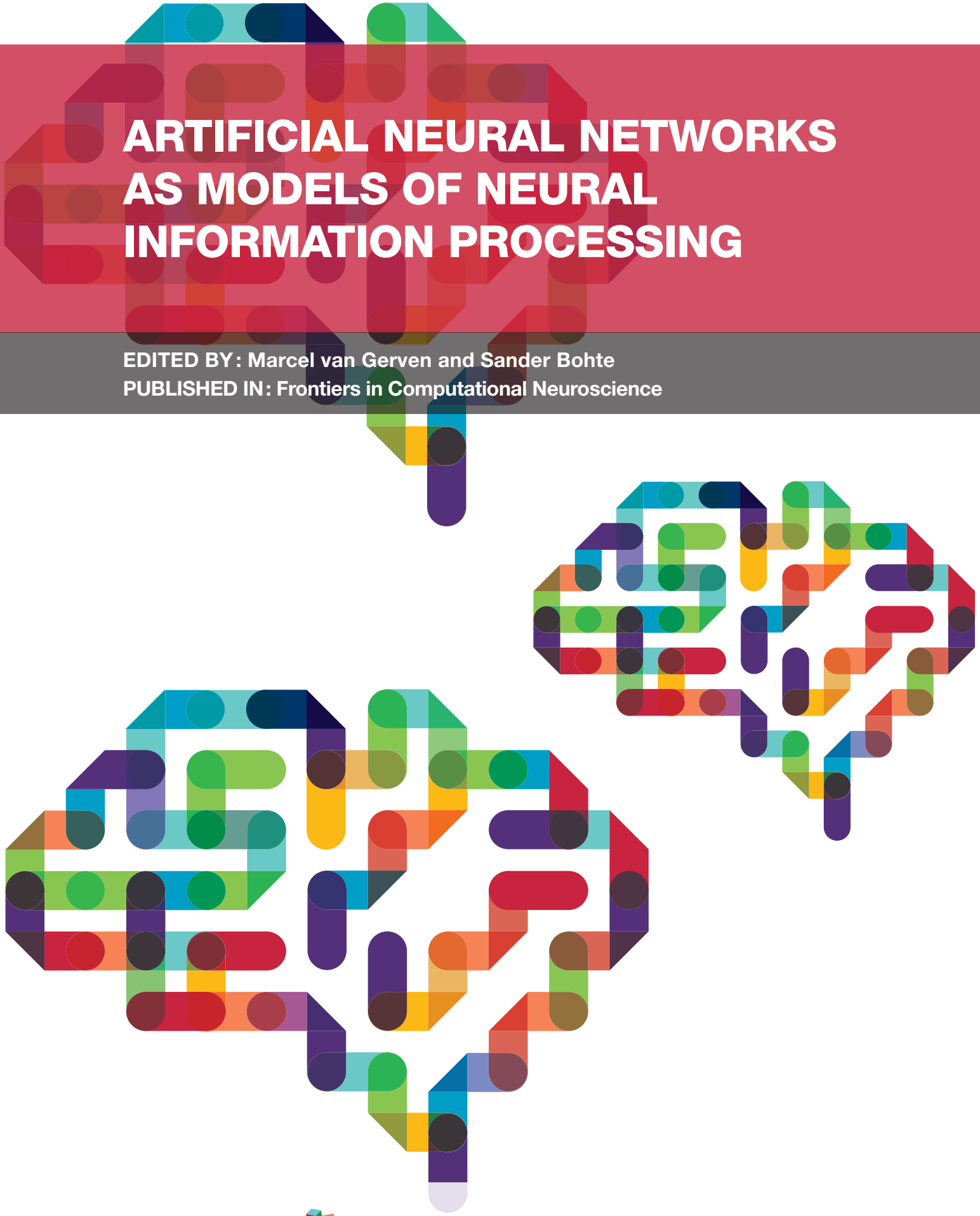
ARTIFICIAL NEURAL NETWORKS AS MODELS OF NEURAL INFORMATION PROCESSING

EDITED BY: Marcel van Gerven and Sander Bohte

PUBLISHED IN: Frontiers in Computational Neuroscience



frontiers Research Topics

frontiers

ISSN 1664-8714
ISBN 978-2-88945-401-3
DOI 10.3389/978-2-88945-401-3

About Frontiers

Frontiers is more than just an open-access publisher of scholarly articles: it is a pioneering approach to the world of academia, radically improving the way scholarly research is managed. The grand vision of Frontiers is a world where all people have an equal opportunity to seek, share and generate knowledge. Frontiers provides immediate and permanent online open access to all its publications, but this alone is not enough to realize our grand goals.

Frontiers Journal Series

The Frontiers Journal Series is a multi-tier and interdisciplinary set of open-access, online journals, promising a paradigm shift from the current review, selection and dissemination processes in academic publishing. All Frontiers journals are driven by researchers for researchers; therefore, they constitute a service to the scholarly community. At the same time, the Frontiers Journal Series operates on a revolutionary invention, the tiered publishing system, initially addressing specific communities of scholars, and gradually climbing up to broader public understanding, thus serving the interests of the lay society, too.

Dedication to Quality

Each Frontiers article is a landmark of the highest quality, thanks to genuinely collaborative interactions between authors and review editors, who include some of the world's best academicians. Research must be certified by peers before entering a stream of knowledge that may eventually reach the public - and shape society; therefore, Frontiers only applies the most rigorous and unbiased reviews.
Frontiers revolutionizes research publishing by freely delivering the most outstanding research, evaluated with no bias from both the academic and social point of view.
By applying the most advanced information technologies, Frontiers is catapulting scholarly publishing into a new generation.

What are Frontiers Research Topics?

Frontiers Research Topics are very popular trademarks of the Frontiers Journals Series: they are collections of at least ten articles, all centered on a particular subject. With their unique mix of varied contributions from Original Research to Review Articles, Frontiers Research Topics unify the most influential researchers, the latest key findings and historical advances in a hot research area! Find out more on how to host your own Frontiers Research Topic or contribute to one as an author by contacting the Frontiers Editorial Office: **researchtopics@frontiersin.org**

ARTIFICIAL NEURAL NETWORKS AS MODELS OF NEURAL INFORMATION PROCESSING

Topic Editors:
Marcel van Gerven, Radboud Universiteit Nijmegen, Netherlands
Sander Bohte, Centrum Wiskunde & Informatica, Netherlands

Modern neural networks gave rise to major breakthroughs in several research areas. In neuroscience, we are witnessing a reappraisal of neural network theory and its relevance for understanding information processing in biological systems. The research presented in this book provides various perspectives on the use of artificial neural networks as models of neural information processing. We consider the biological plausibility of neural networks, performance improvements, spiking neural networks and the use of neural networks for understanding brain function.

Citation: van Gerven, M., Bohte, S., eds. (2018). Artificial Neural Networks as Models of Neural Information Processing. Lausanne: Frontiers Media. doi: 10.3389/978-2-88945-401-3

Table of Contents

UNDERSTANDING BRAIN FUNCTION

frontiers in Computational Neuroscience

EDITORIAL
published: 19 December 2017
doi: 10.3389/fncom.2017.00114

Editorial: Artificial Neural Networks as Models of Neural Information Processing

Marcel van Gerven[1]* *and Sander Bohte*[2]

[1] Department of Artificial Intelligence, Donders Institute for Brain, Cognition and Behaviour, Radboud University Nijmegen, Nijmegen, Netherlands, [2] Department of Machine Learning, Centrum Wiskunde and Informatica, Amsterdam, Netherlands

Keywords: neural networks, artificial intelligence, computational neuroscience, rate coding, spiking neural networks

Editorial on the Research Topic

Artificial Neural Networks as Models of Neural Information Processing

INTRODUCTION

In artificial intelligence (AI), new advances make it possible that artificial neural networks (ANNs) learn to solve complex problems in a reasonable amount of time (LeCun et al., 2015). To the computational neuroscientist, ANNs are theoretical vehicles that aid in the understanding of neural information processing (van Gerven). These networks can take the form of the rate-based models that are used in AI or more biologically plausible models that make use of spiking neurons (Brette, 2015). The objective of this special issue is to explore the use of ANNs in the context of computational neuroscience from various perspectives.

BIOLOGICAL PLAUSIBILITY

Biological plausibility is an important topic in neural networks research. That is, are ANNs simply convenient computational models or do they also inform about the computations that take place in our own brains?

Marblestone et al. carefully lay out the rapid advances in deep learning and contrast these developments with current practice and views in neuroscience. Their main insight is that biological learning may be driven by the optimization of cost functions using successive neural network layers.

A classic question that has haunted ANNs for years is whether backpropagation is biologically plausible (Crick, 1989). Scellier and Bengio introduce Equilibrium Propagation as a new learning framework for energy-based models. The algorithm computes the gradient of an objective function without relying on separate circuits for error propagation that integrate non-local signals.

While acetylcholine (Ach) and dopamine (DA) are neuromodulators that are known to have profound and lasting effects on the neural responses to stimuli, it is unknown what their respective functional roles are. Holca-lamarre et al. develop a neural network model that is combined with the physiological release schedules of ACh and DA.

OPEN ACCESS

Edited and reviewed by:
Paul Miller,
Brandeis University, United States

***Correspondence:**
Marcel van Gerven
m.vangerven@donders.ru.nl

Received: *24 November 2017*
Accepted: *12 December 2017*
Published: *19 December 2017*

Citation:
van Gerven M and Bohte S (2017) Editorial: Artificial Neural Networks as Models of Neural Information Processing. Front. Comput. Neurosci. 11:114. doi: 10.3389/fncom.2017.00114

IMPROVING PERFORMANCE

Several papers propose new mechanisms to improve the perfomance of ANNs.

Li et al. investigate chunking, which is a phenomenon referring to the grouping of items when performing a memory task, leading to improvements in task performance. The authors show that chunking can have computational benefits as it allows the use of synapses with narrow dynamic range and low precision when performing a memory task.

An important limitation of Hopfield networks is their limited storage capacity. Folli et al. show that by allowing non-zero diagonal elements on the weight matrix, maximal storage capacity is obtained when the number of stored memory patterns exceeds the network size.

McClure and Kriegeskorte introduce representational distance learning (RDL) as a stochastic gradient descent method that drives the representational space of a student model to approximate the representational space of a teacher model.

SPIKING NEURAL NETWORKS

An important endeavor in computational neuroscience is to further our understanding of biological and artificial spiking neural networks.

How sensory stimuli relate to the activity of neurons is one of the big open questions in neuroscience, and determining this relationship between the input a neuron receives and the outgoing spike-train has remained a challenge. Zeldenrust et al. propose a new ANN-based method to measure *in vitro* how much information a neuron transfers in this process.

The rate with which spikes are emitted is often mapped to the analog activation values of artificial neurons, but it is well-known that this relationship captures only part of the information processing in real neurons. Carrillo-medina and Latorre develop networks of spiking neurons that operate based on the principles developed for so-called signature neural networks.

How does the central nervous system develop the hierarchy of sensory maps that reflect different internal or external patterns and/or states? Chen shows how simple recurrent and reentrant neuronal networks can discriminate different inputs and generate sensory maps.

Understanding Brain Function

ANNs have also been embraced as a new tool for understanding neural information processing in the brain. In this special issue, a number of advances in this area are put forward.

One question is whether supervised or unsupervised neural networks provide better explanations of neural information processing. Testolin et al. taught neural networks to learn an explicit mapping between different spatial reference frames. They show that both network architecture and the employed learning paradigm affect neural coding properties.

An elusive property of our own brains is that we engage in dreaming during sleep. Horikawa and Kamitani used deep neural networks in an effort to decode what people dream about. They found that decoded features from dream fMRI data positively correlated with those associated with the object categories that related to the dream content.

An important question in neuroscience is how neural representations to sensory input are functionally organized. Güçlü and van Gerven show that neural responses to sensory input can be modeled using recurrent neural networks that can be trained end-to-end.

CONCLUSION

Neural networks are experiencing a revival that not only transforms AI but also provides new insights about neural computation in biological systems. The contributions in this special issue describe new advances in neural networks that increase their efficacy or plausibility from a biological point of view. A closer interaction between the AI and neuroscience communities is expected to lead to various other theoretical and practical breakthroughs in the years to come.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct and intellectual contribution to the work, and approved it for publication.

REFERENCES

Brette, R. (2015). Philosophy of the spike: Rate-based vs spike-based theories of the brain. *Front. Syst. Neurosci.* 9:151. doi: 10.3389/fnsys.2015.00151

Crick, F. (1989). The recent excitement about neural networks. *Nature* 337, 129–132.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature* 521, 436–444. doi: 10.1038/nature14539

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Computational Neuroscience

REVIEW
published: 07 December 2017
doi: 10.3389/fncom.2017.00112

Computational Foundations of Natural Intelligence

*Marcel van Gerven**

Computational Cognitive Neuroscience Lab, Department of Artificial Intelligence, Donders Institute for Brain, Cognition and Behaviour, Radboud University, Nijmegen, Netherlands

New developments in AI and neuroscience are revitalizing the quest to understanding natural intelligence, offering insight about how to equip machines with human-like capabilities. This paper reviews some of the computational principles relevant for understanding natural intelligence and, ultimately, achieving strong AI. After reviewing basic principles, a variety of computational modeling approaches is discussed. Subsequently, I concentrate on the use of artificial neural networks as a framework for modeling cognitive processes. This paper ends by outlining some of the challenges that remain to fulfill the promise of machines that show human-like intelligence.

Keywords: natural intelligence, strong AI, cognition, artificial neural networks, machine learning

1. INTRODUCTION

Understanding how mind emerges from matter is one of the great remaining questions in science. How is it possible that organized clumps of matter such as our own brains give rise to all of our beliefs, desires and intentions, ultimately allowing us to contemplate ourselves as well as the universe from which we originate? This question has occupied cognitive scientists who study the computational basis of the mind for decades. It also occupies other breeds of scientists. For example, ethologists and psychologists focus on the complex behavior exhibited by animals and humans whereas cognitive, computational and systems neuroscientists wish to understand the mechanistic basis of processes that give rise to such behavior.

The ambition to understand natural intelligence as encountered in biological organisms can be contrasted with the motivation to build intelligent machines, which is the subject matter of artificial intelligence (AI). Wouldn't it be amazing if we could build synthetic brains that are endowed with the same qualities as their biological cousins? This desire to mimic human-level intelligence by creating artificially intelligent machines has occupied mankind for many centuries. For instance, mechanical men and artificial beings appear in Greek mythology and realistic human automatons had already been developed in Hellenic Egypt (McCorduck, 2004). The engineering of machines that display human-level intelligence is also referred to as strong AI (Searle, 1980) or artificial general intelligence (AGI) (Adams et al., 2012), and was the original motivation that gave rise to the field of AI (Newell, 1991; Nilsson, 2005).

Excitingly, major advances in various fields of research now make it possible to attack the problem of understanding natural intelligence from multiple angles. From a theoretical point of view we have a solid understanding of the computational problems that are solved by our own brains (Dayan and Abbott, 2005). From an empirical point of view, technological breakthroughs allow us to probe and manipulate brain activity in unprecedented ways, generating new neuroscientific insights into brain structure and function (Chang, 2015). From an engineering perspective, we are finally able to build machines that learn to solve complex tasks, approximating and sometimes surpassing human-level performance (Jordan and Mitchell, 2015). Still, these efforts

OPEN ACCESS

Edited by:
Florentin Wörgötter,
University of Göttingen, Germany

Reviewed by:
Sebastian Herzog,
Max Planck Institute for Dynamics and Self Organization (MPG), Germany
Carme Torras,
Consejo Superior de Investigaciones Científicas (CSIC), Spain

***Correspondence:**
Marcel van Gerven
m.vangerven@donders.ru.nl

Received: *01 August 2017*
Accepted: *22 November 2017*
Published: *07 December 2017*

Citation:
van Gerven M (2017) Computational Foundations of Natural Intelligence. Front. Comput. Neurosci. 11:112. doi: 10.3389/fncom.2017.00112

have not yet provided a full understanding of natural intelligence, nor did they give rise to machines whose reasoning capacity parallels the generality and flexibility of cognitive processing in biological organisms.

The core thesis of this paper is that natural intelligence can be better understood by the coming together of multiple complementary scientific disciplines (Gershman et al., 2015). This thesis is referred to as *the great convergence*. The advocated approach is to endow artificial agents with synthetic brains (i.e., cognitive architectures, Sun, 2004) that mimic the thought processes that give rise to ethologically relevant behavior in their biological counterparts. A motivation for this approach is given by Braitenberg's law of uphill analysis and downhill invention, which states that it is much easier to understand a complex system by assembling it from the ground up, rather than by reverse engineering it from observational data (Braitenberg, 1986). These synthetic brains, which can be put to use in virtual or real-world environments, can then be validated against neuro-behavioral data and analyzed using a multitude of theoretical tools. This approach not only elucidates our understanding of human brain function but also paves the way for the development of artificial agents that show truly intelligent behavior (Hassabis et al., 2017).

The aim of this paper is to sketch the outline of a research program which marries the ambitions of neuroscientists to understand natural intelligence and AI researchers to achieve strong AI (**Figure 1**). Before embarking on our quest to build synthetic brains as models of natural intelligence, we need to formalize what problems are solved by biological brains. That is, we first need to understand how adaptive behavior ensues in animals and humans.

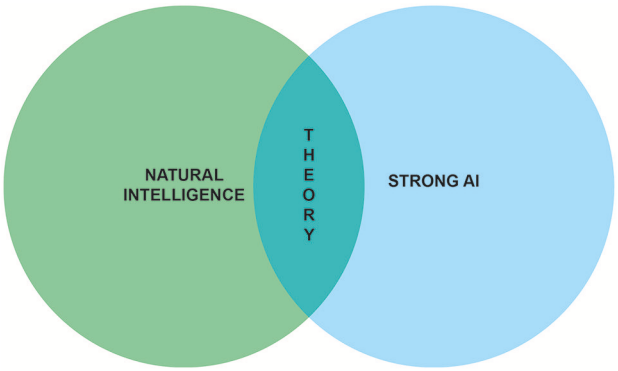


FIGURE 1 | Understanding natural intelligence and achieving strong AI are seen as relying on the same theoretical foundations and require the convergence of multiple scientific and engineering disciplines.

2. ADAPTIVE BEHAVIOR IN BIOLOGICAL AGENTS

Ultimately, organisms owe their existence to the fact that they promote survival of their constituent genes; the basic physical and functional units of heredity that code for an organism (Dawkins, 2016). At evolutionary time scales, organisms developed a range of mechanisms which ensure that they live long enough such as to produce offspring. For example, single-celled protozoans already show rather complex ingestive, defensive and reproductive behavior, which is regulated by molecular signaling (Swanson, 2012; Sterling and Laughlin, 2016).

2.1. Why Do We Need a Brain?

About 3.5 billion years ago, multicellular organisms started to appear. Multicellularity offers several competitive advantages over unicellularity. It allows organisms to increase in size without the limitations set by unicellularity and permits increased complexity by allowing cellular differentiation. It also increases life span since an organism can live beyond the demise of a single cell. At the same time, due to their increased size and complexity, multicellular organisms require more intricate mechanisms for signaling and regulation.

In multicellular organisms, behavior is regulated at multiple scales, ranging from intracellular molecular signaling all the way up to global regulation via the interactions between different organ systems. Hence, the nervous system allows for fast responses via electrochemical signaling and for slow responses by acting on the endocrine system. Nervous systems are found in almost all multicellular animals, but vary greatly in complexity. For example, the nervous system of the nematode roundworm *Caenorhabditis elegans* (*C. elegans*) is made up of 302 neurons and 7,000 synaptic connections (White et al., 1986; Varshney et al., 2011). In contrast, the human brain contains about 20 billion neocortical neurons that are wired together via as many as 0.15 quadrillion synapses (Pakkenberg and Gundersen, 1997; Pakkenberg et al., 2003).

In vertebrates, the nervous system can be partitioned into the central nervous system (CNS), consisting of the brain and the spinal cord, and the peripheral nervous system (PNS), which connects the CNS to every other part of the body. The brain allows for centralized control and efficient information transmission. It can be partitioned into the forebrain, midbrain and hindbrain, each of which contain dedicated neural circuits that allow for integration of information and generation of coordinated activity. The spinal cord connects the brain to the body by allowing sensory and motor information to travel back and forth between the brain and the body. It also coordinates certain reflexes that bypass the brain altogether.

The interplay between the nervous system, the body and the environment is nicely captured by Swanson's four system model of nervous system organization (Swanson, 2000), as shown in **Figure 2**. Briefly, the brain exerts centralized control on the body by sending commands to the motor system based on information received via the sensory system. It exerts this control by way of the cognitive system, which drives voluntary initiation of behavior, as well as the state system, which refers to the intrinsic activity that controls global behavioral state. The motor system can also be influenced directly by the sensory system via spinal cord reflexes. Output of the motor system induces visceral responses that affect bodily state as well as somatic responses that act on the environment. It is also able to drive the secretion of

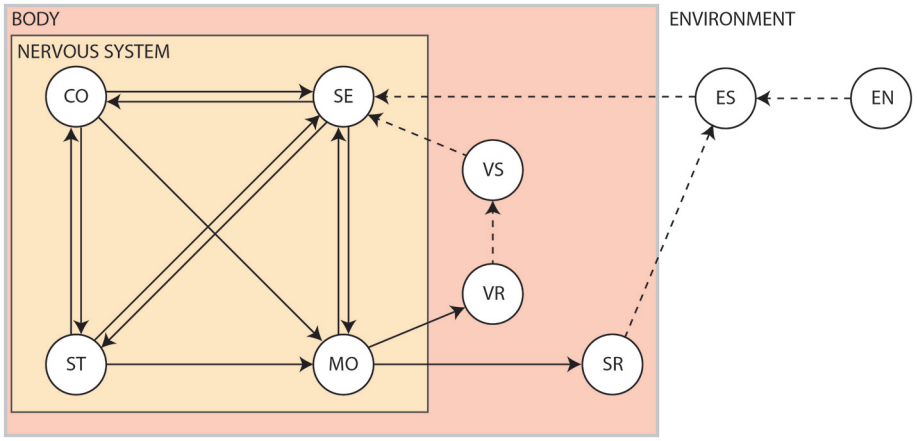


FIGURE 2 | The four system model of nervous system organization. CO, Cognitive system; EN, Environment; ES, Environmental stimuli; MO, Motor system; SE, Sensory system; SR, Somatic responses; ST, Behavioral state system; VR, Visceral responses; VS, Visceral stimuli. Solid arrows show influences pertaining to the nervous system. Dashed arrows show interactions produced by the body or the environment[1].

hormones that act more globally on the body. Both the body and the environment generate sensations that are processed by the sensory system. This closed-loop system, tightly coupling sensation, thought and action, is known as the *perception-action cycle* (Dewey, 1896; Sperry, 1952; Fuster, 2004).

Summarizing, the brain, together with the spinal cord and the peripheral nervous system, can be seen as an organ that exploits sensory input such as to generate adaptive behavior through motor outputs. This ensures an organism's long-term survival in a world that is dominated by uncertainty, as a result of partial observability, noise and stochasticity. The upshot of this interpretation is that action, which drives the generation of adaptive behavior, is the ultimate reason why we have a brain in the first place. Citing Sperry (1952): "the entire output of our thinking machine consists of nothing but patterns of motor coordination." To understand how adaptive behavior ensues, we therefore need to identify the ultimate causes that determine an agent's actions (Tolman, 1932).

2.2. What Makes us Tick?

In biology, ultimately, all evolved traits must be connected to an organism's survival. This implies that, from the standpoint of evolutionary psychology, natural selection favors those behaviors and thought processes that provide the organism with a selective advantage under ecological pressure (Barkow et al., 1992). Since causal links between behavior and long-term survival cannot be sensed or controlled directly, an agent needs to rely on other, directly accessible, ways to promote its survival. This can take the form of (1) evolving optimal sensors and effectors that allow it to maximize its control given finite resources and (2) evolving a behavioral repertoire that maximizes the information gained from the environment and generates optimal actions based on available sensory information.

[1]Figure modified from http://larrywswanson.com/?page_id=1523 with permission.

In practice, behavior is the result of multiple competing needs that together provide an evolutionary advantage. These needs arise because they provide particular rewards to the organism. We distinguish *primary rewards*, *intrinsic rewards* and *extrinsic rewards*.

Primary Rewards

Primary rewards are those necessary for the survival of one's self and offspring, which includes homeostatic and reproductive rewards. Here, homeostasis refers to the maintenance of optimal settings of various biological parameters (e.g., temperature regulation) (Cannon, 1929). A slightly more sophisticated concept is *allostasis*, which refers to the predictive regulation of biological parameters in order to prevent deviations rather than correcting them *post hoc* (Sterling, 2012). An organism can use its nervous system (muscle signaling) or endocrine system (endocrine signaling) to globally control or adjust the activities of many systems simultaneously. This allows for visceral responses that ensure proper functioning of an agent's internal organs as well as basic drives such as ingestion, defense and reproduction that help ensure an agent's survival (Tinbergen, 1951).

Intrinsic Rewards

Intrinsic rewards are unconditioned rewards that are attractive and motivate behavior because they are inherently pleasurable (e.g., the experience of joy). The phenomenon of intrinsic motivation was first identified in studies of animals engaging in exploratory, playful and curiosity-driven behavior in the absence of external rewards or punishments (White, 1959).

Extrinsic Rewards

Extrinsic rewards are conditioned rewards that motivate behavior but are not inherently pleasurable (e.g., praise or monetary reward). They acquire their value through learned association with intrinsic rewards. Hence, extrinsic motivation refers to our tendency to perform activities for known external rewards,

whether they be tangible or psychological in nature (Brown, 2007).

Summarizing, the continual competition between multiple drives and incentives that have adaptive value to the organism and are realized by dedicated neural circuits is what ultimately generates behavior (Davies et al., 2012). In humans, the evolutionary and cultural pressures that shaped our own intrinsic and extrinsic motivations have allowed us to reach great achievements, ranging from our mastery of the laws of nature to expressions of great beauty as encountered in the liberal arts. The question remains how we can gain an understanding of how our brains generate the rich behavioral repertoire that can be observed in nature.

3. UNDERSTANDING NATURAL INTELLIGENCE

In a way, the recipe for understanding natural intelligence and achieving strong AI is simple. If we can construct synthetic brains that mimic the adaptive behavior displayed by biological brains in all its splendor then our mission has succeeded. This entails equipping synthetic brains with the same special purpose computing machinery encountered in real brains, solving those problems an agent may be faced with. In practice, of course, this is easier said than done given the incomplete state of our knowledge and the daunting complexity of biological systems.

3.1. Levels of Analysis

The neural circuits that make up the human brain can be seen as special-purpose devices that together guarantee the selection of (near-)optimal actions. David Marr in particular advocated the view that the nervous system should be understood as a collection of information processing systems that solve particular problems an organism is faced with (Marr, 1982). His work gave rise to the field of computational neuroscience and has been highly influential in shaping ideas about neural information processing (Willshaw et al., 2015). Marr and Poggio (1976) proposed that an understanding of information processing systems should take place at distinct levels of analysis, namely the *computational level*, which specifies what problem the system solves, the *algorithmic level*, which specifies how the system solves the problem, and the *implementational level*, which specifies how the system is physically realized.

A canonical example of a three-level analysis is prey localization in the barn owl (Grothe, 2003). At the computational level, the owl needs to use auditory information to localize its prey. At the algorithmic level, this can be implemented by circuits composed of delay lines and coincidence detectors that detect inter-aural time differences (Jeffress, 1948). At the implementational level, neurons in the nucleus laminaris have been shown to act as coincidence detectors (Carr and Konishi, 1990).

Marr's levels of analysis sidestep one important point, namely how a system gains the ability to solve a computational problem in the first place. That is, it is also crucial to understand how an organism (or species as a whole) is able to learn and evolve the computations and representations that allow it to survive in the natural world (Poggio, 2012). Learning itself takes place at the level of the individual organism as well as of the species. In the individual, one can observe lasting changes in the brain throughout its lifetime, which is referred to as neural plasticity. At the species level, natural selection is responsible for evolving the mechanisms that are involved in neural plasticity (Poggio, 2012). As argued by Poggio, an understanding at the level of learning in the individual and the species is sufficiently powerful to solve a problem and can thereby act as an explanation of natural intelligence. To illustrate the relevance of this revised model, in the prey localization example it would be imperative to understand how owls are able to adapt to changes in their environment (Huo and Murray, 2009), as well as how owls were equipped with such machinery during evolution.

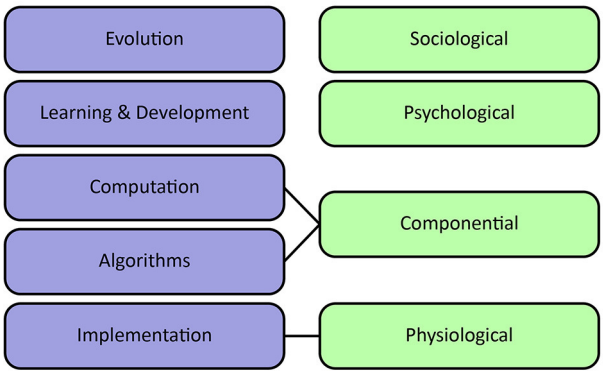


FIGURE 3 | Levels of analysis. Left column shows Poggio's extension of Marr's levels of analysis, emphasizing learning at various timescales. Right column shows Sun's levels of analysis, emphasizing individual beliefs and socio-cultural processes.

Sun et al. (2005) propose an alternative organization of levels of cognitive modeling. They distinguish sociological, psychological, componential and physiological levels. The sociological level refers to the collective behavior of agents, including interactions between agents as well as their environment. It stresses the importance of socio-cultural processes in shaping cognition. The psychological level covers individual behaviors, beliefs, concepts, and skills. The componential level describes inter-agent processes specified in terms of Marr's computational and algorithmic levels. Finally, the physiological level describes the biological substrate which underlies the generation of adaptive behavior, corresponding to Marr's implementational level. It can provide valuable input about important computations and plausible architectures at a higher level of abstraction.

Figure 3 visualizes the different interpretations of levels of analysis. Without committing to a definitive stance on levels of analysis, all described levels provide important complementary perspectives concerning the modeling and understanding of natural intelligence.

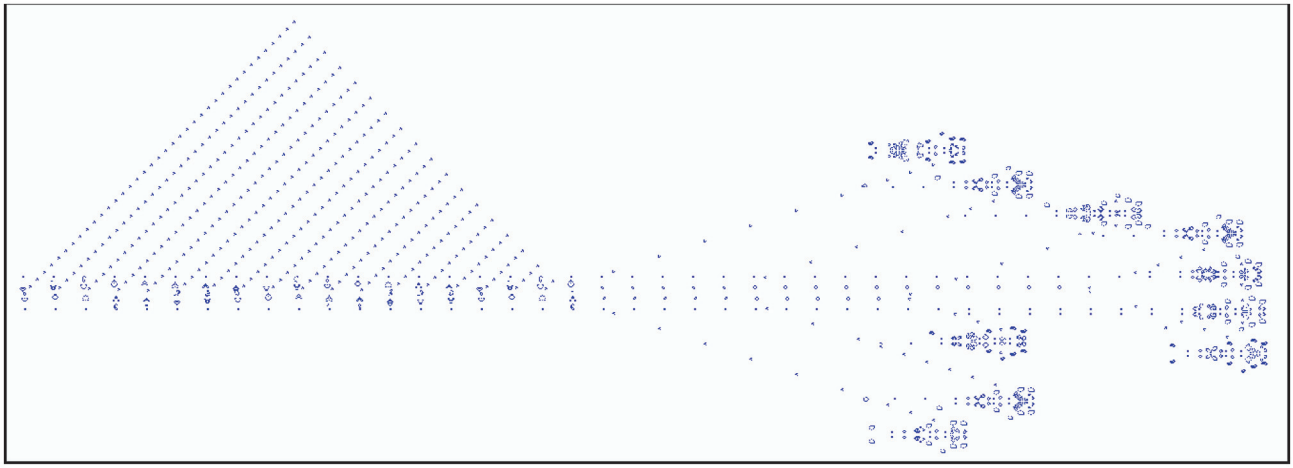


FIGURE 4 | Example of the Game of Life, where each cell state evolves according to a set of deterministic rules that depend on the states of neighboring cells. Depicted is a *breeder* pattern that moves across the universe (here from left to right), leaving behind debris. The breeder produces *Gosper guns* which periodically emit *gliders*; the small patterns that together form the triangular shape on the left-hand side.

3.2. Modeling approaches

The previous section suggests that different approaches to understanding natural intelligence and developing cognitive architectures can be taken depending on the levels of analysis one considers. We briefly review a number of core approaches.

Artificial Life

Artificial life is a broad area of research encompassing various different modeling strategies which all have in common that they aim to explain the emergence of life and, ultimately, cognition in a bottom-up manner (Steels, 1993; Bedau, 2003).

A canonical example of an artificial life system is the cellular automaton, first introduced by von Neumann (1966) as an approach to understand the fundamental properties of living systems. Cellular automata operate within a universe consisting of cells, whose states change over multiple generations based on simple local rules. They have been shown to be capable of acting as universal Turing machines, thereby giving them the capacity to compute any fixed partial computable function (Wolfram, 2002).

A famous example of a cellular automaton is Conway's Game of Life. Here, every cell can assume an "alive" or a "dead" state. State changes are determined by its interactions with its eight direct neighbors. At each time step, a live cell with fewer than two or more than three live neighbors dies and a dead cell with exactly three live neighbors will become alive. **Figure 4** shows an example of a breeder pattern which produces Gosper guns in the Game of Life. Gosper guns have been used to prove that the game of life is Turing complete (Gardner, 2001). SmoothLife (Rafler, 2011), as a continuous-space extension of the Game of Life, shows emerging structures that bear some superficial resemblance to biological structures.

In principle, by virtue of their universality, cellular automata offer the capacity to explain how self-replicating adaptive (autopoeietic, Maturana and Varela, 1980) systems emerge from basic rules. This bottom-up approach is also taken by physicists who aim to explain life and, ultimately, cognition purely from thermodynamic principles (Dewar, 2003, 2005; Grinstein and Linsker, 2007; Wissner-Gross and Freer, 2013; Perunov et al., 2014; Fry, 2017).

Biophysical Modeling

A more direct way to model natural intelligence is to presuppose the existence of the building blocks of life which can be used to create realistic simulations of organisms *in silico*. The reasoning is that biophysically realistic models can eventually mimic the information processing capabilities of biological systems. An example thereof is the OpenWorm project which has as its ambition to understand how the behavior of *C. elegans* emerges from its underlying physiology purely via bottom-up biophysical modeling (Szigeti et al., 2014) (**Figure 5A**). It also acknowledges the importance of including not only a model of the worm's nervous system but also of its body and environment in the simulation. That is, adaptive behavior depends on the organism being both embodied and embedded in the world (Anderson, 2003). If successful, then this project would constitute the first example of a digital organism.

It is a long stretch from the worm's 302 neurons to the 86 billion neurons that comprise the human brain (Herculano-Houzel and Lent, 2005). Still, researchers have set out to develop large-scale models of the human brain. Biophysical modeling can be used to create detailed models of neurons and their processes using coupled systems of differential equations. For example, action potential generation can be described in terms of the Hodgkin-Huxley equations (**Figure 5B**) and the flow of electric current along neuronal fibers can be modeled using cable theory (Dayan and Abbott, 2005). This approach is used in the Blue Brain project (Markram, 2006) and its successor, the Human Brain Project (HBP) (Amunts et al., 2016). See de Garis et al. (2010) for a review of various artificial brain projects.

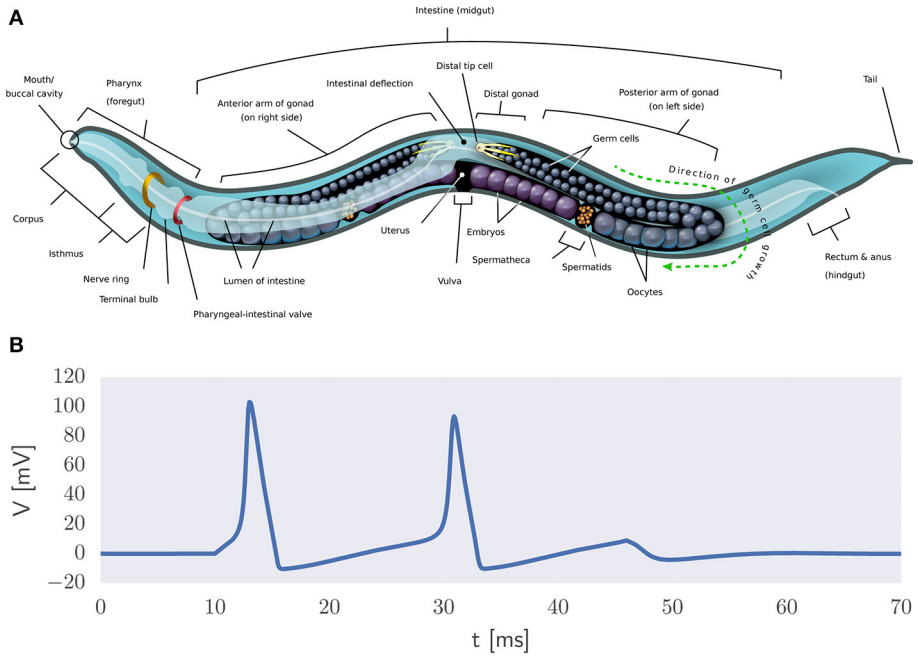


FIGURE 5 | Biophysical modeling. **(A)** Body plan of *C. elegans*[2]. The OpenWorm project aims to provide an accurate bottom-up simulation of the worm acting in its environment. **(B)** Example of action potential generation via the Hodgkin-Huxley equations in the presence of a constant input current.

Connectionism

Connectionism refers to the explanation of cognition as arising from the interplay between basic (sub-symbolic) processing elements (Smolensky, 1987; Bechtel, 1993). It has close links to cybernetics, which focuses on the development of control structures from which intelligent behavior emerges (Rid, 2016).

Connectionism came to be equated with the use of artificial neural networks that abstract away from the details of biological neural networks. An artificial neural network (ANN) is a computational model which is loosely inspired by the human brain as it consists of an interconnected network of simple processing units (artificial neurons) that learns from experience by modifying its connections. Alan Turing was one of the first to propose the construction of computing machinery out of trainable networks consisting of neuron-like elements (Copeland and Proudfoot, 1996). Marvin Minsky, one of the founding fathers of AI, is credited for building the first trainable ANN, called SNARC, out of tubes, motors, and clutches (Seising, 2017).

Artificial neurons can be considered abstractions of (populations of) neurons while the connections are taken to be abstractions of modifiable synaptic connections (**Figure 6**). The behavior of an artificial neuron is fully determined by the connection strengths as well as how input is transformed into output. Contrary to detailed biophysical models, ANNs make use of basic matrix operations and nonlinear transformations as their fundamental operations. In its most basic incarnation, an artificial neuron simply transforms its input $\mathbf{x}$ into a response y through an activation function f, as shown in **Figure 6**. The activation function operates on an input activation which is typically taken to be the inner product between the input $\mathbf{x}$ and the parameters (weight vector) $\mathbf{w}$ of the artificial neuron. The weights are interpreted as synaptic strengths that determine how presynaptic input is translated into postsynaptic firing rate. This yields a simple linear-nonlinear mapping of the form

$$y = f(\mathbf{w}^{\mathsf{T}}\mathbf{x}) . \tag{1}$$

By connecting together multiple neurons, one obtains a neural network that implements some non-linear function $\mathbf{y} = \mathbf{f}(\mathbf{x}; \boldsymbol{\theta})$, where the f_i are nonlinear transformations and $\boldsymbol{\theta}$ stands for the network parameters (i.e., weight vectors). After training a neural network, representations become encoded in a distributed manner as a pattern which manifests itself across all its neurons (Hinton et al., 1986).

Throughout the course of their history ANNs have fallen in and out of favor multiple times. At the same time, each next generation of neural networks has yielded new insights about how complex behavior may emerge through the collective action of simple processing elements. Modern neural networks perform so well on several benchmark problems that they obliterate all competition in, e.g., object recognition (Krizhevsky et al., 2012), natural language processing (Sutskever et al., 2014), game playing (Mnih et al., 2015; Silver et al., 2017) and robotics (Levine et al., 2015), often matching and sometimes surpassing human-level performance (LeCun et al., 2015). Their success relies on combining classical ideas (Widrow and Lehr, 1990; Hochreiter

[2]Figure by K. D. Schroeder, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=26958836. Used with permission.

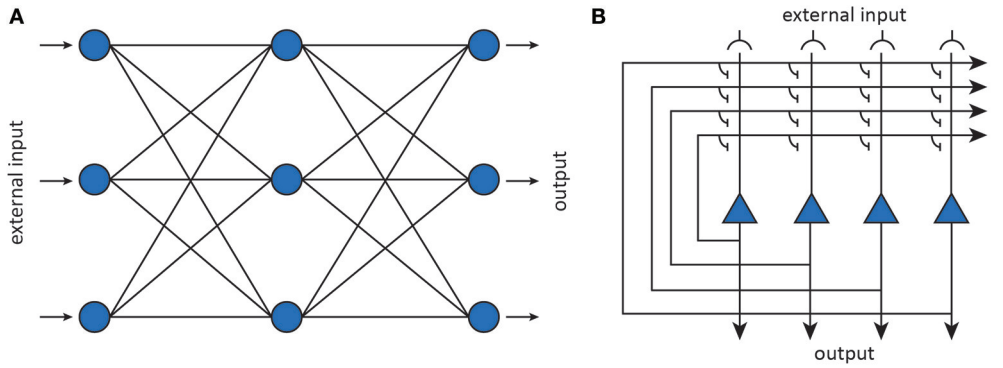


FIGURE 6 | Artificial neural networks (Yuste, 2015). **(A)** Feedforward neural networks map inputs to outputs using nonlinear transformations. **(B)** Recurrent neural networks implement dynamical systems by feeding back output activity to the input layer, where it is combined with external input.

and Schmidhuber, 1997; LeCun et al., 1998) with new algorithmic developments (Hinton et al., 2006; Srivastava et al., 2014; He et al., 2015; Ioffe and Szegedy, 2015; Zagoruyko and Komodakis, 2017), while using high-performance graphical processing units (GPUs) to massively speed up training of ANNs on big datasets (Raina et al., 2009).

Cognitivism

A conceptually different approach to the explanation of cognition as emerging from bottom-up principles is the view that cognition should be understood in terms of formal symbol manipulation. This computationalist view is associated with the cognitivist program which arose in response to earlier behaviorist theories. It embraces the notion that, in order to understand natural intelligence, one should study internal mental processes rather than just externally observable events. That is, cognitivism asserts that cognition should be defined in terms of formal symbol manipulation, where reasoning involves the manipulation of symbolic representations that refer to information about the world as acquired by perception.

This view is formalized by the physical symbol system hypothesis (Newell and Simon, 1976), which states that "a physical symbol system has the necessary and sufficient means for intelligent action." This hypothesis implies that artificial agents, when equipped with the appropriate symbol manipulation algorithms, will be capable of displaying intelligent behavior. As Newell and Simon (1976) wrote, the physical symbol system hypothesis also implies that "the symbolic behavior of man arises because he has the characteristics of a physical symbol system." This also suggests that the specifics of our nervous system are not relevant for explaining adaptive behavior (Simon, 1996).

Cognitivism gave rise to cognitive science as well as artificial intelligence, and spawned various cognitive architectures such as ACT-R (Anderson et al., 2004) (see **Figure 7**) and SOAR (Laird, 2012) that employ rule-based approaches in the search for a unified theory of cognition (Newell, 1991).[3]

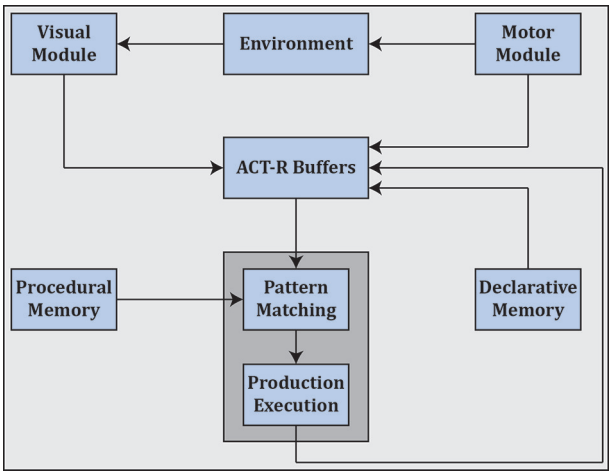


FIGURE 7 | ACT-R as an example cognitive architecture which employs symbolic reasoning. ACT-R interfaces with different modules through buffers. Cognition unfolds as a succession of activations of production rules as mediated by pattern matching and execution[4].

Probabilistic Modeling

Modern cognitive science still embraces the cognitivist program but has since taken a probabilistic approach to the modeling of cognition. As stated by Griffiths et al. (2010), this probabilistic approach starts from the notion that the challenges faced by the mind are often of an inductive nature, where the observed data are not sufficient to unambiguously identify the process that generated them. This precludes the use of approaches that are founded on mathematical logic and requires a quantification of the state of the world in terms of degrees of belief as afforded by probability theory (Jaynes, 1988). The probabilistic approach operates by identifying a hypothesis space representing solutions to the inductive problem. It then prescribes how an agent should revise her belief in the hypotheses given the information provided

[3]In fact, ACT-R also uses some subsymbolic elements and can therefore be considered a *hybrid* architecture.

[4]Figure modified from http://act-r.psy.cmu.edu/about with permission.

by observed data. Hypotheses are typically formulated in terms of probabilistic graphical models that capture the independence structure between random variables of interest (Koller and Friedman, 2009). An example of such a graphical model is shown in **Figure 8**.

Belief updating in the probabilistic sense is realized by solving a statistical inference problem. Consider a set of of hypotheses $\mathcal{H}$ that might explain the observed data. Let $p(h)$ denote our belief in a hypothesis $h \in \mathcal{H}$, reflecting the state of the world, before observing any data (known as the *prior*). Let $p(\mathbf{x} \mid h)$ indicate the probability of observing data $\mathbf{x}$ if h were true (known as the *likelihood*). Bayes' rule tells us how to update our belief in a hypothesis after observing data. It states that the *posterior probability* $p(h \mid \mathbf{x})$ assigned to h after observing $\mathbf{x}$ should be

$$p(h \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid h)p(h)}{\sum_{h \in \mathcal{H}} p(\mathbf{x} \mid h)p(h)} \tag{2}$$

where the denominator is a normalizing constant known as the *evidence* or *marginal likelihood*[5]. Importantly, it can be shown that degrees of belief are coherent only if they satisfy the axioms of probability theory (Ramsey, 1926).

The beauty of the probabilistic approach lies in its generality. It not only explains how our moment-to-moment percepts change as a function of our prior beliefs and incoming sensory data (Yuille and Kersten, 2006) but also places learning, as the construction of internal models, under the same umbrella by viewing it as an inference problem (MacKay, 2003). In the probabilistic framework, mental processes are modeled using algorithms for approximating the posterior (Koller and Friedman, 2009) and neural processes are seen as mechanisms for implementing these algorithms (Gershman and Beck, 2016).

The probabilistic approach also provides a basis for making optimal decisions under uncertainty. This is realized by extending probability theory with decision theory. According to decision theory, a rational agent ought to select that action which maximizes the expected utility (von Neumann and Morgenstern, 1953). This is known as the maximum expected utility (MEU) principle. In real-life situations, biological (and artificial) agents need to operate under bounded resources, trading off precision for speed and effort when trying to attain their objectives (Gigerenzer and Goldstein, 1996). This implies that MEU calculations may be intractable. Intractability issues have led to the development of algorithms that maximize a more general form of expected utility which incorporates the costs of computation. These algorithms can in turn be adapted so as to select the best approximation strategy in a given situation (Gershman et al., 2015). Hence, at the algorithmic level, it has been postulated that brains use approximate inference algorithms (Andrieu et al., 2003; Blei et al., 2016) such as to produce good enough solutions for fast and frugal decision making.

Summarizing, by appealing to Bayesian statistics and decision theory, while acknowledging the constraints biological agents are faced with, cognitive science arrives at a theory of bounded rationality that agents should adhere to. Importantly, this normative view dictates that organisms must operate as Bayesian inference machines that aim to maximize expected utility. If they do not, then, under weak assumptions, they will perform suboptimally. This would be detrimental from an evolutionary point of view.

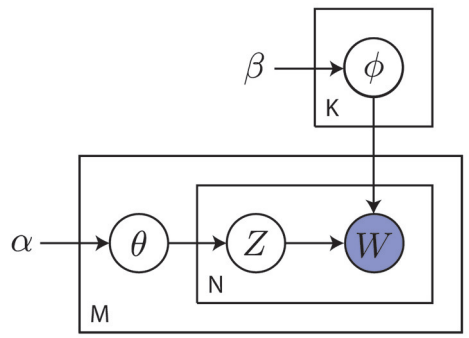


FIGURE 8 | Example of a probabilistic graphical model capturing the statistical relations between random variables of interest. This particular plate model describes a smoothed version of latent Dirichlet allocation as used in topic modeling (Blei et al., 2003). Here, α and β are hyper-parameters, θ_m is the topic distribution for document m, ϕ_k is the word distribution for topic k, z_{nm} is the topic for the n-th word in document m and w_{mn} is a specific word. Capital letters K, M and N denote the number of topics, documents and words, respectively. The goal is to discover abstract topics from observed words. This general approach of inferring posteriors over latent variables from observed data is common to the probabilistic approach.

3.3. Bottom-up Emergence vs. Top-down Abstraction

The aforementioned modeling strategies each provide an alternative approach toward understanding natural intelligence and achieving strong AI. The question arises which of these strategies will be most effective in the long run.

While the strictly bottom-up approach used in artificial life research may lead to fundamental insights about the nature of self-replication and adaptability, in practice it remains an open question how emergent properties that derive from a basic set of rules can reach the same level of organization and complexity as can be found in biological organisms. Furthermore, running such simulations would be extremely costly from a computational point of view.

The same problem presents itself when using detailed biophysical models. That is, bottom-up approaches must either restrict model complexity or run simulations for limited periods of time in order to remain tractable (O'Reilly et al., 2012). Biophysical models additionally suffer from a lack of data. For example, the original aim of the Human Brain Project was to model the human brain within a decade (Markram et al., 2011). This ambition may be hard to realize given the plethora of data required for model estimation. Furthermore, the resulting models may be difficult to link to cognitive function. Izhikevich, reflecting on his simulation of another large biophysically realistic brain model (Izhikevich and Edelman, 2008), states:

[5]Beliefs over continuous quantities can be expressed by replacing summation with integration.

"Indeed, no significant contribution to neuroscience could be made by simulating one second of a model, even if it has the size of the human brain. However, I learned what it takes to simulate such a large-scale system[6]."

Connectionist models, in contrast, abstract away from biophysical details, thereby making it possible to train large-scale models on large amounts of sensory data, allowing cognitively challenging tasks to be solved. Due to their computational simplicity, they are also more amenable to theoretical analysis (Hertz et al., 1991; Bishop, 1995). At the same time, connectionist models have been criticized for their inability to capture symbolic reasoning, their limitations when modeling particular cognitive phenomena, and their abstract nature, restricting their biological plausibility (Dawson and Shamanski, 1994).

Cognitivism has been pivotal in the development of intelligent systems. However, it has also been criticized using the argument that systems which operate via formal symbol manipulation lack intentionality (Searle, 1980)[7]. Moreover, the representational framework that is used is typically constructed by a human designer. While this facilitates model interpretation, at the same time, this programmer-dependence may bias the system, leading to suboptimal solutions. That is, idealized descriptions may induce a semantic gap between perception and possible interpretation (Vernon et al., 2007).

The probabilistic approach to cognition is important given its ability to define normative theories at the computational level. At the same time, it has also been criticized for its treatment of cognition as if it is in the business of selecting some statistical model. Proponents of connectionism argue that computation-level explanations of behavior that ignore mechanisms associated with bottom-up emergence are likely to fall short (McClelland et al., 2010).

The different approaches provide complementary insights into the nature of natural intelligence. Artificial life informs about fundamental bottom-up principles, biophysical models make explicit how cognition is realized via specific mechanisms at the molecular and systems level, connectionist models show how problem solving capacities emerge from the interactions between basic processing elements, cognitivism emphasizes the importance of symbolic reasoning and probabilistic models inform how particular problems could be solved in an optimal manner.

Notwithstanding potential limitations, given their ability to solve complex cognitively challenging problems, connectionist models are taken to provide a promising starting point for understanding natural intelligence and achieving strong AI. They also naturally connect to the different modeling strategies. That is, they connect to artificial life principles by having network architectures emerge through evolutionary strategies (Real et al., 2016; Salimans et al., 2017) and connect to the biophysical level by viewing them as (rate-based) abstractions of biological neural networks (Dayan and Abbott, 2005). They also connect to the computational level by grounding symbolic representations in real-world sensory states (Harnad, 1990) and connect to the probabilistic approach through the observation that emergent computations effectively approximate Bayesian inference (Gal, 2016; Orhan and Ma, 2016; Ambrogioni et al., 2017; Mandt et al., 2017). It is for these reasons that, in the following, we will explore how ANNs, as canonical connectionist models, can be used to promote our understanding of natural intelligence.

[6]From: https://www.izhikevich.org/human_brain_simulation/why.htm

[7]Intentionality or "aboutness" refers to the quality of mental states as being directed toward an object or state of affairs.

4. ANN-BASED MODELING OF COGNITIVE PROCESSES

We will now explore in more detail the ways in which ANNs can be used to understand and model aspects of natural intelligence. We start by addressing how neural networks can learn from data.

4.1. Learning

The capacity of brains to behave adaptively relies on their ability to modify their own behavior based on changing circumstances. The appeal of neural networks stems from their ability to mimic this learning behavior in an efficient manner by updating network parameters $\boldsymbol{\theta}$ based on available data $\mathcal{D} = \{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(N)}\}$, allowing the construction of large models that are able to solve complex cognitive tasks.

Learning proceeds by making changes to the network parameters $\boldsymbol{\theta}$ such that its output starts to agree more and more with the objectives of the agent at hand. This is formalized by assuming the existence of a cost function $\mathcal{J}(\boldsymbol{\theta})$ which measures the degree to which an agent deviates from its objectives. $\mathcal{J}$ is computed by running a neural network in forward mode (from input to output) and comparing the predicted output with the desired output. During its lifetime, the agent obtains data from its environment (sensations) by sampling from a data-generating distribution p_{data}. The goal of an agent is to reduce the expected risk

$$\mathcal{J}^*(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{z}\sim p_{\text{data}}}\left[\ell(\mathbf{z}, \boldsymbol{\theta})\right] \tag{3}$$

where ℓ is the incurred loss per datapoint $\mathbf{z}$. In practice, an agent only has access to a finite number of datapoints which the agent experiences during its lifetime, yielding a training set $\mathcal{D}$. This training set can be represented in the form of an empirical distribution $\hat{p}(\mathbf{z})$ which equals $1/N$ if $\mathbf{z}$ is equal to one of the N examples and zero otherwise. In practice, the aim therefore is to minimize the empirical risk

$$\mathcal{J}(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{z}\sim \hat{p}}\left[\ell(\mathbf{z}, \boldsymbol{\theta})\right] \tag{4}$$

as an approximation of $\mathcal{J}^*$. In reality, the brain is thought to optimize a multitude of cost functions pertaining to the many objectives it aims to achieve in concert (Marblestone et al., 2016).

Risk minimization can be accomplished by making use of a gradient descent procedure. Let $\boldsymbol{\theta}$ be the parameters of a neural network (i.e., the synaptic weights). We can define learning as a search for the optimal parameters $\boldsymbol{\theta}^*$ based on available training data $\mathcal{D}$ such that

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta}) . \tag{5}$$

A convenient way to approximate $\boldsymbol{\theta}^*$ is by measuring locally the change in slope of $\mathcal{J}(\boldsymbol{\theta})$ as a function of $\boldsymbol{\theta}$ and taking a step in the direction of steepest descent. This procedure, known as *gradient descent*, is based on the observation that if $\mathcal{J}$ is defined and differentiable in the neighborhood of a point $\boldsymbol{\theta}$, then $\mathcal{J}$ decreases fastest if one goes from $\boldsymbol{\theta}$ in the direction of the negative gradient $-\nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta})$. In other words, if we use the update rule

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \epsilon \nabla_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta}) \tag{6}$$

with small enough learning rate ϵ then $\boldsymbol{\theta}$ is guaranteed to converge to a (local) minimum of $\mathcal{J}(\boldsymbol{\theta})$[8]. Importantly, the gradient can be computed for arbitrary ANN architectures by running the network in backward mode (from output to input) and computing the gradient using automatic differentiation procedures. This forms the basis of the widely used backpropagation algorithm (Widrow and Lehr, 1990).

One might argue that the backpropagation algorithm fails to connect to learning in biology due to implausible assumptions such as the fact that forward and backward passes use the same set of synaptic weights. There are a number of responses here. First, one might hold the view that backpropagation is just an efficient way to obtain effective network architectures, without committing to the biological plausibility of the learning algorithm *per se*. Second, if biologically plausible learning is the research objective then one is free to exploit other (Hebbian) learning schemes that may reflect biological learning more closely (Miconi, 2017). Finally, researchers have started to put forward arguments that backpropagation may not be that biologically implausible after all (Roelfsema and van Ooyen, 2005; Lillicrap et al., 2016; Scellier and Bengio, 2017).

4.2. Perceiving

One of the core skills any intelligent agent should possess is the ability to recognize patterns in its environment. The world around us consists of various objects that may carry significance. Being able to recognize edible food, places that provide shelter, and other agents will all aid survival.

Biological agents are faced with the problem that they need to be able to recognize objects from raw sensory input (vectors in $\mathbb{R}^n$). How can a brain use the incident sensory input to learn to recognize those things that are of relevance to the organism? Recall the artificial neuron formulation $y = f(\mathbf{w}^\mathsf{T}\mathbf{x})$. By learning proper weights $\mathbf{w}$, this neuron can learn to distinguish different object categories. This is essentially equivalent to a classical model known as the perceptron (Rosenblatt, 1958), which was used to solve simple pattern recognition problems via a simple error-correction mechanism. It also corresponds to a basic linear-nonlinear (LN) model which has been used extensively to model and estimate the receptive field of a neuron or a population of neurons (van Gerven, 2017).

[8]In practice, it is more efficient to iterate over subsets of datapoints, known as mini-batches, in sequence. That is, training is organized in terms of epochs in which all datapoints are processed by iterating over mini-batches. Note that, whenever we are not processing all data points in parallel, we are not exactly following the gradient. Therefore, any such procedure is known as *stochastic gradient descent*.

Single-layer ANNs such as the perceptron are capable of solving interesting learning problems. At the same time, they are limited in scope since they can only solve linearly separable classification problems (Minsky and Papert, 1969). To overcome the limitations of the perceptron we can extend its capabilities by relaxing the constraint that the inputs are directly coupled to the outputs. A multilayer perceptron (MLP) is a feedforward network which generalizes the standard perceptron by having a hidden layer that resides between the input and the output layers. We can write an MLP with multiple output units as

$$\mathbf{y} = \mathbf{g}\left(\mathbf{W}\mathbf{f}\left(\mathbf{V}\mathbf{x}\right)\right) \tag{7}$$

where $\mathbf{V}$ denotes the hidden layer weights and $\mathbf{W}$ denotes the output layer weights. By introducing a hidden layer, MLPs gain the ability to learn internal representations (Rumelhart et al., 1986). Importantly, an MLP can approximate any continuous function to an arbitrary degree of accuracy, given a sufficiently large but finite number of hidden neurons (Cybenko, 1989; Hornik, 1991).

Complex systems tend to be hierarchical and modular in nature (Simon, 1962). The nervous system itself can be thought of as a hierarchically organized system. This is exemplified by Felleman & van Essen's hierarchical diagram of visual cortex (Felleman and Van Essen, 1991), the proposed hierarchical organization of prefrontal cortex (Badre, 2008), the view of the motor system as a behavioral control column (Swanson, 2000) and the proposition that anterior and posterior cortex reflect hierarchically organized executive and perceptual systems (Fuster, 2001). Representations at the top of these hierarchies correspond to highly abstract statistical invariances that occupy our ecological niche (Quian Quiroga et al., 2005; Barlow, 2009). A hierarchy can be modeled by a deep neural network (DNN) composed of multiple hidden layers (LeCun et al., 2015), written as

$$\begin{aligned}\mathbf{y} &= \mathbf{f}_{L+1}\big(\mathbf{W}_{L+1}\mathbf{f}_L\big(\mathbf{W}_L \cdots \mathbf{f}_1\big(\mathbf{W}_1\mathbf{x}\big)\cdots\big)\\ &= \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})\end{aligned} \tag{8}$$

where $\mathbf{W}_l$ is the weight matrix associated with layer l. Even though an MLP can already approximate any function to an arbitrary degree of precision, it has been shown that many classes of functions can be represented much more compactly using thin and deep neural networks compared to shallow and wide neural networks (Bengio and LeCun, 2007; Bengio, 2009; Le Roux and Bengio, 2010; Delalleau and Bengio, 2011; Mhaskar et al., 2016).

A DNN corresponds to a stack of LN models, generalizing the concept of basic receptive field models. They have been shown to yield human-level performance on object categorization tasks (Krizhevsky et al., 2012). The latest DNN incarnations are even capable of predicting the cognitive states of other agents. One example is the prediction of apparent personality traits from multimodal sensory input (Güçlütürk et al., 2016). Deep architectures have been used extensively in neuroscience to model hierarchical processing (Selfridge, 1959; Fukushima, 1980, 2013; Riesenhuber and Poggio, 1999; Lehky and Tanaka, 2016). Interestingly, it has been shown that the representations

encoded in DNN layers correspond to the representations that are learned by areas that make up the sensory hierarchies of biological agents (Güçlü and van Gerven, 2015, 2017a; Güçlü et al., 2016). Multiple reviews discuss this use of DNNs in sensory neuroscience (Cox and Dean, 2014; Kriegeskorte, 2015; Robinson and Rolls, 2015; Marblestone et al., 2016; Yamins and DiCarlo, 2016; Kietzmann et al., 2017; Peelen and Downing, 2017; van Gerven, 2017; Vanrullen, 2017).

4.3. Remembering

Being able to perceive the environment also implies that agents can store and retrieve past knowledge about objects and events in their surroundings. In the feedforward networks considered in the previous section, this knowledge is encoded in the synaptic weights as a result of learning. Memories of the past can also be stored, however, in moment-to-moment neural activity patterns. This does require the availability of lateral or feedback connections in order to enable recurrent processing (Singer, 2013; Maass, 2016). Recurrent processing can be implemented by a recurrent neural network (RNN) (Jordan, 1987; Elman, 1990), defined by

$$\mathbf{y}_n = \mathbf{f}\left(\mathbf{W}\mathbf{y}_{n-1} + \mathbf{U}\mathbf{x}_n\right) \tag{9}$$

such that the neuronal activity at time n depends on the activity at time $n-1$ as well as instantaneous bottom-up input. RNNs can be interpreted as numerical approximations of differential equations that describe rate-based neural models (Dayan and Abbott, 2005) and have been shown to be universal approximators of dynamical systems (Funahashi and Nakamura, 1993)[9]. Their parameters can be estimated using a variant of backpropagation, referred to as backpropagation through time (Mozer, 1989).

When considering perception, feedforward architectures may seem sufficient. For example, the onset latencies of neurons in monkey inferior-temporal cortex during visual processing are about 100 ms (Thorpe and Fabre-Thorpe, 2001), which means that there is ample time for the transmission of just a few spikes. This suggests that object recognition is largely an automatic feedforward process (Vanrullen, 2007). However, recurrent processing is important in perception as well since it provides the ability to maintain state. This is important in detecting salient features in space and time (Joukes et al., 2014), as well as for integrating evidence in noisy or ambiguous settings (O'Reilly et al., 2013). Moreover, perception is strongly influenced by top-down processes, as mediated by feedback connections (Gilbert and Li, 2013). RNNs have also been used to model working memory (Miconi, 2017) as well as hippocampal function, which is involved in a variety of memory-related processes (Willshaw et al., 2015; Kumaran et al., 2016).

A special kind of RNN is the Hopfield network (Hopfield, 1982), where $\mathbf{W}$ is symmetric and $\mathbf{U} = \mathbf{0}$. Learning in a Hopfield net is based on a Hebbian learning scheme. Hopfield nets are attractor networks that converge to a state that is a local minimum of an energy function. They have been used extensively as models of associative memory (Wills et al., 2005). It has even been postulated that dreaming can be seen as an unlearning process which gets rid of spurious minima in attractor networks, thereby improving their storage capacity (Crick and Mitchison, 1983).

[9] The ability of simple RNNs to integrate information over time remains limited, which led to the introduction of various extensions that perform more favorably in this regard (Hochreiter and Schmidhuber, 1997; Cho et al., 2014; Neil et al., 2016; Wu et al., 2016).

4.4. Acting

As already described, the ability to generate appropriate actions is what ultimately drives behavior. In real-world settings, such actions typically need to be inferred from reward signals r_t provided by the environment. This is the subject matter of reinforcement learning (RL) (Sutton and Barto, 1998). Define a policy $\pi(s,a)$ as the probability of selecting an action a given a state s. Let the return $R = \sum_{t=0}^{\infty} \gamma^t r_{t+1}$ be the total reward accumulated in an episode, with γ a discount factor that downweighs future rewards. The goal in RL is to identify an optimal policy π^* that maximizes the expected return

$$\pi^* = \arg\max_{\pi} \mathbb{E}[R \mid \pi]\,. \tag{10}$$

Reinforcement learning algorithms have been crucial in training neural networks that have the capacity to act. Such networks learn to generate suitable actions purely by observing the rewards entailed by previously generated actions. RL algorithms come in model-free and model-based variants. In the model-free setting, optimal actions are learned purely based on the reward that is gained by performing actions in the past. In the model-based setting, in contrast, an explicit model of the environment is used to predict the consequences of actions that are being executed. Importantly, model-free and model-based reinforcement learning approaches have clear correspondences with habitual and goal-directed learning in neuroscience (Daw, 2012; Buschman et al., 2014).

Various model-free reinforcement learning approaches have been used to develop a variety of neural networks for action generation. For example, Q-learning was used to train networks that play Atari games (Mnih et al., 2015) and policy gradient methods have been used to play board games (Silver et al., 2017) and solve problems in (simulated) robotics (Silver et al., 2014; Schulman et al., 2015), effectively closing the perception-action cycle. Evolutionary strategies are also proving to become an useful approach for solving challenging control problems (Salimans et al., 2017). Similar successes have been achieved using model-based reinforcement learning approaches (Schmidhuber, 2015; Mujika, 2016; Santana and Hotz, 2016).

Another important ingredient required for generating optimal actions is recurrent processing, as described in the previous section. Action generation must depend on the ability to integrate evidence over time since, otherwise, we are guaranteed to act suboptimally. That is, states that are qualitatively different can appear the same to the decision maker, leading to suboptimal policies. Consider for example the sensation of a looming object. The optimal decision depends crucially on whether this object is approaching or receding, which can only be determined by

taking past sensations into account. This phenomenon is known as perceptual aliasing (Whitehead and Ballard, 1991).

A key ability of biological organisms which requires recurrent processing is their ability to navigate in their environment, as mediated by the hippocampal formation (Moser et al., 2015). Recent work shows that particular characteristics of hippocampal place cells, such as stable tuning curves that remap between environments, are recovered by training neural networks on navigation tasks (Kanitscheider and Fiete, 2016). The ability to integrate evidence also allows agents to selectively sample the environment, such as to maximize the amount of information gained. This process, known as active sensing, is crucial for understanding perceptual processing in biology (Yarbus, 1967; Regan and Noë, 2001; Friston et al., 2010; Schroeder et al., 2010; Gordon and Ahissar, 2012). Active sensing, in the form of saccade planning, has been implemented using a variety of recurrent neural network architectures (Larochelle and Hinton, 2010; Gregor et al., 2014; Mnih et al., 2014). RNNs that implement recurrent processing have also been used to model various other action-related processes such as timing (Laje and Buonomano, 2013), sequence generation (Rajan et al., 2015) and motor control (Sussillo et al., 2015).

Recurrent processing and reinforcement learning are also essential in modeling higher-level processes, such as cognitive control as mediated by frontal brain regions (Fuster, 2001; Miller and Cohen, 2001). Examples are models of context-dependent processing (Mante et al., 2013) and perceptual decision-making (Carnevale et al., 2015). In general, RNNs that have been trained using RL on a variety of cognitive tasks have been shown to yield properties that are consistent with phenomena observed in biological neural networks (Song et al., 2016; Miconi, 2017).

4.5. Predicting

Modern theories of human brain function appeal to the idea that the brain can be viewed as a prediction machine, which is in the business of continuously generating top-down predictions that are integrated with bottom-up sensory input (Lee and Mumford, 2003; Yuille and Kersten, 2006; Clark, 2013; Summerfield and de Lange, 2014). This view of the brain as a prediction machine that performs unconscious inference has a long history, going back to the seminal work of Alhazen and Helmholtz (Hatfield, 2002). Modern views cast this process in terms of Bayesian inference, where the brain is updating its internal model of the environment in order to explain away the data that impinge upon its senses, also referred to as the Bayesian brain hypothesis (Jaynes, 1988; Doya et al., 2006). The same reasoning underlies the free-energy principle, which assumes that biological systems minimize a free energy functional of their internal states that entail beliefs about hidden states in their environment (Friston, 2010). Predictions can be seen as central to the generation of adaptive behavior, since anticipating the future will allow an agent to select appropriate actions in the present (Schacter et al., 2007; Moulton and Kosslyn, 2009).

Prediction is central in model-based RL approaches since it requires agents to plan their actions by predicting the outcomes of future actions (Daw, 2012). This is strongly related to the notion of preplay of future events subserving path planning (Corneil and Gerstner, 2015). Such preplay has been observed in hippocampal place cell sequences (Dragoi and Tonegawa, 2011), giving further support to the idea that the hippocampal formation is involved in goal-directed navigation (Corneil and Gerstner, 2015). Prediction also allows an agent to prospectively act on expected deviations from optimal conditions. This focus on error-correction and stability is also prevalent in the work of the cybernetic movement (Ashby, 1952). Note further that predictive processing connects to the concept of allostasis, where the agent is actively trying to predict future states such as to minimize deviations from optimal homeostatic conditions. It is also central to optimal feedback control theory, which assumes that the motor system corrects only those deviations that interfere with task goals (Todorov and Jordan, 2002).

The notion of predictive processing has been very influential in neural network research. For example, it provides the basis for predictive coding models that introduce specific neural network architectures in which feedforward connections are used to transmit the prediction errors that result from discrepancies between top-down predictions and bottom-up sensations (Rao and Ballard, 1999; Huang and Rao, 2011). It also led to the development of a wide variety of generative models that are able to predict their sensory states, also referred to as fantasies (Hinton, 2013). Such fantasies may play a role in understanding cognitive processing involved in imagery, working memory and dreaming. In effect, these models aim to estimate a distribution over latent causes $\mathbf{z}$ in the environment that explain observed sensory data $\mathbf{x}$. In this setting, the most probable explanation is given by

$$\begin{aligned} \mathbf{z}^* &= \arg\max_{\mathbf{z}} p(\mathbf{z} \mid \mathbf{x}) \\ &= \arg\max_{\mathbf{z}} \big[p(\mathbf{x} \mid \mathbf{z}) p(\mathbf{z}) \big] . \end{aligned} \tag{11}$$

Generative models also offer a way to perform unsupervised learning, since if a neural network is able to generate predictions then the discrepancy between predicted and observed stimuli can serve as a teaching signal. A canonical example is the Boltzmann machine, which is a stochastic variant of a Hopfield network that is able to discover regularities in the training data using a simple unsupervised learning algorithm (Hinton and Sejnowski, 1983; Ackley et al., 1985). Another classical example is the Helmholtz machine, which incorporates both bottom-up and top-down processing (Dayan et al., 1995). Other, more recent examples of ANN-based generative models are deep belief networks (Hinton et al., 2006), variational autoencoders (Kingma and Welling, 2014) and generative adversarial networks (Goodfellow et al., 2014). Recent work has started to use these models to predict future sensory states from current observations (Lotter et al., 2016; Mathieu et al., 2016; Xue et al., 2016).

4.6. Reasoning

While ANNs are now able to solve complex tasks such as acting in natural environments or playing difficult board games, one could still argue that they are "just" performing sophisticated pattern

recognition rather than showing the symbolic reasoning abilities that characterize our own brains. The question of whether connectionist systems are capable of symbolic reasoning has a long history, and has been debated by various researchers in the cognitivist (symbolic) program (Pinker and Mehler, 1988). We will not settle this debate here but point out that efforts are underway to endow neural networks with sophisticated reasoning capabilities.

One example is the development of "differentiable computers" that learn to implement algorithms based on a finite amount of training data (Graves et al., 2014; Weston et al., 2015; Vinyals et al., 2017). The resulting neural networks perform variable binding and are able to deal with variable length structures (Graves et al., 2014), which are two objections that were originally raised against using ANNs to explain cognitive processing (Fodor and Pylyshyn, 1988).

Another example is the development of neural networks that can answer arbitrary questions about text (Bordes et al., 2015), images (Agrawal et al., 2016) and movies (Tapaswi et al., 2015), thereby requiring deep semantic knowledge about the experienced stimuli. Recent models have also been shown to be capable of compositional reasoning (Johnson et al., 2017; Lake et al., 2017; Yang et al., 2017), which is an important ingredient for explaining the systematic nature of human thought (Fodor and Pylyshyn, 1988). These architectures often make use of distributional semantics, where words are encoded as real vectors that capture word meaning (Mikolov et al., 2013; Ferrone and Zanzotto, 2017).

Several other properties characterize human thought processes, such as intuitive physics, intuitive psychology, relational reasoning and causal reasoning (Kemp and Tenenbaum, 2008; Lake et al., 2017). Another crucial hallmark of intelligent systems is that they are able to explain what they are doing (Brachman, 2002). This requires agents to have a deep understanding of their world. These properties should be replicated in neural networks if they are to serve as accurate models of natural intelligence. New neural network architectures are slowly starting to take steps in this direction (e.g., Louizos et al., 2017; Santoro et al., 2017; Zhu et al., 2017).

5. TOWARD STRONG AI

We have reviewed the computational foundations of natural intelligence and outlined how ANNs can be used to model a variety of cognitive processes. However, our current understanding of natural intelligence remains limited and strong AI has not yet been attained. In the following, we will touch upon a number of important topics that will be of importance for eventually reaching these goals.

5.1. Surviving in Complex Environments

Contemporary neural network architectures tend to excel at solving one particular problem well. However, in practice, we want to arrive at intelligent machines that are able to survive in complex environments. This requires the agent to deal with high-dimensional naturalistic input, be able to solve multiple tasks depending on context, and devise optimal strategies to ensure long-term survival.

The research community has embraced these desiderata by creating virtual worlds that allow development and testing of neural network architectures (e.g., Todorov et al., 2012; Beattie et al., 2016; Brockman et al., 2016; Kempka et al., 2016; Synnaeve et al., 2016)[10]. While most work in this area has focused on environments with fully observable states, reward functions with low delay, and small action sets, research is shifting toward environments that are partially observable, require long-term planning, show complex dynamics and have noisy and high-dimensional control interfaces (Synnaeve et al., 2016).

A particular challenge in these naturalistic environments is that networks need to be able to exhibit continual (life-long) learning (Thrun and Mitchell, 1995), adapting continuously to the current state of affairs. This is difficult due to the phenomenon of catastrophic forgetting (McCloskey and Cohen, 1989; French, 1999), where previously acquired skills are overwritten by ongoing modification of synaptic weights. Recent algorithmic developments attenuate the detrimental effects of catastrophic forgetting (Kirkpatrick et al., 2015; Zenke et al., 2015), offering a (partial) solution to the stability vs. plasticity dilemma (Abraham and Robins, 2005). Life-long learning is further complicated by the exploration-exploitation dilemma, where agents need to decide on whether to accrue either information or reward (Cohen et al., 2007). Another challenge is the fact that reinforcement learning of complex actions is notoriously slow. Here, progress is being made using networks that make use of differentiable memories (Santoro et al., 2016; Pritzel et al., 2017). Survival in complex environments also requires that agents learn to perform multiple tasks well. This learning process can be facilitated through multitask learning (Caruana, 1997) (also referred to as learning to learn Baxter, 1998 or transfer learning Pan and Fellow, 2009), where learning of one task is facilitated by knowledge gained through learning to solve another task. Multitask learning has been shown to improve convergence speed and generalization to unseen data (Scholte et al., 2017). Finally, effective learning also calls for agents that can generalize to cases that were not encountered before, which is known as zero-shot learning (Palatucci et al., 2009), and can learn from rare events, which is known as one-shot learning (Fei-Fei et al., 2006; Vinyals et al., 2016; Kaiser and Roy, 2017).

While the use of virtual worlds allows for testing the capabilities of artificial agents, it does not guarantee that the same agents are able to survive in the real world (Brooks, 1992). That is, there may exist a reality gap, where skills acquired in virtual worlds do not carry over to the real world. In contrast to virtual worlds, acting in the real world requires the agent to deal with unforeseen circumstances resulting from the complex nature of reality, the agent's need for a physical body, as well as its engagement with a myriad of other agents (Anderson, 2003). Moreover, the continuing interplay between an organism and its environment may itself shape and, ultimately, determine cognition (Gibson, 1979; Maturana

[10]See SHRDLU for an early example of such a virtual world (Winograd, 1972).

and Varela, 1987; Brooks, 1996; Edelman, 2015). Effectively dealing with these complexities may not only require plasticity in individual agents but also the incorporation of developmental change, as well as learning at evolutionary time scales (Marcus, 2009). From a developmental perspective, networks can be more effectively trained by presenting them with a sequence of increasingly complex tasks, instead of immediately requiring the network to solve the most complex task (Elman, 1993). This process is known as curriculum learning (Bengio et al., 2009) and is analogous to how a child learns by decomposing problems into simpler subproblems (Turing, 1950). Evolutionary strategies have also been shown to be effective in learning to solve challenging control problems (Salimans et al., 2017). Finally, to learn about the world, we may also turn toward cultural learning, where agents can offload task complexity by learning from each other (Bengio, 2014).

As mentioned in section 2.2, adaptive behavior is the result of multiple competing drives and motivations that provide primary, intrinsic and extrinsic rewards. Hence, one strategy for endowing machines with the capacity to survive in the real world is to equip neural networks with drives and motivations that ensure their long-term survival[11]. In terms of primary rewards, one could conceivably provide artificial agents with the incentive to minimize computational resources or maximize offspring via evolutionary processes (Stanley and Miikkulainen, 2002; Floreano et al., 2008; Gauci and Stanley, 2010). In terms of intrinsic rewards, one can think of various ways to equip agents with the drive to explore the environment (Oudeyer, 2007). We briefly describe a number of principles that have been proposed in the literature. Artificial curiosity assumes that internal reward depends on how boring an environment is, with agents avoiding fully predictable and unpredictably random states (Schmidhuber, 1991, 2003; Pathak et al., 2017). A related notion is that of information-seeking agents (Bachman et al., 2016). The autotelic principle formalizes the concept of flow where an agent tries to maintain a state where learning is challenging, but not overwhelming (Csikszentmihalyi, 1975; Steels, 2004). The free-energy principle states that an agent seeks to minimize uncertainty by updating its internal model of the environment and selecting uncertainty-reducing actions (Friston, 2009, 2010). Empowerment is founded on information-theoretic principles and quantifies how much control an agent has over its environment, as well as its ability to sense this control (Klyubin et al., 2005a,b; Salge et al., 2013). In this setting, intrinsically motivated behavior is induced by the maximization of empowerment. Finally, various theories embrace the notion that optimal prediction of future states drives learning and behavior (Der et al., 1999; Kaplan and Oudeyer, 2004; Ay et al., 2008). In terms of extrinsic rewards, one can think of imitation learning, where a teacher signal is used to inform the agent about its desired outputs (Schaal, 1999; Duan et al., 2017).

[11]The notion of *wanting* agents was already present in the writings of Thurstone (1923), who wrote: "My main thesis is that conduct originates in the organism itself and not in the environment in the form of a stimulus. [...] All mental life may be looked upon as incomplete behavior which is in the process of being formed. [...] Perception is the discovery of the suitable stimulus which is often anticipated imaginally. The appearance of the stimulus is one of the last events in the expression of impulses in conduct. The stimulus is not the starting point for behavior."

5.2. Bridging the Gap between Artificial and Biological Neural Networks

To reduce the gap between artificial and biological neural networks, it makes sense to assess their operation on similar tasks. This can be done either by comparing the models at a neurobiological level or at a behavioral level. The former refers to comparing the internal structure or activation patterns of artificial and biological neural networks. The latter refers to comparing their behavioral outputs (e.g., eye movements, reaction times, high-level decisions). Moreover, comparisons can be made under changing conditions, i.e., during learning and development (Elman et al., 1996). As such, ANNs can serve as explanatory mechanisms in cognitive neuroscience and behavioral psychology, embracing recent model-based approaches (Forstmann and Wagenmakers, 2015).

From a psychological perspective, ANNs have been compared explicitly with their biological counterparts. Connectionist models were widely used in the 1980's to explain various psychological phenomena, particularly by the parallel distributed processing (PDP) movement, which stressed the parallel nature of neural processing and the distributed nature of neural representations (McClelland, 2003). For example, neural networks have been used to explain grammar acquisition (Elman, 1991), category learning (Kruschke, 1992) and the organization of the semantic system (Ritter and Kohonen, 1989). More recently, deep neural networks have been used to explain human similarity judgments (Peterson et al., 2016). With new developments in cognitive and affective computing, where neural networks become more adept at solving high-level cognitive tasks, such as predicting people's (apparent) personality traits (Güçlütürk et al., 2016), their use as a tool to explain psychological phenomena is likely to increase. This will also require embracing insights about how humans solve problems at a cognitive level (Tenenbaum et al., 2011).

ANNs have also been related explicitly to brain function. For example, the perceptron has been used in the modeling of various neuronal systems, including sensorimotor learning in the cerebellum (Marr, 1969) and associative memory in cortex (Gardner, 1988), sparse coding has been used to explain receptive field properties (Olshausen and Field, 1996), topographic maps have been used to explain the formation of cortical maps (Obermayer, 1990; Aflalo, 2006), Hebbian learning has been used to explain neural tuning to face orientation (Leibo et al., 2017), and networks trained by backpropagation have been used to model the response properties of posterior parietal neurons (Zipser and Andersen, 1988). Neural networks have also been used to model central pattern generators that drive behavior (Duysens and Van de Crommert, 1998; Ijspeert, 2008) as well as the perception of rhythmic stimuli (Torras i Genís, 1986; Gasser, Eck and Port, 1999). Furthermore, reinforcement learning algorithms used to train neural networks for action selection have strong ties with the brain's reward system (Schultz

et al., 1997; Sutton and Barto, 1998). It has been shown that RNNs trained to solve a variety of cognitive tasks using reinforcement learning replicate various phenomena observed in biological systems (Song et al., 2016; Miconi, 2017). Crucially, these efforts go beyond descriptive approaches in that they may explain *why* the human brain is organized in a certain manner (Barak, 2017).

Rather than using neural networks to explain certain observed neural or behavioral phenomena, one can also directly fit neural networks to neurobehavioral data. This can be achieved via an indirect approach or via a direct approach. In the *indirect* approach, neural networks are first trained to solve a task of interest. Subsequently, the trained network's responses are fitted to neurobehavioral data obtained as participants engage in the same task. Using this approach, deep convolutional neural networks trained on object recognition, action recognition and music tagging have been used to explain the functional organization of visual as well as auditory cortex (Güçlü and van Gerven, 2015, 2017a; Güçlü et al., 2016). The indirect approach has also been used to train RNNs via reinforcement learning on a probabilistic categorization task. These networks have been used to fit the learning trajectories and behavioral responses of humans engaged in the same task (Bosch et al., 2016). Mante et al. (2013) used RNNs to model the population dynamics of single neurons in prefrontal cortex during a context-dependent choice task. In the *direct* approach, neural networks are trained to directly predict neural responses. For example, Mcintosh et al. (2016) trained convolutional neural networks to predict retinal responses to natural scenes, Joukes et al. (2014) trained RNNs to predict neural responses to motion stimuli, and Güçlü and van Gerven (2017b) used RNNs to predict cortical responses to naturalistic video clips. This ability of neural networks to explain neural recordings is expected to become increasingly important (Sompolinsky, 2014; Marder, 2015), given the emergence of new imaging technology where the activity of thousands of neurons can be measured in parallel (Ahrens et al., 2013; Churchland and Sejnowski, 2016; Lopez et al., 2016; Pachitariu et al., 2016; Yang and Yuste, 2017). Better understanding will also be facilitated by the development of new data analysis techniques to elucidate human brain function (Kass et al., 2014)[12], the use of ANNs to decode neural representations (Schoenmakers et al., 2013; Güçlütürk et al., 2017), as well as the development of approaches that elucidate the functioning of ANNs (e.g., Nguyen et al., 2016; Kindermans et al., 2017; Miller, 2017)[13].

5.3. Next-Generation Artificial Neural Networks

The previous sections outlined how neural networks can be made to solve challenging tasks and provide explanations of neural and behavioral responses in biological agents. In this final section, we consider some developments that are expected to fuel the next generation of ANNs.

First, a major driving force in neural network research will be theoretical and algorithmic developments that inform why ANNs work so well in practice, what their fundamental limitations are, as well as how to overcome these. From a theoretical point of view, substantial advances have already been made pertaining to, for example, understanding the nature of representations (Anselmi and Poggio, 2014; Lin and Tegmark, 2016; Shwartz-Ziv and Tishby, 2017), the statistical mechanics of neural networks (Sompolinsky, 1988; Advani et al., 2013), as well as the expressiveness (Pascanu et al., 2013; Bianchini and Scarselli, 2014; Kadmon and Sompolinsky, 2016; Mhaskar et al., 2016; Poole et al., 2016; Raghu et al., 2016; Weichwald et al., 2016), generalizability (Kawaguchi et al., 2017) and learnability (Dauphin et al., 2014; Saxe et al., 2014; Schoenholz et al., 2017) of DNNs.

From an algorithmic point of view, great strides have been made in improving training of deep (Srivastava et al., 2014; He et al., 2015; Ioffe and Szegedy, 2015) and recurrent neural networks (Hochreiter and Schmidhuber, 1997; Pascanu et al., 2012), overcoming the reality gap (Tobin et al., 2017), adding modularity to neural networks (Fernando et al., 2017), as well as on improving the efficacy of reinforcement learning algorithms (Schulman et al., 2015; Mnih et al., 2016; Pritzel et al., 2017).

Second, it is expected that as neural network models become more plausible from a biological point of view, model fit and task performance will further improve (Cox and Dean, 2014). This is important in driving new developments in model-based cognitive neuroscience but also in developing intelligent machines that show human-like behavior. One example is to match the object recognition capabilities of extremely deep neural networks with more biologically plausible RNNs of limited depth (O'Reilly et al., 2013; Liao and Poggio, 2016) and achieving category selectivity in a more realistic manner (Peelen and Downing, 2017; Scholte et al., 2017). Another example is to incorporate predictive coding principles in neural network architectures (Lotter et al., 2016). Furthermore, more human-like perceptual systems can be arrived at by including attentional mechanisms (Mnih et al., 2014) as well as mechanisms for saccade planning (Najemnik and Geisler, 2005; Larochelle and Hinton, 2010; Gregor et al., 2014).

In general, ANN research can benefit from a close interaction between the AI and neuroscience communities (Yuste, 2015; Hassabis et al., 2017). For example, neural network research may be shaped by general guiding principles of brain function at different levels of analysis (O'Reilly, 1998; Maass, 2016; Sterling and Laughlin, 2016). We may also strive to incorporate more biological detail. For example, to obtain accurate models of neural information processing we may need to embrace spike-based rather than rate-based neural networks (Brette, 2015)[14]. Efforts are underway to effectively train spiking neural

[12] But see Jonas and Kording (2017) for a critical appraisal of the informativeness of such techniques.

[13] These techniques aim to overcome the interpretability problem raised by Mozer and Smolensky (1989), who state: "One thing that connectionist networks have in common with brains is that if you open them up and peer inside, all you can see is a big pile of goo."

[14] While there surely exists neurobiological evidence for temporal coding with spikes (Segundo et al., 1966; Barrio and Buno, 1990; Bohte, 2004), it remains an

networks (Maass, 1997; Gerstner and Kistler, 2002; Gerstner et al., 2014; O'Connor and Welling, 2016; Huh and Sejnowski, 2017) and endow them with the same cognitive capabilities as their rate-based cousins (Thalmeier et al., 2015; Abbott et al., 2016; Kheradpisheh et al., 2016; Lee et al., 2016; Zambrano and Bohte, 2016).

In the same vein, researchers are exploring how probabilistic computations can be performed in neural networks (Nessler et al., 2013; Pouget et al., 2013; Gal, 2016; Orhan and Ma, 2016; Ambrogioni et al., 2017; Heeger, 2017; Mandt et al., 2017) and deriving new biologically plausible synaptic plasticity rules (Brea and Gerstner, 2016; Brea et al., 2016; Schiess et al., 2016). Biologically-inspired principles may also be incorporated at a more conceptual level. For instance, researchers have shown that neural networks can be protected from adversarial attacks (i.e., the construction of stimuli that cause networks to make mistakes) by integrating the notion of nonlinear computations encountered in the branched dendritic structures of real neurons (Nayebi and Ganguli, 2016).

Finally, research is invested in implementing ANNs in hardware, also referred to as neuromorphic computing (Mead, 1990). These brain-based parallel chip architectures hold the promise of devices that operate in real time and with very low power consumption (Schuman et al., 2017), driving new advances in cognitive computing (Modha et al., 2011; Neftci et al., 2013; Van de Burgt et al., 2017). On a related note, nanotechnology may 1 day drive the development of new neural network architectures whose operation is closer to the molecular machines that mediate the operation of biological neural networks (Drexler, 1992; Strukov, 2011). In the words of Feynman (1992): "There's plenty of room at the bottom."

6. CONCLUSION

As cognitive scientists, we live in exciting times. Cognitivism offers an interpretation of agents as information processing systems that are engaged in formal symbol manipulation. The probabilistic approach to cognition extends this interpretation by viewing organisms as rational agents that need to act in the face of uncertainty under limited resources. Finally, emergentist approaches such as artificial life and connectionism indicate that concerted interactions between simple processing elements can achieve human-level performance at certain cognitive tasks. While these different views have stirred substantial debate in the past, they need not be irreconcilable. Surely we are capable of formal symbol manipulation and decision making under uncertainty in real-life settings. At the same time, these capabilities must be implemented by the neural circuits that make up our own brains, which themselves rely on noisy long-range communication between neuronal populations.

The thesis of this paper is that natural intelligence can be modeled and understood by constructing artificial agents whose synthetic brains are composed of (rate-based) neural networks. To act as explanations of natural intelligence, these synthetic brains should show a functional correspondence with their biological counterparts. To identify such correspondence we can embrace the rich sources of data provided by biology, neuroscience and psychology, providing a link to Marr's implementational level. At the same time, we can use sophisticated machinery developed in mathematics, computer science and physics to gain a better understanding of these systems. Ultimately, these synthetic brains should be able to show the capabilities that are prescribed by normative theories of intelligent behavior, providing a link to Marr's computational level.

The supposition that artificial neural networks are sufficient for modeling all of cognition may seem premature. For example, state-of-the-art question-answering systems such as IBM's Watson (Ferrucci et al., 2010) use ANN technology as a minor component within a larger (symbolic) framework and the AlphaGo system (Silver et al., 2017), which learns to play the game of Go beyond grandmaster level without any human intervention, combines neural networks with Monte Carlo tree search. While it is true that ANNs remain wanting when it comes to logical reasoning, inferring causal relationships or planning, the pace of current research may very well bring these capabilities within reach in the foreseeable future. Such neural networks may turn out to be quite different from current neural network architectures and their operation may be guided by complementary yet-to-be-discovered learning rules.

The quest for natural intelligence can be contrasted with a pure engineering approach. From an engineering perspective, understanding natural intelligence may be considered irrelevant since the main interest is in building devices that do the job. To quote Edsger Dijkstra, "the question whether machines can think [is] as relevant as the question whether submarines can swim." At the same time, our quest for natural intelligence may facilitate the development of strong AI given the proven ability of our own brains to generate intelligent behavior. Hence, biologically inspired architectures may not only provide new insights into human brain function but could also in the long run yield superior curious and perhaps even conscious machines that surpass humans in terms of intelligence, creativity, playfulness, and empathy (Boden, 1998; Moravec, 2000; Der and Martius, 2011; Modha et al., 2011; Harari, 2017).

AUTHOR CONTRIBUTIONS

The author confirms being the sole contributor of this work and approved it for publication.

ACKNOWLEDGMENTS

This work is supported by a VIDI grant (639.072.513) from the Netherlands Organization for Scientific Research. I would like to thank Nadine Dijkstra, Gabriëlle Ras, Andrew Reid, Katja Seeliger and the reviewers for their useful comments.

open question if temporal coding is absolutely necessary for the generation of adaptive behavior. In the end, computing with spikes may have emerged chiefly to promote efficiency and allow long-distance neuronal communication (Laughlin and Sejnowski, 2003).

REFERENCES

Abbott, L. F., Depasquale, B., and Memmesheimer, R.-M. (2016). Building functional networks of spiking model neurons. *Nat. Neurosci.* 19, 350–355. doi: 10.1038/nn.4241

Abraham, W. C., and Robins, A. (2005). Memory retention - the synaptic stability versus plasticity dilemma. *Trends Neurosci.* 28, 73–78. doi: 10.1016/j.tins.2004.12.003

Ackley, D., Hinton, G. E., and Sejnowski, T. (1985). A learning algorithm for Boltzmann machines. *Cogn. Sci.* 9, 147–169. doi: 10.1016/S0364-0213(85)80012-4

Adams, S. S., Arel, I., Bach, J., Coop, R., Furlan, R., Goertzel, B., et al. (2012). Mapping the landscape of human-level artificial general intelligence. *AI Mag.* 33, 25–42. doi: 10.1609/aimag.v33i1.2322

Advani, M., Lahiri, S., and Ganguli, S. (2013). Statistical mechanics of complex neural systems and high dimensional data. *J. Stat. Mech. Theory Exp.* 2013:P03014. doi: 10.1088/1742-5468/2013/03/P03014

Aflalo, T. N. (2006). Possible origins of the complex topographic organization of motor cortex: reduction of a multidimensional space onto a two-dimensional array *J. Neurosci.* 26, 6288–6297. doi: 10.1523/JNEUROSCI.0768-06.2006

Agrawal, A., Lu, J., Antol, S., Mitchell, M., Zitnick, C. L., Batra, D., et al. (2016). VQA: visual question answering. ArXiv:1505.00468, 1–25.

Ahrens, M. B., Orger, M. B., Robson, D. N., Li, J. M., and Keller, P. J. (2013). Whole-brain functional imaging at cellular resolution using light-sheet microscopy. *Nat. Methods* 10, 413–420. doi: 10.1038/nmeth.2434

Ambrogioni, L., Güçlü, U., Maris, E., and van Gerven, M. A. J. (2017). Estimating nonlinear dynamics with the ConvNet smoother. ArXiv:1702.05243, 1–8.

Amunts, K., Ebell, C., Muller, J., Telefont, M., Knoll, A., and Lippert, T. (2016). The Human Brain Project: creating a European research infrastructure to decode the human brain. *Neuron* 92, 574–581. doi: 10.1016/j.neuron.2016.10.046

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., and Qin, Y. (2004). An integrated theory of the mind. *Psychol. Rev.* 111, 1036–1060. doi: 10.1037/0033-295X.111.4.1036

Anderson, M. L. (2003). Embodied cognition: a field guide. *Artif. Intell.* 149, 91–130. doi: 10.1016/S0004-3702(03)00054-7

Andrieu, C., De Freitas, N., Doucet, A., and Jordan, M. I. (2003). An introduction to MCMC for machine learning. *Mach. Learn.* 50, 5–43. doi: 10.1023/A:1020281327116

Anselmi, F., and Poggio, T. A. (2014). *Representation Learning in Sensory Cortex: A Theory*. Tech. Rep. CBMM Memo 026, MIT.

Ashby, W. (1952). *Design for a Brain*. London: Chapman & Hall. doi: 10.1007/978-94-015-1320-3

Ay, N., Bertschinger, N., Der, R., Güttler, F., and Olbrich, E. (2008). Predictive information and explorative behavior of autonomous robots. *Eur. Phys. J. B* 63, 329–339. doi: 10.1140/epjb/e2008-00175-0

Bachman, P., Sordoni, A., and Trischler, A. (2016). Towards information-seeking agents. ArXiv:1612.02605v1, 1–11.

Badre, D. (2008). Cognitive control, hierarchy, and the rostro-caudal organization of the frontal lobes. *Trends Cogn. Sci.* 12, 193–200. doi: 10.1016/j.tics.2008.02.004

Barak, O. (2017). Recurrent neural networks as versatile tools of neuroscience research. *Curr. Opin. Neurobiol.* 46, 1–6. doi: 10.1016/j.conb.2017.06.003

Barkow, J. H., Cosmides, L., and Tooby, J. (eds.). (1992). *The Adapted Mind: Evolutionary Psychology and the Generation of Culture*. New York, NY: Oxford University Press. doi: 10.1017/s0730938400018700

Barlow, H. (2009). "Grandmother cells, symmetry, and invariance: how the term arose and what the facts suggest," in *Cognitive Neurosciences*, ed M. S. Gazzaniga (Cambridge, MA: MIT Press), 309–320.

Barrio, L. C., and Buno, W. (1990). Temporal correlations in sensory-synaptic interactions: example in crayfish stretch receptors. *J. Neurophys.* 63, 1520–1528.

Baxter, J. (1998). "Theoretical models of learning to learn," in *Learning to Learn*, eds S. Thrun and L. Pratt (Norwell, MA: Kluwer Academic Publishers), 71–94. doi: 10.1007/978-1-4615-5529-2_4

Beattie, C., Leibo, J. Z., Teplyashin, D., Ward, T., Wainwright, M., Lefrancq, A., et al. (2016). DeepMind lab. ArXiv:1612.03801v2, 1–11.

Bechtel, W. (1993). The case for connectionism. *Philos. Stud.* 71, 119–154. doi: 10.1007/bf00989853

Bedau, M. A. (2003). Artificial life: organization, adaptation and complexity from the bottom up. *Trends Cogn. Sci.* 7, 505–512. doi: 10.1016/j.tics.2003.09.012

Bengio, Y. (2009). Learning deep architectures for AI. *Found. Trends Mach. Learn.* 2, 1–87. doi: 10.1561/2200000006

Bengio, Y. (2014). "Evolving culture vs local minima," in *Growing Adaptive Machine*, eds T. Kowaliw, N. Bredeche, and R. Doursat (Berlin: Springer-Verlag), 109–138. doi: 10.1007/978-3-642-55337-0_3

Bengio, Y., and LeCun, Y. (2007). "Scaling learning algorithms towards AI," in *Large Scale Kernel Machines*, eds L. Bottou, O. Chapelle, D. DeCoste, and J. Weston (Cambridge, MA: The MIT Press), 321–360.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). "Curriculum learning," in *Proceedings of the 26th Annual International Conference on Machine Learning* (Montreal), 1–8. doi: 10.1145/1553374.1553380

Bianchini, M., and Scarselli, F. (2014). On the complexity of neural network classifiers: a comparison between shallow and deep architectures. *IEEE Trans. Neural Netw. Learn. Syst.* 25, 1553–1565. doi: 10.1109/TNNLS.2013.2293637

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford: Oxford University Press.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2016). Variational inference: a review for statisticians. ArXiv:1601.00670v5, 1–33.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.* 3, 993–1022. doi: 10.1162/jmlr.2003.3.4-5.993

Boden, M. A. (1998). Creativity and artificial intelligence. *Artif. Intell.* 103, 347–356. doi: 10.1016/S0004-3702(98)00055-1

Bohte, S. M. (2004). The evidence for neural information processing with precise spike-times: a survey. *Nat. Comput.* 3, 195–206. doi: 10.1023/b:naco.0000027755.02868.60

Bordes, A., Chopra, S., and Weston, J. (2015). Large-scale simple question answering with memory networks. ArXiv:1506.02075v1, 1–10.

Bosch, S. E., Seeliger, K., and van Gerven, M. A. J. (2016). Modeling cognitive processes with neural reinforcement learning. BioArxiv, 1–19.

Brachman, R. J. (2002). Systems that know what they're doing. *IEEE Intell. Syst.* 17, 67–71. doi: 10.1109/mis.2002.1134363

Braitenberg, V. (1986). *Vehicles: Experiments in Synthetic Psychology*. Cambridge, MA: The MIT Press. doi: 10.1016/0004-3702(85)90057-8

Brea, J., Gaál, A. T., Urbanczik, R., and Senn, W. (2016). Prospective coding by spiking neurons. *PLoS Comput. Biol.* 12:e1005003. doi: 10.1371/journal.pcbi.1005003

Brea, J., and Gerstner, W. (2016). Does computational neuroscience need new synaptic learning paradigms? *Curr. Opin. Behav. Sci.* 11, 61–66. doi: 10.1016/j.cobeha.2016.05.012

Brette, R. (2015). Philosophy of the spike: rate-based vs spike-based theories of the brain. *Front. Syst. Neurosci.* 9:151. doi: 10.3389/fnsys.2015.00151

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., et al. (2016). OpenAI gym. ArXiv:1606.01540v1, 1–4.

Brooks, R. A. (1992). "Artificial life and real robots," in *Toward a Practice of Autonomous Systems, Proceedings of First European Conference on Artificial Life*, eds F. J. Varela and P. Bourgine (Cambridge, MA: The MIT Press, Bradford Books).

Brooks, R. A. (1996). "Prospects for human level intelligence for humanoid robots," in *Proceedings of the First International Symposium on Humanoid Robots* (Tokyo), 17–24.

Brown, L. V. (2007). *Psychology of Motivation*. New York, NY: Nova Publishers.

Buschman, T. J., Miller, E. K., and Miller, E. K. (2014). Goal-direction and top-down control. *Philos. Trans. R. Soc. B* 369, 1–9. doi: 10.1098/rstb.2013.0471

Cannon, W. B. (1929). Organization for physiological homeostasis. *Physiol. Rev.* 9, 399–431.

Carnevale, F., de Lafuente, V., Romo, R., Barak, O., and Parga, N. (2015). Dynamic control of response criterion in premotor cortex during perceptual detection under temporal uncertainty. *Neuron* 86, 1067–1077. doi: 10.1016/j.neuron.2015.04.014

Carr, C. E., and Konishi, M. (1990). A circuit for detection of interaural time differences in the brain stem of the barn owl. *J. Neurosci.* 10, 3227–3246.

Caruana, R. (1997). Multitask learning. *Mach. Learn.* 28, 41–75. doi: 10.1109/TCBB.2010.22

Chang, E. F. (2015). Towards large-scale, human-based, mesoscopic neurotechnologies. *Neuron* 86, 68–78. doi: 10.1016/j.neuron.2015.03.037

Cho, K., van Merrienboer, B., Bahdanau, D., and Bengio, Y. (2014). "On the properties of neural machine translation: encoder-decoder approaches," in *Proceedings of the SSST-8, Eighth Work Syntax Semantics and Structure in Statistical Translation* (Doha), 103–111. doi: 10.3115/v1/w14-4012

Churchland, P. S., and Sejnowski, T. J. (2016). Blending computational and experimental neuroscience. *Nat. Rev. Neurosci.* 17, 667–668. doi: 10.1038/nrn.2016.114

Clark, A. (2013). Whatever next? Predictive brains, situated agents, and the future of cognitive science. *Behav. Brain Sci.* 36, 181–204. doi: 10.1017/s0140525x12000477

Cohen, J. D., McClure, S. M., and Yu, A. J. (2007). Should I stay or should I go? How the human brain manages the trade-off between exploitation and exploration. *Philos. Trans. R. Soc. B* 362, 933–942. doi: 10.1098/rstb.2007.2098

Copeland, B. J., and Proudfoot, D. (1996). On Alan Turing's anticipation of connectionism. *Synthese* 108, 361–377. doi: 10.1007/bf00413694

Corneil, D., and Gerstner, W. (2015). "Attractor network dynamics enable preplay and rapid path planning in maze-like environments," in *Advances in Neural Information Processing Systems 28* (Montreal), 1–9.

Cox, D. D., and Dean, T. (2014). Neural networks and neuroscience-inspired computer vision. *Curr. Biol.* 24, R921–R929. doi: 10.1016/j.cub.2014.08.026

Crick, F., and Mitchison, G. (1983). The function of dream sleep. *Nature* 304, 111–114. doi: 10.1038/304111a0

Csikszentmihalyi, M. (1975). *Beyond Boredom and Anxiety: Experiencing Flow in Work and Play*. Hoboken, NJ: John Wiley & Sons Inc. doi: 10.2307/2065805

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Math. Control Signals Syst.* 2, 303–314. doi: 10.1007/BF02134016

Dauphin, Y., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., and Bengio, Y. (2014). Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. ArXiv:1406.2572, 1–14.

Davies, N. B., Krebs, J. R., and West, S. A. (2012). *An Introduction to Behavioral Ecology, 4th Edn*. Hoboken, NJ: John Wiley & Sons. doi: 10.1037/026600

Daw, N. D. (2012). "Model-based reinforcement learning as cognitive search: neurocomputational theories," in *Cognitive Search: Evolution, Algorithms, and the Brain*, eds P. M. Todd, T. T. Hills, and T. W. Robbins (Cambridge, MA: The MIT Press), 195–208. doi: 10.7551/mitpress/9780262018098.001.0001

Dawkins, R. (2016). *The Selfish Gene, 4th Edn*. Oxford: Oxford University Press. doi: 10.4324/9781912281251

Dawson, M. R. W., and Shamanski, K. S. (1994). Connectionism, confusion, and cognitive science. *J. Intell. Syst.* 4, 215–262. doi: 10.1515/jisys.1994.4.3-4.215

Dayan, P., and Abbott, L. F. (2005). *Theoretical Neuroscience*. Cambridge, MA: MIT Press.

Dayan, P., Hinton, G. E., Neal, R., and Zemel, R. (1995). The Helmholtz machine. *Neural Comput.* 7, 1–16. doi: 10.1162/neco.1995.7.5.889

de Garis, H., Shuo, C., Goertzel, B., and Ruiting, L. (2010). A world survey of artificial brain projects, Part I Large-scale brain simulations. *Neurocomputing* 74, 3–29. doi: 10.1016/j.neucom.2010.08.004

Delalleau, O., and Bengio, Y. (2011). "Shallow vs. deep sum-product networks," in *Advances in Neural Information Processing Systems 24* (Granada), 666–674.

Der, R., and Martius, G. (2011). *The Playful Machine: Theoretical Foundation and Practical Realization of Self-Organizing Robots*. Berlin: Springer Verlag. doi: 10.1007/978-3-642-20253-7

Der, R., Steinmetz, U., and Pasemann, F. (1999). Homeokinesis - a new principle to back up evolution with learning. *Comput. Intell. Model. Control. Autom.* 55, 43–47.

Dewar, R. C. (2003). Information theory explanation of the fluctuation theorem, maximum entropy production and self-organized criticality in non-equilibrium stationary states. *J. Phys. A Math. Gen.* 36, 631–641. doi: 10.1088/0305-4470/36/3/303

Dewar, R. C. (2005). Maximum entropy production and the fluctuation theorem. *J. Phys. A Math. Gen.* 38, L371–L381. doi: 10.1088/0305-4470/38/21/L01

Dewey, J. (1896). The reflex arc concept in psychology. *Psychol. Rev.* 3, 357–370. doi: 10.1037/11304-041

Doya, K., Ishii, S., Pouget, A., and Rao, R. P. N. (eds.). (2006). *Bayesian Brain: Probabilistic Approaches to Neural Coding*. Cambridge, MA: The MIT Press.

Dragoi, G., and Tonegawa, S. (2011). Hippocampal cellular assemblies. *Nature* 469, 397–401. doi: 10.1038/nature09633

Drexler, K. E. (1992). *Nanosystems: Molecular Machinery, Manufacturing, and Computation*. New York, NY: Wiley Interscience. doi: 10.1016/S0010-8545(96)90165-4

Duan, Y., Andrychowicz, M., Stadie, B. C., Ho, J., Schneider, J., Sutskever, I., et al. (2017). One-shot imitation learning. ArXiv:1703.07326v2, 1–23.

Duysens, J., and Van de Crommert, H. W. A. A. (1998). Neural control of locomotion; The central pattern generator from cats to humans. *Gait Posture* 7, 131–141. doi: 10.1016/S0966-6362(97)00042-8

Edelman, S. (2015). The minority report: some common assumptions to reconsider in the modelling of the brain and behavior. *J. Exp. Theor. Artif. Intell.* 3079, 1–26. doi: 10.1080/0952813X.2015.1042534

Elman, J. L. (1990). Finding structure in time. *Cogn. Sci.* 14, 179–211. doi: 10.1016/0364-0213(90)90002-E

Elman, J. L. (1991). Distributed representations, simple recurrent networks, and grammatical structure. *Mach. Learn.* 7, 195–225. doi: 10.1023/A:1022699029236

Elman, J. L. (1993). Learning and development in neural networks - The importance of starting small. *Cognition* 48, 71–99. doi: 10.1016/S0010-0277(02)00106-3

Elman, J. L., Bates, E. A., Johnson, M. H., Karmiloff-Smith, A., Parisi, D., and Plunkett, K. (1996). *Rethinking Innateness: A Connectionist Perspective on Development*. Cambridge, MA: The MIT Press. doi: 10.1017/s0272263198333070

Fei-Fei, L., Fergus, R., Member, S., and Perona, P. (2006). One-shot learning of object categories. *IEEE Trans. Patt. Anal. Mach. Intell.* 28, 594–611. doi: 10.1109/tpami.2006.79

Felleman, D. J., and Van Essen, D. C. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cereb. Cortex* 1, 1–47. doi: 10.1093/cercor/1.1.1

Fernando, C., Banarse, D., Blundell, C., Zwols, Y., Ha, D., Rusu, A., et al. (2017). PathNet: evolution channels gradient descent in super neural networks. ArXiv:1701.08734v1.

Ferrone, L., and Zanzotto, F. M. (2017). Symbolic, distributed and distributional representations for natural language processing in the era of deep learning: a survey. ArXiv:1702.00764, 1–25.

Ferrucci, D., Brown, E., Chu-carroll, J., Fan, J., Gondek, D., Kalyanpur, A. A., et al. (2010). Building Watson: an overview of the DeepQA project. *AI Mag.* 31, 59–79. doi: 10.1609/aimag.v31i3.2303

Feynman, R. (1992). There's plenty of room at the bottom. *J. Microelectromech. Syst.* 1, 60–66. doi: 10.1109/84.128057

Floreano, D., Dürr, P., and Mattiussi, C. (2008). Neuroevolution: from architectures to learning. *Evol. Intell.* 1, 47–62. doi: 10.1007/s12065-007-0002-4

Fodor, J. A., and Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: a critical analysis. *Cognition* 28, 3–71. doi: 10.1016/0010-0277(88)90031-5

Forstmann, B. U., and Wagenmakers, E.-J. (2015). *Model-Based Cognitive Neuroscience: A Conceptual Introduction*. New York, NY: Springer. doi: 10.1007/978-1-4939-2236-9_7

French, R. M. (1999). Catastrophic forgetting in connectionist networks. *Trends Cogn. Sci.* 3, 128–135. doi: 10.1016/s1364-6613(99)01294-2

Friston, K. J. (2009). The free-energy principle: a rough guide to the brain? *Trends Cogn. Sci.* 13, 293–301. doi: 10.1016/j.tics.2009.04.005

Friston, K. J. (2010). The free-energy principle: a unified brain theory? *Nat. Rev. Neurosci.* 11, 127–138. doi: 10.1038/nrn2787

Friston, K. J., Daunizeau, J., Kilner, J., and Kiebel, S. J. (2010). Action and behavior: a free-energy formulation. *Biol. Cybern.* 102, 227–260. doi: 10.1007/s00422-010-0364-z

Fry, R. L. (2017). Physical intelligence and thermodynamic computing. *Entropy* 19, 1–27. doi: 10.20944/PREPRINTS201701.0097.V1

Fukushima, K. (1980). Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybern.* 36, 193–202. doi: 10.1007/bf00344251

Fukushima, K. (2013). Artificial vision by multi-layered neural networks: neocognitron and its advances. *Neural Netw.* 37, 103–119. doi: 10.1016/j.neunet.2012.09.016

Funahashi, K.-I., and Nakamura, Y. (1993). Approximation of dynamical systems by continuous time recurrent neural networks. *Neural Netw.* 6, 801–806. doi: 10.1016/s0893-6080(05)80125-x

Fuster, J. M. (2001). The prefrontal cortex - An update: time is of the essence. *Neuron* 30, 319–333. doi: 10.1016/S0896-6273(01)00285-9

Fuster, J. M. (2004). Upper processing stages of the perception-action cycle. *Trends Cogn. Sci.* 8, 143–145. doi: 10.1016/j.tics.2004.02.004

Gal, Y. (2016). Dropout as a Bayesian approximation: representing model uncertainty in deep learning. ArXiv:1506.02142v6, 1–12.

Gardner, E. (1988). The space of interactions in neural network models. *J. Phys. A. Math. Gen.* 21, 257–270. doi: 10.1088/0305-4470/21/1/030

Gardner, M. (2001). *The Colossal Book of Mathematics: Classic Puzzles, Paradoxes, and Problems*. New York, NY: W. W. Norton & Company.

Gasser, M., Eck, D., and Port, R. (1999). Meter as mechanism: a neural network model that learns metrical patterns. *Conn. Sci.* 11, 187–216. doi: 10.1080/095400999116331

Gauci, J., and Stanley, K. O. (2010). Autonomous evolution of topographic regularities in artificial neural networks. *Neural Comput.* 22, 1860–1898. doi: 10.1162/neco.2010.06-09-1042

Gershman, S. J., and Beck, J. M. (2016). "Complex probabilistic inference: from cognition to neural computation," in *Computational Models of Brain and Behavior*, ed A. Moustafa (Chichester, UK: Wiley-Blackwell), 1–17. doi: 10.1002/9781119159193.ch33

Gershman, S. J., Horvitz, E. J., and Tenenbaum, J. B. (2015). Computational rationality: a converging paradigm for intelligence in brains, minds, and machines. *Science* 349, 273–278. doi: 10.1126/science.aac6076

Gerstner, W., and Kistler, W. M. (2002). *Spiking Neuron Models*. Cambridge: Cambridge University Press.

Gerstner, W., Kistler, W. M., Naud, R., and Paninski, L. (2014). *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*. Cambridge: Cambridge University Press. doi: 10.1017/CBO9781107447615

Gibson, J. (1979). *The Ecological Approach to Visual Perception*. Boston, MA: Houghton Mifflin. doi: 10.1002/bs.3830260313

Gigerenzer, G., and Goldstein, D. G. (1996). Reasoning the fast and frugal way: models of bounded rationality. *Psychol. Rev.* 103, 650–669. doi: 10.1037//0033-295x.103.4.650

Gilbert, C. D., and Li, W. (2013). Top-down influences on visual processing. *Nat. Rev. Neurosci.* 14, 350–363. doi: 10.1038/nrn3476

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). Generative adversarial nets. ArXiv:1406.2661v1, 1–9.

Gordon, G., and Ahissar, E. (2012). Hierarchical curiosity loops and active sensing. *Neural Netw.* 32, 119–129. doi: 10.1016/j.neunet.2012.02.024

Graves, A., Wayne, G., and Danihelka, I. (2014). Neural turing machines. ArXiv:1410.5401, 1–26.

Gregor, K., Danihelka, I., Graves, A., and Wierstra, D. (2014). DRAW: a recurrent neural network for image generation. ArXiv:1502.04623v1, 1–16.

Griffiths, T., Chater, N., and Kemp, C. (2010). Probabilistic models of cognition: exploring representations and inductive biases. *Trends Cogn. Sci.* 14, 357–364. doi: 10.1016/j.tics.2010.05.004

Grinstein, G., and Linsker, R. (2007). Comments on a derivation and application of the 'maximum entropy production' principle. *J. Phys. A Math. Theor.* 40, 9717–9720. doi: 10.1088/1751-8113/40/31/n01

Grothe, B. (2003). New roles for synaptic inhibition in sound localization. *Nat. Rev. Neurosci.* 4, 540–550. doi: 10.1038/nrn1136

Güçlü, U., Thielen, J., Hanke, M., and van Gerven, M. A. J. (2016). "Brains on beats," in *Advances in Neural Information Processing Systems 29* (Barcelona), 1–12.

Güçlü, U., and van Gerven, M. (2017a). Increasingly complex representations of natural movies across the dorsal stream are shared between subjects. *Neuroimage* 145, 329–336. doi: 10.1016/j.neuroimage.2015.12.036

Güçlü, U., and van Gerven, M. A. J. (2015). Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. *J. Neurosci.* 35, 10005–10014. doi: 10.1523/JNEUROSCI.5023-14.2015

Güçlü, U., and van Gerven, M. A. J. (2017b). Modeling the dynamics of human brain activity with recurrent neural networks. *Front. Comput. Neurosci.* 11:7. doi: 10.3389/fncom.2017.00007

Güçlütürk, Y., Güçlü, U., Seeliger, K., Bosch, S., van Lier, R., and van Gerven, M. A. J. (2017). "Deep adversarial neural decoding," in *Advances in Neural Information Processing Systems 30* (Long Beach), 1–12.

Güçlütürk, Y., Güçlü, U., van Gerven, M. A. J., and van Lier, R. (2016). "Deep impression: audiovisual deep residual networks for multimodal apparent personality trait recognition," in *Proceedings of the 14th European Conference on Computer Vision* (Amsterdam).

Harari, Y. N. (2017). *Homo Deus: A Brief History of Tomorrow, 1st Edn*. New York, NY: Vintage Books.

Harnad, S. (1990). The symbol grounding problem. *Phys. D Nonlin. Phenom.* 42, 335–346. doi: 10.1016/0167-2789(90)90087-6

Hassabis, D., Kumaran, D., Summerfield, C., and Botvinick, M. (2017). Neuroscience-inspired artificial intelligence. *Neuron* 95, 245–258. doi: 10.1016/j.neuron.2017.06.011

Hatfield, G. (2002). "Perception and the physical world: psychological and philosophical issues in perception," in *Perception and the Physical World: Psychological and Philosophical Issues in Perception*, eds D. Heyer and R. Mausfeld (Hoboken, NJ: John Wiley and Sons), 113–143. doi: 10.1002/0470013427.ch5

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. ArXiv:1512.03385, 1–12.

Heeger, D. J. (2017). Theory of cortical function. *Proc. Natl. Acad. Sci. U.S.A.* 114, 1773–1782. doi: 10.1073/pnas.1619788114

Herculano-Houzel, S., and Lent, R. (2005). Isotropic fractionator: a simple, rapid method for the quantification of total cell and neuron numbers in the brain. *J. Neurosci.* 25, 2518–2521. doi: 10.1523/JNEUROSCI.4526-04.2005

Hertz, J. A., Krogh, A. S., and Palmer, R. G. (1991). *Introduction to the Theory of Neural Computation*. Boulder, CO: Westview Press. doi: 10.1063/1.2810360

Hinton, G. (2013). Where do features come from? *Cogn. Sci.* 38, 1078–1101. doi: 10.1111/cogs.12049

Hinton, G. E., McLelland, J. L., and Rumelhart, D. E. (1986). "Distributed representations," in *Parallel Distributed Processing Explorations in the Microstructure of Cognition*, vol. 1, eds D. E. Rumelhart and J. L. McClelland (Cambridge, MA: MIT Press), 77–109.

Hinton, G. E., Osindero, S., and Teh, Y. (2006). A fast learning algorithm for deep belief nets. *Neural Comput.* 18, 1527–1554. doi: 10.1162/neco.2006.18.7.1527

Hinton, G. E., and Sejnowski, T. J. (1983). "Optimal perceptual inference," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Washington, DC).

Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.* 9, 1735–1780. doi: 10.1162/neco.1997.9.8.1735

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. U.S.A.* 79, 2554–2558. doi: 10.1073/pnas.79.8.2554

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Netw.* 4, 251–257. doi: 10.1016/0893-6080(91)90009-T

Huang, Y., and Rao, R. P. N. (2011). Predictive coding. *WIREs Cogn. Sci.* 2, 580–593. doi: 10.1002/wcs.142

Huh, D., and Sejnowski, T. J. (2017). Gradient descent for spiking neural networks. ArXiv:1706.04698, 1–10.

Huo, J., and Murray, A. (2009). The adaptation of visual and auditory integration in the barn owl superior colliculus with spike timing dependent plasticity. *Neural Netw.* 22, 913–921. doi: 10.1016/j.neunet.2008.10.007

Ijspeert, A. J. (2008). Central pattern generators for locomotion control in animals and robots: a review. *Neural Netw.* 21, 642–653. doi: 10.1016/j.neunet.2008.03.014

Ioffe, S., and Szegedy, C. (2015). Batch normalization: accelerating deep network training by reducing internal covariate shift. ArXiv:1502.03167, 1–11.

Izhikevich, E. M., and Edelman, G. M. (2008). Large-scale model of mammalian thalamocortical systems. *Proc. Natl. Acad. Sci. U.S.A.* 105, 3593–3598. doi: 10.1073/pnas.0712231105

Jaynes, E. (1988). How does the brain do plausible reasoning? *Maximum Entropy Bayesian Methods Sci. Eng.* 1, 1–24. doi: 10.1007/978-94-009-3049-0_1

Jeffress, L. A. (1948). A place theory of sound localization. *J. Comp. Physiol. Psychol.* 41, 35–39. doi: 10.1037/h0061495

Johnson, J., Hariharan, B., van der Maaten, L., Hoffman, J., Fei-Fei, L., Zitnick, C. L., et al. (2017). Inferring and executing programs for visual reasoning. ArXiv:1705.03633.

Jonas, E., and Kording, K. P. (2017). Could a neuroscientist understand a microprocessor? *PloS Comput. Biol.* 13:e1005268. doi: 10.1371/journal.pcbi.1005268

Jordan, M. I. (1987). "Attractor dynamics and parallelism in a connectionist sequential machine," in *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, 531–546.

Jordan, M. I., and Mitchell, T. M. (2015). Machine learning: trends, perspectives, and prospects. *Science* 349, 255–260. doi: 10.1126/science.aaa8415

Joukes, J., Hartmann, T. S., and Krekelberg, B. (2014). Motion detection based on recurrent network dynamics. *Front. Syst. Neurosci.* 8:239. doi: 10.3389/fnsys.2014.00239

Kadmon, J., and Sompolinsky, H. (2016). "Optimal architectures in a solvable model of deep networks," in *Advances in Neural Information Processing Systems 29* (Barcelona), 1–9.

Kaiser, Ł., and Roy, A. (2017). "Learning to remember rare events," in *5th International Conference on Learning Representations* (Toulon), 1–10.

Kanitscheider, I., and Fiete, I. (2016). Training recurrent networks to generate hypotheses about how the brain solves hard navigation problems. ArXiv:1609.09059, 1–10.

Kaplan, F., and Oudeyer, P.-Y. (2004). Maximizing learning progress: an internal reward system for development. *Embodied Artif. Intell.* 3139, 259–270. doi: 10.1007/b99075

Kass, R., Eden, U., and Brown, E. (2014). *Analysis of Neural Data*. New York, NY: Springer. doi: 10.1007/978-1-4614-9602-1

Kawaguchi, K., Kaelbling, L. P., and Bengio, Y. (2017). Generalization in deep learning. ArXiv: 1710.045468v1, 1–15.

Kemp, C., and Tenenbaum, J. B. (2008). The discovery of structural form. *Proc. Natl. Acad. Sci. U.S.A.* 105:10687. doi: 10.1073/pnas.0802631105

Kempka, M., Wydmuch, M., Runc, G., Toczek, J., and Ja, W. (2016). ViZDoom: a Doom-based AI research platform for visual reinforcement learning. ArXiv:1605.02097v2, 1–8.

Kheradpisheh, S. R., Ganjtabesh, M., and Thorpe, S. J. (2016). STDP-based spiking deep neural networks for object recognition. ArXiv:1611.01421v1, 1–16.

Kietzmann, T. C., McClure, P., and Kriegeskorte, N. (2017). Deep neural networks in computational neuroscience. BioRxiv, 1–23. doi: 10.1101/133504

Kindermans, P.-J., Schütt, K. T., Alber, M., Müller, K.-R., and Dähne, S. (2017). PatternNet and PatternLRP – improving the interpretability of neural networks. ArXiv:1705.05598, 1–11.

Kingma, D. P., and Welling, M. (2014). Auto-encoding variational Bayes. ArXiv:1312.6114, 1–14.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., and Rusu, A. A. (2015). Overcoming catastrophic forgetting in neural networks. ArXiv:1612.00796v1, 1–13.

Klyubin, A., Polani, D., and Nehaniv, C. (2005a). Empowerment: a universal agent-centric measure of control. *IEEE Congr. Evol. Comput.* 1, 128–135. doi: 10.1109/CEC.2005.1554676

Klyubin, A. S., Polani, D., and Nehaniv, C. L. (2005b). "All else being equal be empowered," in *Lecture Notes in Computer Science*, Vol. 3630 (Canterbury), 744–753. doi: 10.1007/11553090_75

Koller, D., and Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA: The MIT Press.

Kriegeskorte, N. (2015). Deep neural networks: a new framework for modeling biological vision and brain information processing. *Annu. Rev. Vis. Sci.* 1, 417–446. doi: 10.1146/annurev-vision-082114-035447

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25* (Lake Tahoe), 1106–1114.

Kruschke, J. K. (1992). ALCOVE: an exemplar-based connectionist model of category learning. *Psychol. Rev.* 99, 22–44. doi: 10.1037/0033-295X.99.1.22

Kumaran, D., Hassabis, D., and McClelland, J. L. (2016). What learning systems do intelligent agents need? Complementary learning systems theory updated. *Trends Cogn. Sci.* 20, 512–534. doi: 10.1016/j.tics.2016.05.004

Laird, J. E. (2012). *The SOAR Cognitive Architecture*. Cambridge, MA: The MIT Press.

Laje, R., and Buonomano, D. V. (2013). Robust timing and motor patterns by taming chaos in recurrent neural networks. *Nat. Neurosci.* 16, 925–933. doi: 10.1038/nn.3405

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. (2017). Building machines that learn and think like people. *Behav. Brain Sci*. doi: 10.1017/s0140525x16001837

Larochelle, H., and Hinton, G. E. (2010). "Learning to combine foveal glimpses with a third-order Boltzmann machine," in *Advances in Neural Information Processing Systems 23*, Vol. 40 (Vancouver), 1243–1251.

Laughlin, S. B., and Sejnowski, T. J. (2003). Communication in neuronal networks. *Science* 301, 1870–1874. doi: 10.1126/science.1089662

Le Roux, N., and Bengio, Y. (2010). Deep belief networks are compact universal approximators. *Neural Comput.* 22, 2192–2207. doi: 10.1162/neco.2010.08-09-1081

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature* 521, 436–444. doi: 10.1038/nature14539

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 2278–2324. doi: 10.1109/5.726791

Lee, J. H., Delbruck, T., and Pfeiffer, M. (2016). Training deep spiking neural networks using backpropagation. ArXiv:1608.08782, 1–10.

Lee, T., and Mumford, D. (2003). Hierarchical Bayesian inference in the visual cortex. *J. Opt. Soc. Am. A* 20, 1434–1448. doi: 10.1364/josaa.20.001434

Lehky, S. R., and Tanaka, K. (2016). Neural representation for object recognition in inferotemporal cortex. *Curr. Opin. Neurobiol.* 37, 23–35. doi: 10.1016/j.conb.2015.12.001

Leibo, J. Z., Liao, Q., Anselmi, F., Freiwald, W. A., and Poggio, T. (2017). View-tolerant face recognition and Hebbian learning imply mirror-symmetric neural tuning to head orientation. *Curr. Biol.* 27, 62–67. doi: 10.1016/j.cub.2016.10.015

Levine, S., Finn, C., Darrell, T., and Abbeel, P. (2015). End-to-end training of deep visuomotor policies. ArXiv:1504.00702v1, 1–12.

Liao, Q., and Poggio, T. (2016). Bridging the gaps between residual learning, recurrent neural networks and visual cortex. ArXiv:1604.03640, 1–16.

Lillicrap, T. P., Cownden, D., Tweed, D. B., and Akerman, C. J. (2016). Random feedback weights support learning in deep neural networks. *Nat. Commun.* 7, 1–10. doi: 10.1038/ncomms13276

Lin, H. W., and Tegmark, M. (2016). Why does deep and cheap learning work so well? ArXiv:1608.08225, 1–14.

Lopez, C. M., Mitra, S., Putzeys, J., Raducanu, B., Ballini, M., Andrei, A., et al. (2016). "A 966-electrode neural probe with 384 configurable channels in 0.13μm SOI CMOS," in *Solid State Circuits Conference Dig Technical Papers* (San Francisco, CA), 21–23. doi: 10.1109/ISSCC.2016.7418072

Lotter, W., Kreiman, G., and Cox, D. (2016). Deep predictive coding networks for video prediction and unsupervised learning. ArXiv:1605.08104, 1–12.

Louizos, C., Shalit, U., Mooij, J., Sontag, D., Zemel, R., and Welling, M. (2017). Causal effect inference with deep latent-variable models. ArXiv:1705.08821, 1–12.

Maass, W. (1997). Networks of spiking neurons: the third generation of neural network models. *Neural Netw.* 10, 1659–1671.

Maass, W. (2016). Searching for principles of brain computation. BioRxiv, 1–16.

MacKay, D. J. C. (2003). *Information Theory, Inference and Learning Algorithms*. Cambridge: Cambridge University Press. doi: 10.1108/03684920410534506

Mandt, S., Hoffman, M. D., and Blei, D. M. (2017). Stochastic gradient descent as approximate Bayesian inference. ArXiv:1704.04289v1, 1–30.

Mante, V., Sussillo, D., Shenoy, K. V., and Newsome, W. T. (2013). Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature* 503, 78–84. doi: 10.1038/nature12742

Marblestone, A. H., Wayne, G., and Kording, K. P. (2016). Towards an integration of deep learning and neuroscience. *Front. Comput. Neurosci.* 10:94. doi: 10.3389/fncom.2016.00094

Marcus, G. (2009). How does the mind work? Insights from biology. *Top. Cogn. Sci.* 1, 145–172. doi: 10.1111/j.1756-8765.2008.01007.x

Marder, E. (2015). Understanding brains: details, intuition, and big data. *PLoS Biol.* 13:e1002147. doi: 10.1371/journal.pbio.1002147

Markram, H. (2006). The blue brain project. *Nat. Rev. Neurosci.* 7, 153–160. doi: 10.1038/nrn1848

Markram, H., Meier, K., Lippert, T., Grillner, S., Frackowiak, R., Dehaene, S., et al. (2011). Introducing the human brain project. *Proc. Comput. Sci.* 7, 39–42. doi: 10.1016/j.procs.2011.12.015

Marr, D. (1969). A theory of cerebellar cortex. *J. Physiol.* 202, 437–470. doi: 10.2307/1776957

Marr, D. (1982). *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*.

Cambridge, MA: MIT Press. doi: 10.7551/mitpress/9780262514620.001.0001

Marr, D., and Poggio, T. (1976). *From Understanding Computation to Understanding Neural Circuitry*. Tech. Rep. MIT.

Mathieu, M., Couprie, C., and LeCun, Y. (2016). "Deep multi-scale video prediction beyond mean square error," in *4th International Conference on Learning Representations* (San Juan), 1–14.

Maturana, H., and Varela, F. (1980). *Autopoiesis and Cognition: The Realization of the Living, 1st Edn*. Dordrecht: D. Reidel Publishing Company. doi: 10.1007/978-94-009-8947-4

Maturana, H., and Varela, F. (1987). *The Tree of Knowledge - The Biological Roots of Human Understanding*. London: New Science Library.

McClelland, J. L. (2003). The parallel distributed processing approach to semantic cognition. *Nat. Rev. Neurosci.* 4, 310–322. doi: 10.1038/nrn1076

McClelland, J. L., Botvinick, M. M., Noelle, D. C., Plaut, D. C., Rogers, T. T., Seidenberg, M. S., et al. (2010). Letting structure emerge: connectionist and dynamical systems approaches to cognition. *Trends Cogn. Sci.* 14, 348–356. doi: 10.1016/j.tics.2010.06.002

McCloskey, M., and Cohen, N. J. (1989). Catastrophic inference in connectionist networks: the sequential learning problem. *Psychol. Learn. Motiv.* 24, 109–165. doi: 10.1016/s0079-7421(08)60536-8

McCorduck, P. (2004). *Machines Who Think, 2nd Edn*. Natick, MA: A. K. Peters, Ltd.

Mcintosh, L. T., Maheswaranathan, N., Nayebi, A., Ganguli, S., and Baccus, S. A. (2016). "Deep learning models of the retinal response to natural scenes," in *Advances in Neural Information Processing Systems 29* (Barcelona), 1–9.

Mead, C. (1990). Neuromorphic electronic systems. *Proc. IEEE* 78, 1629–1636 doi: 10.1109/5.58356

Mhaskar, H., Liao, Q., and Poggio, T. (2016). Learning functions: when is deep better than shallow. ArXiv:1603.00988v4, 1–12.

Miconi, T. (2017). Biologically plausible learning in recurrent neural networks for flexible decision tasks. *Elife* 6:e20899. doi: 10.16373/j.cnki.ahr.150049

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). "Efficient estimation of word representations in vector space," in *1st International Conference on Learning Representations* (Scottsdale).

Miller, E. K., and Cohen, J. D. (2001). An integrative theory of prefrontal cortex function. *Annu. Rev. Neurosci.* 24, 167–202. doi: 10.1146/annurev.neuro.24.1.167

Miller, T. (2017). Explanation in artificial intelligence: insights from the social sciences. ArXiv:1706.07269v1, 1–57.

Minsky, M., and Papert, S. (1969). *Perceptrons. An Introduction to Computational Geometry*. Cambridge, MA: MIT Press.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., et al. (2016). Asynchronous methods for deep reinforcement learning. ArXiv:1602.01783, 1–28.

Mnih, V., Heess, N., Graves, A., and Kavukcuoglu, K. (2014). "Recurrent models of visual attention," *Advances in Neural Information Processing Systems 27* (Montreal), 1–9.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., et al. (2015). Human-level control through deep reinforcement learning. *Nature* 518, 529–533. doi: 10.1038/nature14236

Modha, D. S., Ananthanarayanan, R., Esser, S. K., Ndirango, A., Sherbondy, A. J., and Singh, R. (2011). Cognitive computing. *Commun. ACM* 54, 62–71. doi: 10.1145/1978542.1978559

Moravec, H. P. (2000). *Robot: Mere Machine to Transcendent Mind*. New York, NY: Oxford University Press.

Moser, M.-B., Rowland, D. C., and Moser, E. I. (2015). Place cells, grid cells, and memory. *Cold Spring Harb. Perspect. Biol.* 7:a021808 doi: 10.1101/cshperspect.a021808

Moulton, S. T., and Kosslyn, S. M. (2009). Imagining predictions: mental imagery as mental emulation. *Philos. Trans. R. Soc. B* 364, 1273–1280. doi: 10.1098/rstb.2008.0314

Mozer, M. C. (1989). A focused back-propagation algorithm for temporal pattern recognition. *Complex Syst.* 3, 349–381.

Mozer, M. C., and Smolensky, P. (1989). Using relevance to reduce network size automatically. *Conn. Sci.* 1, 3–16. doi: 10.1080/09540098908915626

Mujika, A. (2016). Multi-task learning with deep model based reinforcement learning. ArXiv:1611.01457, 1–11.

Najemnik, J., and Geisler, W. S. (2005). Optimal eye movement strategies in visual search. *Nature* 434, 387–391. doi: 10.1038/nature03390

Nayebi, A., and Ganguli, S. (2016). Biologically inspired protection of deep networks from adversarial attacks. ArXiv:1703.09202v1, 1–11.

Neftci, E., Binas, J., Rutishauser, U., Chicca, E., Indiveri, G., and Douglas, R. J. (2013). Synthesizing cognition in neuromorphic electronic systems. *Proc. Natl. Acad. Sci. U.S.A.* 110, E3468–E3476. doi: 10.1073/pnas.1212083110

Neil, D., Pfeiffer, M., and Liu, S.-C. (2016). Phased LSTM: accelerating recurrent network training for long or event-based sequences. ArXiv:1610.09513v1, 1–9.

Nessler, B., Pfeiffer, M., Buesing, L., and Maass, W. (2013). Bayesian computation emerges in generic cortical microcircuits through spike-timing-dependent plasticity. *PLoS Comput. Biol.* 9:e1003037. doi: 10.1371/journal.pcbi.1003037

Newell, A. (1991). *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press.

Newell, A., and Simon, H. A. (1976). Computer science as empirical inquiry: symbols and search. *Commun. ACM* 19, 113–126. doi: 10.1145/360018.360022

Nguyen, A., Dosovitskiy, A., Yosinski, J., Brox, T., and Clune, J. (2016). Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. ArXiv:1605.09304, 1–29.

Nilsson, N. (2005). Human-level artificial intelligence? Be serious! *AI Mag.* 26, 68–75. doi: 10.1609/aimag.v26i4.1850

Obermayer, K. (1990). A principle for the formation of the spatial structure of cortical feature maps. *Proc. Natl. Acad. Sci. U.S.A.* 87, 8345–8349. doi: 10.1073/pnas.87.21.8345

O'Connor, P., and Welling, M. (2016). Deep spiking networks. ArXiv:1602.08323, 1–10.

Olshausen, B. A., and Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* 381, 607–609. doi: 10.1038/381607a0

O'Reilly, R. (1998). Six principles for biologically based computational models of cortical cognition. *Trends Cogn. Sci.* 2, 1–8. doi: 10.1016/s1364-6613(98)01241-8

O'Reilly, R., Hazy, T., and Herd, S. (2012). "The Leabra cognitive architecture: how to play 20 principles with nature and win!," in *The Oxford Handbook of Cognitive Science*, ed S. E. F. Chipman (Oxford: Oxford University Press), 1–31. doi: 10.1093/oxfordhb/9780199842193.013.8

O'Reilly, R. C., Wyatte, D., Herd, S., Mingus, B., and Jilk, D. J. (2013). Recurrent processing during object recognition. *Front. Psychol.* 4:124. doi: 10.3389/fpsyg.2013.00124

Orhan, A. E., and Ma, W. J. (2016). Probabilistic inference in generic neural networks trained with non-probabilistic feedback. ArXiv:1601.03060v4, 1–30.

Oudeyer, P.-Y. (2007). "Intrinsically motivated machines," in *Lecture Notes Computer Science*, Vol. 4850, eds M. Lungarella, F. Iida, J. Bongard, and R. Pfeifer (Berlin: Springer), 304–314. doi: 10.1007/978-3-540-77296-5_27

Pachitariu, M., Stringer, C., Schröder, S., Dipoppa, M., Rossi, L. F., Carandini, M., et al. (2016). Suite2p: beyond 10,000 neurons with standard two-photon microscopy. BioRxiv, 1–14.

Pakkenberg, B., and Gundersen, H. (1997). Neocortical neuron number in humans: effect of sex and age. *J. Comp. Neurol.* 384, 312–320.

Pakkenberg, B., Pelvig, D., Marner, L., Bundgaard, M., Gundersen, H., Nyengaard, J., et al. (2003). Aging and the human neocortex. *Exp. Gerontol.* 38, 95–99. doi: 10.1016/s0531-5565(02)00151-1

Palatucci, M., Pomerleau, D., Hinton, G. E., and Mitchell, T. (2009). "Zero-shot learning with semantic output codes," in *Advances in Neural Information Processing Systems 22*, eds Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta (Vancouver), 1410–1418.

Pan, S. J., and Fellow, Q. Y. (2009). A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* 22, 1–15. doi: 10.1109/TKDE.2009.191

Pascanu, R., Mikolov, T., and Bengio, Y. (2012). "On the difficulty of training recurrent neural networks," in *Proceedings of the 30th International Conference on Machine Learning* (Atlanta), 1310–1318.

Pascanu, R., Montufar, G., and Bengio, Y. (2013). On the number of response regions of deep feed forward networks with piece-wise linear activations. ArXiv:1312.6098, 1–17.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. (2017). Curiosity-driven exploration by self-supervised prediction. ArXiv:1705.05363, 1–12.

Peelen, M. V., and Downing, P. E. (2017). Category selectivity in human visual cortex: beyond visual object recognition. *Neuropsychologia* 105, 1–7. doi: 10.1016/j.neuropsychologia.2017.03.033

Perunov, N., Marsland, R., and England, J. (2014). Statistical physics of adaptation. ArXiv:1412.1875, 1–24.

Peterson, J. C., Abbott, J. T., and Griffiths, T. L. (2016). Adapting deep network features to capture psychological representations. ArXiv:1608.02164, 1–6.

Pinker, S., and Mehler, J. (eds.). (1988). *Connections and Symbols*. Cambridge, MA: The MIT Press.

Poggio, T. (2012). The levels of understanding framework, revised *Perception* 41, 1017–1023. doi: 10.1068/p7299

Poole, B., Lahiri, S., Raghu, M., Sohl-Dickstein, J., and Ganguli, S. (2016). Exponential expressivity in deep neural networks through transient chaos. ArXiv:1606.05340, 1–16.

Pouget, A., Beck, J. M., Ma, W. J., and Latham, P. E. (2013). Probabilistic brains: knowns and unknowns. *Nat. Neurosci.* 16, 1170–1178. doi: 10.1038/nn.3495

Pritzel, A., Uria, B., Srinivasan, S., Puigdomènech, A., Vinyals, O., Hassabis, D., et al. (2017). Neural episodic control. ArXiv:1703.01988, 1–12.

Quian Quiroga, R., Reddy, L., Kreiman, G., Koch, C., and Fried, I. (2005). Invariant visual representation by single neurons in the human brain. *Nature* 435, 1102–1107. doi: 10.1038/nature03687

Rafler, S. (2011). Generalization of Conway's "Game of Life" to a continuous domain - SmoothLife. ArXiv:1111.1567v2, 1–4.

Raghu, M., Kleinberg, J., Poole, B., Ganguli, S., and Sohl-Dickstein, J. (2016). Survey of expressivity in deep neural networks. ArXiv:1611.08083v1, 1–5.

Raina, R., Madhavan, A., and Ng, A. (2009). "Large-scale deep unsupervised learning using graphics processors," in *Proceedings of the 26th Annual International Conference on Machine Learning* (Montreal), 1–8. doi: 10.1145/1553374.1553486

Rajan, K., Harvey, C. D., and Tank, D. W. (2015). Recurrent network models of sequence generation and memory. *Neuron* 90, 1–15. doi: 10.1016/j.neuron.2016.02.009

Ramsey, F. P. (1926). "Truth and probability," in *The Foundations of Mathematics and other Logical Essays*, ed R. B. Braithwaite (Abingdon: Routledge), 156–198. doi: 10.1007/978-3-319-20451-2_3

Rao, R. P. and Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nat. Neurosci.* 2, 79–87. doi: 10.1038/4580

Real, E., Moore, S., Selle, A., Saxena, S., Suematsu, Y. L., Le, Q., et al. (2016). Large-scale evolution of image classifiers. ArXiv:1703.01041v1, 1–10.

Regan, J. K. O., and Noë, A. (2001). A sensorimotor account of vision and visual consciousness. *Behav. Brain Sci.* 24, 939–1031. doi: 10.1017/s0140525x01000115

Rid, T. (2016). *Rise of the Machines: A Cybernetic History*. New York, NY: W. W. Norton & Company.

Riesenhuber, M., and Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nat. Neurosci.* 2, 1019–1025.

Ritter, H., and Kohonen, T. (1989). Self-organizing semantic maps. *Biol. Cybern.* 61, 241–254. doi: 10.1007/bf00203171

Robinson, L., and Rolls, E. T. (2015). Invariant visual object recognition: biologically plausible approaches. *Biol. Cybern.* 209, 505–535. doi: 10.1007/s00422-015-0658-2

Roelfsema, P. R., and van Ooyen, A. (2005). Attention-gated reinforcement learning of internal representations for classification. *Neural Comput.* 17, 2176–2214. doi: 10.1162/0899766054615699

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol. Rev.* 65, 386–408. doi: 10.1037/h0042519

Rumelhart, D., Hinton, G., and Williams, R. (1986). "Learning internal representations by error propagation," in *Parallel Distributed Processing, Explorations in the Microstructure of Cognition*, eds D. E. Rumelhart and J. L. McClelland (Cambridge, MA: MIT Press), 318–362.

Salge, C., Glackin, C., and Polani, D. (2013). Empowerment - An introduction. ArXiv:1310.1863, 1–46.

Salimans, T., Ho, J., Chen, X., and Sutskever, I. (2017). Evolution strategies as a scalable alternative to reinforcement learning. ArXiv:1703.03864v2, 1–12.

Santana, E., and Hotz, G. (2016). Learning a driving simulator. ArXiv:1608.01230, 1–8.

Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., and Lillicrap, T. (2016). One-shot learning with memory-augmented neural networks. ArXiv:1605.06065v1, 1–13.

Santoro, A., Raposo, D., Barrett, D. G. T., Malinowski, M., Pascanu, R., Battaglia, P., et al. (2017). A simple neural network module for relational reasoning. ArXiv:1706.01427v1, 1–16.

Saxe, A., McClelland, J., and Ganguli, S. (2014). "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks Andrew," in *2nd International Conference on Learning Representations* (Banff), 1–22.

Scellier, B., and Bengio, Y. (2017). Equilibrium propagation: bridging the gap between energy-based models and backpropagation. *Front. Comput. Neurosci.* 11:24. doi: 10.3389/fncom.2017.00024

Schaal, S. (1999). Is imitation learning the route to humanoid robots? *Trends Cogn. Sci.* 3, 233–242. doi: 10.1016/s1364-6613(99)01327-3

Schacter, D. L., Addis, D. R., and Buckner, R. L. (2007). Remembering the past to imagine the future: the prospective brain. *Nat. Rev. Neurosci.* 8, 657–661. doi: 10.1038/nrn2213

Schiess, M., Urbanczik, R., and Senn, W. (2016). Somato-dendritic synaptic plasticity and error-backpropagation in active dendrites. *PLoS Comput. Biol.* 12:e1004638. doi: 10.1371/journal.pcbi.1004638

Schmidhuber, J. (1991). "Curious model-building control systems," in *Proceedings of International Joint Conference on Neural Networks*, Vol. 2 (Singapore), 1458–1463. doi: 10.1109/IJCNN.1991.170605

Schmidhuber, J. (2003). "Exploring the predictable," in *Advances in Evolutionary Computing*, eds A. Ghosh and S. Tsutsui (Berlin: Springer), 579–612. doi: 10.1017/CBO9781107415324.004

Schmidhuber, J. (2015). On learning to think: algorithmic information theory for novel combinations of reinforcement learning controllers and recurrent neural world models. ArXiv:1511.09249, 1–36.

Schoenholz, S. S., Gilmer, J., Ganguli, S., and Sohl-Dickstein, J. (2017). "Deep information propagation," in *5th International Conference on Learning Representations* (Toulon), 1–18.

Schoenmakers, S., Barth, M., Heskes, T., and van Gerven, M. A. J. (2013). Linear reconstruction of perceived images from human brain activity. *Neuroimage* 83, 951–961. doi: 10.1016/j.neuroimage.2013.07.043

Scholte, H. S., Losch, M. M., Ramakrishnan, K., de Haan, E. H. F., and Bohte, S. M. (2017). Visual pathways from the perspective of cost functions and deep learning. BioRxiv,1–16.

Schroeder, C. E., Wilson, D. A., Radman, T., Scharfman, H., and Lakatos, P. (2010). Dynamics of active sensing and perceptual selection. *Curr. Opin. Neurobiol.* 20, 172–176. doi: 10.1016/j.conb.2010.02.010

Schulman, J., Levine, S., Moritz, P., Jordan, M., and Abbeel, P. (2015). Trust region policy optimization. ArXiv:1502.05477v4, 1–16.

Schultz, W., Dayan, P., and Montague, P. R. (1997). A neural substrate of prediction and reward. *Science* 275, 1593–1599. doi: 10.1126/science.275.5306.1593

Schuman, C. D., Potok, T. E., Patton, R. M., Birdwell, J. D., Dean, M. E., Rose, G. S., et al. (2017). A survey of neuromorphic computing and neural networks in hardware. ArXiv:1705.06963, 1–88.

Searle, J. R. (1980). Minds, brains and Programs. *Behav. Brain Sci.* 3, 417–424. doi: 10.1017/s0140525x00005756

Segundo, J. P., Perkel, D. H., and Moore, G. P. (1966). Spike probability in neurones: influence of temporal structure in the train of synaptic events. *Kybernetik* 3, 67–82. doi: 10.1007/BF00299899

Seising, R. (2017). Marvin Lee Minsky (1927-2016). *Artif. Intell. Med.* 75, 24–31. doi: 10.1016/j.artmed.2016.12.001

Selfridge, O. (1959). "Pandemonium: a paradigm for learning," in *Symposium on the Mechanization of Thought Processes* (Teddington), 513–526.

Shwartz-Ziv, R., and Tishby, N. (2017). Opening the black box of deep neural networks via information. ArXiv:1703.00810, 1–19.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. (2014). "Deterministic policy gradient algorithms," in *2nd International Conference on Learning Representations* (Banff), 387–395.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., et al. (2017). Mastering the game of Go without human knowledge. *Nature* 550, 354–359. doi: 10.1038/nature24270

Simon, H. A. (1962). The architecture of complexity. *Proc. Am. Philos. Soc.* 106, 467–482. doi: 10.1007/978-1-4899-0718-9_31

Simon, H. A. (1996). *The Sciences of the Artificial, 3rd Edn*. Cambridge, MA: The MIT Press.

