EMERGENT ASSEMBLY-BASED COMPUTATION IN A MODEL OF THE BRAIN

A Dissertation
Presented to
The Academic Faculty

By

Max Dabagia

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in
Algorithms, Combinatorics, & Optimization

Georgia Institute of Technology

April 2025

EMERGENT ASSEMBLY-BASED COMPUTATION IN A MODEL OF THE BRAIN

Thesis committee:

Dr. Santosh Vempala
School of Computer Science
Georgia Institute of Technology

Dr. Christos Papadimitriou
Department of Computer Science
Columbia University

Dr. Jack Gallant
Department of Psychology
University of California, Berkeley

Dr. Will Perkins
School of Computer Science
Georgia Institute of Technology

Dr. Eva Dyer
School of Biomedical Engineering
Georgia Institute of Technology

Date approved: April 20, 2025

Marco Polo describes a bridge, stone by stone.

"But which is the stone that supports the bridge?" Kublai Khan asks.

"The bridge is not supported by one stone or another," Marco answers, "but by the line of the arch that they form."

Kublai Kahn remains silent, reflecting. Then he adds: "Why do you speak to me of the stones? It is only the arch that matters to me."

Polo answers: "Without stones there is no arch."

Italo Calvino, trans. by William Weaver

TABLE OF CONTENTS

LIST OF ACRONYMS

ANN artificial neural network

w.h.p. with high probability, meaning probability tending to 1 as some parameter tends to infinity

WTA winner(s)-take-all

SUMMARY

The brain is a phenomenally complex physical system. Understanding how it gives rise to cognition remains a fundamental challenge for science, as well as a wellspring of inspiration of artificial intelligence. Despite enormous progress since the genesis of modern neuroscience more than a century ago in characterizing individual neurons, and in correlating macroscopic regions of the brain with various aspects of information processing, the intermediate steps – wherein billions of neurons somehow organize themselves to implementation the computations comprising intelligent behavior – remain essentially mysterious. Thus, the essence of understanding the brain is unraveling the mechanisms by which this self-organization occurs. The starting point of this thesis is a model of the brain, called NEMO, which arises from a few basic ingredients: discretely-firing point neurons, connected with each other randomly and controlled by local inhibition, with Hebbian plasticity to learn. These ingredients are abstractions of biological mechanisms well-known to experimental neuroscience, comprising in some sense a "minimal" model of the brain. Under its dynamics assemblies emerge in response to stimuli, sets of simultaneously-firing neurons which are hypothesized to be the basic unit of representation in the brain. We explore a few simple algorithms which are implementable under its dynamics, including learning linear classifiers, sequence memorization, automata memorization and simulation, and estimation and sampling from simple statistical models, both in theory and simulation. Taken together, these results comprise the foundation of a novel theory of how the brain could compute, and add theoretical support to the idea that assemblies are fundamental to representation in the brain.

CHAPTER 1
INTRODUCTION AND BACKGROUND

How can it be that human cognition, in all its complexity, creativity, and diversity, can be reduced to mere electrical activity of the neurons which comprise the brain? This question, a refinement of one which has tormented philosophers since the dawn of rational inquiry, remains an outstanding one for science.

Of course, this is not for lack of trying. Over the last century, cellular neuroscientists have done a great deal to characterize individual neurons, from precise physical models of their electrochemical properties to how they respond to experiences of the organism as a whole. This we might think of as the "low" level. At the "high" level, entire disciplines are devoted to describing various aspects of *cognition*, while cognitive neuroscientists have mapped out the correspondence between various cognitive systems and large-scale brain structures. However, this leaves an obvious explanatory gap: How does the brain's enormous pool of neurons somehow coordinate itself to perform these incredible feats of intellect? By way of analogy with the electronic computer, it is as though we understand the transistor quite well, and we've gotten a good handle on how one can open a spreadsheet and play video games, but the really interesting part from the perspective of understanding *how this machine works*, where the voltage level of a bunch of transistors somehow gives rise to all the things a computer can do, remains completely unknown to us. The brain's program, language, algorithm, which translates patterns of neural activity into thoughts and actions, is a deep mystery.

This thesis presents the first, tentative steps of an attempt towards such a program. It takes the view that a mature theory of the brain will be essentially computational: It will describe the brain's physiology as a mathematically precise model of computation, and cognition as a collection of algorithms, implemented within this model yet encompassing

every function of the brain.

1.1 The Computational Brain

Whether the human brain is a computer is an immensely contentious issue, although since much of the debate seems to be terminological [1], it is worth clarifying how exactly the equivalence should be understood in this thesis. To a computer scientist, a "computer" simply means a system that realizes a computable function; that is, any function which is the output of a Turing machine, or a logically equivalent mathematical construct [2]. This turns out to be a rather expansive category; since all of known physics is apparently completely described by mathematical equations, and these equations can be computed to arbitrary accuracy by a Turing machine, *any known physical system* seems to realize a computable function. Indeed, this is precisely the content of the strong Church-Turing thesis, which states that any physically computable function is computable by a Turing machine [2]. So, assuming that the brain doesn't rely on any exotic physics, the statement that the brain is a computer (to a computer scientist) is merely an assertion of *physicalism*: That there is nothing more to the brain than its physical embodiment, i.e. the rejection of dualism.

On the other hand, the mind/brain is different from other emergent phenomena (e.g. the weather, which "runs" on the atmosphere) in that it is all but explicitly *designed by evolution* to collect data from its environment and make decisions based on those data; in other words, to *process information*. This implies that what the brain is doing is something which is likely to be well-described as computation in the lay sense. The celebrated neuroscientist David Marr articulated an immensely influential idea about the "levels of analysis" one must consider in a information processing (i.e. computational) system [3]. (Marr was motivated by his work in the visual system, but the point is a general one.) At the top of the hierarchy, one has the *computational* level: What problems does the system solve, and why? The intermediate level is *algorithmic*: What representations are used, and

what processes applied to them, to solve the problem? Finally, the lowest level is *physical*: How is the system realized in the substrate of the world? Hence, the claim that the brain is essentially computational should be understood to mean that (i) there are problems that brain solves; (ii) it does so via applying processes to representations of information; and (iii) it is a physical system. Of course, what defines a "problem" and what it means to "solve" one are also vague concepts, and difficult to define precisely such that the things brains and electronic computers do qualify and the things that the weather and waterfalls do don't. Hence, let us leave aside the issue of definitions here, with the understanding that a computational theory of the brain will only be a meaningful perspective if it offers a rich analysis of how the brain works at all three of Marr's levels.

1.2 The Role of Theory in Neuroscience

It is a common view among neuroscientists that we have entered a "Moore's law" for neural recording [4, 5]; that is, the number of neurons which can be simultaneously recorded may well increase geometrically for the foreseeable future. Thanks to deep learning and an abundance of cheap computation, we also have a suite of sophisticated data analysis methods to apply to these datasets [.] What is the role of theory in this intellectual ecosystem? One dark cloud in this apparently sunny forecast is that the analysis methods must necessarily work, that is, those which find patterns in neural activity had better recover the patterns significant to the underlying computations, and those which decode neural activity had better be interpretable. There is some evidence that this isn't the case; for one, deep neural networks are notoriously difficult to make sense of [6], so it is unclear how helpful even an highly predictive deep network model of neural activity will be in understanding the brain.

Of course, one way to resolve this issue is simply to run these analysis methods on a system where the ground-truth is known – a computational system with a complete characterization at all three of Marr's levels (see Section 1.1) – to test whether the methods

manage to elucidate the underlying information processing. Jones & Kording did precisely this with a microprocessor as a model organism, treating individual transistors as "neurons" and applying the standard experimental treatments and analysis methods of neuroscience to it [7]. They note that

> We find that many measures are surprisingly similar between the brain and the processor but that our results do not lead to meaningful understanding of the processor. The analysis cannot produce the hierarchical understanding of information processing that most students of electrical engineering obtain. It suggests that the availability of unlimited data, as we have for the processor, is in no way sufficient to allow a real understanding of the brain.

The challenge for theorists is clear: Making sense of the wealth of data may require clear hypotheses about what to look for. Of course, the space of possible hypotheses is enormous. Hence, there is an evident need for new strategies for exploring, selecting, and testing theories of brain computation.

More abstractly, it may be surprising to find that a body of work ostensibly about developing a *hypothesis of brain function* is mostly devoted to the design and analysis of certain algorithms. There is a simple rejoinder: A successful theory of the brain will ultimately take shape through the scientific method – proposing hypotheses, making predictions, performing experiments – but notably it is not specified *how* to select a hypothesis. This is especially relevant in neuroscience as the space of possible hypotheses of brain function is enormous and severely underconstrained by empirical data. A good model of the brain (in light of the available data) is one which conforms to what is known about neurophysiology, while also excelling at the various cognitive tasks that we know the brain to perform. Hence, algorithm design has a role in describing how a particular model of neuroanatomy can perform a cognitive task, while the methods of algorithm analysis are a natural perspective to take in quantifying how well the model performs those tasks.

However, in our present state of knowledge, there is a disturbing possibility: The com-

putations performed by the brain could be sufficiently difficult to abstract that they are essentially incomprehensible to human intellect [8]. In other words, although there is an "algorithmic" level of understanding brain computation, the algorithms could be so complex that describing them does not allow us to make sense of the system. This is not totally infeasible; although the traditional notion of an "algorithm" is designed by humans and thus comprehensible to them, an important new class of algorithms – those described by a deep neural network – now approaches or exceeds human performance at tasks commonly thought to require intelligence, and the most effective algorithms in this class are *not designed* by humans, but rather found by a search procedure (the training process). In spite of an enormous research effort these networks remain difficult to interpret or reverse-engineer, even though (unlike the brain) there is no technical barrier to the experiments that can be performed on an artificial neural network. Nonetheless, the process for training artificial neural networks – the dataset, its preprocessing, the architecture of the network, the training objective, and the optimization algorithm – are all fairly easy to describe satisfactorily, as well as understood intuitively (if not rigorously) to some extent. So, even if it turns out to be true that brain's algorithms are beyond human comprehension, there is still hope that the *meta-algorithms* by which these algorithms are developed can be fully understood, and a clear role for theory in uncovering and describing them.

1.3 On Neurobiology

The following discussion is intended to describe a small portion of what is known about the brain, specifically its primary computational substrate – the neuron. Given these facts we will indicate some basic assumptions about how they can be abstracted into a tractable model, as a starting point in this thesis. Some of these ideas are very well-established, while others are more controversial among neuroscientists. In the electronic computer analogy, what we want is a description of the brain at the level of logic gates, rather than at the level of individual electrons, or transistors, or the operating system. Accordingly,

these abstractions certainly lose some of the detail of neuroanatomy. Nonetheless, they are reasonable starting points for grounded thinking about the brain, in that they are consistent with what is known and seem not to postulate any additional mechanisms beyond that.

This a difficult issue for theoretical neuroscience: A model that is too abstract will cease to have anything useful to say about the brain. On the other hand, in the absence of a complete understanding of the brain, a model which is too detailed will not be able to produce testable predictions either. Abstraction is also useful in that it makes the model easier to understand, to analyze, and to think about – and some degree of abstraction is necessary in any model, lest it simply reproduce, in perfect detail, the phenomenon it is meant to capture. It is worth recalling the celebrated short story of Jorge Luis Borges, "On Exactitude in Science", in which the quest for precision leads an overzealous cartographers guild to construct a map of the same scale as the thing it is meant to describe [9]. As long as assumptions are made judiciously, simplifying models that result from them can be productive explanatory tools.

Neurons. Neurons are the substrate of computation in the brain; the rest of the brain's anatomy exists to support the neurons. Neurons are either excitatory or inhibitory; excitatory neurons encode information, while inhibitory neurons serve to coordinate and regulate the firing of excitatory neurons.

Spiking. Gradations in the output of a neuron is unimportant. Its output is binary; it either emits an action potential or it doesn't. Furthermore, spiking can be additionally discretized in time; the entirety of a neuron's output is described by whether a neuron emits a spike within a small enough window (say, 10ms).

Synapses. Neurons communicate with each other exclusively through synapses. Each synapse between excitatory neurons has its own positive "weight", which is subject to plasticity. Neurons form a single synapse with each other. The internal anatomy of a

neuron is irrelevant; action potentials at all synapses contribute equally (after their weights are taken into account) to driving the neuron's output.

Connectivity. Connectivity among all neurons within a sufficiently small area is effectively random. Longer distance connections are combined into fibers, which means that entire local areas can be viewed as connected to each other.

Inhibition. The effect of inhibitory neurons is a "blanket of inhibition" cast over a local neighborhood of excitatory neurons. Local inhibitory circuits are tightly coupled, effectively acting together as a unit to enforce a dynamic on the excitatory neurons they regulate. More specifically, the effect of local inhibition is to enforce a set level of activity among a population of excitatory neurons, by quickly increasing the inhibitory signal if many neurons are beginning to fire and quickly decreasing it if very few are.

Plasticity. The connectivity graph of neurons is fixed over moderate timescales. (Say, minutes to hours.) Weights are always subject to long-term, fast-acting Hebbian plasticity: Whenever a presynaptic neuron fires shortly before the postsynaptic neuron (i.e. shortly enough that it participates in causing the postsynaptic neuron to fire) the synaptic weight between them is immediately subject to a small, persistent increase in its weight. In addition, among some populations of excitatory neurons, homeostasis is periodically applied on a longer timescale, which effectively reweights a neuron's incoming synapses to maintain a fixed sum.

1.4 The Assembly of Neurons

1.4.1 Representation in the Brain

Two perspectives on the basic encoding strategy of the brain have prevailed in modern neuroscience [10, 11]. The *neuron doctrine* proposes that the neuron is the basic structural and functional unit of the nervous system [12]; its origins in the work of Santiago Ramón y

Cajal [13] and Charles Sherrington [14] mark the birth of modern neuroscience. Although it is universally accepted that the substrate of the brain is made up of neurons, whether the neuron is also the right level to think about function is more contentious. The usual way in which this proposal is made more precise is that each neuron fires precisely when it detects a specific feature in a stimulus (ranging in abstraction from a particular pattern on the retina to the idea of one's grandmother). Early experimental work by Horace Barlow [15, 16] show the existence of apparent fly-detecting cells in the frog retina, but this view reached its zenith with the work of Hubel & Wiesel [17], who systematized the exploration of the mammalian visual cortex and discovered neurons receptive specifically to both the position and form of stimulus in the visual field. The dominance of the neuron doctrine, especially in work on the neural mechanisms of perception, is easily understood in light of the facts that multi-unit recording at scale only became feasible within the last few decades, and that the dominant experimental paradigms emphasized very simple stimuli rather than natural scenes. It is now understood that neurons in higher cortical areas – especially the prefrontal cortex [18, 19] – have a much higher breadth and flexibility of tuning which makes the interpretation of their function as feature detection far less clear. We also now understand that neurons die quite frequently [20]; the fact that we maintain concepts for our entire lives means their representations must be spread out over multiple neurons.

In contrast we have the *assembly hypothesis* [21, 10], first articulated precisely by Donald Hebb in his landmark book [22]. Hebb proposed that the basic unit of neural representation is a set of neurons, encoded in their sustained and synchronized firing, which collectively represent the gestalt of a concept or event. This basic idea was anticipated even earlier by Rafael Lorente de Nó, who observed that the recurrent connectivity observed across the brain could generate "functional reverberations", or self-sustaining patterns of activity, after external stimulation has dissipated [23], and indeed even Sherrington [24]:

> The brain region which we may call 'mental' is not a concentration into one cell, but an enormous expansion into millions of cells. They are it is true richly

> interconnected. Where it is a question of 'mind' the nervous system does not integrate itself by centralization upon one pontifical cell. Rather it elaborates a million-fold democracy whose each unit is a cell.

Assemblies are known by various names – ensembles, engrams, among others – but the key point is that the collective activity of a group of neurons generates a functional state that is not ascertainable by measuring individual neurons, and moreover this functional state is characterized by some degree of synchronization and persistence. Notably, assemblies are not necessarily mutually exclusive with feature detecting neurons; an assembly could be comprised of the neurons which respond to its features, but what makes it an assembly is the coordination among them, and the perseverance of their activity after external stimulation ceases.

Moreover, Hebb postulated how assemblies would form fairly concretely:

> It is proposed first that a repeated stimulation of specific receptors will lead slowly to the formation of an 'assembly' of association-area cells which can act briefly as a closed system after stimulation has ceased; this prolongs the time during which the structural changes of learning can occur and constitutes the simplest instance of a representative process (image or idea).

Hebb's work is an early triumph for theory in neuroscience; to support the formation and maintenance of assemblies, he predicted that

> when an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells that A's efficiency, as one of the cells firing B, is increased.

(However, Hebb himself noted that this was "an old idea".) This form of neural plasticity is now known as "Hebbian" in his honor, and is the primary form of plasticity considered in this thesis. Although often aphorized as "fire together, wire together", what Hebb said

is actually more precise: There must be some causality in neuron A's interaction with B, which is only present when A's firing *precedes* that of B. Hebb's postulate predated by several decades the discovery of this form of long-term potentiation (i.e. a sustained increase in the efficacy of a synapse) in the hippocampus [25] and later across the neocortex [26, 27, 28].

The assembly would also enjoy experimental verification in its various aspects, albeit only more recently, given the difficulty of multi-unit recording at the necessary scale [29]. Here we touch on a small fraction of the work on population coding; assembly-specific evidence is addressed in the sequel. Perhaps the simplest evidence for a population code is when individual neurons respond to a diverse set of stimuli, yet these stimuli can nonetheless be distinguished at the population level [11]; examples of this are plentiful in higher-level brain areas, even sensory specific ones, including inferior temporal cortex (the highest vision-specific area) [30, 31], prefrontal cortex [32, 33], and somatosensory cortex [34], among thousands of other experiments.

Moreover, various structural properties of neuroanatomy seem to implicate distributed representations. In the mammalian brain, most neurons receive input from and broadcasts output to an enormous number of other neurons [35, 36], and most excitatory connections are weak [37], such that a given neuron's behavior depends (but only slightly) on a large number of neighbors. On the other hand, most inhibitory interneurons broadcast their inhibitory signals in a "blanket", densely and non-specifically, across their excitatory neighbors [38]. Taken together, these facts strongly suggest that most neurons are only one voice among many in a much larger neural circuit; as we will see, a model which formalizes these ideas also leads quite naturally to the formation of assemblies.

A note on the definition. The word "assembly" has different meanings to theorists and experimentalists. A theorist may understand an assembly as a population of densely interconnected neurons which are capable of exciting each other, and so gives rise to a dis-

tributed representation. For an experimentalist, these properties – dense interconnectivity, recurrent excitability – are difficult to measure, especially in a population of neurons from which activity is also being recorded. Hence, when an experimentalist refers to an assembly they may be talking about a population of neurons whose firing is merely correlated. Notably, this is implied by – but does not imply – the theorist's structural definition, as it does not preclude the possibility that correlation is brought about by some external drive. For the purposes of this thesis, we will be sure to distinguish between these two notions when it is relevant.

1.4.2 Experimental Evidence for Assemblies

Given their longevity in neuroscience, there is a great deal of interest in finding assemblies in the brain and an enormous body of experiments which have made steps in this direction. The task has both technical and conceptual hurdles. We have already discussed some of the conceptual hurdles: It is not obvious with the experimental tools we have how to distinguish between a population of neurons which has functionally synchronized firing from one with simply correlated firing, nor a population which encodes a concept from one which is simply associated with it [39]. On the technical side, we have only recently developed the technology to reliably and accurately record from substantial populations of neurons, and given the presumed sparsity of assemblies, and the huge number of neurons in the brain, even several thousand neurons in a local area may be inadequate to sample from a significant fraction of any one assembly. Additionally, precise experimental interventions in the brain, which might allow assemblies to be spontaneously activated or even created, are still in their infancy. There is an enormous literature which provides evidence for population coding, which is beyond the scope of this discussion; here, we will focus specifically on coordinated firing which occurs both in a representation context and spontaneously, independent of any sensory stimulation or motor behavior. As we will see next, there is ample evidence for neural activity of this type, across the cerebral cortex and the hippocampus

[40].

Although the hippocampus is not part of the cerebral cortex, and its apparent specialization for memory storage and navigation would seem to contraindicate the presence of assemblies there, it is nonetheless the first place where many key fingerprints of assembly activity were observed and thus essential to the study of the brain. The landmark work of Harris et al. [21] identified synchronous activity among groups of pyramidal neurons (henceforth "ensembles"), apparently clocked to the hippocampal theta wave. This activity seems to be anatomically distributed and quite sparse, with different ensembles perhaps having an inhibitory effect on each other [41]. Later work would find synchronized, sequential activity spanning milliseconds to seconds [42, 43, 44, 45, 46, 47, 48]. Interestingly, the sequential activity observed during behavior is later "replayed" during rest and sleep [49, 50, 48], which suggests that it arises from neuronal dynamics rather than external stimulation. A "preplay" phenomenon has also been observed [51, 52], where the ensembles displaying synchronized behavior during a novel task were similarly synchronized *before the task*, which raises the possibility that assemblies are not created by experience, but merely recruited from a pre-existing pool [53, 54]. Crucially, disruption of the synchronization of these ensembles is associated with memory deficits [55, 56, 57], which is evidence that they play a functional role.

In the cerebral cortex, evidence for assemblies is especially strong in primary sensory regions, and largely replicates the findings in the hippocampus. Persistent and anatomically sparse activity has been observed both spontaneously and in stimulus-evoked contexts [58, 59, 60, 61, 62, 63], with synchronization apparent at the 10ms timescale [64]. As in hippocampus, some degree of both preplay and replay is present [64, 61, 65, 66], and there is an apparent competitive effect among different ensembles [67, 68], as well as pattern completion [69, 70]. Among experimental interventions, the groundbreaking work of Carrilo-Reid et al. [62] used optogenetics to induce synchronous activity among a set of neurons in visual cortex, which displayed spontaneous activation, pattern completion

from the induced firing of a single one of its members, and persisted for several days. In visual discrimination tasks, the deactivation of the ensembles correlated with task variables totally impairs visually-guided behavior, while induced activation of the same ensembles substitutes for missing visual stimuli [63, 71, 72], which strongly suggests that they are fundamental to the representation of these concepts in the brain.

1.5 Contributions

The overall contribution of this thesis is to examine what sorts of algorithms for fundamental cognitive problems emerge from the interaction of assemblies, under non-algorithm-specific dynamics. Instead, the structure of an algorithm is encoded in the connectivity between neurons, and in the inhibitory properties of the network. The model of the brain we consider is NEMO, which attempts to strike a balance between tractability and biological fidelity. We are particularly interested in the most generally useful cognitive algorithms – those which are realizable with no specialized hardware. All of our results have both theoretical and experimental components, so an additional contribution beyond the work contained within this thesis is a collection of open-source software in the Python programming language for simulating NEMO efficiently.

We begin with a discussion of NEMO in Chapter 2, which includes a precise definition of each aspect of the model, and a comparison to other models of the brain. NEMO consists of a connectivity graph, and a set of dynamics parameterized by that graph, by which the initial state of the model evolves, under the dynamics and the influence of external input. This model is a natural formalization of a few, by now venerable, assumptions about how the brain works: (i) Connectivity is essentially random on within a sufficiently small area, and between neurons in more distal regions which happen to be connected by a fiber; (ii) a neuron's output is binary, defined by whether it emitted a spike within a small window of time (roughly the time it takes the synaptic potential to decay) or not; (iii) the effect of local inhibition is to uniformly suppress the excitatory population in a small area, to

maintain a certain level of activity; (iv) the primary mode of long-term synaptic plasticity is Hebbian. Broadly, the remainder of this thesis is a exploration of what algorithms are implementable in a model equipped with essentially only these mechanisms. From prior work, a particular interesting and fundamental operation is *the creation of assemblies*: The recruitment of a group of neurons, often with strong internal connectivity, to represent a concept. Assemblies form the basic unit of encoding which the more complex algorithms we examine here act upon.

The first algorithmic problem we discuss is linear classification (Chapter 3). In this setting, streams of stimuli, each representing examples of a single class, are presented one at a time. This evokes a sequence of neural responses in the model, which quickly form an assembly representing that class. If the classes are sufficiently distinguishable (i.e. linearly separable with a large margin) the assemblies for different classes will be almost entirely disjoint. We provide both theoretical results and experiments, including on simple real-world datasets, such as the machine learning benchmark MNIST. This represents a substantial generalization of prior work in NEMO, which considered only the creation of an assembly to represent a single stimulus, and gave no guarantees about the formation of multiple assemblies which might overlap.

Next, we consider the possibility of forming sequential assemblies (Chapter 4), well-connected chains of neurons, to represent sequentially structured stimuli. The most basic operation in this setting is sequence memorization – the formation of a chain to represent a sequence of distinct stimuli, which has a pattern completion, or sequence prediction, property: The presentation of any prefix of the stimulus sequence triggers the entire chain to fire sequentially. We also find that the recruitment of additional brain areas improves the speed of memorization, which may be connected to the well-known practical phenomenon that it is easier to memorize multiple sequences together (e.g. setting the alphabet to a tune). Beyond single sequences, we show that chains corresponding to the states of an arbitrary finite state machine can be realized, so that the model effectively implements a finite state

machine and can recognize the associated languages. When equipped with a memory (also realizable within a slight extension of NEMO), this shows that NEMO is Turing complete, a significant strengthening of previous results regarding the computational power of NEMO-derived models.

The final algorithmic problem we explore is statistical learning, that is, the estimation of certain statistical properties of a pool of samples. We demonstrate that Hebbian plasticity within NEMO readily estimates conditional probabilities from samples of random variables with outcomes encoded as assemblies; moreover, when NEMO is augmented with a source of random noise (in the form of i.i.d. perturbations to neuronal potentials) it quite naturally samples from this estimated distribution. This basic gadget then enables sampling from a variety of probability distributions which have straightforward conditional representations, particularly Markov chains, and as an extension, n-gram models of natural language.

Finally, we conclude with a discussion of open problems and future directions. Some of these are more technical, such as the number of stable assemblies which can be formed within NEMO. Others are more conceptual, such as how a NEMO-like model can perform more sophisticated learning, or form structured representations of a family of stimuli. A third category is the broad sorts of experimental findings that would support or refute NEMO's approach.

CHAPTER 2
NEMO, A MODEL OF THE BRAIN

In this section we describe NEMO, the mathematical model of the brain we use, along with certain novel extensions, mainly a family of more refined plasticity rules.

2.1 Modeling Assemblies: Prior Art

2.1.1 Early Work

As previously discussed, the earliest inklings of assemblies appear in the writing of Lorente de Nó, who realized recurrent connectivity could allow certain patterns of activity to sustain themselves [23], and Hebb, who of course coined the term, and hypothesized that suitable patterns of connectivity were engrained by associative learning (i.e. Hebbian plasticity) [22]. Moreover, Hebb hypothesized that assemblies could support an early form of auto-associative memory, or pattern completion, in that the entire assembly could quickly be brought online when only a small fraction of its neurons initially fired. These early thinkers were working without the benefit of various key discoveries about the brain that are foundational to modern neuroscience – not least, the existence of inhibitory neurons, which provide the negative feedback necessary to keep a strongly recurrent system from runaway excitation.

2.1.2 Marr's Models of Associative Memory

Although not framed in terms of assemblies, the early work of David Marr in the late 1960s and early 1970s on memory and learning in the hippocampus and cerebral cortex are foundational to the modern conception of assemblies, and even more so the work in the thesis [73, 74]. Marr pushed the field substantially forwards by framing what his models were

meant to accomplish in computational terms, and making his models mathematically precise – all of the concrete details are laid out – so that things can be proved about them to the standards of mathematical rigor. In this early work the implementation and computational perspectives are clearly present, albeit difficult to distinguish, preceding and perhaps motivating Marr's later formulation of the three levels of computational modeling – implementation, algorithmic, and computational.

Marr's models of the cerebellum [75], cerebral cortex [73], and hippocampus [74] are closely related. The crucial difference is the computational task they are meant to perform: Marr's cerebellum and cerebral cortex extract structure in the patterns that stimulate them, forming classifiers, while hippocampus is seen as a mere temporary storage for memories. We frame the discussion here in terms of the memory problem that hippocampus is meant to address but the situation for classification is essentially the same.

The hippocampal model [73] is a neural network with three layers: input, "codon", and output. Units in this network have binary outputs. Connectivity is directed and random, although not simply layer to layer; the input and codon layers are divided into blocks, with input units only having a chance to connect with codon units in the same block, while the output layer is recurrently connected. There is a fixed fraction of neurons which can be simultaneously active in each layer; neurons fire if they have the largest input, within this fraction, and do not otherwise. A synapse is said to be active if its presynaptic neuron fired; a synapse is said to be modified if it was set as such during training.

The items to be stored ("events") are binary patterns on the input layer, which lead to a "simple" representation of the event in the output layer after their activity is propagated through the network. The intended retrieval behavior is considered to occur when presentation of an arbitrary fraction of the active neurons in an event at the input layer evokes activity in the output layer very close to the simple representation of that event. (Marr intended to develop a model of how activity would then propagate back to the input layer to recover the original event in a later paper, which unfortunately never materialized.) The

dynamics of activity during the memorization and retrieval phases are different. During the memorization phase neurons fire in a given layer if their number of activated synapses from the previous layer exceeds a threshold, which is set to ensure that the correct fraction is active. During the retrieval phase neurons fire in a given layer if their number of activated & modified synapses from the previous layer exceeds one threshold, *and* the fraction of their activated synapses from the previous layer which are modified exceeds another. The two thresholds are set jointly to ensure the activity constraint is met.

This additional mechanism is useful because during the retrieval phase for an event, the neurons which fire may be a mix of correct units (which did fire during the memorization of that event) and spurious units (which didn't). This means that the number of active synapses to neurons in the subsequent layer will be different, in general, from during the memorization phase. However, all synapses between correct units will be modified, while synapses from spurious to correct units will not be. Thus, so long as a sufficiently large fraction of units are correct, many spurious units in the subsequent layer can be filtered by ensuring that enough of their active synapses are modified. The point is to gradually increase the fraction of modified synapses required as information moves through the network (including through time, once the recurrence of the output layer has taken effect) so that the fraction of correct units increases consistently. For a certain choice of the parameters, Marr showed that only about $1/40$ of the inputs associated with an event need to be active to ensure that the entire simple representation is eventually recovered.

The influence of this model on this thesis is profound. Several core assumptions of Marr's model – random connectivity, and local inhibition as a dynamic threshold which ensures a constant level of activity – are by now well-enshrined ideas about how the brain works that have been carried forward into NEMO. In particular, Marr's dynamic thresholds amount to precisely a k-winners-take-all dynamic. Additionally, the recurrent connectivity at the output layer means that Marr's simple memories are assemblies in the strongly-connected sense; the activation of a sufficiently large fraction of the memory is enough

to bring and keep the entire memory active over several time steps. Besides these details, Marr's broader perspective on theorizing about the brain is likely even more influential; he merrily interleaves implementational and algorithmic considerations in developing his model, taking into account both facts about the brain and the structure of the computational task to be performed.

2.1.3 Hopfield's Model of Associative Memory

In the early 1980s, the physicist John Hopfield took a more general approach to the problem [76]. Specifically, drawing from the core insight of statistical physics that large groups of simple, interacting components often display collective phenomena (e.g. the spontaneous magnetization of a chunk of iron), Hopfield deployed the tools and models of that field to analyze the emergence of computation among networks of simple neurons. Hopfield was certainly aware of Marr's work and perhaps dissatisfied with the level of familiarity with neuroanatomy which Marr demanded of his readership; he tried to abstract away from perfectly matching what was known about the brain and emphasized instead the qualitative phenomena one might expect to see.

The key observation that Hopfield made is that networks of recurrent, thresholding neurons can encode "attractors" within their weights – states which are not only stable once reached but also draw in nearby states, under the dynamics of the network. Moreover, such attractors correspond to densely connected groups of neurons. This justifies thinking of these attractors as assemblies, because they display reverberation and pattern completion in virtue of strong recurrent connectivity. More precisely, a Hopfield network is a set of neurons with outputs $v_i(t) \in \{-1, 1\}$, for $i = 1, \ldots, n$ and $t = 0, 1, 2, \ldots$. (Whether the binary values are $\{0, 1\}$ or $\{-1, 1\}$ depends on the precise model but does not much effect on its behavior.) Each neuron has a threshold θ_i and between each pair of neurons i and j there is a weight, $w_{i,j}$; these are fixed parameters of the system. The recurrence relation

which defines the network is

$$v_i(t+1) = \begin{cases} +1 & \sum_{j=1}^{n} w_{i,j} v_j(t) > \theta_i \\ -1 & \sum_{j=1}^{n} w_{i,j} v_j(t) \leq \theta_i \end{cases}$$

In the tradition of statistical physics, there is a natural energy function associated with the state of a Hopfield network:

$$E_{w,\theta}(v_1, \ldots, v_n) = \sum_{i=1}^{n} v_i \left(\theta_i - \frac{1}{2} \sum_{j=1}^{n} w_{i,j} v_j \right).$$

It is usually assumed that the weights are symmetric ($w_{i,j} = w_{j,i}$) and $w_{i,i} = 0$ for all i. This is because these are sufficient conditions to guarantee that the recurrence relation finds and stays at a local minimum of the energy, when each unit is updated individually:

Theorem 1 (Hopfield Convergence)**.** *Suppose that $w_{i,j}$ are symmetric and zero diagonal. Let $v_1, \ldots, v_n \in \{+1, -1\}$, and define*

$$u_1 = \sum_{j=2}^{n} w_{i,j} v_j - \theta_i$$

with $v_1' = +1$ if $u_1 > 0$ and -1 otherwise. Then

$$E_{w,\theta}(v_1', v_2, \ldots, v_n) = E_{w,\theta}(v_1, v_2, \ldots, v_n) + u_1 \cdot (v_1 - v_1') \leq E_{w,\theta}(v_1, v_2, \ldots, v_n).$$

Proof. Note that

$$E - \hat{E} = (v_1 - v_1') \left(\theta_1 - \sum_{i=2}^{n} \frac{w_{1,i} + w_{i,1}}{2} v_i \right).$$

Then, by symmetry we have $\frac{w_{1,i}+w_{i,1}}{2} = w_{1,i}$, so

$$E - \hat{E} = (v_1 - v_1') \left(\theta_1 - \sum_{i=2}^{n} w_{1,i} v_i \right) = u_1 \cdot (v_1' - v_1)$$

which rearranges to the claimed equality. For the inequality, by definition of u_1, if $v_1' = +1$ we have $u_1 \geq 0$ and $v_1' - v_1 \geq 0$ so their product is as well. If $v_1' = -1$ then $u_1 < 0$ and $v_1' - v_1 \leq 0$ so their product is nonnegative. In either case, $E \leq \hat{E}$. □

So, to find an attractor in a given Hopfield network it suffices to repeatedly update individual units. This is analogous to the retrieval phase in Marr's model. However, what about embedding a desired pattern within the network? It turns out that for $p_1, \ldots, p_n \in \{-1, 1\}$, setting the weights as $w_{i,j} = p_i p_j$ for all $i \neq j$ and the thresholds $\theta_i = 0$ yields a network from which $(p_1, \ldots, p_n)$ can be recovered by starting from any state with one more than half its units correct.

Theorem 2 (Hopfield Retrieval)**.** *Let $v_1, \ldots, v_n \in \{-1, 1\}$ such that $\sum_i p_i v_i > 1$. Then for any i, we have*

$$p_i \sum_{j \neq i} w_{i,j} v_j > 0.$$

Hence, updating any unit will maintain its value if correct or flip it if incorrect.

Proof. Fix i, and suppose that $\sum_i p_i v_i > 0$. We have

$$p_i \sum_{j \neq i} w_{i,j} v_j = p_i \sum_{j \neq i} p_i p_j v_j = \sum_{j \neq i} p_j v_j = \sum_{j=1}^{n} p_j v_j - p_i v_i > 1 - p_i v_i \geq 0.$$

□

Even in this network storing only a single pattern, the Hopfield network suffers from storing spurious patterns: The pattern $-p_1, \ldots, -p_n$ is also an attractor, and will be converged to from any pattern with one more than half of its units incorrect. One can also store

more than one pattern in the network, say $p^1, \ldots, p^m$, by setting

$$w_{i,j} = \sum_{k=1}^{m} p_i^k p_j^k.$$

This property is of particular interest to neuroscientists because it is essentially Hebb's rule: If the patterns had been made to fire in some arbitrary order, it is as though the weight between units i and j was incremented for each pattern in which they took the same sign and decremented each time they differed. Of course, then there is a question of how many patterns can be stored and retrieved successfully. Again, in the tradition of statistical physics, analysis proceeds with uniformly random patterns over $\{-1, +1\}^n$, as $n \to \infty$, and retrieval is defined such that with probability tending to 1, any state with a $1/2 + \epsilon$ fraction of its units matching one of the stored patterns will converge to that pattern (where ϵ is an arbitrarily small constant with respect to n). On the basis of some small-scale numerical experiments Hopfield initially guessed that capacity was linear with a small multiplier, $0.15n$; later work would eventually show that the asymptotic capacity is in fact $\frac{n}{4 \log n}$ [77].

There is also a more general computational perspective on Hopfield networks [78]: Optimization problems can be encoded in a network's weights, so that its lowest energy states correspond to optimal solutions to the original problem. This applies even to computationally hard optimization problems, such as the traveling salesman problem. (P $\neq$ NP remains open because the Hopfield dynamics seem to take exponentially long to find these states, if they don't get stuck in a non-optimal attractor along the way.) Recall the traveling salesman problem: Given a set of n cities, with symmetric distances $d_{i,j} = d_{j,i}$ between cities, find an ordering of the cities $i_1, \ldots, i_n$ which minimizes the sum $d_{i_1,i_2} + \cdots + d_{i_{n-1},i_n} + d_{i_n,i_1}$. The encoding of this problem in a Hopfield network will have n^2 units, say (i, j) for $i, j = 1, \ldots, n$, arranged in a table with n rows and n columns; we interpret a 1 at (i, j) to indicate that $i_j = i$. The weights of this network need to accomplish two things. The first

is to grant lower energies to valid tours, those for which there is exactly one 1 per row and column; the second is to grant lower energies to shorter tours. An energy function which accomplishes both is

$$E(v) = \frac{A}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{j\neq i} v_{ij}v_{ik} + \frac{B}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k\neq i} v_{ij}v_{kj} + \frac{C}{2}\left(\sum_{i=1}^{n}\sum_{j=1}^{n} v_{ij} - n\right)^2 + \frac{D}{2}\sum_{i=1}^{n}\sum_{j\neq i}\sum_{k=1}^{n} d_{i,j}(v_{ik}v_{j(k+1)} + v_{ik}v_{j(k-1)}$$

Note that the first term is zero exactly when there is no more than one 1 per column, the second when there is no more than one 1 per row, and the third when there are a total of n 1's in the entire table. The fourth term then corresponds to the length of the tour whenever the unit values specify a valid one. Crucially, however, this is the energy function of a Hopfield network with weights

$$w_{ij,kl} = -A\mathbb{1}_{i=k}\mathbb{1}_{j\neq l} - B\mathbb{1}_{j=l}\mathbb{1}_{i\neq k} - C - Dd_{i,k}(\mathbb{1}_{l=j-1} + \mathbb{1}_{l=j-1}).$$

So, the Hopfield dynamics will find a local minimum for this problem. Of course, it is difficult to get any provable guarantees about the quality of such a point, and indeed Bruck & Goodman [79] showed that no polynomial-sized network (i.e. the number of units grows as a polynomial in the number of cities) can find even a e-approximate point in general unless $P = NP$.

Of course, Hopfield's model is a step backwards from Marr's in terms of biological fidelity. Most egregiously, inhibition in the Hopfield network is simply a consequence of negative weights, as units play both excitatory and inhibitory roles for their neighbors depending on their weights and outputs. Additionally, connections between units are dense, and symmetric, both of which are difficult constraints for the brain to fulfill directly between neurons. These latter two issues could be resolved by re-envisioning the Hopfield unit as the aggregate of a larger group of neurons – but the negative weights require

precisely-targeted lateral inhibition between these groups, which is at odds with what is known about the predominate form of inhibition in the brain, local and largely untargeted. So, this raises some serious concerns about whether the Hopfield network actually captures the dynamics leading to assemblies in the brain, or merely exhibits the reoccurence of this phenomenon under various realizations. Nonetheless, Hopfield networks continue to loom large in modern neuroscience theory and are a common point of comparison for NEMO.

2.1.4 Valiant's Neuroidal Model

Leslie Valiant's work around the turn of the millennium [80, 81, 82] represents a return to Marr's highly neuroanatomically-constrained approach, albeit with an eye towards principles of organization over the exact measurements (which is important for a unified perspective over species). NEMO inherited a great deal, in terms of assumptions about the brain, how they are abstracted into a model, and the computational tasks that are considered, from Valiant's perspective. Beyond Valiant's specific model there is the trail he blazed for theoretical computer scientists to venture into neuroscience, without which this thesis would not exist.

Concretely, Valiant opens with a few observations about neuroanatomy: The mammalian cerebral cortex has between 10 million and 10 billion neurons (with mice at the lower end of the spectrum and the great apes, cetaceans, and elephants at the other), each of which communicates with tens of thousands of its neighbors, and apparently needs a spike from tens to hundreds of them in order to undergo a spike itself. Finally, since the cycle of emitting a spike takes about 10 ms, and significant cognition can occur in fractions of a second, the basic algorithms of the brain certainly do not require more than 100 steps. He then proposes a model which respects all of these constraints and implements two operations: "join" forms a new memory which represents the conjunction of existing memories, and "link" binds existing memories together so that they trigger each other to fire.

The neuroidal model builds on previous work, considering memory formation in net-

works of excitatory, thresholding linear neurons. Early work by Griffith [83] proposed that memories could be sequences of k-neuron sets in a fully-connected graph; if each synapse's strength is at least $1/k$ as large as the threshold, a signal can be propagated down the chain. Far later, Moshe Abeles [37] generalized this idea to sequences of k neuron sets, where each neuron connects to $h \leq k$ neurons in the next set of the chain, allowing for memories to be stored as "synfire chains", essentially sequential assemblies, and showed that for suitably dense random graphs such chains exist. Of course, this property alone is insufficient, computationally; many such chains need to exist to store a large number of memories, and they need to be discoverable quickly and bound to their memory by a neural algorithm. So, memories are represented by the synchronous firing of sets of roughly r neurons, and considers the choices of parameters (number of neurons, edge density, threshold, and memory size r) under which joining and linking are possible for arbitrary pre-existing memories.

The neuroidal model is defined on a connectivity graph as follows. Each neuroid and synapse has its state. The state of a neuroid specifies whether it is firing, while the next state depends on the current state and the total input (i.e. the sum of inputs from firing neighbors). The state of a synapse specifies its weight; the next state of a synapse depends on the current state of the synapse, the firing status of its pre-synaptic neuroid, and the current state and total input of the post-synaptic neuroid. An algorithm in this model is a specification of the transition function at each neuroid and synapse. Valiant's algorithm for join makes choosing a set of neurons to represent a conjunction of pre-existing memories particularly simple: it is simply the set of neurons with enough input from those combined memories to fire.

Although Valiant intended this model to be so simple that "there can be no doubt that neurons are capable of doing at least that much", it is unclear how well neuroids actually satisfy that criterion; arbitrary state and synaptic weight changes are quite a powerful computational tool that goes well beyond the relatively small set of neuronal firing and plasticity motifs that are known experimentally. This power is perhaps necessary in a model such as

this one where there is no signal to coordinate the neurons; however, as we will see, simple abstractions of local inhibition turn out to be quite a powerful and robust tool for allowing coordination. Additionally, a wider range of parameters is possible when one does not need to enforce that very nearly r neurons exceed their threshold due to connectivity, for instance, and this is instead enforced by a winner-take-all mechanism, as in NEMO.

