COMPLEX NEURAL COMPUTATION WITH SIMPLE DIGITAL NEURONS

by

Andrew Thomas Nere

A dissertation submitted in partial fulfillment of

the requirements for the degree of

Doctor of Philosophy

(Electrical Engineering)

at the

UNIVERSITY OF WISCONSIN–MADISON

2013

dedication

ACKNOWLEDGMENTS

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

COMPLEX NEURAL COMPUTATION WITH SIMPLE DIGITAL NEURONS

Andrew Thomas Nere

Under the supervision of Professor Mikko H. Lipasti

At the University of Wisconsin-Madison

The desire to understand, simulate, and capture the computational capability of the brain is not a new one. However, in recent years, many advances have been made towards building better models of neurons and cortical networks. Furthermore, a number of high profile projects have proposed, designed, and fabricated *neuromorphic* substrates inspired by the structure, organization, and behavior of biological brains.

This dissertation explores both the *software* and *hardware* elements of these neuromorphic systems. On the *software* side, this dissertation begins with an exploration of the leaky integrate-and-fire (LIF) spiking neuron, and demonstrates that a network composed of simple LIF neurons is capable of simple object recognition and motion detecting tasks. Furthermore, a number of complex neuronal behaviors which significantly extend the computational power of the LIF neuron are identified. This dissertation proposes that an extended LIF neuron model can be used to construct a large scale functional model of the visual cortex with *metastable attractor* dynamics. This hierarchical metastable attractor is capable of invariant object recognition, image reconstruction, working memory tasks, and demonstrates functional integration across multiple modeled regions.

On the *hardware* side, this dissertation investigates the challenges associated with neu-

romorphic hardware in the context of IBM's Neurosynaptic Core. This neuromorphic substrate, composed of simple digital neurons, highlights the *neuromorphic semantic gap* that exists between software models such as the ones described in this dissertation, and the hardware on which they will be deployed. This dissertation demonstrates how this semantic gap can be effectively bridged, and proposes a number of automated techniques for deploying large scale cortical models on the Neurosynaptic Core hardware.

Mikko H. Lipasti

ABSTRACT

The desire to understand, simulate, and capture the computational capability of the brain is not a new one. However, in recent years, many advances have been made towards building better models of neurons and cortical networks. Furthermore, a number of high profile projects have proposed, designed, and fabricated *neuromorphic* substrates inspired by the structure, organization, and behavior of biological brains.

This dissertation explores both the *software* and *hardware* elements of these neuromorphic systems. On the *software* side, this dissertation begins with an exploration of the leaky integrate-and-fire (LIF) spiking neuron, and demonstrates that a network composed of simple LIF neurons is capable of simple object recognition and motion detecting tasks. Furthermore, a number of complex neuronal behaviors which significantly extend the computational power of the LIF neuron are identified. This dissertation proposes that an extended LIF neuron model can be used to construct a large scale functional model of the visual cortex with *metastable attractor* dynamics. This hierarchical metastable attractor is capable of invariant object recognition, image reconstruction, working memory tasks, and demonstrates functional integration across multiple modeled regions.

On the *hardware* side, this dissertation investigates the challenges associated with neuromorphic hardware in the context of IBM's Neurosynaptic Core. This neuromorphic substrate, composed of simple digital neurons, highlights the *neuromorphic semantic gap* that exists between software models such as the ones described in this dissertation, and the hardware on which they will be deployed. This dissertation demonstrates how this

semantic gap can be effectively bridged, and proposes a number of automated techniques for deploying large scale cortical models on the Neurosynaptic Core hardware.

1 INTRODUCTION

1.1 Motivation: Challenges Faced by the von Neumann Architecture

Because of its multi-purpose and generic design, the von Neumann architecture has formed the backbone of nearly every computing system created in the past several decades. However, these traditional computing systems bring with them some fundamental limitations. The von Neumann architecture still is bounded by the *von Neumann bottleneck*; that is, the separation of data (or memory) from the processing elements limits the performance of the computing system architecture. Thus, the processor cannot execute a program faster than it can fetch instructions and data from memory. The addition of cache hierarchies between the processor and main memory has partially alleviated this problem, though the improvement is often greatly affected by the cache size. Other methods have been used to improve processor performance without addressing the von Neumann bottleneck. For many years, simply increasing clock speeds allowed chips to improve performance without addressing the von Neumann bottleneck, though frequency scaling has likely reached its limit [16]. Chip multiprocessor designs have improved processor performance in the absence of further frequency scaling, utilizing the number of resources provided by Moore's law to simply increase the number of cores per chip. However, simple multicore scaling of this type will be ultimately limited by power constraints [37], as well as the parallelizability of the applications that will run on them. Furthermore, as technology

scales, the reliability of devices degrades, as even a slight process variation may alter the behavior of these transistors and limit the performance of a chip [79].

1.2 Inspiration: The Cortex as a Computing Model

While these fundamental limitations will continue to affect von Neumann architectures, it is interesting to note that biology has created a computing system capable of harnessing a large number of inherently faulty components, is highly parallel, energy efficient, and fault tolerant. The brain, and more specifically the mammalian cerebral cortex, has become a frontrunner for inspiring non von Neumann computing systems. Tasks such as learning a new game, recognizing a face, or speech to text are almost trivial to humans, though programming such tasks takes a massive amount of effort. For these reasons, computing models inspired by the cortex have become a promising candidate model for future computing devices. Instead of separating the memory and processing elements, this biological system stores memory and performs computation in the same elements. Neurons perform computation by propagating spikes, and their synapses (or connections between neurons) store memories through particular connectivities and their relative strengths.

Towards these goals, neural networks were introduced as an alternative, biologically inspired computing framework. Neural networks have seen a significant amount of success in various problems, such as image recognition and data classification [78, 74, 90]. With the introduction of recurrent connections (both feedforward information from lower levels, as well as feedback information from higher levels), neural networks are also able to exhibit

many important properties observed in biological brains, such as autoassociative recall, pattern completion, error correction, and activity retention [65].

However, to date, neural networks have fallen short of achieving the complexity and functionality of the biological cortex. Among other limitations, over-simplified neuron models, rigid organizations of multilayered neural networks, and over-simplified learning and plasticity rules have been blamed for these shortcomings [96, 60]. However, as neurobiological and neuroscientific understanding of the brain continues to improve, and the limitations of the von Neumann architecture continue to impact performance, researchers again look to model the cerebral cortex at a more realistic and useful level.

1.3 Spiking Neuron Models and Neuromorphic Hardware

In recent years, research has shifted from simple artificial neuron models to more biologically realistic spiking neuron models. These spiking neuron models are not only useful for large-scale cortical network simulation, but also demonstrate powerful computational capabilities for engineering applications - some of which are explored in this dissertation. This interest in spiking neurons has also inspired the development of many *neuromorphic* hardware implementations, specifically designed for simulating cortical networks or executing neurally-inspired applications. While the enthusiasm for large-scale cortical models and neuromorphic hardware continues to grow, there is little consensus on the degree of biological fidelity that is appropriate for these neuromorphic systems.

1.4 Objectives and Contributions

This dissertation makes contributions on both the *software* and *hardware* sides of neuromorphic systems. In *software*, this thesis focuses on the computing capabilities of spiking neuron models, investigates the complex behaviors exhibited by biological neurons, and justifies their use in large scale models of the cortex. In *hardware*, a number of issues regarding the deployment of cortical models on neuromorphic hardware are explored, in the context of IBM's recent Neurosynaptic Core design.

1.4.1 Computing Capabilities of Simple Spiking Neurons

The first major contribution of this dissertation is an exploration of the computational capabilities of a minimal model of a spiking neuron. Paired with a biologically-inspired learning rule which accounts for bursting activity and global neuromodulators, it is shown that a network composed of even the simplest spiking neuron model is capable of tasks such as invariant object recognition, motion detection, and top-down attentional modulation.

1.4.2 Identifying Useful Complex Neuronal Dynamics

While simple spiking neuron models may demonstrate computational capabilities that exceed traditional artificial neural network techniques, it is clear that biological neurons leverage complex behaviors not captured by such simple models. This dissertation identifies a number of these complex neural behaviors and justifies their inclusion in successful models of the cerebral cortex. The minimal model of the spiking neuron is extended to use

each of these complex neuronal behaviors.

1.4.3 Modeling the Visual Cortex as a Hierarchical Attractor Network

As will be discussed in this dissertation, a number of researchers have proposed that the brain exhibits semi-stable *attractor* states. However, to date, most spiking attractor-based networks have investigated simple decision-making or working-memory based tasks. This dissertation demonstrates that the aforementioned complex neuronal behaviors can be leveraged to compose a large-scale hierarchical attractor-based model of the visual cortex.

1.4.4 Visual Cortex Model Applications

When organized as a hierarchical attractor network, the Visual Cortex model demonstrates noise resilience, pattern completion, and can leverage short-term memory for object recognition tasks. Furthermore, it is demonstrated that the attractor organization allows the model to integrate information processed in different neural regions for a *scene understanding* task.

1.4.5 Addressing the Neuromorphic Semantic Gap

In designing a neurally-inspired hardware substrate, a number of approximations and simplifications must be made, as hardware-modeled neurons are far less flexible than software-modeled neurons. Regardless of the reason or justification, such design decisions introduce a *neuromorphic semantic gap* between software models that leverage complex neural behaviors and hardware implementations that cannot afford to realize every possible

complex neuronal behavior. This dissertation looks at the neuromorphic semantic gap in the context of IBM's Neurosynaptic Core hardware, and demonstrates that complex neural behaviors can effectively be emulated by using circuits composed of the available simple digital primitives.

1.4.6 Automatic Approaches for Deploying Cortical Models on Neuromorphic Substrates

Deploying a large scale cortical model on neuromorphic hardware entails challenges beyond the neuromorphic semantic gap. As hardware cannot feasibly support all-to-all connectivity at large scales, many neuromorphic hardwares opt instead for designs that leverage local connectivity on spatially distributed tiles. This, however, makes it difficult for a cortical model developer to easily deploy a software design, where the constraints of connectivity are not an issue. This dissertation describes a compiler-like tool capable of performing the appropriate network segmentation and routing (while maintaining functional equivalence of the software model) of a large network across multiple Neurosynaptic Cores.

1.5 Related Published Work

This dissertation encompasses these previously published works

- **Bridging the Semantic Gap: Emulating Biological Neuronal Behaviors with Simple Digital Neurons (HPCA - 2013).** This paper describes and addresses the neuromorphic

semantic gap that exists between IBM's Neurosynaptic Core hardware and biologically inspired models of cortical networks [101]. This paper was coauthored by Atif Hashmi and Mikko Lipasti.

- **A Neuromorphic Architecture for Object Recognition and Motion Anticipation Using Burst-STDP (PLoS ONE - 2012).** This article presents a biologically-plausible learning rule for leaky integrate-and-fire neurons and demonstrates their capability at vision related tasks [102]. This paper was coauthored by Umberto Olcese, David Balduzzi, and Giulio Tononi.

- **Neuromorphic ISAs (ASPLOS - 2011).** This paper proposes the need for an abstract level to separate neural algorithms and models from the execution substrate (neuromorphic or traditional computing hardware) on which they are deployed [59]. This paper was coauthored by Atif Hashmi, James Jamal Thomas, and Mikko Lipasti.

1.6 Dissertation Structure

The rest of this dissertation is organized as: Chapter 2 provides background material relating to artificial neuron models, recurrent neural networks, and a brief discussion on the structure and organization of the visual cortex. Chapter 3 highlights a number of high profile neuromorphic hardware projects, and motivates the choice to target IBM's Neurosynaptic Core hardware. Chapter 4 describes an implementation of a leaky integrate-and-fire neuron and demonstrates how it is an effective building block for neural networks capable

of motion detection and object recognition tasks. Chapter 5 extends on the foundational models of Chapter 4. In this Chapter, a number of useful complex neural behaviors are identified and employed to construct a large scale attractor-based model of the visual cortex. Chapter 6 demonstrates the usefulness of the hierarchical attractor approach for several different tasks. Chapter 7 highlights the challenges of deploying such large scale cortical models onto neuromorphic hardware, using the aforementioned Neurosynaptic Core as the target substrate. Chapter 8 concludes the dissertation and discusses future extensions to the research described herein.

2 ARTIFICIAL NEURON MODELS AND NEURAL NETWORKS

This chapter provides relevant background material for the concepts discussed in this dissertation. First, a brief history of artificial neurons and neural networks is given. Next, the motivation for the move to spiking neuron models is discussed, as well as several biologically inspired learning mechanisms of spiking neurons. This chapter also discusses the organization and behavior of the cortex, especially the visual cortex, which has been the inspiration for many computational models built on spiking neurons (including the models described in this dissertation). Finally, several relevant recurrent neural networks and attractor-based networks proposed by other researchers are described.

2.1 A Brief History of Artificial Neural Networks

The ambition of capturing the structure, connectivity, and behavior of biological brains is not a new one. The history of brain-inspired computing begins with the development of the McCulloch-Pitts neuron [91]. In 1943, Warren McCulloch and Walter Pitts proposed that biological neurons were simple threshold units and the inputs and outputs of the neuron were 0's and 1's. The McCulloch-Pitts neuron behaves as a logical gate for linearly-separable inputs; however, it is not capable of solving linearly-inseparable problems such as XOR. While this step was an important milestone in the road to building models of the brain, this model was significantly limited by its simplicity.

One of the next key milestones came about in 1949, when Donald Hebb proposed that

the connections between neurons change over time [62]. Now known as *Hebbian plasticity*, Hebb's work proposed that the connections between neurons change as a function of their firing behavior, stating, "When one cell repeatedly assists in firing another, the axon of the first cell develops synaptic knobs (or enlarges them if they already exist) in contact with the soma of the second cell" [62]. These plastic changes were identified as being essential to learning and memory. Often, the description of Hebbian learning is abbreviated to "neurons that fire together, wire together".

Some time later, the combination of these two fundamental discoveries (the McCulloch-Pitts model and Hebbian learning) led to the development of probably the most widely recognized artificial neuron model, the *perceptron*. Frank Rosenblatt developed the first implementation of the perceptron in the 1960s. Like the McCulloch-Pitts neuron, perceptrons are typically fairly simple units with multiple inputs and a single output. However, leveraging the understanding that connectivities between neurons exhibit a broad range of strengths, the perceptron's inputs are weighted by some value. Furthermore, the perceptron model was not limited as a simple threshold unit, but could utilize a number of different functions, such as Gaussian, piecewise linear, or a sigmoid. The hype surrounding artificial neurons grew, and many researchers actively explored their applications. While these changes broadened the scope of functions the artificial neuron was able to perform, the perceptron was still limited to linearly-separable problems. Ultimately, disappointment in the application scope of perceptrons, paired with criticism from artificial intelligence researchers, lead to the first demise of artificial neurons.

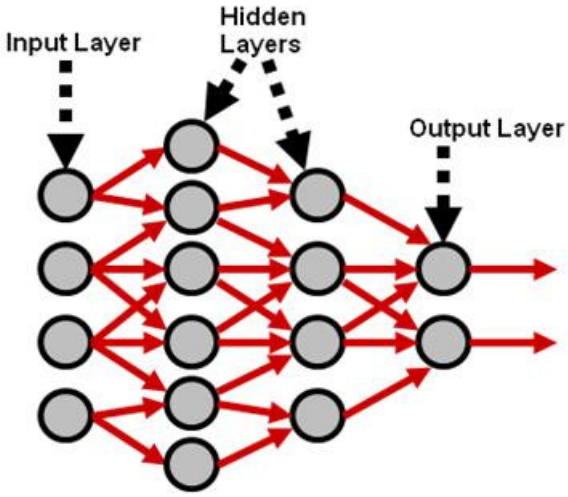


Figure 2.1: A typical multi-layered perceptron (MLP) is composed of an input layer, output layer, and one or more hidden layers.

The late 1970s and early 1980s saw a resurgence of interest in artificial neuron research. One of the key contributors to the renewed interest was the development of the backpropagation learning algorithm. The history of backpropagation is an interesting one, having been formalized by Bryson and Ho in 1969 [20], rediscovered in 1974 by Werbos, and rediscovered again in 1986 by Rumelhart, Hinton and Williams [116]. This supervised learning method proved very useful for training a network composed of multiple perceptron layers (a multi-layered perceptron, or MLP), as shown in Figure 2.1. Furthermore, the MLP, with one or more *hidden layers* (that is, any of the layers between the input and output layers), was capable of solving non-linearly separable problems, such as the XOR problem.

Another key contributor to the renewed interest in artificial neuron research was the development of the Hopfield Network, a *Recurrent Neural Network* (RNN), by John Hopfield [65]. These RNNs are capable of acting as a content-addressable memory, can perform

pattern completion, and are applicable to optimization problems [26]. The Hopfield Network is discussed in greater detail later in this chapter.

In most implementations of MLPs and RNNs, the perceptron-like neuron models are described as *rate-coded* neurons. Because the output of these artificial neuron is typically a value in the range of [0,1], the output is considered to be the average (normalized) firing rate of the neuron (or population of neurons).

While such advancements in neural network research broadened the scope of application, neural network research faded again in the 1990s. Several factors contributed to the diminished enthusiasm. First, the 1990s were dominated by scientific computing; considering that neural networks (such as a backpropagation-trained MLP) provide approximations of functions, they were too inaccurate for broad scientific computing applications. Second, the 1990s saw an explosion in more accurate machine learning techniques, such as Support Vector Machines [31]. Finally, as neuroscientific understanding of the brain grew, it became quite clear that these rate-coded neurons had little in common with biological brains, so researchers developing cortical models also lost interest.

2.2 Spiking Neuron Models

Considering the limitations of classic artificial neural network (ANN) techniques, research has shifted towards more biologically realistic models of neurons and biological neural networks. In particular, the focus has shifted from rate-coded perceptron-like units to a neuron model with a higher degree of biological fidelity: the spiking neuron. Figure 2.2

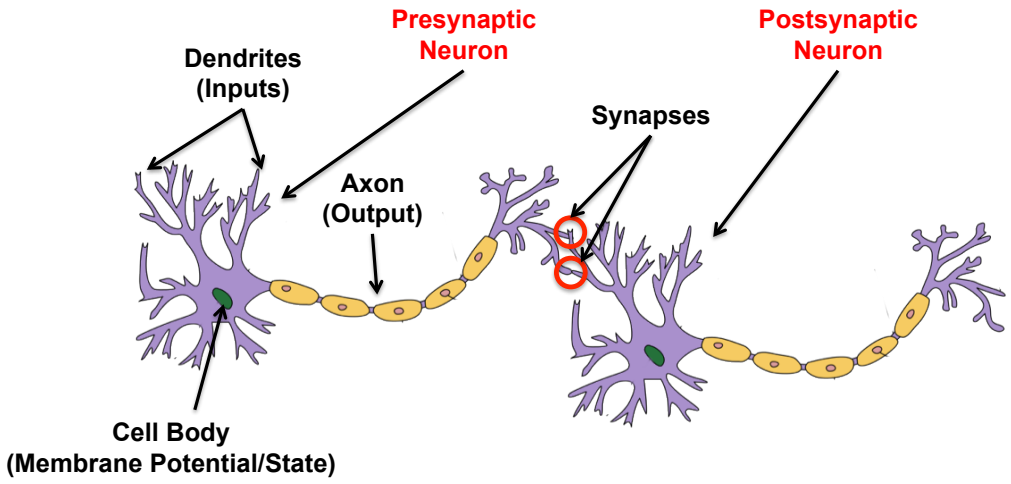


Figure 2.2: Biological neurons integrate the inputs on their dendrites. When the neuron fires, it propagates a spike through its axon, which synapses to the dendrites of other neurons. (Figure adapted from Wikipedia).

highlights some of the major components of a biological neuron: the dendrites (its inputs), the axon (its output), and the cell body (which stores the neuron's current state).

Spiking neuron models have gained interest from those in the neuroscientific community who are interested in accurately modeling the brain, but also among engineers interested in leveraging biological understanding to solve problems. In fact, research has suggested that these more complex and biologically accurate models are computationally more powerful than any of the classic ANN techniques [85].In recent years, many models of spiking neurons have been proposed, ranging from simple integrate and fire models to complex synaptic-conductance based models which use a large set of differential equations to describe the behavior of the neuron and its synapses [69]. However, the commonality between these different models is that neurons communicate via spikes (as opposed to rates) and integrate these spikes over time, giving spiking neurons a concept of time that is

absent from more traditional rate-coded models.

In this section, three classes of spiking neuron model are discussed: the Hodgkin-Huxley model, the Izhikevich model, and the leaky integrate and fire model. While many other models exist, these three are a representative sampling of the varying degrees of biological fidelity and computational complexity associated with spiking neuron models.

2.2.1 The Hodgkin-Huxley Model

The Hodgkin-Huxley neuron model is considered to be one of the most biologically accurate. Based on electrophysiological experiments on the squid giant axon, Hodgkin and Huxley developed a detailed mathematical model to describe how action potentials are generated and propagated in a neuron. The Hodgkin-Huxley model is known as a *conductance-based* model since it accounts for the physical change in conductance of the neuron as a function of its sodium (Na) and calcium (K) channels. Typically, this model consists of four nonlinear ordinary differential equations and many parameters to describe the membrane potential and current flow through ion channels [69]. While these equations were initially developed to describe an entire neuron, even more complex and biologically accurate multi-compartment models have used the same equations to detail biological neurons at an even finer granularity [95]. From the neurobiolical perspective, the Hodgkin-Huxley model clearly and accurately describes these biological neurons in a way that is meaningful and measurable.

While the Hodgkin-Huxley model stands out as a significant neuroscientific achieve-

ment in terms of its biological detail, its computational complexity severely limits its applicability to large scale models and engineering applications. Furthermore, it is unclear from a computational perspective whether the exact details of the ion channels are necessary, or if they are simply artifacts of biology's implementation. Modeling at the level of the Hodgkin-Huxley neuron may be key to building a one-to-one corresponding model of the brain, but the computational requirement of such a model makes it hard to justify its use towards engineering and application specific tasks.

2.2.2 The Izhikevich Model

More recently, the Izhikevich model was developed to reduce the computational complexity associated with conductance-based models like the Hodgkin-Huxley without sacrificing biological fidelity [69]. The Izhikevich model uses just two differential equations and four parameters. Furthermore, the Izhikevich model can be tuned to faithfully exhibit many different neuron behaviors observed in biological experiments such as tonic spiking, bursting, and spike-frequency adaptation, as shown in Figure 2.3.

Clearly, the reduction in computational complexity makes the Izhikevich model much better suited for large simulations, as evidenced by a recent simulation of 10 million thalamocortical neurons [71]. However, the big picture that is still missing is the computational function of each of these behaviors and their role in information processing in the brain.

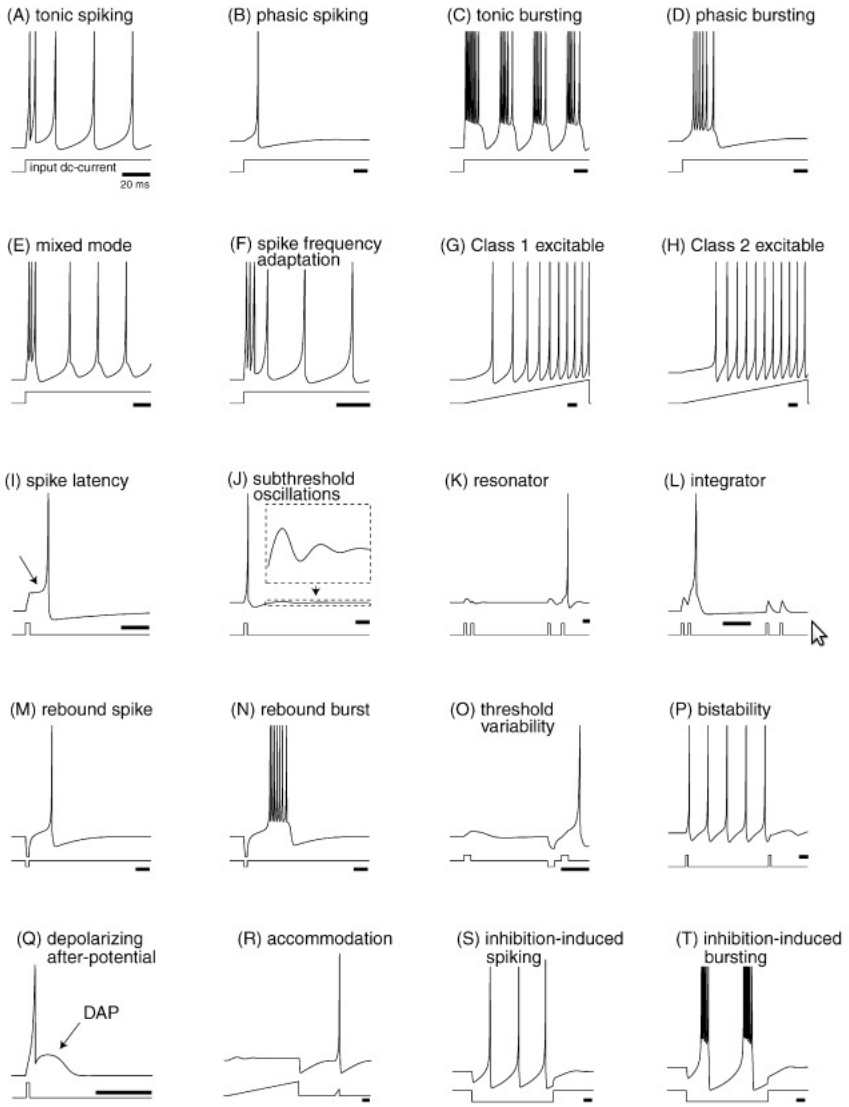


Figure 2.3: The Izhikevich model is capable of mimicking a number of different neuron behaviors that have been experimentally observed. Figure adapted from [69]

2.2.3 The Leaky Integrate and Fire Model

The leaky integrate-and-fire (LIF) neuron model attempts to capture the most basic properties of biological neurons, namely that neurons communicate through spikes and neurons integrate spikes over time. In its simplest implementation, a single differential equation can be used to describe the LIF neuron:

$$\frac{dV}{dt} = \frac{1}{\tau_m}(-V + IR_m) \tag{2.1}$$

Here, V is the current membrane potential of the neuron, R_m is the membrane resistance, I is the input current to the neuron, and τ_m is the time constant of the membrane. If the membrane potential reaches a *firing threshold*, the neuron emits a spike, and is reset to a *reset potential*.

Figure 2.4 illustrates the structural and functional aspects of a LIF spiking neuron. The inputs to a neuron are its dendrites; as shown in Figure 2.2, a neuron has dendritic branches that span in many directions, allowing the neuron to receive many different inputs from other neurons. As shown in Figure 2.4, the LIF neuron model also captures the synaptic efficacy, or weight, of a particular incoming connection. The LIF neuron has three dendrites, each of which as a synaptic weight assigned to it (W_0, W_1, and W_2).

The neuron's cell body, or soma, is typically regarded as the basic processing unit of the LIF neuron. The soma maintains the neuron's *membrane potential*, or current state of the neuron. For clarity, it is helpful to think of a neuron as an electrical device; when a neuron receives inputs, its internal voltage is changed (in the positive direction for excitatory inputs,

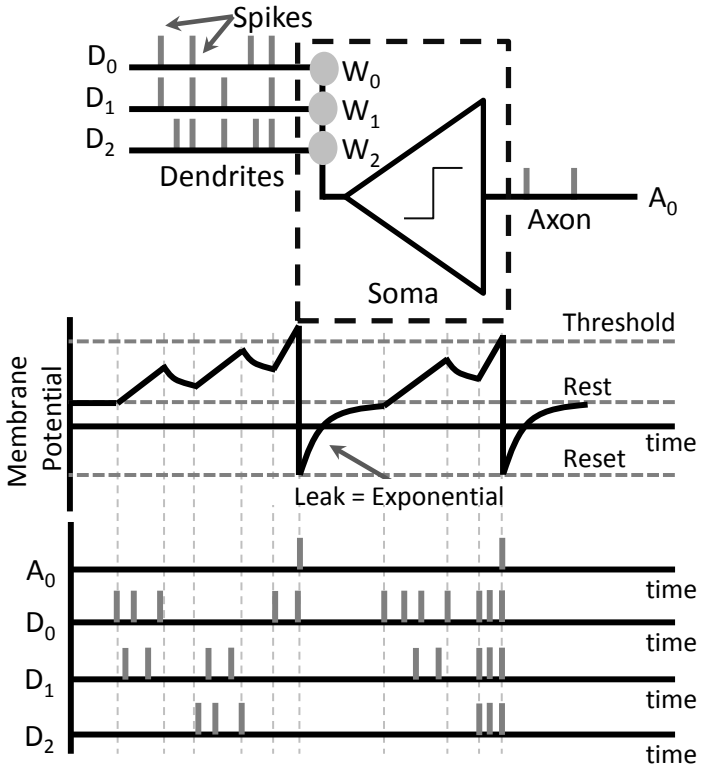


Figure 2.4: LIF neuron–structure and operation.

negative direction for inhibitory inputs). This membrane potential decays as a function of time in the absence of inputs (referred to as the membrane *leak*), and eventually stabilizes at a *resting* voltage. However, if a neuron receives many strong excitatory inputs (at once, or across time at a rate greater than the membrane leak), the membrane reaches a critical *firing threshold*, produces a spike, and is set to a *reset* voltage. This spike travels down the neurons axon (its output), which synapses with the dendrites of other neurons. It should be noted that the communication between neurons (that is, the communication between the output of one neuron and the input of the other) is typically considered to be a chemical, rather than electrical, process. The axon of the presynaptic neuron releases neurotransmitters after an output spike, which in turn, are absorbed by the neuroreceptors on the dendrites

of the postsynaptic neuron.

LIF neuron models, such as the one shown in Figure 2.4, typically have a set of parameterizable variables, such as the synaptic weights corresponding to each of the dendrites, the *reset* and a *resting* membrane potential values, the *firing threshold*, the *leak* of the membrane potential (typically either a linear leak value or a time constant used for an exponential leak), and other stochastic elements and transcendental operations [77]. Figure 2.4 also highlights the role of each of these parameters during the LIF neuron's computation. At each time step, the soma integrates its inputs by evaluating the dot-product of the neuron's synaptic weights (W_0, W_1, and W_2) and the activations on the dendrites (D_0, D_1, and D_2). This value is then added to the membrane potential of the LIF neuron. Next, the updated membrane potential is compared with the *threshold* value, and if the membrane potential exceeds the threshold, the neuron generates a *spike* that propagates down the axon to other neurons. The membrane potential of the neuron is then set to a pre-specified *reset* potential. If the neuron does not generate a spike, the membrane potential *leaks*, which models the tendency of biological neurons to drift towards a *resting* potential.

The LIF neuron can be further extended to include other biological phenomena, such as absolute and relative refractory periods (that is, the shortest amount of time before a neuron can spike again). The LIF neuron can also be extended to show the various spiking behaviors highlighted by the Izhikevich model, albeit at a possibly greater computational complexity. Finally, as will be discussed in Chapter 5, the LIF model can be extended to include complex neural behaviors that extend the models computational capability, such

as voltage-dependent synapses and short-term plasticity.

The choice to target the LIF model in this dissertation stems from a bottom-up approach. With no clear indication what is inherently necessary to a neuron model and what is simply an artifact of biology (ion channel modeling, dozens of spiking behaviors, etc.), beginning with the most basic building block and progressively investigating more complex neuronal behaviors and their computational function significantly simplifies experiments and analysis.

2.3 Biologically Inspired Learning Mechanisms

In neural network and neurobiology literature, various learning paradigms have been hypothesized and/or measured to explain the formation of memories, a sampling of which are presented below.

2.3.1 Hebbian Learning

As was mentioned above, one of the most fundamentally important discoveries of biological neurons is the fact that the connections between them are *plastic* - that is, they change over time. In 1949, Hebb formulated the idea that a synapse should be strengthened if the firing of a presynaptic neuron contributes to the firing of a postsynaptic neuron. It is this plasticity that gives biological brains the capability to adapt to new situations, form new memories, and enhance task performance. Many models of the cortex include some form of Hebbian learning. Furthermore, a number of task-specific engineering applications have

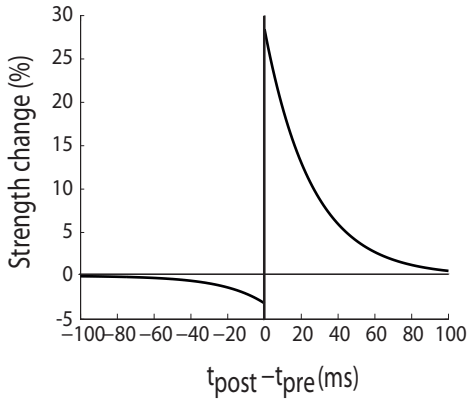


Figure 2.5: Spike-timing dependent plasticity proposes that the relative strength of synaptic change is controlled by the timing between pre and postsynaptic spikes.

also leveraged this biological learning rule. Hebbian learning has been used for object recognition [152], speech recognition applications [39], as well as robotics [146].

2.3.2 Spike Timing Dependent Plasticity

Spike Timing Dependent Plasticity (STDP) is a type of Hebbian learning where, not only is the temporal correlation of spiking behavior important, but also the relative timing of the presynaptic and postsynaptic spikes. If a presynaptic spike is followed by a postsynaptic one within a short span of time, the synapse will be potentiated. Conversely, if the postsynaptic spike is followed by the presynaptic, the connection will be depressed. Figure 2.5 shows an implementation of an STDP learning rule and demonstrates how the magnitude of strength change depends on the delay between the two spikes: the smaller the delay, the greater the change [13, 130]. As shown in the figure, typically the window for synaptic change is fairly short, 10's to 100ms [13].

2.3.3 Variants of STDP

A number of experiments have demonstrated that the precise timing of presynaptic and postsynaptic spike pairs results in synaptic change that closely follows the classic STDP curve of Figure 2.5 [87, 13, 151]. However, other researchers have argued that there is no conclusive evidence that neurons use pair-wise correlations to govern all plastic changes [105]. In some simulations, it has been shown that an STDP rule which leverages *triplets* of spikes better matches the synaptic changes observed in biological experiments [105]. This triplet rule accounts for one presynaptic spike with two postsynaptic spikes, or one postsynaptic spike with two presynaptic spikes. While such a learning rule accounts for more temporal spiking correlations (i.e. three spikes instead of two), it can be argued that this extension marginally increases the accuracy of the STDP learning rule, and an even more biologically accurate model would take into account other factors such as the calcium concentration or postsynaptic membrane potential [105].