Singer, W. (2013). Cortical dynamics revisited. *Trends Cogn. Sci.* 17, 616–626. doi: 10.1016/j.tics.2013.09.006

Smolensky, P. (1987). Connectionist AI, symbolic AI, and the brain. *Artif. Intell. Rev.* 1, 95–109. doi: 10.1007/BF00130011

Sompolinsky, H. (1988). Statistical mechanics of neural networks. *Phys. Today* 40, 70–80. doi: 10.1063/1.881142

Sompolinsky, H. (2014). Computational neuroscience: beyond the local circuit. *Curr. Opin. Neurobiol.* 25, 1–6. doi: 10.1016/j.conb.2014.02.002

Song, H. F., Yang, G. R., and Wang, X.-J. (2016). Reward-based training of recurrent neural networks for diverse cognitive and value-based tasks. *Elife* 6, 1–51. doi: 10.1101/070375

Sperry, R. W. (1952). Neurology and the mind-brain problem. *Am. Sci.* 40, 291–312.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15, 1929–1958. doi: 10.1214/12-AOS1000

Stanley, K., and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evol. Comput.* 10, 1–30. doi: 10.1162/106365602320169811

Steels, L. (1993). The artificial life roots of artificial intelligence. *Artif. Life* 1, 75–110. doi: 10.1162/artl.1993.1.1_2.75

Steels, L. (2004). "The autotelic principle," in *Embodied Artificial Intelligence. Lecture Notes in Computer Science*, eds F. Iida, R. Pfeifer, L. Steels, and Y. Kuniyoshi (Berlin; Heidelberg: Springer), 231–242. doi: 10.1007/978-3-540-27833-7_17

Sterling, P. (2012). Allostasis: a model of predictive regulation. *Physiol. Behav.* 106, 5–15. doi: 10.1016/j.physbeh.2011.06.004

Sterling, P., and Laughlin, S. (2016). *Principles of Neural Design*. Cambridge, MA: MIT Press. doi: 10.7551/mitpress/9780262028707.001.0001

Strukov, D. B. (2011). Smart connections. *Nature* 476, 403–405. doi: 10.1038/476403a

Summerfield, C., and de Lange, F. P. (2014). Expectation in perceptual decision making: neural and computational mechanisms. *Nat. Rev. Neurosci.* 15, 745–756. doi: 10.1038/nrn3838

Sun, R. (2004). Desiderata for cognitive architectures. *Philos. Psychol.* 17, 341–373. doi: 10.1080/0951508042000286721

Sun, R., Coward, L. A., and Zenzen, M. J. (2005). On levels of cognitive modeling. *Philos. Psychol.* 18, 613–637. doi: 10.1080/09515080500264248

Sussillo, D., Churchland, M. M., Kaufman, M. T., and Shenoy, K. V. (2015). A neural network that finds a naturalistic solution for the production of muscle activity. *Nat. Neurosci.* 18, 1025–1033. doi: 10.1038/nn.4042

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems 27* (Montreal), 3104–3112.

Sutton, R. S., and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press. doi: 10.1016/j.brainres.2010.09.091

Swanson, L. W. (2000). Cerebral hemisphere regulation of motivated behavior. *Brain Res.* 886, 113–164. doi: 10.1016/s0006-8993(00)02905-x

Swanson, L. W. (2012). *Brain Architecture: Understanding the Basic Plan, 2nd Edn*. Oxford: Oxford University Press.

Synnaeve, G., Nardelli, N., Auvolat, A., Chintala, S., Lacroix, T., Lin, Z., et al. (2016). TorchCraft: a library for machine learning research on real-time strategy games. ArXiv:1611.00625v2, 1–6.

Szigeti, B., Gleeson, P., Vella, M., Khayrulin, S., Palyanov, A., Hokanson, J., et al. (2014). OpenWorm: an open-science approach to modeling Caenorhabditis elegans. *Front. Comput. Neurosci.* 8:137. doi: 10.3389/fncom.2014.00137

Tapaswi, M., Zhu, Y., Stiefelhagen, R., Torralba, A., Urtasun, R., and Fidler, S. (2015). MovieQA: understanding stories in movies through question-answering. ArXiv:1512.02902, 1–10.

Tenenbaum, J. B., Kemp, C., Griffiths, T. L., and Goodman, N. D. (2011). How to grow a mind: statistics, structure, and abstraction. *Science* 331, 1279–1285. doi: 10.1126/science.1192788

Thalmeier, D., Uhlmann, M., Kappen, H. J., Memmesheimer, R.-M., and May, N. C. (2015). Learning universal computations with spikes. ArXiv:1505.07866v1, 1–35.

Thorpe, S. J., and Fabre-Thorpe, M. (2001). Seeking categories in the brain. *Science* 291, 260–262. doi: 10.1126/science.1058249

Thrun, S., and Mitchell, T. M. (1995). Lifelong robot learning. *Robot. Auton. Syst.* 15, 25–46. doi: 10.1016/0921-8890(95)00004-y

Thurstone, L. (1923). The stimulus-response fallacy in psychology. *Psychol. Rev.* 30:354369. doi: 10.1037/h0074251

Tinbergen, N. (1951). *The Study of Instinct*. Oxford: Oxford University Press.

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., and Abbeel, P. (2017). Domain randomization for transferring deep neural networks from simulation to the real world. ArXiv:1703.06907, 1–8.

Todorov, E., Erez, T., and Tassa, Y. (2012). "MuJoCo: a physics engine for model-based control," in *International Conference on Intelligent Robots and Systems* (Vilamoura), 5026–5033. doi: 10.1109/iros.2012.6386109

Todorov, E., and Jordan, M. I. (2002). Optimal feedback control as a theory of motor coordination. *Nat. Neurosci.* 5, 1226–1235. doi: 10.1038/nn963

Tolman, E. (1932). *Purposive Behavior in Animals and Men*. New York, NY: Century.

Torras i Genís, C. (1986). Neural network model with rhythm-assimilation capacity. *IEEE Trans. Syst. Man Cybern.* 16, 680–693. doi: 10.1109/TSMC.1986.289312

Turing, A. M. (1950). Computing machinery and intelligence. *Mind* 49, 433–460. doi: 10.1093/mind/LIX.236.433

Van de Burgt, Y., Lubberman, E., Fuller, E. J., Keene, S. T., Faria, G. C., Agarwal, S., et al. (2017). A non-volatile organic electrochemical device as a low-voltage artificial synapse for neuromorphic computing. *Nat. Mater.* 16, 414–419. doi: 10.1038/NMAT4856

van Gerven, M. A. J. (2017). A primer on encoding models in sensory neuroscience. *J. Math. Psychol.* 76, 172–183. doi: 10.1016/j.jmp.2016.06.009

Vanrullen, R. (2007). The power of the feed-forward sweep. *Adv. Cogn. Psychol.* 3, 167–176. doi: 10.2478/v10053-008-0022-3

Vanrullen, R. (2017). Perception science in the age of deep neural networks. *Front. Psychol.* 8:142. doi: 10.3389/fpsyg.2017.00142

Varshney, L. R., Chen, B. L., Paniagua, E., Hall, D. H., and Chklovskii, D. B. (2011). Structural properties of the Caenorhabditis elegans neuronal network. *PLoS Comput. Biol.* 7:e1001066. doi: 10.1371/journal.pcbi.1001066

Vernon, D., Metta, G., and Sandini, G. (2007). A survey of artificial cognitive systems: implications for the autonomous development of mental capbilities in computational agents. *IEEE Trans. Evol. Comput.* 11, 1–30. doi: 10.1109/TEVC.2006.890274

Vinyals, O., Blundell, C., Lillicrap, T., and Kavukcuoglu, K. (2016). Matching networks for one shot learning. ArXiv:1606.04080v1, 1–12.

Vinyals, O., Brain, G., Fortunato, M., Jaitly, N., and Brain, G. (2017). Pointer networks. ArXiv:1506.03134v2, 1–9.

von Neumann, J. (1966). *Theory of Self-Reproducing Automata*. Champaign, IL: University of Illinois Press.

von Neumann, J., and Morgenstern, O. (1953). *Theory of Games and Economic Behavior, 3rd Edn*. Princeton, NJ: Princeton University Press.

Weichwald, S., Fomina, T., Schölkopf, B., and Grosse-Wentrup, M. (2016). Optimal coding in biological and artificial neural networks. ArXiv:1605.07094v2, 1–10.

Weston, J., Chopra, S., and Bordes, A. (2015). "Memory networks," in *3rd International Conference on Learning Representations* (San Diego), 1–14.

White, R. W. (1959). Motivation reconsidered: the concept of competence. *Psychol. Rev.* 66, 297–333. doi: 10.1037/h0040934

White, S. G., Southgate, E., Thomson, J., and Brenner, S. (1986). The structure of the nervous system of the nematode *C. elegans*. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* 314, 1–340. doi: 10.1098/rstb.1986.0056

Whitehead, S. D., and Ballard, D. H. (1991). Learning to perceive and act by trial and error. *Mach. Learn.* 7, 45–83. doi: 10.1007/bf00058926

Widrow, B., and Lehr, M. A. (1990). 30 Years of adaptive neural networks: perceptron, madaline, and backpropagation. *Proc. IEEE* 78, 1415–1442. doi: 10.1109/5.58323

Wills, T. J., Lever, C., Cacucci, F., Burgess, N., and Keefe, J. O. (2005). Attractor ddynamics in the hippocampal representation of the local environment. *Science* 308, 873–876. doi: 10.1126/science.1108905.Attractor

Willshaw, D. J., Dayan, P., and Morris, R. G. M. (2015). Memory, modelling and Marr: a commentary on Marr (1971) 'Simple memory: A theory of archicortex'. *Philos. Trans. R. Soc. B* 370:20140383. doi: 10.1098/rstb.2014.0383

Winograd, T. (1972). Understanding natural language. *Cogn. Psychol.* 3, 1–191. doi: 10.1016/0010-0285(72)90002-3

Wissner-Gross, A. D., and Freer, C. E. (2013). Causal entropic forces. *Phys. Rev. Lett.* 110:168702. doi: 10.1103/physrevlett.110.168702

Wolfram, S. (2002). *A New Kind of Science*. Champaign, IL: Wolfram Media.

Wu, Y., Zhang, S., Zhang, Y., Bengio, Y., and Salakhutdinov, R. (2016). On multiplicative integration with recurrent neural networks. ArXiv:1606.06630v2, 1–11.

Xue, T., Wu, J., Bouman, K. L., and Freeman, W. T. (2016). Visual dynamics: probabilistic future frame synthesis via cross convolutional networks. ArXiv:1607.02586, 1–11.

Yamins, D. L. K., and DiCarlo, J. J. (2016). Using goal-driven deep learning models to understand sensory cortex. *Nat. Neurosci.* 19, 356–365. doi: 10.1038/nn.4244

Yang, G. R., Song, H. F., Newsome, W. T., and Wang, X. (2017). Clustering and compositionality of task representations in a neural network trained to perform many cognitive tasks. BioRXiv, 1–44. doi: 10.1101/183632

Yang, W., and Yuste, R. (2017). *In vivo* imaging of neural activity. *Nat. Methods* 14, 349–359. doi: 10.1038/nmeth.4230

Yarbus, A. L. (1967). *Eye Movements and Vision*. New York, NY: Plenum.

Yuille, A., and Kersten, D. (2006). Vision as Bayesian inference: analysis by synthesis? *Trends Cogn. Sci.* 10, 301–308. doi: 10.1016/j.tics.2006.05.002

Yuste, R. (2015). From the neuron doctrine to neural networks. *Nat. Rev. Neurosci.* 16, 487–497. doi: 10.1038/nrn3962

Zagoruyko, S., and Komodakis, N. (2017). DiracNets: training very deep neural networks without skip-connections. ArXiv:1706.00388, 1–11.

Zambrano, D., and Bohte, S. M. (2016). Fast and efficient asynchronous neural computation with adapting spiking neural networks. ArXiv:1609.02053, 1–14.

Zenke, F., Poole, B., and Ganguli, S. (2015). Improved multitask learning through synaptic intelligence. ArXiv:1703.04200v2, 1–9.

Zhu, Y., Gordon, D., Kolve, E., and Fox, D. (2017). Visual semantic planning using deep successor representations. ArXiv:1705.08080v1, 1–13.

Zipser, D., and Andersen, R. (1988). A back-propagation programmed network that simulates response properties of a subset of posterior parietal neurons. *Nature* 331, 679–684. doi: 10.1038/331679a0

Conflict of Interest Statement: The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Computational Neuroscience

HYPOTHESIS AND THEORY
published: 14 September 2016
doi: 10.3389/fncom.2016.00094

Toward an Integration of Deep Learning and Neuroscience

Adam H. Marblestone[1]*, *Greg Wayne*[2] *and Konrad P. Kording*[3]

[1] Synthetic Neurobiology Group, Massachusetts Institute of Technology, Media Lab, Cambridge, MA, USA, [2] Google Deepmind, London, UK, [3] Rehabilitation Institute of Chicago, Northwestern University, Chicago, IL, USA

Neuroscience has focused on the detailed implementation of computation, studying neural codes, dynamics and circuits. In machine learning, however, artificial neural networks tend to eschew precisely designed codes, dynamics or circuits in favor of brute force optimization of a cost function, often using simple and relatively uniform initial architectures. Two recent developments have emerged within machine learning that create an opportunity to connect these seemingly divergent perspectives. First, structured architectures are used, including dedicated systems for attention, recursion and various forms of short- and long-term memory storage. Second, cost functions and training procedures have become more complex and are varied across layers and over time. Here we think about the brain in terms of these ideas. We hypothesize that (1) the brain optimizes cost functions, (2) the cost functions are diverse and differ across brain locations and over development, and (3) optimization operates within a pre-structured architecture matched to the computational problems posed by behavior. In support of these hypotheses, we argue that a range of implementations of credit assignment through multiple layers of neurons are compatible with our current knowledge of neural circuitry, and that the brain's specialized systems can be interpreted as enabling efficient optimization for specific problem classes. Such a heterogeneously optimized system, enabled by a series of interacting cost functions, serves to make learning data-efficient and precisely targeted to the needs of the organism. We suggest directions by which neuroscience could seek to refine and test these hypotheses.

Keywords: cost functions, neural networks, neuroscience, cognitive architecture

Edited by:
Sander Bohte,
Centrum Wiskunde & Informatica, Netherlands

Reviewed by:
Mattia Rigotti,
IBM T.J. Watson Research Center, USA
H. Steven Scholte,
University of Amsterdam, Netherlands
Petia D. Koprinkova-Hristova,
Bulgarian Academy of Sciences, Bulgaria

***Correspondence:**
Adam H. Marblestone
adam.h.marblestone@gmail.com

Received: *24 June 2016*
Accepted: *24 August 2016*
Published: *14 September 2016*

Citation:
Marblestone AH, Wayne G and Kording KP (2016) Toward an Integration of Deep Learning and Neuroscience. Front. Comput. Neurosci. 10:94. doi: 10.3389/fncom.2016.00094

1. INTRODUCTION

Machine learning and neuroscience speak different languages today. Brain science has discovered a dazzling array of brain areas (Solari and Stoner, 2011), cell types, molecules, cellular states, and mechanisms for computation and information storage. Machine learning, in contrast, has largely focused on instantiations of a single principle: function optimization. It has found that simple optimization objectives, like minimizing classification error, can lead to the formation of rich internal representations and powerful algorithmic capabilities in multilayer and recurrent networks (LeCun et al., 2015; Schmidhuber, 2015). Here we seek to connect these perspectives.

The artificial neural networks now prominent in machine learning were, of course, originally inspired by neuroscience (McCulloch and Pitts, 1943). While neuroscience has continued to play a role (Cox and Dean, 2014), many of the major developments were guided by insights into the

mathematics of efficient optimization, rather than neuroscientific findings (Sutskever and Martens, 2013). The field has advanced from simple linear systems (Minsky and Papert, 1972), to nonlinear networks (Haykin, 1994), to deep and recurrent networks (LeCun et al., 2015; Schmidhuber, 2015). Backpropagation of error (Werbos, 1974, 1982; Rumelhart et al., 1986) enabled neural networks to be trained efficiently, by providing an efficient means to compute the gradient with respect to the weights of a multi-layer network. Methods of training have improved to include momentum terms, better weight initializations, conjugate gradients and so forth, evolving to the current breed of networks optimized using batch-wise stochastic gradient descent. These developments have little obvious connection to neuroscience.

We will argue here, however, that neuroscience and machine learning are again ripe for convergence. Three aspects of machine learning are particularly important in the context of this paper. First, machine learning has focused on the optimization of cost functions (**Figure 1A**).

Second, recent work in machine learning has started to introduce complex cost functions, those that are not uniform across layers and time, and those that arise from interactions between different parts of a network. For example, introducing the objective of temporal coherence for lower layers (non-uniform cost function over space) improves feature learning (Sermanet and Kavukcuoglu, 2013), cost function schedules (non-uniform cost function over time) improve[1] generalization (Saxe et al., 2013; Goodfellow et al., 2014b; Gülçehre and Bengio, 2016) and adversarial networks—an example of a cost function arising from internal interactions—allow gradient-based training of generative models (Goodfellow et al., 2014a)[2]. Networks that are easier to train are being used to provide "hints" to help bootstrap the training of more powerful networks (Romero et al., 2014).

Third, machine learning has also begun to diversify the architectures that are subject to optimization. It has introduced simple memory cells with multiple persistent states (Hochreiter and Schmidhuber, 1997; Chung et al., 2014), more complex elementary units such as "capsules" and other structures (Delalleau and Bengio, 2011; Hinton et al., 2011; Tang et al., 2012; Livni et al., 2013), content addressable (Graves et al., 2014; Weston et al., 2014) and location addressable memories (Graves et al., 2014), as well as pointers (Kurach et al., 2015) and hard-coded arithmetic operations (Neelakantan et al., 2015).

These three ideas have, so far, not received much attention in neuroscience. We thus formulate these ideas as three hypotheses about the brain, examine evidence for them, and sketch how experiments could test them. But first, let us state the hypotheses more precisely.

[1]Hyper-parameter optimization shows that complicated schedules of training, which differ across parts of the network, lead to optimal performance (Maclaurin et al., 2015).

[2]In adversarial networks, a generator network is trained to fool a discriminator network into being unable to distinguish generated samples from real data samples, while the discriminator network is trained to prevent the generator network from fooling it in this way.

1.1. *Hypothesis 1* – The Brain Optimizes Cost Functions

The central hypothesis for linking the two fields is that biological systems, like many machine-learning systems, are able to optimize cost functions. The idea of cost functions means that neurons in a brain area can somehow change their properties, e.g., the properties of their synapses, so that they get better at doing whatever the cost function defines as their role. Human behavior sometimes approaches optimality in a domain, e.g., during movement (Körding, 2007), which suggests that the brain may have learned optimal strategies. Subjects minimize energy consumption of their movement system (Taylor and Faisal, 2011), and minimize risk and damage to their body, while maximizing financial and movement gains. Computationally, we now know that optimization of trajectories gives rise to elegant solutions for very complex motor tasks (Harris and Wolpert, 1998; Todorov and Jordan, 2002; Mordatch et al., 2012). We suggest that cost function optimization occurs much more generally in shaping the internal representations and processes used by the brain. Importantly, we also suggest that this requires the brain to have mechanisms for efficient credit assignment in multilayer and recurrent networks.

1.2. *Hypothesis 2* – Cost Functions Are Diverse across Areas and Change over Development

A second realization is that cost functions need not be global. Neurons in different brain areas may optimize different things, e.g., the mean squared error of movements, surprise in a visual stimulus, or the allocation of attention. Importantly, such a cost function could be locally generated. For example, neurons could locally evaluate the quality of their statistical model of their inputs (**Figure 1B**). Alternatively, cost functions for one area could be generated by another area. Moreover, cost functions may change over time, e.g., guiding young humans to understanding simple visual contrasts early on, and faces a bit later[3]. This could allow the developing brain to bootstrap more complex knowledge based on simpler knowledge. Cost functions in the brain are likely to be complex and to be arranged to vary across areas and over development.

1.3. *Hypothesis 3* – Specialized Systems Allow Efficient Solution of Key Computational Problems

A third realization is that structure matters. The patterns of information flow seem fundamentally different across brain areas, suggesting that they solve distinct computational problems. Some brain areas are highly recurrent, perhaps making them predestined for short-term memory storage (Wang, 2012). Some areas contain cell types that can switch between qualitatively different states of activation, such as a persistent firing mode vs. a transient firing mode, in response to particular neurotransmitters (Hasselmo, 2006). Other areas, like

[3]Psychologists have been quantifying the subtleties of many such developmental stagings, e.g., of our perceptual and motor performance, e.g., Nardini et al. (2010), Dekker and Nardini (2015), and McKone et al. (2009).

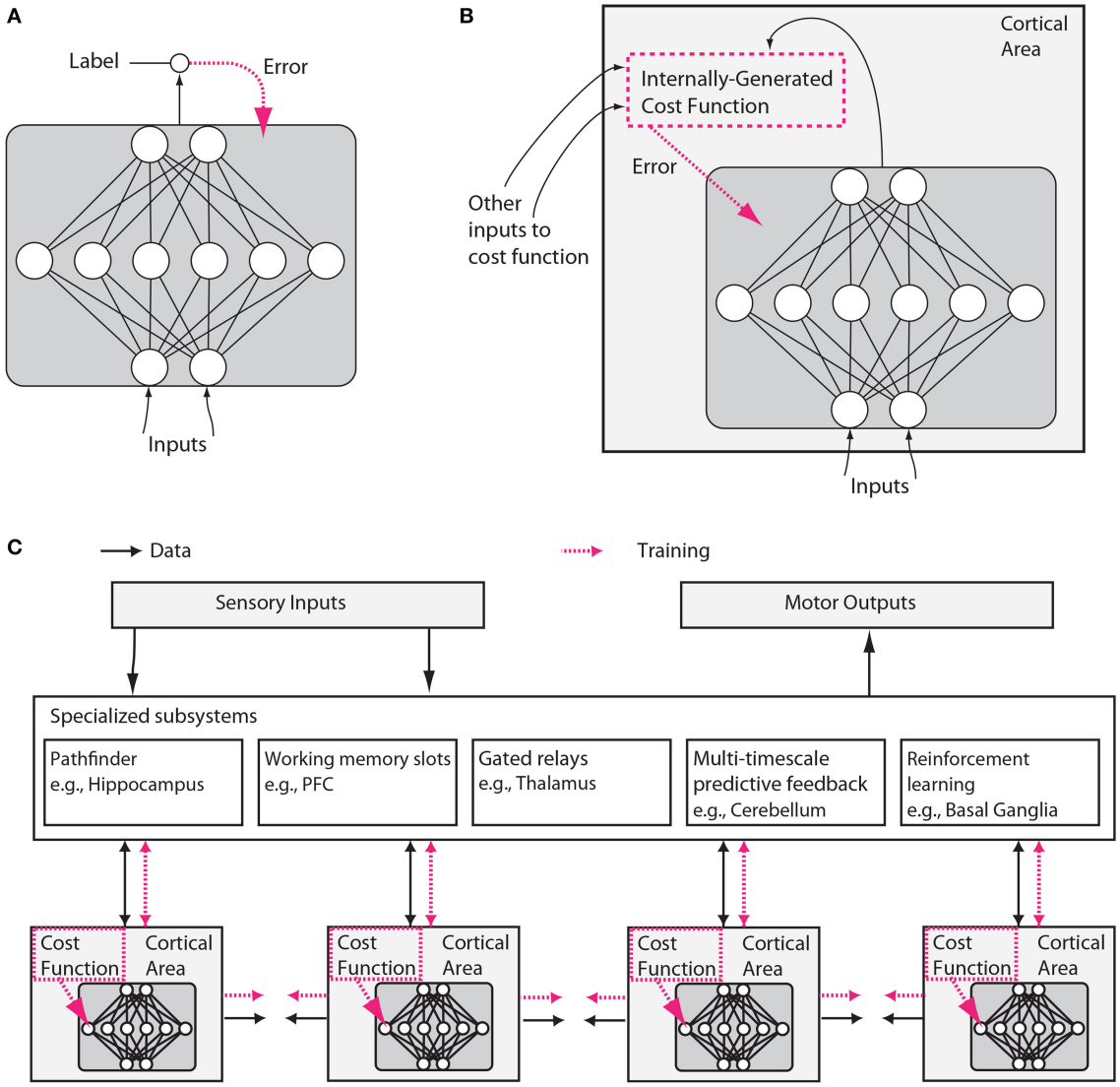


FIGURE 1 | Putative differences between conventional and brain-like neural network designs. (A) In conventional deep learning, supervised training is based on externally-supplied, labeled data. **(B)** In the brain, supervised training of networks can still occur via gradient descent on an error signal, but this error signal must arise from internally generated cost functions. These cost functions are themselves computed by neural modules specified by both genetics and learning. Internally generated cost functions create heuristics that are used to bootstrap more complex learning. For example, an area which recognizes faces might first be trained to detect faces using simple heuristics, like the presence of two dots above a line, and then further trained to discriminate salient facial expressions using representations arising from unsupervised learning and error signals from other brain areas related to social reward processing. **(C)** Internally generated cost functions and error-driven training of cortical deep networks form part of a larger architecture containing several specialized systems. Although the trainable cortical areas are schematized as feedforward neural networks here, LSTMs or other types of recurrent networks may be a more accurate analogy, and many neuronal and network properties such as spiking, dendritic computation, neuromodulation, adaptation and homeostatic plasticity, timing-dependent plasticity, direct electrical connections, transient synaptic dynamics, excitatory/inhibitory balance, spontaneous oscillatory activity, axonal conduction delays (Izhikevich, 2006) and others, will influence what and how such networks learn.

the thalamus appear to have the information from other areas flowing through them, perhaps allowing them to determine information routing (Sherman, 2005). Areas like the basal ganglia are involved in reinforcement learning and gating of discrete decisions (Doya, 1999; Sejnowski and Poizner, 2014). As every programmer knows, specialized algorithms matter for efficient solutions to computational problems, and the brain is likely to make good use of such specialization (**Figure 1C**).

These ideas are inspired by recent advances in machine learning, but we also propose that the brain has major differences from any of today's machine learning techniques. In particular, the world gives us a relatively limited amount of information that we could use for supervised learning (Fodor and Crowther, 2002). There is a huge amount of information available for unsupervised learning, but there is no reason to assume that a *generic* unsupervised algorithm, no matter how powerful, would

learn the precise things that humans need to know, in the order that they need to know it. The evolutionary challenge of making unsupervised learning solve the "right" problems is, therefore, to find a sequence of cost functions that will deterministically build circuits and behaviors according to prescribed developmental stages, so that in the end a relatively small amount of information suffices to produce the right behavior. For example, a developing duck imprints (Tinbergen, 1965) a template of its parent, and then uses that template to generate goal-targets that help it develop other skills like foraging.

Generalizing from this and from other studies (Minsky, 1977; Ullman et al., 2012), we propose that many of the brain's cost functions arise from such an internal bootstrapping process. Indeed, we propose that biological development and reinforcement learning can, in effect, program the emergence of a sequence of cost functions that precisely anticipates the future needs faced by the brain's internal subsystems, as well as by the organism as a whole. This type of developmentally programmed bootstrapping generates an internal infrastructure of cost functions which is diverse and complex, while simplifying the learning problems faced by the brain's internal processes. Beyond simple tasks like familial imprinting, this type of bootstrapping could extend to higher cognition, e.g., internally generated cost functions could train a developing brain to properly access its memory or to organize its actions in ways that will prove to be useful later on. The potential bootstrapping mechanisms that we will consider operate in the context of unsupervised and reinforcement learning, and go well beyond the types of curriculum learning ideas used in today's machine learning (Bengio et al., 2009).

In the rest of this paper, we will elaborate on these hypotheses. First, we will argue that both local and multi-layer optimization is, perhaps surprisingly, compatible with what we know about the brain. Second, we will argue that cost functions differ across brain areas and change over time and describe how cost functions interacting in an orchestrated way could allow bootstrapping of complex function. Third, we will list a broad set of specialized problems that need to be solved by neural computation, and the brain areas that have structure that seems to be matched to a particular computational problem. We then discuss some implications of the above hypotheses for research approaches in neuroscience and machine learning, and sketch a set of experiments to test these hypotheses. Finally, we discuss this architecture from the perspective of evolution.

2. THE BRAIN CAN OPTIMIZE COST FUNCTIONS

Much of machine learning is based on efficiently optimizing functions, and, as we will detail below, the ability to use backpropagation of error (Werbos, 1974; Rumelhart et al., 1986) to calculate gradients of arbitrary parametrized functions has been a key breakthrough. In Hypothesis 1, we claim that the brain is also, at least in part[4], an optimization machine. But what exactly does it mean to say that the brain can optimize cost functions? After all, many processes can be viewed as optimizations. For example, the laws of physics are often viewed as minimizing an action functional, while evolution optimizes the fitness of replicators over a long timescale. To be clear, our main claims are: that (a) the brain has powerful mechanisms for credit assignment during learning that allow it to optimize global functions in multi-layer networks by adjusting the properties of each neuron to contribute to the global outcome, and that (b) the brain has mechanisms to specify exactly which cost functions it subjects its networks to, i.e., that the cost functions are highly tunable, shaped by evolution and matched to the animal's ethological needs. Thus, the brain uses cost functions as a key driving force of its development, much as modern machine learning systems do.

To understand the basis of these claims, we must now delve into the details of how the brain might efficiently perform credit assignment throughout large, multi-layered networks, in order to optimize complex functions. We argue that the brain uses several different types of optimization to solve distinct problems. In some structures, it may use genetic pre-specification of circuits for problems that require only limited learning based on data, or it may exploit local optimization to avoid the need to assign credit through many layers of neurons. It may also use a host of proposed circuit structures that would allow it to actually perform, in effect, backpropagation of errors through a multi-layer network, using biologically realistic mechanisms—a feat that had once been widely believed to be biologically implausible (Crick, 1989; Stork, 1989). Potential such mechanisms include circuits that literally backpropagate error derivatives in the manner of conventional backpropagation, as well as circuits that provide other efficient means of approximating the effects of backpropagation, i.e., of rapidly computing the approximate gradient of a cost function relative to any given connection weight in the network. Lastly, the brain may use algorithms that exploit specific aspects of neurophysiology—such as spike timing dependent plasticity, dendritic computation, local excitatory-inhibitory networks, or other properties—as well as the integrated nature of higher-level brain systems. Such mechanisms promise to allow learning capabilities that go even beyond those of current backpropagation networks.

[4]Our point in this section will not be that all learning in the brain can be captured by cost function optimization, but rather, somewhat more narrowly, our claim is that the algorithms for optimization like backpropagation in deep learning may have correspondences in biological brains. We feel that it is an important task for neuroscience to determine whether and how brains implement these algorithms. The brain may also disclose dynamics that are unlike these algorithms, so we are not disclaiming the possibility of broader theories. In machine learning, many useful algorithms are not explicitly formulated as cost function optimization; for example, many algorithms are based on linear algebra procedures like singular value decomposition, rather than explicit optimization. Such methods can be made nonlinear by using nonlinear kernels—relatedly, some brain circuits run specialized computations using fixed nonlinear basis functions (e.g., in cerebellum). Moreover, while an implicit cost function can be attributed to account for many dynamical processes, as well as many popular learning algorithms, our claim is not merely that the brain uses other learning procedures that lead to solutions which implicitly minimize a cost function, but rather that it actually finds its solutions by performing a powerful form of optimization as such.

2.1. Local Self-organization and Optimization without Multi-layer Credit Assignment

Not all learning requires a general-purpose optimization mechanism like gradient descent[5]. Many theories of cortex (George and Hawkins, 2009; Kappel et al., 2014) emphasize potential self-organizing and unsupervised learning properties that may obviate the need for multi-layer backpropagation as such. Hebbian plasticity, which adjusts weights according to correlations in pre-synaptic and post-synaptic activity, is well established[6]. Various versions of Hebbian plasticity (Miller and MacKay, 1994), e.g., with nonlinearities (Brito and Gerstner, 2016), can give rise to different forms of correlation and competition between neurons, leading to the self-organized formation of ocular dominance columns, self-organizing maps and orientation columns (Miller et al., 1989; Ferster and Miller, 2000). Often these types of local self-organization can also be viewed as optimizing a cost function: for example, certain forms of Hebbian plasticity can be viewed as extracting the principal components of the input, which minimizes a reconstruction error (Pehlevan and Chklovskii, 2015).

To generate complex temporal patterns, the brain may also implement other forms of learning that do not require any equivalent of full backpropagation through a multilayer network. For example, "liquid-" (Maass et al., 2002) or "echo-state machines" (Jaeger and Haas, 2004) are randomly connected recurrent networks that form a basis set (also known as a "reservoir") of random filters, which can be harnessed for learning with tunable readout weights. Variants exhibiting chaotic, spontaneous dynamics can even be trained by feeding back readouts into the network and suppressing the chaotic activity (Sussillo and Abbott, 2009). Learning only the readout layer makes the optimization problem much simpler (indeed, equivalent to regression for supervised learning). Additionally, echo state networks can be trained by reinforcement learning as well as supervised learning (Bush, 2007; Hoerzer et al., 2014). Reservoirs of random nonlinear filters are one interpretation of the diverse, high-dimensional, mixed-selectivity tuning properties of many neurons, e.g., in the prefrontal cortex (Enel et al., 2016). Other variants of learning rules that modify only a fraction of the synapses inside a random network are being developed as models of biological working memory and sequence generation (Rajan et al., 2016).

2.2. Biological Implementation of Optimization

We argue that the above mechanisms of local self-organization are likely insufficient to account for the brain's powerful learning performance (Brea and Gerstner, 2016). To elaborate on the need for an efficient means of gradient computation in the brain, we will first place backpropagation into its computational context (Hinton, 1989; Baldi and Sadowski, 2015). Then we will explain how the brain could plausibly implement approximations of gradient descent.

2.2.1. The Need for Efficient Gradient Descent in Multi-layer Networks

The simplest mechanism to perform cost function optimization is sometimes known as the "twiddle" algorithm or, more technically, as "serial perturbation." This mechanism works by perturbing (i.e., "twiddling"), with a small increment, a single weight in the network, and verifying improvement by measuring whether the cost function has decreased compared to the network's performance with the weight unperturbed. If improvement is noticeable, the perturbation is used as a direction of change to the weight; otherwise, the weight is changed in the opposite direction (or not changed at all). Serial perturbation is therefore a method of "coordinate descent" on the cost, but it is slow and requires global coordination: each synapse in turn is perturbed while others remain fixed.

Weight perturbation (or parallel perturbation) perturbs all of the weights in the network at once. It is able to optimize small networks to perform tasks but generally suffers from high variance. That is, the measurement of the gradient direction is noisy and changes drastically from perturbation to perturbation because a weight's influence on the cost is masked by the changes of all other weights, and there is only one scalar feedback signal indicating the change in the cost[7]. Weight perturbation is dramatically inefficient for large networks. In fact, parallel and serial perturbation learn at approximately the same rate if the time measure counts the number of times the network propagates information from input to output (Werfel et al., 2005).

Some efficiency gain can be achieved by perturbing neural activities instead of synaptic weights, acknowledging the fact that any long-range effect of a synapse is mediated through a neuron. Like weight perturbation and unlike serial perturbation, minimal global coordination is needed: each neuron only needs to receive a feedback signal indicating the global cost. The variance of node perturbation's gradient estimate is far smaller than that of weight perturbation under the assumptions that either all neurons or all weights, respectively, are perturbed and that they are perturbed at the same frequency. In this case, node perturbation's variance is proportional to the number of cells in the network, not the number of synapses.

[5]Of course, some circuits may also be heavily genetically pre-specified to minimize the burden on learning. For instance, particular cell adhesion molecules (Hattori et al., 2007) expressed on particular parts of particular neurons defined by a genetic cell type (Zeisel et al., 2015), and the detailed shapes and placements of neuronal arbors, may constrain connectivity in some cases, though in other cases local connectivity is thought to be only weakly constrained (Kalisman et al., 2005). Genetics is sufficient to specify complex circuits involving hundreds of neurons, such as central pattern generators (Yuste et al., 2005) which create complex self-stabilizing oscillations, or the entire nervous systems of small worms. Genetically guided wiring should not be thought of as fixed "hard-wiring" but rather as a programmatic construction process that can also accept external inputs and interact with learning mechanisms (Marcus, 2004).

[6]Hebbian plasticity even has a well-understood biological basis in the form of the NMDA receptors, which are activated by the simultaneous occurrence of chemical transmitter delivered from the pre-synaptic neuron, and voltage depolarization of the post-synaptic neuron.

[7]The variance can be mitigated by averaging out many perturbations before making a change to the baseline value of the weights, but this would take significant time for a network of non-trivial size as the variance of weight perturbation's estimates scales in proportion to the number of synapses in the network.

All of these approaches are slow either due to the time needed for serial iteration over all weights or the time needed for averaging over low signal-to-noise ratio gradient estimates. To their credit however, none of these approaches requires more than knowledge of local activities and the single global cost signal. Real neural circuits in the brain have mechanisms (e.g., diffusible neuromodulators) that appear to code the signals relevant to implementing those algorithms. In many cases, for example in reinforcement learning, the cost function, which is computed based on interaction with an unknown environment, cannot be differentiated directly, and an agent has no choice but to deploy clever twiddling to explore at some level of the system (Williams, 1992).

Backpropagation, in contrast, works by computing the sensitivity of the cost function to each weight based on the layered structure of the system. The derivatives of the cost function with respect to the last layer can be used to compute the derivatives of the cost function with respect to the penultimate layer, and so on, all the way down to the earliest layers[8]. Backpropagation can be computed rapidly, and for a single input-output pattern, it exhibits no variance in its gradient estimate. The backpropagated gradient has no more noise for a large system than for a small system, so deep and wide architectures with great computational power can be trained efficiently.

2.2.2. Biologically Plausible Approximations of Gradient Descent

To permit biological learning with efficiency approaching that of machine learning methods, some provision for more sophisticated gradient propagation may be suspected. Contrary to what was once a common assumption, there are now many proposed "biologically plausible" mechanisms by which a neural circuit could implement optimization algorithms that, like backpropagation, can efficiently make use of the gradient. These include Generalized Recirculation (O'Reilly, 1996), Contrastive Hebbian Learning (Xie and Seung, 2003), random feedback weights together with synaptic homeostasis (Lillicrap et al., 2014; Liao et al., 2015), spike timing dependent plasticity (STDP) with iterative inference and target propagation (Bengio et al., 2015a; Scellier and Bengio, 2016), complex neurons with backpropagating action-potentials (Körding and König, 2000), and others (Balduzzi et al., 2014). While these mechanisms differ in detail, they all invoke feedback connections that carry error phasically. Learning occurs by comparing a prediction with a target, and the prediction error is used to drive top-down changes in bottom-up activity.

As an example, consider O'Reilly's temporally eXtended Contrastive Attractor Learning (XCAL) algorithm (O'Reilly et al., 2012, 2014b). Suppose we have a multilayer neural network with an input layer, an output layer, and a set of hidden layers in between. O'Reilly showed that the same functionality as backpropagation can be implemented by a bidirectional network with the same weights but symmetric connections. After computing the outputs using the forward connections only, we set the output neurons to the values they should have. The dynamics of the network then cause the hidden layers' activities to evolve toward a stable attractor state linking input to output. The XCAL algorithm performs a type of local modified Hebbian learning at each synapse in the network during this process (O'Reilly et al., 2012). The XCAL Hebbian learning rule compares the local synaptic activity (pre x post) during the early phase of this settling (before the attractor state is reached) to the final phase (once the attractor state has been reached), and adjusts the weights in a way that should make the early phase reflect the later phase more closely. These contrastive Hebbian learning methods even work when the connection weights are not precisely symmetric (O'Reilly, 1996). XCAL has been implemented in biologically plausible conductance-based neurons and basically implements the backpropagation of error approach.

Approximations to backpropagation could also be enabled by the millisecond-scale timing of of neural activities (O'Reilly et al., 2014b). Spike timing dependent plasticity (STDP) (Markram et al., 1997), for example, is a feature of some neurons in which the sign of the synaptic weight change depends on the precise millisecond-scale relative timing of pre-synaptic and post-synaptic spikes. This is conventionally interpreted as Hebbian plasticity that measures the potential for a causal relationship between the pre-synaptic and post-synaptic spikes: a pre-synaptic spike could have contributed to causing a post-synaptic spike only if it occurs shortly beforehand[9]. To enable a backpropagation mechanism, Hinton has suggested an alternative interpretation: that neurons could encode the types of error derivatives needed for backpropagation in the temporal derivatives of their firing rates (Hinton, 2007, 2016). STDP then corresponds to a learning rule that is sensitive to these error derivatives (Xie and Seung, 2000; Bengio et al., 2015b). In other words, in an appropriate network context, STDP learning could give rise to a biological implementation of backpropagation[10].

[8]If the error derivatives of the cost function with respect to the last layer of unit activities are unknown, then they can be replaced with node-perturbation-like correlations, as is common in reinforcement learning.

[9]Interestingly, STDP is not a unitary phenomenon, but rather a diverse collection of different rules with different timescales and temporal asymmetries (Sjöström and Gerstner, 2010; Mishra et al., 2016). Effects include STDP with the inverse temporal asymmetry, symmetric STDP and STDP with different temporal window sizes. STDP is also frequency dependent, which can be explained by rules that depend on triplets rather than pairs of spikes (Pfister and Gerstner, 2006). In some cortical neurons, STDP even switches its sign as the synapse moves away from the neuron's soma into the dendritic tree (Letzkus et al., 2006). While STDP is often included explicitly in models, biophysical derivations of STDP from various underlying phenomena are also being attempted, some of which involve the post-synaptic voltage (Clopath and Gerstner, 2010) or a local dendritic voltage (Urbanczik and Senn, 2014). Meanwhile, other theories suggest that STDP may enable the use of precise timing codes based on temporal coincidence of inputs, the generation and unsupervised learning of temporal sequences (Abbott and Blum, 1996; Fiete et al., 2010), enhancements to distal reward processing in reinforcement learning (Izhikevich, 2007), stabilization of neural responses (Kempter et al., 2001), or many other higher-level properties (Nessler et al., 2013; Kappel et al., 2014).

[10]Hinton has suggested (Hinton, 2007, 2016) that this could take place in the context of autoencoders and recirculation (Hinton and McClelland, 1988). Bengio and colleagues have proposed (Bengio, 2014; Bengio and Fischer, 2015; Scellier and Bengio, 2016) another context in which the connection between STDP and plasticity rules that depend on the temporal derivative of the post-synaptic firing rate can be exploited for biologically plausible multilayer credit assignment. This setting relies on clamping of outputs and stochastic relaxation in energy-based models (Ackley et al., 1958), which leads to a continuous network

Another possible mechanism, by which biological neural networks could approximate backpropagation, is "feedback alignment" (Lillicrap et al., 2014; Liao et al., 2015). There, the feedback pathway in backpropagation, by which error derivatives at a layer are computed from error derivatives at the subsequent layer, is replaced by a set of random feedback connections, with no dependence on the forward weights. Subject to the existence of a synaptic normalization mechanism and approximate sign-concordance between the feedforward and feedback connections (Liao et al., 2015), this mechanism of computing error derivatives works nearly as well as backpropagation on a variety of tasks. In effect, the forward weights are able to adapt to bring the network into a regime in which the random backwards weights actually carry the information that is useful for approximating the gradient. This is a remarkable and surprising finding, and is indicative of the fact that our understanding of gradient descent optimization, and specifically of the mechanisms by which backpropagation itself functions, are still incomplete. In neuroscience, meanwhile, we find feedback connections almost wherever we find feedforward connections, and their role is the subject of diverse theories (Callaway, 2004; Maass et al., 2007). It should be noted that feedback alignment as such does not specify exactly how neurons represent and make use of the error signals; it only relaxes a constraint on the transport of the error signals. Thus, feedback alignment is more a primitive that can be used in fully biological (approximate) implementations of backpropagation, than a fully biological implementation in its own right. As such, it may be possible to incorporate it into several of the other schemes discussed here.

The above "biological" implementations of backpropagation still lack some key aspects of biological realism. For example, in the brain, neurons tend to be either excitatory or inhibitory but not both, whereas in artificial neural networks a single neuron may send both excitatory and inhibitory signals to its downstream neurons. Fortunately, this constraint is unlikely to limit the functions that can be learned (Parisien et al., 2008; Tripp and Eliasmith, 2016). Other biological considerations, however, need to be looked at in more detail: the highly recurrent nature of biological neural networks, which show rich dynamics in time, and the fact that most neurons in mammalian brains communicate via spikes. We now consider these two issues in turn.

2.2.2.1. Temporal credit assignment:

The biological implementations of backpropagation proposed above, while applicable to feedforward networks, do not give a natural implementation of "backpropagation through time" (BPTT) (Werbos, 1990) for recurrent networks, which is widely used in machine learning for training recurrent networks on sequential processing tasks. BPTT "unfolds" a recurrent network across multiple discrete time steps and then runs backpropagation on the unfolded network to assign credit to particular units at particular time steps[11]. While the network unfolding procedure of BPTT itself does not seem biologically plausible, to our intuition, it is unclear to what extent temporal credit assignment is truly needed (Ollivier and Charpiat, 2015) for learning particular temporally extended tasks.

If the system is given access to appropriate memory stores and representations (Buonomano and Merzenich, 1995; Gershman et al., 2012, 2014) of temporal context, this could potentially mitigate the need for temporal credit assignment as such—in effect, memory systems could "spatialize" the problem of temporal credit assignment[12]. For example, memory networks (Weston et al., 2014) store everything by default up to a certain buffer size, eliminating the need to perform credit assignment over the write-to-memory events, such that the network only needs to perform credit assignment over the read-from-memory events. In another example, certain network architectures that are superficially very deep, but which possess particular types of "skip connections," can actually be seen as ensembles of comparatively shallow networks (Veit et al., 2016); applied in the time domain, this could limit the need to propagate errors far backwards in time. Other, similar specializations or higher-levels of structure could, potentially, further ease the burden on credit assignment.

Can generic recurrent networks perform temporal credit assignment in in a way that is more biologically plausible than BPTT? Indeed, new discoveries are being made about the capacity for supervised learning in continuous-time recurrent networks with more realistic synapses and neural integration properties. In internal FORCE learning (Sussillo and Abbott, 2009), internally generated random fluctuations inside a chaotic recurrent network are adjusted to provide feedback signals that drive weight changes internal to the network while the outputs are clamped to desired patterns. This is made possible by a learning procedure that rapidly adjusts the network output to a state where it is close to the clamped values, and exerts continuous control to keep this difference small throughout the learning process[13]. This procedure is able to control and exploit the chaotic dynamical patterns that are spontaneously generated by the network.

Werbos has proposed in his "error critic" that an online approximation to BPTT can be achieved by learning to predict the backward-through-time gradient signal (costate) in a manner analogous to the prediction of value functions in reinforcement

dynamics (Hopfield, 1984) in which hidden units are perturbed toward target values (Bengio and Fischer, 2015), loosely similar to that which occurs in XCAL. This dynamics then allows the STDP-based rule to correspond to gradient descent on the energy function with respect to the weights (Scellier and Bengio, 2016). This scheme requires symmetric weights, but in an autoencoder context, Bengio notes that these can arise spontaneously (Arora et al., 2015).

[11]Even BPTT has arguably not been completely successful in recurrent networks. The problems of vanishing and exploding gradients led to long short term memory networks with gated memory units. An alternative is to use optimization methods that go beyond first order derivatives (Martens and Sutskever, 2011). This suggests the need for specialized systems and structures in the brain to mitigate problems of temporal credit assignment.

[12]Interestingly, the hippocampus seems to "time stamp" memories by encoding them into ensembles with cellular compositions and activity patterns that change gradually as a function of time on the scale of days (Rubin et al., 2015; Cai et al., 2016), and may use "time cells" to mark temporal positions within episodes on a timescale of seconds (Kraus et al., 2013).

[13]Control theory concepts also appear to be useful for simplifying optimization problems in certain other settings (Todorov, 2009; Hennequin et al., 2014).

learning (Werbos and Si, 2004). This kind of idea was recently applied in (Jaderberg et al., 2016) to allow decoupling of different parts of a network during training and to facilitate backpropagation through time. Broadly, we are only beginning to understand how neural activity can itself represent the time variable (Xu et al., 2014; Finnerty et al., 2015)[14], and how recurrent networks can learn to generate trajectories of population activity over time (Liu and Buonomano, 2009). Moreover, as we discuss below, a number of cortical models also propose means, other than BPTT, by which networks could be trained on sequential prediction tasks, even in an online fashion (O'Reilly et al., 2014b; Cui et al., 2015; Brea et al., 2016). A broad range of ideas can be used to approximate BPTT in more realistic ways.

2.2.2.2. Spiking networks:

It has been difficult to apply gradient descent learning directly to spiking neural networks[15,16], although there do exist learning rules for doing so in specific representational contexts and network structures (Bekolay et al., 2013). A number of optimization procedures have been used to generate, indirectly, spiking networks which can perform complex tasks, by performing optimization on a continuous representation of the network dynamics and embedding variables into high-dimensional spaces with many spiking neurons representing each variable (Thalmeier et al., 2015; Abbott et al., 2016; DePasquale et al., 2016; Komer and Eliasmith, 2016). The use of recurrent connections with multiple timescales can remove the need for backpropagation in the direct training of spiking recurrent networks (Bourdoukan and Denève, 2015). Fast connections maintain the network in a state where slow connections have local access to a global error signal. While the biological realism of these methods is still unknown, they all allow connection weights to be learned in spiking networks.

These and other novel learning procedures illustrate the fact that we are only beginning to understand the connections between the temporal dynamics of biologically realistic networks, and mechanisms of temporal and spatial credit assignment. Nevertheless, we argue here that existing evidence suggests that biologically plausible neural networks can solve these problems—in other words, it is possible to efficiently optimize complex functions of temporal history in the context of spiking networks of biologically realistic neurons. In any case, there is little doubt that spiking recurrent networks using realistic population coding schemes can, with an appropriate choice of connection weights, compute complicated, cognitively relevant functions[17]. The question is how the developing brain efficiently learns such complex functions.

2.3. Other Principles for Biological Learning

The brain has mechanisms and structures that could support learning mechanisms different from typical gradient-based optimization algorithms employed in artificial neural networks.

2.3.1. Exploiting Biological Neural Mechanisms

The complex physiology of individual biological neurons may not only help explain how some form of efficient gradient descent could be implemented within the brain, but also could provide mechanisms for learning that go beyond backpropagation. This suggests that the brain may have discovered mechanisms of credit assignment quite different from those dreamt up by machine learning.

One such biological primitive is dendritic computation, which could impact prospects for learning algorithms in several ways. First, real neurons are highly nonlinear (Antic et al., 2010), with the dendrites of each *single* neuron implementing[18] something computationally similar to a three-layer neural network (Mel, 1992)[19]. Individual neurons thus should not be regarded as single "nodes" but as multi-component sub-networks. Second, when a neuron spikes, its action potential propagates back from the soma into the dendritic tree. However, it propagates more strongly into the branches of the dendritic tree that have been active (Williams and Stuart, 2000), potentially simplifying the problem of credit assignment (Körding and König, 2000). Third, neurons can have multiple somewhat independent dendritic compartments, as well as a somewhat independent somatic compartment, which means that the neuron should be thought of as storing more than one variable. Thus, there is the possibility for a neuron to store both its activation itself, and the error derivative of a cost function with respect to its activation, as required in backpropagation, and biological implementations of backpropagation based on this principle have been proposed (Körding and König, 2001; Schiess et al., 2016)[20]. Overall, the implications of dendritic computation for credit assignment in deep networks are only beginning to

[14] In one intriguing study of interval timing, single neurons exhibited response patterns over time which were scaled to the interval duration, and cooling the brain to slow down neural dynamics led to longer intervals being computed by the brain (Xu et al., 2014).

[15] Analogs of weight perturbation and node perturbation are known for spiking networks (Seung, 2003; Fiete and Seung, 2006). Seung (2003) also discusses implications of gradient based learning algorithms for neuroscience, echoing some of our considerations here.

[16] A related, but more general, question is how to learn over many layers of non-differentiable structures. One option is to perform updates via finite-sized rather than infinitesimal steps, e.g., via target-propagation (Bengio, 2014).

[17] Eliasmith and others have shown (Eliasmith and Anderson, 2004; Eliasmith et al., 2012; Eliasmith, 2013) that complex functions and control systems can be compiled onto such networks, using nonlinear encoding and linear decoding of high-dimensional vectors.

[18] Dendritic computation may also have other functions, e.g., competitive interactions between dendrites in a single neuron could also allow neurons to contribute to multiple different ensembles (Legenstein and Maass, 2011).

[19] Localized activity in dendrites drives localized plasticity, with inhibitory interneurons, and interactions between inputs at different parts of the dendritic tree, controlling the local sign and spatial distribution of this plasticity (Sjöström and Häusser, 2006; Cichon and Gan, 2015).

[20] In the model of Körding and König (2001), single spikes are used to transmit activations and burst spikes are used to transmit error information. In other models, including the dendritic voltage in a plasticity rule leads to error-driven and predictive learning that can approximate backpropagation *inside* a single complex neuron (in effect backpropagating from the net somatic output, through nonlinearities at the dendritic branch points, all the way back to the individual input synaptic weights) and that generalize to a reinforcement learning context (Urbanczik and Senn, 2014; Schiess et al., 2016). Single neurons with active dendrites and many synapses may also embody learning rules of greater complexity, such as the storage and recall of temporal patterns (Hawkins and Ahmad, 2016).

be considered[21]. But it is clear that the types of bi-directional, non-linear, multi-variate interactions that are possible *inside* a single neuron could support gradient descent learning or other powerful optimization mechanisms.

Beyond dendritic computation, diverse mechanisms (Marblestone and Boyden, 2014) like retrograde (post-synaptic to pre-synaptic) signals using cannabinoids (Wilson and Nicoll, 2001), or rapidly-diffusing gases such as nitric oxide (Arancio et al., 1996), are among many that could enable learning rules that go beyond conventional conceptions of backpropagation. Harris has suggested (Harris, 2008; Lewis and Harris, 2014) how slow, retroaxonal (i.e., from the outgoing synapses back to the parent cell body) transport of molecules like neurotrophins could allow neural networks to implement an analog of an exchangeable currency in economics, allowing networks to self-organize to efficiently provide information to downstream "consumer" neurons that are trained via faster and more direct error signals. The existence of these diverse mechanisms may call into question traditional, intuitive notions of "biological plausibility" for learning algorithms.

Another potentially important biological primitive is neuromodulation. The same neuron or circuit can exhibit different input-output responses and plasticity depending on a global circuit state, as reflected by the concentrations of various *neuromodulators* like dopamine, serotonin, norepinephrine, acetylcholine, and hundreds of different neuropeptides such as opiods (Bargmann, 2012; Bargmann and Marder, 2013). These modulators interact in complex and cell-type-specific ways to influence circuit function. Interactions with glial cells also play a role in neural signaling and neuromodulation, leading to the concept of "tripartite" synapses that include a glial contribution (Perea et al., 2009). Modulation could have many implications for learning. First, modulators can be used to gate synaptic plasticity on and off selectively in different areas and at different times, allowing precise, rapidly updated orchestration of where and when cost functions are applied. Furthermore, it has been argued that a single neural circuit can be thought of as multiple overlapping circuits with modulation switching between them (Bargmann, 2012; Bargmann and Marder, 2013). In a learning context, this could potentially allow sharing of synaptic weight information between overlapping circuits. Dayan (2012) discusses further computational aspects of neuromodulation. Overall, neuromodulation seems to expand the range of possible algorithms that could be used for optimization.

2.3.2. Learning in the Cortical Sheet

A number of models attempt to explain cortical learning on the basis of specific architectural features of the 6-layered cortical sheet. These models generally agree that a primary function of the cortex is some form of unsupervised learning via prediction (O'Reilly et al., 2014b; Brea et al., 2016)[22]. Some cortical learning models are explicit attempts to map cortical structure onto the framework of message-passing algorithms for Bayesian inference (Lee and Mumford, 2003; Dean, 2005; George and Hawkins, 2009), while others start with particular aspects of cortical neurophysiology and seek to explain those in terms of a learning function, or in terms of a computational function, e.g., hierarchical clustering (Rodriguez et al., 2004). For example, the nonlinear and dynamical properties of cortical pyramidal neurons—the principal excitatory neuron type in cortex (Shepherd, 2014)—are of particular interest here, especially because these neurons have multiple dendritic zones that are targeted by different kinds of projections, which may allow the pyramidal neuron to make comparisons of top-down and bottom-up inputs[23].

Other aspects of the laminar cortical architecture could be crucial to how the brain implements learning. Local inhibitory neurons targeting particular dendritic compartments of the L5

[21] Interestingly, some connectomic studies are finding more obvious connectivity structure at the level of dendritic organization than at the cellular level (Morgan et al., 2016).

[22] An interesting recent study explored this idea in the context of a model of modular cortical-column-like units (Piekniewski et al., 2016). Local units are multi-layer perceptrons trained to minimize a prediction error by gradient descent. Within each unit, predictive autoencoders form a data compression in their middle layers, which is then fed up to higher levels as well as laterally. This system is suggestive of the power of using modular units of intermediate complexity, each of which minimizes a prediction error locally, e.g., in a local few-layer network. The system currently uses a fixed format for transmission of vectors from one unit to another, but ideally the inter-module connections should also be trained by gradient descent as well or by reinforcement learning rather than being fixed. The cortical-column-like modules could also be made more complex and could be organized into higher-order structures like Minsky's semantic networks, frames and K-lines (Minsky, 1988) rather than in simple hierarchies, or such an architecture could self-organize via reinforcement learning or other mechanisms for defining inter-column connections. Such a system also needs connections with specific kinds of memory and long-range information routing systems.

[23] This idea has been used by Hawkins and colleagues to suggest mechanisms for continuous online sequence learning (Cui et al., 2015; Hawkins and Ahmad, 2016) and by Larkum and colleagues for comparison of top-down and bottom-up signals (Larkum, 2013). The Larkum model focuses on the layer 5 (L5) pyramidal neuron type. The cell body of this neuron lies in L5 but extends its "apical" dendritic tree all the way up to a tuft at the top of the cortex in layer 1 (L1), which is a primary target of feedback projections. In the model, interactions between local spiking in these different dendritic zones, which are targeted by different kinds of projections, are crucial to the learning function. The model of Hawkins (Cui et al., 2015; Hawkins and Ahmad, 2016) also focused on the unique dendritic structure of the L5 pyramidal neuron, and distinguishes internal states of the neuron, which impact its responsiveness to other inputs, from activation states, which directly translate into spike rates. Three integration zones in each neuron, and dendritic NMDA spikes (Palmer et al., 2014) acting as local coincidence detectors (Shai et al., 2015), allow temporal patterns of dendritic input to impact the cell's internal state. Intra-column inhibition is also used in this model. Other cortical models pay less attention to the details of dendritic computation, but still provide detailed interpretations of the inter-laminar projection patterns of the neocortex. For example, in O'Reilly et al. (2014b), an architecture is presented for continuous learning based on prediction of the next input. Time is discretized into 100 ms bins via an alpha oscillation, and the deep vs. shallow layers maintain different information during these time bins, with deep layers maintaining a record of the previous time step, and shallow layers representing the current state. The stored information in the deep layers leads to a prediction of the current state, which is then compared with the actual current state. Periodic bursting locked to the oscillation provides a kind of clock that causes the current state to be shifted into the deep layers for maintenance during the subsequent time step, and recurrent loops with the thalamus allow this representation to remain stable for sufficiently long to be used to generate the prediction. Other theories utilize the biophysics of dendritic computation and spike timing dependent plasticity to explain how neurons could learn to make predictions (Brea et al., 2016) on a timescale of seconds using neurons with intrinsic plasticity time constants of a few tens of milliseconds.

pyramidal could be used to exert precise control over when and how the relevant feedback signals and associative mechanisms are utilized. Notably, local inhibitory networks could also give rise to competition (Petrov et al., 2010) between different representations in the cortex, perhaps allowing one cortical column to suppress others nearby, or perhaps even to send more sophisticated messages to gate the state transitions of its neighbors (Bach and Herger, 2015). Moreover, recurrent connectivity with the thalamus, structured bursts of spiking, and cortical oscillations (not to mention other mechanisms like neuromodulation) could control the storage of information over time, to facilitate learning based on temporal prediction. These concepts begin to suggest preliminary, exploratory models for how the detailed anatomy and physiology of the cortex could be interpreted within a machine-learning framework that goes beyond backpropagation. But these are early days: we still lack detailed structural/molecular and functional maps of even a single local cortical microcircuit.

2.3.3. One-shot Learning

Human learning is often one-shot: it can take just a single exposure to a stimulus to never forget it, as well as to generalize from it to new examples. One way of allowing networks to have such properties is what is described by I-theory, in the context of learning invariant representations for object recognition (Anselmi et al., 2015). Instead of training via gradient descent, image templates are stored in the weights of simple-complex cell networks while objects undergo transformations, similar to the use of stored templates in HMAX (Serre et al., 2007). The theories then aim to show that you can invariantly and discriminatively represent objects using a single sample, even of a new class (Anselmi et al., 2015)[24].

Additionally, the nervous system may have a way of quickly storing and replaying sequences of events. This would allow the brain to move an item from episodic memory into a long-term memory stored in the weights of a cortical network (Ji and Wilson, 2007), by replaying the memory over and over. This solution effectively uses many iterations of weight updating to fully learn a single item, even if one has only been exposed to it once. Alternatively, the brain could rapidly store an episodic memory and then retrieve it later without the need to perform slow gradient updates, which has proven to be useful for fast reinforcement learning in scenarios with limited available data (Blundell et al., 2016).

Finally, higher-level systems in the brain may be able to implement Bayesian learning of sequential programs, which is a powerful means of one-shot learning (Lake et al., 2015). This type of cognition likely relies on an interaction between multiple brain areas such as the prefrontal cortex and basal ganglia.

These potential substrates of one-shot learning rely on mechanisms other than simple gradient descent. It should be noted, though, that recent architectural advances, including specialized spatial attention and feedback mechanisms (Rezende et al., 2016), as well as specialized memory mechanisms (Santoro et al., 2016), do allow some types of one-shot generalization to be driven by backpropagation-based learning.

2.3.4. Active Learning

Human learning is often active and deliberate. It seems likely that, in human learning, actions are chosen so as to generate interesting training examples, and sometimes also to test specific hypotheses. Such ideas of active learning and "child as scientist" go back to Piaget and have been elaborated more recently (Gopnik et al., 2000). We want our learning to be based on maximally informative samples, and active querying of the environment (or of internal subsystems) provides a way route to this.

At some level of organization, of course, it would seem useful for a learning system to develop explicit representations of its uncertainty, since this can be used to guide the system to actively seek the information that would reduce its uncertainty most quickly. Moreover, there are population coding mechanisms that could support explicit probabilistic computations (Zemel and Dayan, 1997; Sahani and Dayan, 2003; Rao, 2004; Ma et al., 2006; Eliasmith and Martens, 2011; Gershman and Beck, 2016). Yet it is unclear to what extent and at what levels the brain uses an explicitly probabilistic framework, or to what extent probabilistic computations are emergent from other learning processes (Orhan and Ma, 2016)[25,26].

[24]I-theory can perhaps be viewed as a generalized alternative paradigm to the online optimization of cost functions via multi-layer gradient descent, as used in deep learning. It exploits similar network architectures as conventional deep learning, e.g., hierarchical convolutional networks for the case of feedforward vision, but rather than backpropagating errors, it uses local circuits and learning rules to store templates against which new inputs are compared. This relies on a theory of generalization in learning based on combinations of tuned units (Poggio and Bizzi, 2004), which has been applied to both vision and motor control. Neurons with the required Gaussian-like tunings to stored templates could be obtained through canonical, local, normalization-based circuits (Kouh and Poggio, 2008), which can also be tweaked to implement other aspects of a vision architecture like softmax operations and pooling.

[25]One alternative picture that contrasts with straightforward cost function optimization emphasizes the types of computation that appear most naturally suited to heterogeneous, stochastic, noisy, continually changing neural circuitry (Maass, 2016). On this view, network plasticity is viewed as a sampling-based approximation to Bayesian inference (Kappel et al., 2015) where transiently changing synapses sample from a posterior distribution of network configurations, rather than as gradient descent on a cost function. This view emphasizes Monte-Carlo sampling procedures, rather than cost function optimization.

[26]Sampling based inference procedures are used widely in Bayesian statistics, and efforts have been made to connect these procedures with circuit-based models of computations (Mansinghka and Jonas, 2014). It currently appears difficult, however, to reconcile generic Marcov Chain Monte Carlo (MCMC) dynamics, which mix slowly, with the fast time scales of human psychophysics. But Bayesian methods are powerful and come with a methodology for model comparison (Ghahramani, 2005). In machine learning, variational Bayesian methods have recently become popular precisely because they are capable of fast though approximate posterior inference (inferring causes from observables), but seem to be powerful enough to create strong models. For example, stochastic gradient descent optimization is beginning to be used for variational Bayesian inference (Kingma and Welling, 2013). Restricted Boltzmann Machines (RBMs) also achieve fast inference in shallow architectures—with only a small number of iterations of mixing required—but they do not mix quickly when stacked into deep hierarchies as deep Boltzmann machines. The greedy, layer-wise pre-training of a deep belief network (Hinton et al., 2006) provides a heuristic way to stack the RBMs by auto-encoding, but these have achieved less competitive results than current variational Bayesian models. The problem of fast inference in MCMC models is the subject of current research, including at the interface with biologically

Standard gradient descent does not incorporate any such adaptive sampling mechanism, e.g., it does not deliberately sample data so as to maximally reduce its uncertainty. Interestingly, however, stochastic gradient descent can be used to generate a system that samples adaptively (Alain et al., 2015; Bouchard et al., 2015). In other words, a system can learn, by gradient descent, how to choose its own input data samples in order to learn most quickly from them by gradient descent.

Ideally, the learner learns to choose actions that will lead to the largest improvements in its prediction or data compression performance (Schmidhuber, 2010). In Schmidhuber (2010), this is done in the framework of reinforcement learning, and incorporates a mechanisms for the system to measure its own rate of learning. In other words, it is possible to reinforcement-learn a policy for selecting the most interesting inputs to drive learning. Adaptive sampling methods are also known in reinforcement learning that can achieve optimal Bayesian exploration of Markov Decision Process environments (Sun et al., 2011; Guez et al., 2012).

These approaches achieve optimality in an arbitrary, abstract environment. But of course, evolution may also encode its implicit knowledge of the organism's natural environment, the behavioral goals of the organism, and the developmental stages and processes which occur inside the organism, as priors or heuristics[27] which would further constrain the types of adaptive sampling that are optimal in practice. For example, simple heuristics like seeking certain perceptual signatures of novelty, or more complex heuristics like monitoring situations that other people seem to find interesting, might be good ways to bias sampling of the environment so as to learn more quickly. Other such heuristics might be used to give internal brain systems the types of training data that will be most useful to those particular systems at any given developmental stage.

We are only beginning to understand how active learning might be implemented in the brain. We speculate that multiple mechanisms, specialized to different brain systems and spatio-temporal scales, could be involved. The above examples suggest that at least some such mechanisms could be understood from the perspective of optimizing cost functions.

2.4. Differing Biological Requirements for Supervised and Reinforcement Learning

We have suggested ways in which the brain could implement learning mechanisms of comparable power to backpropagation. But in many cases, the system may be more limited by the available training signals than by the optimization process itself. In machine learning, one distinguishes supervised learning, reinforcement learning and unsupervised learning, and the training data limitation manifests differently in each case.

Both supervised and reinforcement learning require some form of teaching signal, but the nature of the teaching signal in supervised learning is different from that in reinforcement learning. In supervised learning, the trainer provides the entire vector of errors for the output layer and these are back-propagated to compute the gradient: a locally optimal direction in which to update all of the weights of a potentially multi-layer and/or recurrent network. In reinforcement learning, however, the trainer provides a scalar evaluation signal, but this is not sufficient to derive a low-variance gradient. Hence, some form of trial and error twiddling must be used to discover how to increase the evaluation signal. Consequently, reinforcement learning is generally much less efficient than supervised learning.

Reinforcement learning in shallow networks is simple to implement biologically. For reinforcement learning of a deep network to be biologically plausible, however, we need a more powerful learning mechanism, since we are learning based on a more limited evaluation signal than in the supervised case: we do not have the full target pattern to train toward. Nevertheless, approximations of gradient descent can be achieved in this case, and there are cases in which the scalar evaluation signal of reinforcement learning can be used to efficiently update a multi-layer network by gradient descent. The "attention-gated reinforcement learning" (AGREL) networks of Stanisor et al. (2013), Brosch et al. (2015), and Roelfsema and van Ooyen (2005), and variants like KickBack (Balduzzi, 2014), give a way to compute an approximation to the full gradient in a reinforcement learning context using a feedback-based attention mechanism for credit assignment within the multi-layer network. The feedback pathway, together with a diffusible reward signal, together gate plasticity. For networks with more than three layers, this gives rise to a model based on columns containing parallel feedforward and feedback pathways (Roelfsema and van Ooyen, 2005), and for recurrent networks that settle into attractor states it gives a reinforcement-trained version (Brosch et al., 2015) of the Almeida/Pineda recurrent backpropagation algorithm (Pineda, 1987). The process is still not as efficient or generic as backpropagation, but it seems that this form of feedback can make reinforcement learning in multi-layer networks more efficient than a naive node perturbation or weight perturbation approach.

The machine-learning field has recently been tackling the question of credit assignment in deep reinforcement learning. Deep Q-learning (Mnih et al., 2015) demonstrates reinforcement learning in a deep network, wherein most of the network is trained via backpropagation. In regular Q learning, we define a function Q, which estimates the best possible sum of future rewards (the return) if we are in a given state and take a given action. In deep Q learning, this function is approximated by a neural network that, in effect, estimates action-dependent returns in a given state. The network is trained using backpropagation of local errors in Q estimation, using the fact that the return decomposes into the current reward plus the discounted estimate of future return at the next moment. During training, as the agent acts in the environment, a series of loss functions is generated at each step, defining target patterns that can be used as the supervision signal for backpropagation. As Q is a highly nonlinear function of the state, tricks are needed to make deep Q learning efficient and stable, including experience replay and

plausible models (Bengio et al., 2016). When these models are made to perform fast inference, they actually become somewhat similar to variational Bayesian methods, since they rely on feedforward approximate inference, at least to initialize the system.

[27] Heuristics are widely used to simplify motor planning and control, e.g., McLeod and Dienes (1996).

a particular type of mini-batch training. It is also necessary to store the outputs from the previous iteration (or clone the entire network) in evaluating the loss function for the subsequent iteration[28].

This process for generating learning targets provides a kind of bridge between reinforcement learning and efficient backpropagation-based gradient descent learning[29]. Importantly, only temporally local information is needed making the approach relatively compatible with what we know about the nervous system.

Even given these advances, a key remaining issue in reinforcement learning is the problem of long timescales, e.g., learning the many small steps needed to navigate from London to Chicago. Many of the formal guarantees of reinforcement learning (Williams and Baird, 1993), for example, suggest that the difference between an optimal policy and the learned policy becomes increasingly loose as the discount factor shifts to take into account reward at longer timescales. Although the degree of optimality of human behavior is unknown, people routinely engage in adaptive behaviors that can take hours or longer to carry out, by using specialized processes like *prospective memory* to "remember to remember" relevant variables at the right times, permitting extremely long timescales of coherent action. Machine learning has not yet developed methods to deal with such a wide range of timescales and scopes of hierarchical action. Below we discuss ideas of hierarchical reinforcement learning that may make use of callable procedures and sub-routines, rather than operating explicitly in a time domain.

As we will discuss below, some form of deep reinforcement learning may be used by the brain for purposes beyond optimizing global rewards, including the training of local networks based on diverse internally generated cost functions. Scalar reinforcement-like signals are easy to compute, and easy to deliver to other areas, making them attractive mechanistically. If the brain does employ internally computed scalar reward-like signals as a basis for cost functions, it seems likely that it will have found an efficient means of reinforcement-based training of deep networks, but it is an open question whether an analog of deep Q networks, AGREL, or some other mechanism entirely, is used in the brain for this purpose. Moreover, as we will discuss further below, it is possible that reinforcement-type learning is made more efficient in the context of specialized brain systems like short term memories, replay mechanisms, and hierarchically organized control systems. These specialized systems could reduce reliance on a need for powerful credit assignment mechanisms for reinforcement learning. Finally, if the brain uses a diversity of scalar reward-like signals to implement different cost functions, then it may need to mediate delivery of those signals via a comparable diversity of molecular substrates. The great diversity of neuromodulatory signals, e.g., neuropeptides, in the brain (Bargmann, 2012; Bargmann and Marder, 2013) makes such diversity quite plausible, and moreover, the brain may have found other, as yet unknown, mechanisms of diversifying reward-like signaling pathways and enabling them to act independently of one another.

3. THE COST FUNCTIONS ARE DIVERSE ACROSS BRAIN AREAS AND TIME

In the last section, we argued that the brain can optimize functions. This raises the question of what functions it optimizes. Of course, in the brain, a cost function will itself be created (explicitly or implicitly) by a neural network shaped by the genome. Thus, the cost function used to train a given sub-network in the brain is a key innate property that can be built into the system by evolution. It may be much cheaper in biological terms to specify a cost function that allows the rapid learning of the solution to a problem than to specify the solution itself.

In Hypothesis 2, we proposed that the brain optimizes not a single "end-to-end" cost function, but rather a diversity of internally generated cost functions specific to particular brain functions[30]. To understand how and why the brain may use a diversity of cost functions, it is important to distinguish the differing types of cost functions that would be needed for supervised, unsupervised and reinforcement learning. We can also seek to identify types of cost functions that the brain may need to generate from a functional perspective, and how each may be implemented as supervised, unsupervised, reinforcement-based or hybrid systems.

3.1. How Cost Functions May Be Represented and Applied

What additional circuitry is required to actually impose a cost function on an optimizing network? In the most familiar case, supervised learning may rely on computing a vector of errors at the output of a network, which will rely on some comparator circuitry[31] to compute the difference between the network outputs and the target values. This difference could then be backpropagated to earlier layers. An alternative way to impose a cost function is to "clamp" the output of the network, forcing it to occupy a desired target state. Such clamping is actually assumed in some of the putative biological implementations of backpropagation described above, such as XCAL and target propagation. Alternatively, as described above, scalar reinforcement signals are attractive as internally-computed cost functions, but using them in deep networks requires special mechanisms for credit assignment.

[28]Many other reinforcement learning algorithms, including REINFORCE (Williams, 1992), can be implemented as fully online algorithms using "eligibility traces," which accumulate the sensitivity of action distributions to parameters in a temporally local manner (Sutton and Barto, 1998).

[29]Zaremba and Sutskever (2015) also bridges reinforcement learning and backpropagation learning in the same system, in the context of a neural network controlling discrete interfaces, and illustrates some of the challenges of this approach: compared to an end-to-end backpropagation-trained Neural Turing Machine (Graves et al., 2014), reinforcement based training allows training of only relatively simple algorithmic tasks. Special measures need to be taken to make reinforcement efficient, including limiting the number of possible actions, subtracting a baseline reward, and training the network using a curriculum schedule.

[30]This is distinct from a game-theoretic scenario in which multiple actors can achieve an equilibrium, e.g., Gemp and Mahadevan (2015).

[31]Single neurons act as comparators in the motor system, e.g., Brownstone et al. (2015), and networks in the retina adapt so as to report local differences in space or time rather than absolute values, a form of predictive coding (Hosoya et al., 2005).

In unsupervised learning, cost functions may not take the form of externally supplied training or error signals, but rather can be built into the dynamics inherent to the network itself, i.e., there may be no need for a *separate* circuit to compute and impose a cost function on the network. For example, specific spike-timing-dependent and homeostatic plasticity rules have been shown to give rise to gradient descent on a prediction error in recurrent neural networks (Galtier and Wainrib, 2013). Thus, specific unsupervised objectives could be implemented implicitly through specific local network dynamics[32] and plasticity rules inside a network without explicit computation of cost function, nor explicit propagation of error derivatives.

Alternatively, explicit cost functions could be computed, delivered to an optimizing network, and used for unsupervised learning, following a variety of principles being discovered in machine learning (e.g., Radford et al., 2015; Lotter et al., 2015). These networks rely on backpropagation as the sole learning rule, and typically find a way to encode the desired cost function into the error derivatives which are backpropagated. For example, prediction errors naturally give rise to error signals for unsupervised learning, as do reconstruction errors in autoencoders, and these error signals can also be augmented with additional penalty or regularization terms that enforce objectives like sparsity or continuity, as described below. Then these error derivatives can be propagated throughout the network via standard backpropagation. In such systems, the objective function and the optimization mechanism can thus be mixed and matched modularly. In the next sections, we elaborate on these and other means of specifying and delivering cost functions in different learning contexts.

3.2. Cost Functions for Unsupervised Learning

There are many objectives that can be optimized in an unsupervised context, to accomplish different kinds of functions or guide a network to form particular kinds of representations.

3.2.1. Matching the Statistics of the Input Data Using Generative Models

In one common form of unsupervised learning, higher brain areas attempt to produce samples that are statistically similar to those actually seen in lower layers. For example, the wake-sleep algorithm (Hinton et al., 1995) requires the sleep mode to sample potential data points whose distribution should then match the observed distribution. Unsupervised pre-training of deep networks is an instance of this (Erhan and Manzagol, 2009), typically making use of a stacked auto-encoder framework. Similarly, in target propagation (Bengio, 2014), a top-down circuit, together with lateral information, has to produce data that directs the local learning of a bottom-up circuit and vice-versa. Ladder autoencoders make use of lateral connections and local noise injection to introduce an unsupervised cost function, based on internal reconstructions, that can be readily combined with supervised cost functions defined on the networks top layer outputs (Valpola, 2015). Compositional generative models generate a scene from discrete combinations of template parts and their transformations (Wang and Yuille, 2014), in effect performing a rendering of a scene based on its structural description. Hinton and colleagues have also proposed cortical "capsules" (Hinton et al., 2011; Tang et al., 2012, 2013) for compositional inverse rendering. The network can thus implement a statistical goal that embodies some understanding of the way that the world produces samples[33].

Learning rules for generative models have historically involved local message passing of a form quite different from backpropagation, e.g., in a multi-stage process that first learns one layer at a time and then fine-tunes via the wake-sleep algorithm (Hinton et al., 2006). Message-passing implementations of probabilistic inference have also been proposed as an explanation and generalization of deep convolutional networks (Chen et al., 2014; Patel et al., 2015). Various mappings of such processes onto neural circuitry have been attempted (George and Hawkins, 2009; Lee and Yuille, 2011; Sountsov and Miller, 2015), and related models (Makin et al., 2013, 2016) have been used to account for optimal multi-sensory integration in the brain. Feedback connections tend to terminate in distinct layers of cortex relative to the feedforward ones (Felleman and Van Essen, 1991; Callaway, 2004) making the idea of separate but interacting networks for recognition and generation potentially attractive[34]. Interestingly,

[32]Beginning with Hopfield's definition of an energy function for inference in certain classes of symmetric network (Hopfield, 1982), researchers have discovered networks with inherent dynamics that implicitly optimizes certain objectives even while the connection weights are fixed, such as statistical reconstruction of the input via stochastic relaxation in Boltzmann machines (Ackley et al., 1958). Fast approximations of some of these inference procedures are perhaps biologically plausible and could rely on dendritic computation (Bengio et al., 2016). Iterative local Hebbian-like *learning* rules are often used to train the *weights* of such networks, without explicitly propagating error derivatives in the manner of backpropagation. In an appropriate network context, many other combinations of network dynamics and plasticity rules can give rise to inference and learning procedures that implicitly descend cost functions in activity space and/or weight space.

[33]Dreams arguably illustrate that the brain uses generative models which also involve selective recall and recombination of episodic memories.

[34]Much is known about the architecture of cortical feedback vs. feedforward connections. For example, canonically, feedforward connections project from superficial cortical layers to layer 4 of the recipient layer, while feedback connections terminate outside layer 4 and often originate in deeper layers. These types of relationships can be used anatomically to define the hierarchical organization of visual areas, as in Felleman and Van Essen (1991), although the original studies were performed in primates and the precise generalization to rodent cortex is not fully clear (Berezovskii et al., 2011), and there may be various alternate or overlapping anatomical pathways (Callaway, 2004), e.g., with some pathways involved in specific functions like gain control, others routed through specific gating mechanisms, and so forth. Advances in connectomics should allow this architecture to be studied more directly. The study of receptive field properties in the visual cortical hierarchy has led to many insights into this hierarchical system. For example, while each neuron in V1 has a classical local receptive field, neural responses at a given location in V1 also depend on visual locations far from the classical receptive field, e.g., through various forms of surround suppression. These studies have allowed an understanding of the spatial scales over which feedback connections operate in the early visual system (Angelucci et al., 2002). In particular, feedback connections are invoked to account for longer-range receptive field interactions, whereas horizontal connections are invoked to account for shorter-range receptive field interactions (Schwabe et al., 2006). Feedforward and feedback pathways are also distinguished dynamically, e.g., by propagating different oscillatory frequencies (Van Kerkoerle et al., 2014; Bastos et al., 2015), and molecularly, e.g., with NMDA receptors playing an important role in feedback processing.

such sub-networks might even be part of the same neuron and map onto "apical" vs. "basal" parts of the dendritic tree (Körding and König, 2001; Urbanczik and Senn, 2014).

Generative models can also be trained via backpropagation. Recent advances have shown how to perform variational approximations to Bayesian inference inside backpropagation-based neural networks (Kingma and Welling, 2013), and how to exploit this to create generative models (Goodfellow et al., 2014a; Gregor et al., 2015; Radford et al., 2015; Eslami et al., 2016). Through either explicitly statistical or gradient descent based learning, the brain can thus obtain a probabilistic model that simulates features of the world.

3.2.2. Cost Functions That Approximate Properties of the World

A perceiving system should exploit statistical regularities in the world that are not present in an arbitrary dataset or input distribution. For example, objects are sparse, at least in certain representations: there are far fewer objects than there are potential places in the world, and of all possible objects there is only a small subset visible at any given time. As such, we know that the output of an object recognition system must have sparse activations. Building the assumption of sparseness into simulated systems replicates a number of representational properties of the early visual system (Olshausen and Field, 1997; Rozell et al., 2008), and indeed the original paper on sparse coding obtained sparsity by gradient descent optimization of a cost function (Olshausen and Field, 1996). A range of unsupervised machine learning techniques, such as the sparse autoencoders (Le et al., 2012) used to discover cats in YouTube videos, build sparseness into neural networks. Building in such spatio-temporal sparseness priors should serve as an "inductive bias" (Mitchell, 1980) that can accelerate learning.

But we know much more about the regularities of objects. As young babies, we already know (Bremner et al., 2015) that objects tend to persist over time. The emergence or disappearance of an object from a region of space is a rare event. Moreover, object locations and configurations tend to be coherent in time. We can formulate this prior knowledge as a cost function, for example by penalizing representations which are not temporally continuous. This idea of continuity is used in a great number of artificial neural networks and related models (Wiskott and Sejnowski, 2002; Földiák, 2008; Mobahi et al., 2009). Imposing continuity within certain models gives rise to aspects of the visual system including complex cells (Körding et al., 2004), specific properties of visual invariance (Isik et al., 2012), and even other representational properties such as the existence of place cells (Wyss et al., 2006; Franzius et al., 2007). Unsupervised learning mechanisms that maximize temporal coherence or slowness are increasingly used in machine learning[35].

We also know that objects tend to undergo predictable sequences of transformations, and it is possible to build this assumption into unsupervised neural learning systems (George and Hawkins, 2009). The minimization of prediction error explains a number of properties of the nervous system (Friston and Stephan, 2007; Huang and Rao, 2011), and biologically plausible theories are available for how cortex could learn using prediction errors by exploiting temporal differences (O'Reilly et al., 2014b) or top-down feedback (George and Hawkins, 2009). In one implementation, a system can simply predict the next input delivered to the system and can then use the difference between the actual next input and the predicted next input as a full vectorial error signal for supervised gradient descent. Thus, rather than optimization of prediction error being implicitly implemented by the network dynamics, the prediction error is used as an explicit cost function in the manner of supervised learning, leading to error derivatives which can be back-propagated. Then, no special learning rules beyond simple backpropagation are needed. This approach has recently been advanced within machine learning (Lotter et al., 2015, 2016). Recently, combining such prediction-based learning with a specific gating mechanism has been shown to lead to unsupervised learning of disentangled representations (Whitney et al., 2016). Neural networks can also be designed to learn to invert spatial transformations (Jaderberg et al., 2015). Statistically describing transformations or sequences is thus an unsupervised way of learning representations.

Furthermore, there are multiple modalities of input to the brain. Each sensory modality is primarily connected to one part of the brain[36]. But higher levels of cortex in each modality are heavily connected to the other modalities. This can enable forms of self-supervised learning: with a developing visual understanding of the world we can predict its sounds, and then test those predictions with the auditory input, and vice versa. The same is true about multiple parts of the same modality: if we understand the left half of the visual field, it tells us an awful lot about the right. Indeed, we can use observations of one part of a visual scene to predict the contents of other parts (Noroozi and Favaro, 2016; van den Oord et al., 2016), and optimize a cost function that reflects the discrepancy. Maximizing mutual information is a natural way of improving learning (Becker and Hinton, 1992; Mohamed and Rezende, 2015), and there are many other ways in which multiple modalities or processing streams could mutually train one

[35]Temporal continuity is exploited in Poggio (2015), which analyzes many properties of deep convolutional networks with respect to their biological plausibility, including their apparent need for large amounts of supervised training data, and concludes that the environment may in fact provide a sufficient number of "implicitly," though not explicitly, labeled examples to train a deep convolutional network for object recognition. Implicit labeling of object identity, in this case, arises from temporal continuity: successive frames of a video are likely to have the same objects in similar places and orientations. This allows the brain to derive an invariant signature of object identity which is independent of transformations like translations and rotations, but which does not yet associate the object with a specific name or label. Once such an invariant signature is established, however, it becomes basically trivial to associate the signature with a label for classification (Anselmi et al., 2015). Poggio (2015) also suggests specific means, in the context of I-theory (Anselmi et al., 2015), by which this training could occur via the storage of image templates using Hebbian mechanisms among simple and complex cells in the visual cortex. Thus, in this model, the brain has used its implicit knowledge of the temporal continuity of object motion to provide a kind of minimal labeling that is sufficient to bootstrap object recognition. Although not formulated as a cost function, this shows how usefully the assumption of temporal continuity could be exploited by the brain.

[36]Although, some multi-sensory integration appears to occur even in the early sensory cortices (Cappe et al., 2012).

another. This way, each modality effectively produces training signals for the others[37]. Evidence from psychophysics suggests that some kind of training via detection of sensory conflicts may be occurring in children (Nardini et al., 2010).

3.3. Cost Functions for Supervised Learning

In what cases might the brain use supervised learning, given that it requires the system to "already know" the exact target pattern to train toward? One possibility is that the brain can store records of states that led to good outcomes. For example, if a baby reaches for a target and misses, and then tries again and successfully hits the target, then the difference in the neural representations of these two tries reflects the direction in which the system should change. The brain could potentially use a comparator circuit to directly compute this vectorial difference in the neural population codes and then apply this difference vector as an error signal.

Another possibility is that the brain uses supervised learning to implement a form of "chunking," i.e., a consolidation of something the brain already knows how to do: routines that are initially learned as multi-step, deliberative procedures could be compiled down to more rapid and automatic functions by using supervised learning to train a network to mimic the overall input-output behavior of the original multi-step process. Such a process is assumed to occur in cognitive models like ACT-R (Servan-Schreiber and Anderson, 1990), and methods for compressing the knowledge in neural networks into smaller networks are also being developed (Ba and Caruana, 2014). Thus supervised learning can be used to train a network to do in "one step" what would otherwise require long-range routing and sequential recruitment of multiple systems.

3.4. Repurposing Reinforcement Learning for Diverse Internal Cost Functions

Certain generalized forms of reinforcement learning may be ubiquitous throughout the brain. Such reinforcement signals may be repurposed to optimize diverse internal cost functions. These internal cost functions could be specified at least in part by genetics.

Some brain systems such as in the striatum appear to learn via some form of temporal difference reinforcement learning (Tesauro, 1995; Foster et al., 2000). This is reinforcement learning based on a global value function (O'Reilly et al., 2014a) that predicts total future reward or utility for the agent. Reward-driven signaling is not restricted to the striatum, and is present even in primary visual cortex (Chubykin et al., 2013; Stanisor et al., 2013). Remarkably, the reward signaling in primary visual cortex is mediated in part by glial cells (Takata et al., 2011), rather than neurons, and involves the neurotransmitter acetylcholine (Chubykin et al., 2013; Hangya et al., 2015). On the other hand, some studies have suggested that visual cortex learns the basics of invariant object recognition in the absence of reward (Li and Dicarlo, 2012), perhaps using reinforcement only for more refined perceptual learning (Roelfsema et al., 2010).

But beyond these well-known global reward signals, we argue that the basic mechanisms of reinforcement learning may be widely re-purposed to train local networks using a variety of internally generated error signals. These internally generated signals may allow a learning system to go beyond what can be learned via standard unsupervised methods, effectively guiding or steering the system to learn specific features or computations (Ullman et al., 2012).

3.4.1. Cost Functions for Bootstrapping Learning in the Human Environment

Special, internally-generated signals are needed specifically for learning problems where standard unsupervised methods—based purely on matching the statistics of the world, or on optimizing simple mathematical objectives like temporal continuity or sparsity—will fail to discover properties of the world which are statistically weak in an objective sense but nevertheless have special significance to the organism (Ullman et al., 2012). Indigo bunting birds, for example, learn a template for the constellations of the night sky long before ever leaving the nest to engage in navigation-dependent tasks (Emlen, 1967). This memory template is directly used to determine the direction of flight during migratory periods, a process that is modulated hormonally so that winter and summer flights are reversed. Learning is therefore a multi-phase process in which navigational cues are memorized prior to the acquisition of motor control.

In humans, we suspect that similar multi-stage bootstrapping processes are arranged to occur. Humans have innate specializations for social learning. We need to be able to read one another's expressions as indicated with hands and faces. Hands are important because they allow us to learn about the set of actions that can be produced by agents (Ullman et al., 2012). Faces are important because they give us insight into what others are thinking. People have intentions and personalities that differ from one another, and their feelings are important. How could we hack together cost functions, built on simple genetically specifiable mechanisms, to make it easier for a learning system to discover such behaviorally relevant variables?

Some preliminary studies are beginning to suggest specific mechanisms and heuristics that humans may be using to bootstrap more sophisticated knowledge. In a groundbreaking study, Ullman et al. (2012) asked how could we explain hands, to a system that does not already know about them, in a cheap way, without the need for labeled training examples? Hands are common in our visual space and have special roles in the scene: they move objects, collect objects, and caress babies. Building these biases into an area specialized to detect hands could guide the right kind of learning, by providing a downstream learning system with many likely positive examples of hands on the basis of innately-stored, heuristic signatures about how hands tend to look or behave (Ullman et al., 2012). Indeed, an internally supervised learning algorithm containing specialized,

[37]Other brain-inspired unsupervised objectives are being developed for unsupervised visual learning. One recent paper (Higgins et al., 2016) uses an objective function that seeks representations of statistically independent factors in images, by introducing a regularization term that pushes the distribution of latent factors learned in a generative model to be close to a unit Gaussian. This is based on a theory that the ventral visual stream is optimized to disentangle factors of variation in images.

hard-coded biases to detect hands, on the basis of their typical motion properties, can be used to bootstrap the training of an image recognition module that learns to recognize hands based on their appearance. Thus, a simple, hard-coded module bootstraps the training of a much more complex algorithm for visual recognition of hands.

Ullman et al. (2012) then further exploits a combination of hand and face detection to bootstrap a predictor for gaze direction, based on the heuristic that faces tend to be looking toward hands. Of course, given a hand detector, it also becomes much easier to train a system for reaching, crawling, and so forth. Efforts are underway in psychology to determine whether the heuristics discovered to be useful computationally are, in fact, being used by human children during learning (Yu and Smith, 2013; Fausey et al., 2016).

Ullman refers to such primitive, inbuilt detectors as innate "proto-concepts" (Ullman et al., 2012). Their broader claim is that such pre-specification of mutual supervision signals can make learning the relevant features of the world far easier, by giving an otherwise unsupervised learner the right kinds of hints or heuristic biases at the right times. Here we call these approximate, heuristic cost functions "bootstrap cost functions." The purpose of the bootstrap cost functions is to reduce the amount of data required to learn a specific feature or task, but at the same time to avoid a need for fully unsupervised learning.

Could the neural circuitry for such a bootstrap hand-detector be pre-specified genetically? The precedent from other organisms is strong: for example, it is famously known that the frog retina contains circuitry sufficient to implement a kind of "bug detector" (Lettvin et al., 1959). Ullman's hand detector, in fact, operates via a simple local optical flow calculation to detect "mover" events. This type of simple, local calculation could potentially be implemented in genetically-specified and/or spontaneously self-organized neural circuitry in the retina or early dorsal visual areas (Bülthoff et al., 1989), perhaps similarly to the frog's "bug detector."

How could we explain faces without any training data? Faces tend to have two dark dots in their upper half, a line in the lower half and tend to be symmetric about a vertical axis. Indeed, we know that babies are very much attracted to things with these generic features of upright faces starting from birth, and that they will acquire face-specific cortical areas[38] in their first few years of life if not earlier (McKone et al., 2009). It is easy to define a local rule that produces a kind of crude face detector (e.g., detecting two dots on top of a horizontal line), and indeed some evidence suggests that the brain can rapidly detect faces without even a single feed-forward pass through the ventral visual stream (Crouzet and Thorpe, 2011). The crude detection of human faces used together with statistical learning should be analogous to semi-supervised learning (Sukhbaatar et al., 2014) and could allow identifying faces with high certainty.

Humans have areas devoted to emotional processing, and the brain seems to embody prior knowledge about the structure of emotional expressions and how they relate to causes in the world: emotions should have specific types of strong couplings to various other higher-level variables such as goal-satisfaction, should be expressed through the face, and so on (Phillips et al., 2002; Skerry and Spelke, 2014; Baillargeon et al., 2016; Lyons and Cheries, 2016). What about agency? It makes sense to describe, when dealing with high-level thinking, other beings as optimizers of their own goal functions. It appears that heuristically specified notions of goals and agency are infused into human psychological development from early infancy and that notions of agency are used to bootstrap heuristics for ethical evaluation (Hamlin et al., 2007; Skerry and Spelke, 2014). Algorithms for establishing more complex, innately-important social relationships such as joint attention are under study (Gao et al., 2014), building upon more primitive proto-concepts like face detectors and Ullman's hand detectors (Ullman et al., 2012). The brain can thus use innate detectors to create cost functions and training procedures to train the next stages of learning. This prior knowledge, encoded into brain structure via evolution, could allow learning signals to come from the right places and to appear developmentally at the right times.

It is intuitive to ask whether this type of bootstrapping poses a kind of "chicken and egg" problem: if the brain already has an inbuilt heuristic hand detector, how can it be used to train a detector that performs any better than those heuristics? After all, isn't a trained system only as good as its training data? The work of Ullman et al. (2012) illustrates why this is not the case. First, the "innate detector" can be used to train a downstream detector that operates based on different cues: for example, based on the spatial and body context of the hand, rather than its motion. Second, once multiple such pathways of detection come into existence, they can be used to improve each other. In Ullman et al. (2012), appearance, body context, and mover motion are all used to bootstrap off of one another, creating a detector that is better than any of its training heuristics. In effect, the innate detectors are used not as supervision signals *per se*, but rather to guide or steer the learning process, enabling it to discover features that would otherwise be difficult. If such affordances can be found in other domains, it seems likely that the brain would make extensive use of them to ensure that developing animals learn the precise patterns of perception and behavior needed to ensure their later survival and reproduction.

Thus, generalizing previous ideas (Ullman et al., 2012; Poggio, 2015), we suggest that the brain uses optimization with respect to internally generated heuristic[39] detection signals

[38] In the visual system, it is still unknown why a clustered spatial pattern of representational categories arises, e.g., a physically localized "area" that seems to correspond to representations of faces (Kanwisher et al., 1997), another area for representations of visual word forms (McCandliss et al., 2003), and so on. It is also unknown why this spatial pattern seems to be largely reproducible across individuals. Some theories are based on bottom-up correlation-based clustering or neuronal competition mechanisms, which generate category-selective regions as a byproduct. Other theories suggest a computational reason for this organization, in the context of I-theory (Anselmi et al., 2015), involving the limited ability to generalize transformation-invariances learned for one class of objects to other classes (Leibo et al., 2015b). Areas for abstract culture-dependent concepts, like the visual word form area, suggest that the decomposition cannot be "purely genetic." But it is conceivable that these areas could at least in part reflect different local cost functions.

[39] Psychologists have postulated other innate heuristics, e.g., in the context of object tracking (Franconeri et al., 2012). That infant object concepts are trainable but only

to bootstrap learning of biologically relevant features which would otherwise be missed by an unsupervised learner. In one possible implementation, such bootstrapping may occur via reinforcement learning, using the outputs of the innate detectors as local reinforcement signals, and perhaps using mechanisms similar to Stanisor et al. (2013), Rombouts et al. (2015), Brosch et al. (2015), and Roelfsema and van Ooyen (2005) to perform reinforcement learning through a multi-layer network. It is also possible that the brain could use such internally generated heuristic detectors in other ways, for example to bias the inputs delivered to an unsupervised learning network toward entities of interest to humans via an attentional process (Joscha Bach, personal communication), to bias hippocampal replay (Kumaran et al., 2016) or other aspects of memory access, or to directly train simple classifiers (Ullman et al., 2012).

3.4.2. Cost Functions for Learning by Imitation and through Social Feedback

It has been widely observed that the capacity for imitation and social learning may be a feature that is uniquely human, and that enables other human traits (Ramachandran, 2000). Humans need to learn more from the environment by than trial and error can provide for, and more than genetically orchestrated internal bootstrapping signals can effectively guide. Hence, babies spend a long time watching adults, especially adults they are attached to Meltzoff (1999), and later use specific kinds of social cues from their parents to shape their development. Babies and children learn about cause and effect through models based on goals, outcomes and agents, not just pure statistical inference. For example, young children make inferences about causality selectively in situations where a human is trying to achieve an outcome (Meltzoff et al., 2012, 2013). Minsky (2006) discusses how we derive not just skills but also goals from our attachment figures, through socially induced emotions like pride and shame. To do all this requires a powerful infrastructure of mental abilities: we must attribute social feedback to particular aspects of our goals or actions, and hence we need to signal to each other positively and negatively, to draw attention to these aspects. Minsky speculates (Minsky, 2006) that the development of such "learning by being told" led to language by selecting for the development of increasingly precise parsing of synatatic structures in relation to our representations of agents and action-plans.

How does this connect with cost functions? The idea of goals is central here, as we need to be able to identify the goals of others, update our own goals based on feedback, and measure the success of actions relative to goals. It has been proposed that human intrinsically use a model based on abstract goal and costs to underpin learning about the social world (Jara-Ettinger et al., 2016). Perhaps we even learn about our "selves" by inferring a model of our own goals and cost functions. Relatedly, machine learning in some settings can infer their cost functions from samples of behavior (Ho and Ermon, 2016).

along certain dimensions (Scholl, 2004) also suggests the notion of a heuristically "guided" or "bootstrapped" learning process in this context.

3.4.3. Cost Functions for Story Generation and Understanding

It has been widely noticed in cognitive science and AI that the generation and understanding of stories are crucial to human cognition. Researchers such as Winston have framed story understanding as the key to human-like intelligence (Winston, 2011). Stories consist of a linear sequence of episodes, in which one episode refers to another through cause and effect relationships, with these relationships often involving the implicit goals of agents. Many other cognitive faculties, such as conceptual grounding of language, could conceivably emerge from an underlying internal representation in terms of stories.

Perhaps the ultimate series of bootstrap cost functions would be those which would direct the brain to utilize its learning networks and specialized systems so as to construct representations that are specifically useful as components of stories, to spontaneously chain these representations together, and to update them through experience and communication. How could such cost functions arise? One possibility is that they are bootstrapped through imitation and communication, where a child learns to mimic the story-telling behavior of others. Another possibility is that useful representations and primitives for stories emerge spontaneously from mechanisms for learning state and action chunking in hierarchical reinforcement learning and planning. Yet another is that stories emerge from learned patterns of saliency-directed memory storage and recall (e.g., Xiong et al., 2016). In addition, priors that direct the developing child's brain to learn about and attend to social agency seem to be important for stories.

In this section, we have seen how cost functions can be specified that could lead to the learning of increasingly sophisticated mental abilities in a biologically plausible manner. Importantly, however, cost functions and optimization are not the whole story. To achieve more complex forms of optimization, e.g., for learning to understand complex patterns of cause and effect over long timescales, to plan and reason prospectively, or to effectively coordinate many widely distributed brain resources, the brain seems to invoke specialized, pre-constructed data structures, algorithms and communication systems, which in turn facilitate specific kinds of optimization. Moreover, optimization occurs in a tightly orchestrated multi-stage process, and specialized, pre-structured brain systems need to be invoked to account for this meta-level of control over when, where and how each optimization problem is set up. We now turn to how these pre-specialized systems may orchestrate and facilitate optimization.

4. OPTIMIZATION OCCURS IN THE CONTEXT OF SPECIALIZED STRUCTURES

Optimization of initially unstructured "blank slate" networks is not sufficient to generate complex cognition in the brain, we argue, even given a diversity of powerful genetically-specified cost functions and local learning rules, as we have posited above. Instead, in Hypothesis 3, we suggest that specialized, pre-structured architectures are needed for at least two purposes.

First, pre-structured architectures are needed to allow the brain to find efficient solutions to certain types of problems. When we write computer code, there are a broad range of algorithms and data structures employed for different purposes: we may use dynamic programming to solve planning problems, trees to efficiently implement nearest neighbor search, or stacks to implement recursion. Having the right kind of algorithm and data structure in place to solve a problem allows it to be solved efficiently, robustly and with a minimum amount of learning or optimization needed. This observation is concordant with the increasing use of pre-specialized architectures and specialized computational components in machine learning (Graves et al., 2014; Weston et al., 2014; Neelakantan et al., 2015). In particular, to enable the learning of efficient computational solutions, the brain may need pre-specialized systems for planning and executing sequential multi-step processes, for accessing memories, and for forming and manipulating compositional and recursive structures[40].

Second, the training of optimization modules may need to be coordinated in a complex and dynamic fashion, including delivering the right training signals and activating the right learning rules in the right places and at the right times. To allow this, the brain may need specialized systems for storing and routing data, and for flexibly routing training signals such as target patterns, training data, reinforcement signals, attention signals, and modulatory signals. These mechanisms may need to be at least partially in place in advance of learning.

Looking at the brain, we indeed seem to find highly conserved structures, e.g., cortex, where it is theorized that a similar type of learning and/or computation is happening in multiple places (Braitenberg and Schutz, 1991; Douglas and Martin, 2004). But we also see a large number of specialized structures, including thalamus, hippocampus, basal ganglia and cerebellum (Solari and Stoner, 2011). These structures evolutionarily pre-date (Lee et al., 2015) the cortex, and hence the cortex may have evolved to work in the context of such specialized mechanisms. For example, the cortex may have evolved as a trainable module for which the training is orchestrated by these older structures.

Even within the cortex itself, microcircuitry within different areas may be specialized: tinkered variations on a common ancestral microcircuit scaffold could potentially allow different cortical areas, such as sensory areas vs. prefrontal areas, to be configured to adopt a number of qualitatively distinct computational and learning configurations (Yuste et al., 2005; Marcus et al., 2014a,b), even while sharing a common gross physical layout and communication interface. Within cortex, over forty distinct cell types—differing in such aspects as dendritic organization, distribution throughout the six cortical layers, connectivity pattern, gene expression, and electrophysiological properties—have already been found (Markram et al., 2015; Zeisel et al., 2015). Central pattern generator circuits provide an example of the kinds of architectures that can be pre-wired into neural microcircuitry, and may have evolutionary relationships with cortical circuits (Yuste et al., 2005). Thus, while the precise degree of architectural specificity of particular cortical regions is still under debate (Marcus et al., 2014a,b), various mechanisms could offer pre-specified heterogeneity.

In this section, we explore the kinds of computational problems for which specialized structures may be useful, and attempt to map these to putative elements within the brain. Our preliminary sketch of a functional decomposition can be viewed as a summary of suggestions for specialized functions that have been made throughout the computational neuroscience literature, and is influenced strongly by the models of O'Reilly, Eliasmith, Grossberg, Marcus, Hayworth and others (Marcus, 2001; O'Reilly, 2006; Eliasmith et al., 2012; Hayworth, 2012; Grossberg, 2013). The correspondence between these models and actual neural circuitry is, of course, still the subject of extensive debate.

Many of the computational and neural concepts sketched here are preliminary and will need to be made more rigorous through future study. Our knowledge of the functions of particular brain areas, and thus our proposed mappings of certain computations onto neuroanatomy, also remains tentative. Finally, it is still far from established which processes in the brain emerge from optimization of cost functions, which emerge from other forms of self-organization, which are pre-structured through genetics and development, and which rely on an interplay of all these mechanisms[41]. Our discussion here should therefore be viewed as a sketch of potential directions for further study.

4.1. Structured Forms of Memory

One of the central elements of computation is memory. Importantly, multiple different kinds of memory are needed (Squire, 2004). For example, we need memory that is stored for a long period of time and that can be retrieved in a number of ways, such as in situations similar to the time when the memory was first stored (content addressable memory). We also need memory that we can keep for a short period of time and that we can rapidly rewrite (working memory). Lastly, we need the kind of implicit memory that we cannot explicitly recall, similar to the kind of memory that is classically learned using

[40]Of course, specialized architecture also enters the picture at the level of the pre-structuring of trainable/optimizable modules themselves. Just as in deep learning, convolutional networks, LSTMs, residual networks and other specific architectures are used to make learning efficient and fast, even though more generic architectures like multilayer perceptrons or generally RNNs are universal function approximators.

[41]It is interesting to consider how standard neural network models of vision would fit into this categorization. Consider convolutional neural networks, for example, with the convolutional filters optimized via supervised backpropagation. This is by no means a completely unstructured prior to backpropagation-based training. Indeed, these networks typically contain max-pooling and normalization layers with fixed computations that are not altered during learning, as well as fixed architectural features such as number and arrangement of layers, size and stride of the sliding window, and so forth. Likewise "hierarchical max-pooling" (HMAX) models (Serre et al., 2007) of the ventral stream are so-named because of these fixed architectural aspects. Thus, in a hypothetical biological implementation of such systems, these aspects would be pre-structured by genetics even if the convolutional weights would be trained via some kind of gradient descent optimization. There are some plausible neural circuits that would implement these standardized normalization and max pooling operations (Kouh and Poggio, 2008). Moreover, in a biological implementation, the machinery necessary to carry out the optimization itself would need to be embodied by appropriate, genetically structured circuitry.

gradient descent on errors, i.e., sculpted into the weight matrix of a neural network.

4.1.1. Content Addressable Memories

Content addressable memories[42] are classic models in neuroscience (Hopfield, 1982). Most simply, they allow us to recognize a situation similar to one that we have seen before, and to "fill in" stored patterns based on partial or noisy information, but they may also be put to use as sub-components of many other functions. Recent research has shown that including such memories allows deep networks to learn to solve problems that previously were out of reach, even of LSTM networks that already have a simpler form of local memory and are already capable of learning long-term dependencies (Graves et al., 2014; Weston et al., 2014). Hippocampal area CA3 may act as an auto-associative memory[43] capable of content-addressable pattern completion, with pattern separation occurring in the dentate gyrus (Rolls, 2013). If no similar pattern is available, an unfamiliar input will be stored as a new memory (Kumaran et al., 2016). Such systems could permit the retrieval of complete memories from partial cues, enabling networks to perform operations similar to database retrieval or to instantiate lookup tables of historical stimulus-response mappings, among numerous other possibilities.

Of course, memory systems may be organized—through cost function optimization or other mechanisms—into higher-order structures. Cost functions might be used to bias memory representations to adopt particular structures, e.g., to be organized into data structures like like Minskys frames and trans-frames (Minsky, 2006).

4.1.2. Working Memory Buffers

Cognitive science has long characterized properties of the working memory. Its capacity is somewhat limited, with the old idea being that verbal working memory has a capacity of "seven plus or minus two" (Miller, 1956), while visual working memory has a capacity of four (Luck and Vogel, 1997) (or, other authors defend, one). There are many models of working memory (O'Reilly and Frank, 2006; Singh and Eliasmith, 2006; Warden and Miller, 2007; Wang, 2012; Buschman and Miller, 2014), some of which attribute it to persistent, self-reinforcing patterns of neural activation (Goldman et al., 2003) in the recurrent networks of the prefrontal cortex. Prefrontal working memory appears to be made up of multiple functionally distinct subsystems (Markowitz et al., 2015). Neural models of working memory can store not only scalar variables (Seung, 1998), but also high-dimensional vectors (Eliasmith and Anderson, 2004; Eliasmith et al., 2012) or sequences of vectors (Choo and Eliasmith, 2010). Working memory buffers seem crucial for human-like cognition, e.g., reasoning, as they allow short-term storage while also—in conjunction with other mechanisms—enabling generalization of operations across anything that can fill the buffer.

4.1.3. Storing State in Association with Saliency

Saliency, or interestingness, measures can be used to tag the importance of a memory (Gonzalez Andino and Grave de Peralta Menendez, 2012). This can allow removal of the boring data from the training set, allowing a mechanism that is more like optimal experimentation. Moreover, saliency can guide memory replay or sampling from generative models, to generate more training data drawn from a distribution useful for learning (Ji and Wilson, 2007; Mnih et al., 2015). Conceivably, hippocampal replay could allow a batch-like training process, similar to how most machine learning systems are trained, rather than requiring all training to occur in an online fashion. Plasticity mechanisms in memory systems which are gated by saliency are starting to be uncovered in neuroscience (Dudman et al., 2007). Importantly, the notions of "saliency" computed by the brain could be quite intricate and multi-faceted, potentially leading to complex schemes by which specific kinds of memories would be tagged for later context-dependent retrieval. As a hypothetical example, representations of both timing and importance associated with memories could perhaps allow retrieval only of important memories that happened within a certain window of time (MacDonald et al., 2011; Kraus et al., 2013; Rubin et al., 2015). Storing and retrieving information selectively based on specific properties of the information itself, or of "tags" appended to that information, is a powerful computational primitive that could enable learning of more complex tasks. Relatedly, we know that certain pathways become associated with certain kinds of memories, e.g., specific pathways for fear-related memory in mice.

4.2. Structured Routing Systems

To use its information flexibly, the brain needs structured systems for routing data. Such systems need to address multiple temporal and spatial scales, and multiple modalities of control. Thus, there are several different kinds of information routing systems in the brain which operate by different mechanisms and under different constraints.

4.2.1. Attention

If we can focus on one thing at a time, we may be able to allocate more computational resources to processing it, make better use of scarce data to learn about it, and more easily

[42] Attractor models of memory in neuroscience tend to have the property that only one memory can be accessed at a time (although a brain can have many such memories that can be accessed in parallel). Recent machine learning systems, however, have constructed differentiable addressable memory (Graves et al., 2014) and gating (Whitney et al., 2016) systems by allowing weighted superpositions of memory registers or gates to be queried—it is unclear whether the brain uses such mechanisms.

[43] Computational analogies have also been drawn between associative memory storage and object recognition (Leibo et al., 2015a), suggesting the possibility of closely related computations occurring in parts of neocortex and hippocampus. Indeed, the hippocampus and olfactory cortex (a more ancient and simpler structure than the neocortex Shepherd, 2014; Fournier et al., 2015) are few-layer structures described in comparative anatomy as "allocortex," as opposed to the six-layered "neocortex," and both types of cortex have some anatomical similarities (particularly for CA1 and subiculum, though less so for CA3 and dentate gyrus) such as the presence of pyramidal neurons. It has been suggested that the hippocampus can be thought of as the top of the cortical hierarchy (Hawkins and Blakeslee, 2007), responsible for handling and remembering information that could not be fully explained by lower levels of the hierarchy. These computational connections are still tentative.

store and retrieve it from memory[44]. Notably in this context, attention allows improvements in learning: if we can focus on just a single object, instead of an entire scene, we can learn about it more easily using limited data. Formal accounts in a Bayesian framework talk about attention reducing the sample complexity of learning (Chikkerur et al., 2010). Likewise, in models, the processes of applying attention, and of effectively making use of incoming attentional signals to appropriately modulate local circuit activity, can themselves be learned by optimizing cost functions (Jaramillo and Pearlmutter, 2004; Mnih et al., 2014). The right kinds of attention make processing and learning more efficient, and also allow for a kind of programmatic control over multi-step perceptual tasks.

How does the brain determine where to allocate attention, and how is the attentional signal physically mediated? Answering this question is still an active area of neuroscience. Higher-level cortical areas may be specialized in allocating attention. The problem is made complex by the fact that there seem to be many different types of attention—such as object-based, feature-based and spatial attention in vision—that may be mediated by interactions between different brain areas. The frontal eye fields (area FEF), for example, are important in visual attention, specifically for controlling saccades of the eyes to attended locations. Area FEF contains "retinotopic" spatial maps whose activation determines the saccade targets in the visual field. Other prefrontal areas such as the dorsolateral prefrontal cortex and inferior frontal junction are also involved in maintaining representations that specify the targets of certain types of attention. Certain forms of attention may require a complex interaction between brain areas, e.g., to determine targets of attention based on higher-level properties that are represented across multiple areas, like the identity and spatial location of a specific face (Baldauf and Desimone, 2014).

There are many proposed neural mechanisms of attention, including the idea that synchrony plays a role (Baldauf and Desimone, 2014), perhaps by creating resonances that facilitate the transfer of information between synchronously oscillating neural populations in different areas[45]. Other proposed mechanisms include specific circuits for attention-dependent signal routing (Anderson and Van Essen, 1987; Olshausen et al., 1993). Various forms of attention also have specific neurophysiological signatures, such as enhancements in synchrony among neural spikes and with the ambient local field potential, changes in the sharpness of neural tuning curves, and other properties. These diverse effects and signatures of attention may be consequences of underlying pathways that wire up to particular elements of cortical microcircuits to mediate different attentional effects (Bobier et al., 2014).

4.2.2. Buffers

One possibility is that the brain uses distinct groups of neurons, which we can call "buffers," to store distinct variables, such as the subject or object in a sentence (Frankland and Greene, 2015). Having memory buffers allows the abstraction of a variable.

Once we establish that the brain has a number of memory buffers, we need ways for those buffers to interact. We need to be able to take a buffer, do a computation on its contents and store the output into another buffer. But if the representations in each of two groups of neurons are learned, and hence are coded differently, how can the brain "copy and paste" information between these groups of neurons? Malsburg argued that such a system of separate buffers is impossible because the neural pattern for "chair" in buffer 1 has nothing in common with the neural pattern for "chair" in buffer 2—any learning that occurs for the contents of buffer 1 would not automatically be transferable to buffer 2. Various mechanisms have been proposed to allow such transferability, which focus on ways in which all buffers could be trained jointly and then later separated so that they can work independently when they need to[46].

4.2.3. Discrete Gating of Information Flow between Buffers

Dense connectivity is only achieved locally, but it would be desirable to have a way for any two cortical units to talk to one another, if needed, regardless of their distance from one another, and without introducing crosstalk[47]. It is therefore critical to be able to dynamically turn on and off the transfer of information between different source and destination regions, in much the manner of a switchboard. Together with attention, such dedicated routing systems can make sure that a brain area receives exactly the information it needs. Such a discrete routing system is, of course, central to cognitive architectures like ACT-R (Anderson, 2007). The key feature of ACT-R is the ability to evaluate the IF clauses of tens of thousands of symbolic rules

[44]Attention also arguably solves certain types of perceptual binding problem (Reynolds and Desimone, 1999).

[45]The precise roles of synchrony in information routing and other processes, and when it should be viewed as a causal factor vs. as an epiphenomenon of other mechanisms, is still being worked out. In some theories, oscillations occur as consequences of certain recurrent processing loops, e.g., thalamo-cortico-striatal loops (Eliasmith et al., 2012). In other models, so-called "dynamic circuit motifs," involving specific combinations of cellular and synaptic sub-types, both generate synchronies (e.g., in part via intrinsically rhythmic pacemaker neurons) and exploit them for specific computational roles, particularly in the rapid dynamic formation of communication networks (Womelsdorf et al., 2014).

[46]One idea for achieving such transferability is that of a partitionable (Hayworth, 2012) or annexable (Bostrom, 1996) network. These models posit that a large associative memory network links all the different buffers. This large associative memory network has a number of stable attractor states. These are called "global" attractor states since they link across all the buffers. Forcing a given buffer into an activity pattern resembling that of its corresponding "piece" of an attractor state will cause the entire global network to enter that global attractor state. During training, all of the connections between buffers are turned on, so that their learned contents, though not identical, are kept in correspondence by being part of the same attractor. Later, the connections between specific buffers can be turned off to allow them to store different information. Copy and paste is then implemented by turning on the connections between a source buffer and a destination buffer (Hayworth, 2012). Copying between a source and destination buffer can also be implemented, i.e., learned, in a deep learning system using methods similar to the addressing mechanisms of the Neural Turing Machine (Graves et al., 2014).

[47]Micro-stimulation experiments, in which an animal learns to behaviorally report stimulation of electrode channels located in diverse cortical regions, suggest that many areas can be routed or otherwise linked to behavioral "outputs" (Histed et al., 2013), although the mechanisms behind this—e.g., whether this stimulation gives rise to a high-level percept that the animal then uses to make a decision—are unclear. Likewise, it is possible to reinforcement-train an animal to control the activity of individual neurons (Fetz, 1969, 2007).

(called "productions"), in parallel, approximately every 50 ms. Each rule requires equality comparisons between the contents of many constant and variable memory buffers, and the execution of a rule leads to the conditional routing of information from one buffer to another.

What controls which long-range routing operations occur when, i.e., where is the switchboad and what controls it? Several models, including ACT-R, have attributed such parallel rule-based control of routing to the action selection circuitry (Gurney et al., 2001; Terrence Stewart, 2010) of the basal ganglia (BG) (O'Reilly and Frank, 2006; Stocco et al., 2010), and its interaction with working memory buffers in the prefrontal cortex. In conventional models of thalamo-cortico-striatal loops, competing actions of the direct and indirect pathways through the basal ganglia can inhibit or disinhibit an area of motor cortex, thereby gating a motor action[48]. Models like (O'Reilly and Frank, 2006; Stocco et al., 2010; Terrence Stewart, 2010) propose further that the basal ganglia can gate not just the transfer of information from motor cortex to downstream actuators, but also the transfer of information between cortical areas. To do so, the basal ganglia would dis-inhibit a thalamic relay (Sherman, 2005, 2007) linking two cortical areas. Dopamine-related activity is thought to lead to temporal difference reinforcement learning of such gating policies in the basal ganglia (Frank and Badre, 2012). Beyond the basal ganglia, there are also other, separate pathways involved in action selection, e.g., in the prefrontal cortex (Daw et al., 2006). Thus, multiple systems including basal ganglia and cortex could control the gating of long-range information transfer between cortical areas, with the thalamus perhaps largely constituting the switchboard itself.

How is such routing put to use in a learning context? One possibility is that the basal ganglia acts to orchestrate the training of the cortex. The basal ganglia may exert tight control[49] over the cortex, helping to determine when and how it is trained. Indeed, because the basal ganglia pre-dates the cortex evolutionarily, it is possible that the cortex evolved as a flexible, trainable resource that could be harnessed by existing basal ganglia circuitry. All of the main regions and circuits of the basal ganglia are conserved from our common ancestor with the lamprey more than five hundred million years ago. The major part of the basal ganglia even seems to be conserved from our common ancestor with insects (Strausfeld and Hirth, 2013). Thus, in addition to its real-time action selection and routing functions, the basal ganglia may sculpt how the cortex learns.

4.3. Structured State Representations to Enable Efficient Algorithms

Certain algorithmic problems benefit greatly from particular types of representation and transformation, such as a grid-like representation of space. In some cases, rather than just waiting for them to emerge via gradient descent optimization of appropriate cost functions, the brain may be pre-structured to facilitate their creation.

4.3.1. Continuous Predictive Control

We often have to plan and execute complicated sequences of actions on the fly, in response to a new situation. At the lowest level, that of motor control, our body and our immediate environment change all the time. As such, it is important for us to maintain knowledge about this environment in a continuous way. The deviations between our planned movements and those movements that we actually execute continuously provide information about the properties of the environment. Therefore, it seems important to have a specialized system, optimized for high-speed continuous processing, that takes all our motor errors and uses them to update a dynamical model of our body and our immediate environment that can predict the delayed sensory results of our motor actions (McKinstry et al., 2006).

It appears that the cerebellum is such a structure, and lesions to it abolish our way of dealing successfully with a changing body. Incidentally, the cerebellum has more connections than the rest of the brain taken together, apparently in a largely feedforward architecture, and the tiny cerebellar granule cells, which may form a randomized high-dimensional input representation (Marr, 1969; Jacobson and Friedrich, 2013), outnumber all other neurons. The brain clearly needs a dedicated way of quickly and continuously correcting movements to minimize errors, without needing to rely on slow and complex association learning in the neocortex in order to do so.

Newer research shows that the cerebellum is involved in a broad range of cognitive problems (Moberget et al., 2014) as well, potentially because they share computational problems with motor control. For example, when subjects estimate time intervals, which are naturally important for movement, it appears that the brain uses the cerebellum even if no movements are involved (Gooch et al., 2010). Even individual cerebellar Purkinjie cells may learn to generate precise timings of their outputs (Johansson et al., 2014). The brain also appears to use inverse models to rapidly predict motor activity that would give rise to a given sensory target (Hanuschkin et al., 2013; Giret et al., 2014). Such mechanisms could be put to use far beyond motor control, in bootstrapping the training of a larger architecture by exploiting continuously changing error signals to update a real-time model of the system state.

[48]Conventionally, models of the basal ganglia involve all or none gating of an action, but recent evidence suggests that the basal ganglia may also have continuous, analog outputs (Yttri and Dudman, 2016).

[49]It has been suggested that the basic role of the BG is to provide tonic inhibition to other circuits (Grillner et al., 2005). Release of this inhibition can then activate a "discrete" action, such as a motor command. A core function of the BG is thus to choose, based on patterns detected in its input, which of a finite set of actions to initiate via such release of inhibition. In many models of the basal gangliaàĂŹs role in cognitive control, the targets of inhibition are thalamic relays (Sherman, 2005), which are set in a default "off" state by tonic inhibition from the basal ganglia. Upon disinhibition of a relay, information is transferred from one cortical location to another—a form of conditional "gating" of information transfer. For example, the BG might be able to selectively "clamp" particular groups of cortical neurons in a fixed state, while leaving others free to learn and adapt. It could thereby enforce complex training routines, perhaps similar to those used to force the emergence of disentangled representations in (Kulkarni et al., 2015). The idea that the basal ganglia can train the cortex is not new, and already appears to have considerable experimental and anatomical support (Pasupathy and Miller, 2005; Ashby et al., 2007, 2010; Turner and Desmurget, 2010).

4.3.2. Hierarchical Control

Importantly, many of the control problems we appear to be solving are hierarchical. We have a spinal cord, which deals with the fast signals coming from our muscles and proprioception. Within neuroscience, it is generally assumed that this system deals with fast feedback loops and that this behavior is learned to optimize its own cost function. The nature of cost functions in motor control is still under debate. In particular, the timescale over which cost functions operate remains unclear: motor optimization may occur via real-time responses to a cost function that is computed and optimized online, or via policy choices that change over time more slowly in response to the cost function (Körding, 2007). Nevertheless, the effect is that central processing in the brain has an effectively simplified physical system to control, e.g., one that is far more linear. So the spinal cord itself already suggests the existence of two levels of a hierarchy, each trained using different cost functions.

However, within the computational motor control literature (see e.g., DeWolf and Eliasmith, 2011), this idea can be pushed far further, e.g., with a hierarchy including spinal cord, M1, PMd, frontal, prefrontal areas. A low level may deal with muscles, the next level may deal with getting our limbs to places or moving objects, a next layer may deal with solving simple local problems (e.g., navigating across a room) while the highest levels may deal with us planning our path through life. This factorization of the problem comes with multiple aspects: First, each level can be solved with its own cost functions, and second, every layer has a characteristic timescale. Some levels, e.g., the spinal cord, must run at a high speed. Other levels, e.g., high-level planning, only need to be touched much more rarely. Converting the computationally hard optimal control problem into a hierarchical approximation promises to make it dramatically easier.

Does the brain solve control problems hierarchically? There is evidence that the brain uses such a strategy (Botvinick et al., 2009; Botvinick and Weinstein, 2014), beside neural network demonstrations (Wayne and Abbott, 2014). The brain may use specialized structures at each hierarchical level to ensure that each operates efficiently given the nature of its problem space and available training signals. At higher levels, these systems may use an abstract syntax for combining sequences of actions in pursuit of goals (Allen et al., 2010). Subroutines in such processes could be derived by a process of chunking sequences of actions into single actions (Graybiel, 1998; Botvinick and Weinstein, 2014). Some brain areas like Broca's area, known for its involvement in language, also appear to be specifically involved in processing the hierarchical structure of behavior, as such, as opposed to its detailed temporal structure (Koechlin and Jubault, 2006).

At the highest level of the decision making and control hierarchy, human reward systems reflect changing goals and subgoals, and we are only beginning to understand how goals are actually coded in the brain, how we switch between goals, and how the cost functions used in learning depend on goal state (Buschman and Miller, 2014; O'Reilly et al., 2014b; Pezzulo et al., 2014). Goal hierarchies are beginning to be incorporated into deep learning (Kulkarni et al., 2016).

Given this hierarchical structure, the optimization algorithms can be fine-tuned. For the low levels, there is sheer unlimited training data. For the high levels, a simulation of the world may be simple, with a tractable number of high-level actions to choose from. Finally, each area needs to give reinforcement to other areas, e.g., high levels need to punish lower levels for making planning complicated. Thus this type of architecture can simplify the learning of control problems.

Progress is being made in both neuroscience and machine learning on finding potential mechanisms for this type of hierarchical planning and goal-seeking. This is beginning to reveal mechanisms for chunking goals and actions and for searching and pruning decision trees (O'Reilly et al., 2014a; Huys et al., 2015; Balaguer et al., 2016; Krishnamurthy et al., 2016; Tamar et al., 2016). The study of model-based hierarchical reinforcement learning and prospective optimization (Sejnowski and Poizner, 2014), which concerns the planning and evaluation of nested sequences of actions, implicates a network coupling the dorsolateral prefontral and orbitofrontal cortex, and the ventral and dorsolateral striatum (Botvinick et al., 2009). Hierarchical RL relies on a hierarchical representation of state and action spaces, and it has been suggested that error-driven learning of an optimal such representation in the hippocampus[50] gives rise to place and grid cell properties (Stachenfeld, 2014), with goal representations themselves emerging in the amygdala, prefrontal cortex and other areas (O'Reilly et al., 2014a).

The question of how control problems can be successfully divided into component problems remains one of the central questions in neuroscience (Wolpert and Flanagan, 2016) and machine learning (Kulkarni et al., 2016), and the cost functions involved in learning to create such decompositions are still unknown. These considerations may begin to make plausible, however, how the brain could not only achieve its remarkable feats of motor learning—such as generating complex "innate" motor programs, like walking in the newborn gazelle almost immediately after birth—but also the kind of planning that allows a human to prepare a meal or travel from London to Chicago.

4.3.3. Spatial Planning

Spatial planning requires solving shortest-path problems subject to constraints. If we want to get from one location to another, there are an arbitrarily large number of simple paths that could be taken. Most naive implementations of such shortest paths problems are grossly inefficient. It appears that, in animals, the hippocampus aids—at least in part through "place cell" and "grid cell" systems—in efficient learning about new environments and in targeted navigation in such environments (Brown et al., 2016). Interestingly, once an environment becomes familiar, it appears that areas of the neocortex can take over the role of navigation (Hasselmo and Stern, 2015).

In some simple models, targeted navigation in the hippocampus is achieved via the dynamics of "bump attractors" or propagating waves in a place cell network with Hebbian plasticity and adaptation (Hopfield, 2009; Buzsáki and Moser, 2013; Ponulak and Hopfield, 2013), which allows the network to effectively chart out a path in the space of place cell

[50]Like many brain areas, the hippocampus is richly innervated by a variety of reward-related and other neuromodulatory systems (Verney et al., 1985; Colino and Halliwell, 1987; Hasselmo and Wyble, 1997).

representations. Other navigation models make use of the grid cell system. The place cell network may[51] take input from a grid cell network that computes precise distances and directions, perhaps by integrating head direction and velocity signals—grid cells fire when the animal is on any node of a regularly spaced hexagonal grid. Different parts of the entorhinal cortex contain grid cells with different grid spacings, and place cells may combine information from multiple such grids in order to build up responses to particular single positions. These systems are highly structured temporally, e.g., containing nested gamma and theta oscillation structures that are phased locked to sequences of place-cell responses, interfering oscillators frequency-shifted by the animal's motion velocity (Zilli and Hasselmo, 2010), tuned cellular resonances (Giocomo et al., 2007; Buzsáki, 2010), and other neural phenomena that lie far outside a conventional artificial neural network description. It seems that an intricate interplay of spatial and temporal network structures may be essential for encoding sequences of spatiotemporal events across multiple scales, and using them to drive multiple forms of learning, e.g., supporting forward and reverse sequence replay with various temporal compression factors (Buzsáki, 2010).

Higher-level cognitive tasks such as prospective planning appear to share computational sub-problems with path-finding (Hassabis and Maguire, 2009)[52]. Interaction between hippocampus and prefrontal cortex could perhaps support a more abstract notion of "navigation" in a space of goals and sub-goals. Interestingly, there is preliminary evidence from fMRI that abstract concepts are also represented according to grid-cell-like hexagonal grid structures in humans (Constantinescu et al., 2016), as well as preliminary evidence that social relationships may also be represented through a hippocampal map (Tavares et al., 2015). Having specialized structures for path-finding could thus simplify a variety of computational problems at different levels of abstraction.

4.3.4. Variable Binding

Language and reasoning appear to present a problem for neural networks (Minsky, 1991; Marcus, 2001; Hadley, 2009): we seem to be able to apply common grammatical rules to sentences regardless of the content of those sentences, and regardless of whether we have ever seen even remotely similar sentences in the training data. While this is achieved automatically in a computer with fixed registers, location addressable memories, and hard-coded operations, how it could be achieved in a biological brain, or emerge from an optimization algorithm, has been under debate for decades.

As the putative key capability underlying such operations, variable binding has been defined as "the transitory or permanent tying together of two bits of information: a variable (such as an X or Y in algebra, or a placeholder like subject or verb in a sentence) and an arbitrary instantiation of that variable (say, a single number, symbol, vector, or word)" (Marcus et al., 2014a,b). A number of potential biologically plausible binding mechanisms (Eliasmith et al., 2012; Hayworth, 2012; Kriete et al., 2013; Goertzel, 2014) are reviewed in Marcus et al. (2014a) and Marcus et al. (2014b). Some, such as vector symbolic architectures[53], which were proposed in cognitive science (Plate, 1995; Stewart and Eliasmith, 2009; Eliasmith, 2013), are also being considered in the context of efficiently-trainable artificial neural networks (Danihelka et al., 2016)—in effect, these systems learn how to use variable binding.

Variable binding could potentially emerge from simpler memory systems. For example, the Scrub-Jay can remember the place and time of last visit for hundreds of different locations, e.g., to determine whether high-quality food is currently buried at any given location (Clayton and Dickinson, 1998). It is conceivable that such spatially-grounded memory systems enabled a more general binding mechanism to emerge during evolution, perhaps through integration with routing systems or other content-addressable or working memory systems.

4.3.5. Hierarchical Syntax

Fixed, static hierarchies (e.g., the hierarchical organization of cortical areas Felleman and Van Essen, 1991) only take us so far: to deal with long chains of arbitrary nested references, we need *dynamic* hierarchies that can implement recursion on the fly. Human language syntax has a hierarchical structure, which Berwick et al described as "composition of smaller forms like words and phrases into larger ones" (Berwick et al., 2012; Miyagawa et al., 2013). The extent of recursion in human language and thought may be captured by a class of automata known as higher-order pushdown automata, which can be implemented via finite state machines with access to nested stacks (Rodriguez and Granger, 2016). Specific fronto-temporal networks may be involved in representing and generating such hierarchies (Dehaene et al., 2015), e.g., with the hippocampal system playing a key role in implementing some analog of a pushdown stack (Rodriguez and Granger, 2016)[54].

Little is known about the underlying circuit mechanisms for such dynamic hierarchies, but it is clear that specific affordances for representing such hierarchies in an efficient way would be beneficial. This may be closely connected with the issue of variable binding, and it is possible that operations similar to pointers could be useful in this context, in both the brain and artificial neural networks (Kriete et al., 2013; Kurach et al., 2015). Augmenting neural networks with a differentiable analog of a

[51]It remains unclear whether place cells take input from the grid cell system or vice versa (Hasselmo, 2015).

[52]Other spatial problems such as mental rotation may require learning architectures specialized for geometric coordinate transformations (Hinton et al., 2011; Jaderberg et al., 2015) or binding mechanisms that support structural, compositional, parametric descriptions of a scene (Hayworth et al., 2011).

[53]There is some direct fMRI evidence for anatomically separate registers representing the contents of different sentence roles in the human brain (Frankland and Greene, 2015), which is suggestive of a possible anatomical binding mechanism, but also consistent with other mechanisms like vector symbolic architectures. More generally, the substrates of symbolic processing in the brain may bear an intimate connection with the representation of objects in working memory in the prefrontal cortex, and specifically with the question of how the PFC represents multiple objects in working memory simultaneously. This question is undergoing extensive study in primates (Warden and Miller, 2007, 2010; Siegel et al., 2009; Rigotti et al., 2013).

[54]There is controversy around claims that recursive syntax is also present in songbirds (Van Heijningen et al., 2009).

push-down stack is another such affordance being pursued in machine learning (Joulin and Mikolov, 2015).

4.3.6. Mental Programs and Imagination

Humans excel at stitching together sub-actions to form larger actions (Verwey, 1996; Acuna et al., 2014; Sejnowski and Poizner, 2014). Structured, serial, hierarchical probabilistic programs have recently been shown to model aspects of human conceptual representation and compositional learning (Lake et al., 2015). In particular, sequential programs were found to enable one-shot learning of new geometric/visual concepts (Lake et al., 2015). Generative programs have also been proposed in the context of scene understanding (Battaglia et al., 2013). The ability to deal with problems in terms of sub-problems is central both in human thought and in many successful algorithms.

One possibility is that the hippocampus supports the rapid construction and learning of sequential programs, e.g., in multi-step planning. An influential idea—known as the "complementary learning systems hypothesis"—is that the hippocampus plays a key role in certain processes where learning must occur quickly on the basis of single episodes, whereas the cortex learns more slowly by aggregating and integrating patterns across large amounts of data (Herd et al., 2013; Leibo et al., 2015a; Blundell et al., 2016; Kumaran et al., 2016). The hippocampus appears to explore, in simulation, possible future trajectories to a goal, even those involving previously unvisited locations (Ólafsdóttir et al., 2015). Hippocampal-prefrontal interaction has been suggested to allow rapid, subconscious evaluation of potential action sequences during decision-making, with the hippocampus in effect simulating the expected outcomes of potential actions that are generated and evaluated in the prefrontal (Mushiake et al., 2006; Wang et al., 2015). The role of the hippocampus in imagination, concept generation (Kumaran et al., 2009), scene construction (Hassabis and Maguire, 2007), mental exploration and goal-directed path planning (Hopfield, 2009; Ólafsdóttir et al., 2015; Brown et al., 2016) suggests that it could help to create generative models to underpin more complex inference such as program induction (Lake et al., 2015) or common-sense world simulation (Battaglia et al., 2013). For example, a sequential, programmatic process, mediated jointly by the basal ganglia, hippocampus and prefrontal cortex might allow one-shot learning of a new concept, as in the sequential computations underlying a process like Bayesian Program Learning (Lake et al., 2015).

Another related possibility is that the cortex itself intrinsically supports the construction and learning of sequential programs (Bach and Herger, 2015). Recurrent neural networks have been used for image generation through a sequential, attention-based process (Gregor et al., 2015), although their correspondence with the brain is unclear[55].

[55]The above mechanisms are spontaneous and subconscious. In conscious thought, too, the brain can clearly visit the multiple layers of a program one after the other. We make high-level plans that we fill with lower-level plans. Humans also have memory for their own thought processes. We have some ability to put "on hold" our current state of mind, start a new train of thought, and then come back to our original thought. We also are able to ask, introspectively, whether we have had a given thought before. The neural basis of these processes is unclear, although one may speculate that the hippocampus is involved.

4.4. Other Specialized Structures

Importantly, there are many other specialized structures known in neuroscience, which arguably receive less attention than they deserve, even for those interested in higher cognition. In the above, in addition to the hippocampus, basal ganglia and cortex, we emphasized the key roles of the thalamus in routing, of the cerebellum as a fast and rapidly trainable control and modeling system, of the amygdala and other areas as a potential source of utility functions, of the retina or early visual areas as a means to generate detectors for motion and other features to bootstrap more complex visual learning, and of the frontal eye fields and other areas as a possible source of attention control. We ignored other structures entirely, whose functions are only beginning to be uncovered, such as the claustrum (Crick and Koch, 2005), which has been speculated to be important for rapidly binding together information from many modalities. Our overall understanding of the functional decomposition of brain circuitry still seems very preliminary.

4.5. Relationships with Other Cognitive Frameworks Involving Specialized Systems

A recent analysis (Lake et al., 2016) suggested directions by which to modify and enhance existing neural-net-based machine learning toward more powerful and human-like cognitive capabilities, particularly by introducing new structures and systems which go beyond data-driven optimization. This analysis emphasized that systems should construct generative models of the world that incorporate compositionality (discrete construction from re-usable parts), inductive biases reflecting causality, intuitive physics and intuitive psychology, and the capacity for probabilistic inference over discrete structured models (e.g., structured as graphs, trees, or programs) (Tervo et al., 2016) to harness abstractions and enable transfer learning.

We view these ideas as consistent with and complementary to the framework of cost functions, optimization and specialized systems discussed here. One might seek to understand how optimization and specialized systems could be used to implement some of the mechanisms proposed in Lake et al. (2016) inside neural networks. Lake et al. (2016) emphasize how incorporating additional structure into trainable neural networks can potentially give rise to systems that use compositional, causal and intuitive inductive biases and that "learn to learn" using structured models and shared data structures. For example, sub-dividing networks into units that can be modularly and dynamically combined, where representations can be copied and routed, may present a path toward improved compositionality and transfer learning (Andreas et al., 2015). The control flow for recombining pre-existing modules and representations could be learned via reinforcement learning (Andreas et al., 2016). How to implement the broad set of mechanisms discussed in Lake et al. (2016) is a key computational problem, and it remains open at which levels (e.g., cost functions and training procedures vs. specialized computational structures vs. underlying neural primitives) architectural innovations will need to be introduced to capture these phenomena.

Primitives that are more complex than those used in conventional neural networks—for instance, primitives that act as state machines with complex message passing (Bach and Herger, 2015) or networks that intrinsically implement Bayesian inference (George and Hawkins, 2009)—could potentially be useful, and it is plausible that some of these may be found in the brain. Recent findings on the power of generic optimization also do not rule out the idea that the brain may explicitly generate and use particular types of structured representations to constrain its inferences; indeed, the specialized brain systems discussed here might provide a means to enforce such constraints. It might be possible to further map the concepts of Lake et al. (2016) onto neuroscience via an infrastructure of interacting cost functions and specialized brain systems under rich genetic control, coupled to a powerful and generic neurally implemented capacity for optimization. For example, it was recently shown that complex probabilistic population coding and inference can arise automatically from backpropagation-based training of simple neural networks (Orhan and Ma, 2016), without needing to be built in by hand. The nature of the underlying primitives in the brain, on top of which learning can operate, is a key question for neuroscience.

5. MACHINE LEARNING INSPIRED NEUROSCIENCE

Hypotheses are primarily useful if they lead to concrete, experimentally testable predictions. As such, we now want to go through the hypotheses and see to which level they can be directly tested, as well as refined, through neuroscience.

5.1. *Hypothesis 1*– Existence of Cost Functions

There are multiple general strategies for addressing whether and how the brain optimizes cost functions. A first strategy is based on observing the endpoint of learning. If the brain uses a cost function, and we can guess its identity, then the final state of the brain should be close to optimal for the cost function. We could thus compare (Güçlü and van Gerven, 2015) receptive fields that are optimized in a simulation, according to a particular cost function, with the measured receptive fields. Various techniques exist to carry out such comparisons in fRMI studies, including population receptive field estimation (Dumoulin and Wandell, 2008; Güçlü and van Gerven, 2015) and representational dissimilarity matrices (Kriegeskorte et al., 2008; Khaligh-Razavi and Kriegeskorte, 2014). This strategy is only beginning to be used at the moment, perhaps because it has been difficult to measure the receptive fields or other representational properties across a large population of *individual* neurons (fMRI operates at a much coarser level), but this situation is beginning to improve technologically with the emergence of large-scale recording methods (Hasselmo, 2015).

A second strategy could directly quantify how well a cost function describes learning. If the dynamics of learning minimize a cost function then the underlying vector field should have a strong gradient descent type component and a weak rotational component, i.e., weight changes will primarily move down the gradient rather than drifting in the nullspace. If we could somehow continuously monitor the synaptic strengths, while externally manipulating them, then we could, in principle, measure the vector field in the space of synaptic weights, and calculate its divergence as well as its rotation. For at least the subset of synapses that are being trained via some approximation to gradient descent, the divergence component should be strong relative to the rotational component. This strategy has not been developed yet due to experimental difficulties with monitoring large numbers of synaptic weights[56].

A third strategy is based on perturbations: cost function based learning should undo the effects of perturbations which disrupt optimality, i.e., the system should return to local minima after a perturbation, and indeed perhaps to the same local minimum after a sufficiently small perturbation. If we change synaptic connections, e.g., in the context of a brain machine interface, we should be able to produce a reorganization that can be predicted based on a guess of the relevant cost function. This strategy is starting to be feasible in motor areas.

Lastly, if we knew structurally which cell types and connections mediated the delivery of error signals vs. input data or other types of connections, then we could stimulate specific connections so as to impose a user-defined cost function. In effect, we would use the brain's own networks as a trainable deep learning substrate, and then study how the network responds to training. Brain machine interfaces can be used to set up specific local learning problems, in which the brain is asked to create certain user-specified representations, and the dynamics of this process can be monitored (Sadtler et al., 2014). Likewise, brain machine interfaces can be used to give the brain access to new datastreams, and to investigate how those datastreams are incorporated into task performance, and whether such incorporation is governed by optimality principles (Dadarlat et al., 2015). In order to do this kind of experiment fully and optimally, we must first understand more about how the system is wired to deliver cost signals. Much of the structure that would be found in connectomic circuit maps, for example, would not just be relevant for short-timescale computing, but also for creating the infrastructure that supports cost functions and their optimization.

Many of the learning mechanisms that we have discussed in this paper make specific predictions about connectivity or dynamics. For example, the "feedback alignment" approach to biological backpropagation suggests that cortical feedback connections should, at some level of neuronal grouping, be largely sign-concordant with the corresponding feedforward connections, although not necessarily of concordant weight (Liao et al., 2015), and feedback alignment also makes predictions for synaptic normalization mechanisms (Liao et al., 2015). The Kickback model for biologically plausible backpropagation has a specific role for NMDA receptors (Balduzzi et al., 2014). Some models that incorporate dendritic coincidence detection for learning temporal sequences predict that a given axon should make only a small number of synapses on a given dendritic

[56] Fluorescent techniques like (Hayashi-Takagi et al., 2015) might be helpful.

segment (Hawkins and Ahmad, 2016). Models that involve STDP learning will make predictions about the dynamics of changing firing rates (Hinton, 2007, 2016; Bengio et al., 2015a,b; Bengio and Fischer, 2015), as well as about the particular network structures, such as those based on autoencoders or recirculation, in which STDP can give rise to a form of backpropagation.

It is critical to establish the unit of optimization. We want to know the scale of the modules that are trainable by some approximation of gradient descent optimization. How large are the networks which share a given error signal or cost function? On what scales can appropriate training signals be delivered? It could be that the whole brain is optimized end-to-end, in principle. In this case we would expect to find connections that carry training signals from each layer to the preceding ones. On successively smaller scales, optimization could be within a brain area, a microcircuit[57], or an individual neuron (Mel, 1992; Körding and König, 2000, 2001; Hawkins and Ahmad, 2016). Importantly, optimization may co-exist across these scales. There may be some slow optimization end-to-end, with stronger optimization within a local area and very efficient algorithms within each cell. Careful experiments should be able to identify the scale of optimization, e.g., by quantifying the extent of learning induced by a local perturbation.

The tightness of the structure-function relationship is the hallmark of molecular and to some extent cellular biology, but in large connectionist learning systems, this relationship can become difficult to extract: the same initial network can be driven to compute many different functions by subjecting it to different training[58,59]. It can be hard to understand the way a neural network solves its problems.

How could one tell the difference, then, between a gradient-descent trained network vs. untrained or random networks vs. a network that has been trained against a different kind of task? One possibility would be to train artificial neural networks against various candidate cost functions, study the resulting neural tuning properties (Todorov, 2002), and compare them with those found in the circuit of interest (Zipser and Andersen, 1988). This has already been done to aid the interpretation of the neural dynamics underlying decision making in the PFC (Sussillo, 2014), working memory in the posterior parietal cortex (Rajan et al., 2016) and object or action representation in the visual system (Tacchetti et al., 2016; Yamins and DiCarlo, 2016a,b). Some have gone on to suggest a direct correspondence between cortical circuits and optimized, appropriately regularized (Sussillo et al., 2015), recurrent neural networks (Liao and Poggio, 2016). In any case, effective analytical methods to reverse engineer complex machine learning systems (Jonas and Kording, 2016), and methods to reverse engineer biological brains, may have some commonalities.

Does this emphasis on function optimization and trainable substrates mean that we should give up on reverse engineering the brain based on detailed measurements and models of its specific connectivity and dynamics? On the contrary: we should use large-scale brain maps to try to better understand (a) how the brain implements optimization, (b) where the training signals come from and what cost functions they embody, and (c) what structures exist, at different levels of organization, to constrain this optimization to efficiently find solutions to specific kinds of problems. The answers may be influenced by diverse local properties of neurons and networks, such as homeostatic rules of neural structure, gene expression and function (Marder and Goaillard, 2006), the diversity of synapse types, cell-type-specific connectivity (Jiang et al., 2015), patterns of inter-laminar projection, distributions of inhibitory neuron types, dendritic targeting and local dendritic physiology and plasticity (Markram et al., 2015; Bloss et al., 2016; Morgan et al., 2016; Sandler et al., 2016) or local glial networks (Perea et al., 2009). They may also be influenced by the integrated nature of higher-level brain systems, including mechanisms for developmental bootstrapping (Ullman et al., 2012), information routing (Gurney et al., 2001; Stocco et al., 2010), attention (Buschman and Miller, 2010) and hierarchical decision making (Lee et al., 2015). Mapping these systems in detail is of paramount importance to understanding how the brain works, down to the nanoscale dendritic organization of ion channels and up to the real-time global coordination of cortex, striatum and hippocampus, all of which are computationally relevant in the framework we have explicated here. We thus expect that large-scale, multi-resolution brain maps would be useful in testing these framework-level ideas, in inspiring their refinements, and in using them to guide more detailed analysis.

5.2. *Hypothesis 2*– Biological Fine-Structure of Cost Functions

Clearly, we can map differences in structure, dynamics and representation across brain areas. When we find such differences,

[57] The use of structured microcircuits rather than individual neurons as the units of learning can ease the burden on the learning rules possessed by individual neurons, as exemplified by a study implementing Helmholtz machine learning in a network of spiking neurons using conventional plasticity rules (Roudi and Taylor, 2015; Sountsov and Miller, 2015). As a simpler example, the classical problem of how neurons with only one output axon could communicate both activation and error derivatives for backpropagation ceases to be a problem if the unit of optimization is not a single neuron. Similar considerations hold for the issue of weight symmetry, or approximate sign-concordance in the case of feedback alignment (Liao et al., 2015).

[58] Within this framework, networks that adhere to the basic statistics of neural connectivity, electrophysiology and morphology, such as the initial cortical column models from the Blue Brain Project (Markram et al., 2015), would recapitulate some properties of the cortex, but—just like untrained neural networks—would not spontaneously generate complex functional computation without being subjected to a multi-stage training process, naturalistic sensory data, signals arising from other brain areas and action-driven reinforcement signals.

[59] Not only in applied machine learning, but also in today's most advanced neuro-cognitive models such as SPAUN (Eliasmith et al., 2012; Eliasmith, 2013), the detailed local circuit connectivity is obtained through an optimization process of some kind to achieve a particular functionality. In the case of modern machine learning, training is often done via end-to-end backpropagation through an architecture that is only structured at the level of higher-level "blocks" of units, whereas in SPAUN each block is optimized (Eliasmith and Anderson, 2004) separately according to a procedure that allows the blocks to subsequently be stitched together in a coherent way. Technically, the Neural Engineering Framework (Eliasmith and Anderson, 2004) used in SPAUN uses singular value decomposition, rather than gradient descent, to compute the connections weights as optimal linear decoders. This is possible because of a nonlinear mapping into a high-dimensional space, in which approximating any desired function can be done via a hyperplane regression (Tapson and van Schaik, 2013).

the question remains as to whether we can interpret these as resulting from differences in the internally-generated cost functions, as opposed to differences in the input data, or from differences that reflect other constraints unrelated to cost functions. If we can directly measure aspects of the cost function in different areas, then we can also compare them across areas. For example, methods from inverse reinforcement learning[60] might allow backing out the cost function from observed plasticity (Ng and Russell, 2000).

Moreover, as we begin to understand the "neural correlates" of particular cost functions—perhaps encoded in particular synaptic or neuromodulatory learning rules, genetically-guided local wiring patterns, or patterns of interaction between brain areas—we can also begin to understand when differences in observed neural circuit architecture reflect differences in cost functions.

We expect that, for each distinct learning rule or cost function, there may be specific molecularly identifiable types of cells and/or synapses. Moreover, for each specialized system there may be specific molecularly identifiable developmental programs that tune it or otherwise set its parameters. This would make sense if evolution has needed to tune the parameters of one cost function without impacting others.

How many different types of internal training signals does the brain generate? When thinking about error signals, we are not just talking about dopamine and serotonin, or other classical reward-related pathways. The error signals that may be used to train specific sub-networks in the brain, via some approximation of gradient descent or otherwise, are not necessarily equivalent to reward signals. It is important to distinguish between cost functions that may be used to drive optimization of specific sub-circuits in the brain, and what are referred to as "value functions" or "utility functions," i.e., functions that predict the agent's aggregate future reward. In both cases, similar reinforcement learning mechanisms may be used, but the interpretation of the cost functions is different. We have not emphasized global utility functions for the animal here, since they are extensively studied elsewhere (e.g., O'Reilly et al., 2014a; Bach, 2015), and since we argue that, though important, they are only a part of the picture, i.e., that the brain is not solely an end-to-end reinforcement trained system.