2.1.5 Artificial Neural Networks at Large

Given that deep learning has made progress in leaps and bounds on problems that were previously thought to be fundamental to human intelligence, it is tempting to ask whether there might be some deeper functional similarity between the brain and artificial neural networks trained by deep learning. Indeed, there is some anecdotal evidence in this direction: Deep neural networks optimized solely for task performance have been shown to have activity predictive of and predicted by activity in the associated regions of the brain, especially in vision [84, 85, 86], but also audition [87], language [88], and motor control [89], among others. On the other hand, it may be that this convergence in representation arises under different dynamics in the brain and in artificial neural network (ANN)s, especially since the brain seems to lack the precise feedback connectivity required for backpropagation [90]. The broader "neoconnectionist" research program [91, 8] takes the stance that issues with specific types of ANNs does not diminish the potential of the entire category (of which, of course, NEMO is a member) to model the brain, which our approach here is essentially aligned with. NEMO simply arises from the hypothesis that some simple, effective mechanisms for fundamental problems may emerge from relatively well-understood and empirically-supported ideas about how the brain works. In other words, this thesis is the exploration of a specific class of ANNs which are intended to better capture biological detail.

2.2 Connectivity

An instantiation of NEMO consists of a finite number of *brain areas*, each of which consists of n neurons. Each brain area is recurrently connected by a directed random graph, where each directed edge (or synapse) is present with probability p, independently of all others. Each pair of brain areas may be directionally connected, and if so by a directed bipartite random graph with the same edge probability p. All synapses have a weight, assumed to be initially 1. Stimuli are presented in special *input* areas.

2.3 The Dynamical System

Time proceeds in discrete steps. At each time step, every neuron in the graph computes its total input, which the is the sum of all weights of incoming synapses from neurons which fired on the previous step. In each area, the top k neurons with highest total input fire, and the rest do not. This k-winners-take-all (winner(s)-take-all (WTA)) process, or k-cap[1], models local inhibition and abstracts excitatory/inhibitory balance. Synaptic weights are nonnegative and subject to Hebbian plasticity: each time neuron j fires followed by neuron i, the weight w_{ij} from j to i increases by $\pi(w_{ij})$, where $\pi\colon [0,\infty) \to [0,\infty)$ is the plasticity function. Formally, with $x_A(t)$ the set of neurons firing in A, $W_{A,B}$ the weights from area B to area A, and $\mathcal{A}$ the set of areas, the update equations are

$$x_A(t+1) = k\text{-cap}\left(\sum_{B\in\mathcal{A}} W_{A,B}(t)x_B(t)\right) \qquad \forall A \in \mathcal{A}$$

$$W_{A,B}(t+1) = W_{A,B}(t) + x_A(t+1)\cdot x_B(t)^\top \odot \pi(W_{A,B}(t)) \qquad \forall A, B \in \mathcal{A}$$

[1] We will also refer to the set of firing neurons in a brain area as a "cap".

where $\odot$ is element-wise multiplication and the scalar function π is applied element-wise[2]. The function k-cap maps a n-dimensional vector to the indicator of its k largest elements, with ties broken according to some fixed order, and so $x_A(t)$ is a $\{0,1\}^n$ vector with either 0 or k nonzero elements. Neurons may also have their activity perturbed by random noise, in which case the update is

$$x_A(t+1) = k\text{-cap}\left(\sum_{B\in\mathcal{A}} W_{B,A}(t)x_B(t) + z_A(t)\right)$$

where $z_A(t)$ is a random vector with components independent of each other and the random graph.

In more complex algorithms, we will sometimes assume that brain areas are periodically completely inhibited, or the connections between them are deactivated, especially in patterns of alternation or "lazy" alternation – for instance, area A is allowed to fire once (i.e. form a top k cap) while B is inhibited, then area B is allowed to fire a fixed number of times while A is inhibited, and this process repeats. Since this inhibitory activity does not depend on the state of the model we exclude it from the update equations above for simplicity. In the brain, such functionality might be achieved by long-range interneurons which inhibit one area when excited by another, or as a result of the brain's more global rhythms [39].

2.4 Plasticity

In full generality, the synapses of NEMO may be subject to arbitrary, positive, Hebbian plasticity. That is, there exists a function π mapping a current weight to a weight increment; the Hebbian condition is that when neuron i fires at time t followed by neuron j at $t+1$, with $w_{ij}(t)$ the weight of the synapse between them at time t, the dynamic on the weight is $w_{ij}(t+1) = w_{ij}(t) + \pi(w_{ij}(t))$, and $w_{ij}(t+1) = w_{ij}(t)$ otherwise. Positivity simply

[2]Since we do not distinguish between non-existent synapses and those with weight 0, we will always assume that $\pi(0) = 0$ so that the update rule only affects extant synapses.

mandates that $\pi \geq 0$. In practice, we focus on rules which are multiplicative ($\pi(w) = \beta w$ for some $\beta > 0$) or exponentially decay ($\pi(w) = \min\{\alpha, c\dot{e}^{-\lambda w}\}$ for $\alpha, c, \lambda > 0$).

2.5 Assemblies in NEMO

Concretely, an *assembly* in NEMO is a set of roughly k neurons in a single brain area with strong recurrent connectivity: the input to any neuron in the set from the rest of the set is substantially higher than kp, the expectation if recurrent connections had weight 1. Hence, here we imagine that an assembly is confined to a local region, which is more of a convenient definition than the necessary place for assemblies within a broader theory of cognition. One can easily imagine "meta-assemblies", assemblies of assemblies, which are distributed across many brain areas. A *cap* is a set of k neurons within a brain area, which happen to fire together at least once. It may be a distributed representation of something (much like an assembly) but makes no assumptions about the neurons' ability to excite each other; more generally it is a convenient term for referring to the neurons selected to fire by inhibition.

2.6 Mathematical Preliminaries

2.6.1 Probability and Random Variables

Probabilistic analysis is essential to every result in NEMO. What follows is a brief recapitulation of the key ideas; see for instance Billingsley's *Probability and Measure* [92] for a comprehensive treatment.

Basics of Probability A probability space is a pair $(\Omega, \Pr)$, where Ω is a set of outcomes, and $\Pr$ maps subsets of Ω to probabilities (values in $[0, 1]$). A random variable X is a function on Ω, which maps elements of Ω to observations or outcomes. Intuitively, an element $\omega \in \Omega$ is a complete specification of the world (without any uncertainty), while $X(\omega)$ is a measurement of the world that leaves uncertainty about the precise value of ω.

X must be sufficiently well-behaved so that the probability of outcomes are well-defined; that is, we may write $\Pr(X \in A) = \Pr(X^{-1}(A))$ for a fairly comprehensive class of sets in the co-domain of X. (In measure-theoretic terms, X is a measurable function.) When the co-domain of X is a field, one can define its expectation as $\mathbb{E}X = \int_\Omega X d\Pr$, and its variance as $\operatorname{Var} X = \mathbb{E}X^2 - (\mathbb{E}X)^2$, whenever these integrals exist. An important special case is when the range of X is finite or countably infinite; in this case, the expectation of any function of X can be written more simply as

$$\mathbb{E}f(X) = \sum_{x_i \in X(\Omega)} f(x_i) \cdot \Pr(X = x_i).$$

The interesting part of probability theory is when one has a set of random variables on the same probability space; in this case there are many interesting questions about how they interact. One especially interesting structure is independence, which holds for a set of random variables $X_1, \ldots, X_n$ when $\Pr(X_1 \in A_1, \ldots, X_n \in A_n) = \Pr(X_1 \in A_1) \cdots \Pr(X_n \in A_n)$ for all measurable $A_1, \ldots, A_n$. An infinite set of random variables is independent if all its finite subsets are. An extraordinarily useful property is that for any random variables X, Y, $\mathbb{E}[X + Y] = \mathbb{E}X + \mathbb{E}Y$, *regardless of independence*. However, if X and Y are independent, we also have $\operatorname{Var}(X + Y) = \operatorname{Var} X + \operatorname{Var} Y$. It is common to consider ensembles of independent and identically distributed random variables, which we abbreviate as i.i.d.

Any random variable X has a well-defined cumulative distribution function, $F_X(x) = \Pr(X \leq x)$. If there exists a function f_X such that

$$F_X(x) = \int_{-\infty}^{x} f_X(y)\, dy$$

then it is called the density of X, and can be used to compute expectations with respect to X as

$$\mathbb{E}f(X) = \int_{-\infty}^{\infty} f(x) f_X(x)\, dx.$$

Two important classes of distributions throughout this thesis are the binomial distribution and the normal distribution. The binomial distribution $\mathcal{B}(n, p)$ is the number of successes in n independent trials, each of which has probability of success p. The normal distribution $\mathcal{N}(\mu, \sigma^2)$ is the distribution with density $\phi(t) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$. When a random variable follows a distribution $\mathcal{D}$, we will write $X \sim \mathcal{D}$.

Convergence and Limits of Random Variables There are three important modes of convergence of sequences of random variables, $X_1, X_2, \ldots$ to a limiting random variable X. Almost sure convergence occurs when $\Pr(\lim_{n\to\infty} X_n = X)$. Convergence in probability occurs when for all $\epsilon > 0$, $\lim_{n\to\infty} \Pr(|X_n - X| > \epsilon) = 0$. Convergence in distribution occurs when $\lim_{n\to\infty} F_{X_n}(x) = F_X(x)$ for all x at which F_X is continuous. Notably, each of these notions is weaker than the previous one, so almost sure convergence implies convergence in probability implies convergence in distribution.

One important result which makes the entire field of statistics possible is the law of large numbers, which states that the empirical mean of a collection of i.i.d. random variables converges to their (shared) expectation in all three of the above senses. In mathematical notation, for i.i.d. $X_1, X_2, \ldots$ such that $\mathbb{E}X_1 = \mu$, define

$$M_n = \frac{X_1 + \cdots + X_n}{n}.$$

Then $M_n \to \mu$ almost surely. On the other hand, the random variable

$$Y_n = \frac{X_1 + \cdots + X_n - n\mu}{\sigma\sqrt{n}}$$

can be seen to have zero mean and unit variance. The classical central limit theorem states that $Y_n \to Z \sim \mathcal{N}(0, 1)$ in distribution, which is the start of the normal distribution's central role in probability theory.

So, it should then be clear that $\mathcal{B}(n, p)$ satisfies the central limit theorem: If $X_n \sim$

$\mathcal{B}(n,p)$, then $\frac{X_n - np}{\sqrt{np(1-p)}} \to Z$ in distribution. However, in many cases p will depend on n. In any case, we may also want quantiative bounds on how $\Pr(X_n \leq t)$ compares to $\Pr(Z \leq \frac{t-np}{\sqrt{np(1-p)}})$. Both cases are covered by the Berry-Esseen theorem:

Theorem 3 (Berry & Esseen)**.** *Let $X_1, \ldots, X_n$ be independent random variables with $\mathbb{E}[X_i] = 0, \mathbb{E}[X_i^2] = \sigma_i^2$ and $\mathbb{E}[|X_i|^3] = \rho_i < \infty$. Let*

$$S = \left(\sum_{i=1}^{n} \sigma_i^2\right)^{-1/2} \left(\sum_{i=1}^{n} X_i\right)$$

Denote by F_S the CDF of S, and Φ the standard normal CDF. There exists some constant C such that

$$\sup_{x \in \mathbb{R}} |F(x) - \Phi(x)| \leq C \left(\sum_{i=1}^{n} \sigma_i^2\right)^{-1/2} \max_i \frac{\rho_i}{\sigma_i^2}$$

Tail Bounds. Crucial to the analysis of probability is the tail bound, which provides some control on the probability that a rare event occurs (for example, a random variable is very far from its mean). An elementary result about binomial random variables is the ubiquitous Chernoff bound, which states that the probability of being far from the mean drops off exponentially quickly.

Lemma 4 (Chernoff Bound)**.** *Let $X_1, \ldots, X_n$ be independent random variables, with X their sum and $\mu = \mathbb{E}X$. We have*

$$\Pr(X \geq (1+\epsilon)\mu) \leq \exp\left(-\frac{\epsilon^2 \mu}{2+\epsilon}\right)$$
$$\Pr(X \leq (1-\epsilon)\mu) \leq \exp\left(-\frac{\epsilon^2 \mu}{2}\right)$$

There is also a tail bound for the Gaussian distribution with similar behavior:

Lemma 5 (Gaussian Tail)**.** *For $X \sim \mathcal{N}(0,1)$ and $t > 0$,*

$$\Pr(X > t) \leq \frac{1}{\sqrt{2\pi}t} e^{-t^2/2}.$$

For $X \sim \mathcal{N}(\mu, \sigma^2)$ and $P(X > t) = p$, we have

$$t = \mu + \sigma\sqrt{2\ln(1/p) + \ln(2\ln(1/p))} + o(1)$$

where $o(1)$ is w.r.t. $1/p$.

Proof. Recall that

$$\Pr(X > t) = \int_t^\infty \frac{1}{\sqrt{2\pi}} e^{-\tau^2/2} \mathrm{d}\tau$$

Then observing that for $\tau \geq t$,

$$e^{-\tau^2/2} \leq \frac{\tau}{t} e^{-\tau^2/2}$$

we have

$$\Pr(X > t) \leq \frac{1}{\sqrt{2\pi}t} \int_t^\infty \tau e^{-\tau^2/2} = \frac{1}{\sqrt{2\pi}t} e^{-t^2/2}$$

For the second part, simply solve for t. □

Order Statistics. This thesis in particular uses and pushes forward the theory of order statistics. The kth order statistic of a set of random variables $X_1, \ldots, X_n$ is the kth largest value among them, often denoted as $X_{(k)}$. This notion is crucial here because we will often have a set of randomly distributed neuronal potentials, of which the k-winners-take-all process chooses the k largest to fire; all of their potentials will meet or exceed the $(n-k)$th order statistic, and barring ties, k-WTA is equivalent to setting a firing threshold at the $(n-k)$th order statistic.

Our first result is a fairly elementary one about the order statistic of i.i.d. binomial random variables, derived using the Chernoff bound.

Lemma 6 (Binomial Order Statistics)**.** *Let $X_1, \ldots, X_n$ be i.i.d. $\mathcal{B}(m, p)$ random variables, with $mp \geq 3\log n$. Let $1 \leq k \leq n/4$, and Y_k be the kth largest value among $X_1, \ldots, X_n$.*

Then

$$\Pr\left(Y_k \geq mp + \sqrt{3mp\log\frac{2n}{k}}\right) \leq e^{-k/8}$$

Proof. First, for the upper bound, for any t, observe that $Y_k \geq t$ implies that $\sum_{i=1}^{n} \mathbb{1}_{X_i \geq t} \geq k$, and we have

$$\sum_{i=1}^{n} \mathbb{1}_{X_i \geq t} \sim \mathcal{B}\left(n, \Pr(X_1 \geq t)\right).$$

Therefore, applying the Chernoff bound (Lemma 43),

$$\Pr(Y_k \geq t) \leq \Pr\left(\sum_{i=1}^{n} \mathbb{1}_{X_i \geq t} \geq k\right) \leq \exp\left(-\frac{(k - n \cdot \Pr(X_1 \geq t)))^2}{2k}\right).$$

whenever $\Pr(X_1 \geq t) \leq k/n$. Again by the Chernoff bound, we have

$$\Pr\left(X_1 \geq mp + \sqrt{3mp\log\frac{2n}{k}}\right) \leq \exp\left(\frac{3mp\log\frac{2n}{k}}{2mp + \sqrt{3mp\log\frac{2n}{k}}}\right) \leq \frac{k}{2n}$$

and so

$$\Pr(Y_k \geq mp + \sqrt{3mp\log\frac{2n}{k}}) \leq e^{-k/8}.$$

□

The following proceeds similarly but for a different range of parameters:

Lemma 7. *For $m \in [n]$, let $X_1, \ldots, X_n$ be i.i.d. (m, p) binomial random variables, with $X_{(i)}$ their ith order statistic. Let $\Delta = |m - n/2|$ and set $k = \max\{\lfloor n/2 + \frac{\sqrt{np}}{6} \cdot \Delta \rfloor, n\}$. Then the following hold:*

1. *For $m \geq n/2 + 6p^{-1}$, we have*

$$\Pr(X_{(n-k)} \leq np/2) \leq e^{-\Omega(\Delta^2 p)}$$

2. *For $m \leq n/2 - 6p^{-1}$, we have*

$$\Pr(X_{(k)} \geq np/2) \leq e^{-\Omega(\Delta^2 p)}$$

Proof. First, if $\Delta \geq \sqrt{4n \log n}$, then the Chernoff bound gives

$$\Pr(X_1 \leq np/2) \leq \exp\left(-\frac{4 \log n}{1 + 2\epsilon}\right) \leq \frac{1}{n^2}$$

and so by the union bound we have $X_{(1)} > np/2$ w.p. $1 - 1/n$. For $\Delta < \sqrt{4n \log n}$, set

$$Y = \sum_{i=1}^{n} \mathbb{1}_{\{X_i > np/2\}}$$

and note that Y is a (n, q_m)-binomial random variable, where

$$q_m = \Pr(X_1 > np/2)$$

Moreover,

$$\Pr(X_{(n-k)} \leq np/2) = \Pr(Y \leq k) \leq \exp\left(-\frac{(nq_m - k)^2}{2nq_m}\right)$$

via the Chernoff bound. Then using the Berry-Esseen theorem,

$$1 - q_m = \Pr(X_1 \leq np/2) \leq \Phi\left(-\frac{\Delta p}{\sqrt{mp(1-p)}}\right) + \frac{1}{2\sqrt{mp(1-p)}}$$

where $\Phi \colon \mathbb{R} \to [0, 1]$ is the normal CDF. By Taylor's theorem, there exists $t \in [0, \Delta p]$ such that

$$\Phi\left(-\frac{\Delta p}{\sqrt{mp(1-p)}}\right) = \Phi(0) - \frac{\Delta p}{\sqrt{mp(1-p)}} \cdot \phi\left(\frac{t}{\sqrt{mp(1-p)}}\right)$$

where $\phi \colon \mathbb{R} \to \mathbb{R}$ is the normal PDF. Noting that ϕ is monotone decreasing for $t \geq 0$, we

have

$$\Phi\left(-\frac{\Delta p}{\sqrt{mp(1-p)}}\right) \leq \frac{1}{2} - \frac{\Delta p}{\sqrt{mp(1-p)}} \cdot \frac{\exp\left(-\frac{(\Delta p)^2}{2mp(1-p)}\right)}{\sqrt{2\pi}}$$

As $\Delta = o(n)$, $m \leq n$, and $\Delta p \geq 6 \geq 2\sqrt{2\pi}$, it follows that

$$q_m \geq \frac{1}{2} + (1 - o(1))\frac{\Delta p}{\sqrt{8\pi np}}$$

Hence,

$$nq_m - k \geq \frac{n}{2} + (1 - o(1))\sqrt{\frac{np}{8\pi}}\Delta - \frac{n}{2} - \frac{\sqrt{np}}{6}\Delta \geq \frac{\sqrt{np}}{2}\Delta$$

which gives the claim:

$$\Pr(X_{(n-k)} \leq np/2) \leq \exp\left(-\frac{\Delta^2 \cdot np}{8nq_m}\right) \leq \exp\left(-\frac{\Delta^2 p}{4}\right)$$

The proof of the second claim is similar. □

A classic result about order statistics is that the npth order statistic of $X_1, \ldots, X_n$ satisfies a central limit theorem as n trends to ∞ and p is constant [93]. More precisely,

$$\frac{f_X(F_X^{-1}(p))\sqrt{n}}{\sqrt{p(1-p)}}\left(X_{\lceil np \rceil} - F_X^{-1}(p)\right) \to Z \sim \mathcal{N}(0,1).$$

Here, we prove a substantial generalization of this result by giving sufficient conditions for a collection of independent, but *not identically distributed* sequence of random variables to have order statistics which converge in distribution to the normal. We do not expect that this is a previously unknown result but were unable to find a citation in the literature.

Lemma 8. *Let $X_1, \ldots, X_n$ be independent but not necessarily identically distributed random variables, where X_i has continuous p.d.f. and c.d.f. f_i, F_i, respectively. Let $X_{(k)}$ denote their kth order statistic, and $\overline{f}_n = \frac{1}{n}\sum_{i=1}^n f_i$, $\overline{F}_n = \frac{1}{n}\sum_{i=1}^n F_i$. Suppose that the*

following conditions hold:

(i) $\lim_{n\to\infty} k/n \in (0,1)$*;*

(ii) There exists a convergent sequence $(t_n)_n$ *such that* $|\overline{F}_n(t_n) - k/n| = o(n^{-1/2})$*;*

(iii) With $\sigma_n^2 = \frac{1}{n}\sum_{i=1}^n F_i(t_n)(1-F_i(t_n))$*, the limit* $\lim_{n\to\infty}\sigma_n^2$ *exists and is positive;*

(iv) $\lim_{n\to\infty}\overline{f}_n(t_n) \in (0,\infty)$.

Then

$$\overline{f}_n(t_n)\sqrt{n}\frac{X_{(k)} - t_n}{\sigma_n}$$

converges to the standard normal in distribution.

Remark. An important special case of the above lemma is when $F_i(t) = F(t-Y_i)$, where F is a continuous and invertible c.d.f. and $(Y_n)_n$ are i.i.d. random variables. In this case, $\overline{F}_n$ is invertible so we may find a sequence $(t_n)_n$ such that $\overline{F}_n(t_n) = k/n$, which by continuity must converge to the unique t which satisfies $\mathbb{E}F(t-Y_1) = \lambda$, so condition (ii) is satisfied. On the other hand the strong law of large numbers shows that conditions (iii) and (iv) hold almost surely.

Proof of Lemma 8. Observe that for any $t \in \mathbb{R}$,

$$\Pr(X_{(k)} \le t) = \Pr\left(\sum_{i=1}^n \mathbb{1}_{\{X_i \le t\}} \ge k\right)$$

Each random variable $\mathbb{1}_{\{X_i\le t\}}$ is independent, and moreover their first three moments are

$$\mathbb{E}\mathbb{1}_{\{X_i\le t\}} = F_i(t)$$
$$\mathbb{E}(\mathbb{1}_{\{X_i\le t\}} - F_i(t))^2 = F_i(t)(1-F_i(t))$$
$$\mathbb{E}(\mathbb{1}_{\{X_i\le t\}} - \Pr(X_i \le t))^3 \le F_i(t)(1-F_i(t))$$

Now, fix $s \in \mathbb{R}$. By Taylor's theorem, there exists x between t_n and $t_n + s/\sqrt{n}$ such that

$$F_i\left(t_n + \frac{s}{\sqrt{n}}\right) = F_i(t_n) + f_i(x)\frac{s}{\sqrt{n}} = F_i(t_n) + f_i(t_n)O(n^{-1/2})$$

since s is constant and f_i is continuous. So,

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n} F_i(t_n + \frac{s}{\sqrt{n}})(1 - F_i(t_n + \frac{s}{\sqrt{n}})) &= \frac{1}{n}\sum_{i=1}^{n}\left(F_i(t_n)(1 - F_i(t_n)) + O(n^{-1/2})f_i(t_n)\right) \\ &= \sigma_n^2 + O(n^{-1/2})\end{aligned}$$

as $\sum_{i=1}^{n} f_i(t_n) = O(n)$ by assumption. Then

$$\frac{\sum_{i=1}^{n}\mathbb{E}(\mathbb{1}_{\{X_i \le t\}} - F_i(t))^3}{\left(\sum_{i=1}^{n}\mathbb{E}(\mathbb{1}_{\{X_i \le t\}} - F_i(t))^2\right)^{3/2}} \le \frac{\sum_{i=1}^{n} F_i(t)(1 - F_i(t))}{\left(\sum_{i=1}^{n} F_i(t)(1 - F_i(t))\right)^{3/2}} = \frac{1}{\sigma_n\sqrt{n}} \to 0$$

Hence Lyapunov's CLT condition is satisfied, and so the random variable

$$S_t = \frac{\sum_{i=1}^{n}\left(\mathbb{1}_{\{X_i \le t\}} - F_i(t)\right)}{\sigma_n\sqrt{n}}$$

converges to the standard normal in distribution for $t = t_n + s/\sqrt{n}$. Now applying the second order version of Taylor's theorem,

$$F_i(t_n + s/\sqrt{n}) = F_i(t_n) + f_i(t)\frac{s}{\sqrt{n}} + O(n^{-1}).$$

which gives

$$\frac{\sum_{i=1}^{n} F_i(t_n + \frac{s}{\sqrt{n}}) - k}{\sqrt{n}} = \sqrt{n}\left(\frac{1}{n}\sum_{i=1}^{n}(F_i(t_n) + f_i(t_n)\frac{s}{\sqrt{n}} + O(n^{-1})) - \frac{k}{n}\right) = s\cdot\overline{f}_n(t_n) + o(1)$$

as by assumption

$$|\overline{F}_n(t_n) - k/n| = o(n^{-1/2}).$$

Then we have

$$\frac{k - \sum_{i=1}^{n} F_i(t_n + \frac{s}{\sqrt{n}})}{\sqrt{\sum_{i=1}^{n} F_i(t_n + \frac{s}{\sqrt{n}})(1 - F_i(t_n + \frac{s}{\sqrt{n}}))}} = -\frac{s}{\sigma_n}\overline{f}_n(t_n) + o(1).$$

Combining all of the above, with $Z \sim \mathcal{N}(0,1)$, we may conclude that

$$\Pr\left(\sqrt{n}\bar{f}_n(t_n)\frac{X_{(k)} - t_n}{\sigma_n} \leq s\right) = \Pr(S_t \geq -s + o(1)) \to \Pr(Z \leq s)$$

As this holds for all $s \in \mathbb{R}$, the claim follows. □

2.6.2 Automata and Turing Machines

The basic constructs of computer science are models of computation; in complete generality an arbitrary time-indexed dynamical system could be a model of computation, but the most typical classes (and important to this thesis) are discrete time, have a discrete state space, and the dependence of the dynamics on the current state is restricted somehow. A model of this type can easily be conceived of as a machine or automaton operating in time, which maintains a state and reads a new symbol of input at each step. After reading the current symbol, it computes a new state (which may depend on both the current state and the current symbol), possibly outputs something, and moves to the next input symbol. The first and most basic example is the finite state machine (FSM), which describes algorithms which use an uniformly bounded amount of memory, no matter what their input is. In the machine metaphor, these are the machines whose state space is not only discrete but finite.

Definition 9 (Finite State Machines)**.** *A finite state acceptor $M = (\Sigma, Q, \delta, q_0, q_A)$ is a tuple of a finite alphabet Σ, a finite state space Q, a transition function $\delta\colon \Sigma \times Q \to Q$, and initial and accepting states $q_0, q_A \in Q$, respectively. Given a string $s = \sigma_1 \ldots \sigma_n \in \Sigma^*$, we say that M accepts s if*

$$\delta(\sigma_n, \delta(\sigma_{n-1}, \delta(\ldots \delta(\sigma_1, q_0)))) = q_A$$

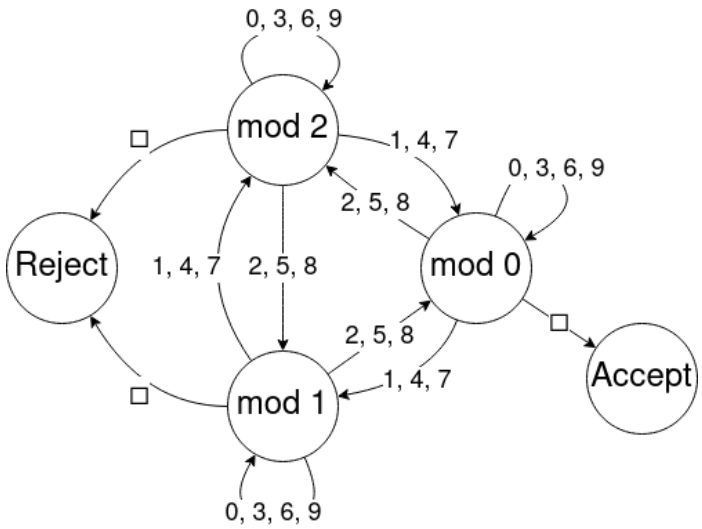


Figure 2.1: Caption

and rejects otherwise. A finite-state transducer is a FSA also equipped with a function $\gamma : \Sigma \times Q \to \Sigma$ *which determines an output symbol depending on the current input symbol and state.*

Note that the transition function δ is recursively applied to the result of the previous transition; it is this that imposes the structure of time on the FSM.

A *language* on the alphabet Σ is a subset of Σ^*; we say that a language is recognized by an automaton if that automaton "accepts" (enters an accepting state) when given exactly the strings of that language as input. The class of languages recognized by finite state machines is called the regular languages, and contains some interesting languages. For example, numbers divisible by 3 written in base-10 can be recognized by a FSM with only three states (see Figure 2.1), one of each of $\{0, 1, 2\}$. The transition function maps $\delta(d, q)$ to $d + q \pmod 3$.

As a model of computation, FSMs are highly resource-limited: The time they can devote to a calculation is precisely the length of their input, and the space they can use is constant (just enough to keep track of the current state). For this reason, the key measure of the complexity of a FSM is its size (the number of states). Moreover, they clearly do not capture the sorts of computations humans can do; every elementary school student can

readily be taught to perform on the order of d^2 operations to multiply two d-digit numbers, which can be written with a mere $2d$ symbols. As it turns out, the language which consists of all multiplication equations (e.g. the base-10 representation of a, followed by the base-10 representation of b, followed by the base-10 representation of ab, for all $a, b \in \mathbb{N}$) is not regular – no FSM can recognize it. a simpler language which is not regular is the strings of the form $a^n b^n$, for $n \geq 0$. A proof of this fact is beyond the scope of this discussion, but intuitively one can note that whatever state the machine is in after seeing the final a must encode n; since a FSM has only finitely many states, it cannot do this for infinitely many n.

So, a clear way to make the FSM more powerful would be to give it more memory. The first way to do so is the push-down automaton (PDA). A PDA is essentially a FSM augmented with a *stack*: an arbitrarily large list of symbols, which the automaton can interact with by reading or deleting ("popping") the current top symbol, or adding ("pushing") a new top symbol. The change of state and interaction with the stack can then depend not only on the current state and input symbol but also the top symbol of the stack.

Definition 10 (Push-Down Automaton)**.** *A push-down automaton (PDA)* $M = (\Sigma, Q, \Gamma, \delta, q_0, q_A, \varepsilon)$ *is a tuple of an input alphabet* Σ*, a state space* Q*, a stack alphabet* Γ*, a transition function* $\delta \colon \Sigma \times Q \times \Gamma \to Q \times \Gamma \cup \{POP, NOP\}$*, start and accepting states* q_0, q_A*, and empty stack symbol* $\varepsilon \in \Gamma$*. At time* t *let* $d(t)$ *denote the depth of the stack, with symbols* $\gamma_1(t), \ldots, \gamma_{d(t)}(t), \varepsilon$ *written on it, and suppose the PDA is in state* q *reading symbol* σ *from the input. It then computes* $\delta(\sigma, q, \gamma_1(t)) = (q', \gamma)$*, and the state on* $t+1$ *becomes* q'*. The stack update is more complex; if* $\gamma \in \Gamma$*, the operation is a push and we have* $d(t+1) = d(t) + 1$ *while* $\gamma_1(t+1) = \gamma$ *and* $\gamma_{i+1}(t+1) = \gamma_i(t)$ *for all* $1 \leq i \leq d(t)$*. If* $\gamma = POP$*, the operation is a pop and we have* $d(t+1) = d(t) - 1$ *and* $\gamma_i(t+1) = \gamma_{i+1}(t)$ *for all* $1 \leq i \leq d(t) - 1$*. If* $\gamma = NOP$*, no action is taken on the stack and we have* $d(t+1) = d(t)$ *and* $\gamma_i(t+1) = \gamma_i$ *for all* $1 \leq i \leq d(t)$*. Accordingly, we require that a pop is never performed when the top symbol is* ε*, i.e. the stack is empty, since there is nothing*

to remove.

The language $a^n b^n$ *can* be recognized by a PDA; it simply must push a symbol onto the stack for every consecutive a, pop a symbol for every consecutive b after that, and reject if it reaches the empty stack before the end of the input or if it sees any other order of symbols (a b followed by an a). The class of languages recognized by PDAs as we have defined them are the *deterministic context-free* languages. The reasons for this name are beyond the current discussion; it suffices for now to note that there are languages which seem to be recognizable by humans, and thus captured by computation, that are not context-free. In particular, the language of all strings of the form $a^n b^n c^n$ is not deterministic context-free.

So, the final model of computation we will discuss is the Turing machine (TM). A TM may be thought of as a machine which is given its input written on an infinite tape, which it can read from, write to, and move along. During each of these operations it maintains a state, which may influence the choice of operation and the next state. Alternatively, it is equivalent to a PDA which has access to two stacks (effectively the two halves of its tape) and which can continue to compute for as long as required after reading the input, signaling the end of the computation by "halting" instead.

Definition 11 (Turing Machine)**.** *A Turing machine is a tuple* $(\Sigma, Q, \delta, b, q_0, q_A, q_R)$ *where* Σ *is the finite alphabet,* Q *is the finite state space,* $\delta \colon \Sigma \times Q \setminus \{q_A, q_R\} \to \Sigma \times Q \times \{L, R\}$ *is the transition function,* b *is the blank symbol, and* q_0, q_A, q_R *are the initial, accepting, and rejecting states, respectively. The tape of a Turing machine is a sequence of symbols* $(\sigma_i(t))_{i \in \mathbb{Z}}$ *indexed by the integers and by time* $t \geq 0$*;* $\sigma_1(0) \cdots \sigma_n(0)$ *constitute its input and* $\sigma_i(0) = b$ *for all* $i \leq 0$ *and* $i \geq n + 1$*. At time* t*, if the TM is in position* i *and state* q *and evaluates its transition function as* $\delta(\sigma_i(t), q) = (\sigma, q', d)$*, the tape updates as* $\sigma_i(t+1) = \sigma$ *and* $\sigma_j(t + 1) = \sigma_j(t)$ *for all* $j \neq i$*. Meanwhile, the machine updates its state to* q' *and its position to* $i + 1$ *(if* $d = R$*) or* $i - 1$ *(if* $d = L$*). The machine accepts an input if it reaches state* q_A*, and rejects it if it reaches state* q_R*.*

Unlike the FSM or PDA, the TM is universally agreed to formalize what is meant

by "computation" in the fullest generality. Intuitively, a function $f\colon \Sigma^* \to \Sigma^*$ might be considered "computable" if it can be calculated by following precisely a finite set of instructions. A Turing machine formalizes this notion – the finite set of instructions is given by its transition function – so a rigorous definition of a computable function is one which can be realized by a Turing machine. This is obviously a philosophical (rather than mathematical) claim but has been upheld since it was first postulated by Alonzo Church and Alan Turing in the 1930s. Systems which are equivalent in power to Turing machines are called Turing-complete.

Turing was motivated in his definition by the "computers" of his day: actually humans, equipped with practically arbitrary amounts of writing implements and surfaces, and trained to carry out precise sets of instructions. It was only later that electronic computers would be invented, which could carry out these operations far more quickly and accurately than a human. From the very beginning, Turing's theory of computation was a theory of human cognition. Thus, it is important for a computational theory of the brain to ultimately describe a model which is Turing complete.

2.7 Prior Results in NEMO

2.7.1 Assembly Projection

The first major result within NEMO was the *creation of assemblies*: repeated stimulation of a brain area by an arbitrary set of neurons results in a sequence of caps which rapidly converges to a single cap which fires repeatedly, strengthening its internal connections, and thus forming an assembly. Moreover, this assembly can be said to "represent" the stimulus because in the future additional presentation will immediately activate the assembly. A nonrigorous proof of this result was given by Papadimitriou & Vempala [94], relying on Gaussian approximations to the binomial distribution which are invalid for many choices of parameters; we recapitulate a slight generalization of their result here with a rigorous proof.

Theorem 12 (Papadimitriou & Vempala)**.** *Let A be a brain area and S a set of m neurons external to it connected to A. Let $p \geq 3\frac{\log n}{k}$. Suppose that S fires repeatedly, evoking a sequence of caps $A_1, \ldots, A_t, \ldots$. Suppose the plasticity is multiplicative with parameter β, satisfying*

$$\beta \geq \beta^* = .$$

Then for $t \geq t^*$, we have $A_t \subseteq \bigcup_{s=1}^{t^*} A_s$.

Proof. Let μ_t be the fraction of newcomers on step t, so that $\mu_1 = 1$ and $\mu_t = k^{-1}|A_t \setminus \bigcup_{1 \leq s < t} A_s|$ for $t \geq 2$. So, we need to show that $\mu_t = 0$ for all $t > t^*$. Let $X_i(t)$ denote the input to neuron $i \in A$ on round t Let c_t be the threshold to make A_t. In other words, every neuron in A_t has at least c_t input from $S \cup A_{t-1}$, and every neuron not in A_t has less. We note that for a neuron to be a newcomer on round t, it needs to be among the $\mu_t k$ neurons with largest input from $S \cup A_{t-1}$ out of $n - |\bigcup_{s<t} A_s|$ neurons. Let $Z_1, \ldots, Z_n$ be a set of i.i.d. $\mathcal{B}(m+k, p)$ random variables, and note that the $\mu_t k$th largest value out of the X_i's where $i \notin \mu$ □

2.7.2 The Assembly Calculus

With the ability to create assemblies established, Papadimitriou & Vempala proposed a small suite of operations on assemblies, which they called the Assembly Calculus (AC), and effectively functioned as a sort of programming language for NEMO. The variables of this programming language are assemblies and brain areas; operations take assemblies and brain areas and perform operations with them. For example, the command $\mathrm{project}(X, A, Y)$ realizes the projection operation in Theorem 12 by enabling firing in area A, and repeatedly fires the set of neurons X, finally denoting the resulting assembly (the projection of X) as Y. The label of this assembly would then be available for use in future operations. A more complex operation is reciprocal projection, notated as $\mathrm{reciprocal\text{-}project}(X, A, Y)$, which creates a new assembly Y in area A now linked *bidirectionally* with X, which may have been perturbed in the process. More complex still is

merge, $\mathrm{merge}(X, Y, A, Z)$, which creates a new assembly Z in area A linked bidirectionally with both X and Y, again possibly perturbing them in the process.

The culmination of this line of thought was a Turing completeness result, showing that this set of operations was Turing-complete under certain assumptions. This works out to describing a layout of brain areas and a "program" of AC commands which simulate an arbitrary Turing machine (TM), in the sense that the model moves through a series of states which are in one-to-one correspondence with the states of the TM. The key assumptions are about what happens when multiple operations of the AC occur. Notably, Theorem 12 asserts that projection convergence occurs from a tabula rasa but says nothing about whether convergence occurs when another assembly already exists in the brain area. The precise assumptions are the following:

- m-non-interference: m AC operations can occur without any interference. For instance, if assembies $X_1, \ldots, X_m$ have been linked to assemblies $Y_1, \ldots, Y_m$ by projection, then firing X_i causes only Y_i to fire.
- Obliviousness: Consider the sequence of operations

$$\mathrm{reciprocal\text{-}project}(X, A, Y); \quad \mathrm{reciprocal\text{-}project}(X, A, Z); \quad \mathrm{fire}(X).$$

  Obliviousness holds if when $\mathrm{fire}(X)$ occurs, *only* Z (and not Y) fires in response.

We may then state the following theorem:

Theorem 13 (Papadimitriou, Vempala, Mitropolsky, Collins, & Maass)**.** *Under the m-non-interference and obliviousness assumptions, the Assembly Calculus can simulate an arbitrary Turing machine using time T and space m in time $O(T)$.*

The role of the AC in the simulation is store the (complete) state of the Turing machine; that is, both the state of the machine and the complete tape. The logic of the transition function is realized via a program of AC commands, which includes the conditional logic

necessary to implement a finite state machine. Thus, the entire algorithmic aspect of a Turing machine has been pushed into the control commands of the AC, which do not have a ready biological interpretation. A second task which has been considered is rudimentary *language acquisition*: How does the developing brain acquire the representations of words which enable it to make sense of language? Mitropolsky & Papadimitriou devised one possible scheme for acquiring contextualized words, that is, paired linguistic and sensory stimuli. It is assumed that the brain contains some structure common across languages – for example, brain areas for nouns and verbs – but is initially agnostic to language-specific information, such as word order. Hence, it is necessary for the structure of a particular language to be inferred from context. An example sentence of the type they consider is "dogs run", which might be paired with the visual stimulus of a dog, the sound of barking, the motor representation of running, etc. Initially, the model forms spurious representation of both words in both the noun and verb areas. However, by seeing many contextualized sentences, the spurious assemblies are eventually eliminated, and the words are correctly linked to parts of speech.

2.7.3 Language

A separate line of work in NEMO has focused on designing specific algorithms within NEMO by which it can perform the fundamental tasks of language usage. One of these tasks is *parsing*: Given a linguistic utterance, break it down into more basic lexical units (clauses, parts of speech, etc). The first paper to design a parser for a non-trivial subset of natural language was by Mitropolsky, Collins, & Papadimitriou [95]. This paper augmented the basic version of NEMO [96] with state-dependent inhibition and disinhibition of enitre brain areas; this mechanism also plays an important role in the Turing simulation presented in this thesis (see Subsection 4.1.4). Specifically, "words" are encoded as assemblies in a designated lexicon area; the onset and offset of these assemblies are each associated with an "action", a set of (dis)inhibitions of other brains areas associated with

various parts of speech. This means that the firing of a word assembly results in some maintained activity in other areas, a sort of stored state of the parse, which interacts with subsequent words to capture syntactic dependencies. The resulting parser is capable of handling syntactic patterns consisting of a single clause, which is to say those recognizable by a finite state machine. Mitropolsky et al. [97] would then expand this construction to handle patterns which multiple clauses, both dependent and embedded. The basic idea is to use an arbitrarily large working memory to store the portion of the sentence that has been parsed. When an embedded clause is encountered, the parser processes it, and when it is concluded it returns to the beginning of working memory and re-processes the initial portion of the sentence to restore its state. This process can additionally be formalized as an extension of a finite state machine which is provably equivalent (in the sense of which languages it can recognize) to the pushdown automaton.

CHAPTER 3

BARELY-SUPERVISED LEARNING WITH ASSEMBLIES

This chapter presents material from the paper "Assemblies of Neurons Learn to Classify Well-Separated Distributions" [98]. Its results represent collaborative work with the authors of that paper, Christos Papadimitriou and Santosh Vempala.

3.1 Overview

The brain has been a productive source of inspiration for AI, from the perceptron and the neocognitron to deep neural nets. Machine learning has since advanced to dizzying heights of analytical understanding and practical success, but the study of the brain has lagged behind in one important dimension: After half a century of intensive effort by neuroscientists (both computational and experimental), and despite great advances in our understanding of the brain at the level of neurons, synapses, and neural circuits, we still have no plausible mechanism for explaining intelligence, that is, the brain's performance in planning, decision-making, language, etc. As Nobel laureate Richard Axel put it, "we have no logic for translating neural activity into thought and action" [99].

Recently, a high-level computational framework was developed with the explicit goal to fill this gap: the Assembly Calculus (AC) [96], a computational model whose basic data type is the *assembly of neurons*. Assemblies, called "the alphabet of the brain" [100], are large sets of neurons whose simultaneous excitation is tantamount to the subject's thinking of an object, idea, episode, or word (see Piantadosi et al. [101]). Dating back to the birth of neuroscience, the "million-fold democracy" by which groups of neurons act collectively without central control was first proposed by Sherrington [14] and was the empirical phenomenon that Hebb attempted to explain with his theory of plasticity [22]. Assemblies are initially created to record memories of external stimuli [102], and

are believed to be subsequently recalled, copied, altered, and manipulated in the non-sensory brain [103, 104]. The Assembly Calculus provides a repertoire of operations for such manipulation, namely `project`, `reciprocal-project`, `associate`, `pattern-complete`, and `merge` encompassing a complete computational system. Since the Assembly Calculus is, to our knowledge, the only extant computational system whose purpose is to bridge the gap identified by Axel in the above quote [99], it is of great interest to establish that complex cognitive functions can be plausibly expressed in it. Indeed, significance progress has been made over the past year, including a parser of English [95] and planning program in the blocks world [105], both written in the AC programming system. Yet despite these recent advances, one fundamental question is left unanswered: If the Assembly Calculus is a meaningful abstraction of cognition and intelligence, why does it not have a `learn` command? *How can the brain learn through assembly representations?*

This is the question addressed and answered in this paper. As assembly operations are a new learning framework and device, one has to start from the most basic questions: Can this model classify assembly-encoded stimuli that are separated through clustering, or by half spaces? Recall that learning linear thresholds is a theoretical cornerstone of supervised learning, leading to a legion of fundamental algorithms: Perceptron, Winnow, multiplicative weights, isotron, kernels and SVMs, and many variants of gradient descent.