Other STDP variants have placed less emphasis on pairs or triplets of spikes, but rather the overall *bursty* behavior of the neurons [102, 32, 80]. Such *burst-STDP* rules place less emphasis on the individual spike-by-spike transmissions, but instead consider that a burst of spikes is inherently more informative, and thus should induce synaptic change. Burst-STDP can include rules that consider presynaptic bursting behavior paired with a single postsynaptic spike [102, 32], or a single presynaptic spike paired with a postsynaptic burst of spikes [80].

2.3.4 Reward Based Learning Paradigms

Classic MLP networks are trained for classification tasks via back-propagation; that is, the correct classification of an object is known and the weights in each layer are adjusted based on this label to minimize the classification error [117]. This form of learning, known as supervised learning, has little biological justification. In biology, it is much more likely that learning is accomplished via unsupervised and/or semi-supervised learning. The STDP and Hebbian learning mechanism are forms of unsupervised learning - plasticity happens as simply a function of the neuron's own behaviors. Semi-supervised learning fits somewhere between; while certainly a "correct answer" cannot be imposed on a biological network of neurons by switching them on and off and propagating an error signal to update synaptic weights, biology does appear to leverage semi-supervised "teacher" signals through global *neuromodulators* [126].

While error-propagating learning mechanisms have little biological justification, it is clear that neuromodulators such as dopamine and noradrenaline can be indicative of a global reward. For example, after an animal has found food, the release of these global neuromodulators helps strengthen recently active neural circuitry (which likely played an active role in finding the food). Temporal Difference (TD) learning, a reinforcement learning rule, has been compared to the dopamine system in the brain [131]. In monkey experiments, a dopamine release was observed after the monkey received a juice reward; over time, the dopamine release migrated to the first indication that the monkey would receive juice [121]. In the absence of a juice reward, dopamine levels would drop below

normal. Similarly, TD learning uses the difference between an expected and received reward to calculate an error function. However, here the network uses a single error (the difference between the predicted and received rewards), as opposed to backpropagation, which requires the precise state of the outcome (i.e. which neurons should be active, which should be inactive) to correct synaptic weights.

Other modeled learning rules have simply considered the release of global neuromodulators to simply bias the sign of Hebbian or STDP learning [102]. Here, a positive global reward biases connections towards potentiation, while a negative global reward biases towards depression. Another possibility is to use an "eligibility trace" variable, which tracks the synapses that have exhibited the pre and postsynaptic firing pairs in recent history [70]. In this learning model, synaptic change is only induced if there is a release of dopamine within a critical time window of a few seconds. At present, the complete role these neuromodulators play in synaptic plasticity is largely unknown; however, the fact that these neuromodulatory systems exhibit such a long reach in their effects is indicative of their importance to memory and learning [55, 126].

2.4 Introduction to the Cerebral Cortex

While the above sections highlight that many different spiking neuron models have been proposed, the ultimate goal of many researchers is to develop scalable models and simulations to better understand the cerebral cortex. The cerebral cortex (or simply - the cortex) is the outermost sheet of neural tissue surrounding the brain of mammals. The cortex of a

typical adult human is composed of around 11.5 billion neurons connected via 360 trillion synapses [114], accounting for approximately 77% of the entire human brain [132]. The cortex itself has been a major focal point of neuroscientific study, as well as the inspiration for engineering applications based on spiking neural networks, for several key reasons.

First, it is believed that the cortex is the part of the brain that is responsible for high level skills, such as mathematics, music, language, perception, and planning, and plays key roles in memory formation and access, attention, and consciousness [76, 36, 35]. Reverse-engineering such processes would fundamentally enhance the capabilities of brain-inspired computing. Second, it has been observed that the cortex is structurally very uniform, composed of millions of nearly identical functional units [98]. This means that a fundamental biologically inspired computing unit (such as the LIF spiking neuron) may be sufficient to perform the broad range of tasks and computations performed by the biological cortex. Finally, the hierarchical organization of the cortex [22, 64, 98] appears to be quite important to its functionality. It has been shown that neurons in higher cortical areas respond with a higher degree of invariance [89], while lower level neurons are more sensitive to particular inputs. This intriguing property shows that the cortex is capable of computing by abstraction in a hierarchical and distributed way [56].

While the neocortex appears to be composed of a very uniform substrate, researchers have identified different regions which process the information of a particular sensory modality. The auditory cortex primarily processes auditory inputs, the somatosensory cortex primarily processes the sense of touch, and the motor cortex performs the planning,

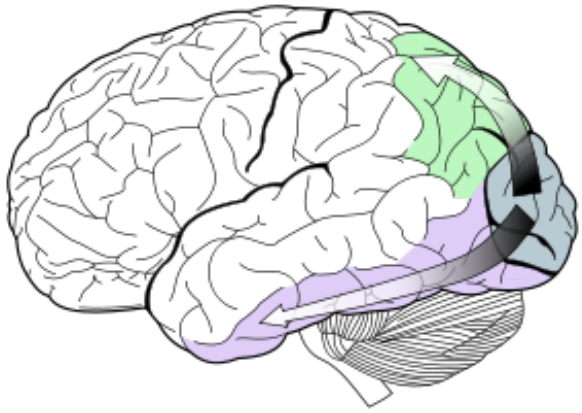


Figure 2.6: The ventral stream, shown in purple, processes the "what", while the dorsal stream, shown in green, processes the "where". Both streams begin in the primary (V1) prestriate (V2) cortices (blue). (Figure from Wikipedia).

control, and execution of motor movements. The models developed in this dissertation primarily focus on the visual cortex: the region of the brain responsible for processing all visual data. Primarily, the focus is on the visual system because it has been studied in greater detail than other regions of the neocortex, and the application scope for modeling the visual cortex is quite large.

2.5 The Visual Cortex

The visual cortex is responsible for processing the sensory input received by the retina. While one may consider vision as a single sense, there are, in fact, multiple streams of processing in the visual cortex. Color, motion, location, and form (that is, the recognition of objects based on shape) are all processed by the visual cortex. While it should not be inferred that each of these information streams is completely disjoint, there is evidence that

distinct regions are responsible for different types of vision processing. For example, as shown in Figure 2.6, much of the processing for object recognition (the "what") is performed by neural regions in the ventral stream, while the processing of object location and motion information (the "where") is performed by neural regions in the dorsal stream.

Figure 2.7 shows a graphical representation of the processing of object recognition in the ventral stream. Visual information is received by the retina, which in turn propagates activations to a region of the thalamus known as the Lateral Geniculate Nucleus (LGN) [73]. For the most part, these LGN cells have center-surround receptive fields which are sensitive to contrast differences. Center-on cells react most strongly to a point of illumination surrounded by darkness, while center-off cells respond strongly to a dark point surrounded by light.

Outputs from the LGN project to the primary visual cortex, also known as the V1. Neurons in the V1 have relatively small receptive fields and are often though of as a tiled set of spatio-temporal filters. In processing form, these V1 cells are thought to respond maximally to very simple features, such as an edge of a particular orientation, and are often modeled using a set of Gabor filter banks [111, 127].

The V1 region sends strong feedforward connections to the prestriate cortex, or the V2. As with the V1, cells in the V2 are similarly tuned for simple spatio-temporal patterns, however, with a larger receptive field and greater degree of pattern complexity [10]. The V1 and V2 regions send strong feedforward connections in two directions: the ventral stream, which is ultimately responsible for object recognition, and the dorsal stream, which

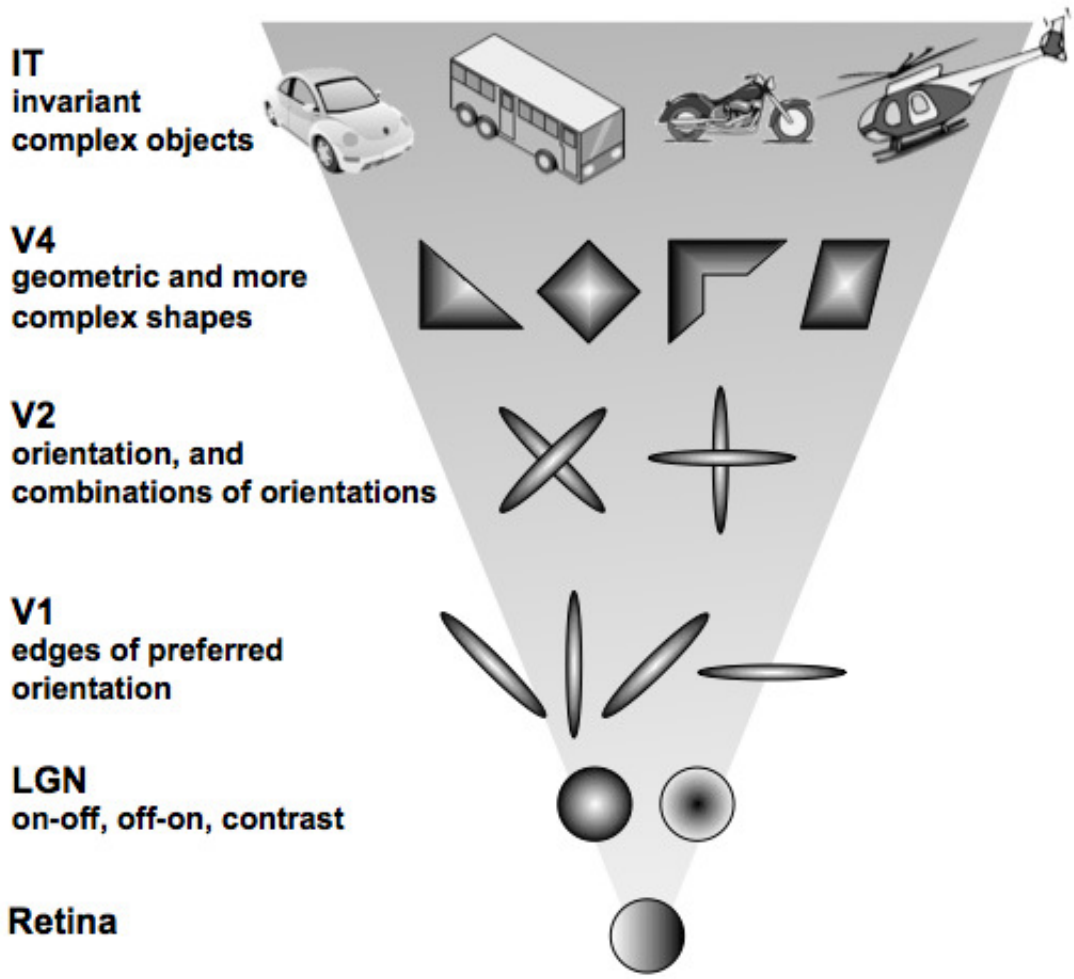


Figure 2.7: A simplified representation of processing in the ventral stream of the visual cortex. Each higher level of the visual cortex responds to progressively more complex shapes and objects. Furthermore, the receptive field grows for each layer of the visual cortex (with respect to the retinotopic input to the system).

processes object location and motion information.

In the ventral stream, the V4 cortical region is believed to respond to objects and features of an intermediate complexity, such as simple geometric shapes. Finally, the top of the ventral stream is the inferior temporal cortex, or IT, where cells respond to complex visual objects and patterns. The receptive fields of IT cells are quite large, allowing the cells to respond to a particular object often with a high degree of invariance [113].

In the dorsal stream, where motion is processed, V1 cells appear to be sensitive to a particular direction of motion within their receptive field [104]. V2 neurons receive feedforward activations from these V1 cells. Both the V1 and V2 cells send strong feedforward connections to the middle temporal cortex, often referred to as the V5 or MT. This region integrates the local motion processing of the lower visual cortex regions, and in turn, processes the global motion of more complex objects [17].

Visual processing information from retina/LGN enters the V1 and flow up the hierarchy, while top-down (or feedback) signals from the IT or MT project to the lower layers in the hierarchy [148]. The full extent of these feedback signals is not yet completely understood, but many of their roles are quite clear. Top-down signalling provides context-dependent pattern completion as well as predictive information from higher cortical regions. Attention is often considered to be predominately driven by top-down signalling as well.

2.6 Recurrent Neural Networks

As was mentioned above, one of the contributing factors to the resurgence of ANN research in the 1980s was the invention of the recurrent neural network (RNN). As any biologically-realistic model of the cortex includes recurrent connectivity between neurons, it is worth discussing some of the RNNs that have inspired the hierarchical attractor-based networks proposed in this dissertation.

2.6.1 The Hopfield Attractor Neural Network

In its most basic definition, an attractor is a state towards which a dynamical system will gravitate over time. It is often the case that many different initial points or states will evolve towards a particular attractor state. In the realm of RNNs, such attractor states play several integral parts. The behavior of such attractor neural networks is often thought of as being essential to working memory in the neocortex, allowing for autoassociative memories to recall and reconstruct from partial matches, and ultimately the ability to generalize memories.

One of the first implementations of this type of RNN was proposed by John Hopfield [65]. In the early 1980s, the Hopfield network demonstrated how recurrent connections could extend the behavior of the neural network, in particular, to store memories. Initially, Hopfield was interested in this type of network to simulate certain properties in physics, so often attractor states are described in terms of *energy*. Typically, Hopfield networks are built around rate-coded neurons such as the perceptron model with a nonlinear activation

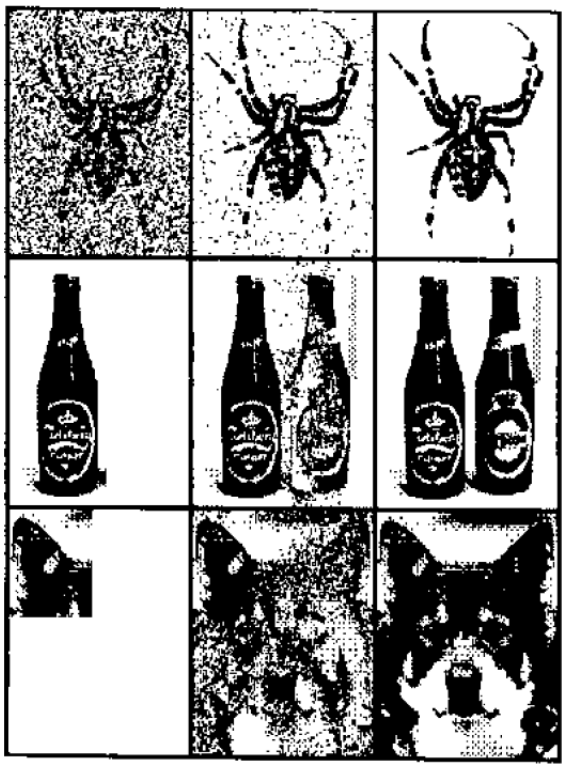


Figure 2.8: Hopfield neural networks can retrieve and restore patterns based on partial images and noisy inputs. The left panels show input images, center shows activity after several iterations, and right panel shows the final convergence of the network. (Figure adapted from [21]).

function. One of the big advantages of this network is its ability to restore or retrieve patterns based on partial matches and corrupted input data, as seen in Figure 2.8.

The absolute memory capacity of a Hopfield network is determined by the number of neurons and connections in the system. However, it has been proven that in a fully connected network, the maximum number of stored pattern that can be effectively recalled is between $N/(2\log N)$ and $N/(4\log N)$, where N is the number of neurons in the network [92]. Furthermore, the Hopfield network has been shown to be useful at storing and retrieving patterns on a discrete time basis - that is, retrieving a pattern at time T has no affect on the

pattern that is retrieved at time $T + 1$. The pattern that is retrieved is simply a product of the pattern (or partial pattern) that is presented to the trained Hopfield network. Thus, the Hopfield network is better suited for discrete inputs, rather than continuous and real-time inputs.

2.6.2 Transient and Metastable Attractor Networks

RNNs have further been extended to include transient state dynamics, or behaviors that vary with the passing of time. With Hopfield-like RNNs, once an attractor state has been reached, essentially the network must be restarted "by hand" in order to leave the current attractor state. While it is clear from biological data that the cortex exhibits attractor state behavior, it is also true that such biological neural systems exhibit dynamical fluctuations between attractor states without requiring a "by hand" reset of the entire network state [43]. Claudius Gros has provided a framework for neural networks with sparse coding and transient state dynamics [51, 52, 53]. Gros proposes that a neural network should be constructed with *clique encoding* - where a clique is a set of neurons that exhibit all-to-all excitatory links. Within the network, this clique encoding corresponds to a *several-winners-take-all* competitive network, where members of the same clique excite each other, while inhibiting or suppressing the neurons outside of the clique. Like the Hopfield network, every neuron is connected to every other neuron in the system, though within any clique, all connections must be excitatory, while between cliques they can be a combination of inhibitory and excitatory. The RNN proposed by Gros also allows individual neurons to

be members of multiple cliques.

Gros' proposed networks also do away with the "reset-by-hand" attribute of Hopfield networks, and instead use neural dynamics which exhibit a transiently stable activity pattern. Once the network is perturbed into a particular attractor state, neurons within the active clique continue to excite each other. A balance of inhibition prevents other cliques from becoming activated at the same time. The transient states are achieved using a slow reservoir variable associated with each neuron, where the activity level of a neuron is dependent on the value of the reservoir. After an extended period of activity, the neurons in the active clique deplete their reservoir and are no longer able to inhibit other neighboring cliques. Finally, the next clique receiving excitatory inputs would be able to enter a transient state of high activity, while the reservoir of the original clique replenishes. Gros describes each of the cliques in the neural network as a *semantic memory*, and proposes that hierarchical memory states can be achieved simply by strengthening particular connections between various semantic memories. For instance, given semantic memories about a person's clothing and that one wants to recall a red shirt, the link (red)-(shirt) would be much stronger than the link (red)-(pants) or (red)-(hat) [50].

However, Gros only considers the transient dynamics and interactions of pre-created cliques - that is, the network defines the connections of and between cliques a priori. In an extended experiment, a very simple artificial environment presents stimuli to the attractor network [53]. Hebbian learning strengthens the connections between an input layer and the presently-active clique in the network, essentially learning an association between a

particular active clique and a particular input stimulus. While such learned connections can help drive the next attractor state from the outside world, it does not provide any information regarding how cliques form, how hierarchical memory states are built, or how sequences may be stored in memory.

Lundqvist et al. have proposed a modular attractor memory using modeled cortical columns [83, 82]. These authors, among others (including Hashmi et al. [57, 58]), have proposed that minicolumns, rather than individual neurons, may be the basic functional unit of computation in the neocortex. Hypercolumns contain a number of minicolumns which exhibit a high degree of interaction. It has been proposed that local lateral inhibition between minicolumns can promote a competitive learning paradigms and enforce soft winner-take-all dynamics [83, 82, 57, 58]. Within a minicolumn, Lundqvist et al. set the connectivity strengths of excitatory pyramidal neurons and inhibitory basket neurons to replicate biological measurements. Finally, long-range excitatory connections synapse with minicolumns in other hypercolumns, forming a very modular and distributed attractor network.

In spiking neural network models, long-timescale neural receptors are often modeled to allow attractor states to stabilize. [83, 82]. Such long-timescale modulation can be thought of as a biological explanation for the reservoir variables proposed in Gros' work. However, Lundqvist et al. showed, as a result of their distributed attractor network, such long-timescale neurotransmitters may not be necessary so long as short-term excitatory synapses and recurrent connectivity are significantly increased [82]. Ultimately, Lundqvist

et al. show this distributed modular network is able to perform the desired attractor behaviors in a very biologically realistic model, such as stability of attractor states, pattern completion, and pattern rivalry.

However, the limitations of Lundqvist et al.'s model are similar to those pointed out with Gros' attractor networks. Each clique, or attractor state, is hand-coded a priori: the minicolumns that encode the same memory are simply hard-wired together via long range excitatory connections. While it is easy to accept that the high level of recurrent connectivity within in a minicolumn is simply part of the inherent structure of the cortex, it is unclear how long range connections between minicolumns and hypercolumns are formed. Furthermore, while Lundqvist et al. propose that each minicolumn is a distinct feature of a particular pattern being stored by the attractor state, they fail to investigate multiple attractor states that have some features, or minicolumns, in common. If the cortex is truly using distributed minicolumns to store the various features of a particular memory, it would likely re-use minicolumns to encode shared memory features efficiently.

In spite of these limitations, the modular attractor networks proposed by Lundqvist appear to be quite robust, grounded in biological data, and effective as realistic building blocks for large scale neural networks. Furthermore, Lundqvist's model, while implemented with highly detailed Hodgkin-Huxley neurons, shows a surprising amount in common with the learning algorithm proposed by Hashmi et al. which models hypercolumns and minicolumns at a much more abstract level. However, many of the basic features are the same, including pattern completion capabilities, competitive learning, and

distributed representations of learned patterns.

2.6.3 Liquid State Machines

In attractor networks like those described above, some number of stable states are stored and can be retrieved by the neural network, whether the states are transiently stable or must be reset through direct stimulus. However, *Liquid State Machines* (LSMs) attempt to model the real-time computing capabilities of biological neural systems without storing and retrieving particular attractor states. LSMs could be considered to have only one stable attractor state - the resting state [86]. LSMs utilize recurrent connections, not to reach a stable state, but rather to capture a *fading memory* about the current and previous inputs to the system. The *liquid* in the name comes from the analogy of dropping a stone into a body of water, where the result of the input (the dropped stone) is converted into a spatio-temporal pattern observed in the liquid.

In the typical case, the recurrent connections of the LSM are randomly generated [86]. However, the statistical structure of the recurrent connections (such as the ratio of local to distal connections, ratio of excitatory to inhibitory connections, and synaptic weight distributions) are often based on biological structures, such as the connectivity within a minicolumn [86]. Furthermore, it has been shown that utilizing more complex neural behaviors and dynamics can further improve the performance of the LSM [54].

Typically training and plasticity are not used or necessary within the LSM. Rather, in most implementations, a memoryless output (or readout) layer connects to the LSM,

which is trained to produce a desired output at a particular time. The output layer must be trained with a supervised learning algorithm; however this training does not need to account for the temporal aspects of the learned task, as the LSM itself performs all temporal processing. The output layer of the LSM is often composed of simple linear readouts, a layer of perceptrons (in cases where the states produced by the LSM are linearly separable), or even multi-layered perceptrons (in cases where the states produced by the LSM are not linearly separable).

The computational capability of LSMs paired with easily-trainable readout layers have made LSMs attractive and quite successful for speech recognition and computer vision applications [72, 120, 141, 40]. However, several criticisms of LSMs arise. First and foremost, LSMs don't necessarily assist in explaining the functionality and connectivity of the brain. At best, the LSM is a "black box" that may perform useful computations necessary for current and previous inputs; however, there is no guaranteed or simple method to determine exactly what is happening or what computations are being performed. Another criticism is that LSMs are not well suited for fault tolerance. If certain neurons on the LSM become faulty, output units must be retrained with the altered liquid machine [61].

2.6.4 Recurrent Neural Networks Summary

This section discusses several high-profile recurrent neural network architectures, though many other networks that make use of recurrent connectivity have been proposed. However, each of the RNNs discussed here have influenced the design of the Visual Cortex model

of this dissertation. Specifically, the goal is to develop a model that is capable of vision related tasks, exhibits metastable attractor states that flexibly change with new inputs, and ultimately shows functional integration across different neural regions. These ideas will be further explored in Chapters 5 and 6.

2.7 Summary

This chapter details and explains a number of different concepts important to the understanding of this dissertation. The desire to implement artificial neurons to represent their biological counterpart dates back to the 1940s. Since then, artificial neurons have evolved considerably; presently, research focuses on spiking neuron implementations and biologically-inspired learning rules. Going forward, the simple leaky integrate-and-fire spiking neuron model is used in this dissertation. At the level of neural networks, many architectures have been proposed, including multi-layered perceptrons, Hopfield and attractor networks, and liquid state machines. This dissertation particularly focuses on modeling the structure and abilities of the visual cortex as a recurrent neural network, which will be described in detail in later chapters.

3 NEUROMORPHIC HARDWARE

The computational advantages of spiking neurons, paired with the distributed and hierarchical processing of the cortex, has led researchers to pursue hardware substrates inspired by the brain. These *Neuromorphic Architectures* have attracted interest in recent years, and a number of different designs have been proposed, designed, and fabricated [27, 118, 2, 11, 125, 67, 93, 128]. Despite the fact that the primitives provided by these substrates, their implementations (analog, digital, or mixed circuits), and application scope may vary significantly, the majority of neuromorphic architectures share some common properties. These include modeling the basic processing elements after biological spiking neurons, designing the functional memory store after biological synapses, and leveraging massive amounts of fine-grained parallelism inspired by the parallel processing of biological brains. This section briefly introduces and describes several high profile neuromorphic substrates, and motivates the selection of IBM's Neurosynaptic Core as the target substrate for this dissertation.

3.1 Neurogrid

Neurogrid is a neuromorphic computing substrate that was specifically designed to simulate cortical networks with up to 1 million neurons and 6 billion synapses [27]. Impressively, the Neurogrid performs this simulation on a power budget of 5W [3].

Each Neurogrid contains 16 Neurocores. These Neurocores contain 64K analog circuit

neurons as well as an on-chip RAM for storing the cortical network's connectivity. While the hardware neurons themselves are implemented as analog circuits, the communication between neurons is performed in the digital domain. The Neurogrid hardware neurons contain two sub-cellular compartments, which allows cortical network modelers to emulate many of the nonlinear neural behaviors observed in biological studies. One of the key features of the Neurogrid design is the large number of user-definable parameters. These parameters allow the Neurogrid to simulate multiple neuron behaviors (e.g. bursting or bistable), synapse types (e.g. excitatory or inhibitory), and synaptic strengths.

In all, 18 binary and 61 graded parameters per hardware neuron cover a broad range of cortical neuron models. The Neurogrid is programmed via a Python script which defines the neuron connectivity and various parameters, and an interactive GUI allows researchers to visualize the behavior of their cortical models.

3.2 The BrainScaleS Neuromorphic Processor

The BrainScaleS Project (formerly the Fast Analog Computing with Emergent Transient States, or FACETS project) is driven by the goal to understand information processing in the brain at both the individual neuron level and the level of functional brain areas [4]. This research includes *in vivo* biological experiments, neural simulation on supercomputers, and the development of a novel mixed-circuit neuromorphic processor.

The BrainScaleS hardware implements a leaky integrate-and-fire neuron with conductance based synapses using analog circuits. Analog circuits allow the neuron hardware to

operate in the continuous-time domain. Furthermore, a number of analog neuron implementations have been shown to accurately mimic many of the behaviors observed in their biological counterparts [67, 93, 128]. The synaptic weights and the network interconnect of the BrainScaleS hardware are digital designs [118]. These neurons can be configured to exhibit both short-term and long-term plastic changes [119].

The basic building block of BrainScaleS neuromorphic hardware is the High Input Count Analog Neural Network (HICANN) chip [118]. Each HICANN chip has a total of 512 neuron membrane circuits and 128,000 synapses. The hardware can be configured to operate as 8 analog neurons with 16,000 input synapses each, or 512 analog neurons with 256 input synapses each. To scale to large network simulations, the BrainScaleS hardware is optimized for *wafer-scale* integration. Each wafer contains 384 HICANN chips, meaning a system of 196,608 neurons and 49,152,000 synapses can be deployed on a single wafer (assuming a defect-free fabrication of the wafer).

This project has developed the PyNN programming model to provide researchers with a common API that runs on various neuronal simulators such as NEST and NEURON, as well as the BrainScaleS/FACETS neuromorphic hardware [19]. A highly detailed model of the early visual system (retina, LGN, and primary visual cortex) which matches a large amount of biological data and recording has been implemented for the BrainScaleS project [1].

By taking an analog approach to neuron implementation, the BrainScaleS hardware operates at timescales thousands of times faster than biological neurons. Biological neurons exhibit slow firing behavior (typically 10's to 100 Hz) as well as a slow membrane time

constants (typically in the 10's of milliseconds). These timescales require capacitive devices too large for an integrated circuit design; hence, the BrainScaleS hardware targets operating frequencies many times greater than biological neurons. As such, the BrainScaleS hardware appears to be a best suited studying the brain through accelerated simulation, as opposed to performing real-time computation at biological timescales.

3.3 SpiNNaker

SpiNNaker (which stands for Spiking Neural Network Architecture) is a novel computer architecture specifically designed for simulating neurons. The SpiNNaker project has targeted applications in neuroscientific simulation, robotics, and nondeterministic (and massively parallel) computing. Rather than building hardware neurons, this architecture simulates the neurons on traditional von Neumann processors, but focuses on a system organization and interconnect that is optimized for the communication patterns displayed by biological brains.

The target SpiNNaker system will contain a million ARM9 cores and a total of 7Tbytes of RAM. 18 cores plus 128Mbytes of off-die SDRAM are grouped into a node called a System-in-Package (SiP). While the cores have a variety of ways to communicate, the predominant form of communication in the SpiNNaker architecture is through a packet-switched network. Because biological neurons and brains are largely asynchronous and stochastic in their behavior, the SpiNNaker architecture is able to maximize communication bandwidth by doing away with memory coherence, synchronization, and determinism

- principles that are considered essential for the correct behavior of traditional parallel von Neumann machines [45]. When operating, the SpiNNaker system consumes 90kW of electrical power, several orders of magnitude below simulation on a modern supercomputer.

One of the key goals of the SpiNNaker project is to simulate large-scale cortical models at biological timescales. To target such an ambitious goal, the SpiNNaker hardware is optimized for simulating the simplified *point neuron model*. This neuron model fits under the broader category of leaky integrate-and-fire neurons, but explicitly ignores any detail of the dendritic structure. Rather, input spikes are simply weighed by the synaptic strength and added to the membrane potential. While the ARM9 cores lack support for division and transcendental functions, the point neuron model can be expressed as a simple polynomial equation compatible with the available primitives of the processor [108]. In the debate over biological fidelity (what neural behaviors and dynamics are essential, and which are artifacts?), the SpiNNaker project is biased towards a simpler solution of artificial neurons [45].

3.4 IBM's Neurosynaptic Core

In the context of this dissertation, IBM's recent Neurosynaptic Core design is chosen as the neuromorphic substrate [94, 11, 106]. The goal of the Neurosynaptic Core is to create a system capable of interpreting real-time inputs at a biologically realistic clock rate. The final neuromorphic design seeks to model 10 billion neurons and 100 trillion synapses [147], and seeks to rival the brain in terms of area and power consumption. In this section, the

choice to target the Neurosynaptic Core is motivated, and the functional behavior of the Neurosynaptic Core is described in detail.

3.4.1 Why Target the Neurosynaptic Core?

The broad range of neuromorphic hardware implementations, several of which have been described above, may make it difficult to select an appropriate substrate on which to deploy software models of the cortex. Each neuromorphic hardware design has a unique set of advantages, such as power consumption, degree of biological fidelity, or flexibility of the neuron and synapse models.

From an initial glance, the choice of IBM's Neurosynaptic Core is not an obvious one. While the Neurosynaptic Core is optimized for dynamic power consumption on the same order as biological neurons [94], characteristics such as binary-only synapses, reliance on conventional CMOS digital logic and memory, and lack of transcendental functions (such as an exponential decay membrane leak), at first, appear questionable. However, the arguments below justify that each of these choices is both sensible and attractive for a neuromorphic design that targets areal and power efficiency at biological timescales.

First of all, avoiding exotic device, fabrication, or design technologies dramatically improves the odds of success, since the design team is able to leverage existing tools and design flows, ongoing improvements in digital CMOS technology, and existing design expertise carried over from prior projects. Second, this expertise in digital CMOS design contributes significantly towards the key goals of areal efficiency and low power operation.

Third, prior theoretical work shows that neurons with binary synapses are fundamentally no less powerful than real-valued synapses [124]. This final design point allows synaptic connectivity to be captured by a space-efficient SRAM, where each row represents the output of a neuron, each column represents the input of a neuron, and a set bit indicates a connection between two neurons. These design choices are clear and easy to justify in the context of this work.

The final choice–reliance on fixed-point arithmetic with no support for transcendental functions–is a bit harder to justify. Considering the design goals of areal efficiency and low power consumption, support for floating point computation, division, and nonlinear operation is quite difficult, as realizing many of the inherent nonlinearities of biological components requires highly complicated digital circuits [97]. However, there is no clear evidence from the neuroscientific literature that a lack of complex neuronal behaviors will (or will not) compromise the computational capability of the neuromorphic hardware. Therefore, the Neurosynaptic Core design appears to side with the SpiNNaker project on the debate of biological fidelity; that is, simpler spiking neuron models are valid until it is proven that more complex neuronal behaviors are more than artifacts of biology. In this dissertation (Chapters 5 and 6) , it is argued that a number of these more complex neuronal behaviors provide significantly enhanced computational and signalling abilities, a contradiction to the design philosophy of the Neurosynaptic Core hardware. However, as will be discussed in Chapter 7, many of these important behaviors can be effectively emulated on the Neurosynaptic Core hardware. Therefore, it can be argued that the design

decision for simple linear-operation-only neurons is a good one, assuming that the majority of the applications and models deployed on the hardware make sparing use of these more complex neuronal behaviors and nonlinear functions.

3.4.2 Description and Operation of the Neurosynaptic Core

The Neurosynaptic Core is composed of simple linear leaky integrate-and-fire (LLIF) neurons. Rather than modeling the leak mechanism using an exponential decay, a LLIF processing element simply subtracts a linear leak value from its membrane potential at each time step. IBM has developed two Neurosynaptic Core designs based around the LLIF neuron: one with [125], and one without online learning [94, 11]. This dissertation focuses in particular on the Neurosynaptic Core which does not feature online learning, but rather, targets low dynamic energy consumption.