Progress in brain mapping could soon allow us to classify the types of reward signals in the brain, follow the detailed anatomy and connectivity of reward pathways throughout the brain, and map in detail how reward pathways are integrated into striatal, cortical, hippocampal and cerebellar microcircuits. This program is beginning to be carried out in the fly brain, in which twenty specific types of dopamine neuron project to distinct anatomical compartments of the mushroom body to train distinct odor classifiers operating on a set of high-dimensional odor representations (Caron et al., 2013; Aso et al., 2014a,b; Cohn et al., 2015). It is known that, even within the same system, such as the fly olfactory pathway, some neuronal wiring is highly specific and molecularly programmed (Hattori et al., 2007; Hong and Luo, 2014), while other wiring is effectively random (Caron et al., 2013), and yet other wiring is learned (Aso et al., 2014a). The interplay between such design principles could give rise to many forms of "division of labor" between genetics and learning. Likewise, it is believed that birdsong learning is driven by reinforcement learning using a specialized cost function that relies on comparison with a memorized version of a tutor's song (Fiete et al., 2007), and also that it involves specialized structures for controlling song variability during learning (Aronov et al., 2011). These detailed pathways underlying the construction of cost functions for vocal learning are beginning to be mapped (Mandelblat-Cerf et al., 2014). Starting with simple systems, it should become possible to map the reward pathways and how they evolved and diversified, which would be a step on the way to understanding how the system learns.

These types of mapping efforts would be a first step toward the ability to create a concrete model of the brain's optimization architecture. Our discussion here has focused on trying to anticipate, based on known neuroscience knowledge and on approaches becoming successful in machine learning, the *kinds* of local cost functions that the brain may rely on, and how specialized brain systems may enable efficient solutions to optimization problems. However, this framework-level discussion is not a formal specification, either of the architecture, or of a notion of biologically applied cost function that could be directly measured based on neural data. In order to move toward a more formal specification of the kind of model we are proposing here, it would be useful to map the architecture of the brain's reward systems and to identify other biological pathways that may mediate the generation and delivery of error signals. Based on such maps, one could identify regions which are proposed to be subject to a single cost function. Otherwise, the problem of inference of the cost function, e.g., based on neural dynamics becomes ill-posed: one can define a local cost function for an *arbitrary* dynamics by integrating the trajectory of the system, but this approach in general lacks explanatory power and also, crucially, lacks any circuit-level relationship with the brain's actual neural mechanisms of optimization, i.e., such a defined cost function does not necessarily correspond to the cost functions that the biological machinery is actually organized to optimize. Notably, some of the relevant biological pathways mediating cost functions and error signals may involve key biomolecular or gene expression aspects, not just real-time patterns of neural activity.

Another related consideration, in trying to formalize this type approach and to infer cost functions from neural measurements, is that not all neurons in the circuit may be subject to optimization: after all, some neurons may be needed to generate the error signals themselves, or to mediate the optimization process for other neurons, or to perform other unrelated functions. Furthermore, within a given region, there may be multiple sub-circuits subject to different optimization pressures.

[60]There is a rich tradition of trying to estimate the cost function used by human beings (Ng and Russell, 2000; Finn et al., 2016; Ho and Ermon, 2016). The idea is that we observe (by stipulation) behavior that is optimal for the human's cost function. We can then search for the cost function that makes the observed behavior most probable and simultaneously makes the behaviors that could have been observed, but were not, least probable. Extensions of such approaches could perhaps be used to ask which cost functions the brain is optimizing.

It is the claim that the brain actually has structured biological machinery to generate, route and apply specific cost functions that gives substance to our proposal, over and above the trivial claim that many kinds of dynamics can be viewed as optimizations, but our knowledge of this machinery is still limited. This is not to mention the difficulties involved in inferring cost functions in the presence of noise or constraints on the dynamics. Thus, one cannot blindly collect the neurons in an arbitrary region, measure their dynamics, and hope to infer their cost function by solving an inverse problem—instead, a rich interplay between structural mapping, dynamic mapping, hypothesis generation, modeling and perturbation is likely to be necessary in order to gain a detailed knowledge of which cost functions the brain uses and how it does so.

5.3. *Hypothesis 3*– Embedding within a Pre-structured Architecture

If different brain structures are performing distinct types of computations with a shared goal, then optimization of a joint cost function will take place with different dynamics in each area. If we focus on a higher level task, e.g., maximizing the probability of correctly detecting something, then we should find that basic feature detection circuits should learn when the features were insufficient for detection, that attentional routing structures should learn when a different allocation of attention would have improved detection and that memory structures should learn when items that matter for detection were not remembered. If we assume that multiple structures are participating in a joint computation, which optimizes an overall cost function (but see Hypothesis 2), then an understanding of the computational function of each area leads to a prediction of the measurable plasticity rules.

6. NEUROSCIENCE INSPIRED MACHINE LEARNING

Machine learning may be equally transformed by neuroscience. Within the brain, a myriad of subsystems and layers work together to produce an agent that exhibits general intelligence. The brain is able to show intelligent behavior across a broad range of problems using only relatively small amounts of data. As such, progress at understanding the brain promises to improve machine learning. In this section, we review our three hypotheses about the brain and discuss how their elaboration might contribute to more powerful machine learning systems.

6.1. *Hypothesis 1*– Existence of Cost Functions

A good practitioner of machine learning should have a broad range of optimization methods at their disposal as different problems ask for different approaches. The brain, we have argued, is an implicit machine learning mechanism which has been evolved over millions of years. Consequently, we should expect the brain to be able to optimize cost functions efficiently, across many domains and kinds of data. Indeed, across different animal phyla, we even see *convergent* evolution of certain brain structures (Shimizu and Karten, 2013; Güntürkün and Bugnyar, 2016), e.g., the bird brain has no cortex yet has developed homologous structures which—as the linguistic feats of the African Gray Parrot demonstrate—can give rise to quite complex intelligence. It seems reasonable to hope to learn how to do truly general-purpose optimization by looking at the brain.

Indeed, there are multiple kinds of optimization that we may expect to discover by looking at the brain. At the hardware level, the brain clearly manages to optimize functions efficiently despite having slow hardware subject to molecular fluctuations, suggesting directions for improving the hardware of machine learning to be more energy efficient. At the level of learning rules, the brain solves an optimization problem in a highly nonlinear, non-differentiable, temporally stochastic, spiking system with massive numbers of feedback connections, a problem that we arguably still do not know how to efficiently solve for neural networks. At the architectural level, the brain can optimize certain kinds of functions based on very few stimulus presentations, operates over diverse timescales, and clearly uses advanced forms of active learning to infer causal structure in the world.

While we have discussed a range of theories (O'Reilly, 1996; Körding and König, 2001; Hinton, 2007, 2016; Roelfsema et al., 2010; Balduzzi et al., 2014; Lillicrap et al., 2014; O'Reilly et al., 2014a; Bengio et al., 2015a) for how the brain can carry out optimization, these theories are still preliminary. Thus, the first step is to understand whether the brain indeed performs multi-layer credit assignment in a manner that approximates full gradient descent, and if so, how it does this. Either way, we can expect that answer to impact machine learning. If the brain does *not* do some form of backpropagation, this suggests that machine learning may benefit from understanding the tricks that the brain uses to avoid having to do so. If, on the other hand, the brain does do backpropagation, then the underlying mechanisms clearly can support a very wide range of efficient optimization processes across many domains, including learning from rich temporal data-streams and via unsupervised mechanisms, and the architectures behind this will likely be of long-term value to machine learning[61]. Moreover, the search for biologically

[61] Successes of deep learning are already being used, speculatively, to rationalize features of the brain. It has been suggested that large networks, with many more neurons available than are strictly needed for the target computation, make learning easier (Goodfellow et al., 2014b). In concordance with this, visual cortex appears to be a 100-fold over-complete representation of the retinal output (Lewicki and Sejnowski, 2000). Likewise, it has been suggested that biological neurons stabilized (Turrigiano, 2012) to operate far below their saturating firing rates mirror the successful use of rectified linear units in facilitating the training of artificial neural networks (Roudi and Taylor, 2015). Hinton and others have also suggested a biological motivation (Roudi and Taylor, 2015) for "dropout" regularization (Srivastava et al., 2014), in which a fraction of hidden units is stochastically set to zero during each round of training: such a procedure may correspond to the noisiness of neural spike trains, although other theories interpret spikes as sampling in probabilistic inference (Buesing et al., 2011), or in many other ways. Randomness of spiking has some support in neuroscience (Softky and Koch, 1993), although recent experiments suggest that spike trains in certain areas may be less noisy than previously thought (Hires et al., 2015). The key role of proper initialization in enabling effective gradient descent is an important recent finding (Saxe et al., 2013; Sutskever and Martens, 2013) which may also be reflected by biological mechanisms of neural homeostasis or self-organization that would enforce appropriate initial conditions for learning. Retinal

plausible forms of backpropagation has already led to interesting insights, such as the possibility of using random feedback weights (feedback alignment) in backpropagation (Lillicrap et al., 2014), or the unexpected power of internal FORCE learning in chaotic, spontaneously active recurrent networks (Sussillo and Abbott, 2009). This and other findings discussed here suggest that there are still fundamental things we don't understand about backpropagation—which could potentially lead not only to more biologically plausible ways to train recurrent neural networks, but also to fundamentally simpler and more powerful ones.

6.2. *Hypothesis 2*– Biological Fine-structure of Cost Functions

A good practitioner of machine learning has access to a broad range of learning techniques and thus implicitly is able to use many different cost functions. Some problems ask for clustering, others for extracting sparse variables, and yet others for prediction quality to be maximized. The brain also needs to be able to deal with many different kinds of datasets. As such, it makes sense for the brain to use a broad range of cost functions appropriate for the diverse set of tasks it has to solve to thrive in this world.

Many of the most notable successes of deep learning, from language modeling (Sutskever et al., 2011), to vision (Krizhevsky et al., 2012), to motor control (Levine et al., 2015), have been driven by end-to-end optimization of single task objectives. We have highlighted cases where machine learning has opened the door to multiplicities of cost functions that shape network modules into specialized roles. We expect that machine learning will increasingly adopt these practices in the future.

In computer vision, we have begun to see researchers re-appropriate neural networks trained for one task (e.g., ImageNet classification) and then deploy them on new tasks other than the ones they were trained for or for which more limited training data is available (Oquab et al., 2014; Yosinski et al., 2014; Noroozi and Favaro, 2016). We imagine this procedure will be generalized, whereby, in series and in parallel, diverse training problems, each with an associated cost function, are used to shape visual representations. For example, visual data streams can be segmented into elements like foreground vs. background, objects that can move of their own accord vs. those that cannot, all using diverse unsupervised criteria (Ullman et al., 2012; Poggio, 2015). Networks so trained can then be shared, augmented, and retrained on new tasks. They can be introduced as front-ends for systems that perform more complex objectives or even serve to produce cost functions for training other circuits (Watter et al., 2015). As a simple example, a network that can discriminate between images of different kinds of architectural structures (pyramid, staircase, etc.) could act as a critic for a building-construction network.

Scientifically, determining the order in which cost functions are engaged in the biological brain will inform machine learning about how to construct systems with intricate and hierarchical behaviors via divide-and-conquer approaches to learning problems, active learning, and more.

6.3. *Hypothesis 3*– Embedding within a Pre-structured Architecture

A good practitioner of machine learning should have a broad range of algorithms at their disposal. Some problems are efficiently solved through dynamic programming, others through hashing, and yet others through multi-layer backpropagation. The brain needs to be able to solve a broad range of learning problems without the luxury of being reprogrammed. As such, it makes sense for the brain to have specialized structures that allow it to rapidly learn to approximate a broad range of algorithms.

The first neural networks were simple single-layer systems, either linear or with limited non-linearities (Rashevsky, 1939). The explosion of neural network research in the 1980s (Rumelhart et al., 1986) saw the advent of multilayer networks, followed by networks with layer-wise specializations as in convolutional networks (Fukushima, 1980; LeCun and Bengio, 1995). In the last two decades, architectures with specializations for holding variables stable in memory like the LSTM (Hochreiter and Schmidhuber, 1997), the control of content-addressable memory (Graves et al., 2014; Weston et al., 2014), and game playing by reinforcement learning (Mnih et al., 2015) have been developed. These networks, though formerly exotic, are now becoming mainstream algorithms in the toolbox of any deep learning practitioner. There is no sign that progress in developing new varieties of structured architectures is halting, and the heterogeneity and modularity of the brain's circuitry suggests that diverse, specialized architectures are needed to solve the diverse challenges that confront a behaving animal.

The brain combines a jumble of specialized structures in a way that works. Solving this problem *de novo* in machine learning promises to be very difficult, making it attractive to be inspired by observations about how the brain does it. An understanding of the breadth of specialized structures, as well as the architecture that combines them, should be quite useful.

7. DID EVOLUTION SEPARATE COST FUNCTIONS FROM OPTIMIZATION ALGORITHMS?

Deep learning methods have taken the field of machine learning by storm. Driving the success is the separation of the problem of learning into two pieces: (1) An algorithm, backpropagation, that allows efficient distributed optimization, and (2) Approaches to turn any given problem into an optimization problem, by designing a cost function and training procedure which will result in the desired computation. If we want to apply deep learning to a new domain, e.g., playing Jeopardy, we do not need to change the optimization algorithm—we just need to cleverly set up the right cost function. A lot of work in deep learning, perhaps the majority, is now focused on setting up the right cost functions.

We hypothesize that the brain also acquired such a separation between optimization mechanisms and cost functions. If neural

fixation has been tentatively connected with robustness of convolutional networks to adversarial perturbations in images (Luo et al., 2015). But making these speculative claims of biological relevance more rigorous will require researchers to first evaluate *whether* biological neural circuits are performing multi-layer optimization of cost functions in the first place.

circuits, such as in cortex, implement a general-purpose optimization algorithm, then any improvement to that algorithm will improve function across the cortex. At the same time, different cortical areas solve different problems, so tinkering with each area's cost function is likely to improve its performance. As such, functionally and evolutionarily separating the problems of optimization and cost function generation could allow evolution to produce better computations, faster. For example, common unsupervised mechanisms could be combined with area-specific reinforcement-based or supervised mechanisms and error signals, much as recent advances in machine learning have found natural ways to combine supervised and unsupervised objectives in a single system (Rasmus and Berglund, 2015).

This suggests interesting questions[62]: When did the division between cost functions and optimization algorithms occur? How is this separation implemented? How did innovations in cost functions and optimization algorithms evolve? And how do our own cost functions and learning algorithms differ from those of other animals?

There are many possibilities for how such a separation might be achieved in the brain. Perhaps the six-layered cortex represents a common optimization algorithm, which in different cortical areas is supplied with different cost functions. This claim is different from the claim that all cortical areas use a single unsupervised learning algorithm and achieve functional specificity by tuning the inputs to that algorithm. In that case, both the optimization mechanism and the implicit unsupervised cost function would be the same across areas (e.g., minimization of prediction error), with only the training data differing between areas, whereas in our suggestion, the optimization mechanism would be the same across areas but the cost function, *as well as* the training data, would differ. Thus the cost function itself would be like an ancillary input to a cortical area, in addition to its input and output data. Some cortical microcircuits could then, perhaps, compute the cost functions that are to be delivered to other cortical microcircuits. Another possibility is that, within the same circuitry, certain aspects of the wiring and learning rules specify an optimization mechanism and are relatively fixed across areas, while others specify the cost function and are more variable. This latter possibility would be similar to the notion of cortical microcircuits as molecularly and structurally configurable elements, akin to the cells in a field-programmable gate array (FPGA) (Marcus et al., 2014a,b), rather than a homogenous substrate. The biological nature of such a separation, if any exists, remains an open question. For example, individual parts of a neuron may separately deal with optimization and with the specification of the cost function, or different parts of a microcircuit may specialize in this way, or there may be specialized types of cells, some of which deal with signal processing and others with cost functions.

8. CONCLUSIONS

Due to the complexity and variability of the brain, pure "bottom up" analysis of neural data faces potential challenges of interpretation (Robinson, 1992; Jonas and Kording, 2016). Theoretical frameworks can potentially be used to constrain the space of hypotheses being evaluated, allowing researchers to first address higher-level principles and structures in the system, and then "zoom in" to address the details. Proposed "top down" frameworks for understanding neural computation include entropy maximization, efficient encoding, faithful approximation of Bayesian inference, minimization of prediction error, attractor dynamics, modularity, the ability to subserve symbolic operations, and many others (Pinker, 1999; Marcus, 2001; Bialek, 2002; Knill and Pouget, 2004; Bialek et al., 2006; Friston, 2010). Interestingly, many of the "top down" frameworks boil down to assuming that the brain simply optimizes a single, given cost function for a single computational architecture. We generalize these proposals assuming both a heterogeneous combination of cost functions unfolding over development, and a diversity of specialized sub-systems.

Much of neuroscience has focused on the search for "the neural code," i.e., it has asked which stimuli are good at driving activity in individual neurons, regions, or brain areas. But, if the brain is capable of generic optimization of cost functions, then we need to be aware that rather simple cost functions can give rise to complicated stimulus responses. This potentially leads to a different set of questions. Are differing cost functions indeed a useful way to think about the differing functions of brain areas? How does the optimization of cost functions in the brain actually occur, and how is this different from the implementations of gradient descent in artificial neural networks? What additional constraints are present in the circuitry that remain fixed while optimization occurs? How does optimization interact with a structured architecture, and is this architecture similar to what we have sketched? Which computations are wired into the architecture, which emerge through optimization, and which arise from a mixture of those two extremes? To what extent are cost functions explicitly computed in the brain, vs. implicit in its local learning rules? Did the brain evolve to separate the

[62]It would be interesting to study these questions in specific brain systems. The primary visual cortex, for example, is still only understood very incompletely (Olshausen and Field, 2004). It serves as a key input modality to both the ventral and dorsal visual pathways, one of which seems to specialize in object identity and the other in motion and manipulation. Higher-level areas like STP draw on both streams to perform tasks like complex action recognition. In some models (e.g., Jhuang et al., 2007), both ventral and dorsal streams are structured hierarchically, but the ventral stream primarily makes use of the spatial filtering properties of V1, whereas the dorsal stream primarily makes use of its spatio-*temporal* filtering properties, e.g., temporal frequency filtering by the space-time receptive fields of V1 neurons. Given this, we can ask interesting questions about V1. Within a framework of multilayer optimization, do both dorsal and ventral pathways impose cost functions that help to shape V1's response properties? Or is V1 largely pre-structured by genetics and local self-organization, with different optimization principles in the ventral and dorsal streams only having effects at higher levels of the hierarchy? Or, more likely, is there some interplay between pre-structuring of the V1 circuitry and optimization according to multiple cost functions? Relatedly, what establishes the differing roles of the downstream ventral vs. dorsal cortical areas, and can their differences be attributed to differing cost functions? This relates to ongoing questions about the basic nature of cortical circuitry. For example, DiCarlo et al. (2012) suggests that visual cortical regions containing on the order of 10000 neurons are locally optimized to perform disentangling of the manifolds corresponding to their local views of the transformations of an object, allowing these manifolds to be linearly separated by readout areas. Yet, DiCarlo et al. (2012) also emphasizes the possibility that certain computations such as normalization are pre-initialized in the circuitry prior to learning-based optimization.

mechanisms involved in cost function generation from those involved in the optimization of cost functions, and if so how? What kinds of meta-level learning might the brain apply, to learn when and how to invoke different cost functions or specialized systems, among the diverse options available, to solve a given task? What crucial mechanisms are left out of this framework? A more in-depth dialog between neuroscience and machine learning could help elucidate some of these questions.

Much of machine learning has focused on finding ever faster ways of doing end-to-end gradient descent in neural networks. Neuroscience may inform machine learning at multiple levels. The optimization algorithms in the brain have undergone a couple of hundred million years of evolution. Moreover, the brain may have found ways of using heterogeneous cost functions that interact over development so as to simplify learning problems by guiding and shaping the outcomes of unsupervised learning. Lastly, the specialized structures evolved in the brain may inform us about ways of making learning efficient in a world that requires a broad range of computational problems to be solved over multiple timescales. Looking at the insights from neuroscience may help machine learning move toward general intelligence in a structured heterogeneous world with access to only small amounts of supervised data.

In some ways our proposal is opposite to many popular theories of neural computation. There is not one mechanism of optimization but (potentially) many, not one cost function but a host of them, not one kind of a representation but a representation of whatever is useful, and not one homogeneous structure but a large number of them. All these elements are held together by the optimization of internally generated cost functions, which allows these systems to make good use of one another. Rejecting simple unifying theories is in line with a broad range of previous approaches in AI. For example, Minsky and Papert's work on the Society of Mind (Minsky, 1988)—and more broadly on ideas of genetically staged and internally bootstrapped development in connectionist systems (Minsky, 1977)—emphasizes the need for a system of internal monitors and critics, specialized communication and storage mechanisms, and a hierarchical organization of simple control systems.

At the time these early works were written, it was not yet clear that gradient-based optimization could give rise to powerful feature representations and behavioral policies. One can view our proposal as a renewed argument against simple end-to-end training and in favor of a heterogeneous approach. In other words, this framework could be viewed as proposing a kind of "society" of cost functions and trainable networks, permitting internal bootstrapping processes reminiscent of the Society of Mind (Minsky, 1988). In this view, intelligence is enabled by many computationally specialized structures, each trained with its own developmentally regulated cost function, where both the structures and the cost functions are themselves optimized by evolution like the hyperparameters in neural networks.

AUTHOR CONTRIBUTION

All authors contributed ideas and co-wrote the paper.

ACKNOWLEDGMENTS

We thank Ken Hayworth for key discussions that led to this paper. We thank Ed Boyden, Chris Eliasmith, Gary Marcus, Shimon Ullman, Tomaso Poggio, Josh Tenenbaum, Dario Amodei, Alex Williams, Erik Peterson, Tom Dean, David Sussillo, Matthew Botvinick, Joscha Bach, Mohammad Gheshlaghi Azar, Joshua Glaser, Marco Nardini, Ali Hummos, David Markowitz, David Rolnick, Sam Rodriques, Nick Barry, Matthew Larkum, Walter Senn, Eric Drexler, Vikash Mansinghka, Darcy Wayne, Lyra and Neo Marblestone, and all of the participants of a Kavli Salon on Cortical Computation (Feb/Oct 2015) for helpful comments. We thank Miles Brundage for an excellent Twitter feed of deep learning papers. We acknowledge the support of NIH grant R01MH103910.

REFERENCES

Abbott, L., DePasquale, B., and Memmesheimer, R. (2016). *Building Functional Networks of Spiking Model Neurons*. Available online at: neurotheory.columbia.edu.

Abbott, L. F., and Blum, K. I. (1996). Functional significance of long-term potentiation for sequence learning and prediction. *Cereb. Cortex* 6, 406–416.

Ackley, D., Hinton, G., and Sejnowski, T. (1958) A learning algorithm for Boltzmann machines. *Cogn. Sci.* 9, 147–169.

Acuna, D. E., Wymbs, N. F., Reynolds, C. A., Picard, N., Turner, R. S., Strick, P. L., et al. (2014). Multifaceted aspects of chunking enable robust algorithms. *J. Neurophysiol.* 112, 1849–1856. doi: 10.1152/jn.00028.2014

Alain, G., Lamb, A., Sankar, C., Courville, A., and Bengio, Y. (2015). Variance reduction in SGD by distributed importance sampling. arXiv:1511.06481.

Allen, K., Ibara, S., and Seymour, A. (2010). Abstract structural representations of goal-directed behavior. *Psychol. Sci.* 21, 1518–1524. doi: 10.1177/0956797610383434

Anderson, C. H., and Van Essen, D. C. (1987). Shifter circuits: a computational strategy for dynamic aspects of visual processing. *Proc. Natl. Acad. Sci. U.S.A.* 84, 6297–6301.

Anderson, J. R. (2007). *How Can the Human Mind Occur in the Physical Universe?* Oxford, UK: Oxford University Press.

Andreas, J., Rohrbach, M., Darrell, T., and Klein, D. (2015). Deep compositional question answering with neural module networks. arXiv:151 1.02799.

Andreas, J., Rohrbach, M., Darrell, T., and Klein, D. (2016). Learning to compose neural networks for question answering. arXiv:1601.01705.

Angelucci, A., Levitt, J. B., Walton, E. J., Hupe, J.-M., Bullier, J., and Lund, J. S. (2002). Circuits for local and global signal integration in primary visual cortex. *J. Neurosci.* 22, 8633–8646. Available online at: http://www.jneurosci.org/content/22/19/8633.long

Anselmi, F., Leibo, J. Z., Rosasco, L., Mutch, J., Tacchetti, A., and Poggio, T. (2015). Unsupervised learning of invariant representations. *Theor. Comput. Sci.* 633, 112–121. doi: 10.1016/j.tcs.2015.06.048

Antic, S. D., Zhou, W.-L., Moore, A. R., Short, S. M., and Ikonomu, K. D. (2010). The decade of the dendritic nmda spike. *J. Neurosci. Res.* 88, 2991–3001. doi: 10.1002/jnr.22444

Arancio, O., Kiebler, M., Lee, C. J., Lev-Ram, V., Tsien, R. Y., Kandel, E. R., et al. (1996). Nitric oxide acts directly in the presynaptic neuron to produce long-term potentiation in cultured hippocampal neurons. *Cell* 87, 1025–1035.

Aronov, D., Veit, L., Goldberg, J. H., and Fee, M. S. (2011). Two distinct modes of forebrain circuit dynamics underlie temporal patterning in the vocalizations of young songbirds. *J. Neurosci.* 31, 16353–16368. doi: 10.1523/JNEUROSCI.3009-11.2011

Arora, S., Liang, Y., and Ma, T. (2015). Why are deep nets reversible: a simple theory, with implications for training. arXiv:1511.05653.

Ashby, F. G., Ennis, J. M., and Spiering, B. J. (2007). A neurobiological theory of automaticity in perceptual categorization. *Psychol. Rev.* 114:632. doi: 10.1037/0033-295X.114.3.632

Ashby, F. G., Turner, B. O., and Horvitz, J. C. (2010). Cortical and basal ganglia contributions to habit learning and automaticity. *Trends Cogn. Sci.* 14, 208–215. doi: 10.1016/j.tics.2010.02.001

Aso, Y., Hattori, D., Yu, Y., Johnston, R. M., Iyer, N. A., Ngo, T.-T. B., et al. (2014a). The neuronal architecture of the mushroom body provides a logic for associative learning. *eLife* 3:e04577. doi: 10.7554/eLife.04577

Aso, Y., Sitaraman, D., Ichinose, T., Kaun, K. R., Vogt, K., Belliart-Guérin, G., et al. (2014b). Mushroom body output neurons encode valence and guide memory-based action selection in Drosophila. *eLife* 3:e04580. doi: 10.7554/eLife.04580

Bach, J. (2015). "Modeling motivation in MicroPsi 2," in *8th International Conference, AGI 2015, AGI 2015, Berlin, Germany, Proceedings*, Vol. 9205 (Cham: Springer International Publishing), 3–13.

Bach, J., and Herger, P. (2015). "Request confirmation networks for neuro-symbolic script execution," in *Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches at NIPS*, eds T. Besold, A. Avila Garcez, G. Marcus, and R. Miikkulainen (Montreal, QC).

Baillargeon, R., Scott, R. M., and Bian, L. (2016). Psychological reasoning in infancy. *Annu. Rev. Psychol.* 67, 159–186. doi: 10.1146/annurev-psych-010213-115033

Ba, J., and Caruana, R. (2014). Do deep nets really need to be deep? *Adv. Neural Inform. Process.* 27, 2654–2662. Available online at: https://papers.nips.cc/paper/5484-do-deep-nets-really-need-to-be-deep

Balaguer, J., Spiers, H., Hassabis, D., and Summerfield, C. (2016). Neural mechanisms of hierarchical planning in a virtual subway network. *Neuron* 90, 893–903. doi: 10.1016/j.neuron.2016.03.037

Baldauf, D., and Desimone, R. (2014). Neural mechanisms of object-based attention. *Science* 344, 424–427. doi: 10.1126/science.1247003

Baldi, P., and Sadowski, P. (2015). The Ebb and flow of deep learning: a theory of local learning. arXiv:1506.06472.

Balduzzi, D. (2014). "Cortical prediction markets," in *Proceedings of the 2014 International Conference on Autonomous AgentsMultiagent Systems (AAMAS)* (Paris).

Balduzzi, D., Vanchinathan, H., and Buhmann, J. (2014). Kickback cuts Backprop's red-tape: biologically plausible credit assignment in neural networks. arXiv:1411.6191.

Bargmann, C. I. (2012). Beyond the connectome: how neuromodulators shape neural circuits. *Bioessays* 34, 458–465. doi: 10.1002/bies.201100185.

Bargmann, C. I., and Marder, E. (2013). From the connectome to brain function. *Nat. Methods* 10, 483–490. doi: 10.1038/nmeth.2451

Bastos, A. M., Vezoli, J., Bosman, C. A., Schoffelen, J.-M., Oostenveld, R., Dowdall, J. R., et al. (2015). Visual areas exert feedforward and feedback influences through distinct frequency channels. *Neuron* 85, 390–401. doi: 10.1016/j.neuron.2014.12.018.

Battaglia, P. W., Hamrick, J. B., and Tenenbaum, J. B. (2013). Simulation as an engine of physical scene understanding. *Proc. Natl. Acad. Sci. U.S.A.* 110, 18327–18332. doi: 10.1073/pnas.1306572110

Becker, S., and Hinton, G. E. (1992). Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature* 355, 161–163.

Bekolay, T., Kolbeck, C., and Eliasmith, C. (2013). "Simultaneous unsupervised and supervised learning of cognitive functions in biologically plausible spiking neural networks," in *Proceedings of the 35th Annual Conference of the Cognitive Science Society* (Berlin), 169–174.

Bengio, Y. (2014). How auto-encoders could provide credit assignment in deep networks via target propagation. arXiv:1407.7906.

Bengio, Y., and Fischer, A. (2015). Early inference in energy-based models approximates back-propagation. arXiv:1510.02777.

Bengio, Y., Lee, D.-H., Bornschein, J., and Lin, Z. (2015a). Towards biologically plausible deep learning. arXiv:1502.04156.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). "Curriculum learning," in *Proceedings of the 26th Annual International Conference on Machine Learning* (Montreal, QC), 41–48.

Bengio, Y., Mesnard, T., Fischer, A., Zhang, S., and Wu, Y. (2015b). STDP as presynaptic activity times rate of change of postsynaptic activity. arXiv:1509.05936.

Bengio, Y., Scellier, B., Bilaniuk, O., Sacramento, J., and Senn, W. (2016). Feedforward initialization for fast inference of deep generative networks is biologically plausible. arXiv:1606.01651.

Berezovskii, V. K., Nassi, J. J., and Born, R. T. (2011). Segregation of feedforward and feedback projections in mouse visual cortex. *J. Comp. Neurol.* 519, 3672–3683. doi: 10.1002/cne.22675

Berwick, R. C., Beckers, G. J. L., Okanoya, K., and Bolhuis, J. J. (2012). A bird's eye view of human language evolution. *Front. Evol. Neurosci.* 4:5. doi: 10.3389/fnevo.2012.00005

Bialek, W. (2002). "Thinking about the brain," in *Physics of Bio-Molecules and Cells*, Vol. 75, eds F. Flyvbjerg, F. Jülicher, P. Ormos, and F. David (Berlin; Heidelberg: Springer), 485–578.

Bialek, W., De Ruyter Van Steveninck, R., and Tishby, N. (2006). "Efficient representation as a design principle for neural coding and computation," in *2006 IEEE International Symposium on Information Theory* (Los Alamitos: IEEE), 659–663.

Bloss, E. B., Cembrowski, M. S., Karsh, B., Colonell, J., Fetter, R. D., and Spruston, N. (2016). Structured dendritic inhibition supports branch-selective integration in CA1 pyramidal cells. *Neuron* 89, 1016–1030. doi: 10.1016/j.neuron.2016.01.029

Blundell, C., Uria, B., Pritzel, A., Li, Y., Ruderman, A., Leibo, J. Z., et al. (2016). Model-free episodic control. arXiv:1606.04460.

Bobier, B., Stewart, T. C., and Eliasmith, C. (2014). A unifying mechanistic model of selective attention in spiking neurons. *PLoS Comput. Biol.* 10:e1003577. doi: 10.1371/journal.pcbi.1003577

Bostrom, N. (1996). Cortical integration: possible solutions to the binding and linking problems in perception, reasoning and long term memory. Available online at: http://www.nickbostrom.com/old/cortical.html

Botvinick, M. M., Niv, Y., and Barto, A. C. (2009). Hierarchically organized behavior and its neural foundations: a reinforcement learning perspective. *Cognition* 113, 262–280. doi: 10.1016/j.cognition.2008.08.011

Botvinick, M., and Weinstein, A. (2014). Model-based hierarchical reinforcement learning and human action control. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* 369:20130480. doi: 10.1098/rstb.2013.0480

Bouchard, G., Trouillon, T., Perez, J., and Gaidon, A. (2015). Accelerating stochastic gradient descent via online learning to sample. arXiv:1506.09016.

Bourdoukan, R., and Denève, S. (2015). "Enforcing balance allows local supervised learning in spiking recurrent networks," in *Advances in Neural Information Processing Systems* (Montreal, QC), 982–990.

Braitenberg, V., and Schutz, A. (1991). *Anatomy of the Cortex: Studies of Brain Function*. Berlin: Springer.

Brea, J., Gaál, A. T., Urbanczik, R., and Senn, W. (2016). Prospective coding by spiking neurons. *PLoS Comput. Biol.* 12:e1005003. doi: 10.1371/journal.pcbi.1005003

Brea, J., and Gerstner, W. (2016). Does computational neuroscience need new synaptic learning paradigms? *Curr. Opin. Behav. Sci.* 11, 61–66. doi: 10.1016/j.cobeha.2016.05.012

Bremner, J. G., Slater, A. M., and Johnson, S. P. (2015). Perception of object persistence: the origins of object permanence in infancy. *Child Dev. Perspect.* 9, 7–13. doi: 10.1111/cdep.12098

Brito, C. S., and Gerstner, W. (2016). Nonlinear hebbian learning as a unifying principle in receptive field formation. arXiv:1601.00701.

Brosch, T., Neumann, H., and Roelfsema, P. R. (2015). Reinforcement learning of linking and tracing contours in recurrent neural networks. *PLoS Comput. Biol.* 11:e1004489. doi: 10.1371/journal.pcbi.1004489

Brownstone, R. M., Bui, T. V., and Stifani, N. (2015). Spinal circuits for motor learning. *Curr. Opin. Neurobiol.* 33, 166–173. doi: 10.1016/j.conb.2015.04.007

Brown, T. I., Carr, V. A., LaRocque, K. F., Favila, S. E., Gordon, A. M., Bowles, B., et al. (2016). Prospective representation of navigational goals in the human hippocampus. *Science* 352, 1323–1326. doi: 10.1126/science.aaf0784.

Buesing, L., Bill, J., Nessler, B., and Maass, W. (2011). Neural dynamics as sampling: a model for stochastic computation in recurrent networks of spiking neurons. *PLoS Comput. Biol.* 7:e1002211. doi: 10.1371/journal.pcbi.1002211

Buonomano, D. V., and Merzenich, M. M. (1995). Temporal information transformed into a spatial code by a neural network with realistic properties. *Science* 267, 1028–1030.

Bülthoff, H., Little, J., and Poggio, T. (1989). A parallel algorithm for real-time computation of optical flow. *Nature* 337, 549–553.

Buschman, T. J., and Miller, E. K. (2010). Shifting the spotlight of attention: evidence for discrete computations in cognition. *Front. Hum. Neurosci.* 4:194. doi: 10.3389/fnhum.2010.00194

Buschman, T. J., and Miller, E. K. (2014). Goal-direction and top-down control. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* 369:20130471. doi: 10.1098/rstb.2013.0471

Bush, K. A. (2007). *An Echo State Model of Non-markovian Reinforcement Learning*. Doctoral Dissertation. Colorado State University.

Buzsáki, G. (2010). Neural syntax: cell assemblies, synapsembles, and readers. *Neuron* 68, 362–385. doi: 10.1016/j.neuron.2010.09.023

Buzsáki, G., and Moser, E. I. (2013). Memory, navigation and theta rhythm in the hippocampal-entorhinal system. *Nat. Neurosci.* 16, 130–138. doi: 10.1038/nn.3304

Cai, D. J., Aharoni, D., Shuman, T., Shobe, J., Biane, J., Song, W., et al. (2016). A shared neural ensemble links distinct contextual memories encoded close in time. *Nature* 534, 115–118. doi: 10.1038/nature17955

Callaway, E. (2004). Feedforward, feedback and inhibitory connections in primate visual cortex. *Neural Netw.* 17, 625–632. doi: 10.1016/j.neunet.2004.04.004

Cappe, C., Rouiller, E. M., and Barone, P. (2012). "The neural bases of multisensory processes," in *Cortical and Thalamic Pathways for Multisensory and Sensorimotor Interplay*, eds M. M. Murray and M. T. Wallace (Boca Raton, FL: CRC Press/Taylor & Francis).

Caron, S. J. C., Ruta, V., Abbott, L. F., and Axel, R. (2013). Random convergence of olfactory inputs in the Drosophila mushroom body. *Nature* 497, 113–117. doi: 10.1038/nature12063

Chen, L.-C., Schwing, A. G., Yuille, A. L., and Urtasun, R. (2014). Learning deep structured models. arXiv:1407.2538.

Chikkerur, S., Serre, T., Tan, C., and Poggio, T. (2010). What and where: a Bayesian inference theory of attention. *Vis. Res.* 50, 2233–2247. doi: 10.1016/j.visres.2010.05.013

Choo, F. X., and Eliasmith, C. (2010). "A spiking neuron model of serial-order recall," in *32nd Annual Conference of the Cognitive Science Society* (Portland).

Chubykin, A. A., Roach, E. B., Bear, M. F., and Shuler, M. G. H. (2013). A cholinergic mechanism for reward timing within primary visual cortex. *Neuron* 77, 723–735. doi: 10.1016/j.neuron.2012.12.039

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv:1412.3555.

Cichon, J., and Gan, W.-B. (2015). Branch-specific dendritic ca2$^+$ spikes cause persistent synaptic plasticity. *Nature* 520, 180–185. doi: 10.1038/nature14251

Clayton, N. S., and Dickinson, A. (1998). Episodic-like memory during cache recovery by scrub jays. *Nature* 395, 272–274.

Clopath, C., and Gerstner, W. (2010). Voltage and spike timing interact in STDP–a unified model. *Front. Synaptic Neurosci.* 2:25. doi: 10.3389/fnsyn.2010.00025

Cohn, R., Morantte, I., and Ruta, V. (2015). Coordinated and compartmentalized neuromodulation shapes sensory processing in drosophila. *Cell* 163, 1742–1755. doi: 10.1016/j.cell.2015.11.019

Colino, A., and Halliwell, J. (1987). Differential modulation of three separate K-conductances in hippocampal CA1 neurons by serotonin. *Nature* 328, 73–77. doi: 10.1038/328073a0

Constantinescu, A. O., O'Reilly, J. X., and Behrens, T. E. (2016). Organizing conceptual knowledge in humans with a gridlike code. *Science* 352, 1464–1468. doi: 10.1126/science.aaf0941

Cox, D. D., and Dean, T. (2014). Neural networks and neuroscience-inspired computer vision. *Curr. Biol.* 24, R921–R929. doi: 10.1016/j.cub.2014.08.026

Crick, F. (1989). The recent excitement about neural networks. *Nature* 337, 129–132.

Crick, F. C., and Koch, C. (2005). What is the function of the claustrum? *Philos. Trans. R. Soc. Lond. B Biol. Sci.* 360, 1271–1279. doi: 10.1098/rstb.2005.1661

Crouzet, S. M., and Thorpe, S. J. (2011). Low-level cues and ultra-fast face detection. *Front. Psychol.* 2:342. doi: 10.3389/fpsyg.2011.00342

Cui, Y., Surpur, C., Ahmad, S., and Hawkins, J. (2015). Continuous online sequence learning with an unsupervised neural network model. arXiv:1512.05463.

Dadarlat, M. C., O'Doherty, J. E., and Sabes, P. N. (2015). A learning-based approach to artificial sensory feedback leads to optimal integration. *Nat. Neurosci.* 18, 138–144. doi: 10.1038/nn.3883

Danihelka, I., Wayne, G., Uria, B., Kalchbrenner, N., and Graves, A. (2016). Associative long short-term memory. arXiv:1602.03032.

Daw, N. D., Niv, Y., and Dayan, P. (2006). "Actions, policies, values and the basal ganglia," in *Recent Breakthroughs in Basal Ganglia Research*, ed E. Bezard (Hauppauge, NY: Nova Science), 91–106.

Dayan, P. (2012). Twenty-five lessons from computational neuromodulation. *Neuron* 76, 240–256. doi: 10.1016/j.neuron.2012.09.027

Dean, T. (2005). "A computational model of the cerebral cortex," in *Proceedings of the 20th National Conference on Artificial Intelligence* (Pittsburg, PA).

Dehaene, S., Meyniel, F., Wacongne, C., Wang, L., and Pallier, C. (2015). The neural representation of sequences: from transition probabilities to algebraic patterns and linguistic trees. *Neuron* 88, 2–19. doi: 10.1016/j.neuron.2015.09.019

Dekker, T. M., and Nardini, M. (2015). Risky visuomotor choices during rapid reaching in childhood. *Dev. Sci.* 19, 427–439. doi: 10.1111/desc.12322

Delalleau, O., and Bengio, Y. (2011). "Shallow vs. deep sum-product networks," in *Advances in Neural Information Processing Systems* (Grenada), 666–674.

DePasquale, B., Churchland, M., and Abbott, L. (2016). Using firing-rate dynamics to train recurrent networks of spiking model neurons. arXiv:1601.07620.

DeWolf, T., and Eliasmith, C. (2011). The neural optimal control hierarchy for motor control. *J. Neural Eng.* 8:065009. doi: 10.1088/1741-2560/8/6/065009

DiCarlo, J. J., Zoccolan, D., and Rust, N. C. (2012). How does the brain solve visual object recognition? *Neuron* 73, 415–434. doi: 10.1016/j.neuron.2012.01.010

Douglas, R. J., and Martin, K. A. C. (2004). Neuronal circuits of the neocortex. *Annu. Rev. Neurosci.* 27, 419–451. doi: 10.1146/annurev.neuro.27.070203.144152

Doya, K. (1999). What are the computations of the cerebellum, the basal ganglia and the cerebral cortex? *Neural Netw.* 12, 961–974.

Dudman, J. T., Tsay, D., and Siegelbaum, S. A. (2007). A role for synaptic inputs at distal dendrites: instructive signals for hippocampal long-term plasticity. *Neuron* 56, 866–879. doi: 10.1016/j.neuron.2007.10.020

Dumoulin, S. O., and Wandell, B. A. (2008). Population receptive field estimates in human visual cortex. *Neuroimage* 39, 647–660. doi: 10.1016/j.neuroimage.2007.09.034

Eliasmith, C. (2013). *How to Build a Brain: A Neural Architecture for Biological Cognition*. Oxford, UK: Oxford University Press.

Eliasmith, C., and Anderson, C. H. (2004). *Neural Engineering: Computation, Representation, and Dynamics in Neurobiological Systems*. Cambridge, MA: MIT Press.

Eliasmith, C., and Martens, J. (2011). Normalization for probabilistic inference with neurons. *Biol. Cybern.* 104, 251–262. doi: 10.1007/s00422-011-0433-y

Eliasmith, C., Stewart, T. C., Choo, X., Bekolay, T., DeWolf, T., Tang, Y., et al. (2012). A large-scale model of the functioning brain. *Science* 338, 1202–1205.

Emlen, S. T. (1967). Migratory orientation in the indigo bunting, passerina cyanea: part i: evidence for use of celestial cues. *Auk* 84, 309–342.

Enel, P., Procyk, E., Quilodran, R., and Dominey, P. F. (2016). Reservoir computing properties of neural dynamics in prefrontal cortex. *PLoS Comput. Biol.* 12:e1004967. doi: 10.1371/journal.pcbi.1004967

Erhan, D., and Manzagol, P. (2009). The difficulty of training deep architectures and the effect of unsupervised pre-training. *Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS), JMLR Workshop and Conference Procedings* (Clearwater Beach, FL), 153–160.

Eslami, S., Heess, N., and Weber, T. (2016). Attend, infer, repeat: fast scene understanding with generative models. arXiv:1603.08575.

Fausey, C. M., Jayaraman, S., and Smith, L. B. (2016). From faces to hands: changing visual input in the first two years. *Cognition* 152, 101–107. doi: 10.1016/j.cognition.2016.03.005

Felleman, D. J., and Van Essen, D. C. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cereb. cortex* 1, 1–47.

Ferster, D., and Miller, K. D. (2000). Neural mechanisms of orientation selectivity in the visual cortex. *Annu. Rev. Neurosci.* 23, 441–471. doi: 10.1146/annurev.neuro.23.1.441

Fetz, E. E. (1969). Operant conditioning of cortical unit activity. *Science* 163, 955–958.

Fetz, E. E. (2007). Volitional control of neural activity: implications for brain–computer interfaces. *J. Physiol.* 579, 571–579. doi: 10.1113/jphysiol.2006.127142

Fiete, I. R., Fee, M. S., and Seung, H. S. (2007). Model of birdsong learning based on gradient estimation by dynamic perturbation of neural conductances. *J. Neurophysiol.* 98, 2038–2057. doi: 10.1152/jn.01311.2006

Fiete, I. R., Senn, W., Wang, C. Z. H., and Hahnloser, R. H. R. (2010). Spike-time-dependent plasticity and heterosynaptic competition organize networks to produce long scale-free sequences of neural activity. *Neuron* 65, 563–576.

Fiete, I. R., and Seung, H. S. (2006). Gradient learning in spiking neural networks by dynamic perturbation of conductances. *Phys. Rev. Lett.* 97:048104. doi: 10.1103/PhysRevLett.97.048104

Finnerty, G., and Shadlen, M., Jazayeri M., Nobre A. C., Buonomano D. V. (2015). Time in Cortical Circuits. *J. Neurosci.* 35, 13912–13916. doi: 10.1523/JNEUROSCI.2654-15.2015

Finn, C., Levine, S., and Abbeel, P. (2016). Guided cost learning: deep inverse optimal control via policy optimization. arXiv:1603.00448.

Fodor, J. D., and Crowther, C. (2002). Understanding stimulus poverty arguments. *Ling. Rev.* 18, 105–145. doi: 10.1515/tlir.19.1-2.105

Földiák, P. (2008). Learning invariance from transformation sequences. *J. Neural Comput.* 3, 194–200. doi: 10.1162/neco.1991.3.2.194

Foster, D. J., Morris, R. G. M., Dayan, P. (2000). Models of hippocampally dependent navigation, using the temporal difference learning rule. *Hippocampus* 10, 1–16. doi: 10.1002/(SICI)1098-1063(2000)10:1<1::AID-HIPO1<3.0.CO;2-1

Fournier, J., Müller, C. M., and Laurent, G. (2015). Looking for the roots of cortical sensory computation in three-layered cortices. *Curr. Opin. Neurobiol.* 31, 119–126. doi: 10.1016/j.conb.2014.09.006

Franconeri, S. L., Pylyshyn, Z. W., and Scholl, B. J. (2012). A simple proximity heuristic allows tracking of multiple objects through occlusion. *Atten. Percept. Psychophys.* 74, 691–702. doi: 10.3758/s13414-011-0265-9

Frankland, S. M., and Greene, J. D. (2015). An architecture for encoding sentence meaning in left mid-superior temporal cortex. *Proc. Natl. Acad. Sci. U.S.A.* 112, 11732–11737. doi: 10.1073/pnas.1421236112

Frank, M. J., and Badre, D. (2012). Mechanisms of hierarchical reinforcement learning in corticostriatal circuits 1: computational analysis. *Cereb. Cortex* 22, 509–526. doi: 10.1093/cercor/bhr114

Franzius, M., Sprekeler, H., and Wiskott, L. (2007). Slowness and sparseness lead to place, head-direction, and spatial-view cells. *PLoS Comput. Biol.* 3:e166. doi: 10.1371/journal.pcbi.0030166

Friston, K. (2010). The free-energy principle: a unified brain theory? *Nat. Rev. Neurosci.* 11, 127–138. doi: 10.1038/nrn2787

Friston, K. J., and Stephan, K. E. (2007). Free-energy and the brain. *Synthese* 159, 417–458. doi: 10.1007/s11229-007-9237-y

Fukushima, K. (1980). Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybern.* 36, 193–202. doi: 10.1007/BF00344251

Galtier, M. N., and Wainrib, G. (2013). A biological gradient descent for prediction through a combination of stdp and homeostatic plasticity. *Neural Comput.* 25, 2815–2832. doi: 10.1162/NECO_a_00512

Gao, T., Harari, D., Tenenbaum, J., and Ullman, S. (2014). When computer vision gazes at cognition. arXiv:1412.2672.

Gemp, I., and Mahadevan, S. (2015). "Modeling context in cognition using variational inequalities," in *Modeling Changing Perspectives—Reconceptualizing Sensorimotor Experiences: Papers from the 2014 AAAI Fall Symposium* (Arlington, TX).

George, D., and Hawkins, J. (2009). Towards a mathematical theory of cortical micro-circuits. *PLoS Comput. Biol.* 5:e1000532. doi: 10.1371/journal.pcbi.1000532

Gershman, S. J., and Beck, J. M. (2016). "Complex probabilistic inference: from cognition to neural computation," in *Computational Models of Brain and Behavior*, ed A. Moustafa (Hoboken, NJ: Wiley-Blackwell).

Gershman, S. J., Moore, C. D., Todd, M. T., Norman, K. A., and Sederberg, P. B. (2012). The successor representation and temporal context. *Neural Comput.* 24, 1553–1568. doi: 10.1162/NECO_a_00282

Gershman, S. J., Moustafa, A. A., and Ludvig, E. A. (2014). Time representation in reinforcement learning models of the basal ganglia. *Front. Comput. Neurosci.* 7:194. doi: 10.3389/fncom.2013.00194

Ghahramani, I. M. Z. (2005). *A Note On the Evidence and Bayesian Occam's Razor*. Gatsby Unit Technical Report GCNU-TR 2005–003.

Giocomo, L. M., Zilli, E. A., Fransén, E., and Hasselmo, M. E. (2007). Temporal frequency of subthreshold oscillations scales with entorhinal grid cell field spacing. *Science* 315, 1719–1722. doi: 10.1126/science.1139207

Giret, N., Kornfeld, J., Ganguli, S., and Hahnloser, R. H. R. (2014). Evidence for a causal inverse model in an avian cortico-basal ganglia circuit. *Proc. Natl. Acad. Sci. U.S.A.* 111, 6063–6068. doi: 10.1073/pnas.1317087111

Goertzel, B. (2014). "How might the brain represent complex symbolic knowledge?" in *International Joint Conference on Neural Networks (IJCNN)* (Beijing).

Goldman, M. S., Levine, J. H., Major, G., Tank, D. W., and Seung, H. (2003). Robust persistent neural activity in a model integrator with multiple hysteretic dendrites per neuron. *Cereb. Cortex* 13, 1185–1195. doi: 10.1093/cercor/bhg095

Gonzalez Andino, S. L. and Grave de Peralta Menendez, R. (2012). Coding of saliency by ensemble bursting in the amygdala of primates. *Front. Behav. Neurosci.* 6:38. doi: 10.3389/fnbeh.2012.00038

Gooch, C. M., Wiener, M., Wencil, E. B., and Coslett, H. B. (2010). Interval timing disruptions in subjects with cerebellar lesions. *Neuropsychologia* 48, 1022–1031.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014a). Generative adversarial networks. arXiv:1406.2661.

Goodfellow, I. J., Vinyals, O., and Saxe, A. M. (2014b). Qualitatively characterizing neural network optimization problems. arXiv:1412.6544.

Gopnik, A., Meltzoff, A. N., and Kuhl, P. K. (2000). *The Scientist in the Crib: What Early Learning Tells us About the Mind*. New York, NY: Harper Paperbacks.

Graves, A., Wayne, G., and Danihelka, I. (2014). Neural Turing Machines. arXiv:1410.5401.

Graybiel, A. M. (1998). The basal ganglia and chunking of action repertoires. *Neurobiol. Learn. Mem.* 70, 119–136.

Gregor, K., Danihelka, I., Graves, A., Rezende, D. J., and Wierstra, D. (2015). DRAW: a recurrent neural network for image generation. arXiv:1502.04623.

Grillner, S., Hellgren, J., Ménard, A., Saitoh, K., and Wikström, M. A. (2005). Mechanisms for selection of basic motor programs–roles for the striatum and pallidum. *Trends Neurosci.* 28, 364–370. doi: 10.1016/j.tins.2005.05.004

Grossberg, S. (2013). Adaptive resonance theory: how a brain learns to consciously attend, learn, and recognize a changing world. *Neural Netw.* 37, 1–47. doi: 10.1016/j.neunet.2012.09.017

Güçlü, U., and van Gerven, M. A. (2015). Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. *J. Neurosci.* 35, 10005–10014. doi: 10.1523/JNEUROSCI.5023-14.2015

Guez, A., Silver, D., and Dayan, P. (2012). Efficient bayes-adaptive reinforcement learning using sample-based search. arXiv:1205.3109.

Gülçehre, Ç., and Bengio, Y. (2016). Knowledge matters: importance of prior information for optimization. *J. Mach. Learn. Res.* 17, 1–32. Available online at: http://jmlr.org/papers/v17/gulchere16a.html

Güntürkün, O., and Bugnyar, T. (2016). Cognition without cortex. *Trends Cogn. Sci.* 20, 291–303. doi: 10.1016/j.tics.2016.02.001

Gurney, K., Prescott, T. J., and Redgrave, P. (2001). A computational model of action selection in the basal ganglia. I. A new functional anatomy. *Biol. Cybern.* 84, 401–410. doi: 10.1007/PL00007984

Hadley, R. F. (2009). The problem of rapid variable creation. *Neural Comput.* 21, 510–532. doi: 10.1162/neco.2008.07-07-572

Hamlin, J. K., Wynn, K., and Bloom, P. (2007). Social evaluation by preverbal infants. *Nature* 450, 557–559. doi: 10.1038/nature06288

Hangya, B., Ranade, S., Lorenc, M., and Kepecs, A. (2015). Central cholinergic neurons are rapidly recruited by reinforcement feedback. *Cell* 162, 1155–1168. doi: 10.1016/j.cell.2015.07.057

Hanuschkin, A., Ganguli, S., and Hahnloser, R. H. R. (2013). A hebbian learning rule gives rise to mirror neurons and links them to control theoretic inverse models. *Front. Neural Circ.* 7:106. doi: 10.3389/fncir.2013.00106

Harris, C. M., and Wolpert, D. M. (1998). Signal-dependent noise determines motor planning. *Nature* 394, 780–784.

Harris, K. (2008). Stability of the fittest: organizing learning through retroaxonal signals. *Trends Neurosci.* 31, 130–136. doi: 10.1016/j.tins.2007.12.002

Hassabis, D., and Maguire, E. (2009). The construction system of the brain. *Philos. Trans. R. Soc. B.* 364, 1263–1271. doi: 10.1098/rstb.2008.0296

Hassabis, D., and Maguire, E. A. (2007). Deconstructing episodic memory with construction. *Trends Cogn. Sci.* 11, 299–306. doi: 10.1016/j.tics.2007.05.001

Hasselmo, M. E. (2006). The role of acetylcholine in learning and memory. *Curr. Opin. Neurobiol.* 16, 710–715. doi: 10.1016/j.conb.2006.09.002

Hasselmo, M. E. (2015). If i had a million neurons: Potential tests of cortico-hippocampal theories. *Progr. Brain Res.* 219, 1–19. doi: 10.1016/bs.pbr.2015.03.009

Hasselmo, M. E., and Stern, C. E. (2015). Current questions on space and time encoding. *Hippocampus* 25, 744–752. doi: 10.1002/hipo.22454

Hasselmo, M. E., and Wyble, B. P. (1997). Free recall and recognition in a network model of the hippocampus: simulating effects of scopolamine on human memory function. *Behav. Brain Res.* 89, 1–34.

Hattori, D., Demir, E., Kim, H. W., Viragh, E., Zipursky, S. L., and Dickson, B. J. (2007). Dscam diversity is essential for neuronal wiring and self-recognition. *Nature* 449, 223–227. doi: 10.1038/nature06099

Hawkins, J., and Ahmad, S. (2016). Why neurons have thousands of synapses, a theory of sequence memory in neocortex. *Front. Neural Circ.* 10:23. doi: 10.3389/fncir.2016.00023

Hawkins, J., and Blakeslee, S. (2007). *On Intelligence*. New York, NY: Henry Holt and Company.

Hayashi-Takagi, A., Yagishita, S., Nakamura, M., Shirai, F., Wu, Y. I., Loshbaugh, A. L., et al. (2015). Labelling and optical erasure of synaptic memory traces in the motor cortex. *Nature* 525, 333–338. doi: 10.1038/nature15257

Haykin, S. S. (1994). *Neural Networks: A Comprehensive Foundation*. New York, NY: Macmillan.

Hayworth, K. J. (2012). Dynamically partitionable autoassociative networks as a solution to the neural binding problem. *Front. Comput. Neurosci.* 6:73. doi: 10.3389/fncom.2012.00073

Hayworth, K. J., Lescroart, M. D., and Biederman, I. (2011). Neural encoding of relative position. *J. Exp. Psychol. Hum. Percept. Perform.* 37, 1032–1050. doi: 10.1037/a0022338

Hennequin, G., Vogels, T. P., and Gerstner, W. (2014). Optimal control of transient dynamics in balanced networks supports generation of complex movements. *Neuron* 82, 1394–1406. doi: 10.1016/j.neuron.2014.04.045

Herd, S. A., Krueger, K. A., Kriete, T. E., Huang T. R., Hazy T. E., and O'Reilly R. C. (2013). Strategic cognitive sequencing: a computational cognitive neuroscience approach. *Comput. Intell. Neurosci.* 2013:149329. doi: 10.1155/2013/149329

Higgins, I., Matthey, L., Glorot, X., Pal, A., Uria, B., Blundell, C., et al. (2016). Early visual concept learning with unsupervised deep learning. arXiv:1606.05579 .

Hinton, G. (1989). Connectionist learning procedures. *Artif. Intell.* 40, 185–234. doi: 10.1016/0004-3702(89)90049-0

Hinton, G. (2007). "How to do backpropagation in a brain," in *Invited Talk at the NIPS'2007 Deep Learning Workshop* (Vancouver, BC).

Hinton, G. (2016). "Can the brain do back-propagation?," in *Invited talk at Stanford University Colloquium on Computer Systems* (Stanford, CA).

Hinton, G. E., Dayan, P., Frey, B. J., and Neal, R. M. (1995). The "wake-sleep" algorithm for unsupervised neural networks. *Science* 268, 1158–1161.

Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Comput.* 18, 1527–1554. doi: 10.1162/neco.2006.18.7.1527

Hinton, G., and McClelland, J. (1988). Learning representations by recirculation. *Neural information processing*.

Hinton, G. E., Krizhevsky, A., and Wang, S. (2011). "Transforming auto-encoders," in *Artificial Neural Networks and Machine Leaning*, eds T. Honkela, W. Duch, M. Girolami, and S. Kaski (Helsinki), 44–51.

Hires, S. A., Gutnisky, D. A., Yu, J., O'Connor, D. H., and Svoboda, K. (2015). Low-noise encoding of active touch by layer 4 in the somatosensory cortex. *eLife* 4. doi: 10.7554/eLife.06619

Histed, M. H., Ni, A. M., and Maunsell, J. H. (2013). Insights into cortical mechanisms of behavior from microstimulation experiments. *Progr. Neurobiol.* 103, 115–130. doi: 10.1016/j.pneurobio.2012.01.006

Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.* 9, 1735–1780.

Hoerzer, G. M., Legenstein, R., and Maass, W. (2014). Emergence of complex computational structures from chaotic neural networks through reward-modulated Hebbian learning. *Cereb. Cortex* 24, 677–690. doi: 10.1093/cercor/bhs348

Ho, J., and Ermon, S. (2016). Generative adversarial imitation learning. arXiv:1606.03476.

Hong, W., and Luo, L. (2014). Genetic control of wiring specificity in the fly olfactory system. *Genetics* 196, 17–29. doi: 10.1534/genetics.113.154336

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. U.S.A.* 79, 2554–2558.

Hopfield, J. J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. Natl. Acad. Sci. U.S.A.* 81, 3088–3092.

Hopfield, J. J. (2009). Neurodynamics of mental exploration. *Proc. Natl. Acad. Sci. U.S.A.* 107, 1648–1653. doi: 10.1073/pnas.0913991107

Hosoya, T., Baccus, S. A., and Meister, M. (2005). Dynamic predictive coding by the retina. *Nature* 436, 71–77. doi: 10.1038/nature03689

Huang, Y., and Rao, R. (2011). Predictive coding. *Wiley Interdiscip. Rev. Cogn. Sci.* 2, 580–593. doi: 10.1002/wcs.142

Huys, Q. J., Lally, N., Faulkner, P., Eshel, N., Seifritz, E., Gershman, S. J.,et al. (2015). Interplay of approximate planning strategies. *Proc. Natl. Acad. Sci. U.S.A.* 112, 3098–3103. doi: 10.1073/pnas.1414219112.

Isik, L., Leibo, J. Z., and Poggio, T. (2012). Learning and disrupting invariance in visual recognition with a temporal association rule. *Front. Comput. Neurosci.* 6:37. doi: 10.3389/fncom.2012.00037

Izhikevich, E. M. (2006). Polychronization: computation with spikes. *Neural Comput.* 18, 245–282. doi: 10.1162/089976606775093882

Izhikevich, E. M. (2007). Solving the distal reward problem through linkage of STDP and dopamine signaling. *Cereb. Cortex* 17, 2443–2452. doi: 10.1093/cercor/bhl152

Jacobson, G. A., and Friedrich, R. W. (2013). Neural circuits: random design of a higher-order olfactory projection. *Curr. Biol.* 23, R448–R451. doi: 10.1016/j.cub.2013.04.016

Jaderberg, M., Czarnecki, W. M., Osindero, S., Vinyals, O., Graves, A., and Kavukcuoglu, K. (2016). Decoupled neural interfaces using synthetic gradients. arXiv:1608.05343.

Jaderberg, M., Simonyan, K., Zisserman, A., and Kavukcuoglu, K. (2015). "Spatial transformer networks," in *Advances in Neural Information Processing Systems 28 (NIPS 2015)*. arXiv:1506.02025.

Jaeger, H., and Haas, H. (2004). Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. *Science* 304, 78–80. doi: 10.1126/science.1091277

Jara-Ettinger, J., Gweon, H., Schulz, L. E., and Tenenbaum, J. B. (2016). The naïve utility calculus: computational principles underlying commonsense psychology. *Trends Cogn. Sci.* 20, 589–604. doi: 10.1016/j.tics.2016.05.011

Jaramillo, S., and Pearlmutter, B. A. (2004). A normative model of attention: receptive field modulation. *Neurocomputing* 58, 613–618. doi: 10.1016/j.neucom.2004.01.103

Jhuang, H., Serre, T., Wolf, L., and Poggio, T. (2007). "A biologically inspired system for action recognition," in *IEEE 11th International Conference on Computer Vision, 2007* (Rio de Janeiro: ICCV), 2007, 1–8.

Jiang, X., Shen, S., Cadwell, C. R., Berens, P., Sinz, F., Ecker, A. S., et al. (2015). Principles of connectivity among morphologically defined cell types in adult neocortex. *Science* 350, aac9462. doi: 10.1126/science.aac9462

Ji, D., and Wilson, M. A. (2007). Coordinated memory replay in the visual cortex and hippocampus during sleep. *Nat. Neurosci.* 10, 100–107. doi: 10.1038/nn1825

Johansson, F., Jirenhed, D.-A., Rasmussen, A., Zucca, R., and Hesslow, G. (2014). Memory trace and timing mechanism localized to cerebellar Purkinje cells. *Proc. Natl. Acad. Sci. U.S.A.* 111, 14930–14934. doi: 10.1073/pnas.1415371111

Jonas, E., and Kording, K. (2016). Could a neuroscientist understand a microprocessor? *bioRxiv*. doi: 10.1101/055624

Joulin, A., and Mikolov, T. (2015). Inferring algorithmic patterns with stack-augmented recurrent nets. arXiv:1503.01007.

Kalisman, N., Silberberg, G., and Markram, H. (2005). The neocortical microcircuit as a tabula rasa. *Proc. Natl. Acad. Sci. U.S.A.* 102, 880–885. doi: 10.1073/pnas.0407088102

Kanwisher, N., McDermott, J., and Chun, M. M. (1997). The fusiform face area: a module in human extrastriate cortex specialized for face perception. *J. Neurosci.* 17, 4302–4311.

Kappel, D., Habenschuss, S., Legenstein, R., and Maass, W. (2015). Network plasticity as bayesian inference. *PLoS Comput. Biol.* 11:e1004485. doi: 10.1371/journal.pcbi.1004485

Kappel, D., Nessler, B., and Maass, W. (2014). STDP installs in Winner-Take-All circuits an online approximation to hidden Markov model learning. *PLoS Comput. Biol.* 10:e1003511. doi: 10.1371/journal.pcbi.1003511

Kempter, R., Gerstner, W., and van Hemmen, J. L. (2001). Intrinsic stabilization of output rates by spike-based Hebbian learning. *Neural Comput.* 13, 2709–2741. doi: 10.1162/089976601317098501

Khaligh-Razavi, S.-M., and Kriegeskorte, N. (2014). Deep supervised, but not unsupervised, models may explain it cortical representation. *PLoS Comput. Biol.* 10:e1003915. doi: 10.1371/journal.pcbi.1003915

Kingma, D. P., and Welling, M. (2013). Auto-Encoding Variational Bayes. arXiv:1312.6114.

Knill, D., and Pouget, A. (2004). The Bayesian brain: the role of uncertainty in neural coding and computation. *Trends Neurosci.* 27, 712–719. doi: 10.1016/j.tins.2004.10.007

Koechlin, E., and Jubault, T. (2006). Broca's area and the hierarchical organization of human behavior. *Neuron* 50, 963–974. doi: 10.1016/j.neuron.2006.05.017

Komer, B., and Eliasmith, C. (2016). A unified theoretical approach for biological cognition and learning. *Curr. Opin. Behav. Sci.* 11, 14–20. doi: 10.1016/j.cobeha.2016.03.006

Körding, K. (2007). Decision theory: what "should" the nervous system do? *Science* 318, 606–610. doi: 10.1126/science.1142998

Körding, K., and König, P. (2000). A learning rule for dynamic recruitment and decorrelation. *Neural Netw.* 13, 1–9. doi: 10.1016/S0893-6080(99)00088-X

Körding, K. P., and König, P. (2001). Supervised and unsupervised learning with two sites of synaptic integration. *J. Comput. Neurosci.* 11, 207–215. doi: 10.1023/A:1013776130161

Körding, K. P., Kayser, C., Einhäuser, W., and König, P. (2004). How are complex cell properties adapted to the statistics of natural stimuli? *J. Neurophysiol.* 91, 206–212. doi: 10.1152/jn.00149.2003

Kouh, M., and Poggio, T. (2008). A canonical neural circuit for cortical nonlinear operations. *Neural Comput.* 20, 1427–1451. doi: 10.1162/neco.2008.02-07-466

Kraus, B. J., Robinson, R. J. II, White, J. A., Eichenbaum, H., and Hasselmo, M. E. (2013). Hippocampal time cells: time versus path integration. *Neuron* 78, 1090–1101. doi: 10.1016/j.neuron.2013.04.015

Kriegeskorte, N., Mur, M., and Bandettini, P. A. (2008). Representational similarity analysis-connecting the branches of systems neuroscience. *Front. Syst. Neurosci.* 2:4. doi: 10.3389/neuro.06.004.2008

Kriete, T., Noelle, D. C., Cohen, J. D., and O'Reilly, R. C. (2013). Indirection and symbol-like processing in the prefrontal cortex and basal ganglia. *Proc. Natl. Acad. Sci. U.S.A.* 110, 16390–16935. doi: 10.1073/pnas.1303547110

Krishnamurthy, R., Lakshminarayanan, A. S., Kumar, P., and Ravindran, B. (2016). Hierarchical reinforcement learning using spatio-temporal abstractions and deep neural networks. arXiv:1605.05359.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems* (Lake Tahoe, CL), 1097–1105.

Kulkarni, T. D., Narasimhan, K. R., Saeedi, A., and Tenenbaum, J. B. (2016). Hierarchical deep reinforcement learning: integrating temporal abstraction and intrinsic motivation. arXiv:1604.06057.

Kulkarni, T. D., Whitney, W., Kohli, P., and Tenenbaum, J. B. (2015). Deep Convolutional Inverse Graphics Network. arXiv:1503.03167.

Kumaran, D., Hassabis, D., and McClelland, J. L. (2016). What learning systems do intelligent agents need? complementary learning systems theory updated. *Trends Cogn. Sci.* 20, 512–534. doi: 10.1016/j.tics.2016.05.004

Kumaran, D., Summerfield, J. J., Hassabis, D., and Maguire, E. A. (2009). Tracking the emergence of conceptual knowledge during human decision making. *Neuron* 63, 889–901. doi: 10.1016/j.neuron.2009.07.030

Kurach, K., Andrychowicz, M., and Sutskever, I. (2015). Neural Random-Access Machines. arXiv:1511.06392.

Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. (2015). Human-level concept learning through probabilistic program induction. *Science* 350, 1332–1338. doi: 10.1126/science.aab3050

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. (2016). Building machines that learn and think like people. arXiv:1604.00289.

Larkum, M. (2013). A cellular mechanism for cortical associations: an organizing principle for the cerebral cortex. *Trends Neurosci.* 36, 141–151. doi: 10.1016/j.tins.2012.11.006

LeCun, Y., and Bengio, Y. (1995). "Convolutional networks for images, speech, and time series," in *The Handbook of Brain Theory and Neural Networks*, ed M. A. Arbib (Cambridge, MA: MIT Press), 3361.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature* 521, 436–444. doi: 10.1038/nature14539

Lee, A. M., Tai, L.-H., Zador, A., and Wilbrecht, L. (2015). Between the primate and 'reptilian' brain: rodent models demonstrate the role of corticostriatal circuits in decision making. *Neuroscience* 296, 66–74. doi: 10.1016/j.neuroscience.2014.12.042

Lee, T., and Yuille, A. (2011). Efficient coding of visual scenes by grouping and segmentation: theoretical predictions and biological evidence. *Department of Statistics, UCLA*.

Lee, T. S., and Mumford, D. (2003). Hierarchical Bayesian inference in the visual cortex. *J. Opt. Soc. Am. A Opt. Image Sci. Vis.* 20, 1434–1448. doi: 10.1364/JOSAA.20.001434

Legenstein, R., and Maass, W. (2011). Branch-specific plasticity enables self-organization of nonlinear computation in single neurons. *J. Neurosci.* 31, 10787–10802. doi: 10.1523/JNEUROSCI.5684-10.2011

Leibo, J. Z., Cornebise, J., Gómez, S., and Hassabis, D. (2015a). Approximate hubel-wiesel modules and the data structures of neural computation. arXiv:1512.08457v1.

Leibo, J. Z., Liao, Q., Anselmi, F., and Poggio, T. (2015b). The invariance hypothesis implies domain-specific regions in visual cortex. *PLoS Comput. Biol.* 11:e1004390. doi: 10.1371/journal.pcbi.1004390

Le, Q., Ranzato, M., Monga, R., Devin, M., Chen, K., Corrado, G., et al. (2012). "Building high-level features using large scale unsupervised learning," in *International Conference in Machine Learning* (Edinburg).

Lettvin, J., Maturana, H., McCulloch, W., and Pitts, W. (1959). What the frog's eye tells the frog's brain. *Proc. IRE* 47, 1940–1951.

Letzkus, J. J., Kampa, B. M., and Stuart, G. J. (2006). Learning rules for spike timing-dependent plasticity depend on dendritic synapse location. *J. Neurosci.* 26, 10420–10429. doi: 10.1523/JNEUROSCI.2650-06.2006

Levine, S., Finn, C., Darrell, T., and Abbeel, P. (2015). End-to-end training of deep visuomotor policies. arXiv:1504.00702.

Lewicki, M. S., and Sejnowski, T. J. (2000). Learning overcomplete representations. *Neural Comput.* 12, 337–365. doi: 10.1162/089976600300015826

Lewis, S. N., and Harris, K. D. (2014). *The Neural Marketplace: I. General Formalism and Linear Theory*. Technical Report. bioRxiv:013185.

Liao, Q., and Poggio, T. (2016). Bridging the gaps between residual learning, recurrent neural networks and visual cortex. arXiv:1604.03640.

Liao, Q., Leibo, J. Z., and Poggio, T. (2015). How important is weight symmetry in backpropagation? arXiv:1510.05067.

Lillicrap, T. P., Cownden, D., Tweed, D. B., and Akerman, C. J. (2014). Random feedback weights support learning in deep neural networks. arXiv:1411.0247.

Li, N., and Dicarlo, J. J. (2012). Neuronal learning of invariant object representation in the ventral visual stream is not dependent on reward. *J. Neurosci.* 32, 6611–6620. doi: 10.1523/JNEUROSCI.3786-11.2012

Liu, J. K., and Buonomano, D. V. (2009). Embedding multiple trajectories in simulated recurrent neural networks in a self-organizing manner. *J. Neurosci.* 29, 13172–13181. doi: 10.1523/JNEUROSCI.2358-09.2009

Livni, R., Shalev-Shwartz, S., and Shamir, O. (2013). An algorithm for training polynomial networks. arXiv:1304.7045.

Lotter, W., Kreiman, G., and Cox, D. (2015). Unsupervised learning of visual structure using predictive generative networks. arXiv:1511.06380.

Lotter, W., Kreiman, G., and Cox, D. (2016). Deep predictive coding networks for video prediction and unsupervised learning. arXiv:1605.08104.

Luck, S. J., and Vogel, E. K. (1997). The capacity of visual working memory for features and conjunctions. *Nature* 390, 279–281.

Luo, Y., Boix, X., Roig, G., Poggio, T., and Zhao, Q. (2015). Foveation-based mechanisms alleviate adversarial examples. arXiv:1511.06292.

Lyons, A. B., and Cheries, E. W. (2016). Inferring social disposition by sound and surface appearance in infancy. *J. Cogn. Dev.* doi: 10.1080/15248372.2016.1200048

Ma, W. J., Beck, J. M., Latham, P. E., and Pouget, A. (2006). Bayesian inference with probabilistic population codes. *Nat. Neurosci.* 9, 1432–1438. doi: 10.1038/nn1790

Maass, W. (2016) Searching for principles of brain computation. *Curr. Opin. Behav. Sci.* 11, 81–92. doi: 10.1016/j.cobeha.2016.06.003

Maass, W., Joshi, P., and Sontag, E. D. (2007). Computational aspects of feedback in neural circuits. *PLoS Comput. Biol.* 3:e165. doi: 10.1371/journal.pcbi.0020165

Maass, W., Natschläger, T., and Markram, H. (2002). Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Comput.* 14, 2531–2560. doi: 10.1162/089976602760407955

MacDonald, C. J., Lepage, K. Q., Eden, U. T., and Eichenbaum, H. (2011). Hippocampal time cells bridge the gap in memory for discontiguous events. *Neuron* 71, 737–749. doi: 10.1016/j.neuron.2011.07.012

Maclaurin, D., Duvenaud, D., and Adams, R. (2015). Gradient-based hyperparameter optimization through reversible learning. arXiv:1502.03492.

Makin, J. G., Dichter, B. K., and Sabes, P. N. (2016). Recurrent exponential-family harmoniums without backprop-through-time. arXiv:1605.05799.

Makin, J. G., Fellows, M. R., and Sabes, P. N. (2013). Learning multisensory integration and coordinate transformation via density estimation. *PLoS Comput. Biol.* 9:e1003035. doi: 10.1371/journal.pcbi.1003035

Mandelblat-Cerf, Y., Las, L., Denisenko, N., and Fee, M. S. (2014). A role for descending auditory cortical projections in songbird vocal learning. *eLife* 3:e02152. doi: 10.7554/eLife.02152

Mansinghka, V., and Jonas, E. (2014). Building fast bayesian computing machines out of intentionally stochastic, digital parts. arXiv:1402.4914.

Marblestone, A. H., and Boyden, E. S. (2014). Designing tools for assumption-proof brain mapping. *Neuron* 83, 1239–1241. doi: 10.1016/j.neuron.2014.09.004

Marcus, G. (2001). *The Algebraic Mind: Integrating Connectionism and Cognitive Science*. Cambridge, MA: MIT Press.

Marcus, G. (2004). *The Birth of the Mind: How a Tiny Number of Genes Creates the Complexities of Human Thought*. New York, NY: Basic Books.

Marcus, G., Marblestone, A., and Dean, T. (2014a). Frequently asked question for: the atoms of neural computation. arXiv:1410.8826.

Marcus, G., Marblestone, A., and Dean, T. (2014b). The atoms of neural computation. *Science* 346, 551–552. doi: 10.1126/science.1261661

Marder, E., and Goaillard, J.-M. (2006). Variability, compensation and homeostasis in neuron and network function. *Nat. Rev. Neurosci.* 7, 563–574. doi: 10.1038/nrn1949

Markowitz, D. A., Curtis, C. E., and Pesaran, B. (2015). Multiple component networks support working memory in prefrontal cortex. *Proc. Natl. Acad. Sci. U.S.A.* 112, 11084–11089. doi: 10.1073/pnas.1504172112

Markram, H., Lübke, J., Frotscher, M., and Sakmann, B. (1997). Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science* 275, 213–215.

Markram, H., Muller, E., Ramaswamy, S., Reimann, M. W., Abdellah, M., Sanchez, C. A., et al. (2015). Reconstruction and simulation of neocortical microcircuitry. *Cell* 163, 456–492. doi: 10.1016/j.cell.2015.09.029

Marr, D. (1969). A theory of cerebellar cortex. *J. Physiol.* 202, 437–470.

Martens, J., and Sutskever, I. (2011). "Learning recurrent neural networks with hessian-free optimization," in *Proceedings of the 28th International Conference on Machine Learning* (Bellevue, WA).

McCandliss, B. D., Cohen, L., and Dehaene, S. (2003). The visual word form area: expertise for reading in the fusiform gyrus. *Trends Cogn. Sci.* 7, 293–299. doi: 10.1016/S1364-6613(03)00134-7

McCulloch, W. S., and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* 5, 115–133. doi: 10.1007/BF02478259

McKinstry, J. L., Edelman, G. M., and Krichmar, J. L. (2006). A cerebellar model for predictive motor control tested in a brain-based device. *Proc. Natl. Acad. Sci. U.S.A.* 103, 3387–3392. doi: 10.1073/pnas.0511281103

McKone, E., Crookes, K., and Kanwisher, N. (2009). "The cognitive and neural development of face recognition in humans," in *The Cognitive Neurosciences, 4th Edn*, eds S. G. Michael (Cambridge, MA: MIT Press), 4, 467–482.

McLeod, P., and Dienes, Z. (1996). Do fielders know where to go to catch the ball or only how to get there? *J. Exp. Psychol. Hum. Percept. Perform.* 22, 531.

Mel, B. (1992). The clusteron: toward a simple abstraction for a complex neuron. *Adv. Neural Inf. Process. Syst.* 4, 35–42.

Meltzoff, A. N. (1999). Born to learn: what infants learn from watching us. *Role Early Exp. Infant Dev.* 145–164.

Meltzoff, A. N., Waismeyer, A., and Gopnik, A. (2012). Learning about causes from people: observational causal learning in 24-month-old infants. *Dev. Psychol.* 48, 1215–1258. doi: 10.1037/a0027440

Meltzoff, A. N., Williamson, R. A., and Marshall, P. J. (2013). "11 developmental perspectives on action science: lessons from infant imitation and cognitive neuroscience," in *Action Science: Foundations of an Emerging Discipline*, eds W. Prinz, M. Beisert, and A. Herwig (Cambridge, MA: MIT Press), 281–306.

Miller, G. A. (1956). The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychol. Rev.* 63, 81–97.

Miller, K. D., Keller, J. B., and Stryker, M. P. (1989). Ocular dominance column development: analysis and simulation. *Science*, 245, 605–615.

Miller, K. D., and MacKay, D. J. C. (1994). The role of constraints in hebbian learning. *Neural Comput.* 6, 100–126.

Minsky, M. (1977). "Plain talk about neurodevelopmental epistemology," in *IJCAI'77 Proceedings of the 5th International Joint Conference on Artificial Intelligence* (San Francisco, CA: Morgan Kaufmann Publishers Inc.), 1083–1092. Available online at: http://hdl.handle.net/1721.1/5763

Minsky, M. (1988). *Society of Mind*. New York, NY: Simon and Schuster.

Minsky, M. (2006). The Emotion Machine. New York, NY: Pantheon.

Minsky, M. L. (1991). Logical versus analogical or symbolic versus connectionist or neat versus scruffy. *AI magazine* 12, 34–51.

Minsky, M. L., and Papert, S. (1972). *Perceptrons: An Introduction to Computational Geometry*. Cambridge, MA: MIT Press.

Mishra, R. K., Kim, S., Guzman, S. J., and Jonas, P. (2016). Symmetric spike timing-dependent plasticity at ca3-ca3 synapses optimizes storage and recall in autoassociative networks. *Nat. Commun.* 7:1552. doi: 10.1038/ncomms11552

Mitchell, T. M. (1980). "The need for biases in learning generalizations," in *Readings in Machine Learning*, eds J. W. Shavlik and T. G. Dietterich (Morgan Kauffman), 184–191. Available online at: http://www.cs.nott.ac.uk/~bsl/G52HPA/articles/Mitchell:80a.pdf

Miyagawa, S., Berwick, R. C., and Okanoya, K. (2013). The emergence of hierarchical structure in human language. *Front. Psychol.* 4:71. doi: 10.3389/fpsyg.2013.00071

Mnih, V., Heess, N., Graves, A., and Kavukcuoglu, K. (2014). "Recurrent models of visual attention," in *Advances in Neural Information Processing Systems* (Montreal, QC), 2204–2212.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., et al. (2015). Human-level control through deep reinforcement learning. *Nature* 518, 529–533. doi: 10.1038/nature14236

Mobahi, H., Collobert, R., and Weston, J. (2009). "Deep learning from temporal coherence in video," in *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09* (New York, NY: ACM Press), 1–8.

Moberget, T., Gullesen, E. H., Andersson, S., Ivry, R. B., and Endestad, T. (2014). Generalized role for the cerebellum in encoding internal models: evidence from semantic processing. *J. Neurosci.* 34, 2871–2878. doi: 10.1523/JNEUROSCI.2264-13.2014

Mohamed, S., and Rezende, D. J. (2015). Variational information maximisation for intrinsically motivated reinforcement learning. arXiv:1509.08731

Mordatch, I., Todorov, E., and Popović, Z. (2012). Discovery of complex behaviors through contact-invariant optimization. *ACM Trans. Graph.* 31:43. doi: 10.1145/2185520.2185539

Morgan, J. L., Berger, D. R., Wetzel, A. W., and Lichtman, J. W. (2016). The fuzzy logic of network connectivity in mouse visual thalamus. *Cell* 165, 192–206. doi: 10.1016/j.cell.2016.02.033

Mushiake, H., Saito, N., Sakamoto, K., Itoyama, Y., and Tanji, J. (2006). Activity in the lateral prefrontal cortex reflects multiple steps of future events in action plans. *Neuron* 50, 631–641. doi: 10.1016/j.neuron.2006.03.045

Nardini, M., Bedford, R., and Mareschal, D. (2010). Fusion of visual cues is not mandatory in children. *Proc. Natl. Acad. Sci. U.S.A.* 107, 17041–17046. doi: 10.1073/pnas.1001699107

Neelakantan, A., Le, Q. V., and Sutskever, I. (2015). Neural programmer: inducing latent programs with gradient descent. arXiv:1511.04834.

Nessler, B., Pfeiffer, M., Buesing, L., and Maass, W. (2013). Bayesian computation emerges in generic cortical microcircuits through spike-timing-dependent

plasticity. *PLoS Comput. Biol.* 9:e1003037. doi: 10.1371/journal.pcbi.1003037

Ng, A., and Russell, S. (2000). "Algorithms for inverse reinforcement learning," in *ICML '00 Proceedings of the Seventeenth International Conference on Machine Learning* (San Francisco, CA).

Noroozi, M., and Favaro, P. (2016). Unsupervised learning of visual representations by solving jigsaw puzzles. arXiv:1603.09246.

Ólafsdóttir, H. F., Barry, C., Saleem, A. B., Hassabis, D., and Spiers, H. J. (2015). Hippocampal place cells construct reward related sequences through unexplored space. *eLife* 4:e06063. doi: 10.7554/eLife.06063

Ollivier, Y., and Charpiat, G. (2015). Training recurrent networks online without backtracking. arXiv:1507.07680.

Olshausen, B. A., Anderson, C. H., and Van Essen, D. C. (1993). A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information. *J. Neurosci.* 13, 4700–4719.

Olshausen, B. A., and Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* 381, 607–609.

Olshausen, B. A., and Field, D. J. (1997). Sparse coding with an overcomplete basis set: a strategy employed by V1? *Vis. Res.* 37, 3311–3325.

Olshausen, B. A., and Field, D. J. (2004). What is the other 85% of v1 doing. *Prob. Syst. Neurosci.* 4, 182–211. doi: 10.1093/acprof:oso/9780195148220.003.0010

Oquab, M., Bottou, L., Laptev, I., and Sivic, J. (2014). "Learning and transferring mid-level image representations using convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Columbus, OH), 1717–1724.

O'Reilly, R. C. (1996). Biologically plausible error-driven learning using local activation differences: the generalized recirculation algorithm. *Neural Comput.* 8, 895–938.

O'Reilly, R. C. (2006). Biologically based computational models of high-level cognition. *Science* 314, 91–94. doi: 10.1126/science.1127242

O'Reilly, R. C., and Frank, M. J. (2006). Making working memory work: a computational model of learning in the prefrontal cortex and basal ganglia. *Neural Comput.* 18, 283–328. doi: 10.1162/089976606775093909

O'Reilly, R. C., Hazy, T. E., Mollick, J., Mackie, P., and Herd, S. (2014a). Goal-driven cognition in the brain: a computational framework. arXiv:1404.7591.

O'Reilly, R. C., Munakata, Y., Frank, M. J., and Hazy, T. E. (2012). *Computational Cognitive Neuroscience, 1st Edn.* Wiki Book. Available online at: http://ccnbook.colorado.edu

O'Reilly, R. C., Wyatte, D., and Rohrlich, J. (2014b). Learning through time in the thalamocortical loops. arXiv:1407.3432, 37.

Orhan, A. E., and Ma, W. J. (2016). The inevitability of probability: probabilistic inference in generic neural networks trained with non-probabilistic feedback arXiv:1601.03060.

Palmer, L. M., Shai, A. S., Reeve, J. E., Anderson, H. L., Paulsen, O., and Larkum, M. E. (2014). NMDA spikes enhance action potential generation during sensory input. *Nat. Neurosci.* 17, 383–390. doi: 10.1038/nn.3646

Parisien, C., Anderson, C. H., and Eliasmith, C. (2008). Solving the problem of negative synaptic weights in cortical models. *Neural Comput.* 20, 1473–1494. doi: 10.1162/neco.2008.07-06-295

Pasupathy, A., and Miller, E. K. (2005). Different time courses of learning-related activity in the prefrontal cortex and striatum. *Nature* 433, 873–876. doi: 10.1038/nature03287

Patel, A., Nguyen, T., and Baraniuk, R. (2015). A probabilistic theory of deep learning. arXiv:1504.00641.

Pehlevan, C., and Chklovskii, D. B. (2015). "Optimization theory of hebbian/anti-hebbian networks for pca and whitening," in *53rd Annual Allerton Conference on Communication, Control, and Computing* (Monticello, IL), 1458–1465.

Perea, G., Navarrete, M., and Araque, A. (2009). Tripartite synapses: astrocytes process and control synaptic information. *Trends Neurosci.* 32, 421–431. doi: 10.1016/j.tins.2009.05.001

Petrov, A. A., Jilk, D. J., and O'Reilly, R. C. (2010). The Leabra architecture: specialization without modularity. *Behav. Brain Sci.* 33, 286–287. doi: 10.1017/S0140525X10001160

Pezzulo, G., Verschure, P. F. M. J., Balkenius, C., and Pennartz, C. M. A. (2014). The principles of goal-directed decision-making: from neural mechanisms to computation and robotics. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* 369, 20130470. doi: 10.1098/rstb.2013.0470

Pfister, J.-P., and Gerstner, W. (2006). Triplets of spikes in a model of spike timing-dependent plasticity. *J. Neurosci.* 26, 9673–9682. doi: 10.1523/JNEUROSCI.1425-06.2006

Phillips, A. T., Wellman, H. M., and Spelke, E. S. (2002). Infants' ability to connect gaze and emotional expression to intentional action. *Cognition* 85, 53–78. doi: 10.1016/S0010-0277(02)00073-2

Piekniewski, F., Laurent, P., Petre, C., Richert, M., Fisher, D., and Hylton, T. (2016). Unsupervised learning from continuous video in a scalable predictive recurrent network. arXiv:1607.06854.

Pineda, F. J. (1987). Generalization of back-propagation to recurrent neural networks. *Phys. Rev. Lett.* 59:2229. doi: 10.1103/PhysRevLett.59.2229

Pinker, S. (1999). How the mind works. *Ann. N.Y. Acad. Sci.* 882, 119–127.

Plate, T. A. (1995). Holographic reduced representations. *IEEE Trans. Neural Netw.* 6, 623–641.

Poggio, T. (2015). *What if...*, MIT Center for Brains Minds and Machines Memo.

Poggio, T., and Bizzi, E. (2004). Generalization in vision and motor control. *Nature* 431, 768–774. doi:10.1038/nature03014

Ponulak, F., and Hopfield, J. J. (2013). Rapid, parallel path planning by propagating wavefronts of spiking neural activity. *Front. Comput. Neurosci.* 7:98. doi: 10.3389/fncom.2013.00098

Radford, A., Metz, L., and Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv:1511.06434.

Rajan, K., Harvey, C. D., and Tank, D. W. (2016). Recurrent network models of sequence generation and memory. *Neuron* 90, 128–142. doi: 10.1016/j.neuron.2016.02.009

Ramachandran, V. S. (2000). *Mirror Neurons and Imitation Learning as the Driving Force Behind "the Great Leap Forward" in Human Evolution*. Available online at: https://www.edge.org/conversation/mirror-neurons-and-imitation-learning-as-the-driving-force-behind-the-great-leap-forward-in-human-evolution

Rao, R. P. (2004). Bayesian computation in recurrent neural circuits. *Neural Comput.* 16, 1–38. doi: 10.1162/08997660460733976

Rashevsky, N. (1939). Mathematical biophysics: physico-mathematical foundations of biology. *Bull. Amer. Math. Soc.* 45, 223–224. doi: 10.1090/S0002-9904-1939-06963-2

Rasmus, A., and Berglund, M. (2015). Semi-supervised learning with ladder networks. arXiv:1507.02672.

Reynolds, J. H., and Desimone, R. (1999). The role of neural mechanisms of attention in solving the binding problem. *Neuron* 24, 19–29.

Rezende, D. J., Mohamed, S., Danihelka, I., Gregor, K., and Wierstra, D. (2016). One-shot generalization in deep generative models. arXiv:1603.05106.

Rigotti, M., Barak, O., Warden, M. R., Wang, X.-J., Daw, N. D., Miller, E. K., et al. (2013). The importance of mixed selectivity in complex cognitive tasks. *Nature* 497, 585–590. doi: 10.1038/nature12160

Robinson, D. (1992). Implications of neural networks for how we think about brain function. *Behav. Brain Sci.* 15, 644–655.

Rodriguez, A., and Granger, R. (2016). The grammar of mammalian brain capacity. *Theor. Comput. Sci.* 633, 100–111. doi: 10.1016/j.tcs.2016.03.021

Rodriguez, A., Whitson, J., and Granger, R. (2004). Derivation and analysis of basic computational operations of thalamocortical circuits. *J. Cogn. Neurosci.* 16, 856–877. doi: 10.1162/089892904970690

Roelfsema, P. R., van Ooyen, A., and Watanabe, T. (2010). Perceptual learning rules based on reinforcers and attention. *Trends Cogn. Sci.* 14, 64–71. doi: 10.1016/j.tics.2009.11.005

Roelfsema, P. R., and van Ooyen, A. (2005). Attention-gated reinforcement learning of internal representations for classification. *Neural Comput.* 17, 2176–2214. doi: 10.1162/0899766054615699

Rolls, E. T. (2013). The mechanisms for pattern completion and pattern separation in the hippocampus. *Front. Syst. Neurosci.* 7:74. doi: 10.3389/fnsys.2013.00074

Rombouts, J. O., Bohte, S. M., and Roelfsema, P. R. (2015). How attention can create synaptic tags for the learning of working memories in sequential tasks. *PLoS Comput. Biol.* 11:e1004060. doi: 10.1371/journal.pcbi.1004060

Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., and Bengio, Y. (2014). Fitnets: hints for thin deep nets. *arXiv* arXiv:1412.6550.

Roudi, Y., and Taylor, G. (2015). Learning with hidden variables. *Curr. Opin. Neurobiol.* 35, 110–118. doi: 10.1016/j.conb.2015.07.006

Rozell, C. J., Johnson, D. H., Baraniuk, R. G., and Olshausen, B. A. (2008). Sparse coding via thresholding and local competition in neural circuits. *Neural Comput.* 20, 2526–2563. doi: 10.1162/neco.2008.03-07-486

Rubin, A., Geva, N., Sheintuch, L., and Ziv, Y. (2015). Hippocampal ensemble dynamics timestamp events in long-term memory. *eLife* 4:e12247. doi: 10.7554/eLife.12247

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature* 323, 533–536. doi: 10.1038/323533a0

Rumelhart, D. E., and Zipser, D. (1986). "Feature discovery by competitive learning," in *Parallel Distributed Processing*, Vol. 1, eds D. E Rumel hart and J. L. McClelland (Cambridge, MA: MIT Press), 151–163.

Sadtler, P. T., Quick, K. M., Golub, M. D., Chase, S. M., Ryu, S. I., Tyler-Kabara, E. C., et al. (2014). Neural constraints on learning. *Nature* 512, 423–426. doi: 10.1038/nature13665

Sahani, M., and Dayan, P. (2003). Doubly distributional population codes: simultaneous representation of uncertainty and multiplicity. *Neural Comput.* 15, 2255–2279. doi: 10.1162/089976603322362356

Sandler, M., Shulman, Y., and Schiller, J. (2016). A novel form of local plasticity in tuft dendrites of neocortical somatosensory layer 5 pyramidal neurons. *Neuron* 90, 1028–1042. doi: 10.1016/j.neuron.2016.04.032

Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., and Lillicrap, T. (2016). One-shot learning with memory-augmented neural networks. 13. arXiv:1605.06065. Available online at: https://arxiv.org/abs/1605.06065

Saxe, A. M., McClelland, J. L., and Ganguli, S. (2013). Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. arXiv:1312.6120.

Scellier, B., and Bengio, Y. (2016). Towards a biologically plausible backprop. arXiv:1602.05179.

Schiess, M., Urbanczik, R., and Senn, W. (2016). Somato-dendritic synaptic plasticity and error-backpropagation in active dendrites. *PLoS Comput. Biol.* 12:e1004638. doi: 10.1371/journal.pcbi.1004638

Schmidhuber, J. (2010). Formal theory of creativity, fun, and intrinsic motivation (19902010). *Auton. Ment. Dev. IEEE.* 2, 230–247. doi: 10.1109/TAMD.2010.2056368

Schmidhuber, J. (2015). Deep learning in neural networks: an overview. *Neural Netw.* 61, 85–117. doi: 10.1016/j.neunet.2014.09.003

Scholl, B. J. (2004). Can infants' object concepts be trained? *Trends Cogn. Sci.* 8, 49–51. doi: 10.1016/j.tics.2003.12.006

Schwabe, L., Obermayer, K., Angelucci, A., and Bressloff, P. C. (2006). The role of feedback in shaping the extra-classical receptive field of cortical neurons: a recurrent network model. *J. Neurosci.* 26, 9117–9129. doi: 10.1523/JNEUROSCI.1253-06.2006

Sejnowski, T., and Poizner, H. (2014). Prospective optimization. *Proc. IEEE.* 102, 799–811. doi: 10.1109/jproc.2014.2314297

Sermanet, P., and Kavukcuoglu, K. (2013). "Pedestrian detection with unsupervised multi-stage feature learning," in *Proceedings of Computer Vision and Pattern Recognition (CVPR)* (Portland, OR).

Serre, T., Oliva, A., and Poggio, T. (2007). A feedforward architecture accounts for rapid categorization. *Proc. Natl. Acad. Sci. U.S.A.* 104, 6424–6429. doi: 10.1073/pnas.0700622104

Servan-Schreiber, E., and Anderson, J. (1990). Chunking as a mechanism of implicit learning. *J. Exp. Psychol.* 16, 592–608.

Seung, H. S. (1998). Continuous attractors and oculomotor control. *Neural Netw.* 11, 1253–1258. doi: 10.1016/S0893-6080(98)00064-1

Seung, H. S. (2003). Learning in spiking neural networks by reinforcement of stochastic synaptic transmission. *Neuron* 40, 1063–1073. doi: 10.1016/S0896-6273(03)00761-X

Shai, A. S., Anastassiou, C. A., Larkum, M. E., and Koch, C. (2015). Physiology of layer 5 pyramidal neurons in mouse primary visual cortex: coincidence detection through bursting. *PLoS Comput. Biol.* 11:e1004090. doi: 10.1371/journal.pcbi.1004090

Shepherd, G. M. (2014). The microcircuit concept applied to cortical evolution: from three-layer to six-layer cortex. *Front. Neuroanat.* 5:30. doi: 10.3389/fnana.2011.00030

Sherman, S. M. (2005). Thalamic relays and cortical functioning. *Prog. Brain Res.* 149, 107–126. doi: 10.1016/S0079-6123(05)49009-3

Sherman, S. M. (2007). The thalamus is more than just a relay. *Curr. Opin. Neurobiol.* 17, 417–422. doi: 10.1016/j.conb.2007.07.003

Shimizu, T., and Karten, H. J. (2013). Multiple origins of neocortex: contributions of the dorsal. *Neocortex* 200:75. doi: 10.1007/978-1-4899-0652-6_8

Siegel, M., Warden, M. R., and Miller, E. K. (2009). Phase-dependent neuronal coding of objects in short-term memory. *Proc. Natl. Acad. Sci. U.S.A.* 106, 21341–21346. doi: 10.1073/pnas.0908193106

Singh, R., and Eliasmith, C. (2006). Higher-dimensional neurons explain the tuning and dynamics of working memory cells. *J. Neurosci.* 26, 3667–3678. doi: 10.1523/JNEUROSCI.4864-05.2006

Sjöström, J., and Gerstner, W. (2010). Spike-timing dependent plasticity. *Scholarpedia* 5:1362. doi: 10.4249/scholarpedia.1362

Sjöström, P. J., and Häusser, M. (2006). A cooperative switch determines the sign of synaptic plasticity in distal dendrites of neocortical pyramidal neurons. *Neuron* 51, 227–238. doi: 10.1016/j.neuron.2006.06.017

Skerry, A. E., and Spelke, E. S. (2014). Preverbal infants identify emotional reactions that are incongruent with goal outcomes. *Cognition* 130, 204–216. doi: 10.1016/j.cognition.2013.11.002

Softky, W., and Koch, C. (1993). The highly irregular firing of cortical cells is inconsistent with temporal integration of random EPSPs. *J. Neurosci.* 13, 334–350.

Solari, S. V. H., and Stoner, R. (2011). Cognitive consilience: primate non-primary neuroanatomical circuits underlying cognition. *Front. Neuroanat.* 5:65. doi: 10.3389/fnana.2011.00065

Sountsov, P., and Miller, P. (2015). Spiking neuron network Helmholtz machine. *Front. Comput. Neurosci.* 9:46. doi: 10.3389/fncom.2015.00046

Squire, L. R. (2004). Memory systems of the brain: a brief history and current perspective. *Neurobiol. Learn. Memory* 82, 171–177. doi: 10.1016/j.nlm.2004.06.005

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15, 1929–1958. Available online at: http://www.jmlr.org/papers/v15/srivastava14a.html

Stachenfeld, K. (2014). "Design principles of the hippocampal cognitive map," in *Advances in Neural Information Processing Systems* (Montreal, QC).

Stanisor, L., van der Togt, C., Pennartz, C. M. A., and Roelfsema, P. R. (2013). A unified selection signal for attention and reward in primary visual cortex. *Proc. Natl. Acad. Sci. U.S.A.* 110, 9136–9141. doi: 10.1073/pnas.1300117110

Stewart, T., and Eliasmith, C. (2009). "Compositionality and biologically plausible models," in *Oxford Handbook of Compositionality*, eds W. Hinzen, E. Machery, and M. Werning (Oxford University Press). doi: 10.1093/oxfordhb/9780199541072.013.0029

Stocco, A., Lebiere, C., and Anderson, J. R. (2010). Conditional routing of information to the cortex: a model of the basal ganglia's role in cognitive coordination. *Psychol. Rev.* 117, 541–574. doi: 10.1037/a0019077

Stork, D. G. (1989). "Is backpropagation biologically plausible?," in *International Joint Conference on Neural Networks*, Vol. 2 (Washington, DC: IEEE), 241–246.

Strausfeld, N. J., and Hirth, F. (2013). Deep homology of arthropod central complex and vertebrate basal ganglia. *Science (New York, N.Y.)* 340, 157–161. doi: 10.1126/science.1231828

Sukhbaatar, S., Bruna, J., Paluri, M., Bourdev, L., and Fergus, R. (2014). Training convolutional networks with noisy labels. *arXiv preprint arXiv:1406.2080*.

Sun, Y., Gomez, F., and Schmidhuber, J. (2011). "Planning to be surprised: optimal bayesian exploration in dynamic environments," in *Artificial General Intelligence* (Mountain View, CA: Springer), 41–51. doi: 10.1007/978-3-642-22887-2_5

Sussillo, D. (2014). Neural circuits as computational dynamical systems. *Curr. Opin. Neurobiol.* 25, 156–163. doi: 10.1016/j.conb.2014.01.008

Sussillo, D., and Abbott, L. (2009). Generating coherent patterns of activity from chaotic neural networks. *Neuron* 63, 544–557. doi: 10.1016/j.neuron.2009.07.018

Sussillo, D., Churchland, M. M., Kaufman, M. T., and Shenoy, K. V. (2015). A neural network that finds a naturalistic solution for the production of muscle activity. *Nat. Neurosci.* 18, 1025–1033. doi: 10.1038/nn.4042

Sutskever, I., and Martens, J. (2013). "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30 th International Conference on Machine Learning* (Atlanta: JMLR:W&CP).

Sutskever, I., Martens, J., and Hinton, G. E. (2011). "Generating text with recurrent neural networks," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)* (Bellevue), 1017–1024.

Sutton, R. S., and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press.

Tacchetti, A., Isik, L., and Poggio, T. (2016). Spatio-temporal convolutional neural networks explain human neural representations of action recognition. *arXiv preprint arXiv:1606.04698*.

Takata, N., Mishima, T., Hisatsune, C., Nagai, T., Ebisui, E., Mikoshiba, K., and Hirase, H. (2011). Astrocyte calcium signaling transforms cholinergic modulation to cortical plasticity *in vivo*. *J. Neurosci.* 31, 18155–18165. doi: 10.1523/JNEUROSCI.5289-11.2011

Tamar, A., Levine, S., and Abbeel, P. (2016). Value iteration networks. *arXiv preprint arXiv:1602.02867*.

Tang, Y., Salakhutdinov, R., and Hinton, G. (2012). Deep mixtures of factor analysers. arXiv:1206.4635

Tang, Y., Salakhutdinov, R., and Hinton, G. (2013). "Tensor analyzers," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)* (Atlanta, GA).

Tapson, J., and van Schaik, A. (2013). Learning the pseudoinverse solution to network weights. *Neural Netw.* 45, 94–100. doi: 10.1016/j.neunet.2013.02.008

Tavares, R. M., Mendelsohn, A., Grossman, Y., Williams, C. H., Shapiro, M., Trope, Y., et al. (2015). A map for social navigation in the human brain. *Neuron* 87, 231–243. doi: 10.1016/j.neuron.2015.06.011

Taylor, S. V., and Faisal, A. A. (2011). Does the cost function of human motor control depend on the internal metabolic state? *BMC Neurosci.* 12(Suppl. 1):P99. doi: 10.1186/1471-2202-12-S1-P99

Terrence Stewart, C. E., Choo, X., and Eliasmith, C. (2010). "Symbolic reasoning in spiking neurons: a model of the cortex/basal ganglia/thalamus loop," in *32nd Annual Meeting of the Cognitive Science Society* (Portland, OR).

Tervo, D. G. R., Tenenbaum, J. B., and Gershman, S. J. (2016). Toward the neural implementation of structure learning. *Curr. Opin. Neurobiol.* 37, 99–105. doi: 10.1016/j.conb.2016.01.014

Tesauro, G. (1995). Temporal difference learning and TD-Gammon. *Commun. ACM*. 38, 58–68.

Thalmeier, D., Uhlmann, M., Kappen, H. J., and Memmesheimer, R.-M. (2015). Learning universal computations with spikes. arXiv:1505.07866.

Tinbergen, N. (1965). "Behavior and natural selection," in *Ideas in Modern Biology: proceedings of the 16th International Zoological Congress*, ed J. A. Moore (Washington, DC), 521–542.

Todorov, E. (2002). Cosine tuning minimizes motor errors. *Neural Comput.* 14, 1233–1260. doi: 10.1016/j.conb.2016.01.014

Todorov, E. (2009). Efficient computation of optimal actions. *Proc. Natl. Acad. Sci. U.S.A.* 106, 11478–11483. doi: 10.1073/pnas.0710743106

Todorov, E., and Jordan, M. I. (2002). Optimal feedback control as a theory of motor coordination. *Nat. Neurosci.* 5, 1226–1235. doi: 10.1038/nn963

Tripp, B., and Eliasmith, C. (2016). Function approximation in inhibitory networks. *Neural Netw.* 77, 95–106. doi: 10.1016/j.neunet.2016.01.010

Turner, R. S., and Desmurget, M. (2010). Basal ganglia contributions to motor control: a vigorous tutor. *Curr. Opin. Neurobiol.* 20, 704–716. doi: 10.1016/j.conb.2010.08.022

Turrigiano, G. (2012). Homeostatic synaptic plasticity: local and global mechanisms for stabilizing neuronal function. *Cold Spring Harb. Perspect. Biol.* 4:a005736. doi: 10.1101/cshperspect.a005736

Ullman, S., Harari, D., and Dorfman, N. (2012). From simple innate biases to complex visual concepts. *Proc. Natl. Acad. Sci. U.S.A.* 109, 18215–18220. doi: 10.1073/pnas.1207690109

Urbanczik, R., and Senn, W. (2014). Learning by the dendritic prediction of somatic spiking. *Neuron* 81, 521–528. doi: 10.1016/j.neuron.2013.11.030

Valpola, H. (2015). From neural PCA to deep unsupervised learning. arXiv:1411.7783.

van den Oord, A., Kalchbrenner, N., and Kavukcuoglu, K. (2016). Pixel recurrent neural networks. arXiv:1601.06759.

Van Heijningen, C. A., De Visser, J., Zuidema, W., and Ten Cate, C. (2009). Simple rules can explain discrimination of putative recursive syntactic structures by a songbird species. *Proc. Natl. Acad. Sci. U.S.A.* 106, 20538–20543. doi: 10.1073/pnas.0908113106

Van Kerkoerle, T., Self, M. W., Dagnino, B., Gariel-Mathis, M.-A., Poort, J., Van Der Togt, C., et al. (2014). Alpha and gamma oscillations characterize feedback and feedforward processing in monkey visual cortex. *Proc. Natl. Acad. Sci. U.S.A.* 111, 14332–14341. doi: 10.1073/pnas.1402773111

Veit, A., Wilber, M., and Belongie, S. (2016). Residual networks are exponential ensembles of relatively shallow networks. arXiv:1605.06431.

Verney, C., Baulac, M., Berger, B., Alvarez, C., Vigny, A., and Helle, K. (1985). Morphological evidence for a dopaminergic terminal field in the hippocampal formation of young and adult rat. *Neuroscience* 14, 1039–1052. doi: 10.1016/0306-4522(85)90275-1

Verwey, W. B. (1996). Buffer loading and chunking in sequential keypressing. *J. Exp. Psychol.* 22:544.

Wang, J. X., Cohen, N. J., and Voss, J. L. (2015). Covert rapid action-memory simulation (CRAMS): a hypothesis of hippocampal-prefrontal interactions for adaptive behavior. *Neurobiol. Learn. Memory* 117, 22–33. doi: 10.1016/j.nlm.2014.04.003

Wang, J., and Yuille, A. (2014). Semantic part segmentation using compositional model combining shape and appearance. arXiv:1412.6124.

Wang, X.-J. (2012). "The prefrontal cortex as a quintessential "cognitive-type" neural circuit," in *Principles of Frontal Lobe Function*, Edited by D. T. Stuss and R. T. Knight (Oxford University Press), 226–248.

Warden, M. R., and Miller, E. K. (2007). The representation of multiple objects in prefrontal neuronal delay activity. *Cereb. Cortex* (New York, N.Y.: 1991) 17(Suppl. 1):i41–i50. doi: 10.1093/cercor/bhm070

Warden, M. R., and Miller, E. K. (2010). Task-dependent changes in short-term memory in the prefrontal cortex. *J. Neurosci.* 30, 15801–15810. doi: 10.1523/JNEUROSCI.1569-10.2010

Watter, M., Springenberg, J., Boedecker, J., and Riedmiller, M. (2015). "Embed to control: a locally linear latent dynamics model for control from raw images," in *Advances in Neural Information Processing Systems* (Montreal, QC), 2728–2736.

Wayne, G., and Abbott, L. F. (2014). Hierarchical control using networks trained with higher-level forward models. *Neural Comput.* 26, 2163–2193. doi: 10.1162/NECO_a_00639

Werbos, P. (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Doctoral Dissertation, Harvard University, Harvard.

Werbos, P. (1982). Applications of advances in nonlinear sensitivity analysis. *Syst. Model. Optim.* 38, 762–770. doi: 10.1007/bfb0006203

Werbos, P. (1990). Backpropagation through time: what it does and how to do it. *Proc. IEEE.* 78, 1550–1560. doi: 10.1109/5.58337

Werbos, P. J., and Si, J. (eds.). (2004). *Handbook of Learning and Approximate Dynamic Programming*, Vol. 2. Playa del Carmen: John Wiley & Sons.

Werfel, J., Xie, X., and Seung, H. S. (2005). Learning curves for stochastic gradient descent in linear feedforward networks. *Neural Comput.* 17, 2699–2718. doi: 10.1162/089976605774320539

Weston, J., Chopra, S., and Bordes, A. (2014). Memory networks. arXiv:1410.3916.

Whitney, W. F., Chang, M., Kulkarni, T., and Tenenbaum, J. B. (2016). Understanding visual concepts with continuation learning. arXiv:1602.06822.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.* 8, 229–256. doi: 10.1007/BF00992696

Williams, R. J., and Baird, L. C. (1993). *Tight Performance Bounds on Greedy Policies based on Imperfect Value Functions*. Technical Report, Citeseer.

Williams, S. R., and Stuart, G. J. (2000). Backpropagation of physiological spike trains in neocortical pyramidal neurons: implications for temporal coding in dendrites. *J. Neurosci.* 20, 8238–8246.

Wilson, R. I., and Nicoll, R. A. (2001). Endogenous cannabinoids mediate retrograde signalling at hippocampal synapses. *Nature* 410, 588–592. doi: 10.1038/35069076

Winston, P. (2011). "The strong story hypothesis and the directed perception hypothesis," in *AAAI Fall Symposium Series* (Association for the Advancement of Artificial Intelligence).

Wiskott, L., and Sejnowski, T. J. (2002). Slow feature analysis: unsupervised learning of invariances. *Neural Comput.* 14, 715–770. doi: 10.1162/089976602317318938

Wolpert, D. M., and Flanagan, J. R. (2016). Computations underlying sensorimotor learning. *Curr. Opin. Neurobiol.* 37, 7–11. doi: 10.1016/j.conb.2015.12.003

Womelsdorf, T., Valiante, T. A., Sahin, N. T., Miller, K. J., and Tiesinga, P. (2014). Dynamic circuit motifs underlying rhythmic gain control, gating and integration. *Nat. Neurosci.* 17, 1031–1039. doi: 10.1038/nn.3764

Wyss, R., König, P., and Verschure, P. F. M. J. (2006). A model of the ventral visual system based on temporal stability and local memory. *PLoS Biol.* 4:e120. doi: 10.1371/journal.pbio.0040120

Xie, X., and Seung, H. (2000). "Spike-based learning rules and stabilization of persistent neural activity," in *Advances in Neural Information Processing System* (Denver).

Xie, X., and Seung, H. S. (2003). Equivalence of backpropagation and contrastive Hebbian learning in a layered network. *Neural Comput.* 15, 441–454. doi: 10.1162/089976603762552988

Xiong, C., Merity, S., and Socher, R. (2016). Dynamic memory networks for visual and textual question answering. *arXiv preprint arXiv:160 3.01417*.

Xu, M., Zhang, S.-Y., Dan, Y., and Poo, M.-M. (2014). Representation of interval timing by temporally scalable firing patterns in rat prefrontal cortex. *Proc. Natl. Acad. Sci. U.S.A.* 111, 480–485. doi: 10.1073/pnas.1321314111

Yamins, D. L., and DiCarlo, J. J. (2016a). Using goal-driven deep learning models to understand sensory cortex. *Nat. Neurosci.* 19, 356–365. doi: 10.1038/nn.4244

Yamins, D. L., and DiCarlo, J. J. (2016b). Eight open questions in the computational modeling of higher sensory cortex. *Curr. Opin. Neurobiol.* 37, 114–120. doi: 10.1016/j.conb.2016.02.001

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). "How transferable are features in deep neural networks?," in *Advances in Neural Information Processing Systems* (Montreal, QC), 3320–3328.

Yttri, E. A., and Dudman, J. T. (2016). Opponent and bidirectional control of movement velocity in the basal ganglia. *Nature* 533, 402–406. doi: 10.1038/nature17639

Yu, C., and Smith, L. B. (2013). Joint attention without gaze following: human infants and their parents coordinate visual attention to objects through eye-hand coordination. *PLoS ONE* 8:e79659. doi: 10.1371/journal.pone.0079659

Yuste, R., MacLean, J. N., Smith, J., and Lansner, A. (2005). The cortex as a central pattern generator. *Nat. Rev. Neurosci.* 6, 477–483. doi: 10.1038/nrn1686

Zaremba, W., and Sutskever, I. (2015). Reinforcement learning neural turing machines. *arXiv preprint arXiv:1505.00521*.

Zeisel, A., Manchado, A. B. M., Codeluppi, S., Lönnerberg, P., La Manno, G., Juréus, A., et al. (2015). Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science* 347, 1138–1142. doi: 10.1126/science.aaa1934

Zemel, R. S., and Dayan, P. (1997). "Combining probabilistic population codes," in *International Joint Conference on Artificial Intelligence* (Nagoya), 1114–1119.

Zilli, E. A., and Hasselmo, M. E. (2010). Coupled noisy spiking neurons as velocity-controlled oscillators in a model of grid cell spatial firing. *J. Neurosci.* 30, 13850–13860. doi: 10.1523/JNEUROSCI.0547-1 0.2010

Zipser, D., and Andersen, R. (1988). A back-propagation programmed network that simulates response properties of a subset of posterior parietal neurons. *Nature* 331, 679–684. doi: 10.1038/331679a0

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Computational Neuroscience

ORIGINAL RESEARCH
published: 04 May 2017
doi: 10.3389/fncom.2017.00024

Equilibrium Propagation: Bridging the Gap between Energy-Based Models and Backpropagation

Benjamin Scellier * *and Yoshua Bengio* †

Département d'Informatique et de Recherche Opérationnelle, Montreal Institute for Learning Algorithms, Université de Montréal, Montreal, QC, Canada

We introduce Equilibrium Propagation, a learning framework for energy-based models. It involves only one kind of neural computation, performed in both the first phase (when the prediction is made) and the second phase of training (after the target or prediction error is revealed). Although this algorithm computes the gradient of an objective function just like Backpropagation, it does not need a special computation or circuit for the second phase, where errors are implicitly propagated. Equilibrium Propagation shares similarities with Contrastive Hebbian Learning and Contrastive Divergence while solving the theoretical issues of both algorithms: our algorithm computes the gradient of a well-defined objective function. Because the objective function is defined in terms of local perturbations, the second phase of Equilibrium Propagation corresponds to only nudging the prediction (fixed point or stationary distribution) toward a configuration that reduces prediction error. In the case of a recurrent multi-layer supervised network, the output units are slightly nudged toward their target in the second phase, and the perturbation introduced at the output layer propagates backward in the hidden layers. We show that the signal "back-propagated" during this second phase corresponds to the propagation of error derivatives and encodes the gradient of the objective function, when the synaptic update corresponds to a standard form of spike-timing dependent plasticity. This work makes it more plausible that a mechanism similar to Backpropagation could be implemented by brains, since leaky integrator neural computation performs both inference and error back-propagation in our model. The only local difference between the two phases is whether synaptic changes are allowed or not. We also show experimentally that multi-layer recurrently connected networks with 1, 2, and 3 hidden layers can be trained by Equilibrium Propagation on the permutation-invariant MNIST task.

Keywords: artificial neural network, backpropagation algorithm, biologically plausible learning rule, contrastive hebbian learning, deep learning, fixed point, Hopfield networks, spike-timing dependent plasticity

OPEN ACCESS

Edited by:
Marcel van Gerven, Radboud University Nijmegen, Netherlands

Reviewed by:
Stefan Frank, Radboud University Nijmegen, Netherlands
Petia D. Koprinkova-Hristova, Institute of Information and Communication Technologies (BAS), Bulgaria

***Correspondence:**
Benjamin Scellier
benjamin.scellier@polytechnique.edu

† *Senior Fellow of CIFAR.*

Received: *01 December 2016*
Accepted: *28 March 2017*
Published: *04 May 2017*

Citation:
Scellier B and Bengio Y (2017) Equilibrium Propagation: Bridging the Gap between Energy-Based Models and Backpropagation. Front. Comput. Neurosci. 11:24. doi: 10.3389/fncom.2017.00024

1. INTRODUCTION

The Backpropagation algorithm to train neural networks is considered to be biologically implausible. Among other reasons, one major reason is that Backpropagation requires a special computational circuit and a special kind of computation in the second phase of training. Here, we introduce a new learning framework called Equilibrium Propagation, which requires only one computational circuit and one type of computation for both phases of training. Just like

Backpropagation applies to any differentiable computational graph (and not just a regular multi-layer neural network), Equilibrium Propagation applies to a whole class of energy based models (the prototype of which is the continuous Hopfield model).

In Section 2, we revisit the continuous Hopfield model (Hopfield, 1984) and introduce Equilibrium Propagation as a new framework to train it. The model is driven by an energy function whose minima correspond to preferred states of the model. At prediction time, inputs are clamped and the network relaxes to a fixed point, corresponding to a local minimum of the energy function. The prediction is then read out on the output units. This corresponds to the first phase of the algorithm. In the second phase of the training framework, when the target values for output units are observed, the outputs are nudged toward their targets and the network relaxes to a new but nearby fixed point which corresponds to slightly smaller prediction error. The learning rule, which is proved to perform gradient descent on the squared error, is a kind of contrastive Hebbian learning rule in which we *learn* (make more probable) the second-phase fixed point by reducing its energy and *unlearn* (make less probable) the first-phase fixed point by increasing its energy. However, our learning rule is not the usual contrastive Hebbian learning rule and it also differs from Boltzmann machine learning rules, as discussed in Sections 4.1 and 4.2.

During the second phase, the perturbation caused at the outputs propagates across hidden layers in the network. Because the propagation goes from outputs backward in the network, it is better thought of as a "back-propagation." It is shown by Bengio and Fischer (2015) and Bengio et al. (2017) that the early change of neural activities in the second phase corresponds to the propagation of error derivatives with respect to neural activities. Our contribution in this paper is to go beyond the early change of neural activities and to show that the second phase also implements the (back)-propagation of error derivatives with respect to the synaptic weights, and that this update corresponds to a form of spike-timing dependent plasticity, using the results of Bengio et al. (2017).

In Section 3, we present the general formulation of Equilibrium Propagation: a new machine learning framework for energy-based models. This framework applies to a whole class of energy based models, which is not limited to the continuous Hopfield model but encompasses arbitrary dynamics whose fixed points (or stationary distributions) correspond to minima of an energy function.

In Section 4, we compare our algorithm to the existing learning algorithms for energy-based models. The recurrent back-propagation algorithm introduced by Pineda (1987) and Almeida (1987) optimizes the same objective function as ours but it involves a different kind of neural computation in the second phase of training, which is not satisfying from a biological perspective. The contrastive Hebbian learning rule for continuous Hopfield nets described by Movellan (1990) suffers from theoretical problems: learning may deteriorate when the free phase and clamped phase land in different modes of the energy function. The Contrastive Divergence algorithm (Hinton, 2002) has theoretical issues too: the CD_1 update rule may cycle indefinitely (Sutskever and Tieleman, 2010). The equivalence of back-propagation and contrastive Hebbian learning was shown by Xie and Seung (2003) but at the cost of extra assumptions: their model requires infinitesimal feedback weights and exponentially growing learning rates for remote layers.

Equilibrium Propagation solves all these theoretical issues at once. Our algorithm computes the gradient of a sound objective function that corresponds to local perturbations. It can be realized with leaky integrator neural computation which performs both *inference* (in the first phase) and *back-propagation of error derivatives* (in the second phase). Furthermore, we do not need extra hypotheses such as those required by Xie and Seung (2003). Note that algorithms related to ours based on infinitesimal perturbations at the outputs were also proposed by O'Reilly (1996) and Hertz et al. (1997).

Finally, we show experimentally in Section 5 that our model is trainable. We train recurrent neural networks with 1, 2, and 3 hidden layers on MNIST and we achieve 0.00% training error. The generalization error lies between 2 and 3% depending on the architecture. The code for the model is available[1] for replicating and extending the experiments.

2. THE CONTINUOUS HOPFIELD MODEL REVISITED: EQUILIBRIUM PROPAGATION AS A MORE BIOLOGICALLY PLAUSIBLE BACKPROPAGATION

In this section, we revisit the continuous Hopfield model (Hopfield, 1984) and introduce Equilibrium Propagation, a novel learning algorithm to train it. Equilibrium Propagation is similar in spirit to Backpropagation in the sense that it involves the propagation of a signal from output units to input units backward in the network, during the second phase of training. Unlike Backpropagation, Equilibrium Propagation requires only one kind of neural computations for both phases of training, making it more biologically plausible than Backpropagation. However, several points still need to be elucidated from a biological perspective. Perhaps the most important of them is that the model described in this section requires symmetric weights, a question discussed at the end of this paper.

2.1. A Kind of Hopfield Energy

Previous work (Hinton and Sejnowski, 1986; Friston and Stephan, 2007; Berkes et al., 2011) has hypothesized that, given a state of sensory information, neurons are collectively performing inference: they are moving toward configurations that better "explain" the observed sensory data. We can think of the neurons' configuration as an "explanation" (or "interpretation") for the observed sensory data. In the energy-based model presented here, that means that the units of the network gradually move toward lower energy configurations that are more probable, given the sensory input and according to the current "model of the world" associated with the parameters of the model.

[1] https://github.com/bscellier/Towards-a-Biologically-Plausible-Backprop

We denote by u the set of units of the network[2], and by $\theta = (W, b)$ the set of free parameters to be learned, which includes the synaptic weights W_{ij} and the neuron biases b_i. The units are continuous-valued and would correspond to averaged voltage potential across time, spikes, and possibly neurons in the same minicolumn. Finally, ρ is a non-linear activation function such that $\rho(u_i)$ represents the firing rate of unit i.

We consider the following energy function E, a kind of Hopfield energy, first studied by Bengio and Fischer (2015), Bengio et al. (2015a,b), and Bengio et al. (2017):

$$E(u) := \frac{1}{2}\sum_i u_i^2 - \frac{1}{2}\sum_{i\neq j} W_{ij}\rho(u_i)\rho(u_j) - \sum_i b_i\rho(u_i). \quad (1)$$

Note that the network is recurrently connected with symmetric connections, that is $W_{ij} = W_{ji}$. The algorithm presented here is applicable to any architecture (so long as connections are symmetric), even a fully connected network. However, to make the connection to backpropagation more obvious, we will consider more specifically a layered architecture with no skip-layer connections and no lateral connections within a layer (**Figure 1**).

In the supervised setting studied here, the units of the network are split in three sets: the inputs x which are always clamped (just like in other models such as the conditional Boltzmann machine), the hidden units h (which may themselves be split in several layers) and the output units y. We use the notation d for the targets, which should not be confused with the outputs y. The set of all units in the network is $u = \{\mathrm{x}, h, y\}$. As usual in the supervised learning scenario, the output units y aim to replicate their targets d. The discrepancy between the output units y and the targets d is measured by the quadratic cost function:

$$C := \frac{1}{2}\|y - d\|^2. \quad (2)$$

The cost function C also acts as an external potential energy for the output units, which can drive them toward their target. A novelty in our work, with respect to previously studied energy-based models, is that we introduce the "total energy function" F, which takes the form:

$$F := E + \beta C, \quad (3)$$

where $\beta \geq 0$ is a real-valued scalar that controls whether the output y is pushed toward the target d or not, and by how much. We call β the "influence parameter" or "clamping factor." The total energy F is the sum of two potential energies: the internal potential E that models the interactions within the network, and the external potential βC that models how the targets influence the output units. Contrary to Boltzmann Machines where the visible units are either free or (fully) clamped, here the real-valued parameter β allows the output units to be *weakly clamped*.

[2]For reasons of convenience, we use the same symbol u to mean both the set of units and the value of those units.

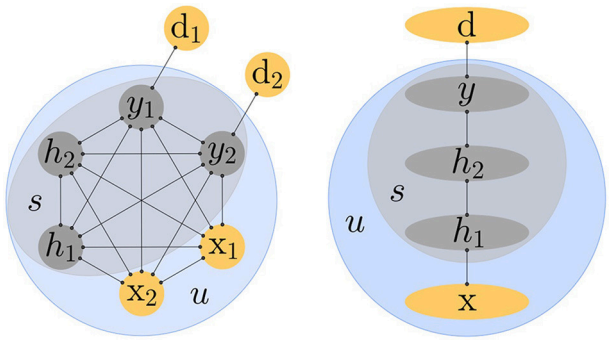


FIGURE 1 | The input units *x* are always clamped. The state variable s includes the hidden units h and output units y. The targets are denoted by d. The network is recurrently connected with symmetric connections. **Left.** Equilibrium Propagation applies to any architecture, even a fully connected network. **Right.** The connection with Backpropagation is more obvious when the network has a layered architecture.

2.2. The Neuronal Dynamics

We denote the state variable of the network by $s = \{h, y\}$ which does not include the input units x since they are always clamped. We assume that the time evolution of the state variable s is governed by the gradient dynamics:

$$\frac{ds}{dt} = -\frac{\partial F}{\partial s}. \quad (4)$$

Unlike more conventional artificial neural networks, the model studied here is a continuous-time dynamical system described by the differential equation of motion (Equation 4). The total energy of the system decreases as time progresses (θ, β, x, and d being fixed) since:

$$\frac{dF}{dt} = \frac{\partial F}{\partial s}\cdot\frac{ds}{dt} = -\left\|\frac{ds}{dt}\right\|^2 \leq 0. \quad (5)$$

The energy stops decreasing when the network has reached a fixed point:

$$\frac{dF}{dt} = 0 \quad \Leftrightarrow \quad \frac{ds}{dt} = 0 \quad \Leftrightarrow \quad \frac{\partial F}{\partial s} = 0. \quad (6)$$

The differential equation of motion (Equation 4) can be seen as a sum of two "forces" that act on the temporal derivative of s:

$$\frac{ds}{dt} = -\frac{\partial E}{\partial s} - \beta\frac{\partial C}{\partial s}. \quad (7)$$

The "internal force" induced by the internal potential (the Hopfield energy, Equation 1) on the i-th unit is:

$$-\frac{\partial E}{\partial s_i} = \rho'(s_i)\left(\sum_{j\neq i} W_{ij}\rho(u_j) + b_i\right) - s_i, \quad (8)$$

while the "external force" induced by the external potential (Equation 2) on h_i and y_i is, respectively:

$$-\beta\frac{\partial C}{\partial h_i}=0 \qquad \text{and} \qquad -\beta\frac{\partial C}{\partial y_i}=\beta(d_i-y_i). \tag{9}$$

The form of Equation (8) is reminiscent of a leaky integrator neuron model, in which neurons are seen as performing leaky temporal integration of their past inputs. Note that the hypothesis of symmetric connections ($W_{ij} = W_{ji}$) was used to derive Equation (8). As discussed in Bengio and Fischer (2015), the factor $\rho'(s_i)$ would suggest that when a neuron is saturated [firing at the maximal or minimal rate so that $\rho'(s_i) \approx 0$], its state is not sensitive to external inputs, while the leaky term drives it out of the saturation regime, toward its resting value $s_i = 0$.

The form of Equation (9) suggests that when $\beta = 0$, the output units are not sensitive to the outside world d. In this case, we say that the network is in the *free phase* (or first phase). On the contrary, when $\beta > 0$, the "external force" drives the output unit y_i toward the target d_i. In this case, we say that the network is in the *weakly clamped phase* (or second phase).

Finally, a more likely dynamics would include some form of noise. The notion of fixed point is then replaced by that of stationary distribution. In Appendix C, we present a stochastic framework that naturally extends the analysis presented here.

2.3. Free Phase, Weakly Clamped Phase, and Backpropagation of Errors

In the first phase of training, the inputs are clamped and $\beta = 0$ (the output units are free). We call this phase the *free phase* and the state toward which the network converges is the *free fixed point* u^0. The prediction is read out on the output units y at the fixed point.

In the second phase (which we call *weakly clamped phase*), the influence parameter β is changed to a small positive value $\beta > 0$ and the novel term βC added to the energy function (Equation 3) induces a new "external force" that acts on the output units (Equation 9). This force models the observation of d: it nudges the output units from their free fixed point value in the direction of their target. Since this force only acts on the output units, the hidden units are initially at equilibrium at the beginning of the weakly clamped phase, but the perturbation caused at the output units will propagate in the hidden units as time progresses. When the architecture is a multi-layered net (**Figure 1**, Right), the perturbation at the output layer propagates backwards across the hidden layers of the network. This propagation is thus better thought of as a "back-propagation." The net eventually settles to a new fixed point (corresponding to the new positive value of β) which we call *weakly clamped fixed point* and denote by u^β.

Remarkably, the perturbation that is (back-)propagated during the second phase corresponds to the propagation of error derivatives. It was first shown by Bengio and Fischer (2015) that, starting from the free fixed point, the early changes of neural activities during the weakly clamped phase (caused by the output units moving toward their target) approximate some kind of error derivatives with respect to the layers' activities. They considered a regular multi-layer neural network with no skip-layer connections and no lateral connections within a layer.

In this paper, we show that the weakly clamped phase also implements the (back)-propagation of error derivatives with respect to the synaptic weights. In the limit $\beta \to 0$, the update rule:

$$\Delta W_{ij} \propto \frac{1}{\beta}\left(\rho\left(u_i^\beta\right)\rho\left(u_j^\beta\right)-\rho\left(u_i^0\right)\rho\left(u_j^0\right)\right) \tag{10}$$

gives rise to stochastic gradient descent on $J := \frac{1}{2}\left\|y^0-d\right\|^2$, where y^0 is the state of the output units at the free fixed point. We will state and prove this theorem in a more general setting in Section 3. In particular, this result holds for any architecture and not just a layered architecture (**Figure 1**) like the one considered by Bengio and Fischer (2015).

The learning rule (Equation 10) is a kind of contrastive Hebbian learning rule, somewhat similar to the one studied by Movellan (1990) and the Boltzmann machine learning rule. The differences with these algorithms will be discussed in Section 4.

We call our learning algorithm Equilibrium Propagation. In this algorithm, leaky integrator neural computation (as described in Section 2.2), performs both *inference* (in the free phase), and *error back-propagation* (in the weakly clamped phase).

2.4. Connection to Spike-Timing Dependent Plasticity

Spike-Timing Dependent Plasticity (STDP) is believed to be a prominent form of synaptic change in neurons (Markram and Sakmann, 1995; Gerstner et al., 1996), and see Markram et al. (2012) for a review.

The STDP observations relate the expected change in synaptic weights to the timing difference between post-synaptic and pre-synaptic spikes. This is the result of experimental observations in biological neurons, but its role as part of a learning algorithm remains a topic where more exploration is needed. Here, is an attempt in this direction.

Experimental results by Bengio et al. (2015b) show that if the temporal derivative of the synaptic weight W_{ij} satisfies:

$$\frac{dW_{ij}}{dt} \propto \rho(u_i)\frac{du_j}{dt}, \tag{11}$$

then one recovers the experimental observations by Bi and Poo (2001) in biological neurons. Xie and Seung (2000) have studied the learning rule:

$$\frac{dW_{ij}}{dt} \propto \rho(u_i)\frac{d\rho(u_j)}{dt}. \tag{12}$$

Note, that the two rules (Equations 11, 12) are the same up to a factor $\rho'(u_j)$. An advantage of Equation (12) is that it leads to a more natural view of the update rule in the case of tied weights studied here ($W_{ij} = W_{ji}$). Indeed, the update should take into account the pressures from both the i to j and j to i synapses, so that the total update under constraint is:

$$\frac{dW_{ij}}{dt} \propto \rho(u_i)\frac{d\rho(u_j)}{dt}+\rho(u_j)\frac{d\rho(u_i)}{dt}=\frac{d}{dt}\rho(u_i)\rho(u_j). \tag{13}$$

By integrating Equation (13) on the path from the free fixed point u^0 to the weakly clamped fixed point u^β during the second phase, we get:

$$\Delta W_{ij} \propto \int \frac{dW_{ij}}{dt} dt = \int \frac{d}{dt} \rho(u_i)\rho(u_j) dt = \int d\left(\rho(u_i)\rho(u_j)\right) = \rho\left(u_i^\beta\right)\rho\left(u_j^\beta\right) - \rho\left(u_i^0\right)\rho\left(u_j^0\right), \quad (14)$$

which is the same learning rule as Equation (10) up to a factor $1/\beta$. Therefore, the update rule (Equation 10) can be interpreted as a continuous-time integration of Equation (12), in the case of symmetric weights, on the path from u^0 to u^β during the second phase.

We propose two possible interpretations for the synaptic plasticity in our model.

First view. In the first phase, a anti-Hebbian update occurs at the free fixed point $\Delta W_{ij} \propto -\rho\left(u_i^0\right)\rho\left(u_j^0\right)$. In the second phase, a Hebbian update occurs at the weakly-clamped fixed point $\Delta W_{ij} \propto +\rho\left(u_i^\beta\right)\rho\left(u_j^\beta\right)$.

Second view. In the first phase, no synaptic update occurs: $\Delta W_{ij} = 0$. In the second phase, when the neurons' state move from the free fixed point u^0 to the weakly-clamped fixed point u^β, the synaptic weights follow the "tied version" of the continuous-time update rule $\frac{dW_{ij}}{dt} \propto \rho(u_i)\frac{d\rho(u_j)}{dt} + \rho(u_j)\frac{d\rho(u_i)}{dt}$.

3. A MACHINE LEARNING FRAMEWORK FOR ENERGY BASED MODELS

In this section we generalize the setting presented in Section 2. We lay down the basis for a new machine learning framework for energy-based models, in which Equilibrium Propagation plays a role analog to Backpropagation in computational graphs to compute the gradient of an objective function. Just like the Multi Layer Perceptron is the prototype of computational graphs in which Backpropagation is applicable, the continuous Hopfield model presented in Section 2 appears to be the prototype of models which can be trained with Equilibrium Propagation.

In our new machine learning framework, the central object is the total energy function F: all quantities of interest (fixed points, cost function, objective function, gradient formula) can be defined or formulated directly in terms of F.

Besides, in our framework, the "prediction" (or fixed point) is defined *implicitly* in terms of the data point and the parameters of the model, rather than *explicitly* (like in a computational graph). This implicit definition makes applications on digital hardware (such as GPUs) less practical as it needs long inference phases involving numerical optimization of the energy function. But we expect that this framework could be very efficient if implemented by analog circuits, as suggested by Hertz et al. (1997).

The framework presented in this section is deterministic, but a natural extension to the stochastic case is presented in Appendix C.

3.1. Training Objective

In this section, we present the general framework while making sure to be consistent with the notations and terminology introduced in Section 2. We denote by s the state variable of the network, v the state of the external world (i.e., the data point being processed), and θ the set of free parameters to be learned. The variables s, v, and θ are real-valued vectors. The state variable s spontaneously moves toward low-energy configurations of an energy function $E(\theta, \mathrm{v}, s)$. Besides that, a cost function $C(\theta, \mathrm{v}, s)$ measures how "good" is a state is. The goal is to make low-energy configurations have low cost value.

For fixed θ and v, we denote by $s^0_{\theta,\mathrm{v}}$ a local minimum of E, also called fixed point, which corresponds to the "prediction" from the model:

$$s^0_{\theta,\mathrm{v}} \in \arg\min_s E(\theta, \mathrm{v}, s). \quad (15)$$

Here, we use the notation arg min to refer to the set of local minima. The objective function that we want to optimize is:

$$J(\theta, \mathrm{v}) := C\left(\theta, \mathrm{v}, s^0_{\theta,\mathrm{v}}\right). \quad (16)$$

Note the distinction between the cost function C and the objective function J: the cost function is defined for any state s, whereas the objective function is the cost associated to the fixed point $s^0_{\theta,\mathrm{v}}$.

Now that the objective function has been introduced, we define the training objective (for a single data point v) as:

$$\text{find} \quad \arg\min_\theta J(\theta, \mathrm{v}). \quad (17)$$

A formula to compute the gradient of J will be given in Section 3.3 (Theorem 1). Equivalently, the training objective can be reformulated as the following constrained optimization problem:

$$\text{find} \quad \arg\min_{\theta,s} C(\theta, \mathrm{v}, s) \quad (18)$$

$$\text{subject to} \quad \frac{\partial E}{\partial s}(\theta, \mathrm{v}, s) = 0, \quad (19)$$

where the constraint $\frac{\partial E}{\partial s}(\theta, \mathrm{v}, s) = 0$ is the fixed point condition. For completeness, a solution to this constrained optimization problem is given in Appendix B as well. Of course, both formulations of the training objective lead to the same gradient update on θ.

Note that, since the cost $C(\theta, \mathrm{v}, s)$ may depend on θ, it can include a regularization term of the form $\lambda\Omega(\theta)$, where $\Omega(\theta)$ is a L_1 or L_2 norm penalty for example.

In Section 2 we had $s = \{h, y\}$ for the state variable, $\mathrm{v} = \{\mathrm{x}, d\}$ for the state of the outside world, $\theta = (W, b)$ for the set of learned parameters, and the energy function E and cost function C were of the form $E(\theta, \mathrm{v}, s) = E(\theta, \mathrm{x}, h, y)$ and $C(\theta, \mathrm{v}, s) = C(y, d)$.

3.2. Total Energy Function

Following Section 2, we introduce the total energy function:

$$F(\theta, \mathrm{v}, \beta, s) := E(\theta, \mathrm{v}, s) + \beta\, C(\theta, \mathrm{v}, s), \quad (20)$$

where β is a real-valued scalar called "influence parameter." Then we extend the notion of fixed point for any value of β. The fixed point (or energy minimum), denoted by $s^{\beta}_{\theta,\mathrm{v}}$, is characterized by:

$$\frac{\partial F}{\partial s}\left(\theta, \mathrm{v}, \beta, s^{\beta}_{\theta,\mathrm{v}}\right) = 0 \tag{21}$$

and $\frac{\partial^2 F}{\partial s^2}\left(\theta, \mathrm{v}, \beta, s^{\beta}_{\theta,\mathrm{v}}\right)$ is a symmetric positive definite matrix. Under mild regularity conditions on F, the implicit function theorem ensures that, for fixed v, the funtion $(\theta, \beta) \mapsto s^{\beta}_{\theta,\mathrm{v}}$ is differentiable.

3.3. The Learning Algorithm: Equilibrium Propagation

Theorem 1 (Deterministic version)**.** *The gradient of the objective function with respect to θ is given by the formula:*

$$\frac{\partial J}{\partial \theta}(\theta, \mathrm{v}) = \lim_{\beta \to 0} \frac{1}{\beta}\left(\frac{\partial F}{\partial \theta}\left(\theta, \mathrm{v}, \beta, s^{\beta}_{\theta,\mathrm{v}}\right) - \frac{\partial F}{\partial \theta}\left(\theta, \mathrm{v}, 0, s^{0}_{\theta,\mathrm{v}}\right)\right), \tag{22}$$

or equivalently

$$\frac{\partial J}{\partial \theta}(\theta, \mathrm{v}) = \frac{\partial C}{\partial \theta}\left(\theta, \mathrm{v}, s^{0}_{\theta,\mathrm{v}}\right) + \lim_{\beta \to 0} \frac{1}{\beta}\left(\frac{\partial E}{\partial \theta}\left(\theta, \mathrm{v}, s^{\beta}_{\theta,\mathrm{v}}\right) - \frac{\partial E}{\partial \theta}\left(\theta, \mathrm{v}, s^{0}_{\theta,\mathrm{v}}\right)\right). \tag{23}$$

Theorem 1 will be proved in Appendix A. Note that the parameter β in Theorem 1 need not be positive (We only need $\beta \to 0$). Using the terminology introduced in Section 2, we call $s^{0}_{\theta,\mathrm{v}}$ the free fixed point, and $s^{\beta}_{\theta,\mathrm{v}}$ the nudged fixed point (or weakly-clamped fixed point in the case $\beta > 0$). Moreover, we call a free phase (resp. nudged phase or weakly-clamped phase) a procedure that yields a free fixed point (resp. nudged fixed point or weakly-clamped fixed point) by minimizing the energy function F with respect to s, for $\beta = 0$ (resp. $\beta \neq 0$). Theorem 1 suggests the following two-phase training procedure. Given a data point v:

1. Run a free phase until the system settles to a free fixed point $s^{0}_{\theta,\mathrm{v}}$ and collect $\frac{\partial F}{\partial \theta}\left(\theta, \mathrm{v}, 0, s^{0}_{\theta,\mathrm{v}}\right)$.
2. Run a nudged phase for some $\beta \neq 0$ such that $|\beta|$ is "small," until the system settles to a nudged fixed point $s^{\beta}_{\theta,\mathrm{v}}$ and collect $\frac{\partial F}{\partial \theta}\left(\theta, \mathrm{v}, \beta, s^{\beta}_{\theta,\mathrm{v}}\right)$.
3. Update the parameter θ according to

$$\Delta\theta \propto -\frac{1}{\beta}\left(\frac{\partial F}{\partial \theta}\left(\theta, \mathrm{v}, \beta, s^{\beta}_{\theta,\mathrm{v}}\right) - \frac{\partial F}{\partial \theta}\left(\theta, \mathrm{v}, 0, s^{0}_{\theta,\mathrm{v}}\right)\right). \tag{24}$$

Consider the case $\beta > 0$. Starting from the free fixed point $s^{0}_{\theta,\mathrm{v}}$ (which corresponds to the "prediction"), a small change of the parameter β (from the value $\beta = 0$ to a value $\beta > 0$) causes slight modifications in the interactions in the network. This small perturbation makes the network settle to a nearby weakly-clamped fixed point $s^{\beta}_{\theta,\mathrm{v}}$. Simultaneously, a kind of contrastive update rule for θ is happening, in which the energy of the free fixed point is increased and the energy of the weakly-clamped fixed point is decreased. We call this learning algorithm Equilibrium Propagation.

Note that in the setting introduced in Section 2.1 the total energy function (Equation 3) is such that $\frac{\partial F}{\partial W_{ij}} = -\rho(u_i)\rho(u_j)$, in agreement with Equation (10). In the weakly clamped phase, the novel term $\frac{1}{2}\beta \left\| y - d \right\|^2$ added to the energy E (with $\beta > 0$) slightly attracts the output state y to the target d. Clearly, the weakly clamped fixed point is better than the free fixed point in terms of prediction error. The following proposition generalizes this property to the general setting.

Proposition 2 (Deterministic version)**.** *The derivative of the function*

$$\beta \mapsto C\left(\theta, \mathrm{v}, s^{\beta}_{\theta,\mathrm{v}}\right) \tag{25}$$

at $\beta = 0$ is non-positive.

Proposition 2 will also be proved in Appendix A. This proposition shows that, unless the free fixed point $s^{0}_{\theta,\mathrm{v}}$ is already optimal in terms of cost value, for $\beta > 0$ small enough, the weakly-clamped fixed point $s^{\beta}_{\theta,\mathrm{v}}$ achieves lower cost value than the free fixed point. Thus, a small perturbation due to a small increment of β would nudge the system toward a state that reduces the cost value. This property sheds light on the update rule (Theorem 1), which can be seen as a kind of contrastive learning rule (somehow similar to the Boltzmann machine learning rule) where we *learn* (make more probable) the slightly better state $s^{\beta}_{\theta,\mathrm{v}}$ by reducing its energy and *unlearn* (make less probable) the slightly worse state $s^{0}_{\theta,\mathrm{v}}$ by increasing its energy.

However, our learning rule is different from the Boltzmann machine learning rule and the contrastive Hebbian learning rule. The differences between these algorithms will be discussed in section 4.

3.4. Another View of the Framework

In Sections 3.1 and 3.2 (as well as in Section 2) we first defined the energy function E and the cost function C, and then we introduced the total energy $F := E + \beta C$. Here, we propose an alternative view of the framework, where we reverse the order in which things are defined.

Given a total energy function F (which models all interactions within the network as well as the action of the external world on the network), we can define all quantities of interest in terms of F. Indeed, we can define the energy function E and the cost function C as:

$$E(\theta, \mathrm{v}, s) := F(\theta, \mathrm{v}, 0, s) \qquad \text{and} \qquad C(\theta, \mathrm{v}, s) := \frac{\partial F}{\partial \beta}(\theta, \mathrm{v}, 0, s), \tag{26}$$

where F and $\frac{\partial F}{\partial \beta}$ are evaluated with the argument β set to 0. Obviously the fixed points $s^{0}_{\theta,\mathrm{v}}$ and $s^{\beta}_{\theta,\mathrm{v}}$ are directly defined in terms of F, and so is the objective function $J(\theta, \mathrm{v}) := C\left(\theta, \mathrm{v}, s^{0}_{\theta,\mathrm{v}}\right)$. The learning algorithm (Theorem 1) is also formulated in terms

of F[3]. From this perspective, F contains all the information about the model and can be seen as the central object of the framework. For instance, the cost C represents the marginal variation of the total energy F due to a change of β.

As a comparison, in the traditional framework for Deep Learning, a model is represented by a (differentiable) computational graph in which each node is defined as a function of its parents. The set of functions that define the nodes fully specifies the model. The last node of the computational graph represents the cost to be optimized, while the other nodes represent the state of the layers of the network, as well as other intermediate computations.

In the framework for machine learning proposed here (the framework suited for Equilibrium Propagation), the analog of the set of functions that define the nodes in the computational graph is the total energy function F.

3.5. Backpropagation vs. Equilibrium Propagation

In the traditional framework for Deep Learning (**Figure 2**, left), each node in the computational graph is an *explicit* differentiable function of its parents. The state of the network $\widehat{s} = f_\theta(\mathrm{v})$ and the objective function $J(\theta, \mathrm{v}) = C\left(\theta, \mathrm{v}, f_\theta(\mathrm{v})\right)$ are computed *analytically*, as functions of θ and v, in the forward pass. The Backpropagation algorithm (a.k.a automatic differentiation) enables to compute the error derivatives analytically too, in the backward pass. Therefore, the state of the network $\widehat{s} = f_\theta(\mathrm{v})$ (forward pass) and the gradient of the objective function $\frac{\partial J}{\partial \theta}(\theta, \mathrm{v})$ (backward pass) can be computed *efficiently* and *exactly*[4].

In the framework for machine learning that we propose here (**Figure 2**, right), the free fixed point $\widehat{s} = s^0_{\theta,\mathrm{v}}$ is an *implicit* function of θ and v, characterized by $\frac{\partial E}{\partial s}(\theta, \mathrm{v}, s^0_{\theta,\mathrm{v}}) = 0$. The free fixed point is computed *numerically*, in the free phase (first phase). Similarly the nudged fixed point $s^\beta_{\theta,\mathrm{v}}$ is an implicit function of θ, v, and β, and is computed numerically in the nudged phase (second phase). Equilibrium Propagation *estimates* (for the particular value of β chosen in the second phase) the gradient of the objective function $\frac{\partial J}{\partial \theta}(\theta, \mathrm{v})$ based on these two fixed points. The requirement for numerical optimization in the first and second phases make computations *inefficient* and *approximate*. The experiments in Section 5 will show that the free phase is fairly long when performed with a discrete-time computer simulation. However, we expect that the full potential of the proposed framework could be exploited on analog hardware (instead of digital hardware), as suggested by Hertz et al. (1997).

4. RELATED WORK

In Section 2.3, we have discussed the relationship between Equilibrium Propagation and Backpropagation. In the weakly clamped phase, the change of the influence parameter β creates a perturbation at the output layer which propagates backwards in the hidden layers. The error derivatives and the gradient of the objective function are encoded by this perturbation.

In this section, we discuss the connection between our work and other algorihms, starting with Contrastive Hebbian Learning. Equilibrium Propagation offers a new perspective on the relationship between Backpropagation in feedforward nets and Contrastive Hebbian Learning in Hopfield nets and Boltzmann machines (**Table 1**).