Following Papadimitriou et al. [96], we model the brain as a directed graph of excitatory neurons with dynamic edge weights (due to plasticity). The brain is subdivided into *areas*, for simplicity each containing n neurons connected through a $G_{n,p}$ random directed graph [106]. Certain ordered pairs of areas are also connected, through random bipartite graphs. We assume that neurons fire in discrete time steps. At each time step, each neuron in a brain area will fire if its synaptic input from the firings of the previous step is among the top k highest out of the n neurons in its brain area. This selection process is called k*-cap*, and is an abstraction of the process of *inhibition* in the brain, in which a separate population

of inhibitory neurons is induced to fire by the firing of excitatory neurons in the area, and through negatively-weighted connections prevents all but the most stimulated excitatory neurons from firing. Synaptic weights are altered via Hebbian plasticity and homeostasis (see Section 2 for a full description). In this stylized mathematical model, reasonably consistent with what is known about the brain, it has been shown that the operations of the Assembly Calculus converge and work as specified (with high probability relative to the underlying random graphs). These results have also been replicated by simulations in the model above, and also in more biologically realistic networks of spiking neurons [107, 96]. *In this paper we develop, in the same model, mechanisms for learning to classify well-separated classes of stimuli*, including clustered distributions and linear threshold functions with margin. Moreover, considering that the ability to learn from few examples, and with mild supervision, are crucial characteristics of any brain-like learning algorithm, we show that learning with assemblies does both quite naturally.

The learning mechanism. For the purpose of demonstrating learning within the framework of NEMO, we consider the specific setting described below. First, there is a special area, called the *sensory area,* in which training and testing data are encoded as assembly-like representations called *stimuli*. There is only one other brain area (besides the sensory area), and that is where learning happens, through the formation of assemblies in response to sequences of stimuli.

A *stimulus* is a set of about k neurons firing simultaneously ("presented") in the sensory area. Note that, exceptionally in the sensory area, a number of neurons that is a little different from k may fire at a step. A *stimulus class* A is a distribution over stimuli, defined by three parameters: two scalars $r, q \in [0, 1], r > q$, and a set of k neurons S_A in the sensory area. To generate a stimulus $x \in \{0, 1\}^n$ in the class A, each neuron $i \in S_A$ is chosen with probability r, while for each $i \notin S_A$, the probability of choosing neuron i is qk/n. It follows immediately that, in expectation, an r fraction of the neurons in the

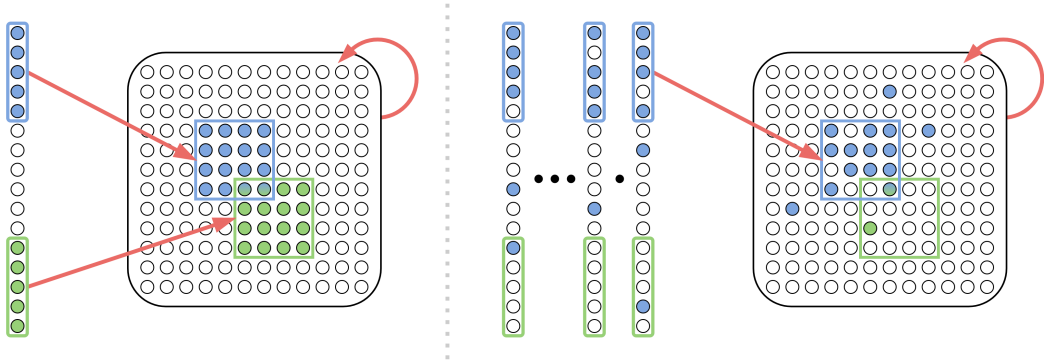


Figure 3.1: A mathematical model of learning in the brain. Our model (left) has a sensory area (column) connected to a brain area (square), both made up of spiking neurons. Two different stimuli classes (with their core sets in blue/green on the left) project from the sensory area via synaptic connections (arrows). Assemblies in the brain area (with core sets in the corresponding colors) form in response to these stimuli classes, each of which consistently fires when a constant fraction of the associated stimuli class's core set does. Our learning algorithm (right) consists of presenting a stream of stimuli from each class.

stimulus core are set to 1 and the number of neurons outside the core that are set to 1 is also $O(k)$.

The presentation of a sequence of stimuli from a class A in the sensory area evokes in the learning system a *response* R, a *distribution over assemblies* in the brain area. We show that, as a consequence of plasticity and k-cap, this distribution R will be highly concentrated, in the following sense: Consider the set S_R of all assemblies x that have positive probability in R. Then the numbers of neurons in both the intersection $R^* = \bigcap_{x \in S_R} x$, called the *core* of R and the union $\bar{R} = \bigcup_{x \in S_R} x$ are close to k, in particular $k - o(k)$ and $k + o(k)$ respectively.[1] In other words, neurons in R^* fire far more often on average than neurons in $\bar{R} \setminus R^*$.

Finally, our learning protocol is this: Beginning with the brain area at rest, stimuli are repeatedly sampled from the class, and made to fire. After a small number of training samples, the brain area returns to rest, and then the same procedure is repeated for the next stimulus class, and so on. Then testing stimuli are presented in random order to test the extent of learning. (see Algorithm 1 in an AC-derived programming language.)

[1]The larger the plasticity, the closer these two values are (see Papadimitriou & Vempala [94], Fig. 2).

Input: a set of stimulus classes $A_1, \ldots, A_c$; $T \geq 1$
Output: A set of assemblies $y_1, \ldots, y_c$ in the brain area encoding these classes
foreach *stimulus class* i **do**
    $\text{inh}(B) \leftarrow 0$;
    foreach *time step* $1 \leq t \leq T$ **do**
        Sample $x \sim A_i$ and fire x;
    end
    $y_i \leftarrow \texttt{read}(B)$;
    $\text{inh}(B) \leftarrow 1$;
end

Algorithm 1: The learning mechanism. (B denotes the brain area.)

That is, we sample T stimuli x from each class, fire each x to cause synaptic input in the brain area, and after the Tth sample has fired we record the assembly which has been formed in the brain area. This is the representation for this class.

Related work

There are numerous learning models in the neuroscience literature. In a variation of the model we consider here, Rangamani & Gandhi [108] have considered supervised learning of Boolean functions using assemblies of neurons, by setting up separate brain areas for each label value. Amongst other systems with rigorous guarantees, assemblies are superficially similar to the "items" of Valiant's neuroidal model [109], in which supervised learning experiments have been conducted [110, 111], where an output neuron is clamped to the correct label value, while the network weights are updated under the model. The neuroidal model is considerably more powerful than ours, allowing for arbitrary state changes of neurons and synapses; in contrast, our assemblies rely on only two biologically sound mechanisms, plasticity and inhibition.

Hopfield nets [76] are recurrent networks of neurons with symmetric connection weights which will converge to a memorized state from a sufficiently similar one, when properly trained using a local and incremental update rule. In contrast, the memorized states our model produces (which we call assemblies) emerge through plasticity and randomization from the structure of a random directed network, whose weights are asymmetric and non-

negative, and in which inhibition — not the sign of total input — selects which neurons will fire.

Stronger learning mechanisms have recently been proposed. Inspired by the success of deep learning, a large body of work has shown that cleverly laid-out microcircuits of neurons can approximate backpropagation to perform gradient descent [112, 113, 114, 115, 116, 90]. These models rely crucially on novel types of neural circuits which, although biologically possible, are not presently known or hypothesized in neurobiology, nor are they proposed as a theory of the way the brain works. These models are capable of matching the performance of deep networks on many tasks, which are more complex than the simple, classical learning problems we consider here. The difference between this work and ours is, again, that here we are showing that learning arises naturally from well-understood mechanisms in the brain, in the context of the assembly calculus.

3.2 Results

Very few stimuli sampled from an input distribution are activated sequentially at the sensory area. The only form of supervision required is that all training samples from a given class are presented consecutively. Plasticity and inhibition alone ensure that, in response to this activation, an assembly will be formed for each class, and that this same assembly will be recalled at testing upon presentation of other samples from the same distribution. In other words, learning happens. And in fact, despite all these limitations, we show that the device is an efficient learner of interesting concept classes.

Our first theorem is about the creation of an assembly in response to inputs from a stimulus class. This is a generalization of a theorem from Papadimitriou & Vempala [94], where the input stimulus was held constant; here the input is a stream of random samples from the same stimulus class. Like all our results, it is a statement holding with high probability (with high probability, meaning probability tending to 1 as some parameter tends to infinity (w.h.p.)), where the underlying random event is the random graph and the

random samples. When sampled stimuli fire, the assembly in the brain area changes. The neurons participating in the current assembly (those whose synaptic input from the previous step is among the k highest) are called the current *winners.* A *first-time winner* is a current winner that participated in no previous assembly (for the current stimulus class).

Theorem 14 (Creation)**.** *Consider a stimulus class A projected to a brain area. Assume that*

$$\beta \geq \beta_0 = \frac{1}{r^2} \frac{\left(\sqrt{2} - r^2\right) \sqrt{2 \ln\left(\frac{n}{k}\right)} + \sqrt{6}}{\sqrt{kp} + \sqrt{2 \ln\left(\frac{n}{k}\right)}}$$

Then w.h.p. no first-time winners will enter the cap after $O(\log k)$ rounds, and moreover the total number of winners $\bar{A}$ can be bounded as

$$|\bar{A}| \leq \frac{k}{1 - \exp(-(\frac{\beta}{\beta_0})^2)} \leq k + O\left(\frac{\log n}{r^3 p \beta^2}\right)$$

Remark. The theorem implies that for a small constant c, it suffices to have plasticity parameter

$$\beta \geq \frac{1}{r^2} \frac{c}{\sqrt{kp/(2\ln(n/k))} + 1}.$$

Our second theorem is about *recall* for a single assembly, when a new stimulus from the same class is presented. We assume that examples from an an assembly class A have been presented, and a response assembly A^* encoding this class has been created, by the previous theorem.

Theorem 15 (Recall)**.** *w.h.p. over the stimulus class, the set C_1 firing in response to a test assembly from the class A will overlap A^* by a fraction of at least $1 - e^{-kpr}$, i.e.*

$$\frac{|C_1 \setminus A^*|}{k} \leq e^{-kpr}$$

The proof entails showing that the average weight of incoming connections to a neuron in

A^* from neurons in S_A is at least

$$1 + \frac{1}{\sqrt{r}} \left(\sqrt{2} + \sqrt{\frac{2}{kpr} \ln \left(\frac{n}{k} \right) + 2} \right)$$

Our third theorem is about the creation of a second assembly corresponding to a second stimulus class. This can easily be extended to many classes and assemblies. As in the previous theorem, we assume that $O(\log k)$ examples from assembly class A have been presented, and $\bar{A}$ has been created. Then we introduce B, a second stimulus class, with $|S_A \cap S_B| = \alpha k$, and present $O(\log k)$ samples to induce a series of caps, $B_1, B_2, \ldots$, with B^* as their union.

Theorem 16 (Multiple Assemblies)**.** *The total support of B^* can be bounded w.h.p. as*

$$|B^*| \leq \frac{k}{1 - \exp(-(\frac{\beta}{\beta_0})^2)} \leq k + O\left(\frac{\log n}{r^3 p \beta^2} \right)$$

Moreover, w.h.p., the overlap in the core sets A^* and B^* will preserve the overlap of the stimulus classes, so that $|A^* \cap B^*| \leq \alpha k$.

This time the proof relies on the fact that the average weight of incoming connections to a neuron in A^* is *upper-bounded* by

$$\gamma \leq 1 + \frac{\sqrt{2 \ln \left(\frac{n}{k} \right)} - \sqrt{2 \ln((1+r)/r\alpha)}}{\alpha r \sqrt{kp}}$$

Our fourth theorem is about classification after the creation of multiple assemblies, and shows that random stimuli from any class are mapped to their corresponding assembly. We state it here for two stimuli classes, but again it is extended to several. We assume that stimulus classes A and B overlap in their core sets by a fraction of α, and that they have been projected to form a distribution of assemblies A^* and B^*, respectively.

Theorem 17 (Classification)**.** *If a random stimulus chosen from a particular class (WLOG,*

say B) fires to cause a set C_1 of learning area neurons to fire, then w.h.p. over the stimulus class the fraction of neurons in the cap C_1 and in B^* will be at least

$$\frac{|C_1 \cap B^*|}{k} \geq 1 - 2\exp\left(-\frac{1}{2}(\gamma\alpha - 1)^2 kpr\right)$$

where γ is a lower bound on the average weight of incoming connections to a neuron in A^* (resp. B^*) from neurons in S_A (resp. S_B).

Taken together, the above results guarantee that this mechanism can learn to classify well-separated distributions, where each distribution has a constant fraction of its nonzero coordinates in a subset of k input coordinates. The process is *naturally interpretable:* an assembly is created for each distribution, so that random stimuli are mapped to their corresponding assemblies, and the assemblies for different distributions overlap in no more than the core subsets of their corresponding distributions.

Finally, we consider the setting where the labeling function is a linear threshold function, parameterized by an arbitrary nonnegative vector v and margin Δ. We will create a single assembly to represent examples on one side of the threshold, i.e. those for which $v \cdot X \geq \|v\|_1 k/n$. We define $\mathcal{D}_+$ denote the distribution of these examples, where each coordinate is an independent Bernoulli variable with mean $\mathbb{E}(X_i) = k/n + \Delta v_i$, and define $\mathcal{D}_-$ to be the distribution of negative examples, where each coordinate is again an independent Bernoulli variable yet now all identically distributed with mean k/n. (Note that the support of the positive and negative distributions is the same; there is a small probability of drawing a positive example from the negative distribution, or vice versa.) To serve as a classifier, a fraction $1 - \epsilon_+$ of neurons in the assembly must be guaranteed to fire for a positive example, and a fraction $\epsilon_- < 1 - \epsilon_+$ guaranteed *not* to fire for a negative one. A test example is then classified as positive if at least a $1 - \epsilon$ fraction of neurons in the assembly fire (for $\epsilon \in [\epsilon_-, 1 - \epsilon_+]$), and negative otherwise. The last theorem shows that this can in fact be done with high probability, as long as the normal vector v of the linear

threshold is neither too dense nor too sparse. Additionally, we assume synapses are subject to homeostasis in between training and evaluation; that is, all of the incoming weights to a neuron are normalized to sum to 1.

Theorem 18 (Learning Linear Thresholds)**.** *Let v be a nonnegative vector normalized to be of unit Euclidean length ($\|v\|_2 = 1$). Assume that $\Omega(k) = \|v\|_1 \leq \sqrt{n}/2$ and*

$$\Delta^2\beta \geq \sqrt{\frac{2k}{p}}(\sqrt{2\ln(n/k)+2)}+1).$$

Then, sequentially presenting $\Omega(\log k)$ samples drawn at random from $\mathcal{D}^+$ forms an assembly A^* that correctly separates D^+ from D^-: with probability $1-o(1)$ a randomly drawn example from $\mathcal{D}^+$ will result in a cap which overlaps at least $3k/4$ neurons in A^*, and an example from $\mathcal{D}^-$ will create a cap which overlaps no more than $k/4$ neurons in A^*.

Remark. The bound on $\Delta^2\beta$ leads to two regimes of particular interest: In the first,

$$\beta \geq \frac{\sqrt{2\ln(n/k)+2}+1}{\sqrt{kp}}$$

and $\Delta \geq \sqrt{k}$, which is similar to the plasticity parameter required for a fixed stimulus [94] or stimulus classes; in the second, β is a constant, and

$$\Delta \geq \left(\frac{2k}{\beta^2 p}\right)^{1/4}\left(\sqrt{2\ln(n/k)+2}+1\right)^{1/2}.$$

Remark. We can ensure that the number of neurons outside of A^* for a positive example or in A^* for a negative example are both $o(k)$ with small overhead[2], so that plasticity can be active during the classification phase.

Since our focus in this paper is on highlighting the brain-like aspects of this learning mechanism, we emphasize stimulus classes as a case of particular interest, as they are a probabilistic generalization of the single stimuli considered in Papadimitriou & Vempala

[2]i.e. increasing the plasticity constant β by a factor of $1+o(1)$

[94]. Linear threshold functions are an equally natural way to generalize a single k-sparse stimulus, say v; all the 0/1 points on the positive side of the threshold $v^\top x \geq \alpha k$ have at least an α fraction of the k neurons of the stimulus active.

Finally, reading the output of the device by the Assembly Calculus is simple: Add a *readout area* to the two areas so far (stimulus and learning), and project to this area one of the assemblies formed in the learning area for each stimulus class. The assembly in the learning area that fires in response to a test sample will cause the assembly in the readout area corresponding to the class to fire, and this can be sensed through the `readout` operation of the AC.

Proof overview. The proofs of all five theorems can be found in the Appendix. The proofs hinge on showing that large numbers of certain neurons of interest will be included in the cap on a particular round — or excluded from it. More specifically:

- To create an assembly, the sequence of caps should converge to the assembly's core set. In other words, w.h.p. an increasing fraction of the neurons selected by the cap in a particular step will also be selected at the next one.
- For recall, a large fraction of the assembly should fire (i.e. be included in the cap) when presented with an example from the class.
- To differentiate stimuli (i.e. classify), we need to ensure that a large fraction of the correct assembly will fire, while no more than a small fraction of the other assemblies do.

Following Papadimitriou & Vempala [94], we observe that if the probability of a neuron having input at least t is no more than ϵ, then no more than an ϵ fraction of the cohort of neurons will have input exceeding t (with constant probability). By approximating the total input to a neuron as Gaussian and using well-known bounds on Gaussian tail probabilities, we can solve for t, which gives an explicit input threshold neurons must surpass to make

a particular cap. Then, we argue that the advantage conferred by plasticity, combined with the similarity of examples from the same class, gives the neurons of interest enough of an advantage that the input to all but a small constant fraction will exceed the threshold.

3.3 Experiments

The learning algorithm has been run on both synthetic and real-world datasets, as illustrated in the figures below. Code for experiments is available at https://github.com/mdabagia/learning-with-assemblies.

Beyond the basic method of presenting a few examples from the same class and allowing plasticity to alter synaptic weights, the training procedure is slightly different for each of the concept classes (stimulus classes, linearly-separated, and MNIST digits). In each case, we renormalize the incoming weights of each neuron to sum to one after concluding the presentation of each class, and classification is performed on top of the learned assemblies by predicting the class corresponding to the assembly with the most neurons on.

- For stimulus classes, we estimate the assembly for each class as composed of the neurons which fired in response to the last training example, which in practice are the same as those most likely to fire for a random test example.
- For a linear threshold, we only present positive examples, and thus only form an assembly for one class. As with stimulus classes, the neurons in the assembly can be estimated by the last training cap or by averaging over test examples. We classify by comparing against a fixed threshold, generally half the cap size.

Additionally, it is important to set the plasticity parameter (β) large enough that assemblies are reliably formed. We had success with $\beta = 0.1$ for stimulus classes and $\beta = 1.0$ for linear thresholds.

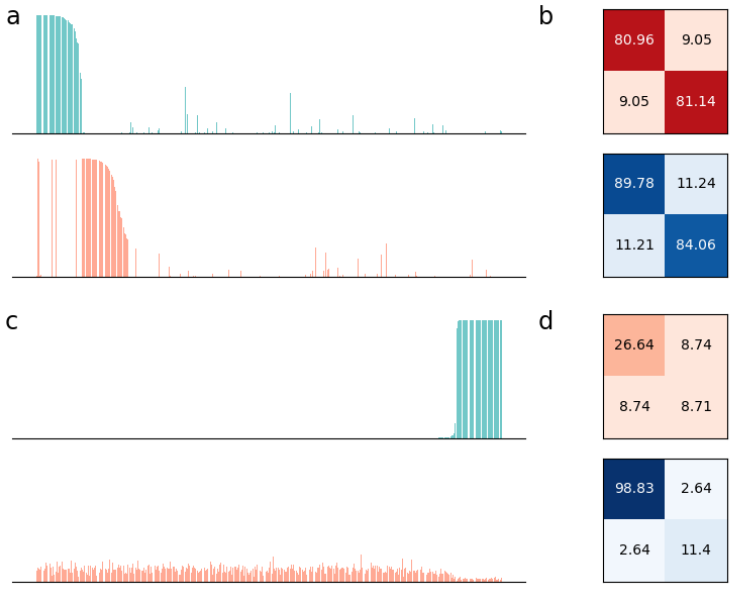


Figure 3.2: Assemblies learned for various concept classes. On the top two lines, we show assemblies learned for stimulus classes, and on the bottom two lines, for a linear threshold with margin. In (a) & (c) we exhibit the distribution of firing probabilities over neurons of the learning area. In (b) & (d) we show the average overlap of different input samples (red square) and the overlaps of the corresponding representations in the assemblies (blue square). Using a simple sum readout over assembly neurons, both stimulus classes and linear thresholds are classified with 100% accuracy. Here, $n = 10^3, k = 10^2, p = 0.1, r = 0.9, q = 0.1, \Delta = 1.0$, with 5 samples per class, and $\beta = 0.01$ (stimulus classes) and $\beta = 1.0$ (linear threshold).

In Figure 3.2 (a) & (b), we demonstrate learning of two stimulus classes, while in Figure 3.2 (c) & (d), we demonstrate the result of learning a well-separated linear threshold function with assemblies. Both had perfect accuracy. Additionally, assemblies readily generalize to a larger number of classes (see Figure 3.6 in the appendix). We also recorded sharp threshold transitions in classification performance as the key parameters of the model are varied (see Figures 3.3 & 3.4).

There are a number of possible extensions to the simplest strategy, where within a single brain region we learn an assembly for each concept class and classify based on which assembly is most activated in response to an example. We compared the performance of various classification models on MNIST as the number of features increases. The high-level model is to extract a certain number of features using one of the five different methods,

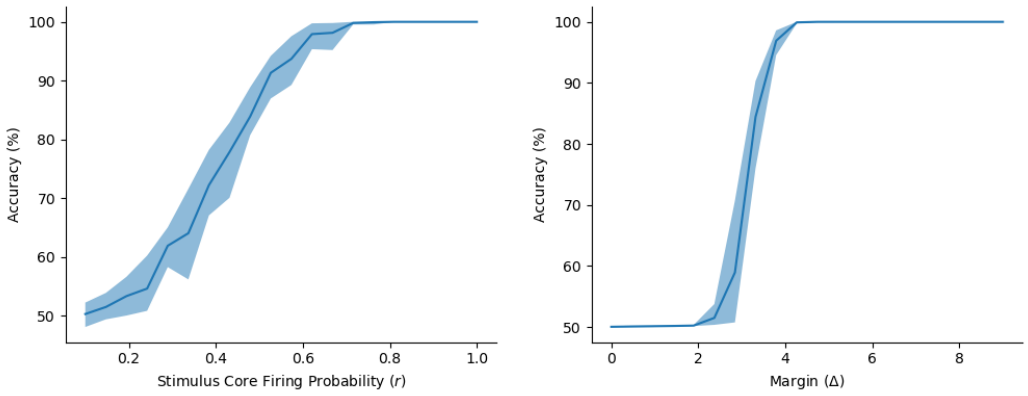


Figure 3.3: Mean (dark line) and range (shaded area) of classification accuracy for two stimulus classes (left) and a fixed linear threshold (right) over 20 trials, as the classes become more separable. For stimulus classes, we vary the firing probability of neurons in the stimulus core while fixing the probability for the rest at k/n, while for the linear threshold, we vary the margin. For both we used 5 training examples with $n = 1000, k = 100, p = 0.1$, and $\beta = 0.1$ (stimulus classes), $\beta = 1.0$ (linear threshold).

and then find the best linear classifier (of the training data) on these features to measure performance (on the test data). The five different feature extractors are:

- Linear features. Each feature's weights are sampled i.i.d. from a Gaussian with standard deviation 0.1.

- Nonlinear features. Each feature is a binary neuron: it has 784 i.i.d. Bernoulli(0.2) weights, and 'fires' (has output 1, otherwise 0) if its total input exceeds the expected input (70×0.2).

- Large area assembly features. In a single brain area of size m with cap size $m/10$, we attempt to form an assembly for each class. The area sees a sequence of 5 examples from each class, with homeostasis applied after each class. Weights are updated according to Hebbian plasticity with $\beta = 1.0$. Additionally, we apply a negative bias: A neuron which has fired for a given class is heavily penalized against firing for subsequent classes.

- 'Random' assembly features. For a total of m features, we create $m/100$ different

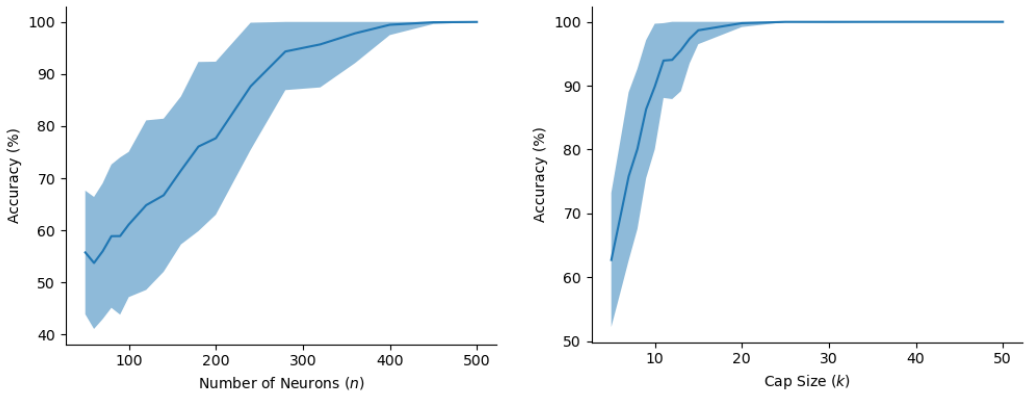


Figure 3.4: Mean (dark line) and range (shaded area) of classification accuracy of two stimulus classes for various values of the number of neurons (n, left) and the cap size (k, right). For variable n, we let $k = n/10$; for variable k, we fix $n = 1000$. Other parameters are fixed, as $p = 0.1, r = 0.9, q = k/n$, and $\beta = 0.1$.

areas of 100 neurons each, with cap size 10. We then repeat the large area training procedure above in each area, with the order of the presentation of classes randomized for each area.

- 'Split' assembly features: For a total of m features, we create 10 different areas of $m/10$ neurons each, with cap size $m/100$. Area i sees a sequence of 5 examples from class i. Weights are updated according to Hebbian plasticity, and homeostasis is applied after training.

After extracting features, we train the linear classification layer to minimize cross-entropy loss on the standard MNIST training set (60000 images) and finally test on the full test set (10000 images).

The results as the total number of features ranges from 1000 to 10000 is shown in Fig. 3.5. 'Split' assembly features are ultimately the best of the five, with 'split' features achieving 96% accuracy with 10000 features. However, nonlinear features outperform 'split' and large-area features and match 'random' assembly features when the number of features is less than 8000. For reference, the linear classifier gets to 89%, while a two-layer neural network with width 800 trained end-to-end gets to 98.4%.

Going further, one could even create a hierarchy of brain areas, so that the areas in the first "layer" all project to a higher-level area, in hopes of forming assemblies for each digit in the higher-level area which are more robust. In this paper, our goal was to highlight the potential to form useful representations of a classification dataset using assemblies, and so we concentrated on a single layer of brain areas with a very simple classification layer on top. It will be interesting to explore what is possible with more complex architectures.

3.4 Discussion

Assemblies are widely believed to be involved in cognitive phenomena, and the AC provides evidence of their computational aptitude. Here we have made the first steps towards understanding how *learning* can happen in assemblies. Normally, an assembly is associated with a stimulus, such as Grandma. We have shown that this can be extended to *a distribution over stimuli.* Furthermore, for a wide range of model parameters, distinct assemblies can be formed for multiple stimulus classes in a single brain area, so long as the

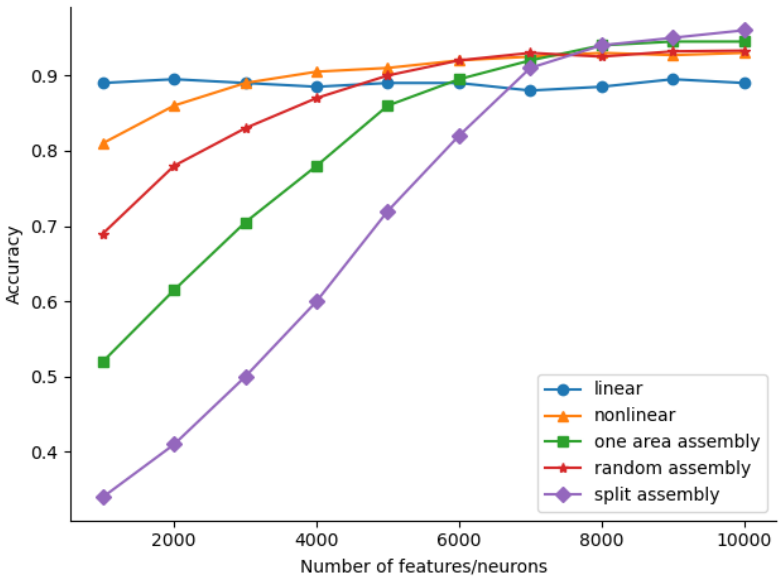


Figure 3.5: MNIST test accuracy as the number of features increases, for various classification models. 'Split' assembly features, which forms an assembly for class i in area i, achieves the highest accuracy with the largest number of features.

classes are reasonably differentiated.

A model of the brain at this level of abstraction should allow for the kind of classification that the brain does effortlessly — e.g., the mechanism that enables us to understand that individual frames in a video of an object depict the same object. With this in mind, the learning algorithm we present is remarkably parsimonious: it generalizes from a handful of examples which are seen only once, and requires no outside control or supervision other than ensuring multiple samples from the same concept class are presented in succession (and this latter requirement could be relaxed in a more complex architecture which channels stimuli from different classes). Finally, even though our results are framed within the Assembly Calculus and the underlying brain model, we note that they have implications far beyond this realm. In particular, they suggest that *any* recurrent neural network, equipped with the mechanisms of plasticity and inhibition, will naturally form an assembly-like group of neurons to represent similar patterns of stimuli.

But of course, many questions remain. In this first step we considered a single brain area — whereas it is known that assemblies draw their computational power from the interaction, through the AC, among many areas. We believe that a more general architecture encompassing a hierarchy of interconnected brain areas, where the assemblies in one area act like stimulus classes for others, can succeed in learning more complex tasks — and even within a single brain area improvements can result from optimizing the various parameters, something that we have not tried yet.

In another direction, here we only considered Hebbian plasticity, the simplest and most well-understood mechanism for synaptic changes. Evidence is mounting in experimental neuroscience that the range of plasticity mechanisms is far more diverse [117], and in fact it has been demonstrated recently [118] that more complex rules are sufficient to learn harder tasks. Which plasticity rules make learning by assemblies more powerful?

We showed that assemblies can learn nonnegative linear threshold functions with sufficiently large margins. Experimental results suggest that the requirement of nonnegativity is

a limitation of our proof technique, as empirically assemblies readily learn arbitrary linear threshold functions (with margin). What other concept classes can assemblies provably learn? We know from support vector machines that linear threshold functions can be the basis of far more sophisticated learning when their input is pre-processed in specific ways, while the celebrated results of Rahimi & Recht [119] demonstrated that certain families of random nonlinear features can approximate sophisticated kernels quite well. What would constitute *a kernel* in the context of assemblies? The sensory areas of the cortex (of which the visual cortex is the best studied example) do pre-process sensory inputs extracting features such as edges, colors, and motions. Presumably learning by the non-sensory brain — which is our focus here — operates on the output of such pre-processing. We believe that studying the implementation of kernels in cortex is a very promising direction for discovering powerful learning mechanisms in the brain based on assemblies.

Appendix: Further experimental results

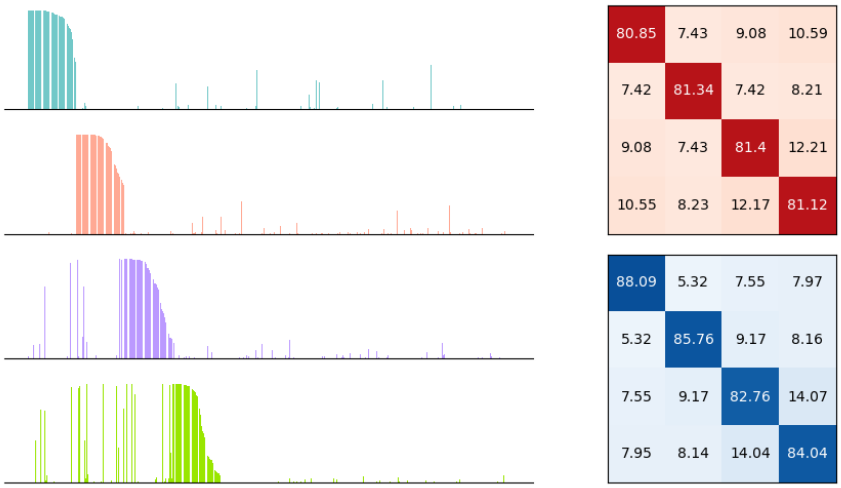


Figure 3.6: Assemblies learned for four stimulus classes. On the left, the distributions of firing probabilities over neurons; on the right, the average overlap of the assemblies. Each additional class overlaps with previous ones, yet a simple readout over assembly neurons allows for perfect classification accuracy. Here, $n = 10^3, k = 10^2, p = 0.1, r = 0.9, q = 0.1, \beta = 0.1$, with 5 samples per class.

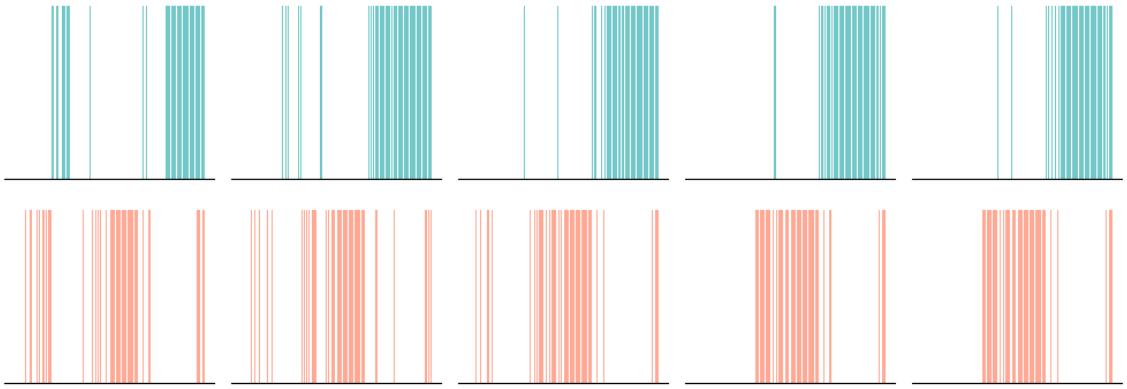


Figure 3.7: Assemblies formed during training. Each row is the response of the neural population to examples from the respective class. At each step, a new example from the appropriate class is presented. Due to plasticity, a core set emerges for the class after only a few rounds. (Here, five rounds are shown.) Inputs are drawn from two stimulus classes, with $n = 10^3, k = 10^2, p = 0.1, r = 0.9, q = 0.1$ and $\beta = 0.1$.

3.5 Proofs

3.5.1 Preliminaries

We will need a few lemmas. The Berry-Esseen theorem (Theorem 3) implies the following:

Lemma 19. *Let $X_1, \ldots, X_{2n}$ denote the weights of the edges incoming to a neuron in the brain area from its neighbors (i.e. $X_i = w_i$ w.p. p, and 0 otherwise). Denote their sum as $S = \sum_i X_i$, and consider the normal random variable*

$$Y \sim \mathcal{N}(\mathbb{E}[S], \mathrm{Var}[S])$$

Then for $p = o(1)$, and $w_i = o(\sqrt{np})$,

$$\sup_{x \in \mathbb{R}} |\Pr(S < t) - \Pr(Y < t)| = o(1)$$

Proof. Let $\tilde{X}_i = X_i - \mathbb{E}[X_i]$. We have

$$\mathbb{E}[\tilde{X}_i^2] = p(1-p)^2 w_i^2 + (1-p)p^2 w_i^2 = p(1-p)w_i^2$$

and

$$\mathbb{E}[|\tilde{X}_i|^3] = p(1-p)^3 w_i^3 + (1-p)p^3 w_i^3 \leq p(1-p)w_i^3$$

Let

$$\tilde{S} = \frac{S - \mathbb{E}[S]}{\sqrt{\mathrm{Var}[S]}} = \left(\sum_{i=1}^{2n} \mathbb{E}[\tilde{X}_i^2]\right)^{-1/2} \left(\sum_{i=1}^{2n} \tilde{X}_i\right)$$

Then by Lemma **??**, there exists some constant C so that

$$\sup_{x\in\mathbb{R}} |F_{\tilde{S}}(x) - \Phi(x)| \leq C \left(\sum_{i=1}^{2n} \mathbb{E}[\tilde{X}_i^2]\right)^{-1/2} \max_i \frac{\mathbb{E}[|\tilde{X}_i|^3]}{\mathbb{E}[\tilde{X}_i^2]}$$

As $1 \leq w_i = o(\sqrt{np})$, it follows that

$$\sum_{i=1}^{2n} \mathbb{E}[\tilde{X}_i^2] = p(1-p) \sum_{i=1}^{2n} w_i^2 \geq 2p(1-p)n$$

and

$$\frac{\mathbb{E}[|\tilde{X}_i|^3]}{\mathbb{E}[\tilde{X}_i^2]} \leq w_i = o(\sqrt{np})$$

Hence,

$$\sup_{x\in\mathbb{R}} |F_{\tilde{S}}(x) - \Phi(x)| \leq C \frac{w_i}{\sqrt{2p(1-p)n}} = o\left(\frac{1}{\sqrt{1-p}}\right) = o(1)$$

for $p = o(1)$. Noting that

$$\Pr(S < t) = \Pr\left(\tilde{S} < \frac{t - \mathbb{E}[S]}{\sqrt{\mathrm{Var}[S}}\right) = F_{\tilde{S}}\left(\frac{t - \mathbb{E}[S]}{\sqrt{\mathrm{Var}[S]}}\right)$$

and using the affine property of normal random variables completes the proof. □

Next, we will use the distribution of a normal random variable, conditioned on its sum with another normal random variable.

Lemma 20. *For $X \sim \mathcal{N}(\mu_x, \sigma_x^2)$, $Y \sim \mathcal{N}(\mu_y, \sigma_y^2)$, and $Z = X + Y$, then conditioning X*

on Z gives

$$X|(Z=z) \sim \mathcal{N}\left(\frac{\sigma_x^2}{\sigma_x^2+\sigma_y^2}z + \frac{\sigma_y^2\mu_x - \sigma_x^2\mu_y}{\sigma_x^2+\sigma_y^2}, \frac{\sigma_x^2\sigma_y^2}{\sigma_x^2+\sigma_y^2}\right)$$

Proof. Bayes' theorem provides

$$f_{X|Z=z}(x,z) = \frac{f_{Z|X=x}(z,x)f_X(x)}{f_Z(z)}$$

where f_W is the probability density function for variable W. From the definition, $f_{Z|X=x}(z,x) = f_Y(z-x)$. Then substituting the Gaussian probability density function and simplifying, we have:

$$\begin{aligned} f_{X|Z=z}(x,z) &= \frac{\frac{1}{\sqrt{2\pi\sigma_y^2}}\exp\left(-\frac{(z-x-\mu_y)^2}{2\sigma_y^2}\right)\frac{1}{\sqrt{2\pi\sigma_x^2}}\exp\left(-\frac{(x-\mu_x)^2}{2\sigma_x^2}\right)}{\frac{1}{\sqrt{2\pi(\sigma_x^2+\sigma_y^2)}}\exp\left(-\frac{(z-(\mu_x+\mu_y))^2}{2(\sigma_x^2+\sigma_y^2)}\right)} \\ &= \frac{1}{\sqrt{2\pi\frac{\sigma_x^2\sigma_y^2}{\sigma_x^2+\sigma_y^2}}}\exp\left(-\frac{1}{2\frac{\sigma_x^2\sigma_y^2}{\sigma_x^2+\sigma_y^2}}\left(x-\left(\frac{\sigma_x^2}{\sigma_x^2+\sigma_y^2}z+\frac{\sigma_y^2\mu_x-\sigma_x^2\mu_y}{\sigma_x^2+\sigma_y^2}\right)\right)^2\right) \end{aligned}$$

□

The following is the distribution of a binomial variable X, given that we know the value of another binomial variable Y which uses X as its number of trials.

Lemma 21. *Denote by $\mathcal{B}(n,p)$ the binomial distribution over n trials with probability of success p. Let $X \sum \mathcal{B}(n,p)$ and $Y|X \sim \mathcal{B}(X,q)$. Then*

$$X|Y \sim Y + \mathcal{B}(n-Y, \tfrac{p(1-q)}{1-pq})$$

Proof. Via Bayes' rule,

$$\Pr(X=x|Y=y) = \frac{\Pr(Y=y|X=x)\Pr(X=x)}{\Pr(Y=y)}$$

It is well-known that $Y \sim \mathcal{B}(n, pq)$. Hence, using the formulae for the distributions and simplifying,

$$\begin{aligned}
\Pr(X = x|Y = y) &= \frac{\binom{x}{y} q^y (1-q)^{x-y} \binom{n}{x} p^x (1-p)^{n-x}}{\binom{n}{y} (pq)^y (1-pq)^{n-y}} \\
&= \binom{n-y}{x-y} \frac{(p(1-q))^{x-y}(1-p)^{n-x}}{(1-pq)^{n-y}} \\
&= \binom{n-y}{x-y} \left(\frac{p(1-q)}{1-pq}\right)^{x-y} \left(\frac{1-p}{1-pq}\right)^{n-x} \\
&= \binom{n-y}{x-y} \left(\frac{p(1-q)}{1-pq}\right)^{x-y} \left(1 - \frac{p(1-q)}{1-pq}\right)^{n-x}
\end{aligned}$$

Note that for $Z \sim \mathcal{B}(n-y, \frac{p(1-q)}{1-pq})$, we have

$$\Pr(X = x|Y = y) = \Pr(Z = x - y)$$

and so $X|Y \sim Y + Z$. □

The next observation is useful: Exponentiating a random variable by a base close to one will increase its concentration.

Lemma 22. *Let $X \sim \mathcal{N}(\mu_x, \sigma_x^2)$ be a normal variable, and let $Y = (1+\beta)^X$. Then Y is lognormal with*

$$\begin{aligned}
\mathbb{E}(Y) &= (1+\beta)^{\mu_x} (1+\beta)^{\ln(1+\beta)\sigma_x^2/2} \\
\operatorname{Var} Y &= ((1+\beta)^{\ln(1+\beta)\sigma_x^2} - 1)(1+\beta)^{2\mu_x} (1+\beta)^{\ln(1+\beta)\sigma_x^2}
\end{aligned}$$

In particular, for $\ln(1+\beta)\sigma_x^2$ close to 0, Y is highly concentrated at $(1+\beta)^{\mu_x}$.