The Neurosynaptic Core design incorporates a number of configurable components and parameters. Earlier publications regarding the Neurosynaptic Core [94] indicated 1024 input axons per core, while later sources [106] have indicated only 256 input axons. In this dissertation, it is assumed that each Neurosynaptic Core is composed of 256 LLIF processing elements, 256 axons, and a 256x256 SRAM crossbar memory for synapses, as shown in Figure 3.1. This chip does not incorporate online learning capabilities. To distinguish these hardware-implemented LLIF processing elements from the general concept of LIF neurons, they will subsequently be referred to as Neurosynaptic Core Neurons, or NCNs. Each of the 256 axons is responsible for propagating the input spikes to the system. In terms of

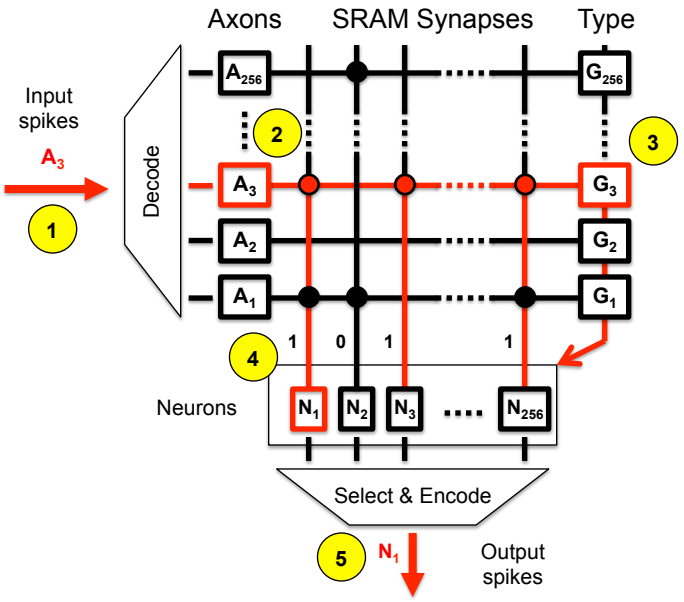


Figure 3.1: IBM's Neurosynaptic Core [11].

these digital neurons, one can think of a spike to simply mean that the axon is held to a value of '1', while a non-spike means the axon is held to '0'. These axons can be configured to route in spikes from off-core elements or to other NCN's residing on the same core, allowing recurrent connectivity. Here, it is assumed that each NCN is assigned a single output axon; therefore, an NCN can be parameterized to either route its axon recurrently back to the same Neurosynaptic Core, or it can project its axon off core. The SRAM crossbar is a configurable set of binary synapses between the incoming axons (horizontal lines) and the digital neuron's dendrites (vertical lines), as shown in Figure 3.1. A set bit (shown as a circle in the figure) indicates that a particular NCN must integrate the spikes arriving on the corresponding axon, while an unset bit (no circle) means that this particular NCN should ignore that axon's behavior.

One of the primary power-saving design points of the Neurosynaptic Core is its event-driven operation, where the propagation of spikes drive the operation of the system [68]. However, each Neurosynaptic Core receives a clock tick at 1kHz to discretize the neuron dynamics in 1 millisecond time steps. Not only does the slow clock rate contribute to the low power consumption of the hardware design, but it also ensures that the Neurosynaptic Core operates on biologically-realistic timescales. Incoming spikes are decoded and routed to their appropriate axon buffer A_j. At each time step t, a NCN cycles through all of its input axons buffers. If the axon buffer contains an incoming spike for this particular time step, the axon j is held at a value of '1', causing a readout of the appropriate SRAM row.

When the chip is initially configured, each axon j is assigned an axon type G_j which can be parameterized to one of three values (0, 1, 2). Likewise, each NCN i has three parameterizable synaptic weight values (S_i^0, S_i^1, S_i^2) which correspond to each of the three axon types. In this way, each NCN in the system can be configured to have both excitatory and inhibitory connections of different strengths, but the overall crossbar can be a dense 256x256 bit SRAM. This binary connectivity, between axon j and NCN i, is defined as W_{ij}.

Each NCN also has a parameterized linear leak term L, which models the tendency of neurons to drift to a resting potential. This linear leak term may include stochastic behavior [106], allowing for a more diverse range of neural behaviors. Finally, the present state of the neuron, its membrane potential, is captured by the variable $V_i(t)$. Taken all together, at each time step, the membrane potential of NCN i is updated by:

$$V_i(t+1) = V_i(t) + L_i + \sum_{j=1}^{K} A_j W_{ji} S_i^{G_j} \tag{3.1}$$

Here, $K = 256$, the total number of axon inputs of the Neurosynaptic Core. When $V_i(t)$ exceeds a parameterizable firing threshold θ, the NCN produces a spike output, and its membrane potential V_i is reset to 0.

Figure 3.1 also demonstrates the operation of the Neurosynaptic Core for a single time step. Prior to time step t, an incoming spike A_3 arrives at the Neurosynaptic Core (indicated by the yellow "1" in Figure 3.1). At the time step t, the NCN cycles through each of its incoming axons (indicated by the yellow "2" in Figure 3.1). An incoming spike in the buffer of axon 3 is detected (i.e. A_3 is set to 1, due to the aforementioned arrival of the spike). As a results, axon 3 is pulled to a value of 1 (as indicated by the red horizontal line), which causes a readout of line 3 from the synapse SRAM. At the same time, the axon's assigned type (as defined by G_3) is read (indicated by the yellow "3" in Figure 3.1). In the figure, the SRAM has been configured so that NCNs N_1, N_2, and N_M synapse with axon 3, and thus, receive values of '1' from the SRAM, while NCN N_2 receives a value of '0'. Next, each of the neurons which receives a value of '1' from the SRAM increments its membrane potential by the appropriate synaptic weight value, $S_1^{G_3}$, $S_2^{G_3}$, and $S_M^{G_3}$ respectively (indicated by the yellow "4" in Figure 3.1). It should be noted that $S_1^{G_3}$, $S_2^{G_3}$, and $S_M^{G_3}$ may all be configured to different values (e.g. 100, -2, and 15). Recent publications have also indicated that these synaptic weights can also be configured as stochastic [106]. After the NCNs have integrated all the spikes in this time step, NCN N_1 has crossed the firing threshold θ and has produced

an output spike that will become an input spike to the chip at time step $t + 1$ (indicated by the yellow "5" in Figure 3.1).

3.4.3 The Neurosynaptic Core with Online Learning

IBM has also implemented a second Neurosynaptic Core design which includes two simple forms of stochastic online learning [125]. Most of the neuron parameters for the online-learning hardware are the same as for the non-learning hardware described above. However, this alternative design features all-to-all synaptic plasticity, which allows each bit cell of the SRAM to be not only read during operation, but written as well, thus emulating online-learning by altering synaptic connectivity.

For the rest of this dissertation, the choice to target the non-learning Neurosynaptic Core is based on two primary reasons. First, depending on the cortical network model of interest, the all-to-all synaptic plasticity provided by the learning Neurosynaptic Core may be significantly underutilized. Here, one must consider that to support all-to-all online learning, every neuron and axon must incur additional area and power overheads to support transposable SRAMs, spike-time counters, and linear feedback shift registers [125]. Second, the learning Neurosynaptic Core implements only four simple learning schemes (Hebbian, anti-Hebbian, STDP, and anti-STDP) which modify a single binary synapse [125]. As will be shown in Chapter 7, using circuits composed of multiple NCNs, each of these learning rules can be emulated. Furthermore, using the composable NCN circuit approach, many more aspects of the STDP and Hebbian learning schemes can be parameterized and

novel learning mechanisms can be developed.

3.5 Summary

This chapter presents four very different neuromorphic substrates. While each design is motivated by the same goal of emulating the brain, the focus and approach of each neuromorphic design is quite different. As evidenced by the different hardware methodologies of the Neurogrid, FACETS, SpiNNaker, and SyNAPSE projects, it is still unclear what future neuromorphic hardwares will look like, but it is clear they will be significantly different from the the traditional von Neumann architecture. However, considering the scope and the impact of the projects mentioned here, it is likely just a matter of time before these types of neuromorphic systems become commodity components as well. The rest of the dissertation considers IBM's Neurosynaptic Core as the targeted neuromorphic substrate.

4 MODELING SPIKING NEURONS AND BIOLOGICALLY INSPIRED LEARNING MECHANISMS

This chapter describes in detail the linear leaky integrate-and-fire (LLIF) spiking neuron model which serves as the basic functional unit for the networks described in this dissertation. Furthermore, it also elucidates the biologically inspired learning and homeostatic renormalization mechanisms believed to govern plasticity in the brain. The LLIF neuron, paired with the aforementioned learning mechanisms, is used to construct a minimal neural network architecture based on several regions of the visual cortex. Finally, the computational capability of this minimal spiking neuron model is explored in several experiments relating to object recognition and motion detection.

4.1 Leaky Integrate-and-Fire Spiking Neuron Model

As was discussed in Chapter 2, many models of spiking neurons have been proposed, ranging significantly in their biological fidelity and computational complexity. However, this work begins with a minimal model of a biological spiking neuron, a choice that has a few key motivations. First, by starting with a minimal model, the behavior, structure, and dynamics of the spiking neuron (and populations of spiking neurons) can be better understood, which simplifies analysis and discussion. Second, by taking a bottom-up approach, the computational usefulness of more complex neuronal behaviors (such as short-term plasticity, discussed in detail in Chapter 5) can be better understood in a step-by-step

process. Finally, choosing a minimal neuron model improves its viability for deployment on first generation neuromorphic substrates, which, as was discussed in Chapter 3, must make a number of approximations and simplifications over biological neurons and brains.

To further simplify the LIF neuron implementation, a linear leak factor is applied at each time step, as opposed to the more traditional exponentially decaying membrane potential. The following equations describe the basic operation of the LLIF model:

$$V_i(t) = V_i(t-1) + \sum_{j=1}^{K} A_j(t-1) W_{ji} - L_i \tag{4.1}$$

$$A_i = \begin{cases} 1, \text{if } V_i(t) >= \theta_i \\ 0, \text{ otherwise} \end{cases} \tag{4.2}$$

$$V_i(t) = \begin{cases} V_i^{reset}, \text{if } A_i == 1 \\ V_i(t), \text{ otherwise} \end{cases} \tag{4.3}$$

Here, $V_i(t)$ is the neuron's membrane potential. On each simulated time step, the neuron integrates its inputs by taking the dot-product of the spikes that were produced at the previous time step ($A_j(t-1)$) with the synaptic weight of the connection (W_{ji}). The total input is added to the membrane potential from the previous time step ($V_i(t-1)$), and the linear leak is applied (L_i). After the updates to the membrane potential have been

completed, the membrane is compared to the neuron's firing threshold (θ_i), as shown in equation 4.2. If the neuron fires, its membrane potential is reset to its reset voltage (V_i^{reset}) - otherwise the membrane potential remains unchanged (equation 4.3).

In this simple implementation, the membrane potential $V_i(t)$ is kept to a value of 0 or above. In the absence of input spikes, the leak parameter L_i drifts the membrane potential of the neuron to a resting voltage, V_i^{rest}. When V_i^{rest} is nonzero, the linear leak factor must also support leak reversal; that is, if a strong inhibitory input or the reset voltage V_i^{reset} set the membrane potential below V_i^{rest}, the sign of the leak parameter is inverted until the membrane potential reaches (or crosses) the resting membrane potential.

4.2 Learning with Bursts of Spikes

Biological neurons communicate through spikes. Therefore, it can be assumed that the only way for a neuron to communicate the importance of its output is to modulate its firing rate over time. However, spikes do not come for free; it has been proposed that the largest component of a neuron's energy consumption is devoted to spiking [12]. Considering metabolic and energy constraints, it is reasonable to assume that the brain as a whole should minimize the number of spikes needed to convey information, and save more "bursty" behavior for truly important events.

In this model, the plasticity mechanism exploits the idea that burst events are expensive and therefore important. This simply means that synaptic potentiating events are weighted by the relative burstiness of the input. In detail, the level of presynaptic burstiness of each

connection is modeled as:

$$\mathrm{burst}_{\mathrm{pre}}(t+1) = \mathrm{burst}_{\mathrm{pre}}(t) + \lambda_{\mathrm{inc}} \cdot \mathrm{spike}_{\mathrm{pre}}(t) - \lambda_{\mathrm{dec}} \tag{4.4}$$

Here $\mathrm{burst}_{\mathrm{pre}}$ is a trace buffer which captures the inputs burstiness, where $\mathrm{spike}_{\mathrm{pre}}(t)$ is 1 if the presynaptic neuron fired and 0 otherwise. λ_{inc} defines the increment of the burst trace on every spike and λ_{dec} defines the decay in the burst trace at every time step.

The burst-STDP learning rule is applied each time the postsynaptic neuron spikes. Therefore, each synaptic weight is updated according to the following equation:

$$w(t+1) = w(t) + k \cdot \mathrm{spike}_{\mathrm{post}}(t) \cdot \mathrm{burst}_{\mathrm{pre}}(t) \tag{4.5}$$

Here k represents the learning rate.

It should be noted that the learning rule, as described above, results only in the potentiation (i.e. strengthening) and never the depression (i.e. weakening) of synapses. The choice of a potentiation-based learning rule is two-fold. First, in keeping with the goal of a minimal neuronal model, it simplifies the learning rule. Second, a growing body of literature has demonstrated that learning during wake is dominated by long-term potentiation (LTP) [30, 142, 49, 29]. While a potentiation-dominated learning rule can destabilize network activity and lead to a saturation of connections, this potentiation is counterbalanced by a homeostatic renormalization mechanism described below.

4.3 Value Dependent Learning

While biological studies have demonstrated that synaptic potentiation is driven by pre and postsynaptic neuron spiking, the brain also uses neuromodulators such as dopamine and noradrenaline to influence plasticity [126, 55, 70, 102]. These types of neuromodulatory systems can be considered much more global in nature, as their effect reaches across multiple brain regions [121]. Since these types of neurotransmitters typically indicate an important event with a broad reach, it is likely that they are a key component of the learning mechanisms in the brain.

The learning mechanisms of this model are extended to include the role of neuromodulators for *value dependent learning*. Similar to other simulation studies, modeling these neuromodulators allows for signalling reward and punishment during learning tasks [129, 102]. In this model, a global reward system evaluates only whether the network has correctly (or incorrectly) responded to the learning task at hand. In turn, a global reward (or possibly a punishment) mechanism is invoked to modulate the appropriate synaptic connections contributing to the response. This type of reward learning is much more biologically plausible than traditional back-propagation learning methods, as it only requires a single reward signal such as dopamine or noradrenaline for correct behaviors.

A correct response is rewarded by multiplying the burst-STDP plastic change calculated in Equation 4.5 by a positive constant, while an incorrect response is multiplied by zero - allowing the value dependent learning to be implemented with minimal change over the proposed plasticity mechanisms described above. Subsequently, the synapses which are

modulated by value dependent learning are updated according to:

$$k = g(r, t)\, k_0 \tag{4.6}$$

Here k_0 is the predetermined value of the learning rate (see the previous section) and $g(r, t)$ is the modulation performed by value-gating, which depends on both rewards r and time t. In summary, when the network responds correctly to a stimulus, k is positive, otherwise k is zero, resulting in no synaptic change.

4.4 Homeostatic Renormalization

A growing body of literature has indicated that the average strength of synapses, as well as neural activity, increase during wake and decrease during sleep [107, 49, 143, 144, 81] in a self-regulatory fashion [15]. The burst-STDP learning rule described above captures the average gain in synaptic strength - which in turn increases neural activity. While a global punishment signal like the one described above can be modified to also include long-term depression (LTD), this model instead utilizes another mechanism to ensure that synaptic potentiation is kept in balance. *Homeostatic Renormalization* of synaptic strength has been hypothesized to take place during sleep [136, 137] and may be responsible for counterbalancing the predominance of potentiation occurring during waking time.

Beyond simply preventing runaway synaptic potentiation (that is, uncontrolled all-to-all strong connectivity), studies have shown its importance to memory and performance

related tasks. Hill et al. showed with a simple linear rescaling of synaptic strength, the signal-to-noise ratio was significantly improved for a simple motor-memory task, in both human subjects and a simulated neural network [63]. Furthermore, Olcese et al. demonstrated with an activity-dependent homeostatic renormalization mechanism an improvement in sequence learning and memory consolidation in a large scale model of the thalamocortical system [103]. The renormalization implemented for this model follows the method proposed by Hill, where the synaptic strengths are linearly rescaled offline so the strongest connection is set to '1'. The i^{th} connection w_i is therefore changed according to:

$$w_i^{renormalized} = w_i / w_{max} \tag{4.7}$$

Renormalization promotes memory consolidation by setting strengthened connections to a value of one and progressively weakening unused ones, until they become negligible. Under these plasticity rules, the plastic synapses of a neural network are renormalized simultaneously after a predetermined number of simulation steps - comparable to a period of sleep after a long period of waking.

In the context of porting a trained neural network to digital hardware, this type of normalization process can be quite useful. Important synapses will often be used, so their connection strength will be high, while other synapses will gradually move towards zero. Since over time the synapse strengths will gravitate towards *very strong* or *very weak*, a simple threshold function can be used to binarize the synapse values. The major benefit to this binarization is that such simple synaptic weights are much more easily realized in

actual hardware like the Neurosynaptic Core.

4.5 Preliminary Spiking Model of the Visual System

This section describes a foundational model of the visual cortex composed of LLIF spiking neurons. Using the neuron model and learning mechanisms described above, this model demonstrates how a simple LLIF spiking neurons can achieve motion detection, simple invariant pattern recognition, noise resilience, and top-down attentional modulation.

Figure 4.1 details the organization of this foundational model of the visual cortex, inspired by the anatomical and functional connectivity of several different brain areas. Through these modeled processing streams of the visual cortex, this neural network is able to learn translation invariant representations of simple shapes, as well as different trajectories of motion, and ultimately produces learned motor outputs for each particular stimulus. The following subsections describe the modeled processing streams and network capabilities in greater detail.

4.5.1 Shape Categorization Module

It is well understood that the brain uses automatic abstraction to create robust and invariant representations of objects, people, and places [56]. As was described in Chapter 2, object recognition in the visual cortex primarily occurs in the V1, V2, V4 and IT regions. The Shape Categorization Module uses simple spiking LLIF neurons for the same task: translation-invariant recognition of simple objects in the environment.

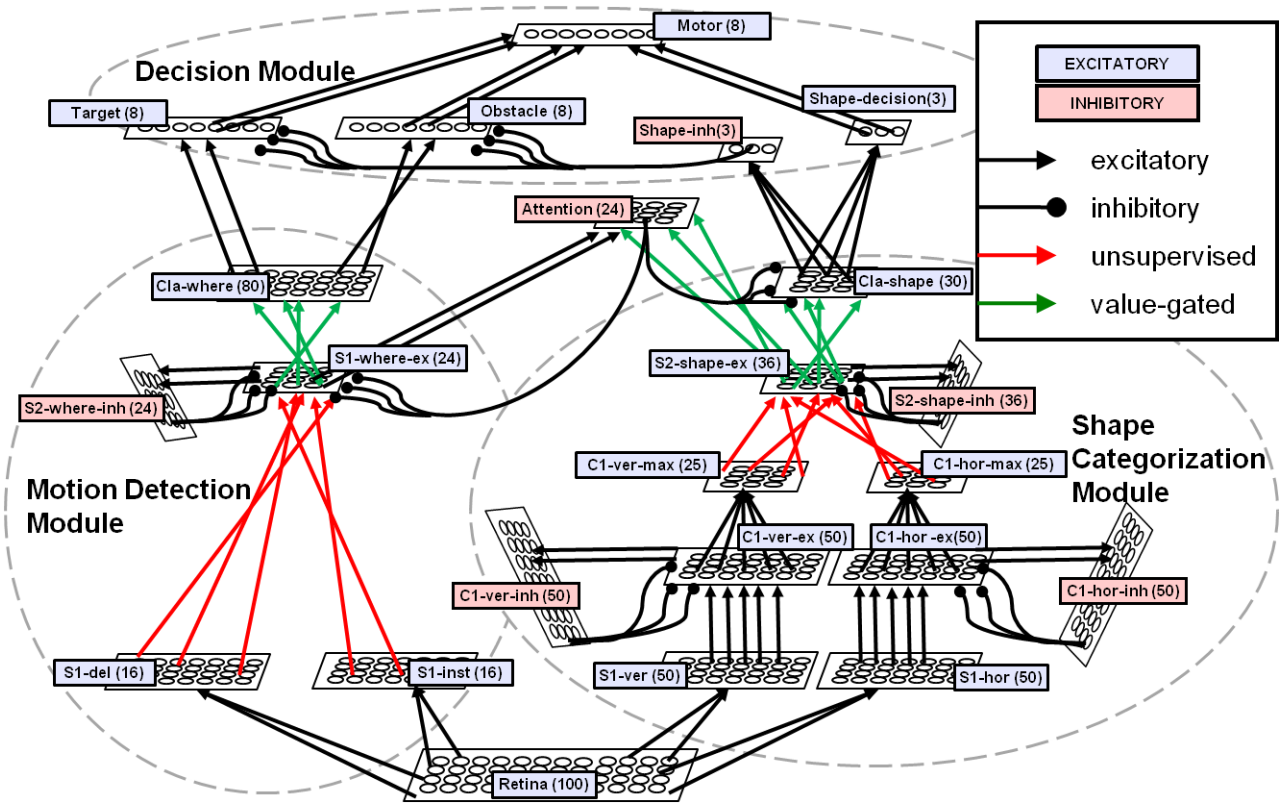


Figure 4.1: The architecture the foundational model of the visual cortex. Each layer is depicted as a grid of cells (dimensions do not correspond to actual layer sizes). Parallel or converging connections represent topographic connectivity without or with dimensionality reduction; overlapping connections represent random connectivity. Subsystems consisting of multiple neural layers are grouped with dashed lines. All hard wired synaptic connections are black, and burst-STDP/homeostatic renormalization learned connections are colored.

The general organization of the Shape Categorization Module borrows from Poggio's HMAX [111, 127] algorithm as well as Masquelier's STDP implementation of HMAX [90]. Like these visual system models, the Shape Categorization Module alternates simple cells (S) which elicit a spiking response for their preferred input and complex cells (C) which provide translation invariance by using a max-pooling operation over a population of simple cells. The overall architecture of the Shape Categorization Module utilizes multiple layers of hierarchical processing, with the top level of the hierarchy being a classifier (see Figure 4.1).

As in the V1 area of the visual cortex, neurons in the first layer ($S1-ver$ and $S1-hor$) have small receptive fields and respond maximally to simple features, such as an edge of a particular orientation [112]. Neurons which find a preferred edge will spike in response, and these activations propagate to the second layer (C1). The C1 layer (in both the vertical and horizontal edge processing streams) uses three neuron populations to perform the max-pooling operation over the S1 cells. The first layer of excitatory neurons ($C1-ver-ex$ and $C1-hor-ex$) are connected 1-to-1 with the S1 cells below them, firing whenever their corresponding S1 cell has fired. This population is recurrently connected with an inhibitory cell population ($C1-ver-inh$, $C1-hor-inh$) which imposes a weakly-enforced winner-take-all (WTA) competition among the excitatory C1 cells. This is not considered a strictly-enforced WTA, since nothing prevents two (or more) C1 cells from firing at the exact same time; however, the recurrent inhibition puts the first neurons that fire at an advantage. Finally, the excitatory C1 cells propagate converging connections to the $C1-ver-max$

and $C1 - hor - max$ populations. This simple circuitry allows the most distinguishing lower level features to propagate spikes up to progressively more invariant regions (due to converging connectivity).

The S2 layer ($S2 - shape - ex$), in turn, has an even broader receptive field, and uses online learning to recognize simple combinations of features. That is, when a particular stimulus is presented to the network, composed of many edges of different orientations, the S2 uses the burst-STDP and homeostatic renormalization rules to learn the conjunction of different edges. Similarly to the C1 neuron layer, the S2 is composed of multiple types of neurons, using inhibitory cells to create a weakly-enforced WTA competitive network to encourage different S2 cells to learn different features. In this way, the S2 cells that first respond to the activations of the C1 neurons will reinforce their connectivity via the burst-STDP learning rule. Again, it is not a strictly-enforced WTA, since nothing prevents two (or more) S2 cells from learning the same feature when initialized from random synaptic weight strengths. Because of the initial random connectivity of this neural layer, the S2 neurons' receptive fields are not topographically organized as in the lower level neural layers. While this connectivity means that a certain level of detail is lost to the upper levels of the shape categorization module, the examples below show that for smaller scale neural networks, performance is not affected.

Finally, the uppermost layer of the shape categorization module is the classifier ($Cla - shape$), which learns to invariantly recognize a particular object anywhere in the visual environment, similar to the Inferior Temporal (IT) region of the visual cortex. A pool of

ten neurons are dedicated to each of the classes to be learned. The connections between the S2 layer and $Cla - Shape$ layer are learned using the value dependent burst-STDP mechanism described above, as opposed to the C1 to S2 connections, which do not require a global reward signal. While the S2 neurons can utilize simple inhibitory competition to learn unique features, the $Cla - shape$ layer must be trained to classify translated inputs as the same item.

Preliminary experiments showed that for linearly-separable classes, a single neuron is sufficient to categorize the input. However, learning rates were improved when using a pool of classifier neurons. The main advantage of using a neuron pool is that the response of each neuron in the pool will depend on its initial weak and random connectivity, which in turn will elicit rewards or punishment. The more neurons that are activated, the more the reward system induces the value dependent burst-STDP and drives the classifier layer to strengthen appropriate connections. Furthermore, other works have shown that populations of simple neurons are quite capable of learning non-linearly separable classes given appropriate constraints and learning mechanisms [123].

4.5.2 Motion Detection Module

The simple spiking LLIF building blocks can also be used to model the regions of the visual cortex which detect and process motion. As shown in Figure 4.2, the motion of a simple line of pixels can be detected using a simple circuit of LLIF spiking neurons. First, three neurons (E_1, I_1 and C_1) are used to detect a contrast change within a receptive field. When

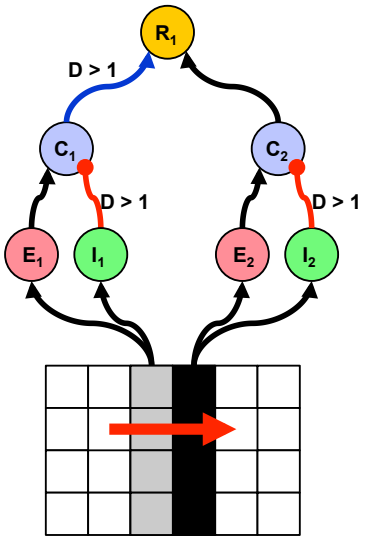


Figure 4.2: A simple circuit for motion detection.

a moving line enters the receptive field of neuron E_1, it spikes in response. The inhibitory neuron I_1 shares the same receptive field. Both neurons project their output axons to the contrast detection neuron C_1, but the connection from the inhibitory neuron requires a longer delay. In this way, C_1 will fire for the positive contrast change (caused by excitatory neuron E_1), but will be silenced by the inhibitory neuron shortly after.

In the neighboring receptive field, another three neurons (E_2, I_2 and C_2) similarly detect contrast changes. Finally, a motion detecting cell (R_1) receives excitatory inputs from both of the contrast detecting cells, with one of the inputs using an axonal delay (blue axon). In this example, a contrast change on the left (from the past) paired with the current contrast change on the right signifies the detection of a rightward moving line.

The Motion Detection Module in Figure 4.1 builds on this basic circuit of neurons. The $S1-inst$ and $S1-del$ populations each implement the same contrast detecting circuitry as shown in Figure 4.2; however, the outputs of $S1-inst$ are received by the $S1-where-ex$

population instantaneously, while the outputs of $S1-del$ include an axonal delay. The necessity for both the $S1-inst$ and $S1-del$ populations stems from the choice to only give one value of axonal delay per neuron, hence the duplication.

Neurons in the $S1-where-ex$ population use the burst-STDP learning rule to detect coincidence firings of contrast changes in nearby receptive fields. The $S1-where-ex$ is recurrently connected with the inhibitory population $S1-where-inh$ to again implement the weak WTA circuitry to encourage different cells to detect different directions of local motion. In this way, cells in the $S1-where-ex$ region learn to fire for local motion in one of the four cardinal directions (up, down, left, right). As with the Shape Categorization Module described above, the lower levels of the Motion Detection Module rely on topographically organized receptive fields. The overlapping receptive field of each $S1-where-ex$ neuron covers a 4x4 area of the retina. These cells show no preference for a particular object, but simply fire when visual features are moving in its preferred direction.

The $Cla-where$ layer uses the value dependent burst-STDP learning rule to classify each of the local motions into one of the four cardinal directions. In this sense, the $Cla-where$ cells are able to learn longer range trajectories and directions of motion over a much broader receptive field (and in the case of this simple model, the entire visual field). Neurons in the $Cla-where$ layer are pooled into groups of ten neurons. After sufficient training with reward and punishment, the appropriate $Cla-where$ neuron pool fires consistently as an object moves across the retina. This architecture is a simple yet effective

model of cortical motion detection systems.

4.5.3 Attention Module

Because of the vast amount of raw data the retina provides to the visual cortex, it is useful to have a mechanism to discriminate important features and objects from other distractors. *Attention* accomplishes this important task by providing top-down signalling to the lower level visual processing areas through *back connections* to place emphasis on important features and filter out distractors. It is known that in the cortex, feedback connections are just as numerous as the forward connections, though their full functionality is far from understood. In this foundational model, these feedback connections are used to provide focus on the objects in the visual receptive field determined to be the most important, while silencing the neurons firing for distractor objects.

As can be seen in Figure 4.1, the attention module receives excitatory input from both the Shape and Motion Modules. The Attention Module receives hardwired one-to-one connections from the motion detection module, while the synapses received from the shape module are initially random and strengthened through value dependent burst-STDP. In this way, the Attention Module learns through reward the associations of a particular important shape and direction of motion. After learning, inhibitory neurons project outputs back to the motion detection and shape categorization modules to silence neurons that are propagating distractor shapes and their motions.

4.5.4 Decision Module and Motor Outputs

Finally, the decision module is responsible for determining the motor output reaction to the state of the visual environment. This decision module helps the network cope with the presence of noise in the input environment. In particular, it evaluates whether the classifications performed by the Shape and Motion Detection Modules are consistent over time or just sporadic activations (and thus likely to be erroneous detections caused by noise). In this preliminary network, one population of neurons responds to *target* objects, and another for *obstacle* objects. Both populations are trained through supervised value-gated burst-STDP learning, and respond maximally as evidence of each object (and its direction of motion) is accumulated over time. This Decision module drives the motor outputs of the entire neural network, either to approach target objects or avoid obstacles. Presently, such motor outputs are only considered at a very high decision level, though future extensions to the system will likely entail performing more detailed, fine grained, and self-correcting motor outputs. Biologically one can consider the simple decision and motor output module to be similar to the high level decisions an animal makes to seek food or avoid predators, while lower level motor outputs actually orchestrate the motion and minute actions.

4.6 Experimental Results

This section highlights several experiments that test the computational capability of the simple LLIF neuron model. In the following sections, the network of LLIF neurons demon-

Figure 4.3: Simple shapes learned by the network of LLIF neurons.

strates object recognition, motion detection, and simple top-down attentional modulation.

4.6.1 Experiment 1: Shape Categorization

In this experiment, the ability of the foundational model to discriminate multiple learned categories is tested. The Shape Categorization Module is trained on three simple letters, a "T", "J", and "L", as shown in Figure 4.3. For this task, the network is trained in a noiseless environment on the objects randomly placed in a 10x10 pixel visual stimulus environment. The object remained still (clamped) for 100ms (100 time steps) to allow the burst-STDP learning rule to modify connections.

After training, each object was randomly placed in the retina. Figure 4.4 shows the classification results for the task. While the learning task is quite simple, it demonstrates that the minimal spiking LLIF model used in these simulations is quite capable of object discrimination when paired with the burst-STDP/homeostatic renormalization learning mechanisms.

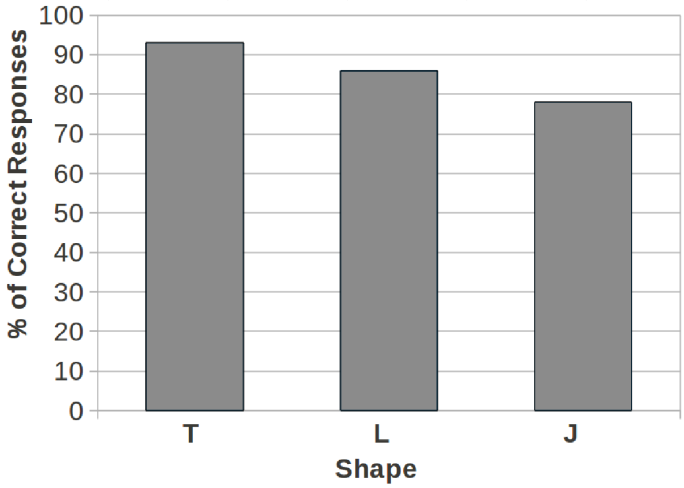


Figure 4.4: Classification accuracy after a single epoch of burst-STDP training and offline homeostatic renormalization.

4.6.2 Experiment 2: Catching Targets and Avoiding Obstacles

In this experiment, the network is tested on a task where a single moving object is present at a time, and the network must choose the correct motor output for the object. To simplify analysis, only two objects were used in this experiment - the "T" object was trained as the target object, while the "L" object was trained as the avoidance object. The object, the starting position, and direction of motion are chosen at random. A response is considered correct if the correct Motor output neuron fires before the object has moved out of the visual environment. The response is considered incorrect if the motor output is to the wrong location for target and avoidance objects. Finally, all other responses are categorized as non decisions, in which no motor output was chosen at all. Furthermore, the amount of noise in the visual receptive field was varied between 0 and 10%. That is, in the 10x10 pixel environment, if there is 8% noise injection, on average 8 pixels may be flipped at any

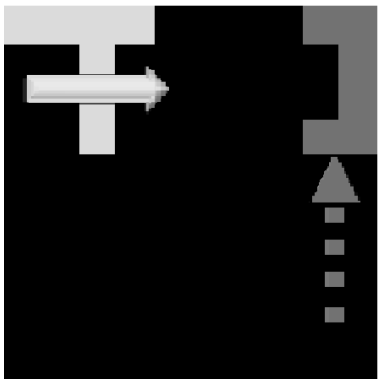


Figure 4.5: 10x10 pixel retina. An object (white T) appears in the upper left corner and moves along the top edge of the visual field. After learning, motor output moves the catcher object (dark gray) to the correct location and orientation.