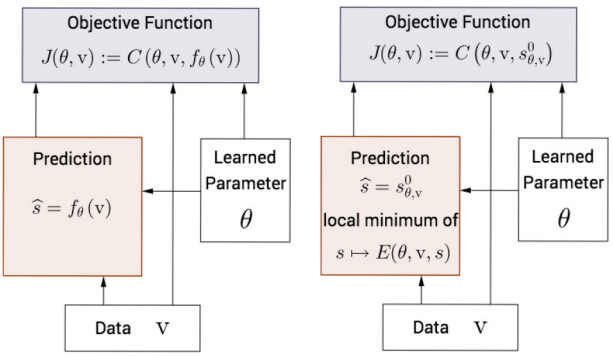


FIGURE 2 | Comparison between the traditional framework for Deep Learning and our framework. Left. In the traditional framework, the state of the network $f_\theta(\mathrm{v})$ and the objective function $J(\theta, \mathrm{v})$ are *explicit* functions of θ and v and are computed *analytically*. The gradient of the objective function is also computed analytically thanks to the Backpropagation algorithm (a.k.a automatic differentiation). **Right.** In our framework, the free fixed point $s^0_{\theta,\mathrm{v}}$ is an *implicit* function of θ and v and is computed *numerically*. The nudged fixed point $s^\beta_{\theta,\mathrm{v}}$ and the gradient of the objective function are also computed numerically, following our learning algorithm: Equilibrium Propagation.

4.1. Link to Contrastive Hebbian Learning

Despite the similarity between our learning rule and the Contrastive Hebbian Learning rule (CHL) for the continuous Hopfield model, there are important differences.

First, recall that our learning rule is:

$$\Delta W_{ij} \propto \lim_{\beta \to 0} \frac{1}{\beta} \left(\rho\left(u_i^\beta\right) \rho\left(u_j^\beta\right) - \rho\left(u_i^0\right) \rho\left(u_j^0\right) \right), \qquad (27)$$

where u^0 is the free fixed point and u^β is the *weakly* clamped fixed point. The Contrastive Hebbian Learning rule is:

$$\Delta W_{ij} \propto \rho\left(u_i^\infty\right) \rho\left(u_j^\infty\right) - \rho\left(u_i^0\right) \rho\left(u_j^0\right), \qquad (28)$$

where u^∞ is the *fully* clamped fixed point (i.e., fixed point with fully clamped outputs). We choose the notation u^∞ for the fully clamped fixed point because it corresponds to $\beta \to +\infty$ with the notations of our model. Indeed Equation (9) shows that in the limit $\beta \to +\infty$, the output unit y_i moves infinitely fast toward y_i, so y_i is immediately clamped to y_i and is no longer sensitive to the "internal force" (Equation 8). Another way to see it is by

[3]The proof presented in Appendix A will show that E, C, and F need not satisfy Equation (20) but only Equation (26).

[4]Here, we are not considering numerical stability issues due to the encoding of real numbers with finite precision.

TABLE 1 | Correspondence of the phases for different learning algorithms: Back-propagation, Equilibrium Propagation (our algorithm), Contrastive Hebbian Learning (and Boltzmann Machine Learning) and Almeida-Pineida's Recurrent Back-Propagation.

	Backprop	Equilibrium Prop	Contrastive Hebbian Learning	Almeida-Pineida
First Phase	Forward Pass	Free Phase	Free Phase (or Negative Phase)	Free Phase
Second Phase	Backward Pass	Weakly Clamped Phase	Clamped Phase (or Positive Phase)	Recurrent Backprop

considering Equation (3): as $\beta \to +\infty$, the only value of y that gives finite energy is d.

The objective functions that these two algorithms optimize also differ. Recalling the form of the Hopfield energy (Equation 1) and the cost function (Equation 2), Equilibrium Propagation computes the gradient of:

$$J = \frac{1}{2} \left\| y^0 - d \right\|^2, \tag{29}$$

where y^0 is the output state at the free phase fixed point u^0, while CHL computes the gradient of:

$$J_{\text{CHL}} = E\left(u^\infty\right) - E\left(u^0\right). \tag{30}$$

The objective function for CHL has theoretical problems: it may take negative values if the clamped phase and free phase stabilize in different modes of the energy function, in which case the weight update is inconsistent and learning usually deteriorates, as pointed out by Movellan (1990). Our objective function does not suffer from this problem, because it is defined in terms of local perturbations, and the implicit function theorem guarantees that the weakly clamped fixed point will be close to the free fixed point (thus in the same mode of the energy function).

We can also reformulate the learning rules and objective functions of these algorithms using the notations of the general setting (Section 3). For Equilibrium Propagation we have:

$$\Delta\theta \propto -\lim_{\beta \to 0} \frac{1}{\beta} \left(\frac{\partial F}{\partial \theta} \left(\theta, \mathrm{v}, \beta, s^\beta_{\theta,\mathrm{v}} \right) - \frac{\partial F}{\partial \theta} \left(\theta, \mathrm{v}, 0, s^0_{\theta,\mathrm{v}} \right) \right)$$

and

$$J(\theta, \mathrm{v}) = \frac{\partial F}{\partial \beta} \left(\theta, \mathrm{v}, 0, s^0_{\theta,\mathrm{v}} \right). \tag{31}$$

As for Contrastive Hebbian Learning, one has

$$\Delta\theta \propto -\left(\frac{\partial F}{\partial \theta} \left(\theta, \mathrm{v}, \infty, s^\infty_{\theta,\mathrm{v}} \right) - \frac{\partial F}{\partial \theta} \left(\theta, \mathrm{v}, 0, s^0_{\theta,\mathrm{v}} \right) \right)$$

and

$$J_{\text{CHL}}(\theta, \mathrm{v}) = F(\theta, \mathrm{v}, \infty, s^\infty_{\theta,\mathrm{v}}) - F(\theta, \mathrm{v}, 0, s^0_{\theta,\mathrm{v}}), \tag{32}$$

where $\beta = 0$ and $\beta = \infty$ are the values of β corresponding to free and (fully) clamped outputs, respectively.

Our learning algorithm is also more flexible because we are free to choose the cost function C (as well as the energy funtion E), whereas the contrastive function that CHL optimizes is fully determined by the energy function E.

4.2. Link to Boltzmann Machine Learning

Again, the log-likelihood that the Boltzmann machine optimizes is determined by the Hopfield energy E, whereas we have the freedom to choose the cost function in the framework for Equilibrium Propagation.

As discussed in Section 2.3, the second phase of Equilibrium Propagation (going from the free fixed point to the weakly clamped fixed point) can be seen as a brief "backpropagation phase" with weakly clamped target outputs. By contrast, in the positive phase of the Boltzmann machine, the target is fully clamped, so the (correct version of the) Boltzmann machine learning rule requires two separate and independent phases (Markov chains), making an analogy with backprop less obvious.

Our algorithm is also similar in spirit to the CD algorithm (Contrastive Divergence) for Boltzmann machines. In our model, we start from a free fixed point (which requires a long relaxation in the free phase) and then we run a short weakly clamped phase. In the CD algorithm, one starts from a positive equilibrium sample with the visible units clamped (which requires a long positive phase Markov chain in the case of a general Boltzmann machine) and then one runs a short negative phase. But there is an important difference: our algorithm computes the *correct* gradient of our objective function (in the limit $\beta \to 0$), whereas the CD algorithm computes a *biased estimator* of the gradient of the log-likelihood. The CD_1 update rule is provably not the gradient of any objective function and may cycle indefinitely in some pathological cases (Sutskever and Tieleman, 2010).

Finally, in the supervised setting presented in Section 2, a more subtle difference with the Boltzmann machine is that the "output" state y in our model is best thought of as being part of the latent state variable s. If we were to make an analogy with the Boltzmann machine, the visible units of the Boltzmann machine would be $\mathrm{v} = \{\mathrm{x}, d\}$, while the hidden units would be $s = \{h, y\}$. In the Boltzmann machine, the state of the external world is inferred directly on the visible units (because it is a probabilistic generative model that maximizes the log-likelyhood of the data), whereas in our model we make the choice to integrate in s special latent variables y that aim to match the target d.

4.3. Link to Recurrent Back-Propagation

Directly connected to our model is the work by Pineda (1987) and Almeida (1987) on recurrent back-propagation. They consider the same objective function as ours, but formulate the problem as a constrained optimization problem. In Appendix B, we derive another proof for the learning rule (Theorem 1) with the Lagrangian formalism for constrained optimization problems. The beginning of this proof is in essence the same as the one proposed by Pineda (1987); Almeida (1987), but there is a major

difference when it comes to solving Equation (75) for the costate variable λ^*. The method proposed by Pineda (1987) and Almeida (1987) is to use Equation (75) to compute λ^* by a fixed point iteration in a linearized form of the recurrent network. The computation of λ^* corresponds to their second phase, which they call *recurrent back-propagation*. However, this second phase does not follow the same kind of dynamics as the first phase (the free phase) because it uses a linearization of the neural activation rather than the fully non-linear activation[5]. From a biological plausibility point of view, having to use a different kind of hardware and computation for the two phases is not satisfying.

By contrast, like the continuous Hopfield net and the Boltzmann machine, our model involves only one kind of neural computations for both phases.

4.4. The Model by Xie and Seung

Previous work on the back-propagation interpretation of contrastive Hebbian learning was done by Xie and Seung (2003).

The model by Xie and Seung (2003) is a modified version of the Hopfield model. They consider the case of a layered MLP-like network, but their model can be extended to a more general connectivity, as shown here. In essence, using the notations of our model (Section 2), the energy function that they consider is:

$$E_{X\&S}(u) := \frac{1}{2}\sum_i \gamma^i u_i^2 - \sum_{i<j} \gamma^j W_{ij}\rho(u_i)\rho(u_j) - \sum_i \gamma^i b_i \rho(u_i). \tag{33}$$

The difference with Equation (1) is that they introduce a parameter γ, assumed to be small, that scales the strength of the connections. Their update rule is the contrastive Hebbian learning rule which, for this particular energy function, takes the form:

$$\begin{aligned}\Delta W_{ij} \propto\ & -\left(\frac{\partial E_{X\&S}}{\partial W_{ij}}\left(u^\infty\right) - \frac{\partial E_{X\&S}}{\partial W_{ij}}\left(u^0\right)\right) \\ &= \gamma^j\left(\rho\left(u_i^\infty\right)\rho\left(u_j^\infty\right) - \rho\left(u_i^0\right)\rho\left(u_j^0\right)\right)\end{aligned} \tag{34}$$

for every pair of indices (i,j) such that $i < j$. Here, u^∞ and u^0 are the (fully) clamped fixed point and free fixed point, respectively. Xie and Seung (2003) show that in the regime $\gamma \to 0$ this contrastive Hebbian learning rule is equivalent to back-propagation. At the free fixed point u^0, one has $\frac{\partial E_{X\&S}}{\partial s_i}(u^0) = 0$ for every unit s_i[6], which yields, after dividing by γ^i and rearranging the terms:

$$s_i^0 = \rho'\left(s_i^0\right)\left(\sum_{j<i} W_{ij}\rho\left(u_j^0\right) + \sum_{j>i}\gamma^{j-i}W_{ij}\rho\left(u_j^0\right) + b_i\right). \tag{35}$$

In the limit $\gamma \to 0$, one gets $s_i^0 \approx \rho'(s_i^0)\left(\sum_{j<i} W_{ij}\rho(u_j^0) + b_i\right)$, so that the network almost behaves like a feedforward net in this regime.

[5]Reccurent Back-propagation corresponds to Back-propagation Through Time (BPTT) when the network converges and remains at the fixed point for a large number of time steps.

[6]Recall that in our notations, the state variable s does not include the clamped inputs x, whereas u includes x.

As a comparison, recall that in our model (Section 2) the energy function is:

$$E(u) := \frac{1}{2}\sum_i u_i^2 - \sum_{i<j} W_{ij}\rho(u_i)\rho(u_j) - \sum_i b_i\rho(u_i), \tag{36}$$

the learning rule is:

$$\begin{aligned}\Delta W_{ij} \propto\ & -\lim_{\beta\to 0}\frac{1}{\beta}\left(\frac{\partial E}{\partial W_{ij}}\left(u^\beta\right) - \frac{\partial E}{\partial W_{ij}}\left(u^0\right)\right) \\ &= \lim_{\beta\to 0}\frac{1}{\beta}\left(\rho\left(u_i^\beta\right)\rho\left(u_j^\beta\right) - \rho\left(u_i^0\right)\rho\left(u_j^0\right)\right),\end{aligned} \tag{37}$$

and at the free fixed point, we have $\frac{\partial E}{\partial s_i}(u^0) = 0$ for every unit s_i, which gives:

$$s_i^0 = \rho'\left(s_i^0\right)\left(\sum_{j\neq i} W_{ij}\rho\left(u_j^0\right) + b_i\right). \tag{38}$$

Here, are the main differences between our model and theirs. In our model, the feedforward and feedback connections are both strong. In their model, the feedback weights are tiny compared to the feedforward weights, which makes the (recurrent) computations look almost feedforward. In our second phase, the outputs are weakly clamped. In their second phase, they are fully clamped. The theory of our model requires a unique learning rate for the weights, while in their model the update rule for W_{ij} (with $i < j$) is scaled by a factor γ^j (see Equation 34). Since γ is small, the learning rates for the weights vary on many orders of magnitude in their model. Intuitively, these multiple learning rates are required to compensate for the small feedback weights.

5. IMPLEMENTATION OF THE MODEL AND EXPERIMENTAL RESULTS

In this section, we provide experimental evidence that our model described in Section 2 is trainable, by testing it on the classification task of MNIST digits (LeCun and Cortes, 1998). The MNIST dataset of handwritten digits consists of 60,000 training examples and 10,000 test examples. Each example x in the dataset is a gray-scale image of 28 by 28 pixels and comes with a label $d \in \{0, 1, \ldots, 9\}$. We use the same notation y for the one-hot encoding of the target, which is a 10-dimensional vector.

Recall that our model is a recurrently connected neural network with symmetric connections. Here, we train multi-layered networks with 1, 2, and 3 hidden layers, with no skip-layer connections and no lateral connections within layers. Although we believe that analog hardware would be more suited for our model, here we propose an implementation on digital hardware (a GPU). We achieve 0.00% training error. The generalization error lies between 2 and 3% depending on the architecture (**Figure 3**).

For each training example (x, d) in the dataset, training proceeds as follows:

1. Clamp x.

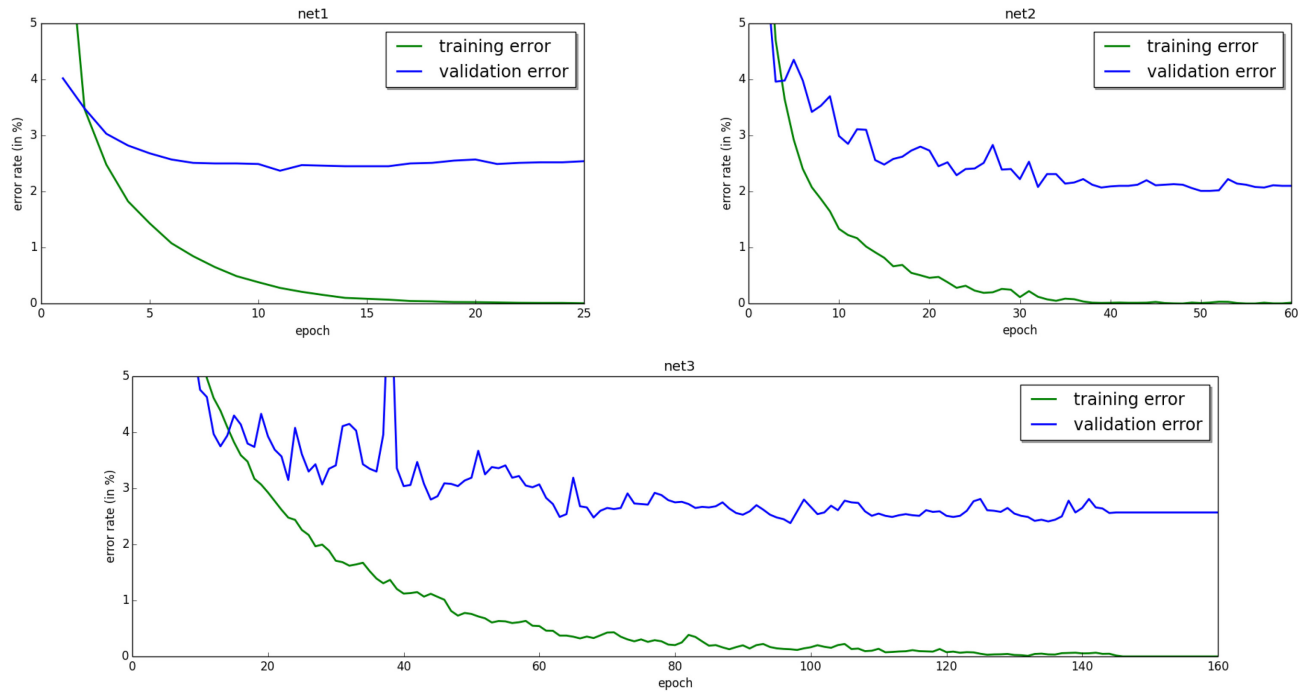


FIGURE 3 | Training and validation error for neural networks with one hidden layer of 500 units (top left), two hidden layers of 500 units **(top right)**, and three hidden layers of 500 units **(bottom)**. The training error eventually decreases to 0.00% in all three cases.

2. Run the free phase until the hidden and output units settle to the free fixed point, and collect $\rho\left(u_i^0\right)\rho\left(u_j^0\right)$ for every pair of units i, j.
3. Run the weakly clamped phase with a "small" $\beta > 0$ until the hidden and output units settle to the weakly clamped fixed point, and collect $\rho\left(u_i^\beta\right)\rho\left(u_j^\beta\right)$.
4. Update each synapse W_{ij} according to

$$\Delta W_{ij} \propto \frac{1}{\beta}\left(\rho\left(u_i^\beta\right)\rho\left(u_j^\beta\right) - \rho\left(u_i^0\right)\rho\left(u_j^0\right)\right). \tag{39}$$

The prediction is made at the free fixed point u^0 at the end of the first phase relaxation. The predicted value y_{pred} is the index of the output unit whose activation is maximal among the 10 output units:

$$y_{\text{pred}} := \arg\max_i y_i^0. \tag{40}$$

Note that no constraint is imposed on the activations of the units of the output layer in our model, unlike more traditional neural networks where a softmax output layer is used to constrain them to sum up to 1. Recall that the objective function that we minimize is the square of the difference between our prediction and the one-hot encoding of the target value:

$$J = \frac{1}{2}\left\|d - y^0\right\|^2. \tag{41}$$

5.1. Finite Differences

5.1.1. Implementation of the Differential Equation of Motion

First we clamp x. Then the obvious way to implement Equation (4) is to discretize time into short time lapses of duration ϵ and to update each hidden and output unit s_i according to

$$s_i \leftarrow s_i - \epsilon\frac{\partial F}{\partial s_i}(\theta, \mathrm{v}, \beta, s). \tag{42}$$

This is simply one step of gradient descent on the total energy F, with step size ϵ.

For our experiments, we choose the hard sigmoid activation function $\rho(s_i) = 0 \vee s_i \wedge 1$, where $\vee$ denotes the max and $\wedge$ the min. For this choice of ρ, since $\rho'(s_i) = 0$ for $s_i < 0$, it follows from Equations (8) and (9) that if $h_i < 0$ then $\frac{\partial F}{\partial h_i}(\theta, \mathrm{v}, \beta, s) = -h_i > 0$. This force prevents the hidden unit h_i from going in the range of negative values. The same is true for the output units. Similarly, s_i cannot reach values above 1. As a consequence s_i must remain in the domain $0 \leq s_i \leq 1$. Therefore, rather than the standard gradient descent (Equation 42), we will use a slightly different update rule for the state variable s:

$$s_i \leftarrow 0 \vee \left(s_i - \epsilon\frac{\partial F}{\partial s_i}(\theta, \mathrm{v}, \beta, s)\right) \wedge 1. \tag{43}$$

This little implementation detail turns out to be very important: if the i-th hidden unit was in some state $h_i < 0$, then Equation (42) would give the update rule $h_i \leftarrow (1-\epsilon)h_i$, which would imply

again $h_i < 0$ at the next time step (assuming $\epsilon < 1$). As a consequence h_i would remain in the negative range forever.

5.1.2. Choice of the Step Size ϵ

We find experimentally that the choice of ϵ has little influence as long as $0 < \epsilon < 1$. What matters more is the *total duration of the relaxation* $\Delta t = n_{\text{iter}} \times \epsilon$ (where n_{iter} is the number of iterations). In our experiments we choose $\epsilon = 0.5$ to keep $n_{\text{iter}} = \Delta t/\epsilon$ as small as possible so as to avoid extra unnecessary computations.

5.1.3. Duration of the Free Phase Relaxation

We find experimentally that the number of iterations required in the free phase to reach the free fixed point is large and grows fast as the number of layers increases (**Table 2**), which considerably slows down training. More experimental and theoretical investigation would be needed to analyze the number of iterations required, but we leave that for future work.

5.1.4. Duration of the Weakly Clamped Phase

During the weakly clamped phase, we observe that the relaxation to the weakly clamped fixed point is not necessary. We only need to "initiate" the movement of the units, and for that we use the following heuristic. Notice that the time constant of the integration process in the leaky integrator equation (Equation 8) is $\tau = 1$. This time constant represents the time needed for a signal to propagate from a layer to the next one with "significant amplitude." So the time needed for the error signals to back-propagate in the network is $N\tau = N$, where N is the number of layers (hiddens and output) of the network. Thus, we choose to perform N/ϵ iterations with step size $\epsilon = 0.5$.

5.2. Implementation Details and Experimental Results

To tackle the problem of the long free phase relaxation and speed-up the simulations, we use "persistent particles" for the latent variables to re-use the previous fixed point configuration for a particular example as a starting point for the next free phase relaxation on that example. This means that for each training example in the dataset, we store the state of the hidden layers at the end of the free phase, and we use this to initialize the state of the network at the next epoch. This method is similar in spirit to the PCD algorithm (Persistent Contrastive Divergence) for sampling from other energy-based models like the Boltzmann machine (Tieleman, 2008).

We find that it helps regularize the network if we choose the sign of β at random in the second phase. Note that the weight updates remain consistent thanks to the factor $1/\beta$ in the update rule $\Delta W_{ij} \propto \frac{1}{\beta}\left(\rho\left(u_i^\beta\right)\rho\left(u_j^\beta\right) - \rho\left(u_i^0\right)\rho\left(u_j^0\right)\right)$. Indeed, the left-derivative and the right-derivative of the function $\beta \mapsto \rho\left(u_i^\beta\right)\rho\left(u_j^\beta\right)$ at the point $\beta = 0$ coincide.

Although the theory presented in this paper requires a unique learning rate for all synaptic weights, in our experiments we need to choose different learning rates for the weight matrices of different layers to make the algorithm work. We do not have a clear explanation for this fact yet, but we believe that this is due to the finite precision with which we approach the fixed points. Indeed, the theory requires to be exactly at the fixed points, but in practice we minimize the energy function by numerical optimization, using Equation (43). The precision with which we approach the fixed points depends on hyperparameters such as the step size ϵ and the number of iterations n_{iter}.

Let us denote by $h_0, h_1, \cdots, h_N$ the layers of the network (where $h_0 = x$ and $h_N = y$) and by W_k the weight matrix between the layers h_{k-1} and h_k. We choose the learning rate α_k for W_k so that the quantities $\frac{\|\Delta W_k\|}{\|W_k\|}$ for $k = 1, \cdots, N$ are approximately the same in average (over training examples), where $\|\Delta W_k\|$ represents the weight change of W_k after seeing a minibatch.

The hyperparameters chosen for each model are shown in **Table 2** and the results are shown in **Figure 3**. We initialize the weights according to the Glorot-Bengio initialization (Glorot and Bengio, 2010). For efficiency of the experiments, we use minibatches of 20 training examples.

6. DISCUSSION, LOOKING FORWARD

From a biological perspective, a troubling issue in the Hopfield model is the requirement of symmetric weights between the units. Note that the units in our model need not correspond exactly to actual neurons in the brain (it could be groups of neurons in a cortical microcircuit, for example). It remains to be shown how a form of symmetry could arise from the learning procedure itself (for example from autoencoder-like unsupervised learning) or if a different formulation could eliminate the symmetry requirement. Encouraging cues come from the observation that denoizing autoencoders without tied weights often end up learning symmetric weights (Vincent et al., 2010). Another encouraging piece of evidence, also linked to autoencoders, is the theoretical result from Arora et al. (2015), showing that the symmetric solution minimizes the autoencoder reconstruction error between two successive layers of rectifying

TABLE 2 | Hyperparameters.

Architecture	Iterations (first phase)	Iterations (second phase)	ϵ	β	α_1	α_2	α_3	α_4
784-500-10	20	4	0.5	1.0	0.1	0.05		
784-500-500-10	100	6	0.5	1.0	0.4	0.1	0.01	
784-500-500-500-10	500	8	0.5	1.0	0.128	0.032	0.008	0.002

The learning rate ϵ is used for iterative inference (Equation 43). β is the value of the clamping factor in the second phase. α_k is the learning rate for updating the parameters in layer k.

(ReLU) units, suggesting that symmetry may arise as the result of an additional objective function making successive layers form an autoencoder. Also, Lillicrap et al. (2014) show that the backpropagation algorithm for feedforward nets also works when the feedback weights are random, and that in this case the feedforward weight tend to "align" with the feedback weights.

Another practical issue is that we would like to reduce the negative impact of a lengthy relaxation to a fixed point, especially in the free phase. A possibility is explored by Bengio et al. (2016) and was initially discussed by Salakhutdinov and Hinton (2009) in the context of a stack of RBMs: by making each layer a good autoencoder, it is possible to make this iterative inference converge quickly after an initial feedforward phase, because the feedback paths "agree" with the states already computed in the feedforward phase.

Regarding synaptic plasticity, the proposed update formula can be contrasted with theoretical synaptic learning rules which are based on the Hebbian product of pre- and post-synaptic activity, such as the BCM rule (Bienenstock et al., 1982; Intrator and Cooper, 1992). The update proposed here is particular in that it involves the temporal derivative of the post-synaptic activity, rather than the actual level of postsynaptic activity.

Whereas our work focuses on a rate model of neurons, see Feldman (2012) for an overview of synaptic plasticity that goes beyond spike timing and firing rate, including synaptic cooperativity (nearby synapses on the same dendritic subtree) and depolarization (due to multiple consecutive pairings or spatial integration across nearby locations on the dendrite, as well as the effect of the synapse's distance to the soma). In addition, it would be interesting to study update rules which depend on the statistics of triplets or quadruplets of spikes timings, as in Froemke and Dan (2002) and Gjorgjievaa et al. (2011). These effects are not considered here but future work should consider them.

Another question is that of time-varying input. Although this work makes back-propagation more plausible for the case of a static input, the brain is a recurrent network with time-varying inputs, and back-propagation through time seems even less plausible than static back-propagation. An encouraging direction is that proposed by Ollivier et al. (2015) and Tallec and Ollivier (2017), which shows that computationally efficient estimators of the gradient can be obtained using a forward method (online estimation of the gradient), which avoids the need to store all past states in training sequences, at the price of a noisy estimator of the gradient.

AUTHOR CONTRIBUTIONS

BS: main contributor to the theory developed in Section 3 and the experimental part (Section 5). YB: main contributor to the theory developed in Section 2.

ACKNOWLEDGMENTS

The authors would like to thank Akram Erraqabi, Alex Lamb, Alexandre Thiery, Mihir Mongia, Samira Shabanian, and Asja Fischer and Devansh Arpit for feedback and discussions, as well as NSERC, CIFAR, Samsung and Canada Research Chairs for funding, and Compute Canada for computing resources. We would also like to thank the developers of Theano[7], for developing such a powerful tool for scientific computing.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: http://journal.frontiersin.org/article/10.3389/fncom.2017.00024/full#supplementary-material

[7] http://deeplearning.net/software/theano/

REFERENCES

Almeida, L. B. (1987). "A learning rule for asynchronous perceptrons with feedback in a combinatorial environment," in *Proceedings of the IEEE First International Conference on Neural Networks*, Vol. 2 (San Diego, CA; New York, NY: IEEE), 609–618.

Arora, S., Liang, Y., and Ma, T. (2015). *Why Are Deep Nets Reversible: A Simple Theory, with Implications for Training*. Technical Report, arXiv:1511.05653, Princeton University.

Bengio, Y., and Fischer, A. (2015). *Early Inference in Energy-Based Models Approximates Back-Propagation*. Technical Report, arXiv:1510.02777, Universite de Montreal.

Bengio, Y., Lee, D.-H., Bornschein, J., and Lin, Z. (2015a). Towards biologically plausible deep learning. arXiv:1502.04156.

Bengio, Y., Mesnard, T., Fischer, A., Zhang, S., and Wu, Y. (2015b). STDP as presynaptic activity times rate of change of postsynaptic activity. arXiv:1509.05936.

Bengio, Y., Mesnard, T., Fischer, A., Zhang, S., and Wu, Y. (2017). STDP as presynaptic activity times rate of change of postsynaptic activity approximates backpropagation. *Neural Comput.* 29, 1–23.

Bengio, Y., Scellier, B., Bilaniuk, O., Sacramento, J., and Senn, W. (2016). Feedforward initialization for fast inference of deep generative networks is biologically plausible. arXiv:1606.01651.

Berkes, P., Orban, G., Lengyel, M., and Fiser, J. (2011). Spontaneous cortical activity reveals hallmarks of an optimal internal model of the environment. *Science* 331, 83–87. doi: 10.1126/science.1195870

Bi, G., and Poo, M. (2001). Synaptic modification by correlated activity: Hebb's postulate revisited. *Annu. Rev. Neurosci.* 24, 139–166. doi: 10.1146/annurev.neuro.24.1.139

Bienenstock, E. L., Cooper, L. N., and Munro, P. W. (1982). Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *J. Neurosci.* 2, 32–48.

Feldman, D. E. (2012). The spike timing dependence of plasticity. *Neuron* 75, 556–571. doi: 10.1016/j.neuron.2012.08.001

Friston, K. J., and Stephan, K. E. (2007). Free-energy and the brain. *Synthese* 159, 417–458. doi: 10.1007/s11229-007-9237-y

Froemke, R. C., and Dan, Y. (2002). Spike-timing-dependent synaptic modification induced by natural spike trains. *Nature* 416, 433–438. doi: 10.1038/416433a

Gerstner, W., Kempter, R., van Hemmen, J., and Wagner, H. (1996). A neuronal learning rule for sub-millisecond temporal coding. *Nature* 386, 76–78.

Gjorgjievaa, J., Clopathb, C., Audetc, J., and Pfister, J.-P. (2011). A triplet spike-timing dependent plasticity model generalizes the bienenstock cooper munro rule to higher-order spatiotemporal correlations. *Proc. Natl. Acad. Sci. U.S.A.* 108, 19383–19388. doi: 10.1073/pnas.1105933108

Glorot, X., and Bengio, Y. (2010). "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of AISTATS' 2010* (Montréal, QC), 249–256.

Hertz, J. A., Krogh, A., Lautrup, B., and Lehmann, T. (1997). Nonlinear backpropagation: doing backpropagation without derivatives of the activation function. *IEEE Trans. Neural Netw.* 8, 1321–1327. doi: 10.1109/72.641455

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Comput.* 14, 1771–1800. doi: 10.1162/089976602760128018

Hinton, G. E., and Sejnowski, T. J. (1986). "Learning and releaming in boltzmann machines," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol. 1, eds D. E. Rumelhart and J. L. McClelland (Cambridge, MA: MIT Press), 282–317.

Hopfield, J. J. (1984). Neurons with graded responses have collective computational properties like those of two-state neurons. *Proc. Natl. Acad. Sci. U.S.A.* 81, 3088–3092.

Intrator, N., and Cooper, L. N. (1992). Objective function formulation of the BCM theory of visual cortical plasticity: statistical connections, stability conditions. *Neural Netw.* 5, 3–17. doi: 10.1016/S0893-6080(05)80003-6

LeCun, Y., and Cortes, C. (1998). *The MNIST database of handwritten digits.* Available online at: http://yann.lecun.com/exdb/mnist/

Lillicrap, T. P., Cownden, D., Tweed, D. B., and Akerman, C. J. (2014). Random feedback weights support learning in deep neural networks. arXiv:1411.0247.

Markram, H., Gerstner, W., and Sjöström, P. (2012). Spike-timing-dependent plasticity: a comprehensive overview. *Front. Synaptic Neurosci.* 4:2. doi: 10.3389/fnsyn.2012.00002

Markram, H., and Sakmann, B. (1995). Action potentials propagating back into dendrites triggers changes in efficacy. *Soc. Neurosci. Abs.* 21.

Mesnard, T., Gerstner, W., and Brea, J. (2016). Towards deep learning with spiking neurons in energy based models with contrastive hebbian plasticity. arXiv:1612.03214.

Movellan, J. R. (1990). "Contrastive Hebbian learning in the continuous Hopfield model," in *Proceedings of the 1990 Connectionist Models Summer School* (San Mateo, CA).

Ollivier, Y., Tallec, C., and Charpiat, G. (2015). *Training Recurrent Networks Online without Backtracking.* Technical report, arXiv:1507.07680, Centre National de la Recherche Scientifique.

O'Reilly, R. C. (1996). Biologically plausible error-driven learning using local activation differences: the generalized recirculation algorithm. *Neural Comput.* 8, 895–938.

Pineda, F. J. (1987). Generalization of back-propagation to recurrent neural networks. 59, 2229–2232.

Salakhutdinov, R., and Hinton, G. E. (2009). "Deep Boltzmann machines," in *International Conference on Artificial Intelligence and Statistic (AISTATS'2009)* (Toronto, ON), 448–455.

Sutskever, I., and Tieleman, T. (2010). "On the convergence properties of contrastive divergence," in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, Vol. 9, eds Y. W. Teh and M. Titterington (Toronto, ON), 789–795.

Tallec, C., and Ollivier, Y. (2017). Unbiased online recurrent optimization. arXiv:1702.05043.

Tieleman, T. (2008). "Training restricted boltzmann machines using approximations to the likelihood gradient," in *Proceedings of the 25th International Conference on Machine Learning* (New York, NY: ACM), 1064–1071.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.* 11, 3371–3408.

Xie, X., and Seung, H. S. (2000). "Spike-based learning rules and stabilization of persistent neural activity," in *Advances in Neural Information Processing Systems 12*, eds S. Solla, T. Leen, and K. Müller (Boston, MA: MIT Press), 199–208.

Xie, X., and Seung, H. S. (2003). Equivalence of backpropagation and contrastive Hebbian learning in a layered network. *Neural Comput.* 15, 441–454. doi: 10.1162/089976603762552988

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

The reviewer SF and handling Editor declared their shared affiliation, and the handling Editor states that the process nevertheless met the standards of a fair and objective review.

frontiers in Computational Neuroscience

ORIGINAL RESEARCH
published: 22 June 2017
doi: 10.3389/fncom.2017.00054

Models of Acetylcholine and Dopamine Signals Differentially Improve Neural Representations

Raphaël Holca-Lamarre[1,2]*, *Jörg Lücke*[3,4†] and *Klaus Obermayer*[1,2†]

[1] Neural Information Processing Group, Fakultät IV, Technische Universität Berlin, Berlin, Germany, [2] Bernstein Center for Computational Neuroscience, Berlin, Germany, [3] Cluster of Excellence Hearing4all and Research Center Neurosensory Science, Carl von Ossietzky Universität Oldenburg, Oldenburg, Germany, [4] Machine Learning Lab, Department of Medical Physics and Acoustics, Carl von Ossietzky Universität Oldenburg, Oldenburg, Germany

Biological and artificial neural networks (ANNs) represent input signals as patterns of neural activity. In biology, neuromodulators can trigger important reorganizations of these neural representations. For instance, pairing a stimulus with the release of either acetylcholine (ACh) or dopamine (DA) evokes long lasting increases in the responses of neurons to the paired stimulus. The functional roles of ACh and DA in rearranging representations remain largely unknown. Here, we address this question using a Hebbian-learning neural network model. Our aim is both to gain a functional understanding of ACh and DA transmission in shaping biological representations and to explore neuromodulator-inspired learning rules for ANNs. We model the effects of ACh and DA on synaptic plasticity and confirm that stimuli coinciding with greater neuromodulator activation are over represented in the network. We then simulate the physiological release schedules of ACh and DA. We measure the impact of neuromodulator release on the network's representation and on its performance on a classification task. We find that ACh and DA trigger distinct changes in neural representations that both improve performance. The putative ACh signal redistributes neural preferences so that more neurons encode stimulus classes that are challenging for the network. The putative DA signal adapts synaptic weights so that they better match the classes of the task at hand. Our model thus offers a functional explanation for the effects of ACh and DA on cortical representations. Additionally, our learning algorithm yields performances comparable to those of state-of-the-art optimisation methods in multi-layer perceptrons while requiring weaker supervision signals and interacting with synaptically-local weight updates.

Keywords: acetylcholine, dopamine, neuromodulator, sensory representations, neural networks, biology-inspired learning, representation learning

OPEN ACCESS

Edited by:
Sander Bohte,
Centrum Wiskunde & Informatica, Netherlands

Reviewed by:
Malte J. Rasch,
Beijing Normal University, China
Wouter Kruijne,
VU University Amsterdam, Netherlands

***Correspondence:**
Raphaël Holca-Lamarre
raphael@bccn-berlin.de

[†] *Joint last authorship*

Received: *01 December 2016*
Accepted: *07 June 2017*
Published: *22 June 2017*

Citation:
Holca-Lamarre R, Lücke J and Obermayer K (2017) Models of Acetylcholine and Dopamine Signals Differentially Improve Neural Representations. Front. Comput. Neurosci. 11:54. doi: 10.3389/fncom.2017.00054

1. INTRODUCTION

Neurons in the cortex represent countless features of sensory signals, from the frequencies of photons falling on the retina to high-level attributes like quantities and numbers. The particular form a sensory representation takes is critical to perception. For instance, experienced musicians display enhanced sensory representations which putatively explain their finer perceptual abilities

(Elbert et al., 1995; Pantev et al., 1998, 2001). This view is further supported by the observation that, following discrimination training, improvements in perceptual sensitivity correlate with the degree of reorganization in cortical representations (Recanzone et al., 1992, 1993; Weinberger, 2003; Polley et al., 2006). On the other hand, perceptual disorders like phantom limb pain (Ramachandran et al., 1992; Halligan et al., 1993; Flor et al., 2006) or tinnitus (Eggermont and Roberts, 2004) appear to be correlates of degenerate sensory representations.

In animals, sensory representations undergo modifications in various circumstances, for instance following extensive perceptual training (Weinberger and Bakin, 1998; Harris et al., 2001; Schoups et al., 2001; Fletcher and Wilson, 2002; Fritz et al., 2003; Wang et al., 2003; Bao et al., 2004; Yang and Maunsell, 2004; Polley et al., 2006; Poort et al., 2015), repeated sensory exposure (Han et al., 2007; Kim and Bao, 2009), cortical stimulation, (Godde et al., 2002; Dinse et al., 2003; Tegenthoff et al., 2005), or sensory deprivation (Calford and Tweedale, 1988; Allard et al., 1991; Gambino and Holtmaat, 2012). Additionally, the neuromodulators acetylcholine (ACh) and dopamine (DA) bear potent effects on cortical representations. In particular, repeated efflux of either ACh (Kilgard and Merzenich, 1998a; Froemke et al., 2007, 2013; Gu, 2003; Weinberger, 2003) or DA (Bao et al., 2001; Frankó et al., 2010) coinciding with a stimulus strengthens the responses of neurons to this stimulus and enlarges its cortical representation.

ACh and DA are critical to forms of learning which require modifications of sensory representations. For instance, lesion of the cholinergic (Butt and Hodge, 1995; Fletcher and Wilson, 2002; Conner et al., 2003; Wilson et al., 2004; Conner et al., 2010) or dopaminergic (Kudoh and Shibuki, 2006; Molina-Luna et al., 2009; Hosp et al., 2011; Luft and Schwarz, 2009; Schicknick et al., 2012) system disrupts perceptual and motor learning as well as the associated plasticity in cortical maps. These observations suggest that the neuromodulators orchestrate plastic changes that refine cortical representations and give rise to perceptual and motor learning.

In physiological conditions, ACh transmission appears to signal attentional effort, a construct reflecting both the relevance and difficulty of a task (Himmelheber et al., 2000; Arnold et al., 2002; Kozak et al., 2006; Sarter et al., 2006). DA carries information relative to reward-prediction errors (RPEs) (Schultz et al., 1997; Schultz, 2007, 2010). Although their release properties are relatively well defined, the functional roles these signals serve in shaping neural representations is unclear.

Much like the cortex, artificial neural networks (ANNs) represent input data in the form of neural activation. As for other machine learning algorithms, the performance of ANNs critically depends on the representation data take. The most widely used learning rule for ANNs, the error back-propagation algorithm (Werbos, 1974; Rumelhart et al., 1985), learns representations optimised for specific tasks. Although, the back-propagation algorithm yields remarkable performances, it is unlikely to be implemented in biological neural structures and it also bears its own limitations. For instance, in order to compute the error function, a target output must be specified for each training example, making training data expensive to acquire. Additionally, weight updates require information not available locally at the weights which limits the use of the back-propagation algorithm in physical devices like neuromorphic chips.

In the present work we explore the use of signals inspired from ACh and DA for learning in a neural network model. This effort serves two aims: first, to shed light on the functional roles of ACh and DA in shaping cortical representations and, second, to provide inspiration for novel training methods for ANNs.

Previous studies examine the roles of ACh and DA in neural information processing. Weinberger and Bakin (1998) develop a model of ACh signaling to investigate its function in classical conditioning. Li and Cleland (2013) present a detailed biophysical model of ACh neuromodulation in the olfactory bulb. However, these studies do not see to the perceptual benefits of long-term plasticity induced by ACh. Other work tackle the question of DA-modulated plasticity in neural networks. Roelfsema and colleagues show that a signal inspired from DAergic signaling allows a network to learn various classification tasks (Roelfsema and Ooyen, 2005; Roelfsema et al., 2010; Rombouts et al., 2012). Similarly, other models make use of DA-like reinforcement signals to learn stimulus-response associations (e.g., Law and Gold, 2009; Liu et al., 2010). In these cases, however, the models for the plastic effects of DA were chosen to carry out reinforcement learning rather than to tally with experimental observations.

In contrast with previous work, we base our modeling effort on the well-documented observation that pairing ACh or DA release with a stimulus boosts neural responses to the stimulus. We use this model to study the perceptual benefits of ACh- and DA-induced plasticity under natural release conditions. In more details, we make use of a Hebbian-learning neural network and simulate the physiological release schedules of ACh and DA. In the model, ACh activation approximates attentional demand while DA activation arises from RPEs. We find that the neuromodulators trigger distinct changes in representations that both improve the network's classification performance. Specifically, ACh leads to changes in synaptic weights such that more neurons are dedicated to stimuli that are challenging for the network. DA adapts synaptic weights to the reward contingencies of a task, thereby sharpening neural tuning with respect to the classes of the task. These results provide a functional explanation for the roles of cholinergic and dopaminergic signals in refining cortical representations.

Our learning algorithm offers several advantages from a practical perspective. First, the network achieves performances comparable to those of state-of-the-art optimisation methods used to train multi-layer perceptrons (MLPs) while requiring weaker supervision signals. Second, learning takes place even in the absence of environmental feedback. And third, weight updates are based on synaptically-local information and on two signals broadcasted identically to all neurons. These features may make the algorithm interesting for

functional applications such as learning in neuromorphic processors.

2. METHODS

2.1. Hebbian Network Model

For our study, we make use of a Hebbian-learning neural network model introduced by Keck et al. (2012). The learning mechanisms implemented in this model achieve approximately optimal learning in terms of maximum likelihood estimation (see original publication for a detailed discussion). As a theoretically well-founded and biologically realistic model, this network is a natural starting point for our work. In this section, we briefly present the original model and then describe our simulation of the neuromodulators ACh and DA.

The network consists of three layers, an input, a representation, and a classification layer (**Figure 1**). Input values activate neurons in the first layer; activity then propagates through the network in the following steps.

2.1.1. Feedforward Inhibition

In mammals, the responses of sensory neurons are largely invariant to contrast in sensory stimuli (Sclar et al., 1990; Stopfer et al., 2003; Mante et al., 2005; Assisi et al., 2007; Olsen and Wilson, 2008), in part due to rapid feedforward inhibition (Pouille and Scanziani, 2001; Swadlow, 2003; Mittmann et al., 2005; Wehr and Zador, 2005; Pouille et al., 2009; Isaacson and Scanziani, 2011). To emulate this process, neural activations in the input layer are normalized:

$$y_d = (A - D)\frac{\tilde{y}_d}{\sum_{d'=1}^{D} \tilde{y}_{d'}} + 1\,, \tag{1}$$

where $\vec{\tilde{y}}$ are input data, A is a normalization constant, and D is the number of input neurons. This form of normalization yields contrast-invariant responses in representation neurons. For the dataset used in this work, $D = 28 \times 28 = 784$ input neurons.

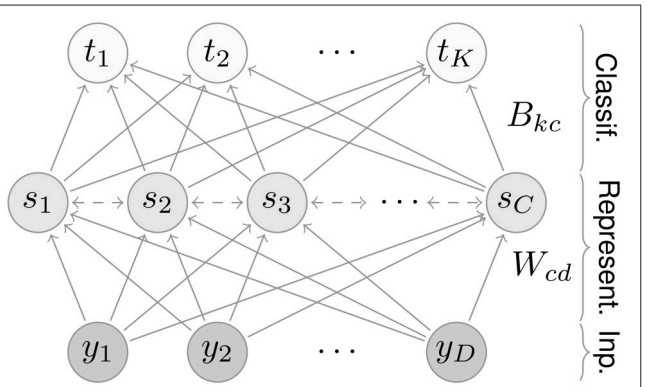


FIGURE 1 | Network architecture. The network contains three layers: an input, a representation, and a classification layer. For the MNIST dataset, the input and classification layers contain $D = 28 \times 28 = 784$ and $K = 10$ neurons, respectively. The number of representation neurons is variable; for most results we use $C = 7 \times 7 = 49$ neurons.

For other hyper-parameters, values are determined through grid search to maximize classification performance (see **Table A1** in Appendix section).

2.1.2. Input Integration

Neurons in the representation layer integrate their input through a weighted sum:

$$I_c = \sum_{d=1}^{D} S(W_{cd}) y_d\,, \tag{2}$$

where W is the weight matrix between the input and representation layers and $S(\cdot)$ is a linearised logarithm function given by:

$$S(W_{cd}) = \begin{cases} W_{cd} & \text{if } W_{cd} < 1 \\ \log(W_{cd}) + 1 & \text{if } W_{cd} \geq 1\,. \end{cases} \tag{3}$$

Taking the logarithm of W_{cd} guarantees approximate optimal learning of the weights, with the linearisation ensuring that the function is never negative for $W_{cd} \geq 0$.

2.1.3. Lateral Inhibition

The integrated input is fed through a softmax function that models global lateral inhibition:

$$s_c = \frac{\exp(I_c)}{\sum_{c'} \exp(I_{c'})}\,. \tag{4}$$

2.1.4. Hebbian Learning

Hebbian learning takes place between the input and representation neurons:

$$\Delta W_{cd} = \epsilon \cdot (s_c y_d - s_c W_{cd})\,, \tag{5}$$

where ϵ is the learning rate.

2.1.5. Classification

We subject the network to a classification task of images of handwritten digits from the MNIST dataset (LeCun et al., 1998b). These input images provide stimuli of intermediate complexity and high-dimensionality akin to natural sensory stimuli, making them a popular dataset to study neural information processing (Nessler et al., 2013; Schmuker et al., 2014). These data consist of gray-scale images with pixel values in the range $[0, 255]$ fed as input $\vec{\tilde{y}}$ to the first layer.

In the classification layer, we use statistical inference to decode activity in the representation layer. Given an input pattern $\vec{y}$ and the model parameters Θ, we want to infer the class of the input pattern, that is, to compute the posterior $\Pr(k \,|\, \vec{y}, \Theta)$. Here, we approximate the posteriors using the labels of the input images. We first compute a value B_{kc}:

$$B_{kc} := \frac{1}{N_m} \sum_{n=1}^{N_m} \Pr(c|\vec{y}^{(n)}, W) = \frac{1}{N_m} \sum_{n=1}^{N_m} s_c^{(n)}\,, \tag{6}$$

with N_m input patterns $\vec{y}^{(n)}$ bearing a label $m = k$. The matrix B can be interpreted as the weights between the representation

and classification layers. This matrix is updated after every presentation of 100 images, or roughly 600 times during one iteration over the dataset. The posteriors are approximated as:

$$\Pr(k \mid \vec{y}, \Theta) \approx t_k = \sum_{c=1}^{C} \frac{B_{kc} s_c}{\sum_{k'=1}^{K} B_{k'c}} . \tag{7}$$

As a classification result $\hat{m}$, we take the unit with the largest value of approximation to the posterior:

$$\hat{m} = \underset{k=1}{\overset{K}{\operatorname{argmax}}}(t_k) . \tag{8}$$

This hierarchical formulation allows to decode activity in the representation layer, providing a probabilistic classification of the input images.

Previous work based on a fully probabilistic description of the Hebbian-learning network model (Forster et al., 2016; Forster and Lücke, 2017) shows that local Hebbian learning converges to the weight matrix B without requiring the non-local summation over k. This is true also when using a small fraction ($\approx$ 1%) of labeled training examples. Learning the classification weights can therefore be achieved while respecting biological constraints. For this work, we mainly focus on the standard fully labeled setting, as is customary (Keck et al., 2012; Nessler et al., 2013; Schmuker et al., 2014; Diehl and Cook, 2015; Neftci et al., 2015), but also provide results for experiments with very few labels.

2.2. Model of the Neuromodulators

2.2.1. Effects on Plasticity

We extend the network model described above to emulate the effects of ACh and DA on neural representations. Specifically, we simulate the impact of the neuromodulators as a modulation of the network's learning rate:

$$\text{acetylcholine:} \quad \Delta W_{cd} = \epsilon \cdot ACh \cdot (s_c y_d - s_c W_{cd}) , \tag{5a}$$

$$\text{dopamine:} \quad \Delta W_{cd} = \epsilon \cdot DA \cdot (s_c y_d - s_c W_{cd}) , \tag{5b}$$

where ACh and DA represent the activation of the corresponding neuromodulatory system. This model is in general agreement with experimental observations in that both ACh (Bröcher et al., 1992; Chun et al., 2013) and DA (Blond et al., 2002; Sun et al., 2005; Matsuda et al., 2006) are reported to promote synaptic plasticity. This model for the neuromodulators was chosen so as to reproduce the results of pairing experiments in mammals (see Results section).

2.2.2. Acetylcholine and Attentional Efforts

ACh release in the mammalian neocortex is tightly linked with attentional processes. For instance, as rats detect a behaviorally meaningful sensory cue, a spike in cortical ACh accompanies the reorientation of their attention towards the cue (Parikh et al., 2007). Additionally, when rats perform a task requiring sustained attention, the concentration of ACh in their prefrontal cortices more than doubles compared to control (Arnold et al., 2002; Kozak et al., 2006). In the course of such tasks, distractors that further tax the animals' attentional systems trigger supplemental ACh release (Himmelheber et al., 2000; Kozak et al., 2006). These observations indicate that the cholinergic system responds to events demanding an animal's attention such as relevant stimuli or challenging tasks. In this sense, ACh transmission reflects the cognitive construct of attentional effort defined as a subject's motivated effort to maintain performance under challenging conditions (Sarter et al., 2006).

In the present work, we model ACh activation to approximate attentional demand. To quantify how demanding a stimulus is for the network, we use the network's classification confidence. Classification confidence is measured as the classifier's maximal posterior over the digit classes, $\kappa = \max_{k=1}^{K}(t_k)$. Classification confidence strongly correlates with classification accuracy ($r = 0.89$, **Figure 2A**) indicating that this measure is suitable to quantify stimulus demand. For each stimulus, the value of the ACh variable is given by:

$$ACh = \frac{\alpha}{1.0 + \exp(\beta \cdot (\bar{\kappa}_{\hat{m}}/\bar{\kappa} - 1.0))} \tag{10}$$

where $\bar{\kappa}_{\hat{m}}$ is the network's average classification confidence for the inferred class of the current stimulus, $\bar{\kappa}$ is the average classification confidence for all stimuli, and α and β are hyper-parameters of the sigmoid function whose values are determined through grid search (**Figure 2**). According to this formulation, the lower the classification confidence (i.e., the greater the stimulus difficulty), the larger the ACh activation. Note that, to compute the average classification confidence over the digit classes, we use the network's inferred classification ($\hat{m}$) and not the stimulus label. Thus, for a given stimulus, ACh activation is evaluated without requiring immediate environmental information. Also note that the classification confidence for the same stimulus may vary during training as the network's weight matrices W and B are updated.

2.2.3. Dopamine and Reward Prediction Errors

DA efflux in animals follows RPEs (Schultz et al., 1997; Satoh et al., 2003; Tobler et al., 2005; Schultz, 2010). We reproduce this release schedule in the model as follows. First, we allow explorative decision making by injecting additive noise in the activation of representation neurons (**Figure 3**):

$$I_c = \sum_{d=1}^{D} S(W_{cd}) y_d + \eta_c ,$$

$$\eta_c \sim \mathcal{N}(0, \upsilon) ,$$

where $\mathcal{N}$ is a normal distribution with zero mean and variance υ. This method for exploration approximates the softmax rule for action selection in reinforcement learning (Sutton and Barto, 1998). Following this rule, actions are selected stochastically with the probability of selecting an action proportional to its expected reward. The parameter υ corresponds to the temperature parameter of the softmax rule: for $\upsilon \to \infty$, all classification decisions have equal probabilities; for $\upsilon \to 0^+$, classification is purely exploitative. We find the optimal value for υ through grid search.

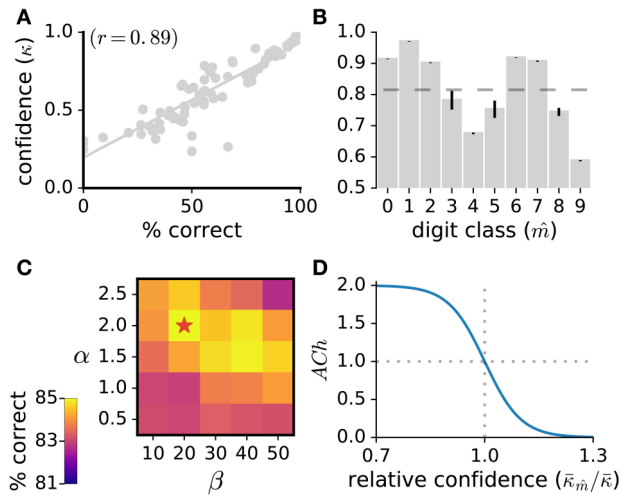


FIGURE 2 | (A) Classification confidence strongly correlates with classification accuracy. Here, we measure the network's classification confidence for the test images of the MNIST dataset, bin the classification confidence (bin size of 0.02%) and calculate the average correct classification for each bin. **(B)** Average classification confidence $\bar{\kappa}_{\hat{m}}$ for the 10 digit classes, with the mean confidence over all classes $\bar{\kappa}$ indicated as a dashed line. Data are the mean of 10 runs, error bars indicate the standard deviation across runs. **(C)** Parameter exploration for the α and β parameters of the ACh release function. A star indicates the parameter set yielding maximal accuracy. **(D)** ACh activation function (Equation 10) taking as input the relative confidence $\bar{\kappa}_{\hat{m}}/\bar{\kappa}$. This ratio quantifies the demand of the current stimulus.

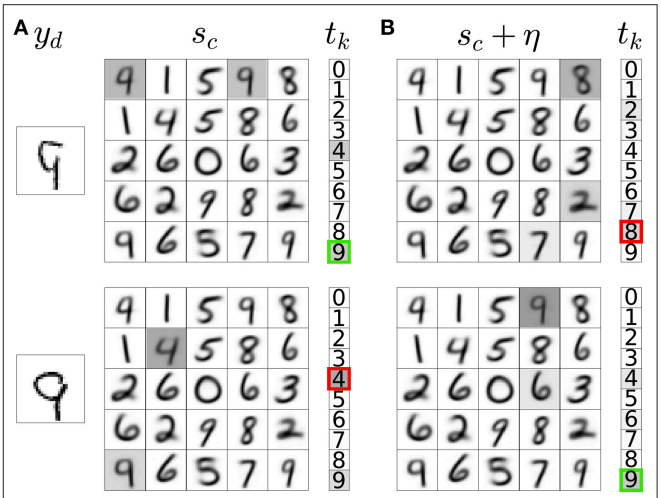


FIGURE 3 | Noisy neural activation in the representation layer allows explorative classification decisions. **(A)** Activations of input neurons (y_d), weights of a subset of representation neurons with their corresponding activations prior to noise injection (s_c, gray highlights), and activations of classification neurons (t_k) with the network's classification output indicated as a bold colored outline. The example input images are correctly (top) and incorrectly classified (bottom). **(B)** Noise addition in the activations of representation neurons leads to incorrect (top) and correct (bottom) explorative classification decisions. In these two different outcomes of exploration, the variable *DA* in Equation 5b takes a distinct value ($\delta_{-/-}$ and $\delta_{-/+}$, respectively).

We then compute the classification output for each $\vec{y}$ with and without the addition of noise η. If noise addition results in a classification decision that is different from the decision without noise addition, the classification is labeled as explorative; otherwise it is labeled as exploitative. If the network takes an exploitative decision it is said to predict a reward ($+pred$); if it takes an explorative decision it is said to not predict a reward ($-pred$). The network is rewarded for taking correct classification decisions ($+rew$) and not rewarded for incorrect decisions ($-rew$). The difference between the predicted and delivered rewards gives rise to a RPE. There are four possible RPE scenarios. In each of these cases, the *DA* variable in Equation 5b takes a distinct value:

$$DA = \begin{cases} \delta_{+/+} & \text{if } +pred \text{ and } +rew \\ \delta_{+/-} & \text{if } +pred \text{ and } -rew \\ \delta_{-/+} & \text{if } -pred \text{ and } +rew \\ \delta_{-/-} & \text{if } -pred \text{ and } -rew \end{cases} \tag{12}$$

where $\delta_{./.}$ are constants whose values are determined through 4-dimensional parameter search to maximize classification performance.

2.2.4. Critical period

We are interested in changes in sensory representations triggered by neuromodulators in adult animals. Adult animals possess stable neural representations of their environment learned in early life during a brief window of heightened plasticity. During this so-called critical period, the response properties of neurons rapidly adjust to the statistical structure of sensory stimuli (Sengpiel et al., 1999; de Villers-Sidani et al., 2007; Han et al., 2007; Barkat et al., 2011).

As a model of this critical period, we pre-train the network solely through Hebbian learning (Equation 5). The network then learns synaptic weights based on correlations in the activation of input neurons, with weights that resemble the different digit classes. The weights in the representation layer are then learned solely through the statistics of the input images and do not reflect the task to be performed. As learning progresses, performance on the classification task increases and eventually saturates. Once performance reaches a plateau, we allow the release of ACh or DA. As an additional control condition, we also continue training the network through Hebbian learning. Omitting the pre-training results in the same functional performance but, without it, the optimal DA activation values found through parameter search differ (see **Figure 6**).

3. RESULTS

3.1. Pairing Experiment

In animals, coupling a stimulus with the release of either ACh (Kilgard and Merzenich, 1998a; Weinberger, 2003; Froemke et al., 2007, 2013) or DA (Bao et al., 2001; Frankó et al., 2010) triggers long-lasting changes in sensory representations. Specifically, sensory neurons increase their responses to the paired stimulus, resulting in more neurons preferring this

stimulus. To test whether our model of ACh and DA is in agreement with this observation, we perform a similar experiment. The experiment consists of coupling all stimuli of a target class with *ACh* or $DA = \rho$ in Equations 5a or 5b, where ρ is a constant > 1 (**Figure 4A**). Stimuli of all other classes have *ACh* and $DA = 1$. We then examine the distribution of class preferences in the network. The preferred digit class of a neuron is determined by taking $\text{argmax}_{k=1}^{K}(B_{kc})$ which gives the class to which neuron c maximally responds to. We find that the pairing protocol increases the responses of individual neurons to the paired stimulus class and augments the number of neurons preferring this class, in agreement with experimental data (**Figures 4B–G**). Furthermore, the procedure reduces the number of units tuned to classes close to the paired one (class closeness is measured as the Euclidean distance between the averages of all training examples of each class). These findings are in line with pairing experiments with DA showing that the cortical representations of frequencies neighboring a paired tone shrink as a result of the pairing procedure (**Figure 4G**; Bao et al., 2001). This observation however contrasts with pairing experiments with ACh which result in enlargements of the cortical representations of both the paired frequency and adjacent ones (Kilgard and Merzenich, 1998a). For this work, this difference in the effects of ACh and DA is not taken into account.

3.2. Physiological Release Schedule

3.2.1. Optimal Release Values

With our model in general agreement with the results of pairing experiments, we can now study the effects of the natural release schedules of ACh and DA. We first pre-train the network through Hebbian learning. As training progresses, performance saturates (**Figure 5**, inset). After this point, we allow the release of ACh or DA. We perform parameter search to identify the optimal values for parameters α and β in Equation 10 (**Figure 2C**) and for the $\delta_{./.}$ constants in Equation 12 (**Figure 6**). In the case of the $\delta_{./.}$ constants, we find that for surprising rewards ($-pred$, $+rew$) the optimal $\delta_{-/+}$ is positive while in the absence of an expected reward ($+pred$, $-rew$) the optimal $\delta_{+/-}$ is negative. For correctly predicted rewards (either $+pred$, $+rew$ or $-pred$, $-rew$) the optimal $\delta_{+/+}$ and $\delta_{-/-}$ are close to zero. This optimal activation profile matches that observed in primates (Schultz et al., 1997; Tobler et al., 2005), **Figures 6B–C**).

3.2.2. Effects of ACh

Visual inspection of the weights of the network (**Figure 7A**) indicates that ACh alters the number of neurons dedicated to the different digit classes. For instance, there are more neurons resembling a "4" and fewer neurons resembling a "1" after training with ACh. We quantify this redistribution by determining the preferred class of a representation neuron. For Hebbian learning, the distribution of preferred classes is close to uniform but not entirely so (**Figure 7B**). There is a positive correlation between the number of neurons dedicated to a class and the network's performance on this class ($r = 0.22$,

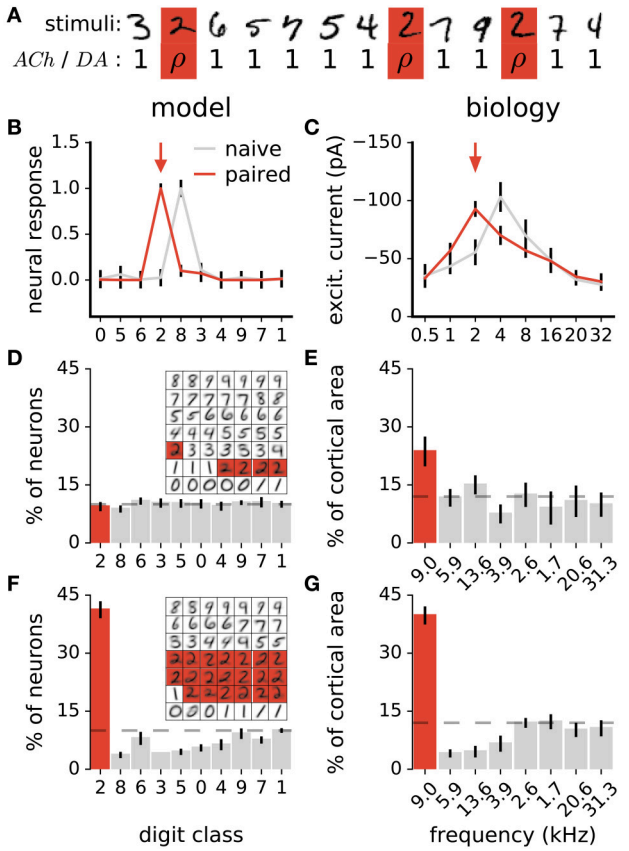


FIGURE 4 | Stimulus pairing with ACh or DA enhances the stimulus' representation. **(A)** Simulation of the neuromodulator-stimulus pairing protocol. The *ACh* or *DA* variables in Equations 5a or 5b is set to a constant value $\rho > 1$ for stimuli of the paired class ("2", in this case) and to 1 for all other classes. **(B)** Mean responses of a neuron to the digit classes. Traces are before and after pairing images labeled as "2" (arrow) with neuromodulator activation $\rho = 20$. Classes are ordered with their distance from class "0" (see text for details). Error bars are a standard deviation. **(C)** Synaptic tuning curves of a neuron in the rat primary auditory cortex. Traces are before and after a 2 kHz tone (arrow) is paired with high-frequency electrical stimulation of the nucleus basalis triggering ACh release. Error bars are the standard error of the mean. Reproduced from Froemke et al. (2007), with permission. In the model and in animals, the pairing procedure boosts responses of individual neurons to the paired stimulus. **(D,F)** Histogram of class preferences in the network model before **(D)** and after **(F)** the pairing manipulation. Classes are ordered with their distance from the paired class. Dashed line is a uniform distribution, data are the mean of 10 runs, error bars indicate a standard deviation. Inset: weights of the representation neurons for an example network; highlights indicate neurons whose preferred classes are "2". **(E,G)** Histogram of best frequencies in the auditory cortices of rats before **(E)** and after **(G)** a 9 kHz tone is paired with stimulation of midbrain dopaminergic neurons. Frequencies are ordered to their difference from the paired tone. Modified from Bao et al. (2001), with permission. In the simulation as in biology, the pairing protocol enhances the representation of the paired stimulus and suppresses that of neighboring ones.

Figure 7D), suggesting that representing a class with more neurons is beneficial to performance.

Training with ACh redistributes class preferences in the network, leading to a less uniform distribution. Specifically, ACh increases the number of neurons dedicated to challenging classes while easier classes are represented with fewer units.

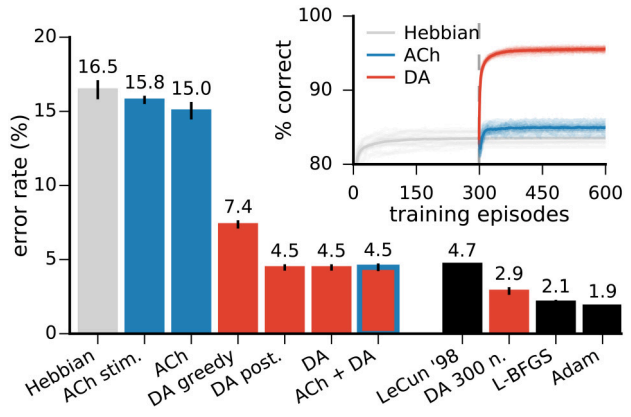


FIGURE 5 | Neuromodulator release improves the network's classification performances. Left bar plots: error rates on the MNIST test dataset in networks with 49 representation neurons. All approaches with neuromodulators lead to significant improvements over Hebbian learning alone. "ACh stim." is for stimulus-wise ACh activation, "DA greedy" is for a network without exploration, "DA post." is for a network using the classifier's posteriors as an approximation to the expected value of the reward (see text for details). DA alone and DA with ACh yield the best performance. Right bar plots: comparison with other training methods for MLPs. All results are for networks of the same architecture, namely a single hidden layer with 300 units. LeCun '98 are the original results from LeCun et al. (1998a) on the MNIST dataset. L-BFGS and Adam are optimisation methods for MLPs (see Appendix for details). Inset: progression of the test performance for the networks with 49 neurons. Darker traces are averages over 20 runs, lighter traces are individual runs. Data for the bar plots are the mean of 20 runs, error bars indicate a standard deviation.

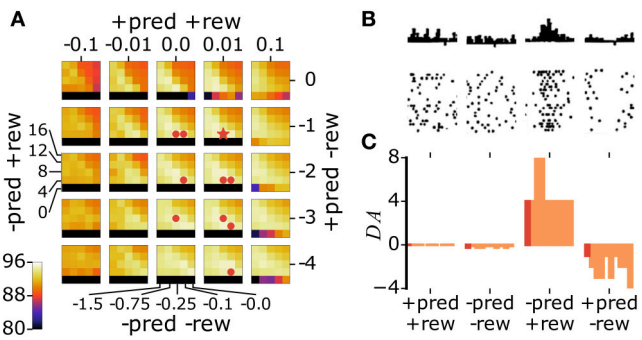


FIGURE 6 | The model's optimal DA activation profile matches the one reported in mammals. **(A)** We explore different values for the four $\delta_{./.}$ constants through grid search and report the classification performance of the network (colored axis, data are averages over 10 runs). A star indicates the best parameter set, dots indicate parameter sets yielding performances not statistically significantly different from that of the best set ($p > 0.01$). **(B)** Firing of dopaminergic neurons in monkeys in RPE scenarios equivalent to those of the model (modified from Schultz et al. (1997), with permission). **(C)** Bar plot of the best parameter set (dark red) and sets not significantly different from best (light red). The parameter sets are sorted in decreasing order of their classification accuracies, from left to right.

Consider for example the classes "1" and "4," the stimuli on which the network performs best and worst, respectively (**Figure 7C**, top row). ACh release leads to a respective decrease and increase in the number of neurons preferring these classes (**Figure 7B**). The redistribution of neurons elicited by ACh raises the network's accuracy on the difficult classes (e.g., "4") and lowers performance on the easy classes (e.g., "1," **Figure 7C**, middle row). ACh thus reverses the correlation between neuron count and performance ($r = -0.79$, **Figure 7D**). On average over all classes, performance rises from 83.5 ± 0.7% with Hebb's rule alone to 85.0 ± 0.6% when supplemented with ACh, corresponding to a relative decrease of 12% in the error rate.

In addition to ACh activation computed as an average over the classes $\hat{m}$, we experiment with stimulus-wise ACh activation. Here, the value of the *ACh* variable is determined for each individual stimulus based on the classifier's posterior for this stimulus (specifically, we use the term κ instead $\bar{\kappa}_{\hat{m}}$ in Equation 10). Although this approach also improves performance, the gains in accuracy are of smaller magnitude than if ACh activation is computed as an average over the classes (**Figure 5**, "ACh stim."). We explain this outcome as the learning mechanism attributing a too great representational importance to demanding but detrimental data, for instance miss-labeled or outlier data points.

3.2.3. Effects of DA

In contrast with ACh signalling, DA bears little effect on the number of neurons responsive to the different classes (**Figure 7B**). For both Hebbian and DA-based learning, the distribution of the neurons' preferred digit class is close to uniform. The positive correlation between neuron count and classification performance also remains after training with DA (Hebbian: $r = 0.22$, DA: $r = 0.20$).

Visual inspection of the weights suggests that DA makes neurons' weights more selective to specific digit classes. Consider the example weights shown in **Figure 8A**. Weights in one column are for corresponding neurons in a Hebbian and DA network (the networks were initialised with the same random seed). Weights in the Hebbian model are rather poorly tuned to the digit classes (e.g., the neuron resembling a "3," "5," and "8" in the second column of **Figure 8A**). On the other hand, DA-based learning leads to weights that more closely correspond to specific digits. This observation can be quantified by measuring the average responses of neurons to the different classes (first and third rows in **Figure 8A**). The measure shown indicates that Hebbian learning yields neurons exhibiting strong responses to multiple stimulus classes, i.e., with a broad tuning. Training with DA yields more sharply tuned weights as units respond almost exclusively to a single digit category.

On average over all neurons, DA generates a 17% increase in neurons' activations to their preferred classes, accompanied by a 84% reduction to non-preferred classes (**Figure 7E**). These modifications amount to neuron weights being more selective to specific digits, or having a sharper tuning. We quantify such neural selectivity as the difference between a neuron's mean response to stimuli of its preferred class and its mean response to stimuli of all other classes:

$$\zeta_c = \frac{\bar{s}_c^{\bullet} - \bar{s}_c^{\circ}}{\bar{s}_c^{\bullet}}, \tag{13}$$

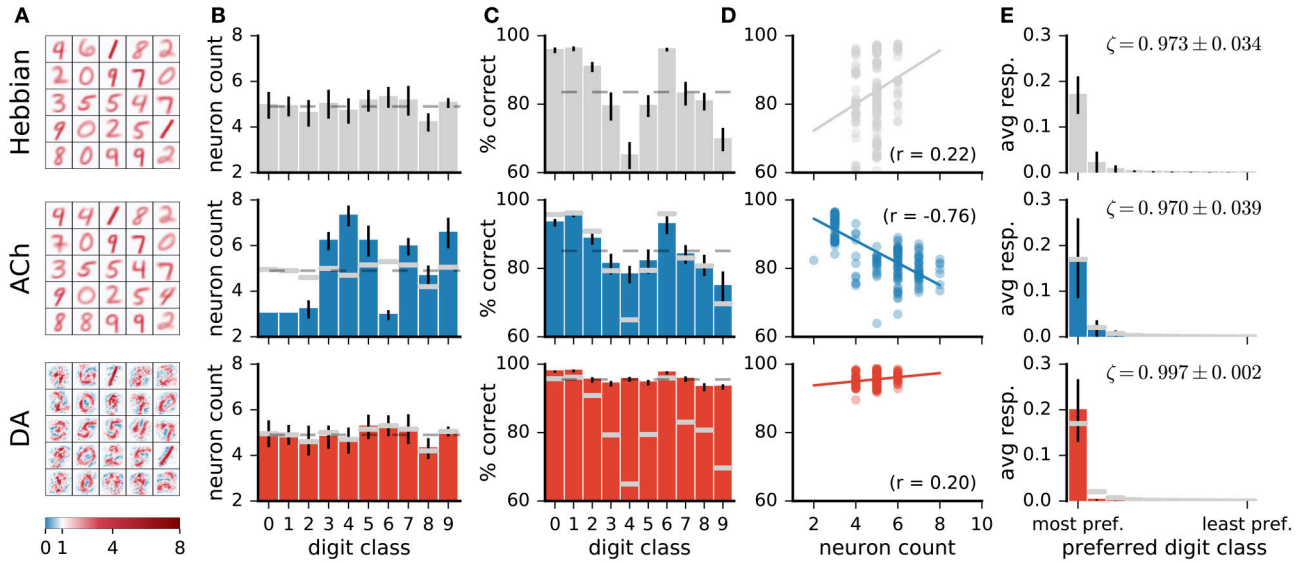


FIGURE 7 | Network changes following neuromodulator release. **(A)** Weights of a subset of representation neurons (25 out of 49). **(B)** Histogram of class preferences. Dashed line is a uniform distribution. ACh increases the number of neurons preferring the more challenging classes. **(C)** Performance of the network on the different classes. Dashed line is average over all classes. **(D)** Performance on a class as a function of the number of neurons preferring this class. There is a positive correlation for Hebbian and DA-based learning. The learning mechanism in ACh reverses this correlation. **(E)** Average responses of neurons to the digit classes, with the classes ordered by the neurons' preference. ζ indicates mean neural selectivity (see text). DA sharpens the responses of neurons, enhancing their activations to their preferred classes and reducing their activations to non-preferred classes. Data are for 20 runs, error bars indicate a standard deviation. Gray overlaid bars in (**B,C,** and **E**) are values for Hebbian learning for comparison.

where $\bar{s}_c^{\bullet}$ and $\bar{s}_c^{\circ}$ are the average responses of neuron c to stimuli of its preferred and non-preferred classes, respectively. Here, $\zeta_c = [0, 1]$, where $\zeta_c = 0$ is a neuron that responds equally strongly to all stimuli and $\zeta_c = 1$ is a neuron that responds exclusively to one digit category. Selectivities of individual neurons are indicated on **Figure 8A**; selectivities averaged over all neurons of a network, ζ, are indicated on **Figure 7E**. We can also quantify a neural network's selectivity for a specific digit class m as the sum of the selectivity of the neurons whose preferred stimulus class is m, ζ_m (see **Figure 8-B,C**). Training with DA statistically significantly boosts neural selectivity ($p < 0.001$).

DA induces large improvements in classification accuracy ($95.53 \pm 0.05\%$ for DA compared to $83.5 \pm 0.7\%$ for Hebbian learning, $p < 0.0001$), corresponding to a 72.7% reduction in the error rate. Performance for a class strongly correlates with neural selectivity for this class, for both the Hebbian and DA networks ($r = 0.996$ and $r = 0.920$, respectively, **Figures 8B,C**). These strong correlations suggest that enhanced neural selectivity explains the rise in correct responses following training with DA.

We can further visualize the outcome of DA learning by reducing the dimensionality of input images to 2 features (using t-SNE, Maaten and Hinton, 2008) and train the network on these data (**Figure 9**). In Hebbian learning, the neural network acts as a clustering algorithm and, as the learning mechanism is agnostic to the labels of the stimuli, the classification boundaries miss some aspects of the data classes. In particular, boundaries are poorly defined between close-by clusters such as "3," "5," and "8." Following DA signalling, weights adjust to match the boundaries for the conditions for reward delivery of the task.

In the model for DA activation presented above, reward predictions are binary, reflecting solely whether a decision is explorative or not. An alternative approach is to use the classifier's posterior for the output class (i.e., its classification confidence) as an approximation to the expected value of the predicted reward. This posterior probability strongly correlates with the empirically-measured reward probability ($r = 0.98$), validating the approximation. However, we find that this approach does not improve the network's accuracy over binary reward predictions (**Figure 5**, "DA post").

In order to assess the role of exploration in DA-based learning, we train a network without allowing explorative decision making. This greedy network achieves a classification score of $92.51 \pm 0.07\%$ (**Figure 5**, "DA greedy"), compared with $95.53 \pm 0.05\%$ with exploration. Exploration thus accounts for a further 18% relative drop in the error rate.

3.2.4. Learning on Non-uniformly Distributed Data

For the results on the MNIST dataset, ACh yields modest reductions in error rates relative to DA. This less important effect may be explained in part by the almost even distribution of training examples over the classes in the dataset. In more natural settings, some classes may contain many more examples than others while a high classification performance is equally important on all classes. For instance, a gatherer may see many more examples of "green leaves" than "berries" but still requires a low error rate for both classes. We test the

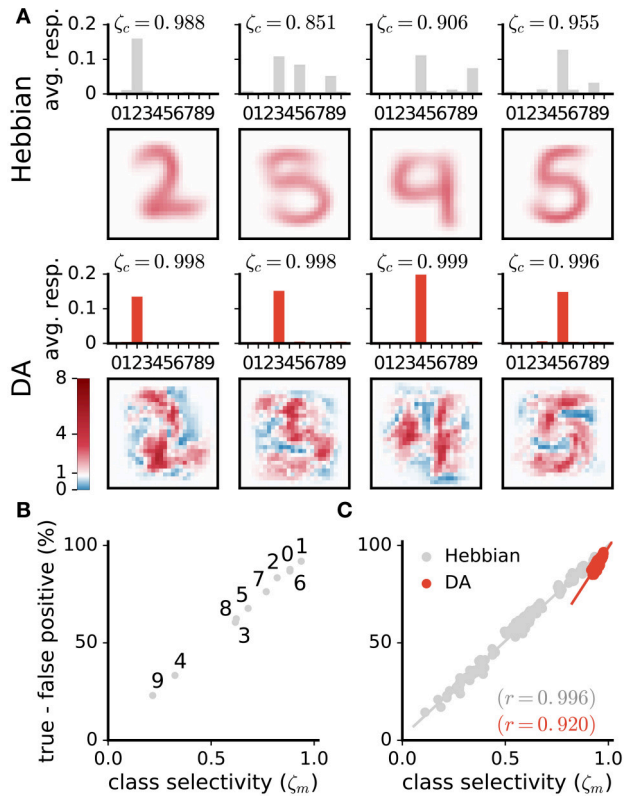


FIGURE 8 | DA enhances class selectivity in neurons. **(A)** 1st and 3rd rows: average responses of example neurons to the different integer classes. 2nd and 4th rows: depictions of the neurons' weights. The color axis represents weight strength. While neurons in the Hebbian network respond to stimuli of multiple classes, those trained with DA respond almost exclusively to a single class. This observation is quantified as the selectivity ζ_c of neurons (Equation 13). **(B,C)** Difference between rates of true and false positives for each digit class as a function of a network's selectivity for this class. **(B)** Data are for a single Hebbian neural network. **(C)** Data are for 20 networks for both Hebbian- and DA-based learning. DA enhances neural selectivity which translates to greater classification accuracy.

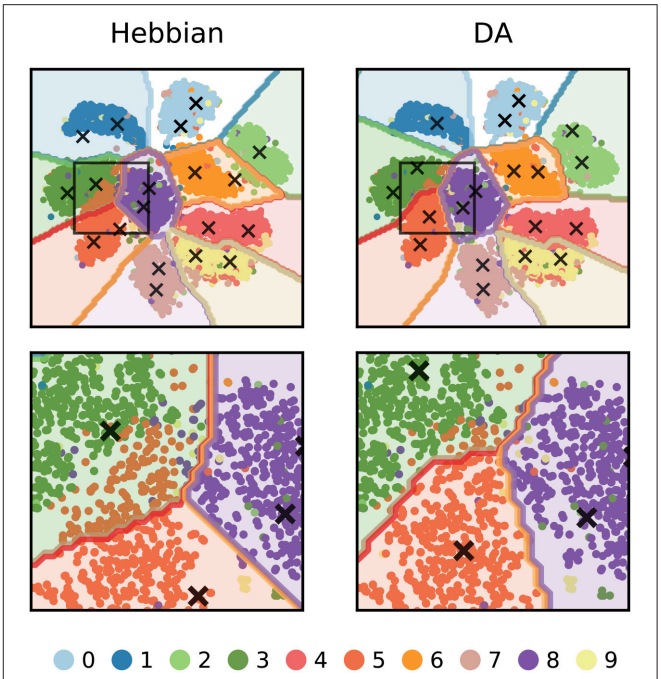


FIGURE 9 | 2-dimensional visualization of the outcome of DA learning. The dimensionality of input images are reduced from 784 to 2 features and a network is trained on these data. The input stimuli are depicted as colored dots, the weights of representation neurons as black crosses, and the classification boundaries as colored outlines. Hebbian learning performs density estimation: weights represent clusters of data points agnostic to the points' labels. For classes that are well separated from others, the network retrieves close to perfect boundaries (e.g., "1" or "0"). However, for close-by classes (e.g., "3", "5", and "8", magnified in the bottom row), the boundaries poorly match the true labels. DA transmission adapts weights so that they better agree with the class boundaries of the task.

impact of ACh in a modified version of MNIST in which a subset of the classes are over-represented. Here, the training dataset contains the classes "0," "2," "3," "5," and "8," and there are 60 times more "0" and "2" (the "leaves") than the other classes (the "berries"). To model equal importance of the classes, we take the test dataset to be uniformly distributed over the classes. For Hebbian learning, the network performs poorly on the under-represented classes as it dedicates only few neurons to these classes (**Figure 10**, top row). Neuromodulation significantly improves accuracy and, on these data, ACh yields gains comparable in size to those of DA. As with the standard MNIST dataset, ACh carries its effect by attributing more neurons to classes on which performance is low (those that are under-represented). DA only has minimal effects on the distribution of class preference; increases in performance derive from boosting neural selectivity.

In addition to training the network with ACh and DA separately, we combine the two neuromodulators by allowing first ACh release and then DA. This procedure leads to a redistribution of the class preferences (due to ACh) followed by an enhancement in neural selectivity (due to DA). The combined activations of ACh and DA result in a further decrease in error rates compared to either modulator alone, indicating that the effects of ACh and DA can successfully combine (**Figure 10**).

3.2.5. Impact of Code Sparseness

Lateral inhibition sparsifies the network's neural code so that inputs activate only one or a few neurons at a time (**Figure 11A**). Such a strong sparse code facilitates learning with neuromodulators as it avoids the credit-assignment problem. Additionally, the global neuromodulator signals are then essentially computed for a single neuron at a time. To examine the extent of the impact of the code's sparseness on learning, we introduce a temperature parameter τ to the softmax function determining the strength of the lateral competition:

$$s_c = \frac{\exp(I_c/\tau)}{\sum_{c'} \exp(I_{c'}/\tau)}. \tag{14}$$

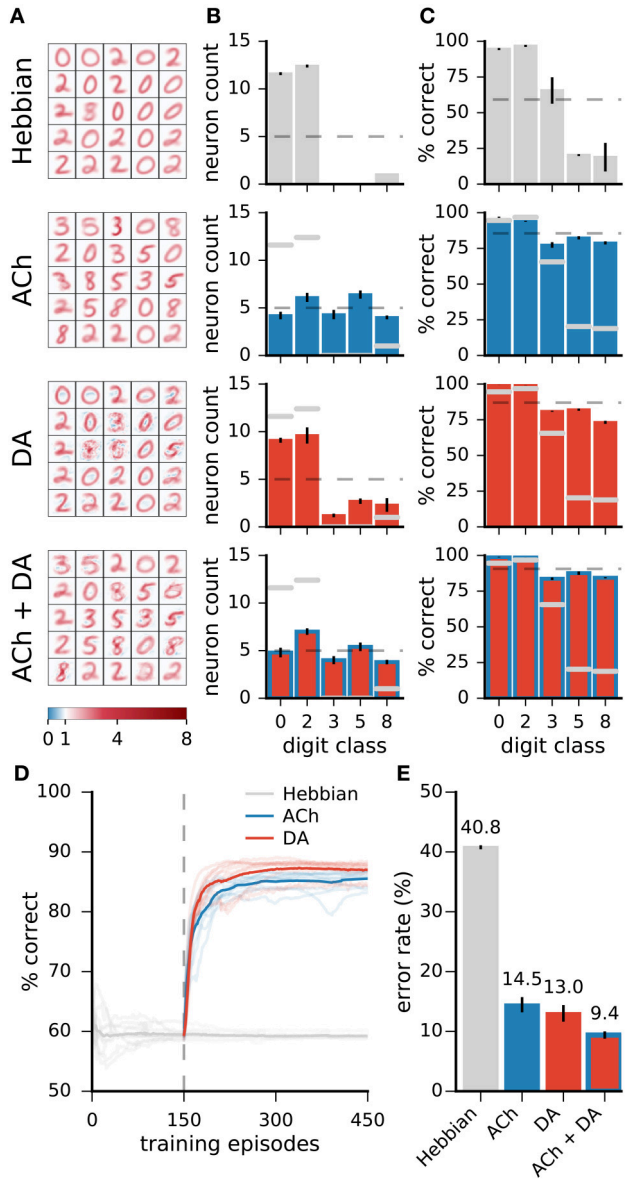


FIGURE 10 | On non-uniformly distributed data, ACh and DA yield gains in accuracy of similar magnitudes. **(A)** Weights of the networks. **(B)** Distribution of the neurons' preferred digit classes. Dashed line is a uniform distribution. **(C)** Rates of correct classification on the test dataset. Dashed line is the mean over classes. **(D)** Progression of the test error for Hebbian, ACh, and DA. Lighter traces are individual runs, darker traces are the mean of 10 runs. **(E)** Error rates of the different methods. Data are the mean of 10 runs, error bars indicate the standard deviation, gray overlaid bars in **(B,C)** are data for the Hebbian network for comparison. On non-uniformly distributed data, ACh and DA bear effects of comparable magnitudes. The refinements in weights brought by the two modulators can combine to bring further decrease in error rates.

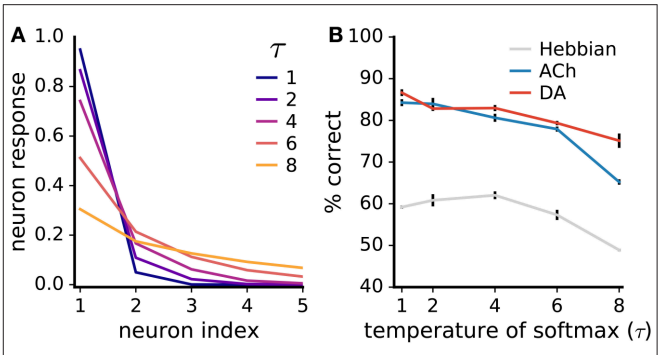


FIGURE 11 | Neuromodulator-based learning improves performance also for low code sparseness. **(A)** Impact of the temperature parameter τ of the softmax function (Equation 14) on the sparseness of the neural code. The neuron indices are ordered from highest to lowest neural responses; the five most active neurons are shown. **(B)** Rate of correct classification for different τ values. Performance is on the non-uniform MNIST dataset. Data are mean of 3 runs, error bars are the standard deviation. Although performances drop with weaker competition, neuromodulators boost accuracy even for low code sparseness.

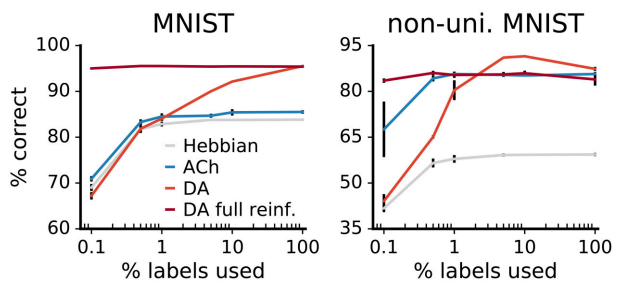


FIGURE 12 | Comparison of label reliance for ACh and DA. Network performance as a function of the percentage of training labels used. For "DA full reinforcement" we use the indicated portion of labels to train the classifier but use all labels to provide the reward feedback. The size of the gains deriving from ACh are not statistically different from each others for any label fractions ($p > 0.01$). Data are the mean of 3 runs, error bars indicate the standard deviation.

For $\tau \to 0^+$, the softmax function gives rise to a winner-take-all competition with a single active neuron; for $\tau \to \infty$, neural responses are uniformly distributed. We train networks with different τ values on the non-uniform MNIST dataset (we use the non-uniform dataset to better discern the effects on ACh-based learning). We find that the networks' performance drops as code sparseness decreases (**Figure 11B**). However, the neuromodulators give rise to large and statistically significant improvements even for low code sparseness, indicating that strong competition is not required for effective neuromodulator-based learning.

3.2.6. Impact of Label Availability

We examine the impact of label availability on learning by training networks with a varying fraction of labels, from 100% down to 0.1%. The accuracies of the networks decrease with label scarcity, both for learning with Hebb's rule and with neuromodulators (**Figure 12**). For the Hebbian network, labels only affect the classification layer; the decay in performance therefore derives exclusively from lower classifier accuracy.

For the neuromodulators, while label scarcity affects them both, the consequences are more substantial for DA. In particular,

when less than 1% of labels is used, the benefits of DA drop below those of ACh, this for both versions of the MNIST dataset. In error-based learning, labels are necessary to determine the correctness of an output. Reducing the ratio of labeled data consequently substantially hinders DA learning. On the other hand, the ACh signal yields gains in performance that are not statistically significantly different for all label fractions ($p > 0.01$). The constant improvements over declining label availability suggest that ACh learning relies effectively only minimally on labels, making ACh signaling beneficial even for scarcely labeled data.

DA-based learning does not require labels per se but only indications of whether outputs are right or wrong. We train an additional network using a fraction of the labels for the classifier but all labels for the reward feedback. The results show that performance remains high even for small label fractions, indicating that DA performs well in scenarios where true labels are in short supply but reinforcement feedback is available.

3.2.7. Performance Benchmark

In order to benchmark the functional performance of our algorithm, we compare it to MLPs trained with error back-propagation. We use the same architecture for our network and the MLPs (in this case, 784 input, 300 hidden, and 10 output neurons) and report the test error rate on the MNIST data. We train the MLPs using two state-of-the-art optimisation methods, the L-BFGS (Zhu et al., 1997) and Adam algorithms (Kingma and Ba, 2014) (see Appendix). In the original publication of benchmark results on the MNIST dataset, LeCun et al. (1998a) report a test error of 4.7% for an MLP of the architecture described above. Our biology-inspired algorithm yields a mean error rate of $2.88 \pm 0.05\%$, outperforming this original result. The MLPs with the L-BFGS and Adam optimisers yield an error rate of $2.15 \pm 0.04\%$ and $1.88 \pm 0.02\%$, respectively (**Figure 5**). In comparison, spiking neural networks intended for neuromorphic systems reach error rates of 5.0% (6,400 hidden spiking neurons, Diehl and Cook, 2015) and 4.4% (500 hidden spiking neurons, Neftci et al., 2015).

4. DISCUSSION

4.1. Learning Mechanisms

We study the effects of two modulatory signals on the representation and classification performance of a neural network. In our model, both signals act identically on synaptic plasticity but follow different release schedules, putatively those of ACh and DA. We find that these two signals give rise to distinct modifications in neural representations that both improve classification performance. Our model allows us to formulate hypotheses regarding the functional roles of ACh and DA in cortical representation learning. These roles can be explained as follows.

Consider the input $\vec{y}^{(n)}$ and the weights $\vec{W}_c$ as vectors in a high-dimensional space. The activation of a neuron s_c is computed as the dot product between an input and the weight vectors. Lateral inhibition introduces a soft winner-take-all competition resulting in a few neurons having strong responses and other neurons being silent. Hebbian learning then induces weight modifications $\Delta\vec{W}_c = \epsilon \cdot s_c(\vec{y} - \vec{W}_c)$ (Equation 5). We note that, for each weight, $\Delta\vec{W}_c$ points from the weight towards the current input. Both the variables *ACh* and *DA* modulate the magnitude of $\Delta\vec{W}_c$, $\|\Delta\vec{W}_c\|$ (Equations 5a and 5b).

Hebbian learning in the network performs density estimation: the distribution of the weights is determined by the density of data points in the input space. Modulating the learning rate of the network is similar to modifying data point density in that presenting a training image twice is comparable to presenting this image once but with a twice larger learning rate. For ACh-based learning, input images that are more challenging will trigger greater ACh activation, or have a larger learning rate. A cluster of data points associated with greater ACh activation is thus similar to having more data points in this cluster, inducing more neurons to represent the cluster. Or in other words, data points with $ACh > 1$ will have $\|\Delta\vec{W}_c\|$ of a greater magnitude, thereby exerting an increased "pull" on the weights.

For DA-based learning, the variable *DA* takes a value $\delta_{./.}$ specified by the current RPE scenario. According to the parameter search, for correct reward predictions ($+pred, +rew$ or $-pred, -rew$), the optimal $\delta_{+/+}$ and $\delta_{-/-}$ are of approximatively zero. In both cases, $\|\Delta\vec{W}_c\| \approx 0$; all the network's weights remain unchanged. When the network takes an exploitative decision that turns out to be wrong ($+pred, -rew$), the optimal $\delta_{+/-}$ is inferior to zero. The vector $\Delta\vec{W}_c$ is negated so that it points away from the current input (**Figure 13A**). Active neurons will have their weights move away from the current input and are then less likely to win the softmax competition at future presentations of this input. When the network takes an explorative decision that is surprisingly correct ($-pred, +rew$), the optimal $\delta_{-/+}$ is positive. The weights of active neurons move towards the input (**Figure 13B**). The explorative decision (expected incorrect) turned out to be right; this decision should be taken again on future presentation of the same stimulus. DA-based learning can be understood as reinforcement learning at the level of sensory representations.

These learning mechanisms are related to several known machine learning algorithms. In the purely Hebbian case, the network is akin to a Kohonen map (Kohonen, 1982) in that learning proceeds iteratively through neural competition and weight adaptation (without however the cooperation aspect which confers the topological organization to Kohonen maps). The ACh learning mechanism is reminiscent of boosting methods, for instance AdaBoost (Freund et al., 1999), which attribute greater weights to misclassified training examples. The DA learning mechanism is closely related to algorithms such as REINFORCE (Williams, 1992) which make use of a reinforcement signal acting on the learning rate of a neural network's weight update rule. It is interesting to note that, despite this close correspondence, the decision to model DA as a modulation of the network's learning rate was made not to match those rules but rather to mirror biology. Indeed, our model of DA (and ACh) emulates the observation that stimuli coinciding with release of the neuromodulators are over-represented in animal sensory cortices (**Figure 4**). The close similarity between our model of DA and REINFORCE's learning rule can thus

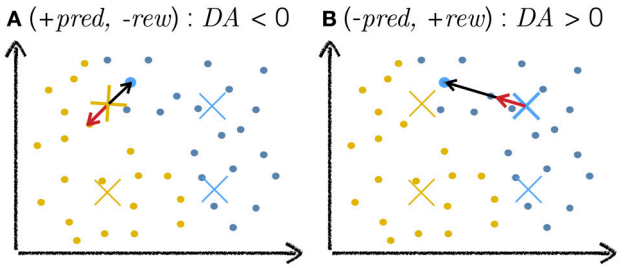


FIGURE 13 | Cartoon explanation of DA-based learning. The plots depict a toy example of a two-dimensional input space with dots as training examples and crosses as neuron weights, the colors of which indicate classes. The highlighted blue training example is the current input to the network. The black arrow depicts the weight change vector $\Delta\vec{W}$. The red arrow depicts the same vector $\Delta\vec{W}$ after modification by the *DA* variable. As a scalar multiplier, *DA* only affects the magnitude (and sign) of $\Delta\vec{W}$ and leaves its direction unchanged. **(A)** The network makes an incorrect exploitative decision: the blue input activates the yellow cross. The network expected a reward but none is delivered. In this case, $DA < 0$, negating $\Delta\vec{W}$ and moving the weight away from the training example (red arrow). **(B)** The networks makes a correct explorative decision: the exploitative scenario would have activated the yellow weight near the current input but noise injection in the activation of neurons led to another (blue) neuron being more active. This decision is surprisingly correct and the network is rewarded. In this case, the value of *DA* is positive, moving the weight towards the current input (red arrow).

be taken as further support for the biological realism of the latter.

4.2. Acetylcholine

Activation of the cholinergic system in mammals appears to follow attentional efforts. Sarter et al. (2006) review evidence suggesting that deteriorating performances, as indicated by a rise in error rates and a decline in reward rates, trigger effortful cognitive control to prevent erroneous behavior. Attentional efforts are paralleled by a heightened activation of cholinergic neurons in the basal forebrain (Himmelheber et al., 2000; Passetti et al., 2000; Dalley et al., 2001; Arnold et al., 2002; McGaughy et al., 2002; Kozak et al., 2006) which in turn broadcast this signal to the cortical mantle (Hasselmo and Sarter, 2011). For instance, engaging in a demanding motor (Conner et al., 2010) or tactile (Butt et al., 1997) task enhances ACh release in the motor and somatosensory cortices, respectively.

There is broad evidence that ACh acts as a permissive plasticity agent at its projection sites (Buchanan et al., 2010; Giessel and Sabatini, 2010), for instance promoting alterations of neural representations in sensory cortices (Greuel et al., 1988; Bröcher et al., 1992; Kilgard and Merzenich, 1998a,b; Ji et al., 2001; Ma and Suga, 2005; Suga, 2012; Chun et al., 2013). The scientific literature contains several hypotheses regarding the functional role of the modifications elicited by ACh. Froemke et al. (2007) suggest that shifts in neural tunings toward a stimulus paired with ACh activation serves as a long-term enhancement of attention to this stimulus. Others postulate that this modification stores the behavioral relevance of the stimulus (Kilgard and Merzenich, 1998a; Weinberger, 2003) or generally improves signal processing (Gu, 2003; Froemke et al., 2013). Here, we show that a signal modulating synaptic plasticity as a function of task difficulty improves the quality of a neural representation with respect to a classification task. The gains in performance result from assigning more neurons to challenging stimulus classes. Our model suggests that ACh serves this role in mammalian cortices.

Experimental evidence offer support for this hypothesis. For instance, motor skill acquisition and the accompanying enlargement of relevant representations in the motor cortex require ACh activation (Conner et al., 2003, 2010). Conversely, discrimination abilities rise for a tone whose representation is expanded as a result of repeated pairing with ACh activation (Reed et al., 2011). More generally, ACh antagonists or lesion of the cholinergic system impairs perceptual (Butt and Hodge, 1995; Fletcher and Wilson, 2002; Wilson et al., 2004; Leach et al., 2013) and motor skill learning (Conner et al., 2003). These results indicate that the cholinergic system is crucial for forms of learning involving modifications in sensory maps, especially those affecting the relative extent of cortical representations, as suggested in this work.

Our model of ACh is in line with a previous simulation study by Weinberger and Bakin (1998). The authors make use of a modified version of Hebb's rule and simulate the action of ACh as an amplification in the post-synaptic activation of target neurons. An *in vivo* micro-stimulation study validates this model. For the Hebbian rule used in this work, the two models of ACh are mathematically equivalent; this previous work thus offers support to the simulation employed here.

4.3. Dopamine

Dopaminergic neurons of the midbrain encode various features of rewards (Satoh et al., 2003; Tobler et al., 2005) and, in particular, strongly respond to the difference between predicted and received rewards (Schultz et al., 1997; Schultz, 2010). Midbrain neurons project to the entire cortex (Haber and Knutson, 2010) and the reward signals they carry modulate neural activity in most cortical areas (Vickery et al., 2011) including primary sensory cortices (Pleger et al., 2009; Brosch et al., 2011; Arsenault et al., 2013).

DA affects plasticity at the sites where it is released, as measured both at the level of synapses (Otani et al., 1998; Centonze et al., 1999; Blond et al., 2002; Bissière et al., 2003; Li et al., 2003; Sun et al., 2005; Matsuda et al., 2006; Calabresi et al., 2007; Navakkode et al., 2007) and behaviorally (Brembs et al., 2002; Wise, 2004; Graybiel, 2005; Kudoh and Shibuki, 2006; Klein et al., 2007; Luft and Schwarz, 2009; Molina-Luna et al., 2009; Hosp et al., 2011; Schicknick et al., 2012; Ott et al., 2014). In sensory cortices, DA efflux, triggered either by electric stimulation of the midbrain or by reward delivery, elicits plastic changes in the responses of primary sensory neurons (Bao et al., 2001, 2003; Beitel et al., 2003; Frankó et al., 2010; Poort et al., 2015).

The role of the plastic modifications induced by DA are usually understood in terms of reinforcement learning, for instance to learn the appetitive value of stimuli (Brembs et al., 2002; Wise, 2004; Frankó et al., 2010) or to learn reward-directed behaviors (Watkins and Dayan, 1992; Dayan and Balleine,

2002; Wise, 2004; Schicknick et al., 2012; Ott et al., 2014). In sensory representations, the changes brought forth by DA were previously hypothesized to enhance the saliency of stimuli predictive of rewards (Bao et al., 2001) and to adapt cortical representations to task requirements (Brosch et al., 2011).

Here, we show that a signal modulating plasticity as a function of RPEs adapts synaptic weights to the reward contingencies of a task, thereby improving performance on the task. Specifically, in our model, the responses of neurons become matched to the boundaries in conditions for reward delivery. In the digit classification task, this results in neurons being better tuned to the distinct digit classes, in this way improving classification performance. We suggest that, in mammals, dopamine carries this role of adapting sensory representations to the reward contingencies of a task.

After training monkeys on a visual discrimination task, neural responses become matched to the stimulus features that discriminate between the reward conditions of the task (Sigala and Logothetis, 2002). This process is comparable to the effect of DA in our model. We thus postulate that DA orchestrates these changes and predict that lesioning the dopaminergic system would prevent this form of learning. Animal experiments show that interfering with DA signaling impairs sensory discrimination learning (Kudoh and Shibuki, 2006; Schicknick et al., 2012), supporting this prediction.

The optimal values of the $\delta_{/.}$ constants we find through parameter exploration are in close qualitative agreement with the release properties of DA observed in primates (Schultz et al., 1997; Tobler et al., 2005) (**Figure 6**). Both in animals and in the present model, unpredicted rewards lead to a rise in dopaminergic activation while the absence of predicted rewards lead to a reduction in activation. Correctly predicted rewards leave dopaminergic activation essentially unchanged. The release values in the model were selected to maximize performance on a discrimination task. It is conceivable that the dopaminergic activation schedule in animals was similarly selected through evolutionary pressures to maximize perceptual abilities.

We tested the effect of explorative decision-making while training with DA and found that exploration yields an additional relative reduction of 18% in error rates. Studies show that human subjects actively engage in exploratory behavior when making decisions (Daw et al., 2006). Explorative decision-making is usually understood as a method to sample available choices with the prospect of discovering an option richer than the current optimum. Our model suggests that, in perceptual decision making, such explorative behavior may additionally serve the purpose of refining cortical sensory representations.

4.4. Comparing Acetylcholine and Dopamine

On the non-uniform dataset, ACh gives rise to improvements comparable in size to those of DA. This result highlights the relevance of ACh in scenarios where training examples are largely non-uniformly distributed over the classes, as is often the case in natural conditions. Furthermore, in contrast to DA, the ACh signal yields gains in accuracy of constant magnitude over decreasing label availability. This finding points to a particularly beneficial role for ACh when environmental feedback is scarce.

On the non-uniform dataset, the combined effects of the two neuromodulators are greater than either one separately. This result indicates that the weight modifications brought by ACh and DA are distinct and complimentary, and that they can successfully combine.

4.5. Functional Performances and Outlook

The learning mechanisms presented in this work yield error rates close to that of state-of-the-art optimisation methods used to train MLPs for comparable network architectures. Since evolutionary pressures must have favored well performing learning mechanisms in the brain, any candidate model of cortical learning must offer strong functional performances. Our model meets this criteria, making it a suitable model for learning in biological neural structures.

In line with recent studies of biologically-plausible learning (Keck et al., 2012; Nessler et al., 2013; Schmuker et al., 2014; Diehl and Cook, 2015; Neftci et al., 2015), we used correct classification as a measure of performance. This measure facilitates the study of the functional roles of neuromodulators and the comparison with previous work. Our neuromodulator-based learning method can be extended to tasks beyond classification, for instance by generaliz ing the softmax competition to k-winner-take-all (O'reilly, 2001) or soft-k-winner-take-all (Lücke, 2009) competition.

Even in the sole context of classification, however, our approach offers several interesting advantages. For instance, compared to the traditional approach of gradient descent on a classification error, neuromodulator-based learning requires a weaker supervision signal, making use of binary rewards instead of explicit labels. Additionally, our model learns even in the absence of environmental feedback through Hebbian learning. Finally, weight modifications are based on synaptically-local information and on two signals broadcasted identically to all neurons, which matches capabilities of biological neural networks.

On the functional side, learning with DA and ACh has been shown to decisively improve classification performance in our model system. Although it was not the main focus of this study, we note that very high classification performances even for relatively small networks (compare sizes in Diehl and Cook, 2015; Neftci et al., 2015) could be achieved using neuromodulation. The use of neuromodulation in spiking neural systems for neuromorphic chips (Diehl and Cook, 2015; Neftci et al., 2015) is therefore likely to result in performance gains. Similarly, neuromodulation is expected to further improve performance of novel hierarchical networks with Hebbian learning (Forster et al., 2016) which have a functional focus on learning from data with very few labels.

It is interesting to note that, since the initial publication of the MNIST dataset, advances in gradient-based learning resulted in continuous and substantial decreases in error rates. The biologically-inspired method presented in this work is at a relatively early stage and we may expect similar improvements from future research.

AUTHOR CONTRIBUTIONS

RH carried out the simulations, analyzed the data, and designed the study with contributions of JL and KO. JL provided theoretical background and support for the neural network model. JL and KO helped revising the manuscript. All authors read and approved the final version of the manuscript.

FUNDING

This work was supported by Quebec's National Fund for Research in Nature and Technology (181120), the German National Academic Foundation, the German Research Foundation (GRK 1589 and LU 1196/5-1), and the Cluster of Excellence EXC 1077/1 Hearing4all.

REFERENCES

Allard, T., Clark, S., Jenkins, W., and Merzenich, M. (1991). Reorganization of somatosensory area 3b representations in adult owl monkeys after digital syndactyly. *J. Neurophysiol.* 66, 1048–1058.

Arnold, H., Burk, J., Hodgson, E., Sarter, M., and Bruno, J. (2002). Differential cortical acetylcholine release in rats performing a sustained attention task versus behavioral control tasks that do not explicitly tax attention. *Neuroscience* 114, 451–460. doi: 10.1016/S0306-4522(02)00292-0

Arsenault, J. T., Nelissen, K., Jarraya, B., and Vanduffel, W. (2013). Dopaminergic reward signals selectively decrease fmri activity in primate visual cortex. *Neuron* 77, 1174–1186. doi: 10.1016/j.neuron.2013.01.008

Assisi, C., Stopfer, M., Laurent, G., and Bazhenov, M. (2007). Adaptive regulation of sparseness by feedforward inhibition. *Nat. Neurosci.* 10, 1176–1184. doi: 10.1038/nn1947

Bao, S., Chan, V. T., and Merzenich, M. M. (2001). Cortical remodelling induced by activity of ventral tegmental dopamine neurons. *Nature* 412, 79–83. doi: 10.1038/35083586

Bao, S., Chan, V. T., Zhang, L. I., and Merzenich, M. M. (2003). Suppression of cortical representation through backward conditioning. *Proc. Natl. Acad. Sci. U.S.A.* 100, 1405–1408. doi: 10.1073/pnas.0337527100

Bao, S., Chang, E. F., Woods, J., and Merzenich, M. M. (2004). Temporal plasticity in the primary auditory cortex induced by operant perceptual learning. *Nat. Neurosci.* 7, 974–981. doi: 10.1038/nn1293

Barkat, T. R., Polley, D. B., and Hensch, T. K. (2011). A critical period for auditory thalamocortical connectivity. *Nat. Neurosci.* 14, 1189–1194. doi: 10.1038/nn.2882

Beitel, R. E., Schreiner, C. E., Cheung, S. W., Wang, X., and Merzenich, M. M. (2003). Reward-dependent plasticity in the primary auditory cortex of adult monkeys trained to discriminate temporally modulated signals. *Proc. Natl. Acad. Sci. U.S.A.* 100, 11070–11075. doi: 10.1073/pnas.1334187100

Bissière, S., Humeau, Y., and Lüthi, A. (2003). Dopamine gates ltp induction in lateral amygdala by suppressing feedforward inhibition. *Nat. Neurosci.* 6, 587–592. doi: 10.1038/nn1058

Blond, O., Crépel, F., and Otani, S. (2002). Long-term potentiation in rat prefrontal slices facilitated by phased application of dopamine. *Eur. J. Pharmacol.* 438, 115–116. doi: 10.1016/S0014-2999(02)01291-8

Brembs, B., Lorenzetti, F. D., Reyes, F. D., Baxter, D. A., and Byrne, J. H. (2002). Operant reward learning in aplysia: neuronal correlates and mechanisms. *Science* 296, 1706–1709. doi: 10.1126/science.1069434

Bröcher, S., Artola, A., and Singer, W. (1992). Agonists of cholinergic and noradrenergic receptors facilitate synergistically the induction of long-term potentiation in slices of rat visual cortex. *Brain Res.* 573, 27–36. doi: 10.1016/0006-8993(92)90110-U

Brosch, M., Selezneva, E., and Scheich, H. (2011). Representation of reward feedback in primate auditory cortex. *Front. Syst. Neurosci.* 5:5. doi: 10.3389/fnsys.2011.00005

Buchanan, K. A., Petrovic, M. M., Chamberlain, S. E. L., Marrion, N. V., and Mellor, J. R. (2010). Facilitation of long-term potentiation by muscarinic m1 receptors is mediated by inhibition of sk channels. *Neuron* 68, 948–963. doi: 10.1016/j.neuron.2010.11.018

Butt, A. E., and Hodge, G. K. (1995). Acquisition, retention, and extinction of operant discriminations in rats with nucleus basalis magnocellularis lesions. *Behav. Neurosci.* 109:699. doi: 10.1037/0735-7044.109.4.699

Butt, A. E., Testylier, G., and Dykes, R. W. (1997). Acetylcholine release in rat frontal and somatosensory cortex is enhanced during tactile discrimination learning. *Psychobiology* 25, 18–33.

Calabresi, P., Picconi, B., Tozzi, A., and Di Filippo, M. (2007). Dopamine-mediated regulation of corticostriatal synaptic plasticity. *Trends Neurosci.* 30, 211–219. doi: 10.1016/j.tins.2007.03.001

Calford, M. B., and Tweedale, R. (1988). Immediate and chronic changes in responses of somatosensory cortex in adult flying-fox after digit amputation. *Nature* 332, 446–448. doi: 10.1038/332446a0

Centonze, D., Gubellini, P., Picconi, B., Calabresi, P., Giacomini, P., and Bernardi, G. (1999). Unilateral dopamine denervation blocks corticostriatal ltp. *J. Neurophysiol.* 82, 3575–3579.

Chun, S., Bayazitov, I. T., Blundon, J. A., and Zakharenko, S. S. (2013). Thalamocortical long-term potentiation becomes gated after the early critical period in the auditory cortex. *J. Neurosci.* 33, 7345–7357. doi: 10.1523/JNEUROSCI.4500-12.2013

Conner, J., Kulczycki, M., and Tuszynski, M. (2010). Unique contributions of distinct cholinergic projections to motor cortical plasticity and learning. *Cereb. Cortex* 20, 2739–2748. doi: 10.1093/cercor/bhq022

Conner, J. M., Culberson, A., Packowski, C., Chiba, A. A., and Tuszynski, M. H. (2003). Lesions of the basal forebrain cholinergic system impair task acquisition and abolish cortical plasticity associated with motor skill learning. *Neuron* 38, 819–829. doi: 10.1016/S0896-6273(03)00288-5

Dalley, J. W., McGaughy, J., O'Connell, M. T., Cardinal, R. N., Levita, L., and Robbins, T. W. (2001). Distinct changes in cortical acetylcholine and noradrenaline efflux during contingent and noncontingent performance of a visual attentional task. *J. Neurosci.* 21, 4908–4914.

Daw, N. D., O'Doherty, J. P., Dayan, P., Seymour, B., and Dolan, R. J. (2006). Cortical substrates for exploratory decisions in humans. *Nature* 441, 876–879. doi: 10.1038/nature04766

Dayan, P., and Balleine, B. W. (2002). Reward, motivation, and reinforcement learning. *Neuron* 36, 285–298. doi: 10.1016/S0896-6273(02)00963-7

de Villers-Sidani, E., Chang, E. F., Bao, S., and Merzenich, M. M. (2007). Critical period window for spectral tuning defined in the primary auditory cortex (a1) in the rat. *J. Neurosci.* 27, 180–189. doi: 10.1523/JNEUROSCI.3227-06.2007

Diehl, P. U., and Cook, M. (2015). Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Front. Comput. Neurosci.* 9:99. doi: 10.3389/fncom.2015.00099

Dinse, H. R., Ragert, P., Pleger, B., Schwenkreis, P., and Tegenthoff, M. (2003). Pharmacological modulation of perceptual learning and associated cortical reorganization. *Sci. Signal.* 301, 91. doi: 10.1126/science.1085423

Eggermont, J. J., and Roberts, L. E. (2004). The neuroscience of tinnitus. *Trends Neurosci.* 27, 676–682. doi: 10.1016/j.tins.2004.08.010

Elbert, T., Pantev, C., Wienbruch, C., Rockstroh, B., and Taub, E. (1995). Increased cortical representation of the fingers of the left hand in string players. *Science* 270, 305.

Fletcher, M. L., and Wilson, D. A. (2002). Experience modifies olfactory acuity: acetylcholine-dependent learning decreases behavioral generalization between similar odorants. *J. Neurosci.* 22:RC201.

Flor, H., Nikolajsen, L., and Jensen, T. S. (2006). Phantom limb pain: a case of maladaptive cns plasticity? *Nat. Rev. Neurosci.* 7, 873–881. doi: 10.1038/nrn1991

Forster, D., and Lücke, J. (2017). Truncated variational em for semi-supervised neural simpletrons. *arXiv preprint arXiv:1702.01997*.

Forster, D., Sheikh, A.-S., and Lücke, J. (2016). Neural simpletrons–minimalistic directed generative networks for learning with few labels. *Stat* 1050:23.

Frankó, E., Seitz, A. R., and Vogels, R. (2010). Dissociable neural effects of long-term stimulus–reward pairing in macaque visual cortex. *J. Cogn. Neurosc.* 22, 1425–1439. doi: 10.1162/jocn.2009.21288

Freund, Y., Schapire, R., and Abe, N. (1999). A short introduction to boosting. *J. Jpn. Soc. Artif. Intell.* 14, 1612.

Fritz, J., Shamma, S., Elhilali, M., and Klein, D. (2003). Rapid task-related plasticity of spectrotemporal receptive fields in primary auditory cortex. *Nat. Neurosci.* 6, 1216–1223. doi: 10.1038/nn1141

Froemke, R. C., Carcea, I., Barker, A. J., Yuan, K., Seybold, B. A., Martins, A. R. O., et al. (2013). Long-term modification of cortical synapses improves sensory perception. *Nat. Neurosci.* 16, 79–88. doi: 10.1038/nn.3274

Froemke, R. C., Merzenich, M. M., and Schreiner, C. E. (2007). A synaptic memory trace for cortical receptive field plasticity. *Nature* 450, 425–429. doi: 10.1038/nature06289

Gambino, F., and Holtmaat, A. (2012). Spike-timing-dependent potentiation of sensory surround in the somatosensory cortex is facilitated by deprivation-mediated disinhibition. *Neuron* 75, 490–502. doi: 10.1016/j.neuron.2012.05.020

Giessel, A. J., and Sabatini, B. L. (2010). M1 muscarinic receptors boost synaptic potentials and calcium influx in dendritic spines by inhibiting postsynaptic sk channels. *Neuron* 68, 936–947. doi: 10.1016/j.neuron.2010.09.004

Godde, B., Leonhardt, R., Cords, S. M., and Dinse, H. R. (2002). Plasticity of orientation preference maps in the visual cortex of adult cats. *Proc. Natl. Acad. Sci. U.S.A.* 99, 6352–6357. doi: 10.1073/pnas.082407499

Graybiel, A. M. (2005). The basal ganglia: learning new tricks and loving it. *Curr. Opin. Neurobiol.* 15, 638–644. doi: 10.1016/j.conb.2005.10.006

Greuel, J. M., Luhmann, H. J., and Singer, W. (1988). Pharmacological induction of use-dependent receptive field modifications in the visual cortex. *Science* 242, 74–77. doi: 10.1126/science.2902687

Gu, Q. (2003). Contribution of acetylcholine to visual cortex plasticity. *Neurobiol. Learn. Mem.* 80, 291–301. doi: 10.1016/S1074-7427(03)00073-X

Haber, S. N., and Knutson, B. (2010). The reward circuit: linking primate anatomy and human imaging. *Neuropsychopharmacology* 35, 4–26. doi: 10.1038/npp.2009.129

Halligan, P. W., Marshall, J. C., Wade, D. T., Davey, J., and Morrison, D. (1993). Thumb in cheek? sensory reorganization and perceptual plasticity after limb amputation. *Neuroreport* 4, 233–236. doi: 10.1097/00001756-199303000-00001

Han, Y. K., Köver, H., Insanally, M. N., Semerdjian, J. H., and Bao, S. (2007). Early experience impairs perceptual discrimination. *Nat. Neurosci.* 10, 1191–1197. doi: 10.1038/nn1941

Harris, J. A., Harris, I. M., and Diamond, M. E. (2001). The topography of tactile learning in humans. *J. Neurosci.* 21, 1056–1061.

Hasselmo, M. E., and Sarter, M. (2011). Modes and models of forebrain cholinergic neuromodulation of cognition. *Neuropsychopharmacology* 36, 52–73. doi: 10.1038/npp.2010.104

Himmelheber, A. M., Sarter, M., and Bruno, J. P. (2000). Increases in cortical acetylcholine release during sustained attention performance in rats. *Cogn. Brain Res.* 9, 313–325. doi: 10.1016/S0926-6410(00)00012-4

Hosp, J. A., Pekanovic, A., Rioult-Pedotti, M. S., and Luft, A. R. (2011). Dopaminergic projections from midbrain to primary motor cortex mediate motor skill learning. *J. Neurosci.* 31, 2481–2487. doi: 10.1523/JNEUROSCI.5411-10.2011

Isaacson, J. S., and Scanziani, M. (2011). How inhibition shapes cortical activity. *Neuron* 72, 231–243. doi: 10.1016/j.neuron.2011.09.027

Ji, W., Gao, E., and Suga, N. (2001). Effects of acetylcholine and atropine on plasticity of central auditory neurons caused by conditioning in bats. *J. Neurophysiol.* 86, 211–225.

Keck, C., Savin, C., and Lücke, J. (2012). Feedforward inhibition and synaptic scaling–two sides of the same coin? *PLoS Comput. Biol.* 8:e1002432. doi: 10.1371/journal.pcbi.1002432

Kilgard, M. P., and Merzenich, M. M. (1998a). Cortical map reorganization enabled by nucleus basalis activity. *Science* 279, 1714–1718. doi: 10.1126/science.279.5357.1714

Kilgard, M. P., and Merzenich, M. M. (1998b). Plasticity of temporal information processing in the primary auditory cortex. *Nat. Neurosci.* 1, 727–731. doi: 10.1038/3729

Kim, H., and Bao, S. (2009). Selective increase in representations of sounds repeated at an ethological rate. *J. Neurosci.* 29, 5163–5169. doi: 10.1523/JNEUROSCI.0365-09.2009

Kingma, D., and Ba, J. (2014). Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, abs/1412.6980.

Klein, T. A., Neumann, J., Reuter, M., Hennig, J., von Cramon, D. Y., and Ullsperger, M. (2007). Genetically determined differences in learning from errors. *Science* 318, 1642–1645. doi: 10.1126/science.1145044

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biol. Cybern.* 43, 59–69. doi: 10.1007/BF00337288

Kozak, R., Bruno, J. P., and Sarter, M. (2006). Augmented prefrontal acetylcholine release during challenged attentional performance. *Cereb. Cortex* 16, 9–17. doi: 10.1093/cercor/bhi079

Kudoh, M., and Shibuki, K. (2006). Sound sequence discrimination learning motivated by reward requires dopaminergic d2 receptor activation in the rat auditory cortex. *Learn. Mem.* 13, 690–698. doi: 10.1101/lm.390506

Law, C.-T., and Gold, J. I. (2009). Reinforcement learning can account for associative and perceptual learning on a visual decision task. *Nat. Neurosci.* 12:655. doi: 10.1038/nn.2304

Leach, N. D., Nodal, F. R., Cordery, P. M., King, A. J., and Bajo, V. M. (2013). Cortical cholinergic input is required for normal auditory perception and experience-dependent plasticity in adult ferrets. *J. Neurosci.* 33, 6659–6671. doi: 10.1523/JNEUROSCI.5039-12.2013

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998a). Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 2278–2324. doi: 10.1109/5.726791

LeCun, Y., Cortes, C., and Burges, C. J. (1998b). *The Mnist Database of Handwritten Digits.*

Li, G., and Cleland, T. A. (2013). A two-layer biophysical model of cholinergic neuromodulation in olfactory bulb. *J. Neurosci.* 33, 3037–3058. doi: 10.1523/JNEUROSCI.2831-12.2013

Li, S., Cullen, W. K., Anwyl, R., and Rowan, M. J. (2003). Dopamine-dependent facilitation of ltp induction in hippocampal ca1 by exposure to spatial novelty. *Nat. Neurosci.* 6, 526–531. doi: 10.1038/nn1049

Liu, J., Lu, Z.-L., and Dosher, B. A. (2010). Augmented hebbian reweighting: interactions between feedback and training accuracy in perceptual learning. *J. Vis.* 10:29. doi: 10.1167/10.10.29

Lücke, J. (2009). Receptive field self-organization in a model of the fine-structure in V1 cortical columns. *Neural Comput.* 21, 2805–2845. doi: 10.1162/neco.2009.07-07-584

Luft, A. R., and Schwarz, S. (2009). Dopaminergic signals in primary motor cortex. *Int. J. Dev. Neurosc.* 27, 415–421. doi: 10.1016/j.ijdevneu.2009.05.004

Ma, X., and Suga, N. (2005). Long-term cortical plasticity evoked by electric stimulation and acetylcholine applied to the auditory cortex. *Proc. Natl. Acad. Sci. U.S.A.* 102, 9335–9340. doi: 10.1073/pnas.0503851102

Maaten, L. V. D., and Hinton, G. (2008). Visualizing data using t-sne. *J. Mach. Learn. Res.* 9, 2579–2605.

Mante, V., Frazor, R. A., Bonin, V., Geisler, W. S., and Carandini, M. (2005). Independence of luminance and contrast in natural scenes and in the early visual system. *Nat. Neurosci.* 8, 1690–1697. doi: 10.1038/nn1556

Matsuda, Y., Marzo, A., and Otani, S. (2006). The presence of background dopamine signal converts long-term synaptic depression to potentiation in rat prefrontal cortex. *J. Neurosc.* 26, 4803–4810. doi: 10.1523/JNEUROSCI.5312-05.2006

McGaughy, J., Dalley, J., Morrison, C., Everitt, B., and Robbins, T. (2002). Selective behavioral and neurochemical effects of cholinergic lesions produced by intrabasalis infusions of 192 igg-saporin on attentional performance in a five-choice serial reaction time task. *J. Neurosci.* 22, 1905–1913.

Mittmann, W., Koch, U., and Häusser, M. (2005). Feed-forward inhibition shapes the spike output of cerebellar purkinje cells. *J. physiol.* 563, 369–378. doi: 10.1113/jphysiol.2004.075028

Molina-Luna, K., Pekanovic, A., Rohrich, S., Hertler, B., Schubring-Giese, M., Rioult-Pedotti, M.-S., et al. (2009). Dopamine in motor cortex is necessary for skill learning and synaptic plasticity. *PLoS ONE* 4:e7082. doi: 10.1371/journal.pone.0007082

Navakkode, S., Sajikumar, S., and Frey, J. U. (2007). Synergistic requirements for the induction of dopaminergic d1/d5-receptor-mediated ltp in

hippocampal slices of rat ca1 in vitro. *Neuropharmacology* 52, 1547–1554. doi: 10.1016/j.neuropharm.2007.02.010

Neftci, E. O., Pedroni, B. U., Joshi, S., Al-Shedivat, M., and Cauwenberghs, G. (2015). Unsupervised learning in synaptic sampling machines. *arXiv preprint arXiv:1511.04484*.

Nessler, B., Pfeiffer, M., Buesing, L., and Maass, W. (2013). Bayesian computation emerges in generic cortical microcircuits through spike-timing-dependent plasticity. *PLoS Comput. Biol.* 9:e1003037. doi: 10.1371/journal.pcbi.1003037

Olsen, S. R., and Wilson, R. I. (2008). Lateral presynaptic inhibition mediates gain control in an olfactory circuit. *Nature* 452, 956–960. doi: 10.1038/nature06864

O'reilly, R. C. (2001). Generalization in interactive networks: The benefits of inhibitory competition and hebbian learning. *Neural Comput.* 13, 1199–1241. doi: 10.1162/08997660152002834

Otani, S., Blond, O., Desce, J.-M., and Crepel, F. (1998). Dopamine facilitates long-term depression of glutamatergic transmission in rat prefrontal cortex. *Neuroscience* 85, 669–676. doi: 10.1016/S0306-4522(97)00677-5

Ott, T., Jacob, S. N., and Nieder, A. (2014). Dopamine receptors differentially enhance rule coding in primate prefrontal cortex neurons. *Neuron* 84, 1317–1328. doi: 10.1016/j.neuron.2014.11.012

Pantev, C., Oostenveld, R., Engelien, A., Ross, B., Roberts, L. E., and Hoke, M. (1998). Increased auditory cortical representation in musicians. *Nature* 392, 811–814. doi: 10.1038/33918

Pantev, C., Roberts, L. E., Schulz, M., Engelien, A., and Ross, B. (2001). Timbre-specific enhancement of auditory cortical representations in musicians. *Neuroreport* 12, 169–174. doi: 10.1097/00001756-200101220-00041

Parikh, V., Kozak, R., Martinez, V., and Sarter, M. (2007). Prefrontal acetylcholine release controls cue detection on multiple timescales. *Neuron* 56, 141–154. doi: 10.1016/j.neuron.2007.08.025

Passetti, F., Dalley, J., O'connell, M., Everitt, B., and Robbins, T. (2000). Increased acetylcholine release in the rat medial prefrontal cortex during performance of a visual attentional task. *Eur. J. Neurosci.* 12, 3051–3058. doi: 10.1046/j.1460-9568.2000.00183.x

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: machine learning in python. *J. Mach. Learn. Res.* 12, 2825–2830.

Pleger, B., Ruff, C. C., Blankenburg, F., Klöppel, S., Driver, J., and Dolan, R. J. (2009). Influence of dopaminergically mediated reward on somatosensory decision-making. *PLoS Biol.* 7:e1000164. doi: 10.1371/journal.pbio.1000164

Polley, D. B., Steinberg, E. E., and Merzenich, M. M. (2006). Perceptual learning directs auditory cortical map reorganization through top-down influences. *J. Neurosci.* 26, 4970–4982. doi: 10.1523/JNEUROSCI.3771-05.2006

Poort, J., Khan, A. G., Pachitariu, M., Nemri, A., Orsolic, I., Krupic, J., et al. (2015). Learning enhances sensory and multiple non-sensory representations in primary visual cortex. *Neuron* 86, 1478–1490. doi: 10.1016/j.neuron.2015.05.037

Pouille, F., Marin-Burgin, A., Adesnik, H., Atallah, B. V., and Scanziani, M. (2009). Input normalization by global feedforward inhibition expands cortical dynamic range. *Nat. Neurosci.* 12, 1577–1585. doi: 10.1038/nn.2441

Pouille, F., and Scanziani, M. (2001). Enforcement of temporal fidelity in pyramidal cells by somatic feed-forward inhibition. *Science* 293, 1159–1163. doi: 10.1126/science.1060342

Ramachandran, V. S., Stewart, M., and Rogers-Ramachandran, D. (1992). Perceptual correlates of massive cortical reorganization. *Neuroreport* 3, 583–586. doi: 10.1097/00001756-199207000-00009

Recanzone, G. A., Schreiner, C., and Merzenich, M. M. (1993). Plasticity in the frequency representation of primary auditory cortex following discrimination training in adult owl monkeys. *J. Neurosci.* 13, 87–103.

Recanzone, G. H., Merzenich, M. M., Jenkins, W. M., Grajski, K. A., and Dinse, H. R. (1992). Topographic reorganization of the hand representation in cortical area 3b owl monkeys trained in a frequency-discrimination task. *J. Neurophysiol.* 67, 1031–1056.

Reed, A., Riley, J., Carraway, R., Carrasco, A., Perez, C., Jakkamsetti, V., et al. (2011). Cortical map plasticity improves learning but is not necessary for improved performance. *Neuron* 70, 121–131. doi: 10.1016/j.neuron.2011.02.038

Roelfsema, P. R., and Ooyen, A. v. (2005). Attention-gated reinforcement learning of internal representations for classification. *Neural Comput.* 17, 2176–2214. doi: 10.1162/0899766054615699

Roelfsema, P. R., van Ooyen, A., and Watanabe, T. (2010). Perceptual learning rules based on reinforcers and attention. *Trends Cogn. Sci.* 14:64. doi: 10.1016/j.tics.2009.11.005

Rombouts, J., Roelfsema, P., and Bohte, S. M. (2012). "Neurally plausible reinforcement learning of working memory tasks," in *Advances in Neural Information Processing Systems*, eds F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Lake Tahoe, NV: NIPS), 1871–1879.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). *Learning Internal Representations by Error Propagation.* Technical Report, DTIC Document.

Sarter, M., Gehring, W. J., and Kozak, R. (2006). More attention must be paid: the neurobiology of attentional effort. *Brain Res. Rev.* 51, 145–160. doi: 10.1016/j.brainresrev.2005.11.002

Satoh, T., Nakai, S., Sato, T., and Kimura, M. (2003). Correlated coding of motivation and outcome of decision by dopamine neurons. *J. Neurosci.* 23, 9913–9923.

Schicknick, H., Reichenbach, N., Smalla, K.-H., Scheich, H., Gundelfinger, E. D., and Tischmeyer, W. (2012). Dopamine modulates memory consolidation of discrimination learning in the auditory cortex. *Eur. J. Neurosci.* 35, 763–774. doi: 10.1111/j.1460-9568.2012.07994.x

Schmuker, M., Pfeil, T., and Nawrot, M. P. (2014). A neuromorphic network for generic multivariate data classification. *Proc. Natl. Acad. Sci. U.S.A.* 111, 2081–2086. doi: 10.1073/pnas.1303053111

Schoups, A., Vogels, R., Qian, N., and Orban, G. (2001). Practising orientation identification improves orientation coding in v1 neurons. *Nature* 412, 549–553. doi: 10.1038/35087601

Schultz, W. (2007). Behavioral dopamine signals. *Trends Neurosci.* 30, 203–210. doi: 10.1016/j.tins.2007.03.007

Schultz, W. (2010). Review dopamine signals for reward value and risk: basic and recent data. *Behav. Brain Funct.* 6:24. doi: 10.1186/1744-9081-6-24

Schultz, W., Dayan, P., and Montague, P. R. (1997). A neural substrate of prediction and reward. *Science* 275, 1593–1599. doi: 10.1126/science.275.5306.1593

Sclar, G., Maunsell, J. H., and Lennie, P. (1990). Coding of image contrast in central visual pathways of the macaque monkey. *Vis. Res.* 30, 1–10. doi: 10.1016/0042-6989(90)90123-3

Sengpiel, F., Stawinski, P., and Bonhoeffer, T. (1999). Influence of experience on orientation maps in cat visual cortex. *Nat. Neurosci.* 2, 727–732. doi: 10.1038/11192

Sigala, N., and Logothetis, N. K. (2002). Visual categorization shapes feature selectivity in the primate temporal cortex. *Nature* 415, 318–320. doi: 10.1038/415318a

Stopfer, M., Jayaraman, V., and Laurent, G. (2003). Intensity versus identity coding in an olfactory system. *Neuron* 39, 991–1004. doi: 10.1016/j.neuron.2003.08.011

Suga, N. (2012). Tuning shifts of the auditory system by corticocortical and corticofugal projections and conditioning. *Neurosci. Biobehav. Rev.* 36, 969–988. doi: 10.1016/j.neubiorev.2011.11.006

Sun, X., Zhao, Y., and Wolf, M. E. (2005). Dopamine receptor stimulation modulates ampa receptor synaptic insertion in prefrontal cortex neurons. *J. Neurosc.* 25, 7342–7351. doi: 10.1523/JNEUROSCI.4603-04.2005

Sutton, R. S., and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*, Vol. 135. Cambridge, MA: MIT press Cambridge.

Swadlow, H. A. (2003). Fast-spike interneurons and feedforward inhibition in awake sensory neocortex. *Cereb. Cortex* 13, 25–32. doi: 10.1093/cercor/13.1.25

Tegenthoff, M., Ragert, P., Pleger, B., Schwenkreis, P., Förster, A.-F., Nicolas, V., et al. (2005). Improvement of tactile discrimination performance and enlargement of cortical somatosensory maps after 5 hz rtms. *PLoS Biol.* 3:e362. doi: 10.1371/journal.pbio.0030362

Tobler, P. N., Fiorillo, C. D., and Schultz, W. (2005). Adaptive coding of reward value by dopamine neurons. *Science* 307, 1642–1645. doi: 10.1126/science.1105370

Vickery, T. J., Chun, M. M., and Lee, D. (2011). Ubiquity and specificity of reinforcement signals throughout the human brain. *Neuron* 72, 166–177. doi: 10.1016/j.neuron.2011.08.011

Wang, Y., Sereno, J. A., Jongman, A., and Hirsch, J. (2003). fMRI evidence for cortical modification during learning of mandarin lexical tone. *J. Cogn. Neurosci.* 15, 1019–1027. doi: 10.1162/089892903770007407

Watkins, C. J., and Dayan, P. (1992). Q-learning. *Mach. Learn.* 8, 279–292. doi: 10.1007/BF00992698

Wehr, M., and Zador, A. M. (2005). Synaptic mechanisms of forward suppression in rat auditory cortex. *Neuron* 47, 437–445. doi: 10.1016/j.neuron.2005.06.009

Weinberger, N. M. (2003). The nucleus basalis and memory codes: auditory cortical plasticity and the induction of specific, associative behavioral memory. *Neurobiol. Learn. Mem.* 80, 268–284. doi: 10.1016/S1074-7427(03)00072-8

Weinberger, N. M., and Bakin, J. S. (1998). Learning-induced physiological memory in adult primary auditory cortex: receptive field plasticity, model, and mechanisms. *Audiol. Neurotol.* 3, 145–167. doi: 10.1159/000013787

Werbos, P. (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral sciences*. PhD Thesis, Harvard University.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.* 8, 229–256. doi: 10.1007/BF00992696

Wilson, D. A., Fletcher, M. L., and Sullivan, R. M. (2004). Acetylcholine and olfactory perceptual learning. *Learn. Mem.* 11, 28–34. doi: 10.1101/lm.66404

Wise, R. A. (2004). Dopamine, learning and motivation. *Nat. Rev. Neurosci.* 5, 483–494. doi: 10.1038/nrn1406

Yang, T., and Maunsell, J. H. (2004). The effect of perceptual learning on neuronal responses in monkey visual area v4. *J. Neurosci.* 24, 1617–1626. doi: 10.1523/JNEUROSCI.4442-03.2004

Zhu, C., Byrd, R. H., Lu, P., and Nocedal, J. (1997). Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transact. Math. Soft.* 23, 550–560. doi: 10.1145/279232.279236

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

A. APPENDIX

A.1. Code and data

The model was written in the Python programming language and was run on a computer cluster. The code for the neural network is available at https://github.com/raphaelholca/hebbianRL. The original MNIST dataset is available at http://yann.lecun.com/exdb/mnist/. The dataset is randomly split into training and testing sets; the network's performance is reported on the testing images not seen during training. The network with 300 hidden units used for performance comparison with other work was trained with the full (unbalanced) dataset. For all other results, the datasets were balanced so that they contain the same number of examples for each digit class. This balancing has negligible effects on the results.

A.2. Weight initialisation

We pre-compute the activation of input neurons $\vec{y}$ through Equation 1 for the whole training dataset. Learning proceeds through full iterations over the dataset during which $\vec{y}$ are presented in a random order to the network. Weights of representation neurons are initialised using the statistics of the input images. Specifically, we initialise the weights with the mean activation of input neurons taken over the whole dataset, with the addition of noise to break symmetry:

$$W_{cd} = \mu(y_d) - \sigma^2(y_d) \cdot \eta_{\text{init}}\,, \tag{A1}$$

where $\mu(\cdot)$ and $\sigma^2(\cdot)$ are the mean and variance taken over all training images N, respectively, and η_{init} is noise drawn from a uniform distribution in the interval [0,0, 2.0). Activations propagate through the network as a succession of Equations 2, 4, 5. Values for all hyper-parameters were found through grid search (see **Table A1**).

A.3. Batch learning

To speed up computation, we train the network using mini-batches; weight updates are computed over batches of 50 training examples. Using mini-batches only negligibly affects representation learning and the network's performance.

In the case of DA-based learning, negative learning rates (for absent expected rewards, $+pred\ -rew$) could potentially result in negative weights. For biological realism and computational stability, we prevent this by excluding weight updates for a representation neuron c if any weight W_{cd} would become negative after the weight update. For the parameter set presented in **Table A1** in Supplementary Methods, this rule only rarely prevents learning ($\sim$ 0.1% of all batch updates). However, when performing parameter exploration of the $\delta_{./.}$ variables, some parameter sets lead to rapid decay to negative weight values, and this rule is then necessary to ensure computational stability.

TABLE A1 | Hyper-parameters used in training the network. Values were determined through parameter exploration.

Parameter	Description	Value
α	Amplitude of the sigmoid function for ACh release	2.0
β	Slope of the sigmoid function for ACh release	20.0
$\delta_{+/+}$	DA activation for correctly predicted reward	0.01
$\delta_{+/-}$	DA activation for incorrectly predicted reward	−1.0
$\delta_{-/+}$	DA activation for unexpected reward	4.0
$\delta_{-/-}$	DA activation for correctly predicted absence of reward	−0.25
A	Normalization constant for feedforward inhibition	1.0×10^{3}
ϵ	Learning rate	5.0×10^{-3}
υ	Variance of the normal distribution of noise η	0.3

TABLE A2 | Hyper-parameters for the benchmarking algorithms, as implemented in the *Scikit-learn* module.

Parameter	Description	Value
hidden_layer_sizes	Number of neurons in the hidden layer	300
activation	Activation function for the hidden layer	'relu'
algorithm	Algorithm for weight optimisation	'adam' or 'l-bfgs'
alpha	Regularisation term (L2 penalty)	1e-06
batch_size	Size of mini-batches for stochastic optimisation	200
learning_rate_init	Initial learning rate (Adam)	0.001
beta_1	Exp. decay for estimates of 1st moment (Adam)	0.8
beta_2	Exp. decay for estimates of 2nd moment (Adam)	0.9
epsilon	Value for numerical stability (Adam)	1e-08

A.4. Comparison benchmarks

The MLP algorithm was obtained from the *Scikit-learn* module Pedregosa et al., 2011) (version 18.dev0, downloaded on 04/29/16). We used 3-fold cross-validation and grid search to determine the values of the hyper-parameters (see **Table A2**). The two optimisation methods used to train the MLP were the Adam and L-BFGS algorithms. Adam is a first-order stochastic optimisation method that uses individual adaptive learning rates for the different parameters. L-BFGS (Limited-memory Broyden-Fletcher-Goldfarb-Shanno) is a quasi-Newton method.

frontiers in Computational Neuroscience

ORIGINAL RESEARCH
published: 20 December 2016
doi: 10.3389/fncom.2016.00136

Hierarchical Chunking of Sequential Memory on Neuromorphic Architecture with Reduced Synaptic Plasticity

Guoqi Li[1†], Lei Deng[1†], Dong Wang[1†], Wei Wang[2], Fei Zeng[3], Ziyang Zhang[1], Huanglong Li[1], Sen Song[4], Jing Pei[1*] and Luping Shi[1*]*

[1] Department of Precision Instrument, Center for Brain Inspired Computing Research, Tsinghua University, Beijing, China, [2] School of Automation Science and Electric Engineering, Beihang University, Beijing, China, [3] Department of Materials Science and Engineering, Tsinghua University, Beijing, China, [4] School of Medicine, Tsinghua University, Beijing, China

Chunking refers to a phenomenon whereby individuals group items together when performing a memory task to improve the performance of sequential memory. In this work, we build a bio-plausible hierarchical chunking of sequential memory (HCSM) model to explain why such improvement happens. We address this issue by linking hierarchical chunking with synaptic plasticity and neuromorphic engineering. We uncover that a chunking mechanism reduces the requirements of synaptic plasticity since it allows applying synapses with narrow dynamic range and low precision to perform a memory task. We validate a hardware version of the model through simulation, based on measured memristor behavior with narrow dynamic range in neuromorphic circuits, which reveals how chunking works and what role it plays in encoding sequential memory. Our work deepens the understanding of sequential memory and enables incorporating it for the investigation of the brain-inspired computing on neuromorphic architecture.

Keywords: chunking, synaptic plasticity, sequential memory, neuromorphic engineering, memristor

OPEN ACCESS

Edited by:
Marcel Van Gerven, Radboud University Nijmegen, Netherlands

Reviewed by:
Pablo Varona, Autonomous University of Madrid, Spain
Alexantrou Serb, University of Southampton, UK

***Correspondence:**
Guoqi Li
liguoqi@mail.tsinghua.edu.cn
Jing Pei
peij@mail.tsinghua.edu.cn
Luping Shi
lpshi@mail.tsinghua.edu.cn

[†] These authors have contributed equally to this work.

Received: *12 August 2016*
Accepted: *01 December 2016*
Published: *20 December 2016*

Citation:
Li G, Deng L, Wang D, Wang W, Zeng F, Zhang Z, Li H, Song S, Pei J and Shi L (2016) Hierarchical Chunking of Sequential Memory on Neuromorphic Architecture with Reduced Synaptic Plasticity. Front. Comput. Neurosci. 10:136. doi: 10.3389/fncom.2016.00136

1. INTRODUCTION

The word "Chunking," a phenomenon whereby individuals group items together when performing a memory task, was initiated by (Miller, 1956). (Lindley, 1966) showed that groups produced by chunking have concept meanings to the participant. Therefore, this strategy makes it easier for an individual to maintain and recall information in memory. For example, when recalling a number sequence 01122014, if we group the numbers as 01, 12, and 2014, mnemonic meanings for each group as a day, a month and a year are created. Furthermore, studies found evidence that the firing event of a single cell is associated with a particular concept, such as personal names of Bill Clinton or Jennifer Aniston (Kreiman et al., 2000, 2001).

Psychologists believe that chunking plays as an essential role in joining the elements of a memory trace together through a particular hierarchical memory structure (Tan and Soon, 1996; Edin et al., 2009). At a time when information theory started to be applied in psychology, Miller claimed that short-term memory is not rigid but amenable to strategies (Miller, 1956) such as chunking that can expand the memory capacity (Gobet et al., 2001). According to this information, it is possible to increase short-term memory capacity by effectively recoding a large amount of low-information-content items into a smaller number of high-information-content items (Cowan, 2001; Chen and Cowan, 2005). Therefore, when chunking is evident in recall tasks, one can expect a

higher proportion of correct recalls. Patients with Alzheimer's disease typically experience working memory deficits; chunking is also an effective method to improve patients' verbal working memory performance (Huntley et al., 2011).

However, to this day, the mechanism why chunking improves human memory is unclear. This is mainly due to two difficulties. Firstly, no mathematical model is applicable to describe the memory processing in human brain. Secondly, no bio-plausible validation system that allows to emulate how chunking can be merged into a proper memory model. Although researchers have a long way to go before synthetic systems can match the capability of the natural brain, there are breakthroughs in neuroscience and neuromorphic engineering studies (Mead, 1989):

(1) *The discovery of the link between transient metastability and sequential memory in the brain.* Advances in non-invasive brain imaging (Gholipour et al., 2007) allow assessing the structural connectivity of the brain and corresponding evolution of the spatio-temporal activity in details. This makes the structure and dynamics of functional brain networks useful for building theoretical memory models. Among these results, one popular view is that, sequential memory, which refers to the functionality of the brain to encode and represent the temporal order of discrete elements occurring in a sequence, plays a key role in organizing the brain memory. The metastable state (Rabinovich et al., 2008, 2014; Mante et al., 2013; Tognoli and Kelso, 2014) is a significant feature of sequential memory. Experimental and modeling studies suggest that most of the sequential memories are the result of transient activities of large-scale brain networks in the presence of noise (Rabinovich et al., 2008; Maass, 2014).

(2) *The discovery of the bridge between the synapse and the memristor.* A synapse is a functional unit of the brain, which permits a neuron to pass an electrical or chemical signal to another neuron. In the last few years, it is believed that a synapse bears striking resemblance to a two-terminal electrical device termed as "memristor" (memory + resistor) (Chua, 1971; Strukov et al., 2008; Kim et al., 2012). The memristor resistive states can be modified by controlling the voltage applied across its terminals or the current flowing through it, which makes it promising to emulate the biological synapse (Jo et al., 2010; Chang et al., 2011; Kuzum et al., 2011; Yu et al., 2011; Alibart et al., 2012; Jackson et al., 2013; Kuzum et al., 2013; He et al., 2014). Clearly, advancements in memristor technology are establishing entirely new fashions in brain-inspired chip design.

Based on above breakthroughs, we set out to investigate why chunking improves sequential memory performance. To achieve this, we build a bio-plausible hierarchical chunking of sequential memory (HCSM) model shown in **Figure 1** using memristors as synapses. More specifically, our works are summarized as follows. Firstly, a HCSM model consisting multilayered neural networks is proposed. Each layer is divided into different chunks of neurons. Within each chunk, neurons are all-to-all connected (**Figure 3**); while chunks in different layers might be correlated through an activation signal denoted by the dotted arrows in **Figure 1**. In particular, a chunk in the upper layer and its connected ones in the adjacent lower layer are termed as parent chunk (PC) and child chunks (CCs), respectively. A winner neuron in a PC activates its connected child chunk (CC) to form a hierarchical structure (Figure S4). The winnerless competition (WLC) (Rabinovich et al., 2001) principle is applied between neurons, i.e., the winner is temporary or "metastable" because it switches from one neuron to another. The HCSM model selects the necessary metastable states and link them together to form a sequential memory through the hierarchical organization. When a recall cue is given, the model presents a memory trace containing temporary winner neurons among different chunks. The trace reflects the sequential memory recall. Secondly, to emulate the synapse with ideal synaptic plasticity, we use iron oxide (He et al., 2014) as the memristor resistive layer. A memristor with a typical sandwich structure, $0.25\ \mu m^2 - size\ TiW/Pt/FeO_x/Pt/TiW$, is fabricated, as shown in Figure S1. The well-known I-V hysteresis loops of memristor (Chua, 2011) under applied triangle-wave-shape DC voltage sweeps are observed. The conductance of this memristor can be monotonically and consecutively modulated among the intermediate states, which is crucial for the synaptic plasticity emulation. Lastly, we provide a neuromorphic chip implementation (**Figure 3**, Figures S2–S4) in which the memristor crossbar is used for emulating the synapse matrix of each chunk in the proposed HCSM model, and a scheme for encoding the sequential memory is presented. The key to encode memory in a bio-neural network is to exploit its ability of changing the synaptic weights (Zeng et al., 2001), which is also known as synaptic plasticity. In fact, synaptic plasticity is widely believed to be essential for creating the memory and learning ability of the brain (Hebb, 1949; Bi and Poo, 1998; Song et al., 2000; Han et al., 2011; Ramanathan et al., 2012; Carrillo-Reid et al., 2015).

With the HCSM model, the chunking mechanism can be linked to the synaptic plasticity. Usually, the dynamic range of the synapse, i.e., a memristor when considering a neuromorphic chip, is required to be much wider if the same length of sequential memory is encoded without chunking. By contrast, we observed that only a narrower dynamic range and imprecise state of the synaptic weight is required to encode a sequential memory with chunking mechanism. Thus, it is shown that chunking improves sequential memory by reducing the requirements of synapse plasticity in memory encoding. Our work reveals how chunking works and what role it plays in encoding sequential memory.