Proof. Observe that $Y = e^{\ln(1+\beta)X}$, so it is clearly lognormal. So, define $\tilde{X} = \ln(1+$

$\beta)X \sim \mathcal{N}(\ln(1+\beta)\mu_x, \ln(1+\beta)^2\sigma_x^2)$. Then we have

$$\begin{aligned}\mathbb{E}(Y) &= \exp\left(\mathbb{E}(\tilde{X}) + \frac{1}{2}\operatorname{Var}\tilde{X}\right) \\ &= \exp\left(\ln(1+\beta)\mu_x + \frac{1}{2}\ln(1+\beta)^2\sigma_x^2\right) \\ &= (1+\beta)^{\mu_x}(1+\beta)^{\ln(1+\beta)\sigma_x^2/2}\end{aligned}$$

and

$$\begin{aligned}\operatorname{Var} Y &= \left(\exp\left(\operatorname{Var}\tilde{X}\right) - 1\right)\exp\left(2\mathbb{E}(\tilde{X}) + \operatorname{Var}\tilde{X}\right) \\ &= \left(\exp\left(\ln(1+\beta)^2\sigma_x^2\right) - 1\right)\exp\left(2\ln(1+\beta)\mu_x + \ln(1+\beta)^2\sigma_x^2\right) \\ &= ((1+\beta)^{\ln(1+\beta)\sigma_x^2} - 1)(1+\beta)^{2\mu_x}(1+\beta)^{\ln(1+\beta)\sigma_x^2}\end{aligned}$$

Furthermore, if $\ln(1+\beta)\sigma_x^2 \approx 0$, then $(1+\beta)^{\ln(1+\beta)\sigma_x^2} \approx 1$ and we obtain the concentration.

□

For learning a linear threshold function with an assembly (Theorem 18), we will require an additional lemma.

Lemma 23. *Let $X_1, \ldots, X_n$ and $Y_1, \ldots, Y_n$ be independent Bernoulli variables, and let $Z = \sum_{j=1}^n X_i Y_i$. Then for any $t \geq 0$,*

$$\mathbb{E}(Y_i | Z \geq \mathbb{E}(Z) + t) \geq \mathbb{E}(Y_i)$$

Proof. Bayes' rule gives

$$\mathbb{E}(Y_i|Z \geq \mathbb{E}(Z)+t) = \Pr(Y_i = 1|Z \geq \mathbb{E}(Z)+t) = \frac{\Pr(Z \geq \mathbb{E}(Z)+t|Y_i=1)\Pr(Y_i=1)}{\Pr(Z \geq \mathbb{E}(Z)+t)}$$

Then observe that the events $Z \geq \mathbb{E}Z + t$ and $Y_i = 1$ are positively correlated, and so

$$\Pr(Z \geq \mathbb{E}Z + t | Y_i = 1) \geq \Pr(Z \geq \mathbb{E}Z + t)$$

Then substituting gives

$$\begin{aligned}\mathbb{E}(Y_i | Z \geq \mathbb{E}(Z) + t) &\geq \frac{\Pr\left(Z \geq \mathbb{E}(Z) + t\right) \Pr(Y_i = 1)}{\Pr(Z \geq \mathbb{E}(Z) + t)} \\ &= \mathbb{E}(Y_i)\end{aligned}$$

as required. □

Lastly, the following lemma allows us to translate a bound on the weight between certain synapses into a bound on the number of rounds (or samples) required.

Lemma 24. *Consider a neuron i, connected by a synapse to a neuron j with weight initially 1, and equipped with a plasticity parameter β. Assume that j fires with probability p and i fires with probability q on each round, and that there are at least T rounds, with*

$$T \geq \frac{1}{pq} \frac{\ln \gamma}{\ln(1 + \beta)}$$

Then the synapse will have weight at least γ in expectation.

We are now equipped to prove the theorems.

3.5.2 Proof of Theorem 14

Let μ_t be the fraction of first-timers in the cap on round t. The process stabilizes when $\mu_t < 1/k$, as then no new neurons have entered the cap.

For a given neuron i, let $X(t)$ and $Y(t)$ denote the input from connections to the k neurons in S_A and the $n - k$ neurons outside of S_A, respectively, on round t. For a neuron

which has never fired before, they are distributed approximately as

$$X(t) \sim \mathcal{N}(kpr, kpr) \quad Y(t) \sim \mathcal{N}(kpq, kpq)$$

which follows from Lemma 19, for a total input of $X(t) + Y(t) \sim \mathcal{N}(kpr + kpq, kpr + kpq)$. (Note that we ignore small second-order terms in the variance.) To determine which neurons will make the cap on the first round, we need a threshold that roughly k of n draws from $X(1) + Y(1)$ will exceed, with constant probability. In other words, we need the probability that $X(1)+Y(1)$ exceeds this threshold to be about k/n. Taking $L = 2\ln(n/k)$ and using the tail bound in Lemma 5, we find the threshold for the first cap to be at least

$$C_1 = kp(r + q) + \sqrt{kp(r + q)L}$$

On subsequent rounds, there is additional input from connections to the previous cap, distributed as $\mathcal{N}(kp, kp(1-p))$. Using μ_t as the fraction of first-timers, a first-time neuron must be in the top $\mu_t k$ of the $n - k \sim n$ neurons left out of the previous cap. The activation threshold is thus

$$C_t = kp(1 + r) + kpq + \sqrt{kp(1 + r + q)(L + 2\ln(1/\mu_t))}$$

Now consider a neuron i which fired on the first round. We know that $X(1) + Y(1) \geq C_1$, so using Lemma 20,

$$X(1)|(X(1) + Y(1) = C_1) \sim \mathcal{N}\left(\frac{r}{r+q}C_1, kp\frac{rq}{r+q}\right)$$

If $X(1) = x$, Lemma 21 indicates that the true number of connections with stimulus neurons is distributed roughly as $\tilde{X}|(X(1) = x) \sim \mathcal{N}(x + (k-x)p(1-r), (k-x)p(1-r))$. Conditioning on $X(1) + Y(1) = C_1$, ignoring second-order terms, and bounding the vari-

ance as kp, we have

$$\tilde{X}|(X(1)+Y(1)=C_1) \sim \mathcal{N}\left(kp(1-r)+\frac{r}{r+q}C_1, kp\right)$$

On the second round, the synapses between neuron i and stimulus neurons which fired have had their weights increased by a factor of $1+\beta$, and these stimulus neurons will fire on the second round with probability r. An additional $\tilde{X}-X(1)$ stimulus neurons have a chance to fire for the first time. Neuron i also receives recurrent input from the k other neurons which fired the previous round, which it is connected to with probability p. So, the total input to neuron i is roughly

$$\mathcal{N}\left((1+\beta)\frac{r^2}{r+q}C_1+kpr(1-r), kp\left(1+\frac{rq}{r+q}\right)\right)+\mathcal{N}(kp(1+q), kp(1+q))$$

In order for i to make the second cap, we need that its input exceeds the threshold for first-timers, i.e.

$$(1+\beta)\frac{r^2}{r+q}C_1+kp(1+r(1-r)+q)+Z \geq C_2$$

where $Z \sim \mathcal{N}(0, kp(1+r+q))$. Taking $\mu=\mu_2$, we have the following:

$$\begin{aligned}
&\Pr(i \in C_2 | i \in C_1) = 1-\mu \\
&\quad \geq \Pr\left(Z \geq C_2-(1+\beta)\frac{r^2}{r+q}C_1-kp(1+r(1-r)+q)\right) \\
&\quad \geq \Pr\left(Z \geq -\beta kpr^2-(1+\beta)\frac{r^2}{\sqrt{r+q}}\sqrt{kpL}+\sqrt{kp(1+r+q)(L+2\ln(1/\mu))}\right)
\end{aligned}$$

Now, normalizing Z to $\mathcal{N}(0,1)$ we have (again by the tail bound)

$$1-\mu \geq 1-\exp\left(-\frac{\left(\beta\sqrt{kp}r^2+(1+\beta)\frac{r^2}{\sqrt{r+q}}\sqrt{L}+\sqrt{(1+r+q)(L+2\ln(1/\mu))}\right)^2}{2(1+r+q)}\right)$$

More clearly, this means

$$\sqrt{2(1+r+q)\ln(1/\mu)} \leq \beta\sqrt{kp}r^2 + (1+\beta)\frac{r^2}{\sqrt{r+q}}\sqrt{L} + \sqrt{(1+r+q)(L+2\ln(1/\mu))}$$

Then taking

$$\beta \geq \beta_0 = \frac{\sqrt{r+q}}{r^2}\frac{\left(\sqrt{1+r+q} - \frac{r^2}{\sqrt{r+q}}\right)\sqrt{L} + \sqrt{2(1+r+q)}}{\sqrt{kp}+\sqrt{L}}$$

gives $\mu \leq 1/e$, i.e. the overlap between the first two caps is at least a $1 - 1/e$ fraction.

Now, we seek to show that the probability of a neuron leaving the cap drops off exponentially the more rounds it makes it in. Suppose that neuron i makes it into the first cap and stays for t consecutive caps. Each of its connections with stimulus neurons will be strengthened by the number of times that stimulus neuron fired, roughly $\mathcal{N}(tr, tr(1-r))$ times. Using Lemma 22, the weight of the connection with a stimulus neuron is highly concentrated around $(1+\beta)^{tr}$. Furthermore we know that i has at least $\tilde{X}|(X(1)+Y(1) = C_1) \sim \mathcal{N}\left(kp(1-r) + \frac{r}{r+q}C_1, kp\right)$ such connections, of which $\mathcal{N}\left(kpr(1-r) + \frac{r^2}{r+q}C_1, kpr\right)$ will fire. So, the input to neuron i will be at least

$$(1+\beta)^{tr}(kpr(1-r) + \frac{r^2}{r+q}C_1) + kp(1+q) + Z$$

where $Z \sim \mathcal{N}(0, kp(1+(1+\beta)^{2tr}r+q))$

To stay in the $(t+1)$th cap, it suffices that this input is greater than C_{t+1}, the threshold

for first-timers. Using $\mu = \mu_{t+1}$ and reasoning as before:

$$\begin{aligned}
\Pr(i \in C_{t+1} | i \in C_1 \cap \ldots \cap C_t) &= 1 - \mu \\
&\geq \Pr\left(Z > C_{t+1} - (1+\beta)^{tr}\left(kpr(1-r) + \frac{r^2}{r+q}C_1\right) - kp(1+q)\right) \\
&= \Pr\left(Z > -t\beta kpr - (1+tr\beta)\frac{r^2}{\sqrt{r+q}}\sqrt{kpL} + \sqrt{kp(1+r+q)(L+2\ln(1/\mu))}\right) \\
&\geq 1 - \exp\left(-\frac{\left(tr\beta\sqrt{kp} + (1+tr\beta)\frac{r^2}{\sqrt{r+q}}\sqrt{L} - \sqrt{(1+r+q)(L+2\ln(1/\mu))}\right)^2}{2(1+r+q)}\right)
\end{aligned}$$

where in the last step we approximately normalized Z to $\mathcal{N}(0,1)$. Then

$$\beta \geq \frac{1}{tr^2}\frac{\sqrt{(1+r+q)(L+2t^2)} - r^2)\sqrt{L} + t\sqrt{2}}{\sqrt{kp} + \sqrt{L}}$$

will ensure $\mu \leq e^{-t^2}$, which is no more than β_0.

Now, let neuron i be a first time winner on round t. Let $X \sim \mathcal{N}(kpr, kpr)$ denote the input from stimulus neurons, $Y \sim \mathcal{N}(kp, kp)$ the input from recurrent connections to neurons in the previous cap, and $Z \sim \mathcal{N}(kpq, kpq)$ the input from nonstimulus neurons. Then conditioned on $X + Y + Z = C_t$, the second lemma indicates that

$$\begin{aligned}
X|(X+Y+Z=C_t) &\sim \mathcal{N}\left(\frac{r}{1+r+q}C_t, kp\frac{r(1+q)}{1+r+q}\right) \\
Y|(X+Y+Z=C_t) &\sim \mathcal{N}\left(\frac{1}{1+r+q}C_t, kp\frac{r+q}{1+r+q}\right)
\end{aligned}$$

So, the input on round $t+1$ is at least

$$(1+\beta)\frac{1-\mu_t+r^2}{1+r+q}C_t + kpr(1-r) + kp\mu_t + kpq + Z$$

where $Z \sim \mathcal{N}\left(0, kp\left((1+\beta)^2\frac{r^2(1+q)+(1-\mu_t)(r+q)}{1+r+q} + r(1-r) + q\right)\right)$. By the usual argu-

ment we have

$$\begin{aligned}\Pr(i \in C_{t+1} | i \in C_t) &= 1 - \mu_{t+1} \\ &\geq \Pr\left(Z \geq C_{t+1} - (1+\beta)\frac{1-\mu_t+r^2}{1+r+q}C_t - kpr(1-r) - kp\mu_t - kpq\right)\end{aligned}$$

So, we will have $\mu_{t+1} < e^{-1}\mu_t$ when

$$\beta \geq \frac{1}{1-\mu_t+r^2}\frac{\frac{r(1-r)+q+\mu_t}{\sqrt{1+r+q}}\sqrt{L+2\ln(1/\mu_t)}+\sqrt{2\ln(1/\mu_t)}}{\sqrt{kp}+\sqrt{\frac{L+2\ln(1/\mu_t)}{1+r+q}}}$$

which is smaller than β_0. Assuming $r+q \sim 1$, we may simplify β_0, so that

$$\beta_0 = \frac{1}{r^2}\frac{\left(\sqrt{2}-r^2\right)\sqrt{L}+\sqrt{6}}{\sqrt{kp}+\sqrt{L}}$$

So, if $\beta \geq \beta_0$, the probability of leaving the cap once in the cap t times drops off exponentially. We can conclude that no more than $\ln(k)$ rounds will be required for convergence. Additionally, assuming that a neuron enters the cap at time t, let $1-p_\tau$ denote the probability it leaves after τ rounds. Then its probability of staying in the cap on all subsequent rounds is

$$\prod_{\tau \geq 1} p_\tau \geq \prod_{\tau \geq 1}\left(1-\exp\left(-\tau^2\left(\frac{\beta}{\beta_0}\right)^2\right)\right) \geq 1-\exp\left(-\left(\frac{\beta}{\beta_0}\right)^2\right)$$

Thus, every neuron that makes it into the cap has a probability at least $1-\exp(-(\beta/\beta_0)^2))$ of making every subsequent cap, so the total support of all caps together is no more than $k/(1-\exp(-(\beta/\beta_0)^2)))$ in expectation. ■

Proof of Theorem 15

Let μ denote the fraction of newcomers in the cap. A neuron in A^* can expect an input of

$$X_a = \gamma kpr + kpq + Z_a$$

where $Z_a \sim \mathcal{N}(0, \gamma^2 kpr + kpq)$, while neurons outside of A^* can expect an input of

$$X = kpr + kpq + Z$$

where $Z \sim \mathcal{N}(0, kpr + kpq)$. Then the threshold is roughly

$$C_1 = kpr + kpq + \sqrt{kp(r+q)(L + 2\ln(1/\mu))}$$

For a neuron i in A^* to make the cap, it needs to exceed this threshold. We have

$$\begin{aligned}
\Pr(i \in C_1 | i \in A^*) &= 1 - \mu \\
&\geq \Pr(X_a \geq C_1) \\
&= \Pr\left(Z_a \geq -(\gamma - 1)kpr + \sqrt{kp(r+q)(L + 2\ln(1/\mu))}\right)
\end{aligned}$$

Applying the tail bound gives

$$\sqrt{2\ln(1/\mu)} \leq (\gamma - 1)\frac{r}{\sqrt{r+q}}\sqrt{kp} - \sqrt{L + 2\ln(1/\mu)}$$

so for $r + q \sim 1$ and

$$\gamma \geq 1 + \frac{1}{\sqrt{r}}\left(\sqrt{2} + \sqrt{L/kpr + 2}\right)$$

we will have $\mu \leq e^{-kpr}$. ■

Proof of Theorem 16

Let ν_t be the fraction of neurons in A^* included in the cap on round t, and let μ_t be the fraction of true first-timers. The input on the first round for first-timers will be $\mathcal{N}(kpr, kpr) + \mathcal{N}(kpq, kpq)$, while for neurons in A^* will be

$$\mathcal{N}(\gamma\alpha kpr, \gamma^2\alpha kpr) + \mathcal{N}((1-\alpha)kpr, (1-\alpha)kpr) + \mathcal{N}(kpq, kpq)$$

For a neuron not in A^* to make the cap, it needs to be in the top $(1-\nu_1)k$ of $n-k \sim n$ draws. Thus, the threshold is at least

$$C_1 = kp(r+q) + \sqrt{kp(r+q)(L - 2\ln(1-\nu_1))}$$

For a neuron in A^* to make the cap, it needs to exceed this threshold. Thus, we have

$$\begin{aligned}
\Pr(i \in C_1 | i \in A^*) &= \nu_1 \\
&\leq \Pr(Z > C_1 - \gamma\alpha kpr - (1-\alpha)kpr - kpq) \\
&= \Pr\left(Z > -(\gamma-1)\alpha kpr + \sqrt{kp(r+q)(L - 2\ln(1-\nu_1))}\right) \\
&\leq \exp\left(-(\sqrt{(r+q)(L - 2\ln(1-\nu_1))} - (\gamma-1)\alpha r\sqrt{kp})^2/2\right)
\end{aligned}$$

Rearranging we have

$$\sqrt{2\ln(1/\nu_1)} \leq \sqrt{(r+q)(L - 2\ln(1-\nu_1))} - (\gamma-1)\alpha r\sqrt{kp}$$

Thus, so as long as

$$\gamma \leq 1 + \frac{\sqrt{(r+q)L} - \sqrt{2\ln((1+r)/r\alpha)}}{\alpha r\sqrt{kp}}$$

we will have $\nu_1 \leq r\alpha/(1+r)$. On any round t after the first, a neuron in $S_A \setminus C_{t-1}$ will receive

$$(1-\alpha)kpr + \gamma\alpha kpr + \gamma\nu_{t-1}kp + (1-\nu_{t-1})kp + kpq$$

while the threshold for first-timers is at least

$$kp(1+r+q) + \sqrt{kp(1+r+q)(L+2\ln(1/\mu_t))}$$

where $\mu_t \leq e^{-t^2\beta/\beta_0}$, as in the proof of Theorem 1. The neurons in S_A which make the cap on round $t+1$ consist of those that made the previous cap and this one, and those that are first-timers. From Theorem 1, we know $\Pr(i \in C_{t+1} | i \in C_t) \geq 1 - e^{-t^2}$, so take $\Pr(i \in C_{t+1} | i \in S_A \cap C_t) \sim \nu_t$. We need only to find $\Pr(i \in C_{t+1} | i \in S_A \setminus C_t) = \nu'_t$. On the second round, we have $\nu_1 = \frac{r\alpha}{1+r}$. So, letting $Z \sim \mathcal{N}(0, kp(\gamma^2-1)\alpha(1+r) + kpq)$, it follows that:

$$\begin{aligned}
\Pr(i \in C_2 | i \in S_A \setminus C_1) &= \nu'_2 \\
&\leq \Pr\left(Z > -(\gamma-1)\frac{\alpha(2+r)}{1+r})kpr + \sqrt{kp(1+r+q)(L)}\right) \\
&\leq \exp\left(-\frac{1}{2}\left(\sqrt{(1+r+q)(L)} - \frac{\alpha r(2+r)}{1+r})\sqrt{kp}\right)^2\right)
\end{aligned}$$

and so if

$$\gamma \leq 1 + \frac{\sqrt{(1+r+q)L} - \sqrt{2\ln((1+r)/\alpha)}}{\alpha\frac{r^2+2r}{1+r}\sqrt{kp}}$$

will ensure that $\nu'_2 \leq \alpha/(1+r)$, which is very nearly the bound for the first cap. Thus, no more than an α fraction of neurons in the second cap are in S_A.

Now, we seek to show that if $\nu_t \leq \alpha$, then we will have $\nu_{t+1} \leq \alpha + 1/k$, since this will ensure that no new neurons from S_A have entered the cap. Reasoning as before, let

$\Pr(i \in C_{t+1} | i \in S_A \setminus C_t) = \nu'_{t+1}$, and then we have

$$\begin{aligned}\Pr(i \in C_{t+1} | i \in S_A \setminus C_t) &= \nu'_{t+1} \\ &\leq \Pr\left(Z > -2(\gamma-1)\alpha kpr + \sqrt{kp(1+r+q)(L+2\ln(1/\mu_t))}\right) \\ &\leq \exp\left(-\left(\sqrt{(1+r+q)(L+2\ln(1/\mu_t))} - 2(\gamma-1)\alpha r\sqrt{kp}\right)^2/2\right)\end{aligned}$$

Then solving for γ, we find that we will have $\nu'_t \leq 1/k$ as long as

$$\gamma \leq 1 + \frac{\sqrt{(1+r+q)(L+2t^2)} - \sqrt{2\ln(k)}}{2\alpha r\sqrt{kp}}$$

which is the least upper bound so far. Thus, taking $r + q \sim 1$, so long as

$$\gamma \leq 1 + \frac{\sqrt{L} - \sqrt{2\ln((1+r)/r\alpha)}}{\alpha r\sqrt{kp}}$$

the overlap of any cap with A^* will never exceed αk neurons. By Theorem 1, an assembly B will form with high probability after $\ln(k)$ examples, and we conclude that $|A^* \cap B^*| \leq \alpha k$. ■

Proof of Theorem 17

Let μ be the fraction of first-timers included in the cap C_1, and ν be the fraction of neurons in A^* in the cap. A neuron in B^* receives

$$X_b = \gamma kpr + kpq + Z_b$$

where $Z_b \sim \mathcal{N}(0, \gamma^2 kpr + kpq)$ while a neuron in A^* receives

$$\begin{aligned}X_a &= \gamma\alpha kpr + \gamma(1-\alpha)kp\left(\frac{qk}{n}\right) + kpq + Z_a \\ &= \gamma\alpha kpr + kpq + Z_a + O(1)\end{aligned}$$

where $Z_a \sim \mathcal{N}(0, \gamma^2 \alpha kpr + kpq)$. The threshold to be in the top νk of $|A^*| \sim k$ draws from this distribution is

$$C_1 = \gamma \alpha kpr + kpq + \sqrt{2kp(\gamma^2 \alpha r + q) \ln(1/\nu)}$$

Neurons in B^* will make the first cap if they exceed this threshold. So, we have

$$\begin{aligned}
\Pr(i \in C_1 | i \in B^*) &= 1 - \nu \\
&\geq \Pr(X_b \geq C_1) \\
&= \Pr\left(Z_b \geq -\gamma(1-\alpha)kpr + \sqrt{2kp(\gamma^2 \alpha r + q) \ln(1/\nu)}\right) \\
&\geq 1 - \exp\left(-\frac{1}{2} \frac{\left(\gamma(1-\alpha) r \sqrt{kp} - \sqrt{2(\gamma^2 \alpha r + q) \ln(1/\nu)}\right)^2}{\gamma^2 r + q}\right)
\end{aligned}$$

where the last step follows from Lemma 5. Solving for ν gives

$$\nu \leq \exp\left(-\frac{(1-\alpha)^2 kpr}{2(1+\alpha)}\right)$$

Similarly, neurons in neither of the two assembly cores must also exceed the threshold to make the cap, which means

$$\begin{aligned}
\Pr(i \in C_1 | i \notin A^* \cup B^*) &= \mu \\
&\leq \Pr(X > C_1) \\
&= \Pr\left(Z > (\gamma\alpha - 1)kpr + \sqrt{2kp(\gamma^2 \alpha r + q) \ln(1/\nu)}\right) \\
&\leq \exp\left(-\frac{1}{2(r+q)} \left((\gamma\alpha - 1) r \sqrt{kp} + \sqrt{2(\gamma^2 \alpha r + q) \ln(1/\nu)}\right)^2\right) \\
&\leq \exp\left(-\frac{1}{2}(\gamma\alpha - 1)^2 kpr\right) \cdot \nu^{\gamma^2 \alpha}
\end{aligned}$$

Then the fraction of neurons in $C_1 \setminus B^*$ will be

$$\nu + \mu \leq \exp\left(-\frac{(1-\alpha)^2 kpr}{2(1+\alpha)}\right) + \exp\left(-\frac{1}{2}(\gamma\alpha - 1)^2 kpr\right)$$

■

Proof of Theorem 18

For each round $1, \ldots, t, \ldots$, define μ_t to be the fraction of neurons firing for the first time on that round, and let $X(t)$ be an example sampled from $\mathcal{D}_+$. For each neuron i, let $W_j^i(t)$ represent the weight of the synapses between neuron i and input neuron j on round t. Since each synapse is present independently with probability p, the number of synapses (i.e. the norm $\|W^i\|_0$) between the input region and neuron i is a binomial random variable, with expectation np. Furthermore, the Chernoff bound shows that the number of synapses is sharply concentrated about its mean as

$$\Pr(|\|W^i\|_0 - np| \geq n^{1-\epsilon}p) \leq 2\exp\left(-\frac{n^{1-2\epsilon}p}{3}\right)$$

for any $\epsilon > 0$. Thus, once normalized, the weight of a particular synapse $W_j^i(0)$ will be between, say, $\frac{1}{np - n^{1/3}p}$ and $\frac{1}{np + n^{1/3}p}$ with high probability. Since both of the quantities approach $\frac{1}{np}$ for large enough n, we will regard the weight of every synapse at the outset as $\frac{1}{np}$, so that $\mathbb{E}W_j^i(0) = \frac{p}{np} = \frac{1}{n}$.

If i has not yet fired, then the round t input $W^i(t) \cdot X(t)$ is approximately normal, with mean

$$\mathbb{E}(W^i(t) \cdot X(t)) = \sum_{j=1}^{n} \mathbb{E}(W_j^i(t))\mathbb{E}(X_j(t)) \geq \sum_{j=1}^{n} \frac{1}{n}\left(\frac{k}{n} + \Delta v_j\right) = \frac{k}{n} + \frac{\Delta}{n}\|v\|_1$$

and variance approximately $\frac{k}{n^2 p}$. So, using Lemma 5, a neuron will be in the top $\mu_t k$ of the

approximately n neurons which have never fired before if its input from $X(1)$ exceeds

$$C_1 = \frac{k}{n} + \frac{\Delta}{n}\|v\|_1 + \frac{\sqrt{kpL}}{np}$$

where $L = 2\ln(n/k)$, and including input from the previous cap $N(k/n, k/n^2p)$ on subsequent rounds,

$$C_t = \frac{2k}{n} + \frac{\Delta}{n}\|v\|_1 + \frac{1}{np}\sqrt{2kp(L + 2\ln(1/\mu_t))}$$

Now, let i be a neuron which made the first cap. Then $W^i(1)\cdot X(1) \geq C_1$ and each nonzero component in $W^i(1)$ will be increased by a factor of $1+\beta$ if the corresponding component of $X(1)$ was nonzero as well. By Lemma 23, the conditional expectation of $W_j^i(2)$ will be

$$\begin{aligned}\mathbb{E}(W_j^i(2)|W^i(1)\cdot X(1) \geq C_1) &\geq (1+\beta\Pr(X(1)=1))\,\mathbb{E}(W_j^i(1)) \\ &\geq \frac{1}{n} + \beta\frac{1}{n}\left(\frac{k}{n} + \Delta v_j\right)\end{aligned}$$

Then we have

$$\begin{aligned}\mathbb{E}(W^i(2)\cdot X(2)|i \in C_1) &= \sum_{j=1}^{n}\mathbb{E}(W_j^i(2)|i\in C_1)\mathbb{E}(X_j(2)) \\ &\geq \sum_{j=1}^{n}\frac{1}{n}\left(\frac{k}{n}+\Delta v_j\right) + \beta\frac{1}{n}\left(\frac{k}{n}+\Delta v_j\right)^2 \\ &\geq \frac{k}{n} + \frac{\Delta}{n}\|v\|_1 + \frac{\beta\Delta^2}{n}\end{aligned}$$

Then letting $Y \sim \mathcal{N}(k/n, k/n^2p)$ denote the input from the previous cap, we have

$$\begin{aligned}\Pr(i \in C_2|i\in C_1) &= 1-\mu_2 \\ &\geq \Pr(W^i(2)\cdot X(2) + Y \geq C_2)\end{aligned}$$

Taking $Z \sim \mathcal{N}(0, k/n^2 p)$ to be an underestimate of the variance,

$$\begin{aligned} 1 - \mu_2 &\geq \Pr\left(Z \geq \frac{1}{np}\sqrt{2kp(L + 2\ln(1/\mu_2))} - \frac{\beta\Delta^2}{n}\right) \\ &\geq 1 - \exp\left(-\frac{1}{2kp}\left(\beta\Delta^2 p - \sqrt{2kp(L + 2\ln(1/\mu_2))}\right)^2\right) \end{aligned}$$

following from the tail bound in Lemma 5. Then so as long as

$$\Delta^2\beta \geq \frac{\sqrt{2k(L+2)} + \sqrt{2k}}{\sqrt{p}}$$

we will have $\mu_2 \leq 1/e$.

Now, suppose i fired on all of the first t rounds. Input neuron j is expected to fire at least $(\frac{k}{n} + \Delta v_i)t$ times, so by Lemma 22 the weight of an extant synapse $W_j^i(t+1)$ will be concentrated about its mean, which is at least $\frac{1}{np}(1+\beta)^{(\frac{k}{n}+\Delta v_i)t}$. The expected input $W^i(t+1) \cdot X(t+1)$ is thus

$$\begin{aligned} \mathbb{E}(W^i(t+1) \cdot X(t+1) | i \in C_1 \cap \ldots \cap C_t) &= \sum_{j=1}^{n} \mathbb{E}(W_j^i(t+1) | i \in C_1 \cap \ldots \cap C_t)\hat{\mathbb{E}}(X_j(t+1)) \\ &\geq \sum_{j=1}^{n} \frac{1}{n}\left(\frac{k}{n} + \Delta v_j\right) + \beta t \frac{1}{n}\left(\frac{k}{n} + \Delta v_j\right)^2 \\ &\geq \frac{k}{n} + \frac{\Delta}{n}\|v\|_1 + \frac{\Delta^2}{n}\beta t \end{aligned}$$

Then including recurrent input $Y \sim \mathcal{N}(kp, kp)$, we have

$$\begin{aligned} \Pr(i \in C_{t+1} | i \in C_1 \cap \ldots \cap C_t) &= 1 - \mu_{t+1} \\ &\geq \Pr(W^i(t+1) \cdot X(t+1) + Y \geq C_{t+1}) \end{aligned}$$

Again taking $Z \sim \mathcal{N}(0, k/n^2p)$ to be the variance, we have

$$\begin{aligned} 1 - \mu_{t+1} &\geq \Pr\left(Z \geq \frac{1}{np}\sqrt{2kp(L + 2\ln(1/\mu_{t+1}))} - \frac{\Delta^2}{n}\beta t\right) \\ &\geq 1 - \exp\left(-\frac{1}{2kp}\left(\Delta^2\beta tp - \sqrt{2kp(L + 2\ln(1/\mu_{t+1}))}\right)^2\right) \end{aligned}$$

again following from the tail bound. Then we will have $\mu_{t+1} \leq e^{-t}$ as long as

$$\Delta^2\beta \geq \frac{\sqrt{2k(L + 2t)} + \sqrt{2kt}}{t\sqrt{p}}$$

which is certainly no more than β_0.

So, if $\beta \geq \beta_0$, the probability of a neuron that made the cap t times missing the next one drops off exponentially. It follows that after $\ln(k)$ rounds the process will have converged, and each neuron in a particular cap has a probability of at least $1 - \exp(-(\beta/\beta_0)^2))$ of making every subsequent cap. So, the total support of all caps together is no more than $k/(1 - \exp(-(\beta/\beta_0)^2))) = k + o(k)$ in expectation.

Now, suppose the training process continues until the cap converges to some set A^* of size $k + o(k)$, which takes roughly $\ln(k)$ rounds with high probability. For neuron $i \in A^*$, the weight of an extant synapse $W^i_j(\ln(k))$ will be, in expectation,

$$W^i_j(\ln(k)) = \frac{1}{np}(1 + \beta)^{\left(\frac{k}{n} + \Delta v_j\right)\ln(k)}$$

and on average over the neurons in A^*, we will have

$$\sum_{j=1}^{n} W^i_j \geq \sum_{j=1}^{n} \frac{p}{np}(1 + \beta)^{\left(\frac{k}{n} + \Delta v_j\right)\ln(k)} \geq 1 + \frac{\beta\ln(k)}{n}\Delta\|v\|_1$$

The neurons are then allowed to return to rest, and each neuron renormalizes the weights

of its incoming synapses to sum to 1, so that

$$W_j^i = \frac{(1+\beta)^{\left(\frac{k}{n}+\Delta v_j\right)\ln(k)}}{np + \beta p \ln(k)\Delta\|v\|_1} + o(n^{-1})$$

Choose examples X^+ and X^- at random, so that

$$v^\top X^+ \geq \frac{k}{n}\|v\|_1 + \Delta \qquad v^\top X^- \leq \frac{k}{n}\|v\|_1 - \Delta$$

Letting C_+, C_- denote the caps formed after presenting the respective example once, define ϵ_+ to be the fraction of neurons in C_+ formed in response to X^+ which are not in A^*, and ϵ_- to be the fraction of neurons in C_- which are in A^*, i.e.

$$\epsilon_+ = \frac{|C_+ \setminus A^*|}{k} \qquad \epsilon_- = \frac{|C_- \cap A^*|}{k}$$

As before, the input for a neuron i outside of A^* will be nearly normal with expectation

$$\begin{aligned}
\mathbb{E}(W^i \cdot X^+) &= \sum_{j=1}^{n} \mathbb{E}(W_j^i)\mathbb{E}(X_j^+) \\
&\geq \sum_{j=1}^{n} \frac{1}{n}\left(\frac{k}{n} + \Delta v_j\right) \\
&= \frac{k}{n} + \frac{\Delta}{n}\|v\|_1
\end{aligned}$$

and variance nearly $Z \sim \mathcal{N}(0, k/n^2 p)$, so that the threshold to make the top $\epsilon_+ k$ neurons will be at least

$$C_+ = \frac{k}{n} + \frac{\Delta}{n}\|v\|_1 + \frac{1}{np}\sqrt{kp(L + 2\ln(1/\epsilon_+))}$$

Now, consider $i \in A^*$. Its input in expectation will be

$$\begin{aligned}\mathbb{E}(W^i \cdot X^+ | i \in A^*) &= \sum_{j=1}^{n} \mathbb{E}(W_j^i)\mathbb{E}(X_j^+) \\ &\geq \sum_{j=1}^{n} \frac{\frac{k}{n} + \Delta v_j}{np + \beta p \ln(k)\Delta\|v\|_1} + \frac{\beta \ln(k) \left(\frac{k}{n} + \Delta v_j\right)^2}{np + \beta p \ln(k)\Delta\|v\|_1} \\ &\geq \frac{k + \Delta\|v\|_1 + \beta \ln(k)\Delta^2}{np + \beta p \ln(k)\Delta\|v\|_1}\end{aligned}$$

while its variance will be

$$\begin{aligned}\mathrm{Var}(W^i \cdot X^+ | i \in A^*) &= \mathbb{E}((W^i \cdot X^+)^2 | i \in A^*) - \mathbb{E}(W^i \cdot X^+ | i \in A^*)^2 \\ &\geq \sum_{j=1}^{n} p\left(\frac{k}{n} + \Delta v_j\right) \frac{(1+\beta)^{(\frac{k}{n} + \Delta v_j) 2\ln(k)}}{(np + \beta \ln(k) p \Delta\|v\|_1)^2} \\ &\geq \frac{k}{(n + \beta \ln(k)\Delta\|v\|_1)^2 p}\end{aligned}$$

Then letting $Z \sim \mathcal{N}(0, \frac{k}{(n+\beta \ln(k)\Delta\|v\|_1)^2 p})$ denote the uncertainty, we have

$$\begin{aligned}\Pr(i \in C_+ | i \in A^*) &= 1 - \epsilon_+ \\ &\geq \Pr(W^i \cdot X^+ \geq C_+) \\ &\geq \Pr\left(Z \geq \beta \ln(k) \frac{\frac{\Delta\|v\|_1}{n}(k + \Delta\|v\|_1) - \Delta^2}{n + \beta \ln(k)\Delta\|v\|_1} + \frac{1}{np}\sqrt{kp(L + 2\ln(1/\epsilon_+))}\right)\end{aligned}$$

Then using Lemma 5,

$$\epsilon_+ \leq \exp\left(-\frac{1}{2kp}\left(p\beta \ln(k)\left(\Delta^2 - \frac{\Delta\|v\|_1(k+\Delta\|v\|_1)}{n}\right) - \frac{n+\beta \ln(k)\Delta\|v\|_1^2}{n}\sqrt{kp(L + 2\ln(1/\epsilon_+))}\right)^2\right)$$

So, for fixed ϵ_+ we need

$$\Delta^2 - \frac{\Delta\|v\|_1}{n}(k + \Delta\|v\|_1) \geq \frac{\left(1 + \beta \ln(k)\frac{\Delta\|v\|_1}{n}\right)\sqrt{kp(L + 2\ln(1/\epsilon_+))} + \sqrt{2kp\ln(1/\epsilon_+)}}{p\beta \ln(k)}$$

Now, note that if $\Delta \geq \frac{2k}{\sqrt{n}}$, recalling that $\|v\|_1 \leq \sqrt{n}/2$, we have must have $\Delta \geq \frac{k\|v\|_1}{n/2-\|v\|_1^2}$, and so

$$\begin{aligned}\Delta^2 - \frac{\Delta\|v\|_1}{n}\left(k + \Delta\|v\|_1\right) &= \Delta^2 - \left(\Delta\frac{k\|v\|_1}{n/2-\|v\|_1^2}\frac{n/2-\|v\|_1^2}{n} + \frac{\Delta^2\|v\|_1^2}{n}\right) \\ &\geq \Delta^2 - \left(\frac{\Delta^2}{2} - \frac{\Delta^2\|v\|_1^2}{n} + \frac{\Delta^2\|v\|_1^2}{n}\right) \\ &= \frac{\Delta^2}{2}\end{aligned}$$

Additionally, if $\Delta \geq 2\sqrt{\frac{k}{np}}\sqrt{L + 2\ln(1/\epsilon_+)}$ (a much milder bound compared to the previous one, for $p = \Omega(k^{-1})$ and ϵ_+ fixed) then it suffices that

$$\Delta^2\beta \geq \frac{4\sqrt{k}}{\ln(k)\sqrt{p}}\left(\sqrt{L + 2\ln(1/\epsilon_+)} + \sqrt{2\ln(1/\epsilon_+)}\right)$$

Now, consider the example X^-. The input to a neuron i outside of A^* will be nearly normal with expectation

$$\begin{aligned}\hat{\mathbb{E}}(W^i \cdot X^-) &= \sum_{j=1}^{n}\mathbb{E}(W_j^i)\mathbb{E}(X_j^-) \\ &= \sum_{j=1}^{n}\frac{k}{n^2} \\ &= \frac{k}{n}\end{aligned}$$

For $i \in A^*$,

$$\begin{aligned}\mathbb{E}(W^i \cdot X^- | i \in A^*) &= \sum_{j=1}^{n}\hat{\mathbb{E}}(W_j^i)\mathbb{E}(X_j^-) \\ &\geq \sum_{j=1}^{n}\frac{(1+\beta)^{(\frac{k}{n}+\Delta v_j)\ln(k)}}{\sum(1+\beta)^{(\frac{k}{n}+\Delta v_l)\ln(k)}}\frac{k}{n} \\ &= \frac{k}{n}\end{aligned}$$

with variance no larger than $\frac{k}{n^2 p}$. It follows that the threshold for a neuron in A^* to make the top $\epsilon_- k$ of the k neurons in A^* will be no more than

$$C_- = \frac{k}{n} + \frac{\sqrt{2kp\ln(1/\epsilon_-)}}{np}$$

So, for $Z \sim \mathcal{N}(0, k/n^2 p)$ we have

$$\begin{aligned}
\Pr(i \in C_- | i \notin A^*) &= (1-\epsilon_-)\frac{k}{n} \\
&\geq \Pr\left(W^i \cdot X^- \geq C_-)\right) \\
&\geq \Pr\left(Z \geq \frac{\sqrt{2kp\ln(1/\epsilon_-)}}{np}\right) \\
&= \epsilon_-
\end{aligned}$$

Rearranging shows that

$$\epsilon_- \leq \frac{\frac{k}{n}}{1+\frac{k}{n}} \leq \frac{k}{n}$$

Now, comparing all of above bounds on Δ when $\epsilon_+ \leq 1/e$:

$$\Delta \geq \frac{2k}{\sqrt{n}} \tag{3.1}$$

$$\Delta \geq 2\sqrt{\frac{k}{np}}\sqrt{L+2} \tag{3.2}$$

$$\Delta^2\beta \geq \frac{4\sqrt{k}}{\beta\ln(k)\sqrt{p}}\left(\sqrt{L+2}+\sqrt{2}\right) \tag{3.3}$$

$$\Delta^2\beta \geq \frac{\sqrt{2k(L+2)}+\sqrt{2k}}{\sqrt{p}} \tag{3.4}$$

Since $k \leq n$, and assuming β is no larger than a constant, the bound in (4) is the strongest. Thus, taking

$$\Delta^2\beta \geq \frac{\sqrt{2k(L+2)}+\sqrt{2k}}{\sqrt{p}}$$

and $\Delta \geq \sqrt{L+2}$ is sufficient to ensure that the assembly creation process converges within

$\ln(k)$ steps, and $\epsilon_+, \epsilon_- \leq 1/e$. It follows that if more than half (resp. less than half) of the neurons in A^* fire in response to any given example, it is a positive (resp. negative) one with high probability. ■

CHAPTER 4

SEQUENCES OF ASSEMBLIES

Many of the brain's remarkable capabilities rely on working with *sequences* (of stimuli, words, places, etc), with the human brain's acumen for language being a particularly striking example. The capacity to memorize sequences is widely attested in cognitive neuroscience [120, 121]. Experiments have documented the creation and activation of assemblies in sequence in the animal brain after training on tasks that involve sequential decisions [44, 43, 51]. Ikegaya et al. [58] observed that precisely-timed patterns of activation across large groups of neurons in the mouse neocortex are frequently repeated, suggesting memorization, and moreover that these patterns are combined into higher-order sequences (i.e., sequences of sequences). In the hippocampi of rats performing a sequential decision-making task, Pastalkova et al. [43] identified assembly sequences which predicted the decisions made. Across both navigational and memory tasks, the particular sequence which was exhibited depended closely on initial conditions and the structure of the task instance. In a further investigation by Dragoi & Tonegawa [51], sequences of neurons that were observed during a novel experience also occurred in spontaneous activity during rest before the experience was initiated again, a phenomenon known as "preplay". This finding suggests that the structure of sequence representations is largely determined by the intrinsic connectivity of the network.

Arguably, it is through sequences of stimuli and their representations that brains deal with the all-important concept of *time*. The question arises: Can NEMO capture this capability of the animal brain? In past work, NEMO did not have to deal explicitly with sequences or time. In the English parser implemented in [95], the input sentence is presented sequentially, and the order of its words is not memorized by the device. In subsequent work on parsing [97], the need to memorize subsequences of the input language became appar-

ent in connection to the *center embedding* of sentences; however, no mechanism for this memorization was proposed.

In this paper, we demonstrate the emergent formation of sequences of assemblies in NEMO. When a brain area is stimulated by the same sequence of stimuli a handful of times, assemblies are reliably created, and the entire sequence will subsequently be recalled when only the beginning of the stimulus sequence is presented. Importantly, the underlying mechanism involves the capture of time precedence between sequences through the establishment, via plasticity, of high synaptic weights between stimuli representations, in the direction of time. Moreover, we demonstrate that involving additional brain areas during presentation (essentially forming a "scaffold" of interconnected assemblies) makes memorization faster and more robust, and more so if this new area already contains another memorized sequence. This provides theoretical support to experience: It is easier to memorize a sequence of stimuli when each stimulus is mentally associated by the subject with an element of an already memorized sequence — for example, a familiar tune, or the sequence of buildings next to the subject's home.