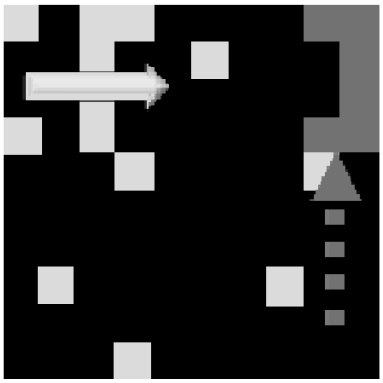


Figure 4.6: Visual environment with 8% noise injection. The object (here a white T) appears in the upper left corner and moves along the top edge of the visual field.

given simulation cycle. Figures 4.5 and 4.6 show the network's task with 0 and 8% noise, respectively.

Figure 4.7 shows the results of the network performance on this task (as noise is varied). The testing phase (for each percent of noise injection) consisted of 100 object presentations, and target object and avoidance object were chosen with equal probability. With no noise injection, the correct response is chosen 96% of the time. As the level of noise increases, the correct motor response degrades gracefully as the number of non decisions increases. For a range of 0 to 10% noise injection, the number of incorrect responses is negligible.

Since the cells modeled in the neural network are LLIF, the cell membranes still potentiate in response to noisy inputs and maintain a memory across multiple time steps, so long as the noise is within a reasonable limit. As a result, the network has an inherent resilience to filter out much noise on its own, as even noisy inputs will eventually cause the cell tuned

Figure 4.7: Performance of the network with single object presentation and a varying level of noise.

Figure 4.8: Performance of the network with two simultaneous objects presented and a varying level of noise.

for a particular edge, feature, or object to fire. Additionally, the decision module makes the network more robust by determining if classifications performed by the shape and motion detection modules are consistent over a reasonable time interval.

4.6.3 Experiment 3: Anticipating a Target Object Location with Multiple Objects

Finally, the network was also tested in an environment where multiple objects could appear in the presence of noise. In this experiment, the system was tested on a total of 100 presentations (for each percent of noise injection). On each presentation, 25% of the time a target object "T" appeared, 25% of the time the avoidance object "L" appeared, and 50% of the time both appeared. The starting position and direction of motion of all objects were

chosen independently and randomly. During training, the attention neurons were value gated to reward firing for presentations of the "T" - that is, to pay attention to the target object over the avoidance object. For this learning task, the network was trained to catch the target object, regardless of what other objects may be present.

Figure 4.8 details the performance of the network with a variable amount of noise injection. Here, the correct motor response is to catch the "T" if the "T" is present and avoid the "L" if the target object "T" is not present. The performance is similar to the results of Figure 4.7, though the number of correct responses is slightly lower. The number of incorrect responses varies between 5 and 14%. However, even in the presence of multiple moving objects, the network responds correctly over 70% of the time with a 3% noise injection.

4.7 Summary

This chapter serves to demonstrate the computational power of one of the simplest implementations of biological spiking neurons: the leaky integrate-and-fire (LIF) neuron. Paired with simple biologically-inspired learning mechanisms, a network of LIF neurons demonstrates its ability to perform simple tasks such as invariant object recognition and motion detection. Furthermore, this simple network takes advantage of feedback connections to achieve top-down attentional modulation. With a strong understanding of the capabilities of the simple LIF neuron, the subsequent chapters investigate a number of more complex neuronal behaviors that can enhance these capabilities.

5 VISUAL CORTEX MODEL

This chapter provides a detailed description of the biologically inspired model of the visual cortex. Many of the components and features of this model build upon the foundational model described in Chapter 4. However, this chapter also explores a set of more complex neuronal behaviors leveraged by biological brains and explains their usefulness in the visual cortex model. Specifically, these complex neuronal behaviors allow the Visual Cortex model to be organized as a *hierarchical metastable attractor*. Next, Chapter 6 details the abilities of this attractor-based network.

5.1 Extending the LLIF Neuron Model

Chapter 4 demonstrated how a minimal linear leaky integrate-and-fire neuron could be used as a basic building block for a visual system capable of invariant object recognition, motion detection, and noise filtering. However, it is known that biological neurons leverage a host of nonlinear behaviors, which some researchers believe enhance their computational capability [14]. In developing a model of the visual cortex, a number of these complex neuronal behaviors are identified, and their usefulness is described in detail.

5.1.1 Short-Term Plasticity

While many neuronal network models include long-term plasticity rules such as Hebbian or STDP learning (or burst-STDP, as discussed in the previous chapter), few consider the

much shorter timescale plastic changes known to exist in biological neurons. These types of synaptic modulations last for tens of milliseconds to minutes [109, 88]. Short-term plasticity appears to be primarily driven by the presynaptic firing rate, though several factors relating to the postsynaptic neuron are known to be related, including postsynaptic receptors which have become saturated [145, 42] or desensitized [25, 140].

Short-term synaptic plasticity can either be potentiating (that is, a frequency-dependent strengthening of the synapse occurs) or depressing (that is, the strength of the synapse decays as a function of presynaptic firing frequency). In fact, experimental evidence has shown that the same neuron may exhibit short-term potentiation on its synapses to one neuron group, while showing short-term depression on its synapses to another [134, 135]. Such behavior appears computationally more powerful, as the same neuron's outputs can exhibit differential effects to different neurons.

For example, short-term depression can be used to signal the presence of a novel input to a downstream neuron. Initially, the presynaptic neuron is at rest. When it detects an input stimulus, it propagates spikes to the postsynaptic neuron with a strong synaptic connection. If the input persists and the presynaptic neuron continues to fire at a high rate, the synapse to the downstream neuron becomes depressed [28]. Hence, when a new/novel input first appears, its effect is stronger than an input that has persisted for an extended period of time. Other work has explored the use of short-term plasticity for developing directional-selective circuits [41, 23].

In the context of a model of the visual cortex, the usefulness of short-term depression is

quite clear. Feedforward connections from lower sensory-input layers can use short-term depressing synapses to indicate a change in the input. In a real-time system with streaming sensory input, such a feature can provide important signalling to a new object or feature of interest. Short-term potentiation can also play important roles. For example, one could control the firing rate of an excitatory neuron (or population of neurons) by recurrently connecting it to an inhibitory neuron (or population of neurons). If the excitatory neuron projects short-term potentiating synapses to the inhibitory neuron, the response of the inhibitory neuron will increase as an effect of the excitatory neurons firing frequency, and in turn, project more inhibition back. By leveraging these types of behaviors, a large scale model of the visual cortex can respond quickly to new inputs, and maintain balanced firing rates.

The LLIF neuron is extended to include a short-term plasticity model based on work by Markram et al. [88]. For each synapse modulated by short-term plasticity, the dynamics are governed by two variables, x and u, using two differential equations:

$$dx/dt = (1 - x)/\tau_d \tag{5.1}$$

$$du/dt = (U - u)/\tau_f \tag{5.2}$$

Here, τ_d is the time constant that governs synaptic depression, while τ_f is the time constant for synaptic potentiation. U is a constant variable chosen from the range $[0, 1]$ which

indicates the percent of synaptic strength that becomes unavailable after a presynaptic spike. When the presynaptic neuron spikes, the variables x and u are updated using the equations below:

$$x(t) = x(t-1) * (1 - u(t-1)) \tag{5.3}$$

$$u(t) = u(t-1) + U * (1 - u(t-1)) \tag{5.4}$$

Synapses are then modulated through scaling the synaptic weight by $u * x$. By choosing the parameters of U, τ_d, and τ_f, the synapses can be tuned to model synaptic potentiation or depression.

5.1.2 NMDA Modulated Synapses

While feedforward connections are primarily responsible for driving signals from sensory inputs up through the various processing regions of the cortex, the role of feedback (that is, top-down connections from higher cortical regions) appears to be quite different. As opposed to feedforward, these feedback connections are not well understood. However, these feedback connections are thought to be more modulatory, rather than driving, in nature, providing top-down context from higher cortical regions. Studies have suggested the role of feedback connections in attentional modulation [34] and figure-ground segregation [66].

A recent experimental study examined the biological differences between feedforward

and feedback connections in the visual cortex [122] and revealed that excitatory feedforward synapses are dominated by AMPA (2-amino-3-hydroxy-5-methyl-4-isoxazolepropionic acid) receptors, while excitatory feedback synapses are dominated by NMDA (N-methyl-D-aspartic acid) receptors. The activation time course of AMPA receptors is fairly short; after a presynaptic spike the effect on the postsynaptic neuron lasts a few milliseconds. Conversely, it has been found that the time course for NMDA-mediated synapses is substantially longer, lasting tens to hundreds of milliseconds [75]. NMDA-mediated synapses are also distinct in the fact that their activation is *voltage-dependent* [75]. That is, the activation of NMDA receptors is dependent on an initial depolarization of the postsynaptic neuron. In simple terms, a neuron must receive stimuli at voltage-independent synapses (e.g. those mediated by AMPA receptors) before the stimuli at the voltage-dependent synapses can affect the postsynaptic neuron membrane potential.

As with short-term plasticity, there appear to be unique advantages of NMDA-mediated synapses for a visual cortex model. Because of the long lasting effects of NMDA-mediated synapses, top-down activations can provide long-timescale contextual signalling to lower cortical regions. Furthermore, as NMDA receptors are more modulatory, rather than driving in nature, this top-down signalling can be quite diffuse without causing interference. Only neurons which receive feedforward evidence (through AMPA-mediated synapses) are affected by these top-down context signals.

The neuron model is first extended to include the long time constant behavior of NMDA-mediated synapses. Each presynaptic neuron is extended to include a variable s. When

the presynaptic neuron spikes, s is updated via:

$$s(t) = (1 - s(t-1))/\tau_{NMDArise} \tag{5.5}$$

For each time step the presynaptic neuron doesn't spike, s is updated via:

$$s(t) = (s(t-1))/\tau_{NMDAdecay} \tag{5.6}$$

Here, the $\tau_{NMDArise}$ parameter controls the rise time of NMDA, while $\tau_{NMDAdecay}$ controls its decay. For synapses that are mediated by NMDA-receptors, the synaptic weight is scaled by s on each simulated time step and integrated with the postsynaptic neuron's membrane potential (as opposed to other synapses, which are integrated only in the time steps where a spike is present). So long as the value of s is non-zero, it is integrated at each time step.

To capture the voltage-dependent behavior of NMDA-mediated synapses, a simple "depolarization threshold" variable, D_{th}, is used. The neuron model first integrates the inputs from all the voltage-independent connections the same as before (Equation 5.7). If the membrane potential $V_i(t)$ exceeds D_{th}, the inputs from the NMDA-mediated synapses are integrated. This is an extreme simplification of voltage-dependent behavior, but nonetheless captures the general principle.

$$V_i(t) = V_i(t) + \sum_{l=1}^{L} s * W_{li} \tag{5.7}$$

After this integration, the neuron may fire and reset its potential (as before in Equations 4.2 and 4.3).

5.2 The Visual Cortex as a Hierarchical Metastable Attractor

This section describes a large-scale model of the cortex, on the order of 100,000 LLIF neurons. Building upon the model outlined in Chapter 4, the Visual Cortex model is composed of a number of hierarchical processing levels, which perform feature detection in lower areas and ultimately invariant object recognition in the highest areas [89]. Furthermore, this section proposes that the visual cortex is a metastable *attractor* network, driven by feedforward input from the retina and stabilized through recurrent connections between different neural areas.

5.2.1 Hierarchical Organization and the Feedforward Pathways

Figure 5.1 shows the hierarchical model which captures the organization of the ventral stream of the visual cortex. Much of the organization builds upon the foundational model of the visual cortex outlined in Chapter 4, though here, each of the regions of the visual cortex are modeled at a larger scale and with a greater degree of complexity.

The input to the model is a 64x32 pixel retina, where simple objects (uniform color) appear and move across the retina. These inputs project (feedforward) to the Lateral

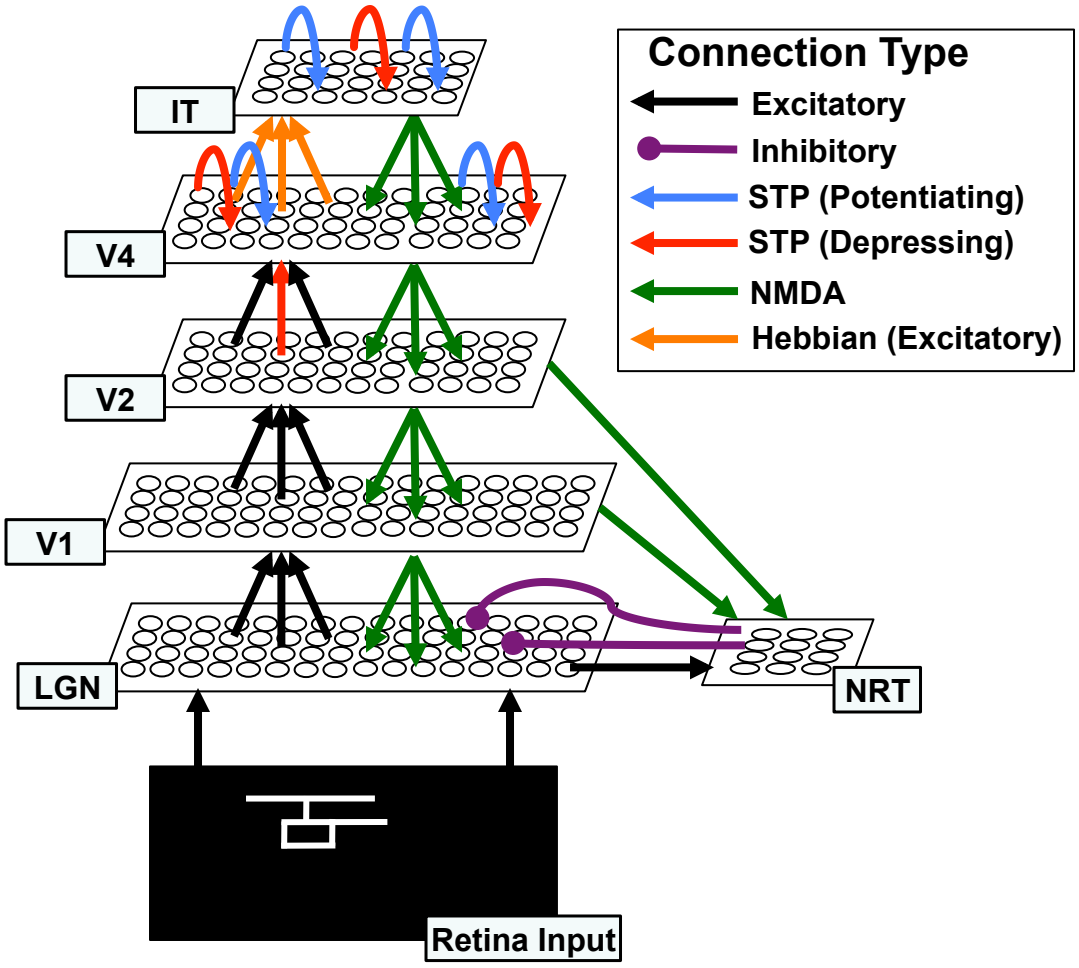


Figure 5.1: A block diagram of the Visual Cortex model. All modeled regions contain recurrent connectivity (unshown for simplicity) and both excitatory and inhibitory neurons, except the NRT, which is an entirely inhibitory population.

Geniculate Nucleus (LGN) cells. The LGN cells have small (typically 3x3 pixels) center-surround receptive fields; center-on cells respond with their highest firing rate when their input is a light pixel surrounded by dark pixels, while center-off cells respond maximally when their input is a dark pixel surrounded by light pixels. Thus, the modeled LGN cells provide the first step toward salient edge detection and convert the retinal input into spikes for the rest of the model.

As shown in the figure, the LGN is tightly coupled with another modeled region of the thalamus, the thalamic reticular nucleus (NRT). The NRT is an entirely inhibitory population of neurons. It does not project to the cortex, but rather, modulates the activity of other regions in the thalamus through inhibition [150], helping to regulate the information coming in through the sensory pathways. In this way, the NRT provides a high degree of noise filtering for the inputs propagated from the retina. Furthermore, since feedforward activations begin at the LGN (and feedback is voltage-dependent), the NRT plays a critical role in modulating the overall firing rate of the network through its inhibition.

In the feedforward direction, spikes propagate from the LGN to the modeled primary visual cortex (V1). Here, the modeled LIF neurons have slightly larger receptive fields (typically 5x5 LGN cells) which maximally respond to an edge of a preferred orientation. The modeled LIF neurons are topographically organized with overlapping receptive fields; each neuron is tuned to detect either a horizontal or a vertical edge.

The V2 neurons perform a max-pooling operation across a local neighborhood of V1 cells (typically a 3x3 region of V1 cells), thus providing a higher degree of invariant edge detection. V2 cells only pool neurons of the same type (i.e. a V2 cell will only pool horizontally-oriented V1 cells, or only vertically-oriented V1 cells).

In turn, the activations of these V2 cells propagate to the V4 region, where neurons respond to more complex features over a larger receptive field. A portion of these feedforward connections exhibit short-term depression; since the V2 provide a degree of translation invariance, they fire for longer periods of time than the V1 cells (when an object is moving).

However, when a new V2 cell detects an edge within its receptive field, it responds with an initially strong connection to indicate a new input; after a period of high firing, its feed-forward connection strength is moderately depressed. The V4 neurons typically receive inputs from a 4x4 region of V2 cells; here, the same V4 cells receive inputs from both the horizontally and vertically-oriented V2 cells. In this way, the V4 cells can be tuned to detect a more invariant, but more complex shapes, such as conjunctions of multiple edges.

Finally, the activations of the V4 cells propagate to the modeled IT. Neurons in the IT have a receptive field over the entire V4 region (and thus, have a receptive field over the entire retina). These neurons respond maximally to the recognition of complex objects composed of multiple features detected in the V4 region. While the synaptic weights of the lower regions are hard-wired (i.e. the edge detectors, invariant edge detectors, and simple feature detectors), the connections to the neurons in the IT can be trained via Hebbian learning (though other paradigms such as STDP and burst-STDP work as well). Thus, when a new object is presented to the Visual Cortex model, the LGN through the V4 perform feature extraction, while cells in the IT learn to recognize the combination of these features compose an object.

5.2.2 Lateral Connections within Modeled Areas

Each of the modeled regions in Figure 5.1 contains both excitatory and inhibitory populations of neurons, except for the NRT, which is inhibitory neurons only. Aside from the NRT (which is 100% inhibitory), typically 80% of cells are excitatory and 20% are inhibitory,

which is consistent with biological evidence [115]. While the figure does not show all connections (for simplicity), each of these neural regions contains recurrent connectivity between excitatory and inhibitory neurons, primarily to balance the firing rate and add stability to the system.

In the upper levels of the model (The V4 and IT), many of the excitatory-to-excitatory connections are modulated by short-term potentiation and short-term depression. As discussed below, these modulatory connections help ensure that attractor states form and dissolve quickly with changing inputs.

5.2.3 Invariant Object Recognition Through Feedforward Pathways

To perform invariant object recognition, the Visual Cortex model requires only its feedforward connections. In the Visual Cortex model, the connections between the higher regions (V4 and IT) are plastic, learned during training sequences, while the lower levels of the model use hardwired connections. As the lower levels capture much simpler features (center-surround contrasts, edges, and simpler conjunctions of edges), there is little benefit provided by plastic changes in these areas. Furthermore, experimental evidence has pointed out that much of the tuning of the lower thalamic and cortical areas may take place before birth or early thereafter [24, 47]. However, there is benefit to having an IT that can learn the conjunctions of different features and ultimately recognize invariant objects.

During training, the network was presented with two simple objects: a helicopter and a car. The IT was configured with two populations of neurons (80% excitatory and 20% in-

hibitory [115]) recurrently connected, with initially weak random feedforward connections from the V4. As feedback connections are voltage-dependent (that is, only modulatory rather than driving in nature), they were initialized to a predetermined strength and did not require plasticity; however, it has been proposed that these feedback connections also exhibit plasticity in biological systems [139].

On each training epoch, either the car or the helicopter was randomly placed in the retina for 1000 ms (or stimulated time steps). Activation percolated up through the network, ultimately reaching the IT. As with the foundational model presented in Chapter 4, the learning rule was modulated by a value-dependent learning - which also required competitive inhibition between the two populations of the IT. When the correct population in the IT had the higher firing rate, Hebbian learning strengthened the corresponding connections. After training, the competitive inhibition between the IT populations were removed (since after training, the network was capable of recognizing multiple objects at a time). The network was able to recognize the helicopter and the car moving through the retina, invariant of their position.

5.2.4 Why Attractors?

The brain is autonomously active, exhibits transiently stable internal dynamical states as well as sensitivity to incoming stimuli [51, 52]. A number of studies have considered these *attractor states* to underlie the mechanisms and processes of the brain, including decision making [7, 46, 18], short-term working memory [8], and the storage and recall of long term

memories [65]. Each of these examples consider that a single spike doesn't provide enough evidence to signify an important event; therefore it is more likely that neurons (or groups of neurons, minicolumns [99], etc.) signal important outputs through a temporary but stable increased firing rate.

Besides the applicability to decision making and memory storage networks, the question still exists: why would the brain (or in the context of this dissertation, a model of the visual cortex) be organized as an attractor network? It has been proposed that the brain is organized by two fundamental principles: *functional segregation* and *functional integration* [44, 149]. Functional segregation in the cortex is quite clear: different cortices process the information of different sensory pathways. Even within the Visual Cortex model as described above, functionally segregated regions perform different computations. Functional integration in the cortex is the ability of these segregated areas to influence each other [138]. Thus, a global attractor state is capable of integrating information across spatially separated neural regions which may process entirely different information [44].

This type of network is significantly different from successful feedforward classifier networks like Convolution Neural Networks (CNNs) or Poggio's HMAX [111]. Such schemes are highly successful at image classification, since they are organized to create robust invariant representations the visual stimuli they were trained on. However, these types of visual systems completely ignore binding the particular details in sensory input with the invariant representations stored in the higher regions of the cortex.

5.2.5 Integration and Feedback Pathways

As described above, the feedforward pathways (shown in green in Figure 5.1), through primarily converging connectivity, are capable of feature extraction and ultimately invariant object recognition. Neurons in the IT are quite capable of recognizing a particular person with a high degree of invariance (e.g. a close friend can be recognized invariantly whether they are close or far away, angry or smiling, is wearing a new shirt, etc.). However, because the IT is so invariant, these neurons cannot distinguish particular features of that person. Lower area neurons in the V1 however, respond to very particular and localized features (e.g. the edges that outline a person's facial features, the color of their clothes, etc.) but are completely unaware that a face is present - they simply respond to the localized details present in their smaller receptive fields. Thus, from this example, it is easily understood that no single neuron can simply represent all the information necessary to understand real-world inputs. Therefore, recurrent connectivity is necessary to integrate the Visual Cortex model into a single *hierarchical metastable attractor state* which binds the *invariant* neurons in higher cortical regions with the *detail-specific* neurons in lower regions. As a whole, such attractor states are able to represent all the information necessary to understand complex real-world inputs.

To achieve stable state behavior and integration between all modeled neural regions, feedback connections are required. As shown in Figure 5.1, these top-down connections are primarily modeled as NMDA-mediated synapses. Thus, top-down activations are modulatory rather than driving in nature (since NMDA-mediated synapses are voltage-

dependent) and are prolonged in their signalling effect (allowing for stable integration across spatially separated neurons in lower areas).

5.2.6 Metastability of a Hierarchical Attractor

Recent experimental studies have proposed that the dynamics observed in the IT region of monkeys are indicative of attractor states and firing rate adaptation [6]. The recorded populations of neurons showed sustained firing for their preferred stimulus for long periods of time (hundreds of milliseconds). Furthermore, recent computational models have investigated "nested" attractors whose dynamics operate at different timescales [48]. This work investigated two levels of processing in an attractor: a bottom level, consisting of populations operating on a short timescale, and an upper level, consisting of "memory populations", which operate on a much slower timescale. This "nested" attractor was proposed as an explanation for the bistable perception paradigm.

Building on these ideas, the Visual Cortex model proposes that the transient stability of a neural area should reflect its location in the hierarchy and its degree of invariant response. Considering the visual cortex again, it is clear that the primary visual cortex (V1) should exhibit the shortest timescale. Cells in the V1 respond to specific edges, and neural activity must change rapidly with changing retinal input. However, an area like the V4, which responds to much more invariant representations of more complex features, should exhibit a much longer transient stability for visual recognition tasks. And finally, neurons in the IT should exhibit the longest timescale of the visual cortex, since neurons here respond to

objects completely invariant of their position in the retina. While these ideas share much in common with the idea of the "nested" attractors [48], the primary difference is the Visual Cortex model presented here performs all the sensory processing through many levels and allows top-down influence to occur at a timescale proportional to its degree of invariance.

Maintaining a metastable internal state is clearly essential for various tasks, such as decision making, maintaining a short-term memory, or tracking an object even when it is occluded (i.e. object permanence). However, it is also clear that such stable states should be able to fade on their own (i.e. without the need for resetting an attractor "by hand" or with a global inhibitory system) to allow new states to be driven by new stimuli [51, 52]. In short, the brain exhibits the ability to be driven into a particular attractor state and maintain it for a useful amount time, and this state can either fade away or be modified by future stimuli.

To achieve this type of flexible behavior, the Visual Cortex model leverages short-term plastic connections, as shown in Figure 5.1. In the visual cortex model, short-term potentiation and depression are primarily used in the modeled V4 and IT regions, since these regions exhibit the longest transient stability. With short-term potentiation, a rapid boost in connection strength between neurons allows a neural network to enter a stable state rapidly. Short-term depression exhibits a weakening of synaptic strength over time, allowing stable states to dissolve so the system may be driven by new sensory inputs. These modulatory connections essentially shape the attractor states the model enters and leaves.

Figure 5.2 shows two snapshots of the hierarchical attractor formed in the visual cortex when exposed to a simple video of a moving helicopter. Each of the black panels shows

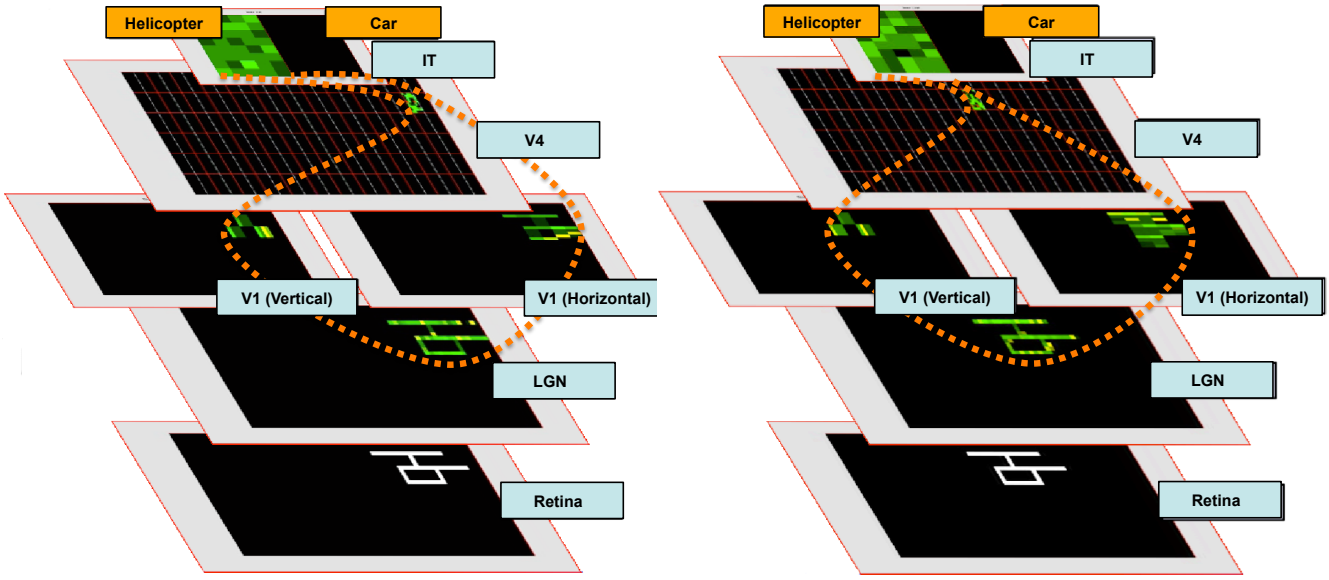


Figure 5.2: The hierarchical attractor is anchored by its "head" in the IT, while its "limbs" in the lower levels flexibly change to adapt with changing/moving inputs.

one of the modeled layers from the Visual Cortex model shown in Figure 5.1 (the LGN and V2 are not shown for simplicity). Black pixels in the model indicate neurons that are not firing, while the intensity of the green pixels indicate the present firing rate of the neuron.

When the helicopter first appears (left panel), a feedforward volley of spikes propagates through the various modeled regions, ultimately reaching the IT, where neurons respond to the recognition of a helicopter. Top-down activations, in turn, stabilize firing in the different regions. The orange dashed line shows the neurons that stabilize at a high rate of firing in this attractor state.

As the helicopter moves through the retina (right panel), the attractor state adapts to the changing input. The "head" of the attractor remains anchored in the IT; as shown in the figure, the firing rate is stable for an extended period of time. So long as there is consistent feedforward evidence of a helicopter, the "head" of the attractor is active

indefinitely. Conversely, the "limbs" of the attractor in the lower cortical areas (V1, LGN, etc.) also show stable activity, but on a much shorter timescale, so that neural activity can be easily driven by changing feedforward inputs from the retina. As shown in the figure, the "limbs" have changed between the two panels due to the small receptive field LGN and V1 cells . Finally, intermediate areas like the V4 (and V2, unshown) fall in between - its transient stability reflects its level of invariance.

Figure 5.3 similarly demonstrates a hierarchical attractor anchored by its "head" in the IT, but here, the time-varying firing rate of neurons in the V4 and IT are shown. Furthermore, in this example, there are two moving objects, a helicopter and a car. As shown in the bottom panel on the right, populations in the V4 are initially strongly activated (by the short-term depressing feedforward connections from the V2 region). As the helicopter moves, new populations receive feedforward evidence and become activated, while the previously firing populations transiently decay in their response. In the top panel, we see two populations of neurons (one encoding the invariant representation of the helicopter, the other, the car) remain stable and active the entire time (though, their firing rates fluctuate with the feedforward evidence provided by the V4 neurons). This long-lasting stable response in the IT is consistent with experimental evidence that has proposed attractor dynamics in the IT [6].

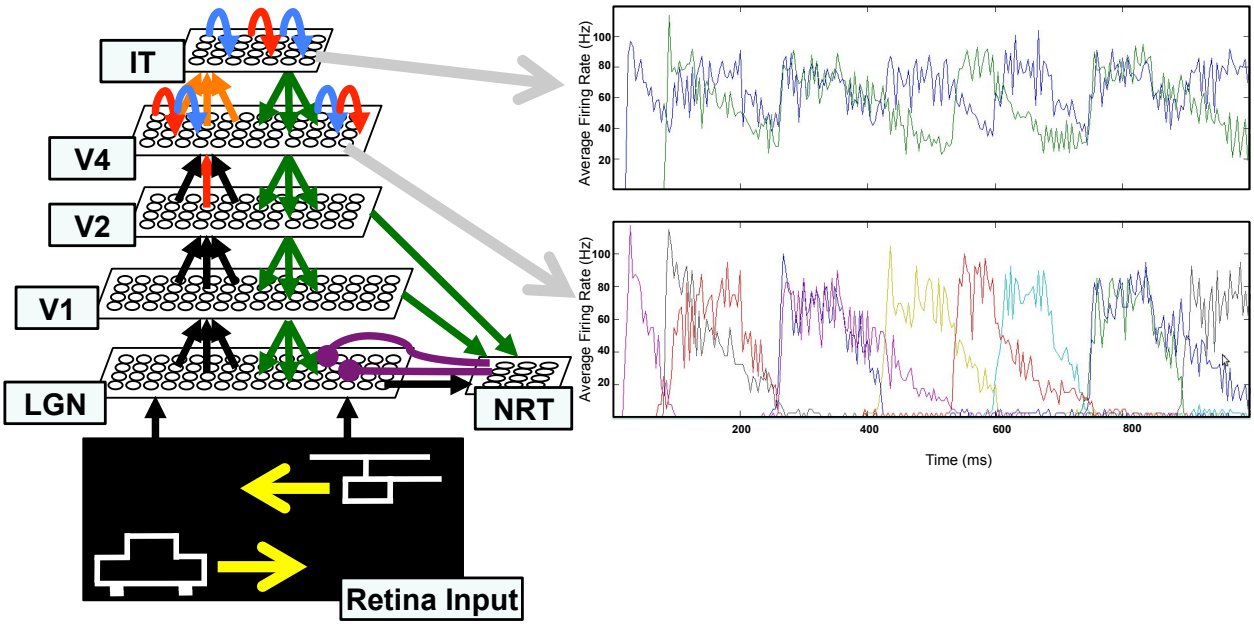


Figure 5.3: On the left, the Visual Cortex model is exposed to a moving helicopter and moving bus. The right shows the average firing rates of neurons responding to the stimulus in the V4 and IT. As shown, the timescale of neuron populations in the IT is much longer.

5.3 Summary

In this chapter, the simple LIF neuron model and basic visual system model from Chapter 4 are extended. A growing body of literature has described the brain as a metastable attractor network, continuously changing and adapting to incoming sensory input. In this chapter, this idea of a hierarchical metastable attractor is used to construct a functional model of the visual cortex. However, to achieve metastable attractor states, capable of forming and dissolving rapidly with changing sensory input, a number of complex neuronal behaviors have been identified. Voltage-dependent synapses, prolonged signalling effects, and short-term plasticity significantly extend the computational capabilities of the LLIF neuron. In

the next chapter, some of the abilities of a hierarchical metastable attractor are explored.

6 ABILITIES OF A HIERARCHICAL ATTRACTOR NETWORK

In the previous chapter, it was proposed that a Visual Cortex model exhibiting metastable attractor states is significantly more powerful than more traditional feedforward neural network architectures. This chapter highlights three experiments which demonstrate the capabilities provided by these metastable attractor states. First, the network's capability to recognize and reconstruct an incomplete image is tested. Next, the short-term working memory capabilities of the metastable attractor are demonstrated. The final experiment shows how a hierarchical attractor network is capable of integrating information across spatially separated neural regions. Here, it is proposed that the hierarchical attractor architecture allows a novel way to *access* information about a visual scene and *route* the information to an appropriate output. As will be detailed below, the results presented here are promising evidence for a scalable network capable of *scene understanding*.