2. MATERIALS AND METHODS

As illustrated in **Figure 2**, this work explains why hierarchical chunking mechanism helps improve the memory performance and provides a promising solution to successfully realize memory dynamics in neuromorphic circuits. Through the reduced "synaptic plasticity" provided by the chunking mechanism, i.e., narrow dynamic range and not so precise state, we

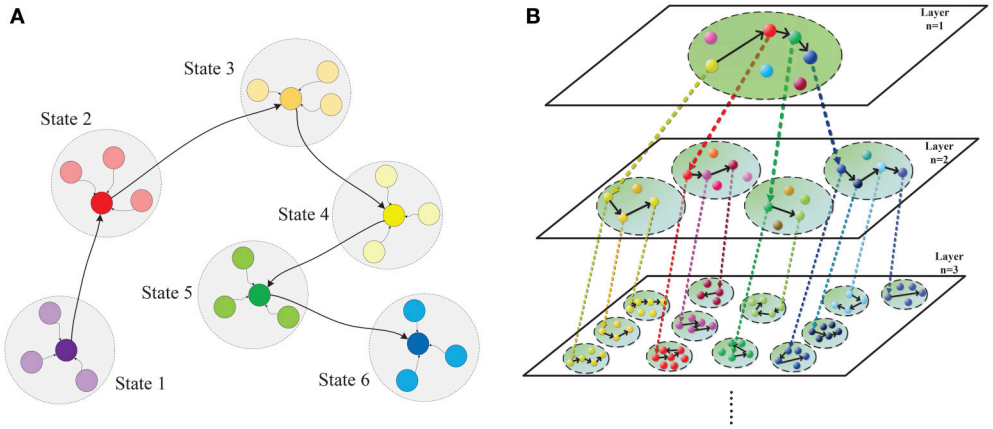


FIGURE 1 | Hierarchical chunking of sequential memory (HCSM) model. (A) The metastable states within a chunk, within which neurons are all-to-all connected. Each colored node denotes a temporal winner neuron in the chunk, which is a metastable state switching from one to another in the sequential memory trace. The shadowed circle encompassing each metastable state represents the domain of attraction for the state. **(B)** An illustrative diagram of the proposed hierarchical chunking sequential memory model. The chunk indexed by (n, m) denotes the mth chunk in layer n. A neuron representing a particular concept in a chunk (termed as PC) in layer n may activate a chunk (termed as CC) in layer $n+1$ that is connected to the PC neuron through a dotted arrow. In a memory recall, as the metastable neuron switches from one to another in a PC, the corresponding CCs is activated in a sequential order.

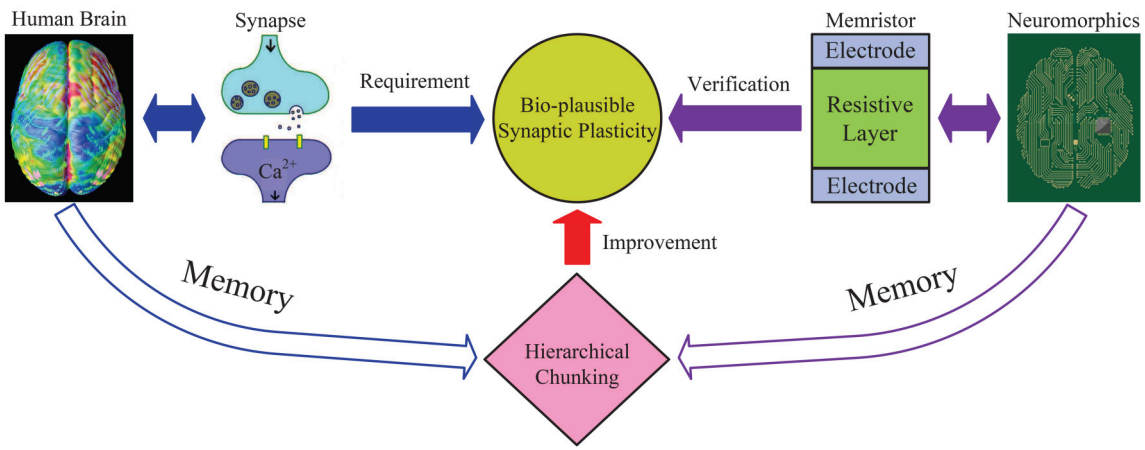


FIGURE 2 | An illustrative diagram of the main idea in this work. General neural networks without chunking require the synapse to have wide dynamic range, especially to memorize a long sequence. While the proposed hierarchical chunking mechanism greatly improves memory performance under a lower requirement for synaptic plasticity, i.e., only requires a narrow dynamic range and not a very precise state, which seems bio-plausible. A neuromorphic architecture is designed based on memristor devices with narrow dynamic range to successfully perform the sequential memory simulation. In this way, the reduced "synaptic plasticity" provided by chunking model bridges the brain memory dynamics and neuromorphic system.

establish a bridge between the brain memory dynamics and the neuromorphic system.

2.1. Synapse and Memristor

The molecular nature of the synaptic plasticity has been mathematically examined to have identical calcium-dependent dynamics, where the synaptic weight is described by a linear equation as follows (Shouval et al., 2002):

$$\frac{dW_i}{dt} = \frac{1}{\tau([Ca^{2+}]_i)} \left(\Omega([Ca^{2+}]_i) - W_i\right), \quad (1)$$

where W_i is the synaptic weight of the i-th input axon. τ is a time constant with respect to the insertion and removal rates of neurotransmitter receptors, which is a function dependent on the concentration of calcium $[Ca^{2+}]$. Ω is another function of $[Ca^{2+}]$ that depends linearly on the number of receptors on the membrane of the neuron. Equation (1) implies that the present synaptic weight between neurons is dependent on the historical weight indirectly, and it can be adjusted by changing $\Omega([Ca^{2+}])$.

To mimic the biological synapse, it is critical to build an artificial synaptic device to emulate its plastic behavior. Fortunately, the memristor (Strukov et al., 2008) was successfully developed and found to bear striking resemblance to the synapse in neural networks. The fundamental characteristic of a memristor is that its present resistance is dependent on its

historical resistances. The resistance of a memristor can be adjusted by changing the applied voltage or current, which controls the transport of charge carriers in the nanoscale device. In this work, iron oxide is adopted as the memristor resistive layer. As shown Figure S1 in the Supplementary Information, a typical memristor of sandwich structure with 0.25 μm^2 − *size* $TiW/Pt/FeO_x/Pt/TiW$ is fabricated.

However, as shown in (Kuzum et al., 2013), the dynamic range of memristor conductance to effectively emulate a synapse is often relatively narrow. For instance, the iron oxide memristor fabricated in this work is an ideal synaptic device due to its monotonous and consecutive state distribution. Note that the resistive ratio of the maximum conductance to the minimum conductance reflects the dynamic range of synaptics weight. As seen in Figure S1C, the ratio is only about $3 \sim 4$. This is consistent with a narrow distribution of biological synaptic weights that generally follows a lognormal distribution (Song et al., 2005; Teramae and Fukai, 2014). With the proposed HCSM model, it will be shown later that the neuromorphic system also works well since HCSM reduces the requirements on the synaptic plasticity.

2.2. Hierarchical Chunking of Sequential Memory (HCSM) Model

We propose a hierarchical chunking of sequential memory (HCSM) model shown in **Figure 1**, which consists of multi-layered networks. Each dashed circle indexed by a unique tuple (n, m) represents the mth chunk in the nth layer. Within each chunk, neurons are all-to-all connected (**Figure 3**). It can be seen that in each layer n, each neuron is connected to a specific sub-chunk in layer $n + 1$ to form a hierarchical structure (Figure S4), through an activation signal denoted by the dotted arrows in **Figure 1**. Thus we refer a chunk in layer n and its connected chunks in layer $n + 1$ as a parent chunk (PC) and child chunks (CCs), respectively. Clearly, a chunk in layer n is a CC, with respect to its PC in layer $n - 1$, and also a PC with respect to its CCs in layer $n + 1$. In other words, PCs and CCs only represent relative relationships between connected chunks from two successive layers.

The winnerless competition principle (WLC) (Rabinovich et al., 2001) among neurons in each network is described by the generalized Lotka-Volterra model in (Bick, 2009). The neuron preserving the maximum activity is a winner neuron. Here "winnerless" implies a winner is only metastable and it will switch from one neuron to another in a sequential memory trace as shown in **Figure 1A**. As each temporary winner neuron in a PC chunk will activate its connected CC, competition exists among different chunks in the same layer. When a recall cue is given, the HCSM model presents a memory trace containing all temporary winner neurons as shown in **Figure 1B**. In order to apply the WLC principle to the model of hierarchical architecture in this work, a time constant ($\tau > 0$) is introduced to reflect the dynamic evolving rate in the generalized Lotka-Volterra model. The WLC in a chunk indexed by (n, m) is then described by the following dynamic equation:

$$\dot{x}_i^{(n,m)} = \tau^{(n,m)} \cdot x_i^{(n,m)} (\sigma_i^{(n,m)} - \sum_{j=1}^{N_0^{(n,m)}} w_{ij}^{(n,m)} \cdot x_j^{(n,m)}) + v_i^{(n,m)} \tag{2}$$

where $\tau^{(n,m)}$ is the time constant that reflects the rate of activation and decay of $N_0^{(n,m)}$ neurons in the chunk (n, m), $\sigma_i^{(n,m)}$ is a fixed bias term that determines the equilibrium neural activity in the absence of external inputs and noise, $x_i^{(n,m)} \geq 0$ is the output neural activity of neuron i, $w_{ij}^{(n,m)} \geq 0$ for $j \neq i$ is the inhibitory weight (Brunel and Wang, 2001) from neuron j to neuron i, $w_{ii}^{(n,m)} = 1$ for $i = 1, ..., N_0^{(n,m)}$, and $v_i^{(n,m)}$ is the external noise in the interval $[-\varepsilon, \varepsilon]$ where ε is a small positive constant. Note that the weight $w_{ij}^{(n,m)}$, and $\sigma_i^{(n,m)}$ for all i and j are required to be encoded when performing specific task. Also, the connection between the nodes in different layers is not reflected in the dynamic Equation (2). However, it is assumed that a neuron in the PC can activate a CC chunk that is connected to the PC neuron through a activation signal denoted by dotted arrow. This implies that different chunks, for example the chunks in layer 2 of **Figure 1**, do not interact with each other directly in the same layer, but they indeed interact with each other in a particular way in the higher layer (layer 1). The time constant $\tau^{(n,m)}$ for a CC is normally smaller than that in a PC, which implies that the chunks in the CC layers possess relatively faster dynamic evolving rates. Since the HCSM model may be a deep architecture, the time constant τ has a wide range. It is known that there is indeed a wide range of time constants for the neurons in the human brain, from hundreds of milliseconds to tens of seconds (Bernacchia et al., 2011).

Note that each neuron represents a particular item in memory such as a digital number or a letter of the alphabet. The neural activity in a dynamic system (Equation 2), which is time-varying, reflects the level of activity of each neuron in a neural network. At a given time instant, the neuron of the maximum neural activity among a chunk becomes the temporal winner. The corresponding item that the winner neuron represents will be recalled.

2.3. Encoding Scheme for the HCSM Neuromorphic Network

Suppose that there are N_0 neurons in a chunk, and each neuron represents a particular item in the memory. Before we encode a sequence containing $\kappa \leq N_0$ metastable states in a chunk as described in Equation (2), the bias parameter of each neuron and the weight between two arbitrary neurons need be determined first. In this work, the bias parameter of neuron i in a specific chunking sequence is chosen as

$$\sigma_i = \begin{cases} F_k, & \text{if neuron } i \text{ is the } k\text{-th term of the chunking sequence.} \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

where F_k is the k-th term of the Fibonacci sequence (Dunlap, 1997) with $F_1 = 1$ and $F_2 = g$. Here g is the "Golden ratio" (Dunlap, 1997; Livio, 2008). The synaptic weight between neurons i and j in the same chunk is then selected as:

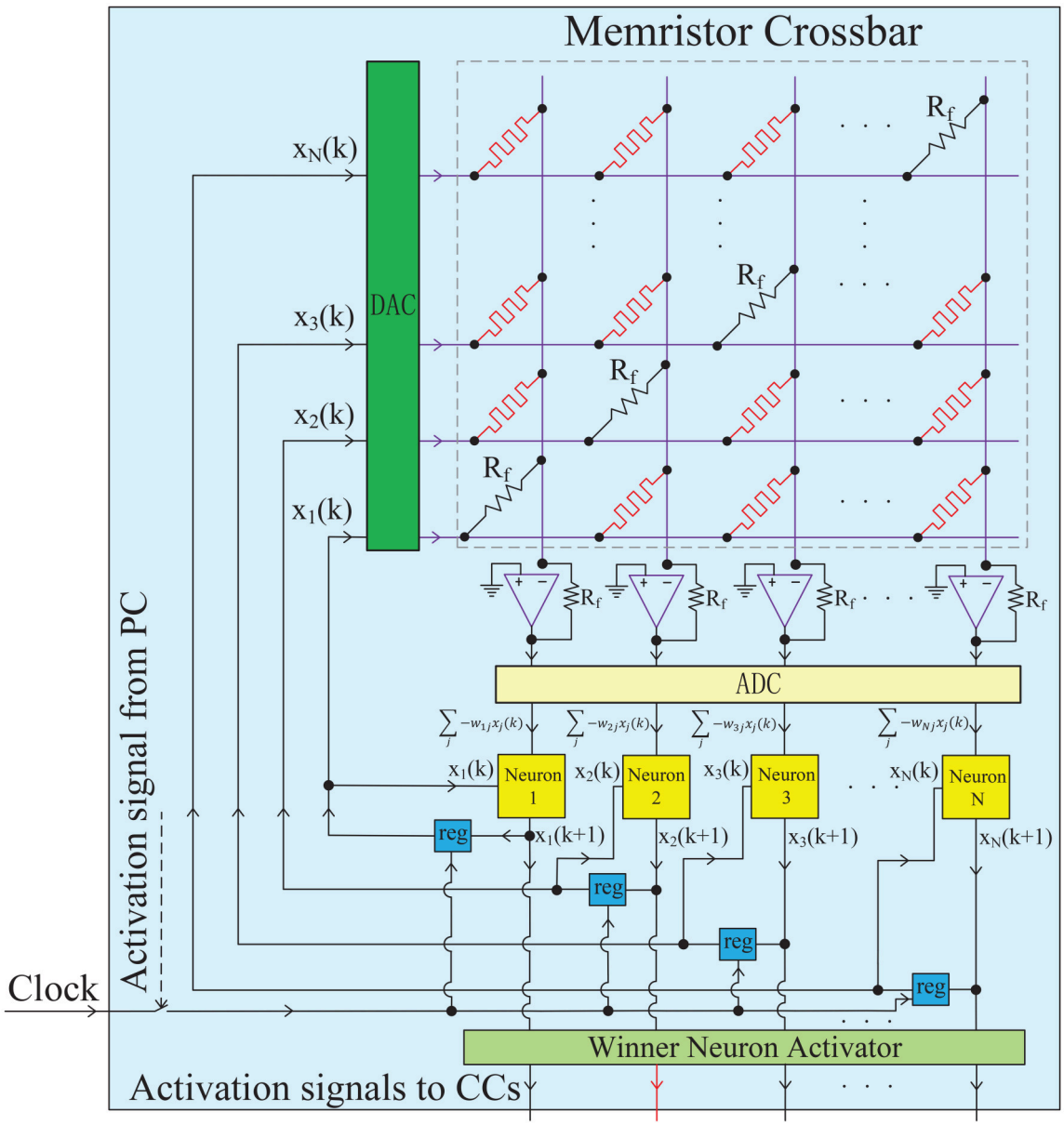


FIGURE 3 | Hardware implementation for a single chunk. The chunk, a PC or a CC, is mainly constructed by an analog memristor crossbar circuit and digital neurons. The memristor crossbar completes a VMM operation in one step, and the digital neurons run the neuronal dynamics described in Equation (6). The diagonal elements of the crossbar are non-plastic, with the same value as R_f to satisfy $w_{ii} = 1$. The DACs and ADCs are required to convert the signal format between analog circuits and digital circuits. The iterative timing sequence $k \rightarrow k+1$ is governed by the clock signal. The winner neuron activator in each chunk (PC) is used to determine the winner neuron at each time step and transmit an excitatory signal to the corresponding connected CC block. When a CC receives an excitatory activation signal from a winner neuron in the PC, the clock will be triggered and the iterative neuronal dynamics in this CC starts to form a pre-defined memory trace.

$$w_{ij} \in \begin{cases} S_1, & \text{if neurons } i \text{ and } j \text{ are adjacent in the chunking sequence.} \\ S_2 & \text{otherwise.} \end{cases} \quad (4)$$

with

$$\begin{aligned} S_1 &= \{x | g - \tfrac{1}{2} < x < g\} \\ S_2 &= \{x | g^{\mathbf{k}-1} + 1 < x < +\infty\} \end{aligned} \quad (5)$$

where **k** is the length of the sequence in this chunk. A detailed motivation to the above encoding scheme associated with the existence of the metastable states in a stable sequential memory trace are provided in Theorem 1 in the supplementary information.

2.4. Hardware Implementation

As well known, the neuromorphic engineering especially the memristive system enables the hardware implementation of

neural networks with ultra-low power, small size, and high speed (Kuzum et al., 2012; Yu et al., 2013; Deng et al., 2016), which aims at the future of mobile intelligence. However, the memristor with good plasticity to emulate synapse usually suffers from a narrow dynamic range. Fortunately, the proposed HCSM model efficiently reduces the requirement on synaptic plasticity. This coincidence motivates us to build a memristive architecture of the chunking model and demonstrate the feasibility, which provides neuromorphic engineers with a promising solution to realize dynamical memory on hardware platform.

We fabricate a FeO_x-based memristor device with typical sandwich structure, whose detailed process and electrical characteristics are shown in Figure S1. The conductance state can be monotonously and consecutively modulated under a series of positive or negative pulses, i.e., with good plasticity. The positive pulse train gradually increases the conductance, corresponding to the short/long term potentiation (STP/LTP) process; while the negative pulse train results in short/long term depression (STP/LTD). The resistive ratio of highest conductance to lowest conductance is only $3 \sim 4$.

The memristor is cascaded with an amplifier to perform as a functional synapse. The one-input-to-one-output structure and multiple-input-to-one-output structure are illustrated in Figures S2A,B, respectively. The equivalent synaptic weight is co-determined by the memristor conductance G and feedback resistance R_f on amplifier, $w = V_{out}/V_{in} = -R_f G$, which is a dimensionless value indicating the voltage transmission efficiency from input to output. In this manner, the negative weight realizes the inhibitory connection in HCSM model. For the case of multiple inputs injected to one amplifier, all the memristors form a parallel circuit and the transfer function is provided in Figure S2. The amplifier is able to accumulate the multiple synaptic inputs, like the integration function of dendrites. This feature efficiently supports the multiplication and accumulation (MAC) operations between the inputs and weights in Equation (2).

The neuron dynamics described by the differential equation (Equation 2) can be numerically solved based on its corresponding difference equation

$$x_i^{(n,m)}(t+dt) = x_i^{(n,m)}(t) + dt \cdot \{\tau^{(n,m)} x_i^{(n,m)}(t) \cdot [\sigma_i^{(n,m)} - \sum_{j=1}^{N_0} w_{ij}^{(n,m)} \cdot x_j^{(n,m)}(t)] + v_i^{(n,m)}\}. \quad (6)$$

If we replace the evolution "$t \rightarrow t + dt$" by "$k \rightarrow k + 1$," we can achieve a numerical iteration process. The one-to-one corresponding digital neuron block is shown in Figure S3A. A Fibonacci sequence block is also necessary to determine $\sigma_i^{(n,m)}$, as well as the upper and lower bounds of the synaptic weights, as shown in Figure S3B.

Based on the element synapse and neuron block, a chunk network can be implemented, as demonstrated in **Figure 3**. Each chunk, a PC or a CC, is mainly constructed by an analog memristor crossbar circuit and digital neurons. More specifically, the weighted synapses which is the most critical part in this neural network, are implemented by a memristor crossbar circuit. Each column in the memristor crossbar and the cascaded amplifier on that column perform a MAC operations as shown in Figure S2B. All columns are assumed to be independent without crosstalk, so that the whole memristor crossbar and the amplifier array can well realize the matrix-vector multiplication (VMM) which is the major operation in neural networks. This indicates that the architecture supports one-time projection from multiple inputs to multiple outputs, with the advantages of small size, high speed and low power. It is worth noting that the diagonal elements of the crossbar are non-plastic resistors (not memristor) with the same value as the feedback resistance R_f on the amplifier. Thus, the requirement of $w_{ii} = 1$ in HCSM model is met. Each neuron block iteratively runs the dynamics of Equation (6). The neuronal outputs at each time step are stored in temporal registers and fed back into the network as synaptic inputs at the next time step. In fact, the timing sequence of the whole network ($k \rightarrow k + 1$...) is governed by the clock signal. Actually, the chunk is an analog-digital hybrid circuits, DACs (digital to analog converters) and ADCs (analog to digital converters) are required to convert the signal format (Li et al., 2013). Furthermore, each chunk circuit can be hierarchically organized together to form a complete HCSM model, as shown in Figure S4. The winner neuron activator in each chunk (PC) is used to determine the winner neuron at each time step and transmit an excitatory signal to the corresponding connected CC circuit. When a CC receives an excitatory activation signal from a winner neuron in the PC, the clock will be triggered and the iterative neuronal dynamics in this CC starts to form a pre-defined memory trace.

When performing a real task, the memristive networks often work in two stages: the write (synaptic modulation) stage and the read (neuronal processing) stage. During the write stage, the memristive crossbar is fully controlled by the pulse modulator block, as presented in Figure S5. The weight calculator block calculates the theoretical weight of each synapse according to a pre-defined chunking sequence based on Equations (3)–(5), and the pulse modulator generates the pulse train (potentiation or depression pulses) to modify the conductance of each memristor to the desired value. Two detailed modulation methods are illustrated in Figures S6, S7. Different from the conventional direct configuration in computer software, neuromorphic implementation has to gradually program the conductance of hardware synapse array from a random initial state to the target state that is produced by the weight calculator. Pulse tuning scheme is more popular, compared to DC switching, since its well controllable modulating increment can achieve relatively high precision. The open-loop modulation directly uses the behavior model of memristor device to determine the direction and number of pulses to move any initial conductance state to the desired one. Considering real device variability, one-time open-loop modulation sometimes cannot reach the ideal state. To this end, the closed-loop modulation repeatedly performs the open-loop modulation until the desired conductance is achieved. More generally, the closed-loop modulation can use the trial-and-error method to gradually tune the conductance without the guidance of theoretical behavior model. Furthermore, the modulation process is flexible by choosing proper pulse amplitude (Kuzum et al., 2011), width (Snider, 2008), and frequency (He et al., 2014) of the pulse train. However, the versatile pulse tuning schemes will drastically

increase the burden of the pulse generator, hence it is often hard to be executed in hardware systems. To mitigate the burden, a number of identical pulses are adopted in this work to modulate the memristor states, as mentioned earlier in Figure S1C.

When updating the conductance of a specific memristor in the crossbar, it would be firstly selected to avoid influencing the states of other unselected memristors. Some typical selective devices are useful such as diode or transistor (Wong et al., 2012), and even selector-free memristor crossbar is possible (Prezioso et al., 2015). The half-selected technique is also used to further prevent unintentional operation on the unselected memristors (Yang et al., 2013). This is because the conductance change only occurs when the amplitude of modulation pulses is above the threshold voltage and no significant conductance change is observed under applied voltages below the threshold (Jo et al., 2010). The full programming voltage V_P and V_D on the selected memristor is above the threshold, while the half voltage $V_P/2$ and $V_D/2$ is configured below the threshold. Then while one synapse is under programming, the others are clamped at their current states with a lower half-selected voltage. During the read stage, the pulse modulator is switched off and the data flows from the memristor crossbar to the neuron block, and then feeds back at next time step. The input voltages are scaled to be sufficiently small that the trained memristor states would not change during the whole read stage.

This paper aims for offering a heuristic solution to guide neuromorphic engineers to embed a dynamical memory model into future neuromorphic platforms. In all the following simulations, we present less peripheral circuit details but pay more attention to the influence of dynamic range and precision of memristor device, which are the two key points narrowing the gap between memristive system and HCSM model. Based on the real memristor data in Figure S1, as well as some existing physical models and behavioral models of the memristor (Strukov et al., 2008; Yang et al., 2008; Guan et al., 2012a; Suri et al., 2012; Deng et al., 2015), we build an iron oxide memristor model whose synaptic behavior shows excellent agreement with the real device experiments (Figure S1D). Furthermore, we use SPICE (a standard circuit simulator) to verify the proposed network model of HCSM shown in **Figure 3** and Figure S4.

3. RESULTS

3.1. Chunking and Synaptic Plasticity

We denote $\sup(S_1^{(n,\,m)})$ as the supremum of set S_1 defined in Equation (5) in the chunk (n, m), and $\inf(S_2^{(n',\,m')})$ as the infimum of set S_2 in the chunk $(n',\ m')$. The ratio of $\inf(S_2^{(n',\,m')})$ to $\sup(S_1^{(n,\,m)})$ generally implies the requirement of synaptic plasticity to recall the sequences. As there are multiple chunks in different layers, a relative synaptic plasticity requirement index φ is defined as

$$\varphi = \frac{\max\{\inf(S_2^{(n',\,m')})\}}{\min\{\sup(S_1^{(n,\,m)})\}} = \frac{g^{\max\{\mathbf{k}^{(n,\,m)}\}-1}+1}{g} = g^{\max\{\mathbf{k}^{(n,\,m)}\}-2} + g^{-1} \tag{7}$$

where $\max\{\mathbf{k}^{(n,\,m)}\}$ is the length of the longest chunking sequence for a particular memory task. It is known that φ in real neurobiological systems should be less than an upper bound, which constitutes the capacity boundary of sequential memory. As mentioned previously, the dynamic range of memristor with good synaptic plasticity is often relatively narrow. This may relate to capacity limitations in the human brain. Note that in the HCSM, a sequence is divided into a series of subsequences with different lengths, and the chunk with the longest sequence mainly determines the requirement on the dynamic range of the memristor. In this regard, the HCSM model is capable of having the neuromorphic system maintaining its performance with a reduced requirement of synaptic plasticity.

As shown in the literature, the synaptic weight distribution (Barbour et al., 2007) in the human brain follows a lognormal distribution (Song et al., 2005; Teramae and Fukai, 2014), as illustrated in **Figure 4A**. This indicates that the synaptic weights mainly locate in a narrow domain. Note that generally it is impossible to estimate the relative synaptic plasticity requirement index φ in human brain and bio-neural systems by applying (Equation 7). Here we define another index $\tilde{\varphi}$ to address this issue. Let $\tilde{w}$ be the median weight and define $\tilde{\varphi} = \frac{w}{\tilde{w}}$ as the measurement of relative synaptic plasticity index in bio-neural systems. Thus, in **Figure 4A**, it is observed that for 80 and 90% of synaptic weights, the synaptic plasticity index

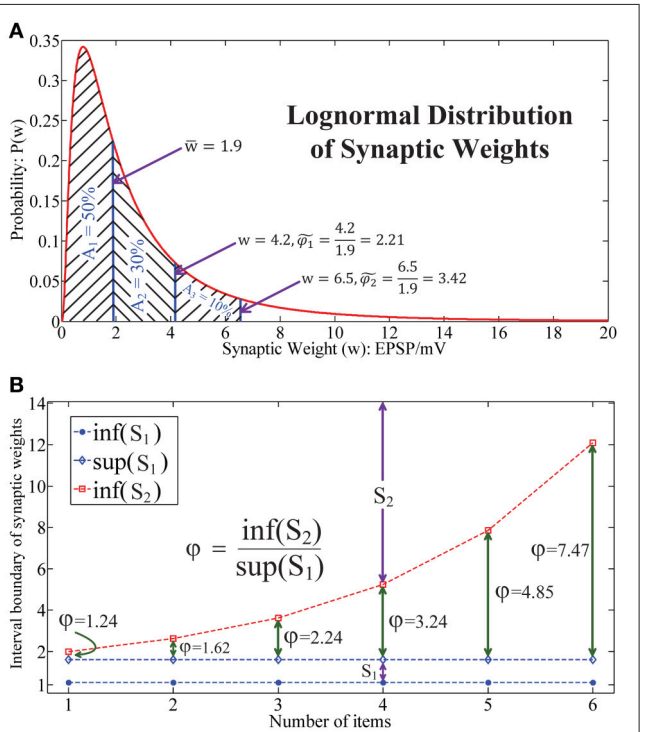


FIGURE 4 | Lognormal distribution of synaptic weights underlies the chunking mechanism. (A) The lognormal distribution of synaptic weights in real neural systems, which implies that the dynamic range of the synapses in the brain is in a narrow interval. **(B)** The relative synaptic plasticity requirement index with respect to the length of the sequence in a particular chunk for HCSM.

is below $\tilde{\varphi}_1 = \frac{4.2}{1.9} = 2.21$ and $\tilde{\varphi}_2 = \frac{6.5}{1.9} = 3.42$, respectively.

While in **Figure 4B**, it is seen that φ increases exponentially with respect to the length of the sequence in HSCM. As a chunking mechanism allows us to encode a long sequence into many shorter subsequences, HCSM could work on the domain which requires a relative narrower dynamic range of the memristor in the neuromorphic network. By combining the results shown in both **Figures 4A,B**, when we apply $\tilde{\varphi}$ to reflect the requirement of synaptic plasticity in the brain/bio-neural systems, we provide a putative reason why the optimal items in sequential memory is 3–4 items, which is a long standing problem pointed out in (Simon, 1974) and (Gobet, 2004).

3.2. Impact on the Precision of the Synaptic Weights in HCSM

We simulate the hardware implementation of HCSM (**Figure 3**) and the results of neuronal activities are shown in **Figure 5**, where an example consisting of four winner neurons in each chunk is illustrated. Specifically, several sub-circuits with the same structure but different parameters, each represents a PC or a CC, have been established to form the complete HCSM model. A 50 Hz square wave is provided as the gated clock signal for all sub-circuits, while each gate is controlled by its activation signal. The clock gate is on when the activation signal is logically high. Then the sub-circuit is activated. The computation module of the activated sub-circuit, consisting of a memristor array, a group of neurons and other peripheral circuits, is then triggered by the 50 Hz clock signal. Output neuronal activities of the activated subcircuit gradually evolve following Equation (6) where each time step is kept smaller than a half clock period. The winner neuron activator then determines the winner neuron with maximum neuronal activity and set the corresponding activation signal logically high to activate its connected CC. It is seen that the ideal memory trace is successfully achieved, where different neurons become the temporal winner in turn. Thus, the trace in the CCs is instantly activated by a corresponding activation signal generated from the PC.

As discussed in (Kuzum et al., 2012), (Yu et al., 2013), (Guan et al., 2012b), and (Yu et al., 2012), the main challenge of memristor-based neuromorphic system is the notable variation of memristive devices during programming, including cycle-to-cycle variation and device-to-device variation. In this case, the performance of the encoded memristor-based neuromorphic network of HCSM in the presence of device variation need to be validated. We introduce different levels of fuzzy dispersion to the final weight values of synaptic weights in HCSM using our fabricated memristor as synapse when recalling a memory trace task. **Figure 6** shows the robustness of HCSM model by analyzing the fault-tolerance performance with respect to the weight variation of memristors. In particular terms, the network can perfectly trace the target sequence under a pessimistic 20% dispersion of the synaptic weights. As expected, the responses of pre-defined winner neurons gradually deviate from the ideal pattern with a rapid increasement of weight dispersion. For example, only three winner neurons successfully trace its memory under 30% dispersion of the synaptic weights, and the number of successful neurons reduces to two when the dispersion level increases to 50%. The trace pattern no longer converges to its stable state when the dispersion is larger than 70%. In general, our proposed HCSM model does not require precise synaptic weights in the encoding scheme, and a great degree of device variation can be tolerated. This suggests that chunking mechanism enables applying low precision synapses when performing a memory task.

3.3. Impact on the Dynamic Range of the Synaptic Weight in HCSM

In simulating the encoding process of a sequential memory on SPICE, two conclusions are obtained: (i) the dynamic range of the synaptic weight is required to be much wider if the same length of sequential memory is encoded without chunking; (ii) the success rate of the encoding in each chunk is a monotonously increasing function of the dynamic range synaptic weights.

Suppose that we encode a sequential memory with $\mathbf{k}$ items such that the square root of $\mathbf{k}$ is an integer. To achieve a lower relative synaptic plasticity requirement index defined in Equation (7), the best way is to encode the sequence in $\sqrt{\mathbf{k}}$ chunks, with each chunk consisting $\sqrt{\mathbf{k}}$ items. Then, the relative synaptic plasticity requirement index φ is obtained by

$$\varphi = \frac{\max\{\inf(S_2^{(n',\,m')})\}}{\min\{\sup(S_1^{(n,\,m)})\}} = g^{\sqrt{\mathbf{k}}-2} + g^{-1} \tag{8}$$

By comparing Equations (7) and (8), it is seen that we require the same dynamic range of the synaptic weight to encode $\mathbf{k}$ items with chunking mechanism and $\sqrt{\mathbf{k}}$ items without chunking mechanism.

We simulate the encoding process of a length of 16-items sequential memory which has 4 chunks, with each chunk consisting of 4 items on SPICE based on the encoding scheme we introduced in Equations (3)–(5). In Equation (5), we notice that the supremum of set S_2 can be positive infinity. However, in real applications it is well known that the synaptic weight can never be infinity. To show the impact of the dynamic range of the synaptic weight in HCSM more clearly, we set $|S_2| = |S_1|$ where $|.|$ denotes the measure/length of an interval, i.e., $S_2 = \{x | g^{\mathbf{k}-1} + 1 < x < g^{\mathbf{k}-1} + \frac{3}{2}\}$. Obviously, we have $|S_2| = |S_1| = \frac{1}{2}$. When both $|S_1|$ and $|S_1|$ are fixed, the relative synaptic plasticity requirement index φ also reflects the requirement of the dynamic range of synaptic weights. In SPICE simulation, φ is chosen from 2–4. In Figures S8–S11 in Supplementary Information, the effects for cases $\varphi = 2.0, 2.4, 3$, and 3.6 are shown in four figures, respectively. It is seen that a small φ usually leads to failure of the encoding of the sequential memory while a larger φ improves such a situation. We repeated the experiments 200 times to estimate the encoding success rate for each fixed φ in **Figure 7**, where it is shown that the encoding success rate in each chunk is a monotonously increasing function of the dynamic range of synaptic weights.

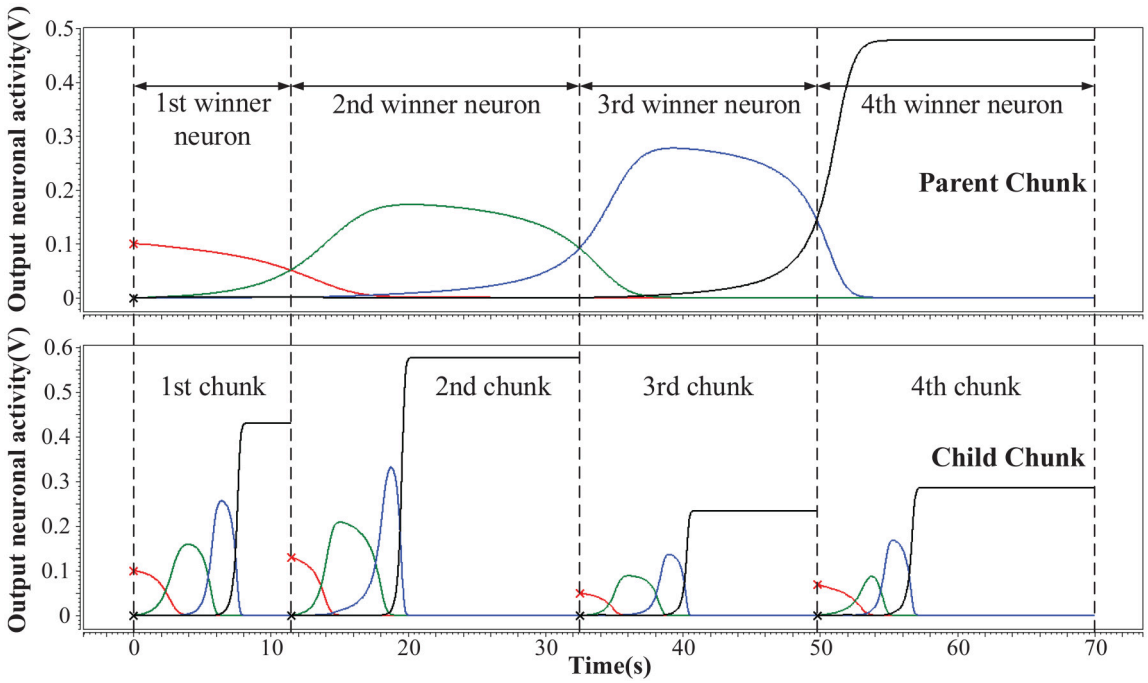


FIGURE 5 | Hierarchical memory traces in one PC and its CCs. The vertical axis is the output neuronal activity and the horizontal axis is the sampling time. At a given moment, the neuron which preserves the maximum activity is the temporal winner neuron.

4. DISCUSSION

In this work, we suggest a link between chunking mechanism and synaptic plasticity to answer a long standing question why chunking improves sequential memory. A hierarchical chunking of sequential memory (HCSM) model and a robust scheme regarding how to encode sequential memory are presented. It is observed from the encoding scheme that chunking mechanism reduces the requirements of synaptic plasticity when recalling a memory trace, including the tolerance of the dynamic range and precision of the synaptic weights. Furthermore, we provide a neuromorphic implementation to verify the proposed memory dynamics under the hardware constraints of narrow dynamic range and device variability. The successful demonstration indicates the feasibility to embed more complex memory models into future neuromorphic systems.

One merit of the proposed HCSM model is the robustness of the encoding method, i.e., the weight can be a random value in a given interval, which makes the model convenient to be realized by memristive devices. However, the disadvantages include (i) the model requires full connection of the neurons in the network; and (ii) the asymmetry in information storage fundamentally impairs the length of the memory trace. Therefore, the investigation of a new model that allows sparse connection of neurons to link metastable states together in a sequential memory would be of great interest. Also, besides chunking mechanism, how can the memory capacity of biological systems be improved deserves investigation. Furthermore, we would like to point out that in HCSM, a particular item in the memory trace is represented by only one neuron. However, experimental studies have revealed that population coding (Pasupathy and Connor, 2002) , a method to represent stimuli (a memory item) by using the joint activities of a number of neurons, is widely used in the brain (Averbeck et al., 2006). This implies that single neural coding method may be inadequate in practical applications. We conjecture that population coding could be applied to our model which deserves further investigations.

This work provides an addition to recent work on learning of chunking sequences (Fonollosa et al., 2015) including specific roles in cognitive process (Varona and Rabinovich, 2016). Specifically, this work provides a useful hardware validation means for many advanced theory researches. We also open up a new application space on neuromorphic platforms to implement not only HCSM, but also various bio-inspired memory models related to the encoding of the visual, acoustic and semantic information and so on. Predictably, the disciplines of cognitive psychology, neuroscience and information technology, and neuromorphic engineering becomes more and more important. The top-down bio-plausible theories fundamentally guide the development of future neuromorphic computing systems; while the bottom-up neuromorphic materials, chips, boards and systems usefully verify these pioneering theories. Although there is still a long road ahead, this work kindles a ray of hope.

The major difficulty preventing its application is the fabrication and management of large-scale memristor crossbar, especially considering the device variability and the crosstalk among adjacent cells. On the other side, the required peripheral circuits are also quite complex, including analog-digital

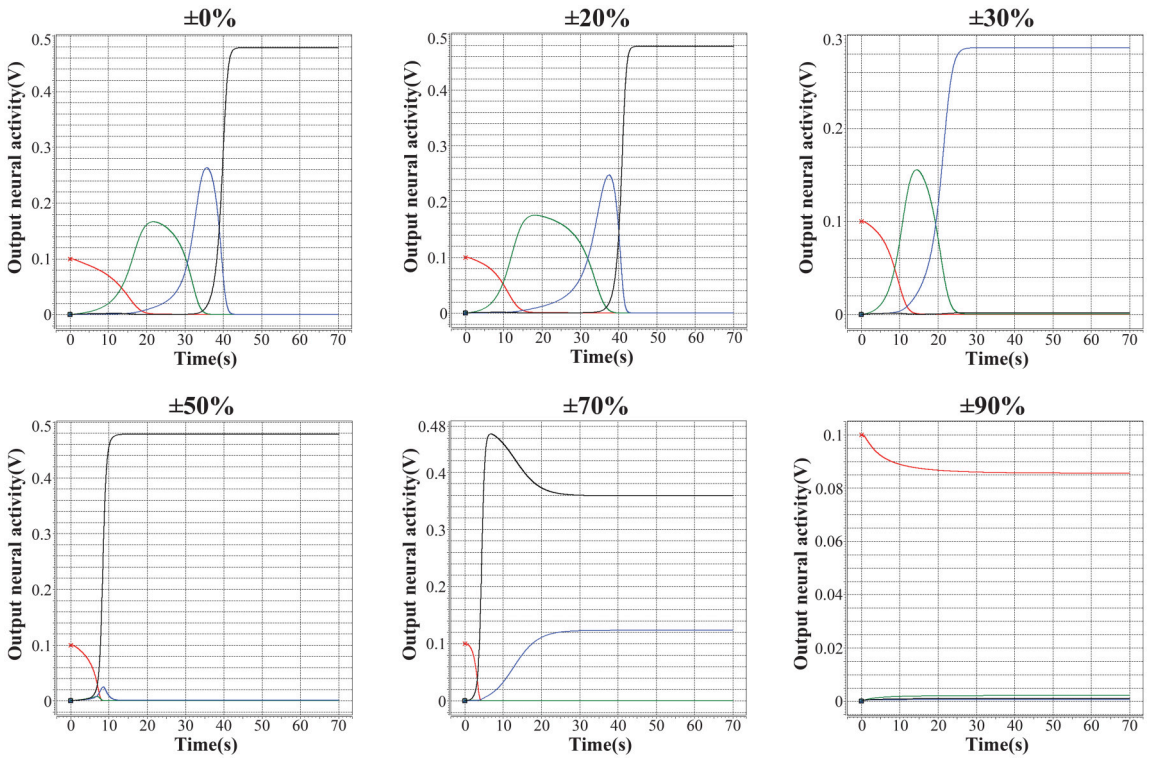


FIGURE 6 | Analysis of the fault-tolerance performance of the proposed HCSM system with respect to the variation of memristors. It can be seen that although with a weight dispersion of up to 20%, the HCSM system can perfectly pass by all the four temporal winners. And the system can pass by three of the four temporal winners even with a dispersion of 30%. Hence, a good tolerance of the proposed HCSM system to the device variation can be shown.

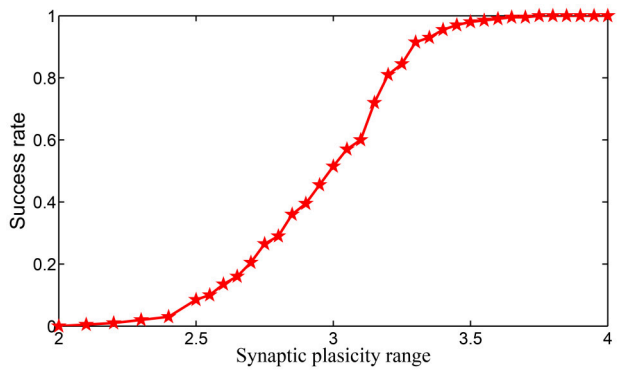


FIGURE 7 | The relationship of encoding success rate and the dynamic range of synaptic weights.

converters, read/write circuits, switching matrix as well as extra computing circuits for learning. Fortunately, some reported memristor-based artificial neural networks have shown that these developments may become feasible in the near future, at least in relatively small scale (Alibart and Zamanidoost, 2013; Garbin et al., 2014; Prezioso et al., 2015). With the development of integration techniques for large scale memristor crossbar or even 3D networks (Yu et al., 2013; Li et al., 2016), as well as memristor for logical or arithmetical computations (Borghetti et al., 2010; Gale, 2015) to reduce complex peripheral circuits by replacing the digital neurons, we envisage a real chip able to perform interesting memory tasks.

AUTHOR CONTRIBUTIONS

GL, LD, and SS proposed the model and designed the experiments. DW and WW did the SPICE simulation. ZZ, HL, and FZ did the device experiments. JP and LS contributed to the main idea of this paper. All authors write the manuscript.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China (No. 61603209 and No. 61475080), Beijing Natural Science Foundation (No. 4164086), and Tsinghua University Initiative Scientific Research Program (No. 20151080467 and No. 20141080934).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: http://journal.frontiersin.org/article/10.3389/fncom.2016.00136/full#supplementary-material

REFERENCES

Alibart, F., Zamanidoost, E. and Strukov, D. B. (2013). Pattern classification by memristive crossbar circuits using *ex situ* and *in situ* training. *Nat. Commun.* 4:2072. doi: 10.1038/ncomms3072

Alibart, F., Pleutin, S., Bichler, O., Gamrat, C., Serrano-Gotarredona, T., Linares-Barranco, B., et al. (2012). A memristive nanoparticle/organic hybrid synapstor for neuroinspired computing. *Adv. Funct. Mater.* 22, 609–616. doi: 10.1002/adfm.201101935

Averbeck, B. B., Latham, P. E., and Pouget, A. (2006). Neural correlations, population coding and computation. *Nat. Rev. Neurosci.* 7, 358–366. doi: 10.1038/nrn1888

Barbour, B., Brunel, N., Hakim, V., and Nadal, J. P. (2007). What can we learn from synaptic weight distributions? *Trends Neurosci.* 30, 622–629. doi: 10.1016/j.tins.2007.09.005

Bernacchia, A., Seo, H., Lee, D., and Wang, X. J. (2011). A reservoir of time constants for memory traces in cortical neurons. *Nat. Neurosci.* 14, 366–372. doi: 10.1038/nn.2752

Bi, G. Q., and Poo, M. M. (1998). Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *J. Neurosci.* 18, 10464–10472.

Bick, C. and Rabinovich, M. I. (2009). Dynamical origin of the effective storage capacity in the brains working memory. *Phys. Rev. Lett.* 103, 218101–218104. doi: 10.1103/PhysRevLett.103.218101

Borghetti, J., Snider, G. S., Kuekes, P. J., Yang, J. J., Stewart, D. R., and Williams, R. S. (2010). 'Memristive' switches enable 'stateful' logic operations via material implication. *Nature* 464, 873–876. doi: 10.1038/nature08940

Brunel, N., and Wang, X. J. (2001). Effects of neuromodulation in a cortical network model of object working memory dominated by recurrent inhibition. *J. Comput. Neurosci.* 11, 63–85. doi: 10.1023/A:1011204814320

Carrillo-Reid, L., Lopez-Huerta, V. G., Garcia-Munoz, M., Theiss, S., and Arbuthnott, G. W. (2015). Cell assembly signatures defined by short-term synaptic plasticity in cortical networks. *Int. J. Neural. Syst.* 25:1550026. doi: 10.1142/S0129065715500264

Chang, T., Jo, S. H., and Lu, W. (2011). Short-term memory to long-term memory transition in a nanoscale memristor. *ACS Nano* 5, 7669–7676. doi: 10.1021/nn202983n

Chen, Z., and Cowan, N. (2005). Chunk limits and length limits in immediate recall: a reconciliation. *J. Exp. Psychol. Learn. Mem. Cogn.* 31, 1235–1249. doi: 10.1037/0278-7393.31.6.1235

Chua, L. O. (1971). Memristor-the missing circuit element. *IEEE Trans. Circuit Theory* 18, 507–519. doi: 10.1109/TCT.1971.1083337

Chua, L. O. (2011). Resistance switching memories are memristors. *Appl. Phys. A* 102, 765–783. doi: 10.1007/s00339-011-6264-9

Cowan, N. (2001). The magical number 4 in short-term memory: a reconsideration of mental storage capacity. *Behav. Brain Sci.* 24, 87–114. doi: 10.1017/S0140525X01003922

Deng, L., Li, G., Deng, N., Wang, D., Zhang, Z., He, W., et al. (2015). Complex learning in bio-plausible memristive networks. *Sci. Rep.* 5:10684. doi: 10.1038/srep10684

Deng, L., Wang, D., Zhang, Z., Tang, P., Li, G., and Pei, J. (2016). Energy consumption analysis for various memristive networks under different learning strategies. *Phys. Lett. A* 380, 903–909. doi: 10.1016/j.physleta.2015.12.024

Dunlap, R. A. (1997). *The Golden Ratio and Fibonacci Numbers.* World Scientific Publishing.

Edin, F., Klingberg, T., Johansson, P., McNab, F., Tegnér, J., and Compte, A. (2009). Mechanism for top-down control of working memory capacity. *Proc. Natl. Acad. Sci. U.S.A.* 106, 6802–6807. doi: 10.1073/pnas.0901894106

Fonollosa, J., Neftci, E., and Rabinovich, M. (2015). Learning of chunking sequences in cognition and behavior. *PLoS Comput. Biol.* 11:e1004592. doi: 10.1371/journal.pcbi.1004592

Gale, E. (2015). Single memristor logic gates: from NOT to a full adder. *arXiv:*1510.05705

Garbin, D., Bichler, O., Vianello, E., Rafhay, Q., Gamrat, C., Perniola, L., et al. (2014). "Variability-tolerant convolutional neural network for pattern recognition applications based on oxram synapses," in *IEEE International Electron Devices Meeting (IEDM)* (San Francisco, CA), 28.4.1–28.4.4. doi: 10.1109/IEDM.2014.7047126

Gholipour, A., Kehtarnavaz, N., Briggs, R., Devous, M., and Gopinath, K. (2007). Brain functional localization: a survey of image registration techniques. *IEEE Trans. Med. Imaging* 26, 427–451. doi: 10.1109/TMI.2007.892508

Gobet, F., Lane, P. C., Croker, S., Cheng, P., Jones, G., Oliver, I., et al. (2001). Chunking mechanisms in human learning. *Trends Cogn. Sci.* 5, 236–243. doi: 10.1016/S1364-6613(00)01662-4

Gobet, F. and Clarkson, G. (2004). Chunks in expert memory: evidence for the magical number four or is it two? *Memory* 12, 732–747. doi: 10.1080/09658210344000530

Guan, X., Yu, S., and Wong, H. S. P. (2012a). A SPICE compact model of metal oxide resistive switching memory with variations. *IEEE Electr. Device Lett.* 33, 1405–1407. doi: 10.1109/LED.2012.2210856

Guan, X., Yu, S., and Wong, H. S. P. (2012b). On the switching parameter variation of metal-oxide RRAM-part I: physical modeling and simulation methodology. *IEEE Trans. Electron. Dev.* 59, 1172–1182. doi: 10.1109/TED.2012.2184545

Han, F., Wiercigroch, M., Fang, J. A., and Wang, Z. (2011). Excitement and synchronization of small-world neuronal networks with short-term synaptic plasticity. *Int. J. Neural. Syst.* 21, 415–425. doi: 10.1142/S0129065711002924

He, W., Huang, K., Ning, N., Ramanathan, K., Li, G., Jiang, Y., et al. (2014). Enabling an integrated rate-temporal learning scheme on memristor. *Sci. Rep.* 4:4755. doi: 10.1038/srep04755

Hebb, D. O. (1949). *The Organization of Behavior: A Neuropsychological Theory*. New York, NY: John Wiley and Sons.

Huntley, J., Bor, D., Hampshire, A., Owen, A., and Howard, R. (2011). Working memory task performance and chunking in early Alzheimers disease. *Brit. J. Psychiat.* 198, 398–403. doi: 10.1192/bjp.bp.110.083857

Jackson, B. L., Rajendran, B., Corrado, G. S., Breitwisch, M., Burr, G. W., Cheek, R., et al. (2013). Nanoscale electronic synapses using phase change devices. *ACM J. Emerg. Tech. Com.* 9:12. doi: 10.1145/2463585.2463588

Jo, S. H., Chang, T., Ebong, I., Bhadviya, B. B., Mazumder, P., and Lu, W. (2010). Nanoscale memristor device as synapse in neuromorphic systems. *Nano Lett.* 10, 1297–1301. doi: 10.1021/nl904092h

Kim, H., Sah, M. P., Yang, C., Roska, T., and Chua, L. O. (2012). Memristor bridge synapses. *Proc. IEEE* 100, 2061–2070. doi: 10.1109/JPROC.2011.2166749

Kreiman, G., Fried, I., and Koch, C. (2001). Single neuron responses in humans during binocular rivalry and flash suppression. *J. Vision* 1:131. doi: 10.1167/1.3.131

Kreiman, G., Koch, C., and Fried, I. (2000). Category-specific visual responses of single neurons in the human medial temporal lobe. *Nat. Neurosci.* 3, 946–953. doi: 10.1038/78868

Kuzum, D., Jeyasingh, R. G., Lee, B., and Wong, H. S. (2011). Nanoelectronic programmable synapses based on phase change materials for brain-inspired computing. *Nano Lett.* 12, 2179–2186. doi: 10.1021/nl201040y

Kuzum, D., Jeyasingh, R. G. D., Yu, S., and Wong, H. S. (2012). Low-energy robust neuromorphic computation using synaptic devices. *IEEE Trans. Electron. Dev.* 59, 3489–3494. doi: 10.1109/TED.2012.2217146

Kuzum, D., Yu, S., and Wong, H. S. P. (2013). Synaptic electronics: materials, devices and applications. *Nanotechnology* 24:382001. doi: 10.1088/0957-4484/24/38/382001

Li, H., Li, K. S., Lin, C. H., Hsu, J. L., Chiu, W. C., Chen, M. C., et al. (2016). "Four-layer 3D vertical RRAM integrated with FinFET as a versatile computing unit for brain-inspired cognitive information processing," in *IEEE Symposium on VLSI Technology* (Honolulu, HI). doi: 10.1109/vlsit.2016.7573431

Li, B., Shan, Y., Hu, M., Wang, Y., Chen, Y., and Yang, H. (2013). "Memristor-based approximated computation," in *Proceedings of the 2013 International Symposium on Low Power Electronics and Design (ISLPED)* (Beijing: IEEE Press), 242–247. doi: 10.1109/islped.2013.6629302

Lindley, R. H. (1966). Recoding as a function of chunking and meaningfulness. *Psychon. Sci.* 6, 393–394. doi: 10.3758/BF03330953

Livio, M. (2008). *The Golden Ratio: the Story of Phi, the World's Most Astonishing Number*. Random House LLC.

Maass, W. (2014). Noise as a resource for computation and learning in networks of spiking neurons. *Proc. IEEE* 102, 860–880. doi: 10.1109/JPROC.2014.2310593

Mante, V., Sussillo, D., Shenoy, K. V., and Newsome., W. T. (2013). Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature* 503, 78–84. doi: 10.1038/nature12742

Mead, C. (1989). *Analog VLSI Implementation of Neural Systems*. Portland: Addison-Wesley.

Miller, G. A. (1956). The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychol. Rev.* 63, 81–97. doi: 10.1037/h0043158

Pasupathy, A., and Connor, C. E. (2002). Population coding of shape in area V4. *Nat. Neurosci.* 5, 1332–1338. doi: 10.1038/972

Prezioso, M., Merrikh-Bayat, F., Hoskins, B. D., Adam, G. C., Likharev, K. K., and Strukov, D. B. (2015). Training and operation of an integrated neuromorphic network based on metal-oxide memristors. *Nature* 521, 61–64. doi: 10.1038/nature14441

Rabinovich, M. I., Huerta, R., and Laurent, G. (2008). Transient dynamics for neural processing. *Science* 321, 48–50. doi: 10.1126/science.1155564

Rabinovich, M. I., Huerta, R., Varona, P., and Afraimovich, V. S. (2008). Transient cognitive dynamics, metastability, and decision making. *PLoS Comput. Biol.* 4:e1000072. doi: 10.1371/journal.pcbi.1000072

Rabinovich, M. I., Varona, P., Tristan, I., and Afraimovich, V. S. (2014). Chunking dynamics: heteroclinics in mind. *Front. Comput. Neurosci.* 8:22. doi: 10.3389/fncom.2014.00022

Rabinovich, M., Volkovskii, A., Lecanda, P., Huerta, R., Abarbanel, H. D. I., and Laurent, G. (2001). Dynamical encoding by networks of competing neuron groups: winnerless competition. *Phys. Rev. Lett.* 87, 0681021–0681024. doi: 10.1103/PhysRevLett.87.068102

Ramanathan, K., Ning, N., Dhanasekar, D., Li, G., Shi, L., and Vadakkepat, P. (2012). Presynaptic learning and memory with a persistent firing neuron and a habituating synapse: a model of short term persistent habituation. *Int. J. Neural Syst.* 22, 12500151-125001520. doi: 10.1142/S0129065712500153

Shouval, H. Z., Castellani, G. C., Blais, B. S., Yeung, L. C., and Cooper, L. N. (2002). Converging evidence for a simplified biophysical model of synaptic plasticity. *Biol. Cybern.* 87, 383–391. doi: 10.1007/s00422-002-0362-x

Simon, H. A. (1974). How big is a chunk. *Science* 183, 482–488. doi: 10.1126/science.183.4124.482

Snider, G. S. (2008). Spike-timing-dependent learning in memristive nanodevices. *IEEE Int. Symposium Nanoscale Architect.* 2008, 85–92. doi: 10.1109/nanoarch.2008.4585796

Song, S., Miller, K. D., and Abbott, L. F. (2000). Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. *Nat. Neurosci.* 3, 919–926. doi: 10.1038/78829

Song, S., Sjöström, P. J., Reigl, M., Nelson, S., and Chklovskii, D. B. (2005). Highly nonrandom features of synaptic connectivity in local cortical circuits. *PLoS Biol.* 3:e68. doi: 10.1371/journal.pbio.0030068

Strukov, D. B., Snider, G. S., Stewart, D. R., and Williams, R. S. (2008). The missing memristor found. *Nature* 453, 80–83. doi: 10.1038/nature06932

Suri, M., Bichler, O., Querlioz, D., Traor, B., Cueto, O., Perniola, L., et al. (2012). Physical aspects of low power synapses based on phase change memory devices. *J. Appl. Phys.* 112:054904. doi: 10.1063/1.4749411

Tan, A. H., and Soon, H. S. (1996). Concept hierarchy memory model: a neural architecture for conceptual knowledge representation, learning, and commonsense reasoning. *Int. J. Neural Syst.* 7, 305–319. doi: 10.1142/S0129065796000270

Teramae, J., and Fukai, T. (2014). Computational implications of lognormally distributed synaptic weights. *Proc. IEEE* 102, 500–512. doi: 10.1109/JPROC.2014.2306254

Tognoli, E., and Kelso, J. A. (2014). The metastable brain. *Neuron* 81, 35–48. doi: 10.1016/j.neuron.2013.12.022

Varona, P., and Rabinovich, M. I. (2016). Hierarchical dynamics of informational patterns and decision-making. *Proc. R. Soc. B* 283:20160475. doi: 10.1098/rspb.2016.0475

Wong, H. S. P., Lee, H. Y., Yu, S., Chen, Y. S., Wu, Y., Chen, P. S., et al. (2012). Metal-oxide RRAM. *Proc. IEEE* 100, 1951–1970. doi: 10.1109/JPROC.2012.2190369

Yang, J. J., Pickett, M. D., Li, X., Ohlberg, D. A., Stewart, D. R., and Williams, R. S. (2008). Memristive switching mechanism for metal/oxide/metal nanodevices. *Nat. Nanotechnol.* 3, 429–433. doi: 10.1038/nnano.2008.160

Yang, J. J., Strukov, D. B., and Stewart, D. R. (2013). Memristive devices for computing. *Nat. Nanotechnol.* 8, 13–24. doi: 10.1038/nnano.2012.240

Yu, S., Chen, H. Y., Gao, B., Kang, J., and Wong, H. S. P. (2013). HfOx-based vertical resistive switching random access memory suitable for bit-cost-effective three-dimensional cross-point architecture. *ACS Nano* 7, 2320–2325. doi: 10.1021/nn305510u

Yu, S., Gao, B., Fang, Z., Yu, H., Kang, J., and Wong, H. S. P. (2013). A low energy oxide-based electronic synaptic device for neuromorphic visual systems with tolerance to device variation. *Adv. Mater.* 25, 1774–1779. doi: 10.1002/adma.201203680

Yu, S., Guan, X., and Wong, H. S. P. (2012). On the switching parameter variation of metal oxide RRAM-part II: model corroboration and device design strategy. *IEEE Trans. Electron. Dev.* 59, 1183–1188. doi: 10.1109/TED.2012.2184544

Yu, S., Wu, Y., Jeyasingh, R., Kuzum, D., and Wong, H. S. P. (2011). An electronic synapse device based on metal oxide resistive switching memory for neuromorphic computation. *IEEE Trans. Elec. Dev.* 58, 2729–2737. doi: 10.1109/TED.2011.2147791

Zeng, H., Chattarji, S., Barbarosie, M., Rondi-Reig, L., Philpot, B. D., Miyakawa, T., et al. (2001). Forebrain-specific calcineurin knockout selectively impairs bidirectional synaptic plasticity and working/episodic-like memory. *Cell* 107, 617–629. doi: 10.1016/S0092-8674(01)00585-2

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Computational Neuroscience

ORIGINAL RESEARCH
published: 10 January 2017
doi: 10.3389/fncom.2016.00144

On the Maximum Storage Capacity of the Hopfield Model

Viola Folli[1]*, *Marco Leonetti*[1] and *Giancarlo Ruocco*[1, 2]

[1] Center for Life Nanoscience, Istituto Italiano di Tecnologia, Rome, Italy, [2] Department of Physics, Sapienza University of Rome, Rome, Italy

Recurrent neural networks (RNN) have traditionally been of great interest for their capacity to store memories. In past years, several works have been devoted to determine the maximum storage capacity of RNN, especially for the case of the Hopfield network, the most popular kind of RNN. Analyzing the thermodynamic limit of the statistical properties of the Hamiltonian corresponding to the Hopfield neural network, it has been shown in the literature that the retrieval errors diverge when the number of stored memory patterns (P) exceeds a fraction ($\approx 14\%$) of the network size N. In this paper, we study the storage performance of a generalized Hopfield model, where the diagonal elements of the connection matrix are allowed to be different from zero. We investigate this model at finite N. We give an analytical expression for the number of retrieval errors and show that, by increasing the number of stored patterns over a certain threshold, the errors start to decrease and reach values below unit for $P \gg N$. We demonstrate that the strongest trade-off between efficiency and effectiveness relies on the number of patterns (P) that are stored in the network by appropriately fixing the connection weights. When $P \gg N$ and the diagonal elements of the adjacency matrix are not forced to be zero, the optimal storage capacity is obtained with a number of stored memories much larger than previously reported. This theory paves the way to the design of RNN with high storage capacity and able to retrieve the desired pattern without distortions.

Keywords: maximum storage memory, feed-forward structure, random recurrent network, Hopfield model, retrieval error

OPEN ACCESS

Edited by:
Marcel Van Gerven, Radboud University Nijmegen, Netherlands

Reviewed by:
Simon R. Schultz, Imperial College London, UK
Fleur Zeldenrust, Donders Institute for Brain, Cognition and Behaviour, Netherlands

***Correspondence:**
Viola Folli
viola.folli@iit.it

Received: *19 September 2016*
Accepted: *20 December 2016*
Published: *10 January 2017*

Citation:
Folli V, Leonetti M and Ruocco G (2017) On the Maximum Storage Capacity of the Hopfield Model. Front. Comput. Neurosci. 10:144. doi: 10.3389/fncom.2016.00144

1. INTRODUCTION

A vast amount of literature deals with neural networks, both as model for brain functioning (Amit, 1989), and as smart artificial systems for practical applications in computation and information handling (Haykin, 1999).

Among the different possible applications of artificial neural networks, those referred to as "associative memory" are particularly important (Rojas, 1996), i.e., circuits with the capability to store and retrieve specific information patterns. According to Amit et al. (1985a,b) there is a natural limit for the usage of an N nodes neural network built according to the Hebbian principle (Hebb, 1949) as associative memory. The association is embedded within the connection matrix which has a dyadic form: the weight connecting neuron i to neuron j is the product of the respective signals. The limit of storage is linear with N: an attempt to store a number P of memory elements larger than $\alpha_c N$, with $\alpha_c \approx 0.14$, results in a "divergent" (order P) number of retrieval errors. In order to be effective (low retrieval error probability) a neural network working as associative memory cannot be efficient (i.e., it can store only a small number of memory elements). This is particularly frustrating in practical applications, as it strongly limits the use of artificial neural networks for information

storage, especially since it is well known that the number of fixed points in randomly connected (symmetric) neural networks shows an exponential relation with N (Tanaka and Edwards, 1980; Sompolinsky et al., 1988; Wainrib and Touboul, 2013).

Contemporaneous to Amit et al., Abu-Mostafa, and St. Jaques (Abu-Mostafa et al., 1985) claimed that the number of fixed points that can be used for memory storage in a Hopfield model with a generic coupling matrix is limited to N (i.e., $P<N$). Soon after, Mc Eliece et al. (1987), considering only the Hebbian dyadic form for the coupling matrix, found a more severe limitation: the maximum P scales as $N/\log(N)$. In a more recent study, Sollacher et al. (2009) designed a network of specific topology, reaching α_c-values larger than 0.14, but still maintaining the limit of a linear N dependence of the maximum storage capacity. The storage problem remains an open research question (Brunel, 2016).

In this letter we show that the existence of a critical P/N-value in the Hebbian scheme for the coupling matrix is only part of the story. As demonstrated in Amit et al. (1985a,b), the limit $P<\alpha_c N$ holds in the region where $P<N$. In all previous studies, the diagonal elements are removed from the dyadic form of the coupling matrix. Here we show the existence of a not yet explored region in the parameter space, with $P \gg N$, where the number of retrieval errors decreases with increasing P and reaches values lower than one. This region can be found by not removing the diagonal elements. Strictly speaking the present model is not a "Hopfield model," as in the latter case the diagonal elements are forced to vanish and—as we will see- bring significant differences in the network behavior. In order to avoid confusion, let us call the present model as "Hopfield model with autapses" or "Generalized Hopfield model." This strategy allows the design of effective and efficient associative memories based on artificial neural networks. In the following we will derive analytically the probability of retrieval errors, validate these results by their comparison with a numerical simulation and study the efficiency of the system as a function of P and N.

2. METHODS

2.1. Network Model

In an artificial neural network working as associative memory, one deals with a network of N neurons of which each one has state s_i ($i = 1...N$) that can be "active" ($s_i = 1$) or "quiescent" ($s_i = -1$). The configuration of the whole network is given by the vector $\bar{s} \equiv \{s_1, s_2, ..s_N\}$ and its temporal evolution follows the parallel non-linear dynamics:

$$s_i(t + \Delta t) = E[s_i(t)] \doteq \text{sign}\Big[\sum_{j=1}^{N} J_{ij} s_j(t)\Big], \tag{1}$$

where $\mathbf{J} = \{J_{ij}\}$ is the connection matrix. We set external inputs to be equal to 0. We assume a symmetric bimodal distribution for the synaptic polarities in the wiring matrix $\mathbf{J}$, so 50% of the connections are excitatory and 50% inhibitory. After a transient time related to the finite value of N, the network reaches a fixed point, $s_i = E[s_i]$, or a limit cycle of length L, $s_i = E^{(L)}[s_i]$.

2.2. The Hebbian Rule and the Storage Memory

Previous work has studied the cycle length and transient time distribution as a function of the properties of $\mathbf{J}$ (Gutfreundt et al., 1988; Sompolinsky et al., 1988; Derrida, 1989; Bastolla et al., 1997). In order to work as an associative memory, the matrix $\mathbf{J}$ must be tailored in such a way that one or more patterns of neurons are fixed points of the dynamics in Equation (1), i.e., they are the "memory elements" stored in the network. To store one pattern $\bar{\xi}$, the connection matrix is simply the dyadic form given by $J_{ij} = \xi_i \xi_j$ [1], while to store a generic number P of patterns $\bar{\xi}^\mu$ ($\mu = 1...P$) one follows the storage prescription of Cooper (1973) and Cooper et al. (1979), who exploited an old idea which goes back to Hebb (1949) and Eccles (1953) and which states that the change in synaptic transmission is proportional to the product of the signals of pre and post-synaptic neurons. The process for which each matrix element is appropriately determined is called *learning*. Specifically, the "Hebbian" rule results in the following expression for the connectivity matrix,

$$J_{ij} = \frac{1}{P} \sum_{\mu=1}^{P} \xi_i^\mu \xi_j^\mu. \tag{2}$$

The set of vectors $\bar{\xi}^m u$ is known as "training set." In this case, it is not guaranteed that each $\bar{\xi}^\mu$ is a fixed point. In other words, $\bar{\xi}^\mu$ is stable in probabilistic sense. Further, the probability for $\bar{\xi}^\mu$ to be a fixed point depends on the values of P and N. This dependence has been first studied by Hopfield (1982); Hopfield et al. (1983); Hopfield (1984) who concluded that the retrieval of the memory stored in the Hebbian matrices is guaranteed up to a P-value which is a critical fraction on the number of network nodes N of the order of 10–20%. Above this value, the associative memory quickly degrades. Following these studies, Amit et al. (1985a,b), who noticed the similarity between the Hopfield model for the associative memory and the spin glasses, developed a statistical theory for the determination of the critical P/N ratio, that turned out to be ≈ 0.14, in good agreement with the previous Hopfield estimation. Above P=0.14N the number of errors is so large that the network based on the Hebbian matrix is no longer capable to work as an associative memory. All these studies assumed a modified form of Equation (2): the diagonal elements of $\mathbf{J}$ are forced to be zero.

2.3. Numerical Simulations and Data Analysis

In order to demonstrate the validity of our analytic results (see Section 3), we perform numerical simulations by evolving the network model as described in Equation (1). We design the default network by fixing the NxN recurrent connections as given in Equation (2), by randomly assigning the value ± 1 to ξ_i^μ and retaining the diagonal elements. So, the $N(N-1)$ connections are 50% excitatory and 50% inhibitory and the N neurons can form self-connections. We then run simulations by varying the size of the network, $N = 50, .., 200$ and the number of stored memories

[1] The evolution $E[\xi_i]$ always return ξ_i since the sum $\sum_j \xi_j^2 = N$.

in Equation (2), $P = 1, ..., 2000$. Finally, for each pair of N and P, we perform 1000 different random realizations.

All P patterns introduced in Equation (2) are given as input to the network and their dynamics is followed until the network reaches the equilibrium state. The initial patterns are chosen among those that were stored in the adjacency matrix and that have been randomly chosen in the designing of the network. Evolved patterns were recorded at each time step and compared with the initial one. Then, if the evolved pattern is different from the initial state, we calculated the temporal evolution of the distortion (number of wrong bits) and determined the probability that one of the bits was wrong, the probability that the whole vector was exactly recovered, and the number of memory patterns that could not be recovered, as a function of N and P. Basically, to calculate the storage capacity, it is sufficient to determine all these quantities by using the distortion between the stimulus (the stored memory) and its first evolved pattern.

3. RESULTS

3.1. The Probability of Recovery

In order to investigate the maximum storage memory of our model, we calculate the one-step dynamical evolution. We give as input a vector of the training set and we calculate a single step of the dynamical evolution according to Equation (1). Then, we compare the output with the input. We aim to look whether or not a vector, $\bar{\xi}^{\mu}$, belonging to the training set, is truly a fixed point. If $\bar{\xi}^{\mu}$ is a fixed point, the output coincides with the input, and the recovery has been successful. If $\bar{\xi}^{\mu}$ is not a fixed point, the two vectors differ for at least a single bit. We now derive an analytical expression for the probability that the recovery of a stored pattern was not successful. The first step is to find the probability p_B that -given the matrix $\mathbf{J}$ of Equation (2)- a single element of the vector (a "bit") was wrong, i.e., the probability that $E[\xi_i^{\mu}] \neq \xi_i^{\mu}$. Basically, we need to evolve a vector $\bar{\xi}^{m}u$ (from the training set) for one step and count how many bits of its time evolution are different from the bits of $\bar{\xi}^{m}u$ itself. Obviously, if $\bar{\xi}^{m}u$ actually is a fixed point, this distance vanishes. On the contrary, $\bar{\xi}^{m}u$ is not a fixed point, the network has made a recovery error. Thus, p_B (or better, p_V, as we see in the next paragraph) measures "how many" training set vectors are not fixed points. The argument of the *sign* function in Equation (1) is $A_i^{\mu} = \sum_{j=1}^{N}\sum_{\nu=1}^{P}\xi_i^{\nu}\xi_j^{\nu}\xi_j^{\mu}$, this contains NP terms among which there are $N+P-1$ terms (those with $j=i$ [2] and those with $\nu=\mu$) where two out of the three ξ of the product are equals to each other ξ_i^{ν} and the third is ξ_i^{μ}. Thus $A_i^{\mu} = (N+P-1)\xi_i^{\mu} + T_i^{\mu}$, with $T_i^{\mu} = \sum_{j\neq i}^{N}\sum_{\nu\neq\mu}^{P}\xi_i^{\nu}\xi_j^{\nu}\xi_j^{\mu}$. The first term is the "coherent" one, its sign is identical to ξ_i^{μ}, and it will -if dominant- guarantee that $\bar{\xi}^{\mu}$ is a fixed point of the dynamics. The second term T_i^{μ}, on the contrary, is "noise" and its presence can either reinforce or weaken the stability of ξ_i^{μ} as fixed point. Specifically, if $|T_i^{\mu}| > (N+P-1)$ and $\mathrm{sign}(T_i^{\mu}) \neq \xi_i^{\mu}$, then the *i-th* bit of the vector $\bar{\xi}^{\mu}$ will turn out to be wrong. The quantity T_i^{μ} is the sum of $(N-1)(P-1)$ statistically independent terms, each one being $+1$ or -1. Therefore, for large enough P and N, its distribution $N(T)$ can be approximated by a gaussian with zero mean and standard deviation $\sqrt{(N-1)(P-1)}$:

$$N(T) = \frac{e^{-T^2/(2(N-1)(P-1))}}{\sqrt{2\pi(N-1)(P-1)}}. \tag{3}$$

It is now straightforward to determine the probability that $|T_i^{\mu}| > (N+P-1)$ and $\mathrm{sign}(T_i^{\mu}) \neq \xi_i^{\mu}$, thus that one of the bits of $E[\xi_i^{\mu}]$ was wrong, as $p_B = \int_{N+P-1}^{\infty} dT\, N(T)$. In conclusion:

$$p_B = \frac{1}{2}\Big[1 - \mathrm{erf}\Big(\frac{N+P-1}{\sqrt{2(N-1)(P-1)}}\Big)\Big]. \tag{4}$$

It is worth to note that this expression is symmetric under the exchange of P with N, and that for large P and N, with $P/N = 1$, it tends to $(1-\mathrm{erf}(\sqrt{2}))/2 \approx 0.02275$ which corresponds to the maximum of probability in a wrong recovery of a single bit (see **Figures 1**, **2**).

The second step is the determination of the probability p_V that one of the P vectors encoded into the connection matrix (the training set) turns out not be a fixed point. If only a single bit of the vector is wrong, the whole vector is considered "wrong." Since there are N bits that can be wrong, the probability p_V will be much higher than p_B. The calculation is straightforward, in order not to be wrong, all the bits of the vector $\bar{\xi}^{\mu}$ must be right, thus $p_V = 1-(1-p_B)^N$, therefore:

$$p_V = 1 - \Big[\frac{1}{2} + \frac{1}{2}\mathrm{erf}\Big(\frac{N+P-1}{\sqrt{2(N-1)(P-1)}}\Big)\Big]^N. \tag{5}$$

Finally, the number, N_V, of memory vectors that are not recovered, i.e., that are not true fixed points of the dynamics is given by Pp_V, that is:

$$N_V = \Big[1 - \Big[\frac{1}{2} + \frac{1}{2}\mathrm{erf}\Big(\frac{N+P-1}{\sqrt{2(N-1)(P-1)}}\Big)\Big]^N\Big]P \tag{6}$$

3.2. The Asymptotical Approximation

Equations (4), (5) and, in particular, Equation (6) represent the main result of this work. Before showing their validity, via a comparison with numerical simulations, and discussing their relevance in the framework of artificial neural networks, it is important to present the asymptotical approximation for N_V. The argument of the error function, for either $P \gg N$ or $P \ll N$, is large, and can be expanded as $\mathrm{erf}(x) \approx 1-\exp(-x^2)/(x\sqrt{\pi})$. Furthermore, as p_B is exponentially small with N (or P) for large N (P), we use, $(1-p_B)^N \approx (1-Np_B)$. Thus, for large N or large P:

$$p_V \approx \frac{N^{3/2}P^{1/2}e^{-\frac{(N+P)^2}{2NP}}}{\sqrt{2\pi}(N+P)} \tag{7}$$

$$N_V \approx \frac{N^{3/2}P^{3/2}e^{-\frac{(N+P)^2}{2NP}}}{\sqrt{2\pi}(N+P)} \tag{8}$$

We note that, while in the exact expression for N_V (Equation 6) the $P \leftrightarrow N$ exchange symmetry is lost, in the approximate form the symmetry is recovered.

[2] These are P terms that are present only if the diagonal elements are kept as they are and are not forced to vanish.

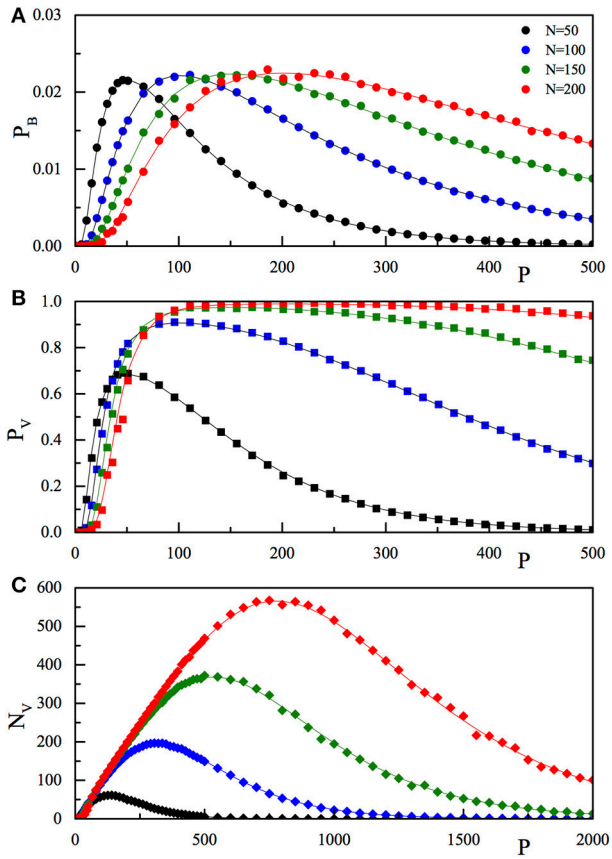


FIGURE 1 | Comparison of the results of the numerical simulation (p_B, full dots A; p_V, full squares, **B**; N_V, full diamonds, **C**) with the corresponding theoretical function (p_B, Equation 4; p_V, Equation 5; N_V, Equation 6) reported as full lines. The three quantities are reported as a function of P for fixed N. The values of N are 50 (black), 100 (blue), 150 (green), and 200 (red). The P range in **(C)** is extended with respect to **(A,B)**.

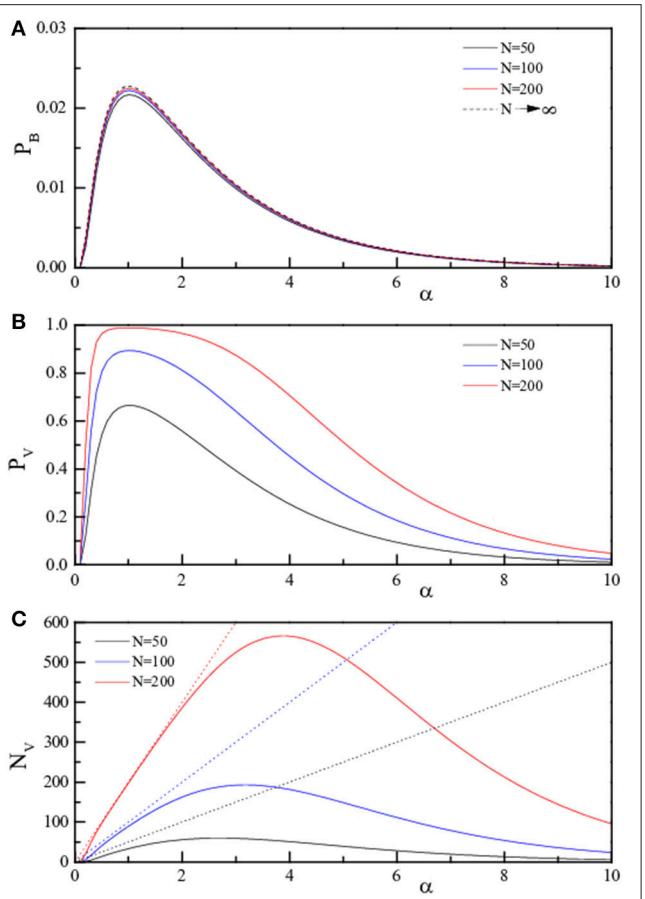


FIGURE 2 | Theoretical curves for the three quantities p_B, p_V, and N_V (p_B, Equation 4, A; p_V, Equation 5, **B**; N_V, Equation 6, **C**) reported as full lines. The three quantities are reported in linear scale as a function of $\alpha = P/N$ for fixed N. The values of N are 50 (black), 100 (blue), and 200 (red). The dotted lines in **(C)** represent $N_V = P$.

For sake of comparison with the previous literature, it is also useful to express the main results as a function of $\alpha \doteq P/N$. Equations (4) (for large N) and (8) read:

$$p_B \approx \frac{1}{2}\left[1 - \mathrm{erf}\left(\frac{1+\alpha}{\sqrt{2\alpha}}\right)\right] \tag{9}$$

$$N_V \approx NP\frac{1}{\sqrt{2\pi}}\frac{\sqrt{\alpha}}{1+\alpha}e^{-\frac{(1+\alpha)^2}{2\alpha}}. \tag{10}$$

While p_B only depends on α, N_V clearly is an extensive observable, being proportional to P and N. Furthermore, both expressions keep their symmetry with respect to the exchange of P and N, thus to the exchange of α with $1/\alpha$. The last observation anticipates that there must exists a region at large α-values where the same features are observed as at small values of α.

3.3. Numerical Results

To check the predictions of our network model, we have simulated the Model (1) and studied the dynamics for several values of N and P, in the range of few hundred, see Section 2.3 for details. In the numerical analysis, the P memory vectors have been randomly chosen and used to construct the connection matrix **J**. Next, we tested whether or not the stored memories were fixed points of the dynamics. The values of p_B, p_V and N_V were calculated by averaging over (up to) 1000 different random realizations of $\bar{\xi}^\mu$. The results of the numerical simulations are reported (dots) in **Figure 1**, together with the analytical Expressions (4)–(6) (lines). The three panels refer to the three quantities p_B (**Figure 1A**), p_V (**Figure 1B**), and N_V (**Figure 1C**) as a function of P for the selected values of N, as reported in the legend. From **Figure 1**, we observe that on increasing P, at fixed N, both the single bit probability error, the probability of recovery error (P_V), and the number of wrong recoveries N_V, after a first fast increase, reach a maximum (equal to 0.02275 for p_B, close to one for p_V, and larger than N for N_V) then start to decrease, tending to zero for very large P-values.

To better emphasize this behavior, the same quantities are reported (analytic results only) as a function of α in **Figure 2** (linear scale) and in **Figure 3** (log scale) for selected N. The dotted lines in panels C of both figures represent $N_V = P$,

i.e., indicate the case of "totally wrong recovery." Due to the already observed $\alpha \leftrightarrow 1/\alpha$ symmetry, the asymptotic curve in **Figure 3A** appears with a left-right symmetry around $\alpha = 1$. From **Figures 2**, **3**, we can clearly identify two regions of high recovery efficiency. The low α region, already studied many years ago by Hopfield (1982); Hopfield et al. (1983); Hopfield (1984) and Amit et al. (1985a,b), shows the existence of a quick transition toward "loss of memory recover" on increasing α around $\alpha \approx 0.14$. The second region at large α-values is not yet explored.

Although the value $\alpha = 1$ ($P = N$) represents traditionally a sort of limit in the computation of the storable memories in a RNN, there is no reason why not to store more than N memory elements in a network of N neurons, that by construction allows 2^N possible patterns. Indeed, the number of fixed points in a (random) symmetric matrix is known to be, for fully connected symmetric matrices as in our case, exponentially large with N (Tanaka and Edwards, 1980). Specifically, the number of fixed points P_o is equal to $P_o = \exp(\gamma N)$, with $\gamma \approx 0.2$. P_o, much larger than N, can be considered a natural limit for P.

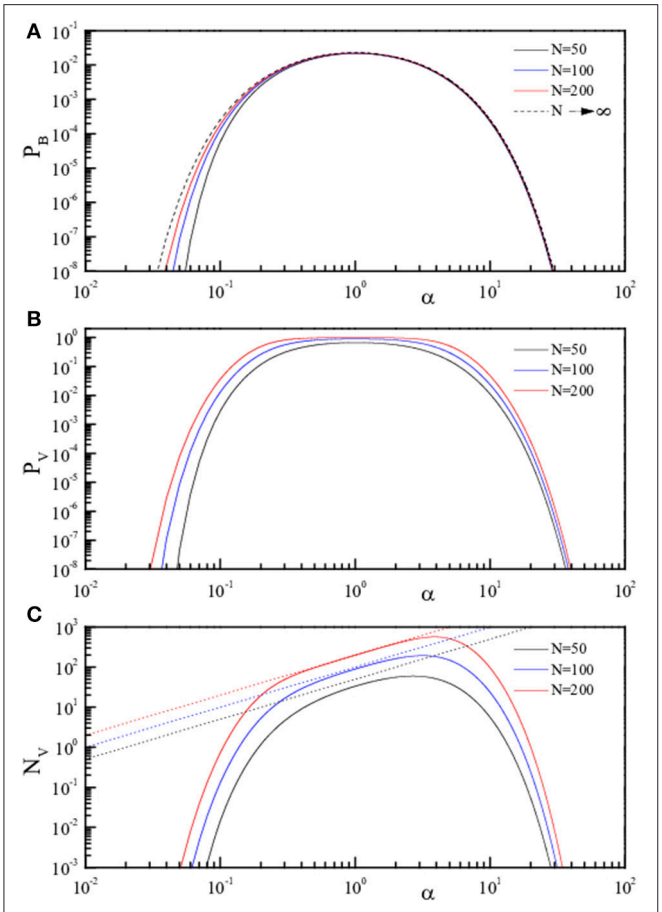


FIGURE 3 | Theoretical curves for the three quantities *pB*, *pV*, and *NV* (*pB*, Equation 4, A; *pV*, Equation 5, **B**; *NV*, Equation 6, **C**) reported in Log-Log scale as full lines. The three quantities are reported in linear scale as a function of $\alpha = P/N$ for fixed N. The values of N are 50 (black), 100 (blue), and 200 (red). The dotted lines in **(C)** represent $NV = P$.

The recovery efficiency increases for large P. In fact, the coherent term in the argument of the sign function increases linearly with P and the noise increases as $P^{1/2}$. For large P, the relative weight of the noise decreases as $P^{-1/2}$, this allows to store a large number of memories in a relatively small neural network.

For practical purposes, as for example in the design of an artificial neural network with high efficiency (large storage capacity) and effectiveness (low recovery error rates), it is important to study (Equation 6, and its approximation in Equation 8) and, in particular, to find the conditions for which the network shows "perfect recovery." Let's define perfect recovery as the state where the number of retrieval errors N_V is smaller than one.

In **Figure 4** we show the contour plot of the (decimal) logarithm of N_V, from Equations (6) and (8), in the P-N range [0–100]. The full lines are the loci of the points where $log_{10}(N_V)$ equals 0, 0.4, 0.8, 1.2, and 1.6, as indicated on the right side of the figure. The dashed lines are the same level lines for the (logarithm of the) approximate form of N_V reported in Equation (8). As can be observed, for $N_V \approx 1$, the approximation (Equation 8) for N_V is highly accurate, indicating that this approximation can be safely applied to find the "perfect recovery" condition.

In the P-N plane the existence of two regions (small and large α) where the perfect recovery ($N_V = 1$, red lines) takes place can be easily observed and the result is symmetric under the exchange of P and N. In the already explored small α region, we also show (full blue line) the $P = 0.14N$ condition. Similar to the high α region, it is important to find a simple relation between N and P identifying the $N_V = 1$ condition. We aim, therefore, to obtain a function $P(N)$ which returns, at given N, the P-value

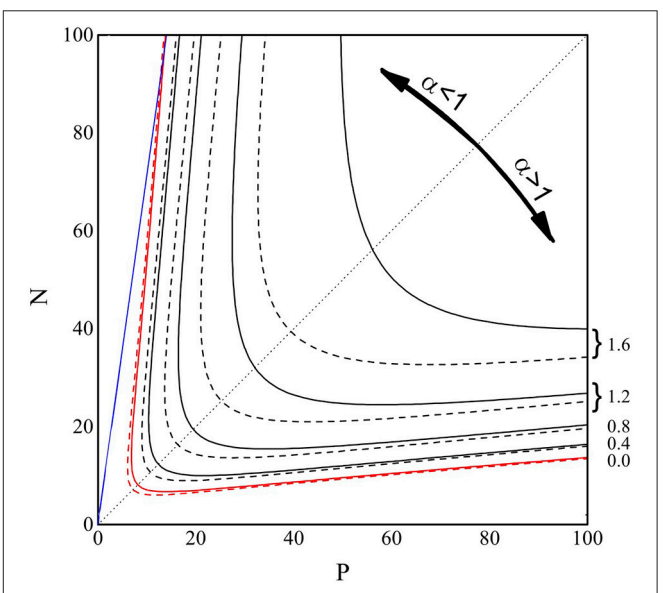


FIGURE 4 | Contour plot of $\log_{10}(N_V)$ from Equations (6) (full lines) and (8) (dashed lines), in the *P* and *N* range 0–100. The lines are the loci of the points where $log_{10}(N_V)$ equals 0.0 (red), 0.4, 0.8, 1.2, and 1.6 (black), as indicated on the right side of the figure. The blue line represents $P = 0.14N$, while the black dotted line is the bisectrix $N = P$, plotted to emphasize the symmetry of the contour lines.

such that $N_V = 1$. We write the prefactor NP in Equation (10) as αN^2 and exploit the $\alpha \gg 1$ limit, so to obtain $N_V \approx N^2\alpha^{1/2}\exp(-\alpha/2)/\sqrt{2\pi}$. The equation $N^2\alpha^{1/2}\exp(-\alpha/2)/\sqrt{2\pi} = 1$ can be squared, $\alpha\ \exp(-\alpha) = 2\pi/N^4$, and solved with respect to α, to give $\alpha = -W_{-1}(-2\pi/N^4)$, where $W_{-1}(x)$ is the second real branch of the Lambert function (Olver et al., 2010). In conclusion, the "perfect recovery condition" is satisfied -for each N-value- if we chose to store a number of memories *larger* than $P(N)$ given by:

$$P(N) = -NW_{-1}(-2\pi/N^4). \tag{11}$$

For practical purposes, for large enough N, we can use the small-argument expansion of the Lambert function $-W_{-1}(-x) \approx -\ln(x)+\ln(-\ln(x))$ (Corless et al., 1996), to have:

$$P(N) = N\left[\ln\left(\frac{N^4}{2\pi}\right) + \ln\left(\ln\left(\frac{N^4}{2\pi}\right)\right)\right]. \tag{12}$$

The results for $P(N)$ are shown in **Figure 5** as a function of N in the range 1–1000. The black line represents the exact, numerical, solution to $N_V = 1$, with N_V in Equation (6), the blue line is the expression for $P(N)$ in Equation (11), while the red line is those in Equation (12).

It is important to note that the presence of a decrease of the retrieval error probability at high P, or α, values is due to the presence of non-zero diagonal elements in the **J** matrix that creates a coherent term of weight P. Indeed, repeating the rationale leading to Equation (4) with the assumption that $J_{ii} = 0$, would give rise to the same (Equations 4–6) but with the numerator of the argument of the error functions equal to $N - 1$ instead of to $N + P - 1$. This is shown graphically in **Figure 6** where we compare for $N = 50$, both theoretically (full line) and numerically (full dots), the quantities p_B, p_V, and N_V as a function of P in the two cases: diagonal elements in Equation (2) (black) and diagonal elements forced to vanish (orange).

The stabilization of the fixed points $\bar{\xi}^\mu$ in the high storage region arises from the presence of the non-zero diagonal elements. Asymptotically, on increasing P, the diagonal elements growth coherently and the **J** matrix tends to become the unit matrix. However, the dynamics (see Equation 1) dictated by the matrix J does not tend to the dynamics dictated by the unit matrix. In the latter case, indeed, all the 2^N state vectors should become fixed points and the network should loose on important feature: the capability to distinguish between the stored memories (the vectors $\bar{\xi}^\mu$, for $\mu = 1...P$) and the spurious fixed points, all the vectors $\bar{\zeta}$ not belonging to the set $\bar{\xi}^\mu$ but such that $E[\bar{\zeta}] = \bar{\zeta}$. To study this property, we have calculated the probability that a (randomly chosen) vector $\bar{\zeta}$ (different from all the $\bar{\xi}^\mu$ used

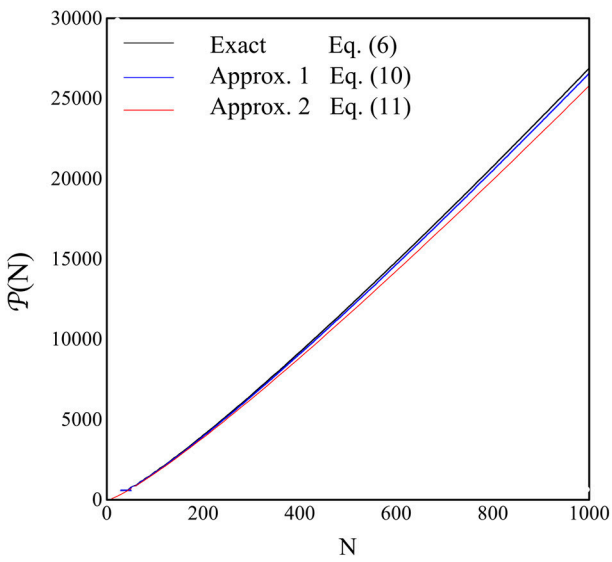


FIGURE 5 | The quantity *P*(*N*), i.e., the *P*-value where the perfect recovery is guaranteed, is shown as a function of *N*. The blue line is the numerical solution of $N_V = 1$ from Equation (6), the blue line is the plot of Equation (11) and the red line is the plot of Equation (12).

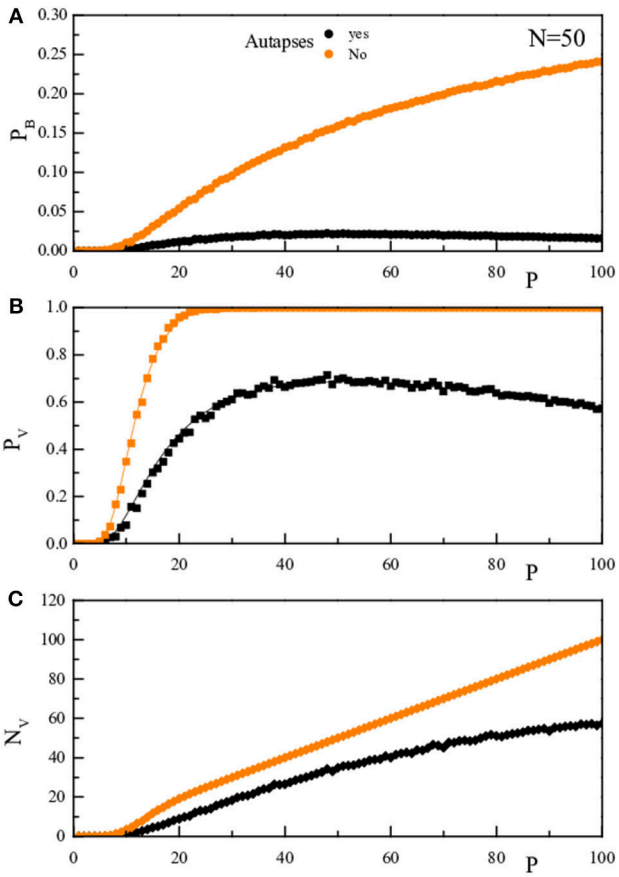


FIGURE 6 | The upper panel (A) reports for a given *N*-value (*N* = 50), as a function of P, the probability p_B that, stimulating the network with a vector inside the training set, there is one bit wrong in the network response. The middle panel **(B)** reports p_V, the probability that, stimulating the network with a vector inside the training set, the vector obtained after one dynamical step is not the stimulating vector. The lower panel **(C)** reports $N_V = Pp_V$. The black symbols/lines refer to the case where the diagonal elements are as determined in Equation (2), while the oranges ones to diagonal elements forced to vanish. The full lines are the theoretical prediction, the full dots are the results of the numerical simulation.

to build the **J** matrix) was recognized as a "memory" from the network dynamics. To be consistent with the previous notation (where we called p_B and p_V the probability of errors, not that of correct retrieval of the memory states) we define $\bar{p}_B$ ($\bar{p}_V$) as the probability of correctly not retrieving a vector not belonging to the training set. More specifically, the quantity $\bar{p}_V$ is the probability that one dynamical step after presenting a vector ζ not belonging to the training set to the network, the output a vector is different from ζ." More specifically, the quantity $\bar{p}_V$ is the probability that presenting a vector ζ not belonging to the training set to the network, after one dynamical step we found as output a vector different from ζ. Similarly for $\bar{p}_B$. It turns out that [3]:

$$\bar{p}_B = \frac{1}{2}\Big[1 - \mathrm{erf}\Big(\frac{P}{\sqrt{2(N-1)(P-1)}}\Big)\Big] \tag{13}$$

$$\bar{p}_V = 1 - \Big[\frac{1}{2} + \frac{1}{2}\mathrm{erf}\Big(\frac{P}{\sqrt{2(N-1)(P-1)}}\Big)\Big]^N. \tag{14}$$

In **Figure 7** we report the comparison of the P dependence of p_B and $\bar{p}_B$ (**Figure 7A**) and that of p_V and $\bar{p}_V$ (**Figure 7B**). As usual, full lines are the theoretical results, while the full dots are the outcome of the numerical simulation. Black data are for the "memory states," while the green ones are for the "spurious state." As can be seen, the spurious state becomes more and more "present" in the set of memories stored by the network as P increases. It seems however that also at high P-values the retrieval of the memory states is reasonably good and that of the spurious states reasonably bad.

To be quantitative on this point, we rewrite Equation (14) in its large N limit:

$$\bar{p}_V \approx \frac{N^{3/2}P^{-1/2}e^{-\frac{P}{2N}}}{\sqrt{2\pi}} \tag{15}$$

and compare it with Equation (7). In particular, is interesting to calculate the ratio, ρ, between the probability of wrong retrieval of a spurious state and that of a memory state: $\rho = \bar{p}_V/p_V$. From Equations (7) and (15) it turns out:

$$\rho = \Big(\frac{N+P}{P}\Big)\ e^{\frac{(N+P)^2}{2NP}} e^{-\frac{P}{2N}}. \tag{16}$$

This quantity only depends on α:

$$\rho = \Big(\frac{1+\alpha}{\alpha}\Big)\ e^{\left(1+\frac{1}{2\alpha}\right)} \tag{17}$$

and ρ has a finite high α limit:

$$\lim_{\alpha\to\infty} \rho = e. \tag{18}$$

In other words, although the number of spurious attractors tends to increase for $P \gg N$, the vectors encoded into the system through the connection matrix are retrieved with an efficiency almost three times better than for the spurious states.

[3]The calculation follows the same steps already depicted before, counting the "coherent" terms, that, in this case, only arise from the diagonal elements (i=j) and not from the μ=ν terms that now do not exist. The weight of the coherent part is equal to P instead of $N + P - 1$. The rest of the demonstration follows straightforward.

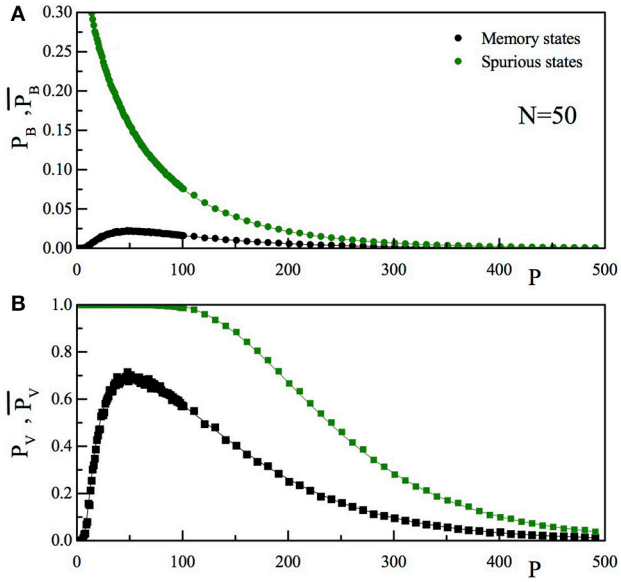


FIGURE 7 | (A) The upper panel reports for a given N-value ($N = 50$), as a function of P, the probability that, stimulating the network with a vector inside (p_B, black) or outside ($\bar{p}_B$, green) the training set, there is one bit differing between the input and the output vector. **(B)** The lower panel reports the probability that, stimulating the network with a vector inside (p_V, black) or outside ($\bar{p}_V$, green) the training set, the vector obtained after one dynamical step is not the stimulating vector. The full lines are the theoretical prediction, the full dots are the results of the numerical simulation.

4. DISCUSSION

In this work we have developed a simple theoretical approach to investigate the computational properties and the storage capacity of feed-forward networks with self-connections. We have worked out an exact expression which gives the probability p_B of having a wrong bit in the recovery of a memory element from a Hebbian N-node neural network, where P memory elements are stored. In disagreement with previous studies we have investigated the case in which the diagonal elements were not forced to vanish. Studying the storage capacity, and deriving the related probability p_V and number N_V of having a wrongly recovered memory element, we discovered that besides the well know $P \ll N$ region, there is another region, at $P \gg N$, where the recovery is highly effective. When $P \gg N$, the efficiency of recall for a large number of encoded vectors in the J matrix is related to the presence of non-zero diagonal elements of the matrix. Basically, the higher storage performance of the network depends on the number of "coherent" terms (the signal) in the quantity A_i^μ (see Section 3.1) with respect to the "incoherent" ones (the noise). The larger the ratio between coherent to incoherent terms, the lower the probability of a wrong recovery. The number of coherent terms is $(N + P - 1)$ in the case of autapses, it is $(N - 1)$ in the case of no autapses. Indeed, the P terms disappear if the diagonal is forced to be zero as in the standard Hopfield model. It is clear that, apart from a transient regime at $P \sim N$, increasing $P \gg N$ strongly reinforces the signal-to-noise ratio and induces a much larger storage capacity. In addition to the vectors encoded into

the system, other unwanted memories also appear in the network. These are the spurious states, fixed points which do not belong to the training set. The presence of spurious states is not a feature specific to the present model, it is a typical characteristic of the standard Hopfield network and its successive improvements. Indeed, as shown by Tanaka and Edwards (1980), a random $N \times N$ matrix has $2^{\gamma N}$ fixed points ($\gamma \approx 2$). As an example, if $N = 100$, the number of fixed points is about one million. A Hebbian 100×100 matrix storing $P = 1000$ patterns, besides the "good" P fixed points have also an overwhelming number of spurious fixed points (or "false memories"). The interest of our approach does not rely in "how many" spurious (i.e., not belonging to the training set) states are present but rather in how the recognition of a vector belonging to the training set is as a "good" one. Obviously, the argument of Tanaka-Edwards applies only to random matrices. The Hebbian form, with or without autapses, is not fully random (there exists correlation among the matrix elements), but we expect a number of fixed points similar to that of a random matrix. It would be interesting to determine such a number, but this is beyond the scope of the present paper. In spite of the overwhelming majority of spurious fixed points, the network—even at very large P-values, maintains the capacity of discriminate between "good" state (belonging to the training set) and "wrong" ones (not belonging to the training set). More specifically, looking at the one-step dynamical evolution and comparing the input vector with the output one, we have posed to the network the question: "is the input vector belonging to the training set"? We have demonstrated that, when the input vector actually belongs to the training set, at large P (similarly to low P) the probability of having a wrong response ("no, it does not belong to the training set) goes to zero. Furthermore, we have demonstrated that when the input vector does not belong to the training set the probability of a wrong response ("yes, it is a fixed point") is much less that in the previous case, asymptotically 2.7 time worst.

In order to identify whether or not a vector belonging to the training set was a fixed point we propose to the system a vector of the training set as input. Then we perform a one-step dynamic evolution of this input state. If after one step the output vector is equal to the input one, this is a fixed point. On the contrary, if after one step the output vector is not equal to the input one, it could be possible that further dynamical steps lead to the input vector. From this point on, as the dynamic is deterministic, the system enters a limit cycle (of length greater than one). Since it is not clear whether or not a limit cycle can be considered a "right recognition," we have excluded this possibility from the counts of the right recognition. Only fixed point are considered "good." For this reason, to determine the probability of "right recognition" one dynamical step is enough. We have also not considered the possibility that, using as input a vector not belonging to the training set, it converges to one of the training vectors. The probability of right recognition reported here is an underestimation of the network capability. A further quantity that it would be interesting to evaluate is the size of the attraction basin of a given fixed point, i.e., how many non-training vectors converge to a given training vector fixed point. The basins size would be an important measure of the network performance, their determination is however difficult to achieve analytically, and is behind the goal of the present paper.

One important finding is summarized in Equation (18). It states that for $P \gg N$, when the connection matrix is dominated by the diagonal term and is still different from the unity matrix (this is due to the great number of off-diagonal elements with zero average and RMS of the order of $1/\sqrt{P}$), the network retains its capacity of give more "good" than "wrong" answers. This property, the fact that the limit in Equation (18) is e and not "1," can be ascribed to the observation that, although the matrix **J** tends to the unit matrix for large P, the dynamics (see Equation 1) dictated by the matrix J does not tend to the dynamics dictated by the unit matrix. This finding opens the way to a much more efficient use of the artificial Hebbian neural network for information storage. In the first region, as well known since 40 years, the storage capacity is limited as the number of encoded vectors becomes of the order N. Indeed, in the high α region, the number of elements is basically unlimited [4], when the number of stored elements is taken *larger* than $\approx 4N\ln(N)$.

AUTHOR CONTRIBUTIONS

GR designed research. VF, ML, and GR performed numerical simulations, analyzed data and wrote the manuscript.

[4]The word unlimited is obviously non-physical. However, the value of P_o arising from the Tanaka-Edwards relation (Tanaka and Edwards, 1980), $P_o = \exp(\gamma N)$, with $\gamma = 0.2$. P_o, already at the N-values reported in Equation (5) is so large ($P_o(N = 1000) \approx 10^{87}$) with respect to $P(N)$ ($P(N = 1000) \approx 2\ 10^4$ to state that P_o is unlimited to any practical purpose.

REFERENCES

Abu-Mostafa, Y. S., and St. Jacques, J.-M. (1985). Information capacity of the Hopfield model. *IEEE Trans. Inf. Theory* IT-31, 461. doi: 10.1109/tit.1985.1057069

Amit, D. J. (1989). *Modelling Brain Function: The World of Attractor Neural Networks*. Cambridge: Cambridge University Press.

Amit, D. J., Gutfreund, H., and Sompolinsky, H. (1985a). Storing infinite numbers of patterns in a spin-glass model of neural networks. *Phys. Rev. Lett.* 55:1530. doi: 10.1103/PhysRevLett.55.1530

Amit, D. J., Gutfreund, H., and Sompolinsky, H. (1985b). Spin-glass models of neural networks. *Phys. Rev. A* 32:1007. doi: 10.1103/physreva.32.1007

Bastolla, U., and Parisi, G. (1997). Attractors in fully asymmetric neural networks. *J. Phys. A Math. Gen.* 30:5613.

Brunel, N. (2016). Is cortical connectivity optimized for storing information? *Nat. Neurosci.* 19:749. doi: 10.1038/nn.4286

Cooper, L. N. (1973). "A possible organization of animal memory and learning," in *Proceedings of the Nobel Symposium on Collective Properties of Physical Systems,* eds B. Lundquist and S. Lundquist (New York, NY: Academic Press), 252–264.

Cooper, L. N., Liberman, F., and Oja, E. (1979). A theory for the acquisition and loss of neuron specificity in visual cortex. *Biol. Cybern.* 33:9. doi: 10.1007/BF00337414

Corless, R. M., Gonnet, G. H., Hare, D. E. G., Jeffrey, D. J., and Knuth, D. E. (1996). On the LambertW function. *Adv. Comput. Math.* 5:329. doi: 10.1007/BF02124750

Derrida, B. (1989). Distribution of the activities in a diluted neural network. *J. Phys. A Math. Gen.* 22:2069. doi: 10.1088/0305-4470/22/12/012

Eccles, J. G. (1953). *The Neurophysiological Basis of Mind*. Oxford: Clarendon.

Gutfreundt, H., Regert, J. D., and Young, A. P. (1988). The nature of attractors in an asymmetric spin glass with deterministic dynamics. *J. Phys. A Math. Gen.* 21:2775. doi: 10.1088/0305-4470/21/12/020

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*. Prentice Hall. Available online at: https://lecteurepub-a507a.firebaseapp.com/3RO4lXk7OXbEm12/Neural%20Networks%20A%20Comprehensive%20Foundation%202nd%20Edition%20Ebooks%20Gratuit.pdf

Hebb, D. O. (1949). *The Organization of Behavior*. New York, NY: Wiley.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proc. Nat. Acad. Sci. U.S.A.* 79:2554. doi: 10.1073/pnas.79.8.2554

Hopfield, J. J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. Nat. Acad. Sci. U.S.A.* 81:3088. doi: 10.1073/pnas.81.10.3088

Hopfield, J. J., Feinstein, D. I., and Palmer, R. G. (1983). 'Unlearning' has a stabilizing effect in collective memories *Nature* 304, 158.

Mc Eliece, R. J., Posner, E. C., Rodemich, E. R., and Venkatesh, S. S. (1987). The capacity of the hopfield associative memory. *IEEE Trans. Inf. Theory* IT-33, 461.

Olver, F. W. J., Lozier, D. W., Boisvert, R. F. and Clark, C. W. (eds.). (2010). *Handbook of Mathematical Functions*. Cambridge: Cambridge University Press.

Rojas, R. (1996). *Neural Networks*. Berlin: Springer-Verlag.

Sollacher, R., and Gao, H. (2009). Towards real-world applications of online learning spiral recurrent neural networks. *J. Intell. Learn. Syst. Appl.* 1:1. doi: 10.4236/jilsa.2009.11001

Sompolinsky, H., Crisanti, A., and Sommers, H. (1988). Chaos in random neural networks. *Phys. Rev. Lett.* 61:259. doi: 10.1103/PhysRevLett.61.259

Tanaka, F., and Edwards, S. F. (1980). Analytic theory of the ground state properties of a spin glass. I. Ising spin glass. *J. Physi. F Met. Phys.* 10:2769. doi: 10.1088/0305-4608/10/12/017

Wainrib, G., and Touboul, J. (2013). Topological and dynamical complexity of random neural networks. *Phys. Rev. Lett.* 118:101259. doi: 10.1103/physrevlett.110.118101

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Computational Neuroscience

ORIGINAL RESEARCH
published: 27 December 2016
doi: 10.3389/fncom.2016.00131

Representational Distance Learning for Deep Neural Networks

Patrick McClure and Nikolaus Kriegeskorte*

MRC Cognition and Brain Sciences Unit, Cambridge, UK

Deep neural networks (DNNs) provide useful models of visual representational transformations. We present a method that enables a DNN (student) to learn from the internal representational spaces of a reference model (teacher), which could be another DNN or, in the future, a biological brain. Representational spaces of the student and the teacher are characterized by representational distance matrices (RDMs). We propose representational distance learning (RDL), a stochastic gradient descent method that drives the RDMs of the student to approximate the RDMs of the teacher. We demonstrate that RDL is competitive with other transfer learning techniques for two publicly available benchmark computer vision datasets (MNIST and CIFAR-100), while allowing for architectural differences between student and teacher. By pulling the student's RDMs toward those of the teacher, RDL significantly improved visual classification performance when compared to baseline networks that did not use transfer learning. In the future, RDL may enable combined supervised training of deep neural networks using task constraints (e.g., images and category labels) and constraints from brain-activity measurements, so as to build models that replicate the internal representational spaces of biological brains.

Keywords: neural networks, transfer learning, distance matrices, visual perception, computational neuroscience

OPEN ACCESS

Edited by:
Marcel Van Gerven,
Radboud University Nijmegen,
Netherlands

Reviewed by:
Michael Hanke,
Otto-von-Guericke University
Magdeburg, Germany
Iris I. A. Groen,
National Institutes of Health (NIH),
USA

****Correspondence:***
Patrick McClure
patrick.mcclure@mrc-cbu.cam.ac.uk

Received: *21 July 2016*
Accepted: *29 November 2016*
Published: *27 December 2016*

Citation:
McClure P and Kriegeskorte N (2016)
Representational Distance Learning
for Deep Neural Networks.
Front. Comput. Neurosci. 10:131.
doi: 10.3389/fncom.2016.00131

1. INTRODUCTION

Deep neural networks (DNNs) have recently been highly successful for machine perception, particularly in the areas of computer vision using convolutional neural networks (CNNs) (Krizhevsky et al., 2012) and speech recognition using recurrent neural networks (RNNs) (Deng et al., 2013). The success of these methods depends on their ability to learn good, hierarchical representations for these tasks (Bengio, 2012). DNNs have not only been useful in achieving engineering goals, but also as models of computations in biological brains. Several studies have shown that DNNs trained only to perform object recognition learn representations that are similar to those found in the human ventral stream (Khaligh-Razavi and Kriegeskorte, 2014; Yamins et al., 2014; Güçlü and van Gerven, 2015). The models benefit from task training, which helps determine the large number of parameters and bring the domain knowledge required for feats of intelligence such as object recognition into the models. This is in contrast to the earlier approach in visual computational neuroscience of using nonlinear systems identification techniques to set the parameters exclusively on the basis of measured neural responses to large sets of stimuli (Naselaris et al., 2011). The latter approach is challenging for deep neural networks, because the high cost of brain-activity measurement limits the amount of data that can be acquired (Yamins and DiCarlo, 2016). Ultimately, task-based constraints will have to be combined with constraints from brain-activity measurements to model information processing in biological brains.

Here we propose a method that enables the training of DNNs with combined constraints on the desired outputs and the internal representations. We demonstrate the method by using another neural net model as the reference system whose internal representations the DNN is to emulate. One method for doing so would be to have a layer in a DNN linearly predict individual measured responses (e.g., fMRI voxels or neurons), and backpropagate the error derivatives from the linear measured-response predictors into the DNN. However, the linear measurement prediction model has a large number of parameters ($n_{units} \times n_{responses}$). An alternative approach is to constrain the DNN to replicate the representational distance matrices (RDMs) estimated from brain responses. In this paper, we take a step in that direction by considering the problem of training a DNN (student) to model the sequence of representational transformations in another artificial system (teacher), a CNN trained on different data.

Our technique falls in the class of transfer learning methods. In the deep learning literature, several such techniques have been proposed both for pulling a DNN's internal representations toward the task target and for transferring knowledge from a teacher DNN to a student DNN. We begin by briefly considering the previous approaches used to accomplish these goals.

1.1. Auxiliary Classifiers: Pulling Internal Representations Toward the Desired Output

Recently, it has been investigated how the error signal reaching an internal layer through backpropagation can be complemented by auxiliary error functions. These more directly constrain internal representations using auxiliary optimization goals. A variety of methods using auxiliary error functions to pull representations toward the desired output have been proposed.

Weston et al. (2012) proposed semi-supervised embeddings to augment the error from the output layer. A reference embedding of the inputs was used to guide representational learning. The embedding constraint was implemented in different ways: inside the network as a layer, as part of the output layer, or as an auxiliary error function that directly affected a particular hidden layer. Weston et al. discussed a variety of embedding methods that could be used, including multidimensional scaling (MDS) (Kruskal, 1964) and Laplacian Eigenmaps (Belkin and Niyogi, 2003). The addition of these semi-supervised error functions led to increased accuracy compared to DNNs trained using output layer backpropagation alone.

Lee et al. (2014) also showed that auxiliary error functions improve DNN representational learning. Instead of using semi-supervised methods, they performed classification with a softmax or L2SVM readout at a given intermediate hidden layer. The softmax layer allowed the output of a network to be treated as a probability distribution by performing normalized exponentiation on the previous layer's activations ($y_i = e^{x_i} / \sum_j e^{x_j}$). The error of the intermediate-level readout was then backpropagated to earlier layers to drive intermediate layers directly toward the target output. The gradients from these classifiers were linearly combined with the gradients from the output layer classifier. This technique resulted in improved accuracies for several datasets.

A challenge in training very deep networks is the problem of vanishing gradients. Layers far from the output may receive only a weak learning signal via conventional backpropagation. Auxiliary error functions were successfully applied to these very deep networks by Szegedy et al. (2015) to inject a complementary learning signal at internal layers by constraining representations to better discriminate between classes. This was implemented in a very large CNN which won the ILSVRC14 classification competition (Russakovsky et al., 2014). In this DNN, two auxiliary networks were used to directly backpropagate from two intermediate layers back through the main network. Similar to the method used in Lee et al. (2014), the parameters for the layers in the main network directly connected to auxiliary networks were updated using a linear combination of the backpropagated gradients from later layers and the auxiliary network.

Wang et al. (2015) investigated the effectiveness of auxiliary error functions in very large CNNs and their optimal placement. They selected where to place these auxiliary functions by measuring the average magnitude of the conventional backpropagation error signal at each layer. Auxiliary networks, similar to those used in Szegedy et al. (2015), were placed after layers with vanishing gradients. These networks consisted of a convolutional layer followed by three fully connected layers and a softmax classifier. As in Lee et al. (2014) and Szegedy et al. (2015), the auxiliary gradients were linearly combined to update the model parameters. Adding these supervised auxiliary error functions led to an improved accuracy for two very large datasets, ILSVRC12 (Russakovsky et al., 2014) and MIT Places (Zhou et al., 2014).

1.2. Transfer Learning: Pulling the Representations of a Student Toward Those of a Teacher

Enabling a student network to learn from a teacher is useful for a number of tasks, for instance model compression (also known as knowledge distillation) and transfer learning (Bengio, 2012). The goal in either case is to use the representational knowledge learned by a teacher neural network to improve the performance of a student network (Bucilua et al., 2006; Ba and Caruana, 2014; Hinton et al., 2015). For model compression, the teacher is a larger or more complex network with higher performance than the student. For knowledge transfer, the representations learned by the teacher network are used to improve the training of a student network on a different tasks or using different data. Several techniques have been proposed for performing these methods.

One technique for model compression is to have the student learn the output representation of the teacher for a given training input. For classification, the neurons before the softmax layer can be constrained to have the same values as the teacher using mean squared error (MSE) as done in Bucilua et al. (2006); Ba and Caruana (2014). Alternatively, the output of the softmax layer can be constrained to represent the same, or similar, output distribution as the teacher. This can be done by minimizing the

cross-entropy between the output distributions of the teacher and student networks for the training inputs (Hinton et al., 2015). However, these techniques assume that the student is learning the same task as the teacher.

Knowledge from different networks can also be transferred at internal layers. Romero et al. (2014) proposed a method for transferring the knowledge of a wide and shallow teacher to a thin and deep student, called FitNet. Pre-trained a network by constraining an intermediate layer of the student network to have representations that could linearly predict "hints" from the teacher network (i.e., activation patterns at a corresponding layer in the teacher network). After this, the network was fine-tuined using the technique proposed in Hinton et al. (2015). The FitNet method was shown to improve the students classification accuracy.

Another prominent technique for performing transfer learning is to initialize the weights of the student network to those of the teacher. The network is then trained on a different task or using different data. This can lead to improved network performance (Yosinski et al., 2014). However, this requires that the teacher and student have the same, or very similar, architectures, which may not be desirable, especially if the teacher is a biological neural network.

In this paper, we introduce an auxiliary error function that enables a student network to learn from the internal representational spaces of a teacher that has a similar or different architecture. The method constrains the student's representational distances in a set of layers to approximate those of the teacher. The student can thus learn the computational transformations discovered by the teacher, leading to improved representational learning during training.

2. METHODS

Our method, representational distance learning (RDL), enables DNNs to learn from the representations of other models to improve performance. As in Lee et al. (2014), Szegedy et al. (2015), and Wang et al. (2015), we utilize auxiliary error functions to train internal layers directly in conjunction with the error from the output layer found via backpropagation. We propose an error function that maximizes the similarity between the representational spaces of a student DNN and that of a teacher model.

2.1. Representational Distance Matrices

In order to compare the representational spaces of models, a method must be used to describe them. As discussed in Weston et al. (2012), a representational space can be characterized by the pairwise distances between representations. This idea has been used in several methods such as MDS, which seeks to reduce the dimensionality of data while minimizing the error between the pairwise distance matrix of the original data and the reduced dimensionality data (Kruskal, 1964).

Kriegeskorte et al. (2008) proposed using the matrix of pairwise dissimilarities between representations of different inputs, which they called representational distance, or dissimilarity, matrices (RDMs), to compare computational models and neurological data. More recently, Khaligh-Razavi and Kriegeskorte (2014) used this technique to analyze several computer vision models, including the CNN proposed in Krizhevsky et al. (2012), and neurological data. Any distance function could be used to compute the pairwise dissimilarities, for instance the Euclidean or correlation distances. An RDM for a DNN can be defined by:

$$RDM(X; f_m)_{i,j} = d(f_m(x_i; W_m), f_m(x_j; W_m)) \quad (1)$$

where X is a set of n inputs (e.g., a mini-batch or a subset of a mini-batch), f_m is the neuron activations at layer m, x_i, and x_j are single inputs, W_m is the weights of the neural network up to layer m, and some distance, or dissimilarity, measure d.

In addition to characterizing the information present in a particular layer of a DNN, RDMs can be used to visualize the representational space of a layer in a DNN (**Figure 1**). information captured by internal layers in a DNN is challenging. Zeiler and Fergus (2014) proposed a method for visualizing the input features which active internal neurons at varying layers using deconvolutional neural networks. Yosinski et al. (2015) also proposed methods for visualizing the activations of a DNNs for a given input. However, these methods do not show the categorical information of each representational layer. Visualizing the similarity of labeled inputs at layers of interest, via an RDM, allow clusters inherent to the learned representational transformations to be viewed.

2.2. Representational Distance Learning

RDL uses an auxiliary error functions that maximizes the similarity between the RDMs of a student and the RDMs of a teacher at several layers. This is motivated by the idea that RDMs, or distance matrices in general, can characterize the representational space of a model. DNNs seek to learn a set of hierarchical representations. For classification, this culminates in finding a representational space where different classes are separable. RDL allows a DNN to learn from the representations of a different, potentially better, model by maximizing the similarity between the RDMs of the DNN being trained and the target model at several layers. Unlike in Bucilua et al. (2006), Ba and Caruana (2014), and Hinton et al. (2015). RDL not only directly trains the output representation, but also the representations of hidden layers. As discussed in Bengio (2012), however, large datasets can prohibit the use of pairwise techniques, since the number of comparisons grows quadratically with dataset size. To address this, our technique only uses a random subset of all pairwise distances for each parameter update. This allows the speed of our method to be constrained by the subset size and not the overall number of training examples, which is usually several orders of magnitude larger.

In order to maximize the similarity between the RDM of a DNN layer being trained and a target RDM, we propose minimizing the mean squared error between the two RDMs. This corresponds to making all possible pairwise distances as similar as possible:

$$E_{aux}(X; f_m; T_m) = \frac{2}{n(n-1)} \sum_{(i,j)|i<j} (RDM(X; f_m)_{i,j} - T_{m,i,j})^2 \quad (2)$$

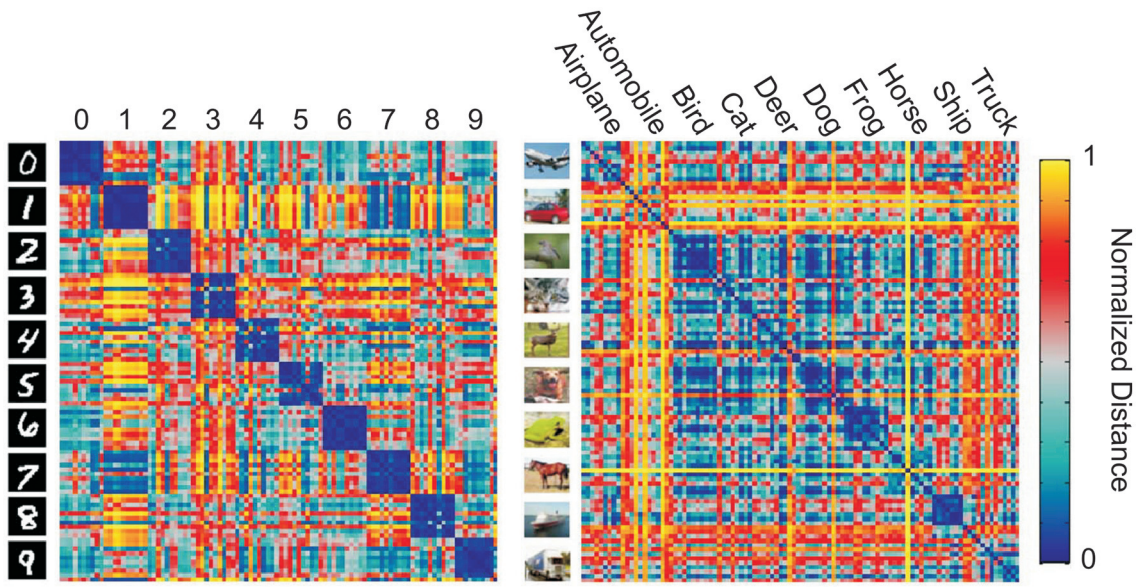


FIGURE 1 | Example representational distance matrices (RDMs) of the output layer of convolutional neural networks (CNNs) for 10 random images of each class from MNIST (left) and CIFAR-10 **(right)** made using the RSA toolbox (Nili et al., 2014).

TABLE 1 | The convolutional neural network (CNN) architecture used for MNIST.

Layer	Kernel size	Number of features	Stride	Non-linearity	Other
Conv-1	5 × 5	32	1	ReLU	–
MaxPool-1	3 × 3	32	3	Max	–
Conv-2	5 × 5	64	1	ReLU	–
MaxPool-2	2 × 2	64	2	Max	–
FC	1500	200	–	ReLU	Dropout ($p = 0.5$)
Linear	200	10	–	–	–

TABLE 2 | The McNemar exact test p-values for the tested CNNs trained on MNIST.

	Baseline	Teacher	Finetuning	Deep supervision	Hints	RDL
Baseline	–	0.38	0.00 ↑	0.11	0.34	0.01 ↑
Teacher	0.38	–	0.01 ↑	0.66	0.89	0.20
Finetuning	0.00 ←	0.01 ←	–	0.14	0.04 ←	0.63
Deep Supervision	0.11	0.66	0.14	–	0.64	0.39
Hints	0.34	0.89	0.04 ↑	0.64	–	0.17
RDL	0.01 ←	0.20	0.63	0.39	0.17	–

Arrows indicate a significant difference ($p < 0.05$, uncorr.) and point to the better model.

where X is a set of n inputs (e.g., a mini-batch or a subset of a mini-batch), f_m is the neuron activations at layer m, and $T_{m,i,j}$ is the distance between the teacher's representations of input x_i and input x_j at layer m. The function d used to calculate the RDMs (Equation 1) could be any dissimilarity or distance function, but we chose to use the mean squared error (MSE). This results in the average auxiliary error with respect to neuron k of f_m, $f_{m,k}$, for input x_i and the weights of the neural network up to layer m, W_m, being defined as:

$$\frac{\partial E_{aux}(x_i; X; f_m; T_m)}{\partial f_{m,k}} = \frac{8}{n(n-1)} \sum_{j|j \neq i} (RDM(X; f_m)_{i,j} - T_{m,i,j})(f_{m,k}|_{x_j}^{x_i}) \quad (3)$$

where $f_{m,k}|_{x_j}^{x_i} = f_{m,k}(x_i; W_m) - f_{m,k}(x_j; W_m)$.

However, calculating the error for every pairwise distance can be computational expensive, so we estimate the error using a random subset, P, of the pairwise distances for each update of a network's parameters. This leads to the auxiliary error gradient being approximated by:

$$\frac{\partial E_{aux}(x_i; X; f_m; T_m)}{\partial f_{m,k}} \approx \frac{8}{|X_P||P_{x_i}|} \sum_{(i,j) \in P_{x_i}} (RDM(X; f_m)_{i,j} - T_{m,i,j})(f_{m,k}|_{x_j}^{x_i}) \quad (4)$$

where X_P is the set of all images contained in P, P_{x_i} is the set of all pairs, (i, j), in P that include input x_i and another input, x_j. If an image is not sampled, its auxiliary error is zero.

The total error of $f_{m,k}$ for input x_i is calculated by taking a linear combination of the auxiliary error at layer m and the error from backpropagation of the output error function and any later auxiliary functions. These terms are combined using weighting hyper parameter α, similar to the method discussed in Lee et al. (2014), Szegedy et al. (2015), and Wang et al. (2015). In RDL,

α is the weight of the RDL error in the overall error function. Subsequently, the error gradient at a layer with an auxiliary error function is defined as:

$$\frac{\partial E_{total}(x_i; y_i; X; f_m; T_m)}{\partial f_{m,k}} = \frac{\partial E_{backprop}(x_i; y_i; f_m)}{\partial f_{m,k}} + \alpha \frac{\partial E_{aux}(x_i; X; f_m; T_m)}{\partial f_{m,k}} \quad (5)$$

This error is then used to calculate the error of earlier layers in the DNN using backpropagation. As discussed by Lee et al. (2014) and Wang et al. (2015), the value of α was decayed as training progressed. Throughout training, α was updated following $\alpha_{t+1} = \alpha_0{}^*(1 - t/t_{max})$ where t is the epoch number and t_{max} is the total number of epochs. By using this decay rule, the auxiliary error function initially helps drive the parameters to good values while allowing the DNN to converge predominantly using the output error by the end of training.

3. RESULTS

To evaluate the effectiveness of RDL, we perform two experiments using four different datasets, MNIST, InfiMNIST, CIFAR-10, and CIFAR-100. For each experiment, we transferred the knowledge of a teacher network trained on a separate dataset to a student network with the a similar architecture using: (1) finetuning after directly copying the weights of the teacher, (2) pre-training an internal layer of the student to linearly predict a corresponding layer in the teacher using "hints," and (3) using RDL. We compared the results to two non-transfer learning networks, a network only constrained at the output layer using the target labels and a deeply supervised network, which constrained both the output layer and internal layers using the target labels. We implemented all of these methods using Torch (Collobert et al., 2011). These experiments show that the knowledge stored in the weights of a teacher network can be transferred to a student network using the representational distances learned by a teacher trained on a related task.

3.1. MNIST

MNIST is a dataset of 28 × 28 images of handwritten digits from 10 classes, 0 through 9 (LeCun et al., 1998). The dataset contains 60,000 training images and 10,000 test images. A 10,000 image subset of the training data was used as a validation set for hyper-parameter tuning. No pre-processing or data augmentation was applied. InfiMNIST is a dataset that extends the MNIST dataset using pseudo-random deformations and translations (Loosli et al., 2007). The first 10,000 non-MNIST InfiMNIST examples were used as a validation set and the next 120,000 examples were used as a training set for the teacher network. Each tested network had the same architecture (**Table 1**), excluding any auxiliary error functions. The deeply supervised network had linear auxiliary softmax classifiers placed after the max pooling layers and α was decayed using $\alpha_{t+1} = \alpha_t^*0.1^*(1 - t/t_{max})$, as proposed in Lee et al. (2014). For the finetuning network, the weights

TABLE 3 | Test errors for MNIST trained convolutional neural networks (CNNs) and the CIFAR-100 trained "Network in Network" (NiN) models.

Method	Error (%)
MNIST	
Baseline CNN	0.63
Teacher	0.56
Teacher with finetuning	0.48
Student with deep supervision	0.55
Student with hints	0.56
Student with RDL	0.49
CIFAR-100	
Baseline NiN	30.68
Teacher with finetuning	38.75
Student with deep supervision	29.46
Student with hints	29.37
Student with RDL	28.77

The performance of the teacher for the CIFAR-100 classification is not shown, since it was trained on CIFAR-10 and, therefore, predicted across 10 not 100 classes, making it unable to perform the CIFAR-100 task.

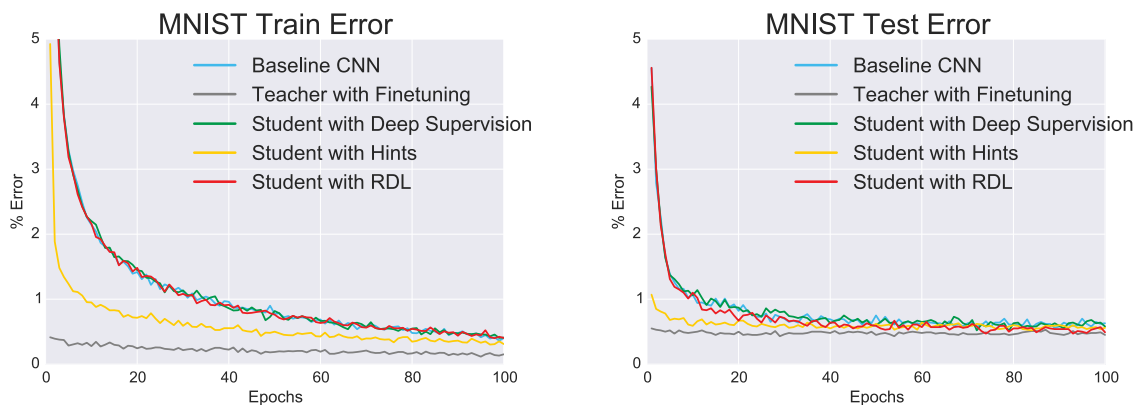


FIGURE 2 | The change in the train and test errors through time as the tested convolutional neural networks (CNNs) are trained on MNIST.

were initialized as the weights of the teacher network instead of being randomly initialized. After this, the network was trained normally. The RDL network had auxiliary error functions after both max pooling layers and the fully connected layer. 5% (500) of the image pairs per mini-batch were used to calculate the RDL auxiliary errors. A momentum of 0.9 and a mini-batch size of 100 were used for all networks trained on MNIST and InfiMNIST.

In addition to the classification error (**Figure 2** and **Table 3**), we used the McNemar exact test (Edwards, 1948) to evaluate whether a network was significantly more accurate in classifying

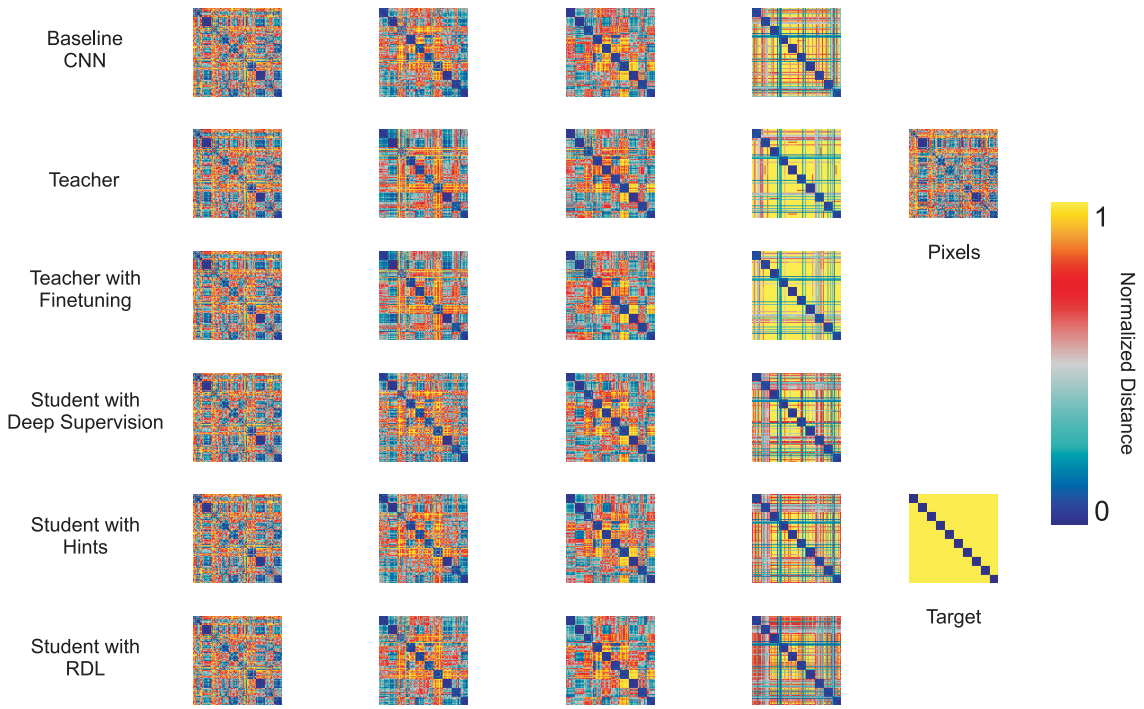


FIGURE 3 | Representational distance matrices (RDMs) using the Euclidean distance for the first and second convolutional layers as well as the fully connected (FC) and softmax layers of the CNN tested methods, the raw pixel data, and the target labels for 10 random class exemplars from MNIST.

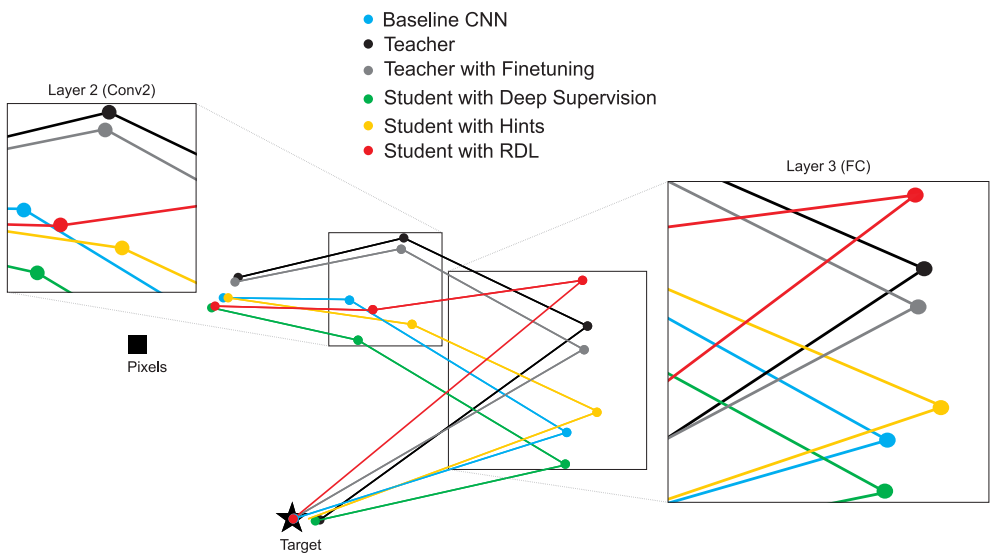


FIGURE 4 | 2-D multi-dimensional scaling (MDS) visualization of the distances between the representational distance matrices (RDMs) for selected layers of the MNIST trained networks. RDMs were generated for each model using 20 bootstrapped samples of 100 images from the test set. For each sampled image set, the correlation distance between the RDMs of the different networks were calculated. These values were then averaged to generate the MDS plot.

a random image from the distribution from which the images in the training and test sets were drawn. The results (**Table 2**) show that the finetuning and RDL methods both signifantly improve accuracy compared to the baseline CNN. They are, however, not significantly different, showing the ability of RDL to indirectly transfer the knowledge of the teacher network. The finetuned network is also significantly better than the teacher and the "hint" network, unlike RDL. This is because RDL actively constrains the student network to imitate the teacher, while finetuning only affects initialization.

In order to further compare the trained networks, RDMs were generated for each fully trained model. **Figure 3** shows RDMs for 100 random test images, 10 from each class. This visualization emphasizes the class clustering as inputs are transformed from pixel space to label space. Some classes are already clustered in pixel space. For instance, 1, 7, and 9 s each have large blocks along the diagonal portion of the pixel RDM. However, by looking at the rows and columns we can see that these classes are difficult to separate from one another. After the first convolutional layer, class clustering increases, especially for the baseline CNN. After the second convolutional layer, class clustering increases for every model and other class relationships become apparent. For instance, 3 and 5 s are becoming increasingly different from other classes, but are still similar to each other. Also, 1s remain similar to many other classes. The fully connected (FC) layer leads to stronger, but not perfect, class cluster. As expected, the softmax layer leads to extremely strong class distinction. However, most of the models still view 1s as similar to other classes, as seen by the large horizontal and vertical gray stripes. The notable exception is the finetuned CNN, which had the lowest testing error.

While viewing the RDMs directly can make certain facts about the transformations performed by the models evident, it can be hard to compare RDMs to each other by visual inspection. To better understand the relationships between the representations of the different models, we calculate the correlation distance between each pair of RDMs and use MDS to create a 2-D plot showing the relative position in representational space of the transformations learned by the various trained networks (**Figure 4**). This allows for drawing several qualitative conclusions. As expected, the RDMs of the networks start close to the pixel-based RDM and become more similar to the target RDM the deeper the layer. The differences between the evaluated techniques can most clearly be seen at the 2nd (Conv2) and 3rd (FC) layers. As expected: (1) the network initialized with the weights of the teacher and then finetuned has the most similar RDMs to the teacher, (2) deep supervision pulls the RDMs of the student toward the target, (3) RDL pulls the RDMs of the student toward and the RDMs of the teacher, especially at 3rd layer.

TABLE 4 | The "Network in Network" (NiN) architecture with batch-normalization (BN) (Ioffe and Szegedy, 2015) used for CIFAR-100.

Layer	Kernel size	Number of features	Stride	Non-linearity	Other
Conv-1	5 × 5	192	1	ReLU	BN
MLPConv-1-1	1 × 1	160	1	ReLU	BN
MLPConv-1-2	1 × 1	96	1	ReLU	BN
MaxPool	3 × 3	96	2	Max	–
Conv-2	5 × 5	192	1	ReLU	BN, Dropout ($p = 0.5$)
MLPConv-2-1	1 × 1	192	1	ReLU	BN
MLPConv-2-2	1 × 1	192	1	ReLU	BN
AveragePool-1	3 × 3	192	2	–	–
Conv-3	5 × 5	192	1	ReLU	BN, Dropout ($p = 0.5$)
MLPConv-3-1	1 × 1	192	1	ReLU	BN
MLPConv-3-2	1 × 1	100	1	ReLU	BN
AveragePool-2	8 × 8	100	–	–	–

3.2. CIFAR-100

In order to test RDL on a more interesting problem, we performed transfer learning from CIFAR-10 to CIFAR-100. This experiment consists of transferring knowledge learned in an easier task to a harder one, something that is useful in many instances. CIFAR-100 is a dataset of 32 × 32 color images each containing one of 100 objects. The dataset contains 50,000 training images and 10,000 test images. A 10,000 image subset of the training data was used as a validation set for hyper-parameter

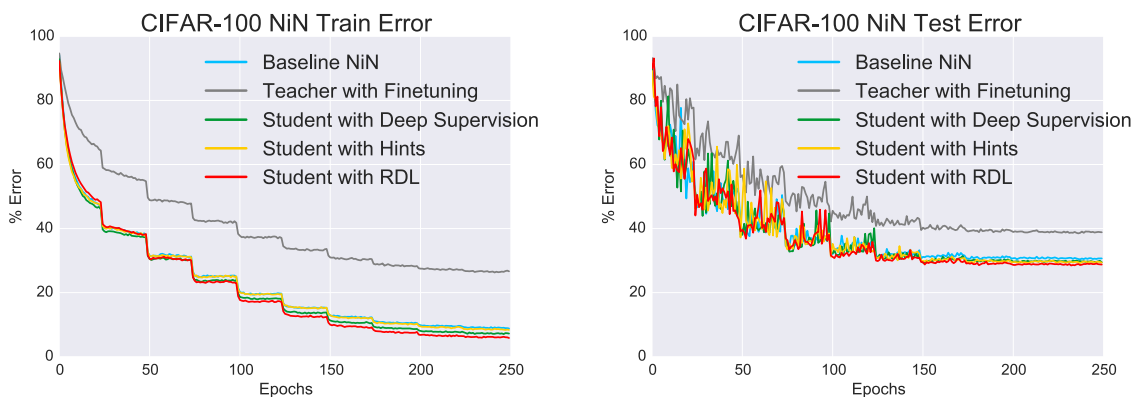


FIGURE 5 | The change in the train and test errors through time for the "Network in Network" (NiN) models trained on CIFAR-100.

tuning. CIFAR-10 is also a dataset of 32 × 32 color images, but containing only 10 distinct classes instead of 100. CIFAR-10 also contains 50,000 training images and 10,000 test images. For both datasets, the data were pre-processed using global contrast normalization. During training, random horizontal flips of the images were performed and the learning rate was halved every 25 epochs.

To evaluate using RDL with a more complex network, we used a "Network in Network" (NiN) architecture (Lin et al., 2013), which use MLPConv layers, convolutional layers that use multi-layered perception (MLP) filters instead of linear filters (**Table 4**).

TABLE 5 | The McNemar exact test *p*-values for the tested "Network in Network" (NiN) models trained on CIFAR-100.

	Baseline	Finetuning	Deep supervision	Hints	RDL
Baseline	—	0.00 ←	0.00 ↑	0.00 ↑	0.00 ↑
Finetuning	0.00 ↑	—	0.00 ↑	0.00 ↑	0.00 ↑
Deep supervision	0.00 ←	0.00 ←	—	0.86	0.08
Hints	0.00 ←	0.00 ←	0.86	—	0.05
RDL	0.00 ←	0.00 ←	0.08	0.05	—

Arrows indicate a significant difference ($p < 0.05$,uncorr.) and point to the better model.

The CIFAR-10 trained teacher network had the same architecture as the baseline CIFAR-100 NiN (**Table 4**) except with a 10-class output layer and had a testing error of 8.0%. The DSN had linear auxiliary softmax classifiers after the first and second pooling layers and α was decayed as proposed in Lee et al. (2014). The finetuning network's weights were initialized using those of the CIFAR-10 teacher network and a linear readout was added. The RDL network had the same architecture as the baseline CIFAR-100 network with randomly initialized weights and the addition of auxiliary error functions that used the RDMs from the CIFAR-10 teacher. For RDL, an additional linear readout was added after the last MLPConv layer since RDL does not specify that each neuron in a representation corresponds to an output class. For RDL, 2.5% (406) of the image pairs per mini-batch of 128 images were used to calculate the RDL auxiliary errors.

As in the previous experiment, the performances of the networks (**Figure 5** and **Table 3**) were statistically compared using the McNemar test. The results are shown in **Table 5**. Unlike in the MNIST experiment, the fine tuned network performed statistically worse than all tested methods. This is likely a combination of the weights being overspecialized for CIFAR-10 classification and the last MLPConv layer having less units. The networks that were trained with deep supervision, hints, and RDL all significantly improved upon the baseline NiN and the

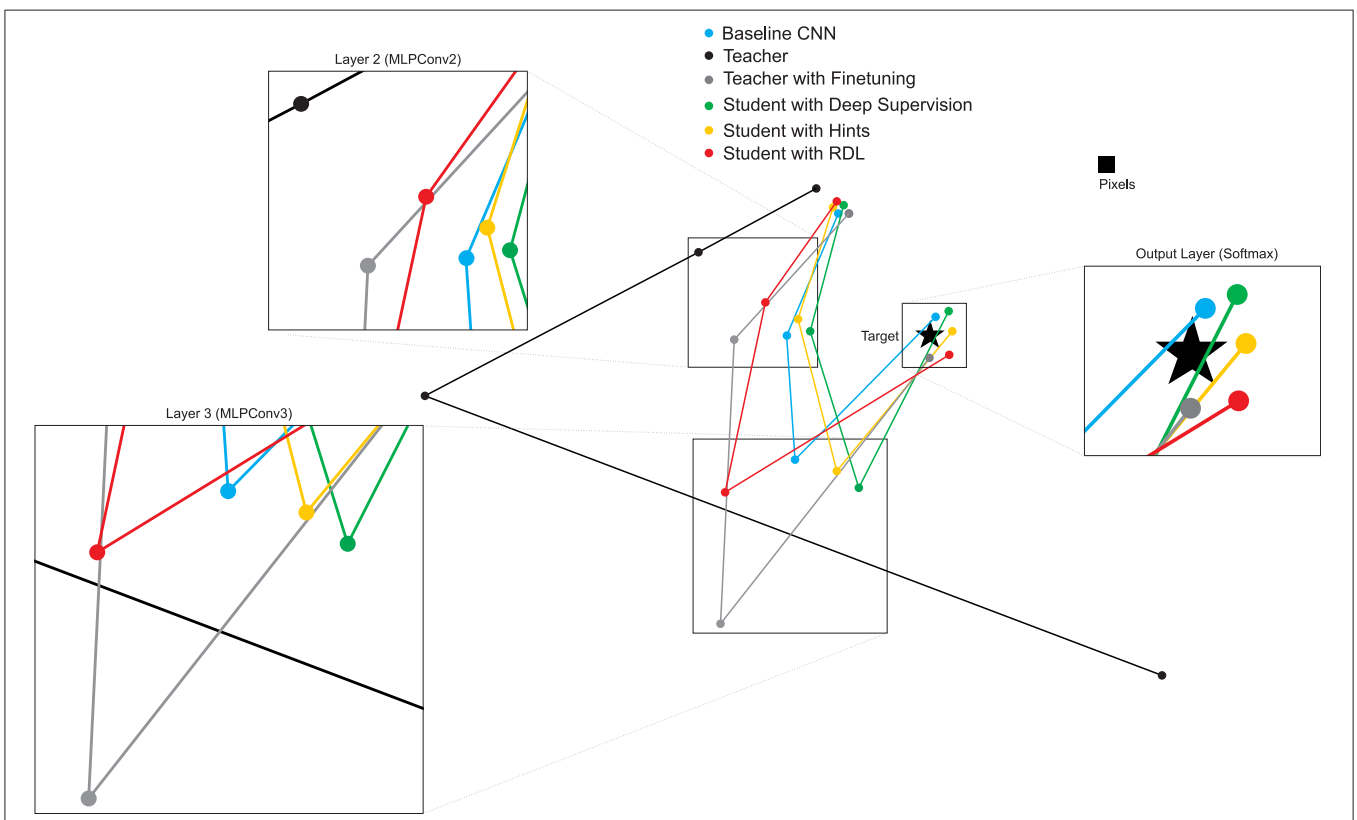


FIGURE 6 | 2-D multi-dimensional scaling (MDS) visualization of the distances between the representational distance matrices (RDMs) for selected layers of the CIFAR-100 trained networks. RDMs were generated for each model using 20 bootstrapped samples of 100 images from the test set. For each sampled image set, the normalized Euclidean distance between the RDMs of the different networks were calculated. These values were then averaged to generate the MDS plot.

finetuned network. These results show that learning from RDMs can extract meaningful information from a teacher network, which leads to improved classification performance.