We use these ideas to further show that, in NEMO, assemblies can be configured to simulate an arbitrary *finite state machine* (FSM, or *finite state automaton*). Recall that FSMs are simple computational devices capable of recognizing and generating the class of sequential patterns known as *regular languages* [122]. Moreover, we show that this configuration can be learned quickly by presenting sequences of stimuli corresponding to state transitions; this captures the brain's ability to learn *algorithms* involving sequences. The implementation and learning of FSMs relies crucially on one last feature of NEMO, namely *long-range interneurons (LRIs).* These are neural populations extrinsic to the brain areas of NEMO, which can be recruited by assemblies in adjacent areas, and whose function is to inhibit or disinhibit remote brain areas, potentially to achieve synchrony and control of the computation [123, 124, 125]. There is evidence in the literature [126] that LRIs are essential for the onset of γ oscillations, often considered coterminous with brain computation.

One interesting byproduct of the mechanism for implementing and learning FSMs is a simple demonstration that NEMO is *Turing complete*. In other words, NEMO with LRIs constitutes a hardware language capable of implementing any computation, within the constraints imposed by the parameters of the model. This is rather significant for a mathematical model that has the ambition to capture a large part of human cognition. The original exposition of NEMO in [96] did contain an argument of Turing-completeness as well; however, that proof relies on a biologically implausible computer-like program, with loops, conditional statements, and variables corresponding to assemblies. The Turing completeness argument in the present paper is carried out strictly within NEMO, and the required program is implemented with LRIs, yielding an entirely hardware-based general computer, consistent with neurobiological principles.

4.0.1 Related work

The NEMO model is distinguished from other general theories of neuronal coordination by its bottom-up approach to the problem: It consists only of tractable abstractions of well-understood biological mechanisms, which can be shown to yield interesting behavior. Among other such theories, Valiant's neuroidal model [80, 81, 127] is powerful but demands neurons and synapses capable of arbitrary state changes, while attempts to translate the success of deep learning via gradient descent into the brain [128, 113, 114, 115, 116, 90] rely on hypothetical feedback connections and exceedingly precise coordination.

Specifically in regard to memorizing and reproducing sequences, there are several other approaches which vary in biological fidelity and corresponding limitations. Eliasmith et al. [129] presented a biologically-plausible model of the entire brain which exhibits sequence memorization and prediction (along with other more complex cognitive tasks), but the mechanisms for doing so are complex and engineered, rather than emerging from simple dynamics. The model of Cui et al. [130] is able to memorize and predict sequences in much the same fashion that we present here, using Hebbian plasticity and k-winners-

take-all to abstract inhibition, but requires more sophisticated compartmental neurons and columnar organization. Lastly, the Tolman-Eichenbaum machine [131] is an abstraction of the hippocampus which can memorize sequences (along with more general structured sets of stimuli), but its encoding and mechanisms are simply assumed and it is trained with gradient descent. In comparison with these, NEMO is distinguished by (i) the simplicity and biological fidelity of its mechanisms and (ii) the fact that its interesting behavior is emergent, with minimal "engineering".

A model very closely related to NEMO and called *the assembly calculus (AC)* was described in Papadimitriou et al. [96], where it was shown that it is Turing complete. The simulation of a Turing machine was accomplished using what amounts to a stored *program* of control commands, entailing variables, conditional statements, and loops, including commands needed for inhibiting and disinhibiting brain areas. A key contribution of this paper is that we eliminate such programs and use biologically plausible LRIs with similar consequences. As a result, our simulation is entirely self-contained, and driven only by the presentation of stimuli — the higher-level connectivity of brain areas and interneurons can implement the general architecture of a Turing machine (i.e. a long tape of symbols and a tape head which keeps track of the current state and reads and writes symbols on the tape), while the weights of connections between neurons store the actual symbols on the tape and the state transitions of the tape head.

Sequences of assemblies in neuroscience. There is a large body of experimental evidence from neuroscience indicating that neural representations of temporally-structured patterns of stimuli often preserve that structure. The assemblies hypothesis goes a step further: these neural representations ought not to arise only in response to the sequence of stimuli, but should also be generated internally by the network, for example by playing out the entire sequence of neural responses in response to only a fragment of the entire pattern of stimuli. Here we recount certain inspiring experimental results which support this view.

In Dragoi & Buzsáki [44], neurons in the hippocampi of rats running a linear maze were recorded, and the pairwise temporal correlations of their spikes were computed. These correlations were significantly larger than predicted by the model where neurons simply fire in response to where the animal currently is, suggesting that neurons which fire in response to one location of the maze become linked with those in nearby locations. Pastalkova et al. [43] trained rats to alternate between two paths of a maze, with a delay period between each run, while again recording from the hippocampus throughout. They found that the activity during the delay period was a strong predictor of which path the rat would take, even on error runs (where the animal failed to alternate between paths correctly). As the environmental context was identical during these delay periods, this points to internal generation of these neural responses. Finally, Dragoi & Tonegawa [51] recorded from the hippocampi of mice during periods of sleep and exploration of an unfamiliar maze, and found that the sequences of firing which occurred during exploration matched those that occurred during prior sleep periods. These sequences were active before the animal even saw the novel maze, which suggests that particular temporal patterns of neural activity arise spontaneously, and these patterns are later recruited to represent environmental stimuli. Although the sequences of firings which arise in our model need not occur spontaneously before the first presentation of the stimulus sequence, they do rely crucially on internal connectivity, and by repeated presentation of the stimulus sequence become linked together to reinforce the replay of the sequence.

4.1 Computation with sequences of assemblies

4.1.1 Sequence memorization

We begin with *sequence projection,* where a sequence of assemblies from one area is projected to another area. The most natural way to project a sequence from one area to another is to simply activate, in the first area, the assemblies for each element of the sequence, one after the other in the given order. Assuming that there is a fiber connecting the first area

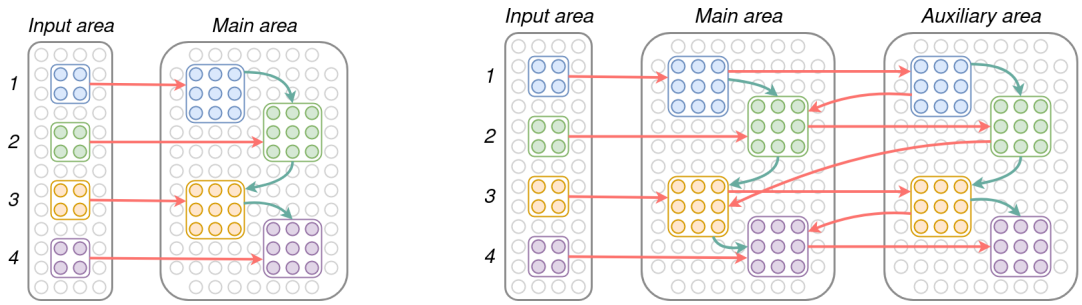


Figure 4.1: Left: Simple copy of a sequence of assemblies. When a sequence is played a few times in one area, a corresponding sequence of assemblies is formed in an adjacent area, and will subsequently be recalled when only the beginning of the sequence is presented in the input area. Right: When the area with the copied sequence is also allowed to send and receive input to/from an auxiliary area, a "scaffolded" sequence of assemblies is formed across the two areas, leading to faster, more reliable memorization.

to the second, and the second area is disinhibited, one would hope that this would result in the creation of a set of corresponding assemblies in the target area, so that activating any newly created assembly results in the sequential activation of the rest of the sequence of new assemblies. This is the essence of our first finding, stated below with quantitative bounds on plasticity and the other parameters.

Theorem 25 (Simple sequence copy). *Suppose that area A receives input from area S. Let $S_1, \ldots, S_L$ be a sequence of subsets of k neurons from S, with $L \leq n/2k$, and $|S_\sigma \cap S_{\sigma'}| \leq \Delta$ for all $\sigma \neq \sigma' \in \{1, 2, \ldots, L\}$, and Δ a positive integer. Suppose this sequence is presented T times (with each presentation beginning from rest) with area A disinhibited, forming a sequence of caps $A_1(t), \ldots, A_L(t)$ in area A on round $t \leq T$. After each round, homeostasis is applied, so that each neuron's incoming weights sum to 1. Then for*

$$kp \geq 3 \ln n, \qquad \Delta \leq \frac{k}{(2 \ln n)^2}, \qquad \beta \leq \frac{\ln \frac{n}{2kL}}{2(\max\{\Delta p, 6 \ln n\})^2}, \qquad T \geq \frac{1}{\beta \ln(n/k)}$$

when any S_i fires once in the input area, and A is allowed to fire $L - i + 1$ times, the resulting sequence of caps $\widehat{A}_i, \ldots, \widehat{A}_L$, satisfies

$$\frac{\mathbb{E}[|\widehat{A}_j \cap A_j(1)|]}{k} \geq 1 - \left(\frac{k}{n}\right)^{2\beta T}$$

for all $j \geq i$. In particular, for

$$T \geq \frac{1}{2\beta}\sqrt{\frac{\ln(nL)}{\ln(n/k)}}$$

w.h.p. we get perfect recall of the entire sequence after projection.

In other words, following T rounds of rehearsal, subsequent presentation of any input assembly in the sequence results in the activation of the corresponding assembly and the rest of the sequence in the target area. We note that the number of presentations needed grows inversely with the plasticity, and crucially, the plasticity cannot be too high.

Memorization with a scaffold. A well-known phenomenon in cognitive science is that memorization is easier by creating associations such as mnemonics. For sequences, learning one sequence by associating with another sequence, element by element (e.g., learning the alphabet to a tune), helps with retention and recall. We consider a very simple form of this: when projecting a sequence, we create two copies of the sequence that "scaffold" each other. Remarkably, this leads to provably better recall of the sequence with about half as many training rounds as simple sequence projection.

Theorem 26 (**Scaffold sequence copy**). *Suppose that areas A and B are connected to each other, and only A receives input from area S. Consider a sequence $S_1, \ldots, S_L$ of sets of k neurons in area S, for $L \leq n/2k$, which satisfy $|S_\sigma \cap S_{\sigma'}| \leq \Delta$ for all $\sigma \neq \sigma'$. Suppose this sequence is presented T times (with each beginning from rest) with areas A and B disinhibited, to form a sequence of caps $A_1(t), \ldots, A_L(t)$ within A, and $B_1(t), \ldots, B_L(t)$ within B on round t. After each round, homeostasis is applied, so that each neuron's incoming weights sum to 1. Then for*

$$kp \geq 3\ln n, \qquad \Delta \leq \frac{k}{(2\ln n)^2}, \qquad \beta \leq \frac{\ln \frac{n}{2kL}}{2(\max\{\Delta p, 6\ln n\})^2}, \qquad T \geq \frac{1}{\beta \ln(n/k))}$$

when any S_i fires once (for $1 \leq i \leq L$), and A and B are allowed to fire $L - i + 1$ times to

form a sequence of caps $\widehat{A}_i, \ldots, \widehat{A}_L$ *and* $\widehat{B}_i, \ldots, \widehat{B}_L$*, we will have*

$$\frac{\mathbb{E}[|\widehat{A}_j \cap A_j(1)|]}{k}, \frac{\mathbb{E}[|\widehat{B}_j \cap B_j(1)|]}{k} \geq 1 - \left(\frac{k}{n}\right)^{4\beta T}$$

for any $j \geq i$*. In particular, for*

$$T \geq \frac{1}{4\beta}\sqrt{\frac{\ln(nL)}{\ln(n/k)}}$$

w.h.p. we get perfect recall of the entire sequence in both areas A *and* B*.*

We note that the bound on the number of training rounds is a factor of two smaller here compared to simple sequence projection (Figure 4.6).

4.1.2 (Dis)inhibition with interneurons

Interneurons serve to inhibit and disinhibit areas. The lemma below shows that by connecting D_A to B and D_B to A, we get alternate firing of the two areas A and B. More precisely, the firing of a cap in area B causes D_A to fire, which causes I_A to cease firing, which allows a cap in A to fire in response to input from S and B (see Figure 4.2).

Time step	Firing
1	S_1, D_A, I_B
2	S_2, A_1, D_B, I_A
3	S_3, B_1, D_A, I_B
4	S_4, A_2, D_B, I_A
5	S_5, B_2, D_A, I_B
⋮	⋮

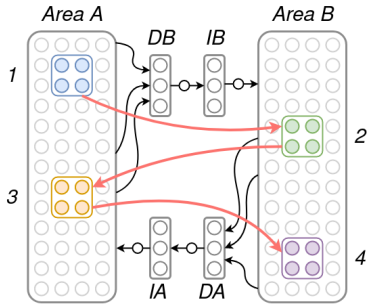


Figure 4.2: An example sequence of firings with interneurons. The table on the left shows the sets of neurons firing simultaneously on each round; on the right is the architecture of the network.

Lemma 27 (Alternation). *Let* D_A *receive input from* B *and* D_B *receive input from* A*, with* A *and* B *connected to each other and both receiving input from* S*. For any sequence*

of inputs $S_1, S_2, \ldots$ in the input area, where D_A and I_B are initially firing, the resulting sequence of activations $S'_1, S'_2 \ldots$ satisfies $S'_\sigma \subseteq A$ for σ odd and $S'_\sigma \subseteq B$ for σ even.

Proof. By induction on σ. For the base case $\sigma = 1$, as I_A is not firing initially and I_B is, only neurons in A will fire. So, $S'_1 \subseteq A$. More generally assume the claim holds for all $\sigma' < \sigma$. If σ is even, then $S'_{\sigma-1} \subseteq A$ and $S'_{\sigma-2} \subseteq B$ so D_B and I_A fired on the previous round. Hence, B is disinhibited and A is inhibited, so $S'_\sigma \subseteq B$. If σ is odd, then $S'_{\sigma-1} \subseteq B$ and $S'_{\sigma-2} \subseteq A$, so D_A and I_B fired on the previous round. Hence, A is disinhibited and A is inhibited, so $S'_\sigma \subseteq A$. □

We will make extensive use of this property in the next section.

4.1.3 Learning finite state machines

Here we demonstrate that NEMO is powerful enough to simulate an arbitrary finite state machine (FSM). In fact, an FSM is learned — that is, memorized — simply by presenting all valid transitions between states in the FSM.

Finite state machines. A finite state machine (FSM) is a tuple $F = (Q, \Sigma, q_0, q_A, q_R, \delta)$, where Q is a set of states, Σ is a finite input alphabet, $q_0 \in Q$ is the initial state, and $\delta \colon Q \times \Sigma \to Q$ is the transition function, which maps a (current state, input symbol) pair to a new state (see Figure 4.3 for an example). Given an input string $\sigma_1 \sigma_2 \ldots$ and starting from the input state q_0, at time $t \geq 1$ the FSM goes from state q_{t-1} to $q_t = \delta(q_{t-1}, \sigma_t)$.

The states $q_A, q_R \in Q$ are special terminal states, along with a special terminal character $\square \in \Sigma$ which only appears at the end of the input string. Every state $q \notin \{q_A, q_R\}$ satisfies $\delta(q, \square) \in \{q_A, q_R\}$. When the machine reaches state q_A or q_R, we say that it accepts or rejects the string, respectively.

Theorem 28. *Let $F = (Q, \Sigma, q_0, q_A, q_R, \delta)$ be an FSM. Consider a network consisting of an input area I and brain areas S and A, where I, S both have connections to and*

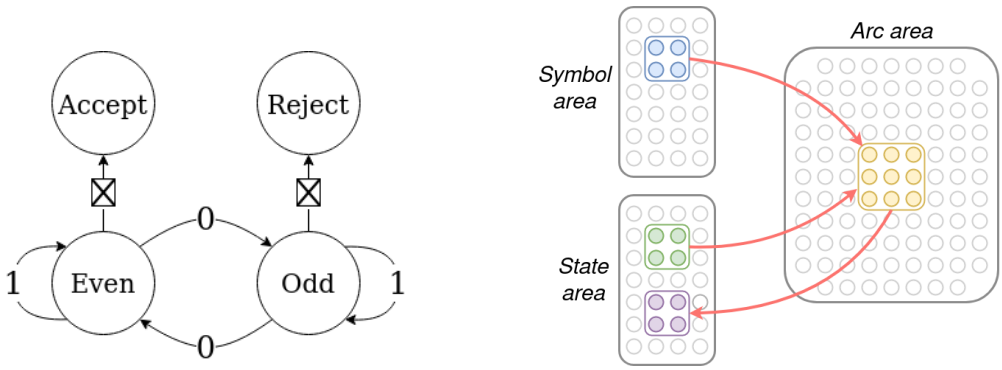


Figure 4.3: On the left is an example of a finite state machine (FSM), which consists of a finite number of states, joined by transitions (arrows). The machine changes states based on input symbols. This FSM accepts binary strings which contain an even number of zeros. On the right is the network architecture used here to simulate finite state machines. There is an assembly for each symbol, state, and transition; each pair of state and symbol assemblies projects to the associated arc assembly, which in turn projects back to the assembly corresponding to the state the FSM would switch to after seeing that state/symbol combination..

from area A, with interneurons D_S, D_A, where D_S disinhibits S upon input from A and D_A disinhibits A upon input from S. Suppose that for each $q \in Q, \sigma \in \Sigma$, there exist designated sets of k neurons $S_q \subseteq S$, $I_\sigma \subseteq I$, where $|S_q \cap S_r|, |I_\sigma \cap I_\rho| \leq \Delta = o(k)$ for all $q \neq r, \sigma \neq \rho$. Then if

$$n \geq |Q|^2|\Sigma|^2, \qquad kp \geq 6 \ln n/k, \qquad \beta \leq \frac{\ln \frac{n}{2kL}}{2(\max\{\Delta p, 6 \ln n\})^2}, \qquad T \geq \frac{12}{\beta}\sqrt{\frac{\ln n}{kp}}$$

w.h.p. this network can simulate F on any input string $\sigma_1\sigma_2 \cdots \sigma_L$ in the following sense: If S_{q_0} is made to fire on round 1, and I_{σ_i} is made to fire on round $2i - 1$, $1 \leq i \leq L$, then after $2L + 2$ rounds, S_{q_A} will fire if F accepts the string and S_{q_R} will fire if F rejects the string.

Note that by Lemma 27 if S fires on some round, A will be permitted to fire on the next, and vice versa. The crux of the simulation is to enable the configuration of synaptic weights of the network so that together, S_q and I_σ project to an assembly $A_{q,\sigma} \subseteq A$, and in turn $A_{q,\sigma}$ projects to $S_{\delta(q,\sigma)}$ (see Figure 4.3 for a schematic). As part of the proof, we

will show that firing the sequence $\{S_q, I_\sigma\}, A_{q,\sigma}, S_{\delta(q,\sigma)}$ at least $7/\beta$ times will suffice to arrange this, so the FSM simulation can actually be configured by simply observing state transitions. With this in place, if S_q and I_σ fire together, two rounds later $S_{\delta(q,\sigma)}$ will fire, simulating a single transition of the FSM. So, with S_{q_0} firing initially and I_{σ_i} firing after $2L$ rounds, S_{q_A} (resp. S_{q_R}) will fire if the FSM accepts (resp. rejects). Hence, the behavior of the FSM on any input string $\sigma_1 \ldots \sigma_L$ can be simulated in the projected FSM by setting S_{q_0} to fire initially in area S and presenting $I_{\sigma_1}, I_{\sigma_2}, \ldots$ every other time step in area I. In Figure 4.9, we show assemblies simulating the FSM from Fig. 4.3.

An illustrative FSM. In Figure 4.4, we show assemblies simulating a FSM which recognizes strings which consist of the base 10 representation of a number divisible by 3. With 3 non-terminal states and an alphabet of 10 symbols, it operates by tracking the cumulative sum of the digits modulo 3, and accepts if this modulus is 0 at the end of the string (rejecting otherwise).

Remark. A finite-state transducer (a finite state machine that produces an output symbol with each transition) with output function $\theta\colon Q \times \Sigma \to \Gamma$ can be simulated using a third area B, which contains designated sets of k neurons B_γ for each $\gamma \in \Gamma$. To accomplish this, one simply fires $B_{\theta(q,\sigma)}$ at the same time as $S_{\delta(q,\sigma)}$ during training. Then, w.h.p., $B_{\theta(q,\sigma)}$ will fire two steps after A_q and I_σ under the same conditions on T.

4.1.4 Turing Completeness

A Turing Machine (TM) is an FSM together with a tape (read-write memory). Concretely, a single-tape TM is a tuple $M = (Q, \Sigma, \{L, R\}, q_0, q_A, q_R, \delta)$ where Q is a set of states, Σ is a set of tape symbols, q_0 is an initial state, and $\delta\colon Q \times \Sigma \to Q \times \Sigma \times \{L, R\}$ is a transition function. The TM has access to a tape of symbols, which initially has written on it the input to the machine, with an infinite number of blank space symbols ($\sqcup$) extending to the left and right of the input. The tape head of the TM indicates the current symbol,

and begins at the first symbol of the input. The transition function of a Turing machine $\delta\colon Q \times \Sigma \to Q \times \Sigma \times \{L, R\}$ maps the current state and symbol indicated by the tape head to a new state, a new symbol to write, and a direction to move the tape head. The operation of the machine is as follows: At each time step, when it is in state q, it reads a symbol σ from its tape. Denote the output of the transition function as $\delta(q, \sigma) = (r, \rho, d)$. It then (i) changes state to r, (ii) replaces σ with ρ on the current tape square, and (iii) moves to the square immediately left of the current one if $d = L$ and right if $\Delta = R$. For simplicity in the simulation, we require that $\rho = \sigma$ if $d = L$, which is easily seen to maintain generality. Notably, the head of a Turing machine may be viewed as a finite state transducer which outputs symbols from the alphabet $\Sigma \times \{L, R\}$, writing symbols back to its (unbounded) input tape.

To simulate an arbitrary transition function, we augment the FSM network consisting of areas I, S, A with areas D and M, which are where movement commands and symbols to be written will be output (respectively). Areas I and M contain assemblies I_σ and M_σ, respectively, for each $\sigma \in \Sigma$, S contains an assembly S_q for each $q \in Q$, and D contains assemblies D_L and D_R. Now, as in Theorem 28, to implement a transition $\delta(q, \sigma) = (r, \rho, d)$, assemblies I_σ and S_q project to an assembly $A_{q,\sigma} \subseteq A$. In turn, $A_{q,\sigma}$ projects to S_r, M_ρ, and D_d.

What remains is to simulate the tape, namely maintain a sequence of symbols and a pointer location, and update both according to the output of the simulated transition function. We will show how to simulate a tape with assemblies, and thus (together with the FSM simulation) a Turing machine in its entirety, momentarily. The idea of the tape simulation is to maintain an assembly for each nonempty tape square, which projects to the assembly corresponding to the appropriate symbol. Each tape assembly is linked with those representing neighboring tape squares, and distributed across several brain areas configured so that when the FSM simulation issues a movement command, the assembly corresponding to the next tape square in the direction of movement will fire. To overwrite the current tape

square, a new assembly is created and its connections with its neighbor and the new symbol are strengthened.

We remark that if the model instead has access to an external "tape", we can achieve general computation with essentially just the FSM. For the environment to implement a tape, it suffices for it to contain a large number of distinguishable spaces, which can each store a representation of a symbol from Σ (imagine a stack of index cards and a writing implement). There is a pointer which indicates the current space, which is driven by the activity of the network such that it moves left or right when D_L or D_R fire in area D. If symbol σ is contained in the space indicated by the pointer, it causes I_σ to fire in area I; and if M_σ fires, symbol σ is written to the space. The system consisting of the model and its environment will implement the action of the Turing machine with transition function δ on whatever is initially written on the external tape.

Tape simulation

To simulate the tape using assemblies, we split the tape into two halves and simulate each half independently. Each half-tape supports two operations: One operation adds a symbol to the beginning of the tape, while the other removes the current symbol so that the succeeding symbol is at the beginning. We implement this tape with assemblies as follows.

Intuitively, the tape is represented by a sequence of assemblies which cycle between the three areas, with the current position of the beginning of the tape represented by the currently firing assembly (see Figure 4.5). To remove the current symbol, the next area in the cycle is disinhibited, which causes the next assembly in the chain to fire (and the previous assembly is effectively forgotten); to add a new symbol, a new assembly is created in the preceding area, linked to the current assembly. The symbol at each position of the tape is stored by strengthened connections between the tape assemblies and symbol assemblies in a designated symbol area, which will fire when the associated tape assemblies do. Precisely, the simulation consists of maintaining a set of assemblies $A_1(t), \ldots, A_{L_t}(t)$

for $1 \leq t \leq T$ with the following properties:

(i) If the t'th operation adds σ to the beginning of the tape, with $A_1(t) \subseteq H_i$, we will have $A_1(t+1) \subseteq H_{i-1}$, and $Aj+1(t+1) = A_j(t-1)$ for all $1 \leq j \leq L_t$.

(ii) If the t'th operation removes σ from the beginning of the tape, we will have $A_i(t+1) = A_{i+1}(t)$ for all $1 \leq i \leq L_t - 1$.

A "delete" operation when assembly $A_1(t)$ in area H_i is firing will simply cause H_{i+1} to fire, and in turn strengthened connections from $A_1(t)$ to $A_2(t)$ will cause $A_2(t)$ to begin firing. An "add" operation is more complicated; with H_{i-1} disinhibited, input from outside of the tape areas will drive creation of a new assembly $A_1(t+1)$, which is linked to $A_1(t) = A_2(t+1)$ and S_σ (where σ is the symbol to be added to the tape) by their simultaneous firing. Thus, we require an lower bound on the plasticity, so that $A_1(t+1)$ can be formed quickly, as well as an upper bound on the plasticity, to ensure that existing assemblies in area H_{i-1} will not overlap $A_1(t+1)$ by too much. Three areas are required so that $A_1(t+1)$ becomes linked only in the forward direction to $A_1(t)$, and not in the backwards direction (i.e. in the future when $A_1(t+1)$ fires it will trigger $A_1(t) = A_2(t+1)$ to fire, but not vice versa).

Lemma 29. *[Tape Simulation] Consider three brain areas H_1, H_2, H_3, each with n neurons, where H_i gives input to H_{i+1} (addition modulo 3) and all give input to an area S. The firing of H_i is governed by interneurons D_i, which receive input from a control area C containing assemblies C_{add} and C_{delete}. Suppose the behavior of D_i is as follows: D_i permits H_i to fire T times when either C_{add} fires together with H_i or H_{i+1}, OR C_{delete} and H_{i-1} fire together. Moreover, there are two areas E_1, E_2, connected via interneurons which disinhibit E_1 for T rounds after E_2 fires T times, and disinhibit E_2 for T rounds after E_1 fires. Suppose that S initially contains assemblies S_σ for each $\sigma \in \Sigma$, with $|S_\sigma \cap S_\rho| \leq 6 \ln n$ for all $\sigma \neq \rho$. Let E be a set of neurons containing a set of k-caps $Z_1, \ldots, Z_L$, where $|Z_t \cap Z_s| \leq 6 \ln n$ for all $t \neq s$, and Z_t fires on rounds $(t-1)T + 1$ through tT. Assume*

that

$$\sqrt{\frac{\ln n}{kp}} \leq \beta \leq \frac{\ln \frac{n}{2kK}}{72 \ln^2 n}$$

and $kp \geq 3 \ln n$. *Suppose there is a sequence of "add" and "delete" operations in the following sense: If the* t*th operation adds symbol* σ*, then* C_{add} *and* S_σ *are made to fire by external control on rounds* $(t-1)T + 1$ *through* tT*, while if the* t*th operation deletes the current symbol, then* C_{delete} *fires on rounds* $(t-1)T + 1$ *through* tT*. For a sequence of length* $L \leq \frac{n}{2k}$*,* NEMO *can simulate this sequence of operations: WHP, if* σ *is the current symbol on round* t*, then* S_σ *will fire on round* $tT + 1$.

Now, by combining simulations of each tape half, we can simulate the entire tape. The symbol under the tape head is represented by the symbol at the beginning of the right half of the tape. A rightward movement of the TM is simulated by removing the first symbol from the right half of the tape, and adding the (potentially changed) symbol to the left half of the tape. A leftward movement is simulated by removing the first symbol from the left half and adding it to the right half (recall that the symbol does not change in a leftward movement). As a consequence, NEMO is Turing complete.

The simulation consists of six areas $H_i^L, H_i^R, i = 1, 2, 3$, three each for the "left" and "right" halves of the tape, a control area D for both halves jointly, and the three areas I, S, A for the FSM, which yields a total of 10 areas ($2 \times 3 + 1 + 3$). The outline of its operation is as follows: The FSM simulation proceeds as in Theorem 4, modified so that if $\delta(q, \sigma) = (p, \rho, d)$, the assembly $A_{q,\sigma}$ causes S_p, I_ρ, and D_d to fire, where $d \in \{L, R\}$. The tape halves are simulated as in Lemma 29, where the firing of D_L signals a "delete" operation for the left tape and an "add" operation for the right tape (and vice versa). The new assembly on the right half becomes linked to I_ρ by their concurrent firing, which effects a leftward movement of the tape head (and similarly for D_R).

Theorem 30. *[TM Simulation] With plasticity*

$$\sqrt{\frac{\ln n}{kp}} \leq \beta \leq \frac{\ln \frac{n}{2kT}}{72 \log^2 n}$$

and $kp \geq 3 \ln n$*, w.h.p.* NEMO *can simulate a Turing machine* $M = (Q, \Sigma, \{L, R\}, q_0, q_A, q_R, \delta)$ *which uses* T *time in time* $O(T)$ *using ten brain areas, each of size* $n \geq 2 \max\{T^2, |Q|^2|\Sigma|^2\}$.

4.2 Experiments

To support our theoretical results, we have simulated the sequence and FSM models extensively. We detail the results and conclusions of these various experiments here.

Number of presentations versus recall. In Figure 4.6 we show how recall of a sequence improves over the course of repeated presentations, for both the single-area and scaffolded models. Significantly, the faster rate of memorization observed for the scaffolded model supports the bound given in Theorem 26. Here, we estimate the assembly corresponding to a given element in the sequence as the set of neurons which fire on the last round of training, and measure recall as the fraction of neurons in that assembly which fire at the appropriate time when only the first element of the sequence is presented.

Sequence memorization capacity. In Figure 4.7, we increase the length of the sequence to be memorized and measure both the recall of the sequence and the maximum overlap between assemblies corresponding to different sequence elements. This disambiguates two failure modes of sequence learning, where either different neurons fire during testing versus training or the assemblies representing different sequence elements become indistinguishable. Notably, the capacity of a given brain area seems to greatly exceed the bound we give in Theorem 25, and indeed even surpasses the quantity n/k which is the largest number of disjoint k-caps which can be formed in a given area. This suggests that sequence learning in NEMO is significantly more robust than our analysis implies.

Effect of parameters on sequence memorization. We examine the effect of area size (Figure 4.8, left) and edge density (Figure 4.8, right) on the success of sequence memorization. In each case, we vary the relevant parameter while training the model to memorize a length 25 sequence, and measure the fraction of neurons in the last assembly of the sequence which fire at the appropriate time when only the first element of the sequence is presented (*recall*), and the largest fraction of overlap between any two assemblies corresponding to distinct elements (*max overlap*). High recall indicates that the neural responses to the sequence of stimuli have been memorized, while low max overlap indicates that the assemblies corresponding to any pair of distinct sequence elements are distinguishable. Our analytical results (where the probability of success increases with n, and we require the product kp to be sufficiently large) predict that increasing either parameters should improve performance (i.e. increase recall and decrease overlap) which is indeed observed experimentally. Notably, reasonably good performance is attained for substantially larger ranges of n and p than we require in the theorems.

FSM memorization. In Figure 4.9 we demonstrate how the performance of each transition of the FSM improves over the course of training, and how performance degrades as the size of the FSM (i.e. total number of transitions) increases, while all parameters of the model are fixed. We measure performance by the recall averaged over all transitions, where here the recall is the fraction of neurons in the appropriate next state assembly which fire two rounds after the assemblies corresponding to a given state/symbol pair are made to fire.

Effect of parameters on FSM memorization. We examine the effect of area size (Figure 4.10, left) and edge density (Figure 4.10, right) on the success of FSM memorization. Using the FSM simulated in Figure 4 (see Figure 4.4 for a diagram), we vary the relevant parameter while training the model to memorize each transition of this FSM. We measure performance through the fraction of neurons in the appropriate next state assembly which fire two rounds after the assemblies corresponding to a given state/symbol pair are made

to fire (*recall*), with the minimum taken over all transitions, and the fraction of a random sample of length 20 input strings which are correctly accepted or rejected (*classification accuracy*), in the sense that the state assembly with the most neurons firing on the last round corresponds to the correct terminal state. Our analytical results (where again the probability of success increases with n, and we require the product kp to be sufficiently large) predict that increasing either n or p should improve performance (i.e. increase both recall and classification accuracy) which is observed experimentally. Notably, classification accuracy reaches a perfect score much sooner than recall.

Effect of input string length on FSM simulation. We examine the effect of the length of the input string on the accuracy of the simulation of the FSM on it (Figure 4.11). For a few choices of the key parameters (area size n and density p), we train the model to memorize each transition of the FSM in Figure 4.4 and then test its performance on a random sample of input strings of various sizes. As above, we measure performance via *classification accuracy*, the fraction of strings which are correctly accepted or rejected (i.e. the state assembly with the most neurons firing on the last round corresponds to the correct terminal state). For n and p such that all transitions are performed accurately, we expect that classification accuracy should be very high even for long strings, while for smaller n and p classification accuracy should decay with string length. This is largely supported by the simulations, where classification accuracy remains nearly perfect for all string lengths for the larger choices of n and p and decays somewhat for the smaller choices, although notably even for long strings the classification accuracy is still substantially higher than chance (which would be 0.2 with the 5 states of the FSM).

4.3 Proofs

Proof sketches. All of our proofs rely on a few basic ingredients. A key observation underpinning all of them is that in the range of parameters assumed, the neurons which

activate in response to a given sequence element/transition do not change over the course of training w.h.p. Intuitively, if the weights change sufficiently slowly, then repeated presentations of the same stimulus should activate the same set of neurons. We then show that the overlap of assemblies corresponding to distinct sequence elements/transitions will not be too large, which will allow them to be distinguished and ensures that increasing weights between one pair of assemblies will not interfere with others. Finally, we bound the number of rounds of training needed to ensure that recall occurs successfully, which simply involves comparing the input a neuron in the correct assembly will receive versus a neuron outside of the correct assembly, and choosing the weight to be sufficiently high so that the neurons in the correct assembly have the highest input w.h.p.

To proceed with the proofs in full, we first need a few crucial lemmas. Given a pair of inputs to a brain area, Lemma 31 provides an upper bound (which depends on the overlap of these inputs) on the overlap of the sets of neurons which fire in response to them, when plasticity is in effect.

Lemma 31. *Let $\delta > 0$. Let I_1, I_2 be sets of neurons providing input to brain area A. (Note that I_1, I_2 might intersect A.) Suppose that I_1 fires, forming a cap C_1, with weights from I_1 to C_1 increased by $1 + \gamma$. Some time later I_2 fires, forming a cap C_2. Suppose that $|I_1|, |I_2| \geq k$, and that $kp \geq 6 \ln \frac{n}{k}$. Then if $|I_1 \cap I_2| = 0$, we have*

$$\Pr\left[|C_1 \cap C_2| \geq 2\max\{\tfrac{k^2}{n}, 3\ln\tfrac{1}{\delta}\}\right] \leq \delta$$

and otherwise, for $\epsilon \geq k/n$ and

$$\gamma \leq \frac{\sqrt{3|I_2|p\ln\frac{n}{k}} - \max\{\sqrt{3|I_1 \cap I_2|p\ln\frac{2n}{\delta}}, 3\ln\frac{2n}{\delta}\} - \sqrt{3|I_2 \setminus I_1|p\ln\frac{2}{\epsilon}}}{2\max\{|I_1 \cap I_2|p, 3\ln\frac{2n}{\delta}\}}$$

we have

$$\Pr\left[|C_1 \cap C_2| \geq \max\left\{\epsilon k, 6\ln\tfrac{2}{\delta}\right\}\right] \leq \delta$$

Lemma 32 controls the number of caps a given neuron will enter in response to a sequence of stimuli with bounded overlap.

Lemma 32. *Let $I_1, \ldots, I_L$ be a set of inputs to a brain area A (which might intersect A) which are presented in sequence, satisfying $|I_1| \geq k, |I_t| = m \geq k$ for all $t > 1$, $|I_s \cap I_t| \leq \Delta$ for all $s \neq t$, and*

$$L \leq 3 \ln n \left(\frac{n}{k}\right)^{(1-\sqrt{\frac{\Delta \ln n}{m \ln \frac{n}{k}}})^2}$$

Let C_t denote the cap denoting from I_t, and $\mathbb{1}_t$ denote the indicator vector for C_t. Then for

$$\beta \leq \frac{(1 - \sqrt{\frac{\Delta \ln n}{m \ln \frac{n}{k}}})^2 \ln \frac{n}{k} - \ln L - \frac{1}{3}}{2(\max\{\Delta p, 6 \ln n\})^2}$$

and $kp \geq 3 \ln n$ we have

$$\sum_{t=1}^{L} \mathbb{1}_t(i) < 6 \ln n$$

w.h.p. for all $i \in A$.

Lemma 33 provides a criterion to ensure that the same set of neurons will fire in response to the same stimulus.

Lemma 33. *Let $\delta > 0$, and let I be a set of neurons providing input to brain area A with $|I|p \geq 3 \ln \frac{1}{\delta}$. Let $C \subseteq A$ be a set of k neurons, where each neuron in C has at least d incoming edges from I, each strengthened by a factor of $1 + \gamma$. Suppose no neuron outside of C has more than m incoming connections from C which have been strengthened, each by a factor no more than $1 + c\gamma$. Then for*

$$\gamma \geq \frac{|I|p + \sqrt{3|I|p \ln \frac{n}{\delta}} - d}{d - cm}$$

if I fires, the resulting cap C' will equal C w.p. $1 - \delta$

Lemma 34 upper bounds the overlap of the caps for a pair of stimuli, at a larger overlap than Lemma 31.

Lemma 34. *Let I_1 and I_2 be two sets of neurons providing input to area A, with $|I_1| = |I_2| = 2k$ and $|I_1 \cap I_2| \leq (1 + o(1))k$, and let A_1, A_2 denote their respective caps. Then*

$$|A_1 \cap A_2| \leq \frac{k}{2}$$

w.p. $1 - o(n^{-1})$.

Finally, Lemma 35 is a fact about the fixed points of a certain class of exponential functions.

Lemma 35. *Let $f(x) = \exp(-a(b(1-x)^2 - 1))$ for $a > 0, b \geq 1 + 2a^{-1}$. Then there exists some $x^* \in \mathbb{R}$ such that $f(x^*) = x^*$ and moreover*

$$x^* \leq \exp(-a(b-1))$$

We can now proceed with the proofs of the theorems.

Proof of Theorem 25. We will first show that on the first presentation, the overlap between the sets of neurons which fire for different elements of the sequence will not be too large for sufficiently small plasticity. Using this fact, we can then show that the sequence of neurons which fire will remain the same over repeated presentations of the stimulus sequence. Finally, this second claim makes it simple to show that after enough presentations the sequence of neurons will be strongly linked enough that when only the first cap fires, the rest of the sequence will as well without external stimulus.

We will prove the first claim by induction on σ. The base case $\sigma = 1$ holds vacuously. For $\sigma \geq 1$, we note that if $|A_\rho(1) \cap A_\sigma(1)| \leq 6 \ln \frac{n}{k}$ for all $\rho < \sigma$, then the input to $A_{\sigma+1}(1)$ ($S_{\sigma+1}$ and $A_\sigma(1)$) overlaps the input to any other set $A_\rho(1)$ by at most $\Delta + 6 \ln \frac{n}{k}$. As

weights are increased by $1+\beta$ and

$$\beta \leq \frac{\ln \frac{n}{kL} - \frac{1}{3}}{2(\max\{\Delta p, 6\ln n\})^2} \leq \frac{\sqrt{6kp\ln\frac{n}{k}} - \max\{\sqrt{6\Delta p \ln Ln}, 6\ln nL\} - \sqrt{6kp\ln k}}{2\max\{\Delta p, 6\ln\frac{n}{k}\}}$$

by Lemma 31 we have $|A_{\sigma+1}(1) \cap A_\rho(1)| \leq 6\ln\frac{n}{k}$ w.p. $1-o(L^{-2})$ as well. A union bound over all $O(L^2)$ events shows that the probability that $|A_\sigma(1) \cap A_\rho(1)| > 6\ln\frac{n}{k}$ for any pair $\rho \neq \sigma$ is $o(1)$.

We will now show that $A_\sigma(t) = A_\sigma(1)$ for all σ, by first proving that $A_\sigma(2) = A_\sigma(1)$ via induction on σ. First note that, by Lemma 32, no neuron in A makes more than $6\ln n$ caps $A_1(1), \ldots, A_L(1)$ w.h.p. Let $X_i(t,\sigma)$ denote the input to i on round t during presentation of S_σ. For the base case $\sigma = 1$, for $i \in A_1(1)$, we have

$$e(S_1, i) \geq kp + \sqrt{3kp\ln\frac{n}{k}}$$

and so its input is

$$X_i(2,1) \geq \frac{(1+\beta)(kp + \sqrt{3kp\ln\frac{n}{k}})}{2np + 2\beta kp}$$

while for $j \notin A_1(1)$,

$$\begin{aligned} X_j(2,1) &\leq \frac{kp + \sqrt{3kp\ln\frac{n}{k}} + (1+\beta)^{6\ln n}(\Delta p + \max\{\Delta p, 3\ln n\})}{2np + (1+\beta)^{6\ln n}(\Delta p + \max\{\Delta p, 3\ln n\})} \\ &\leq \frac{kp + \sqrt{3kp\ln\frac{n}{k}} + 2(1+\beta)^{6\ln n}\max\{\Delta p, 3\ln n\}}{2np + (1+\beta)^{6\ln n}(\Delta p + \max\{\Delta p, 3\ln n\})} \end{aligned}$$

Observe that for

$$\beta \leq\leq \frac{\ln \frac{n}{kL} - \frac{1}{3}}{2(\max\{\Delta p, 6\ln n\})^2} \leq \frac{1}{6\ln n}$$

we have

$$X_j(2,1) \leq \frac{(1+\beta)(kp + \sqrt{3kp\ln\frac{n}{k}})}{2np + 2\beta kp}$$

in which case $X_i(2,1) > X_j(2,1)$ for all $i \in A_1(1), j \notin A_1(1)$.

Now assume that $\sigma > 1$ and $A_\rho(2) = A_\rho(1)$ for all $\rho < \sigma$. For $i \in A_\sigma(1)$,

$$e(S_\sigma \cup A_{\sigma-1}(1), i) \geq 2kp + \sqrt{6kp \ln \frac{n}{k}}$$

and so

$$X_i(2, \sigma) \geq \frac{(1+\beta)(2kp + \sqrt{6kp \ln \frac{n}{k}})}{np + \beta 2kp + \sqrt{6kp \ln \frac{n}{k}}}$$

On the other hand, for all $j \notin A_\sigma(1)$, we have

$$X_j(2, \sigma) \leq \frac{2kp + \sqrt{6kp \ln \frac{n}{k}} + 4(1+\beta)^{6 \ln n} \max\{\Delta p, 3 \ln n\}}{np + 4(1+\beta)^{6 \ln n} \max\{\Delta p, 3 \ln n\}}$$

For

$$\beta \leq \frac{\ln \frac{n}{kL} - \frac{1}{3}}{2(\max\{\Delta p, 6 \ln n\})^2} \leq \frac{1}{12 \ln n}$$

we have

$$X_j(2, \sigma) \leq \frac{(1+\beta)(2kp + \sqrt{6kp \ln \frac{n}{k}})}{np + \beta 2kp + \sqrt{6kp \ln \frac{n}{k}}}$$

and so $X_i(2, \sigma) > X_j(2, \sigma)$ for all $i \in A_\sigma(1), j \notin A_\sigma(1)$, which means $A_\sigma(2) = A_\sigma(1)$.