6.1 Pattern Completion and Noise Resilience

One of the important features of attractors networks is, even when just a subset of the attractor has reached stable activity levels, the attractor tends to complete itself [8]. In this way, the Visual Cortex model can complete patterns and recognize incomplete objects presented in the retina.

The left panel of Figure 6.1 shows a retina viewing a noisy scene of a helicopter with a few features missing. Feedforward driven neural activity appears in green, while feedback

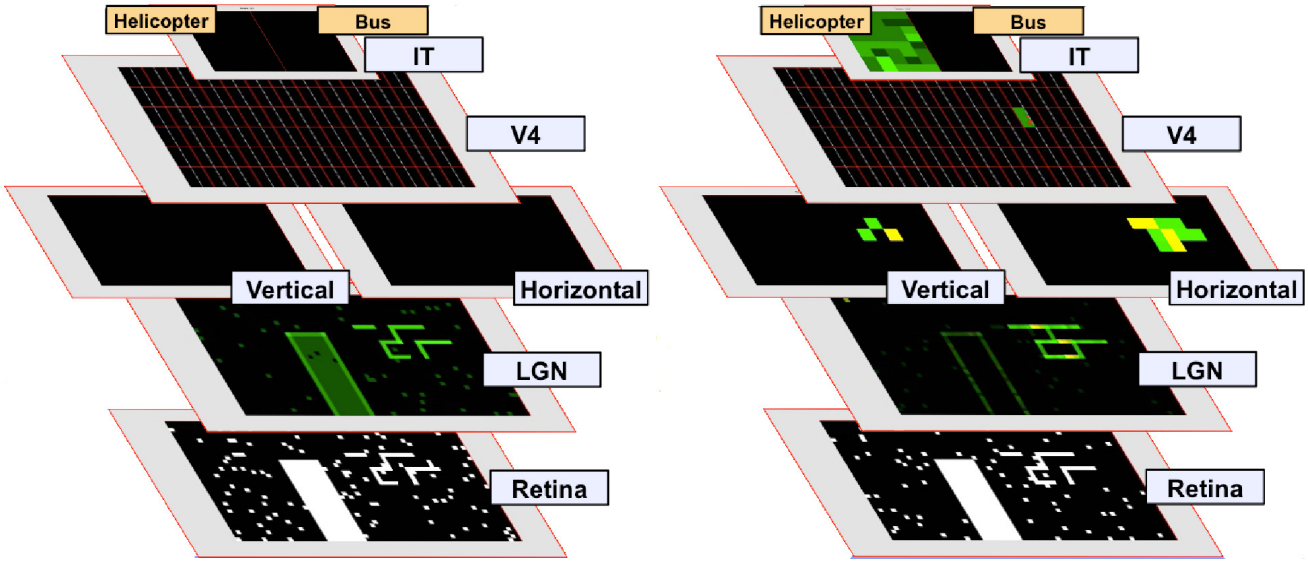


Figure 6.1: Left: An incomplete helicopter pattern in a noisy retina is presented to the visual system. Initial feedforward driven activity in the LGN mirrors the activity of the retina. Right: Pattern completion in hierarchical attractor visual system. Feedforward driven activity appears in green, and top-down pattern completion appears in yellow.

signalling appears in yellow. Initially, the incomplete helicopter pattern, as well as the noise in the retina, are propagated to the LGN. However, when there are sufficient inputs to drive the system towards an attractor state, typically the network stabilizes in less than 150 ms. The right panel of Figure 6.1 shows a snapshot of the networks response after the attractor has stabilized, and feedback driven activations (yellow) in the LGN are able to complete the helicopter pattern. Furthermore, the most salient features of the helicopter are enhanced (through top-down modulation of their firing rate), and the noise propagated from the retina to the modeled LGN is inhibited. As such, hierarchical attractors can exhibit the same powerful properties (such as pattern completion) that Hopfield attractor networks do, while remaining flexible enough to change state with rapidly changing inputs.

Here, the advantage of modeling voltage-dependent synapses is clear. Top-down

connectivity between any two layers is mostly diverging, yet the neural activity is only enhanced in the areas that have received at least some amount of feedforward spiking evidence. Without such voltage-dependent synapses, top-down activity from the IT could activate many more features associated with helicopters in the V4, and from the V4 to the lower levels of the model, resulting in unrecognizable activity in the LGN.

It should be noted that the ability of a neural model to reconstruct an image or pattern does not help it as a classifier; if there is enough information for the network to converge on an attractor state that completes the helicopter pattern, then there is already enough information to classify the noisy image as a helicopter. However, what this experiment does show is that the attractor state is able to *specify* the particular features that are missing, using the feedback connections from the higher, more invariant regions. This feature of the hierarchical attractor network is demonstrated in greater detail below in Section 6.3.

6.2 Object Occlusions (Working Memory)

The next experiment demonstrates the advantage the hierarchical attractor provides for a working memory task. One can consider watching a helicopter fly across a landscape, which then becomes occluded as it flies behind a building. In such a situation, one's brain provides *object permanence*; that is, it is understood that the helicopter still exists, even though it cannot be observed. Furthermore, one *expects* that the helicopter will fly out from behind the building on the opposite side.

In this experiment, the helicopter moves from right to left in a continuous motion

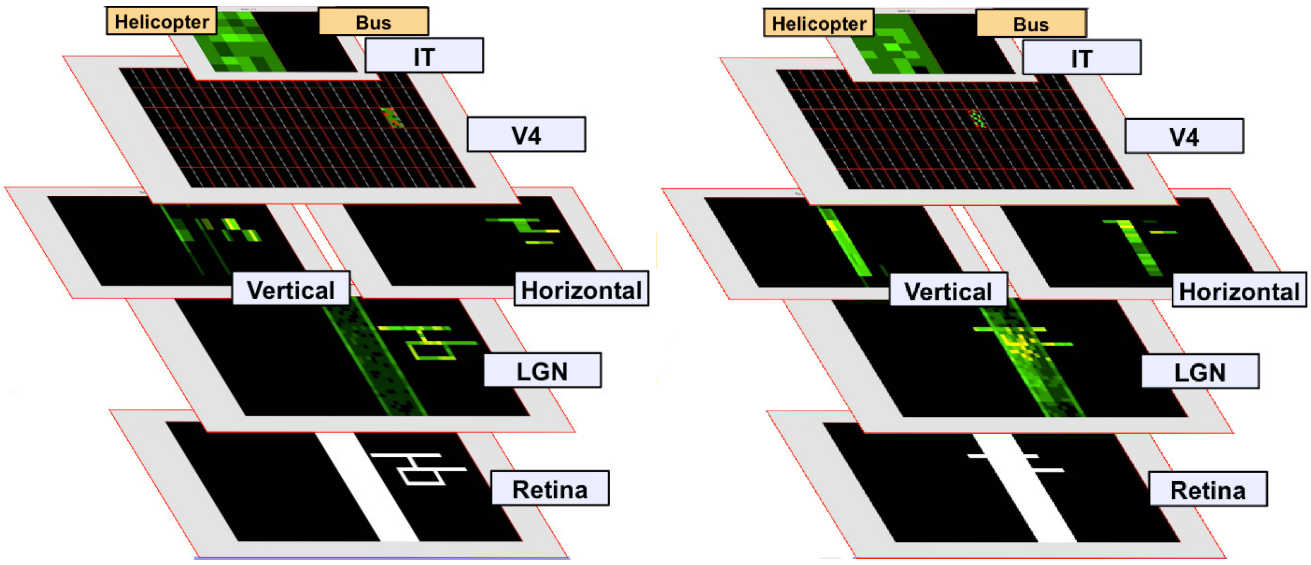


Figure 6.2: Left: A helicopter moves from right to left in the retina. Right: The helicopter is occluded by the building. However, transient stability and hierarchical organization allows the IT and V4 to still show robust responses even in the absence of current feedforward evidence.

(moving approximately 1 pixel location per 25 ms). Figure 6.2 shows two snapshots of the network as the helicopter flies behind a building (the solid rectangular shape). In the panel on the left, the helicopter is still completely visible, and the hierarchical attractor shows a stable response in the modeled LGN, V1, V4, and IT (V2 unshown).

In the panel on the right, the helicopter has become occluded as it moves behind the building. However, due to the transient metastability of the network, there is still a robust response in the IT and V4, indicating a short-term memory of the moving helicopter and its features. Thus, the system displays a sense of object permanence through this working memory, since the concept of a helicopter is still active, even if there is little to no feedforward evidence indicating a helicopter is present. As seen in the modeled retina and LGN, there are clearly not enough features to recognize a helicopter in a feedforward-only

network.

Again, this is possible only because of recurrent long timescale (NMDA) connections and the high degree of metastability in the V4 and IT neural layers. Furthermore, as the helicopter emerges from the left side of the building, the attractor state quickly converges in the LGN and lower modeled areas. Since top-down connectivity is present, the formation of the full attractor state is boosted as feedforward evidence of a helicopter appears.

6.3 Access and Routing in a Hierarchical Attractor

In this final experiment, the ability of a hierarchical attractor to *functionally integrate* information across spatially distributed neurons is demonstrated. As was described in Chapter 2, the cortex uses different neural regions to process different types of information. Even within the visual cortex, different modalities such as motion and location, color, and form are processed through separate streams. Thus, it appears from a first glance that the visual cortex is quite capable of decomposing information into different *concepts* (e.g. the color red, the shape of a helicopter, the location of an object).

In fact, this approach has shown to be quite successful for a number of neurally-inspired object-recognition systems [90, 112, 127, 111]. Requiring only feedforward connectivity, an image can be decomposed into simple features, and these features can be progressively combined for invariant object recognition (using an approach not unlike the feedforward pathways of Visual Cortex model). Such "detector" based neural architectures have shown to be quite successful, capable of robustly invariant object recognition.

Consider a feedforward detector for the invariant recognition of a helicopter. Using a feedforward-only approach, one can construct a robust helicopter detector that classifies a red helicopter, a green helicopter, a helicopter on the ground or in the sky, as a helicopter. However, in taking a feedforward-only approach, these systems have essentially "thrown away" the information regarding the helicopters color and location in favor of robust invariant object recognition. If it is important to classify red helicopters in the sky (say, the detection of an enemy helicopter approaching as compared to a friendly green helicopter grounded on a runway), one must construct a *red flying helicopter* detector. For every combination of concepts across the motion and location, color, and form streams, one must build an individual detector. While this approach would no doubt be able to robustly classify all the combinations of concepts that are deemed important, it has two major flaws. First, this leads to a combinatorial explosion of the number of detectors that must be constructed. Second, this approach requires that these detectors be built *a priori* - one must consider every possibility before constructing the detectors. Though it is far from completely understood how the brain stores memories, it is quite unlikely that the brain would utilize a different neuron or network of neurons for every possible scenario that could possibly be encountered. Such an approach defeats the purpose of having invariant representations of concepts in the brain.

Rather, this dissertation proposes the idea that the cortex uses *integration* across different processing streams to bind multiple concepts. Again, this integration requires the organization of the network as a global attractor which leverages recurrent connectivity between

different regions. To examine this idea as a proof of concept, the Visual Cortex model was scaled down to approximately 2,000 neurons. This smaller version of the model, however, was extended to include three discrete processing streams with their own concepts. The *Form* stream, as with the original Visual Cortex model, performs feature extraction and ultimately invariant object recognition of a helicopter and a car. The *Color* stream contains two concepts: the recognition of the colors red and green. Finally, the *Where* stream simply recognizes the location of an object, completely invariant to what the object is.

Figure 6.3 shows the simplified model of the Visual Cortex with its three discrete processing streams. Here, the retina has been scaled down such that a helicopter or a car can be in one of four locations. Neurons in the LGN, as before, use center-surround receptive fields. However, they are also sensitive to the color in their receptive field; hence, there are green and red-tuned LGN cells. In the Color stream, populations of neurons simply detect whether there is a patch of a particular color (red or green) in their receptive fields. At the top of the Color stream, one population of neurons detects invariantly whether there is a patch of green in the retina, while the other detects whether there is a patch of red. In the Form stream, the first layer contains a population of neurons that respond to simple feature within their receptive field of the LGN (though, for simplicity, this stream does not explicitly model each of the V1, V2, and V4 regions as in the large-scale Visual Cortex model). These object detectors respond invariantly to the color of the object they are detecting (i.e. the same object detector will respond to the detection of a red or green object). At the top of the Form stream, one population invariantly detects the presence of a

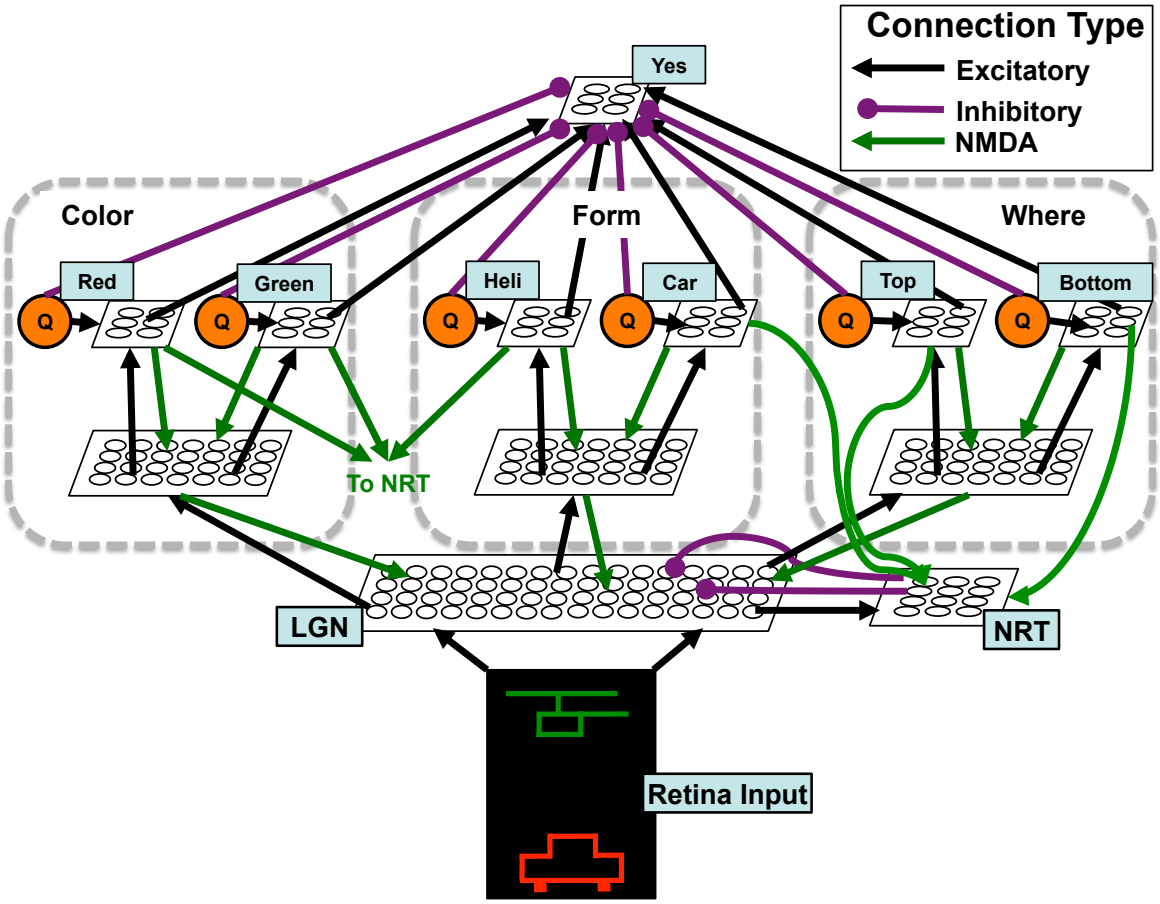


Figure 6.3: The scaled-down version of the Visual Cortex model. Three simple modalities recognize two invariant concepts each. The orange "Q" nodes are activated when the concept is being queried. The "Yes" population is activated when the query is consistent with the visual scene in the retina.

helicopter, while the other detects the presence of a car. Finally, in the Where stream, the first layer contains a population of neurons that simply detects whether there is any object or feature at all in their receptive field, regardless of the shape or color. The top layer of the Where stream pools the response over these cells; one population detects invariantly whether there is an object in the top half of the retina, while the other population detects whether an object exists in the bottom half of the retina.

When an object appears in the retina, each of the different processing streams extracts a different set of features and ultimately recognizes different invariant concepts. For example, when a green helicopter appears in the sky, the *helicopter* concept in the Form stream shows stable activation, as does the *green* concept in the Color stream and the *top* concept in the Where stream. Simply looking at these invariant concept populations, one may argue that the network "understands" what it is looking at; since the *green*, *helicopter*, and *top* are activated, there must be a green helicopter in the top of the retina. However, if a red car also appears on the ground, the invariant concepts *car*, *red*, and *bottom* are all activated, making the color and location of the two objects appear ambiguous, that is, if one is considering only the state of the invariant concept detectors.

To test the ability of the hierarchical attractor to integrate information across different processing streams, the simplified Visual Cortex model is slightly extended to answer "questions" about the scene it is observing. To "ask" the model questions, an additional *invariant auditory concept* is connected to each of the invariant concepts in the Visual Cortex model, as shown in Figure 6.3 (orange circles). Each of these auditory concepts could be considered to be the high level concepts of the auditory cortex; when one of the auditory concepts is active, the model is being asked a "yes/no" question about that concept. For example, if the *helicopter* auditory concept is activated, the Visual Cortex model is being asked "Is there a helicopter?". If the *helicopter*, *top*, and *green* auditory concepts are activated, the model is being asked "Is there a green helicopter in the sky?".

When the auditory concepts are activated during a query, they project excitatory con-

nections to each of their corresponding visual concepts. In turn, the firing rate of the corresponding visual concepts go up, and likewise, project more top-down excitation to the neurons below them. Again, in terms of the stabilization of the hierarchical attractor, the voltage-dependent behavior of these top-down connections is vital; only the appropriate neurons in the level below will see these top-down modulations, while other neurons remain unaffected. As a result, the lower level feature-processing populations of the corresponding streams also experience a higher firing rate as a result of the top-down modulation. Finally, these levels project voltage-dependent feedback connections to the LGN cells which perform the first level of feature extraction in the Visual Cortex model.

It is also important to consider the feedback connections from the cortical areas to the inhibitory NRT population, as shown in Figure 6.3, which is consistent with biological evidence [150]. In this model, the role of these top-down connections allows the NRT to balance the extra excitation during the query with inhibition to the LGN. Since each of the visual and auditory concept populations sends a feedback connection to the NRT, the NRT receives an excitatory boost proportional to the complexity of the query; that is, if asked about a helicopter alone, one feedback pathway shows an enhanced firing rate, while the query "Is there a green helicopter in the sky?" results in three enhanced feedback pathways to the NRT.

The organization as an attractor allows the network to truly integrate information across these spatially separated neural regions. Through the enhanced firing rate provided by the auditory concept populations, the network is able to *access* the appropriate information

in regard to the conjunction of concepts. When the network is asked "Is there a green helicopter in the sky?", the *helicopter*, *green*, and *top* concepts enhance their firing rate, and enhance the firing rate of the corresponding features below them, in a way that could be compared to top-down attention. In turn, the excitatory input to the NRT goes up, which means more inhibition for the LGN. However, since each of the three streams is providing strong feedback to the same cells in the LGN, the attractor state balances.

Finally, the network needs to be extended to effectively *route* this information to a population of neurons that could simply answer "yes/no" questions. As shown in Figure 6.3, each of the visual concepts sends an excitatory input to a single *Yes* population at the top of the network. Each of the auditory concepts sends an inhibitory input to the same *Yes* population. This essentially "primes" the *Yes* population to answer a default "no" by preventing it from firing. In this way, if the Visual Cortex model is asked about a conjunction of two or three concepts, each of these concepts must stabilize at an enhanced firing rate to answer "yes" to the query.

In Figures 6.5 through 6.7, the Visual Cortex model is tested on its ability to correctly answer queries, a task, as has been described, that truly requires an integrated attractor state. In Figure 6.4, the Visual Cortex model is presented with a green helicopter in the top of the retina, and the system is asked "Is there a helicopter?". As shown, the firing rate of the helicopter concept is enhanced by the query, which in turn, projects extra feedback to the corresponding LGN cells. The NRT population enhances its firing rate during the query, but the feedback balances out the extra inhibition; as a result, the helicopter concept

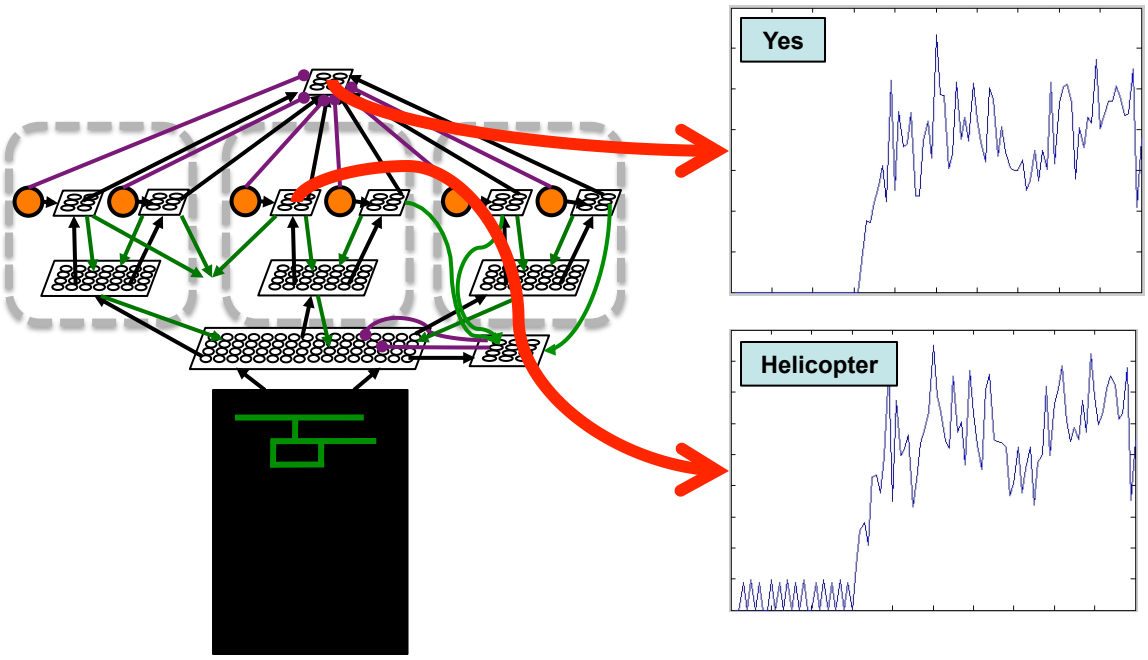


Figure 6.4: The system is asked whether it sees a helicopter. The population which invariantly recognizes the concept *helicopter* stabilizes to a higher firing rate and activates the *Yes* population.

stabilizes at a higher firing rate, and the *Yes* population correctly answers the query with robust firing, as shown in the figure.

Next, under the same conditions, the system is asked "Is there a car?" (see Figure 6.5). In this case, since the feedback connections are voltage-dependent, the system is unable to activate any neurons in the lower levels. As shown, the car concept does not exhibit a high firing rate, and hence its input to the *Yes* population falls short of activating it. The inactivation of the *Yes* population is an implied "no" to the query, hence the system correctly answers this query as well.

Next, a more complex scene is presented to the Visual Cortex model: a green helicopter in the sky and a red car on the ground. In Figure 6.6, the system is asked "Is there a green

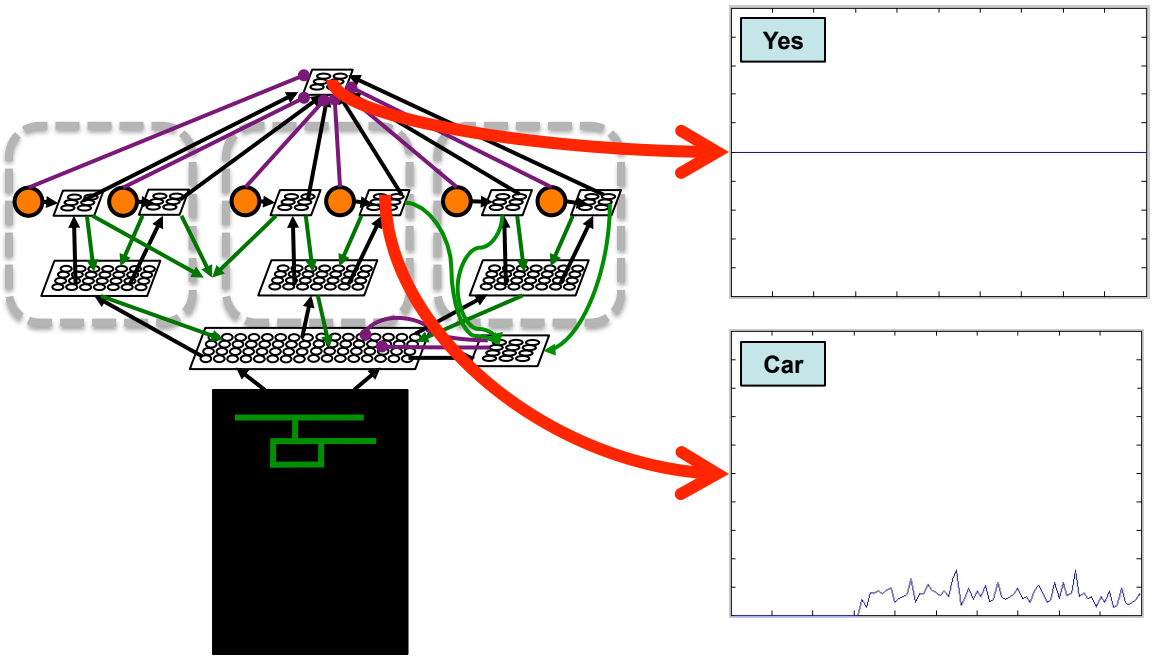


Figure 6.5: The system is asked whether it sees a car. Since feedback connections are modulatory (voltage-dependent), but there is no feedforward evidence of a car, the population does not stabilize at a high firing rate, and the *Yes* population is silent (implied "no").

helicopter in the sky?". All three concepts activate at a higher firing rate due to the query, and subsequently, each send feedback to the same neurons in the LGN population. When the NRT enhances its firing rate proportionally to the query, the feedback excitation and extra inhibition balance out, allowing the cells in the LGN to remain active at their typical levels. As a result, the *helicopter*, *green*, and *top* concepts stabilize at a higher firing rate, and the *Yes* population correctly answer the query, as shown in the figure.

Finally, under the same conditions, the system is queried "Is there a green car?" (see Figure 6.7). In this case, the *green* and *car* concepts are both active, as something green and a car exist in the retinal input. However, top-down activations follow their respective voltage-dependent paths, and the feedback is diffused over two locations in the LGN.

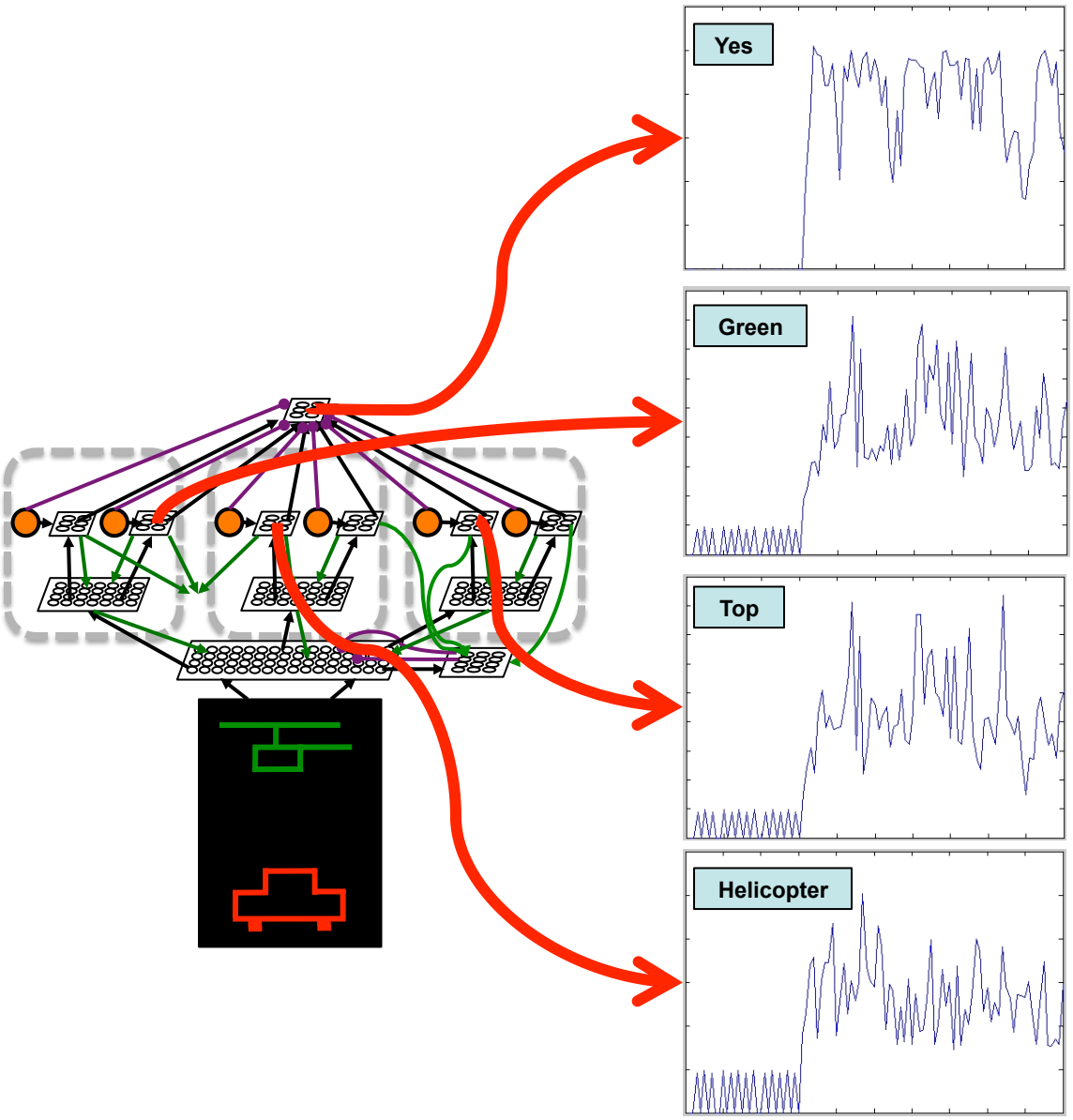


Figure 6.6: The system is asked "Is there a green helicopter on top?". Feedback activations converge to the same neurons in the LGN, balancing the inhibition of the NRT. All three populations stabilize to a high firing rate, and the *Yes* population is activated.

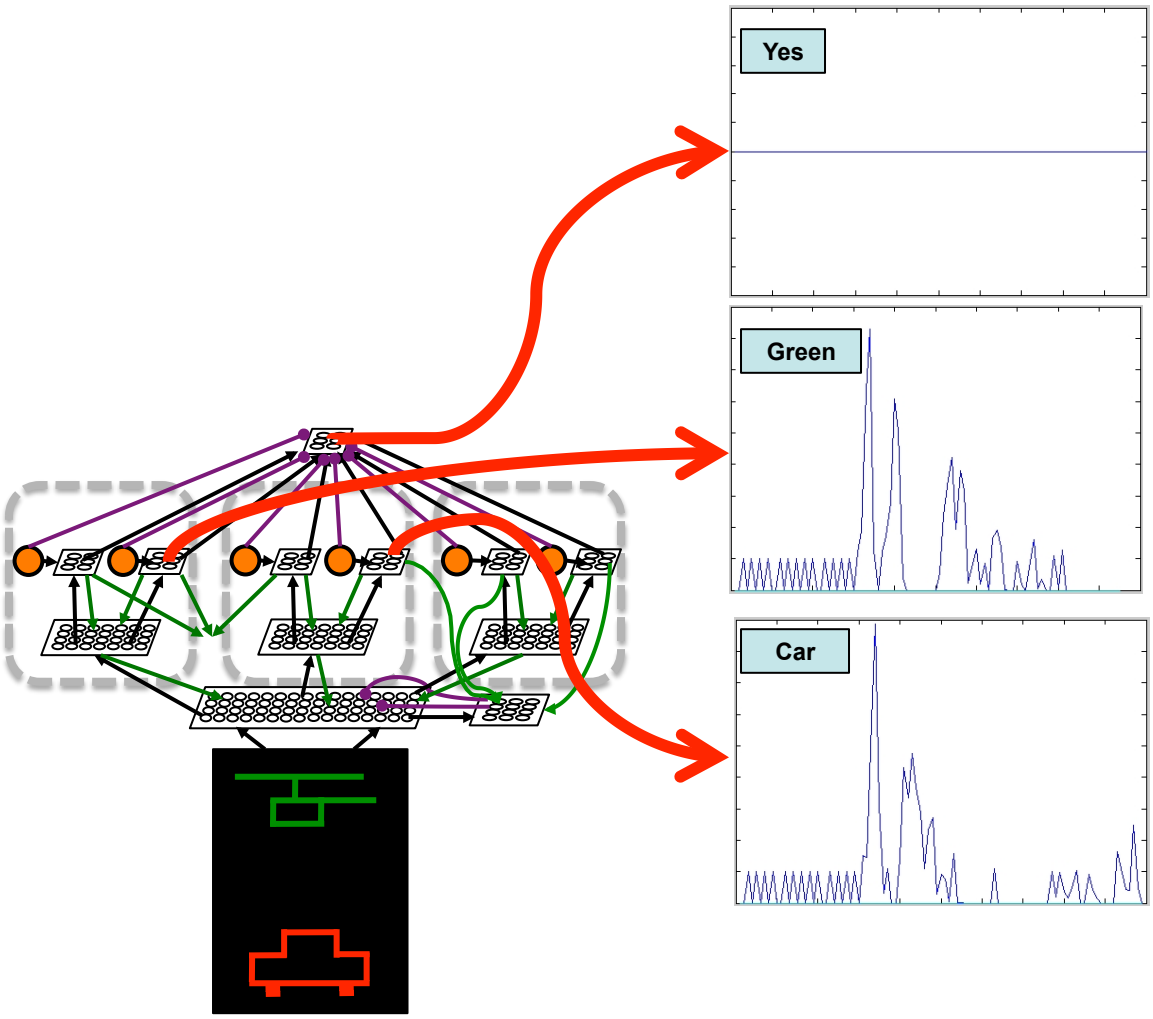


Figure 6.7: The system is asked "Is there a green car?". Feedback activations propagate to different neurons in the LGN, and therefore, feedback is unable to balance out the inhibition of the NRT. As a result, neither the *green* or *car* concepts stabilize to a high firing rate, and the *Yes* population is silent (implied "no").