To investigate the relationships between the representations of the different NiN models, we calculate the correlation between each pair of RDMs and use MDS to create a 2-D plot showing the relative position in representational space of the transformations learned by the various trained networks (**Figure 6**). The MDS plots shows that: (1) the layer 2 and layer 3 RDMs of the network initialized with the weights of the teacher and then finetuned are further from the target than the other non-teacher networks, (2) deep supervision pulls the RDMs of the student toward the target, (3) despite learning a series of transformations that do not map directly to the target, the teacher contains useful information to the students' task, and (4) RDL pulls the RDMs of the student toward and the RDMs of the teacher. This shows the ability of RDL to incorporate both the representational information from the teacher as well as from the classification task.

4. DISCUSSION

In this paper, we proposed RDL, a technique for transferring knowledge from a teacher model to a student DNN. The representational space of the student is pulled toward that of a teacher model during training using stochastic gradient descent. This was performed by minimizing the difference between the pairwise distances between representations of two models at selected layers using auxiliary error functions. Training with RDL was shown to improve classification performance by extracting knowledge from another model trained on a similar task, while allowing architectural differences between the student and teacher. This suggests that RDL can transfer the relationships between class examples learned by the teacher. This information is not present when only constraining internal layers using class labels, as done in the deeply supervised method, since the target vectors for each class are orthogonal. In particular, RDL allows a student network to learn similar sequential transformations to those learned by a teacher network. This could be of potential use in learning transformations similar to those performed in the human visual ventral stream. Such a model might be able to generate brain-like RDMs for novel stimuli. In the future, we plan to train such a model by constraining large DNNs using fMRI-based RDMs from the human visual ventral stream. By learning from brain-activity patterns, RDL has the potential to help build more realistic models of computations in biological brains.

AUTHOR CONTRIBUTIONS

NK and PM conceived of RDL. PM and NK developed the method. PM implemented RDL and performed the training and validation. PM and NK wrote the paper.

ACKNOWLEDGMENTS

This research was funded by the Cambridge Commonwealth, European & International Trust, the UK Medical Research Council (Program MC-A060-5PR20), and a European Research Council Starting Grant (ERC-2010-StG 261352).

REFERENCES

Ba, J., and Caruana, R. (2014). "Do deep nets really need to be deep?" in *Advances in Neural Information Processing Systems* (Montréal, QC), 2654–2662.

Belkin, M., and Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.* 15, 1373–1396. doi: 10.1162/089976603321780317

Bengio, Y. (2012). Deep learning of representations for unsupervised and transfer learning. *Unsupervised Transfer Learn. Challeng. Machine Learn.* 27, 17–36.

Bucilua, C., Caruana, R., and Niculescu-Mizil, A. (2006). "Model compression," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Philadelphia, PA), 535–541.

Collobert, R., Kavukcuoglu, K., and Farabet, C. (2011). "Torch7: a matlab-like environment for machine learning," in *BigLearn, NIPS Workshop* (Granada).

Deng, L., Hinton, G., and Kingsbury, B. (2013). "New types of deep neural network learning for speech recognition and related applications: an overview," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Vancouver, BC), 8599–8603.

Edwards, A. L. (1948). Note on the correction for continuity in testing the significance of the difference between correlated proportions. *Psychometrika* 13, 185–187.

Güçlü, U., and van Gerven, M. A. (2015). Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. *J. Neurosci.* 35, 10005–10014. doi: 10.1523/JNEUROSCI.5023-14.2015

Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. arXiv:1503.02531.

Ioffe, S. and Szegedy, C. (2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of The 32nd International Conference on Machine Learning* (Lille), 448–456.

Khaligh-Razavi, S.-M., and Kriegeskorte, N. (2014). Deep supervised, but not unsupervised, models may explain it cortical representation. *PLoS Computat. Biol.* 10:e1003915. doi: 10.1371/journal.pcbi.1003915

Kriegeskorte, N., Mur, M., and Bandettini, P. A. (2008). Representational similarity analysis-connecting the branches of systems neuroscience. *Front. Sys. Neurosci.* 2:4. doi: 10.3389/neuro.06.004.2008

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems* (Lake Tahoe, NV), 1097–1105.

Kruskal, J. B. (1964). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika* 29, 1–27.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 2278–2324.

Lee, C.-Y., Xie, S., Gallagher, P., Zhang, Z., and Tu, Z. (2014). Deeply-supervised nets. arXiv:1409.5185.

Lin, M., Chen, Q., and Yan, S. (2013). Network in network. arXiv:1312.4400.

Loosli, G., Canu, S., and Bottou, L. (2007). "Training invariant support vector machines using selective sampling," in *Large Scale Kernel Machines*, eds L. Bottou, O. Chapelle, D. DeCoste, and J. Weston (Cambridge, MA: MIT Press), 301–320.

Naselaris, T., Kay, K. N., Nishimoto, S., and Gallant, J. L. (2011). Encoding and decoding in fmri. *Neuroimage* 56, 400–410. doi: 10.1016/j.neuroimage.2010.07.073

Nili, H., Wingfield, C., Walther, A., Su, L., Marslen-Wilson, W., and Kriegeskorte, N. (2014). A toolbox for representational similarity analysis. *PLoS Computat. Biol.* 10:e1003553. doi: 10.1371/journal.pcbi.1003553

Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., and Bengio, Y. (2014). Fitnets: Hints for thin deep nets. arXiv:1412.6550.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2014). Imagenet large scale visual recognition challenge. *Int. J. Comput. Vision* 115, 211–252. doi: 10.1007/s11263-015-0816-y

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2015). "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Boston, MA), 1–9.

Wang, L., Lee, C.-Y., Tu, Z., and Lazebnik, S. (2015). Training deeper convolutional networks with deep supervision. arXiv:1505.02496.

Weston, J., Ratle, F., Mobahi, H., and Collobert, R. (2012). "Deep learning via semi-supervised embedding," in *Neural Networks: Tricks of the Trade*, eds G. Montavon, G. B. Orr, and K.-R. Müller (Heidelberg: Springer), 639–655.

Yamins, D. L. and DiCarlo, J. J. (2016). Using goal-driven deep learning models to understand sensory cortex. *Nat. Neurosci.* 19, 356–365. doi: 10.1038/nn.4244

Yamins, D. L., Hong, H., Cadieu, C. F., Solomon, E. A., Seibert, D., and DiCarlo, J. J. (2014). Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proc. Natl. Acad. Sci. U.S.A.* 111, 8619–8624. doi: 10.1073/pnas.1403112111

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems* (Montréal, QC), 3320–3328.

Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., and Lipson, H. (2015). Understanding neural networks through deep visualization. arXiv:1506.06579.

Zeiler, M. D., and Fergus, R. (2014). "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision*, eds D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars (Cham: Springer), 818–833.

Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., and Oliva, A. (2014). "Learning deep features for scene recognition using places database," in *Advances in Neural Information Processing Systems* (Montréal, QC), 487–495.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Computational Neuroscience

METHODS
published: 15 June 2017
doi: 10.3389/fncom.2017.00049

Estimating the Information Extracted by a Single Spiking Neuron from a Continuous Input Time Series

Fleur Zeldenrust[1]*, *Sicco de Knecht*[2], *Wytse J. Wadman*[2], *Sophie Denève*[3] and *Boris Gutkin*[3, 4]

[1] Department of Neurophysiology, Faculty of Science, Donders Institute for Brain, Cognition and Behaviour, Radboud University, Nijmegen, Netherlands, [2] Cellular and Systems Neurobiology, Swammerdam Institute for Life Sciences, University of Amsterdam, Amsterdam, Netherlands, [3] Group for Neural Theory, Institut National de la Santé et de la Recherche Médicale U960, Institute of Cognitive Studies, École Normale Supérieure, Paris, France, [4] Department of Psychology, Center for Cognition and Decision Making, National Research University Higher School of Economics, Moscow, Russia

Understanding the relation between (sensory) stimuli and the activity of neurons (i.e., "the neural code") lies at heart of understanding the computational properties of the brain. However, quantifying the information between a stimulus and a spike train has proven to be challenging. We propose a new (*in vitro*) method to measure how much information a single neuron transfers from the input it receives to its output spike train. The input is generated by an artificial neural network that responds to a randomly appearing and disappearing "sensory stimulus": the hidden state. The sum of this network activity is injected as current input into the neuron under investigation. The mutual information between the hidden state on the one hand and spike trains of the artificial network or the recorded spike train on the other hand can easily be estimated due to the binary shape of the hidden state. The characteristics of the input current, such as the time constant as a result of the (dis)appearance rate of the hidden state or the amplitude of the input current (the firing frequency of the neurons in the artificial network), can independently be varied. As an example, we apply this method to pyramidal neurons in the CA1 of mouse hippocampi and compare the recorded spike trains to the optimal response of the "Bayesian neuron" (BN). We conclude that like in the BN, information transfer in hippocampal pyramidal cells is non-linear and amplifying: the information loss between the artificial input and the output spike train is high if the input to the neuron (the firing of the artificial network) is not very informative about the hidden state. If the input to the neuron does contain a lot of information about the hidden state, the information loss is low. Moreover, neurons increase their firing rates in case the (dis)appearance rate is high, so that the (relative) amount of transferred information stays constant.

Keywords: neural information processing, artificial neural network, *in vitro* electrophysiology, Bayesian neuron model, information theory

OPEN ACCESS

Edited by:
Sander Bohte,
Centrum Wiskunde & Informatica, Netherlands

Reviewed by:
Valeri Makarov,
Complutense University of Madrid, Spain
Masami Tatsuno,
University of Lethbridge, Canada

***Correspondence:**
Fleur Zeldenrust
f.zeldenrust@neurophysiology.nl

Received: *17 October 2016*
Accepted: *19 May 2017*
Published: *15 June 2017*

Citation:
Zeldenrust F, de Knecht S, Wadman WJ, Denève S and Gutkin B (2017) Estimating the Information Extracted by a Single Spiking Neuron from a Continuous Input Time Series. Front. Comput. Neurosci. 11:49. doi: 10.3389/fncom.2017.00049

Abbreviations: BN, Bayesian Neuron, see Denève (2008a); MSE, Mean-Squared Error (Equation 14); FMSE, Fraction of output MSE relative to input MSE (Equation 16); MSE_P, Fraction of output MSE relative to MSE in Poisson spike train (Equation 15); F, Fraction of information relative to entropy of the hidden state (Equation 2); FI, Fraction of information about the hidden state in output relative to input (Equation 3); FS, SNR_{output}/SNR_{input}.

1. INTRODUCTION

Neuroscientists aim to understand how the brain represents and transforms incoming information by quantifying the relation between (sensory) stimuli and the activity of neurons (i.e., "the neural code"). When researching such information transfer properties of neural systems, and in particular of single neurons, there are two main questions: (1) *what* information is encoded by a neuron (and what information is discarded), and (2) *how much* information is transferred (or lost). The first question is often investigated by fitting functional filter models such as a Linear–Non-linear Poisson model (Chichilnisky, 2001) or a Generalized Linear Model (Paninski, 2004) to the neural input and output (for an overview see Simoncelli et al., 2004; Schwartz et al., 2006). Here, we will focus on the second question: How much information is transferred by single neurons? This question was first posed by MacKay and McCulloch (1952) and de Ruyter van Steveninck and Bialek (1988) were first to develop a way to measure the information transfer in neurons. This quantitative approach to information transfer is important, because it shows how information transfer properties change. For instance, the amount of information a neuron transmits depends on the background activity of the network a neuron is embedded in Panzeri et al. (1999) and Shadlen and Newsome (1998), on neuromodulators such as dopamine (Cruz et al., 2011) and on the type of code that is used (i.e., a "temporal" or "rate" code, Panzeri et al., 2001).

Researchers have attempted to measure the information transfer from presynaptic activity to output spike trains in neurons in different experimental setups and sensory systems *in vivo* and *in vitro* (including the visual system of the fly (de Ruyter van Steveninck and Bialek, 1988) and the whisker system of rats (Panzeri et al., 2001), using different information theoretical measures (for an overview, see Borst and Theunissen, 1999; Dimitrov et al., 2011). However, quantifying the information between a stimulus and a spike train has proven to be challenging. For example, information can be measured by reconstructing the stimulus from a spike train, and estimating the signal-to-noise ratio (Bialek et al., 1991; Rieke et al., 1997). This method requires a large amount of data, since a model needs to be fitted to the neural response (e.g., a linear filter and transfer function) before transferred information can be measured. Alternatively, information can be measured using the so-called "direct method" (de Ruyter van Steveninck et al., 1997; Strong et al., 1998), in which the response variability is used to estimate the mutual information between stimulus and spike train output. Measuring the information between a neuron's input and output this way involves various difficulties and biases, including the need to repeat a stimulus many times (or for a vary long time) and a bias due to limited sample sizes (Treves and Panzeri, 1995; Strong et al., 1998). Moreover, it might be difficult to determine what kind of stimulus to use, and in these setups the stimulus and the measured neuron are often several synapses away, making it difficult to assess where a measured loss of information happens. Finally, the choice of what set of stimuli to use is non-trivial.

Here we present a method to estimate how much information is contained in the spike train of a single neuron in an *in vitro* setup. The neuron is presented with an current input, generated by a population of artificial presynaptic neurons that respond to a randomly appearing and disappearing preferred stimulus: the hidden state (Denève, 2008a; Lochmann and Denève, 2008). This hidden state mimicks for instance a randomly appearing bar with a preferred orientation (for cells in primary visual cortex) or sound with a preferred frequency (for cells in auditory cortex). The information estimate is calculated by comparing the absence/presence of the hidden state and an estimate of the presence of this stimulus, based on the output spike train. The method does not require vast amounts of data or many repetitions. The method can be applied in any *in vitro* setup (so it not limited to sensory systems). Moreover, various experimental parameters such as the autocorrelation time-constant due to the (dis)appearance rate of the hidden state or the specific amount of information in the input and the amplitude of the signal relative to the background noise can systematically be varied, while the input is still close to the natural stimuli neurons normally receive. Finally, since we have a model of the optimal response (the "Bayesian neuron," Denève, 2008a), the quality of the performance of the neuron can be rigorously assessed.

The goal of the method presented here is to define an experimental paradigm with which the information (loss) of the spike-generating process can be quantified and compared (for instance between neuropharmacological states) in an *in vitro* paradigm. This information-calculation is based on previous work (Denève, 2008a; Lochmann and Denève, 2008), where a similar method was used to compare single-compartment models. Here, we add the following to the existing method: Firstly, we replace delta-spikes by exponential kernels to mimick Post-Synaptic Currents (PSCs). Secondly, we define the output of the artificial neural network as a current output, and scale it so that it can be injected in a current-clamp setup. Thirdly, we show that the mutual information in the input current can be kept constant while varying experimental parameters. There is a a trade-off between the autocorrelation time and the firing rates of the artificial presynaptic neurons: if the autocorrelation time is short (i.e., the hidden state appears and disappears with a high rate), a high firing rate of the presynaptic neurons is needed to keep the information in the input current constant[1]. Finally, we provide an example of an *in vitro* experiment where this paradigm is used. We apply the method presented here to pyramidal neurons in region CA1 of the rat hippocampus in an *in vitro* slice, to quantify the information loss from input to output spike train as a function of the stimulus (dis)appearance rate, the input current amplitude, and the information content of the input current (for an overview of other coding properties of these cells, see Hasselmo, 2011).

[1]Note that increasing the number of presynaptic neurons or the firing rates of the presynaptic neurons has the same effect: increasing the stimulus amplitude relative to the background noise. This relative stimulus amplitude is related to, but not the same as, the signal-to-noise ratio (see Supplementary Material).

2. METHODS

Here we present an experimental method to estimate how much information is contained in the spike train of a single neuron. In the first part of this methods section, we summarize and explain the theory behind the method. In order to easily estimate the information in a spike train, the neuron has to respond to a special type of input generated by an artificial neural network, which is explained first in Section 2.1.1. In the next Section 2.1.2, we explain how this special form of a noisy input can be used to quantify the information in the output spike train. The theoretical derivation follows Lochmann and Denève (2008), who compared model-neurons this way. Next, we define an optimal response model (Denève, 2008a; Section 2.1.3), which sets a benchmark for the performance of the recorded neuron.

In the second part of the methods section, we zoom in on the experimental part of the method: in Section 2.2.1 we explain how the activity of the artificial neural network, which is in arbitrary units, can be scaled so that it can be used as a fluctuating current input in an *in vitro* setup. Next, the input parameters used in the experiments are summarized (Section 2.2.2). Finally, the details of the experimental slice preparation and recording are given (Section 2.2.3).

2.1. Theory

2.1.1. Input Generation

Except for sensory receptors, neurons in the brain respond to input generated by other neurons. We assume here that neurons respond to the absence or presence of a preferred stimulus feature, for instance an edge in a preferred orientation (visual system). This absence or presence of the preferred stimulus feature is represented by the hidden state x (**Figure 1**): a binary variable that equals 1 if the preferred stimulus is present, and 0 if it is absemt. We assume that this preferred stimulus appears and disappears randomly following a memoryless (Markov) process with rates r_{on} and r_{off}. Or, stated differently, the quantities $\tau = \frac{1}{r_{\text{on}}+r_{\text{off}}}$ and $p_1 = \frac{r_{\text{on}}}{r_{\text{on}}+r_{\text{off}}}$ quantify respectively how fast the hidden state switches and the probability of finding the hidden state in the "ON" (1) state.

The second assumption in the input generation, is that neurons do not directly observe the hidden state, but receive synaptic inputs from a population of N presynaptic neurons i, whose firing rate is modulated by the stimulus so that each fire Poisson spike trains with rate q^i_{on} when $x = 1$, and q^i_{off} when $x = 0$. These two assumptions are comparable to the assumptions that are implicitly made when estimating tuning curves, for instance by fitting filter models such as a Linear-Nonlinear Poisson model (Chichilnisky, 2001) to sensory stimuli: in both cases it is assumed that a neuron responds only to the present value (so no history or reverberation effects) of a preferred stimulus feature that it does not have direct access to.

Each of the spikes from the population of artificial presynaptic neurons is convolved with an exponential kernel with a time constant of 5 ms and a unitary surface. Moreover, the spike trains from different presynaptic neurons are weighted according to their reliability, i.e., $w_i = \log \frac{q^i_{\text{on}}}{q^i_{\text{off}}}$ (**Figure 1**). This is the third and strongest assumption of the input generation. These values for the weights result in an optimally informative total input current (Denève, 2008a), and can be learned with an unsupervised, local, spike-dependent learning rule (Denève, 2008b). We did not use a learning rule here, but just used the "optimal" weights. The relation between the weights and the firing frequencies makes sense intuitively: we assume that the neuron listens strongly to informative neurons ($q^i_{\text{on}} >> q^i_{\text{off}}$, results in $w_i >> 0$), not to neurons that are not informative ($q^i_{\text{on}} \approx q^i_{\text{off}}$, so $w_i \approx 0$) and neurons that fire more when the preferred stimulus is absent have an inhibitory contribution ($q^i_{\text{on}} << q^i_{\text{off}}$, results in $w_i << 0$). Given these weights, the sum-total synaptic input is given by

$$I = \sum_{i=1}^{N} w_i s_i * k, \tag{1}$$

where $*$ denotes a convolution with the exponential kernel $k(t)$ and $s_i = \sum_{m_i=1}^{M_i} \delta(t - t_{m_i})$ is the spike train of artificial neuron i that depends on the hidden state through q^i_{on} and q^i_{off}. However, this input cannot be injected directly into a neuron in an *in vitro* setup or into a model neuron yet: it has to be scaled from dimensionless units to ampère A, which will be explained in Section 2.2.1. The autocorrelation time constant of the input depends on the switching rates of the hidden state (through $\tau = \frac{1}{r_{\text{on}}+r_{\text{off}}}$ and on the distribution of firing rates in the artificial neural network q^i_{on} and q^i_{off}. Since we do not know anything a priori about the distributions of the firing rates q^i_{on} and q^i_{off}, we make the most simple assumption and draw them from a Gaussian distribution. So q^i_{on} and q^i_{off} are all drawn from a Gaussian distribution with mean μ_q and standard deviation $\sigma_q = \sqrt{\frac{1}{8}}\mu_q$ (the value if σ_q is chosen so that virtually all firing rates are positive). Note that even though the firing rates q^i_{on} and q^i_{off} are drawn from the same distribution, this generally does not mean they have the same value.

2.1.2. Estimating Mutual Information

The mutual information between the hidden state and the the input (MI_I) or any output spike train ($MI_{\text{spike train}}$) in response to this input can easily be estimated, because the input defined in the previous section uses a hidden state x. In this section, we will explain how to estimate this information in a spike train (the method can be applied to any spike train, be it recorded, simulated any other spike train). We start by estimating the entropy of the hidden state. Next, we consider the following two steps: (1) the transformation from hidden state to input (MI_I), and (2) the transformation from input to spike train (i.e., the neural spike generating process, $MI_{\text{spike train}}$). By definition, MI_I and $MI_{\text{spike train}}$ cannot exceed the entropy of the hidden state H_{xx} (determined by p_1, Equation 4). If there would be no information loss, the mutual information between the spike train and the hidden state equals the entropy of the hidden state: $MI_{\text{spike train}} = MI_I = H_{xx}$. However, in practice every step will result in information loss: $MI_{\text{spike train}} < MI_I < H_{xx}$. Since we

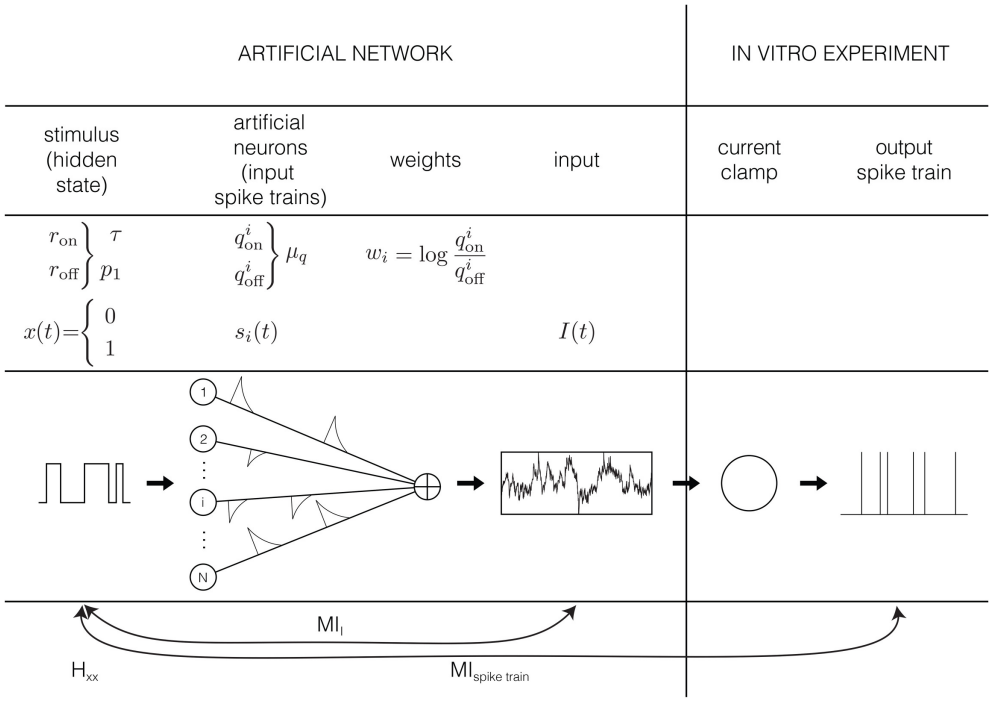


FIGURE 1 | Graphical representation of the input model: the input is generated by *N* artificial neurons *i* that fire Poisson spike trains with rates q^i_{on} and q^i_{off} in response to a hidden Markov model, with rates r_{on} and r_{off}. The spikes of each artificial neuron are convolved with an exponential kernel with a time constant of 5 ms and a weight of $w_i = \log \frac{q^i_{on}}{q^i_{off}}$. This current is injected into hippocampal pyramidal cells in an *in vitro* current clamp setup. The resulting spike trains are recorded and used to reconstruct the hidden state.

have access to $MI_{\text{spike train}}$, MI_I, and H_{xx}, we can estimate the information loss at every step.

The derivation follows Lochmann and Denève (2008) and Denève (2008a). The method requires two assumptions; firstly an ergodic argument: it is assumed that an average over samples can be replaced by an average over time. This means that if in an experiment the setup is not stationary during the time window for which the mutual information is calculated, the approximation fails. Secondly, it is assumed that output spike trains are by approximation Poissonian. The estimate of the mutual information is not strongly sensitive to this assumption, but strong deviations from Poissonian statistics will make the estimate fail. Time is measured in discrete steps, as most simulations and experiments use finite sampling rates. The mutual information is estimated for a single time-step, so it is an information rate (in bits/second). However, for simplicity and since we do not adjust the time step of our simulations of experiments here, we will only report the mutual information (in bits).

In Section 3, we will often use the fraction of transferred entropy

$$F = \langle \frac{\hat{MI}}{\hat{H}_{xx}} \rangle_{\text{samples or simulations}}, \tag{2}$$

where the brackets denote an average over samples orsimulations. This fraction shows how much of the entropy of the hidden state is transferred to the output spike train, and should therefore always have a value between 0 (since information or entropy cannot become negative) and 1 (the mutual information should never exceed the entropy of the hidden state). Similarly, we will use the fraction of transferred information

$$FI = \frac{\hat{MI}_{\text{spike train}}}{\hat{MI}_I}, \tag{3}$$

which should also have a value between 0 (no information about the hidden state in the input was transferred to the output spike train) and 1 (all information in the input was transferred to the output spike train).

2.1.2.1. Entropy of the hidden state

The theoretical value of entropy of the hidden state on each moment in time depends only on the probability that the hidden state is 1 (because a Markov process is memoryless):

$$H_{xx} = -p_1 \log_2(p_1) - (1 - p_1) \log_2(1 - p_1). \tag{4}$$

However, for a given realization (the full sequence of hidden state values up to time t: $x_{0\to t}$), the estimate of the entropy:

$$\begin{aligned} \hat{H}_{xx} \; = \; & -\langle x_{0\to t} \rangle_{\text{time}} \log_2(\langle x_{0\to t} \rangle_{\text{time}}) \\ & - (1 - \langle x_{0\to t} \rangle_{\text{time}}) \log_2(1 - \langle x_{0\to t} \rangle_{\text{time}}). \end{aligned} \tag{5}$$

could show small deviations from its true value given in Equation (4).

2.1.2.2. Conditional entropy

We start with a general estimate of the mutual information between the hidden state x and either an output spike train or the input. We use y to denote the history of the spike train or the input until now, and Y as the set of values y can take. The estimated mutual information MI is defined as the difference between the estimate of the entropy of the hidden state and the estimate of the conditional entropy of the hidden state given the history of y (spike train or input) until now:

$$\hat{MI} = \hat{H}_{xx} - \hat{H}_{xy}. \tag{6}$$

The conditional entropy of x given y is defined by

$$H_{xy} = -\sum_{x\in X, y\in Y} p(x,y)\log_2(p(x|y)), \tag{7}$$

where X is the set of values x can take (i.e., $X = \{0, 1\}$). Since the hidden state can only take the values 0 and 1, we can estimate the conditional entropy by averaging over time:

$$\begin{aligned}\hat{H}_{xy} &= -\langle x\log_2(p(x = 1|y)) + (1-x)\log_2(p(x = 0|y))\rangle_{\text{time}} \\ &= -\langle x\log_2(p(x = 1|y)) \\ &\quad + (1-x)\log_2((1-p(x = 1|y)))\rangle_{\text{time}},\end{aligned} \tag{8}$$

where we used the ergodic argument mentioned before to approximate an average over samples by an average over time. In the following two sections, we will explain how to estimate $p(x = 1|y)$ and $p(x = 0|y)$ based on either the input or an output spike train. Remember that x denotes the *current* value of the hidden state, whereas y signifies the spike train or input *history* up until now.

2.1.2.3. Mutual information between the hidden state and the input

To estimate of the conditional entropy of the hidden state given the input history, we have to estimate the probability of the hidden state being equal to 1 given the history of the input. Following the derivation in Denève (2008a), $L(t)$, the temporal evolution of the posterior log-likelihood of the hidden state being 1 based on the input history

$$L(t) = \log_2\frac{p(x = 1|I_{0\to t})}{p(x = 0|I_{0\to t})} = \log_2\frac{p(x = 1|I_{0\to t})}{1-p(x = 1|I_{0\to t})} \tag{9}$$

can be estimated using the following differential equation:

$$\frac{d\hat{L}}{dt} = r_{\text{on}}(1+e^{-\hat{L}}) - r_{off}(1+e^{\hat{L}}) + I(t) - \theta, \tag{10}$$

where $\theta = \sum_{i=1}^{N} q^i_{\text{on}} - q^i_{\text{off}}$ is the constant offset of the input, which is chosen to be equal to 0 in this paper[2]. So if we generate an input using the method from Section 2.1.1, we can integrate $\hat{L}$ using Equation (10) and estimate the mutual information using Equation (8) and the following estimate of the probability that the hidden state equals 1 given the input history:

$$\hat{p}(x = 1|I_{0\to t}) = \frac{1}{1+e^{-\hat{L}}}. \tag{11}$$

2.1.2.4. Mutual information between the hidden state and a spike train

The conditional entropy and the mutual information between the hidden state and an output spike train $\rho(t) = \sum_{m=1}^{M}\delta(t-t_m)$ can be estimated using the same method as for estimating the mutual information between the hidden state and the input: by integrating the log-likelihood L over time. However, parameter I in Equation (10) should now be replaced by $I_{\text{spike train}}$, generated with the help of Equation (1). In this equation, the exponential kernel k was used, because δ-spikes cannot be used in an experimental setup. However, for the information calculation, δ-spikes are not a problem, so the exponential kernel will be discarded[3]. For a given spike train, we need to estimate both θ and w, so we need to estimate q_{on} and q_{off}:

$$\begin{aligned}\hat{q}_{\text{on}} &= \frac{\int_{t|_{x=1}}\rho(t)dt}{\int_{t|_{x=1}}dt} = \frac{\text{total \# spikes while } x = 1}{\text{total time } x = 1} \\ \hat{q}_{\text{off}} &= \frac{\int_{t|_{x=0}}\rho(t)dt}{\int_{t|_{x=0}}dt} = \frac{\text{total \# spikes while } x = 0}{\text{total time } x = 0}.\end{aligned} \tag{12}$$

Now, we can generate $I_{\text{spike train}}$ for calculating L using Equation (10) and estimating the mutual information using Equations (8) and (11).

2.1.2.5. Hidden state estimate and mean-squared error

With the help of Equation (9), an estimate of the hidden state can be defined: because the hidden state can only take the values 0 and 1, and the estimate of the probability that the hidden state is equal to one $\hat{p}(x = 1|I_{0\to t})$ can only take values between 0 and 1, $\hat{p}(x = 1|I_{0\to t})$ can be viewed as an estimate of the hidden state:

$$\hat{x}(t) = \hat{p}(x = 1|I_{0\to t}) = \frac{1}{1+e^{-\hat{L}(t)}}. \tag{13}$$

This can be used to calculate another measure of how well a spike train represents the hidden state, the mean-squared error (MSE)[4]:

$$MSE = \frac{1}{N_t}\sum_{t=1}^{N_t}(\hat{x}_t - x_t)^2, \tag{14}$$

[2]For large enough N, $\theta \approx 0$. Since q^i_{on} and q^i_{off} are drawn from the same normal distribution with mean μ_q and standard deviation $\sigma_q = \sqrt{\frac{1}{8}}\mu_q$, the difference distribution has mean 0 and standard deviation $\sqrt{2}\sigma_q = \sqrt{\frac{1}{2}}\mu_q$.

[3]Due to the discretization of numerical approaches, "true" δ-spikes cannot be implemented in a computer. Rather, a δ-spike is implemented as a square kernel with width dt and height $\frac{1}{dt}$.

[4]Note that this gives us another estimate of the mutual information, based on the relation between the signal-to-noise ratio and the mutual information $MI = \int_0^\infty \log_2(1+SNR(f))df$ (Shannon, 1984; Cover and Thomas, 1991; Guo et al., 2005; Schultz, 2007) and the noise signal is defined as noise $= x - \hat{x}$. However, this will give us essentially the same results, since it is based on the same estimate $\hat{p}(x = 1)$ and $\hat{L}$.

where we used discretized time. We can normalize this *MSE* by dividing it by the *MSE* of Poisson spike-trains with the same number of spikes

$$MSE_P = \frac{MSE_{\text{spike train}}}{\langle MSE_{\text{Poisson spike train}} \rangle_{\text{simulations}}}. \quad (15)$$

This gives us a quantity that is around 1 when a spike train performs as well as a Poisson spike train (so when there is no information about the hidden state in the spike train) and vanishes when the hidden state can be perfectly inferred from the spike train (so the *MSE* vanishes). We can also normalize $MSE_{\text{spike train}}$ by dividing it by the mean-squared error obtained with the input MSE_I:

$$FMSE = \frac{MSE_{\text{spike train}}}{MSE_I}. \quad (16)$$

This represents how much noise the spike process of the neuron adds to the estimate of the hidden state: if it equals 1 the error of the estimate based on the input has the same size as the error based on the spike train, and the neuron transmits all the information in the input perfectly.

2.1.2.6. Delays

The theoretical form of the input was derived using Dirac-delta spikes (Denève, 2008a). Since an input consisting of delta spikes cannot be used in an experimental setup, we chose to convolve the input with exponential kernels, which mimics cortical PSC shapes. However, since an exponential kernel rises instantaneously, but decays slowly, this introduces a delay in the input relative to the hidden state. On the next level, any neuron that responds to this input will have a non-vanishing membrane time constant, resulting in a further delay. With this reasoning, each processing level adds a few ms delay to the representation of the hidden state. To separate the effects due to delays and other effects influencing the quality of the representation, we also calculate the mutual information between the hidden state and a shifted version of the input or spike train: we calculate the time-value peak of the cross-correlogram between the hidden state and the input/spike train, and shift the input/spike train by this amount. The mutual information resulting from this calculation will be denoted by MI^*.

2.1.3. Optimal Response Model

One of the advantages of creating an input using a hidden Markov model is that we have a model for an optimal response: the "Bayesian neuron" (Denève, 2008a). This model compares the log odds ratio of the stimulus (i.e., the log-likelihood of the hidden state being 1, see Equation 9) based on the input L with the log odds ratio based on the output spike train G, and keeps this difference small by spiking at appropriate times. This neuron only spikes if the likelihood of the hidden state being 1 based on the output spike train is lower than the likelihood of the hidden state being 1 based on the input, thereby only transferring "new" information and making efficient use of its output spikes:

$$\begin{aligned} \frac{dL}{dt} &= r_{\text{on}}(1+e^{-L}) - r_{off}(1+e^{L}) + I(t) - \theta \\ \frac{dG}{dt} &= r_{\text{on}}(1+e^{-G}) - r_{off}(1+e^{G}) \\ \text{if } L > G + \frac{\eta}{2} &: \begin{cases} \text{a spike is fired} \\ G \rightarrow G + \eta \end{cases} \end{aligned} \quad (17)$$

For a given input, the only free parameter in this model is η, the reset and threshold condition which sets the output firing rate of the neuron. The mutual information between a spike train and the hidden state necessarily depends on the firing rate: if a neuron does not spike, the mutual information vanishes. To signal whether the hidden state switches on (or off), the neuron needs to fire at least one spike every on (or off) state. Ideally, the firing rate of a spike train is comparable to $\frac{1}{\tau}$. We use parameter η to match the firing rate of the neurons we measured and the firing rate of the Bayesian neuron, to be able to compare the two.

2.2. Experimental Design

2.2.1. Scaling

The input defined in Section 2.1.1 is dimensionless (the weights only give a relative contribution, scaled to how informative the artificial neuron is about the hidden state). Input currents used in *in vitro* experiments has either unit ampère A (current clamp), volt V (voltage clamp), or siemens S (dynamic clamp). Therefore, the dimensionless theoretical "input current" from the artificial network has to be scaled so that it can be injected into the neuron in a current clamp setup (so we will have to scale the input generated by the artificial network to ampère A).

$$I_{\text{injected}} = I_{\text{hold}} + I_{\text{scale}} I_{\text{Markov}}(t), \quad (18)$$

where $I_{\text{Markov}}(t)$ is the dimensionless "current" defined by Equation (1). Finding I_{hold} and I_{scale} is not a trivial procedure: how "strong" an input current is for a neuron depends on its sensitivity to input current. This sensitivity can depend on several neuronal properties such as its excitability (rheobase, the steepness of the input-frequency curve), but also on the size of the neuron and the strength of the seal of the patch clamp. Here, we chose the following solution:

- **Offset:** In the current clamp measurements the membrane potential was adjusted by a feedback system that injects current (I_{hold}), so that the membrane potential stabilized to a desired value (−65 mV) before the actual measurement was started. From then on the value I_{hold} was fixed.
- **Amplitude:** We used a probe input (see Section 3.1.2) to define the amplitude with which to scale all inputs for a given neuron: we tried factors I_{scale} (with a resolution of 250 pA, so 250, 500, 750, 1000, 1250 pA, etc.) to set the firing rate response of the neuron to about 12 Hz overall (about 20 Hz when $x = 1$).

2.2.2. Parameters

Every parameter set $\{\tau, p_1, \mu_q\}$ defines an input "regime." We chose three "difficult" (i.e., low MI_I) regimes: a "slow" (S) regime,

with a small μ_q and a large τ; a "fast" (F) regime, with a large μ_q and a small τ, and a "probe" (P) regime in between with intermediate μ_q and τ. The probe served to determine the scaling of the input (previous Section 2.2.1). For comparison, we also used a "fast switching—low amplitude" (FL) regime with a very low information content and a "slow switching—high amplitude" (SH) regime with a high information content. The exact values and reasoning behind the regimes will be explained in Section 3.1.2.

As explained in the previous Section 2.2.1, the theoretical input generated by the artificial network needs to be scaled in order to use it in an experimental set-up. We scaled the inputs generated in the different regimes all with the same factor (Equation 18). From then on the value was fixed. To determine I_{scale} we used the probe (P) input, i.e., an input with the same information content as the S and F inputs, but with an intermediate $\tau : \tau_{\text{probe}}$ and $\mu : \mu_{q,\text{probe}}$. The mutual information between the hidden state and a spike train naturally depends on the firing rate. Therefore, we scale the input current so that each neuron responds with about the same firing rate to the probe input: about 12 Hz overall (about 20 Hz when $x = 1$).

The input defined in Section 2.1.1 was generated once for each regime, and consequently used as a "frozen noise" input for the experiments and simulations. The parameters for the generated input are shown in **Table 1**. In Section 3.1.2 we will motivate these choices. The input used in the experiments was 20 s for the probes, and 300 s for each of the other regimes. The mutual information was calculated on 15 consecutive windows of 20 s. Unless mentioned otherwise, we used a sampling rate of 5,000 Hz (so a time step of $dt = 0.2$ ms) for both the input in the experiments and the simulations. Due to the limited time we had for each neuron, we measured in each neuron both the "slow" (S) regime and the "fast" (F) regime, but only the "fast, low amplitude" (FL) OR "slow, high amplitude" (SH) regime in the following order: (1) F, (2) SH, (3) S or (1) S, (2) FL, (3) F. So the switching speed was always changed first, and the amplitude second.

We obtained valid recordings from 6 cells. We measured the mutual information of on traces of 20 s. Since we used 300 s recordings, this means we obtained 15 measurements of the mutual information per neuron and per regime.

2.2.3. Experiments

2.2.3.1. Animals and slice preparation

Electrophysiological experiments were performed using brain slices from 4 to 5 week old C57/Bl6 mice (Harlan, The Netherlands) of either sex (3 animals, 5 different slices in total). All experiments were performed with the approval of the committee on animal bioethics of the University of Amsterdam. Hippocampal acute slices were prepared in ice cold (4°C) modified artificial cerebro spinal fluid (ACSF, in mM)—120 choline Cl, 3.5 KCl, 0.5 $CaCl_2$, 6 $MgSO_4$, 1.25 NaH_2PO_4, 10 D-glucose, 25 $NaHCO_3$. Animals were killed by decapitation, and 350 μm thick slices were cut in the horizontal plane on a vibrating slicer (Leica, VT1200S; Wetzlar, Germany). Slices were kept in a perfusion chamber with ACSF (in mM)—120 NaCl, 3.5 KCl, 2.5 $CaCl_2$, 1.3 $MgSO_4$, 1.25 NaH_2PO_4, 10 Glucose, 25 $NaHCO_3$ at 32°C for 30 min, and then left at room temperature for at least 30 min until recordings started. For further details on the animals and slice preparation, see Wierenga and Wadman (2003).

2.2.3.2. Electrophysiological recordings

Current-clamp recordings were made under constant superfusion of ACSF bubbled with carbogen (95% O_2/5% CO_2) at a temperature of 32°C. We recorded neurons solely from the pyramidal cell layer of region CA1 and identified the pyramidal cells using differential interference contrast (DIC) with a light source of 780 nm (Scientifica; Uckfield, UK), as well as on the basis of their firing properties. Neurons were recorded in whole cell current clamp configuration with the Axopatch 200B amplifier (Axon Instruments Inc.; Forster City, CA, USA). For these recordings we used a pipette solution with (in mM) 131.5 K-gluconate, 8.75 KCl, 10 HEPES, 0.5 EGTA, 4 MgATP, and 0.4 NaGTP, this solution was brought to a pH of 7.3. Glass pipettes with a resistance in the range of 2.5–4 MΩ were used. Signals were low-pass filtered at 5 kHz and sampled at 25 kHz. Series resistances was compensated up to 70%. Data was acquired with in-house MATLAB based routines (MathWorks, 2007b; Natick, MA, United States).

We compensated online for the liquid junction potential (14.5 mV), as calculated from the solutions. To determine I_{hold}, we used a feedback system that stabilized the membrane potential to −65 mV until the actual measurement was started.

3. RESULTS

In order to calculate the information transfer in single neurons in an *in vitro* setup, we designed an input current defined in Sections 2.1.1 and 2.2.1. Before we describe the results of the current clamp experiments, we will first discuss the properties of this input current.

3.1. Input Properties

3.1.1. Information in Input Depends on Switching Speed and Firing Rate

The input defined by Equation (1), depends on the switching speed of the hidden state (r_{on} and r_{off}) and on the firing rates of the artificial presynaptic neurons (q^i_{on} and q^i_{off}, see **Figure 1**). The characteristics of the hidden state are external, i.e., they model how "the world outside of the animal" behaves. The characteristics of the artificial neurons model how neurons presynaptic to the real neuron (inside the animal) respond to the external stimulus. Both the external parameters of the "outside world" and the modeled internal parameters of the artificial neurons influence how much of the entropy of the hidden state (H_{xx} is transferred to the spike trains received by the neuron (mutual information in the input, MI_I). In **Figure 2** we kept the entropy of the hidden state constant ($r_{\text{off}} = 2r_{\text{on}}$, so the probability of the hidden state being 1 equals $p_1 = \frac{1}{3}$ and the entropy of the hidden state is $H_{xx} \approx 0.92$ bits at each moment in time). The switching speed τ of the hidden state and the firing rates μ_q of the artificial presynaptic neurons were independently varied. We calculated the fraction of the entropy in the hidden state that gets transferred to the input

TABLE 1 | Parameter values for the different input regimes.

Regime	Symbols	Abbreviations	r_{on} (Hz)	r_{off} (= 2 r_{on}) (Hz)	τ (ms)	μ_q (Hz)
Slow	•	S	6.7	13.3	50	0.5
Fast	▲	F	$5r_{on,\,slow} = 33.3$	66.7	10	$5\mu_{q,slow} = 2.5$
Probe	■	P	$2.5r_{on,\,slow} = 16.7$	33.3	20	$2.5\mu_{q,\,slow} = 1.3$
Slow, high amplitude	♦	SH	6.7	13.3	50	2.5
Fast, low amplitude	∗	FL	33.3	66.7	10	0.5

(Equation 2). **Figure 2** shows that there is a trade-off between the switching speed of the hidden state and the firing rates of the presynaptic neurons: if the switching speed is high (small τ), a high firing rate of the presynaptic neurons is needed to represent the hidden state, whereas for lower speeds the firing rates can be lower. This was expected: in order to represent x, one or more of spikes are needed to signal each period when x is in the "ON" state (i.e., a period when $x = 1$). A higher switching rate implies that these "ON" periods are shorter and more frequent. Even though the total "ON"-time might be unchanged, there are more separate "ON" states. Therefore, if every "ON"-state needs (at least) one output spike to be visible in the output spike train, more spikes are needed for a fast-switching hidden state (small τ). Note that since the artificial presynaptic neurons fire Poissonian spike trains, a higher overall firing rate can be obtained by either increasing the individual firing rates of the neurons (μ_q), as in **Figure 2**, or by increasing the number of presynaptic neurons N[5]. The relationship between μ_q and τ is almost inversely proportional (black line shows inversely proportional relationship).

Even though the time constant of the hidden state (τ) and the firing rates of the presynaptic neurons (μ_q) have a similar effect on the mutual information between the hidden state and the input, their effects on the shape of the input are quite different: the effect of increasing μ_q is to increase the amplitude of the input (**Figure 2**). Alternatively, increasing τ does not increase the amplitude, but changes the autocorrelation-time τ_{auto} (see Supplementary Material) of the input current signal. So, with τ and μ_q we can vary the input amplitude and autocorrelation-time independently, while keeping the mutual information between the input and the hidden state constant.

3.1.2. Input Regimes

In order to show the power of the method presented here, we designed two inputs with the same mutual information between the input current and the hidden state, but with a different amplitude (μ_q) and time-constant (τ) on the basis of our results from the previous section. The results of the current clamp experiment will be shown in Section 3.2. Here, we explain the design of the experiment (**Figure 2** and **Table 1**). We chose three "difficult" (i.e., low information content) regimes: a "slow" (S) regime (circle •), with a low amplitude and a large τ, a "fast" (F) regime (triangle ▲), with a high amplitude and a small τ, and a "probe" (P) regime in between (square ■) with intermediate firing rates and τ. The probe served to determine the scaling of the input current (Section 2.2.1). For comparison, we also used a "fast switching—low amplitude" (FL) regime with a very low information content (star ∗) and a "slow switching—high amplitude" (SH) regime with a high information content (diamond ♦).

As explained in Section 2.2.1, the theoretical input generated by the artificial network needs to be scaled in order to use it in an experimental set-up. We scaled the inputs from the different regimes all with the same factor (Equation 18). This factor was determined once for each neuron, from then on the value was fixed. To determine I_{scale} we used the probe (P) input defined before, i.e., an input with the same information content as the S and F inputs, but with an intermediate τ_{probe} and $\mu_{q,probe}$. As argued before, the mutual information between the hidden state and a spike train naturally depends on the firing rate. Therefore, we scale the input current so that each neuron responds with about the same firing rate to the probe input: about 12 Hz overall (about 20 Hz when $x = 1$).

3.2. Experimental Results

3.2.1. Representation of the Hidden State by a Single Neuron

3.2.1.1. Neurons perform a non-linear operation on their input

In the previous section, we explained the rationale behind the experiments. In **Figure 3** we show the distributions of the injected input current (left) and the resulting membrane potential (right) of one example neuron (denoted with + in **Figure 5**). The input current distributions of both the S (blue), and the FL (pink) regimes were identical, as expected. There was a small difference between both F regimes (red) and the SH (green) regime, because in the F regime the "ON" state ($x = 1$) and "OFF" state ($x = 0$) are blurred by the exponential shape of the artificial EPSCs (Section 2.1.1). The resulting membrane potential distributions (**Figure 3**, right) are unimodal for both the S and the FL regimes, as expected. However, in the SH regime the (output) membrane potential distribution (green) is bimodal, whereas the (input) current distribution is unimodal for this regime (this effect was found for all cells for which we measured the SH regime). This means that the neuron performs

[5]If we would use delta-spikes to simulate postsynaptic current shapes these two options are completely equivalent (as long as N is large enough, about 100 or more neurons are needed or otherwise the realization of the firing rates and therefore the weights w results in large variations between realizations). For all other PSC-shapes, there could be a small effect if overlapping spikes from different neurons are added different from overlapping spikes from the same neuron, but we will not consider this technical issue here, since the effects are small.

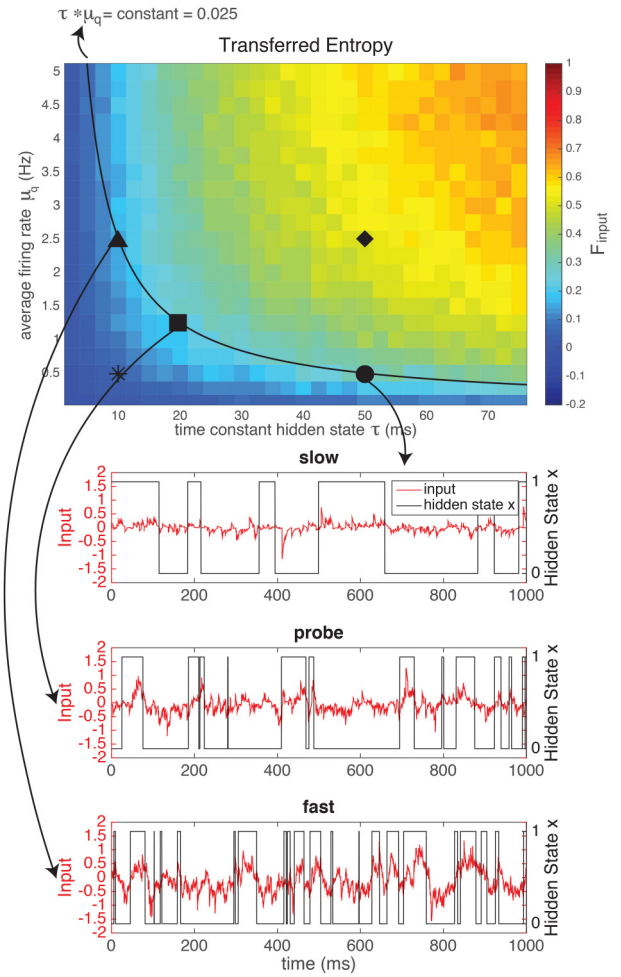


FIGURE 2 | The mutual information between the hidden state and the input generated by a network of $N = 1,000$ artificial neurons as a fraction of the entropy in the hidden state (top). This fraction of transferred information depends both on the firing rate of the neurons in the network (μ_q) and on the switching speed (time constant τ) of the hidden state: if the hidden state switches faster, more spikes (i.e., higher firing rate or more neurons) are needed to reliably represent the hidden state in the input. For the experiments we chose three "difficult" (i.e., low information content) regimes: a "slow" regime (circle), with a low firing rate and a large τ, a "fast" regime (triangle), with a high firing rate and a small τ and a "probe" regime in between (square) with intermediate firing rates and τ, to determine the scaling of the input. For comparison, we also used a "fast switching—low amplitude" regime with a very low information content (star, example shown in Supplementary Material) and a "slow switching—high amplitude" regime with a high information content (diamond, example shown in Supplementary Material). NB Note that even though theoretically $MI \geq 0$, due to our approximation $\hat{MI}$ can take small negative values. However, these effects are negligable (smallest value in this figure is $\hat{MI} = -0.0011$).

a non-linear operation on the input current; with a linear transformation, the shape of the distribution would stay identical. Moreover, the distributions of the membrane potentials in both F regimes (red, full and dotted line) are not identical. This could be due to neural adaptation to the input or to non-stationary experimental conditions (for instance resistance of the seal with the pipette).

3.2.1.2. Neurons transmit information about the hidden state

In **Figure 4** we show the hidden state and the different estimates of the hidden state (Equation 13), in the S (**Figure 4A**) and F (**Figure 4B**) regime, for a single hippocampal (CA1) pyramidal cell (depicted with □ in **Figure 5**). Note that in both regimes, spikes occur mostly in when $x = 1$, even if there is not a spike every time. In **Figure 4C** we calculated the *MSE* between the hidden state and the estimated hidden state, based on the spike times of the recorded neuron and normalized by a Poisson spike train of the same rate (MSE_P, Equation 15). Note that the values in both the slow and fast regime are smaller than but not far from 1, meaning that the estimate is not much better than that of a Poisson process. The neuron performs slightly better in the slow regime: the difference in mean-squared error is small but significant (slow $MSE_P = 0.83 \pm 0.03$, fast $MSE_P = 0.92 \pm 0.01$, Student's t-test on difference $p = 1.2 \cdot 10^{-7}$). In **Figure 4D** it can be seen that the ratio between the *MSE* based on the spike train and the *MSE* based on the the input (Equation 16) is close to 1 (but significantly different; slow $FMSE = 1.26 \pm 0.05$, Student's t-test on difference between 1: $p = 5.8 \cdot 10^{-12}$, fast $FMSE = 1.16 \pm 0.02$, Student's t-test on difference between 1: $p = 5.3 \cdot 10^{-13}$). So even though the neuron does not perform much better than a Poisson process (**Figure 4C**), there is not much information loss between the input and the output spike train. The low mutual information between the spike train and the hidden state is a result of the low information content of the input. Indeed, in **Figures 4E,F** it is shown that the spike train transmits about 40–50% of the information in the input (Equation 3).

Even though the firing rate in the F regime is much higher than that of the S regime, the difference in output-information between the S and the F regime is very small [but significant: (**Figure 4E**) slow $FI = 0.45 \pm 0.05$, fast $FI = 0.37 \pm 0.04$, Student's t-test on difference $p = 1.2 \cdot 10^{-4}$, (**Figure 4F**) slow $FI_{\text{shifted}} = 0.48 \pm 0.06$, fast $FI_{\text{shifted}} = 0.39 \pm 0.04$, Student's t-test on difference $p = 0.0012$]. This means that the recorded neuron represent the hidden state states equally well in both regimes, but it is less *efficient* in the F state: it needs more spikes to transfer the same amount of information. As explained before (Section 3.1.1), more spikes are needed to represent a fast-switching hidden state. The result that the recorded neuron indeed increases its firing rate in the F regime relative to the S regime and the transferred information stays the same in both regimes suggests that the neuron "adapts" to the different regimes to keep the transferred information constant.

In **Figure 5A** we show the *FI* against the firing rate (same as in **Figure 4E**) for all recorded neurons. Different symbols denote different cells, whereas different colors denote the different regimes. The fraction of information about the hidden state in the input that is transmitted into the output spike train, depends on the amount of information in the input: in the very informative regime (SH, green), about 50–60% of the information in the input is transferred to the spike train, whereas in the low informative regime (FL, pink) only about 10% of the information is transmitted. In the intermediate S (blue), F (red), and P (black) regimes, the transmitted information is comparable and between these two extremes.

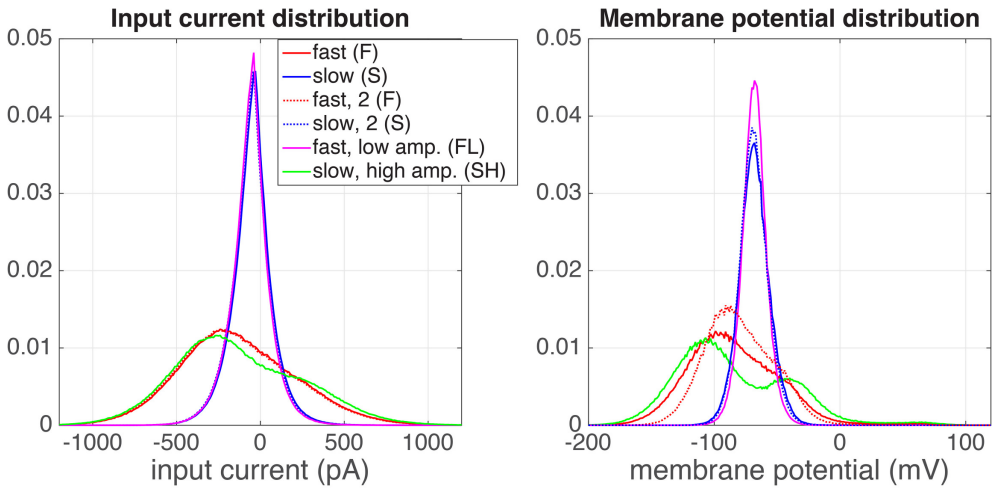


FIGURE 3 | Distributions of the input current **(left)** and output membrane potential **(right)**, for a one of the recorded cells (depicted with symbol + in **Figure 5**). The experiment was performed in the following order of regimes: fast (red), slow, high amplitude (green), slow (blue), slow (blue, dotted), fast, low amplitude (pink), fast (red, dotted).

The "adaptation" (the neuron transmits as much information in the S or F regime, but with different firing rates) seen in **Figure 4** can be seen in 4 ($\circ, \diamond, \square, +$) out of 6 neurons. The other two neurons ($\times, \triangle$) show a very low response in the slow state.

The firing rate of the neurons depends strongly on the amplitude of the input μ_q: in the SH (green) and F (red) regime, that used the same value for μ_q (**Table 1**), the neurons show similar firing rates of around 15 Hz (except for a single neuron denoted with $\diamond$). In the SH (pink) and S (blue) regime, which also used the same value for μ_q (**Table 1**), the neurons show low firing rates, with the firing rates of the FL regime, which has very little information about the hidden state in the input, having a lower artificial network firing rate. So the firing rates of the neurons increase with both the amplitude of the input and the amount of information.

3.2.2. Comparison to an Optimal Response Model

Finally, we compared the responses of the recorded neurons to a model of the optimal response for this input (Denève, 2008a; see Section 2.1.3). The parameters of this "Bayesian Neuron" (BN) are determined by the parameters of the input (i.e., r_{on}, r_{off}, and θ), except for parameter η, which determines the firing rate of the model neuron (changing η has a similar effect as changing the reset value and threshold in a leaky integrate-and-fire model).

In **Figure 5B**, we show how the BN performs in a simulation where we used the same input as we used in the experiments, for different values of η. Overall, the BN performs somewhat better than the recorded neurons, as can be expected from an optimal response model. However, as in the *in vitro* experiments, the BN increases its firing rate in the F state relative to the S state to keep fraction of transferred information relatively constant [compare for instance the F (red) and the S (blue) regime for $\eta = 3.5$, denoted with $\triangle$].

In both the experiments and the simulations, the S and SH regimes seem to form a single curve, as do the F and FL regimes. In the Bayesian neuron this makes sense: the switching speed of the hidden state τ is a parameter of the model, the amplitude of the input μ_q is not. So the BN has the same parameters in the S and SH regimes, and the same is true for the F and FL regimes. The observation that these regimes also form a single curve in the experiments, suggest that the recorded neurons also adapt their response properties to the input statistics. The recorded and simulated neurons all transmit less information for a given firing frequency in the F and FL regimes than in the S and SH regimes, because in the F and FL regimes, more spikes are needed because more spikes are needed to represent a fast-switching hidden state.

For a quantitative comparison between the experiments and the BN, we fitted a saturating function to both slow states (green and blue, fits represented by blue lines) and both fast states (red and pink, fit represented by red lines) to the data from both the experiments and the model:

$$FI = 2f_{\text{sat}}\left(\frac{1}{1+e^{-\nu_{\text{sat}}r}} - \frac{1}{2}\right), \tag{19}$$

where r is the firing rate, f_{sat} is the saturation value and ν_{sat} the saturation rate (in s). Since the BN is an ideal observer model, we expect that the BN transmits more information than the experimentally measured neurons: we expect the saturation value f_{sat} to be higher, which is indeed what we find (**Figure 5C**). The closer the experimentally obtained f_{sat} is to the values from the model, the more "optimal" the information transfer of the hippocampal pyramidal cells.

Finally, for both the S and SH curve and the F and FL curve (**Figure 5**, right), there seems to be an optimal value for parameter η of the BN. This means that the BN appears to have an optimal firing rate: for too low firing rates (larger η) the neuron will miss some periods when $x = 1$, whereas for too

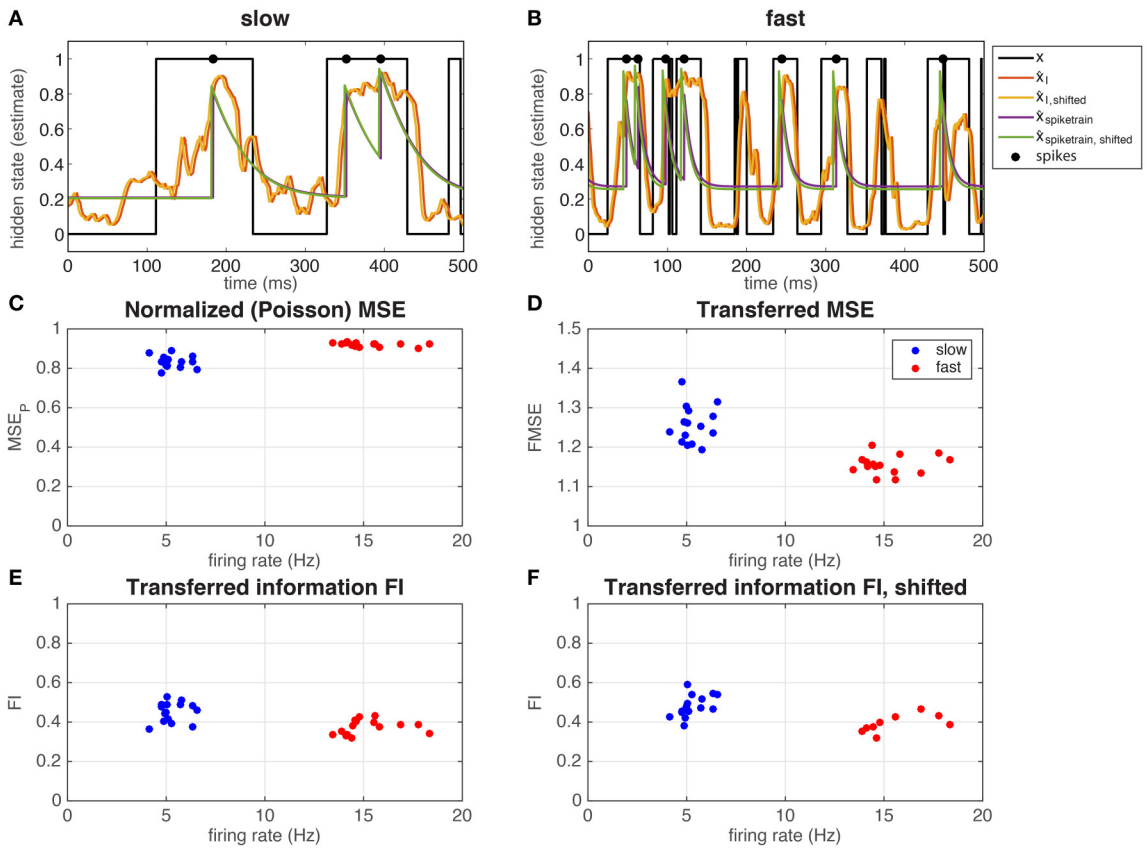


FIGURE 4 | Representation of the hidden state by a single hippocampal pyramidal neuron (depicted with □ in **Figure 5**). The neuron responds to the input by spiking mostly when $x = 1$ in both the slow **(A)** and the fast **(B)** regime. The hidden state (black line) can be estimated from the input (red and yellow line) and from the output spike train (green and purple line), using a correction for the exponential kernel (yellow and green lines, section 2.1.2.6) or not (red and purple lines). In **(C)** the MSE_P (*MSE* normalized by a Poisson spike train, Equation 15) is shown as a function of the overall firing rate, and in **(D)** the *FMSE* (*MSE* of the spike train normalized by the *MSE* of the input, Equation 16). In **(E,F)**, we show the mutual information between the spike train and the hidden state normalized by the mutual information in the input, corrected for the exponential kernel **(F)** or not **(E)**.

high firing rates (smaller η) the neuron will also spike when $x = 0$, making the neuron less informative. This effect is stronger for the fast regimes than for the slow regimes. For all regimes investigated here, this optimal firing rate appears to be around 40 Hz (**Figure 5B**). In the experiments, we scaled the input current to set the firing rate response of the recorded neuron to the probe stimulus to about 12 Hz overall (about 20 Hz when $x = 1$), so this "optimal" firing rate of 40 Hz was never reached. Whether this 40 Hz is optimal for the recorded neurons too, remains to be investigated (see Grienberger et al., 2017 for natural firing regimes for hippocampal neurons).

4. DISCUSSION

An important task of the brain is to infer information about the outside world. Except for sensory receptors, neurons in the brain do not have direct access to sources in the outside world, but have to infer the state of the world from input generated by other neurons. This input from other neurons is often unreliable and noisy (Knill and Richards, 1996; Körding and Wolpert, 2004). Therefore, neurons need enough input samples to keep a reliable estimate. The number of samples can be increased by either increasing the number of presynaptic neurons, or by integrating information over a longer period of time. Which one is feasible or appropriate depends on the characteristics of the local network (How many presynaptic neurons are available? With what frequency do they fire? How informative are they?) and on the characteristics of the outside world itself (How fast does a stimulus change?). Here, we modeled this by creating a current input for a single neuron that has to infer the presence or absence of a hidden state on the basis of noisy Poisson spike-trains of presynaptic neurons. Like in the general case, there is a trade-off between being fast, in which case many sources (presynaptic neurons) are needed, and being precise, in which case a longer integration time is needed, especially if there are not many presynaptic neurons. We propose to use the current stimulus designed here to measure in an *in vitro* setup how single neurons transfer information about a time varying stimulus.

We propose a new method to measure how much information a single neuron transfers from the (current) input it receives

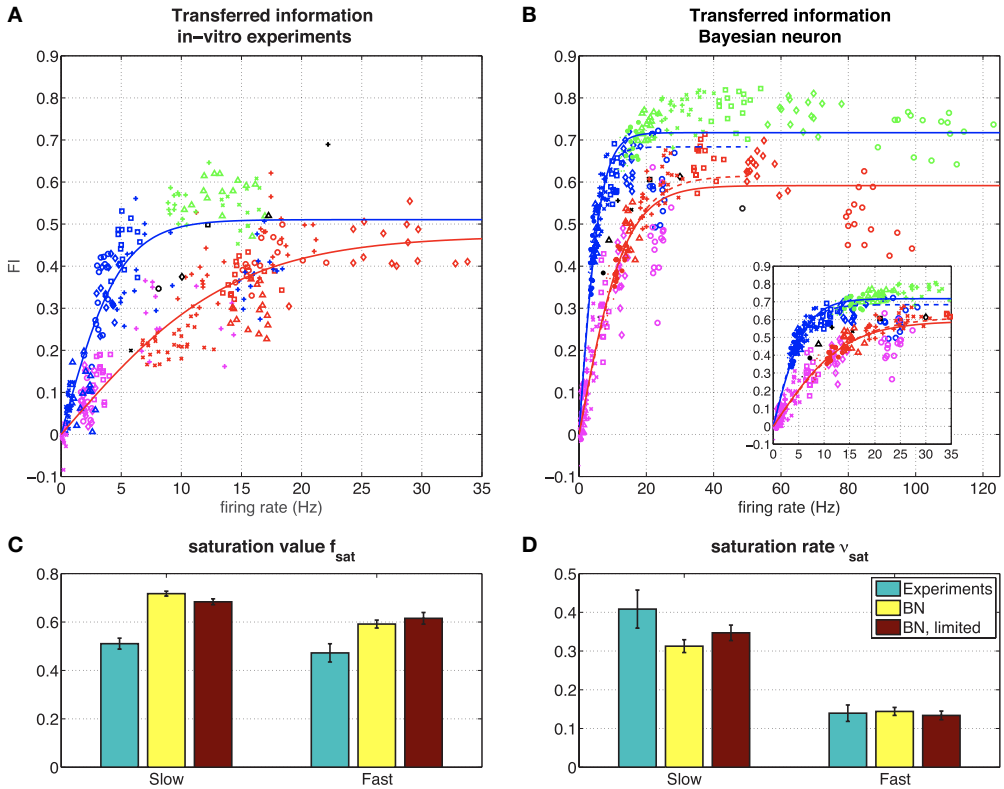


FIGURE 5 | (A) Representation of the hidden state by 6 hippocampal pyramidal neurons (8 measurements). The figure shows the mutual information between the spike train and the hidden state normalized by the mutual information in the input, for 6 different hippocampal pyramidal neurons (depicted with symbols $\circ, \diamond, \square, \times, +, \triangle$), in different regimes (blue: slow, red: fast, green: slow, high amplitude, pink: fast, low amplitude, black: probe). Solid lines: fit with saturating function (Equation 19). NB Note that even though theoretically $MI_I \geq 0$ and $MI_{\text{spike train}} \geq 0$, due to our approximation $\hat{MI}$ can take small negative values, and therefore so can FI. However, these effects are negligable (they only occur when both MI_I and $MI_{\text{spike train}}$ are very small due to vanishing firing rates, which makes $FI \approx \frac{0}{0}$). **(B)** Representation of the hidden state by the Bayesian neuron (Section 2.1.3) for different values of the threshold/reset parameter η. The inset shows the same frequency range as in the experiments. Solid lines are fits using all data, dashed lines ("BN, limited") are fits limited to values for $\eta \geq 2$ (slow) or $\eta \geq 3$ (fast). **(C)** Fitted parameters of saturating function (Equation 19) to data. Error bars denote 95 % confidence intervals of the fit.

to the output spike train it generates. This method is based on generating current input as the response of an artificial population of presynaptic neurons responding to a stimulus randomly switching on and off, and measuring how well this hidden state can be constructed from the output spike train. This gives a lower bound on the mutual information between the spike train (Lochmann and Denève, 2008). This method has several advantages: (1) trials do not have to be repeated, since no estimate of the trial-to-trial variability is needed; (2) since no decoding model needs to be fitted, all recorded data can be used to measure the quantities of interest; (3) for comparison, the properties of an optimal response can be computed easily with the help of the Bayesian neuron (Denève, 2008a); (4) as the method is designed for an *in vitro* setup, stimuli are not limited to sensory stimuli, and neurons outside the sensory systems can be analyzed; (5) since we explicitly control how much information is present in the input, the information loss at the spike generating process itself can be measured; (6) experimental parameters, such as the "time constant of the world" and the number of available sources as discussed above can be systematically varied.

Like any method, the method presented here has several limitations and assumptions. We will discuss these explicitly. Firstly, three assumptions concern generating the input current for the experiments: (1) neurons respond to a randomly appearing and disappearing "preferred stimulus" that (2) they have no access to, and (3) synapses from informative presynaptic neurons are stronger than synapses from non-informative presynaptic neurons. The first two assumptions are comparable to the assumptions that are implicitly made when estimating tuning curves, for instance by fitting filter models such as a Linear–Non-linear Poisson model (Chichilnisky, 2001): in both cases it is assumed that a neuron responds only to the absence or presence (so no history or reverberation effects) of a preferred stimulus feature that it does not have direct access to. However, in the case of filter models, the presence of the preferred stimulus is graded: a preferred stimulus can be "more" or "less" present (i.e., the stimulus can be more or less similar to the preferred stimulus). Here, the stimulus is binary: it is either present or not. Which one is more realistic probably depends on the system in question. Whether the third assumption is realistic depends on the learning rule that was used by the system. Denève (2008b) showed that there exist indeed unsupervised, local, spike-based learning rules by which these synapse strengths could be learned. Secondly, the method requires two additional assumptions for

the output spike trains: (1) an ergodic argument: it is assumed that an average over samples can be replaced by an average over time and (2) it is assumed that spike trains are by approximation Poissonian. The first assumption means that if in an experiment the system is not stationary during the time window for which the mutual information is calculated, the approximation fails. However, such an argument is necessary for almost any experimental measurement. Concerning the second assumption: the estimate of the mutual information is not strongly sensitive to how "Poissonian" the output spike train is, but strong deviations from Poissonian statistics will make the estimate fail. Finally, the fact that we used somatic patch-clamp stimulation, means that we ignored most of the computations that happen in dendritic trees, something that has proven to be substantial in hippocampal pyramidal cells (Spruston, 2008) and that could be essential for the integration of (correlated) inputs (Ujfalussy et al., 2015). This could be partly overcome by using bipolar electrodes and stimulate dendritically, for instance to evoke dendritic calcium spikes. However, the complex spatial distribution of dendritic inputs will be difficult to assess experimentally, although it could be investigated in a biophysical model. Another difference with the natural situation is that normally synaptic input creates conductance fluctuations, which have different (more complex) dynamics than the current injections we used in our model and experiments. For the moment we assume that this difference only creates second order differences.

We designed different input currents with the same amount of information about the hidden state, but with different switching speeds and firing rates (which are realistic for hippocampal neurons, see Grienberger et al., 2017), and injected these into the somata of pyramidal neurons in the CA1 region of mouse hippocampus. We found that the amount of information in the recorded spike trains depended strongly on the firing rate of the neuron: spike trains with more spikes were more informative about the hidden state than spike trains with fewer spikes. However, this effect saturated at around 15 Hz. The slope of the relationship between the firing rate and the mutual information depended on the switching speed of the hidden state: slowly changing inputs were easier to represent, hence contained more information for a given firing rate. However, the neurons responded to two inputs that contained comparable amounts of information about the hidden state, but had different characteristics (a "slow" input with a low amplitude and a "fast" input with a high amplitude) with different firing rates, but kept the amount of information in the recorded output spike trains constant, thereby "adapting[6]" to the characteristics of the stimulus. Strikingly, how much of the information about the hidden state in the input is transferred to the output spike train depended on how informative the input was in the first place: if the input was not very informative, not much information is transferred, whereas a much larger fraction of information about an informative input is transmitted to the output spike train, an effect that is also present in the optimal response of the Bayesian neuron, suggesting that biological neurons approximate an optimal inference process. So the spike-generating process of the recorded neurons has an amplifying effect on information transfer: it reduces the information about a low-informative input stronger than the information about a high-informative input (as explained in the Supplementary Material, the same holds for the relative signal-to-noise ratio: $FS = SNR_{\text{output}}/SNR_{\text{input}}$: the *FS* in response to an input with a low *SNR* is lower than the *FS* in response to an input with a high *SNR*).

The probability density functions of the membrane potential and the input current values show that that the input-current-to-membrane-potential transformation is strongly non-linear and could therefore not be described by for instance a simple leaky integrate-and-fire neuron. The strongly bimodal shape of the membrane potential distribution (as opposed to the input current distribution) can for instance be a result of a saturating (sigmoidal) input-output relation. From this non-linear processing and the amplifying effect on information transfer together we conclude that the neurons we recorded cannot have a simple linear input-output relation, but perform complex transformations on their input. In agreement with this conclusion, Ujfalussy et al. (2015) recently also suggested that the neural computation from presynaptic spikes to the postsynaptic membrane potential should be non-linear for optimal stimulus integration. How such non-linear input-output relationships shape the information processing properties of neurons and how they respond to stimuli with different characteristics (see also Stemmler and Koch, 1999; Brenner et al., 2000; Hong et al., 2008) remains an important topic that needs to be investigated further.

The mutual information between the position of an animal and the spike trains of rat hippocampal CA1 pyramidal cells has been quantified by Barbieri et al. (2004), who also used an estimate of the posterior probability to estimate the mutual information. They concluded that the hippocampal place cells contain a significant amount of information about the location of the animal. However, how much information was present in previous processing layers, and how much information is lost or maintained by these neurons, was not specified. Here, we quantified the information loss of the spike generating process, i.e., the mutual information between the cellular input and the output spike train. In barrel cortex, this information transfer has been quantified, and several studies have shown that spike generation can result in significant information loss (Panzeri et al., 2001; Petersen et al., 2002; Alenda et al., 2010), similar to what has been shown here. In hippocampus, *what* information is encoded in the spike trains has been described extensively since the discovery of place cells (O'Keefe and Dostrovsky, 1971). Moreover, *how* this information is encoded in the spike trains has been suggested to depend on the theta/gamma phase precession (Lisman, 2005). Finally, it has been shown that the nature of this information transfer (for instance the shape of place cell receptive fields) can change significantly, depending on for instance the age of the animal (Tanila et al., 1997). However, *how much* information is transferred by these cells, and how that depends on parameters such as the input characteristics, the state of the network (such as "up" or "down" states or the "high conductance state"; Destexhe et al., 2003) or the presence of neuromodulators such as dopamine

[6]The word "adapting" is between quotation marks, since it is possible that this effect is caused by non-linear but instantaneous processes in the neuron and not by an active adaptive process, compare for instance to Hong et al. (2008).

or acetylcholine (ACh) remains to be quantified. Here, we provide a method to easily measure information transfer or information loss in hippocampus or any other system in an *in vitro* setup.

AUTHOR CONTRIBUTIONS

FZ, SD, and BG designed the method. FZ, WW, and SdK designed the experiments. SdK performed the experiments. FZ wrote the manuscript.

FUNDING

This research was funded by the following grants/institutions: Fondation Pierre Gilles de Gennes, Neuropole Region Île de France (NERF), Amsterdam Brain and Cognition (ABC) Talent Grant, Margaret Olivia Knip Foundation, ERC consolidator grant "predispike," James McDonnell foundation award "Human cognition," and Russian Academic Excellence Project "5–100."

ACKNOWLEDGMENTS

We thank Timm Lochmann for helpful conversations during the early stages of this project.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: http://journal.frontiersin.org/article/10.3389/fncom.2017.00049/full#supplementary-material

REFERENCES

Alenda, A., Molano-Mazón, M., Panzeri, S., and Maravall, M. (2010). Sensory input drives multiple intracellular information streams in somatosensory cortex. *J. Neurosci.* 30, 10872–10884. doi: 10.1523/JNEUROSCI.6174-09.2010

Barbieri, R., Frank, L. M., Nguyen, D., Quirk, M. C., Solo, V., Wilson, M. A., et al. (2004). Dynamic analyses of information encoding in neural. *Neural Comput.* 16, 277–307. doi: 10.1162/089976604322742038

Bialek, W., Rieke, F., de Ruyter van Steveninck, R. R., and Warland, D. (1991). Reading a Neural Code. *Science* 252, 1854–1857.

Borst, A., and Theunissen, F. E. (1999). Information theory and neural coding. *Nat. Neurosci.* 2, 947–957.

Brenner, N., Bialek, W., and de Ruyter van Steveninck, R. R. (2000). Adaptive rescaling maximizes information transmission. *Neuron* 26, 695–702. doi: 10.1016/S0896-6273(00)81205-2

Chichilnisky, E. J. (2001). A simple white noise analysis of neuronal light. *Netw. Comput. Neural Syst.* 12, 199–213. doi: 10.1080/713663221

Cover, T. M., and Thomas, J. A. (1991). *Elements of Information Theory, 2nd Edn.* Hoboken, NJ: Wiley-Interscience.

Cruz, A. V., Mallet, N., Magill, P. J., Brown, P., and Averbeck, B. B. (2011). Effects of dopamine depletion on information flow between the subthalamic nucleus and external globus pallidus. *J. Neurosci.* 106, 2012–2023. doi: 10.1152/jn.00094.2011

de Ruyter van Steveninck, R. R., and Bialek, W. (1988). Real-time performance of a movement-sensitive neuron in the blowfly visual system: coding and information transfer in short spike sequences. *Proc. R. Soc. Lond. B* 234, 379–414.

de Ruyter van Steveninck, R. R., Lewen, G. D., Strong, S. P., Koberle, R., and Bialek, W. (1997). Reproducibility and variability in neural spike trains. *Science* 275, 1805–1808.

Denève, S. (2008a). Bayesian spiking neurons I: inference. *Neural Comput.* 20, 91–117. doi: 10.1162/neco.2008.20.1.91

Denève, S. (2008b). Bayesian spiking neurons II: learning. *Neural Comput.* 20, 118–45. doi: 10.1162/neco.2008.20.1.118

Destexhe, A., Rudolph, M., and Paré, D. (2003). The high-conductance state of neocortical neurons *in vivo*. *Nat. Rev. Neurosci.* 4, 739–751. doi: 10.1038/nrn1198

Dimitrov, A., Lazar, A., and Victor, J. D. (2011). Information theory in neuroscience. *J. Comput. Neurosci.* 30, 1–5. doi: 10.1007/s10827-011-0314-3

Grienberger, C., Milstein, A. D., Bittner, K. C., Romani, S., and Magee, J. C. (2017). Inhibitory suppression of heterogeneously tuned excitation enhances spatial coding in CA1 place cells. *Nat. Neurosci.* 20, 417–426. doi: 10.1038/nn.4486

Guo, D., Shamai, S., and Verdú, S. (2005). Mutual information and minimum mean-square error in Gaussian channels. *IEEE Trans. Inf. Theory* 51, 1261–1282. doi: 10.1109/TIT.2005.844072

Hasselmo, M. E. (2011). Models of hippocampus. *Scholarpedia* 6:1371. doi: 10.4249/scholarpedia.1371

Hong, S., Lundstrom, B. N., and Fairhall, A. L. (2008). Intrinsic gain modulation and adaptive neural coding. *PLoS Comput. Biol.* 4:e1000119. doi: 10.1371/journal.pcbi.1000119

Knill, D. C., and Richards, W. (eds.). (1996). *Perception as Bayesian Inference*. Cambridge, UK: Cambridge University Press.

Körding, K., and Wolpert, D. (2004). Bayesian integration in sensorimotor learning. *Nature* 427, 244–247. doi: 10.1038/nature02169

Lisman, J. (2005). The theta/gamma discrete phase code occuring during the hippocampal phase precession may be a more general brain coding scheme. *Hippocampus* 15, 913–922. doi: 10.1002/hipo.20121

Lochmann, T., and Denève, S. (2008). Information transmission with spiking Bayesian neurons. *New J. Phys.* 10:055019. doi: 10.1088/1367-2630/10/5/055019

MacKay, D. M., and McCulloch, W. S. (1952). The limiting information capacity of a neuronal link. *Bull. Math. Biophys.* 14, 127–135.

O'Keefe, J., and Dostrovsky, J. (1971). The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. *Brain Res.* 34, 171–175.

Paninski, L. (2004). Maximum likelihood estimation of cascade point-process neural encoding models. *Netw. Comput. Neural Syst.* 15, 243–262. doi: 10.1088/0954-898X/15/4/002

Panzeri, S., Petersen, R. S., Schultz, S. R., Lebedev, M., and Diamond, M. E. (2001). The role of spike timing in the coding of stimulus location in rat somatosensory cortex. *Neuron* 29, 769–777. doi: 10.1016/S0896-6273(01)00251-3

Panzeri, S., Schultz, S. R., Treves, A., and Rolls, E. T. (1999). Correlations and the encoding of information in the nervous system. *Proc. R. Soc. Lond. B* 266, 1001–1012. doi: 10.1098/rspb.1999.0736

Petersen, R. S., Panzeri, S., and Diamond, M. E. (2002). Population coding in somatosensory cortex. *Curr. Opin. Neurobiol.* 12, 441–447. doi: 10.1016/S0959-4388(02)00338-0

Rieke, F., Warland, D., de Ruyter van Steveninck, R. R., and Bialek, W. (1997). *Spikes: Exploring the Neural Code*. Cambridge, MA: MIT Press.

Schultz, S. R. (2007). Signal-to-noise ratio in neuroscience. *Scholarpedia* 2:2046. doi: 10.4249/scholarpedia.2046

Schwartz, O., Pillow, J. W., Rust, N. C., and Simoncelli, E. P. (2006). Spike-triggered neural characterization. *J. Vis.* 6, 484–507. doi: 10.1167/6.4.13

Shadlen, M. N., and Newsome, W. T. (1998). The variable discharge of cortical neurons: implications for connectivity, computation, and information coding. *J. Neurosci.* 18, 3870–3896.

Shannon, C. (1984). Editorial note on "Communication in the presence of noise." *Proc. IEEE* 72:1713.

Simoncelli, E. P., Paninski, L., Pillow, J. W., and Schwartz, O. (2004). "Characterization of neural responses with stochastic stimuli," in *The Cognitive Neurosciences*, ed M. Gazzaniga (Cambridge, MA: MIT Press), 1385.

Spruston, N. (2008). Pyramidal neurons: dendritic structure and synaptic integration. *Nat. Rev. Neurosci.* 9, 206–221. doi: 10.1038/nrn2286

Stemmler, M., and Koch, C. (1999). How voltage-dependent conductances can adapt to maximize the information encoded by neuronal firing rate. *Nat. Neurosci.* 2, 521–527.

Strong, S. P., Koberle, R., de Ruyter van Steveninck, R. R., and Bialek, W. (1998). Entropy and information in neural spike trains. *Phys. Rev. Lett.* 80, 197–200.

Tanila, H., Shapiro, M. L., Gallagher, M., and Eichenbaum, H. (1997). Brain aging: changes in the nature of information coding by the hippocampus. *J. Neurosc.* 17, 5155–5166. doi: 10.1111/j.1600-6143.2008.02497.x.Plasma

Treves, A., and Panzeri, S. (1995). The upward bias in measures of information derived from limited data samples. *Neural Comput.* 7, 399–407.

Ujfalussy, B. B., Makara, J. K., Branco, T., and Lengyel, M. (2015). Dendritic nonlinearities are tuned for efficient spike-based computations in cortical circuits. *eLife* 4:e10056. doi: 10.7554/eLife.10056

Wierenga, C. J., and Wadman, W. J. (2003). Functional relation between interneuron input and population activity in the rat hippocampal cornu ammonis 1 area. *Neuroscience* 118, 1129–1139. doi: 10.1016/S0306-4522(03)00060-5

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Computational Neuroscience

ORIGINAL RESEARCH
published: 20 December 2016
doi: 10.3389/fncom.2016.00132

Implementing Signature Neural Networks with Spiking Neurons

José Luis Carrillo-Medina[1] *and Roberto Latorre*[2]*

[1] *Departamento de Eléctrica y Electrónica, Universidad de las Fuerzas Armadas - ESPE, Sangolquí, Ecuador,* [2] *Grupo de Neurocomputación Biológica, Dpto. de Ingeniería Informática, Escuela Politécnica Superior, Universidad Autónoma de Madrid, Madrid, Spain*

Spiking Neural Networks constitute the most promising approach to develop realistic Artificial Neural Networks (ANNs). Unlike traditional firing rate-based paradigms, information coding in spiking models is based on the precise timing of individual spikes. It has been demonstrated that spiking ANNs can be successfully and efficiently applied to multiple realistic problems solvable with traditional strategies (e.g., data classification or pattern recognition). In recent years, major breakthroughs in neuroscience research have discovered new relevant computational principles in different living neural systems. Could ANNs benefit from some of these recent findings providing novel elements of inspiration? This is an intriguing question for the research community and the development of spiking ANNs including novel bio-inspired information coding and processing strategies is gaining attention. From this perspective, in this work, we adapt the core concepts of the recently proposed *Signature Neural Network* paradigm—i.e., neural signatures to identify each unit in the network, local information contextualization during the processing, and multicoding strategies for information propagation regarding the origin and the content of the data—to be employed in a spiking neural network. To the best of our knowledge, none of these mechanisms have been used yet in the context of ANNs of spiking neurons. This paper provides a proof-of-concept for their applicability in such networks. Computer simulations show that a simple network model like the discussed here exhibits complex self-organizing properties. The combination of multiple simultaneous encoding schemes allows the network to generate coexisting spatio-temporal patterns of activity encoding information in different spatio-temporal spaces. As a function of the network and/or intra-unit parameters shaping the corresponding encoding modality, different forms of competition among the evoked patterns can emerge even in the absence of inhibitory connections. These parameters also modulate the memory capabilities of the network. The dynamical modes observed in the different informational dimensions in a given moment are independent and they only depend on the parameters shaping the information processing in this dimension. In view of these results, we argue that plasticity mechanisms inside individual cells and multicoding strategies can provide additional computational properties to spiking neural networks, which could enhance their capacity and performance in a wide variety of real-world tasks.

Keywords: bioinspired ANNs, neural signatures, subcellular plasticity, multicoding, local contextualization, signature neural network, spiking neuron

OPEN ACCESS

Edited by:
Sander Bohte,
Centrum Wiskunde & Informatica, Netherlands

Reviewed by:
Masami Tatsuno,
University of Lethbridge, Canada
Bodo Rückauer,
ETH Zurich, Switzerland
Timothy Rumbell,
IBM Research, USA

***Correspondence:**
Roberto Latorre
roberto.latorre@uam.es

Received: *10 September 2016*
Accepted: *30 November 2016*
Published: *20 December 2016*

Citation:
Carrillo-Medina JL and Latorre R (2016) Implementing Signature Neural Networks with Spiking Neurons. Front. Comput. Neurosci. 10:132. doi: 10.3389/fncom.2016.00132

1. INTRODUCTION

Biological neural circuits are powerful computational systems that efficiently process a great amount of data in real time with extensive plasticity capabilities. This makes the nervous system a source of inspiration when designing engineered tools. In this sense, many Artificial Neural Network (ANN) paradigms mimicking the computational principles performed by living neural systems have been developed to solve real-world problems (Michie et al., 1994; Bishop, 1995). Nevertheless, the bio-inspiration in most cases is limited to a knowledge about neural information processing that was available more than 60 years ago. A challenge in ANN research is related to incorporate novel bio-inspired information coding and processing strategies to the network design since they can contribute to enhance the network capacity to perform a given task (Rumbell et al., 2014).

Information coding in the nervous system is mainly based on the generation, propagation, and processing of action potentials or *spikes* (Bialek et al., 1991; Kandel et al., 1991; Rieke et al., 1999). Most of the neural computation is driven by these events. The classical view of neural coding emphasizes the importance of information carried by the rate at which neurons discharge action potentials. However, experimental evidence indicates that living neural systems use many different information coding strategies (Rabinovich et al., 2006b; Middleton et al., 2011), which greatly enhances their processing capacity as compared to the classical view. In this scenario, temporal coding emerges as a strategy commonly used by neural systems, emphasizing that, unlike (or in addition to) the firing rate paradigm, neural information may be carried by precise individual spike timings (e.g., see Mainen and Sejnowski, 1995; Lestienne, 1996; Diesmann et al., 1999; Reinagel and Reid, 2002).

Traditional ANN paradigms are mostly based on highly simplified information processing mechanisms derived from the neural coding classical view. However, the growing experimental evidence of the importance of temporal code to explain neural computation gave rise to the *Spiking Neural Networks*, nowadays considered the third generation of ANNs (Gerstner, 1995; Maass, 1997b). In the two previous generations, neuron models employ threshold gates and activation functions, such as sigmoid functions, to propagate analog values to their neighbors. In contrast, spiking neurons communicate and encode information using discrete spikes (Gerstner et al., 1993; Deco and Schürmann, 1998; Maass and Bishop, 2001; Gerstner and Kistler, 2002; Bohte, 2004; Brette et al., 2007; Ponulak and Kasinski, 2011). This allows spiking neural networks to solve computational tasks using a firing-rate based strategy as their analog counterparts (O'Connor et al., 2013; Diehl et al., 2015; Esser et al., 2016), but discrete spiking activity provides additional dimensions for information coding (e.g., time, frequency or phase), which makes ANN of spiking neurons a promising approach for solving complex computational tasks. Theoretical efforts try to illustrate that computing and modeling with these networks may be biologically plausible and computationally efficient (Maass, 1997a; Izhikevich, 2004; VanRullen et al., 2005; Cessac et al., 2010). It has been shown that spiking neural networks are at least as computationally powerful as traditional ANN paradigms (Maass, 1996, 1997a; Natschläger and Ruf, 1998; Ruf and Schmitt, 1998). In applied engineering, spiking ANNs have been successfully used in different practical applications, such as motor control, odor recognition, image classification, or spatial navigation between others (see Ponulak and Kasinski, 2011, for an overview).

Although they are closer to their biological counterparts, most ANN paradigms of spiking neurons do not include relevant computational principles experimentally and theoretically studied in the nervous system. For instance, most neuro-inspired paradigms consider network elements as indistinguishable units; they only implement synaptic learning based on adjusting the synaptic weights (Bohte et al., 2002b; Kube et al., 2008; Ponulak and Kasinski, 2011); and individual units are considered integrators that integrate synaptic input over time until a given threshold is reached. Experimental evidence demonstrates that neural computation does not only include synaptic integration and synaptic plasticity, but also *subcellular plasticity*, i.e., intra-unit mechanisms that allow a neuron to tune its intrinsic dynamics and shape the computation of its output response as a function of the incoming information (Zhang and Linden, 2003; Turrigiano and Nelson, 2004; Davis, 2006; Turrigiano, 2007). Likewise, it is commonly considered that the information arriving to a neuron is encoded through a single code, e.g., the rate or the precise timing of spikes, when the need for several simultaneous codes (*multicoding*) in the nervous system seems to be apparent (Latorre et al., 2006; Kayser et al., 2009; Panzeri et al., 2010). Living cells receive many inputs from different sources and send their output to different neurons too. An effective way to improve communications is combining multiple encoding modalities in the same signal. Not all the readers have to be interested in the same modality at the same time, specially when we talk about multifunctional networks. This kind of information processing requires of local information discrimination/contextualization mechanisms that allow a neuron to process the multiple simultaneous codes in its input signals one by one or simultaneously in order to perform different tasks. Subcellular plasticity emerges as a highly relevant strategy to perform this context-dependent information processing.

Signature Neural Networks represent a novel self-organizing bio-inspired ANN paradigm that incorporates some of these concepts (Latorre et al., 2011). Behind this ANN paradigm, there are three main ideas. (1) Each neuron of the network has a signature that allows its unequivocal identification by the rest of the cells. (2) The neuron outputs are signed with the neural signature. Therefore, there are multiple codes in a message regarding the origin and the content of the information. (3) The single neuron discriminates the incoming information and performs a distinct processing as a function of the multiple codes in the network. Nevertheless, in spite of being inspired in a precise temporal structure, signature neural networks are non-spiking ANN. The main goal of this work is to assess whether the information coding and processing strategies proposed by the signature neural network paradigm are plausible for spiking networks. With this aim, we morph the core concepts of the

existing non-spiking paradigm to build an ANN of spiking neurons.

Bursting activity consists of series of high-frequency spikes that alternate with quiescent periods with only subthreshold activity (Izhikevich, 2006). This is particularly suitable to implement multicoding, since it involves the presence of at least two different time scales that can serve to encode distinct informational aspects. It has been also suggested that the burst length or the number of spikes in a burst can be used by living neurons to encode information (Kepecs and Lisman, 2003, 2004). Information can also be encoded in the intraburst firing pattern. In the bursting activity of the leech heartbeat control circuit, the temporal structure of the first spikes in the burst allows predicting the length and number of spikes of the burst (Campos et al., 2007). Another relevant temporal structure within the burst is the intraburst neural signature, in which the signature neural network paradigm is inspired. *Intraburst neural signatures* are very precise and cell-specific spike timings experimentally observed in the bursting activity of cells of different vertebrates and invertebrates living neural circuits (Szücs et al., 2003, 2005; Garcia et al., 2005; Zeck and Masland, 2007; Brochini et al., 2011). In central pattern generators (CPGs), they depend on the synaptic organization of the network (Latorre et al., 2002; Rodríguez et al., 2002; Szücs et al., 2003). These precise temporal structures coexist in the neural signals with relevant information encoded with other encoding modalities. Their possible functional meaning for the neurons that belong to the same or to other neural system is still an open question. Model simulations of CPG circuits (Latorre et al., 2004, 2006, 2007) point out that they can have important implications for the understanding of the origin of the CPG rhythms, the fast and fine tuning to modulation and the signaling mechanisms to other interconnected systems (other CPGs or muscles that the CPG controls). These modeling results have shown that cell-specific intraburst spike timing can be part of a multicoding strategy of bursting neurons. The readers of these signals may be able to read these characteristic firing patterns to perform different tasks in response to the multifunctional signals from each CPG cell.

In the context of ANN, bursting activity has been labeled as a "non-standard" behavior (Kampakis, 2013). However, taking into account the previous considerations, the individual units of the proposed network have bursting behavior. We argue that the additional dimensions to encode information provided by bursting activity can significantly increase the computational power of a spiking network. In particular, here we consider two encoding schemes in the bursting signals: a rhythmic encoding modality, in which information is carried by the bursting frequency; and a spike-timing encoding modality; in which information is carried by specific intraburst spike patterns. Each individual neuron has a characteristic intraburst neural signature that uses to sign its output signals in the spike-timing encoding dimension. Finally, the model incorporates intra-unit history-dependent processing rules to compute the response in the spike-timing encoding dimension as a function of previous incoming signals. This local contextualization mechanism can be considered a particular case of subcellular plasticity. The idea behind this network design is transforming different stimuli and/or different relevant aspects of the inputs into different coexisting spatio-temporal spaces that encode information in a distributed network form.

The analysis of the emerging collective dynamics and the self-organizing properties of the network discussed in this paper points out that novel bio-inspired processing strategies could enhance the spiking ANNs capacity and performance. In particular, we provide a proof-of-concept that combining multiple encoding modalities in the network allows transforming incoming data into different spatio-temporal spaces, from which different aspects of the data, including their source, could be exploited one by one or globally. Different collective processing strategies can be implemented in each information dimension only by tuning the synaptic or intra-unit parameters, which facilitates parallelism and multifunctionality in the network. All these features would potentially increase the computational power of spiking ANNs and their ability to model complex high-dimensional processes.

2. MODELS AND METHODS

2.1. Network Model

Signature neural networks use neural fingerprints to identify each individual unit of the ensemble (Latorre et al., 2011). For the spiking network proposed here, we take inspiration from the CPG circuits and use interspike interval signatures to achieve this feature. Thus, the fingerprint of a neuron (n_i) is a cell-specific intraburst spike timing distribution described as the sequence $S_i = \{ISI_1, ISI_2, ..., ISI_n\}$, where ISI_n represents interspike intervals between consecutive spikes within the same burst. The timing of the last spikes in the bursting activity of the pyloric CPG cells varies from one burst to another; while the first spikes in the burst are highly reliable (Elson et al., 1999; Varona et al., 2001a,b) and contain the neural signature (Szücs et al., 2003, 2005). Mimicking this behavior, we consider two parts in a burst. The first part is used to sign the output messages and contains the signature of the emitter neuron (S_i). The spike timings of the second part of the burst are given by a preferred output pattern ($P_i = \{t_0 = 0, t_1, t_2, ..., t_N\}$) that changes dynamically as a result of the single neuron plasticity (see Section 2.1.3).

Spiking-bursting activity allows the simultaneous propagation of different units of information throughout the network (multicoding). Therefore, different spatio-temporal spaces can be simultaneously used to globally encode and store information. In the network discussed in this paper, we consider two coexisting units of information in each neural signal: the bursting frequency and the neural fingerprints included within the burst. In the first dimension, the network must generate and coordinate spatio-temporal patterns of propagating transient bursting activity (rhythmic encoding modality). To achieve this, we impose two constraints (Wiedemann and Lüthi, 2003; Tabak et al., 2010): (i) predominance of excitatory synapses and (ii) a refractory period in each neuron following hyperexcitation. Information processing in the second dimension is based on the emission and recognition of specific neural signatures (Tristán et al., 2004; Carrillo-Medina and Latorre, 2015), i.e., information

in this dimension propagates encoded in a spike-timing modality. An intra-unit contextualization mechanism drives the signature emission and recognition processes. This does not only allow us to illustrate a novel information processing strategy in the context of spiking neural networks, but also the dynamical richness that subcellular plasticity can provide to these networks.

2.1.1. Neuron Spontaneous Dynamics

Many spiking models generate output bursts depending on the parameter settings and/or the input stimuli (e.g., the models by Hindmarsh and Rose, 1984, Komendantov and Kononenko, 1996, or Liu et al., 1998 that we have previously used to investigate the functional meaning of neural signatures). However, simulations show that the neural signatures in these models mainly depends on the network connectivity (Latorre et al., 2002) and, to our knowledge, none of the existing spiking models displays an adaptive fingerprint as required by our study. A possible alternative to this issue is using the mechanism described in Marin et al. (2014) to tune neuron busting models and produce neural signatures equivalent to those observed in living cells. However, the generation of realistic signatures is out of the scope of this proof-of-concept.

To describe the individual behavior of each unit, we define a stochastic model operating in a discrete event framework. The neuron activity is considered as a discrete variable and characterized in time by $V(t)$, its "membrane potential." **Figure 1A** illustrates schematically the neuron spontaneous dynamics. Our model neuron integrates and processes the information received through its different input channels (synaptic integration), adapts its firing pattern to the incoming information (intra-unit plasticity), and generates a coherent signed output signal. During subthreshold activity, the spontaneous evolution of the neuron activity is determined by the probability p—the transit probability of the internal state per time step. When the membrane potential of a neuron n_i reaches the firing threshold (TH), this generates a sequence of spikes (not a single spike). The temporal distribution of spikes within the response burst is given by a firing sequence composed of concatenating the signature (S_i) and the preferred output spike pattern (P_i) of the neuron. Then, the stochastic dynamics of a single neuron depends on the temporal evolution of the neuron activity and whether it is under (subthreshold activity) or over (spiking-bursting activity) the firing threshold. Formally:

- During subthreshold activity ($V_i(t) < TH$):

$$V_i(t+1) = \begin{cases} V_i(t) + I_{syn} + 1 & \text{with probability } p \\ V_i(t) + I_{syn} & \text{otherwise} \end{cases} \quad (1)$$

where I_{syn} is the synaptic input (Equation 3) and p the transit probability of the internal state per time step.

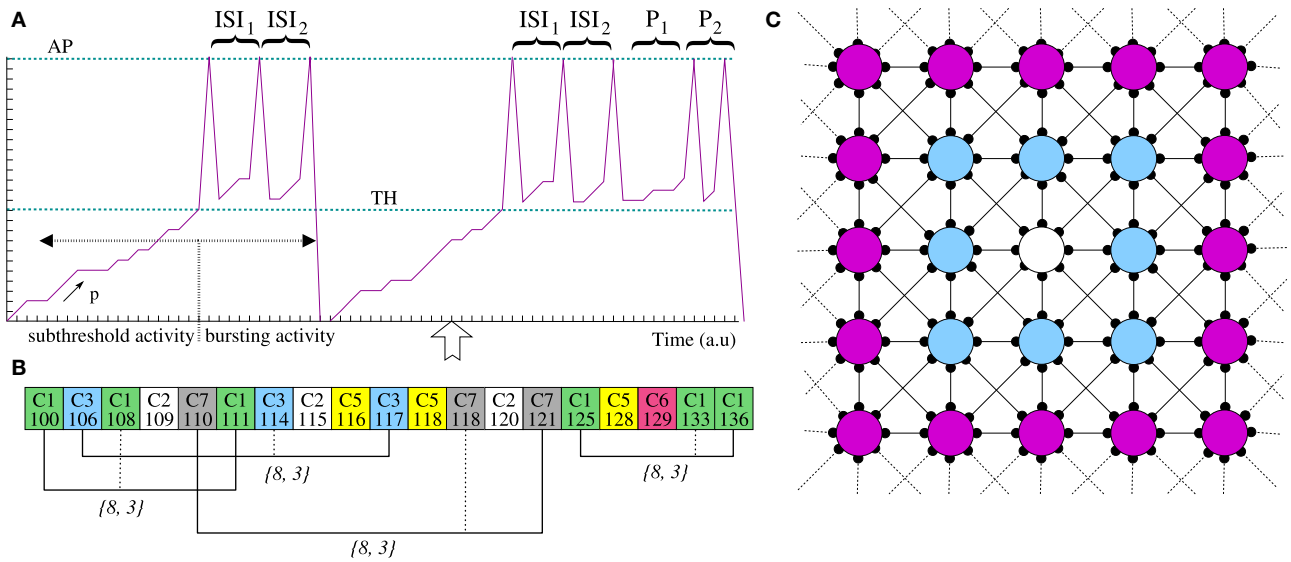


FIGURE 1 | (A) Schematic representation of the stochastic neuron model (see main text for details). $S = \{ISI_1, ISI_2\}$ and $P = \{P_1, P_2\}$ denote the neuron signature and the preferred output pattern, respectively. Note that the intraburst firing pattern is different in the first and the second burst. This is because, as **(B)** illustrates, the neuron recognizes a signature at the time step pointed by the arrow and intra-unit plasticity changes the neuron response in the spike-timing encoding modality. **(B)** Example of signature recognition. For each incoming spike, the local informational context keeps track of the corresponding input channel and spike timing (e.g., C1-100 means that at time step 100 a spikes arrived to the neuron through channel C1). This transient memory provides an intra-unit contextualization mechanism to the single neuron. For example, if the arrow in **(A)** corresponds to time step 136 and an input spike arrives through channel C1, the neuron can contextualize this spike and determine that the signature {8, 3} have been received four times in the recent history. If this value is greater than the learning threshold (L_i), the neuron recognizes this signature and, consequently, modulates its output firing pattern as illustrated in the second burst of **(A)**. **(C)** Network topology. Each neuron is directly connected to its eight nearest neighbors with periodic boundary conditions. Then, neighbors of the white unit are the blue neurons.

- During the generation of the burst ($V_i(t) >= TH$):

$$V_i(t+1) = \begin{cases} AP & \text{if } t = t_1 + t_n \\ TH+1 & \text{if } t = t_1 + t_n + 1\,, \forall n \neq N \\ 0 & \text{if } t = t_1 + t_N + 1 \\ V_i(t)+1 & \text{otherwise with probability } p \\ V_i(t) & \text{otherwise} \end{cases} \quad (2)$$

where N is the number of spikes in the firing sequence ($S_i + P_i$), t_n denotes the timing of the nth spike in this sequence, (i.e., t_1 corresponds to the initial timing of the burst) and being AP the peak membrane potential to generate a spike. Note, that during the burst generation synaptic input (I_{syn}) is not taken into account (cf. Equation 1 and 2). After generating a burst, neurons have a refractory period of RP time steps during which $V_i(t) = 0$. Then, subthreshold dynamics starts again.

2.1.2. Synaptic Input

Synaptic input arrives to a neuron through two kind of input channels: connections with other neurons and an external channel to introduce external stimulation into the network. Each neuron in the network is connected to its eight nearest neighbors (**Figure 1C**) with periodic boundary conditions. As in every spiking neural network, neurons communicate with each other through the generation and propagation of spikes. Then, the interchange rule is defined by:

$$I_{syn} = g_e \cdot pulse_e + \sum_j g_{ji} \cdot pulse_j \quad (3)$$

where g_e defines the weight of the external stimulus, $pulse_e$ is 1 when an action potential is delivered through the external channel and 0 otherwise; and, similarly, g_{ji} is the weight of the connection between neurons n_j and n_i and $pulse_j$ is 1 when $V_j(t-1) = AP$ and 0 otherwise. Note that Equation 3 does not apply neither during the generation of a burst (Equation 2) nor during the refractory period, i.e., in these situations synaptic input is not considered.

It is important to highlight that in this paper we do not discuss synaptic learning (see Section 4). This implies that g_{ji} is constant for all the synapses and, consequently, the neighborhood of every neuron does not change.

2.1.3. Intra-Unit Plasticity

Incorporating subcellular plasticity to a neuron model implies that a mechanism inside the cell allows tuning the neuron dynamics to incoming signals and/or to particular processing states. We consider here a history-dependent contextualization mechanism driving the spike-timing encoding modality. This intra-unit contextualization modulates the preferred output pattern as a function of previous incoming spike patterns.

As in the non-spiking signature neural network paradigm, to implement local contextualization, each individual neuron uses a transient memory, called *local informational context*. The local informational context keeps track of the information received during a time window of M_i time units, providing a history-dependent contextualization mechanism to the single neuron processing. In the case of our spiking network, for each incoming spike, the neuron stores in its local context the joint information about the input channel and the spike timing (**Figure 1B**). In this way, different intra-unit plasticity rules can be defined to take into consideration the input spike timings. In particular, the following rule can be used to recognize specific neural signatures:

- when a spike arrives to a target unit, this checks whether the spike pattern received though the corresponding input channel appears in its local informational context so many times as a given learning threshold, L_i. If so, the receptor recognizes this fingerprint, which implies that the preferred output pattern is overwritten with the recognized fingerprint.

Figures 1A,B illustrate how intra-unit plasticity tunes the output firing pattern in response to the fingerprint recognition. During the generation of the first burst in the time series, the neuron does not recognize any signature. Therefore, there is not a preferred output pattern and the burst only contains the signature of the neuron ($S = \{ISI_1, ISI_2\}$). At time step 136 (pointed by the arrow), an spike arrives through channel C1. The neuron can use its local informational context (**Figure 1B**) to contextualize this spike. In our case, this means to identify the incoming pattern through this channel (in this case $\{8, 3\}$) and to determine that this fingerprint has been received four times from time step 100. Then, assuming that the learning threshold is $L_i = 4$, the neuron's preferred output spike pattern changes due to the recognition of the signature $S' = \{8, 3\}$. As a consequence, the intraburst firing pattern of the second burst in the time series varies to encode additional information (in the example, the sequence $P = \{P_1 = 8, P_2 = 3\}$). The neuron emits the new preferred output pattern until a new fingerprint is recognized or until the recognized fingerprint appears less than L_i times in the local informational context (keep in mind that this is transient memory). Note, that intra-unit plasticity can be used to compute different aspect of the output signal as a function of the local contextualization, not only the spiking firing pattern. For instance, a particular cell could increase/decrease its level of activity or generate an output spike in response to specific incoming patterns independently of the synaptic weight.

During the input processing, channels are checked randomly in each iteration. In this way, when the target neuron recognizes multiple signatures in the same iteration, the last processed prevails over the others. Plasticity rule does not apply during the generation of a burst—i.e., once the neuron starts firing, the output spike pattern cannot change.

2.2. Analysis Methods

2.2.1. Rhythmic Encoding Modality

To illustrate the spatio-temporal patterns generated in the bursting informational dimension, we generate activity movies representing the membrane potential evolving dynamics. In these movies, the evolution in time of the activity of a given unit ($V_i(t)$) is represented with a color scale. Regions with the same color have synchronous behavior. Red corresponds to neurons with a membrane potential over the firing threshold ($V_i(t) > TH$), i.e., they are generating a burst. Intermediate colors between blue and red, represent subthreshold activity. The cooler the color, the lower the level of activity.

Spatio-temporal patterns of spiking or spiking-bursting activity in one dimensional signals are usually detected and analyzed by means of spectral methods. However, in higher dimensions, the coefficients produced by the multidimensional Fourier transform are hard to interpret. On the other hand, wavelet-based techniques have proven to be useful tools for signal analysis (Stollnitz et al., 1996; Mallat, 1999). Unlike the Fourier transform coefficients, the wavelet transform coefficients are determined both by a resolution component and a time (or space) component and, therefore, they represent the resolution content at a given portion of the original signal. Thus, to quantitatively characterize the bursting rhythmic activity in our network, we perform a wavelet-based analysis.

In particular, we use the same discrete wavelet transform (DWT) analysis employed in Latorre et al. (2013a) to characterize the global network dynamics of a model of the inferior olive. The method consists in considering the spiking-bursting spatio-temporal patterns produced by the network as sequences of images evolving in time. As a first step in the characterization, a two-dimensional basis is generated by direct Cartesian product of the one-dimensional Haar basis (Stollnitz et al., 1996). Then, the two dimensional non-standard DWT is calculated for each frame of network activity. The idea behind this characterization method is that the number of wavelet coefficients in a given frame, $C(t)$, provides an estimation of the complexity of the image corresponding to the spatio-temporal pattern at time t. A low number of coefficients means that the image is smooth or is composed of smooth components. In contrast, a high number of coefficients corresponds to complex images. In this way, the DWT analysis transforms the multidimensional spiking-bursting activity in the network, $V_i(t)$, into a one dimensional signal, $C(t)$. This signal provides an useful characterization of the bursting dynamics in which both the frequency and the spatial complexity can be discussed. From the frequency perspective, a simple visual inspection of the evolution of $C(t)$ allows to detect the presence of different rhythmic patterns in the network. Furthermore, these rhythms can now be studied by means of the one dimensional Fourier transform. From the spatial complexity of the patterns, very high values of $C(t)$ correspond to almost random behavior of every neuron, with no patterns present; intermediate high values indicate the presence of complex spatial structures in the patterns; while completely synchronized networks produce a small number of coefficients. Note, that $C(t)$ ranges between 0 and the number of neurons in the network.

2.2.2. Spike-Timing Encoding Modality

The spike-timing encoding is related to the spreading of specific intraburst spike patterns through the network and the synchronization mechanisms that allow a group of neurons to recognize and emit the same signature at a given moment (Tristán et al., 2004; Carrillo-Medina and Latorre, 2015). To graphically illustrate the dynamic spatial organization of the spike patterns within the network, we generate activity movies representing the fingerprint-based evolving dynamics (e.g., see **Figure 5**). Each point in the 50×50 square represents with a color code the neural signature recognized by a given neuron within the network at a given moment. In this manner, neurons with the same color recognize the same signature. White color identifies the units that do not recognize any fingerprint.

To quantitatively analyze this encoding strategy, we compute the evolution of the number of neurons that recognize and emit each individual signature per time unit. This measure provides an estimation over time of the level of activity in the network related to each signature.

3. RESULTS

We have conducted experiments in which multiple datasets are presented to regular networks with different parameters. Independently of the network size and the number of neighbors per neuron, it is possible to find a broad range of synaptic weights and neuron parameters allowing the network to simultaneously encode information in the rhythmic and the spike-timing modality. However, the emerging phenomena that we describe here can be more easily illustrated in autonomous networks with a low level of bursting activity, since in these cases, the spatio-temporal activity in the different dimensions arises due to external stimulation. In autonomous networks, i.e., networks not receiving external input, the level of bursting activity depends on the transit probability of the internal state (p), the firing threshold (TH), and the duration of the refactory period (RP). These parameters modulate the ratio of bursts produced by an isolated neuron. The greater the value of the stochastic probability p, the higher the mean bursting frequency. Similarly, the bursting frequency also grows with low values of TH and RP.

Thus, in the following sections, we focus on neurons where $p = 0.05$, $TH = 50$, $RP = 50$, and $AP = 200$ (units are dimensionless). Note, that AP—the peak membrane potential to generate a spike—has no influence on information processing, the only requirement is being greater than TH. We discuss results of square-shaped networks of 50×50 of such units, with periodic boundary conditions and where each unit is connected through an excitatory synapse ($g_{ji} = 1$) to its eight nearest neighbors as shown **Figure 1C**. External stimuli consist of tonic spiking signals at a given frequency introduced into a randomly chosen cell during a give time period. The neural signature of every neuron has six spikes, with all the ISIs in the range 2–12 (dimensionless). These signatures are randomly generated and assigned at the beginning of the simulation. The rest of parameters are specified in the corresponding experiment description. $V_i(0)$ is chosen randomly between 0 and 40 *a.u.* for all neurons in the network.

3.1. Rhythmic Encoding Modality

The degree of synchrony among the membrane potential of the neurons constituting the network characterizes the global spiking-bursting activity in the network. For fixed values of p and TH, the degree of synchrony varies as a function of the synaptic transmission strength among neurons (i.e., g_{ji} in Equation 3). For small values, each neuron fires nearly independently. As synaptic weights grow, the degree of synchrony increases because the generation of a burst in a given unit sequentially propagates to its neighbors and so on (**Figure 2**). The higher synchrony occurs in networks with combinations of firing thresholds and excitatory synapses that allow a target neuron to reach the firing threshold

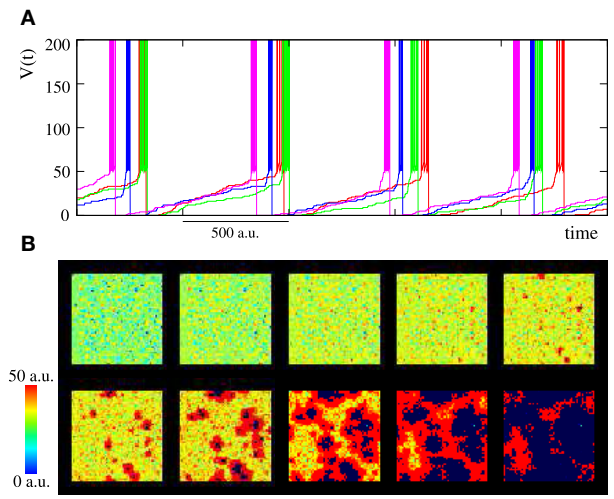


FIGURE 2 | (A) Activity time series of four randomly chosen close neurons in an autonomous network with $M_i = 400$ and $L_i = 4$. Units are dimensionless. Due to the synaptic excitation, the generation of a burst in a given unit propagates to the surrounding units. **(B)** Spatio-temporal patterns of spontaneous activity observed in the network of the top panel. The patterns consist of propagating wave fronts of spiking-bursting activity. Sequences develop in time from left to right and from top to bottom with a time interval between frames of 33 *a.u*.

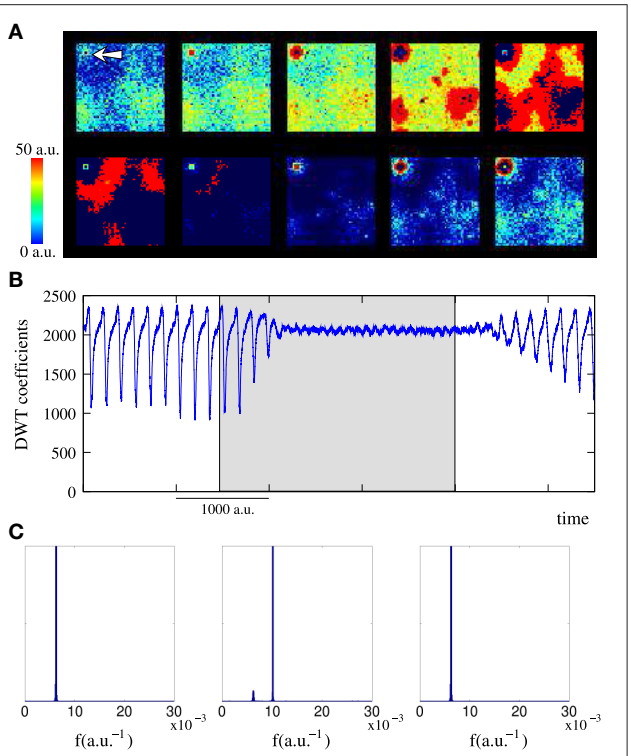


FIGURE 3 | (A) Snapshots of an activity movie illustrating the spiking-bursting spatio-temporal patterns generated by the network of **Figure 2** ($M_i = 400$ *a.u*. and $L_i = 4$) when a tonic input with a period of 100 time units between consecutive spikes is introduced into a single neuron (arrow in the first frame points to the neuron that receives incoming stimulus). Sequences develop in time from left to right and from top to bottom. The time interval between frames is 33 *a.u*. The stimulated unit increases its bursting frequency due to the external stimulation, and this generates new spatio-temporal patterns of transient spiking-bursting activity from this unit. **(B)** Characterization with the DWT coefficients of the activity of the network in the top panel: first without stimuli (snapshots in **Figure 2B** belong to this period), then when the selected neuron receives the incoming data (grayed area identifies the period while the input is active), and finally without any input again. **(C)** Normalized power spectra of the wavelet analysis for the three periods. Left: without stimuli. Middle: during the stimulation. Right: when the stimulation is over and after the reverberation period. Power spectra are calculated using time series of 500,000 time units. The DWT analysis demonstrates that the global network dynamics changes when data are introduced into the network. It also shows that the network is a dynamical working memory of spiking-bursting rhythms, since the network dynamics generated in response to data onset reverberates after the input is retired.

when it receives a burst ($g_{ji} \cdot \#spikes\_in\_burst \geq TH$). However, as we mention above, here we are interested in autonomous with a low level of bursting activity.

Depending on the synaptic parameters, burst propagation provides autonomous networks the ability to generate well-defined spatio-temporal patterns in the form of propagating wave fronts of transient spiking-bursting activity. Note that local contextualization modulates intraburst firing patterns, but it has not any influence on burst timings. To illustrate these spatio-temporal patterns, we generate activity movies representing the membrane potential evolving dynamics (see Section 2.2.1 for details). As representative example of the spontaneous collective bursting rhythms generated by the network, bottom panel in **Figure 2** displays snapshots of the activity movie of the network shown in the top panel.

The spontaneous generation of transient spatio-temporal patterns of spiking or spiking-bursting activity is a feature with relevant functional implications observed in different living neural media. However, we are interested in the network response to stimuli. Therefore, from the encoding perspective, the most interesting feature of the network, appearing even in networks with a small synaptic transmission among neurons, is its ability to develop dynamical patterns of spiking-bursting activity in response to data onset. These patterns allow the network to encode information using the frequency of different bursting rhythms induced by stimuli. To illustrate how the network of **Figure 2** encodes a single input using this spatio-temporal space, **Figure 3A** shows snapshots of its collective spiking-bursting dynamics when a unit in the left-top corner receives an external tonic spiking signal. When the stimulus is introduced into the neuron, its firing frequency increases. Then, the spiking-bursting activity originated in the stimulated neuron propagates to the surrounding units because of excitation. Thus, this neuron becomes the origin of a new rhythm that coexists with those generated spontaneously by the network (if any).

The DWT analysis (Section 2.2.1) corroborates the rhythm encoding in the network transient spiking-bursting dynamics. Changes in the collective spiking-bursting dynamics in response to data onset are reflected in a change in the evolution of the DWT coefficients whose shape characterizes the spiking-bursting activity of the network. As an example, **Figure 3B** illustrates how the collective dynamics of the network in **Figure 3A** changes when data are introduced into the stimulated unit. Initially, no

data is present and the network spontaneously generates spatio-temporal patterns as the ones shown in the bottom panel of **Figure 2**. In this situation, the DWT coefficients oscillate with a nearly homogeneous frequency capturing the spontaneous spiking-bursting rhythm. The spontaneous rhythm frequency depends on the stochastic probability p and can be estimated by means of the Fourier transform of the wavelet analysis of the network activity. For instance, in the network of **Figures 2**, **3**, the spontaneous rhythm frequency is around $6.2 \cdot 10^{-3}$ (see frequency peak in **Figure 3C**, left). On the other hand, the oscillation of the DWT coefficients between a high and an intermediate value indicates, respectively, the nearly independent neuron behavior during subthreshold activity and a high transient synchronization in the network during the spreading of the spiking-bursting wave fronts. Then, the external stimulus is introduced into the network during a given time interval (grayed area). At this point, the network collective dynamics stepwise changes. A first remarkable change in the evolution of DWT coefficients is observed in the oscillation amplitude. Now, the DWT coefficients tend to oscillate around two high values. This change points out the complex spatial structure of the new emerging dynamics. Not obtaining low or intermediate values in the DWT analysis during the stimulation period indicates that, in this network, the propagation of the wave fronts originated in the stimulated unit does not imply a complete transient synchronization in the whole ensemble. Another relevant change in the DWT coefficients during the stimulation is a frequency increase (cf. left and middle power spectra in **Figure 3C**), pointing out that the rhythm evoked by the stimulus prevails over the spontaneous rhythm ($6.2 \cdot 10^{-3}$ vs. $10 \cdot 10^{-3}$ $a.u.^{-1}$). The frequency of the spiking-bursting rhythms evoked by external stimulation depends on the frequency of the input, since the stimulated neuron follows the stimulus. These changes indicate that the network has encoded the incoming information in a characteristic spiking-bursting rhythm. Finally, no input is present again and the network recovers the spiking-bursting autonomous activity (cf. **Figure 3C**, right). The DWT analysis indicates that the stimuli-evoked rhythms can reverberate for long periods after data onset. This implies that the network behaves as a working memory in the spiking-bursting spatio-temporal space. For each network configuration, the mean reverberation period of the rhythms encoding different inputs is nearly the same, i.e., the memory capability of the network in this information dimension is independent of the data and only depends on the synaptic parameters.

The emerging collective dynamics analysis in networks that receive multiple tonic stimuli with different frequencies indicate that spatio-temporal patterns of spiking-bursting activity allow the network to encode information using several coexisting and coordinated rhythms. Top panel in **Figure 4** displays an example of the complex spatial organization of the patterns generated by a network receiving 10 different inputs. The snapshots clearly show the increased complexity of the patterns, since, now, the network organizes clusters of neurons oscillating at different frequencies (cf. top panel in **Figure 3**). Each of the unit receiving external data becomes the source of a rhythm that propagates through the network competing with the rhythms encoding other inputs.

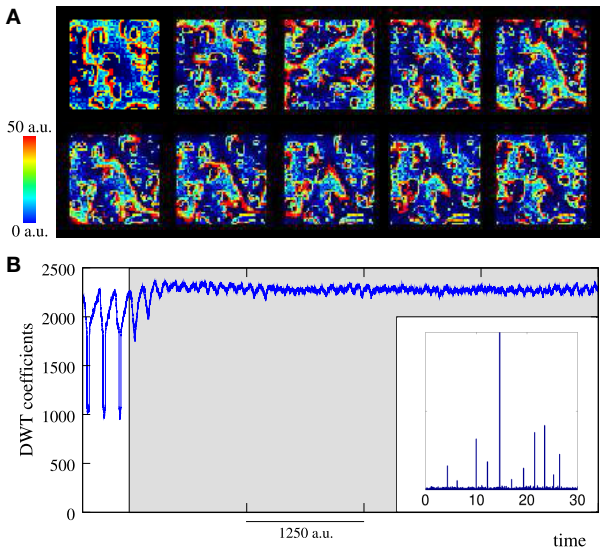


FIGURE 4 | Figure equivalent to Figure 3 but when the network ($M_i = 400$ *a.u.* and $L_i = 4$) receives 10 inputs. (A) Sequences develop in time from left to right and from top to bottom. The time interval between frames is 33 *a.u.* In response to data onset, the network starts generating 10 different coexisting rhythms encoding incoming information. **(B)** The different spiking-bursting rhythms encoded within the network are captured by the DWT analysis. While external inputs are present, the oscillation frequency of the DWT coefficients is not homogeneous (see inset), which reveals the coexistence of the different rhythms. Inset shows the normalized power spectrum of the wavelet analysis of a time series of 500,000 time units while the 10 external stimuli are present. The number of coefficients increase (cf. **Figure 3B**) denotes the increase in the spatial complexity of the patterns.

As we show above, while an input is active, the corresponding rhythm survives in the network. Therefore, when more than one stimulus is present, the competition among the input-evoked spiking-bursting rhythms is a winnerless competition. Note that there is no inhibition in the network nor subcellular plasticity rules limiting the spiking-bursting activity. Winnerless competition allows the encoding of multiple coexisting spiking-bursting rhythms. This competition dynamics is captured by the DWT analysis (bottom panel in **Figure 4**). When multiple data are introduced into the network, the number of DWT coefficients remains high with a non-homogeneous oscillation frequency. This reveals the complex spatial structure of the patterns and, on the other hand, the coexistence of multiple spiking-bursting rhythms within the network.

We have previously shown that the spiking-bursting rhythms evoked by a single stimulus reverberate for a while when the stimulation is over. The reverberation period drastically increases when the network receives multiple stimuli. The greater the number of external inputs, the greater the number of sources of spiking-bursting activity. This translates into a higher spiking-bursting activity in the network and explains the increasing reverberation period. Depending on the synaptic strength and the value of p, in this situation, the network even becomes a long-term memory of spiking-bursting rhythms. We would like to emphasize that the rhythms that survive for longer periods in

short-term memories or the ones that persistently reverberate in long-term memories are not always the higher frequency stimuli-evoked rhythms nor the rhythms encoding the last data presented to the network.

3.2. Spike-Timing Encoding Modality

One of the major characteristics of the proposed network is the intra-unit contextualization of input signals, responsible of the spike-timing encoding modality. In this section, we study the complex collective dynamics induced by this intra-unit information processing strategy. These emerging collective dynamics can give us important clues about the underlying computational properties of the network.

As expected, the ability of an individual unit to recognize specific fingerprints varies as a function of the intra-unit parameters shaping local contextualization, i.e., the maximum size of the local informational context (M_i) and the fingerprint learning threshold (L_i). Depending on the value of these parameters, specific intraburst firing patterns can propagate through autonomous networks. However, the more interesting phenomena from the information processing viewpoint are related to the mechanisms that allow the network to generate and organize spatio-temporal patterns in response to data onset. Therefore, we focus our attention on networks in which the signature recognition does not occur without external stimuli. When these networks receive incoming data, they aid the study of the information encoding in the fingerprint-based spatio-temporal space by analyzing how the signatures of the stimulated units propagate throughout the network.

Again, we first address the analysis of networks receiving a single stimulus. When a neuron receives an external tonic input, this unit increases its bursting frequency (see Section 3.1). This increase can make the neighbor units recognize the neural signature of the stimulated neuron and propagate the corresponding intraburst firing pattern. In this situation, new intriguing collective dynamics arise in the network. To illustrate the dynamic spatial organization of the neural signatures traveling through the network, we generate activity movies representing the fingerprint-based evolving dynamics (see Section 2.2.2 for details). These activity movies point out that the network generates in this dimension well-defined transient patterns of activity in response to data onset. The emerging spatio-temporal patterns are related to the spatial organization and clusterization of the signatures traveling through the network. To give insight into the generation and propagation of these complex spatio-temporal structures, **Figure 5** shows snapshots of the activity movies of two representative networks in which the same unit receives an input. Note, that the only signature traveling through the network corresponds to the stimulated unit. If we consider that at a given moment two neurons that recognize the same signature belong to the same cluster; we can study the specific properties of the dynamic organization of the patterns by calculating the clustering coefficient and the average shortest path between neurons belonging to the same cluster. This analysis indicates that the fingerprint-based spatio-temporal patterns are initially originated in the stimulated unit (see initial frames in the sequences of **Figure 5**). Then, depending on the parameters M_i and L_i, they can propagate locally or globally as transient wave fronts; or as localized clusters with a fixed spatial organization that occasionally become the source of new transient patterns. The generation of localized transient patterns of activity in the fingerprint spatio-temporal space suggests a collective coding strategy based on the emission and recognition of specific neural fingerprints. This mechanism allows the network to encode information regarding the origin of incoming data (input source) in a distributed network form.

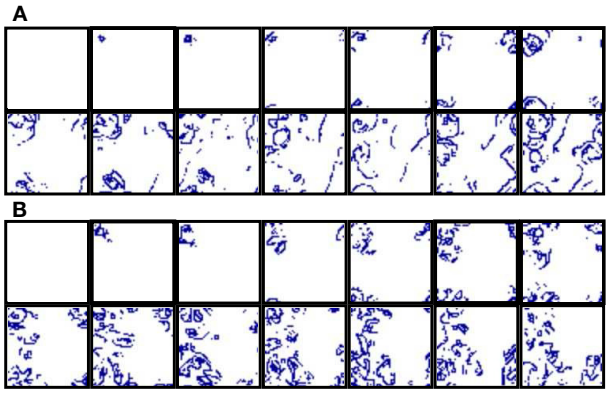


FIGURE 5 | Snapshots of two representative activity movies illustrating the fingerprint-based encoding mechanism. (A) $M_i = 500$ *a.u.* and $L_i = 5$. **(B)** $M_i = 400$ *a.u.* and $L_i = 4$. Sequences develop in time from left to right and from top to bottom. The time interval between frames is 1000 *a.u.* Note that the propagation of the fingerprint-based spatio-temporal patterns is slower than the corresponding spiking-bursting rhythms (cf. bottom panel and **Figure 3**). The color code identifies neurons recognizing the same signature, being white color used for neurons that do not recognize any signature. The first frame in each sequence indicates that, in the absence of stimuli, neural signatures do not propagate in these networks. When the external stimulus is introduced into a neuron located in the left-top corner (second frame in both panels), new collective dynamics emerge and the network organizes transient spatio-temporal patterns of activity related to the propagation of the signature of the stimulated unit (blue regions). Note that this is the only signature that travels throughout the network. These localized patterns of activity encode the *who* of incoming data.

The information encoded in the spike-timing modality and the encoded in the rhythmic modality coexist in the network. A relevant property observed in the simulations is that a neural fingerprint does not necessarily travel over the propagating wave fronts encoding the corresponding spiking-bursting rhythm. The spreading velocity of the fingerprint-based spatio-temporal patterns is always slower than the corresponding spiking-bursting spatio-temporal patterns velocity (cf. time interval between frames in **Figures 3**, **5**; 33 vs. 1000 *a.u*). Likewise, the spatial organization of the patterns in the different spatio-temporal spaces is not correlated. If we consider that at a given moment two neurons over the firing threshold belong to the same cluster, we can calculate the clustering coefficient and the average shortest path for the spiking-bursting patterns and compare the self-organizing properties of the patterns encoded in both information dimension. This analysis points out that the spiking-bursting patterns always consist of propagating transient

wave fronts from the stimulated unit traveling through the whole network. Meanwhile, the fingerprint-based patterns can also be originated in the stimulated unit, but they can propagate locally or globally as transient wave fronts or remain bounded in specific regions of the network.

A simple way to characterize the fingerprint-based dynamics is computing the number of neurons that recognize and emit a given firing pattern. This allows us to identify the signatures encoded in the network. **Figure 6A** depicts the characteristic evolution of the level of activity related to the fingerprint of the data source in three representative networks receiving the same single input during three different stimulation periods. This figure corroborates the results derived from the snapshots shown in **Figure 5**. When the stimulation begins, the signature of the stimulated unit starts propagating through the network. The number of neurons recognizing this fingerprint grows until reaching a stationary level that depends on the value of M_i and L_i. Then, the network dynamics consists of a fluctuation around the steady level (e.g., see blue traces in **Figure 6A**). This dynamic is kept while the stimulation is sustained. When the stimulation ends, the stimulus-evoked activity does not immediately disappear from the network (cf. red and green traces in **Figure 6A**). This is an interesting result that demonstrates that intra-unit contextualization can be a mechanism to implement intrinsic memory in the network, giving rise to both short-term and long-term memories. In short-term memories (bottom and middle panel in **Figure 6A**), the stimuli-evoked dynamics reverberate for a while. This reverberation effect constitutes a mechanism providing the network the ability of acting as a dynamical working memory that transiently stores incoming data. In contrast, in long-term memory networks (top panel in **Figure 6A**), the information survives in the network in a permanent manner (maybe until a new input is received).

The collective dynamics in the fingerprint-based dimension is mainly driven by the intra-unit parameters M_i and L_i. On the one hand, reducing the size of the local informational context of every neuron (M_i) decreases the number of neurons that recognize a given firing pattern. On the other hand, decreasing the learning threshold (L_i) facilitates the recognition of the propagating fingerprints and, therefore, the level of activity in the network grows. The trade-off among the effect of these parameters determines if the network encodes information in the spike-timing modality and the mode of behavior in this dimension. To illustrate this, **Figure 6B** depicts a phase diagram locating the different behaviors in the space of intra-unit parameters.

With the experiments described so far, we investigate the ability of the proposed network to encode and process a single stimulus using an information processing strategy driven by local contextualization. If we repeat the same experiments but now introducing multiple inputs simultaneously, we observe that the presence of multiple stimuli makes the network generate coexisting transient spatio-temporal patterns of activity encoding the origin of the different inputs (**Figure 7**). These experiments reveal additional relevant computational properties that subcellular plasticity can provide to spiking neural networks. When multiple intraburst firing patterns spread through the network, a competition dynamics arises between them. A simple visual inspection of the snapshots shown in **Figure 7** reveals that the self-organizing properties of the patterns drastically change depending on the intra-unit parameters shaping the intra-unit plasticity rules. These define different modes of competition among the spreading fingerprints. This competition affects the global level of activity of each signature in the network and determines the spatial organization of the patterns. The competition dynamics among the different intraburst firing

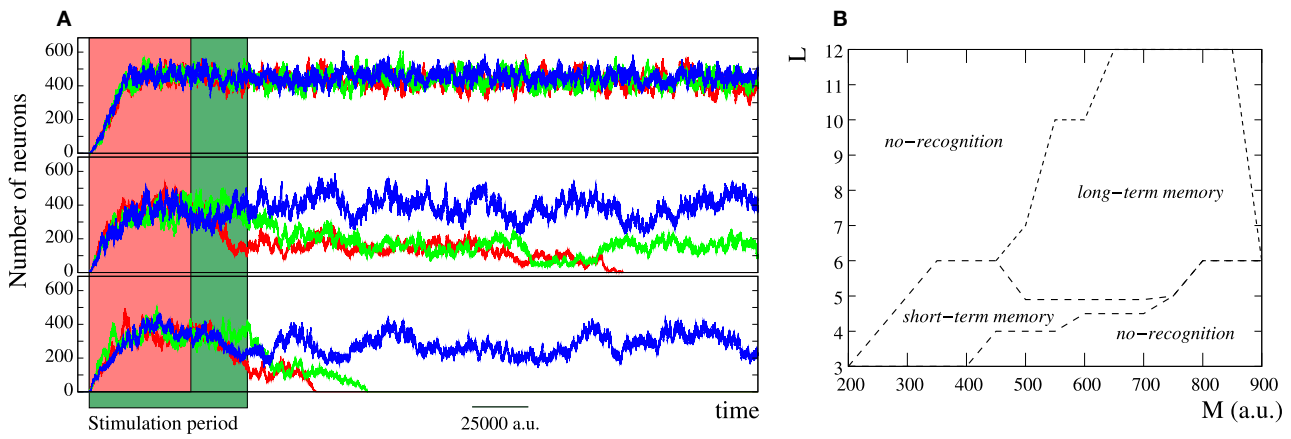


FIGURE 6 | (A) Evolution of the mean number of neurons that recognize and emit the fingerprint of a unit receiving the same data in three different networks during three different periods. Each trace is calculated as the average of 10 experiments with different random seeds and location of the stimulated unit. These plots characterize the stimuli-evoked fingerprint-based dynamics. Top panel: $M_i = 500$ and $L_i = 5$. Middle panel: $M_i = 400$ and $L_i = 4$. Bottom panel: $M_i = 350$ and $L_i = 4$. Units are dimensionless. In red traces, the stimulation period corresponds to the red region. In green traces, to the green region. And in blue traces, data are continuously present. In this spatio-temporal space, the network may act as a long-term memory (top panel) or as a short-term memory (middle and bottom panels) depending on the value of M_i and L_i, i.e., the parameters associated to intra-unit contextualization. **(B)** Phase diagram illustrating the relationship between M_i and L_i in networks where $p = 0.05$, $TH = 50$, $RP = 50$, and $AP = 200$ (units are dimensionless).

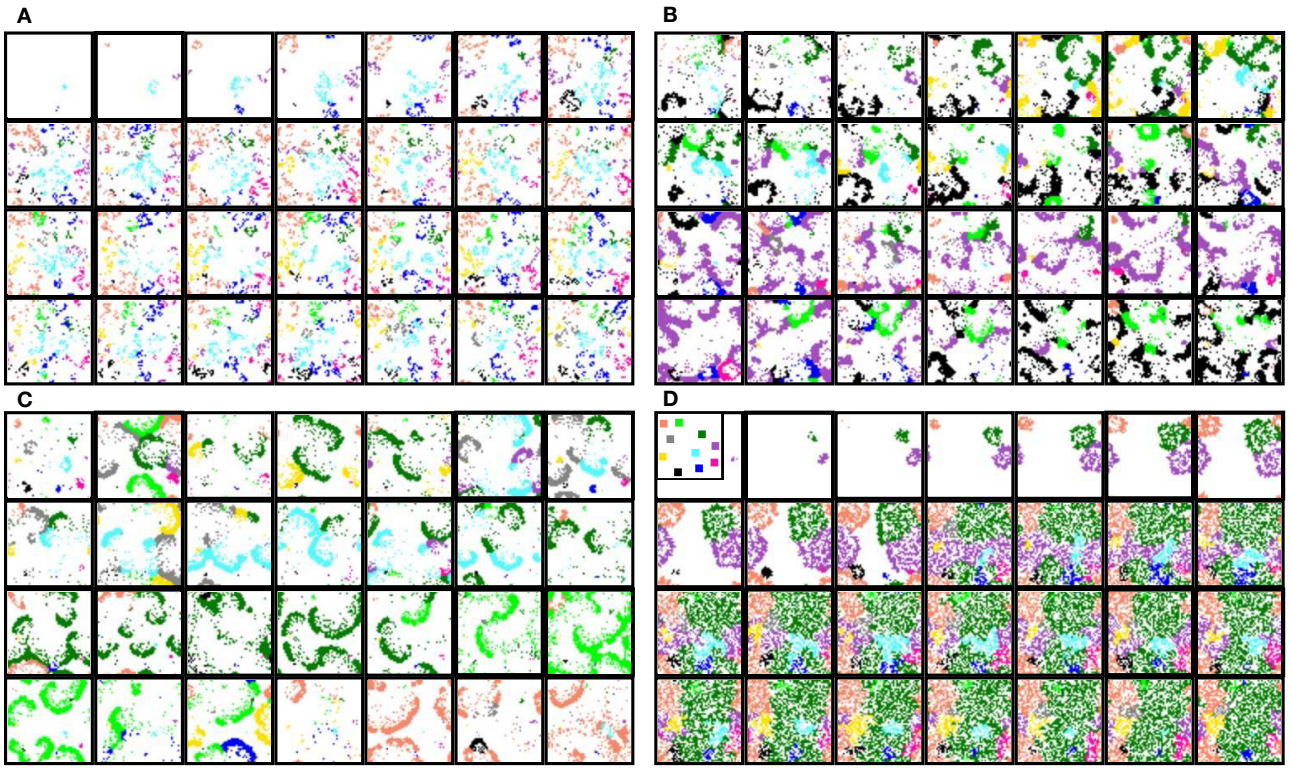


FIGURE 7 | Snapshots of four representative activity movies illustrating the fingerprint-based spatio-temporal patterns generated by networks that receive 10 data simultaneously. The inset in the first frame of **(D)** shows the approximate location of each input. Sequences develop in time from left to right and from top to bottom. The time interval between frames is 2000 *a.u*. Subcellular plasticity induces different competition dynamics among the coexisting patterns in this spatio-temporal space: from winnerless **(A–C)** to winner-take-all **(D)**. These competition regimes are characterized in **Figure 8**. **(A)** $p = 0.05$, $M_i = 400$, and $L_i = 4$. The competition among fingerprints makes the patterns only propagate locally, remaining bounded near the corresponding stimulated unit. **(B)** $p = 0.05$, $M_i = 350$, and $L_i = 4$. Evolving coexisting patterns propagate through the whole ensemble. Each pattern is originated in the unit that receives the corresponding input. **(C)** $p = 0.05$, $M_i = 500$, and $L_i = 5$. The patterns also travel through the whole network, but there exist alternating periods during which only the patterns encoding a given input propagate. After that, a new competing cycle begins until a fingerprint prevails over the others and starts propagating. **(D)** $p = 0.08$, $M_i = 350$, and $L_i = 3$. As result of the competition, only the patterns associated to a limited group of data (the winners) propagate. Note that the different competition regimes arise depending on the values M_i and L_i which shape the intra-unit contextualization mechanism.

patterns determines the coherence and coordination of the coexisting patterns.

We would like to highlight that the competition regimes observed in the activity movies arise in the absence of inhibitory connections, which hints at intra-unit contextualization as an effective mechanism to restrict the activity in networks without inhibition. Note that each neuron can only transmit one recognized firing pattern per burst. This limitation produces somehow a local competition among the patterns received by the neuron where only the "winner" is transmitted. This local competition is the basis of the global competition in the whole network.

The different dynamical modes observed in the activity movies are better characterized by the evolution of the number of neurons that recognize and emit each signature (**Figure 8**). Regardless the number of active inputs, the type of competition depends on the value of the parameters M_i and L_i and may vary from a winnerless (WLC) to a winner-take-all (WTA) competition. In WLC networks, none of the signatures becomes a "winner," and therefore, none of them persistently prevails over the others. Depending on the intra-unit parameters, the network can display different winnerless regimes. **Figure 8A** illustrates a winnerless competition in which the level of activity related to every fingerprint is similar and remains fluctuating nearly a stationary level. This defines a collective dynamics where several coherent spatio-temporal patterns coexist within the network encoding simultaneously a great amount of data (e.g., in **Figure 8A** all the inputs introduced into the network). **Figures 8B,C** show winnerless regimes with alternating periods where some fingerprint has a higher level of activity.

An interesting phenomenon observed with some network settings is that some regions within the network specialize in the emission of firing patterns encoding the origin of different stimuli although they do not receive any external input. This phenomenon occurs without any kind of supervised synaptic nor intra-cellular learning, i.e., it is a self-organizing property of the network. These emitter areas are usually related to winnerless competitions where the prevailing fingerprints change

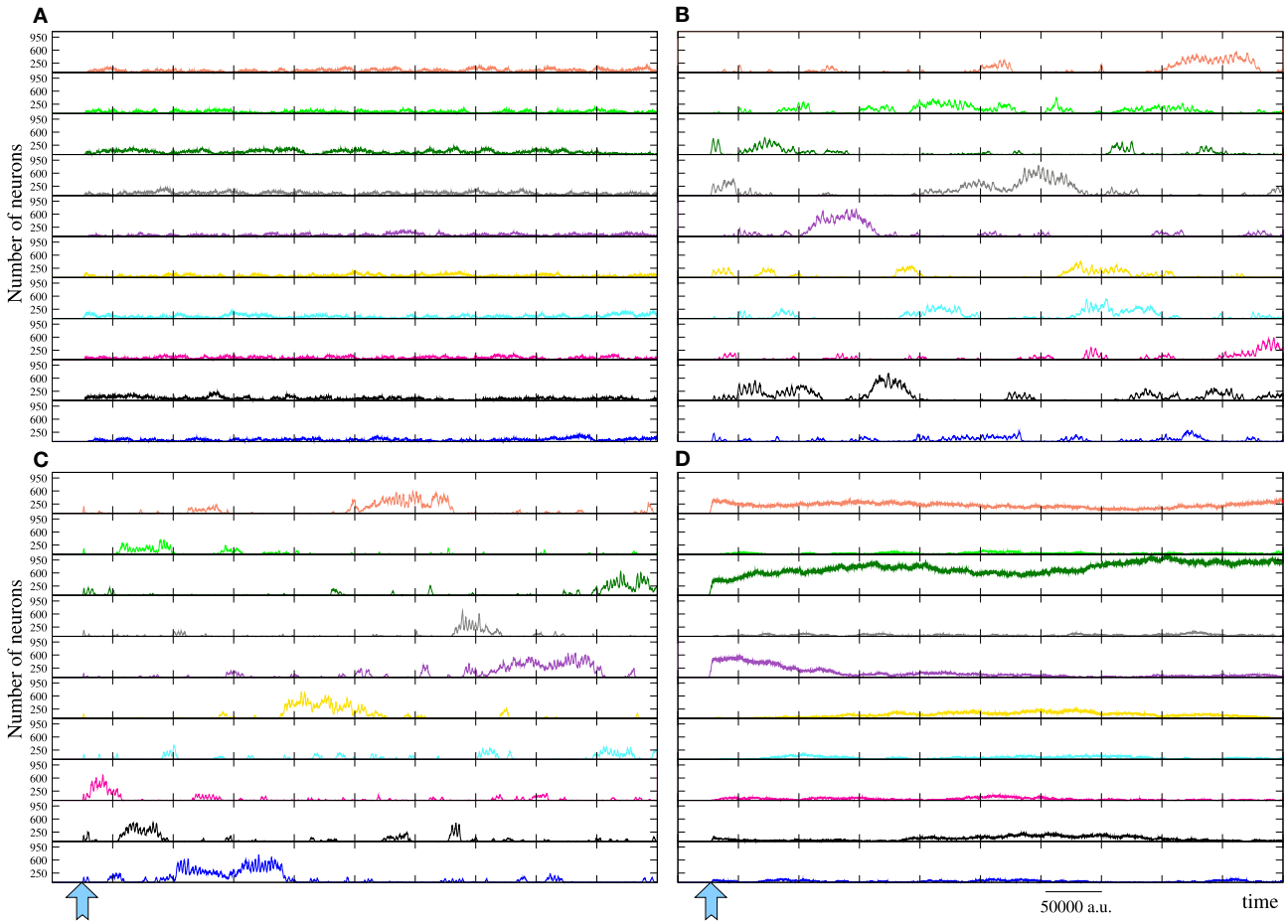


FIGURE 8 | Level of activity related to the 10 neural signatures belonging to the input sources in the networks of Figure 7. The inputs and the color code used to identify them are the same used in this figure. All of them are injected simultaneously from time step pointed out by the arrow to the end of the time series. Each panel corresponds to the equivalent in **Figure 7** and illustrates a different competition regime (see main text for details).

accordingly to the patterns originated in these areas (**Figure 9**). Conversely, when a winner-take-all competition occurs, only the signature or signatures that win the competition propagates through the network (e.g., see **Figure 8D**). In the WTA network shown in **Figure 7D**, all the neurons tend to recognize and emit simultaneously the prevailing fingerprint. However, depending on M_i and L_i, this can also spread as evolving transient patterns equivalent to the shown in **Figure 7C** when the dark green input prevails over the others. Note that, in some sense, the winnerless competitions displayed in **Figures 7B,C** consist of sequences of transient winner-take-all competitions.

The reverberating spatio-temporal patterns encoding the origin of incoming data continue competing even when they are not sustained by an active input. Short-term memory networks have a limited ability to retain previously stored data when new information is introduced into the network. In these cases, the reverberation period drops as compared to networks receiving a single input, and the stored data are almost instantaneously forgotten, i.e., the corresponding patterns disappear because the patterns encoding the last incoming stimulus win the competition. However, in long-term memory networks, coexisting coherent spatio-temporal patterns related to multiple fingerprints can be observed even when the corresponding input is not active.

4. DISCUSSION

The present work introduces a spiking neural network that makes use of multicoding strategies for information propagation and subcellular plasticity to locally contextualize or discriminate data received by a unit. Furthermore, each neuron in the network has a neural signature that allows its unequivocal identification by the rest of the cells. This network is an encoder and generator of spatio-temporal patterns that take advantage of the multiple simultaneous encoding modalities present in the network to transform dynamic inputs into different spatio-temporal spaces, and organize and coordinate coexisting patterns of transient activity in response to data onset.

The discussed experiments are aimed at analyzing the emerging collective dynamics in two information dimensions.