Now, we use double induction on t and σ, with $t = 2$ as the base case. Let $X_i(\sigma, t)$ denote the input to neuron i when S_σ fires on round t. Suppose that $A_\rho(s) = A_\rho(1)$ for all ρ and $s < t$ and $A_\rho(t) = A_\rho(1)$ for all $\rho < \sigma$. For $i \in A_\sigma(1)$, we have

$$X_i(t, \sigma) \geq \frac{(1+\beta) X_i(t-1, \sigma)}{2np + \beta X_i(t-1, \sigma)}$$

For $j \notin A_\sigma(1)$,

$$X_j(t, \sigma) \leq \frac{X_j(t-1, \sigma) + (e^{6\beta \ln n} - 1)(X_j(t-1, \sigma) - 2kp - \sqrt{6kp \ln \frac{n}{k}})}{2np + (e^{6\beta \ln n} - 1)(X_j(t-1, \sigma) - 2kp - \sqrt{6kp \ln \frac{n}{k}})}$$

For

$$\beta \leq \frac{\ln \frac{n}{kL} - \frac{1}{3}}{2(\max\{\Delta p, 6 \ln n\})^2} \leq \frac{(1 - \sqrt{\frac{\Delta \ln n}{k \ln \frac{n}{k}}})^2 \ln \frac{n}{k} - \ln L - \frac{1}{3}}{2(\max\{\Delta p, 6 \ln n\})^2}$$

and $\Delta \leq k/\ln^2 n$, we have $X_j(t,\sigma) \leq X_i(t,\sigma)$ and hence, $A_\sigma(t) = A_\sigma(1)$.

Now, we will prove the last claim. For $\sigma = 1$, the argument is the same as during training since S_1 still fires, so we have $\widehat{A}_\sigma = A_\sigma(1)$. Suppose that $|\widehat{A}_{\sigma-1} \cap A_{\sigma-1}(1)| = (1-\epsilon)k$. Then for $i \in A_\sigma(1)$,

$$X_i(\sigma) \geq \frac{(1-\epsilon)(1+\beta)^T(kp + \sqrt{\frac{3}{2}kp\ln\frac{n}{k}}) + \epsilon kp}{((1+\beta)^T - 1)(kp + \sqrt{3kp\ln\frac{n}{k}}) + 2np} = \tau_\epsilon$$

while for $j \notin A_\sigma(1)$,

$$X_j(\sigma) \sim \frac{\mathcal{B}(k,p)}{2np}$$

Hence,

$$\begin{aligned}
\Pr[j \in \widehat{A}_\sigma \mid j \notin A_\sigma(1)] &\leq \Pr\left[X_j(\sigma) \geq \tau_\epsilon\right] \\
&\leq \exp\left(-\frac{1}{3kp}\left((1-\epsilon)\beta T(kp + \sqrt{\frac{3}{2}kp\ln\frac{n}{k}}) + (1-\epsilon)\sqrt{\frac{3}{2}kp\ln\frac{n}{k}}\right)^2\right) \\
&\leq \exp\left(-\frac{1}{2}(1-2\epsilon)(1+2\beta T)^2\ln\frac{n}{k}\right)
\end{aligned}$$

The fraction of newcomers is hence, in expectation, no more than

$$\frac{n}{k}\exp\left(-\frac{1}{2}(1-2\epsilon)(1+2\beta T)^2\ln\frac{n}{k}\right)$$

We seek a fixed point ϵ^* of this function. By Lemma 35, we have

$$\epsilon^* \leq \left(\frac{k}{n}\right)^{2\beta T}$$

as long as

$$T \geq \frac{1}{\beta\ln\frac{n}{k}}$$

Now, for perfect recall, suppose that

$$T \geq \frac{1}{\beta}\sqrt{\frac{\ln nL}{2\ln \frac{n}{k}}}$$

Then

$$\exp\left(-\frac{1}{2}(1+2\beta T)^2 \ln \frac{n}{k}\right) \leq \frac{1}{nL}$$

Then a union bound over vertices and $1 \leq \sigma \leq L$ shows that the probability $\widehat{A}_\sigma \neq A_\sigma(1)$ for some σ is $o(1)$. □

Proof of Theorem 26. The argument proceeds similarly as in the proof of Theorem 1: We bound the overlap of different caps during the first presentation, show that the sequence of caps will remain the same over subsequent presentations, and show that after enough presentations the sequence of caps will be recalled without external input after the first round. Crucially, a larger fraction of the input to the correct neurons in the main area will still fire in the absence of external stimuli, which allows memorization to occur more quickly than in the absence of the auxiliary area.

We will prove the first claim by induction on σ. The base case $\sigma = 1$ holds vacuously. For $\sigma \geq 1$, we note that if $|A_\rho(1) \cap A_\sigma(1)| \leq 6\ln\frac{n}{k}$ for all $\rho < \sigma$, and $|B_\rho(1) \cap B_\sigma(1)| \leq 6\frac{n}{k}$ for all $\rho < \sigma - 1$, then the input to $A_{\sigma+1}(1)$ ($S_{\sigma+1}$, $A_\sigma(1)$, and $B_{\sigma-1}(1)$) overlaps the input to any other set $A_\rho(1)$ by at most $\Delta + 12\ln\frac{n}{k}$. As weights are increased by $1+\beta$ and

$$\beta \leq \frac{\ln\frac{n}{kL} - \frac{1}{3}}{2(\max\{\Delta p, 6\ln n\})^2} \leq \frac{\sqrt{9kp\ln\frac{n}{k}} - \max\{\sqrt{9\Delta p \ln Ln}, 3\ln nL\} - \sqrt{9kp\ln k}}{3\max\{\Delta p, 6\ln\frac{n}{k}\}}$$

by Lemma 31 we have $|A_{\sigma+1}(1) \cap A_\rho(1)| \leq 6\ln\frac{n}{k}$ w.p. $1 - o(L^{-2})$ as well. The argument proceeds similarly for $B_\sigma(1)$. Then a union bound over all $O(L^2)$ events shows that the probability that $|A_\rho(1) \cap A_\sigma(1)|$ or $|B_\rho(1) \cap B_\rho(1)| > 6\ln\frac{n}{k}$ for any pair $\sigma \neq \rho$ is $o(1)$.

We will now show that $A_\sigma(t) = A_\sigma(1)$ and $B_\sigma(t) = B_\sigma(1)$ for all σ. We note that, by Theorem 1, $B_\sigma(t) = B_\sigma(1)$ for all σ and t w.h.p. as long as $A_\rho(s) = A_\rho(1)$ for all pairs

(ρ, s) where either $s < t$ or $\rho < \sigma$. So, it will suffice to only show that $A_\sigma(s) = A_\sigma(1)$, which we will accomplish by first proving that $A_\sigma(2) = A_\sigma(1)$ via induction on σ. First note that, by Lemma 32, no neuron in A makes more than $6 \ln n$ caps $A_1(1), \ldots, A_L(1)$. Let $X_i(t, \sigma)$ denote the input to i on round t during presentation of S_σ. For the base case $\sigma = 1$, for $i \in A_1(1)$, we have

$$e(S_1, i) \geq kp + \sqrt{3kp \ln \frac{n}{k}}$$

and so its input is

$$X_i(2, 1) \geq \frac{(1+\beta)(kp + \sqrt{3kp \ln \frac{n}{k}})}{2np + 2\beta kp}$$

while for $j \in A \setminus A_1(1)$,

$$\begin{aligned} X_j(2, 1) &\leq \frac{kp + \sqrt{3kp \ln \frac{n}{k}} + (1+\beta)^{6 \ln n}(\Delta p + \max\{\Delta p, 3 \ln n\})}{2np + (1+\beta)^{6 \ln n}(\Delta p + \max\{\Delta p, 3 \ln n\})} \\ &\leq \frac{kp + \sqrt{3kp \ln \frac{n}{k}} + 2(1+\beta)^{6 \ln n} \max\{\Delta p, 3 \ln n\}}{2np + (1+\beta)^{6 \ln n}(\Delta p + \max\{\Delta p, 3 \ln n\})} \end{aligned}$$

Observe that for

$$\beta \leq\leq \frac{\ln \frac{n}{kL} - \frac{1}{3}}{2(\max\{\Delta p, 6 \ln n\})^2} \leq \frac{1}{6 \ln n}$$

we have

$$X_j(2, 1) \leq \frac{(1+\beta)(kp + \sqrt{3kp \ln \frac{n}{k}})}{2np + 2\beta kp}$$

in which case $X_i(2, 1) > X_j(2, 1)$ for all $i \in A_1(1), j \in A \setminus A_1(1)$. Hence, $A_2(1) = A_1(1)$.

Now let $\sigma > 1$, and assume that $A_\rho(2) = A_\rho(1)$ and $B_{\rho-1}(2) = B_{\rho-1}(1)$ for all $\rho < \sigma$. For $i \in B_{\sigma-1}(1)$,

$$e(A_{\sigma-1} \cup B_{\sigma-2}(1), i) \geq 2kp + \sqrt{6kp \ln \frac{n}{k}}$$

and so

$$X_i(2, \sigma) \geq \frac{(1+\beta)(2kp + \sqrt{6kp \ln \frac{n}{k}})}{2np + \beta 2kp + \sqrt{6kp \ln \frac{n}{k}}}$$

On the other hand, for all $j \in B \setminus B_{\sigma-1}(1)$, we have

$$X_j(2,\sigma) \leq \frac{2kp + \sqrt{6kp \ln \frac{n}{k}} + 12(1+\beta)^{6\ln n} \ln \frac{n}{k}}{2np + 12(1+\beta)^{6\ln n} \ln n}$$

For

$$\beta \leq \frac{\ln \frac{n}{kL} - \frac{1}{3}}{2(\max\{\Delta p, 6\ln n\})^2} \leq \frac{1}{12\ln n}$$

, we have

$$X_j(2,\sigma-1) \leq \frac{(1+\beta)(2kp + \sqrt{6kp\ln\frac{n}{k}})}{2np + \beta 2kp + \sqrt{6kp\ln\frac{n}{k}}}$$

and so $X_i(2,\sigma) > X_j(2,\sigma)$ for all $i \in B_{\sigma-1}(1), j \in B \setminus B_{\sigma-1}(1)$, which means $B_{\sigma-1}(2) = B_{\sigma-1}(1)$ Then, for $i \in A_\sigma(1)$,

$$e(S_\sigma \cup A_\sigma(1) \cup B_{\sigma-1}(1), i) \geq 3kp + \sqrt{9kp\ln\frac{n}{k}}$$

and so

$$X_i(2,\sigma) \geq \frac{(1+\beta)(3kp + \sqrt{9kp\ln\frac{n}{k}})}{3np + \beta 3kp + \sqrt{9kp\ln\frac{n}{k}}}$$

On the other hand, for all $j \in A \setminus A_\sigma(1)$, we have

$$X_j(2,\sigma) \leq \frac{3kp + \sqrt{6kp \ln \frac{n}{k}} + 12(1+\beta)^{6\ln n} \ln \frac{n}{k}}{3np + 12(1+\beta)^{6\ln n} \ln n}$$

For

$$\beta \leq \frac{\ln \frac{n}{kL} - \frac{1}{3}}{2(\max\{\Delta p, 6\ln n\})^2} \leq \frac{1}{12\ln n}$$

we have

$$X_j(2,\sigma) \leq \frac{(1+\beta)(3kp + \sqrt{9kp\ln\frac{n}{k}})}{3np + \beta 3kp + \sqrt{9kp\ln\frac{n}{k}}}$$

and so $X_i(2,\sigma) > X_j(2,\sigma)$ for all $i \in A_\sigma(1), j \in A \setminus A_\sigma(1)$, so $A_\sigma(2) = A_\sigma(1)$. Then using induction, we have $A_\sigma(2) = A_\sigma(1)$ and $B_\sigma(2) = B_\sigma(1)$ for all $1 \leq \sigma \leq L$.

Now, we use double induction on t and σ, with $t = 2$ as the base case. Let $X_i(t,\sigma)$

denote the input to neuron i when S_σ fires on round t. Suppose that $A_\rho(s) = A_\rho(1)$ and $B_{\rho-1}(s) = B_{\rho-1}(1)$ for all ρ and $s < t$ and $A_\rho(t) = A_\rho(1)$ and $B_{\rho-1}(t) = B_{\rho-1}(1)$ for all $\rho < \sigma$. For $i \in B_{\sigma-1}(1)$, we have

$$X_i(t,\sigma) \geq \frac{(1+\beta)X_i(t-1,\sigma)}{2np + \beta X_i(t-1,\sigma)}$$

For $j \in B \setminus B_{\sigma-1}(1)$,

$$X_j(t,\sigma) \leq \frac{X_j(t-1,\sigma) + (e^{6\beta \ln n} - 1)(X_j(t) - 2kp - \sqrt{6kp \ln \frac{n}{k}})}{2np + (e^{6\beta \ln n} - 1)(X_j(t-1,\sigma) - 2kp - \sqrt{6kp \ln \frac{n}{k}})}$$

For

$$\beta \leq \frac{\ln \frac{n}{kL} - \frac{1}{3}}{2(\max\{\Delta p, 6 \ln n\})^2} \leq \frac{(1 - \sqrt{\frac{\Delta \ln n}{k \ln \frac{n}{k}}})^2 \ln \frac{n}{k} - \ln L - \frac{1}{3}}{2(\max\{\Delta p, 6 \ln n\})^2}$$

and $\Delta \leq k/\ln^2 n$, we have $X_j(t,\sigma) \leq X_i(t,\sigma)$ and hence, $B_{\sigma-1}(t) = B_{\sigma-1}(1)$.

Now, we will prove the last claim. For $\sigma = 1$, the argument is the same as during training since S_1 still fires, so we have $\widehat{A}_\sigma = A_\sigma(1)$. Suppose that $|\widehat{A}_{\sigma-1} \cap A_{\sigma-1}(1)| = (1-\epsilon)k$. Then for $i \in A_\sigma(1)$,

$$X_i(\sigma) \geq \frac{(1-\epsilon)(1+\beta)^T(kp + \sqrt{\frac{3}{2}kp \ln \frac{n}{k}}) + \epsilon kp}{((1+\beta)^T - 1)(kp + \sqrt{3kp \ln \frac{n}{k}}) + 2np} = \tau_\epsilon$$

while for $j \notin A_\sigma(1)$,

$$X_j(\sigma) \sim \frac{\mathcal{B}(k,p)}{2np}$$

Hence,

$$\begin{aligned}\Pr[j \in \widehat{A}_\sigma \mid j \notin A_\sigma(1)] &\leq \Pr\left[X_j(\sigma) \geq \tau_\epsilon\right] \\ &\leq \exp\left(-\frac{1}{3kp}\left((1-\epsilon)\beta T(kp + \sqrt{\frac{3}{2}kp\ln\frac{n}{k}}) + (1-\epsilon)\sqrt{\frac{3}{2}kp\ln\frac{n}{k}}\right)^2\right) \\ &\leq \exp\left(-\frac{1}{2}(1-2\epsilon)(1+2\beta T)^2\ln\frac{n}{k}\right)\end{aligned}$$

The fraction of newcomers is hence, in expectation, no more than

$$\frac{n}{k}\exp\left(-\frac{1}{2}(1-2\epsilon)(1+2\beta T)^2\ln\frac{n}{k}\right)$$

We seek a fixed point ϵ^* of this function. By Lemma 35, we have

$$\epsilon^* \leq \left(\frac{k}{n}\right)^{2\beta T}$$

as long as

$$T \geq \frac{1}{\beta\ln\frac{n}{k}}$$

Now, for perfect recall, suppose that

$$T \geq \frac{1}{\beta}\sqrt{\frac{\ln nL}{2\ln\frac{n}{k}}}$$

Then

$$\exp\left(-\frac{1}{2}(1+2\beta T)^2\ln\frac{n}{k}\right) \leq \frac{1}{nL}$$

Then a union bound over vertices and $1 \leq \sigma \leq L$ shows that the probability $\widehat{A}_\sigma \neq A_\sigma(1)$ for some σ is $o(1)$. □

Proof of Theorem 28. We first bound the overlap of the arc assemblies associated with different transitions. We then show that after sufficiently many presentations of each transi-

tions, the connections between pairs of state and symbol assemblies and their associated arc assemblies, and between arc assemblies and the correct next state assemblies, will be sufficiently strengthened so that firing any state/symbol pair will cause the appropriate arc assembly, and then in turn the appropriate next state assembly, to fire in their entirety. Once this property is established for every transition, it follows immediately that the FSM can be simulated on an arbitrary input string.

Let $A_{q,\sigma} \subseteq A$ denote the set of neurons which fires in response to $S_q \cup I_\sigma$. We need to show that after T training presentations, if S_q and I_σ fire, then $S_{\delta(q,\sigma)}$ will fire two time steps later w.p. $1 - o(|Q|^{-1}|\Sigma|^{-1})$.

We will first bound the overlap of the sets $A_{q,\sigma}$. Suppose that $(q, \sigma) \neq (q', \sigma')$. Then

$$|(S_q \cup I_\sigma) \cap (S_{q'} \cup I_{\sigma'})| \leq k + \Delta$$

With $\Delta = o(k)$, by Lemma 34 we have $|A_{q,\sigma} \cap A_{q',\sigma'}| \leq k/2$ w.p. at least $1 - o(n^2)$. Then a union bound over all $O(|Q||\Sigma|) = O(n)$ pairs $(q, \sigma) \neq (q', \sigma')$ shows that the bound holds for every pair w.p. $1 - o(1)$.

Now, after training, it is clear that $A_{q,\sigma}$ will fire in response to S_q and I_σ. It remains to ensure that all of $S_{\delta(q,\sigma)}$ will fire. With probability $1 - o(1)$, we have

$$e(A_{q,\sigma}, i) \geq kp - \sqrt{2kp \ln n}$$

for all $i \in S_{\delta(q,\sigma)}$ and $q \in Q, \sigma \in \Sigma$. On the other hand, for any $j \in S$,

$$e(A_{q,\sigma} \cap A_{q',\sigma'}, j) \leq \frac{kp}{2} + \sqrt{\frac{3}{2} kp \ln n}$$

Let

$$X_j = e(A_{q,\sigma} \setminus A_{q',\sigma'}, j)$$

and note that X_j is the sum of i.i.d. Bernoulli random variables with $\mathbb{E}X_j \leq kp/2$. Let $S'_{q,\sigma}$

denote the cap which fires in response to $A_{q,\sigma}$. Then

$$\Pr(j \in S'_{q,\sigma} \mid j \notin S_{\delta(q,\sigma)}) \leq \Pr\left(X_j \geq (1+\beta)^T \left(\frac{kp}{2} - \sqrt{\frac{7}{2}kp\ln n}\right)\right)$$
$$\leq \exp\left(-\frac{2(T\beta\frac{kp}{2} - (1+T\beta)\sqrt{\frac{7}{2}kp\ln n})^2}{3kp}\right)$$

For

$$T \geq \frac{12}{\beta}\sqrt{\frac{\ln n}{kp}}$$

we have

$$\Pr(j \in S'_{q,\sigma} \mid j \notin S_{\delta(q,\sigma)}) \leq \frac{1}{n^2}$$

and hence via a union bound we will have $S'_{q,\sigma} = S_{\delta(q,\sigma)}$ for every $q \in Q, \sigma \in \Sigma$ w.p. $1 - o(1)$. □

Proof of Lemma 29. Suppose the content of the tape after L operations is $\sigma_1, \ldots, \sigma_{\tilde{L}}$. We will show that with probability $1 - o(1)$, a set $A_1, \ldots, A_{\tilde{L}}$ of assemblies will be created, where $A_1 \subseteq H_i$ and $A_t \subseteq H_{i+t \pmod 3}$. This set has the properties that for any $1 \leq t \leq \tilde{L}$, (i) if A_t fires then S_{σ_t} will fire on the next round, and (ii) if A_t fires and $H_{i+t+1 \pmod 3}$ is disinhibited then A_{t+1} will fire on the next round.

We proceed by induction on L, the number of operations. Immediately, if the $(L+1)$th operation is "delete", then after T rounds $H_{i+1 \pmod 3}$ will be disinhibited, and by the inductive hypothesis then $A_2, \ldots, A_{\tilde{L}}$ is a set of assemblies with the required property. So, we consider only "add" operations henceforth. We will show that each "add" operation creates an assembly A' which will fire A_1 and the associated symbol assembly with probability $1 - o(L^{-1})$. As there are at most L operations, a union bound over all operations will show the entire sequence succeeds w.p. $1 - o(1)$.

For the first "add" operation, there is no next symbol, so the set of neurons A_1 which fires randomly only needs to be linked to the correct symbol, S_{σ_1}. By Theorem 3 of Pa-

padimitriou & Vempala [94], for $\beta \geq \sqrt{\ln n/kp}$ A_1 will stabilize after Z_1 fires at least

$$T' = \frac{1}{\beta}\frac{\ln k}{\sqrt{kp}}$$

times. Every neuron in S_{σ_1} will have at least $kp - \sqrt{2kp \ln n}$ connections from A_1, while no neuron outside has more than $kp + \sqrt{3kp \ln n}$. Hence, it suffices that

$$(1+\beta)^{T-T'}\left(kp - \sqrt{2kp \ln n}\right) \geq kp + \sqrt{3kp \ln n}$$

For

$$T \geq 5\frac{\sqrt{\ln n}}{\beta}$$

this bound holds.

Now, suppose that the first t operations were successfully simulated, for $t \geq 1$. Hence, on rounds $tT, tT+1, \ldots, tT+T-1$, some area (say H_i) fired, while C_{delete} will fire on round $tT+T-1$. For a "delete" operation, this implies immediately that H_{i+1} will begin to fire on round $(t+1)T$ and continue firing through round $(t+1)T+T-1$. Moreover, to ensure that the correct set of neurons fires (say $A_{t+1} \subset H_{i+1}$) on round $(t+1)T$, we need that every neuron of A_{t+1} receives more input than every neuron in $H_{i+1} \setminus A_{t+1}$. Once again, $T \geq k + 10/\beta$ will ensure this holds. Assuming A_{t+1} fires, every neuron of $S_{\sigma_{t+1}}$ has its incoming weights from A_{t+1} strengthened by $(1+\beta)^{T-\ln k}$, so $S_{\sigma_{t+1}}$ will also fire under the same conditions on T.

On the other hand, suppose the $(t+1)$'th operation adds a symbol to the beginning of the tape. By induction, some area (say H_i) fired on rounds $tT, tT+1, \ldots, (t+1)T-1$, while C_{add} fires on round $(t+1)T-1$. So, H_{i-1} will begin firing on $(t+1)T$ (along with H_i, which will continue to fire) and continue firing until $(t+2)T-1$. Simultaneously, on rounds $(t+1)T, (t+1)T+1, \ldots, (t+1)T+T-1$ the assemblies Z_{t+1}, $S_{\sigma_{t+1}}$ fire together.

Let A_{t+1} denote the set of neurons that fire in response to stimulus from $E_1 \cup E_2$. Then for

$$\beta T \leq \frac{\sqrt{3k} - \sqrt{36 \ln n}}{12\sqrt{\ln np}}$$

by Lemma 31 w.p. $o(L^{-1})$, we will have $|A_{t+1} \cap A_s| \leq 6 \ln n$. By Lemma 32, no neuron enters more than $6 \ln n$ assemblies.

It remains to show that the correct assembly $A_t \subseteq H_i$ will continue to fire for the next T rounds. As shown above, this will be sufficient to ensure that when A_{t+1} is firing and H_i is disinhibited, that A_t will begin firing. A neuron in A_t has fired alongside all other neurons in A_t at least T times, so its input is at least

$$(3 + \beta T)\left(kp - \sqrt{2kp \ln n}\right)$$

On the other hand any neuron outside of A_t receives at most

$$3kp + \sqrt{9kp \ln n} + 72\beta T \ln^2 n$$

input from A_{t+1}, A_t, and Z_{t+1} combined w.p. $1 - o(L^{-1})$. We then only need that

$$\beta T\left(kp - \sqrt{2kp \ln n} - 72 \ln^2 np\right) \geq \sqrt{9kp \ln n}$$

As for $kp \geq 6 \ln n$ we have

$$5\frac{\sqrt{\ln n}}{\beta} \geq \frac{1}{\beta} \frac{\sqrt{9kp \ln n}}{kp - \sqrt{2kp \ln n} - 12 \ln^2 np}$$

this is satisfied.

□

Proof of Theorem 30. The simulation consists of three areas H_i^L, H_i^R for each of the "right" and "left" halves of the tape, a control area D for both halves of the tape jointly, and

the three areas I, S, A for the FSM. The outline of its operation is as follows: The FSM simulation proceeds as in Theorem 4, modified so that if $\delta(q, \sigma) = (p, \rho, d)$, the assembly $A_{q,\sigma}$ causes S_p, I_ρ, and D_d to fire, where $d \in \{L, R\}$. The interneurons of the FSM simulation now allow their respective areas to fire after T rounds, where T is the number of repetitions required by the tape simulation. The tape halves are configured as in Lemma 29, where the firing of D_L signals a deletion for the left half and a addition of the top symbol from the left half for the right stack. The new assembly on the right half becomes linked to I_σ by their concurrent firing, which effects a leftward movement of the tape head (and similarly for D_R).

By Theorem 4, the FSM simulation in areas I, S, A will simulate the transition function of the Turing machine w.h.p., while by Lemma 29 each of the stack simulations will perform as desired w.h.p. which simulates the TM's tape. We then have three events which each occur w.h.p., so their intersection does as well. □

4.4 Discussion

NEMO is a simple mathematical model of the brain involving random connectivity, Hebbian plasticity, local inhibition, and long-range interneurons, which provably captures key aspects of cognition involving sequences. Here we demonstrated that sequences can be memorized and copied in various modes, and furthermore that, through sequences, our model can learn to recognize patterns modeled by finite state machines; a more sophisticated use of sequences of assemblies is capable of universal computation. While this last point is of course primarily of theoretical interest, it is a rather useful point to make about the power of NEMO as a model of brain computation. Brain-like computation can happen with no explicit control flow or commands — that is to say, no *program.* All that is needed is biologically plausible neural hardware and the presentation of stimuli.

A striking difference in the behavior of the brain and silicon computers is in their mechanisms for the creation and recall of memories. This is the starting point of the assembly

model, continued here. A very interesting and unexpected discovery is rigorous mathematical evidence for the emergence of the utility of mnemonics, memory palaces and other memory aids [132], a uniquely brain-centered phenomenon that has no analogy for computers.

Although we have explored how the brain might encode and memorize sequences here, a crucial gap remains: How the brain generates sequences. Augmenting our FSM simulation with probabilistic transitions would allow it to generate random sequences, effectively capturing the brain's ability to sample from a *probabilistic* finite automaton; this approach could be further generalized to allow sampling from a graphical model. How probabilistic transitions might be realized in NEMO, and moreover learned from streams of stimuli, is an important and urgent future direction for this work, and would be the basis of emergent statistical computation in the brain.

Finally, we must note that, in our discussion of the way in which sequences are memorized and recalled in the brain, we have not yet mentioned *language,* arguably the most remarkable faculty related to sequences that human brains enjoy; see Mitropolsky et al. [95, 97] for work on assemblies and language.

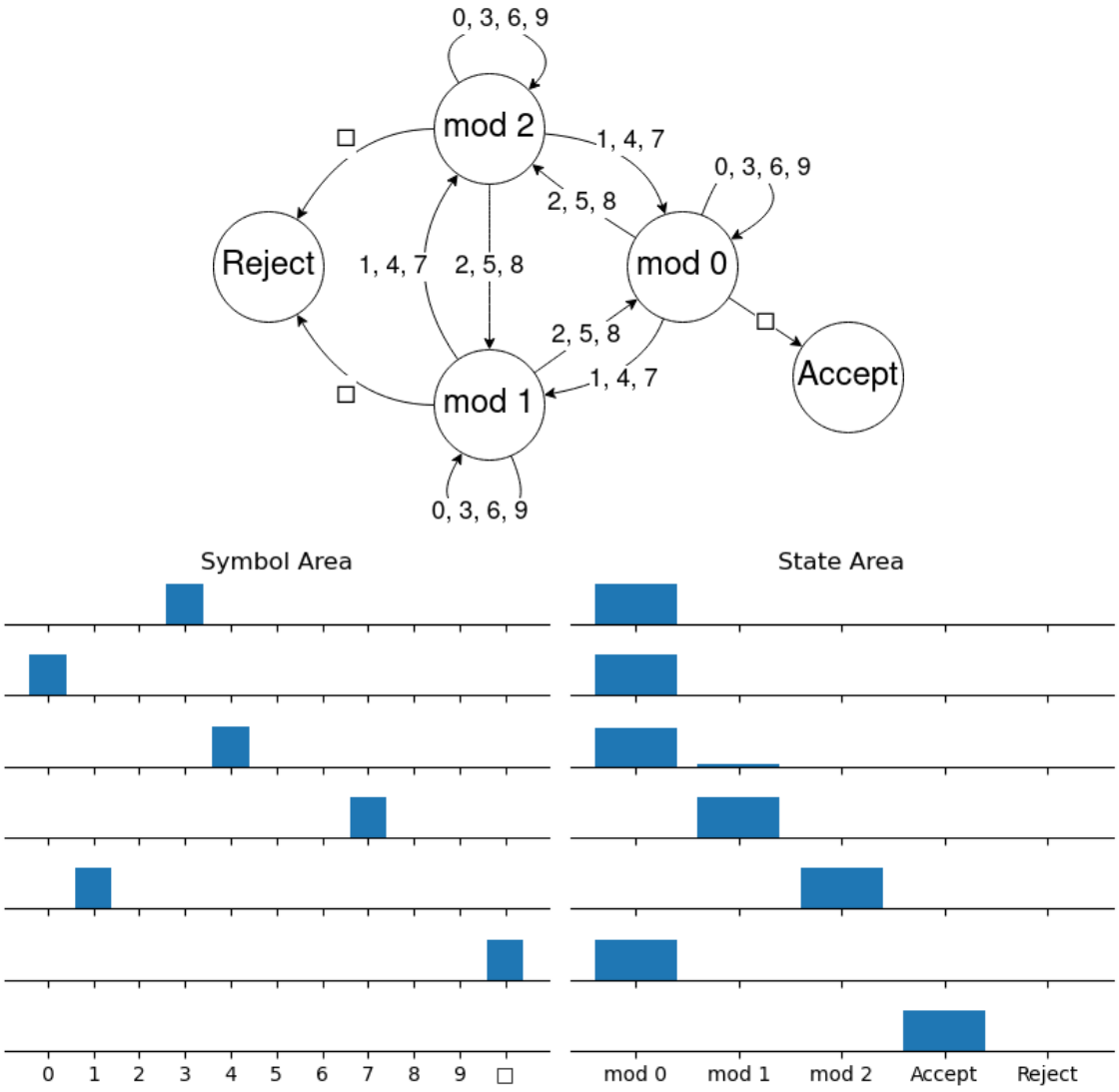


Figure 4.4: Above is the schematic of a FSM which accepts numbers in base 10 which are multiples of 3. The non-terminal states of the machine correspond to the remainder of the sum of digits which have been seen; the FSM accepts if the remainder at the end of the string is 0, and rejects otherwise. This FSM has a total of 5 states, 11 symbols, and 33 transitions. In the lower plot, we display the fraction of overlap between the responses of the trained model and the assemblies representing states and symbols, while simulating this FSM on the test string `30471`. Here, $n = 5000, k = 70, p = 0.4, \beta = 0.1$, and the model was trained with 15 presentations of each transition.

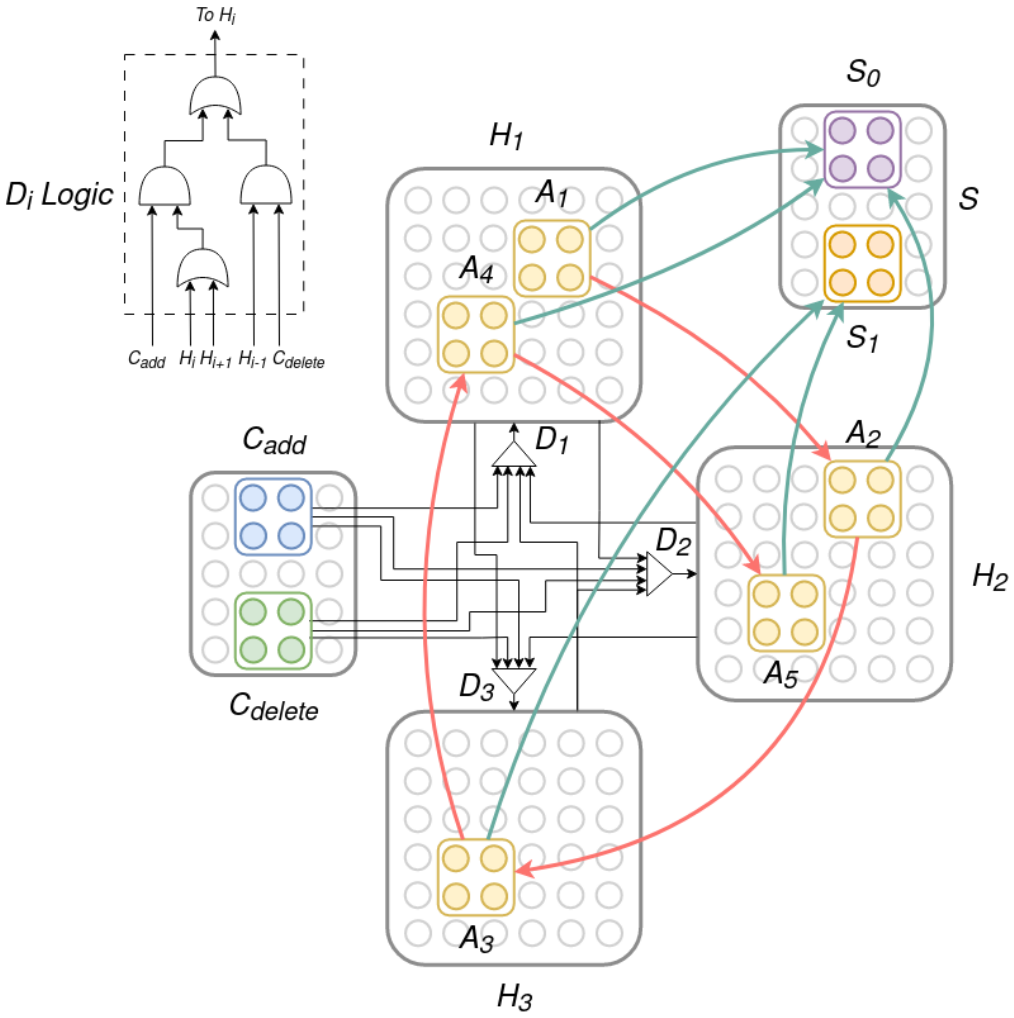


Figure 4.5: Simulating a restricted tape with assemblies. Assemblies $A_1, \ldots, A_5$ each represent a space on the tape, and are placed to cycle between areas H_1, H_2, H_3 (red arrows). C_{add} and C_{delete} are externally-activated control assemblies, which signal operations to areas H_1, H_2, H_3 and via interneuron populations D_1, D_2, D_3 determine the sequence in which they are disinhibited. On the upper left we provide a schematic of the logical function implemented by D_i, which carries out the inhibition and disinhibition of H_i to simulate the operations signalled by the control assemblies. Black arrows in the larger diagram correspond to inputs to (i.e. synapses of) the interneurons, D_1, D_2, D_3. To represent symbol σ being written at space i, assembly A_i is linked to assembly S_σ (teal connections) so that the firing of A_i will cause S_σ to fire. At this point in the simulation, the simulated tape has the string `00101` written on it.

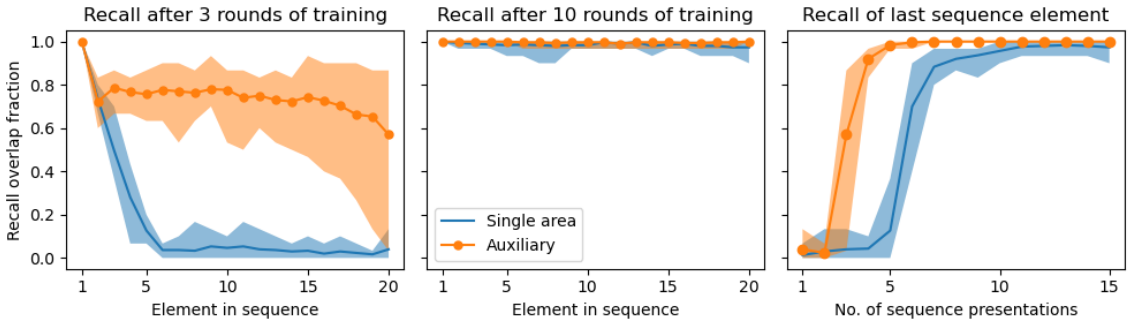


Figure 4.6: Simple vs scaffold sequence memorization: we examine recall of the sequence in response to presentation of only the first item after 3 (left) and 10 (center) presentations of the sequence, by measuring the fraction of neurons in the assembly corresponding to a given element of the sequence which fire during testing. On the right, we demonstrate how the recall of the last element of the sequence (again, after only the first is presented) over the course of training. Here, $n = 1000, k = 30, p = 0.2, \beta = 0.1$, and the sequence is length 20. Dark center line is the mean over 10 trials, while shaded area is the range.

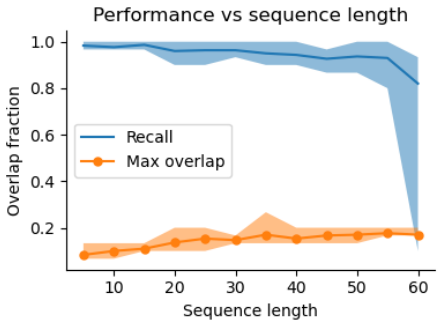


Figure 4.7: For each choice of sequence length, we train the model via repeated presentations of a sequence of the given length. In blue we plot the recall of the last element in the sequence, and in orange we plot the maximum overlap of assemblies corresponding to distinct sequence elements. Here, $n = 1000, k = 30, p = 0.2, \beta = 0.1$, and each sequence is trained via 10 presentations. Dark center line is the mean over 10 trials, while shaded area is the range.

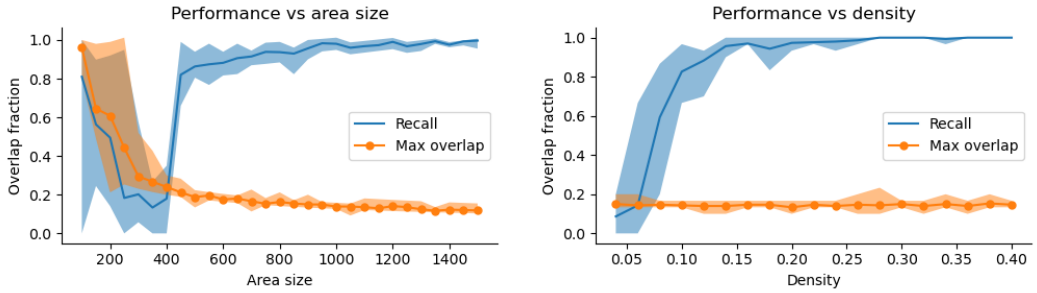
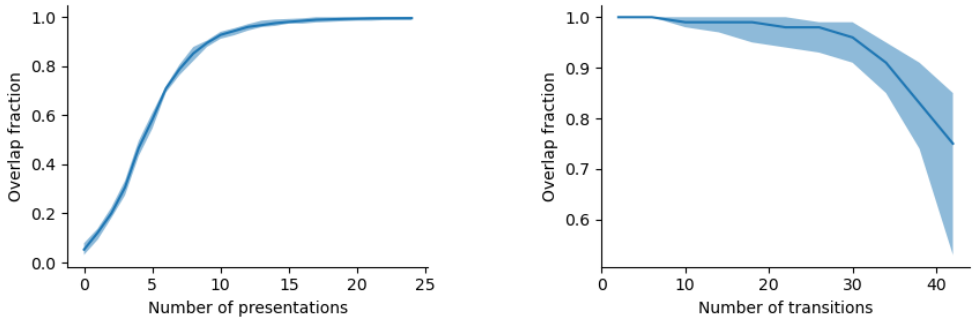


Figure 4.8: We measure performance of sequence memorization as two key parameters of the model (area size and density of connections) are varied. On the left, we vary n from 100 to 1500 while $k = \sqrt{n}$, $p = 0.2$, $\beta = 0.1$. On the right, we vary p from 0.04 to 0.4 while $n = 1000, k = 30, \beta = 0.1$. In both cases the sequence to be memorized is length 25 and training consists of 10 presentations. Dark center line is the average over 10 trials, while shaded area is the range.



(a) Recall over training the FSM

(b) Performance versus size of FSM

Figure 4.9: In (a), we train the model by repeatedly presenting each transition of the FSM in Figure 4.3 and measure performance after a given number of presentations. In (b), we train the model using FSMs of different sizes and again measure the recall. Here, $n = 5000, k = 70, p = 0.4, \beta = 0.1$ and the model is trained with 15 presentations of each transition. Dark center line indicates mean over 10 trials; shaded area indicates the range.

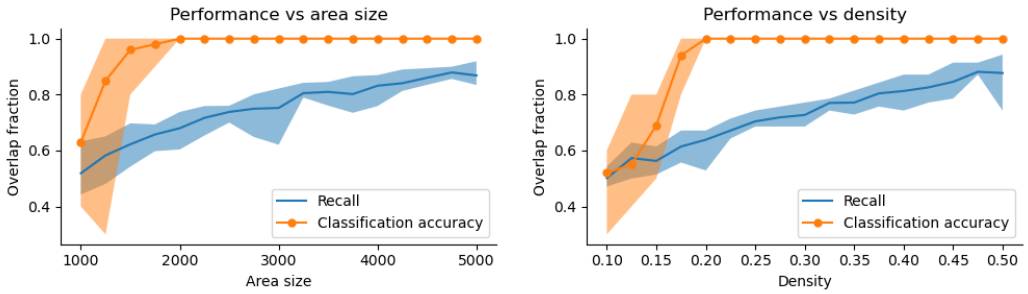


Figure 4.10: We measure performance of FSM memorization as two key parameters of the model (area size and density of connections) are varied. On the left, we vary n from 1000 to 5000 while $k = \sqrt{n}$, $p = 0.5, \beta = 0.1$. On the right, we vary p from 0.1 to 0.5 while $n = 5000, k = 70, \beta = 0.1$. In both cases the FSM is given in Figure 4.4 (with 11 symbols and 3 non-terminal states) and training consists of 15 presentations of each transition. Dark center line is the average over 10 trials, while shaded area is the range.

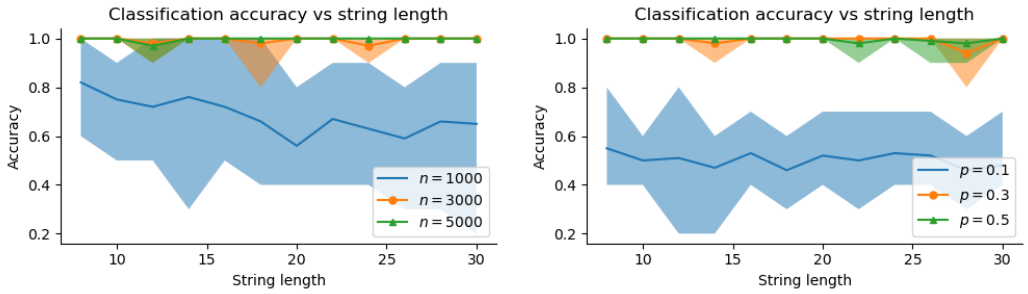


Figure 4.11: We measure performance of FSM simulation as the length of the input string increases, for a few choices of area size (left) and edge density (right). On the left, $k = \sqrt{n}, p = 0.5, \beta = 0.1$; on the right, $n = 5000, k = 70, \beta = 0.1$. In both cases the FSM is given in Figure 4.4 (with 11 symbols and 3 non-terminal states) and training consists of 15 presentations of each transition. Dark center line is the average over 10 trials, while shaded area is the range.