When the firing rate of the NRT is enhanced proportionally to the query, it depresses both locations in the LGN. As a result, there is less voltage-independent feedforward activation to the higher levels in the Visual Cortex model, and both the *green* and *car* concepts do not stabilize at a higher firing rate (see Figure 6.7). As a result, the feedforward excitation to the *Yes* population is insufficient to make it fire; hence, the system correctly answers "no" to this query.

In this simple system, three basic vision-related streams capture six invariant concepts, yet the system can be queried about the location, color, and presence of simple objects. If a feedforward-only architecture was used, the system would need an invariant helicopter detector, a red helicopter detector, a green helicopter detector, a green helicopter on top detector, and so on. The feedforward-only approach would require 20 individual detectors for each of these possible cases. As the number of concepts and processing streams increases, such an approach is clearly quite expensive, requiring a combinatorial explosion of detectors for each possible case. Thus, even at a simple system level, with very few invariant concepts, the advantage of a system organized as a single integrated attractor is quite obvious.

Future work will extend this ability to the large-scale visual cortex model, as well as consider a larger number of invariant concepts across different processing streams. However, the simple example network presented here does not detract from the generality of this approach, which appears promising for larger scale models. By organizing the network as a single integrated hierarchical attractor, the attractor state specifies both the *invariant* concepts in the higher modeled regions as well as the neurons firing for the *specific*

details of the retinal input. As demonstrated, this approach works even with the invariant concepts detected by different processing streams, so long as the attractor state allows the top-down enhanced firing rate to converge on the same topographical locations (in this case, the same LGN cells).

6.4 Summary

In this chapter, the abilities of the hierarchical metastable attractor model of the Visual Cortex are explored. Consistent with other research on attractor networks, the Visual Cortex model demonstrates noise resilience, pattern completion, and is capable of leveraging short-term working memory for the task of object recognition. Furthermore, this chapter also considers how the hierarchical attractor architecture *integrates* information across different streams of processing. Thus, the results of this architecture demonstrate a network that is capable of recognizing invariant concepts, but can also bind related concepts in a much more efficient manner than building a specific detector for every possible conjunction of concepts. Future work will necessarily examine the scalability of this approach and investigate a much larger number of concepts; however, the results described in this chapter serve as a proof of concept, demonstrating the potential of a hierarchical attractor network.

7 DEPLOYMENT ON NEUROMORPHIC SUBSTRATE

This chapter considers the complex relationship between neuromorphic hardware and neural models such as the one described in the previous chapter. First, this chapter highlights the *neuromorphic semantic gap* that exists between state of the art software models and the neuromorphic substrates on which they will be deployed. Second, this chapter details the challenges of deploying large scale neural models on neuromorphic hardware. These challenges are studied in detail using IBM's digital Neurosynaptic Core hardware, where simple digital neurons provide no complex neuronal behavior and each Core is limited to 256 digital neurons with limited fan-in and fan-out capabilities.

7.1 The Neuromorphic Semantic Gap

While a number of different neuromorphic substrates have been proposed and implemented [27, 119, 118, 2, 11, 125, 67, 93, 128], the common goal of such hardware is to leverage the beneficial properties of biological neurons and brains. While neuroscientific research has certainly improved the understanding of the brain in recent years, the full functionality of this complex system is still far from understood. In this way, developing a neuromorphic substrate appears to be a continuously moving target; as new scientific discoveries are made, new neuronal functions and structures may prove vital to the information processing that occurs in biological brains. However, the lack of a universal understanding of the brain does not diminish the importance of developing

neurally-inspired hardware today.

Neuromorphic hardware such as IBM's Neurosynaptic Core presents a set of unique features and benefits: a non von Neumann computing substrate composed of neuron-like processing elements, an architecture that avoids the von Neumann bottleneck by storing processing elements and memory together in a distributed and massively parallel way, and ultra-low power event driven computation [94]. However, the hardware implements a simple LLIF neuron lacking nonlinear neuronal behaviors and transcendental function support. In Chapter 4, the foundational visual system model demonstrated how such a simple neuron model was capable of performing motion detection and object recognition at a small scale. However, in Chapter 5, it was proposed that the visual cortex may leverage a number of nonlinear neuronal behaviors, such as connections modulated by short-term plasticity, or voltage-dependent NMDA-mediated synapses. Furthermore, online learning rules, such as Hebbian or STDP, are also important features, allowing neuronal networks to adapt over time. Such neuronal behaviors are inherently more complex than the simple digital neuron primitives implemented in IBM's Neurosynaptic Core hardware. As a result, there exists a *neuromorphic semantic gap* between the biologically inspired model and the hardware substrate on which it is deployed.

To computer architects, the problem of the semantic gap is not a new one. Several decades ago, the same problem was solved in the context of the reduced instruction set computing (RISC) architecture. The algorithms and programs being developed in high-level languages may not have had directly supported primitive on a RISC architecture; yet,

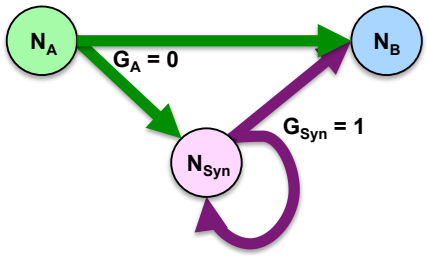


Figure 7.1: NCN circuit emulating short-term potentiation.

NCN	Axon Type	Threshold	S^0	S^1	S^2	Leak	Stochastic Leak
N_A	0	100	120	0	0	10	0
N_B	0	100	20	90	0	10	0
N_{Syn}	1	200	40	204	0	0	5

Table 7.1: One set of parameters for the NCN circuit to exhibit short-term potentiating behavior.

through translation or compilation, these algorithms and programs could be expressed in a way that was compatible with the underlying simple hardware primitives, thus bridging the semantic gap. In a similar vein, the following sections demonstrate that the complex neuronal behaviors leveraged by the Visual Cortex model can effectively be translated to simple digital neuron primitives compatible with the Neurosynaptic Core hardware.

7.1.1 Emulating Short-Term Plasticity

Short-term modulation of synaptic strength, whether potentiating or depressing, enhances the computational power of a neuron, as the same neuron's outputs can exhibit differential effects to different neurons [88]. However, IBM's Neurosynaptic Core features only fixed synapses; connections between neurons are binary, and each neuron is afforded only three synaptic weight values for incoming axons [94, 125, 11]. While it is not possible to configure

a single Neurosynaptic Core Neuron (NCN) to demonstrate this short-term modulation of synapses, simple circuits of NCNs can effectively emulate this behavior.

In its most basic sense, short-term potentiation is marked by a relative increase in synaptic strength as a function of the presynaptic neuron's firing rate. This effect can be emulated using the NCN circuit shown in Figure 7.1. In the figure, NCNs N_A and N_B are the presynaptic and postsynaptic neurons, respectively. A third NCN, N_{Syn}, acts as a detector for NCN N_A's firing rate, and in turn, projects a *potentiating* output to NCN N_B.

Table 7.1 shows one possible configuration of the NCN parameters, which are compatible with IBM's Neurosynaptic Core hardware. NCN N_A projects an axon of type 0 to N_B, which uses a small positive synaptic weight for this connection. As a result, when N_A spikes, the effect on N_B is fairly small. Since the primitives provided by the Neurosynaptic Core hardware mean that each NCN has only one output axon (and it must be assigned a static axon type), N_A also projects an axon of type 0 to the synapse NCN N_{Syn}. The firing threshold and linear leak parameters of N_{Syn} are set such that N_{Syn} will only fire if N_A outputs spikes at a fairly high rate. N_{Syn} projects an axon of type 1 to N_B, which uses a strong synaptic weight value for a potentiating effect. This axon also synapses with N_{Syn} with a large positive synaptic weight value, allowing the potentiating effect to last without the need to accumulate many spikes from the presynaptic N_A again. Finally, N_{Syn} uses a stochastic leak, which ensures that after N_A has stopped firing for a period, N_{Syn} too will leak back to its resting potential. Given these ingredients, the NCN assembly shows short-term potentiating effects when the presynaptic NCN spike rate is high, but eventually

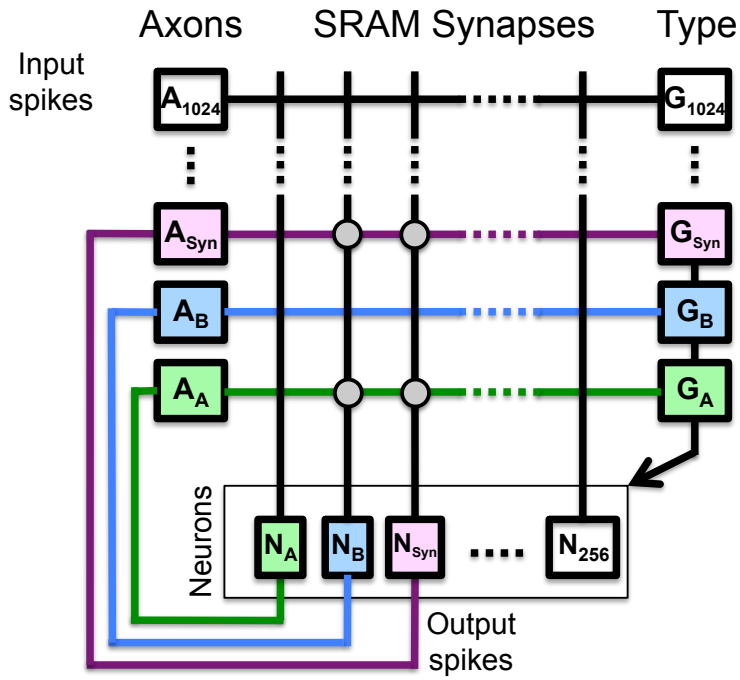
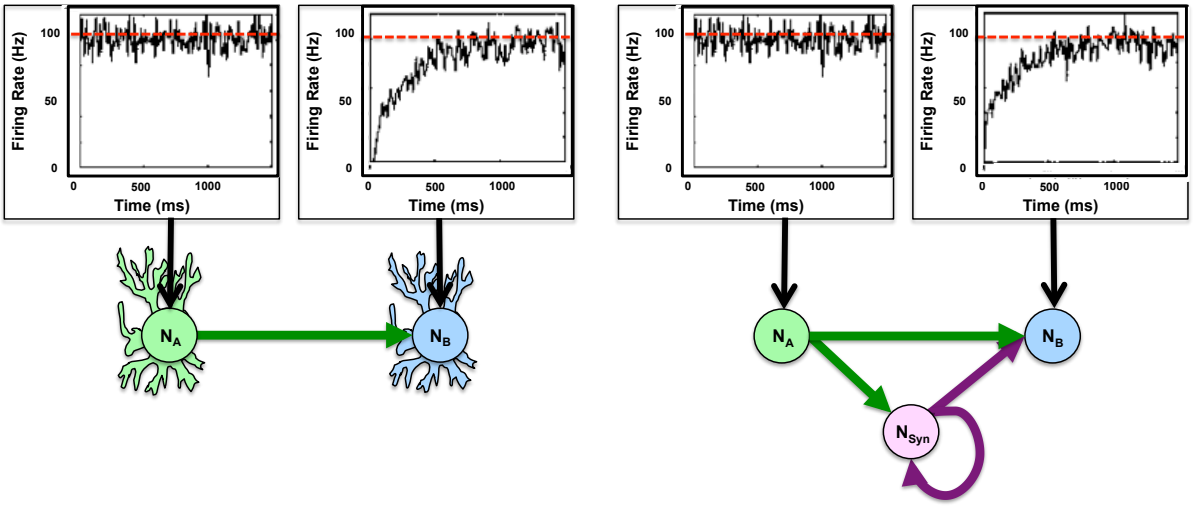


Figure 7.2: Short-term potentiation circuit deployed on Neurosynaptic Core. Each neuron is assigned one axon with a parameterized axon type.

returns to normal synaptic strength once the presynaptic neuron rests. Figure 7.2 illustrates how the NCN circuit maps onto IBM's Neurosynaptic Core.

To validate the NCN circuit for short-term potentiation, its functionality is compared with the software implementation of short-term potentiation presented in Chapter 5. Since the Visual Cortex model uses populations of neurons (rather than individual neurons in each modeled region), this experiment considers the average effect between 100 pairs of presynaptic and postsynaptic neurons exhibiting the complex neuronal behavior. Figure 7.3 (a) shows the desired behavior of short-term potentiation. The average firing rate of the presynaptic neurons is 100 Hz. As a result of the short-term potentiating synapses, the average firing rate of the postsynaptic neurons grows over time, eventually reaching 100



(a) (b)

Figure 7.3: (a) Short-term potentiation behavior modeled in software. (b) Short-term potentiation behavior of NCN circuit.

Hz as well.

Figure 7.3 (b) shows the behavior of the NCN circuits emulating short-term potentiation. As before, the average firing rate of the presynaptic neurons is 100 Hz. In turn, once the N_{Syn} NCNs have become active due to the high rate of the presynaptic NCN, the firing rate of the postsynaptic NCNs (N_B) grows to 100 Hz as well. From the standpoint of functionality, the behavior of the postsynaptic NCN population is equivalent to that shown in Figure 7.3 (a). Importantly, this demonstrates that while the primitives of an individual digital NCN do not allow for such nonlinear functions, the aforementioned neuromorphic semantic gap can be bridged using additional NCNs to emulate the appropriate behavior.

Similarly, short-term depression (that is, a temporary decay in synaptic strength as a function of presynaptic firing rate) can also be emulated using a circuit of digital NCNs, as shown in Figure 7.4. Table 7.2 shows one possible configuration of the NCN parameters

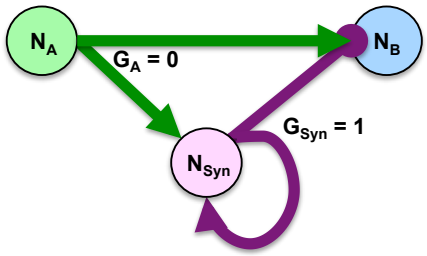


Figure 7.4: NCN circuit emulating short-term depression.

NCN	Axon Type	Threshold	S^0	S^1	S^2	Leak	Stochastic Leak
N_A	0	100	120	0	0	10	0
N_B	0	100	100	-120	0	10	0
N_{Syn}	1	200	40	204	0	0	5

Table 7.2: One set of parameters for the NCN circuit to exhibit short-term depressing behavior.

to emulate short-term depression. NCN N_A projects an axon of type 0 to N_B, which uses a positive synaptic weight value. This synaptic weight value is chosen to be the baseline synaptic strength between the two NCNs. N_A also projects the same axon type to N_{Syn}, which, as before, uses a threshold and leak parameter such that it fires only when the firing rate of NCN N_A is high. N_{Syn}, in turn, projects an axon of type 1 to N_B, which uses a negative synaptic weight value. As N_{Syn} becomes activated, it essentially cancels out the excitatory input received from N_A (to whatever degree is desired, as the synaptic weight values are user-defined parameters). As before, N_{Syn} uses a strong positive recurrent connection to allow the depressing effect to last without the need to accumulate many input spikes from N_A, and a stochastic leak to ensure that after N_A has stopped firing for a period, N_{Syn} leaks back to its resting state.

The short-term depression NCN circuit is experimentally validated against the software

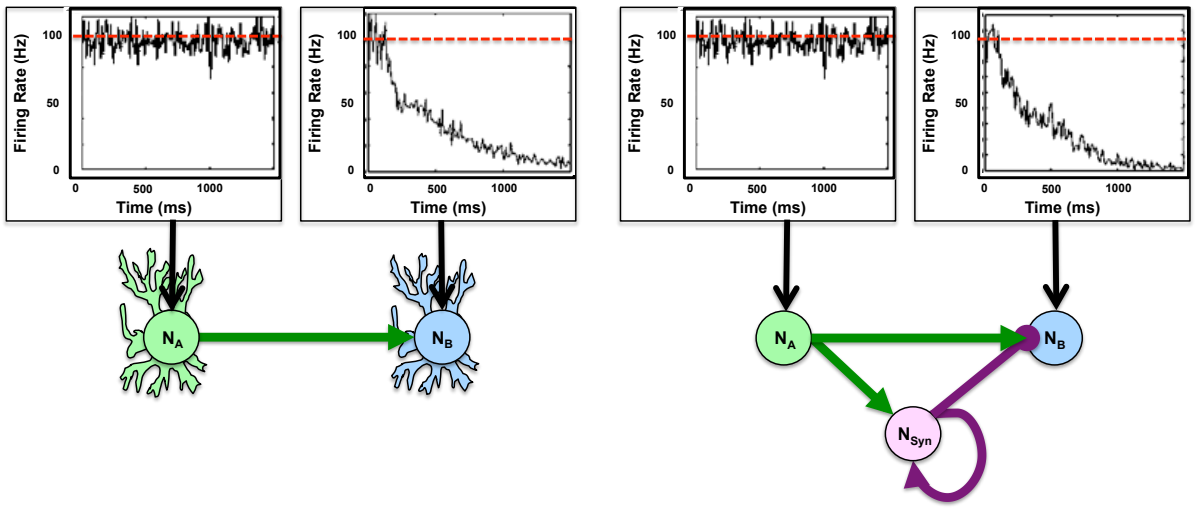


Figure 7.5: (a) Short-term depressing behavior modeled in software. (b) Short-term depressing behavior of NCN circuit.

implementation of short-term depression. As before, the experiment considers the average effect between 100 pairs of presynaptic and postsynaptic neurons. Figure 7.5 (a) shows the desired effect of short-term depression obtained with the software implementation of the synaptic modulation. The average firing rate of the presynaptic neurons is 100 Hz, while the average firing rate of the postsynaptic neurons is initially 100 Hz, but significantly decays. Figure 7.5 (b) shows the average behavior of the 100 NCN circuits emulating this effect. As with short-term facilitation, the functional response of this NCN circuit is equivalent, again demonstrating that the semantic gap can be bridged by using an extra NCN per synapse and appropriately configuring the NCN parameters. Furthermore, this NCN circuit is mapped to the Neurosynaptic Core in the exact same way as presented in Figure 7.4; only the synaptic weight value for the depressing effect is different.

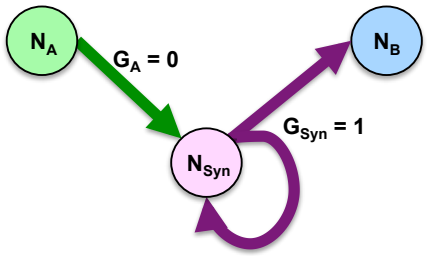


Figure 7.6: NCN circuit emulating the prolonged signalling of NMDA-mediated synapses.

NCN	Axon Type	Threshold	S^0	S^1	S^2	Leak	Stochastic Leak
N_A	0	100	120	0	0	10	0
N_B	0	100	0	120	0	10	0
N_{Syn}	1	155	255	255	0	0	100

Table 7.3: One set of parameters for the NCN circuit to exhibit the prolonged signalling effects of NMDA.

7.1.2 Emulating the Long Timescale Effects of NMDA-Mediated Synapses

For many of the connections in biological nervous systems, a spike from a presynaptic (i.e. source) neuron affects the postsynaptic (i.e. target) neuron for a brief time, typically less than a few milliseconds. The NCNs of the Neurosynaptic Core capture the short timescale effects of these spikes, as each spike lasts for a single time step of the digital hardware (1 ms). However, as discussed in Chapter 5, the Visual Cortex model leverages the prolonged signalling effects captured by NMDA-mediated synapses. These prolonged signalling effects are important, since they provide spiking neurons with a mechanism to integrate temporally correlated activity across spatially distributed neurons.

Figure 7.6 shows an NCN circuit which emulates this prolonged signalling effect. Ta-

ble 7.3 shows one possible configuration of the NCN parameters to emulate the effect. As before, NCNs N_A and N_B are the presynaptic and postsynaptic neurons, respectively. An intermediate NCN, N_{Syn}, is inserted between the presynaptic and postsynaptic NCNs. The threshold of N_{Syn} is set below the value of the synaptic weight value of axon 0 from N_A, such that it fires immediately after the firing of N_A. N_{Syn} uses a recurrent connection to itself with a strong synaptic weight value, so that even after N_A has stopped firing, N_{Syn} will continue to fire for an extended period. N_{Syn} uses a stochastic leak to ensure that it returns to its resting potential when N_A is silent for an extended period of time.

As with short-term plasticity, the NCN circuit emulating the prolonged signalling of NMDA-mediated synapses is validated against a software model of an NMDA-mediated synapse. As before, the results presented consider the average behavior of 100 pairs of presynaptic and postsynaptic neurons. In Figure 7.7 (a) shows a presynaptic neurons, which initially exhibits a firing rate of 100 Hz for 500 ms, followed by silence. The postsynaptic neurons (here, assuming voltage independent connections for simplicity) show a 100 Hz firing rate for the first 500 ms, followed by a slow decay in the average postsynaptic neuron firing rate. This demonstrates that the effect on the postsynaptic neurons last beyond the instantaneous spike of the presynaptic neurons. In Figure 7.7 (b), the NCN circuits show, on average, a qualitatively similar effect. The postsynaptic NCNs show an initially high firing rate, followed by a decaying effect that lasts beyond the instantaneous spikes of the presynaptic NCNs.

In the Visual Cortex model, each presynaptic neuron which projects top-down NMDA-

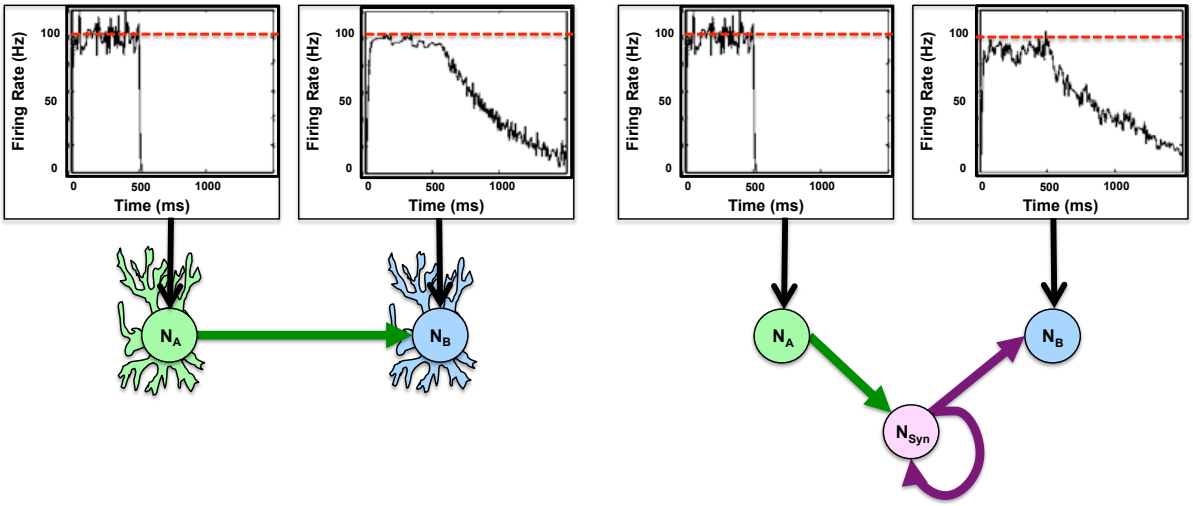


Figure 7.7: (a) NMDA prolonged signalling behavior modeled in software. (b) NMDA prolonged signalling behavior of NCN circuit.

mediated synapses requires one additional N_{Syn}. The voltage dependent behavior of NMDA-mediated synapses is captured by NCN circuits on the postsynaptic neurons, discussed below.

7.1.3 Emulating the Voltage-Dependence of NMDA-mediated Synapses

As was discussed in Chapter 5, synapses mediated by NMDA receptors are voltage-dependent; that is, the postsynaptic neuron only integrates inputs on NMDA-mediated synapses if it has already been depolarized by other voltage-independent inputs. This "conditional integration" appears to be an important feature, especially in the context of top-down connections. In this way, top-down signalling can be both very general and

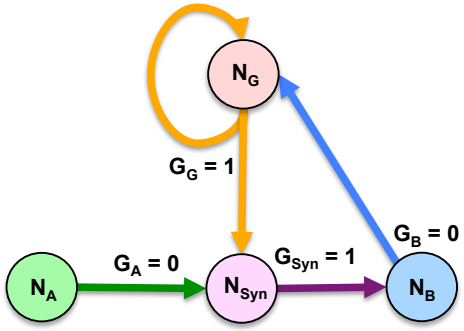


Figure 7.8: NCN circuit realizing voltage-dependent synapses.

NCN	Axon Type	Threshold	S^0	S^1	S^2	Leak	Stochastic Leak
N_A	0	100	120	0	0	10	0
N_B	0	100	120	0	0	10	0
N_{Syn}	1	99	50	50	0	50	0
N_G	1	155	255	255	0	0	100

Table 7.4: One set of parameters for the NCN circuit to display the voltage-dependence of NMDA-mediated synapses.

diverging in structure, and voltage-dependent connections ensure that this top-down signalling is consistent with feedforward spiking evidence.

In the Visual Cortex model, this voltage-dependent behavior is captured by first integrating all voltage-independent inputs, checking whether the membrane potential is above a depolarization threshold (not to be confused with the neuron's firing threshold), and then integrating the voltage-dependent inputs if the threshold is exceeded. The Neurosynaptic Core neurons do not feature a secondary "conditional integration" threshold as it has been described. Rather, NCNs exhibit only voltage-independent inputs: if a connection exists between two neurons, the postsynaptic neuron will integrate spikes from the presynaptic neuron unconditionally.

However, voltage-dependent synapses can be modeled using a circuit of NCNs, as

shown in Figure 7.8. Table 7.4 shows one possible configuration of the NCN parameters to emulate the effect. As with previous NCN circuits, N_A and N_B are the presynaptic and postsynaptic neurons, respectively. Two additional NCNs, N_{Syn} and N_G are required. N_{Syn} acts as the actual synapse, while N_G is used to detect the initial depolarization of the postsynaptic NCN. Initially, both the synapse gate N_G and the synapse N_{Syn} NCNs are inactive. Even if the presynaptic N_A exhibits a high firing rate, the postsynaptic N_B remains unaffected. However, if N_B fires once, indicating that it has received voltage-independent evidence on another synapse, it in turn activates the synapse gate N_G. N_G uses a strong self-connection to remain active for an extended period of time. N_{Syn} exhibits a very high leak parameter, and only fires during coincident inputs from the synaptic gate N_G and the presynaptic NCN N_A; in turn, it passes on the presynaptic spike to the postsynaptic NCN. As with other NCN circuits described above, N_G uses a stochastic leak parameter such that, after the postsynaptic N_B has been inactive for a period of time, the voltage-dependent gate closes (and thus, the postsynaptic NCN must again be depolarized before the top-down NMDA signalling can have an effect). Alternatively, the circuit can be constructed such that the synapse gate N_G receives the same voltage-independent inputs as N_B to detect depolarization, thus allowing the NMDA-mediated synapses to have an effect even before N_B fires.

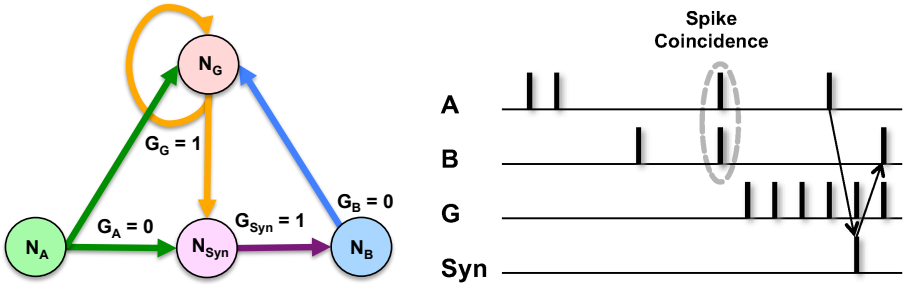


Figure 7.9: (a) NCN circuit emulating Hebbian learning. (b) Spike-time plot of emulated Hebbian learning.

NCN	Axon Type	Threshold	S^0	S^1	S^2	Leak	Stochastic Leak
N_A	0	100	100	0	0	10	0
N_B	0	100	0	100	0	10	0
N_{Syn}	1	100	50	50	0	50	0
N_G	1	100	50	150	0	50	0

Table 7.5: One set of parameters for the NCN circuit to exhibit Hebbian learning.

7.1.4 Emulating Online Learning

Beyond the complex neuronal mechanisms described above, the Visual Cortex model described in Chapter 5 can be trained with online learning. Spike time dependent plasticity (STDP) and Hebbian learning have been identified as two key learning rules supported by biological neurons [33, 13, 130]. Again, the lack of online learning on the Neurosynaptic Core presented by Merolla et al. [94, 11] demonstrates the semantic gap between the Visual Cortex model and the target substrate. However, the following sections demonstrate how NCN circuits can effectively emulate online plasticity.

7.1.4.1 Emulating Hebbian Learning

Hebbian learning has often been described by the adage "neurons that fire together, wire together" - simply meaning that the co-occurrence of spiking activity of both the presynaptic and postsynaptic neurons strengthens the synapse between them. Considering the non-plastic digital neuromorphic hardware, the simplest type of this behavior can be realized two additional NCNs. Figure 7.9(a) shows a presynaptic neuron N_A, a postsynaptic neuron N_B, a synaptic gating NCN N_G, and the synapse NCN N_{Syn}.

Figure 7.9(b) demonstrates the operation of this Hebbian plasticity NCN circuit, while Table 7.5 shows one possible configuration of the NCN parameters. Before learning, the synaptic gating neuron N_G is silent, while neurons N_A and N_B spike independently from non coincident inputs. Once neurons N_A and N_B fire at the same time, the synaptic gating neuron N_G acts as a coincidence detector and fires in response. The synaptic gating neuron will then fire at 1kHz (i.e. the Neurosynaptic Core time step) due to its strong recurrent connection, acting as a latch which stores the connection between the pre and postsynaptic neurons indefinitely. N_G also sends a spike at each time step to the synapse NCN N_{Syn}. When N_{Syn} detects the coincident spikes between the presynaptic neuron N_A and the gating neuron N_G, it propagates the spike to the postsynaptic neuron N_B.

7.1.4.2 Emulating STDP Learning

To emulate this type of plasticity, even with a very simple interpretation, a significantly more complicated NCN circuit is required, as shown in Figure 7.10. With STDP, synaptic

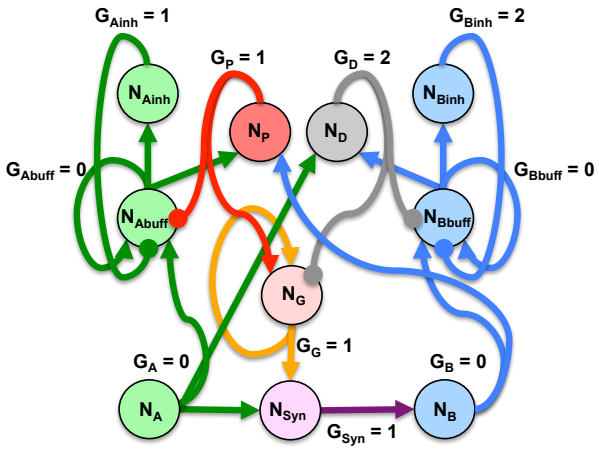


Figure 7.10: NCN circuit emulating STDP learning.

NCN	Axon Type	Threshold	S^0	S^1	S^2	Leak	Stochastic Leak
N_A	0	100	100	0	0	10	0
N_B	0	100	0	100	0	10	0
N_{Syn}	1	100	50	50	0	50	0
N_G	1	100	50	150	-150	50	0
N_P	1	100	50	0	0	50	0
N_D	2	100	50	0	0	50	0
N_{Abuff}	0	100	100	-100	0	0	0
N_{Ainh}	1	20	1	0	0	0	0
N_{Bbuff}	0	100	100	0	-100	0	0
N_{Binh}	2	20	1	0	0	0	0

Table 7.6: One set of parameters for the NCN circuit to exhibit STDP learning.

changes occur when both the presynaptic and postsynaptic spikes fall within a short window, typically 20 ms or so [13]. Again, considering the simple digital NCNs of the Neurosynaptic Core, a NCN has no way of remembering its own firing history. However, an additional NCN can be employed to act as a history buffer which remembers that a neuron has fired in the recent past.

In Figure 7.10, NCN N_{Abuff} (N_{Bbuff}) fires whenever the presynaptic (postsynaptic)

NCN fires, and uses a strong self connection to act as a temporary latch. To ensure that STDP operates within a reasonable window of time (typically 20 ms), the history buffer NCN synapses with an inhibitory NCN (N_{Ainh} and N_{Binh}). The inhibitory NCN is given a high threshold and a zero-value leak, such that it accumulates spikes over time from the history buffer NCN; once it fires, it silences (or resets) the history buffer NCN through a connection with a strong inhibitory synaptic weight value. For example, the synapse from N_{Abuff} to N_{Ainh} can use a synaptic weight value value of 1, while the firing threshold of N_{Ainh} is 20; therefore N_{Ainh} ensures the STDP window is limited to 20 ms.

Two additional NCNs are required for detecting spiking events that cause plasticity, one for synaptic potentiation (N_P), and one for depression (N_D). NCN N_P detects the coincident firing of the postsynaptic neuron N_B and the history buffer of the presynaptic neuron N_{Abuff} (i.e. presynaptic before postsynaptic within the time window results in potentiation). Conversely, N_P detects the coincident firing of the presynaptic neuron N_A and the history buffer of the postsynaptic neuron N_{Bbuff} (i.e. postsynaptic before presynaptic within the time window results in depression). Each time either of the plasticity event detectors fire, they also send a strong inhibitory signal to the history buffer NCNs, which simplifies the learning rule and ensures that plasticity occurs only on a presynaptic-postsynaptic spike pair basis.