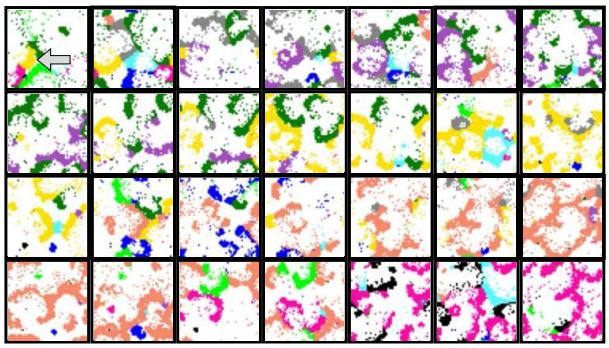


FIGURE 9 | WLC network whose collective dynamics is characterized by an emitter area that generates transient patterns encoding the prevailing fingerprint in the network. Arrow in the first frame denotes the approximate location of this area. Sequences develop in time from left to right and from top to bottom. When neurons in the pointed area start generating patterns encoding a given input, the collective behavior changes accordingly to these patterns and the corresponding fingerprint prevails over the others. Note that the existence of these emitter areas is a self-organizing property of the network.

On one hand, a spiking-bursting spatio-temporal space, where information processing is driven by synaptic transmission. On the other hand, a fingerprint-based spatio-temporal space driven by an intra-unit contextualization mechanism. The specific properties of the dynamic organization of the patterns are different in each information dimension, so that, the life cycle of the information encoded in both encoding schemes is independent. When multiple patterns in the same dimension coexist in the network, a competition emerges between them. We show that various forms of competition can arise without inhibitory connections in the network. Depending on the parameters shaping simple intra-unit plasticity rules, the competition regime may vary from a winnerless (i.e., the network stores multiple data simultaneously) to a winner-take-all competition (i.e., one datum or a group of them prevails over the others). The stimuli-evoked spatio-temporal patterns and the corresponding competing dynamics can survive for long periods after data onset. This reverberation effect allows the network to memorize incoming data. This can display short-term or long-term memory capabilities in the different spatio-temporal spaces. When the network behaves as a short-term memory, the spatio-temporal patterns encoding incoming data in the corresponding scheme transiently reverberate after the stimulation ending. Conversely, in long-term memories, the stimulus leads the network to a new stable state and the patterns persistently survive. The memory ability of the network in each dimension varies as a function of the synaptic and/or intra-unit parameters. Therefore, different simultaneous processing strategies can be implemented within the network.

These results illustrate the dynamical richness and large flexibility of the proposed network to encode and process information in different spatio-temporal spaces. We argue that plasticity mechanisms inside individual cells and multicoding strategies can provide additional computational properties to spiking neural networks, which could enhance their capacity and performance. In particular, local contextualization mechanisms allow individual neurons to process the multiple simultaneous codes in their input signals selectively or globally in order to completely decide or weight the decision about their output in the different encoding schemes. This information processing provides a framework to model complex high-dimensional processes that can be applied to different real-world computational problems. The ideas relating multicoding with local information discrimination have a direct application in problems that benefit from multifunctionality and parallelism. These are desirable features for many technical applications of ANNs, representing a potential advantage when processing large amounts of data or multiple decision-making criteria must be developed, for instance, in multiobjective optimization problems (Saini and Saraswat, 2013; Wang et al., 2014) or in control systems [e.g., multifunctional prosthesis controllers that must quickly detect and classify multiple characteristic simultaneous myoelectric signals (Saridis and Gootee, 1982; Hudgins et al., 1993; Karlik et al., 2003; Li et al., 2010)]. Another straightforward application of these concepts is in problems where a global task is solved by means of solving independent partial tasks. An example is the wide scope of multidimensional sorting problems, specifically when the order in a particular dimension can be independent of the order in other dimensions, or when there is no global sorting criteria in any dimension. Non-spiking signatures neural networks have been successfully applied to this type of problems (Latorre et al., 2011). Areas of application for multidimensional sorting are scheduling, planning and optimization, between others (Catoni, 1998; Aref and Kamel, 2000). On the other hand, the different dynamical modes observed in the network are relevant in the context of multiple technical applications. Winnerless competition is usually associated to sequential information processing (Seliger et al., 2003; Rabinovich et al., 2006a; Arena et al., 2009; Kiebel et al., 2009; Latorre et al., 2013b), which has a wide application in many artificial intelligent systems in tasks such as inference, planning, reasoning, natural language processing, and others (Sun and Giles, 2001; Wörgötter and Porr, 2005). Similarly, pattern recognition in different spiking ANNs is based on winner-take-all dynamics (Bohte et al., 2002a; Gütig and Sompolinsky, 2006; Schmuker et al., 2014).

In this paper, we have imposed some constraints and assumptions in order to facilitate the presentation of our results. Results obtained with larger regular networks (up to 1000×1000); higher levels of bursting activity; and different number and/or distribution of spikes in the neural signatures are equivalent to the results presented in Section 3. In experiments with signatures with an arbitrary number of spikes, new interesting fingerprint-based dynamics emerges in the network and results are not exactly the same. In these simulations, not only the fingerprints belonging to a neuron propagate, but also specific firing sequences built with combinations of these signatures propagate throughout the network. In some sense, these networks do not only encode information regarding the input source, but they also generate new information. It is also important to note that, for simplicity, we only consider two

encoding schemes in the network. However, bursting activity allows easily including additional units of information (e.g., the burst duration or the number of spikes in the burst). In this line, and regarding a selective processing of input messages, experimental evidence indicates that some neural systems exhibit *functional or behavioral neural signatures* representing different states or associated to the task performed at a given moment (Klausberger et al., 2003; Somogyi and Klausberger, 2005; Kaping et al., 2011). The concept of neural fingerprint that underlies the strategy of the discussed network can be extended to consider the emission and recognition of multiple fingerprints with a different meaning within the same signal. In this situation, subcellular plasticity in the form of intra-unit information contextualization mechanisms would allow individual neurons to perform a distinct processing of incoming signals, for example, as a function of specific emitters and/or functional states.

Although not addressed in this paper, subcellular plasticity and multicoding mechanisms for information processing can be combined with the features that underlie information processing in the existing spiking neural network paradigms. In this line, for example, plenty of work has been done on synaptic plasticity in spiking neural networks, since modifications of the synaptic connections are traditionally considered the physiological basis of learning in the nervous system. These works are mostly related to unsupervised synaptic learning methods, such as Spike-Timing Dependent Plasticity (STDP) (Song et al., 2000; Bohte et al., 2002b; Kube et al., 2008; Meftah et al., 2010), with an increasing interest into supervised synaptic learning (Bohte et al., 2002a; Belatreche et al., 2007; Yu et al., 2013). The combination of learning rules including not only the modification of the synaptic weights, but also the parameters that affect the local discrimination of input signals can greatly contribute to enhance the spiking ANNs' computational power. In this vein, our results can be of particular interest in the context of the generation and recognition of spatio-temporal information. Different spiking neural networks have been proposed to process, classify, and store spatio-temporal patterns (Laje and Buonomano, 2013; Yu et al., 2013). We speculate that incorporating multicoding strategies and different types of subcellular plasticity to other successful spiking ANN paradigms can potentially allow these networks to process, classify and store more complex data. For example, a highly relevant application of the referred spiking networks is the analysis of EEG spatio-temporal data. To consider a multicoding mechanism that incorporates the neural fingerprint-based dimension to these networks could permit an analysis of coexisting brain rhythms from multiple simultaneous perspectives. In particular, the fingerprint-based spatio-temporal patterns could facilitate the analysis of the propagation trajectories and the identification of possible information sources and sinks in different cognitive processes.

Because of their functional similarity to biological neurons, spiking neural networks have been extensively used by the computational neuroscience community as a powerful tool for studying neural information processing (e.g., see Izhikevich, 2003; Deco et al., 2008; Izhikevich and Edelman, 2008). Results obtained with our simple model could also be relevant from this perspective Information storage in the nervous system has been typically studied considering the adaptation of the synaptic connection strengths (e.g., see Zipser et al., 1993). Our simulations suggest that mechanisms inside individual cells modulating their intrinsic dynamics could also be an effective mechanism to implement intrinsic memory, both in short- and long-term memory networks. On the other hand, many biological neural systems (including many areas of the human brain) continuously receive a great amount of inputs from many different sources and, nevertheless, they exhibit a low level of activity and only respond to specific inputs (Shoham et al., 2006; Sato et al., 2007; O'Connor et al., 2010; Barth and Poulet, 2012). We hypothesize that neural dynamics based on the propagation of specific neural fingerprints and a contextualization mechanisms like the one studied here could explain why these system are so sparsely active. Target neurons would only fire when they recognize a characteristic firing pattern in their incoming stimuli; while signal not recognized would be simply ignored. Obviously, to test this hypothesis more realistic spiking models for the activity of the neurons must be developed.

AUTHOR CONTRIBUTIONS

RL conceived and designed the study. JC and RL conducted the experiments. JC and RL analyzed the data. JC and RL wrote, read, and approved the manuscript.

FUNDING

This work was supported by UAM-Banco Santander (CEAL-AL/2015-16) and MINECO/FEDER DPI2015-65833-P.

REFERENCES

Aref, W. G., and Kamel, I. (2000). "On multi-dimensional sorting orders," in *Lecture Notes in Computer Science*, Vol. 1873 (Berlin; Heidelberg: Springer), 774–783.

Arena, P., Fortuna, L., Lombardo, D., Pantanè, L., and Velarde, M. G. (2009). The winnerless competition paradigm in cellular nonlinear networks: models and applications. *Int. J. Circ. Theory Appl.* 37, 505–528. doi: 10.1002/cta.567

Barth, A. L., and Poulet, J. F. (2012). Experimental evidence for sparse firing in the neocortex. *Trends Neurosci.* 35, 345–355. doi: 10.1016/j.tins.2012.03.008

Belatreche, A., Maguire, L., and McGinnity, M. (2007). Advances in design and application of spiking neural networks. *Soft Comput.* 11, 239–248. doi: 10.1007/s00500-006-0065-7

Bialek, W., Rieke, F., de Ruyter van Steveninck, R. R., and Warland, D. (1991). Reading a neural code. *Science* 252, 1854–1857. doi: 10.1126/science.2063199

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. New York, NY: Oxford University Press, Inc.

Bohte, S. M. (2004). The evidence for neural information processing with precise spike-times: a survey. *Nat. Comput.* 3, 195–206. doi: 10.1023/B:NACO.0000027755.02868.60

Bohte, S. M., La Poutre, H, and Kok, J. N. (2002a). Error-backpropagation in temporally encoded networks of spiking neurons. *Neurocomputing* 48, 17–37. doi: 10.1016/S0925-2312(01)00658-0

Bohte, S. M., Poutre, H. L., and Kok, J. N. (2002b). Unsupervised clustering with spiking neurons by sparse temporal coding and multilayer rbf networks. *IEEE Trans. Neural Netw.* 13, 426–435. doi: 10.1109/72.991428

Brette, R., Rudolph, M., Carnevale, T., Hines, M., Beeman, D., Bower, J. M., et al. (2007). Simulation of networks of spiking neurons: a review of tools and strategies. *J. Comput. Neurosci.* 23, 349–398. doi: 10.1007/s10827-007-0038-6

Brochini, L., Carelli, P. V., and Pinto, R. D. (2011). Single synapse information coding in intraburst spike patterns of central pattern generator motor neurons. *J. Neurosci.* 31, 12297–12306. doi: 10.1523/JNEUROSCI.1568-11.2011

Campos, D., Aguirre, C., Serrano, E., Rodríguez, F. B., de Polavieja, G. G., and Varona, P. (2007). Temporal structure in the bursting activity of the leech heartbeat CPG neurons. *Neurocomputing* 70, 1792–1796. doi: 10.1016/j.neucom.2006.10.118

Carrillo-Medina, J. L., and Latorre, R. (2015). Neural dynamics based on the recognition of neural fingerprints. *Front. Comput. Neurosci.* 9:33. doi: 10.3389/fncom.2015.00033

Catoni, O. (1998). Solving scheduling problems by simulated annealing. *Siam J. Control Optim.* 36, 1539–1575. doi: 10.1137/S0363012996307813

Cessac, B., Paugam-Moisy, H., and Viéville, T. (2010). Overview of facts and issues about neural coding by spikes. *J. Physiol. Paris* 104, 5–18. doi: 10.1016/j.jphysparis.2009.11.002

Davis, G. W. (2006). Homeostatic control of neural activity: from phenomenology to molecular design. *Annu. Rev. Neurosci.* 29, 307–323. doi: 10.1146/annurev.neuro.28.061604.135751

Deco, G., Jirsa, V. K., Robinson, P. A., Breakspear, M., and Friston, K. (2008). The dynamic brain: from spiking neurons to neural masses and cortical fields. *PLoS Comput. Biol.* 4:e1000092. doi: 10.1371/journal.pcbi.1000092

Deco, G., and Schürmann, B. (1998). The coding of information by spiking neurons: an analytical study. *Network* 9, 303–317. doi: 10.1088/0954-898X_9_3_002

Diehl, P. U., Neil, D., Binas, J., Cook, M., Liu, S.-C., and Pfeiffer, M. (2015). "Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing," in *2015 International Joint Conference on Neural Networks (IJCNN)* (Killarney), doi: 10.1109/IJCNN.2015.7280696

Diesmann, M., Gewaltig, M. O., and Aertsen, A. (1999). Stable propagation of synchronous spiking in cortical neural networks. *Nature* 402, 529–533. doi: 10.1038/990101

Elson, R. C., Huerta, R., Abarbanel, H. D., Rabinovich, M. I., and Selverston, A. I. (1999). Dynamic control of irregular bursting in an identified neuron of an oscillatory circuit. *J. Neurophysiol.* 82, 115–122.

Esser, S. K., Merolla, P. A., Arthur, J. V., Cassidy, A. S., Appuswamy, R., Andreopoulos, A., et al. (2016). Convolutional networks for fast, energy-efficient neuromorphic computing. *Proc. Natl. Acad. Sci. U.S.A.* 113, 11441–11446. doi: 10.1073/pnas.1604850113

Garcia, L., D'Alessandro, G., Fernagut, P.-O., Bioulac, B., and Hammond, C. (2005). Impact of high-frequency stimulation parameters on the pattern of discharge of subthalamic neurons. *J. Neurophysiol.* 94, 3662–3669. doi: 10.1152/jn.00496.2005

Gerstner, W. (1995). Time structure of the activity in neural network models. *Phys. Rev. E Stat. Phys. Plasmas Fluids Relat. Interdiscip. Topics* 51, 738–758. doi: 10.1103/PhysRevE.51.738

Gerstner, W., and Kistler, W. (2002). *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge, MA: Cambridge University Press. doi: 10.1017/cbo9780511815706

Gerstner, W., Ritz, R., and van Hemmen, J. L. (1993). Why spikes? Hebbian learning and retrieval of time-resolved excitation patterns. *Biol. Cybern.* 69, 503–515. doi: 10.1007/BF00199450

Gütig, R., and Sompolinsky, H. (2006). The tempotron: a neuron that learns spike timing-based decisions. *Nat. Neurosci.* 9, 420–428. doi: 10.1038/nn1643

Hindmarsh, J. L., and Rose, R. M. (1984). A model of neuronal bursting using three coupled first order differential equations. *Proc. Roy. Soc. B Biol. Sci.* 221, 87–102. doi: 10.1098/rspb.1984.0024

Hudgins, B., Parker, P., and Scott, R. (1993). A new strategy for multifunction myoelectric control. *IEEE Trans. Biomed. Eng.* 40, 82–94. doi: 10.1109/10.204774

Izhikevich, E. (2006). *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting*. Cambridge, MA: MIT Press.

Izhikevich, E. M. (2003). Simple model of spiking neurons. *IEEE Trans. Neural Netw.* 14, 1569–1572. doi: 10.1109/TNN.2003.820440

Izhikevich, E. M. (2004). Which model to use for cortical spiking neurons? *IEEE Trans. Neural Netw.* 15, 1063–1070. doi: 10.1109/TNN.2004.832719

Izhikevich, E. M., and Edelman, G. M. (2008). Large-scale model of mammalian thalamocortical systems. *Proc. Natl. Acad. Sci. U.S.A.* 105, 3593–3598. doi: 10.1073/pnas.0712231105

Kampakis, S. (2013). Investigating the computational power of spiking neurons with non-standard behaviors. *Neural Netw.* 43C, 41–54. doi: 10.1016/j.neunet.2013.01.011

Kandel, E. R., Schwartz, J., and Jessell, T. M., (eds.). (1991). *Principles of Neural Science, 3rd Edn.* New York, NY: Elsevier Science Publishing Co. Inc.

Kaping, D., Vinck, M., Hutchison, R. M., Everling, S., and Womelsdorf, T. (2011). Specific contributions of ventromedial, anterior cingulate, and lateral prefrontal cortex for attentional selection and stimulus valuation. *PLoS Biol.* 9:e1001224. doi: 10.1371/journal.pbio.1001224

Karlik, B., Tokhi, M. O., and Alci, M. (2003). A fuzzy clustering neural network architecture for multifunction upper-limb prosthesis. *IEEE Trans. Biomed. Eng.* 50, 1255–1261. doi: 10.1109/TBME.2003.818469

Kayser, C., Montemurro, M. A., Logothetis, N. K., and Panzeri, S. (2009). Spike-phase coding boosts and stabilizes information carried by spatial and temporal spike patterns. *Neuron* 61, 597–608. doi: 10.1016/j.neuron.2009.01.008

Kepecs, A., and Lisman, J. (2003). Information encoding and computation with spikes and bursts. *Network* 14, 103–118. doi: 10.1080/net.14.1.103.118

Kepecs, A., and Lisman, J. (2004). How to read a burst duration code. *Neurocomputing* 58–60, 1–6. doi: 10.1016/j.neucom.2004.01.014

Kiebel, S. J., von Kriegstein, K., Daunizeau, J., and Friston, K. J. (2009). Recognizing sequences of sequences. *PLoS Comput. Biol.* 5:e1000464. doi: 10.1371/journal.pcbi.1000464

Klausberger, T., Magill, P. J., Márton, L. F., Roberts, J. D. B., Cobden, P. M., Buzsáki, G., et al. (2003). Brain-state- and cell-type-specific firing of hippocampal interneurons *in vivo*. *Nature* 421, 844–848. doi: 10.1038/nature01374

Komendantov, A. O., and Kononenko, N. I. (1996). Deterministic chaos in mathematical model of pacemaker activity in bursting neurons of snail, helix pomatia. *J. Theor. Biol.* 183, 219–230. doi: 10.1006/jtbi.1996.0215

Kube, K., Herzog, A., Michaelis, B., de Lima, A. D., and Voigt, T. (2008). Spike-timing-dependent plasticity in small-world networks. *Neurocomputing* 71, 1694–1704. doi: 10.1016/j.neucom.2007.03.013

Laje, R., and Buonomano, D. V. (2013). Robust timing and motor patterns by taming chaos in recurrent neural networks. *Nat. Neurosci.* 16, 925–933. doi: 10.1038/nn.3405

Latorre, R., Aguirre, C., Rabinovich, M. I., and Varona, P. (2013a). Transient dynamics and rhythm coordination of inferior olive spatio-temporal patterns. *Front. Neural Circuits* 7:138. doi: 10.3389/fncir.2013.00138

Latorre, R., Levi, R., and Varona, P. (2013b). Transformation of context-dependent sensory dynamics into motor behavior. *PLoS Comput. Biol.* 9:e1002908. doi: 10.1371/journal.pcbi.1002908

Latorre, R., Rodríguez, F. B., and Varona, P. (2002). "Characterization of triphasic rhythms in central pattern generators (i): interspike interval analysis," in *Lecture Notes in Computer Science* (Berlin; Heidelberg: Springer), 160–166. doi: 10.1007/3-540-46084-5_27

Latorre, R., Rodríguez, F. B., and Varona, P. (2004). Effect of individual spiking activity on rhythm generation of central pattern generators. *Neurocomputing* 58, 535–540. doi: 10.1016/j.neucom.2004.01.091

Latorre, R., Rodríguez, F. B., and Varona, P. (2006). Neural signatures: multiple coding in spiking-bursting cells. *Biol. Cybern.* 95, 169–183. doi: 10.1007/s00422-006-0077-5

Latorre, R., Rodríguez, F. B., and Varona, P. (2007). Reaction to neural signatures through excitatory synapses in central pattern generator models. *Neurocomputing* 70, 1797–1801. doi: 10.1016/j.neucom.2006.10.059

Latorre, R., Rodríguez, F. B., and Varona, P. (2011). Signature neural networks: definition and application to multidimensional sorting problems. *IEEE Trans. Neural Netw.* 22, 8–23. doi: 10.1109/TNN.2010.2060495

Lestienne, R. (1996). Determination of the precision of spike timing in the visual cortex of anaesthetised cats. *Biol. Cybern.* 74, 55–61. doi: 10.1007/BF00199137

Li, G., Schultz, A. E., and Kuiken, T. A. (2010). Quantifying pattern recognition-based myoelectric control of multifunctional transradial prostheses. *EEE Trans. Neural Syst. Rehabil. Eng.* 18, 185–192. doi: 10.1109/TNSRE.2009.2039619

Liu, A., Golowasch, J., Marder, E., and Abbott, F. (1998). A model neuron with activity-dependent conductances regulated by multiple calcium sensor. *J. Neurosci.* 18, 2309–2320.

Maass, W. (1996). "Noisy spiking neurons with temporal coding have more computational power than sigmoidal neurons," in *Advances in Neural Information Processing Systems 9, NIPS*, eds M. Mozer, M. I. Jordan, and T. Petsche (Denver, CO: MIT Press), 211–217.

Maass, W. (1997a). Fast sigmoidal networks via spiking neurons. *Neural Comput.* 9, 279–304. doi: 10.1162/neco.1997.9.2.279

Maass, W. (1997b). Networks of spiking neurons: the third generation of neural network models. *Neural Netw.* 10, 1659–1671. doi: 10.1016/S0893-6080(97)00011-7

Maass, W., and Bishop, C. M. (2001). *Pulsed Neural Networks*. Cambridge, MA: MIT Press.

Mainen, Z. F., and Sejnowski, T. J. (1995). Reliability of spike timing in neocortical neurons. *Science* 268, 1503–1506. doi: 10.1126/science.7770778

Mallat, S. (1999). *A Wavelet Tour of Signal Processing*. San Diego, CA; London: Academic Press.

Marin, B., Pinto, R. D., Elson, R. C., and Colli, E. (2014). Noise, transient dynamics, and the generation of realistic interspike interval variation in square-wave burster neurons. *Phys. Rev. E* 90:042718. doi: 10.1103/physreve.90.042718

Meftah, B., Lezoray, O., and Benyettou, A. (2010). Segmentation and edge detection based on spiking neural network model. *Neural Process. Lett.* 32, 131–146. doi: 10.1007/s11063-010-9149-6

Michie, D., Spiegelhalter, D. J., Taylor, C. C., and Campbell, J. (eds.). (1994). *Machine Learning, Neural and Statistical Classification*. Upper Saddle River, NJ: Ellis Horwood.

Middleton, J. W., Yu, N., Longtin, A., and Maler, L. (2011). Routing the flow of sensory signals using plastic responses to bursts and isolated spikes: experiment and theory. *J. Neurosci.* 31, 2461–2473. doi: 10.1523/JNEUROSCI.4672-10.2011

Natschläger, T., and Ruf, B. (1998). Spatial and temporal pattern analysis via spiking neurons. *Network* 9, 319–332. doi: 10.1088/0954-898X_9_3_003

O'Connor, D. H., Peron, S. P., Huber, D., and Svoboda, K. (2010). Neural activity in barrel cortex underlying vibrissa-based object localization in mice. *Neuron* 67, 1048–1061. doi: 10.1016/j.neuron.2010.08.026

O'Connor, P., Neil, D., Liu, S.-C., Delbruck, T., and Pfeiffer, M. (2013). Real-time classification and sensor fusion with a spiking deep belief network. *Front. Neurosci.* 7:178. doi: 10.3389/fnins.2013.00178

Panzeri, S., Brunel, N., Logothetis, N. K., and Kayser, C. (2010). Sensory neural codes using multiplexed temporal scales. *Trends Neurosci.* 33, 111–120. doi: 10.1016/j.tins.2009.12.001

Ponulak, F., and Kasinski, A. (2011). Introduction to spiking neural networks: information processing, learning and applications. *Acta Neurobiol. Exp. (Wars)* 71, 409–433.

Rabinovich, M. I., Huerta, R., Varona, P., and Afraimovich, V. S. (2006a). Generation and reshaping of sequences in neural systems. *Biol. Cybern.* 95, 519–536. doi: 10.1007/s00422-006-0121-5

Rabinovich, M. I., Varona, P., Selverston, A. I., and Abarbanel, H. D. I. (2006b). Dynamical principles in neuroscience. *Rev. Mod. Phys.* 78, 1213–1265. doi: 10.1103/RevModPhys.78.1213

Reinagel, P., and Reid, R. C. (2002). Precise firing events are conserved across neurons. *J. Neurosci.* 22, 6837–6841. Available online at: http://www.jneurosci.org/content/22/16/6837.abstract

Rieke, F., Warland, D., de Ruyter van Steveninck, R., and Bialek, W. (1999). *Spikes: Exploring the Neural Code*. Cambridge, MA: MIT Press.

Rodríguez, F. B., Latorre, R., and Varona, P. (2002). "Characterization of triphasic rhythms in central pattern generators (ii): Burst information analysis," in *Lecture Notes in Computer Science* (Berlin; Heidelberg: Springer), 167–173. doi: 10.1007/3-540-46084-5_28

Ruf, B., and Schmitt, M. (1998). Self-organization of spiking neurons using action potential timing. *IEEE Trans. Neural Netw.* Berlin; Heidelberg: Springer 9, 575–578. doi: 10.1109/72.668899

Rumbell, T., Denham, S. L., and Wennekers, T. (2014). A spiking self-organizing map combining STDP, oscillations, and continuous learning. *IEEE Trans. Neural Netw. Learn. Syst.* 25, 894–907. doi: 10.1109/TNNLS.2013.2283140

Saini, A., and Saraswat, A. (2013). Multi-objective day-ahead localized reactive power market clearing model using {HFMOEA}. *Int. J. Electr. Power Energy Syst.* 46, 376–391. doi: 10.1016/j.ijepes.2012.10.018

Saridis, G. N., and Gootee, T. P. (1982). EMG pattern analysis and classification for a prosthetic arm. *IEEE Trans. Biomed. Eng.* 29, 403–412. doi: 10.1109/TBME.1982.324954

Sato, T. R., Gray, N. W., Mainen, Z. F., and Svoboda, K. (2007). The functional microarchitecture of the mouse barrel cortex. *PLoS Biol.* 5:e189. doi: 10.1371/journal.pbio.0050189

Schmuker, M., Pfeil, T., and Nawrot, M. P. (2014). A neuromorphic network for generic multivariate data classification. *Proc. Natl. Acad. Sci. U.S.A.* 111, 2081–2086. doi: 10.1073/pnas.1303053111

Seliger, P., Tsimring, L. S., and Rabinovich, M. I. (2003). Dynamics-based sequential memory: winnerless competition of patterns. *Phys. Rev. E Stat. Nonlin. Soft. Matter Phys.* 67(1 Pt 1):011905. doi: 10.1103/PhysRevE.67.011905

Shoham, S., O'Connor, D. H., and Segev, R. (2006). How silent is the brain: is there a dark matter problem in neuroscience? *J. Compar. Physiol. A* 192, 777–784. doi: 10.1007/s00359-006-0117-6

Somogyi, P., and Klausberger, T. (2005). Defined types of cortical interneurone structure space and spike timing in the hippocampus. *J. Physiol.* 562(Pt 1), 9–26.

Song, S., Miller, K. D., and Abbott, L. F. (2000). Competitive hebbian learning through spike-timing-dependent synaptic plasticity. *Nat. Neurosci.* 3, 919–926.

Stollnitz, E., DeRose, T., and Salesin, D. (1996). *Wavelets for Computer Graphics: Theory and Applications*. San Francisco, CA: Morgan Kaufmann Publishers Inc.

Sun, R., and Giles, C. (2001). Sequence learning: from recognition and prediction to sequential decision making. *IEEE Intell. Syst.* 16, 67–70. doi: 10.1109/MIS.2001.1463065

Szücs, A., Abarbanel, H. D., Rabinovich, M. I., and Selverston, A. I. (2005). Dopamine modulation of spike dynamics in bursting neurons. *Eur. J. Neurosci.* 21, 763–772. doi: 10.1111/j.1460-9568.2005.03894.x

Szücs, A., Pinto, R. D., Rabinovich, M. I., Abarbanel, H. D., and Selverston, A. I. (2003). Synaptic modulation of the interspike interval signatures of bursting pyloric neurons. *J. Neurophysiol.* 89, 1363–1377. doi: 10.1152/jn.00732.2002

Tabak, J., Mascagni, M., and Bertram, R. (2010). Mechanism for the universal pattern of activity in developing neuronal networks. *J. Neurophysiol.* 103, 2208–2221. doi: 10.1152/jn.00857.2009

Tristán, A., Rodríguez, F. B., Serrano, E., and Varona, P. (2004). Networks of neurons that emit and recognize signatures. *Neurocomputing* 58–60, 41–46. doi: 10.1016/j.neucom.2004.01.020

Turrigiano, G. (2007). Homeostatic signaling: the positive side of negative feedback. *Curr. Opin. Neurobiol.* 17, 318–324. doi: 10.1016/j.conb.2007.04.004

Turrigiano, G. G., and Nelson, S. B. (2004). Homeostatic plasticity in the developing nervous system. *Nat. Rev. Neurosci.* 5, 97–107. doi: 10.1038/nrn1327

VanRullen, R., Guyonneau, R., and Thorpe, S. J. (2005). Spike times make sense. *Trends Neurosci.* 28, 1–4. doi: 10.1016/j.tins.2004.10.010

Varona, P., Torres, J. J., Huerta, R., Abarbanel, H. D., and Rabinovich, M. I. (2001a). Regularization mechanisms of spiking–bursting neurons. *Neural Netw.* 14, 865–875. doi: 10.1016/S0893-6080(01)00046-6

Varona, P., Torres, J. J., Abarbanel, H. D., Rabinovich, M. I., and Elson, R. C. (2001b). Dynamics of two electrically coupled chaotic neurons: experimental observations and model analysis. *Biol. Cybern.* 84, 91–101. doi: 10.1007/s004220000198

Wang, P., Zhu, H., Wilamowska-Korsak, M., Bi, Z., and Li, L. (2014). Determination of weights for multiobjective decision making or machine learning. *IEEE Syst. J.* 8, 63–72. doi: 10.1109/JSYST.2013.2265663

Wiedemann, U. A., and Lüthi, A. (2003). Timing of network synchronization by refractory mechanisms. *J. Neurophysiol.* 90, 3902–3911. doi: 10.1152/jn.00284.2003

Wörgötter, F., and Porr, B. (2005). Temporal sequence learning, prediction, and control: a review of different models and their relation to biological mechanisms. *Neural Comput.* 17, 245–319. doi: 10.1162/0899766053011555

Yu, Q., Tang, H., Tan, K. C., and Li, H. (2013). Precise-spike-driven synaptic plasticity: learning hetero-association of spatiotemporal spike patterns. *PLoS ONE* 8:e78318. doi: 10.1371/journal.pone.0078318

Zeck, G. M., and Masland, R. H. (2007). Spike train signatures of retinal ganglion cell types. *Eur. J. Neurosci.* 26, 367–380. doi: 10.1111/j.1460-9568.2007.05670.x

Zhang, W., and Linden, D. J. (2003). The other side of the engram: experience-driven changes in neuronal intrinsic excitability. *Nat. Rev. Neurosci.* 4, 885–900. doi: 10.1038/nrn1248

Zipser, D., Kehoe, B., Littlewort, G., and Fuster, J. (1993). A spiking network model of short-term active memory. *J. Neurosci.* 13, 3406–3420.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Computational Neuroscience

ORIGINAL RESEARCH
published: 21 April 2017
doi: 10.3389/fncom.2017.00020

Mechanisms of Winner-Take-All and Group Selection in Neuronal Spiking Networks

Yanqing Chen *

The Neurosciences Institute, La Jolla, CA, USA

A major function of central nervous systems is to discriminate different categories or types of sensory input. Neuronal networks accomplish such tasks by learning different sensory maps at several stages of neural hierarchy, such that different neurons fire selectively to reflect different internal or external patterns and states. The exact mechanisms of such map formation processes in the brain are not completely understood. Here we study the mechanism by which a simple recurrent/reentrant neuronal network accomplish group selection and discrimination to different inputs in order to generate sensory maps. We describe the conditions and mechanism of transition from a rhythmic epileptic state (in which all neurons fire synchronized and indiscriminately to any input) to a winner-take-all state in which only a subset of neurons fire for a specific input. We prove an analytic condition under which a stable bump solution and a winner-take-all state can emerge from the local recurrent excitation-inhibition interactions in a three-layer spiking network with distinct excitatory and inhibitory populations, and demonstrate the importance of surround inhibitory connection topology on the stability of dynamic patterns in spiking neural network.

Keywords: neuronal spiking network, phase transition, learning and memory, Winner-take-all (WTA), neural computation, Robotics

OPEN ACCESS

Edited by:
Sander Bohte,
Centrum Wiskunde & Informatica, Netherlands

Reviewed by:
Da-Hui Wang,
Beijing Normal University, China
Victor de Lafuente,
Universidad Nacional Autónoma de México, Mexico

***Correspondence:**
Yanqing Chen
yanqingchen2000@gmail.com

Received: *12 November 2016*
Accepted: *20 March 2017*
Published: *21 April 2017*

Citation:
Chen Y (2017) Mechanisms of Winner-Take-All and Group Selection in Neuronal Spiking Networks. Front. Comput. Neurosci. 11:20. doi: 10.3389/fncom.2017.00020

1. INTRODUCTION

Facing with vast amount of multi-sensory information, Central Nervous System (CNS) seems to process only a small subset of those inputs at any given time, no matter whether they come from external or internal sources. How brain selectively processes such large number of inputs and maintains a unified perception remains a mystery. At the level of neuronal networks, a network in which all neurons respond the same to all stimuli would convey no information about the stimulus. In order to be useful, neurons must come to respond differentially to variety of incoming signals. Many neural models and theories have been proposed to account for such ability. Winner-Take-All (WTA) network is one of such proposed mechanisms for developing feature selectivity through competition in simple recurrent networks, and it has received much attention on both theoretical and experimental grounds. The primary theoretical justification is the ability of such networks to explain how the maps, which are ubiquitous in the cerebral cortex, can arise (Kohonen, 1982; Goodhill, 2007). WTA networks can also explain how a network can come to make useful distinctions between its inputs. WTA networks coupled with synaptic learning rules and homoestatic plasticity can explain how this takes place in a self-organized fashion from an initially undifferentiated state. Finally, WTA models are often employed at the behavioral level in

theoretical models of higher-level cognitive phenomenon such as action-selection, attention (Itti and Koch, 2001; Walther and Koch, 2001) and decision making (Wang, 2002; Furman and Wang, 2008).

Another mechanism proposed for feature selectivity is the phenomenon of spatially localized bumps in neuronal networks (Somers et al., 1995; Laing and Chow, 2001; Wei et al., 2012). If we view multiple neurons within a bump as mulitiple-winners of excitatory and inhibitory competition, bump activity in spiking networks can be treated as a soft WTA or k-Winner-Take-All phenomenon (see Maass, 2000 for their definition). In this paper we use Winner-Take-All (WTA) and "bump activity" inter-changeably to describe the same stable group activity that arises from inter-connected excitatory-inhibitory neuronal networks. On a more general level, both bump activity and WTA phenomenon can be viewed as a type of pattern formation process in networks of excitatory and inhibitory neurons (for example, patterns of stable grid in Wilson and Cowan, 1973; Ermentrout and Cowan, 1979), and an example of activity dependent neuronal group selection process (Edelman, 1987).

Population rate-based WTA models have been extensively studied and are well understood (Dayan and Abbot, 2001). But, the connections between rate models to the real biological neural systems are not direct, because they are different from the real nervous systems whose neurons are spiking. So it is necessary to study the networks of spiking neurons, such that the biological interpretation of spike models can be more directly linked to real nervous systems. Modeling and understanding spiking networks is not simple because spiking neurons are highly nonlinear and their action potentials are discrete. As a result, it is always more difficult to obtain analytical solutions for spiking firing properties than rate models.

Analysis has shown conductance-based spiking models can be approximated by simple rate models under certain conditions (such as in an asynchronous state in Shriki et al., 2003). This approach has been applied to the study of hyper-column in a spiking model of visual cortex (Shriki et al., 2003). The orientation selectivity in their study, is modeled as the appearance of a unimodal "bump"-like spiking activity in a ring-connected spiking network, similar to an earlier study (Laing and Chow, 2001). Both approaches applied approximations from the rate models and used Fourier analysis to calculate the conditions for the appearances of bump activity. Recent work specifically studied recurrent spiking WTA networks, which are closer to real biological systems than previous rate models (Rutishauser and Douglas, 2009; Rutishauser et al., 2011). Even though these newer network models can receive spike input and generate spike output, their network structures are still very simplified. For example, excitatory and inhibitory neurons are modeled into one single population (Laing and Chow, 2001), and inhibitory population are reduced into one unit (Rutishauser et al., 2011), or removed altogether and modeled as direct inhibitory connections among excitatory neurons (Oster et al., 2009).

In a recent report we presented a robust and more biologically-realistic WTA network structure with distinct excitatory and inhibitory populations with arbitrary number of units (Chen et al., 2013). This WTA network has been implemented into a robot that accomplished a sequence learning and mental rotation task (McKinstry et al., 2016). In our spiking models each neuron type has very detailed biological parameters to model different neuronal transmitters and receptor types similar to previous work (Izhikevich and Edelman, 2008). We showed that surround inhibition and longer time constants from NMDA and GABAb conductances are sufficient to achieve stable "bump" spiking activity in a selected winner neuronal group while all the other neurons are inhibited and quiet. However, detailed biological properties, such as STSP (short-term synaptic plasticity), NMDA voltage gating etc., prevented a formal analytical analysis of the whole model. Also, it is not clear any of those biological details or a specific type of synaptic connections are crucial for the emergence of bump activity.

To identify the most important mechanistic factors for the spiking WTA networks, here we study a simplified spiking network after some biological details are removed. For example, based upon what we have noticed previously, turning off STSP, NMDA voltage-gating and excitatory-to-excitatory connections does not change the overall properties of WTA phenomenon. On the other hand, we preserve some important biological features such as the four different synaptic connections and conductance types (AMPA, NMDA, GABAa, and GABAb), because we found that these four individual conductance types contribute to different aspects of the "bump" stability. By examining functions of these individual conductances and the topologies of excitatory-inhibitory connectivity, we provide a detailed analysis of the conditions on which a stable bump activity can emerge from this recurrent spiking network. Our analysis thus provide a mechanistic analysis on how a neuronal group selection process can occur in an activity dependent manner in neural systems.

2. METHODS

2.1. Network Structure

Here we analyze a basic 3-layer spiking neuronal network with different neuron types with realistic biological parameters. The first layer of excitatory neurons (IN – input cells) takes input signals (e.g., arbitrary analog patterns) and translates them into spiking activity. The input signal we considere here in this paper is a type of unstructured random currents evenly distributed within a certain range and injected into the 100 input neurons (IN). IN cells are randomly connected to the next excitatory layer (E) with initial weights evenly distributed between 0 and a maximal value. The random input currents and random connections to the excitatory layer we analyzed here provide a baseline condition in which we test how the recurrent/reentrant connectivity between excitatory and inhibitory neurons by themselves can accomplish winner-take-all competition to random but unstructured input patterns (without obvious firing-rate differences among input neurons) and without synaptic modifications. The successful WTA network structure then can be trained to discriminate more complex and structured patterns through spike-timing dependent learning rules such as STDP. Such learning process will modify the synapses between these two excitatory types so that a selected E and I neurons (the WTA group) will respond to preferred input patterns more

quickly for practical applications. We have demonstrated these in previous reports (Chen et al., 2013; McKinstry et al., 2016) where the same WTA network structures were implemented in a humanoid robot to process real world complex visual inputs, to learn visual-motor association and sequencing, and to accomplish a "mental rotation" and delayed-match-to-sample task.

We also implemented the above network using adaptive exponential spiking models and obtained similar results. For simplicity the analysis below uses the Izhikevich model (Izhikevich and Edelman, 2008), and excitatory (E) and inhibitory (I) neurons use the same parameters in the following equation:

$$C\dot{v} = k(v - v_r)(v - v_t) - u - I_{syn} \tag{1a}$$

$$\dot{u} = a\{b(v - v_r) - u\} \tag{1b}$$

Parameters in these equations are the same as explained before (Izhikevich and Edelman, 2008). That is, v is the membrane voltage in millivolts (mV), C is the membrane capacitance, v_r is the neuron's resting potential, v_t refers to its threshold potential, u represents the recovery variable defined as the difference of all inward and outward voltage-gate currents. I_{syn} is the synaptic current (in pA) originated from spike input from other neurons. a and b are different constants. When the membrane voltage reaches a threshold, i.e., $v > v_{peak}$, the model is said to generate a spike, and two variables in Equations (1a, 1b) are reset according to $v \leftarrow c$ and $u \leftarrow u + d$ while c and d are parameters for different cell type.

We use a simplified synaptic current form with four basic conductances from $AMPA$, $NMDA$, $GABA_A$, and $GABA_B$ channels. For simplification, voltage-gating of NMDA channel is reduced to a constant factor. This is done through calculating an average number for the voltage-gating term for the NMDA conductance (i.e., $[(v + 80)/60]^2/[1 + ((v + 80)/60)^2]$ on Page 11 of the Supplementary Information in (Izhikevich and Edelman, 2008)) for the normal range of voltages: $v = [-60, 60]$, and the result is equivalent to a voltage-independent NMDA channel with smaller gain factor than AMPA channels (see Appendix for the individual conductance gain factors we used). So synaptic current I_{syn} is composed of four different current types originated from those four conductances multiplied with the voltage differences between their individual reversal potentials:

$$I_{syn} = g_{AMPA}(v - 0) + g_{NMDA}(v - 0) + g_{GABA_A}(v + 70) + g_{GABA_B}(v + 90). \tag{2}$$

Shown in the above equation, reversal potentials of $AMPA$ and $NMDA$ channels are 0 and reversal potentials for $GABA_A$ and $GABA_B$ channels are -70 and $-90\,mV$ respectively.

As described before, each conductance has exponential decay with different time constants in millisecond (ms):

$$\dot{g} = -g/\tau, \tag{3}$$

while $\tau = 5$, 150, 6, and 150 for the $AMPA$, $NMDA$, $GABA_A$, and $GABA_B$ channels respectively.

To simplify the analysis, there are equal numbers (400 in all the subsequent analysis) of excitatory (E) and inhibitory (I) neurons in our basic network model in **Figure 1**, although their numbers can be in any ratio. In fact, in our previous published full models (Chen et al., 2013; McKinstry et al., 2016) the ratio of E and I neurons were set at 4:1 to more closely resemble the real cortex. We also explored different types of connection topologies in the connections from excitatory to inhibitory neurons (E to I), the reentrant inhibition from basket cells to pyramidal neurons (I to E) and the inhibitory connections within basket cells themselves (I to I). In our study, Inhibitory to Excitatory and Inhibitory to Inhibitory connections are kept the same topological type and total weights are kept equal. Throughout the simulation the total connection weights to each neuron are normalized to be a constant for each connection type. The total weights for each connection type (E to I and I to E) are two parameters we explored systematically. As a first step, we firstly only consider one type of inhibitory conductance ($GABA_A$) to obtain analytical solutions for the conditions of Winner-Take-All state. $GABA_B$ conductances are added after an analytical solution is found, a comparison of the transition plots can be found in the Appendix.

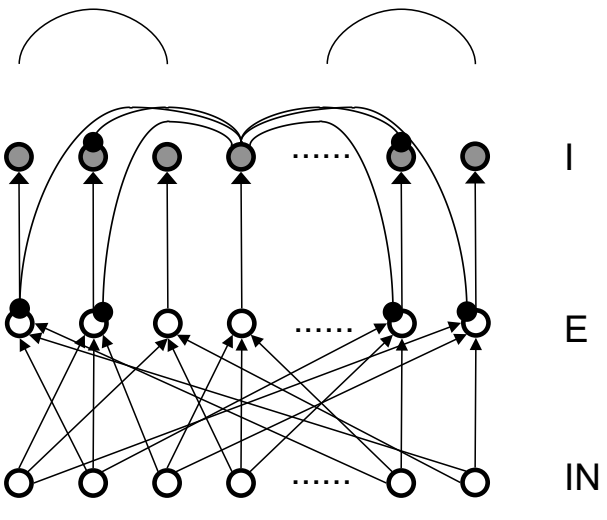


FIGURE 1 | Structure of the basic 3-layer spiking network and a schematic plot of the "surround inhibition" connectivity that supports winner-take-all phenomenon. IN – thalamo-cortical input neurons, E – Excitatory pyramidal neurons, I – Inhibitory neurons. We chose 100 input (IN) cells, 400 E cells, and 400 I cells for total of 900 neurons in the analysis model presented here. Input layer to excitatory layer (IN to E) are all-to-all random connected, excitatory to inhibitory layers are narrow and were simplified into one-to-one connection in our analysis. Inhibitory connections are surround type, that is, I cells do not inhibit its nearest neighboring I and E cells, but only distant surrounding neurons. This connectivity is implemented as two cosine peaks with a flat gap (zero value connectivity) in between. We call this specific network connectivity as surround inhibition type for the one dimensional case and it is a simplified version of the two-dimensional Central-Annual-Surround (CAS) type of topology we described before (Chen et al., 2013).

3. RESULTS

To classify different types of spike dynamics for the surround inhibition network in **Figure 1**, for each neuron, we record the number of spikes between 2 and 3 s after the simulation had reached steady state without synaptic plasticity (STDP off). We then characterize the behavior of the network by the maximum number of spikes generated by any excitatory neuron. **Figure 2E** plots this maximal firing rate for every combination of the E to I weights vs. the I to E weights. The analysis is repeated and plotted in **Figure 2F** for the inhibitory population.

Figure 2 shows different types of dynamic firing patterns in the 2-dimensional parameter space. When only one type of connection weight (excitatory or inhibitory) is high but the other weight is low, either excitatory or inhibitory neurons are in a quasi-random/rhythmic state in which one group of neurons fires in high Gamma frequency range (>40 Hz, see **Figures 2A,G**). When both connection weights are relatively high (see **Figure 2C**), both excitatory and inhibitory neurons have high maximal firing rates where excitatory neurons have a maximal firing rate larger than 35 Hz and inhibitory neurons have a maximal firing rate of larger than 100 Hz. If we look at the corresponding spike raster plot in **Figure 2C**, only a subset of excitatory and inhibitory neurons maintain such high firing rates while majority of other neurons are silent. We call this Winner-Take-All (WTA) state in which only a small subset of neuronal groups persistently fire high frequency and keep the rest of neurons from firing using surround inhibition. The region of the

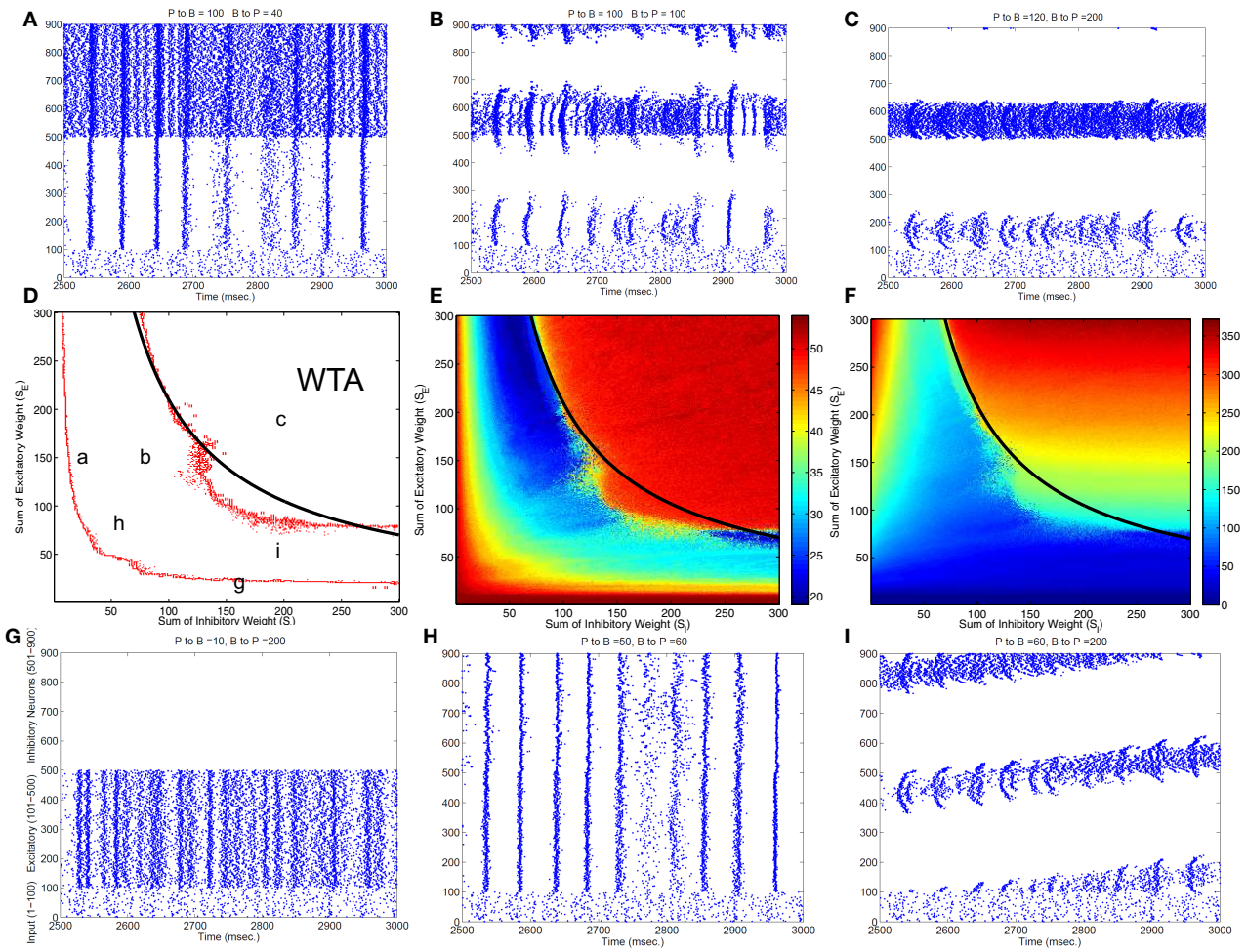


FIGURE 2 | Classification of different dynamic spiking patterns in the surround inhibition network we defined in Figure 1 and the phase diagram for the transition between different firing states. (A–C,G–I) Are raster plots which show all spikes within a half second interval for each neuron in the network. **(E,F)** are maximal firing rates of excitatory and inhibitory neurons in the 2-d parameter space (total excitatory weight in y-axis and total inhibitory weight in the x-axis). Red Points in **(D)** are transition curves constructed from **(E)** where maximal firing rate of excitatory neurons changed from below to above 40 Hz. **(A–C,G–I)** Are example spiking patterns under their specific parameter combinations which are marked on the 2-d plot in **(D)**. Subplot **(C)** represents a winner-take-all (WTA) state where only a small group of excitatory and its corresponding inhibitory neurons fire persistently while others are silent. Black curves on the middle row–**(D,E,F)** are the same analytic transition condition for the WTA region based upon the analysis described in the Appendix (Equation A.14). Basically it is a curve where $S_E \cdot S_I$ equals to a constant defined by Equation (A.14), and it delineates the Winner-Take-All region (i.e., max firing rate large than 40 Hz for only a selected group of excitatory neurons) in the parameter space very well for both the excitatory and inhibitory neurons.

parameter space with such WTA states is delineated by the right curve in **Figure 2D** where the maximal firing rate of excitatory neurons increased to greater than 35 Hz from lower firing rates in the middle region (from the blue area in **Figure 2E** transition to the red area on the top right), and roughly corresponds to a similar increase of maximal firing rate to above 100 Hz for inhibitory neurons in **Figure 2F**.

Subplots **Figures 2A,B,H,I** all belong to an intermediate region in the parameter space in **Figure 2B** between two curves where maximal firing rates for both excitatory and inhibitory neurons are relatively low. Within this parameter range, excitatory and inhibitory neurons are either quasi-synchronized (**Figure 2A**) or precisely synchronized and firing rhythmically (**Figure 2H**), or exhibit moving bump activity (**Figure 2I**) or as combinations of rhythmic and moving bump activity. In all these cases, single excitatory neuron cannot maintain a stable high gamma frequency spiking activity unless connection weights are changed, moving to the WTA region on the top-right of the second curve in **Figure 2D**. **Figure 2D** thus provides a phase diagram for the neuronal network defined in **Figure 1**.

Notice that this maximal firing rate is not the neuron's instantaneous firing rate, but is the total number of spikes within a 1 s window. This definition is useful to discriminate a stable high firing rate neuron vs. a neuron firing a short burst less than 1 s and then becoming quiet (especially for stable vs. traveling activity, see **Figure 2C** vs. **Figure 2I**).

Figure 3 summarizes patterns of spike dynamics with different connection topologies. Compared to the surround inhibition type analyzed above, all the other connection types do not support a Winner-Take-All state manifested as stable bump activity shown in **Figure 2C**. This is because under those connection types, excitatory and inhibitory neurons cannot maintain high maximal firing rates when both excitatory and inhibitory weights are high and did not have a red area on the upper-right region shown in **Figures 2E,F**. The most common firing patterns for those connection types are quasi-rhythmic firings in the 10–20 Hz range for excitatory neurons resembling an epileptic state while some short burst of unstable bump activity in inhibitory neurons. Our results suggest that, among different types of connectivity topologies we analyzed, only

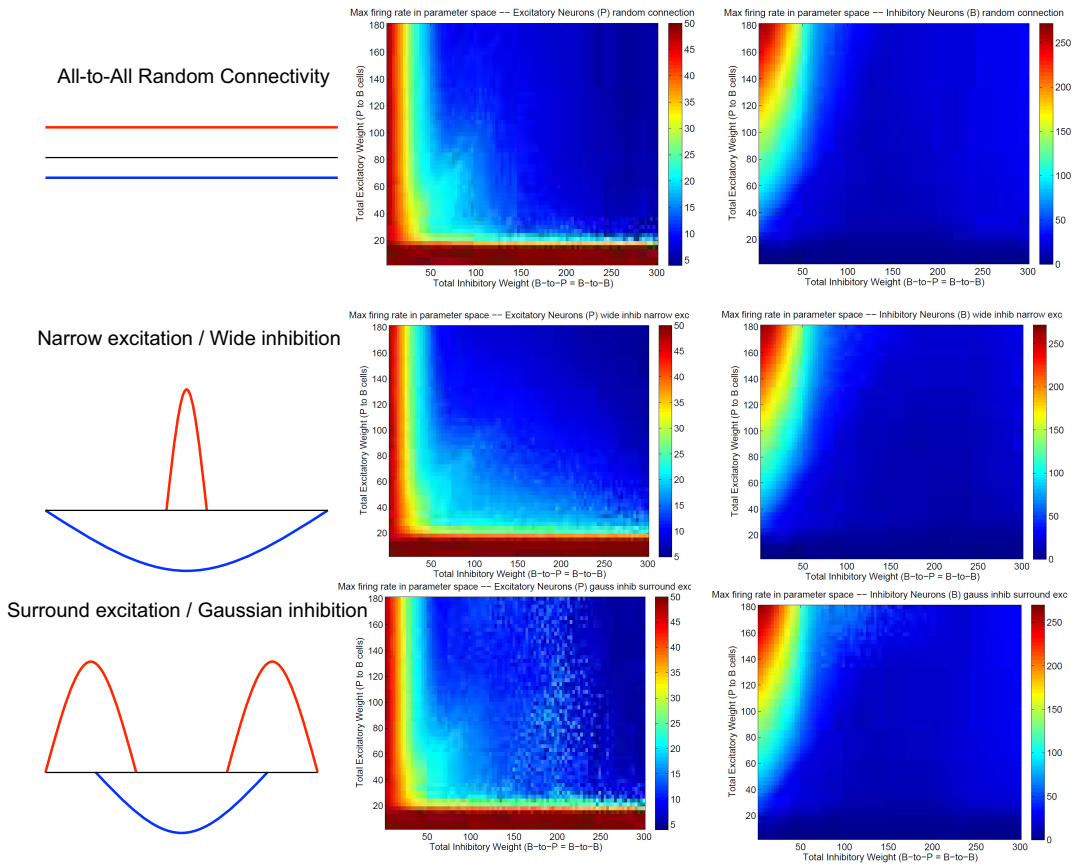


FIGURE 3 | Spiking dynamics under different connection types other than the surround inhibition type defined in Figure 1. Each row represents a connectivity type and the middle and right columns are maximal firing rate of excitatory and inhibitory neurons under specific connectivity. These plots were calculated the same way as **Figures 2E,F**. Notice that all three connectivity types here do not support a winner-take-all (WTA) region in the parameter space (no red region in the upper-right corner). It exists in **Figures 2E,F** as a red region representing a high individual maximal firing rate state when both excitatory and inhibitory weights are relatively high, but it is always absent here on the top right of the 2-d parameter space.

surround inhibition can generate a stable bump spiking activity and maintain a WTA state.

3.1. Mechanism of Winner-Take-All Neuronal Group Selection and Emergence of Bump Activity

Our above analysis suggests that surround inhibitory topology supports emergence of bump activity. To explore the mechanism of WTA and which neuronal properties are essential for such behavior, we applied the same analysis as in **Figure 2** to neuronal network in **Figure 1** when Short-Term-Synaptic-Plasticity (STSP) or NMDA voltage-gating is on, or change excitatory and inhibitory neurons' parameters to different type. In all cases, a similar WTA region was found for every conditions, even though the transition curves that delineate the emergence of stable bump activity are shifted to different positions in the parameter space (see results in Chen et al., 2013). We also analyzed the same neuronal network with a different set of individual spiking models, i.e., the adaptive exponential models and found the similar WTA region as long as the topology of the inhibitory connections are surround type. These analyses suggest that detailed neuronal properties such as exact models of the spiking neuron, STSP or NMDA gating etc., are likely not fundamental for the existence of stable bump activity, but the type of connectivity topology (i.e., surround inhibition) is more important for such behavior.

Both the Izhikevich neural model and the adaptive exponential model we used are conductance based with models of inhibitory and excitatory currents of different time scales. So we suspect that different time constants of NMDA, AMPA, GABAa, and GABAb channel conductance might play some role for the emergence of bump activity. To demonstrate this, **Figure 4** shows the time evolutions of AMPA, NMDA, and GABAa currents along with the spiking activity in the simplified network in **Figure 1** starting from a zero conductance initial condition. It demonstrates the detailed transition from a rhythmic synchronized firing state into a stable bump activity. Looking at detailed dynamic changes of the individual excitatory and inhibitory currents should shed light on how the transition is occurred.

From **Figure 4** we can see, differences in time constants will determine how fast a specific channel conductance returns back to zero after a burst of spiking activity. For excitatory neurons specifically, because of its short time constants (6 ms), AMPA current fluctuates around a similar level with large variances. NMDA currents, on the other hand, are accumulating to higher levels because of longer time constant (150 ms) even though both currents are generated by the same spiking input from the input neurons. Similar phenomenon can be seen for inhibitory neurons. When those neurons fire rhythmically before about 300 millisecond, AMPA conductance jumps to high level (from 7 to 9 nS) after each spike then drops down to zero very fast (red curve in **Figure 4D**), while NMDA conductance only drops a small amount each cycle and overall level still increases to much higher value (red curve in **Figure 4E**). Initially inhibitory neurons fire after excitatory neurons in each rhythmic cycle and they synchronize to each other with a time delay. If excitatory to inhibitory weights (E to I) are larger than a certain value, such that NMDA currents for inhibitory neurons increase faster than excitatory neurons (**Figure 4E**), the delay between inhibitory and excitatory neurons diminishes and GABAa currents become effective within the same cycle to inhibit other neurons. As a result some inhibitory and excitatory neurons stop firing in the rhythmic cycle, eventually lead to a winner group that persistently fires and shuts off their surrounding neighbors. Notice that in the simplified model in **Figure 4**, GABAb (with longer time constant of 150 ms) currents are omitted and set to zero, which lead to a moving bump activity for this specific parameter set. If GABAb conductance is restored to the original level as in the full model, bump activity becomes stable. It implies that time constant of GABAa and GABAb channel conductance is related to the stability of the bump activity.

As a summary, we think the combinations of long and short time constants from excitatory and inhibitory conductance plus the surround inhibitory connectivity support a mechanism for emergence of bump activity and winner-take-all phenomenon in this basic spiking neuronal network. This neuronal group selection mechanism provides a basis for modeling learning and map-formation process for sensory motor integration and other higher cognitive processes.

4. ANALYTICAL ANALYSIS OF THE TRANSITION CURVE FOR WTA PHENOMENON

4.1. Differentiation in Inhibitory Conductances Lead to Spiking Activity Pattern Transition and Neuronal Group Selection

To identify the mechanism of bump activity in spiking networks, based upon the transition plots shown in **Figure 4**, we look at voltages, conductances of all 400 excitatory neurons at one specific time point ($t = 980$ ms. in **Figure 4**). **Figure 5** shows AMPA, NMDA, and GABAa conductances along with voltages for all those excitatory neurons at this specific time point. From the network structure defined in **Figure 1**, we know that the excitatory conductances (AMPA and NMDA) are determined by excitatory synapses from the input layer (because we omitted self-excitation), where inhibitory conductances (GABAa and GABAb) are only determined (triggered) by spikes from inhibitory neurons onto those excitatory neurons. Because of the uniform random connection from input layer, AMPA and NMDA conductances are around the same level and undifferentiated for all excitatory neurons. From **Figure 5**, voltages are above threshold and neurons fire only at locations where GABAa conductances below a certain value. So in order for a bump to emerge and a subgroup of neurons selected to be active, GABAa or GABAb conductances have to be differentiated, i.e., they have to be small for some neurons and remain high for all other neurons. We suspect this condition can only be met by surround type of inhibitory connectivity. It is obvious that lowest GABA conductances lead to highest firing rate for excitatory

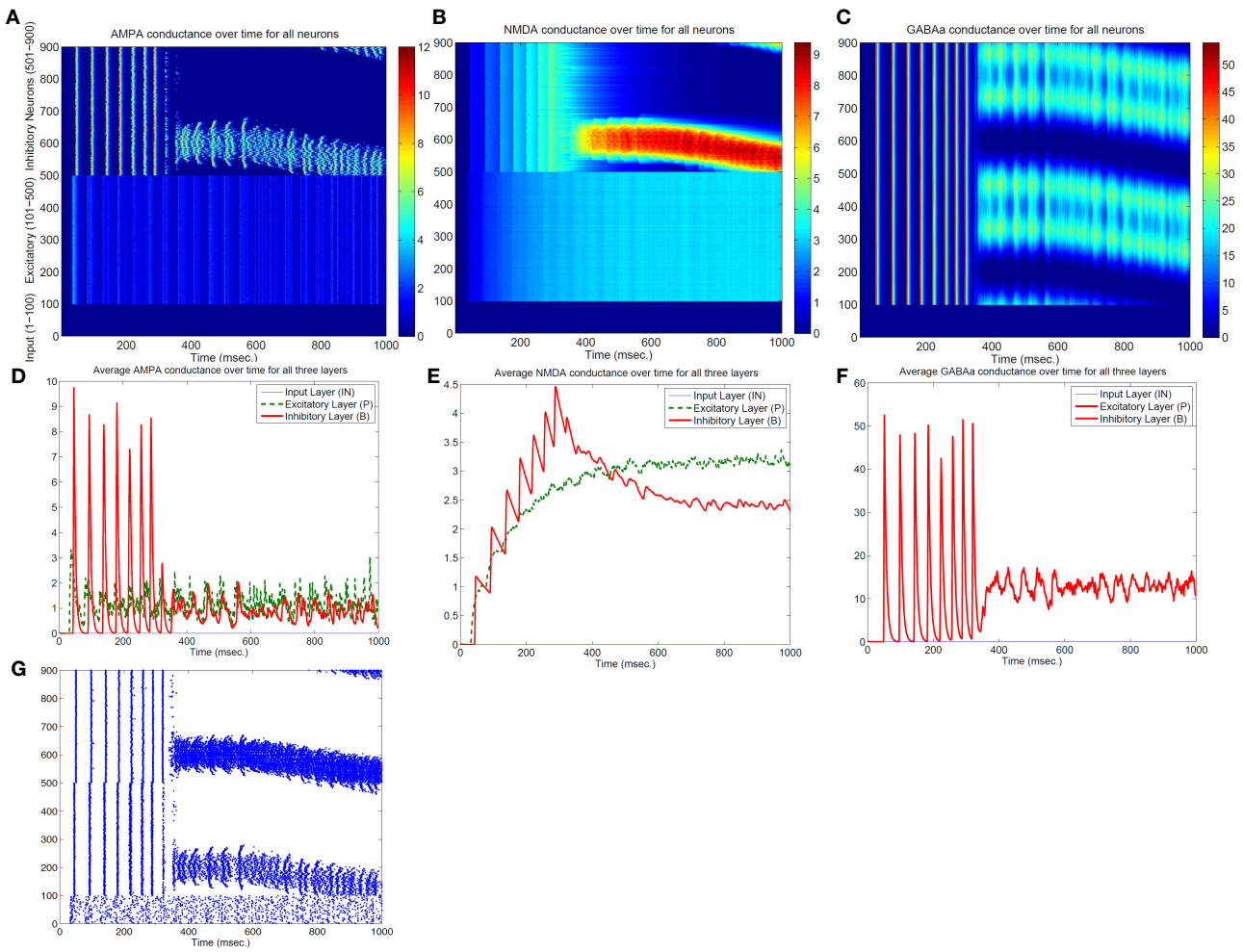


FIGURE 4 | Transition from a synchronized rhythmic firing state to a winner-take-all bump spiking activity–(G). In this simplified surround inhibition network, short-term synaptic plasticity (STSP) and NMDA gating effects are removed, GABAb conductance is set to be zero and individual excitatory and inhibitory spiking models are changed to have the same biological parameter set. First row shows the 1-s spatial-temporal evolutions of individual conductance for AMPA–**(A)**, NMDA–**(B)**, and GABAa–**(C)**. Second row **(D–F)** show the averaged conductance over time for excitatory and inhibitory cells separately. Notice that bump activity in AMPA and NMDA conductance appear in inhibitory neurons first (see **A,B**), while both conductances are spatially uniform even after spiking bump activity emerged after about 400 ms. The fact that spatial uniformity is destroyed in GABAa conductance first in **(C)** suggests that inhibitory neurons might show transition into winner-take-all state early and then bias the transition in excitatory cells.

neurons. Since we have local feedforward excitation to inhibitory neurons in our network, the bump area in inhibitory neurons with highest firing rate should also have lowest inhibitory conductances. This difference in GABA conductances is true for both excitatory and inhibitory neurons because GABA conductances are determined by the same inhibitory spikes. This suggests that in order for a bump to emerge, local inhibition to the nearest neighbors should be lower than inhibition to neurons outside of the bump. Notice the three other inhibition topology in **Figure 3** all have peak (or flat) inhibition locally, so even if a neuronal group emerge with highest firing rate, the strong local inhibition will force their firing rates to decrease, and let the other sub-threshold neurons to fire. So this is likely the reason why we did not obtain stable bump activity using those inhibition connectivities. On the other hand, surround inhibition type defined in **Figure 1** might be the most simple form of inhibition topology that could let a bump emerge and stabilize. Below analysis will further prove this point.

In contrast with models using negative weights to represent inhibitory connections, our spiking models' synaptic weights and excitatory/inhibitory conductances are all positive (which obviously is more biologically realistic). From Equation (2) we can see, it is only because of the differences in reversal potentials between excitatory and inhibitory channels, the current generated by excitatory and inhibitory conductances could have different signs (excitatory current coming into the neuron and inhibitory current coming out of neuron). In order for a neuron to fire, synaptic current has to be below a threshold $I_{syn} < -I_{th}$ where I_{th} is about 100pA (I_{syn} has to be negative for a neuron to fire because it was defined as an outward

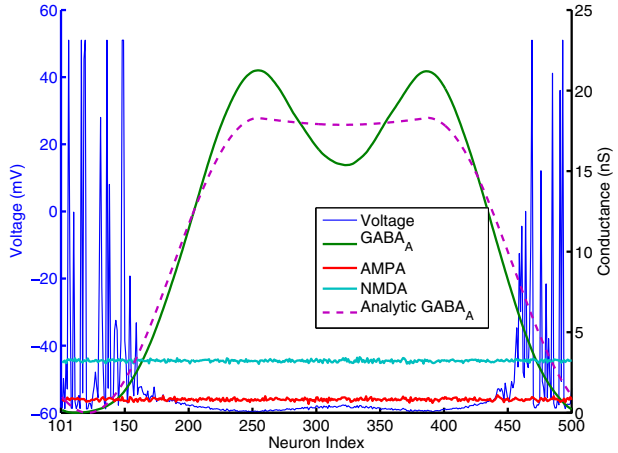


FIGURE 5 | *AMPA*, *NMDA*, $GABA_A$ conductances and voltages for all excitatory neurons (cell index 101–500 in Figure 1) at one specific time point (t = 980 ms. in Figure 4). AMPA and NMDA conductances are basically flat across all excitatory neurons here, while $GABA_A$ conductances are close to 0 for neurons around number 101 and 500, and reach high values else where. Pink dashed lines are analytic GABA conductances from Equation (A.6) in the Appendix and derived from surround inhibition topology, and is a good match for the actual numerical simulation. Notice that excitatory neurons fire spikes and have above threshold voltage values (blue lines) only within neighborhood of neurons having $GABA_A$ conductances close to 0 and below a certain value.

current in Equation 1a). Firing threshold I_{th} can be found from calculating F-I curve for the specific spiking neuron model we used. Figure A.3 (in the Appendix) plots the firing rates vs. amount of injected current (equivalent to $-I_{syn}$) for the Izhikevich neuron model result from numerical simulations. It shows that neurons start to fire when absolute value of the injected current is above 100 pA and then increase their firing rate approximately linearly until above 100 Hz. we can use this information to simplify the spiking activity into a rate model. As indicated on the last paragraph, AMPA and NMDA conductances are approximately uniform for excitatory neurons and they can not contribute to the differences in firing rates, so in order for the excitatory population to fire differentially, the difference between highest and lowest GABA conductances for individual neurons has to be larger than a certain value. This value can be estimated using Equation (2). If $min(g_{GABA_A})$ is 0, for a resting potential of $v_r = -60mV$, $GABA_A$ conductance has to be larger than the following value so injected synaptic current $-I_{syn}$ will be below the firing threshold I_{th}:

$$g_{GABA_A} > (-I_{th} + 60 * (g_{AMPA,E} + g_{NMDA,E}))/10. \quad (4)$$

In **Figure 5**, $g_{AMPA,E} + g_{NMDA,E}$ for excitatory neurons is around 4nS (Appendix will show how this value can be estimated analytically), so g_{GABA_A} has to be larger than 14 nS to keep sub-threshold neurons from firing. This number is consistent with the result plotted on **Figure 4**, **5** that neurons fire and form a bump area where GABA conductances are below 14 nS and areas with GABA conductance larger than 14 nS are completely quiet.

If we consider both the $GABA_A$ and $GABA_B$ conductances based upon Equation (2) and using the same idea as above, conditions for inhibitory conductances will be the following for the winner-take-all state:

$$10 \cdot g_{GABA_A} + 30 \cdot g_{GABA_B} > (-I_{th} + 60 * (g_{AMPA,E} + g_{NMDA,E})). \quad (5)$$

Equations (4, 5) can be used further to identify the exact condition for the WTA state and to locate the transition curve in **Figure 2**. Using two cosine bumps as surround inhibitory connection weights, Appendix gives the analytic form of GABA distribution of a bump solution for neurons connected one dimensionally and uses it to obtain analytic conditions for the WTA state in the parameter space (see Equations A.13, A.14). Such analytic conditions are expressed as formulas combining single neuron property and the conductance parameters (such as time constants, gain factors for different inhibitory and excitatory conductances). Based upon these formulas we can locate the Winner-Take-All and bump activity in the parameter space fairly precisely (see black curves in **Figure 2** and the white curve in Figure A.4 in the Appendix), thus provide a mechanistic explanation for the emergence of winner-take-all state and stationary bump activity in this 3-layer spiking network we analyzed here.

4.2. Origin of Traveling Wave and Instability of Bump Activity – Driven by AMPA Conductances

Parameters in **Figure 4** are located very near the transition curve in the parameter space (see **Figure 2**), so the bump is not spatially stable and moves across different neurons. To identify the origin of such instability, we selectively set AMPA gain of excitatory or inhibitory neurons to 0 in order to see their effects on the bump stability. This is equivalent to selectively block AMPA conductances in either excitatory or inhibitory neurons in real biological neural systems. We found that if AMPA conductances in inhibitory neurons are set to 0 but not in excitatory neurons, bumps become more or less stable. On the other hand, when AMPA conductances are blocked and set to 0 in excitatory neurons but not in inhibitory neurons, we can have a moving bump with a constant spatial speed (see **Figure 6**). In fact, we can estimate the moving speed of bumps based upon the parameters we defined in **Figure 2**. So we believe the source of the bump instability is from the AMPA conductances in inhibitory neurons. Previous studies have associated stabilities of bump activity with dynamic synapses (Fung et al., 2009). Notice that AMPA conductances have much shorter time constants than others thus more associated with faster synapses (similar to GABAa), so in this sense there might be a connection and some agreement between our observation on **Figure 6** and dynamic synapses analyzed by Fung et al. (2009).

5. CONCLUSION AND DISCUSSION

In this paper we derive global properties of spiking neuronal networks related to bump activity and Winner-Take-All state mainly through analysis of the dynamics of excitatory and

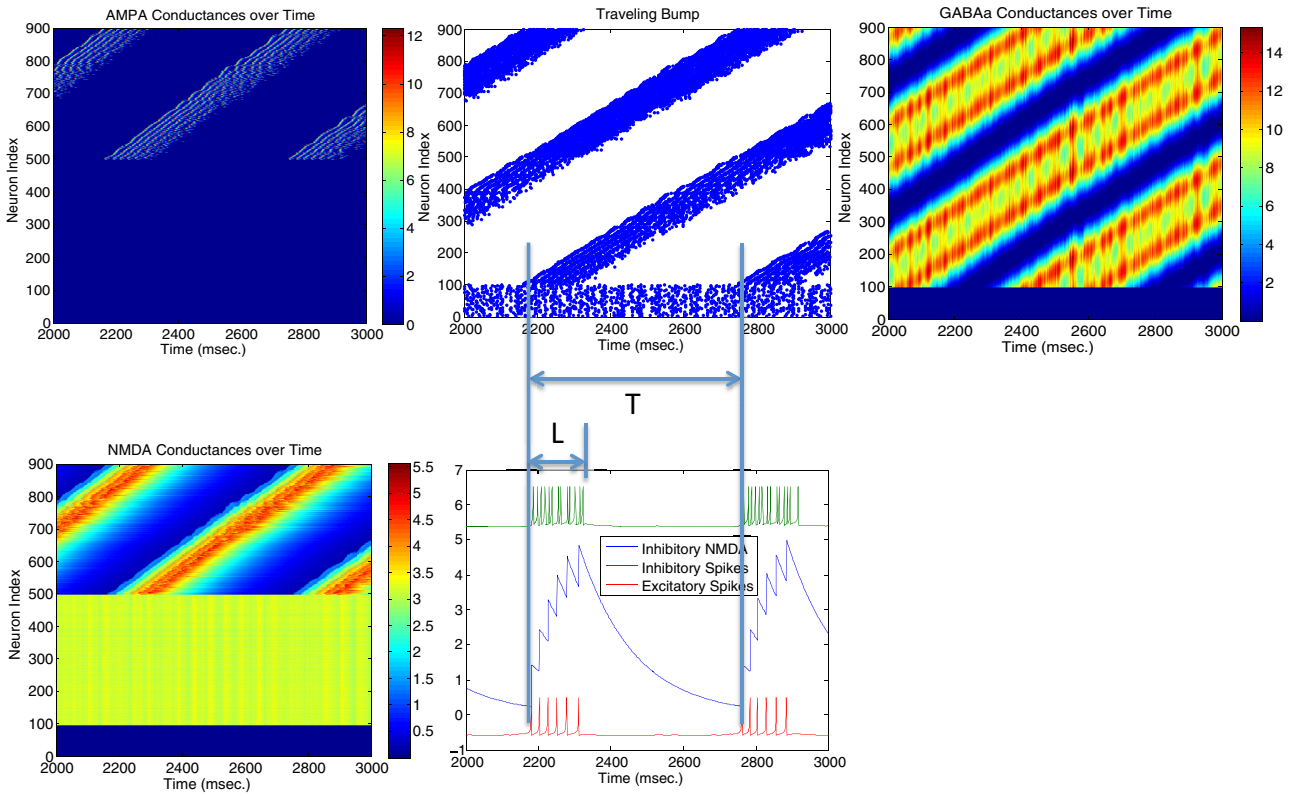


FIGURE 6 | Stability of the bump activity is determined by AMPA conductances. Figures above show that spiking bump activity moves with a constant speed when AMPA conductances in excitatory neurons are blocked and set to 0 (Top left). The speed of the traveling "bump," proportional to the length of firing interval for each neuron (L) and the period of the traveling wave (T), seems relate to the frequency of the inhibitory neurons (hence their AMPA conductances) and the delay time for the NMDA conductances when they reach maximal firing rate. When AMPA conductances are set to 0 in inhibitory neurons, bump activity stops moving and $T \to \infty$.

inhibitory conductances. To achieve the analysis of collective behavior, individual spiking properties are approximated by its firing rate property such as the conductance/firing rate curve (Figure A.3 in the Appendix) or F-I curves. In this regard, detailed properties of individual spiking model might not be crucial for global activity such as the emergence of bump and WTA state. For example, if we use different parameter sets for excitatory and inhibitory neurons such as changing the inhibitory neurons to be basket cell type, we can still found the WTA region in the parameter space in **Figure 2D**, but the exact location of the transition curve is shifted to a different place because basket neurons have different conductance/firing rate curve and different I_{th}, g_{th}, k values in Equation (A.13). This could explain the transition curve in our full model with more detailed biological properties has the same functional form, but in different location in the parameter space (see Chen et al., 2013) because it included more detailed single neuronal spiking properties such as NMDA voltage gating and STSP etc. In fact, we used adaptive exponential spiking model to substitute the Izhikevich neuron models and obtained similar phase plot and transition curve for the bump activity and the WTA state.

We suggest that all conductance-based spiking models with distinct excitatory and inhibitory populations could have the similar collective Winner-Take-All behavior as analyzed here. Detailed spiking model properties such as F-I curve and firing threshold (I_{th} and g_{th}) would determine the exact location of the transition curve in **Figure 2**. Global connectivity topology and different time constants (dynamics) of excitatory and inhibitory conductances are likely to be the determinant of system-wide spiking activity patterns.

5.1. Importance of the Inhibitory Topology

The most important feature of our winner-take-all network is its surround inhibition topology. The reason we chose two sine function peaks as surround inhibitory connection weights is just because of its mathematical convenience, since convolutions of sine/cosine functions are much easier to solve than other types of functions. In fact, connection topologies using Gaussian peaks or torus (for two-dimensional neuronal arrays) were used in our previous model (Chen et al., 2013) and similar stable WTA results were obtained. We believe using other type of function for inhibitory connection would also work, as long as there is a low inhibitory weight locally. Comparing four different connection topologies from **Figures 2**, **3**, the reason why only surround-inhibition supports stable bump activities is because its maximal inhibitory connection weight is not to the nearest neighbors,

but to slightly distant neurons. This gap of zero inhibitory weight can be very small (e.g., w down to 0 and equivalent to a no-self-inhibition case), and we can still find solutions for stable bump or bumps (in fact, w determines how many bumps can emerge and we will have a 2-bump solution when w is close to 0, see Appendix and Figures A.1, A.2). So as long as there is a local valley of inhibitory weight, stationary bumps could emerge because only decreasing inhibition could allow a bump to sustain.

Mechanistically it appears that the most important requirement for a bump solution is the stable differentiation in inhibitory ($GABA_A$ or $GABA_B$) conductance distributions across the neuronal population. That is, for some neuronal groups, GABA conductances should be low to allow bumps to emerge and for the other neurons, they need to be high enough to keep the rest of neuronal population from firing spikes. As long as this condition is met, more detailed biological properties such as local self-excitation, short-term synaptic plasticity (STSP), voltage-gating of NMDA channels etc. can be added to the model without destroying the overall bump stability.

5.2. Why Traditional Center-Surround Topology Might Not Lead to Stable Bumps in Models with Distinct Excitatory and Inhibitory Populations

Previous rate-based population models (Dayan and Abbot, 2001) had most often used center-surround type of connection topology as shown in the middle row of **Figure 3** (Narrow excitation/Wide inhibition). Similarly, many spiking models with excitatory/inhibitory conductances on the same units used the same topology (Laing and Chow, 2001). By simple subtraction, narrow excitation and wide inhibition can lead to a "Mexican Hat" type of effective connectivity which supports winner-take-all in previous firing rate models. But, as we see from the analysis above, in biologically more realistic spiking models with distinct excitatory and inhibitory neuronal populations, multiple types of conductances cannot cancel each other easily because they are generated by precise spike timing and have different time constants. The "classical" center-surround topology can not guarantee a stable "Mexican Hat" type of net connectivity because sensitive spike timing differences between different neurons prevent easy subtraction of excitatory and inhibitory weights at every time point. In fact, as shown in **Figure 4**, the emergence of winner-take-all in spiking networks is a direct result of precise spike-timing–the coincide of excitatory and inhibitory population firing spikes lead to a sub-population of inhibitory neurons fire earlier than the rest of populations which then let them suppress and shut off the other neurons in the network (see **Figure 4G**). This is the reason that we believe a surround-type of inhibitory topology is essential for a stable spiking WTA network because it can support the emergence of a winner-group without shutting off themselves too early.

In summary, WTA network analyzed here demonstrates how variability and randomness in spiking time of individual neurons can lead to global pattern changes and phase transition in collective neuronal groups. Analytic solutions for the phase transition curve provided in this paper will help to increase our understandings of different functional roles of excitatory and inhibitory neural connections on the emergence and stability of firing patterns in the brain.

AUTHOR CONTRIBUTIONS

YC developed the original idea of this paper, performed computational modeling and mathematical derivation for the analytical solution and wrote the paper.

ACKNOWLEDGMENTS

This work was supported in part by DARPA through ONR Grant N00014-08-1-0728 and by AFRL Cooperative Agreement FA8750-11-2-0255 to Neurosciences Research Foundation and by a grant from The G. Harold & Leila Y. Mathers Charitable Foundation. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views, opinions, and/or findings contained in this article are those of the authors and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency, the Air Force Research Laboratory, the Department of Defense, or the U.S. Government.

APPENDIX

The Appendix for this article can be found online at: http://journal.frontiersin.org/article/10.3389/fncom.2017.00020/full#supplementary-material

REFERENCES

Boyland, P. L. (1986). Bifurcations of circle maps: Arnol'd tongues, bistability and rotation intervals. *Comm. Math. Phys.* 106, 353–381.

Chen, Y., McKinstry, J. L., and Edelman, G. M. (2013). Versatile networks of simulated spiking neurons displaying winner-take-all behavior. *Front. Comput. Neurosci.* 7:16. doi: 10.3389/fncom.2013.00016

Coombes, S., and Bressloff, P. C. (1999). Mode locking and Arnold tongues in integrate-and-fire neural oscillators. *Phys. Rev. E.* 60(2 Pt. B), 2086–2096.

Dayan, P., and Abbot, L. F. (2001). *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. Cambridge, MA: MIT Press.

Edelman, G. M. (1987). *Neural Darwinism: The Theory of Neuronal Group Selection*. New York, NY: Basic Books.

Ermentrout, G. B., and Cowan, J. D. (1979). A mathematical theory of visual hallucination patterns. *Biol. Cybern.* 34, 137–150.

Fung, C. C. A., Wong, K. Y. M., and Wu, S. (2009). Tracking dynamics of two-dimensional continuous attractor neural networks. *J. Phys. Conf. Ser* 197:012017. doi: 10.1088/1742-6596/197/1/012017

Furman, M., and Wang, X.-J. (2008). Similarity effect and optimal control of multiple-choice decision making. *Neuron* 60, 1153–1168. doi: 10.1016/j.neuron.2008.12.003

Goodhill, G. J. (2007). Contributions of theoretical modeling to the understanding of neural map development. *Neuron* 56, 301–311. doi: 10.1016/j.neuron.2007.09.027

Itti, L., and Koch, C. (2001). Computational modeling of visual attention. *Nat. Rev. Neurosci.* 2, 194–203. doi: 10.1038/35058500

Izhikevich, E. M., and Edelman, G. M. (2008). Large-scale model of mammalian thalamocortical systems. *Proc. Natl. Acad. Sci. U.S.A.* 105, 3593–3598. doi: 10.1073/pnas.0712231105

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biol. Cybern.* 43, 59–69. doi: 10.1007/BF00337288

Laing, C. R., and Chow, C. C. (2001). Stationary bumps in networks of spiking neurons. *Neural Comput.* 13, 1473–1494. doi: 10.1162/089976601750264974

Maass W. (2000). On the computational power of winner-take-all. *Neural Comput.* 12, 2519–2535. doi: 10.1162/089976600300014827

McKinstry, J., Fleischer, J. G., Chen, Y., Gall, W. E., and Edelman, G. M. (2016). Imagery may arise from associations formed through sensory experience: a network of spiking neurons controlling a robot learns visual sequences in order to perform a mental rotation task. *PLoS ONE* 11:e0162155. doi: 10.1371/journal.pone.0162155

Oster, M., Douglas, R., and Liu, S.-C. (2009). Computation with spikes in a winner-take-all network. *Neural Comput.* 21, 2437–2465. doi: 10.1162/neco.2009.07-08-829

Rutishauser, U., and Douglas, R. J. (2009). State-dependent computation using coupled recurrent networks. *Neural. Comput.* 21, 478–509. doi: 10.1162/neco.2008.03-08-734

Rutishauser, U., Douglas, R. J., and Slotine, J. (2011). Collective stability of networks of winner-take-all circuits. *Neural Comput.* 23, 735–773. doi: 10.1162/NECO_a_00091

Shriki, O., Hansel, D., and Sompolinsky, H. (2003). Rate models for conductance-based cortical neuronal networks. *Neural Comput.* 15, 1809–1841. doi: 10.1162/08997660360675053

Somers, D. C., Nelson, S. B., and Sur, M. (1995). An emergent model of orientation selectivity in cat visual cortical simple cells. *J. Neurosci.* 15, 5448–5465.

Walther, D., and Koch, C. (2001). Modeling attention to salient proto-objects. *Neural Netw.* 19, 1395–1407. doi: 10.1016/j.neunet.2006.10.001

Wang, X.-J. (2002). Probabilistic decision making by slow reverberation in cortical circuits. *Neuron* 36, 955–968. doi: 10.1016/S0896-6273(02)01092-9

Wei, Z., Wang, X.-J., and Wang, D. H. (2012). From distributed resources to limited slots in multiple-item working memory: a spiking network model with normalization. *Neuron* 32, 11228–11240. doi: 10.1523/jneurosci.0735-12.2012

Wilson, H. R., and Cowan, J. D. (1973). A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue. *Kybernetik* 13, 55–80.

Conflict of Interest Statement: The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Computational Neuroscience

ORIGINAL RESEARCH
published: 21 March 2017
doi: 10.3389/fncom.2017.00013

The Role of Architectural and Learning Constraints in Neural Network Models: A Case Study on Visual Space Coding

Alberto Testolin[1]*, *Michele De Filippo De Grazia*[1] and *Marco Zorzi*[1, 2]*

[1] Department of General Psychology and Padova Neuroscience Center, University of Padova, Padova, Italy, [2] San Camillo Hospital IRCCS, Venice, Italy

The recent "deep learning revolution" in artificial neural networks had strong impact and widespread deployment for engineering applications, but the use of deep learning for neurocomputational modeling has been so far limited. In this article we argue that unsupervised deep learning represents an important step forward for improving neurocomputational models of perception and cognition, because it emphasizes the role of generative learning as opposed to discriminative (supervised) learning. As a case study, we present a series of simulations investigating the emergence of neural coding of visual space for sensorimotor transformations. We compare different network architectures commonly used as building blocks for unsupervised deep learning by systematically testing the type of receptive fields and gain modulation developed by the hidden neurons. In particular, we compare Restricted Boltzmann Machines (RBMs), which are stochastic, generative networks with bidirectional connections trained using contrastive divergence, with autoencoders, which are deterministic networks trained using error backpropagation. For both learning architectures we also explore the role of sparse coding, which has been identified as a fundamental principle of neural computation. The unsupervised models are then compared with supervised, feed-forward networks that learn an explicit mapping between different spatial reference frames. Our simulations show that both architectural and learning constraints strongly influenced the emergent coding of visual space in terms of distribution of tuning functions at the level of single neurons. Unsupervised models, and particularly RBMs, were found to more closely adhere to neurophysiological data from single-cell recordings in the primate parietal cortex. These results provide new insights into how basic properties of artificial neural networks might be relevant for modeling neural information processing in biological systems.

Keywords: connectionist modeling, unsupervised deep learning, restricted Boltzmann machines, autoencoders, sparseness, space coding, gain modulation, sensorimotor transformations

Edited by:
Marcel van Gerven, Radboud University Nijmegen, Netherlands

Reviewed by:
Michael W. Spratling, King's College London, UK
Kandan Ramakrishnan, University of Amsterdam, Netherlands

***Correspondence:**
Alberto Testolin
alberto.testolin@unipd.it
Marco Zorzi
marco.zorzi@unipd.it

Received: *30 November 2016*
Accepted: *27 February 2017*
Published: *21 March 2017*

Citation:
Testolin A, De Filippo De Grazia M and Zorzi M (2017) The Role of Architectural and Learning Constraints in Neural Network Models: A Case Study on Visual Space Coding. Front. Comput. Neurosci. 11:13. doi: 10.3389/fncom.2017.00013

INTRODUCTION

Artificial neural network models aim at explaining human cognition and behavior in terms of the emergent consequences of a large number of simple, subcognitive processes (McClelland et al., 2010). Within this framework, the pattern seen in overt behavior (macroscopic dynamics of the system) reflects the coordinated operations of simple biophysical mechanisms (microscopic dynamics of the system), such as the propagation of activation and inhibition among elementary processing units. Though this general tenet is shared by all connectionist models, there is large variability in processing architectures and learning algorithms, which turns into varying degrees of psychological and biological realism (e.g., Thorpe and Imbert, 1989; O'Reilly, 1998). When the aim is to investigate high-level cognitive functions, simplification is essential (McClelland, 2009) and the underlying processing mechanisms do not need to faithfully implement the neuronal circuits supposed to carry out such functions in the brain. However, modelers should strive to consider biological plausibility if this can bridge different levels of description (Testolin and Zorzi, 2016).

Recent theoretical and technical progress in artificial neural networks has significantly expanded the range of tasks that can be solved by machine intelligence. In particular, the advent of powerful parallel computing architectures based on Graphic Processing Units (GPUs), coupled with the availability of "big data," has allowed to create and train large-scale, hierarchical neural networks known as *deep neural networks* (LeCun et al., 2015, for review). These powerful learning systems achieve impressive performance in many challenging cognitive tasks, such as visual object recognition (Krizhevsky et al., 2012), speech processing (Mohamed et al., 2012) and natural language understanding (Collobert et al., 2011). However, while the impact of deep learning for engineering applications is undisputed, its relevance for modeling neural information processing in biological systems still needs to be fully evaluated (for seminal attempts, see Stoianov and Zorzi, 2012; Khaligh-Razavi and Kriegeskorte, 2014; Güçlü and van Gerven, 2015).

One critical aspect of most deep learning systems is the reliance on a feed-forward architecture trained with error backpropagation (Rumelhart et al., 1986), which has been repeatedly shown to yield state-of-the-art performance in a variety of problems (LeCun et al., 2015). However, the assumptions that learning is largely discriminative (e.g., classification or function learning) and that an external teaching signal is always available at each learning event (i.e., all training data is "labeled") are clearly implausible from both a cognitive and a biological perspective (Zorzi et al., 2013; Cox and Dean, 2014). Reinforcement learning is a valuable alternative and it has already shown promising results when combined with deep learning (Mnih et al., 2015; Silver et al., 2016), but there is a broad range of situations where learning seems to be fully unsupervised and its only objective is that of discovering the latent structure of the input data in order to build rich, internal representations of the environment (Hinton and Sejnowski, 1999). We argue that more realistic neurocognitive models should therefore also exploit unsupervised forms of deep learning, where the objective is not to explicitly classify the input patterns but rather to discover internal representations by fitting a hierarchical generative model to the sensory data (Hinton, 2007, 2013; Zorzi et al., 2013). Compared to its supervised counterpart, this modeling approach emphasizes the role of feedback, recurrent connections (Sillito et al., 2006), which carry top-down expectations that are gradually adjusted to better reflect the observed data (Hinton and Ghahramani, 1997; Friston, 2010) and which can be used to implement concurrent probabilistic inference along the whole cortical hierarchy (Lee and Mumford, 2003; Gilbert and Sigman, 2007). Notably, top-down processing is also relevant for understanding attentional mechanisms in terms of modulation of neural information processing (Kastner and Ungerleider, 2000).

A powerful class of stochastic neural networks that learn a generative model of the data is that of Restricted Boltzmann Machines (RBMs), which can efficiently discover internal representations (i.e., latent features) using Hebbian-like learning mechanisms (Hinton, 2002). RBMs constitute the building block of hierarchical generative models such as Deep Belief Networks (Hinton and Salakhutdinov, 2006) and Deep Boltzmann Machines (Salakhutdinov, 2015). These unsupervised deep learning models have been successfully used to simulate a variety of cognitive functions, such as numerosity perception (Stoianov and Zorzi, 2012), letter perception (Testolin et al., under review), location-invariant visual word recognition (Di Bono and Zorzi, 2013), and visual hallucinations in psychiatric syndromes (Reichert et al., 2013). A similar approach has been used to simulate how early visual cortical representations are adapted to statistical regularities in natural images, in order to predict single voxel responses to natural images and identify images from stimulus-evoked multiple voxel responses (Güçlü and van Gerven, 2014). A temporal extension of RBMs has also been recently used to model sequential orthographic processing and spontaneous pseudoword generation (Testolin et al., 2016).

Unsupervised deep learning can be implemented using an alternative architecture based on autoencoders (Bengio et al., 2007), which are deterministic, feed-forward networks whose learning goal is to accurately reconstruct the input data into a separate layer of output units. Single-layer autoencoders are trained using error backpropagation, and can be stacked in order to build more complex, multi-layer architectures. However, despite the common view that RBMs and autoencoders could be considered equivalent (Ranzato et al., 2007), we note that their underlying architectural and learning assumptions are significantly different. In this study we empirically compare RBMs and autoencoders in terms of the type of internal encoding emerging in the hidden neurons. Moreover, we investigate how additional learning constraints, such as sparsity and limitation of computational resources (i.e., hidden layer size), could influence the representations developed by the networks. As a case study, we focus on the problem of learning visuospatial coding for sensorimotor transformations, which is a prominent example of how the emergentist approach based on learning in artificial neural networks has offered important insights into the computations performed by biological neurons (Zipser and Andersen, 1988).

Sensorimotor transformations refer to the process by which sensory stimuli are converted into motor commands. For example, reaching requires to map visual information, represented in retinal coordinates, into a system of coordinates that is centered on the effector. Coordinate transformations can be accomplished by combining sensory information with extra-retinal information, such as postural signals representing the position of eyes, head, or hand, thereby obtaining abstract representations of the space interposed between the sensory input and the motor output (Pouget and Snyder, 2000). Single-neuron recordings from monkey posterior parietal cortex have shown that the response amplitude of many neurons indeed depends on the position of the eyes, thereby unveiling a fundamental coding principle used to perform this type of signal integration (Andersen et al., 1985). The term *gain field* was coined to describe this gaze-dependent response of parietal neurons, and since then the notion of *gain modulation* has been generalized to indicate the multiplicative control of one neuron's responses by the responses of another set of neurons (Salinas and Thier, 2000). Another fundamental property unveiled by neuronal recordings is that the encoding of space used for coordinate transformations involves a variety of different, complementary frames of reference. For example, although many parietal neurons are centered on retinal coordinates (Andersen et al., 1985; Duhamel et al., 1992), others represent space using body-centered (Snyder et al., 1998) or effector-centered (Sakata et al., 1995) coordinate systems. Moreover, some neurons exhibit multiple gain modulation (Chang et al., 2009), suggesting more complex forms of spatial coding. For example, postural information related to both eye and head positions can be combined in order to encode "gaze" direction (Brotchie et al., 1995; Stricanne et al., 1996; Duhamel et al., 1997).

From a computational perspective, the seminal work of Zipser and Andersen (1988) showed that gain modulation could spontaneously emerge in supervised, feed-forward neural networks trained to explicitly map visual targets into head-centered coordinates, giving as input any arbitrary pair of eye and retinal positions. Similar results have been observed using more biologically-plausible learning settings, such as reinforcement learning (Mazzoni et al., 1991) and predictive coding (De Meyer and Spratling, 2011). Note that these learning settings assume that gain modulation emerges because the task implies to establish a mapping between different reference frames. However, it is unclear whether the form of modulation and the distribution of neuronal tuning functions is influenced by the type of learning algorithm and/or by the nature of the learning task (i.e., learning input-output mappings vs. unsupervised learning of internal representations). We also note that a popular alternative framework for modeling sensorimotor transformations is not based on learning, but rather stipulates that parietal neurons represent a set of basis functions that combine visual and postural information (for review, see Pouget and Snyder, 2000).

In summary, space coding represents an interesting case study for testing the adequacy of different neural network architectures and learning algorithms, because it provides a wealth of neurophysiological data (both at the population and single-neuron levels), and it departs from the classic problem of visual object recognition investigated in the large majority of deep learning research.

MATERIALS AND METHODS

In this section we describe the space coding tasks used in our simulations, including training and test stimuli, the different learning architectures, and the procedures for analyzing the emergent neural representations.

Space Coding Tasks

In this study we consider a visual signal in retinotopic coordinates and two different postural signals, one for eye position and another for a generic "effector," which might represent, for example, the position of the hand. We do not consider the integration between different modalities (see Xing and Andersen, 2000, for a computational investigation of multimodal integration in several coordinate frames). We implemented three types of space coding tasks to test the different learning architectures.

Unsupervised Learning with No Coordinate Transformation

The first learning architecture is depicted in **Figure 1A**. Unsupervised learning is represented by undirected arrows, which connect the sensory input to a separate layer of hidden neurons. The input signal to the network consists of a visual map, which represents target location in retinotopic coordinates, and two postural maps, which represent eye and effector positions. The learning goal is only to build a compact representation of these input signals in the hidden layer, which is later read-out by a simple linear associator in order to establish a mapping with the corresponding motor program. Details of input and output representations are provided in Section Dataset and Stimuli. The unsupervised learning phase does not involve any coordinate transformation because information about the motor program is not available.

Unsupervised Learning with Coordinate Transformation

The second learning architecture is depicted in **Figure 1B**. The input signal to the network still consists of a visual map and two postural maps, but in this case we also provide as input the corresponding motor program. In this setting the unsupervised learning phase implicitly involves coordinate transformation (i.e., different coordinate systems become associated). In order to compare the mapping accuracy of different learning architectures using the same method, the motor program is still read-out from hidden neurons via a simple linear associator.

Supervised Learning with Coordinate Transformation

The third learning architecture is depicted in **Figure 1C**, and it corresponds to the model used by Zipser and Andersen (1988). The input is the same of the unsupervised architecture shown in **Figure 1A**, but in this case supervised learning (directed arrows) is used to establish an explicit mapping between input signals

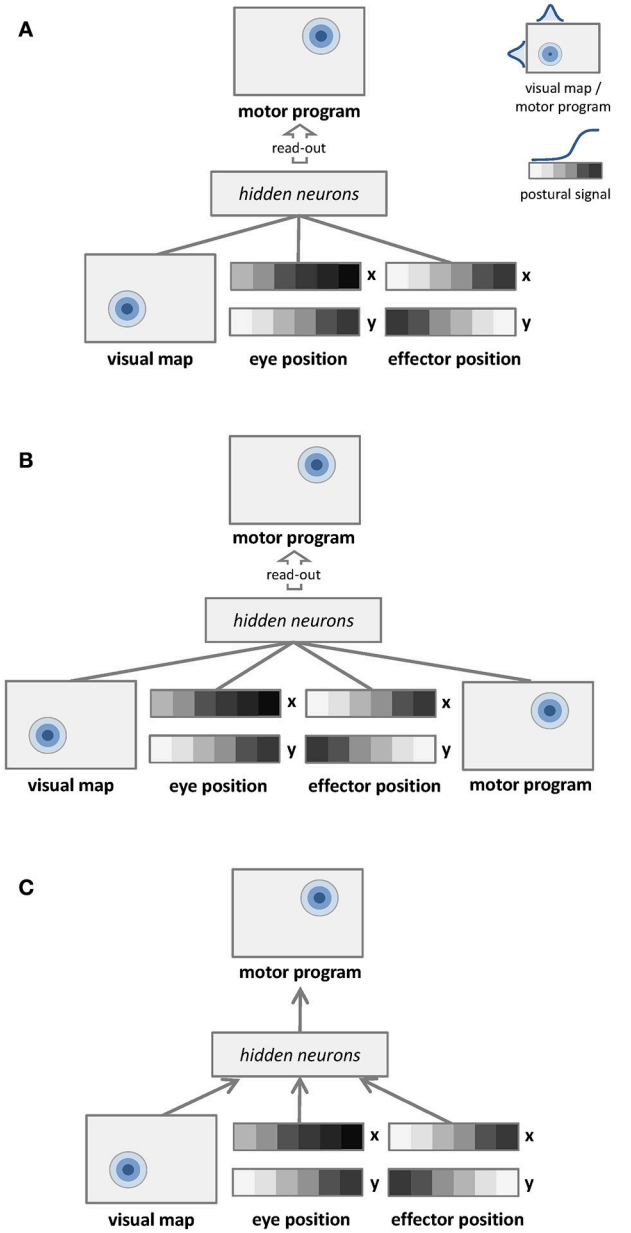


FIGURE 1 | Graphical representations of the learning architectures used to simulate the space coding tasks. Undirected edges entail bidirectional (recurrent) connections, while directed arrows represent feed-forward connections. **(A)** Unsupervised learning with no coordinate transformation. **(B)** Unsupervised learning with coordinate transformation. **(C)** Supervised learning with coordinate transformation.

and motor programs. As for the previous architectures, accuracy of the motor program is also tested by read-out from hidden neurons via linear association.

Dataset and Stimuli

The representation format adopted for the sensory stimuli was the same used in previous computational investigations (Zipser and Andersen, 1988; Pouget and Snyder, 2000; De Filippo De Grazia et al., 2012), which is broadly consistent with neurophysiological data recorded in animals performing tasks involving coordinate transformations (e.g., Andersen et al., 1985).

The visual input to the models consisted in a real-valued vector representing the position of the stimulus as a Gaussian peak of activity in a specific location. These visible neurons simulate the activity of the cortical areas supplying retinotopic sensory information to the posterior parietal cortex. The retinotopic map consisted in a square matrix of 17×17 neurons, which employed a population code with Gaussian tuning functions (standard deviation $= 4°$). Visual receptive fields were uniformly spread between $-9°$ and $+9°$ with increments of $3°$, both in the horizontal and vertical dimensions.

Four postural maps, each one consisting of 17 neurons, were used to represent the horizontal and vertical positions of the eye and the effector. These visible neurons used a sigmoid activation function (steepness parameter $= 0.125$) to represent postural information between -18 and $+18°$, with steps of $3°$.

The motor program consisted in a real-valued vector representing the target position of the stimulus. Similarly to the retinotopic map, it was coded as a square matrix of 25×25 neurons, which employed a population code with Gaussian tuning functions to represent target position in coordinates centered on the effector (standard deviation $= 6°$). Motor programs were uniformly spread between $-9°$ and $+9°$ with increments of $3°$, both in the horizontal and vertical dimensions.

In order to create the stimuli dataset, all possible combinations of visual input and postural signals were first generated, and the corresponding motor program (target location) was computed. We then balanced the patterns to ensure that target locations were equally distributed across the motor map to avoid position biases when decoding the motor program. This resulted in a total of 28,880 patterns, which were randomly split into a training set (20,000 patterns) and an independent test set (8,880 patterns). The latter was used to assess the generalization performance of the models.

Learning Architectures

Despite they differ in several aspects, Boltzmann machines and autoencoders can both be defined within the mathematical framework of energy-based models (Ranzato et al., 2007), where the learning objective is to carve the surface of an energy function so as to minimize the energies of training points and maximize the energies of unobserved points. A set of latent variables is used to learn an internal code that can efficiently represent the observed data points, and since the number of latent variables is usually smaller than that of the observed variables the encoding process can be interpreted as a form of dimensionality reduction (Hinton and Salakhutdinov, 2006). In this unsupervised setting, the model learns the statistical structure of the data without the need for any explicit, external label.

Restricted Boltzmann Machines (RBMs)

Boltzmann machines are stochastic neural networks that use a set of hidden neurons to model the latent causes of the observed data vectors, which are presented to the network through a set of

visible neurons (Ackley et al., 1985). In the "restricted" case, the network connectivity is constrained in order to obtain a bipartite graph (i.e., there are no connections within the same layer; see **Figure 2A** for a graphical representation). The behavior of the network is driven by an energy function E, which defines the joint distribution of the hidden and visible neurons by assigning a probability value to each of their possible configurations:

$$p(v,h) = \frac{e^{-E(v,h)}}{Z}$$

where v and h are the column vectors containing the values of visible and hidden neurons, respectively, and Z is the partition function. The energy function is defined as a linear combination of visible and hidden neurons' activation:

$$E(v,h) = -b^T v - c^T h - h^T W v$$

where W is the matrix of connections weights, b and c are two additional parameters known as unit biases and T denotes the transpose operator. Since there are no connections within the same layer, hidden neurons are conditionally independent given the state of visible neurons (and vice versa). In particular, the activation probability of the neurons in each layer conditioned on the activation of the neurons in the opposite layer can be efficiently computed in one parallel step:

$$P(h_j = 1|v) = \sigma(c_j + \sum_i w_{ij} v_i)$$
$$P(v_i = 1|h) = \sigma(b_i + \sum_j w_{ij} h_j)$$

where σ is the sigmoid function, c_j and b_i are the biases of hidden and visible neurons (h_j and v_i respectively), and w_{ij} is the connection weight between h_j and v_i. Learning in RBMs can be performed through maximum-likelihood, where each weight should be changed at each step according to a Hebbian-like learning rule:

$$\Delta W = \eta(v^+ h^+ - v^- h^-)$$

where η represents the learning rate, $v^+ h^+$ are the visible-hidden correlations computed on the training data (positive phase), and $v^- h^-$ are the visible-hidden correlations computed according to the model's expectations (negative phase). Model's expectations have been traditionally computed by running Gibbs sampling algorithms until the network reached equilibrium (Ackley et al., 1985). However, more efficient algorithms such as contrastive divergence (Hinton, 2002) speed-up learning by approximating the log-probability gradient. The reader is referred to Hinton (2010) and Zorzi et al. (2013) for more details about RBMs and for the discussion of hyper-parameters of the learning algorithm.

In our simulations, RBMs were trained using 1-step contrastive divergence with a learning rate of 0.03, a weight decay of 0.0002 and a momentum coefficient of 0.9, which was initialized to 0.5 for the first few epochs. Learning was performed using a mini-batch scheme, with a mini-batch size of 4 patterns, for a total of 100 learning epochs (reconstruction error always converged). Sparse representations were encouraged by forcing the network's internal representations to rely on a limited number of active hidden units, that is, by driving the probability q of a unit to be active to a certain desired (low) probability p (Lee et al., 2008). For logistic units, this can be practically implemented by first calculating the quantity q-p, which is then multiplied by a scaling factor and added to the biases of each hidden units at every weight update. When the sparsity constraint was applied, we always verified that the average activation of hidden units was indeed maintained below the desired level. All the simulations were performed using an efficient implementation of RBMs on graphic processors (Testolin et al., 2013). The complete source code is available for download[1].

Autoencoders

Similarly to RBMs, autoencoders rely on a single layer of nonlinear hidden units to compactly represent the statistical regularities of the training data. However, autoencoders are feed-forward, deterministic networks trained with error backpropagation (Bengio et al., 2007). The training data is presented to a layer of input units, and the learning goal is to accurately reconstruct such input vector into a separate, output layer. An autoencoder is therefore composed of a set of encoding weights W^1 that are used to compute the activation of hidden h units given the activation of input units v, and a set of decoding weights W^2 that are used to compute the network reconstructions v_rec from the activations of hidden units:

$$h = \sigma(W^1 v + c)$$
$$v\_rec = \sigma(W^2 h + b)$$

where b and c are the vectors of output and hidden unit biases, and σ is the sigmoid function (see **Figure 2B** for a graphical representation). The error function E to be minimized corresponds to the average reconstruction error, which is quantified by the sum across all output units of the squared difference between the original and the reconstructed values:

$$E = \frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K} (v_k - v\_rec_k)^2 + \beta^* \Omega_{sparsity}$$

where K is the number of output units and N is the number of training patterns. Similarly to RBMs, sparse representations can be induced by adding to the cost function a regularization term $\Omega_{sparsity}$ that takes a large value when the average activation value q of each hidden neuron diverges from a certain desired (low) value p. In particular, the sparsity constraint was implemented as the Kullback-Leibler divergence from q to p:

$$\Omega_{sparsity} = \sum_{i=1}^{H} KL(p \,||\, q_i)$$

where H is the number of hidden units. As for RBMs, when sparsity was applied we always verified that the average activation of hidden units was indeed maintained below the desired level.

[1] http://ccnl.psy.unipd.it/research/deeplearning

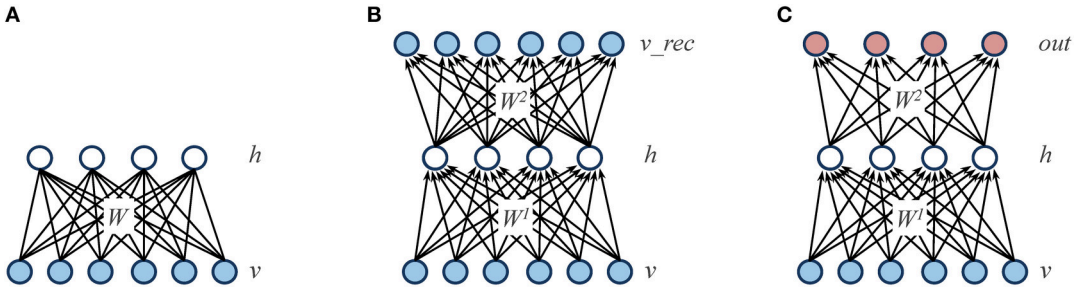


FIGURE 2 | Graphical representations of the different learning architectures used in the simulations. (A) Restricted Boltzmann Machine (RBM): the learning objective is to accurately reconstruct the input patterns presented through the visible layer (*v*) by relying on a set of hidden units (*h*), which represent the latent structure of the data. The reconstruction is performed by using a weight matrix (*W*) that contains symmetric (i.e., undirected) connections. **(B)** Autoencoder: as for RBMs, the learning objective is to accurately reconstruct the input patterns presented through the visible layer (*v*) by relying on a set of hidden units (*h*). However, the reconstruction is performed on a separate layer of units (*v_rec*) by using two weight matrices (W^1 and W^2) that contain directed connections. **(C)** Feed-forward, supervised network: in contrast to RBMs and autoencoders, the learning objective is to minimize the mapping error between the input patterns presented through the visible layer (*v*) and a distinct set of output patterns presented through a dedicated layer (*out*).

In our simulations, we used an efficient implementation of autoencoders provided by the MATLAB Neural Network toolbox (Demuth and Beale, 1993). Learning was performed using standard scaled conjugate gradient descent (Møller, 1993) with adaptive learning rate, using a weight decay factor of 0.0002 and a batch processing scheme, for a total of 150 learning epochs (reconstruction error always converged).

Feed-Forward, Supervised Networks

In order to better assess the impact of the learning regimen, we compared the unsupervised learning architectures described above with a standard, supervised architecture implemented as a feed-forward network with one hidden layer (Zipser and Andersen, 1988). Similarly to autoencoders, learning can be performed using error backpropagation (see **Figure 2C** for a graphical representation). We used an efficient implementation of feed-forward networks provided by the MATLAB Neural Network toolbox[2]. Learning rate was set to 0.05 and training was performed for a total of 2500 learning epochs (output error always converged).

Testing Procedure

For each experimental setting, we run 10 different networks in order to collect simulation statistics. In the results, we therefore always report mean values along with standard deviations.

Decoding Internal Representations by Linear Read-Out

Following unsupervised learning, a linear read-out was performed from the internal (hidden layer) distributed representations of the networks in order to assess how well they could support a supervised mapping to the target motor program through a simple linear projection (Pouget and Snyder, 2000). The read-out was implemented using a linear neural network trained with the delta rule (Widrow and Hoff, 1960). Learning was performed for 250 epochs using mini-batches of 20 patterns. Learning rate was set to 0.07, and weight decay of 0.000001 was used as a regularizer. Classifier performance was always measured on the separate test set. Test errors always matched those obtained on the training set, indicating that the read-out was robust to overfitting.

The output of the classifier was first compared with the target motor program by computing the Root Mean Squared Error (RMSE) between the two matrices. However, a more useful performance measure was obtained by first decoding the Center Of Mass (COM) of the output distribution, which was then compared with the actual coordinates of the motor program. This measure allows to quantify the read-out error in degrees: following Zipser and Andersen (1988), the mapping was considered to be successful if the error was below the distance between the centers of the Gaussian tuning functions in the retinotopic map (i.e., 3°). If the latter mapping accuracy was not achieved, we did not consider the network for subsequent analyses. We found the RMSE and COM measures to be always consistent with each other, so we only report COM results.

Measuring Single-Neuron and Population Sparseness

An index of *single-neuron sparseness* was computed using a well-established procedure employed in neurophysiological investigations (Rolls and Tovee, 1995; Vinje and Gallant, 2000), which describes the activity fraction a of each neuron across stimuli as:

$$a = \frac{(\sum r_i/n)^2}{\sum(r_i^2/n)}$$

where r_i is the firing rate of the neuron to the i-th stimulus in the set of n stimuli. This is a useful measure of the extent of the tail of the distribution, in this case of the firing rates of the neuron to each stimulus. Mean single-neuron sparseness

[2]MATLAB provides several improved versions of the standard backpropagation algorithm. An extended set of preliminary simulations was used to establish the best performing variant. In particular, these training functions were tested: *traingdm* (gradient descent with momentum); *traingda* (gradient descent with adaptive learning rate); *traingdx* (gradient descent with momentum and adaptive learning rate); *trainscg* (scaled conjugate gradient) and *trainrp* (resilient backpropagation). The most stable and accurate learning algorithm was resilient backpropagation (Riedmiller and Braun, 1993).

for each network was then calculated by averaging the activity fraction *a* across all hidden neurons. A low value (minimum value is 0, maximum value is 1) indicates that the distribution has a long tail, which means that, on average, each neuron has high activation levels only for a small subset of input patterns. This method for quantifying sparseness has a number of advantages (Rolls and Tovee, 1995): (a) it results from formal analyses of the capacity of neural networks using an approach derived from theoretical physics (Treves and Rolls, 1991); (b) it can be applied both to binary neurons and to neurons with continuous (graded) firing rates; (c) it makes no assumption about the form of the firing rate distribution and (d) it makes no assumption about the mean and the variance of the firing rate.

Following Froudarakis et al. (2014) we also computed an index of *population sparseness*, on which the activity fraction is computed over the entire hidden layer, that is, by considering r_i as the firing rate of the i-th neuron and n as the total number of neurons. Mean population sparseness for each network was then calculated by averaging the activity fraction *a* across all stimuli. A low value of population sparseness indicates that, on average, each stimulus elicits high activations only for a small subset of hidden neurons.

Receptive Fields Emerging in the Hidden Neurons

In order to qualitatively assess the type of visual features extracted by individual hidden neurons, we first analyzed the weight matrices by separately plotting the strengths of the connections between each hidden neuron and all the visible neurons corresponding to the retinal input. Weights were plotted on a gray scale, with dark colors indicating strong inhibitory connections and light colors representing positive, excitatory connections. This allowed to assess whether hidden neurons learned location-specific receptive fields, for example by developing stronger projections to specific regions of the visual field.

Gain Modulation Indexes

We then analyzed the response of hidden neurons using a standard approach adopted in neurophysiological studies to assess gain modulation in parietal neurons (Andersen et al., 1985). First, we probed the hidden neurons in order to only select the "visual" ones, that is, those responding to the portion of input vectors representing the retinotopic map (De Filippo De Grazia et al., 2012). To this aim, we first recorded all hidden neurons' activations when the network received as input only all possible combinations of eye and effector positions (i.e., the retinotopic map and, if present, the motor program, were set to zero), and for each neuron we selected the positions corresponding to maximum activation. We then probed again each neuron, this time providing as input all possible retinotopic signals along with the preferred combination of postural signals. The neuron was considered as visual if its maximum activity differed by more than 10% from that recorded in the absence of visual input. Non-visual neurons were discarded from subsequent analyses[3]. We then computed a gain modulation index (GMI) for each neuron by recording its response to each target location as a function of eye and effector position (Pouget and Snyder, 2000). We first identified the combination of postural and retinal input producing the maximum neuron activation value. Starting from this input combination, we systematically varied each postural variable (one at a time, keeping all the others fixed) and computed gain modulation as the normalized ratio between the maximum and minimum activation values. Therefore, each neuron was characterized by four different GMIs, representing the gain for each postural variable with respect to horizontal and vertical axes. We finally sorted all hidden neurons into four different categories based on the combination of GMI indexes (using a threshold of 0.5 to establish modulation): (i) no modulation (i.e., purely visual neurons), (ii) modulation by eye position only, (iii) modulation by effector position only, and (iv) modulation by both eye and effector position.

RESULTS

Learning always converged for all models. For unsupervised models, convergence was monitored by measuring the mean reconstruction error on the whole training set. Autoencoders required more learning epochs to converge, but also achieved a lower reconstruction error compared to RBMs. This is probably due to the fact that autoencoders are natively real-valued. Existing real-valued extensions of RBMs (Cho et al., 2011) assume that the input values are normally distributed, which was not our case, so we preferred to use standard RBMs. Learning in the feed-forward, supervised models required almost 20 times more epochs to converge (the number of epochs required by each learning architecture is reported in **Table 1**).

A first, qualitative analysis shows that RBMs and autoencoders developed different types of receptive fields. As shown in **Figure 3**, autoencoders learned homogeneous, location-specific receptive fields that uniformly covered the central regions of the visual input. On the other hand, while some neurons in the RBMs learned location-specific receptive fields resembling those of autoencoders, other neurons developed more complex receptive fields covering larger regions of the visual fields, sometimes also simultaneously covering symmetrical portions of the input image.

The quantitative analyses (see Section Testing Procedure) allowed to group hidden neurons into different categories according to their response profiles. In line with empirical findings (Duhamel et al., 1997), there were always some neurons that did not exhibit any form of gain modulation (i.e., "purely visual" neurons), that is, they responded to visual stimuli at a given spatial location regardless of eye- or effector- positions. However, the majority of neurons developed gain fields, which in some cases were modulated exclusively by either eye or effector position (see, for example, top panels of **Figure 4**), while in other

[3]It turned out that more than 95% of hidden neurons responded to the visual input, with a minimum activation value exceeding a threshold of 0.1.

TABLE 1 | Read-out errors for each learning architecture and space coding task, as a function of hidden layer size.

Space coding task	Layer size	RBMs		Autoencoders		Supervised Feed-forward	
		Read-out	Epochs	Read-out	Epochs	Read-out	Epochs
No transformation	200	1.59 (0.08)	100	1.05 (0.05)	150		
	300	1.39 (0.07)	100	0.91 (0.04)	150		
	400	1.30 (0.08)	100	0.86 (0.04)	150		
	500	1.25 (0.04)	100	0.89 (0.02)	150		
	600	1.23 (0.05)	100	0.90 (0.02)	150		
	700	1.33 (0.04)	100	0.90 (0.03)	150		
Coordinate transformation	500	1.55 (0.15)	100	1.45 (0.05)	150	1.46 (0.06)	2,500
	600	1.47 (0.12)	100	1.46 (0.06)	150	1.45 (0.02)	2,500
	700	1.52 (0.11)	100	1.45 (0.05)	150	1.46 (0.05)	2,500
	800	1.57 (0.11)	100	1.47 (0.08)	150	1.47 (0.08)	2,500
	900	1.56 (0.16)	100	1.45 (0.07)	150	1.47 (0.04)	2,500

Read-out errors are in degrees, and standard deviations are reported in parentheses. The "Epochs" column shows the number of epochs required by each learning architecture to converge.

cases were modulated by both eye and effector position, resulting in multiple gain fields (bottom panels of **Figure 4**).

Unsupervised Learning without Coordinate Transformation

In a first set of simulations, the number of hidden units was fixed to 400[4], while the sparsity constraint was varied between 0.004 (very strong sparsity constraint, requiring low average activation) and 0.3 (mild sparsity constraint). As shown in **Figure 5**, the effect of sparsity constraints on the two unsupervised architectures was markedly different. Levels of sparsity constraints in the first two rows are represented using a color scale, where lighter tones indicate stronger sparsity and dark tones indicate mild sparsity. Gain modulation in RBMs (**Figure 5A**) was not affected by imposing sparsity constraints. In all cases, we found a modest percentage (around 10%) of purely visual neurons, which were not modulated by any postural information. A more consistent percentage of neurons (20–25%) were modulated either by eye or by effector positions, while the remaining neurons (40–50%) exhibited multiple gain fields. Read-out accuracy (**Figure 5C**) was always good, except for the networks trained with very strong sparsity constraints (0.01 and 0.004), where learning failed and read-out accuracy did not achieve a mean error lower than 3°. The lowest read-out error (around 1.3°) was obtained with a sparsity constraint of 0.05. In contrast, autoencoders were extremely sensitive to sparsity constraints: Strong sparsity constraints resulted in a compressed code where the majority of hidden neurons (60%) exhibited multiple gain fields (**Figure 5B**). When the sparsity pressure was reduced gain fields gradually disappeared, and the majority of neurons did not exhibit any modulation at all. Read-out error was generally lower compared to RBMs, and learning failed only for the networks trained with extreme (0.004) or without any sparsity constraints (**Figure 5D**). Notably, also for autoencoders the lowest read-out error (around 0.9°) was obtained with a sparsity constraint of 0.05, which also resulted in a distribution of gain fields more similar to that of RBMs.

Interestingly, the objective indexes of sparseness revealed that RBMs are naturally much sparser than autoencoders (see bottom panels of **Figure 5**). Indeed, the level of sparsity constraint turned out to have a very weak effect on population sparseness in RBMs (**Figure 5E**), as also confirmed by linear regression [$r^2 = 0.32$, b = 0.05, $p < 0.001$, $n = 50$]. Single-neuron sparseness was only affected when the sparsity constraint operated below a critical level of 0.1. In order to measure what would be the "spontaneous" index of sparseness in RBMs, we trained an additional set of networks without imposing any sparsity constraint, which resulted in a single-neuron sparseness of 0.56 and a population sparseness of 0.28, showing that RBMs naturally exhibit a remarkable sparseness. In contrast, sparsity constraints in autoencoders had a marked effect on both single-neuron sparseness and population sparseness (**Figure 5F**), suggesting that this architecture naturally develops extremely distributed internal representations. In particular, the effect of level of sparsity constraint on population sparseness for autoencoders [linear regression: $r^2 = 0.88$, b = 0.43, $p < 0.001$, $n = 50$] was almost one order of magnitude higher compared to RBMs. In order to measure the spontaneous index of sparseness in autoencoders, we trained an additional set of networks with a very low sparsity constraint (0.8), which is the borderline condition that still guaranteed successful learning. The latter simulations yielded sparseness values indicating non-sparse, highly distributed representations (single-neuron sparseness = 0.97; population sparseness = 0.98).

In a second set of simulations, the sparsity constraint for both architectures was fixed to the value leading to the best performance (0.05), while the size of the hidden layer was varied systematically between 200 and 700 neurons in steps of 100. This range allowed to explore the effect of relatively large increases and decreases of hidden layer sizes with respect to the previous simulations, without compromising

[4]The initial size of the hidden layer was determined empirically based on a set of pilot simulations to guarantee reliable and relatively fast convergence of learning.

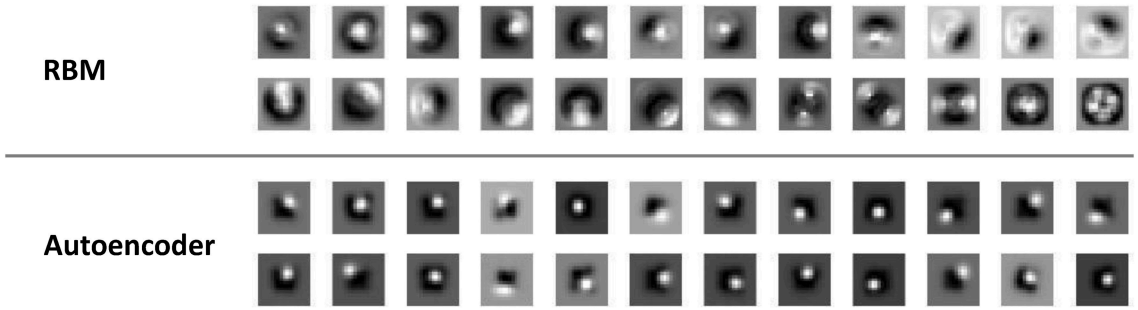


FIGURE 3 | Visual receptive fields. Samples of receptive fields emerging from RBMs (top panel) and autoencoders (bottom panel) on the unsupervised learning task that did not require coordinate transformations. Similar receptive fields emerged from the unsupervised learning task involving coordinate transformations.

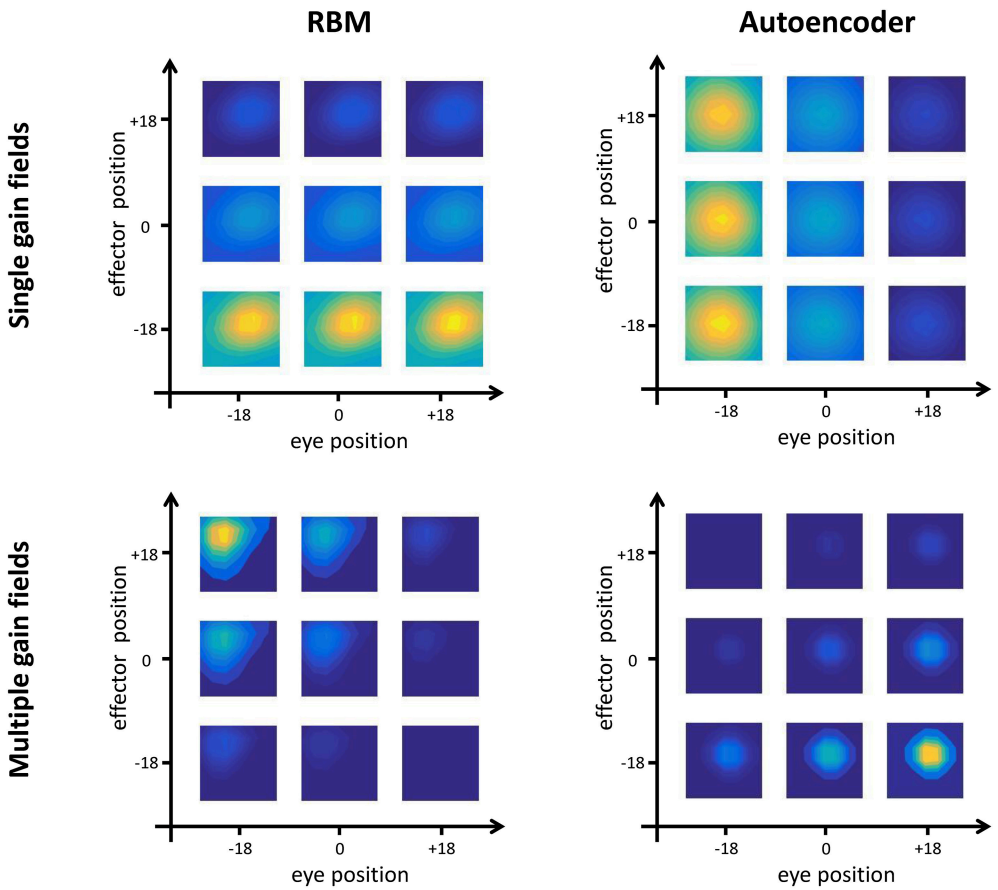


FIGURE 4 | Gain field coding. Examples of single (top panels) and multiple (bottom panels) gain fields emerging in the hidden neurons of RBMs **(left)** and autoencoders **(right)**. Colors represent the amount of activation, with yellow indicating highest activation and dark blue indicating lowest activation. Single gain fields are characterized by a modulation of the neuron's activation that depends only on one postural signal (in the figure, effector position for the RBM and eye position for the autoencoder). In multiple gain fields, the activation is modulated by both signals.

the learning accuracy. For both architectures, the read-out accuracy was not affected by hidden layer size, and the mapping error was always below 2° (read-out errors for all different hidden layer sizes are reported in **Table 1**). However, as shown in **Figure 6**, also in this case the manipulation had different effects for the two architectures (lighter colors indicate smaller sizes). The type of encoding developed by RBMs (**Figure 6A**) was affected by hidden layer size: When the number of hidden neurons decreased the network developed more compressed codes, by increasing the percentage of

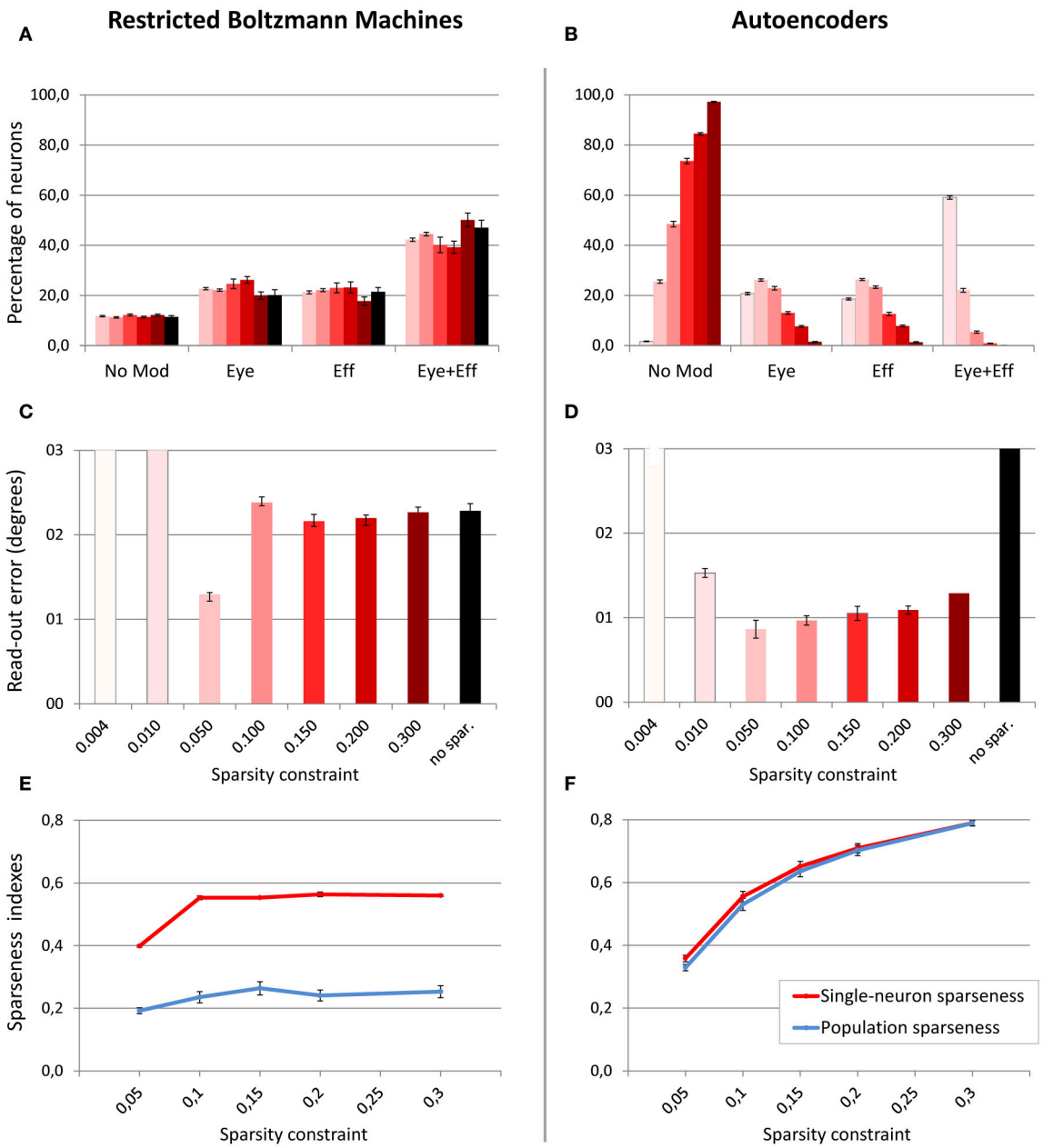


FIGURE 5 | Effect of sparsity constraints. Distribution of gain field types emerging in the hidden neurons of RBMs **(A)** and autoencoders **(B)** with varying levels of sparsity constraint. Sparsity constraints are represented in different columns using a color scale, where lighter tones indicate stronger sparsity constraints and dark red indicates mild sparsity constraints. Read-out errors obtained at each level of sparsity constraint for RBMs **(C)** and autoencoders **(D)**. Single-neuron and population sparseness as a function of sparsity constraints for RBMs **(E)**, and autoencoders **(F)**. Note that small values indicate stronger sparseness.

multiple gain fields and reducing the percentage of neurons modulated by only eye or effector positions. Interestingly, it turned out that the manipulation of hidden layer size had a clear impact also on the underlying sparseness of the representation (**Figure 6C**). Indeed, both single-neuron and population sparseness decreased as a function of number of hidden neurons [linear regressions: single-neuron sparseness, $r^2 = 0.92$, b = 0.22, $p < 0.001$, $n = 60$; population sparseness, $r^2 = 0.96$, b = 0.21, $p < 0.001$, $n = 60$]. This result suggests that the distribution of gain fields in RBMs might in fact be modulated by the underlying sparseness of the representation. This was confirmed by the high correlation between the percentage of multiple gain-fields and the objective sparseness indexes [Pearson correlations: single-neuron sparseness: $r = -0.85$, $p < 0.001$; population sparseness, $r = -0.92$, $p < 0.001$].

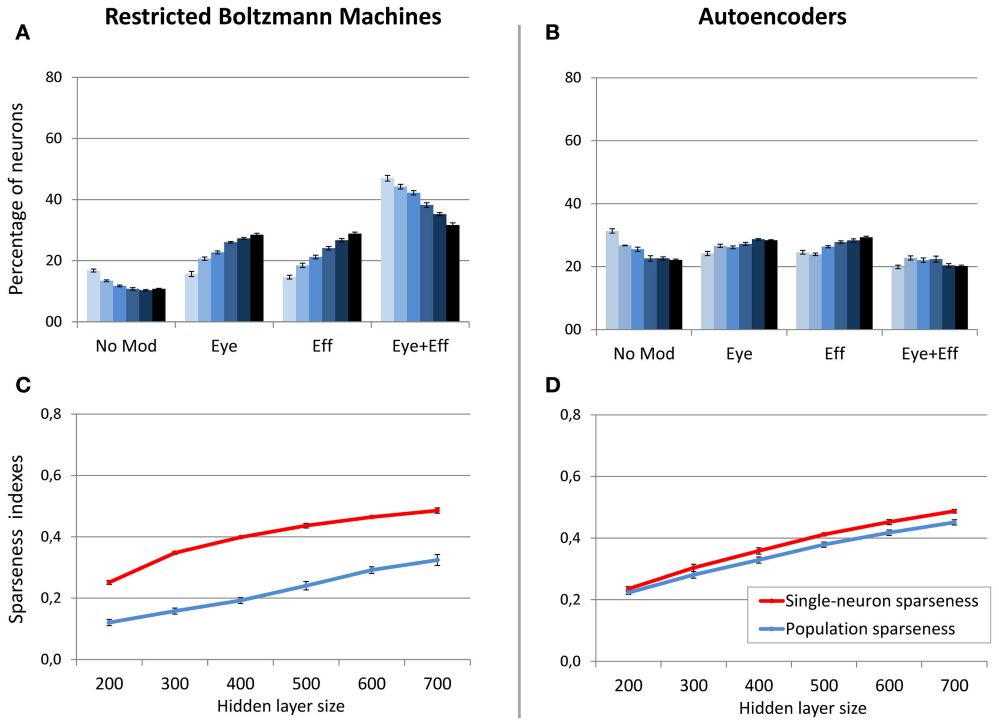


FIGURE 6 | Effect of hidden layer size, with strong sparsity constraint. Distribution of gain field types emerging at the hidden layer of RBMs **(A)** and autoencoders **(B)** with varying number of hidden neurons and sparsity constraint fixed to 0.05. Lighter tones indicate smaller layers and dark blue indicates larger layers. Single-neuron and population sparseness as a function of hidden layer size for RBMs **(C)** and autoencoders **(D)**. Note that small values indicate stronger sparseness.

On the contrary, neuronal tuning functions in autoencoders were not affected by hidden layer size, as this architecture always developed uniformly distributed types of gain fields (**Figure 6B**). Interestingly, as for RBMs the reduction of hidden layer size caused a decrease in both single-neuron sparseness and population sparseness [linear regressions: single-neuron sparseness, $r^2 = 0.98$, b = 0.25, $p < 0.001$, $n = 60$; population sparseness, $r^2 = 0.98$, b = 0.23, $p < 0.001$, $n = 60$]. However, the sparseness indexes did not correlate with the percentage of multiple gain-fields [all $p > 0.05$]. This suggests that similar changes in the underlying sparseness do not produce the same effect on the gain field distribution in RBMs and autoencoders.

In order to better clarify if the size of the hidden layer in RBMs modulates the distribution of gain fields only when sparseness is externally forced (i.e., when using a sparsity constraint of 0.05), in a subsequent set of simulations the sparsity constraint was set to a weak level (0.2) and the size of the hidden layer was manipulated as in the previous condition. In this case the distribution of gain fields did not systematically change (**Figure 7A**) but, notably, also the population sparseness was not affected (**Figure 7C**) [linear regression: $r^2 = 0.24$, b = 0.03, $p < 0.001$, $n = 60$]. Correlation analyses still revealed a correlation between population sparseness and the percentage of multimodal gain fields [$r = -0.54$, $p < 0.001$], while the correlation with single-neuron sparseness was not significant [$p > 0.05$]. These results show that, for RBMs, population sparseness is a robust predictor of the distribution of gain fields: if RBMs must rely only of few active neurons to represent each sensory stimulus, they will develop more compressed spatial codes, such as those based on multiple gain fields. The corresponding simulation with autoencoders was relatively uninformative, because the weak level of sparsity constraint resulted in the absence of multimodal gain fields (**Figure 7B**).

Unsupervised Learning with Coordinate Transformation

As discussed before, in this learning setting the motor program was included as input during unsupervised learning. This implies that two different coordinate systems (i.e., retinotopic and motor) are implicitly associated during training. For these simulations, we focused on hidden layer size, which was varied between 500 and 900 neurons in steps of 100. Note that the larger number of hidden neurons with respect to the previous simulations is motivated by the increased size and complexity of the training patterns. The sparsity constraint was fixed to 0.05, which was the value resulting in more accurate read-outs and more balanced distribution of gain fields for both RBMs and autoencoders in the previous set of simulations. For both architectures, read-out accuracy was always good (mapping error below 2°) and it was not affected by hidden layer size (see **Table 1**). As shown in **Figure 8**, RBMs generally developed a larger percentage of gain fields compared to autoencoders. In particular, the number of multiple gain fields was much higher for RBMs. Interestingly, for both architectures also in this case the manipulation of hidden

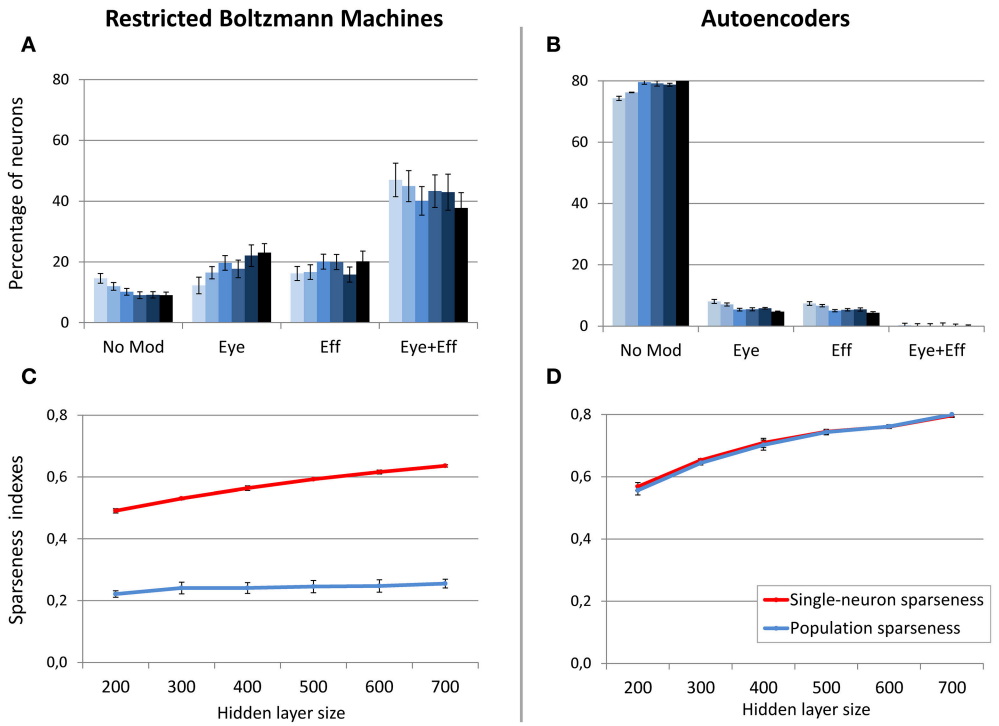


FIGURE 7 | Effect of hidden layer size, with moderate sparsity constraint. Distribution of gain field types emerging at the hidden layer of RBMs **(A)** and autoencoders **(B)** with varying number of hidden neurons and sparsity constraint fixed to 0.2. Lighter tones indicate smaller layers and dark blue indicates larger layers. Single-neuron and population sparseness as a function of hidden layer size for RBMs **(C)** and autoencoders **(D)**. Note that small values indicate stronger sparseness.

layer size produced a systematic change in the sparseness indexes [linear regressions: RBMs single-neuron sparseness, $r^2 = 0.98$, b $= 0.14$, $p < 0.001$, $n = 50$; RBMs population sparseness, $r^2 = 0.95$, b $= 0.07$, $p < 0.001$, $n = 50$; autoencoders single-neuron sparseness, $r^2 = 0.98$, b $= 0.10$, $p < 0.001$, $n = 60$; autoencoders population sparseness, $r^2 = 0.98$, b $= 0.06$, $p < 0.001$, $n = 60$]. For both architectures, population and single-neuron sparseness were highly correlated with the percentage of multiple gain fields [Pearson correlations: RBMs single-neuron sparseness, $r = -0.94$, $p < 0.001$; RBMs population sparseness, $r = -0.96$, $p < 0.001$; autoencoders single-neuron sparseness, $r = -0.88$, $p < 0.001$; autoencoders population sparseness, $r = -0.88$, $p < 0.001$]. This finding corroborates the hypothesis that, especially for RBMs, reducing the number of active neurons results in more compressed codes based on multiple gain fields, which might be particularly advantageous in the current scenario since learning involved coordinate transformations. In contrast, fewer neurons in autoencoders exhibited multiple gain modulation (**Figure 8B**), even if also in this case the percentage of multiple gain fields was proportional to the underlying level of sparseness.

Supervised Learning with Coordinate Transformation

The final set of simulations reproduced the feed-forward, supervised architecture used by Zipser and Andersen (1988). As in their original work, we did not enforce sparse coding. The size of the hidden layer was varied between 500 and 900 in steps of 100. Learning always converged and both the feed-forward mapping error and the read-out error were below 3° (see **Table 1**). As shown in **Figure 9**, this type of learning architecture developed a strikingly lower proportion of gain-modulated neurons in the hidden layer: Almost 80% of the neurons did not exhibit any form of gain field. The remaining ones were almost uniformly distributed across the three other types (about 8% for either eye or effector position; 10% for multiple gain modulation). Moreover, differently from the unsupervised architectures, the type of gain modulation was not affected by changes in the hidden layer size. This result is remarkable, because it suggests that feed-forward, supervised architectures are much less prone to develop efficient forms of space coding based on gain fields. One possible explanation for this finding is that the type of coding used to represent the motor program might have affected the efficiency of error backpropagation, which was not able to properly propagate the error signals across the hidden layer. Indeed, also Zipser and Andersen (1988) found some discrepancy between the type of gain modulations developed when using a monotonic output format compared to the Gaussian output format (which was adopted in the present study). However, the previous simulations with autoencoders showed that backpropagation can give rise to a variety of strong gain modulations when it is applied within an unsupervised learning setting. Another, more critical factor might instead be the absence of sparsity constraints, which were

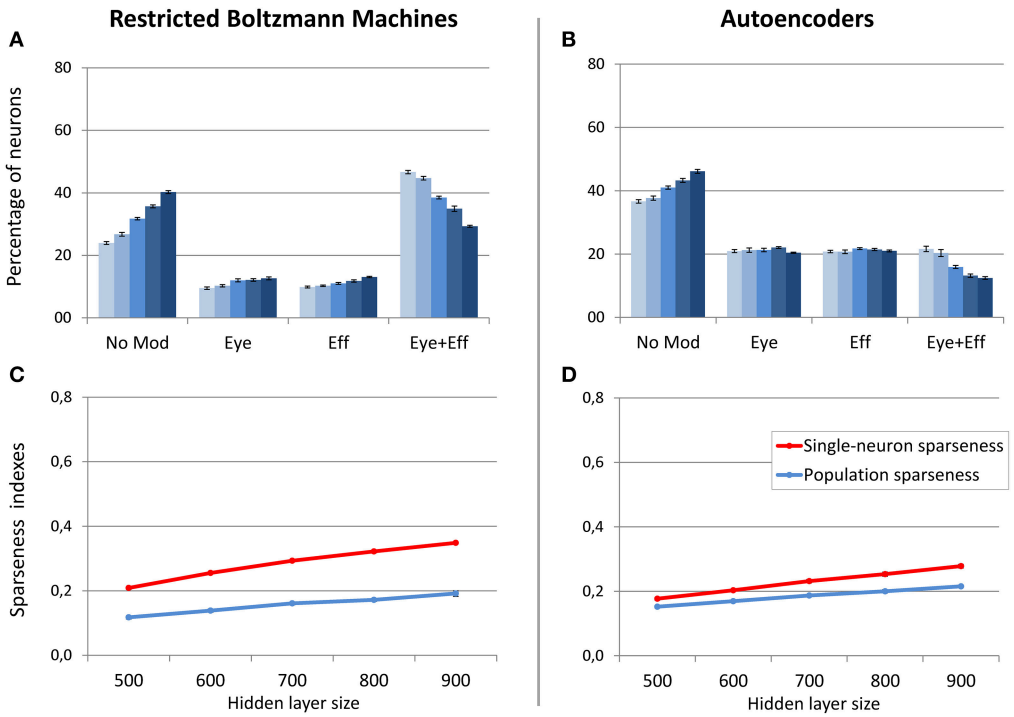


FIGURE 8 | Unsupervised learning involving coordinate transformations. Distribution of gain field types emerging at the hidden layer of RBMs **(A)** and autoencoders **(B)** with varying number of hidden neurons. Lighter tones indicate smaller layers and dark blue indicates larger layers. Single-neuron and population sparseness as a function of hidden layer size for RBMs **(C)** and autoencoders **(D)**. Note that small values indicate stronger sparseness.

not used in the feed-forward models but turned out to be fundamental with autoencoders.

DISCUSSION

In this study we investigated the role of architectural and learning constraints in neural network models that learned to encode spatial information resulting from the combination of visual and postural signals. Results showed that, compared to the supervised architecture originally proposed by Zipser and Andersen (1988), unsupervised architectures like Restricted Boltzmann Machines (RBMs) and autoencoders discover space codes that more closely reproduce the distribution of neuronal tuning functions observed in neurophysiological experiments. In particular, the majority of hidden neurons of RBMs and autoencoders exhibited gain modulation, which in some cases only depended either on eye or effector position, while in other cases depended on both eye and effector positions, thereby resulting in multiple gain fields. In fact, all unsupervised models developed a much higher percentage of gain modulated neurons compared to the supervised models. Although the precise distribution of gain field types in the cerebral cortex depends on the exact recording site (Colby and Goldberg, 1999), our simulations suggest that this efficient form of encoding emerges more naturally if the task requires to reconstruct the whole sensory input, rather than to simply discover a feed-forward mapping to a target motor program. In other words, gain field coding might be useful when the goal is to discover "good" internal representations of the input data, that is, when the aim is to unveil and more explicitly encode the latent factors underlying the input data distribution.

As a general principle, the quality of an internal representation should reflect how well the learned features disentangle as many factors of variation as possible, at the same time discarding as little information about the data as is practical (Bengio et al., 2013). In the specific case of sensorimotor transformations, it has been proposed that good internal representations should have a variety of properties, such as the ability to combine the input signal in a nonlinear way, the ability to fully cover the range of possible input values, and the ability to represent multiple reference frames simultaneously within the same neurons (Pouget and Snyder, 2000). Populations of gain modulated neurons satisfy these requirements, allowing to encode visual space using a flexible set of basis functions. Notably, our simulations showed that this allows to learn coordinate transformations in two separate stages, by first learning the set of basis functions in a completely unsupervised way, and then learning appropriate mappings to target motor commands by relying on explicit supervision or reinforcement signals (Pouget and Snyder, 2000).