CHAPTER 5

FLIPPING A COIN IN THE BRAIN

Any attempt to explain how intelligence arises from the brain must address the organ's remarkable statistical capabilities — extracting, computing with, and reproducing statistical structure in its environment. A century of work and an enormous body of literature in psychology has quantified these capacities in humans and other animals, across many domains [133, 134, 135]. Although the mechanisms underlying statistical learning remain poorly understood [136], they appear to be subconscious [137, 138], and at least partially modality-specific [139]. This raises two questions: What is the most basic level on which statistical learning can be understood to occur — the neuron, the circuit, or the entire brain? Second, what are the mechanisms (ideally simple and general ones) which implement statistical learning?

In neuroscience, the brain is commonly thought to be inherently noisy at many levels, from the stochasticity of developmental processes [140] and synaptic transmission [141, 142] all the way to the response variability of neurons to identical patterns of stimuli [143, 144]. Various proposals for modeling the computation performed by the brain have postulated that this stochasticity serves a purpose, from improving the magnitude of neural responses [145] to boosting their robustness [146]. Of particular interest in the context of statistical learning are the many approaches which frame variability as the result, or enabler, of the brain performing *sampling* [147, 148, 149, 150, 151, 152].

In this work, we propose a stylized neuronal mechanism which leads to learning the statistical properties of stimuli, in the sense that internally-generated randomness leads to sampling from the stimulus distributions. In our model sampling emerges at the level of assemblies: groups of densely-interconnected neurons, which, through their simultaneous firing, encode cognitively-relevant items. Here, the encoded item may be thought of

as an outcome of a random variable, and the probability the assembly fires approximates the observed frequency of that event. The ingredients for this mechanism are simple and eminently biologically plausible: random connectivity, inhibition-induced competition, a Hebbian plasticity rule, and noisy neuronal activations are sufficient. Our model is a slight extension toNEMO [96], a biologically plausible model of the brain in which assemblies of neurons rapidly emerge, and general computation can be carried out by presentation of stimuli [153].

Assembly-level sampling occurs as follows (Theorem 36): A set of neurons subject to a k-winners-take-all dynamic —NEMO's way of modeling local inhibition — gives rise to densely-intraconnected assemblies. When these neurons are excited by some external stimulus, the activation of each neuron is perturbed by random noise. Due to inhibition, only a subset of these neurons fire, which in general includes neurons from several assemblies. The random perturbation ensures that one assembly is quite likely to enjoy a slight numerical advantage over the others, and the dense intraconnectivity of the assemblies ensures that this initial leader quickly dominates and fires alone. Put differently, assemblies may be viewed as the attractors of the dynamical system, and the effect of randomness is to choose an initial state distributed among these various basins of attraction. In the language of probability, these assemblies represent outcomes of a random variable, while the external stimulus represents a conditional event (e.g. the known value of another random variable). The distribution over outcomes is encoded by the weights from the external stimulus (Corollary 37), so that an appropriate Hebbian plasticity rule makes it possible to learn the distribution from observing samples (Lemma 38). This is the fundamental building block of our implementation of statistical learning, by which one assembly can be sampled out of a set of possibilities, according to a conditional distribution which depends on external activity and is learned from experience.

Going further, we show in experiment and in theory how this building block naturally leads to the neural implementation of more complex probabilistic models, such as Markov

chains (Corollary 39) and, as a structured extension thereof, a trigram model of language. This generalizes previous work in theNEMO model regarding the memorization of *deterministic* sequences [153]. The number of assemblies used in these models is proportional to the number of states in the Markov chain, which means that Markov chains can be implemented highly efficiently in terms of neural "memory". As a demonstration, we show that an assembly-based model can extract and reproduce the lower-order statistics of a small natural language corpus, *with no specialized instructions*.

Taken together, our results have several implications. The first is a concrete proposal for how probabilistic computation (and statistical learning more specifically) can be performed by the brain on the level of firing neurons and synapses, which to our knowledge is the first to do so at the level of groups of neurons, or assemblies. Hence, this proposal bolsters the hypothesis that assemblies are the fundamental encoding strategy of the brain, as a sparse population code that supports statistical learning. Moreover, we make precise predictions about the functional role of plasticity rules and brain rhythms. We also contribute to a line of theoretical work [147, 151, 152] positing a key computational role for noise in the brain.

5.1 Background

Plasticity rules. In our theoeretical results and simulations we focus on plasticity rules of the form $w(t+1) - w(t) = \pi(w(t)) = \min\{\alpha, e^{\lambda(1+\beta-w(t))}\}$, where α, β, and λ are the parameters of the rule. See Lemma 38 for a derivation of how this family captures *softmax*. Qualitatively, this rule provides a strong increase to the synaptic weight the first few times neurons fire in sequence, which rapidly tapers off with repeated presentations. In Figure 5.1, we plot this rule and the resulting weights for various parameter choices. Although it does not impose a hard ceiling on the synaptic weights, this rule bears some similarities to other Hebbian plasticity rules which taper off as the weight increases, such as Oja's [154], remarkable for its ability to perform principle component analysis.

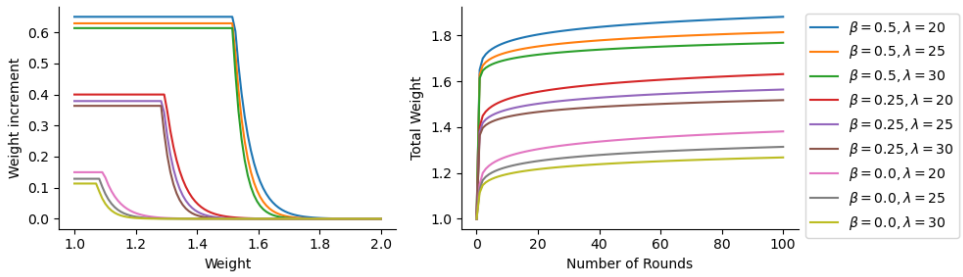


Figure 5.1: On the left we exhibit the plasticity increment, $\min\{\alpha, e^{\lambda(1+\beta-w(t))}\}$, as a function of the weight; on the right is the total weight versus the number of rounds of training. Here, $\alpha = \beta + \log\lambda/\lambda$ as we assume in all of our proofs and experiments.

5.1.1 The Statistical Brain

A wide body of work in psychology and cognitive science has provided evidence for the sensitivity of humans and other animals to the statistics of their environments. A line of foundational studies [155, 156, 157, 158] examined the ability of humans to estimate the proportion of different stimuli in a stream presented to them; the overall consensus is that humans are surprisingly good at this task, especially when the classes of stimuli are all relatively common. Beyond humans, the tendency of an animal to adjust its foraging strategy such that the amount of time it spends at various feeding sites matches the amount of food or probability of receiving food at those sites is exceptionally well-documented, across clades ranging from fish to passerine birds to rats [159, 160, 161]. In the study of animal behavior this is known as the "ideal free distribution", as the animal has *ideal* (complete) knowledge of the distribution of food and *free* movement. Although counterintuitive, since the optimal strategy is to always choose the most rewarding site, the ideal free distribution is believed to be more stable in a competitive environment [162].

Humans and other animals have also demonstrated awareness of higher-order statistics, i.e., the frequency of co-occurrence of stimuli. The landmark study of Saffran et al. [135] in language learning showed that adult humans distinguished between frequently and less-frequently heard pairs of syllables, which was later replicated in infants [163] and tamarin monkeys [164]. Other studies showed similar capabilities in the visual system [165, 138].

Another line of work uses MEG/EEG measures of surprise to gauge subject's responses to sequences of stimuli. Various probabilistic structures have been examined: Furl et al. [166] generated the sequence from a Markov chain, while others considered triplets (i.e. trigrams) of stimuli [167, 168, 169]. After exposure to the sequence, subjects are reliably more surprised by less-probable transitions.

In contrast to this rich body of behavioral work, the neural basis of statistical learning is poorly understood. Various studies have found that processing of statistical structure is correlated with activity in higher-level, sensory modality-specific areas, in both speech [170, 171, 172] and vision [173, 174]. In combination with behavioral studies which show that individual subjects performing statistical learning exhibit different biases [175], uncorrelated performance [176], and the tendency not to generalize statistical structure across different sensory modalities [177], it has been postulated that statistical learning is implemented by mechanisms local to various cortical regions [139]. On the other hand, activity in the brain's general memory systems, especially the hippocampus, is also associated with statistical learning [178, 179], although since subjects with damage to these areas still display statistical learning capabilities [180] it is unclear to what extent they are functionally involved in these processes (see Batterink et al. [136] for a more detailed discussion).

5.1.2 Related Work

NEMO models the brain as a dynamical system arising from a small set of biologically well-grounded mechanisms.NEMO contrasts with Valiant's neuroidal model [80, 82] in which individual neurons are capable of arbitrary state changes; as well as with attempts to implement backpropagation by neural circuitry [90], which rely on hypothetical connectivity and highly-precise coordination; and also with Willshaw and Hopfield network models, which are high-level models of associative memory which deviate from the dynamics and capabilities of the brain.NEMO emphasizes emergent neuronal coordination; assemblies arise to represent external stimuli [94] through the interplay of Hebbian plas-

ticity with inhibition-induced competition. A related model, with connectivity parameterized by a geometric random graph (as opposed to Erdös-Renyi) has been shown to display assembly-like behavior even without plasticity [181]. Later work within NEMO has studied other algorithms that emerge from its dynamics, include learning linear classifiers [98] and memorization of temporal sequences of activations [153], while more engineered constructions have been given for natural language [95, 97, 182] and implementing various automata, including the Turing machine [153].

The brain as a probabilistic model. The observation that many of the problems the brain solves have a probabilistic or statistical character is integral to cognitive science [183, 184] and there have been many attempts to frame computational problems in terms of probabilistic inference [185, 186, 187]. Among neural network models that implement probabilistic inference, the best known are Boltzmann machines [188], which may be thought of as a probabilistic version of the Hopfield network. The weights of a Boltzmann machine parameterize a restricted class of distributions on $\{0,1\}^n$, which can be sampled from using Gibbs sampling (a Markov chain Monte Carlo algorithm). These models drew interest in neuroscience since they can be trained using a biologically-plausible synapse-local algorithm, even though the traditional sampling algorithms do not bear much resemblance to the dynamics of the brain. Later work by Buesing et al. [152] gave a spiking neural network implementation which addresses the latter issue while retaining the biologically-plausible training algorithm. This model inherits two weaknesses from the Boltzmann machine as a model of the brain. The first is that random variables are represented by single spiking units, which requires high reliability of individual neurons and contradicts the widely accepted population coding hypothesis [189]. Recent work in has shown the possibility of forming multi-neuron representations of concepts in spiking neural networks [190], but it remains unclear whether similar ideas apply to the Boltzmann machine implementation. The second is that distributions outside of the class of Boltzmann distributions (i.e. exactly

those that are parameterized by the weights of a Boltzmann machine) require deeper networks; how the brain trains its deep networks and what functions it can learn through them remain major open questions.

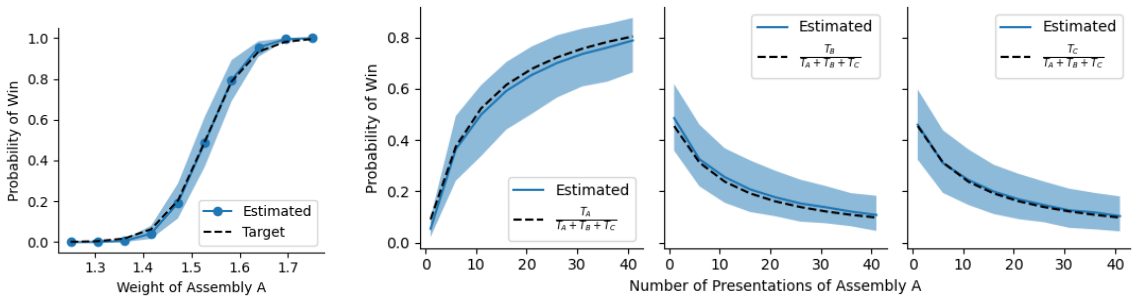
5.2 Computational Experiments

5.2.1 The assembly coin-flip

We first consider the situation where weights from some input assembly I to assemblies $A_1, \ldots, A_m$ in area S are increased (Figure 5.7). Then I fires, while the activations of neurons in S are perturbed by Gaussian noise; in subsequent rounds, only S fires. A preliminary observation is that if any of $A_1, \ldots, A_m$ has sufficiently large weight from I, then after a few rounds of firing the set of firing neurons converges to one of $A_1, \ldots, A_m$. In every experiment in this section, we have $n = 25000, k = 500, p = 0.1$, and the standard deviation of the noise is $5\sqrt{kp}$.

In our first experiment (Figure 5.2(a)), we fix a sample of the random graph, and measure how the probability that the firing neurons converge to A_1 changes as the weight from I to A_1 is varied, while weights from I to A_2 and A_3 are fixed (here at 2). The internal weights of A_1, A_2, A_3 are set to 2 as well. We repeat the process for 1000 realizations of the random noise, and compute the empirical frequency that A_1 fires. We also plot the softmax function of the weights, where the rate parameter λ is chosen empirically to minimize the error between the softmax and the empirical frequency (averaged over graphs). Qualitatively and empirically, this demonstrates that the probability distribution over assemblies is approximately a softmax of the incoming weights to the assemblies (see also Theorem 36).

In the previous experiment, we directly set the synaptic weights between the input and output assemblies, and studied the resulting probability distribution (of the firing neurons converging to an output assembly). In other words, this shows how *sample* a random variable, assuming its distribution has somehow already been learned and "stored" in the synaptic weights. However, the same neural representations of the outcomes of random variable



(a) (b)

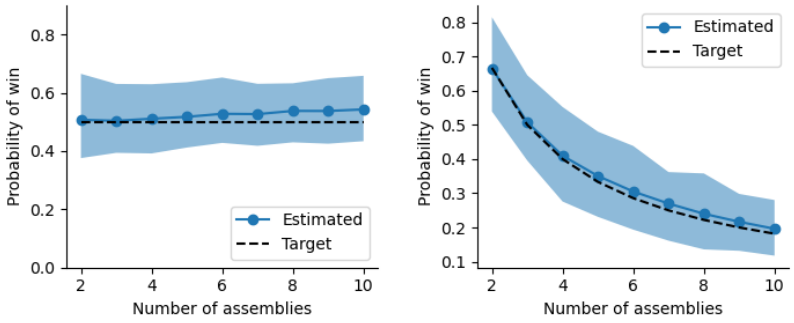
Figure 5.2: Variation in the probability that assembly A wins as its weight from the context assembly is varied, while the weights to B and C are held constant (and equal). On the left the weight is directly adjusted and compared against a best-fitting softmax function (dashed line); on the right all weights are updated incrementally according to a plasticity rule which approximately recovers the observed frequency (dashed line). Dark center line is the mean, while shaded area is the $[0.05, 0.95]$ quantile range, across 100 trials.

A, namely the assemblies A_i, can be used to input *observations* of the random variable during learning. It would be remarkable if, after presenting samples in this way, the probability distribution obtained by running the *previous* experiment (for sampling) yields the empirical distribution seen during training (i.e., the probability the firing neurons converge to A_i is the frequency with which A_i was presented).

This motivates our second experiment (Figure 5.2(b)), where we fix a random graph, and then present training samples to the model as follows: first fire I, and then A_i repeatedly T_i times. We then measure how the probability that each of A_1, A_2, A_3 fire changes as T_1 varies, with $T_2 = T_3 = 5$. We also plot the empirical frequency, $\frac{T_i}{T_1+T_2+T_3}$. The parameters of the plasticity rule are $\alpha = 0.63$, $\beta = 0.5$, and $\lambda = 26$. Remarkably, plasticity updates suffice to arrive at weights that sample from the empirical distribution! We refer the reader to Corollary 37 and Lemma 38 for a formal statement of this result and derivation of this plasticity rule.

Finally, to verify that the same plasticity rule suffices to learn distributions over various numbers of assemblies (outcomes), in Figure 5.3 compare the learned distribution as the number of assemblies that fired during training changes. Notably, a single plasticity rule

approximately recovers the observed frequency, even when the number of outcomes is relatively large. In the first experiment (Figure 5.3(a)) we vary the number of outcomes, while A_1 fires $5(m-1)$ times and $A_2, \ldots, A_m$ fire 5 times during training; in the second (Figure 5.3), A_1 fires 10 times and $A_2, \ldots, A_m$ fire 5 times. For both, we report the empirical frequency of A_1, estimated from 500 samples over 100 trials.



(a) (b)

Figure 5.3: Variation in the probability of an assembly winning, as the support of the training distribution grows to include more assemblies, under a plasticity rule that approximately recovers the training distribution. For two different sequences of distributions, we plot the empirical probability of assembly A_1 firing (blue dotted line) along with the target probability from the training distribution (black dashed). In (a) and (b) $A_2, \ldots, A_m$ fire 5 times during training, for each $m = 2, \ldots, 10$, while in (a) A_1 fires $5(m-1)$ times and in (b) A_1 fires 10 times for each m.

5.2.2 Scaling

One important observation about the assembly coinflip is that a substantially higher scale is required for the phenomenon to emerge compared to previous work inNEMO [98, 153] – say, $k = 500$ versus $k = 30$. Accordingly, we examine this scaling phenomenon quantitatively in Figure 5.4. For simplicity we consider the case where the target distribution over assemblies A and B is uniform, so the weights to each are the same – here, 2. For each choice of cap size (k) we sample 20 connectivity graphs, increase the weights, and estimate the resulting distribution from 500 samples. We then plot the resulting absolute deviation of A firing from $1/2$. Notably, the error does not reach 0, even for very large k, but nor do

we expect it to, since the maximum deviation of 20 $(500, 1/2)$ binomial random variables is roughly 0.1 in expectation (i.e. if the learned distribution were completely unbiased, and the only error was due to sampling).

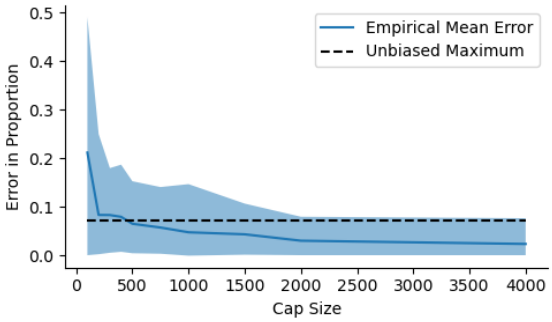


Figure 5.4: Variation in the error of the learned distribution over two assemblies as the cap size increases. For each of 20 trials, a new connectivity graph was sampled, the assemblies A and B were made to fire exactly the same number of times, and the resulting distribution was estimated from 500 samples. The absolute deviation of the frequency with which A won from $1/2$ was recorded for each trial. The dark center line is the mean of this error over trials, the shaded area is the range, and the dashed line is the *expected maximum* if the samples were truly unbiased (i.e. the probability that A comes to fire is exactly $1/2$).

5.2.3 Learning Markov chains

Next, we examine how accurately the transitions of a Markov chain can be learned from a stream of samples, using the basic mechanism demonstrated above (see Corollary 39 for the theoretical formulation of this result). We assume that there ares area A and B which contain an assembly A_s, B_s for each state s of the Markov chain (see Figure 5.8). The areas "lazily" alternate firing, so that A fires once while B is inhibited, then B fires 10 times while A is inhibited, then A fires again once, and so forth. There are connections in both directions between A and B, initially all 1. Internally, weights within each A_s and B_s are strengthened by a factor of 2.

The basic idea of the Markov model is to treat A as the "context" area in the coinflip model, and B as the "state" area, so that assemblies in B are outcomes with probability conditional on the assembly firing in A. The model is trained by presenting a stream of

states sampled from the target Markov chain, say $s_1, \ldots, s_t, \ldots$. Training proceeds by firing A_{s_1}, followed by B_{s_2}, followed by A_{s_2} and so on. In Figure 5.5, the training sequences had length $100 \cdot \#$ of states. At test time, we sample from the Markov chain by simply firing an initial state, and then observing the activity in area A every 10 rounds. To estimate the transition probabilties, for each state, we sample 1000 realizations of random noise and record the empirical frequency of each transition. In Figure 5.5 we display the results of learning two chains: one has three states and every transition has positive probability, and the other has 15 states but each state transitions to only five states with positive probability. For each instance, we display the empirical transition matrix estimated from samples alongside the ground-truth transition matrix from which the training examples were sampled. Here, $n = 25000, k = 500, p = 0.1$, and the noise has standard deviation $5\sqrt{kp}$, while the plasticity parameters are $\alpha = 0.63$, $\beta = 0.5$, and $\lambda = 26$.

A trigram model of text. As a demonstration of these ideas in simulation, we apply the probabilistic machinery described so far to construct a primitive model of language (Figure 5.6). Consider a corpus of language, which is a set of strings comprised of tokens from a lexicon $\mathcal{T}$. We suppose there exist brain areas A, B, C, with connections from A to B, B to A and C, and C to A. The areas contain assemblies A_τ, B_τ, C_τ for each token $\tau \in \mathcal{T}$; the firing of A_τ represents token τ being presented to the model. Internal connections of these assemblies are assumed to be strengthened, and connections from A_τ to B_τ and B_τ to C_τ, but otherwise no weight changes have occurred. Training then proceeds as follows: For a string of tokens $\tau_1, \tau_2, \ldots$, we fire $A_{\tau_1}, A_{\tau_2}, \ldots$. As this activity propagates to areas B and C, the result is strengthening the connections to A_{τ_t} from $B_{\tau_{t-1}}$ and $C_{\tau_{t-2}}$. Sampling then occurs in area A: To sample a third token in the trigram $(\tau_1, \tau_2, \cdot)$, B_{τ_2} and C_{τ_1} fire simultaneously, then A is allowed to fire 10 times, while connections from A to B and from B to C are inhibited, then A is inhibited while inter-areal connections are again enabled. After this process concludes, for some τ_3 we should have B_{τ_3} along

with C_{τ_2} firing, so sampling may repeat to generate a stream of tokens. Here, we take $n = 100,000, k = 500, p = 0.1$, with plasticity parameters $\alpha = 0.63, \beta = 0.5, \lambda = 60$. The model was trained by presenting the entire corpus 5 times. We emphasize that no settings or changes were made toNEMO to specialize to the data or output.

5.3 Theoretical Results

Outline. The presentation of our theoretical results follows the development of the experiments in the previous section. We first establish the basic ingredient of our assembly-level statistical learner – randomly choosing one assembly out of a set of possible "outcome" assemblies to fire, according to a probability distribution which depends on the weights from the context assembly to the outcome assemblies. This sampling procedure is driven by random noise, which we model here as an i.i.d. perturbation to each neuron's activation. The formal claim that this behavior provably occurs inNEMO (when the set of outcomes has size 2) is the content of Theorem 36. This gives the distribution as a function of the weights; in Corollary 37, we determine the weights necessary to achieve a target distribution. Lemma 38 gives a Hebbian plasticity rule which achieves the weights; more precisely, it incrementally increases the weights during training to ensure that the probability that an assembly fires after training replicates its observed frequency in the training phase. Finally, Corollary 39 uses this mechanism as the building block of a more complex model, namely the simulation of a restricted (but completely general) class of Markov chains using assemblies. Proofs follow in Section 5.3.1.

We now proceed with the statement of Theorem 36. At a high level, this theorem gives the distribution over outcome assemblies as a function of the weights from the context assembly to the outcomes (Figure 5.7). It is worth noting that there are two sources of randomness in play: One is the random graph which determines connectivity, and the other arises from the noise model's perturbations to neuronal activations. For our purposes, it is desirable that the realized distribution over outcomes does not depend on the connectivity

graph. Since complete independence is impossible, we settle for approximation with high probability, so that the probability (over the random graph) that the realized distribution deviates significantly from a distribution genuinely independent of the connectivity goes to zero as n (the number of neurons) goes to ∞.

Theorem 36 (**Conditional Coin-Flipping**). *Consider* NEMO *brain areas I and M, with a context set $I_0 \subseteq I$ and outcome sets $A, B \subseteq M$ of size k, satisfying $|A \cap B| \leq \frac{1}{2}\sqrt{k/\log k}$). Let the connections from I_0 to A (resp. B) have weight w_A (resp. w_B), and the connections within A and within B have weight w_R, and all other connections from $I_0 \cup A \cup B$ to M have weight 1. Suppose that I_0 fires once, and the total inputs to neurons in M are perturbed by i.i.d. $\mathcal{N}(\mu, \sigma^2 kp)$ random variables.*

Let C_t denote the set of k neurons firing in M after t rounds. Then there exist p, w_0, w_R, σ, t_0 (see Setting 1 for precise values) such that if $w_A, w_B \geq w_0$, for all $t \geq t_0$ we have

$$\left|\Pr(C_t = A|\mathcal{G}) - \Phi\left(\frac{k\sqrt{2p}}{\sigma}(w_A - w_B)\right)\right| = o(1),$$
$$\left|\Pr(C_t = B|\mathcal{G}) - \Phi\left(\frac{k\sqrt{2p}}{\sigma}(w_B - w_A)\right)\right| = o(1),$$

with high probability over the random graph $\mathcal{G}$ as $n \to \infty$. Here, Φ is the Gaussian CDF[1].

The upshot of Theorem 36 is that there is an explicit function of the weights only, which does not depend on the actual realization of the underlying graph (only on the parameters of the model), which is very close to the actual probability distribution over outcomes with high probability. We now give ranges of the parameters for which the conclusions of Theorem 36 hold.

Setting 1. In Theorem 36, it suffices that $p \geq \max\{\sqrt{\log k/k}, 3\log n/k\}$, $w_0 = 1 + 4(1 + \sigma)\frac{\sqrt{p}}{1-\sqrt{p}}$, $w_R \geq 1 + 2\sqrt{\log n/kp}$, $\sigma \geq \sqrt{\log k}$, and $t_0 = \log k/\log(kp)$.

[1] It is worth noting that the appearance of the Gaussian CDF here is *not* due to the normal distribution of the noise, but rather because under fairly general assumptions sample quantiles are asymptotically normally distributed. We expect that results differing only in the precise constants hold for other continuous noise distributions.

Notably, the condition on the edge density p is significantly stronger than previous results pertaining toNEMO [94, 98, 153] for typical choices of n and k, which only require that $p = \Omega(\log n/k)$ (i.e. a quadratically weaker bound for k polynomial in n). The higher edge density seems necessary to ensure that the assembly which eventually wins is simply the leader (in number of firing neurons) from the first round – more precisely, that the small numerical advantage enjoyed probabilistically by the initial leader is large enough to stand out against the random deviation in connectivity. This does not appear to be a limitation of our analysis; when low densities are used experimentally, convergence to a total winner is typically slow and the eventual winner is often different from the first round leader.

In light of the previous result , a natural next question, in some sense the converse, is whether for an arbitrary target probability distribution (over A and B) there exists a setting of the weights that realizes it, in the sense that after t time steps the probability that A or B is entirely firing matches their probability under the target distribution. The following corollary gives appropriate weights to approximate an arbitrary target distribution.

Corollary 37. *Under the assumptions of Theorem 36 and the parameters of Setting 1, there exist $c, \lambda > 0$ such that, for any $T_A, T_B \geq 1$, if*

$$\left| w_A - \frac{\log cT_A}{\lambda} \right| \leq \frac{1}{50\lambda}, \qquad \left| w_B - \frac{\log cT_B}{\lambda} \right| \leq \frac{1}{50\lambda}$$

then for all $t \geq t_0$ we have

$$\left| \Pr(C_t = A|\mathcal{G}) - \frac{T_A}{T_A + T_B} \right| < \frac{1}{100},$$

with high probability over the realization of $\mathcal{G}$. In particular, taking $\lambda = 1.7\frac{k\sqrt{2p}}{\sigma}$ and $c \geq e^{\lambda w_0}$ suffices.

The essence of this result is just showing that the Gaussian CDF $\Phi(x)$ is close to the logistic function (2-dimensional softmax), $\frac{e^{\lambda x}}{1+e^{\lambda x}}$, for appropriate choice of λ.

We now turn our attention to whether these weights can be *learned*, that is, estimated from examples. Suppose we can present examples to the model by first firing I_0, followed by firing either A or B. We now establish a plasticity rule such that the induced probability distribution over A and B approximates the *observed frequency* with which they each followed I_0 during training.

Lemma 38. *Under the assumptions of Theorem 36 and the parameters of Setting 1, suppose that the weight update is*

$$w_{ij}(t+1) = w_{ij}(t) + \max\{\alpha, e^{\lambda(1+\beta-w_{ij}(t))}\}$$

when neuron j fired on round t and neuron i fired on round $t+1$, and $w_{ij}(t)$ is the weight of the synapse from j to i on round t. Suppose that the number of timesteps where I_0 fires followed by A is T_A, and likewise for B and T_B. Then there exist $T_0, \alpha, \beta, \lambda$ such that for $T_A, T_B \geq T_0$,

$$\left| \Pr(C_t = A \mid \mathcal{G}) - \frac{T_A}{T_A + T_B} \right| < \frac{1}{100}.$$

In particular, $\lambda = 1.7\frac{k\sqrt{2p}}{\sigma}$, $\beta = w_0 - 1 - \log\lambda/\lambda$, and $\alpha = w_0 - 1$ suffice.

Note that the plasticity rule is Hebbian [2], and completely synapse-local; what is specified is only the magnitude of the change in weight, as a function of the current weight. This rule is a natural approximation of the "target" weights for proportional sampling given in Corollary 37, as follows. For $t = 0$, we have $w(0) = 1$ and the target $w(1) = \lambda^{-1}\log c$, so the incremental change due to the plasticity rule is simply $w(1) - w(0) = \lambda^{-1}\log c$. For $t \geq 1$, the increment is

$$w(t+1) - w(t) = \frac{\log(c(t+1)) - \log(ct)}{\lambda} = \frac{\log(1+1/t)}{\lambda}.$$

[2]Hebbian here meaning increasing, and applied only when the presynaptic neuron fires just before the postsynaptic neuron

We make the approximation $\log(1 + 1/t) \approx 1/t$, and then use the assumption that $t = e^{\lambda w(t)}/c$ (which is exactly true for the target weight) to get $w(t+1) - w(t) = \frac{c}{\lambda} e^{-\lambda w(t)}$. Taking $c = \lambda e^{\lambda(1+\beta)}$ and set $\alpha = \beta + \lambda^{-1} \log(\lambda)$ recovers the parameterization of the rule in the lemma.

Using the above result, we show rigorously that a finite, stationary Markov chain with out-degree at most 2 can be not only simulated but also learned from examples inNEMO.

Corollary 39 (Markov Chain)**.** *Consider a Markov chain with states $s \in [m]$ with transition probability from s to state $\rho(s,i)$ equal to $p_{s,i} \geq 0$ for $i = 1, 2$, and 0 for all other states.*

Suppose there are two distinct brain areas A and B, with subsets $A_s \subseteq A, B_s \subseteq B$ for each $s \in [m]$, satisfying $|A_s \cap A_t| \leq \sqrt{k/p}$ and $|B_s \cap B_t| \leq \frac{1}{2}\sqrt{k/\log k}$ for all $s \neq t$. Furthermore, suppose for each $s \in [m]$, $i = 1, 2$, connections from A_s to $B_{\rho(s,i)}$ have weight $\frac{1}{\lambda}\log(c \cdot p_{s,i}) + 1$, and connections from B_s to $A_s \cup B_s$ have weight w_R.

Suppose sampling is executed as follows: a set A_s in A fires, after which B fires for t timesteps while A is inhibited. Then, A is again disinhibited while B is inhibited for a single timestep and the cycle repeats.

Then there exist p, c, λ, σ, t such that for all $s \in [m]$, $i = 1, 2$, if A_s fires, the probability that $A_{\rho(s,i)}$ fires $t+1$ rounds later deviates from $p_{s,i}$ by no more than $1/100$, w.h.p. as $n \to \infty$.

In the notation of the corollary, this means that a model configured as above approximates the Markov chain with states $\{1, \ldots, m\}$, and with transition probability from state s to state $\rho(s,i)$ of $p_{s,i}$. Using the plasticity rule in Lemma 38, the transition probabilities can be learned from a sequence $s_1, s_2, \ldots$ sampled from the target Markov chain by firing the assemblies $A_{s_1}, B_{s_2}, A_{s_2}, B_{s_2}, \ldots$. Furthermore, out-degree 2 Markov chains are general in the following sense:

Remark. Consider a Markov chain with states $[m]$ and transition probability $p_{i,j}$ from state i to state j. Then there exists a Markov chain (X_t) with states $[m] \times [m-1]$ and out-degree

2 such that

$$\Pr(X_{t+\max\{j,m-1\}} = (j,1) \mid X_t = (i,1)) = p_{i,j}.$$

Intuitively, this construction reduces a single random choice of one item out of many to a sequence of binary choices, whether to choose a particular item or discard it and repeat with the smaller set. In terms of the transition graph of a Markov chain, it replaces each random transition with a binary tree of transitions, whose leaf vertices correspond to states in the original chain.

It is worth noting that the barrier to higher-degree Markov chains is a limitation of our analysis; in simulation (Appendix 5.2) our model performs well for a much larger number of outcomes.

Probabilistic computation. The above results have natural consequences for extending the Turing completeness ofNEMO [153] to an efficient simulation of arbitrary probabilistic computation. For simplicity, we make the following definition of a probabilistic Turing machine $M = (Q, \Sigma, \{H,T\}, \{L,R\}, q_0, q_A, q_R, \delta)$, where Q is a set of states, Σ is the alphabet, H and T are the L and R are the directions on the tape, $q_0, q_A, q_R \in Q$ are the (unique) initial, accepting, and rejecting states, respectively, and δ maps $Q \times \Sigma \times \{H,T\}$ to $Q \times \Sigma \times \{L,R\}$. Intuitively, M consists of an infinite read-write tape with a finite number of nonblank symbols on it at any given time, and an infinite read-only tape with a uniformly random sequence of $\{H,T\}$ symbols. The transitions depend on the symbols written on both tapes, but movement along the random tape is always to the right.

It is straightforward to see that given a uniformly random string over $\{H,T\}^m$, encoded as a sequence of firing of two assemblies at appropriate timesteps, the simulation of a deterministic Turing machine withinNEMO [153] in time $O(m)$ and space $O(m^2)$ gives a simulation of a probabilistic Turing machine with the same time and space bounds. So, the problem reduces to constructing a generator for this uniformly random string. The difficulty is that we have only given a sampler for a coin with expectation in $[0.49, 0.51]$,

but since the *expectation of this bias* is itself 0.5, this issue is easily surmounted by taking a majority vote over $\Omega(\log m)$ independent instances; sampling one symbol at a time from the boosted sampler gives a distribution over $\{H, T\}^m$ with $o(1)$ TVD from the uniform distribution. We can then formulate the following theorem:

Theorem 40. *Consider a probabilistic Turing machine which halts in time m. There exist n, k, p such that*NEMO *simulates this machine in time $O(m)$, using $\tilde{O}(\max\{m^2, |Q|^2 \cdot |\Sigma|^2\})$ space and $O(\log m)$ brain areas.*

The noise assumption. In our experiments and proofs, we have assumed that neuronal activations are perturbed by i.i.d. Gaussian noise. Due to the central limit theorem, this is a reasonable way to abstractly model the contribution of many small independent sources of uncertainty. In particular, one might explicitly model the source of randomness in the setting of Theorem 36 by adding an additional area R of size $\sigma^2 n$ with connections to area A, and assume that the only randomness is that a *random set* of k neurons fires in R. The resulting perturbations to the activations of neurons in A will be independent, identically-distributed binomial $(\sigma^2 n, p)$ random variables, which under the CLT converge in distribution to $\mathcal{N}(\sigma^2 np, \sigma^2 np(1-p))$ with appropriate normalization.

5.3.1 Proofs

The proof of Theorem 36 consists of showing that the probability that A obtains a decent advantage on the first round using Lemma 41 is very nearly the specified probability that A wins altogether, and then showing that this initial lead will quickly grow to activate the entire assembly using Lemma 7.

Theorem 36. WLOG, assume that $\mu = 0$. Let $X_i(0)$ denote the input to neuron i when I_0

fires, and $Y_i \sim \mathcal{N}(0, \sigma^2 kp)$ denote the random perturbations; then we have

$$X_i = \begin{cases} w_A \cdot e(I_0, i) + Y_i & i \in A \\ w_B \cdot e(I_0, i) + Y_i & i \in B \\ e(I_0, i) + Y_i & \text{o.w.} \end{cases}$$

We will first establish that $C_0 \subseteq A \cup B$ w.h.p. Note that w.h.p. for all $i \notin A \cup B$, we have $e(I_0, i) \leq kp + \sqrt{3kp \log n}$ by the Chernoff bound; likewise, $Y_i \leq \sigma\sqrt{3kp \log n}$. Hence,

$$X_i \leq kp + (1 + \sigma)\sqrt{3kp \log n}.$$

On the other hand, for all $i \in A \cup B$, we have

$$X_i \geq w_0(kp - \sqrt{2kp \log k}) - \sigma\sqrt{3kp \log k}$$

and so as long as

$$w_0 \geq \frac{kp + (1 + 2\sigma)\sqrt{3kp \log n}}{kp - \sqrt{2kp \log k}}$$

we have

$$\min_{i \in A \cup B} X_i \geq \max_{i \notin A \cup B} X_i.$$

Hence, w.h.p. $C_0 \subseteq A \cup B$.

Now, let T_0 be the median of $\{X_i(0) \colon i \in A \cup B\}$, and note that $i \in C_0$ if and only if $X_i(0) \geq T_0$. By Lemma 41, we have

$$\Pr(|C_0 \cap A| \geq k/2 + \sqrt{k/\log k} \mid \mathcal{G}) \geq \Phi\left(\lambda \frac{k\sqrt{p}}{\sigma}(w_A - w_B)\right)$$
$$\Pr(|C_0 \cap B| \geq k/2 + \sqrt{k/\log k} \mid \mathcal{G}) \geq \Phi\left(\lambda \frac{k\sqrt{p}}{\sigma}(w_B - w_A)\right)$$

Now, suppose that $k - |C_t \cap A| \leq \sqrt{k/\log k}$. Since $\sqrt{k/\log k} \geq 6p^{-1}$, by Lemma 7, we

will have

$$|C_{t+1} \cap A| \geq k/2 + \max\{k/2, \frac{\sqrt{kp}}{6}(|C_t \cap A| - k/2)\}$$

w.p. at least $1 - \exp\left(-(|C_t \cap A| - k/2)^2 p\right)$. Thus, if $|C_0 \cap A| \geq k/2 + \sqrt{k/\log k}$, then for all $t \geq 1$ we have

$$|C_t \cap A| \geq k/2 + \max\{k/2, \left(\sqrt{kp}6\right)^t \sqrt{k/\log k}\}$$

w.p. $1 - o(1)$, and so after $t_0 = O(\log k/\log(kp))$ rounds, we must have $|C_{t_0} \cap A| = k$ w.h.p. A similar argument holds for B. □

Corollary 37. One can verify numerically that

$$\sup_{x \in \mathbb{R}} \left| \frac{e^{1.7x}}{1 + e^{1.7x}} - \Phi(x) \right| < \frac{1}{100}.$$

In particular, for $x = \frac{(w_A - w_B)}{\tau}$, and setting $\lambda = 1.7/\tau$, we have

$$\frac{e^{1.7x}}{1 + e^{1.7x}} = \frac{e^{1.7w_A/\tau}}{e^{1.7w_A/\tau} + e^{1.7w_B/\tau}} = \frac{e^{\lambda w_A}}{e^{\lambda w_A} + e^{\lambda w_B}}$$

Now, for settings of w_A and w_B that satisfy the corollary conditions,

$$\frac{e^{\lambda w_A}}{e^{\lambda w_A} + e^{\lambda w_B}} \approx \frac{cT_A}{cT_A + cT_B} \approx \frac{T_A}{T_A + T_B}$$

where $\approx$ denotes an absolute difference of $o(1)$, and so

$$\left| \Phi\left(\frac{w_A - w_B}{\tau}\right) - \frac{T_A}{T_A + T_B} \right| < \frac{1}{100} + o(1).$$

We then take c and T_0 large enough to ensure the condition on w_0 in Theorem 36 is satisfied.

□

Lemma 38. We first note that for $\alpha = \beta + \lambda^{-1}\log(\lambda)$, one has $w(1) = \lambda^{-1}\log(c \cdot 1)$ for

$c = \lambda e^{\lambda(1+\beta)}$. The substance of the proof is showing that for all $t \geq 2$, we have

$$\frac{\log(ct)}{\lambda} \leq w(t) \leq \frac{\log(c(t + \log t))}{\lambda}.$$

It is readily verified that $w(t)$ satisfying the above also meets the conditions of Corollary 37 and hence the desired probabilities obtain. The above claim reduces to showing that the recurrence relation

$$x_{n+1} = x_n + e^{-x_n} \qquad\qquad x_1 = 0$$

satisfies $\log n \leq x_n \leq \log(n + \log n)$. This can be shown by induction; clearly it holds for $n = 1$. For $n \geq 2$, we have the lower bound

$$x_{n+1} = x_n + e^{-x_n} \geq \log n + \frac{1}{n} \geq \log(n + 1).$$

For the upper bound, it can be verified that the function

$$f(x) = \log(x + \log x) + \frac{1}{x + \log x} - \log(x + 1 + \log(x + 1))$$

has a single extremum, at which it is positive, and since $f(1) = 0$ and $\lim_{x\to\infty} f(x) = 0$ we have $f \geq 0$ on $[1, \infty)$. This gives $x_{n+1} \leq \log(n + 1 + \log(n + 1))$. □

Corollary 39. For each $s = 1, \ldots, m$, Corollary 37 asserts that this choice of weights will ensure that when A_s fires, the probability that $B_{\rho(s,i)}$ fires will be $p_{s,i} \pm 1/50$. For sufficiently large w_R, when $B_{\rho(s,i)}$ fires $A_{\rho(s,i)}$ will fire immediately after. So, by choosing t larger than t_0 in Theorem 36 we can ensure that the entirety of $B_{\rho(s,i)}$ fires when A becomes disinhibited, so the probability that $A_{\rho(s,i)}$ fires in area A t rounds after A_s is very nearly $p_{s,i}$. □

Lemma 41. *Let $X_1, X_1', \ldots, X_n, X_n'$, $Y_1, Y_1', \ldots, Y_n, Y_n'$ be independent, with $X_i, X_i' \sim \mathcal{B}(n, p)$ and $Y_i, Y_i' \sim \mathcal{N}(0, \sigma^2 np(1 - p))$ for all $i = 1, \ldots, n$. Let M denote the median of*

the combined sample $wX_1 + Y_1, w'X'_1 + Y'_1, \ldots, wX_n + Y_n, w'X'_n + Y'_n$. *Then w.h.p. as* $n, \sigma \to \infty$, *we have*

$$\Pr\left(\#\{i = 1, \ldots, n \colon wX_i + Y_i \geq M\} \geq \frac{n}{2} + \sqrt{\frac{n}{\log n}} \,\middle|\, X_1, X'_1 \ldots, X_n, X'_n\right) \geq \Phi\left(\lambda \frac{n\sqrt{p}}{\sigma}(w - w')\right) - o(1)$$

for some constant $\lambda > 0$.