Finally, two additional NCNs act as the synapse between the presynaptic and postsynaptic neurons. The first is the synaptic gate N_G, which is enabled or disabled by the plasticity detector neurons. N_P activates N_G with a strong excitatory connection, while

N_D disables N_G with a strong inhibitory connection. As with many of the NCN circuits described above, N_G uses a strong self connection to latch itself "active" after a potentiating event. Once N_G has been activated by potentiation, N_{Syn} will fire for any coincident spikes received by N_G and the presynaptic N_A, passing on the spike to the postsynaptic N_B One possible set of NCN parameters to exhibit STDP learning are shown in Table 7.6.

Figure 7.11 shows the behavior of the STDP NCN circuit. In Figure 7.11(a), the synapse between N_A and N_B has not been yet potentiated, as indicated by N_G's silence. The presynaptic neuron N_A fires, and in the next cycle, its history buffer N_{Abuff} latches this spike (as indicated by the train of spikes). Later, the postsynaptic neuron N_B spikes. N_P detects the coincident firing of the postsynaptic neuron N_B and the presynaptic history buffer N_{Abuff}, and in turn, activates N_G. Some time later, N_A spikes again, and the spike propagates through N_{Syn} to the postsynaptic neuron N_B. So long as a synaptic depressing event doesn't occur (as would be indicated by a firing of N_D), the synapses is potentiated indefinitely; every spike from N_A is passed to N_B.

Figure 7.11(b) demonstrates the opposite effect: synaptic depression. In this example, the synapse is initially potentiated, as indicated by the spike train of the gating neuron N_G. The postsynaptic neuron N_B fires first, which is latched by its own history buffer N_{Bbuff}. Some time later, the presynaptic neuron fires. N_D detects the coincident firing (the postsynaptic neuron has fired in the recent past, and the presynaptic neuron just fired), and in turn, inhibits the synaptic gate N_G. Later, the presynaptic neuron spikes; however, since the synapse NCN N_{Syn} is not receiving a coincident spike from N_G, it remains silent, and

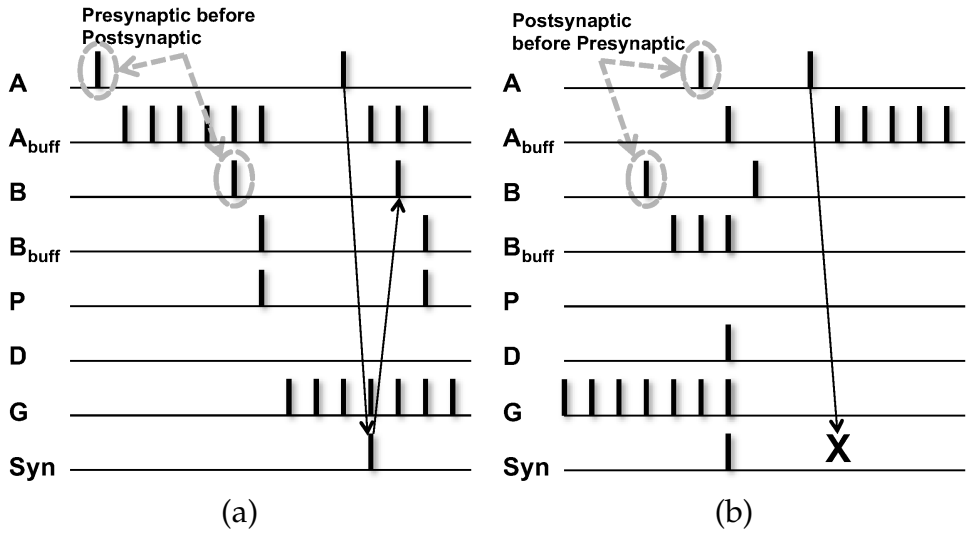


Figure 7.11: (a) Spike-time plot of emulated STDP potentiation. (b) Spike-time plot of emulated STDP depression.

the spike does not propagate to the postsynaptic neuron. So long as a synaptic potentiating event does not occur, the synapse is depressed indefinitely.

7.1.4.3 Extensions to Learning Assemblies

The NCN circuits described above emulate Hebbian and STDP learning on a single synapse, where the strength of the synaptic connection may take on two values: zero (before learning, or in the case of synaptic depression) or a single parameterized value determined at chip configuration (i.e. the NCNs synaptic weight value for the appropriate axon type). However, depending on the learning task at hand, it may be beneficial to have synaptic weights capable of a broader range of values.

In the context of non-learning neuromorphic hardware, such behavior is possible, albeit at a high overhead. As shown in Figure 7.12, multiple NCNs can be recruited for a circuit

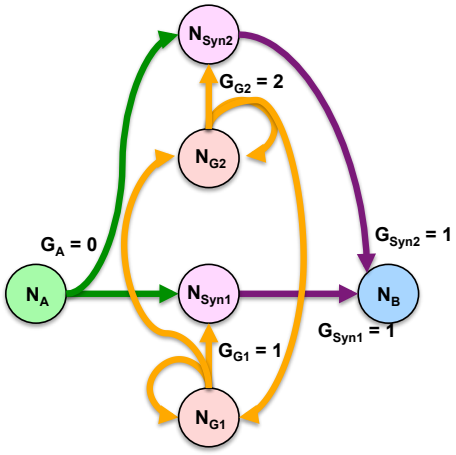


Figure 7.12: NCN circuit emulating chain of synapses.

that acts as a "chain" of plastic synapses between the presynaptic and postsynaptic neurons. This chain of synapses is generalized to work in conjunction with either the Hebbian or STDP assemblies described above. To utilize this chain of synapses for Hebbian learning, each of the synaptic gating NCNs (N_Gs) must receive inputs, and detect coincident spikes, from neurons N_A and N_B. To utilize the chain for STDP learning (Figure 7.10), each of the synaptic gating NCNs must receive connections from the plasticity detectors N_P and N_D.

From an initial state where none of the synapses have been potentiated, N_{G1} is the first to activate after a potentiating event, which allows spikes to propagate through the first "synapse" NCN N_{Syn1}. N_{G1} projects an excitatory connection to N_{G2}. The threshold of N_{G2} is higher than N_{G1} such that it fires only after detecting both a potentiating event and the firing of N_{G1}. An excitatory connection from N_{G2} projects back to N_{G1} such that, when a depressing event occurs, connections are disabled in the opposite order (i.e. if N_{G2} is still activated, but N_{G1} receives inhibitory input from a synaptic depression event, they are balanced, and N_{G1} remains active. However, if N_{G2} is inactive during the same scenario,

N_{G1} will be inhibited and disabled). This scheme generalizes in a way that, while expensive in terms of spiking behavior and number of NCNs required, allows very general plasticity rules to be deployed on hardware designed without online-learning capabilities in mind.

While implementing these learning rules with NCN circuits is expensive in terms of both hardware and spiking behavior, they provide a high degree of parameterization and flexibility. For example, the Hebbian learning assembly can be extended to use history buffers similar to the STDP assembly. This modification allows Hebbian learning to occur over a broader time window. The STDP assembly can also be modified to implement variations of burst-STDP [102, 32, 80] or triplet spike STDP [105].

7.2 Automated Approaches for Neural Network Deployment

While the above sections demonstrate how the neuromorphic semantic gap can be bridged, another set of challenges must be addressed before a neural model can be deployed on neuromorphic hardware. Considering IBM's substrate, each Neurosynaptic Core is composed of only 256 digital neurons with limited fan-in and fan-out capabilities. If one considers the Neurosynaptic Core as a basic building block [106], it becomes clear that a large scale neural model must be partitioned and deployed across many Neurosynaptic Cores. However, the challenge still exists: large scale cortical models must be partitioned and configured such that they can be deployed on such a tiled neuromorphic substrate.

Rather than having the cortical model developer consider the underlying hardware (e.g. the neuron model, the 256 neurons per core, the limited fan-in and fan-out capabilities, etc.), this job is more appropriate for a "compiler-like" tool which can parse the designed cortical model and generate an equivalent network for deployment on multiple Neurosynaptic Cores. The first objective of this compilation tool is to replace each of the complex neuronal behaviors modeled in software (such as NMDA-mediated synapses, or synapses that exhibit Hebbian learning) with the functionally equivalent NCN circuits described above. Next, this tool must partition the cortical model into blocks of 256 neurons (or less) to be deployed on a Neurosynaptic Core. This process can be optimized if different hand-tuned templates are available (as will be discussed below). Once the initial neuron placement is complete, the compiler then must perform the axonal routing between populations that reside on multiple Neurosynaptic Cores. The following section discusses the various solutions that the compiler utilizes to over come the limited fan-in and fan-out of the Neurosynaptic Core. It should be noted that these tools and concepts can be extended to support other neuromorphic hardware and can provide automation capabilities that allow the neural algorithm developer to construct complex cortical network models and easily deploy them onto a neuromorphic hardware.

7.2.1 Templates for Neuronal Populations

Naively, the aforementioned compiler tool can simply parse a cortical network model and assign neurons to a free Neurosynaptic Core without any further thought. However,

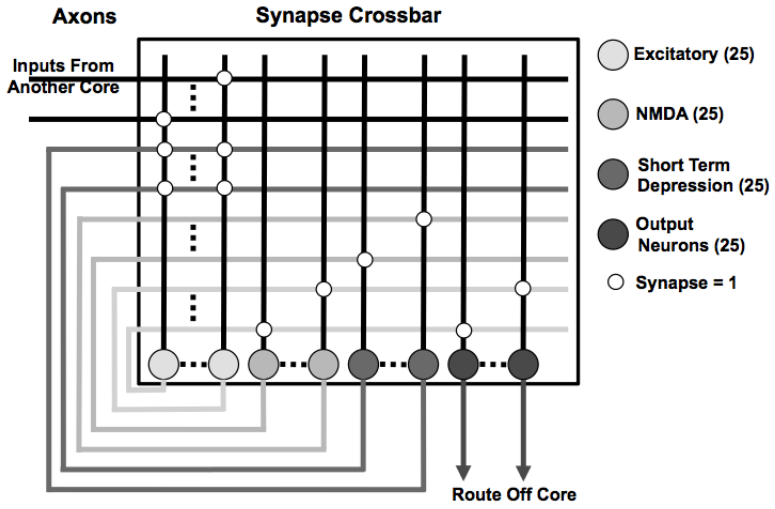


Figure 7.13: An example of a Neurosynaptic Core template from the V4.

for a cortical network model developer with a strong understanding of both the neural algorithm being deployed as well as the underlying neuromorphic substrate, populations of highly interconnected neurons can be grouped into a *template*. These templates are useful for cortical models which are very regular in their structure, connectivities, and other parameters. The Visual Cortex model described in Chapter 5 exhibits high amounts of homogeneous structure within any of the modeled neural areas (e.g. LGN, V4, IT, etc.).

Figure 7.13 shows an example of one of these templates created for the V4 layer of the Visual Cortex model. In the V4, small populations of neurons (in the figure, 25 neurons) exhibit a high degree of regularly structured connectivity, utilize NMDA-mediated and short-term depressing synapses, and share a set of input and output connections. With a large neural layer, regularly structured populations can be easily identified; only a limited number of input and output connections will differ across the neural layer. Each population of neurons matching this template will be deployed together on a single Neurosynaptic

Core. In this way, a designer can create a single template defining how SNN neurons should be matched and deployed on a Neurosynaptic Core, and other populations can take advantage of this hand-tuned organization without requiring the explicit placement of each individual neuron.

When such templates are available, the cortical network model developer only needs to insert commands for template matching when compiling. The compiler, upon recognizing such flags, will organize and place each population matching the template. Another important advantage of this method is, as cortical models are scaled up, the templates created are still relevant so long as the connectivity of the modeled areas remains homogeneous.

7.2.2 Connecting Populations on Distributed Cores

Once an entire cortical network model has been deployed onto Neurosynaptic Cores, the compiler must then perform the appropriate axonal routing to ensure functional equivalence with the unpartitioned SNN. Initially, when a cortical network model is constructed, the network developers do not consider any constraints on the fan-in and fan-out of individual neurons - nor should they. However, considering the hardware constraints outlined in Chapter 3, a NCN can only be assigned a single output axon, and its inputs are limited to the 256 input axons on the Neurosynaptic Core. This imposes serious fan-in and fan-out limitations. Fan-in constraints can be overcome using Firing Population Integration Neurons (FPIs), while the fan-out limitation can be overcome using Copy Neurons and Routing Neurons.

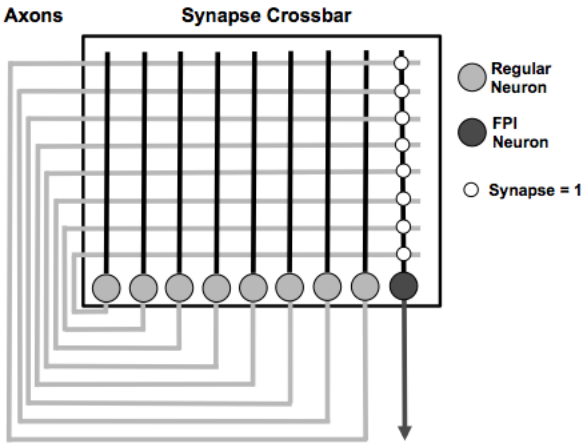


Figure 7.14: An FPI neuron integrates the response of a large neuron population.

Firing Population Integration Neurons, or FPI Neurons, can be used to approximate the signalling between two populations of neurons using a drastically reduced number of axonal projections. As seen in Figure 7.14, an FPI neuron synapses with the axons of a large neuronal population, aggregating their total firing rate and projecting these signals to another Neurosynaptic Core using just one axon. This makes the FPI Neurons an excellent optimization considering the limited fan-in of the Neurosynaptic Core. Since these FPI Neurons integrate over populations of neurons, they do not preserve the actual connectivity of the original cortical network model; as such, the user must indicate directly to the compiler which connections between populations can be replaced with a single FPI Neuron.

To address the fan-out limitations, when the compiler routes the connectivity between the NCNs, it recognizes if a presynaptic neuron must project an axon to two (or more) NCNs on different Neurosynaptic Cores. If there are free NCNs available on the presynaptic neuron's core, a *Copy Neuron* is created. This Copy Neuron replicates all the parameters

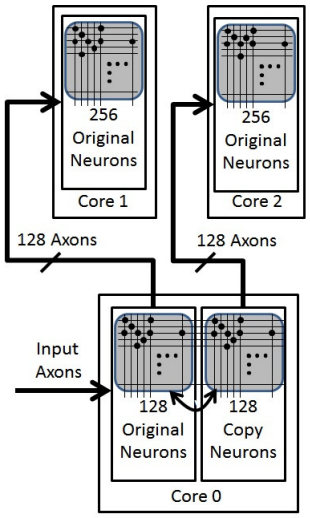


Figure 7.15: Using Copy Neurons for routing.

and dendritic connectivity of the original presynaptic neuron. Hence, the output behavior of both these neurons is functionally equivalent; the only difference is that the axons of both these neurons target different Neurosynaptic Cores. This process is explained in Figure 7.15. In this figure, Core 0 contains 128 neurons referred to as *original neurons* which must project axonal outputs to neurons on both Core 1 and Core 2. When the connection between Core 0 and Core 1 populations is encountered, all 128 axons are projected to Core 1. Next, when the connection between Core 0 and Core 2 neuron populations is encountered, none of the axons from the *original neurons* on Core 0 are available. Since there are 128 free neurons available on Core 0, they are utilized as Copy Neurons, replicating the full set of dendritic connectivities and parameters as the *original neurons*.

However, when an insufficient amount neurons are available to create Copy Neurons, *Routing Neurons* are employed. The concept is quite similar to the Copy Neurons, except these neurons are placed on another Neurosynaptic Core dedicated specifically for routing

the appropriate amount of axonal projections. The only purpose of these Routing Neurons is to forward the activations of the original neuron to multiple target neurons by creating copies of itself. The process of creating Routing Neurons is demonstrated in Figure 7.16. In this figure, two neuron populations reside on Core 0: Population A with 192 neurons and Population B with 64 neurons. Population A projects its axon to Core 1 without any need for additional routing complexity. However, Population B must connect to neurons residing on Cores 2 through 5. Since no free neurons are available on Core 0, a free Neurosynaptic Core (Core 6) is utilized for routing. First, Population B projects its axons to the new Routing Core and creates one-to-one connections with the first free 64 Routing Neurons. These Routing Neurons act as a simple relay, firing for each spike that comes in. Next, the axons of these 64 Routing Neurons on Core 6 route to Core 2. Subsequently, to connect to Cores 3 through 5, additional neurons are created on the Routing Core (Core 6) to achieve the appropriate fan-out. This routing scheme adds a delay of one time step; however, given that spiking neurons collect evidence over multiple time steps, a delay of one time step does not typically impose any functional discrepancies.

The fan-in and fan-out limitations of the Neurosynaptic Core could also be considered part of the *neuromorphic semantic gap* that exists between software models and the neuromorphic hardware on which they are deployed. Cortical network models, like their biological counterpart, can take advantage of thousands of connections [5], while the fan-in and fan-out of a NCN is limited to 256. Furthermore, the NCNs which reside on the same Neurosynaptic Core must share the same set of 256 inputs (i.e. incoming axons). However,

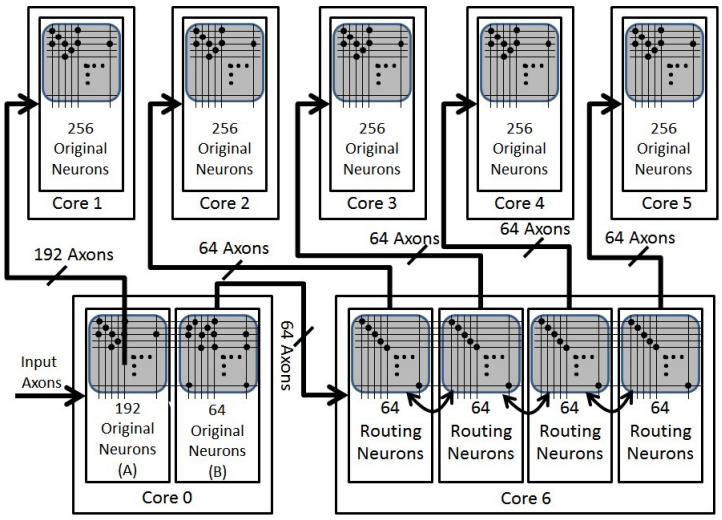


Figure 7.16: Using routing neurons for handling axonal projections.

at present, there is no clear "all-to-all" possible connectivity scheme that could be effectively employed in modern hardware. While mechanisms like the FPI, Routing, and Copy neurons are overhead, they allow a network deployed on multiple Neurosynaptic Cores to leverage large and diverse connectivity patterns without being subject to the limitations of a single Core's fan-in and fan-out.

Alternatively, the SpiNNaker project has proposed custom routers with multicast support to achieve efficient fan-out communication [100]. The clear advantage of this approach is efficient communication of an output spike to thousands of downstream neurons. It was shown that, as fan-out becomes large, the multicast approach (on average) requires an order of magnitude less network resources than a traditional unicast approach. Considering this success likely warrants an investigation of a similar approach for the Neurosynaptic Core.

7.3 Deploying the Visual Cortex Model on Neurosynaptic Cores

Using the NCN circuits, axonal routing techniques, and the compiler-like tool described above, the Visual Cortex model was compiled/translated for deployment on multiple Neurosynaptic Cores. The Visual Cortex model was scaled to approximately 100,000 neurons, many of which used complex neuronal behaviors like NMDA-mediated or short-term plasticity modulated synapses. Furthermore, a number of connections between the modeled V4 and IT regions utilized online Hebbian learning, giving the network the adaptability to learn new categories in the IT region.

Using the compiler-like tool, approximately 200,000 extra NCNs were required to emulate the prolonged signalling and voltage-dependent behavior of NMDA-mediated synapses. Approximately 40,000 NCNs were required to emulate short-term potentiation and depression, while an additional 24,000 NCNs were required to emulate Hebbian learning. To achieve the appropriate axonal projections, a large number of Routing (30,000), Copy (50,000), and FPI (10,000) NCNs were also needed. Thus, a total of 454,000 NCNs are required to implement a model of 100,000 neurons in software (and thus, do not have limitations on connectivity or complex neuronal behaviors).

Ultimately, this means that if the targeted neuromorphic substrate is composed of Neurosynaptic Core like elements, for each complex modeled neuron in software, an average of 4.54 simple digital NCNs must be used to achieve functionally equivalent

behavior. This overhead begs the question whether the simple digital neurons of the Neurosynaptic Core truly capture the correct set of neuromorphic primitives for these types of models. On one hand, if the neural network models are fairly simple (such as those used in traditional neural network engineering applications), the Neurosynaptic Core still appears quite attractive; the hardware provides all the functionality, and few (if any) parameters go unused. On the other hand, it is clear that more biologically realistic cortical models must leverage neuronal behaviors and broad connectivities that are not supported by IBM's Neurosynaptic Core hardware. While this chapter has proved that, in spite of these hardware limitations, large scale cortical models can still be deployed on this neuromorphic substrate, it comes at a substantially high overhead. Adding functionality to the hardware neuron model would likely be more efficient than recruiting extra NCNs; however, if the neural network models being deployed are very simple, such hardware extensions would go largely unused. Therefore, an investigation into the appropriate hardware neuron primitives appears to be an open question for future research. While capturing the power and areal efficiency of biological models will no doubt be a primary goal (as it was with the Neurosynaptic Core design), future implementations of neuromorphic hardware will likely need to consider the types of applications, neuron models, and complex neuronal behaviors they must support.

7.4 Summary

This chapter addresses many of the challenges of deploying a large scale neural model onto a neuromorphic substrate. Considering IBM's Neurosynaptic Core as the target substrate, the first challenge is bridging the *neuromorphic semantic gap* that exists between the simple digital neurons of the hardware substrate and the network model which uses complex neuronal behaviors. The second major challenge is deploying a large-scale model onto Neurosynaptic Cores with limited fan-in and fan-out capabilities and only 256 digital neurons each. The first challenge is overcome by constructing circuits of Neurosynaptic Core neurons that can effectively emulate the desired complex neuronal behaviors, while the second challenge is addressed by developing appropriate routing methods and an automated approach for partitioning a large-scale model across multiple Neurosynaptic Cores.

8 CONCLUSION AND REFLECTIONS

In recent years, a number of high profile neuromorphic projects have emerged [27, 119, 118, 2, 11, 125, 67, 128]. Whether neuromorphic systems exist as hardware accelerators cooperating with traditional von Neumann machines or as stand-alone alternative computing devices, their potential as a fundamentally new and exciting architecture is clear. However, a number of challenges and questions still exist in this domain. What aspects of spiking neuron behavior are computationally useful, and what may simply be an artifact of biological constraints? What are the correct set of hardware primitives that can capture functionally useful behavior, yet still remain energy efficient given current technology? What are the applications that are better solved (or possibly, can only be solved) with systems inspired by the structural and functional properties of biological neurons?

This chapter summarizes this dissertation's contributions towards these challenges, and highlights future contributions that will expand on the work put forward by this dissertation.

8.1 Summary

Neural models, and more recently neuromorphic hardwares, have taken different approaches toward understanding and capturing the power of biological brains. One approach targets extreme biological fidelity, developing neural models that closely mimic every aspect of their biological counterpart [27, 119, 118, 67, 128]; the neuromorphic sub-

strates of this domain seek to develop hardware that allows high biological accuracy for massive cortical simulation. Another approach targets the clearly apparent computational efficiency of biological neurons, focusing on low power implementations with simple abstract models of biological neurons [94, 11, 125].

This dissertation more closely follows the latter method, taking a "bottom-up" approach toward identifying the necessary computational capabilities of spiking neurons. Using only simple leaky integrate-and-fire (LIF) neurons, this dissertation demonstrates how even a minimal model of a spiking neuron can achieve invariant object recognition, motion detection, and even top-down attentional modulation in a hierarchically organized network of neurons.

Building on this minimal framework, this dissertation proposes that the visual cortex can be considered a *hierarchical metastable attractor*, capable of integrating information across different streams of processing in different neural regions. To achieve metastable attractor behavior, however, requires a number of more complex neuronal behaviors that extend the computational capability of the simple LIF model. In this dissertation, several of these complex behaviors are identified. Modeling feedback connection as predominately NMDA-modulated not only matches biological evidence [122], but also allows top-down activations to be modulatory rather than driving in nature (due to the voltage-dependent synapses) and provides a prolonged signalling effect to help attractor states stabilize across spatially separated regions. Short-term potentiation and depression modulated synapses allow neurons to have differential effects on different populations, signal important new

inputs, converge on attractor states rapidly, as well as dissolve rapidly when new feedforward evidence is encountered. With these ingredients, the presented Visual Cortex model is capable of invariant object recognition, pattern completion, and noise resilience. Furthermore, these attractor dynamics allow the network to leverage short-term working memory, affording the model a sense of object permanence in a rapidly changing environment.

Furthermore, the organization of the Visual Cortex model as a hierarchical metastable attractor demonstrates functional integration across different streams of processing. Many traditional neurally-inspired models have shown robust invariant object recognition using only feedforward architectures. However, to achieve robust invariant detection, such approaches essentially "weed out" the details in favor of extracting the gist concept. Conversely, through recurrent connectivity, the Visual Cortex model is capable of top-down excitatory modulation to *access* information regarding the particular details of an object and *route* them to an appropriate output. As was demonstrated in Chapter 6, this type of organization allows the network to integrate information across different streams of processing essentially for free, without the need to consider and build every possible scenario a priori.

Finally, with a justified neuron model and its associated complex synaptic behaviors, this dissertation considers deployment on a state-of-the-art neuromorphic substrate: IBM's Neurosynaptic Core. While the hardware primitives of the Neurosynaptic Core closely match the initial simple implementation of the LIF neuron model, they do not provide support for the more complex behaviors identified, such as NMDA-modulated synapses,

short-term plasticity, or online long-term plasticity. Thus, a *neuromorphic semantic gap* exists between the software models being developed and the hardware substrate on which they will be deployed. However, by using the simple digital Neurosynaptic Core neurons as a basic building block, this dissertation demonstrates how these complex neuronal behaviors can effectively be emulated, effectively bridging the semantic gap. Beyond the semantic gap, the Neurosynaptic Core also has a number of constraints which make it difficult to directly deploy a network model onto the hardware. With only 256 digital neurons per core, and limited axonal fan-in and fan-out capabilities, large scale models must be partitioned and configured across multiple Neurosynaptic Cores. This dissertation presents some of the automated approaches developed to effectively partition a large scale cortical model across multiple Neurosynaptic Cores and perform the axonal routing between them.

8.2 Future Work

The following section outlines some of the various future research avenues that build upon the ideas put forth by this dissertation.

8.2.1 Extending the Visual Cortex Model

The Visual Cortex model presented in this work demonstrates invariant object recognition in spite of noise, incomplete inputs, and spatial translation. However, future extensions to the model should consider invariance to other types of distortions, such as scaling or rotation. The successful HMAX algorithm [111] has already made an argument for utilizing neurons

tuned to different processing scales, especially in the lower modeled neural areas (i.e. the modeled V1 not only has neurons responding to edges of different orientation, but edges of slightly different size). Expanding on the hierarchical attractor based system presented in this work, localized max-pooling operations will allow the construction of scale-invariant feature maps, and ultimately, scale-invariant and translation-invariant representations of objects. Furthermore, the role of top-down voltage dependent connections with be even more important, allowing the system to achieve both highly invariant concepts, but also a high degree of detail specificity which includes features of different scales.

As was presented in Chapter 5, the large scale Visual Cortex model captures the processing of the "form" stream. However, future extensions will add the processing of motion and location as well as color in the large scale model. Many of the ideas regarding simple motion detection, as put forward in Chapter 4, are directly applicable to the large scale model.

8.2.2 Formalizing Large Scale Hierarchical Attractors

The Visual Cortex model, as well as its scaled-down implementation used for the question/answer task, are organized as attractor networks, which ultimately enables their capabilities in pattern completion, short-term working memory, and integration across different processing modalities. However, even in these models, which are orders of magnitude smaller than actual biological networks, it takes a considerable effort to tune connections to achieve the desired behavior of the attractor dynamics.

Other works, which have considered attractor behavior in the context of spiking neurons, have proposed using mean-field techniques to tackle this problem [8, 110]. This approach, modified from statistical physics applications, utilizes a mathematical interpretation of the way populations of neurons can balance each other and achieve stable attractor states. However, to date, mean field techniques have considered only a small number of populations at a time. Though the number of neurons in each population may be quite large, typically the behavior of the neurons within these populations stabilize to the same firing rates. Because the Visual Cortex model presented in this dissertation considers many (dozens to hundreds) populations of neurons in different modeled areas, so far, these mean-field approaches have not been applied. However, future work will consider the applicability of these mathematical formalisms for developing large scale models with many neuron populations.

Future work will also consider the role of plasticity in forming metastable attractor states. As described in this dissertation, Hebbian or burst-STDP plasticity were used to learn invariant representations of simple objects based on the coincident firing of their detected features. However to date, little work has been done to investigate the role of such learning rules for both bottom-up and top-down connections (in terms of both this dissertation as well as neural network research in general). Ultimately, a plasticity rule that adapts both feedforward and feedback connections to achieve metastable attractor states would be quite powerful and alleviate the need to hand-tune the balance of recurrent excitation, inhibition, and complex neural behaviors.

8.2.3 Sequential Learning Between Attractor States

Future work will also investigate the interactions and influence between different metastable attractor states. To a degree, this type of "context sensitivity" already appears in the hierarchical attractor based visual system, since the metastable activity of the higher modeled areas lasts for longer periods of time (and thus the network is able to demonstrate object permanence). However, more extensive influence between attractor states would ultimately allow the network to learn sequences, capably of replaying (or predicting) temporal patters when stimulus input is noisy or weak.

Temporal Difference (TD) learning has been shown in the past to be quite successful at learning sequential patterns for game playing [133, 84]. Future work will consider TD learning, as well as other sequence learning paradigms, to allow episodic-like memory in this system.

8.2.4 Identifying the Appropriate Neuromorphic Primitives

In developing a large-scale model of the visual cortex organized as a hierarchical metastable attractor, a number of complex, but essential, neuronal behaviors were identified. However, the targeted neuromorphic substrate, the Neurosynaptic Core, lacks the hardware primitives necessary to directly implement the types of behaviors; hence, it was shown that circuits of digital Neurosynaptic Core neurons can functionally implement them. While the digital neurons of the Neurosynaptic Core were targeted at extreme power and areal efficiency through simplicity of design, the computational power afforded by these more

complex neuronal behaviors likely justifies a re-examination of the appropriate hardware primitives.

As was previously mentioned, the development of a neuromorphic substrate is somewhat of a moving target. As neuroscientific study advances, researchers gain a better understanding of biological systems and can better identify what is essential and what is simply an artifact of biology. However, given the broad range of neuron model implementations, it may be quite some time before a consensus is reached.

Therefore, perhaps the best approach in developing a neuromorphic substrate at present depends on the scope of its application. For large scale cortical model simulation, the approach taken by analog neuromorphic designs, with their ability to precisely match many neuronal behaviors, is likely appropriate. Other applications have considered neural approaches for more traditional computing workloads such as recognition, mining, and synthesis applications [26]. With little need for extreme biological fidelity, these types of applications instead favor a substrate that is low power and reconfigurable for fault tolerance, therefore favoring a neuromorphic hardware more in line with IBM's Neurosynaptic Core. Finally, brain-inspired applications (but not necessarily brain-modeling applications) such as the Visual Cortex model presented in this dissertation, appear to justify an "in-between" neuromorphic substrate. These types of applications may leverage the functions of biological neurons that are justified in their computational power, and thus, would benefit from efficient hardware support. Future work will consider the tradeoffs between directly implementing NMDA-mediated synapses and short-term plasticity in hardware, and their

implementation as circuits of Neurosynaptic Core neurons.

8.3 Reflections

In this section, I present opinions and thoughts regarding computational models and neuromorphic hardware inspired by the brain. These ideas are based on my research and experience during the past five years of graduate study. I do note that these opinions are my own, and may not reflect the ideas and opinions of the many co-authors I have collaborated with over the years.

8.3.1 Need for Better Neural Programming

A growing body of research has shown many applications and problems that can be solved using neural implementations, several of which have been discussed in this dissertation. Vision related tasks, speech recognition, and robotics applications have all used neural networks; even more traditional computing workloads like file compression and chip layout optimization can be done with neural implementations [26].

However, to date, developing these types of applications requires a reasonable amount of effort and skill. Even when considering the well-understood traditional multilayered perceptron networks trained with backpropagation - a significant amount of hand-tuning of parameters is required to achieve optimal results. Considering the large-scale models developed in this dissertation, an even greater effort is required to balance the excitation and inhibition for the system to exhibit metastable attractor states.

For neuromorphic hardware or neural accelerators to ever become widely-used, we need better and more generalized programming models for these types of applications. There is certainly no shortage of neural network simulators; however, most of these still require the programmer to specify many neuron parameters, network population sizes, and connectivity schemes. If one spends the better part of a graduate degree developing many network models, this is no problem; however, to gain a solid understanding of these ideas would require a significant ramp-up time for the typical programmer.

Rather, what is truly needed is a way to leverage neurally-inspired computing and algorithms in a way that doesn't first require an explicit knowledge of the structure, the semantics of the neuron behavior, and learning algorithms. Even having a general way to specify the connectivity/interaction between neuron populations would significantly enhance the programmability of these types of models. For example, rather than explicitly writing for-loops in code to generate a population of topographically-connected feature detectors, one could simply specify, "I would like a population of vertical and horizontal edge detectors of various scales with receptive fields that overlap by 2 pixels". Furthermore, as mentioned above, when working with spiking neuron models (especially in the context of attractor-state networks), one would rather have an easy way to balance connectivities and firing rates, rather than hand-tuning connections until the desired behavior is achieved. In this dissertation, I do not provide any solutions to this problem, but rather just point to it as one of the current big challenges of neural application development that will necessarily need to be addressed for neuromorphic hardwares to become widely adopted in the future.