Our analyses also highlighted several differences in the spatial codes learned by RBMs and autoencoders, despite the fact that these two unsupervised architectures are often considered

similar, if not equivalent (Ranzato et al., 2007; Coates et al., 2011). Even from a simple, qualitative analysis of the visual receptive fields, it turned out that these models developed different internal representations. Subsequent analyses conducted to investigate the emergence of gain fields further revealed that the distribution of hidden neurons' tuning functions in RBMs and autoencoders was similar only for a very narrow choice of the hyper-parameters. An important finding was that RBMs spontaneously exhibited a remarkable level of sparseness, which made them insensitive to external sparsity constraints, and which encouraged the emergence of compressed forms of spatial coding based on gain modulation. The spontaneous level of sparseness in RBMs could be manipulated only within a narrow range, by imposing an extreme sparsity constraint and jointly reducing the size of the hidden layer. This forced the internal representations to rely on even fewer neurons, and produced an increase in the percentage of multiple gain fields. These findings are consistent with the intuition that reducing the computational resources forces the networks to discover more complex (and compressed) forms of encoding, such as those resulting from the combination of many sensory/postural variables into multiple gain fields. Notably, for RBMs this was the case even when the task did not involve any coordinate transformations, which implied that postural variables were orthogonal. In other words, despite the fact that eye and effector positions were varied independently across training patterns, the RBMs with fewer active neurons often combined these signals together, resulting in an increase of multiple gain fields. Nevertheless, unlike autoencoders, RBMs always dedicated some representational resources also to encode eye and effector positions independently.

Autoencoders turned out to rely on much more distributed representations compared to RBMs, and were therefore extremely sensitive to external sparsity constraints. This implies that, compared to RBMs, autoencoders have an additional hyper-parameter that must be carefully tuned. Notably, when the sparsity pressure was reduced hidden neurons in the autoencoders did not develop any form of gain modulation. Only for specific values of sparsity constraints autoencoders could reproduce the variety of gain field types observed in neurophysiological data (Brotchie et al., 1995; Graziano et al., 1997; Snyder et al., 1998; Chang et al., 2009), with a distribution compatible with that of RBMs. However, in autoencoders the underlying sparseness indexes did not seem to be systematically related to the complexity of the emergent spatial codes. Though these findings alone do not allow to adjudicate between models, they call for a more systematic investigation of these different learning architectures, possibly spanning other domains and using a more direct comparison to neurophysiological data.

A plausible explanation for the striking differences in the spontaneous level of sparseness between RBMs and autoencoders can be found when considering the different processing dynamics embedded in these two neural network models. Indeed, in autoencoders the activation of each hidden neuron is deterministic, and simply corresponds to the (possibly graded) value returned by the non-linear, logistic activation function. In RBMs, instead, the value returned by the logistic function is treated as a probability, and the final activation of each hidden neuron is obtained by performing a stochastic binarization step. This important difference likely produces more sharp neuronal activations, driving RBMs to develop more sparse representations compared to autoencoders.

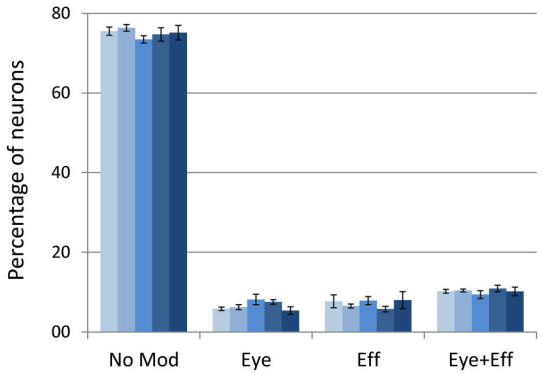


FIGURE 9 | Supervised learning of coordinate transformations. Distribution of gain field types emerging at the hidden layer of a feed-forward, supervised neural network similar to that used by Zipser and Andersen (1988) with varying number of hidden neurons. Lighter tones indicate smaller layers and dark blue indicates larger layers.

From a broader perspective, we believe that stochastic neural networks such as RBMs and their extension into hierarchical generative models will have an increasingly central role in neurocomputational modeling, because they provide a unique bridge between high-level descriptions of cognition in terms of Bayesian computation and low-level, mechanistic explanations inspired by the biophysical properties of real neuronal networks (Testolin and Zorzi, 2016). For example, generative neural networks are compatible with Bayesian approaches based on probabilistic population codes (Ma et al., 2006), which have been successfully used to simulate sensorimotor transformations with basis functions (Pouget and Sejnowski, 1997; Pouget and Snyder, 2000). RBMs extend the basis function approach by explaining how learning might shape the emergent neuronal gain fields, and they could similarly be combined with attractor dynamics to simulate optimal statistical inference over multisensory spatial representations (cf. Pouget et al., 2002) and spatial remapping in attention orienting (cf. Casarotti et al., 2012).

Moreover, the fact that generative networks can simulate both evoked (feed-forward) and intrinsic (feedback) neuronal activity makes them particularly suited to investigate spontaneous brain activity, which has been recognized as a fundamental property of the brain (Raichle, 2015) but whose computational role is still largely unknown. An intriguing hypothesis suggests that intrinsic activity could help with driving the brain close to states that are probable to be valid inferences once an external input arrives, thus potentially shortening the reaction time of the system

(Fiser et al., 2010). Stochastic, generative networks are consistent with this "sampling-based" framework, and also support the idea that neuronal noise could play an important role during sampling (Kirkpatrick et al., 1983), for example by keeping the system in a metastable state that facilitates flexible settling into the most appropriate configuration (Kelso, 2012; Deco et al., 2013). Notably, we are also beginning to better understand how these powerful models could be implemented with biologically more realistic architectures, such as those incorporating temporal dynamics and spike-based communication (Buesing et al., 2011; Nessler et al., 2013).

In conclusion, we hope that the recent breakthroughs in neurally-inspired machine learning will attract the interest of the neuroscience community, as these models hold great promise for improving our understanding of how learning shapes and organizes information processing in complex neuronal networks.

AUTHOR CONTRIBUTIONS

AT, MD, and MZ equally contributed to the research design. AT implemented the simulations. AT and MD analyzed the data. AT and MZ wrote the paper. All the authors are accountable for all aspects of the work in ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.

ACKNOWLEDGMENTS

This research was supported by grants from the European Research Council (no. 210922) and by the University of Padova (Strategic Grant NEURAT) to MZ. We are grateful to the Reviewers for their helpful comments on a previous version of this article.

REFERENCES

Ackley, D., Hinton, G. E., and Sejnowski, T. J. (1985). A learning algorithm for Boltzmann machines. *Cogn. Sci.* 9, 147–169. doi: 10.1207/s15516709cog0901_7

Andersen, R. A., Essick, G. K., and Siegel, R. M. (1985). Encoding of spatial location by posterior parietal neurons. *Science* 230, 456–458. doi: 10.1126/science.4048942

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: a review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 1798–1828. doi: 10.1109/TPAMI.2013.50

Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. (2007). Greedy layer-wise training of deep networks. *Adv. Neural Inf. Process. Syst.* 19, 153–170.

Brotchie, P. R., Andersen, R. A., Snyder, L. H., and Goodman, S. J. (1995). Head position signals used by parietal neurons to encode locations of visual stimuli. *Nature* 375, 232–235. doi: 10.1038/375232a0

Buesing, L., Bill, J., Nessler, B., and Maass, W. (2011). Neural dynamics as sampling: a model for stochastic computation in recurrent networks of spiking neurons. *PLoS Comput. Biol.* 7:e1002211. doi: 10.1371/journal.pcbi.1002211

Casarotti, M., Lisi, M., Umiltà, C., and Zorzi, M. (2012). Paying attention through eye movements: a computational investigation of the premotor theory of spatial attention. *J. Cogn. Neurosci.* 24, 1519–1531. doi: 10.1162/jocn_a_00231

Chang, S. W., Papadimitriou, C., and Snyder, L. H. (2009). Using a compound gain field to compute a reach plan. *Neuron* 64, 744–755. doi: 10.1016/j.neuron.2009.11.005

Cho, K., Ilin, A., and Raiko, T. (2011). "Improved learning algorithms for restricted boltzmann machines," in *International Conference on Artificial Neural Networks* (Espoo), 10–17.

Coates, A., Arbor, A., and Ng, A. Y. (2011). An analysis of single-layer networks in unsupervised feature learning. *Int. Conference Artif. Intell. Stat.* 15, 215–223.

Colby, C. L., and Goldberg, M. E. (1999). Space and attention in parietal cortex. *Annu. Rev. Neurosci.* 22, 319–349. doi: 10.1146/annurev.neuro.22.1.319

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* 12, 2493–2537.

Cox, D. D., and Dean, T. (2014). Neural networks and neuroscience-inspired computer vision. *Curr. Biol.* 24, R921–R929. doi: 10.1016/j.cub.2014.08.026

Deco, G., Jirsa, V. K., and McIntosh, A. R. (2013). Resting brains never rest: computational insights into potential cognitive architectures. *Trends Neurosci.* 36, 268–274. doi: 10.1016/j.tins.2013.03.001

De Filippo De Grazia, M., Cutini, S., Lisi, M., and Zorzi, M. (2012). Space coding for sensorimotor transformations can emerge through unsupervised learning. *Cogn. Process.* 13, 141–146. doi: 10.1007/s10339-012-0478-4

De Meyer, K., and Spratling, M. W. (2011). Multiplicative gain modulation arises through unsupervised learning in a predictive coding model of cortical function. *Neural Comput.* 23, 1536–1567. doi: 10.1162/NECO_a_00130

Demuth, H., and Beale, M. (1993). *Neural Network Toolbox for Use with MATLAB*. Natick, MA: The MathWorks, Inc.

Di Bono, M. G., and Zorzi, M. (2013). Deep generative learning of location-invariant visual word recognition. *Front. Psychol.* 4:635. doi: 10.3389/fpsyg.2013.00635

Duhamel, J. R., Colby, C. L., and Goldberg, M. E. (1992). The updating of the representation of visual space in parietal cortex by intended eye movements. *Science* 255, 90–92. doi: 10.1126/science.1553535

Duhamel, J. R., Bremmer, F., Ben Hamed, S., and Graf, W. (1997). Spatial invariance of visual receptive fields in parietal cortex neurons. *Nature* 389, 845–848. doi: 10.1038/39865

Fiser, J., Berkes, P., Orbán, G., and Lengyel, M. (2010). Statistically optimal perception and learning: from behavior to neural representations. *Trends Cogn. Sci.* 14, 119–130. doi: 10.1016/j.tics.2010.01.003

Friston, K. (2010). The free-energy principle: a unified brain theory? *Nat. Rev. Neurosci.* 11, 127–138. doi: 10.1038/nrn2787

Froudarakis, E., Berens, P., Ecker, A. S., Cotton, R. J., Sinz, F. H., Yatsenko, D., et al. (2014). Population code in mouse V1 facilitates readout of natural scenes through increased sparseness. *Nat. Neurosci.* 17, 851–857. doi: 10.1038/nn.3707

Gilbert, C. D., and Sigman, M. (2007). Brain states: top-down influences in sensory processing. *Neuron* 54, 677–696. doi: 10.1016/j.neuron.2007.05.019

Graziano, M. S., Hu, X. T., and Gross, C. G. (1997). Visuospatial properties of ventral premotor cortex. *J. Neurophysiol.* 77, 2268–2292.

Güçlü, U., and van Gerven, M. A. J. (2015). Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. *J. Neurosci.* 35, 10005–10014. doi: 10.1523/JNEUROSCI.5023-14.2015

Güçlü, U., and van Gerven, M. A. J. (2014). Unsupervised feature learning improves prediction of human brain activity in response to natural images. *PLoS Comput. Biol.* 10:e1003724. doi: 10.1371/journal.pcbi.1003724

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Comput.* 14, 1771–1800. doi: 10.1162/089976602760128018

Hinton, G. E. (2007). Learning multiple layers of representation. *Trends Cogn. Sci.* 11, 428–34. doi: 10.1016/j.tics.2007.09.004

Hinton, G. E. (2010). *A Practical Guide to Training Restricted Boltzmann Machines*. Tech. Rep. UTML TR 2010-003, Univ. Toronto 9, 1.

Hinton, G. E. (2013). Where do features come from?. *Cogn. Sci.* 38, 1–24. doi: 10.1111/cogs.12049

Hinton, G. E., and Ghahramani, Z. (1997). Generative models for discovering sparse distributed representations. *Philos. Trans. R. Soc. Lond. B. Biol. Sci.* 352, 1177–1190. doi: 10.1098/rstb.1997.0101

Hinton, G. E., and Salakhutdinov, R. (2006). Reducing the dimensionality of data with neural networks. *Science* 313, 504–507. doi: 10.1126/science.1127647

Hinton, G. E., and Sejnowski, T. J. (1999). *Unsupervised Learning: Foundations of Neural Computation.* Cambridge, MA: MIT Press. Available at: http://books.google.com/books?hl=it&lr=&id=yj04Y0lje4cC&pgis=1 [Accessed July 4, 2012].

Kastner, S., and Ungerleider, L. G. (2000). Mechanisms of visual attention in the human cortex. *Annu. Rev. Neurosci.* 23, 315–341. doi: 10.1146/annurev.neuro.23.1.315

Kelso, J. A. (2012). Multistability and metastability: understanding dynamic coordination in the brain. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* 367, 906–918. doi: 10.1098/rstb.2011.0351

Khaligh-Razavi, S. M., and Kriegeskorte, N. (2014). Deep Supervised, but not unsupervised, models may explain IT cortical representation. *PLoS Comput. Biol.* 10:e1003915. doi: 10.1371/journal.pcbi.1003915

Kirkpatrick, S., Gelatt, C. D. Jr., and Vecchi, M. P. (1983). Optimization by simmulated annealing. *Science* 220, 671–680. doi: 10.1126/science.220.4598.671

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* 24, 609–616.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature* 521, 436–444. doi: 10.1038/nature14539

Lee, H., Ekanadham, C., and Ng, A. Y. (2008). Sparse deep belief net models for visual area V2. *Adv. Neural Inf. Process. Syst*. 20, 873–880.

Lee, T. S., and Mumford, D. (2003). Hierarchical Bayesian inference in the visual cortex. *J. Opt. Soc. Am. A* 20:1434. doi: 10.1364/josaa.20.001434

Ma, W. J., Beck, J. M., Latham, P. E., and Pouget, A. (2006). Bayesian inference with probabilistic population codes. *Nat. Neurosci.* 9, 1432–1438. doi: 10.1038/nn1790

Mazzoni, P., Andersen, R. A., and Jordan, M. I. (1991). A more biologically plausible learning rule for neural networks. *Proc. Natl. Acad. Sci. U.S. A.* 88:4433. doi: 10.1073/pnas.88.10.4433

McClelland, J. L. (2009). The place of modeling in cognitive science. *Top. Cogn. Sci.* 1, 11–38. doi: 10.1111/j.1756-8765.2008.01003.x

McClelland, J. L., Botvinick, M. M., Noelle, D. C., Plaut, D. C., Rogers, T. T., Seidenberg, M. S., et al. (2010). Letting structure emerge: connectionist and dynamical systems approaches to cognition. *Trends Cogn. Sci.* 14, 348–356. doi: 10.1016/j.tics.2010.06.002

Møller, M. F. (1993). A scaled conjugate gradient algorithm for fast supervised learning. *Neural Netw.* 6, 525–533. doi: 10.1016/S0893-6080(05)80056-5

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., et al. (2015). Human-level control through deep reinforcement learning. *Nature* 518, 529–533. doi: 10.1038/nature14236

Mohamed, A., Dahl, G. E., and Hinton, G. E. (2012). Acoustic modeling using deep belief networks. *IEEE Trans. Audio. Speech. Lang. Proces.* 20, 14–22. doi: 10.1109/TASL.2011.2109382

Nessler, B., Pfeiffer, M., Buesing, L., and Maass, W. (2013). Bayesian computation emerges in generic cortical microcircuits through spike-timing-dependent plasticity. *PLoS Comput. Biol.* 9:e1003037. doi: 10.1371/journal.pcbi.1003037

O'Reilly, R. C. (1998). Six principles for biologically based computational models of cortical cognition. *Trends Cogn. Sci.* 2, 455–462. doi: 10.1016/S1364-6613(98)01241-8

Pouget, A., Deneve, S., and Duhamel, J.-R. (2002). A computational perspective on the neural basis of multisensory spatial representations. *Nat. Rev. Neurosci.* 3, 741–747. doi: 10.1038/nrn914

Pouget, A., and Sejnowski, T. J. (1997). Spatial transformations in the parietal cortex using basis functions. *J. Cogn. Neurosci.* 9, 222–237. doi: 10.1162/jocn.1997.9.2.222

Pouget, A., and Snyder, L. H. (2000). Computational approaches to sensorimotor transformations. *Nat. Neurosci.* 3, 1192–1198. doi: 10.1038/81469

Raichle, M. E. (2015). The restless brain: how intrinsic activity organizes brain function. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* 370, 20140172–20140172. doi: 10.1098/rstb.2014.0172

Ranzato, M. A., Boureau, L., Chopra, S., and LeCun, Y., (2007). "A unified energy-based framework for 913 unsupervised learning," in *Proceedings Conference on AI*. Available online at: 914 http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:A+Unified+Energy-Based+Framework+for+Unsupervised+Learning#0 [Accessed July 16, 2014].

Reichert, D. P., Seriès, P., and Storkey, A. J. (2013). Charles Bonnet syndrome: evidence for a generative model in the cortex? *PLoS Comput. Biol.* 9:e1003134. doi: 10.1371/journal.pcbi.1003134

Riedmiller, M., and Braun, H. (1993). "A direct adaptive method for faster backpropagation learning: The RPROP algorithm," in *IEEE International Conference on Neural Networks* (San Francisco, CA), 586–591.

Rolls, E. T., and Tovee, M. J. (1995). Sparseness of the neuronal representation of stimuli in the primate temporal visual cortex. *J. Neurophysiol.* 73, 713–26.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature* 323, 533–536. doi: 10.1038/323533a0

Sakata, H., Taira, M., Murata, A., and Mine, S. (1995). Neural mechanisms of visual guidance of hand action in the parietal cortex of the monkey. *Cereb. Cortex* 5, 429–438. doi: 10.1093/cercor/5.5.429

Salakhutdinov, R. (2015). Learning deep generative models. *Annu. Rev. Stat. Appl.* 2, 361–385. doi: 10.1146/annurev-statistics-010814-020120

Salinas, E., and Thier, P. (2000). Gain modulation: a major computational principle of the central nervous system. *Neuron* 27, 15–21. doi: 10.1016/S0896-6273(00)00004-0

Sillito, A. M., Cudeiro, J., and Jones, H. E. (2006). Always returning: feedback and sensory processing in visual cortex and thalamus. *Trends Neurosci.* 29, 307–316. doi: 10.1016/j.tins.2006.05.001

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., et al. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature* 529, 484–489. doi: 10.1038/nature16961

Snyder, L. H., Grieve, K. L., Brotchie, P., and Andersen, R. A. (1998). Separate body- and world-referenced representations of visual space in parietal cortex. *Nature* 394, 887–891.

Stoianov, I., and Zorzi, M. (2012). Emergence of a "visual number sense" in hierarchical generative models. *Nat. Neurosci.* 15, 194–196. doi: 10.1038/nn.2996

Stricanne, B., Andersen, R. A., and Mazzoni, P. (1996). Eye-centered, head-centered, and intermediate coding of remembered sound locations in area LIP. *J. Neurophysiol.* 76, 2071–2076.

Testolin, A., Stoianov, I., De Filippo De Grazia, M., and Zorzi, M. (2013). Deep unsupervised learning on a desktop PC : a primer for cognitive scientists. *Front. Psychol.* 4:251. doi: 10.3389/fpsyg.2013.00251

Testolin, A., Stoianov, I., Sperduti, A., and Zorzi, M. (2016). Learning orthographic structure with sequential generative neural networks. *Cogn. Sci.* 40, 579–606. doi: 10.1111/cogs.12258

Testolin, A., and Zorzi, M. (2016). Probabilistic models and generative neural networks: towards an unified framework for modeling normal and impaired neurocognitive functions. *Front. Comput. Neurosci.* 10:73. doi: 10.3389/fncom.2016.00073

Thorpe, S. J., and Imbert, M. (1989). Biological constraints on connectionist modelling. *Connect. Perspect.* 1, 1–36.

Treves, A., and Rolls, E. T. (1991). What determines the capacity of autoassociative memories in the brain? *Netw. Comput. Neural Syst.* 2, 371–397. doi: 10.1088/0954-898X_2_4_004

Vinje, W. E., and Gallant, J. L. (2000). Sparse Coding and decorrelation in primary visual cortex during natural vision. *Science* 287, 1273–1276. doi: 10.1126/science.287.5456.1273

Widrow, B., and Hoff, M. (1960). "Adaptive Switching Circuits," in *IRE WESCON Convention Record*, 96–140. Available online at: http://www-isl.stanford.edu/people/widrow/papers/c1960adaptiveswitching.pdf [Accessed November 29, 2014].

Xing, J., and Andersen, R. A., (2000). Models of the posterior parietal cortex which perform multimodal integration and represent space in several coordinate frames. *J. Cogn. Neurosci.* 12, 601–614. doi: 10.1162/089892900562363

Zipser, D., and Andersen, R. (1988). A back-propagation programmed network that simulates response properties of a subset of posterior parietal neurons. *Nature* 331, 679–684. doi: 10.1038/331679a0

Zorzi, M., Testolin, A., and Stoianov, I. (2013). Modeling language and cognition with deep unsupervised learning: a tutorial overview. *Front. Psychol.* 4:515. doi: 10.3389/fpsyg.2013.00515

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Computational Neuroscience

ORIGINAL RESEARCH
published: 31 January 2017
doi: 10.3389/fncom.2017.00004

Hierarchical Neural Representation of Dreamed Objects Revealed by Brain Decoding with Deep Neural Network Features

Tomoyasu Horikawa[1] *and Yukiyasu Kamitani*[1,2]*

[1] Computational Neuroscience Laboratories, Advanced Telecommunications Research Institute (ATR), Kyoto, Japan,
[2] Graduate School of Informatics, Kyoto University, Kyoto, Japan

Dreaming is generally thought to be generated by spontaneous brain activity during sleep with patterns common to waking experience. This view is supported by a recent study demonstrating that dreamed objects can be predicted from brain activity during sleep using statistical decoders trained with stimulus-induced brain activity. However, it remains unclear whether and how visual image features associated with dreamed objects are represented in the brain. In this study, we used a deep neural network (DNN) model for object recognition as a proxy for hierarchical visual feature representation, and DNN features for dreamed objects were analyzed with brain decoding of fMRI data collected during dreaming. The decoders were first trained with stimulus-induced brain activity labeled with the feature values of the stimulus image from multiple DNN layers. The decoders were then used to decode DNN features from the dream fMRI data, and the decoded features were compared with the averaged features of each object category calculated from a large-scale image database. We found that the feature values decoded from the dream fMRI data positively correlated with those associated with dreamed object categories at mid- to high-level DNN layers. Using the decoded features, the dreamed object category could be identified at above-chance levels by matching them to the averaged features for candidate categories. The results suggest that dreaming recruits hierarchical visual feature representations associated with objects, which may support phenomenal aspects of dream experience.

Keywords: dream, brain decoding, deep neural networks, hierarchical neural representations, functional magnetic resonance imaging

OPEN ACCESS

Edited by:
Marcel Van Gerven,
Radboud University Nijmegen,
Netherlands

Reviewed by:
Jan Lauwereyns,
Kyushu University, Japan
Piotr Wojewnik,
Kyushu University, Japan

***Correspondence:**
Yukiyasu Kamitani
kamitani@i.kyoto-u.ac.jp

Received: *29 November 2016*
Accepted: *16 January 2017*
Published: *31 January 2017*

Citation:
Horikawa T and Kamitani Y (2017) Hierarchical Neural Representation of Dreamed Objects Revealed by Brain Decoding with Deep Neural Network Features. Front. Comput. Neurosci. 11:4. doi: 10.3389/fncom.2017.00004

INTRODUCTION

Dreaming during sleep is a universal human experience and one that is often accompanied by highly realistic visual scenes spontaneously generated by the brain. The most striking characteristic of visual dreaming is its similarity to the visual experience during waking hours, and dreaming generally incorporates features that are typical of the waking experience, such as shapes, objects, and scenes. These phenomenological similarities are considered to be underlain by neural substrates common to both the awake and sleep states, and a number of studies have sought to address the neural commonalities and differences of these contrasting states by analyses of regional brain

activations (Maquet et al., 1996; Braun et al., 1997, 1998; Maquet, 2000; Hong et al., 2009; Dresler et al., 2011), and brain activity patterns for specific visual contents (Horikawa et al., 2013).

A previous work investigated the commonality of neural representations of visual objects and scenes between perception and dreaming, and demonstrated that the dreamed objects/scenes could be predicted from brain activity patterns during sleep using statistical decoders trained to predict viewed object/scene categories (Horikawa et al., 2013). In this study, the authors used decoders trained to predict categorical labels of viewed objects and scenes, the labels of which were constructed from subjects' dream reports. They thereby demonstrated decoding of dream contents from brain activity patterns during sleep using stimulus-trained decoders. The decoders trained on brain activity patterns in higher visual cortex showed higher accuracy than those trained on brain activity patterns in lower visual cortex. Their results suggest that visual dream contents are represented by discriminative brain activity patterns similar to perception at least in higher visual areas.

While this study demonstrated accurate decoding of categorical information on dreamed objects from higher visual areas, it still remains unclear whether or how multiple levels of hierarchical visual features associated with dreamed objects are represented in the brain. Because brain decoding through multi-voxel pattern classification algorithms often obscures what made the labeled brain activity patterns discriminable, it is not clear what levels of visual information, including multiple levels of hierarchical visual features and semantics, enabled the successful decoding.

Several recent studies have addressed this issue by using explicit models of visual features, and investigated neural representations of visual contents (Kay et al., 2008; Khaligh-Razavi and Kriegeskorte, 2014; Horikawa and Kamitani, 2015; Naselaris et al., 2015; Jozwik et al., 2016). These studies used multiple levels of visual features, including Gabor filters and features extracted from hierarchical models, to represent visual images by patterns of visual features. They thereby established links between brain activity patterns and visual features or modeled the representational space of brain activity patterns using visual features to address how each visual feature is used to represent seen or imagined visual images.

Among a large number of visual features, hierarchical visual features, such as those from deep neural networks (DNN) (Khaligh-Razavi and Kriegeskorte, 2014; Horikawa and Kamitani, 2015), would be especially suited to represent objects: They are hierarchical in the sense that higher-level features are composed of the outputs from the previous lower-level features. The reason for their suitability is that those visual features achieve varying levels of invariance to image differences through hierarchical processing, including differences in rotation, position, scale, and other attributes, which are often observed in images even within the same object categories, and acquire robust object-category-specific representations.

Our previous study (Horikawa and Kamitani, 2015) investigated neural representations of hierarchical visual features associated with seen and imagined objects by representing objects using patterns of visual features. In this study, we asked subjects to imagine visual images of presented object names and analyzed imagery-induced brain activity in combination with hierarchical visual features to observe how hierarchical visual feature representations are used during mental imagery. We first used the visual features derived from various computational models to represent an object by a vector of visual features, and then trained statistical regression models to decode feature vectors of viewed objects from brain activity patterns measured as subjects viewed images of objects (stimulus-trained decoder). The trained decoders were then used to decode visual features of seen and imagined objects, and the decoded feature vectors were used to identify the object categories. Our analyses showed that the stimulus-trained decoders better predicted the low/high-level visual features of seen objects from lower/higher visual areas respectively, showing a homology between the brain and DNN. This provided empirical support for the idea that the DNN can be a good proxy for the hierarchical visual system for object recognition. We further demonstrated high decoding accuracy for mid- to high-level features of imagined objects from relatively higher visual areas, suggesting the recruitment of feature-level representations during mental imagery, in particular for the mid- to high-level feature representation. Therefore, the same strategy would also be applicable to investigate hierarchical feature representations of dreamed objects, and may reveal the recruitment of feature-level representations associated with dreamed objects at least for mid- to high-level feature representations as the previous study demonstrated for volitional mental imagery (Horikawa and Kamitani, 2015).

Here, we investigated whether multiple levels of hierarchical visual features associated with dreamed objects are represented in the brain in a manner similar to perception. For this purpose, we applied the same strategy in Horikawa and Kamitani (2015) to the decoding of hierarchical visual features associated with dreamed objects from brain activity patterns during sleep. We used a deep convolutional neural network (DNN) for object recognition as a proxy for hierarchical visual feature representation. We represented images of objects using patterns of visual features derived from DNN models (**Figures 1A,B**). We then performed decoding analyses of DNN features associated with dreamed objects from brain activity patterns obtained from sleeping subjects. We used the decoders trained to decode visual features of seen objects, thereby testing whether visual dream contents are represented by the hierarchical feature representations elicited in visual perception. We also tested whether the decoded feature vector could be used to identify the reported dreamed object by matching it to the averaged feature vectors of images in multiple candidate categories. The results were then compared with those for seen and imagined objects (Horikawa and Kamitani, 2015) to see the differences in hierarchical representation and the ability to decode arbitrary objects beyond training categories ("generic object decoding").

MATERIALS AND METHODS

The data used for this study came from two previous studies performed at our laboratory (Horikawa et al., 2013; Horikawa

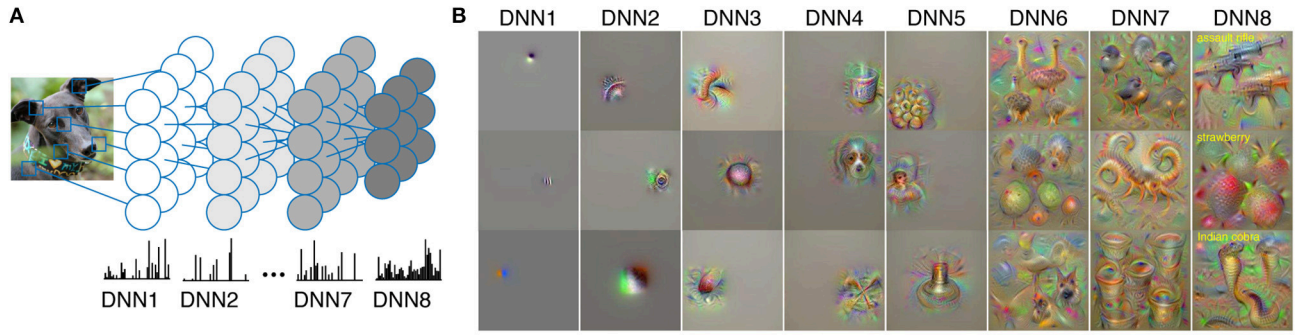


FIGURE 1 | Deep neural network features. (A) Representing an input image by visual feature patterns derived by DNN. The DNN was used to extract feature values of individual units in each layer. **(B)** Preferred images for each DNN layer. The images that highly activate each DNN unit were generated using activation maximization methods (see Materials and Methods: "Synthesis of Preferred Images Using Activation Maximization" for details).

and Kamitani, 2015). In these studies, two subjects (Subjects 1 and 2 in this study) participated in both of the two studies as Participants 1 and 3 in Horikawa et al. (2013) and as Subjects 1 and 2 in Horikawa and Kamitani (2015). Here, we provide a brief description of the subjects, datasets, and preprocessing of the MRI data for the main experiments. For full details, see Horikawa et al. (2013) and Horikawa and Kamitani (2015).

Subjects

Two healthy subjects (males, aged 27 and 42) with normal or corrected-to-normal vision participated in the experiments. Both subjects had considerable experience of participating in fMRI experiments, and were highly trained. Both subjects provided written informed consent for their participation in the experiments, in accordance with the Declaration of Helsinki, and the study protocol was approved by the Ethics Committee of ATR. The experimental of each subject were collected over multiple scanning sessions spanning over 2 years.

Dataset from Horikawa et al. (2013; "Dream" Dataset)

We used an fMRI dataset from the sleep experiments conducted in a previous dream decoding study (Horikawa et al., 2013). This "dream" dataset was used for testing decoding models trained on part of the dataset from Horikawa and Kamitani (2015). A brief description of the dataset is given in the following paragraph (see Horikawa et al., 2013 for all experimental details).

fMRI signals were measured with simultaneous recording of electroencephalography (EEG) while subjects slept in an MRI scanner. They were awakened when a characteristic EEG signature was detected during sleep-onset periods (non-rapid eye movement [NREM] periods) and were then asked to provide a verbal report, freely describing their visual experience (NREM dream) before awakening. This procedure was repeated until at least 200 awakenings associated with a visual report were collected for each subject. From the collected reports, words describing visual objects or scenes were manually extracted and mapped to *WordNet*, a lexical database in which semantically similar words are grouped together as *synsets* (categories), in a hierarchical structure (Fellbaum, 1998). Using the semantic hierarchy, extracted visual words were grouped into *base synsets* that appeared in at least 10 reports from each subject (26 and 16 synsets for Subjects 1 and 2, respectively). The fMRI data obtained before each awakening were labeled with a *visual content vector*, each element of which indicated the presence or absence of a base synset in the subsequent report.

Note that these fMRI data were collected during sleep-onset periods (sleep stage 1 or 2) rather than rapid-eye movement (REM) periods. Although REM sleep and its underlying neurophysiological mechanisms were originally believed to be indispensable for dreaming, there has been accumulating evidence that dreaming is dissociable from REM sleep and can be experienced during NREM sleep periods (Nir and Tononi, 2009).

In addition to the fMRI data, we used visual images presented in the perception experiment described in Horikawa et al. (2013) to construct *category features* (see Materials and Methods: "Category feature vector") for dreamed objects (216 and 240 images for each category for Subjects 1 and 2, respectively).

Datasets from Horikawa and Kamitani (2015; "Training," "Perception," and "Imagery" Datasets)

We used fMRI data from the perception experiment and the imagery experiment conducted in Horikawa and Kamitani (2015). The perception experiment had two sessions: a training image session and a testing image session. Data from the training image session of the perception experiment were used to train decoding models in this study ("training" dataset), which were then tested on the dream dataset from Horikawa et al. (2013). For comparison with the results from the dream dataset, the data from the perception experiment (the testing image session) and the imagery experiment in Horikawa et al. (2013) were also used as test datasets in this study ("perception" and "imagery" datasets). A description of the datasets is given in the following paragraph (see Horikawa and Kamitani, 2015 for all experimental details).

In the perception experiment of Horikawa et al. (2013), stimulus-induced fMRI signals were collected from two distinct

sessions: a training image session and a testing image session, consisting of 24 and 35 separate runs respectively. Each run contained 55 stimulus blocks consisting of 50 blocks with different images and five randomly interspersed repetition blocks where the same image as in the previous block was presented. In each stimulus block an image (image size, $12 \times 12^{\circ}$C) was flashed at 2 Hz for 9 s. Images were presented on the center of the display with a central fixation spot. To indicate the onset of the block, the color of the fixation spot changed for 0.5 s before each stimulus block began. Subjects maintained steady fixation throughout each run, and performed a one-back repetition detection task on the images to maintain their attention on the presented images, responding with a button press for each repetition. In the training image session, a total of 1200 images from 150 different object categories (eight images per each category) were each presented only once. In the testing image session, a total of 50 images from 50 object categories (one image from each category) were presented 35 times (blocks) each. Note that the categories in the testing image session were not used in the training image session. The presentation order of the categories was randomized across runs.

In the imagery experiment of Horikawa et al. (2013), the subjects were required to visually imagine images from one of the 50 object categories used in the testing image session of the perception experiment. The imagery experiment consisted of 20 separate runs, with each run containing 25 imagery blocks. Each imagery block consisted of a 3-s cue period, a 15-s imagery period, a 3-s evaluation period, and a 3-s rest period. During the rest periods, a fixation spot was presented in the center of the display. From 0.8 s before each cue period began, the color of the fixation spot changed for 0.5 s to indicate the onset of the blocks. During the cue period, words describing the names of the 50 categories presented in the testing image session of the perception experiment were visually presented around the center of the display (one target and 49 distractors). The word of the category to be imagined was presented with a red color (target), while the other words were presented in black (distractors). The onset and end of the imagery periods were signaled by beep sounds. Subjects were required to start imagining as many object images pertaining to the category described by the red word as possible. Their eyes were closed from the first beep sound to the second beep sound. After the second beep sound, the word of the target category was presented at the center of the display to allow the subjects to evaluate the vividness of their mental imagery on a five-point scale (very vivid, fairly vivid, rather vivid, not vivid, cannot recognize, or forget the target) by a button press. The 25 categories in each run were pseudo-randomly selected from 50 categories such that the two consecutive runs contained all the 50 categories.

fMRI Data Preprocessing

The first 9 s of scans from each run were discarded to remove instability effects of the MRI scanner. The acquired fMRI data were subjected to three-dimensional motion correction using SPM5 (http://www.fil.ion.ucl.ac.uk/spm). The data were then coregistered to the within-session high-resolution anatomical image of the same slices used for EPI, and then subsequently to a whole-head high-resolution anatomical image common across the two studies. The coregistered data were then reinterpolated to $3 \times 3 \times 3$ mm voxels.

For the dream data from Horikawa et al. (2013), we created data samples by first regressing out nuisance parameters, including a linear trend, and temporal components proportional to six motion parameters from the SPM5 motion correction procedure, from each voxel amplitude for each run, and the data were then *despiked* to reduce extreme values (beyond $\pm$ 3SD for each run). After that, voxel amplitudes around awakening were normalized relative to the mean amplitude during the period 60–90 s prior to each awakening. This period was used as the baseline, as it tended to show relatively stable blood oxygenation level dependent (BOLD) signals over time. The voxel values averaged across the three volumes (9 s) immediately before awakening served as a single data sample (the time window was shifted for time course analysis).

For the perception and imagery data from Horikawa and Kamitani (2015), we created data samples by first regressing out nuisance parameters, including a constant baseline, a linear trend, and temporal components proportional to six motion parameters from the SPM5 motion correction procedure, from each voxel amplitude for each run, and the data were then *despiked* to reduce extreme values (beyond $\pm$ 3SD for each run). The voxel amplitudes were then averaged within each 9-s stimulus block (three volumes) or 15-s imagery block (five volumes), after shifting the data by 3 s (one volume) to compensate for hemodynamic delays.

For testing decoding models with the dream dataset, the trials in which the last 15-s epoch before awakening was classified as *wake* were not used for the following analyses, and those classified as sleep stage 1 or 2 were used. We analyzed the dream fMRI data in two ways: single category-based analysis with averaged trials, and multiple category-based analysis with individual trials. In the single category-based analysis, fMRI samples were further averaged for the dream trials containing the same category while disregarding the other reported categories. Thus, one data sample is labeled only by a single category. This preprocessing yielded 26 and 16 averaged fMRI samples for Subjects 1 and 2, respectively (corresponding to the numbers of the base synsets). In the multiple category-based analysis, individual fMRI samples were labeled by multiple reported categories at each awakening.

For testing with the perception and imagery datasets, the blocks of the same category were averaged (35 and 10 blocks averaged for perception and imagery, respectively). This procedure yielded 50 averaged fMRI samples (corresponding to the 50 test categories) for each of the perception and imagery datasets in each subject.

Region of Interest (ROI) Selection

V1, V2, V3, and V4 were delineated by a standard retinotopy experiment (Engel et al., 1994; Sereno et al., 1995). The lateral occipital complex (LOC), the fusiform face area (FFA), and the parahippocampal place area (PPA) were identified using conventional functional localizers (Kanwisher et al., 1997; Epstein and Kanwisher, 1998; Kourtzi and Kanwisher, 2000). For the analysis of individual visual areas, the following numbers

of voxels were identified for V1, V2, V3, V4, LOC, FFA, and PPA, respectively: 1054, 1079, 786, 763, 570, 614, and 369 voxels for Subject 1; 772, 958, 824, 545, 847, 438, and 317 voxels for Subject 2. A continuous region covering LOC, FFA, and PPA was manually delineated on the flattened cortical surfaces, and the region was defined as the *higher visual cortex* (HVC). Voxels from V1–V4 and HVC were combined to define the *visual cortex* (VC; 4794 and 4499 voxels for Subject 1 and 2, respectively). For full details on the experiments for localizing the regions of interest, see Horikawa et al. (2013) and Horikawa and Kamitani (2015).

Visual Features Derived from Deep Convolutional Neural Network

Using the deep convolutional neural network (DNN) proposed in a previous study (Krizhevsky et al., 2012), we computed visual features from the images used in the fMRI experiments, and also from images from an online image database[1] where images are grouped according to the hierarchy in *WordNet* (Fellbaum, 1998). We used the *MatConvNet* implementation of DNN[2], which was trained with images in ImageNet to classify 1000 object categories. The DNN consisted of five convolutional layers (DNN1–5) and three fully connected layers (DNN6–8), with some of these layers containing a huge number of feature units (e.g., 290,400 units in DNN1). We randomly selected 1000 units in each of the layers from one to seven to reduce the computational load while making sure that the selection was unbiased, and used all 1000 units in the eighth layer. We represented each image by a vector of those units' outputs.

Category Feature Vector

We constructed *category feature vectors* to represent object categories using visual features in each DNN layers. We first computed visual feature vectors for all images of categories in the *ImageNet* database (50 test categories for Horikawa and Kamitani (2015), and 15,314 candidate categories; Deng et al., 2009) and for images used in the perception test experiment in Horikawa et al. (2013) (26 and 16 dreamed categories for Subjects 1 and 2, respectively). Using the computed feature vectors, category feature vectors were constructed for all categories by averaging the feature vectors of images belonging to the same category. These procedures were conducted for each DNN layer to construct feature representations of individual object categories (*single-category feature vectors*). In addition to that, we also constructed *multi-category feature vectors* to represent multiple object categories reported at each awakening in the dream dataset using features in each DNN layer. The multi-category feature vectors were constructed by averaging multiple single-category feature vectors annotated by reported categories at each awakening.

Synthesis of Preferred Images Using Activation Maximization

We used the activation maximization method to generate preferred images for individual units in each layer of the DNN model (Simonyan et al., 2014; Yosinski et al., 2015; Mahendran and Vedaldi, 2016; Nguyen et al., 2016). Synthesis of preferred images starts from a random image and optimizes the input image to maximally activate a target DNN unit by iteratively calculating how the image should be changed via backpropagation. This analysis was implemented using custom software written in MATLAB based on the original Python code provided in blog posts[3].

Visual Feature Decoding Analysis

We constructed multivoxel decoders to predict a visual feature vector of a seen object from fMRI activities in multiple ROIs in the training dataset (Horikawa and Kamitani, 2015) using a set of linear regression models. In this study, we used the sparse linear regression algorithm (SLR; Bishop, 2006), which can automatically select important voxels for decoding, by introducing sparsity into weight estimation through Bayesian parameters estimation with the automatic relevance determination (ARD) prior (see Horikawa and Kamitani, 2015 for detailed descriptions). The decoders were trained to predict the values of individual elements in the feature vector (consisting of 1000 randomly selected units for DNN1–7 and all 1000 units for DNN8) using the training dataset (1200 samples from the perception experiment).

Decoding accuracy was evaluated by the correlation coefficient between the category feature and decoded feature values of each feature unit. The correlation coefficients were pooled across the units and the subjects for each DNN layer and ROI.

These analyses were performed for each combination of DNN layers (DNN1–8) and brain regions of interest (V1, V2, V3, V4, LOC, FFA, and PPA), and the entire visual cortex covering all of the visual subareas listed above (VC). We performed voxel selection prior to the training of the regression model for each feature unit: voxels showing the highest correlation coefficients with the target variable (feature value) in the training data were used (at most 500 voxels for V1, V2, V3, V4, LOC, FFA, and PPA; 1000 voxels for VC). For details of the general procedure of feature decoding, see Horikawa and Kamitani (2015).

Pairwise Identification Analysis

In the pairwise identification analysis, the category of a seen/imagined/dreamed object was identified between true and false categories, using the feature vector decoded from the averaged fMRI activity pattern for each object category. The decoded feature vector was compared with two candidate category feature vectors, one for the true category and the other for a false category selected from the 15,314 candidates. The category with a higher correlation coefficient was selected as the identified category. The analysis was repeated for all combinations of the test categories (50 categories for the perception and imagery datasets; 26 and 16 categories for the dream dataset of Subjects 1 and 2, respectively) and the 15,314 candidate categories. The accuracy for each test category

[1](ImageNet; (Deng et al., 2009); http://www.image-net.org/; 2011 fall release)
[2](http://www.vlfeat.org/matconvnet/)
[3](Mordvintsev, A., Olah, C., Tyka, M., DeepDream—a code example for visualizing Neural Networks, https://github.com/google/deepdream, 2015; Øygard, A.M.,Visualizing GoogLeNet Classes, https://github.com/auduno/deepdraw, 2015)

was evaluated by the ratio of correct identification. This was further averaged across categories and subjects to characterize the accuracies with the dream, perception, and imagery datasets.

Data and Code Availability

The experimental data and codes used in the present study are available from the corresponding author upon request.

RESULTS

We applied the decoders trained with stimulus-induced fMRI signals (stimulus-trained decoders) to dream dataset to test whether multiple levels of visual features associated with dreamed object categories could be decoded from brain activity of sleeping subjects. For this analysis, the dream fMRI samples [three-volume (9 s) averaged fMRI signals immediately before awakening] annotated by each individual object category were averaged across awakenings, and these averaged fMRI samples were used as input to the stimulus-trained decoders (**Figure 2A**). To evaluate the prediction accuracy in each unit, Pearson's correlation coefficient was calculated between the decoded and the single-category feature values for the series of test samples for each subject. The correlation coefficients were then averaged across all feature units obtained from two subjects for each DNN layer. Here, we used the correlation coefficient between the series of the category feature values and the decoded feature values in each unit, instead of the correlation coefficient between the category feature vector and the decoded feature vector for each sample. This was because we constructed decoders for each unit independently, and the baseline amplitude pattern across units alone could lead to spuriously high correlation coefficients between feature vectors.

The correlation coefficients between the features decoded from the dream fMRI dataset and the category features in multiple ROIs are shown in **Figure 2B**. While the absolute values of the correlation coefficients from dream fMRI dataset were lower than those from perception and imagery fMRI datasets (Horikawa and Kamitani, 2015), positive correlation coefficients were observed from decoders trained with relatively higher visual areas at mid- to high-level DNN layers (46 out of 56 pairs of DNN layers and ROIs, one-sided t-test after Fisher's Z transform, uncorrected $p < 0.01$). For most of the DNN layers, the previous study (Horikawa and Kamitani, 2015) showed that the decoding accuracy of seen category features was moderately high from most of visual areas with peak accuracy in V4, whereas the decoding accuracy of imagined category features was high for mid- to higher visual areas. The category feature decoding of dreamed objects showed highest correlation coefficients around the higher visual areas, suggesting the qualitatively similar tendency to the results for imagined rather than seen object categories (Horikawa and Kamitani, 2015).

While we first focused on individual dream categories at a time by averaging fMRI samples at multiple awakenings annotated by the common dreamed object categories (**Figure 2**), we also performed decoding analysis on brain activity patterns from each awakening annotated by multiple dreamed object categories (**Figure 3A**). For this analysis, the same decoders were applied to fMRI samples at each single awakening (three-volume [9 s] averaged fMRI signals immediately before awakening) to obtain decoded features for all awakenings (samples classified as sleep stage 1 or 2). Then, the multi-category features were constructed by averaging the single-category features for multiple object categories reported at subsequent awakenings. The accuracy was evaluated by Pearson's correlation coefficients between decoded feature values and feature values of the multi-category features for the series of test samples. This analysis showed that feature values decoded from brain activity patterns in higher visual areas just before awakening positively correlated with feature values of the multi-category features constructed for object categories reported at subsequent awakening at mid- to high-level DNN layers (**Figure 3B**; 45 out of 56 pairs of DNN layers and ROIs, one-sided t-test after Fisher's Z transform, uncorrected $p < 0.01$). The results suggest that single trial-based fMRI signals contain sufficient information to decode feature-level representations about dreamed object categories while the accuracy was relatively low.

Furthermore, when the time window for the feature decoding analysis was shifted around the time of awakening, the correlation coefficient peaked around 0–10 s before awakening for most of the DNN layers in both of the averaged- and single-trial analyses (**Figure 4**; no correction for hemodynamic delay). This is consistent with the results of the category decoding reported in the previous study (Horikawa et al., 2013). While the high correlations after awakening may be explained by hemodynamic delay and the large time window, the general tendency for the high correlations to be relatively prolonged, especially for higher DNN layers, may reflect feature representations associated with retrieved dream contents during reporting.

Finally, we tested whether the decoded feature vectors can be used to identify dreamed object categories and compared the results with the identification accuracies of seen and imagined object categories reported in a previous study (Horikawa and Kamitani, 2015). We did this by matching the decoded feature vectors and category feature vectors calculated from multiple images of candidate categories in the image database (**Figure 5A**). The pairwise identification accuracy for all combinations of the DNN layers and ROIs are shown in **Figure 5B**. For most combinations of the DNN layers and ROIs, the dreamed object categories can be identified from brain activity patterns with a statistically significant level (43 out of 56 pairs of layers and ROIs, one-sided t-test, uncorrected $p < 0.05$). Additionally, the analysis showed significantly high identification accuracy of dreamed objects from the LOC and FFA for all of the DNN layers (one-sided t-test, uncorrected $p < 0.05$). The pairwise identification accuracies for dreamed, seen, and imagined objects obtained by decoders trained on brain activity pattern in an entire visual cortex are shown in **Figure 5C**. The identification of dreamed objects showed a higher than chance accuracy for most of the DNN layers (one-sided t-test, uncorrected $p < 0.05$, except for DNN7). The identification accuracy for seen and imagined object categories was higher than the chance level for all of the DNN layers (one-sided t-test, uncorrected $p < 0.05$; re-analyzed using the datasets from Horikawa and Kamitani, 2015), with

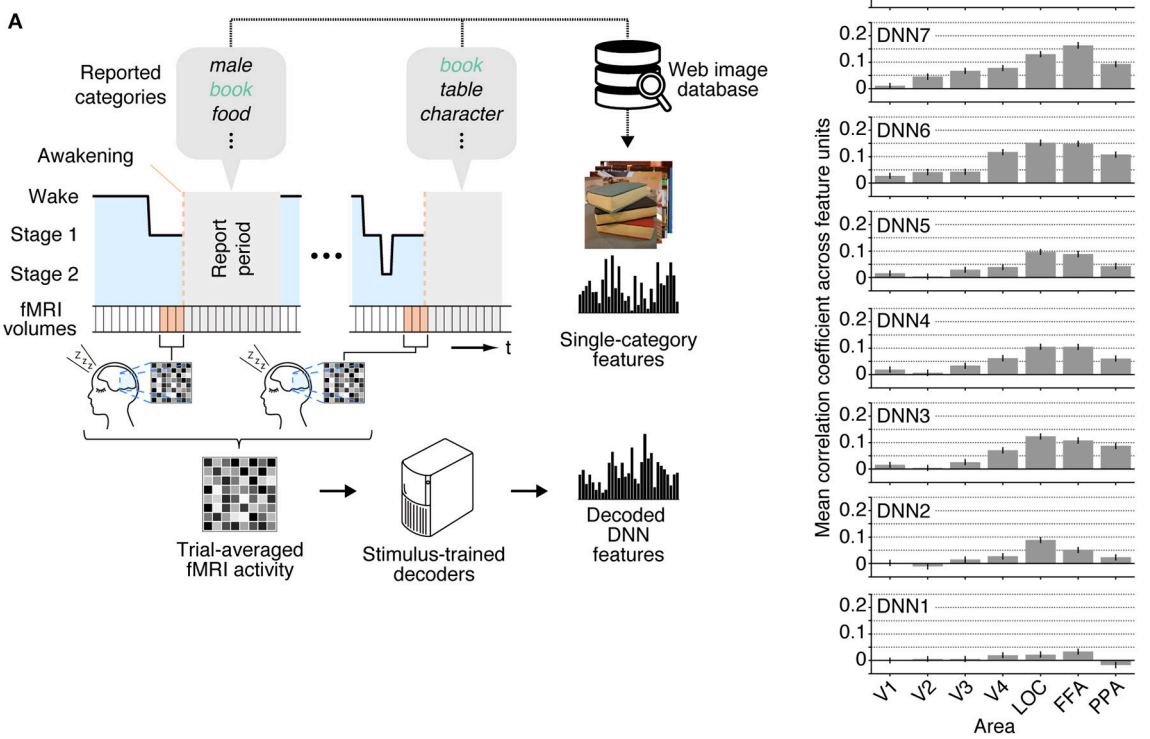


FIGURE 2 | Single category feature decoding with averaged trials. (A) A schematic view of analysis procedure. The decoders trained to decode the DNN feature values calculated from the presented images in the training dataset were applied to an averaged fMRI activity associated with a specific dream category to test decodability of the values of the category features constructed from multiple images annotated by the dream categories. **(B)** Correlation coefficients between the category feature values and the decoded feature values [error bars indicate 95% CI across feature units; two subjects pooled; three-volume (9 s) averaged fMRI signals immediately before awakening].

the highest accuracy shown around the mid-level layers. Similar to the results of the perception and imagery, the identification of the dreamed object categories also showed relatively higher accuracy at mid-level DNN layers around DNN5. However, the accuracy tendency across layers under the dream condition was slightly different from those under the perception and imagery conditions, in the sense that the highest layer, DNN8, also showed higher accuracy, whereas the DNN7 showed poor performance, which may suggest the unique characteristic of dream representations.

DISCUSSION

In this study, we examined whether hierarchical visual feature representations common to perception are recruited to represent dreamed objects in the brain. We used the decoders trained to decode visual features of seen object images and showed that the feature values decoded from brain activity during dreaming positively correlated with feature values associated with dreamed object categories at mid- to high-level DNN layers. This made it possible to discriminate object categories in dreaming at above-chance levels. These results reveal the recruitment of hierarchical visual feature representations shared with perception during dreaming.

In our analyses, we have shown that the multiple-levels of DNN features associated with dreamed objects can be predicted using the stimulus-trained decoders especially from the relatively higher visual areas (**Figures 2B**, **3B**), as in our previous finding with imagined objects (Horikawa and Kamitani, 2015). The present results demonstrated not only semantic or categorical representations but also feature-level representations were recruited during dreaming to represent dreamed objects in a manner similar to perception. While a previous study demonstrated decoding of category information on dreamed objects (Horikawa et al., 2013), it did not clarify whether multiple levels of hierarchical visual feature representations are used to represent dream contents. In contrast to that, the present study demonstrated decoding of hierarchical visual features associated with dreamed object categories (**Figures 2B**, **3B**), especially for features in mid- to high-level DNN layers (see **Figure 1B** for the characteristics of feature units in these layers), providing

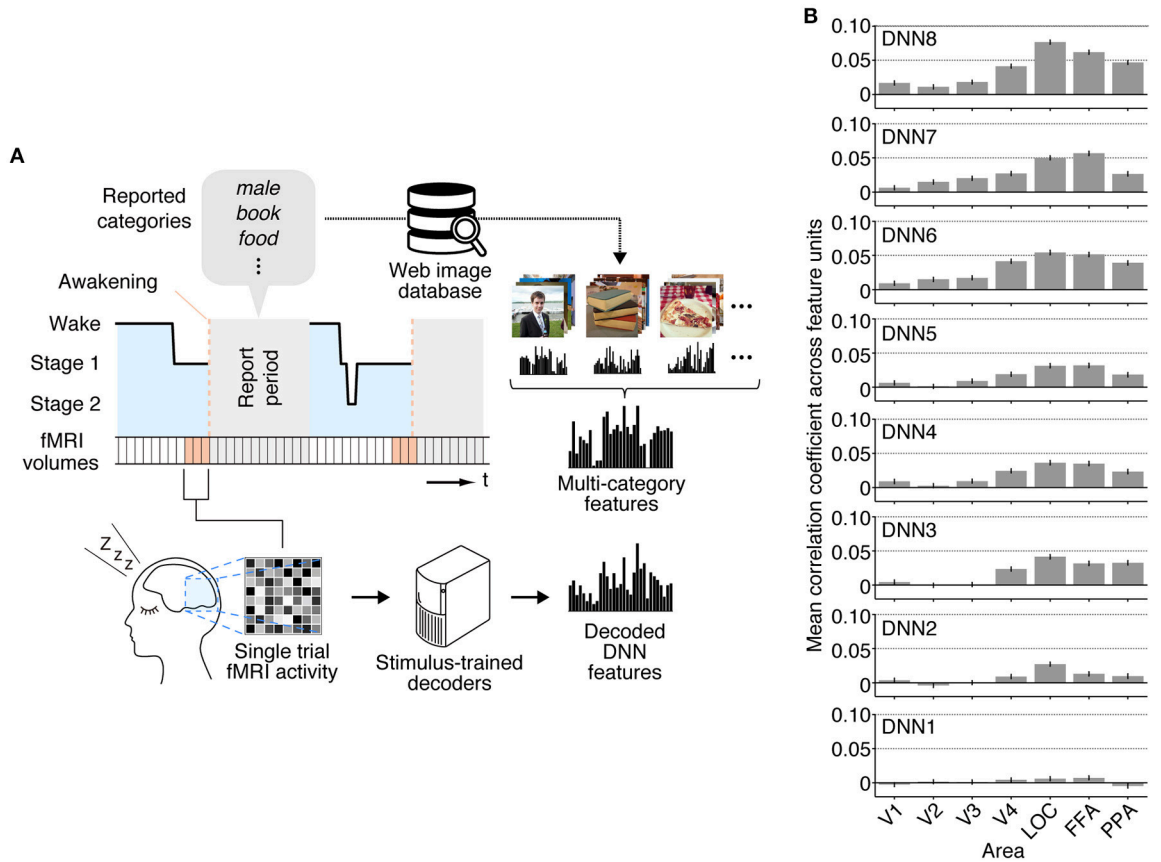


FIGURE 3 | Multi-category feature decoding with individual trials. (A) A schematic view of analysis procedure. The stimulus-trained decoders were applied to fMRI samples at each single awakening (or trial) in the dream fMRI dataset [orange areas, three-volume (9 s) averaged fMRI signals immediately before awakening]. The values of the single-category features for reported dreamed objects were averaged to construct the multi-category features for each awakening. The accuracy was evaluated by Pearson's correlation coefficients between feature values of the decoded and the multi-category features for the series of test samples. **(B)** Correlation coefficients between multi-category features and decoded features (error bars, 95% CI across feature units; two subjects pooled).

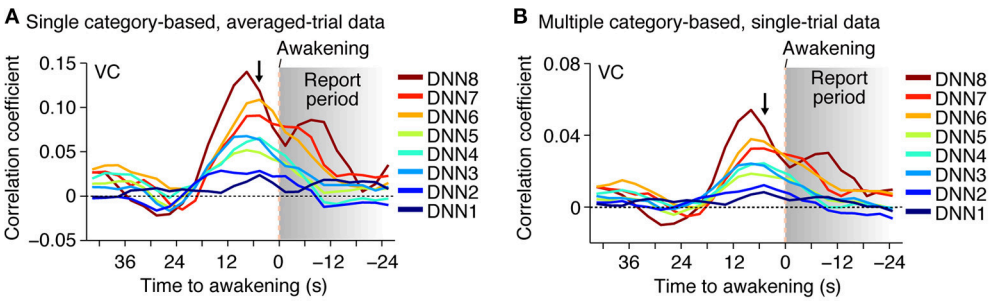


FIGURE 4 | Time course of feature prediction. Correlation coefficients were calculated between the decoded and the category feature values for the series of test samples (two subjects pooled; decoded from VC). The plot shows the mean feature prediction accuracy with the 9-s (three-volume) time window centered at each point (arrowed point for main analyses; cf.). **(A)** Time course of feature prediction with single category-based, averaged-trial data. **(B)** Time course of feature prediction with multiple category-based, single-trial data.

empirical evidence for recruitments of hierarchical visual feature representations during dreaming. These results further our understanding of how dreamed objects are represented in our brain.

Our decoding of feature-level representations of dreamed objects allowed us to discriminate dreamed object categories, although the accuracy is limited, without pre-specifying target categories, and achieved predictions beyond the categories used

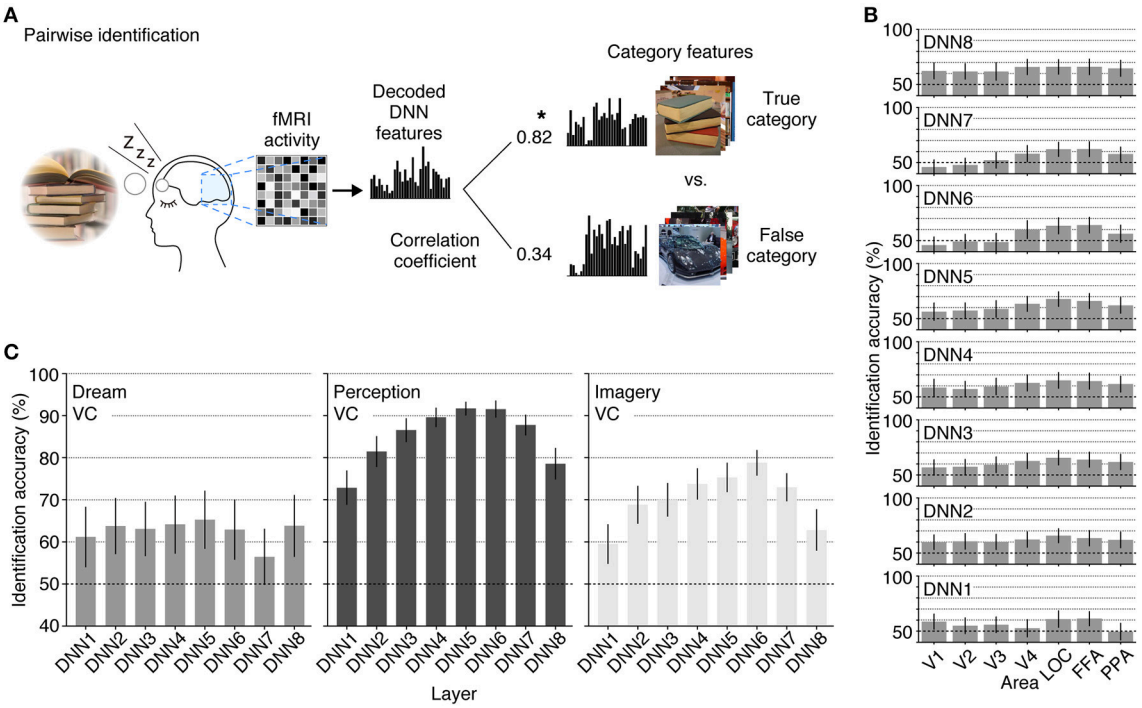


FIGURE 5 | Pairwise identification analysis. (A) A schematic view of the pairwise identification analysis. Given a decoded feature vector, the correlation coefficient was calculated between the decoded feature vector and the category feature vectors of two candidate categories (one true, the other false), and the category with a higher correlation coefficient was selected as the predicted category (star). **(B)** Pairwise identification accuracy of dreamed object categories for each combination of DNN layers and ROIs. The pairwise identification analysis was carried out for all combinations of DNN layers and ROIs. **(C)** Pairwise identification accuracy for the dream (left), perception (middle), and imagery (right) datasets. The visual feature vectors decoded from VC activity were used for identification analyses [error bars, 95% CI across samples; two subjects pooled; three-volume (9 s) averaged fMRI signals immediately before awakening].

for decoder training: this characteristic was conceptualized as "generic object decoding" in our previous brain decoding study (Horikawa and Kamitani, 2015). This framework is also known as the "*zero-data learning*" or "*zero-shot learning*" (Larochelle et al., 2008) in the machine-learning field, in which a model must generalize to classes with no training data. In the previous dream decoding study (Horikawa et al., 2013), the target categories to be decoded from brain activity patterns were determined from and restricted to the reported dream contents consisted of around 20 object categories for each subject. By contrast, we did decoding of dreamed object categories via predictions of DNN features. Thereby, we were able to decode arbitrary categories once the decoders were trained, even though the fMRI data for decoder training were collected irrespective of the reported dream contents. Our results extended the previous results on generic decoding of seen and imagined objects (Horikawa and Kamitani, 2015) to dreamed objects, demonstrating the generalizability of the generic decoding approach across different visual experiences.

The present study extended the previous results reporting recruitments of hierarchical visual feature representations during volitional mental imagery (Horikawa and Kamitani, 2015) to spontaneous mental imagery, in the sense that the feature-level representations were recruited without volitional attempt to visualize images. Taken together with the generalizability of the generic decoding from task-induced brain activity to spontaneous brain activity, we may be able to expect that the generic decoding approach via visual feature prediction is also applicable to decoding of visual information from other types of spontaneously generated subjective experiences, such as mind wandering (Smallwood and Schooler, 2015) or visual hallucination induced by psychedelic drugs (Carhart-Harris et al., 2016), which may help to understand the general principles of neural representations of our visual experience.

Our demonstration of the generic decoding of dreamed object categories indicates the commonality of hierarchical neural representations between perception and dreaming, but there still may be a representational difference between, perception, imagery and dreaming as suggested from different tendency in identification accuracy across DNN layers (**Figure 5C**). In our analyses, the identification accuracy of seen and imagined objects showed a single peak at around mid-level DNN layers (DNN4–7). On the other hand, the identification accuracy of dreamed objects showed poor accuracy at DNN7, whereas DNN8, which showed relatively poor accuracy for the seen and imagined conditions, showed higher accuracy (**Figure 5C**). Additionally, the high-level ROIs (LOC and FFA) rather than the mid-level ROI (V4) tended to show higher dreamed object identification accuracy for

most of DNN layers (**Figure 5B**), while the identification of seen and imagined objects showed highest accuracy from V4 activity (Horikawa and Kamitani, 2015). Because there were differences in test categories between the perception/imagery datasets and dream dataset (50 test categories for perception and imagery; ∼20 reported dream categories for dreaming) such difference may partially affect the results. However, the discontinuous profile of identification accuracy across DNN layers, relatively high accuracy in mid- and top-level DNN layers (DNN5 and 8) and low accuracy in DNN6 and 7, might suggest the involvement of higher cognitive functions, such as memory and abstract knowledge, to generate object representations during dreaming. Our time course analyses of the feature prediction accuracy showed the prolonged high accuracy for the high-level DNN features during reporting periods (**Figure 4**), which may also be explained by higher level cognitive functions related with memory retrieval and verbal reporting. The associations between dreaming and higher cognitive functions (Nir and Tononi, 2009) may lead to robust representations that resemble the high-level DNN layer.

While our analyses showed higher decodability for features at mid- to high-level DNN layers from relatively higher ROIs (**Figures 2B**, **3B**), we were not able to provide evidence on how low-level features of dreamed objects are represented in lower ROIs. This was partly because our analyses were restricted to feature representations associated with object categories. Specifically, while the decoders were trained to decode visual features of individual images, the decoding accuracy was evaluated by correlation coefficients between the decoded features and the category features. Furthermore, the decoding from dream fMRI data was based on category-averaged brain activity and not based on brain activity induced by a specific image. Because of these limitations, our analyses should have reduced the sensitivity to the information on low-level image features. Thus, the poor accuracy for low-level DNN features and ROIs does not necessarily mean that we should reject the possibility of the recruitment of low-level image features in the representation of dream contents. Whether low-level image features, such as color and contrast, are represented in the dreaming brain is a topic that is worth addressing in future study.

AUTHOR CONTRIBUTIONS

TH and YK designed the study. TH performed experiments and analyses. TH and YK wrote the paper.

FUNDING

This research was supported by grants from JSPS KAKENHI Grant number JP26119536, JP26870935, JP15H05920, JP15H05710, ImPACT Program of Council for Science, Technology and Innovation (Cabinet Office, Government of Japan), and the New Energy and Industrial Technology Development Organization (NEDO).

ACKNOWLEDGMENTS

The authors thank Kei Majima and Mitsuaki Tsukamoto for helpful comments on the manuscript

REFERENCES

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. New York, NY: Springer.

Braun, A. R., Balkin, T. J., Wesensten, N. J., Gwadry, F., Carson, R. E., Varga, M., et al. (1998). Dissociated pattern of activity in visual cortices and their projections during human rapid eye movement sleep. *Science* 279, 91–95. doi: 10.1126/science.279.5347.91

Braun, A. R., Balkin, T. J., Wesenten, N. J., Carson, R. E., Varga, M., Baldwin, P., et al. (1997). Regional cerebral blood flow throughout the sleep-wake cycle. An H2(15)O PET study. *Brain* 120, 1173–1197. doi: 10.1093/brain/120.7.1173

Carhart-Harris, R. L., Muthukumaraswamy, S., Roseman, L., Kaelen, M., Droog, W., Murphy, K., et al. (2016). Neural correlates of the LSD experience revealed by multimodal neuroimaging. *Proc. Natl. Acad. Sci. U.S.A.* 113, 4853–4858. doi: 10.1073/pnas.1518377113

Deng, J., Dong, W., Cocher, R., Li, L. J., Li, K., and Fei-Fei, L. (2009). *Imagenet: A large-Scale Hierarchical Image Database*. Miami Beach, FL: IEEE CVPR.

Dresler, M., Koch, S. P., Wehrle, R., Spoormaker, V. I., Holsboer, F., Steiger, A., et al. (2011). Dreamed movement elicits activation in the sensorimotor cortex. *Curr. Biol.* 21, 1833–1837. doi: 10.1016/j.cub.2011.09.029

Engel, S. A., Rumelhart, D. E., Wandell, B. A., Lee, A. T., Glover, G. H., Chichilnisky, E. J., et al. (1994). fMRI of human visual cortex. *Nature* 369, 525. doi: 10.1038/369525a0

Epstein, R., and Kanwisher, N. (1998). A cortical representation of the local visual environment. *Nature* 392, 598–601. doi: 10.1038/33402

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.

Hong, C. C., Harris, J. C., Pearlson, G. D., Kim, J. S., Calhoun, V. D., Fallon, J. H., et al. (2009). fMRI evidence for multisensory recruitment associated with rapid eye movements during sleep. *Hum. Brain. Mapp.* 30, 1705–1722. doi: 10.1002/hbm.20635

Horikawa, T., and Kamitani, Y. (2015). Generic decoding of seen and imagined objects using hierarchical visual features. *arXiv:*1510.06479.

Horikawa, T., Tamaki, M., Miyawaki, Y., and Kamitani, Y. (2013). Neural decoding of visual imagery during sleep. *Science* 340, 639–642. doi: 10.1126/science.1234330

Jozwik, K. M., Kriegeskorte, N., and Mur, M. (2016). Visual features as stepping stones toward semantics: explaining object similarity in IT and perception with non-negative least squares. *Neuropsychologia* 83, 201–226. doi: 10.1016/j.neuropsychologia.2015.10.023

Kanwisher, N., McDermott, J., and Chun, M. M. (1997). The fusiform face area: a module in human extrastriate cortex specialized for face perception. *J. Neurosci.* 17, 4302–4311.

Kay, K. N., Naselaris, T., Prenger, R. J., and Gallant, J. L. (2008). Identifying natural images from human brain activity. *Nature* 452, 352–355. doi: 10.1038/nature06713

Khaligh-Razavi, S. M., and Kriegeskorte, N. (2014). Deep supervised, but not unsupervised, models may explain IT cortical representation. *PLoS Comput. Biol.* 10:e1003915. doi: 10.1371/journal.pcbi.1003915

Kourtzi, Z., and Kanwisher, N. (2000). Cortical regions involved in perceiving object shape. *J. Neurosci.* 20, 3310–3318.

Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). *ImageNet Classification with Deep Convolutional Neural Networks*. Lake Tahoe, CA: NIPS.

Larochelle, H., Erhan, D., and Bengio, Y. (2008). "Zero-data learning of new tasks," in *AAAI Conference on Artificial Intelligence* (Chicago, IL).

Mahendran, A., and Vedaldi, A. (2016). Visualizing deep convolutional neural networks using natural pre-images. *Int. J. Comput. Vis.* 120, 233–255. doi: 10.1007/s11263-016-0911-8

Maquet, P. (2000). Functional neuroimaging of normal human sleep by positron emission tomography. *J. Sleep Res.* 9, 207–231. doi: 10.1046/j.1365-2869.2000.00214.x

Maquet, P., Péters, J., Aerts, J., Delfiore, G., Degueldre, C., Luxen, A., et al. (1996). Functional neuroanatomy of human rapid-eye-movement sleep and dreaming. *Nature* 383, 163–166. doi: 10.1038/383163a0

Naselaris, T., Olman, C. A., Stansbury, D. E., Ugurbil, K., and Gallant, J. L. (2015). A voxel-wise encoding model for early visual areas decodes mental images of remembered scenes. *Neuroimage* 105, 215–228. doi: 10.1016/j.neuroimage.2014.10.018

Nguyen, A., Yosinski, J., and Clune, J. (2016). Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. *arXiv:*1602.03616.

Nir, Y., and Tononi, G. (2009). Dreaming and the brain: from phenomenology to neurophysiology. *Trends. Cogn. Sci.* 14, 88–100. doi: 10.1016/j.tics.2009.12.001

Sereno, M. I., Dale, A. M., Reppas, J. B., Kwong, K. K., Belliveau, J. W., Brady, T. J., et al. (1995). Borders of multiple visual areas in humans revealed by functional magnetic resonance imaging. *Science* 268, 889–893. doi: 10.1126/science.7754376

Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv:*1312.6034.

Smallwood, J., and Schooler, J. W. (2015). The science of mind wandering: empirically navigating the stream of consciousness. *Annu. Rev. Psychol.* 66, 487–518. doi: 10.1146/annurev-psych-010814-015331

Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., and Lipson, H. (2015). "Understanding neural networks through deep visualization," in *Deep Learning Workshop* (Lille: ICML Conference).

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Computational Neuroscience

ORIGINAL RESEARCH
published: 09 February 2017
doi: 10.3389/fncom.2017.00007

Modeling the Dynamics of Human Brain Activity with Recurrent Neural Networks

Umut Güçlü * *and Marcel A. J. van Gerven*

Donders Institute for Brain, Cognition and Behaviour, Radboud University, Nijmegen, Netherlands

Encoding models are used for predicting brain activity in response to sensory stimuli with the objective of elucidating how sensory information is represented in the brain. Encoding models typically comprise a nonlinear transformation of stimuli to features (feature model) and a linear convolution of features to responses (response model). While there has been extensive work on developing better feature models, the work on developing better response models has been rather limited. Here, we investigate the extent to which recurrent neural network models can use their internal memories for nonlinear processing of arbitrary feature sequences to predict feature-evoked response sequences as measured by functional magnetic resonance imaging. We show that the proposed recurrent neural network models can significantly outperform established response models by accurately estimating long-term dependencies that drive hemodynamic responses. The results open a new window into modeling the dynamics of brain activity in response to sensory stimuli.

Keywords: encoding, fMRI, RNN, LSTM, GRU

OPEN ACCESS

Edited by:
Alexandre Gramfort,
Télécom ParisTech, France

Reviewed by:
Iris I. A. Groen,
National Institutes of Health, USA
Karl Friston,
University College London, UK
Seyed-Mahdi Khaligh-Razavi,
Massachusetts Institute of Technology, USA

***Correspondence:**
Umut Güçlü
u.guclu@donders.ru.nl

Received: *05 October 2016*
Accepted: *25 January 2017*
Published: *09 February 2017*

Citation:
Güçlü U and van Gerven MAJ (2017) Modeling the Dynamics of Human Brain Activity with Recurrent Neural Networks. Front. Comput. Neurosci. 11:7. doi: 10.3389/fncom.2017.00007

1. INTRODUCTION

Encoding models (Naselaris et al., 2011) are used for predicting brain activity in response to naturalistic stimuli (Felsen and Dan, 2005) with the objective of understanding how sensory information is represented in the brain. Encoding models typically comprise two main components. The first component is a feature model that nonlinearly transforms stimuli to features (i.e., the independent variables used in fMRI time series analyses). The second component is a response model that linearly transforms features to responses. While encoding models have been successfully used to characterize the relationship between stimuli in different modalities and responses in different brain regions, their performance usually falls short of the expected performance of the true encoding model given the noise in the analyzed data (noise ceiling). This means that there usually is unexplained variance in the analyzed data that can be explained solely by improving the encoding models.

One way to reach the noise ceiling is the development of better feature models. Recently, there has been extensive work in this direction. One example is the use of convolutional neural network representations of natural images or natural movies to explain low-, mid- and high-level representations in different brain regions along the ventral (Agrawal et al., 2014; Cadieu et al., 2014; Khaligh-Razavi and Kriegeskorte, 2014; Yamins et al., 2014; Güçlü and van Gerven, 2015a; Cichy et al., 2016) and dorsal streams (Güçlü and van Gerven, 2015b; Eickenberg et al., 2016) of the human visual system. Another example is the use of manually constructed or statistically estimated representations of words and phrases to explain the semantic representations in different brain

regions (Mitchell et al., 2008; Huth et al., 2012; Murphy et al., 2012; Fyshe et al., 2013; Güçlü and van Gerven, 2015c; Nishida et al., 2015).

Another way to reach the noise ceiling is the development of better response models. There is a long history of estimating hemodynamic response functions (HRFs) in fMRI time series modeling. The standard general linear (convolution) model used in procedures like statistical parametric mapping (SPM) expands the HRF in terms of orthogonal kernels or temporal basis functions that have been motivated in terms of Volterra expansions. Indeed, commonly used software packages such as the SPM software have (hidden) facilities to model second-order Volterra kernels that enable modeling of non-linear hemodynamic effects such as saturation. In reality, the transformation from stimulus features to observed responses is exceedingly complex because of various temporal dependencies that are caused by neurovascular coupling (Logothetis and Wandell, 2004; Norris, 2006) and other more elusive cognitive or neural factors.

Here, our objective is to develop a model that can be trained end to end, captures temporal dependencies and processes arbitrary input sequences for time-continuous fMRI experiments such as watching movies, listening to music or playing video games. Such time-continuous designs are characterized by the absence of discrete experimental events as those found in their block or event-related counterparts. To this end, we use recurrent neural networks (RNNs) as response models in the encoding framework. Recently, RNNs in general and two RNN variants—long short-term memory (Hochreiter and Schmidhuber, 1997) and gated recurrent units (Cho et al., 2014)—in particular have been shown to be extremely successful in various tasks that involve processing of arbitrary input sequences such as handwriting recognition (Graves et al., 2009; Graves, 2013), language modeling (Sutskever et al., 2011; Graves, 2013), machine translation (Cho et al., 2014) and speech recognition (Sak et al., 2014). These models use their internal memories to capture the temporal dependencies that are informative about solving the task at hand. That is, these models base their predictions not only to the information available at a given time, but also to the information that was available in the past. They accomplish this by maintaining an explicit or implicit representation of the past input sequences and use it to make their predictions at each time point. If these models can be used as response models in the encoding framework, it will open a new window into modeling brain activity in response to sensory stimuli since the brain activity is modulated by long temporal dependencies.

While the use of RNNs in the encoding framework has been proposed a number of times (Güçlü and van Gerven, 2015a,b; Kriegeskorte, 2015; Yamins and DiCarlo, 2016a,b), these proposals mainly focused on using RNNs as feature models. In contrast, we have framed our approach in terms of response models used in characterizing distributed or multivariate responses to stimuli in the encoding framework. The key thing that we bring to the table is a generic and potentially useful response model that transforms features to observed (hemodynamic) responses. From the perspective of conventional analyses of functional magnetic resonance imaging (fMRI) time series, this response model corresponds to the convolution model used to map stimulus features (e.g., the presence of biological motion) to fMRI responses. In other words, the stimulus features correspond to conventional stimulus functions that enter standard convolution models of fMRI time series (e.g., the GLM used in statistical parametric mapping).

In brief, we know that the transformation from neuronal responses to fMRI signals is mediated by neuronal and hemodynamic factors that can always be expressed in terms of a non-linear convolution. A general form for these convolutions has been previously considered in the form of Volterra kernels or functional Taylor expansions (Friston et al., 2000). Crucially, it is also well known that RNNs are universal non-linear approximators that can reproduce any Volterra expansion (Wray and Green, 1994). This means that we can use RNNs as an inclusive and flexible way to parameterize the convolution of stimulus features generating hemodynamic responses. Furthermore, we can use RNNs to model not just response of a single voxel but distributed responses over multiple voxels. Having established the parametric form of this convolution, the statistical evidence or significance of each regionally specific convolution can then be assessed using standard (cross-validation) machine learning techniques by comparing the accuracy of the convolution when applied to test data after optimization with training data.

We test our approach by comparing how well a family of RNN models and a family of ridge regression models can predict blood-oxygen-level dependent (BOLD) hemodynamic responses to high-level and low-level features of natural movies using cross-validation. We show that the proposed recurrent neural network models can significantly outperform the standard ridge regression models and accurately estimate hemodynamic response functions by capturing temporal dependencies in the data.

2. MATERIALS AND METHODS

2.1. Data Set

We analyzed the vim-2 data set (Nishimoto et al., 2014), which was originally published by Nishimoto et al. (2011). The experimental procedures are identical to those in Nishimoto et al. (2011). Briefly, the data set has twelve 600 s blocks of stimulus and response sequences in a training set and nine 60 s blocks of stimulus and response sequences in a test set. The stimulus sequences are videos (512 px $\times$ 512 px or $20^\circ \times 20^\circ$, 15 FPS) that were drawn from various sources. The response sequences are BOLD responses (voxel size $= 2 \times 2 \times 2.5\,\text{mm}^3$, TR $= 1$ s) that were acquired from the occipital cortices of three subjects (S1, S2, and S3). The stimulus sequences in the test set were repeated ten times. The corresponding response sequences were averaged over the repetitions. The response sequences have already been preprocessed as described in Nishimoto et al. (2011). Briefly, they have been realigned to compensate for motion, detrended to compensate for drift and z-scored. Additionally, the first six seconds of the blocks were discarded. No further preprocessing was performed. Regions of

interests were localized using the multifocal retinotopic mapping technique on retinotopic mapping data that were acquired in separate sessions (Hansen et al., 2004). As a result, the voxels were grouped into 16 areas. However, not all areas were identified in all subjects (**Table 1**). The last 45 seconds of the blocks in the training set were used as the validation set.

2.2. Problem Statement

Let $\mathbf{x}^t \in \mathbb{R}^n$ and $\mathbf{y}^t \in \mathbb{R}^m$ be a stimulus and a response at temporal interval $[t, t+1]$, where n is the number of stimulus dimensions and m is the number of voxel responses. We are interested in predicting the most likely response $\mathbf{y}^t$ given the stimulus history $\mathbf{X}^t = (\mathbf{x}^0, \ldots, \mathbf{x}^t)$:

$$\hat{\mathbf{y}}^t = \arg\max_{\mathbf{y}^t} \Pr\left(\mathbf{y}^t \middle| \mathbf{X}^t\right) \quad (1)$$

$$= \mathbf{g}\left(\phi\left(\mathbf{x}^0\right), \ldots, \phi\left(\mathbf{x}^t\right)\right) \quad (2)$$

where Pr is an encoding distribution, ϕ is a feature model such that $\phi(\cdot) \in \mathbb{R}^p$, p is the number of feature dimensions, and $\mathbf{g}$ is a response model such that $\mathbf{g}(\cdot) \in \mathbb{R}^m$.

In order to solve this problem, we must define the feature model that transforms stimuli to features and the response model that transforms features to responses. We used two alternative feature models; a scene description model that codes for low-level visual features (Oliva and Torralba, 2001) and a word embedding model that codes for high-level semantic content. We used two response model families that differ in architecture (recurrent neural network family and feedforward ridge regression family) (**Figure 1**). In contrast to standard convolution models for fMRI time series, we are dealing with potentially very large feature spaces. This means that in the absence of constraints the optimization of model parameters can be ill posed. Therefore, we use dropout and early stopping for the recurrent models, and L^2 regularization for the feedforward models.

2.3. Feature Models

2.3.1. High-Level Semantic Model

As a high-level semantic model we used the word2vec (W2V) model by Mikolov et al. (2013a,b,c). This is a one-layer feedforward neural network that is trained for predicting either target words/phrases from source-context words (continuous bag-of-words) or source context-words from target words/phrases (skip-gram). Once trained, its hidden states are used as continuous distributed representations of words/phrases. These representations capture many semantic regularities. We used the pretrained (skip-gram) W2V model to avoid training from scratch (https://code.google.com/archive/p/word2vec/). It was trained on 100 billion-word Google News dataset. It contains 300-dimensional continuous distributed representations of three million words/phrases.

We used the W2V model for transforming a stimulus sequence to a feature sequence on a second-by-second basis as follows: First, each one second of the stimulus sequence is assigned 20 categories (words/phrases). We used the *Clarifai* service (http://www.clarifai.com/) to automatically assign the categories rather than annotating them by hand. *Clarifai* provides a web-based video recognition application, which internally uses a pretrained deep neural network to automatically tag the contents of the video frames on a second-by-second basis. Then, each category is transformed into continuous distributed representations of words/phrases. Next, these representations are averaged over the categories. This resulted in a 300-dimensional feature vector per second of stimulus sequence ($p = 300$).

2.3.2. Low-Level Visual Feature Model

As a low-level visual feature model we used the GIST model (Oliva and Torralba, 2001). The GIST model transforms scenes into spatial envelope representations. These representations capture many perceptual dimensions that represent the dominant spatial structure of a scene and have been used to study neural representations in a number of earlier work (Groen et al., 2013; Leeds et al., 2013; Cichy et al., 2016). We used the implementation that is provided at: http://people.csail.mit.edu/torralba/code/spatialenvelope/.

We used the GIST model for transforming a stimulus sequence to a feature sequence on a second-by-second basis as follows: First, each 16 non-overlapping 8×8 regions of all 15 128×128 frames in one second of the stimulus sequence are filtered with 32 Gabor filters that have eight orientations and four scales. Then, their energies are averaged over the frames. This resulted in a 512-dimensional feature vector per second of stimulus sequence ($p = 512$).

2.4. Response Models

2.4.1. Ridge Regression Family

The response models in the ridge regression family predict feature-evoked responses as a linear combination of features. Each member of this family differs in how it accounts for the hemodynamic delay.

The *R-C* model (i) convolves the features with the canonical hemodynamic response function (Friston et al., 1994) and (ii) predicts the responses as a linear combination of these features:

$$\hat{\mathbf{y}}^t = \left(\mathbf{H}_c \mathbf{F}_c \mathbf{B}^\top\right)^t \quad (3)$$

TABLE 1 | Number of voxels per subject and area.

	V2	V3	V1	IPS	V4	LOC	V7	MT+	V3A	V3B	VO	EBA	OFA	RSC	pSTS	TOS
S1	1,477	1,141	994	2,251	734	885	0	466	252	256	410	0	0	71	45	0
S2	1,659	1,360	1,043	0	1032	614	400	174	337	223	267	319	246	128	0	0
S3	1,377	1,131	1,366	893	750	408	583	263	282	225	0	131	91	8	16	41

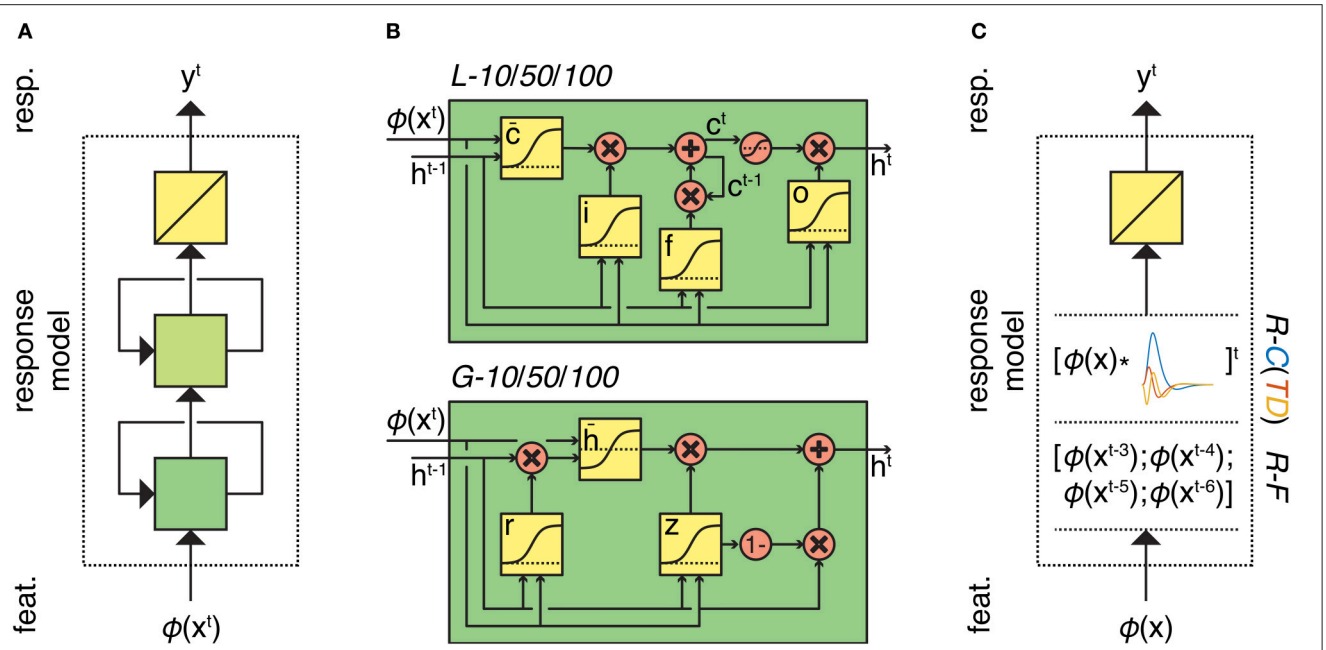


FIGURE 1 | Overview of the response models. (A) Response models in the RNN family. All RNN models process feature sequences via two (recurrent) nonlinear layers and one (nonrecurrent) linear layer but differ in the type and number of artificial neurons. *L-10/50/10* models have 10, 50, or 100 long short-term memory units in both of their hidden layers, respectively. Similarly, *G-10/50/10* models have 10, 50, or 100 gated recurrent units in both of their hidden layers, respectively. **(B)** First-layer long short-term memory and gated recurrent units. Squares indicate linear combination and nonlinearity. Circles indicate elementwise operations. Gates in the units control the information flow between the time points. **(C)** Response models in the ridge regression family. All ridge regression models process feature sequences via one (nonrecurrent) linear layer but differ in how they account for the hemodynamic delay. *R-C(TD)* models convolve the feature sequence with the canonical hemodynamic response function (and its time and dispersion derivatives). *R-F* model lags the feature sequence for 3, 4, 5, and 6 s and concatenates the lagged sequences.

where $\mathbf{H}_c \in \mathbb{R}^{t\times t}$ is the Toeplitz matrix of the canonical HRF. That is, it is a diagonal-constant matrix that contains the shifted versions of the HRF in its columns. Multiplying it with a signal corresponds to convolution of the HRF with the signal. Furthermore, $\mathbf{F}_c = \left[\phi\left(\mathbf{x}^0\right), \ldots, \phi\left(\mathbf{x}^t\right)\right]^\top \in \mathbb{R}^{t\times p}$ and $\mathbf{B} \in \mathbb{R}^{m\times p}$ is the matrix of regression coefficients.

The *R-CTD* model (i) convolves the features with the canonical hemodynamic response function, its temporal derivative and its dispersion derivative (Friston et al., 1998), (ii) concatenates these features and (iii) predicts the responses as a linear combination of these features:

$$\hat{\mathbf{y}}^t = \left(\left[\mathbf{H}_c\mathbf{F}_c, \mathbf{H}_{ct}\mathbf{F}_c, \mathbf{H}_{cd}\mathbf{F}_c\right]\mathbf{B}^\top\right)^t \tag{4}$$

where $\mathbf{H}_{ct} \in \mathbb{R}^{t\times t}$ is the Toeplitz matrix of the the temporal derivative of the canonical HRF, $\mathbf{H}_{cd} \in \mathbb{R}^{t\times t}$ is the Toeplitz matrix of the the dispersion derivative of the canonical HRF and $\mathbf{B} \in \mathbb{R}^{m\times 3p}$ is the matrix of regression coefficients.

The *R-F* model is a finite impulse response (FIR) model that (i) lags the features for 3, 4, 5, and 6 s (Nishimoto et al., 2011), (ii) concatenates these features and (iii) predicts the responses as a linear combination of these features:

$$\hat{\mathbf{y}}^t = \mathbf{F}_f\mathbf{B}^\top \tag{5}$$

where $\mathbf{F}_f = \left[\phi\left(\mathbf{x}^{t-3}\right), \phi\left(\mathbf{x}^{t-4}\right), \phi\left(\mathbf{x}^{t-5}\right), \phi\left(\mathbf{x}^{t-6}\right)\right]^\top \in \mathbb{R}^{t\times 4p}$ and $\mathbf{B} \in \mathbb{R}^{m\times 4p}$ is the matrix of regression coefficients.

We used the validation set for model selection (a regularization parameter per voxel) and the training set for model estimation (a row of $\mathbf{B}$ per voxel). Regularization parameters were selected as explained in Güçlü and van Gerven (2014). The rows of $\mathbf{B}$ were estimated by analytically minimizing the L^2-penalized least squares loss function. In related Bayesian models, this corresponds to applying shrinkage priors to the parameters (weights) of our model.

2.4.2. Recurrent Neural Network Family

The response models in the RNN family are two-layer recurrent neural network models. They use their internal memories for nonlinearly processing arbitrary feature sequences and predicting feature-evoked responses as a linear combination of their second-layer hidden states:

$$\hat{\mathbf{y}}^t = \mathbf{h}_2^t\mathbf{W}^\top \tag{6}$$

where $\mathbf{h}_2^t$ represents the hidden states in the second layer, and $\mathbf{W}$ are the weights. The RNN models differ in the type and number of artificial neurons.

The *L-10*, *L-50*, and *L-100* models are two-layer recurrent neural networks that have 10, 50, and 100 long short-term memory (LSTM) units (Hochreiter and Schmidhuber, 1997) in

their hidden layers, respectively. Each LSTM unit has a cell state that acts as its internal memory by storing information from previous time points. The contents of the cell state are modulated by the gates of the unit and in turn modulate its outputs. As a result, the output of the unit is not only controlled by the present stimulus alone, but also by the stimulus history. The gates are implemented as multiplicative sigmoid functions of the inputs of the unit at the current time point and the outputs of the unit at the previous time point. That is, the gates produce values between zero and one, which are multiplied by (a function of) the cell state to determine the amount of information to store, forget or retrieve at each time point. The first-layer hidden states of an LSTM unit are defined as follows:

$$\mathbf{h}^t = \mathbf{o}^t \odot \tanh\left(\mathbf{c}^t\right) \tag{7}$$

$$\mathbf{o}^t = \sigma\left(\mathbf{U}_o\mathbf{h}^{t-1} + \mathbf{W}_o\phi\left(\mathbf{x}^t\right) + \mathbf{b}_o\right) \tag{8}$$

where $\odot$ denotes elementwise multiplication, $\mathbf{c}^t$ is the cell state, and $\mathbf{o}^t$ are the output gate activities. The cell state maintains information about the previous time points. The output gate controls what information will be retrieved from the cell state. The cell state of an LSTM unit is defined as:

$$\mathbf{c}^t = \mathbf{f}^t \odot \mathbf{c}^{t-1} + \mathbf{i}^t \odot \bar{\mathbf{c}}^t \tag{9}$$

$$\mathbf{f}^t = \sigma\left(\mathbf{U}_f\mathbf{h}^{t-1} + \mathbf{W}_f\phi\left(\mathbf{x}^t\right) + \mathbf{b}_f\right) \tag{10}$$

$$\mathbf{i}^t = \sigma\left(\mathbf{U}_i\mathbf{h}^{t-1} + \mathbf{W}_i\phi\left(\mathbf{x}^t\right) + \mathbf{b}_i\right) \tag{11}$$

$$\bar{\mathbf{c}}^t = \sigma\left(\mathbf{U}_c\mathbf{h}^{t-1} + \mathbf{W}_c\phi\left(\mathbf{x}^t\right) + \mathbf{b}_c\right) \tag{12}$$

where $\mathbf{f}^t$ are the forget gate activities, $\mathbf{i}^t$ are the input gate activities, and $\bar{\mathbf{c}}^t$ is an auxiliary variable. Forget gates control what old information will be discarded from the cell states. Input gates control what new information will be stored in the cell states. Furthermore, **U**s and **W**s are the weights and **b**s are the biases that determine the behavior of the gates (i.e., the learnable parameters of the model).

The *G-10*, *G-50*, and *G-100* models are two-layer recurrent neural networks that have 10, 50, and 100 gated recurrent units (GRU) (Cho et al., 2014) in the their hidden layers, respectively. The GRU units are simpler alternatives to the LSTM units. They combine hidden states with cell states and input gates with forget gates. The first-layer hidden states of a GRU unit is defined as follows:

$$\mathbf{h}^t = \left(1 - \mathbf{z}^t\right) \odot \mathbf{h}^{t-1} + \mathbf{z}^t \odot \bar{\mathbf{h}}^t \tag{13}$$

$$\mathbf{z}^t = \sigma\left(\mathbf{U}_z\mathbf{h}^{t-1} + \mathbf{W}_z\phi\left(\mathbf{x}^t\right) + \mathbf{b}_z\right) \tag{14}$$

$$\mathbf{r}^t = \sigma\left(\mathbf{U}_r\mathbf{h}^{t-1} + \mathbf{W}_r\phi\left(\mathbf{x}^t\right) + \mathbf{b}_r\right) \tag{15}$$

$$\bar{\mathbf{h}}^t = \tanh\left(\mathbf{U}_h\left(\mathbf{r}_t \odot \mathbf{h}^{t-1}\right) + \mathbf{W}_h\phi\left(\mathbf{x}^t\right) + \mathbf{b}_h\right) \tag{16}$$

where $\mathbf{z}^t$ are update gate activities, $\mathbf{r}^t$ are reset gate activities and $\bar{\mathbf{h}}^t$ is an auxiliary variable. Like the gates in LSTM units, those in GRU units control the information flow between the time points. As before, **U**s and **W**s are the weights and **b**s are the biases that determine the behavior of the gates (i.e., the learnable parameters of the model).

The second-layer hidden states are defined similarly to the first-layer hidden states except for replacing the input features with the first-layer hidden states. For each previously identified brain area of each subject, a separate model was trained. That is, the voxels in a given brain area of a given subject shared the same recurrent layers but had different weights for linearly transforming the hidden states of the second recurrent layer to the response predictions. We used truncated backpropagation through time in conjunction with the optimization method Adam (Kingma and Ba, 2014) to train the models on the training set by iteratively minimizing the mean squared error loss function. Dropout (Hinton et al., 2012) was used to regularize the hidden layers. The epoch in which the validation performance was the highest was taken as the best model. The *Chainer* framework (http://chainer.org/) was used to implement the models.

2.5. HRF Estimation

Voxel-specific HRFs were estimated by stimulating the RNN model with an impulse. Let $\mathbf{x}^{-t}, \ldots, \mathbf{x}^0, \ldots, \mathbf{x}^t$ be an impulse such that $\mathbf{x}$ is a vector of zeros at times other than time 0 and a vector of ones at time 0. The period of the impulse before time 0 is used to stabilize the baseline of the impulse response. First, the response of the model to the impulse is simulated:

$$\left[\mathbf{H}_r^*\right]_{-t}^{t} = \mathbf{g}_r\left(\mathbf{x}^{-t}, \ldots, \mathbf{x}^0, \ldots, \mathbf{x}^t\right) \tag{17}$$

where $\left[\mathbf{H}_r^*\right]_{-t}^{t} = \left(\mathbf{H}_r^{*-t}, \ldots, \mathbf{H}_r^{*0}, \ldots, \mathbf{H}_r^{*t}\right)$. Then, the baseline of the impulse response before time 0 is subtracted from itself:

$$\left[\mathbf{H}_r^*\right]_{-t}^{t} = \left[\mathbf{H}_r^*\right]_{-t}^{t} - \mathbf{H}_r^{*-1}. \tag{18}$$

Next, the impulse response is divided by its maximum:

$$\left[\mathbf{H}_r^*\right]_{-t}^{t} = \left[\mathbf{H}_r^*\right]_{-t}^{t} / \max\left[\mathbf{H}_r^*\right]_{-t}^{t}. \tag{19}$$

Finally, the period of the impulse response before time 0 is discarded, and the remaining period of the impulse response is taken as the HRF of the voxels:

$$\left[\mathbf{H}_r\right]_0^t = \left[\mathbf{H}_r^*\right]_0^t. \tag{20}$$

The time when the HRF is at its maximum was taken as the delay of the response, and the time after the delay of the response when the HRF was at its minimum was taken as the delay of undershoot.

2.6. Performance Assessment

The performance of a model for a voxel was defined as the cross-validated Pearson's product-moment correlation coefficient between the observed and predicted responses of the voxel (r)[1]. Its performance for a group of voxels was defined as the median of its performance over the voxels in the group ($\tilde{r}$). The data of all

[1]The cross-validated correlation coefficient automatically penalizes for model complexity and therefore can be used as a proxy for model evidence.

subjects were concatenated prior to analyzing the performance of the models.

In order to make sure that the differences in the performance of a model in different areas are not caused by the differences in the signal-to-noise ratios of the areas, the performance of the model in an area was corrected for the median of the noise ceilings of the voxels in the area ($\tilde{r}^*$) (Kay et al., 2013). Briefly, we performed Monte Carlo simulations in which the correlation coefficient between a signal and a noisy signal is estimated. In each simulation, both the signal and the noise were drawn from a Gaussian distribution. The noisy signal was taken to be the summation of the signal sample and the noise sample. The parameters of the signal and the noise distributions were estimated from the 10 repeated measurements of the responses to the same stimuli. The noise distribution was assumed to be zero mean, and its variance was taken to be the variance of the standard errors of the data. The mean and the variance of the signal distribution were given as the mean of the data, and the difference between the variance of the data and the noise distribution, respectively. The medians of the correlation coefficients that were estimated in the simulations were taken to be the noise ceilings of the voxels, indicating the maximum performance that can be expected from the perfect model due to the noise in the data.

Permutation tests were used for comparing the performance of a model against chance level. First, data were randomly permuted over time for 200 times. Then, a separate model was trained and tested for each of the 200 permutations. Finally, the p-value was taken to be the fraction of the 200 permutations whose performance was greater than the actual performance. The performance was considered significant at $\alpha = 0.05$ if the p-value was less than 0.05 (Bonferroni corrected for number of areas).

Bootstrapping was used for comparing the performance of two models over voxels in a ROI (i.e., all voxels or voxels in an area). For 10,000 repetitions, bootstrap samples (i.e., voxels) were drawn from the ROI with replacement, and the performance difference between the models over these voxels were estimated. The performance difference was considered significant at $\alpha = 0.05$ if the 95% confidence interval of the sampled statistic did not cover zero (Bonferroni corrected for number of models).

3. RESULTS

3.1. Comparison of Response Models

We evaluated the response models by comparing the performance of the response models in the (recurrent) RNN family and (feed-forward) ridge regression family in combination with the (high-level) W2V model and the (low-level) GIST model. Using two feature models of different levels ruled out any potential biases in the performance difference of the response models that can be caused by the feature models. Recall that the models in the RNN family (*G/L-10/50/100* models) differed in the type and number of artificial neurons, whereas the models in the ridge regression family (*R-C/R-CTD/R-F* models) differed in how they account for the hemodynamic delay.

Once the best response models among the RNN family and the ridge regression family were identified, we first compared their performance in detail. Particular attention was paid to the voxels where the performance of the models differed by more than an arbitrary threshold of $r = 0.1$. We then compared the performance of the best response model among the RNN family over the areas along the visual pathway.

3.1.1. Comparison of the Response Models in Combination with the Semantic Model

Figure 2 compares the performance of all response models in combination with the W2V model. The performance of the models in the RNN family that had 50 or 100 artificial neurons was always significantly higher than that of all models in the ridge regression family ($p \leq 0.05$, bootstrapping). However, the performance of the models in the same family was not always significantly different from each other. The performance of the *G-100* model was the highest among the RNN family ($\tilde{r} = 0.16$), and that of the *R-C* model was the highest among the ridge regression family ($\tilde{r} = 0.12$).

The performance of the *G-100* model and the *R-C* model differed from each other by more than the chosen threshold of $r = 0.1$ in 30% of the voxels. The performance of the *G-100* model was higher in 78% of these voxels ($\Delta\tilde{r} = 0.17$), and that of the *R-C* model was higher in 22% of these voxels ($\Delta\tilde{r} = 0.14$).

Figure 3 compares the performance of the *G-100* model in combination with the W2V model over the areas along the visual stream. While the performance of the model was significantly higher than chance throughout the areas ($p \leq 0.05$, permutation test), it was particularly high in downstream areas. For example, it was the highest in TOS ($\tilde{r}^* = 0.55$), OFA ($\tilde{r}^* = 0.38$) and EBA ($\tilde{r}^* = 0.35$), and the lowest in pSTS ($\tilde{r}^* = 0.14$), IPS ($\tilde{r}^* = 0.20$) and V1 ($\tilde{r}^* = 0.24$).

3.1.2. Comparison of the Response Models in Combination with the Low-Level Feature Model

Figure 4 compares the performance of the all response models in combination with the GIST model. The trends that were observed in this figure were similar to those that were observed in **Figure 2**. The *G-100* model was the best among the RNN family ($\tilde{r} = 0.18$), and the *R-C* model was the best among the ridge regression family ($\tilde{r} = 0.14$).

The *G-100* model and the *R-C* differed from each other by more than the threshold of $r = 0.1$ in 27% of the voxels. The *G-100* model was better in 66% of these voxels ($\Delta\tilde{r} = 0.17$). The *R-C* model was better in 34% of these voxels ($\Delta\tilde{r} = 0.14$).

Figure 5 compares the performance of the *G-100* model in combination with the GIST model over the areas along the visual pathway. While the *G-100* model performed significantly better than chance throughout the areas ($p \leq 0.05$, permutation test), it performed particularly well in upstream visual areas. For example, it performed the best in V1 ($\tilde{r}^* = 0.39$), V2 ($\tilde{r}^* = 0.35$) and V3 ($\tilde{r}^* = 0.35$), and the worst in TOS ($\tilde{r}^* = 0.13$), IPS ($\tilde{r}^* = 0.16$) and pSTS ($\tilde{r}^* = 0.16$).

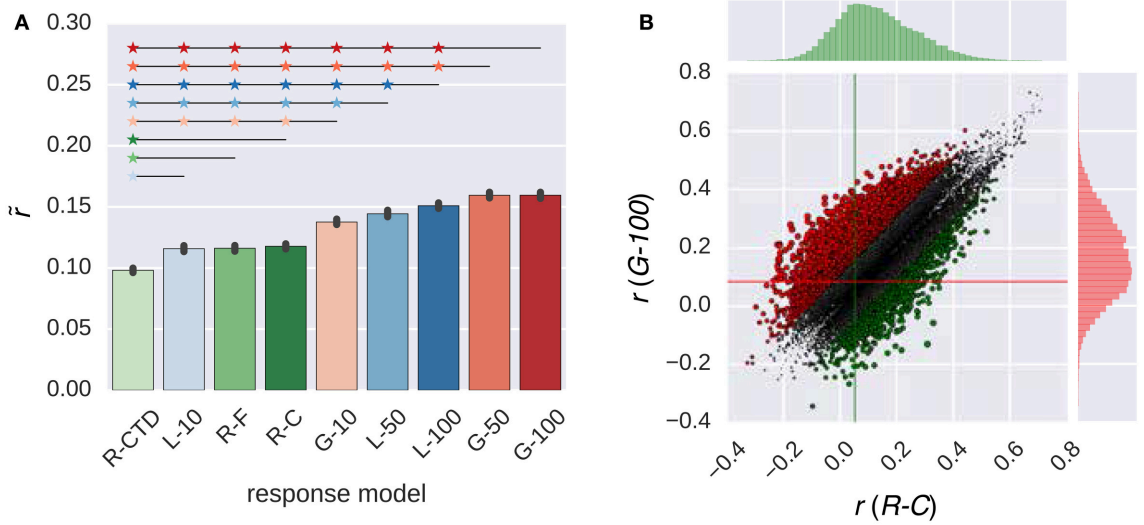


FIGURE 2 | Comparison of the response models in combination with the W2V model. (A) Median performance of response models in RNN (*G-X* and *L-X*) and ridge regression (*R-X*) families over all voxels. Error bars indicate 95% confidence intervals (bootstrapping). Asterisks indicate significant performance difference. All of the individual bars depict significantly above chance-level performance ($p < 0.05$, permutation test). **(B)** Performance of best response models in RNN (*G-100* model) and ridge regression (*R-C* model) families over individual voxels. Points indicate voxels. Gray points indicate voxels where the performance difference is less than $r = 0.1$. Lines indicate (median) performance over all voxels.

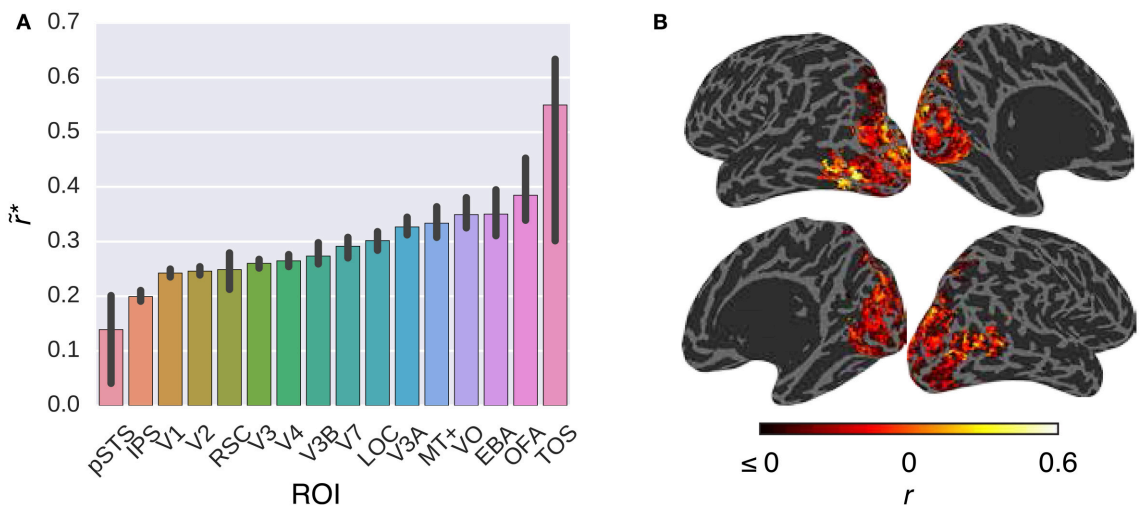


FIGURE 3 | Comparison of the *G-100* model in combination with the W2V model in different areas. (A) Median noise ceiling controlled performance over all voxels in different areas. Error bars indicate 95% confidence intervals (bootstrapping). All of the individual bars depict significantly above chance-level performance ($p < 0.05$, permutation test). **(B)** Projection of performance to cortical surfaces of S3.

3.2. Comparison of Feature Models

Once the efficacy of the proposed RNN models was positively assessed, we performed a validation experiment in which we assessed the extent to which these models can replicate the earlier findings on the low-level and high-level subdivision of the visual cortex. This was accomplished by identifying the voxels that prefer semantic representations vs. low-level representations. Concretely, we compared the performance of the W2V model and the GIST model in combination with the *G-100* model (**Figure 6**).

The performance of the models was significantly different in all areas along the visual stream except for pSTS and V3A ($p \leq 0.05$, bootstrapping). This difference was in favor of semantic representations in downstream areas and low-level representations in upstream areas. The largest difference in favor of semantic representations was in TOS ($\Delta\tilde{r} = 0.11$),

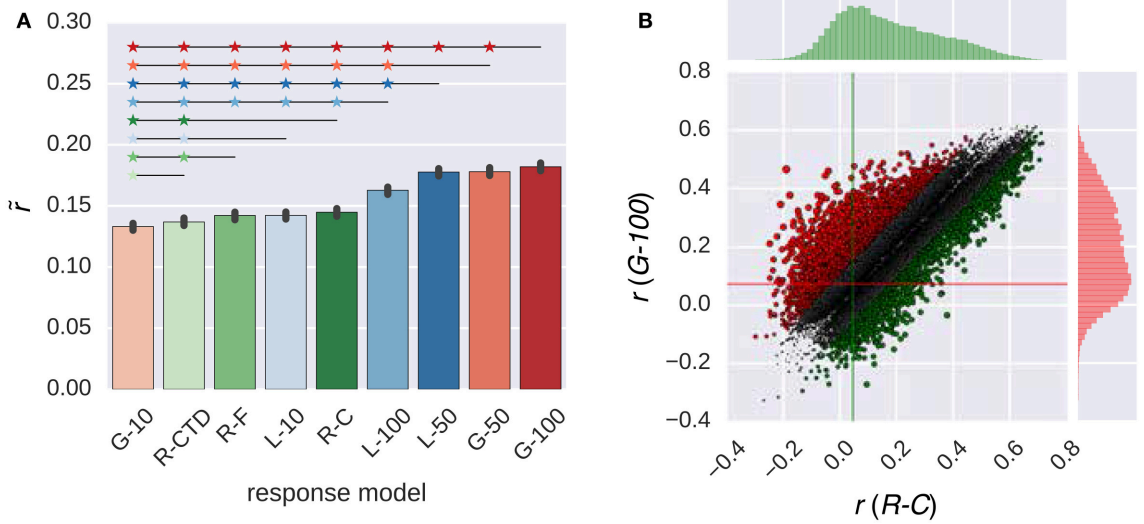


FIGURE 4 | Comparison of the response models in combination with the GIST model. (A) Median performance of response models in RNN (*G-X* and *L-X*) and ridge regression (*R-X*) families over all voxels. Error bars indicate 95% confidence intervals (bootstrapping). Asterisks indicate significant performance difference. All of the individual bars depict significantly above chance-level performance ($p < 0.05$, permutation test). **(B)** Performance of best response models in RNN (*G-100* model) and ridge regression (*R-C model*) families over individual voxels. Points indicate voxels. Gray points indicate voxels where the performance difference is less than $r = 0.1$. Lines indicate median performance over all voxels.

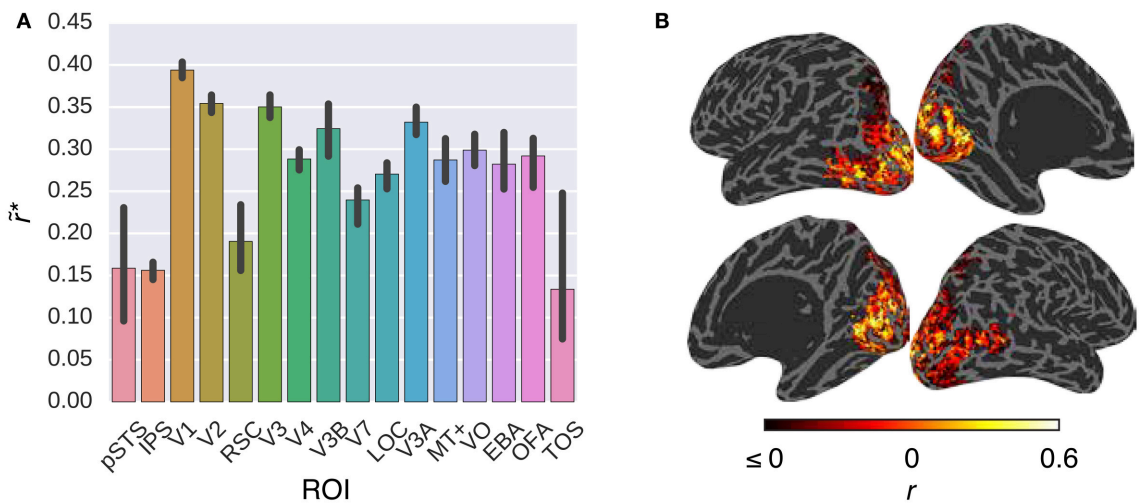


FIGURE 5 | Comparison of the *G-100* model in combination with the GIST model in different areas. (A) Median noise ceiling controlled performance over all voxels in different areas. Error bars indicate 95% confidence intervals (bootstrapping). All of the individual bars depict significantly above chance-level performance ($p < 0.05$, permutation test). **(B)** Projection of performance to cortical surfaces of S3.

OFA ($\Delta\tilde{r} = 0.08$) and MT+ ($\Delta\tilde{r} = 0.04$), and low-level representations was in V1 ($\Delta\tilde{r} = 0.10$), V2 ($\Delta\tilde{r} = 0.07$) and V3 ($\Delta\tilde{r} = 0.05$).

Thirty-nine percent of the voxels preferred either representation by more than the arbitrary threshold of $r = 0.1$. Thirty-four percent of these voxels preferred semantic representations ($\Delta\tilde{r} = 0.16$), and 66% percent of these voxels preferred low-level representations ($\Delta\tilde{r} = 0.18$).

These results are in line with a large number of earlier work that showed similar dissociations between the representations of the upstream and downstream visual areas (Mishkin et al., 1983; Naselaris et al., 2009; DiCarlo et al., 2012; Güçlü and van Gerven, 2015a).

3.3. Analysis of Internal Representations

Next, to gain insight into the temporal dependencies captured by the *G-100* model, we analyzed its internal representations (**Figure 7**).

First, we investigated how the hidden states of the RNN depend on its inputs and output. We constructed

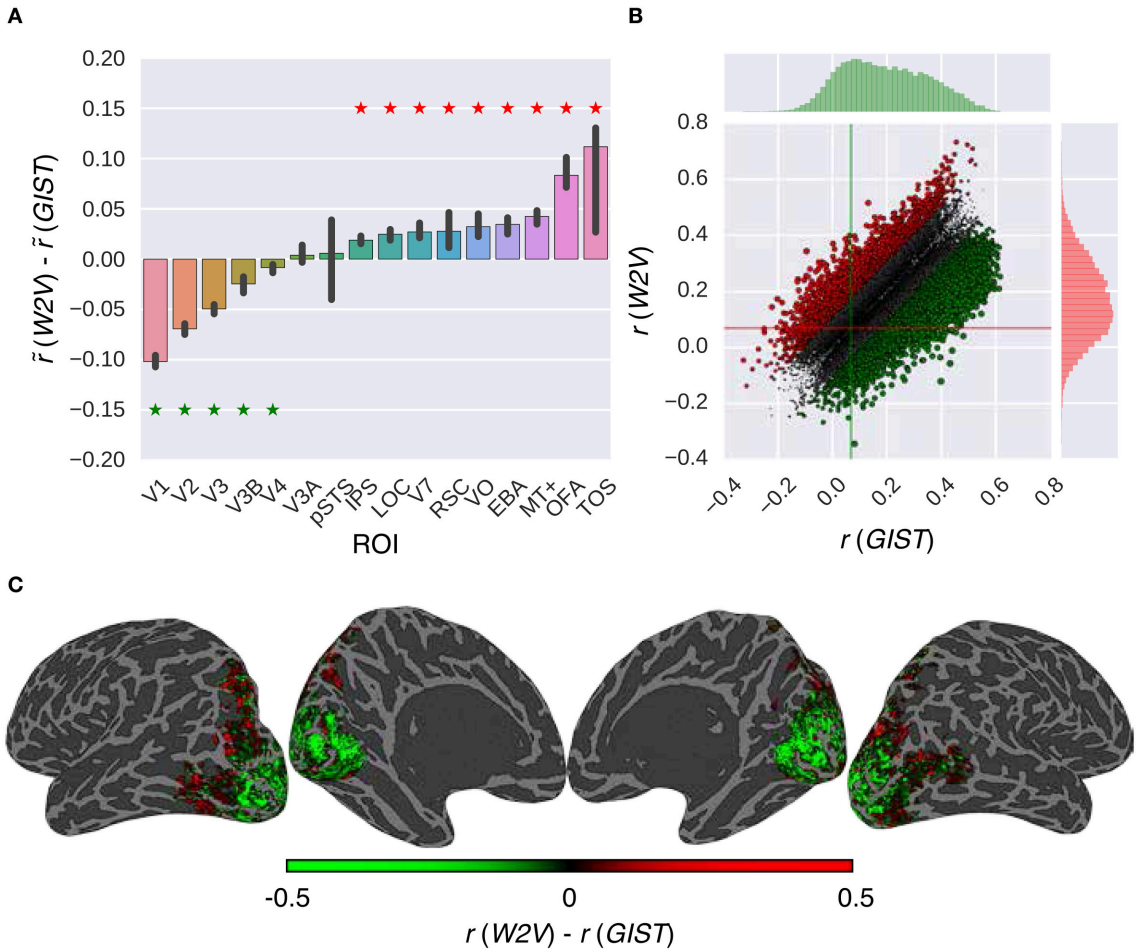


FIGURE 6 | Comparison of the feature models in combination with the *G-100* model. (A) Median performance difference over all voxels in different areas. Asterisks indicate significant performance difference. Error bars indicate 95% confidence intervals (bootstrapping). **(B)** Performance over individual voxels. Points indicate voxels. Gray points indicate voxels where performance difference is less than $r = 0.1$. Lines indicate median performance over all voxels. **(C)** Projection of performance difference to cortical surfaces of S3.

representational dissimilarity matrices (RDMs) of the stimulus sequence in the test set at different stages of the processing pipeline and averaged them over subjects (Kriegeskorte et al., 2008). Per feature model, this resulted in one RDM for the features, two RDMs for the layer 1 and layer 2 hidden states and one RDM for the predicted responses. We correlated the upper triangular parts of the RDMs with one another, which resulted in a value indicating how much the hidden states of the RNN were modulated by its inputs and how much they modulated its outputs at a given time point. We found a gradual increase in correlations of the RDMs. That is, the RDMs at each stage were more correlated with those at the next stage compared to those at the previous stages. Importantly, the hidden state RDMs were highly correlated with the predicted response RDMs ($r = 0.61$ and $r = 0.93$ for layers 1 and 2, respectively) but less so with the feature RDMs ($r = 0.39$ and $r = 0.21$ for layers 1 and 2, respectively). This means that while the hidden states of the RNN modulated its outputs at a given time point, they were not modulated by its inputs to the same extent at the same time point. This suggests that a substantial part of the output at a given time-point is not directly related to the input at the same time-point, but instead to previous time-points. That is, the RNN learned to use the input history to make its predictions as expected.

Then, we investigated which time points in the input history were used by the RNN to make its predictions. We cross-correlated each hidden state with each stimulus feature, and averaged the cross-correlations over the features, which resulted in a value indicating how much a hidden state is selective to different time points in the input history. The time point at which this value was at its maximum was taken as the optimal lag of that hidden unit. We found that different hidden units had different optimal lags. The majority of the hidden units had optimal lags up to -20 s, which are likely capturing the hemodynamic factors. However, there was a non-negligible number of hidden units with optimal lags beyond this period, which might be capturing other

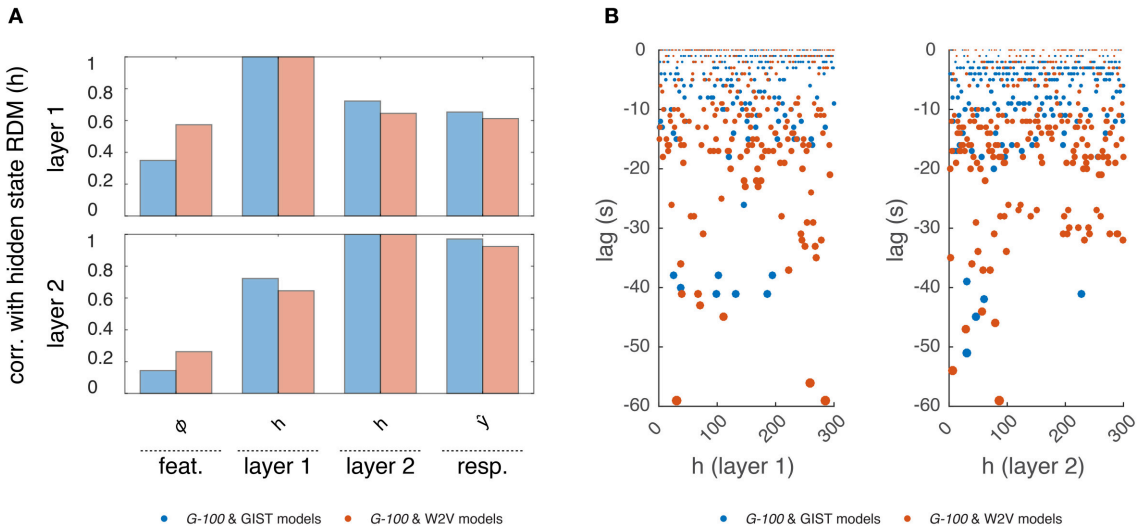


FIGURE 7 | Internal representations of the *G-100* model. (A) Correlation between representational dissimilarity matrices of layer 1 and layer 2 hidden states with each other as well as with those of features and predicted responses. **(B)** Temporal selectivity of layer 1 and layer 2 hidden units. Points indicate lags at which cross-correlations between hidden states and features are highest.

cognitive/neuronal factors or factors related to stimulus/feature statistics. It should be noted that not all hidden units, in particular those with extensive lags, can be attributed to any of these factors, and their behavior might be induced by model definition or estimation. Furthermore, the optimal lags of the hidden units in the *W2V* based model were on average significantly higher than those in the *GIST* based model ($\mu = -9.6$ s vs. $\mu = -4.9$ s, $p < 0.05$, two-sample t-test), which might reflect the differences in the statistics of the features that the models are based on. That is, high-level semantic features tend to be more persistent than the low-level structural features across the input sequence. For example, over a given video sequence, distribution of objects in a scene change relatively slowly compared to that of the edges in the scene.

3.4. Estimation of Voxel-Specific HRFs

Traditionally, models have used analytically derived (Friston et al., 1998) or statistically estimated (Dale, 1999; Glover, 1999) HRFs such as the linear models considered here. Estimation of voxel-specific HRFs is an important problem since using the same HRF for all voxels ignores the variability of the hemodynamic response across the brain, which might adversely affect the model performance. Recent developments have focused on the derivation and estimation of more accurate HRFs. For example, Aquino et al. (2014) has shown that HRFs can be analytically derived from physiology, and Pedregosa et al. (2015) has shown that HRFs can be efficiently estimated from data. Note that, while the methods for statistically estimating HRFs are particularly suited for use in block designs and event related designs, they are less straightforward to use in continuous designs such as the one considered here.

As demonstrated in the previous subsection, one important advantage of the response models in the RNN family is that they can capture certain temporal dependencies in the data, which might correspond to the HRFs of voxels. Here, we evaluate the voxel-specific HRFs that are obtained by stimulating the *G-100* model with an impulse. We used both feature models in combination with the *G-100* model to estimate the HRFs of the voxels where the performance of any model combination was significantly higher than chance (51% of the voxels, $p \leq 0.05$, Student's t-test, Bonferroni correction) (**Figure 8**). The W2V and *G-100* models were used to estimate the HRFs of the voxels where their performance was higher than that of the GIST and *G-100* models, and vice versa.

It was found that the global shape of the estimated HRFs was similar to that of the canonical HRF. However, there was a considerable spread in the estimated delays of responses and the delays of undershoots (median delay of response $= 6.57 \pm 0.02$ s, median delay of undershoot $= 16.95 \pm 0.04$ s), with the delays of responses being significantly correlated with the delays of undershoots (Pearson's $r = 0.45$, $p \leq 0.05$, Student's t-test).

These results demonstrate that RNNs can not only learn (stimulus) feature-response relationships but also can estimate HRFs of voxels, which in turn demonstrate that the nonlinear temporal dynamics that are learned by the RNNs capture biologically relevant temporal dependencies. Furthermore, the variability in the estimated voxel-specific HRFs revealed by the recurrent models might provide a partial explanation of the performance difference between the recurrent and ridge regression models since the ridge regression models use fixed or restricted HRFs, making it difficult for them to take such variability into account.

4. DISCUSSION

Understanding how the human brain responds to its environment is a key objective in neuroscience. This study

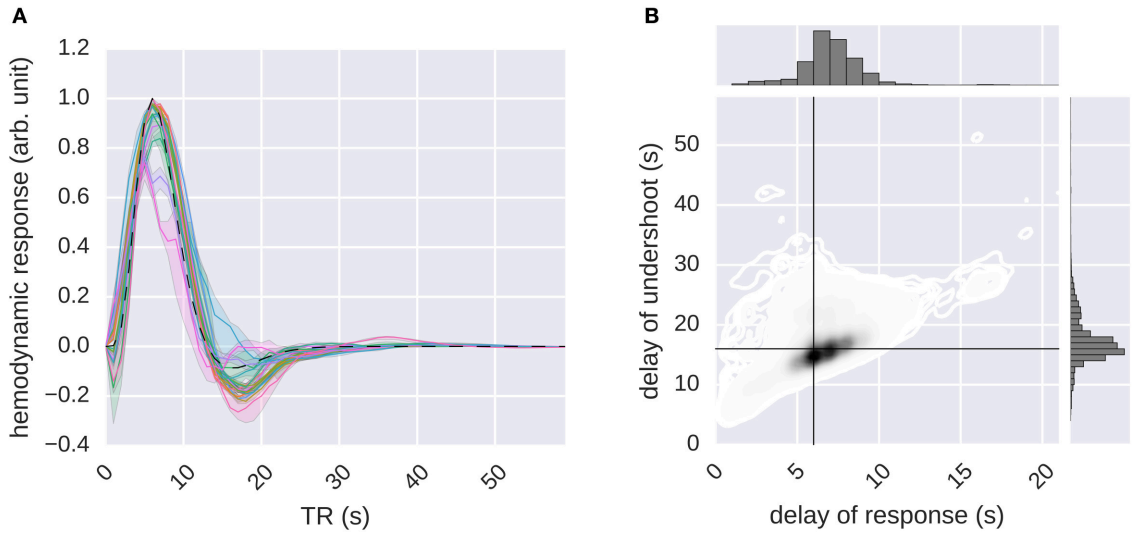


FIGURE 8 | Estimation of the hemodynamic response functions. The *G-100* model was stimulated with an impulse. The impulse response was processed by normalizing its baseline and scale. The result was taken as the HRF. **(A)** Median hemodynamic response functions of all voxels in different areas. Error bands indicate 68% confidence intervals (bootstrapping). Different colors indicate different areas. Dashed line indicates canonical hemodynamic response function. **(B)** Delays of responses and undershoots of all voxels. Black lines indicate canonical delays.

has shown that recurrent neural networks are exquisitely capable of capturing how brain responses are induced by sensory stimulation, outperforming established approaches augmented with ridge regression. This increased sensitivity has important consequences for future studies in this area.

4.1. Testing Hypotheses about Brain Function

Like any other encoding model, RNN based encoding models can be used to test hypotheses about neural representations (Naselaris et al., 2011). That is, they can be used to test whether a particular feature model outperforms alternative feature models when it comes to explaining observed data. As such, we have shown that a low-level visual feature model explains responses in upstream visual areas well, whereas a high-level semantic model explains responses in downstream visual areas well, conforming to the well established early and high-level subdivision of the visual cortex (Mishkin et al., 1983; Naselaris et al., 2009; DiCarlo et al., 2012; Güçlü and van Gerven, 2015a).

Furthermore, RNN-based encoding models can also be used to test hypotheses about the temporal dependencies between features and responses. For example, by constraining the temporal memory capacities of the RNN units, one can identify the optimal scale of the temporal dependencies that different brain regions are selective to.

Here, we used RNNs as response models in an encoding framework. That is, they were used to predict responses to features that were extracted from stimuli with separate feature models. However, use cases of RNNs are not limited to this setting. For example, RNN models can be used as feature models instead of response models in the encoding framework. Like CNNs, RNNs are being used to solve various problems in fields ranging from computer vision (Gregor et al., 2015) to computational linguistics (Zaremba et al., 2014). Internal representations of task-optimized CNNs were shown to correspond to neural representations in different brain regions (Kriegeskorte, 2015; Yamins and DiCarlo, 2016b). It would be interesting to see if the internal representations of task-based RNNs have similar correlates in the brain. For example, it was recently shown that RNNs develop representations that are reminiscent of their biological counterparts when they learn to solve a spatial navigation task (Kanitscheider and Fiete, 2016). Such representations may turn out to be predictive of brain responses recorded during similar tasks.

4.2. Limitations of RNNs for Investigating Neural Representations

RNNs can process arbitrary input sequences in theory. However, they have an important limitation in practice. Like any other contemporary neural network architecture, typical RNN architectures have a very large number of free parameters. Therefore, a very large amount of training data is required for accurately estimating RNN models without overfitting. While there are several methods to combat overfitting in RNNs like different variants of dropout (Hinton et al., 2012; Zaremba et al., 2014; Semeniuta et al., 2016), it is still an important issue to which particular attention needs to be paid.

This can also be the reason why gated recurrent unit architectures were shown to outperform LSTM architectures. That is, the performance difference between the two types of architectures is likely to be caused by difficulties in model estimation in the current data regime rather than one architecture being better suited to the problem at hand than the other.

This also means that RNN models will face difficulties when trying to predict responses to very high-dimensional stimulus features such as the internal representations of convolutional neural networks which range from thousands to hundreds of thousands dimensions. For such features, dimensionality reduction techniques can be utilized for reducing the feature dimensionality to a range that can be handled with RNNs in scenarios with either insufficient computational resources or training data.

Linear response models have been used with great success in the past for gaining insights into neural representations. They have been particularly useful since linear mappings make it easy to interpret factors driving response predictions. One might argue that the nonlinearities introduced by RNNs make the interpretation harder compared to linear mappings. However, the relative difficulty of interpretation is a direct consequence of more accurate response predictions, which can be beneficial in certain scenarios. For example, it was shown that systematic nonlinearities that are not taken into account by linear mappings can lead to less accurate response predictions and tuning functions of V1 voxels (Vu et al., 2011). Furthermore, since more accurate response predictions lead to higher statistical power, the improved model fit afforded by RNNs might make detection of more subtle effects possible. Moreover, when the goal is to compare different feature models, such as the GIST and W2V models used here, maximizing explained variance might become the main criterion of interest. That is, linear models might lead to misleading performance differences between the encoding models in the cases where their assumptions about the underlying temporal dynamics do not hold. In such cases, it would be particularly important to fit the response models as accurately as possible as to ensure that the observed performance difference between two encoding models is driven by their underlying feature representations and not suboptimal model fits. Therefore, RNNs will be particularly useful in settings where temporal dynamics are of primary interest. Finally, combining the present work with recent developments on understanding RNN representations (Karpathy et al., 2015) is expected to improve the interpretations of factors driving response predictions.

4.3. Capturing Temporal Dependencies

RNNs can use their internal memories to capture the temporal dependencies in data. In the context of modeling the dynamics of brain activity in response to naturalistic stimuli, these dependencies can be caused by factors such as neurovascular coupling or stimulus-induced cognitive processes. By providing an RNN with an impulse on the input side, it was shown that, effectively, the RNN learns to represent voxel-specific hemodynamic responses. Importantly, the RNNs allowed us to estimate these HRFs from data collected under a continuous design. To the best of our knowledge this is the first time it has been shown that this is possible in practice. By analyzing the internal representations of an RNN, it was also shown that the RNN learns to represent information from stimulus features at past time points beyond the range of neurovascular coupling. Hence, the predictions of observed brain responses are likely induced by stimulus-related, cognitive or neural factors on top of the hemodynamic response.

4.4. Isolating Neural and Hemodynamic Components

In the introduction, we motivated the use of RNNs as a generic parameterization of any non-linear convolution of stimulus features to hemodynamic responses. Crucially, this could cover both neuronal and hemodynamic convolution. In other words, our black box approach allows for a neuronal convolution of stimulus feature input to produce a neuronal response that is subsequently convolved by hemodynamic operators to produce the observed outcome. This facility may explain the increased cross-validation accuracy observed in our analyses (over and above more restricted models of hemodynamic convolution). In other words, the procedure detailed in this paper can accommodate neuronal convolutions that may be precluded in conventional models.

The cost of this flexibility is that we cannot separate the neuronal and hemodynamic components of the convolution. This follows from the fact that the RNN parameterization does not make an explicit distinction between neuronal and hemodynamic processes. To properly understand the relative contribution of these formally distinct processes, one would have to use a generative model approach with biologically plausible prior constraints on the neuronal and hemodynamic parts of the convolution. This is precisely the objective of dynamic causal modeling that equips a system of neuronal dynamics (and implicit recurrent connectivity) with a hemodynamic model based upon known biophysics (Friston et al., 2003). It would therefore be interesting to examine the form of RNNs in relation to existing dynamic causal models that have a similar architecture.

4.5. Conclusions

We have shown for the first time that RNNs can be used to predict how the human brain processes sensory information. Whereas classical connectionist research has focused on the use of RNNs as models of cognitive processing (Elman, 1993), the present work has shown that RNNs can also be used to probe the hemodynamic correlates of ongoing cognitive processes induced by dynamically changing naturalistic sensory stimuli. The ability of RNNs to learn about long-range temporal dependencies provides the flexibility to couple ongoing sensory stimuli that induce various cognitive processes with delayed measurements of brain activity that depend on such processes. This end-to-end training approach can be applied to any neuroscientific experiment in which sensory inputs are coupled to observed neural responses.

4.6. Data Sharing

The data set that was used in this paper was originally published in Nishimoto et al. (2011) and is available at Nishimoto et al.

(2014). The code that was used in this paper is provided at http://www.ccnlab.net/.

ETHICS STATEMENT

Human fMRI data set that was used in this study was taken from the public data sharing repository http://crcns.org/. The original study was approved by the local ethics committee (Committee for the Protection of Human Subjects at University of California, Berkeley).

AUTHOR CONTRIBUTIONS

UG and MvG designed research; UG performed research; UG and MvG contributed unpublished reagents/analytic tools; UG analyzed data; UG and MvG wrote the paper.

FUNDING

This research was supported by VIDI grant number 639.072.513 of the Netherlands Organization for Scientific Research (NWO).

REFERENCES

Agrawal, P., Stansbury, D., Malik, J., and Gallant, J. L. (2014). Pixels to voxels: modeling visual representation in the human brain. *arXiv:1407.5104 [q-bio.NC]*.

Aquino, K. M., Robinson, P. A., and Drysdale, P. M. (2014). Spatiotemporal hemodynamic response functions derived from physiology. *J. Theor. Biol.* 347, 118–136. doi: 10.1016/j.jtbi.2013.12.027

Cadieu, C. F., Hong, H., Yamins, D. L., Pinto, N., Ardila, D., Solomon, E. A., et al. (2014). Deep neural networks rival the representation of primate IT cortex for core visual object recognition. *PLoS Comput. Biol.* 10:e1003963. doi: 10.1371/journal.pcbi.1003963

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., et al. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv:1406.1078 [cs.CL]*.

Cichy, R. M., Khosla, A., Pantazis, D., Torralba, A., and Oliva, A. (2016). Deep neural networks predict hierarchical spatio-temporal cortical dynamics of human visual object recognition. *arXiv:1601.02970 [cs.CV]*.

Dale, A. M. (1999). Optimal experimental design for event-related fMRI. *Hum. Brain Mapp.* 8, 109–114.

DiCarlo, J. J., Zoccolan, D., and Rust, N. C. (2012). How does the brain solve visual object recognition? *Neuron* 73, 415–434. doi: 10.1016/j.neuron.2012.01.010

Eickenberg, M., Gramfort, A., Varoquaux, G., and Thirion, B. (2016). Seeing it all: convolutional network layers map the function of the human visual system. *NeuroImage*. doi: 10.1016/j.neuroimage.2016.10.001. [Epub ahead of print].

Elman, J. L. (1993). Learning and development in neural networks - the importance of prior experience. *Cognition* 48, 71–99. doi: 10.1016/0010-0277(93)90058-4

Felsen, G., and Dan, Y. (2005). A natural approach to studying vision. *Nat. Neurosci.* 8, 1643–1646. doi: 10.1038/nn1608

Friston, K. J., Harrison, L., and Penny, W. (2003). Dynamic causal modelling. *Neuroimage* 19, 1273–1302. doi: 10.1016/S1053-8119(03)00202-7

Friston, K. J., Holmes, A. P., Worsley, K. J., Poline, J.-P., Frith, C. D., and Frackowiak, R. S. J. (1994). Statistical parametric maps in functional imaging: a general linear approach. *Hum. Brain Mapp.* 2, 189–210. doi: 10.1002/hbm.460020402

Friston, K. J., Josephs, O., Rees, G., and Turner, R. (1998). Nonlinear event-related responses in fMRI. *Magn. Reson. Med.* 39, 41–52. doi: 10.1002/mrm.1910390109

Friston, K. J., Mechelli, A., Turner, R., and Price, C. J. (2000). Nonlinear responses in fMRI: the Balloon model, Volterra kernels, and other hemodynamics. *Neuroimage* 12, 466–477. doi: 10.1006/nimg.2000.0630

Fyshe, A., Talukdar, P., Murphy, B., and Mitchell, T. (2013). "Documents and dependencies: an exploration of vector space models for semantic composition," in *Proceedings of CoNLL* (Sofia).

Glover, G. H. (1999). Deconvolution of impulse response in event-related BOLD fMRI. *NeuroImage* 9, 416–429. doi: 10.1006/nimg.1998.0419

Graves, A. (2013). Generating sequences with recurrent neural networks. *arXiv:1308.0850 [cs.NE]*.

Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., and Schmidhuber, J. (2009). A novel connectionist system for unconstrained handwriting recognition. *IEEE Trans. Patt. Anal. Mach. Intell.* 31, 855–868. doi: 10.1109/TPAMI.2008.137

Gregor, K., Danihelka, I., Graves, A., Jimenez Rezende, D., and Wierstra, D. (2015). DRAW: A recurrent neural network for image generation. *arXiv:1502.04623 [cs.CV]*.

Groen, I. I., Ghebreab, S., Prins, H., Lamme, V. A., and Scholte, H. S. (2013). From image statistics to scene gist: evoked neural activity reveals transition from low-level natural image structure to scene category. *J. Neurosci.* 33, 18814–18824. doi: 10.1523/JNEUROSCI.3128-13.2013

Güçlü, U., and van Gerven, M. A. (2014). Unsupervised feature learning improves prediction of human brain activity in response to natural images. *PLoS Comput. Biol.* 10:e1003724. doi: 10.1371/journal.pcbi.1003724

Güçlü, U., and van Gerven, M. A. J. (2015a). Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. *J. Neurosci.* 35, 10005–10014. doi: 10.1523/JNEUROSCI.5023-14.2015

Güçlü, U., and van Gerven, M. A. J. (2015b). Increasingly complex representations of natural movies across the dorsal stream are shared between subjects. *NeuroImage*. 145, 329–336. doi: 10.1016/j.neuroimage.2015.12.036

Güçlü, U., and van Gerven, M. A. J. (2015c). Semantic vector space models predict neural responses to complex visual stimuli. *arXiv:1510.04738 [q-bio.NC]*.

Hansen, K. A., David, S. V., and Gallant, J. L. (2004). Parametric reverse correlation reveals spatial linearity of retinotopic human V1 BOLD response. *NeuroImage* 23, 233–241. doi: 10.1016/j.neuroimage.2004.05.012

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv:1207.0580 [cs.NE]*.

Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.* 9, 1735–1780. doi: 10.1162/neco.1997.9.8.1735

Huth, A. G., Nishimoto, S., Vu, A. T., and Gallant, J. L. (2012). A continuous semantic space describes the representation of thousands of object and action categories across the human brain. *Neuron* 76, 1210–1224. doi: 10.1016/j.neuron.2012.10.014

Kanitscheider, I., and Fiete, I. (2016). Training recurrent networks to generate hypotheses about how the brain solves hard navigation problems. *arXiv:1609.09059 [q-bio.nc]*.

Karpathy, A., Johnson, J., and Fei-Fei, L. (2015). Visualizing and understanding recurrent networks. *arXiv:1506.02078 [cs.LG]*.

Kay, K. N., Winawer, J., Mezer, A., and Wandell, B. A. (2013). Compressive spatial summation in human visual cortex. *J. Neurophysiol.* 110, 481–494. doi: 10.1152/jn.00105.2013

Khaligh-Razavi, S.-M., and Kriegeskorte, N. (2014). Deep supervised, but not unsupervised, models may explain IT cortical representation. *PLoS Comput. Biol.* 10:e1003915. doi: 10.1371/journal.pcbi.1003915

Kingma, D., and Ba, J. (2014). Adam: a method for stochastic optimization. *arXiv:1412.6980 [cs.LG]*.

Kriegeskorte, N. (2015). Deep neural networks: a new framework for modeling biological vision and brain information processing. *Ann. Rev. Vis. Sci.* 1, 417–446. doi: 10.1146/annurev-vision-082114-035447

Kriegeskorte, N., Mur, M., and Bandettini, P. (2008). Representational similarity analysis - connecting the branches of systems neuroscience. *Front. Syst. Neurosci.* 2:4. doi: 10.3389/neuro.06.004.2008

Leeds, D. D., Seibert, D. A., Pyles, J. A., and Tarr, M. J. (2013). Comparing visual representations across human fMRI and computational vision. *J. Vis.* 13, 25. doi: 10.1167/13.13.25

Logothetis, N. K., and Wandell, B. A. (2004). Interpreting the BOLD signal. *Ann. Rev. Physiol.* 66, 735–769. doi: 10.1146/annurev.physiol.66.082602.092845

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv:1301.3781 [cs.CL]*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. *arXiv:1310.4546 [cs.CL]*.

Mikolov, T., Yih, W.-T., and Zweig, G. (2013c). "Linguistic regularities in continuous space word representations," in *Proceedings of NAACL HLT* (Atlanta, GA).

Mishkin, M., Ungerleider, L. G., and Macko, K. A. (1983). Object vision and spatial vision: two cortical pathways. *Trends Neurosci.* 6, 414–417. doi: 10.1016/0166-2236(83)90190-X

Mitchell, T. M., Shinkareva, S. V., Carlson, A., Chang, K. M., Malave, V. L., Mason, R. A., et al. (2008). Predicting human brain activity associated with the meanings of nouns. *Science* 320, 1191–1195. doi: 10.1126/science.1152876

Murphy, B., Talukdar, P., and Mitchell, T. (2012). "Selecting corpus-semantic models for neurolinguistic decoding," in *Proceedings of First Joint Conference on Lexical and Computational Semantics* (Montréal, QC).

Naselaris, T., Kay, K. N., Nishimoto, S., and Gallant, J. L. (2011). Encoding and decoding in fMRI. *NeuroImage* 56, 400–410. doi: 10.1016/j.neuroimage.2010.07.073

Naselaris, T., Prenger, R. J., Kay, K. N., Oliver, M., and Gallant, J. L. (2009). Bayesian reconstruction of natural images from human brain activity. *Neuron* 63, 902–915. doi: 10.1016/j.neuron.2009.09.006

Nishida, S., Huth, A., Gallant, J. L., and Nishimoto, S. (2015). "Word statistics in large-scale texts explain the human cortical semantic representation of objects, actions, and impressions," in *The 45th Annual Meeting of the Society for Neuroscience* (Chicago, IL).

Nishimoto, S., Vu, A. T., Naselaris, T., Benjamini, Y., Yu, B., and Gallant, J. L. (2011). Reconstructing visual experiences from brain activity evoked by natural movies. *Curr. Biol.* 21, 1641–1646. doi: 10.1016/j.cub.2011.08.031

Nishimoto, S., Vu, A. T., Naselaris, T., Benjamini, Y., Yu, B., and Gallant, J. L. (2014). *Gallant Lab Natural Movie 4T fMRI Data*. Available online at: http://CRCNS.org

Norris, D. G. (2006). Principles of magnetic resonance assessment of brain function. *J. Magn. Reson. Imaging* 23, 794–807. doi: 10.1002/jmri.20587

Oliva, A., and Torralba, A. (2001). Modeling the shape of the scene: a holistic representation of the spatial envelope. *Int. J. Comput. Vis.* 42, 145–175. doi: 10.1023/A:1011139631724

Pedregosa, F., Eickenberg, M., Ciuciu, P., Thirion, B., and Gramfort, A. (2015). Data-driven HRF estimation for encoding and decoding models. *NeuroImage* 104, 209–220. doi: 10.1016/j.neuroimage.2014.09.060

Sak, H., Senior, A., and Beaufays, F. (2014). Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. *arXiv:1402.1128 [cs.NE]*.

Semeniuta, S., Severyn, A., and Barth, E. (2016). Recurrent dropout without memory loss. *arXiv:1603.05118 [cs.CL]*.

Sutskever, I., Martens, J., and Hinton, G. (2011). "Generating text with recurrent neural networks," in *Proceedings of the 28th International Conference on Machine Learning* (Bellevue, WA).

Vu, V. Q., Ravikumar, P., Naselaris, T., Kay, K. N., Gallant, J. L., and Yu, B. (2011). Encoding and decoding v1 fMRI responses to natural images with sparse nonparametric models. *Ann. Appl. Stat.* 5, 1159–1182. doi: 10.1214/11-AOAS476

Wray, J., and Green, G. G. R. (1994). Calculation of the Volterra kernels of non-linear dynamic systems using an artificial neural network. *Biol. Cybern.* 71, 187–195. doi: 10.1007/BF00202758

Yamins, D. L. K., and DiCarlo, J. J. (2016a). Eight open questions in the computational modeling of higher sensory cortex. *Curr. Opin. Neurobiol.* 37, 114–120. doi: 10.1016/j.conb.2016.02.001

Yamins, D. L. K., and DiCarlo, J. J. (2016b). Using goal-driven deep learning models to understand sensory cortex. *Nat. Neurosci.* 19, 356–365. doi: 10.1038/nn.4244

Yamins, D. L. K., Hong, H., Cadieu, C. F., Solomon, E. A., Seibert, D., and DiCarlo, J. J. (2014). Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proc. Natl. Acad. Sci. U.S.A.* 111, 8619–8624. doi: 10.1073/pnas.1403112111

Zaremba, W., Sutskever, I., and Vinyals, O. (2014). Recurrent neural network regularization. *arXiv:1409.2329 [cs.NE]*.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Advantages of publishing in Frontiers

OPEN ACCESS
Articles are free to read, for greatest visibility

COLLABORATIVE PEER-REVIEW
Designed to be rigorous – yet also collaborative, fair and constructive

FAST PUBLICATION
Average 85 days from submission to publication (across all journals)

COPYRIGHT TO AUTHORS
No limit to article distribution and re-use

TRANSPARENT
Editors and reviewers acknowledged by name on published articles

SUPPORT
By our Swiss-based editorial team

IMPACT METRICS
Advanced metrics track your article's impact

GLOBAL SPREAD
5'100'000+ monthly article views and downloads

LOOP RESEARCH NETWORK
Our network increases readership for your article

Frontiers
EPFL Innovation Park, Building I • 1015 Lausanne • Switzerland
Tel +41 21 510 17 00 • Fax +41 21 510 17 01 • info@frontiersin.org
www.frontiersin.org

Find us on