One related idea is the concept of a neuromorphic instruction set architecture, or NISA, as was proposed in [59]. The general notion of a NISA, as with the ISA of traditional von Neumann machines, is that an intermediate representation should captures the structure, semantics, and state of an abstract computational machine. Ideally, the development and widespread-adoption of a particular NISA would help generalize neural applications, and allow them to be deployed on different substrates (whether neuromorphic, or traditional von Neumann) [59].

8.3.2 Need for Flexible Neuromorphic Substrates

As outlined in Chapter 7, one of the target goals of my dissertation research was to develop large-scale cortical models capable of real-world tasks, including invariant object recognition, motion detection, pattern completion, and ultimately, scene-understanding. This task is inherently different from the approach of other large-scale cortical models, which simply aim at deploying a large number of spiking neurons and connecting them in a way that reflects the statistical connectivity of biological brains [9]. While building, simulating, and deploying such models may demonstrate the *weak scaling* capabilities of the underlying hardware, they fail to address many of the challenges addressed in this dissertation. As a reminder, the primary hardware challenges addressed in this dissertation include addressing the *neuromorphic semantic gap* and overcoming the fan-in, fan-out, and routing limitations of the Neurosynaptic Core hardware. Drawing a parallel to the semantic gap that von Neumann machines addressed decades ago, the *neuromorphic semantic gap*

is the difference between the neuron models and algorithms being developed and the neuromorphic hardware primitives on which they will be deployed. For example, the Neurosynaptic Core primitives lack the capability to directly implement many of the important complex neuronal behaviors investigated in this dissertation, including NMDA-mediated synapses and short-term plasticity. Furthermore, achieving a functionally-equivalent neural network connectivity between the software model and its deployment on hardware requires a significant effort. While each Neurosynaptic Core neuron can receive 256 inputs, all 256 digital neurons on a single core must share the *same* 256 inputs. Combined with the limitation that each digital neuron can project only one axon (whether to its own core, or another core) means that any network (whether architected, or randomly connected) must be translated into a structure that fits the rigid constraints of the hardware.

While IBM's Neurosynaptic Core hardware nonetheless shows an impressive design that is dense and ultra-low power [94], it is my opinion that, given the current state of neural models and applications, a more flexible neuromorphic hardware is more useful. While a neuromorphic substrate in line with the FACETS/BrainScaleS hardware may avoid the aforementioned neuromorphic semantic gap with its analog circuit neuron implementations, it has its own challenges (though I note that, not having worked with the FACETS hardware, perhaps there are still differences between the hardware and software models that qualify as part of the neuromorphic semantic gap). Developing large-scale analog and mixed-circuit substrates obviously incurs challenges absent from a purely digital CMOS designs, and wafer-level integration is an interesting problem in its own

right. Furthermore, the availability of the more biologically accurate analog neurons may be overkill for many neurally-inspired applications (though, it should be noted that the primary motivation for the development of the FACETS hardware is to perform accelerated cortical simulations, and not necessarily applications like the vision and scene understanding tasks described in this dissertation).

Given how much room there is for future work in both cortical modeling and neurally-inspired application development, it is my opinion that at present, a neuromorphic substrate in line with the SpiNNaker project seems most appropriate. Since neurons are modeled in software on low-power ARM cores, they allow a flexibility in neuron-model design that is absent from the dedicated hardware-neuron designs [108]. Furthermore, a custom interconnect with multicast capability allows for efficient communication between neurons [100], a feature that is desirable for large-scale cortical modeling (where the fan-in and fan-out of a neuron is on the order of 10,000 connections). Considering how little is known (with absolute certainty) in regard to neurons, the plasticity rules that govern their organization, and the elements of their structure and behavior that are necessary (and not simply artifacts) for information processing and functional integration, this combination of traditional von Neumann machines and an interconnect which supports the communication patterns of biological brains seems to be the most broadly applicable to both scientific study and brain-inspired engineering applications.

In other work, we specifically investigated neural network implementations of traditional computing workloads, such as image classification, file compression, chip layout

optimization, and financial applications [26]. In this context, a neuromorphic hardware accelerator with dedicated hardware neurons (possibly in line with IBM's Neurosynaptic Core) would be highly attractive from an energy consumption standpoint; if the hardware also allows for plasticity and retraining, such a design is also capable of overcoming hardware faults and defects. If the set of applications (or at least the general classes of applications) are known *a priori*, a custom hardware accelerator can be developed without worry of a neuromorphic semantic gap; though one would likely never consider deploying a large-scale cortical model on such a substrate. At a finer granularity (small segments of code), other researchers have proposed neural accelerators for *approximate computing*, gaining energy efficiency by trading off high precision computation [38].

8.3.3 Summary

With an ever-growing interest in understanding the brain, paired with the current and future challenges faced by the von Neumann computing model, the fields of neurally-inspired applications and neuromorphic hardware development are poised to succeed where neural networks of the past have failed. These research endeavors still include significant challenges; neural applications will only flourish if they can be widely adopted by general programmers, and neuromorphic hardwares must be flexible enough that they can accommodate the latest understanding of the computational power of biological neurons. However, considering the capabilities of biological brains, the potential for these neurally-inspired applications, paired with an energy efficient and fault tolerant

neuromorphic substrate, appears quite promising for future computing systems.

BIBLIOGRAPHY

[1] Progress report on model developments and comparison to experimental data. http://facets.kip.uni-heidelberg.de/jss/DisplayPublicDeliverables/, 2007. retrieved: April, 2013.

[2] A universal spiking neural network architecture (spinnaker). http://apt.cs.man.ac.uk/ projects/ SpiNNaker/, 2010. retrieved: Oct, 2011.

[3] The board: Neurogrid. http://www.stanford.edu/group/brainsinsilicon/neurogrid.html, 2013. retrieved: April, 2013.

[4] The brainscales project. http://www.artificialbrains.com/brainscales, 2013. retrieved: April, 2013.

[5] L. F. Abbott, J. A. Varela, K. Sen, and S. B. Nelson. Synaptic depression and cortical gain control. *Science*, 275(5297):221–224, 1997.

[6] A. Akrami, Y. Liu, A. Treves, and B. Jagadeesh. Converging Neuronal Activity in Inferior Temporal Cortex during the Classification of Morphed Stimuli. *Cerebral Cortex*, 19(4):760–776, Apr. 2009.

[7] L. Albantakis and G. Deco. Changes of mind in an attractor network of decision-making. *PLoS computational biology*, 7(6):e1002086+, June 2011.

[8] D. J. Amit and N. Brunel. Model of global spontaneous activity and local structured activity during delay periods in the cerebral cortex. *Cerebral cortex (New York, N.Y. : 1991)*, 7(3):237–252, Apr. 1997.

[9] R. Ananthanarayanan, S. K. Esser, H. D. Simon, and D. S. Modha. The cat is out of the bag: cortical simulations with 109 neurons, 1013 synapses. In *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, SC '09, pages 63:1–63:12, New York, NY, USA, 2009. ACM.

[10] A. Anzai, X. Peng, and D. C. Van Essen. Neurons in monkey visual area V2 encode combinations of orientations. *Nat Neurosci*, 10(10):1313–1321, Oct. 2007.

[11] J. Arthur, P. Merolla, F. Akopyan, R. Alvarez, A. Cassidy, S. Chandra, S. Esser, N. Imam, W. Risk, D. Rubin, R. Manohar, and D. Modha. Building block of a programmable neuromorphic substrate: A digital neurosynaptic core. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pages 1 –8, june 2012.

[12] D. Attwell and S. B. Laughlin. An energy budget for signaling in the grey matter of the brain. *J Cereb Blood Flow Metab*, 21(10):1133–1145, Oct 2001.

[13] G. Q. Bi and M. M. Poo. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *J Neurosci*, 18(24):10464–10472, Dec 1998.

[14] D. Bibitchkov, J. Herrmann, and T. Geisel. Effects of short-time plasticity on the associative memory. *Neurocomputing*, 44–46(0):329 – 335, 2002.

[15] A. A. Borbely and P. Achermann. Sleep homeostasis and models of sleep regulation. *J Biol Rhythms*, 14(6):557–568, Dec 1999.

[16] S. Borkar and A. A. Chien. The future of microprocessors. *Commun. ACM*, 54:67–77, May 2011.

[17] R. T. Born and D. C. Bradley. Structure and function of visual area MT. *Annual Review of Neuroscience*, 28:157–189, 2005.

[18] J. Braun and M. Mattia. Attractors and noise: Twin drivers of decisions and multistability. *NeuroImage*, 52(3):740 – 751, 2010. Computational Models of the Brain.

[19] D. Brüderle, E. Müller, A. Davison, E. Muller, J. Schemmel, and K. Meier. Establishing a novel modeling tool: a python-based interface for a neuromorphic hardware system. *Frontiers in neuroinformatics*, 3, 2009.

[20] E. Bryson and Y. C. Ho. *Applied optimal control: optimization, estimation, and control*. Blaisdell Publishing Company, 1969.

[21] N. Burgess. The Hippocampus and Associative Memory.

[22] W. Calvin. Cortical columns, modules, and hebbian cell assemblies. In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 269–272. MIT Press, Cambridge, MA, 1998.

[23] S. Carver, E. Roth, N. J. J. Cowan, and E. S. S. Fortune. Synaptic Plasticity Can Produce and Enhance Direction Selectivity. *PLoS Computational Biology*, 4(2), Feb. 2008.

[24] B. Chapman, M. P. Stryker, and T. Bonhoeffer. Development of orientation preference maps in ferret primary visual cortex. *Journal of Neuroscience*, 16:6443–6453, 1996.

[25] C. Chen, D. M. Blitz, and W. G. Regehr. Contributions of receptor desensitization and saturation to plasticity at the retinogeniculate synapse. *Neuron*, 33(5):779 – 788, 2002.

[26] T. Chen, Y. Chen, M. Duranton, Q. Guo, A. Hashmi, M. Lipasti, A. Nere, S. Qiu, M. Sebag, and O. Temam. Benchnn: On the broad potential application scope of hardware neural network accelerators. In *IEEE international symposium on workload characterization*, IISWC 2012, 2012.

[27] S. Choudhary, S. Sloan, S. Fok, A. Neckar, E. Trautmann, P. Gao, T. Stewart, C. Eliasmith, and K. Boahen. Silicon neurons that compute. In *Proceedings of the 22nd international conference on Artificial Neural Networks and Machine Learning - Volume Part I*, ICANN'12, pages 121–128, Berlin, Heidelberg, 2012. Springer-Verlag.

[28] S. Chung, X. Li, and S. B. Nelson. Short-term depression at thalamocortical synapses contributes to rapid adaptation of cortical sensory responses in vivo. *Neuron*, 34(3):437–446, Apr. 2002.

[29] C. Cirelli, C. M. Gutierrez, and G. Tononi. Extensive and divergent effects of sleep and wakefulness on brain gene expression. *Neuron*, 41(1):35 – 43, 2004.

[30] C. Cirelli and G. Tononi. Differential Expression of Plasticity-Related Genes in Waking and Sleep and Their Regulation by the Noradrenergic System. *The Journal of Neuroscience*, 20(24):9187–9194, December 2000.

[31] C. Cortes and V. Vapnik. Support-vector networks. In *Machine Learning*, pages 273–297, 1995.

[32] V. Cutsuridis, S. Cobb, and B. P. Graham. How bursts shape the stdp curve in the presence/absence of gabaergic inhibition. In *Proceedings of the 19th International Conference on Artificial Neural Networks: Part I*, ICANN '09, pages 229–238, Berlin, Heidelberg, 2009. Springer-Verlag.

[33] Y. Dan and M.-M. Poo. Spike timing-dependent plasticity of neural circuits. *Neuron*, 44(1):23–30, Sep 2004.

[34] R. Desimone and J. Duncan. Neural Mechanisms of Selective Visual Attention. *Annual Review of Neuroscience*, 18(1):193–222, 1995.

[35] G. Edelman. Naturalizing consciousness: A theoretical framework. *Proceedings of the National Academy of Sciences*, 100:5520–5524, 2003.

[36] G. Edelman and G. Tononi. A universe of consciousness. *New York: Basic Books*, 2000.

[37] H. Esmaeilzadeh, E. Blem, R. S. Amant, K. Sankaralingam, and D. Burger. Dark Silicon and the End of Multicore Scaling. In *Proceedings of the 38th International Symposium on Computer Architecture*, June 2011.

[38] H. Esmaeilzadeh, A. Sampson, L. Ceze, and D. Burger. Neural acceleration for general-purpose approximate programs. In *Proceedings of the 2012 45th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO '12, pages 449–460, Washington, DC, USA, 2012. IEEE Computer Society.

[39] R. Fagundes, F. de Castro, A. Martins, and M. de Castro. Hebbian learning in an automatic gender identification by speech system. In *Neural Information Processing*,

2002. ICONIP '02. Proceedings of the 9th International Conference on, volume 5, pages 2409–2413 vol.5, 2002.

[40] C. Fernando and S. Sojakka. Pattern recognition in a bucket. 2801:588–597, 2003.

[41] E. Fortune and G. Rose. Short-term synaptic plasticity as a temporal filter. *Trends Neurosci*, 24(7):381–385, July 2001.

[42] K. A. Foster, A. C. Kreitzer, and W. G. Regehr. Interaction of postsynaptic receptor saturation with presynaptic mechanisms produces a reliable synapse. *Neuron*, 36(6):1115 – 1126, 2002.

[43] W. Freeman. Strange attractors that govern mammalian brain dynamics shown by trajectories of electroencephalographic (eeg) potential. *Circuits and Systems, IEEE Transactions on*, 35(7):781 –783, jul 1988.

[44] K. J. Friston. Transients, metastability, and neuronal dynamics. *NeuroImage*, 5(2):164 – 171, 1997.

[45] S. B. Furber, D. R. Lester, L. A. Plana, J. D. Garside, E. Painkras, S. Temple, and A. D. Brown. Overview of the spinnaker system architecture. *IEEE Transactions on Computers*, 99(PrePrints), 2012.

[46] S. Fusi, W. F. Asaad, E. K. Miller, and X.-J. Wang. A Neural Circuit Model of Flexible Sensorimotor Mapping: Learning and Forgetting on Multiple Timescales. *Neuron*, 54(2):319–333, Apr. 2007.

[47] L. Garey and C. de Courten. Structural development of the lateral geniculate nucleus and visual cortex in monkey and man. *Behavioural Brain Research*, 10(1):3 – 13, 1983.

[48] G. Gigante, M. Mattia, J. Braun, and P. Del Giudice. Bistable perception modeled as competing stochastic integrations at two levels. *PLoS Comput Biol*, 5(7):e1000430, 2009.

[49] G. F. Gilestro, G. Tononi, and C. Cirelli. Widespread changes in synaptic markers as a function of sleep and wakefulness in drosophila. *Science*, 324(5923):109–112, Apr 2009.

[50] C. Gros. Self-sustained thought processes in a dense associative network. 2005.

[51] C. Gros. Neural networks with transient state dynamics. *New Journal of Physics*, 2007.

[52] C. Gros. Complex and adaptive dynamical systems, a primer. 2008.

[53] C. Gros. Cognitive computation with autonomously active neuralnetworks: an emerging field. *Cognitive Computation*, 1:77–90, 2009.

[54] S. Haeusler and W. Maass. A statistical analysis of information-processing properties of lamina-specific cortical microcircuit models. *Cereb. Cortex*, 17(1):149–162, Jan 2007.

[55] C. Harley. Noradrenergic and locus coeruleus modulation of the perforant path-evoked potential in rat dentate gyrus supports a role for the locus coeruleus in attentional and memorial processes. *Progress in Brain Research*, 88:307–321, 1991.

[56] A. Hashmi, H. Berry, O. Temam, and M. H. Lipasti. Automatic abstraction and fault tolerance in cortical microachitectures. In *ISCA'11*, pages 1–10, 2011.

[57] A. Hashmi and M. Lipasti. Discovering cortical algorithms. In *Proceedings of the 14th International Conference on Cognitive and Neural Systems (ICCNS-14)*, 2010.

[58] A. Hashmi and M. Lipasti. Discovering cortical algorithms. In *Proceedings of the International Conference on Neural Computation (ICNC 2010)*, 2010.

[59] A. Hashmi, A. Nere, J. J. Thomas, and M. Lipasti. A case for neuromorphic isas. In *Proceedings of the sixteenth international conference on Architectural support for programming languages and operating systems*, ASPLOS '11, pages 145–158, New York, NY, USA, 2011. ACM.

[60] J. Hawkins and S. Blakeslee. *On Intelligence*. Henry Holt & Company, Inc., 2005.

[61] H. Hazan and L. Manevit. The liquid state machine is not robust to problems in its components but topological constraints can restore robustness. In *Proc. of the Int. Conf. on Fuzzy Computation and Int. Conf. on Neural Computation*, page 258=264, 2010.

[62] D. O. Hebb. *The Organization of Behavior: A Neuropsychological Theory*. Wiley, new edition edition, June 1949.

[63] S. Hill, G. Tononi, and M. Ghilardi. Sleep improves the variability of motor performance. *Brain Res Bull.*, 76(6):605–611, Aug 2008.

[64] J. Hirsch and L. Martinez. Laminar processing in the visual cortical column. *Current Opinion in Neurobiology*, 16:377–384, 2006.

[65] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, Apr. 1982.

[66] J. M. Hupe, A. C. James, B. R. Payne, S. G. Lomber, P. Girard, and J. Bullier. Cortical feedback improves discrimination between figure and background by V1, V2 and V3 neurons. *Nature*, 394(6695):784–787, Aug. 1998.

[67] K. Hynna and K. Boahen. Neuronal ion-channel dynamics in silicon. In *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*, pages 4 pp.–, 2006.

[68] N. Imam, F. Akopyan, J. Arthur, P. Merolla, R. Manohar, and D. S. Modha. A digital neurosynaptic core using event-driven qdi circuits. In *Proceedings of the 2012 18th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, ASYNC '12, pages 25–32, Washington, DC, USA, 2012. IEEE Computer Society.

[69] E. Izhikevich. Which model to use for cortical spiking neurons? *Neural Networks, IEEE Transactions on*, 15(5):1063 –1070, sep. 2004.

[70] E. M. Izhikevich. Solving the Distal Reward Problem through Linkage of STDP and Dopamine Signaling. *Cerebral Cortex*, 17(10):2443–2452, Oct. 2007.

[71] E. M. Izhikevich and G. M. Edelman. Large-scale model of mammalian thalamocortical systems. *Proceedings of the National Academy of Sciences*, 105(9):3593–3598, Mar. 2008.

[72] W. A. Kaminski and G. M. Wojcik. Liquid state machines built of hodgkin-huxley. In *Neurons âŁ“ Pattern Recognition and Informational Entropy, Annales UMCS Informatica, Vol. I, Lublin*, 2004.

[73] E. Kandel, J. Schwartz, and T. Jessell. *Principles of Neural Science*. McGraw-Hill, 4 edition, 2000.

[74] K. Kavukcuoglu, M. Ranzato, R. Fergus, and Y. Le-Cun. Learning invariant features through topographic filter maps. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1605 –1612, june 2009.

[75] I. C. Kleppe and H. P. Robinson. Determining the activation time course of synaptic ampa receptors from openings of colocalized nmda receptors. *Biophysical Journal*, 77(3):1418 – 1427, 1999.

[76] C. Koch and N. Tsuchiya. Attention and consciousness: two distinct brain processes. *Trends in Cognitive Sciences*, 11(1):16–22, Jan. 2007.

[77] P. Lansky, P. Sanda, and J. He. The parameters of the stochastic leaky integrate-and-fire neuronal model. *Journal of computational neuroscience*, 21(2):211–223, 2006.

[78] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[79] J. Lee, P. Ajgaonkar, and N. S. Kim. Analyzing throughput of gpgpus exploiting within-die core-to-core frequency variation. In *Performance Analysis of Systems and Software (ISPASS), 2011 IEEE International Symposium on*, pages 237 –246, april 2011.

[80] J. J. Letzkus, B. M. Kampa, and G. J. Stuart. Learning rules for spike timing-dependent plasticity depend on dendritic synapse location. *J Neurosci*, 26(41):10420–10429, Oct. 2006.

[81] Z.-W. Liu, U. Faraguna, C. Cirelli, G. Tononi, and X.-B. Gao. Direct evidence for wake-related increases and sleep-related decreases in synaptic strength in rodent cortex. *J Neurosci*, 30(25):8671–8675, Jun 2010.

[82] M. Lundqvist, A. Compte, and A. Lansner. Bistable, irregular firing and population oscillations in a modular attractor memory network. *PLoS Comput Biol*, 6(6):e1000803, 06 2010.

[83] M. Lundqvist, M. Rehn, M. Djurfeldt, and A. Lansner. Attractor dynamics in a modular network model of neocortex. network. In *Network: Computation in Neural Systems*, pages 17–253, 2006.

[84] M. Lynch and S. N. Griffith. An application of temporal difference learning to draughts, 1997.

[85] W. Maass and T. Natschläger. Networks of spiking neurons can emulate arbitrary Hopfield nets in temporal coding. *Network: Computation in Neural Systems*, 8(4):355–372, 1997.

[86] W. Maass, T. Natschläger, and H. Markram. Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Comput*, 14(11):2531–2560, Nov. 2002.

[87] H. Markram, J. LÃ¼bke, M. Frotscher, and B. Sakmann. Regulation of synaptic efficacy by coincidence of postsynaptic aps and epsps. *Science*, 275(5297):213–215, 1997.

[88] H. Markram, Y. Wang, and M. Tsodyks. Differential signaling via the same axon of neocortical pyramidal neurons. *Proceedings of the National Academy of Sciences of the United States of America*, 95(9):5323–5328, Apr. 1998.

[89] K. A. C. Martin. A Brief History of the" Feature Detector". *Cerebral Cortex*, 4(1):1, 1994.

[90] T. Masquelier and S. J. Thorpe. Unsupervised learning of visual features through spike timing dependent plasticity. *PLoS Comput Biol*, 3(2):e31, 02 2007.

[91] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology*, 5(4):115–133, Dec. 1943.

[92] R. McEliece, E. Posner, E. Rodemich, and S. Venkatesh. The capacity of the hopfield associative memory. *Information Theory, IEEE Transactions on*, 33(4):461 – 482, jul 1987.

[93] C. Mead. *Analog VLSI and Neural Systems*. Addison Wesley Publishing Company, 1st edition, Jan. 1989.

[94] P. Merolla, J. Arthur, F. Akopyan, N. Imam, R. Manohar, and D. Modha. A digital neurosynaptic core using embedded crossbar memory with 45pj per spike in 45nm. In *Proceeding of the IEEE Custom Integrated Circuits Conference*, 2011.

[95] C. Metzner, M. Menzinger, A. Schweikard, and B. Zurowski. Early signs of tinnitus in a simulation of the mammalian primary auditory cortex. In *BMC Neuroscience*, volume 12 (Suppl. 1), page P383, 2011.

[96] M. L. Minsky and S. A. Papert. *Perceptrons: Expanded edition*. The MIT Press, Cambridge, MA, 1988.

[97] J. Misra and I. Saha. Artificial neural networks in hardware: A survey of two decades of progress. *Neurocomput.*, 74:239–255, December 2010.

[98] V. Mountcastle. An organizing principle for cerebral function: The unit model and the distributed system. In G. Edelman and V. Mountcastle, editors, *The Mindful Brain*. MIT Press, Cambridge, Mass., 1978.

[99] V. Mountcastle. The columnar organization of the neocortex. *Brain*, 120:701–722, 1997.

[100] J. Navaridas, M. Luján, L. A. Plana, J. Miguel-Alonso, and S. B. Furber. Analytical assessment of the suitability of multicast communications for the spinnaker neuromimetic system. In *Proceedings of the 2012 IEEE 14th International Conference on High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems*, HPCC '12, pages 1–8, Washington, DC, USA, 2012. IEEE Computer Society.

[101] A. Nere, A. Hashmi, and M. Lipasti. Bridging the semantic gap: Emulating biological neuronal behaviors with simple digital neurons. In *Proceedings of the nineteenth international symposium on high-performance computer architecture*, HPCA '13, 2013.

[102] A. Nere, U. Olcese, D. Balduzzi, and G. Tononi. A neuromorphic architecture for object recognition and motion anticipation using burst-stdp. *PLoS ONE*, 7(5):17 pages, 5 2012.

[103] U. Olcese, S. Esser, and G. Tononi. Sleep and synaptic renormalization: A computational study. *J Neurophysiol*, submitted.

[104] J. A. Perrone. A visual motion sensor based on the properties of {V1} and {MT} neurons. *Vision Research*, 44(15):1733 – 1755, 2004.

[105] J.-P. Pfister and W. Gerstner. Triplets of Spikes in a Model of Spike Timing-Dependent Plasticity. *The Journal of Neuroscience*, 26(38):9673–9682, Sept. 2006.

[106] R. Preissl, T. M. Wong, P. Datta, M. Flickner, R. Singh, S. K. Esser, W. P. Risk, H. D. Simon, and D. S. Modha. Compass: A scalable simulator for an architecture for cognitive computing, 2012.

[107] Y. Rao, Z.-W. Liu, E. Borok, R. L. Rabenstein, M. Shanabrough, M. Lu, M. R. Picciotto, T. L. Horvath, and X.-B. Gao. Prolonged wakefulness induces experience-dependent synaptic plasticity in mouse hypocretin/orexin neurons. *J Clin Invest*, 117(12):4022–4033, Dec 2007.

[108] A. Rast, F. Galluppi, X. Jin, and S. Furber. The leaky integrate-and-fire neuron: A platform for synaptic model exploration on the spinnaker chip. In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pages 1–8, 2010.

[109] W. Regehr. Short-term presynaptic plasticity. *Cold Spring Harb. Perspect. Biol.*, 4, 2012.

[110] A. Renart, N. Brunel, and X.-J. Wang. Mean field theory of irregularly spiking neuronal populations and working memory in recurrent cortical networks,. *Computational Neuroscience: A Comprehensive Approach*, pages 431 – 490, 2003.

[111] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2:1019–1025, 1999.

[112] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nat Neurosci*, 2(11):1019–1025, Nov 1999.

[113] E. T. Rolls, N. C. Aggelopoulos, and F. Zheng. The Receptive Fields of Inferior Temporal Cortex Neurons in Natural Scenes. *J. Neurosci.*, 23(1):339–348, Jan. 2003.

[114] G. Roth and U. Dicke. Evolution of brain and intelligence. *TRENDS in Cognitive Sciences*, 5:250–257, 2005.

[115] J. L. R. Rubenstein and M. M. Merzenich. Model of autism: increased ratio of excitation/inhibition in key neural systems. *Genes Brain Behav*, 2(5):255–267, Oct. 2003.

[116] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Parallel distributed processing: explorations in the microstructure of cognition, vol. 1. chapter Learning internal representations by error propagation, pages 318–362. MIT Press, Cambridge, MA, USA, 1986.

[117] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2003.

[118] J. Schemmel, J. Fieres, and K. Meier. Wafer-scale integration of analog neural networks. In *Proceedings of the International Joint Conference on Neural Networks, IJCNN 2008, part of the IEEE World Congress on Computational Intelligence, WCCI 2008, Hong Kong, China, June 1-6, 2008*, pages 431–438. IEEE, 2008.

[119] J. Schemmel, A. Grubl, K. Meier, and E. Mueller. Implementing synaptic plasticity in a vlsi spiking neural network model. In *Neural Networks, 2006. IJCNN '06. International Joint Conference on*, pages 1–6, 2006.

[120] B. Schrauwen, M. D'Haene, D. Verstraeten, and J. Van Campenhout. Compact hardware for real-time speech recognition using a liquid state machine. In *Neural Networks, 2007. IJCNN 2007. International Joint Conference on*, pages 1097 –1102, aug. 2007.

[121] W. Schultz. Predictive reward signal of dopamine neurons. *Journal of Neurophysiology*, 80:1–27, 1998.

[122] M. W. Self, R. N. Kooijmans, H. Supèr, V. A. Lamme, and P. R. Roelfsema. Different glutamate receptors convey feedforward and recurrent processing in macaque V1. *Proceedings of the National Academy of Sciences of the United States of America*, 109(5):11031–6, 2012.

[123] W. Senn and S. Fusi. Convergence of stochastic learning in perceptrons with binary synapses. *Phys. Rev. E*, 71:061907, Jun 2005.

[124] W. Senn and S. Fusi. Learning only when necessary: Better memories of correlated patterns in networks with bounded synapses. *Neural Comput.*, 17(10):2106–2138, 2005.

[125] J.-s. Seo, B. Brezzo, Y. Liu, B. D. Parker, S. K. Esser, R. K. Montoye, B. Rajendran, J. A. Tierno, L. Chang, D. S. Modha, and D. J. Friedman. A 45nm cmos neuromorphic chip with a scalable architecture for learning in networks of spiking neurons. In *Custom Integrated Circuits Conference (CICC), 2011 IEEE*, pages 1 –4, sept. 2011.

[126] G. Seol, J. Ziburkus, S. Huang, et al. Neuromodulators control the polarity of spike-timing-dependent synaptic plasticity. *Neuron*, 55(6):919–929, 2007.

[127] T. Serre, A. Oliva, and T. Poggio. A feedforward architecture accounts for rapid categorization. *Proc. Natl. Acad. Sci. USA*, 104(15):6424–6429, Apr 2007.

[128] M. Simoni, G. Cymbalyuk, M. Sorensen, R. Calabrese, and S. DeWeerth. A multiconductance silicon neuron with biologically matched dynamics. *Biomedical Engineering, IEEE Transactions on*, 51(2):342–354, 2004.

[129] O. Sporns and W. H. Alexander. Neuromodulation and plasticity in an autonomous robot. *Neural Netw*, 15(4-6):761–774, 2002.

[130] D. Standage, S. Jalil, and T. Trappenberg. Computational consequences of experimentally derived spike-time and weight dependent plasticity rules. *Biol Cybern*, 96(6):615–623, Jun 2007.

[131] R. S. Sutton. Learning to predict by the methods of temporal differences. In *MACHINE LEARNING*, pages 9–44. Kluwer Academic Publishers, 1988.

[132] L. Swanson. Mapping the human brain: past, present, and future. *Trends in Neurosciences*, 18(11):471 –474, 1995.

[133] G. Tesauro. Temporal difference learning and td-gammon. *Commun. ACM*, 38:58–68, March 1995.

[134] A. Thomson, J. Deuchars, and D. West. Large, deep layer pyramid-pyramid single axon epsps in slices of rat motor cortex display paired pulse andfrequency-dependent depression, mediated presynaptically and self-facilitation, mediated postsynaptically. *J. Neurophysiol.*, 70:2354–69.

[135] A. M. Thomson. Activity-dependent properties of synaptic transmission at two classes of connections made by rat neocortical pyramidal axons in vitro. *J Physiol*, 502 (Pt 1):131–147, July 1997.

[136] G. Tononi and C. Cirelli. Sleep and synaptic homeostasis: a hypothesis. *Brain Res Bull*, 62(2):143–150, Dec 2003.

[137] G. Tononi and C. Cirelli. Sleep function and synaptic homeostasis. *Sleep Med Rev*, 10(1):49–62, Feb 2006.

[138] G. Tononi, O. Sporns, and G. M. Edelman. A measure for brain complexity: relating functional segregation and integration in the nervous system. *Proceedings of the National Academy of Sciences of the United States of America*, 91(11):5033–5037, May 1994.

[139] D. Tropea, E. Sermasi, and L. Domenici. Synaptic plasticity of feedback connections in rat visual cortex. *Developmental Brain Research*, 118(1âŁ“2):61 – 67, 1999.

[140] L. O. Trussell, S. Zhang, and I. M. Ramant. Desensitization of ampa receptors upon multiquantal neurotransmitter release. *Neuron*, 10(6):1185 – 1196, 1993.

[141] D. Verstraten, B. Schrauwen, D. Stroobandt, and J. Van Campenhout. Isolated word recognition with the liquid state machine: a case study. *Inf. Process. Lett.*, 95:521–528, September 2005.

[142] V. V. Vyazovskiy, C. Cirelli, M. Pfister-Genskow, U. Faraguna, and G. Tononi. Molecular and electrophysiological evidence for net synaptic potentiation in wake and depression in sleep. *Nature Neuroscience*, 11(2):200–208, January 2008.

[143] V. V. Vyazovskiy, C. Cirelli, M. Pfister-Genskow, U. Faraguna, and G. Tononi. Molecular and electrophysiological evidence for net synaptic potentiation in wake and depression in sleep. *Nat Neurosci*, 11(2):200–208, Feb 2008.

[144] V. V. Vyazovskiy, U. Olcese, Y. M. Lazimy, U. Faraguna, S. K. Esser, J. C. Williams, C. Cirelli, and G. Tononi. Cortical firing and sleep homeostasis. *Neuron*, 63(6):865–878, Sep 2009.

[145] J. I. Wadiche and C. E. Jahr. Multivesicular release at climbing fiber-purkinje cell synapses. *Neuron*, 32(2):301 – 313, 2001.

[146] Y. Wang, T. Wu, G. Orchard, P. Dudek, M. Rucci, and B. Shi. Hebbian learning of visually directed reaching by a robot arm. In *Biomedical Circuits and Systems Conference, 2009. BioCAS 2009. IEEE*, pages 205–208, 2009.

[147] T. Wong, R. Preissl, P. Datta, M. Flickner, R. Singh, S. Esser, E. McQuinn, R. Appuswamy, W. Risk, H. Simon, and D. Modha. 1014. *IBM Research Report*, November 2012.

[148] W. Yang, L. Zhang, , and L. Ma. Computational model for rotation-invariant perception. In *International Conference on Natural Computation*, volume 2, pages 144–148, 2007.

[149] S. Zeki. The motion pathways of the visual cortex. *Vision: Coding and Efficiency*, pages 321–345, 1990.

[150] L. Zhang and E. Jones. Corticothalamic inhibition in the thalamic reticular nucleus. *J. Neurophysiol.*, 91:759–766, 2004.

[151] L. I. Zhang, H. W. Tao, C. E. Holt, W. A. Harris, and Poo. A critical window for cooperation and competition among developing retinotectal synapses. *Nature*, 395:37–44, 1998.

[152] Y.-J. Zheng and B. Bhanu. Adaptive object detection based on modified hebbian learning. In *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, volume 4, pages 164–168 vol.4, 1996.