Proof. Set $Z_i = wX_i + Y_i$ and $Z'_i = w'X'_i + Y'_i$, and let $Z_{(k)}, Z'_{(k)}$ denote their kth order statistics, respectively. Note that $\#\{i = 1, \ldots, n \colon Z_i \geq M\} \geq k$ if and only if $Z_{(n-k+1)} \geq Z'_{(k)}$. Let $Z_{(k)}(x) = Z_{(k)} \mid \{X = x\}$, and note that $Z_{(k)}(x)$ is the kth order statistic of the normal random variables $Y_1 + wx_1, \ldots, Y_n + wx_n$, where $Y_i + wx_i$ has mean wx_i and variance $\sigma^2 np(1-p)$. Let $Q(x), R(x)$ be chosen so that

$$\frac{1}{n}\sum_{i=1}^{n} \Phi\left(\frac{Q(x) - wx_i}{\sigma\sqrt{np(1-p)}}\right) = \frac{1}{2} - \frac{1}{\sqrt{\log n}}$$
$$\frac{1}{n}\sum_{i=1}^{n} \Phi\left(\frac{R(x) - w'x_i}{\sigma\sqrt{np(1-p)}}\right) = \frac{1}{2} + \frac{1}{\sqrt{\log n}}$$

noting that this is possible as Φ is continuous. Denote by $\Sigma(x), \Sigma'(x)$ the quantities

$$\Sigma(x) = \sqrt{\sigma np(1-p)}\frac{\sqrt{\sum_{i=1}^{n} \Phi\left(\frac{Q(x)-wx_i}{\sigma\sqrt{np(1-p)}}\right)\left(1 - \Phi\left(\frac{Q(x)-wx_i}{\sigma\sqrt{np(1-p)}}\right)\right)}}{\sum_{i=1}^{n} \phi\left(\frac{Q(x)-wx_i}{\sigma\sqrt{np(1-p)}}\right)}$$
$$\Sigma'(x) = \sqrt{\sigma np(1-p)}\frac{\sqrt{\sum_{i=1}^{n} \Phi\left(\frac{R(x)-w'x_i}{\sigma\sqrt{np(1-p)}}\right)\left(1 - \Phi\left(\frac{R(x)-w'x_i}{\sigma\sqrt{np(1-p)}}\right)\right)}}{\sum_{i=1}^{n} \phi\left(\frac{R(x)-w'x_i}{\sigma\sqrt{np(1-p)}}\right)}$$

and let $S(x) = Z_{n/2-\sqrt{n/\log n}}(x), S'(x') = Z'_{n/2+\sqrt{n/\log n}}(x')$. By Lemma 8 (specifically

Remark 2.6.1), we have that the random variables

$$\frac{S(X) - Q(X)}{\Sigma(X)}, \frac{S'(X') - R(X')}{\Sigma'(X')}$$

converge in distribution to $\mathcal{N}(0,1)$ almost surely (over X and X'). Let x, x' be chosen so that this holds. As $S(x), S'(x')$ are independent, the random variable

$$\frac{Z_{(n/2-\Delta)}(x) - Z'_{(n/2+\Delta)}(x') - (Q(x) - R(x'))}{\sqrt{\Sigma(x)^2 + \Sigma'(x')^2}}$$

also converges to $\mathcal{N}(0,1)$ in distribution. Thus, for $W \sim \mathcal{N}(0,1)$,

$$\Pr(S(x) - S'(x') \geq 0) \geq \Pr\left(W \geq \frac{R(x') - Q(x)}{\sqrt{\Sigma(x)^2 + \Sigma(x')^2}}\right) - o(1)$$

Now, note that if $Q(x) \leq t$, then by monotonicity

$$\sum_{i=1}^{n} \Phi\left(\frac{t - wx_i}{\sigma\sqrt{np(1-p)}}\right) \geq \frac{n}{2} - \sqrt{\frac{n}{\log n}}.$$

Let t_q be chosen so that $\mathbb{E}\Phi\left(\frac{t_q - wX_1}{\sigma\sqrt{np(1-p)}}\right) = q$. As Φ is $1/\sqrt{2\pi}$-Lipschitz, by Lemma 44 we have

$$\operatorname{Var}\Phi\left(\frac{t_q - wX_1}{\sigma\sqrt{np(1-p)}}\right) \leq \frac{w^2}{2\pi}\frac{\operatorname{Var} X_1}{\sigma^2 np(1-p)} = \frac{w^2}{2\pi\sigma^2}.$$

Furthermore, clearly

$$\mathbb{E}\left|\Phi\left(\frac{t_q - wX_1}{\sigma\sqrt{np(1-p)}}\right) - q\right|^3 \leq 1.$$

Set $q = 1/2 - \frac{1}{\sqrt{n \log n}} - \frac{1}{\sqrt{\sigma n}}$. The Berry-Esseen theorem then provides that

$$\begin{aligned}
\Pr(Q(X) \le t_q) &\le \Pr\left(\sum_{i=1}^{n} \Phi\left(\frac{t_q - wX_i}{\sigma\sqrt{np(1-p)}}\right) \ge n/2 - \sqrt{n/\log n}\right) \\
&\le \Phi\left(-\frac{n/2 - \sqrt{n/\log n} - (n/2 - \sqrt{n/\log n} - \sqrt{n/\sigma})}{w\sqrt{n/2\pi\sigma^2}}\right) + o(1) \\
&= \Phi\left(-\sqrt{2\pi\sigma}/w\right) + o(1) \\
&= \Phi(-\omega(1)) + o(1) = o(1).
\end{aligned}$$

A similar argument establishes that $R(X') \le t_{1-q}$ w.p. $1 - o(1)$. If the above both hold, then

$$R(X') - Q(X) \le t_{1-q} - t_q \le (w' - w)np + \sigma\sqrt{np(1-p)} \cdot o(n^{-1/2})$$

Now, note that

$$\mathbb{E}\left[\Phi(\frac{t_q - wX_1}{\sigma\sqrt{np(1-p)}})\left(1 - \Phi(\frac{t_q - X_1}{\sigma\sqrt{np(1-p)}})\right)\right] = q(1-q) - \operatorname{Var}\Phi\left(\frac{t_q - wX_1}{\sigma\sqrt{np(1-p)}}\right)$$

Hence,

$$\mathbb{E}\left[\Phi(\frac{t_q - wX_1}{\sigma\sqrt{np(1-p)}})\left(1 - \Phi(\frac{t_q - wX_1}{\sigma\sqrt{np(1-p)}})\right)\right] \ge q(1-q) - \frac{w^2}{2\pi\sigma^2}.$$

By Hoeffding's inequality,

$$\sum_{i=1}^{n} \Phi(\frac{t_q - X_i}{\sigma\sqrt{np(1-p)}})\left(1 - \Phi(\frac{t_q - X_i}{\sigma\sqrt{np(1-p)}})\right) \ge (1 - o(1))n\left(\frac{1}{4} - \frac{w^2}{2\pi\sigma^2}\right) = (1 - o(1))\frac{n}{4}$$

with high probability. On the other hand, clearly

$$\sum_{i=1}^{n} \phi\left(\frac{Q(x) - wx_i}{\sigma\sqrt{np(1-p)}}\right) \le n$$

and so

$$\sqrt{\Sigma(X)^2 + \Sigma'(X')^2} = \sigma\sqrt{np(1-p)}\sqrt{2\frac{(1-o(1))n/4}{n^2}} = (1-o(1))\sigma\sqrt{np(1-p)/2n}$$

again w.h.p. Finally, with x, x' satisfying all of the above conditions, we have

$$\begin{aligned}\Pr(S(x) \geq S'(x')) &\geq \Pr\left(W \geq \frac{(w'-w)np + \sigma\sqrt{np(1-p)} \cdot o(n^{-1/2})}{\sigma\sqrt{np(1-p)/2n}}\right) - o(1) \\ &= \Pr(W \geq \lambda\frac{n\sqrt{p}}{\sigma}(w'-w) + o(1)) - o(1) \\ &= \Phi\left(\lambda\frac{n\sqrt{p}}{\sigma}(w-w')\right) - o(1).\end{aligned}$$

Since these conditions hold w.h.p. for X and X', the proof is complete. □

Lemma 42. *Let $\Phi\colon \mathbb{R} \to [0,1]$ be the c.d.f. of the standard normal distribution. Then for $t \geq 0$,*

$$\frac{1}{2} + \frac{t}{\sqrt{2\pi}} - O(t^3) \leq \Phi(t) \leq \frac{1}{2} + \frac{t}{\sqrt{2\pi}}.$$

Proof. Let $\phi\colon \mathbb{R} \to \mathbb{R}$ be the p.d.f. of the standard normal distribution. By Taylor's theorem, there exists $0 \leq s \leq t$ such that $\Phi(t) = \Phi(0) + t\phi(s)$. By the monotonicity of ϕ for $x \geq 0$, it follows that

$$\Phi(0) + t\phi(t) \leq \Phi(t) \leq \Phi(0) + t\phi(0).$$

Substituting, we obtain

$$\frac{1}{2} + \frac{t}{\sqrt{2\pi}}e^{-t^2/2} \leq \Phi(t) \leq \frac{1}{2} + \frac{t}{\sqrt{2\pi}}$$

Using the inequality $e^{-t^2/2} \geq 1 - t^2/2$ completes the proof. □

Lemma 43. *Let $X_1, \ldots, X_n$ be independent random variables taking values in $[0,1]$ almost*

surely. Set $\mu = \sum_{i=1}^n \mathbb{E}X_i$. *Then for* $t > 0$,

$$\Pr\left(\sum_{i=1}^n X_i \geq \mu + t\right) \leq \exp\left(-\frac{t^2}{2\mu + t}\right)$$
$$\Pr\left(\sum_{i=1}^n X_i \leq \mu - t\right) \leq \exp\left(-\frac{t^2}{2\mu - t}\right).$$

Proof. Let $p_i = \mathbb{E}X_i$. The Chernoff bound gives for any $\lambda > 0$

$$\Pr\left(\sum_{i=1}^n X_i \geq \mu + t\right) \leq e^{-\lambda(\mu+t)}\mathbb{E}\exp\left(\lambda\sum_{i=1}^n X_i\right).$$

By independence,

$$\mathbb{E}\exp\left(\lambda\sum_{i=1}^n X_i\right) = \prod_{i=1}^n \mathbb{E}e^{\lambda X_i}.$$

By convexity, $e^{\lambda X_i} \leq 1 + X_i(e^\lambda - 1)$ and thus by monotonicity

$$\mathbb{E}e^{\lambda X_i} \leq 1 + p_i(e^\lambda - 1) \leq \exp\left(p_i(e^\lambda - 1)\right).$$

Hence,

$$\Pr\left(\sum_{i=1}^n X_i \geq \mu + t\right) \leq \exp\left(\mu(e^\lambda - 1) - \lambda(\mu + t)\right).$$

Taking $\lambda = \log(1 + \frac{t}{\mu})$ and using the inequality $\log(1 + x) \geq \frac{2x}{2+x}$ gives

$$\Pr\left(\sum_{i=1}^n X_i \geq \mu + t\right) \leq \exp\left(t - 2\frac{t(\mu + t)}{2\mu + t}\right) = \exp\left(-\frac{t^2}{2\mu + t}\right).$$

For the lower tail, for any $\lambda > 0$,

$$\Pr\left(\sum_{i=1}^n X_i \leq \mu - t\right) \leq e^{\lambda(\mu-t)}\prod_{i=1}^n \mathbb{E}e^{-\lambda X_i}.$$

By convexity and monotonicity, we have

$$\mathbb{E}e^{-\lambda X_i} \leq 1 - p_i(1 - e^{-\lambda}) \leq \exp\left(-p_i(1 - e^{-\lambda})\right)$$

and so

$$\Pr\left(\sum_{i=1}^{n} X_i \leq \mu - t\right) \leq \exp\left(\lambda(\mu - t) - \mu(1 - e^{-\lambda})\right).$$

For $\lambda = -\log(1 - \frac{t}{\mu}) \geq \frac{2t}{2\mu - t}$, we have

$$\Pr\left(\sum_{i=1}^{n} X_i \leq \mu - t\right) \leq \exp\left(\frac{2t/\mu}{2 - t/\mu}(\mu - t) - t\right) = \exp\left(-\frac{t^2}{2\mu - t}\right).$$

□

Lemma 44. *Let X be a random variable with $\operatorname{Var} X < \infty$ and $g\colon \mathbb{R} \to \mathbb{R}$ an L-Lipschitz function. Then $\operatorname{Var} g(X) \leq L^2 \operatorname{Var} X$.*

Proof. For any random variable Y, we have

$$\operatorname{Var} Y = \mathbb{E}Y^2 - (\mathbb{E}Y)^2 \leq \mathbb{E}Y^2$$

Hence,

$$\operatorname{Var} g(X) = \operatorname{Var}(g(X) - g(\mathbb{E}X)) \leq \mathbb{E}(g(X) - g(\mathbb{E}X))^2.$$

Since g is L-Lipschitz, we have $(g(X) - g(\mathbb{E}X))^2 \leq L^2 \cdot (X - \mathbb{E}X)^2$ and so by monotonicity

$$\operatorname{Var} g(X) \leq L^2 \mathbb{E}(X - \mathbb{E}X)^2 = L^2 \operatorname{Var} X.$$

□

5.4 Discussion

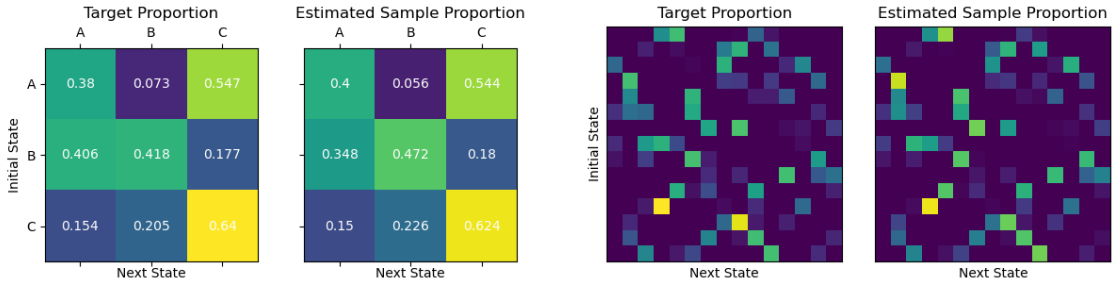
We have described a model of the brain at the neuronal level in which the statistical structure of stimuli can be recorded and reproduced through assemblies and sampling. Key ingredients of this model, inherited fromNEMO, are brain areas, random synapses, plasticity, and inhibition-induced competition; to these we add here random noise, and an additive plasticity rule enabling statistical learning, which may be of interest on its own. We provide both simulations demonstrating statistical learning phenomena, as well as proofs of a theoretically tractable special case. It would be interesting to extend these results to more complex graphical models in efficient ways. Although we have focused on Markov chains, our methods can be used to show that via the presentation of an appropriate sequence of stimuli,NEMO can perform arbitrary probabilistic computation in conjunction with the Turing machine simulation inNEMO previously shown [153].

A particularly interesting observation is the emergence of the softmax, or Boltzmann distribution, as the probability that an assembly fires as a function of its weight. Our model also makes two precise predictions; one is about the exact Hebbian plasticity rule used to implement this form of statistical learning, and another is the alternating inhibition of connected areas ("context" and "outcome") which gives rise to sampling of assemblies.

A fertile source of inspiration for this work was naturally language acquisition (surely one of the human brain's most remarkable abilities), not least due to the substantial amount of experimental work in psychology dedicated to understanding the statistical aspects of this task. We imagine that the statistical sensitivity of our model might support the most rudimentary phases of language learning; for instance, reproducing the phonological structure of words while babbling [191], or discovering word boundaries [163]. Our simple language experiments are in line with this perspective, demonstrating a "babbler" inNEMO which generates locally-plausible word sequences but lacks higher-level structure. In Kahneman's two-systems theory [192], this form of statistical learning might be regarded as the

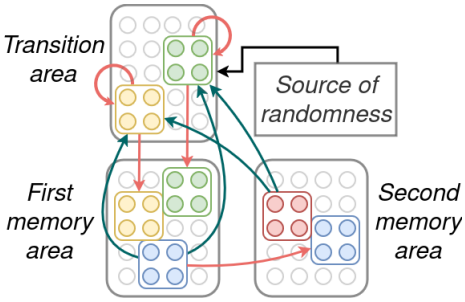
"fast" system, which requires interaction with the symbolic reasoning of the "slow" system to produce grammatical language. This will also surely be a rich direction for future work — in particular, the present model does not assume nor seem to discover any structure in its input beyond local co-occurrence statistics. One can imagine more complex models which learn to construct syntax trees of their input (perhaps again using statistical cues), and sampling could then occur at higher levels in the tree. This parallels the observation in language acquisition that the phase wherein the brain "commits itself" to the particular patterns of its native language seems to be crucial to a speaker's development [193].

Additionally, our work raises some clear open problems of technical interest. One is to extend our theoretical results to deal with random events with more than two outcomes (equivalently, Markov chains of maximum out-degree greater than 2). Our experimental results suggest that similar guarantees should be possible for significantly more outcomes. A second, outstanding one for theNEMO model in general is the question of capacity: how many neurons in a brain area are required to create and maintain a set of assemblies? More precisely, given a set of m stimulus items which can be externally fired, for which values of n do there exists k and p and Hebbian plasticity rules such that some sequence of firing the stimulus items results in the creation of assemblies with the small overlap and high internal connectivity required for simulating a Markov chain? Previous work inNEMO has given only very loose bounds [153], quadratic in m, which do not even recruit every neuron in the brain area to an assembly. A final question is whether the ability for a neuron to participate in multiple assemblies can be exploited to an *advantage* in statistical learning, instead of interfering with learning (and thus avoided) as in this work.



(a) (b)

Figure 5.5: We compare an estimate of the transition matrix learned by the model from a stream of samples from a Markov chain with the true transition matrix. This model is trained by presenting sequences of states, sampled from a ground-truth Markov chain, and then firing the appropriate state and transition assemblies in an alternating fashion, with the weights updated by the appropriate Hebbian rule. In both (a) and (b) the left is the ground truth transition matrix (which training samples were generated from) and the right is estimated empirical transition matrix. In (a) we show a chain with a dense transition graph, and in (b) a chain with a sparse transition graph.




```
the owl and the pussycat went
to sea in a wood a piggy wig
stood with a ring.  they took
some honey and plenty of money
wrapped up in a beautiful pea
green boat.  they took some honey
and plenty of money wrapped up in
a wood a piggy wig stood with a
ring.  they took it away and were
married next day by the light of
the moon.
```

(a) (b)

Figure 5.6: In (a), we exhibit a schematic of a simple trigram model of language withinNEMO. In (b), we show an example string generated after training the model with the text of the poem "The Owl and the Pussy-Cat" by Edward Lear, where each word and punctuation mark is a token. The poem contains 109 tokens comprising 182 unique trigrams.

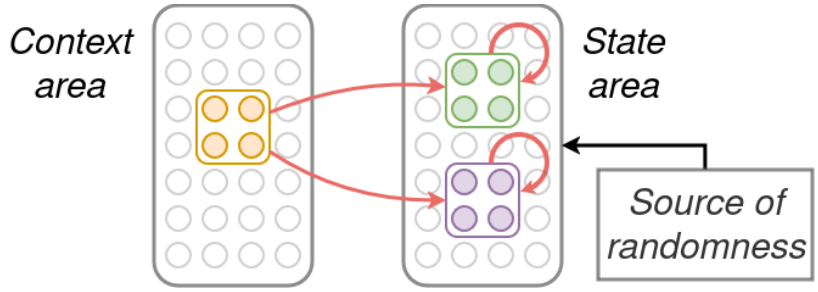


Figure 5.7: Choosing one of two assemblies to fire, based on context. The probability distribution over assemblies depends on the weights of each assembly from the context (Theorem 36).

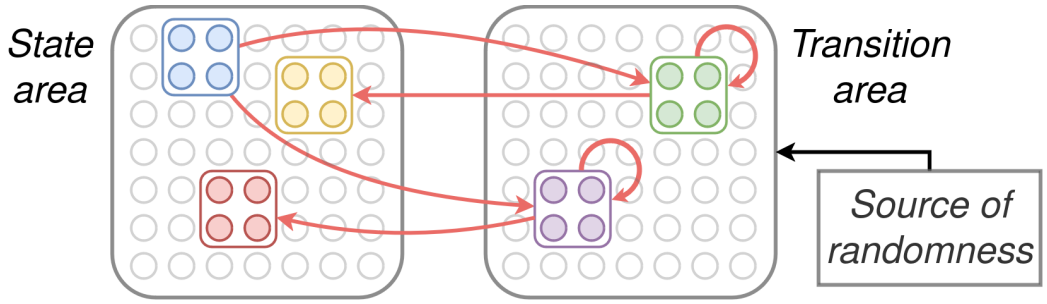


Figure 5.8: Realizing the transitions of a Markov chain using assemblies (Corollary 39). Both the state and transition areas contain an assembly for each state. Assemblies in the state area are connected to transition assemblies, with the synaptic weights from a state assembly to the transition assemblies encoding the probability of transitioning to those states.

CHAPTER 6
CONCLUSION

We have presented NEMO, a model of the brain and of computation, and explored a few algorithms which naturally arise from its dynamics. The basic object that these algorithms act on and through is the *assembly of neurons* – sets of neurons whose synchronized firing represents concepts in the model. Although simple, the algorithms we considered – learning a linear classifier from examples, memorizing & completing sequences, memorizing & simulating automata, and statistical estimation & sampling – are all certainly things the brain is capable of, and moreover are likely to be building blocks for the more complex tasks at the heart of cognition which make the brain such a marvel.

In the remainder of this thesis, we attempt to summarize the important directions for further developing NEMO as a model of the brain. We divide these points into *open problems*, which are relatively clearly formulated technical questions that do not require significant extensions to the model, *empirical questions* about how well NEMO captures the brain, and *future directions*, more open-ended problems regarding how the general approach of developing models of neurobiology and designing algorithms within them can shed light on how intelligence arises from the brain.

6.1 Open Problems

- **Capacity.** What is the largest m such that a brain area of size n can store m assemblies of size k, subject to certain properties? Important, generic properties would include mapping of distinct external stimuli to distinct assemblies, bounded overlap of distinct assemblies, pattern completion whenever a significant fraction of an assembly fires, and the ability to link arbitrary pairs of assemblies A and B such that whenever A fires, B fires shortly thereafter.

- **Neuron sharing.** How can the potential for a neuron to participate in multiple assemblies be made useful? Currently, we view overlap between assemblies as a nuisance, which is raises questions about why the brain would bother with distributed representations and leads to populations of "grandmother neurons", failing to match the experimental evidence for more dynamic tuning of individual neurons. One possibility arising from the previous problem is that it increases the capacity; another is that it allows for complex representations of the structure of a family of stimuli in the overlap structure of assemblies.

- **Adaptive cap size.** How does allowing the cap size to change systematically over time (much like the inhibition of areas) increase the power of NEMO? We have reason to believe that this is helpful in at least two ways. The first is that given a set of assemblies in one brain area which are linked into a sequence, say $A_1, A_2, \ldots$, when A_i is firing, toggling to cap size $2k$ seems to cause A_i and A_{i+1} to fire together, which will continue until the cap size returns to k, at which point only A_{i+1} will fire. This allows recurrently connected assemblies to be linked into a sequence, which can be stepped through by changing the cap size, a substantial improvement on our current sequence memorization protocol, and potentially frees the sequence from being clocked to the dynamics of the network. The second is decreasing the cap size to ϵk, then increasing it to k, allows for many-to-many mappings of assemblies which are as complex as a two-layer neural network (so, for instance, can realize XOR), whereas to enable these arbitrary mappings in our finite state machine simulation we needed an additional area, with a distinct cap for every state-symbol pair.

6.2 The Empirical Outlook for NEMO

Experimental probing of NEMO could address both the biological realism of the mechanisms which it is built upon and whether the phenomena it predicts actually occurs. The following are a few points of particular interest:

- Perhaps the most distinctive assumption of NEMO is that local inhibition can be abstracted as a k-winners-take-all operation on an excitatory population. If this turns out to be essentially incorrect, in that the operation of local inhibition is substantially different from this dynamic, NEMO will be debunked. On the other hand, a "noisy" winners-take-all could serve as a source of randomness for the neural coin-flipping in Chapter 5.

- The precise timing of Hebbian plasticity is important for some of our results. For assembly creation (and its extension, learning linear classifiers) to occur essentially as predicted in NEMO, it is necessary that there is a short-term Hebbian effect induced by single pre- and post-synaptic spike pairs and active on the timescale of 10ms which leads to convergence of an assembly within, say, 100ms, and a longer term Hebbian effect (potentially only induced by the burst of coordinated activity occurring during and after convergence) which ensures the assembly remains stable on task-relevant time scales (seconds to minutes) if not much longer. Our other results – sequence memorization, automata memorization, and statistical learning – do not need short-term Hebbian plasticity and will occur with only the long-term version, although for sequence memorization it is not possible to induce bursts of activity in the pre- and post-synaptic neurons under the protocol we have described, so small increments of plasticity must be induced by single spike pairs.

- NEMO assumes that there is a basic unit of organization in the brain, called here a "brain area", in which connectivity among excitatory neurons is random and local inhibition is tightly coupled to enforce a winner-take-all dynamic across the area. The sharp boundary of an area suggests it might correspond to a cortical column. If the excitatory and inhibitory circuits do not precisely correspond – i.e. there is some overlap among inhibitory circuits – it is unlikely to be a fatal blow for NEMO but will require re-thinking the division of brain areas.

- Our algorithms that involve multiple brain areas rely on the total (dis)inhibition of these brain areas from time to time. Except for the tape of a Turing machine (Chapter 4), which is merely a proof of concept, this inhibition is periodic and thus state-independent, and so we expect it corresponds to (i.e. either drives or is driven by) the brain's more global rhythms. If it is discovered that brain rhythms do not involve the periodic inhibition of anatomically localized regions, this assumption will be invalid. Nonetheless, NEMO will still essentially stand – we will just need to reconsider how algorithms involving multiple brain areas can be coordinated.

- Optogenetics could in principle be used to simulate any of the operations we have described. Can arbitrary sequences of sufficiently large sets of neurons be linked after they are induced to fire in sequence a few times? What about finite state machines, where initially symbol-state pairs, followed by a unique set of neurons for each pair, followed by the next state, are induced to fire? The ultimate test of this would be to fire only the symbol neurons and examine whether the states are traversed correctly.

6.3 Future Directions

- So far, the most sophisticated operations we have managed to implement in NEMO have essentially had a statistical estimation character – no inference has been required. How does more complex, hierarchical learning occur in the brain? A simple, concrete problem is whether NEMO or an extension thereof can learn hidden Markov models. More generally, we are interested in a biologically implementable algorithm for language learning.

- Is there a canonical function for the cerebral cortex which is realizable in a NEMO-like model? In terms of NEMO, this might consist of a generic circuit of brain areas which can be massively parallelized across cortex and yet somehow give rise to cognition. The search for such a canonical function is motivated by the apparent homogeneity

of cortex across brain areas, and the immense flexibility and resilience of the brain in response to massive perturbations, both during development and later in life.

- Recent successes of generative modeling in, for instance, natural language, have reinvigorated the idea that the brain could essentially construct a generative model for its environment. How far can viewing the brain as a generative model for its environment take us?

- The best empirically-motivated reward and feedback signals the brain has access to seem to have a relatively untargeted character – for instance, the release of dopamine into a brain volume containing a fairly large number of neurons. How can the brain learn using such coarse feedback?

- NEMO has restricted dynamics but access to massive parallelism. Is there a model of computation slightly more abstract than NEMO, taking as primitive those operations and structures which arise naturally under the dynamics of NEMO, which provably performs some difficult computation quickly? One might imagine constructing "circuits" out of components comprising a few brain areas, subject to a small selection of control signals which might inhibit the areas, change their cap sizes, etc.

REFERENCES

[1] B. A. Richards and T. P. Lillicrap, "The brain-computer metaphor debate is useless: A matter of semantics," *Frontiers in Computer Science*, vol. 4, p. 810 358, 2022.

[2] B. J. Copeland, "The Church-Turing Thesis," in *The Stanford Encyclopedia of Philosophy*, E. N. Zalta and U. Nodelman, Eds., Winter 2024, Metaphysics Research Lab, Stanford University, 2024.

[3] D. Marr, *Vision: A computational investigation into the human representation and processing of visual information*. MIT press, 2010.

[4] I. H. Stevenson and K. P. Kording, "How advances in neural recording affect data analysis," *Nature neuroscience*, vol. 14, no. 2, pp. 139–142, 2011.

[5] T. J. Sejnowski, P. S. Churchland, and J. A. Movshon, "Putting big data to good use in neuroscience," *Nature neuroscience*, vol. 17, no. 11, pp. 1440–1441, 2014.

[6] G. Marcus, "Deep learning: A critical appraisal," *arXiv preprint arXiv:1801.00631*, 2018.

[7] E. Jonas and K. P. Kording, "Could a neuroscientist understand a microprocessor?" *PLoS computational biology*, vol. 13, no. 1, e1005268, 2017.

[8] A. Doerig *et al.*, "The neuroconnectionist research programme," *Nature Reviews Neuroscience*, vol. 24, no. 7, pp. 431–450, 2023.

[9] J. L. Borges, *Collected Fictions*, trans. by A. Hurley. Penguin, 1999.

[10] R. Yuste, "From the neuron doctrine to neural networks," *Nature reviews neuroscience*, vol. 16, no. 8, pp. 487–497, 2015.

[11] H. Eichenbaum, "Barlow versus hebb: When is it time to abandon the notion of feature detectors and adopt the cell assembly as the unit of cognition?" *Neuroscience letters*, vol. 680, pp. 88–93, 2018.

[12] G. M. Shepherd, *Foundations of the neuron doctrine*. Oxford University Press, 2015.

[13] S. R. y Cajal, *Estructura de los centros nerviosos de las aves*. 1888.

[14] C. S. Sherrington, *The Integrative Action of the Nervous System*. Yale University Press, 1906, vol. 2.

[15] H. B. Barlow, "Summation and inhibition in the frog's retina," *The Journal of physiology*, vol. 119, no. 1, p. 69, 1953.

[16] K. A. Martin, "A brief history of the "feature detector"," *Cerebral cortex*, vol. 4, no. 1, pp. 1–7, 1994.

[17] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of physiology*, vol. 160, no. 1, p. 106, 1962.

[18] J. Duncan, "An adaptive coding model of neural function in prefrontal cortex," *Nature reviews neuroscience*, vol. 2, no. 11, pp. 820–829, 2001.

[19] E. K. Miller and J. D. Cohen, "An integrative theory of prefrontal cortex function," *Annual review of neuroscience*, vol. 24, no. 1, pp. 167–202, 2001.

[20] J. H. Morrison and P. R. Hof, "Life and death of neurons in the aging brain," *Science*, vol. 278, no. 5337, pp. 412–419, 1997.

[21] K. D. Harris, J. Csicsvari, H. Hirase, G. Dragoi, and G. Buzsáki, "Organization of cell assemblies in the hippocampus," *Nature*, vol. 424, no. 6948, pp. 552–556, 2003.

[22] D. O. Hebb, *The organization of behavior: A neuropsychological theory*. Wiley, New York, 1949.

[23] R. Lorente de Nó, "Studies in the structure of the cerebral cortex. i. the area entorhinalis," *Journal of Psychology and Neurology*, vol. 45, pp. 381–438, 1934.

[24] C. Sherrington, *Man on his nature*. Cambridge, 1940.

[25] T. V. Bliss and T. Lømo, "Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path," *The Journal of physiology*, vol. 232, no. 2, pp. 331–356, 1973.

[26] A. Artola and W. Singer, "Long-term potentiation and nmda receptors in rat visual cortex," *Nature*, vol. 330, no. 6149, pp. 649–652, 1987.

[27] A. Iriki, C. Pavlides, A. Keller, and H. Asanuma, "Long-term potentiation in the motor cortex," *Science*, vol. 245, no. 4924, pp. 1385–1387, 1989.

[28] J. Hirsch, G. Barrionuevo, and F. Crepel, "Homo-and heterosynaptic changes in efficacy are expressed in prefrontal neurons: An in vitro study in the rat," *Synapse*, vol. 12, no. 1, pp. 82–85, 1992.

[29] G. Buzsáki, "Large-scale recording of neuronal ensembles," *Nature neuroscience*, vol. 7, no. 5, pp. 446–451, 2004.

[30] M. P. Young and S. Yamane, "Sparse population coding of faces in the inferotemporal cortex," *Science*, vol. 256, no. 5061, pp. 1327–1331, 1992.

[31] N. Matsumoto, M. Okada, Y. Sugase-Miyamoto, S. Yamane, and K. Kawano, "Population dynamics of face-responsive neurons in the inferior temporal cortex," *Cerebral cortex*, vol. 15, no. 8, pp. 1103–1112, 2005.

[32] M. Rigotti *et al.*, "The importance of mixed selectivity in complex cognitive tasks," *Nature*, vol. 497, no. 7451, pp. 585–590, 2013.

[33] A. Farovik, R. J. Place, S. McKenzie, B. Porter, C. E. Munro, and H. Eichenbaum, "Orbitofrontal cortex encodes memories within value-based schemas and represents contexts that guide memory retrieval," *Journal of Neuroscience*, vol. 35, no. 21, pp. 8333–8344, 2015.

[34] M. A. Nicolelis, L. A. Baccala, R. C. Lin, and J. K. Chapin, "Sensorimotor encoding by synchronous neural ensemble activity at multiple levels of the somatosensory system," *Science*, vol. 268, no. 5215, pp. 1353–1358, 1995.

[35] G. M. Shepherd, *The synaptic organization of the brain*. Oxford university press, 2003.

[36] V. Braitenberg and A. Schüz, *Anatomy of the cortex: statistics and geometry*. Springer Science & Business Media, 2013, vol. 18.

[37] M. Abeles, *Corticonics: Neural circuits of the cerebral cortex*. Cambridge University Press, 1991.

[38] M. M. Karnani, M. Agetsuma, and R. Yuste, "A blanket of inhibition: Functional inferences from dense inhibitory connectivity," *Current opinion in Neurobiology*, vol. 26, pp. 96–102, 2014.

[39] G. Buzsáki, "Neural syntax: Cell assemblies, synapsembles, and readers," *Neuron*, vol. 68, no. 3, pp. 362–385, 2010.

[40] R. Yuste, R. Cossart, and E. Yaksi, "Neuronal ensembles: Building blocks of neural circuits," *Neuron*, vol. 112, no. 6, pp. 875–892, 2024.

[41] A. Malvache, S. Reichinnek, V. Villette, C. Haimerl, and R. Cossart, "Awake hippocampal reactivations project onto orthogonal neuronal assemblies," *Science*, vol. 353, no. 6305, pp. 1280–1283, 2016.

[42] E. A. A. Jones and L. M. Giocomo, "Neural ensembles in navigation: From single cells to population codes," *Current opinion in neurobiology*, vol. 78, p. 102 665, 2023.

[43] E. Pastalkova, V. Itskov, A. Amarasingham, and G. Buzsaki, "Internally generated cell assembly sequences in the rat hippocampus," *Science*, vol. 321, no. 5894, pp. 1322–1327, 2008.

[44] G. Dragoi and G. Buzsáki, "Temporal encoding of place sequences by hippocampal cell assemblies," *Neuron*, vol. 50, no. 1, pp. 145–157, 2006.

[45] V. Itskov, E. Pastalkova, K. Mizuseki, G. Buzsaki, and K. D. Harris, "Theta-mediated dynamics of spatial information in hippocampus," *Journal of Neuroscience*, vol. 28, no. 23, pp. 5959–5964, 2008.

[46] V. Villette, A. Malvache, T. Tressard, N. Dupuy, and R. Cossart, "Internally recurring hippocampal sequences as a population template of spatiotemporal information," *Neuron*, vol. 88, no. 2, pp. 357–366, 2015.

[47] C. Drieu, R. Todorova, and M. Zugaro, "Nested sequences of hippocampal assemblies during behavior support subsequent sleep replay," *Science*, vol. 362, no. 6415, pp. 675–679, 2018.

[48] M. El-Gaby *et al.*, "An emergent neural coactivity code for dynamic memory," *Nature neuroscience*, vol. 24, no. 5, pp. 694–704, 2021.

[49] A. K. Lee and M. A. Wilson, "Memory of sequential experience in the hippocampus during slow wave sleep," *Neuron*, vol. 36, no. 6, pp. 1183–1194, 2002.

[50] W. E. Skaggs and B. L. McNaughton, "Replay of neuronal firing sequences in rat hippocampus during sleep following spatial experience," *Science*, vol. 271, no. 5257, pp. 1870–1873, 1996.

[51] G. Dragoi and S. Tonegawa, "Preplay of future place cell sequences by hippocampal cellular assemblies," *Nature*, vol. 469, no. 7330, pp. 397–401, 2011.

[52] A. D. Grosmark and G. Buzsáki, "Diversity in neural firing dynamics supports both rigid and learned hippocampal sequences," *Science*, vol. 351, no. 6280, pp. 1440–1443, 2016.

[53] C. Leibold, "A model for navigation in unknown environments based on a reservoir of hippocampal sequences," *Neural Networks*, vol. 124, pp. 328–342, 2020.

[54] R. Huszár, Y. Zhang, H. Blockus, and G. Buzsáki, "Preconfigured dynamics in the hippocampus are guided by embryonic birthdate and rate of neurogenesis," *Nature Neuroscience*, vol. 25, no. 9, pp. 1201–1212, 2022.

[55] G. Girardeau, K. Benchenane, S. I. Wiener, G. Buzsáki, and M. B. Zugaro, "Selective suppression of hippocampal ripples impairs spatial memory," *Nature neuroscience*, vol. 12, no. 10, pp. 1222–1223, 2009.

[56] A. Fernández-Ruiz, A. Oliva, G. A. Nagy, A. P. Maurer, A. Berényi, and G. Buzsáki, "Entorhinal-ca3 dual-input control of spike timing in the hippocampus by theta-gamma coupling," *Neuron*, vol. 93, no. 5, pp. 1213–1226, 2017.

[57] I. Gridchyn, P. Schoenenberger, J. O'Neill, and J. Csicsvari, "Assembly-specific disruption of hippocampal replay leads to selective memory deficit," *Neuron*, vol. 106, no. 2, pp. 291–300, 2020.

[58] Y. Ikegaya *et al.*, "Synfire chains and cortical songs: Temporal modules of cortical activity," *Science*, vol. 304, no. 5670, pp. 559–564, 2004.

[59] K. D. Harris, "Neural signatures of cell assembly organization," *Nature Reviews Neuroscience*, vol. 6, no. 5, pp. 399–407, 2005.

[60] A. Luczak, P. Barthó, and K. D. Harris, "Spontaneous events outline the realm of possible sensory responses in neocortical populations," *Neuron*, vol. 62, no. 3, pp. 413–425, 2009.

[61] J.-e. K. Miller, I. Ayzenshtat, L. Carrillo-Reid, and R. Yuste, "Visual stimuli recruit intrinsically generated cortical ensembles," *Proceedings of the National Academy of Sciences*, vol. 111, no. 38, E4053–E4061, 2014.

[62] L. Carrillo-Reid, W. Yang, Y. Bando, D. S. Peterka, and R. Yuste, "Imprinting and recalling cortical ensembles," *Science*, vol. 353, no. 6300, pp. 691–694, 2016.

[63] L. Carrillo-Reid, S. Han, W. Yang, A. Akrouh, and R. Yuste, "Controlling visually guided behavior by holographic recalling of cortical ensembles," *Cell*, vol. 178, no. 2, pp. 447–457, 2019.

[64] J. N. MacLean, B. O. Watson, G. B. Aaron, and R. Yuste, "Internal dynamics determine the cortical response to thalamic stimulation," *Neuron*, vol. 48, no. 5, pp. 811–823, 2005.

[65] L. Carrillo-Reid, J.-e. K. Miller, J. P. Hamm, J. Jackson, and R. Yuste, "Endogenous sequential cortical activity evoked by visual stimuli," *Journal of neuroscience*, vol. 35, no. 23, pp. 8813–8828, 2015.

[66] J. Lines and R. Yuste, "Visually evoked neuronal ensembles reactivate during sleep," *bioRxiv*, pp. 2023–04, 2023.

[67] B. O. Watson, J. N. MacLean, and R. Yuste, "Up states protect ongoing cortical activity from thalamic inputs," *PLoS one*, vol. 3, no. 12, e3971, 2008.

[68] M. Agetsuma, J. P. Hamm, K. Tao, S. Fujisawa, and R. Yuste, "Parvalbumin-positive interneurons regulate neuronal ensembles in visual cortex," *Cerebral cortex*, vol. 28, no. 5, pp. 1831–1845, 2018.

[69] G. Molnár *et al.*, "Complex events initiated by individual spikes in the human cerebral cortex," *PLoS biology*, vol. 6, no. 9, e222, 2008.

[70] M. Hemberger, M. Shein-Idelson, L. Pammer, and G. Laurent, "Reliable sequential activation of neural assemblies by single pyramidal cells in a three-layered cortex," *Neuron*, vol. 104, no. 2, pp. 353–369, 2019.

[71] J. H. Marshel *et al.*, "Cortical layer–specific critical dynamics triggering perception," *Science*, vol. 365, no. 6453, eaaw5202, 2019.

[72] L. E. Russell *et al.*, "The influence of visual cortex on perception is modulated by behavioural state," *biorxiv*, p. 706 010, 2019.

[73] D. Marr, "A theory for cerebral neocortex," *Proceedings of the Royal society of London. Series B. Biological sciences*, vol. 176, no. 1043, pp. 161–234, 1970.

[74] D. Marr and G. S. Brindley, "Simple memory: A theory for archicortex," *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, vol. 262, no. 841, pp. 23–81, 1971.

[75] D. Marr, "A theory of cerebellar cortex," *The Journal of Physiology*, vol. 202, no. 2, pp. 437–470, 1969.

[76] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the national academy of sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

[77] R. McEliece, E. Posner, E. Rodemich, and S. Venkatesh, "The capacity of the hopfield associative memory," *IEEE Transactions on Information Theory*, vol. 33, no. 4, pp. 461–482, 1987.

[78] J. J. Hopfield and D. W. Tank, ""neural" computation of decisions in optimization problems," *Biological cybernetics*, vol. 52, no. 3, pp. 141–152, 1985.

[79] J. Bruck and J. W. Goodman, "On the power of neural networks for solving hard problems," *Journal of Complexity*, vol. 6, no. 2, pp. 129–135, 1990.

[80] L. G. Valiant, *Circuits of the Mind*. Oxford University Press on Demand, 2000.

[81] L. G. Valiant, "A neuroidal architecture for cognitive computation," *Journal of the ACM (JACM)*, vol. 47, no. 5, pp. 854–882, 2000.

[82] L. G. Valiant, "Memorization and association on a realistic neural model," *Neural Computation*, vol. 17, no. 3, pp. 527–555, 2005.

[83] J. S. Griffith, "On the stability of brain-like structures," *Biophysical journal*, vol. 3, no. 4, pp. 299–308, 1963.

[84] S.-M. Khaligh-Razavi and N. Kriegeskorte, "Deep supervised, but not unsupervised, models may explain it cortical representation," *PLoS computational biology*, vol. 10, no. 11, e1003915, 2014.

[85] D. L. Yamins, H. Hong, C. F. Cadieu, E. A. Solomon, D. Seibert, and J. J. DiCarlo, "Performance-optimized hierarchical models predict neural responses in higher visual cortex," *Proceedings of the national academy of sciences*, vol. 111, no. 23, pp. 8619–8624, 2014.

[86] R. M. Cichy, A. Khosla, D. Pantazis, A. Torralba, and A. Oliva, "Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence," *Scientific reports*, vol. 6, no. 1, p. 27 755, 2016.

[87] A. J. Kell, D. L. Yamins, E. N. Shook, S. V. Norman-Haignere, and J. H. McDermott, "A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy," *Neuron*, vol. 98, no. 3, pp. 630–644, 2018.

[88] C. Caucheteux and J.-R. King, "Brains and algorithms partially converge in natural language processing," *Nature Communications biology*, vol. 5, no. 1, p. 134, 2022.

[89] J. Merel, M. Botvinick, and G. Wayne, "Hierarchical motor control in mammals and machines," *Nature communications*, vol. 10, no. 1, p. 5489, 2019.

[90] T. P. Lillicrap, A. Santoro, L. Marris, C. J. Akerman, and G. Hinton, "Backpropagation and the brain," *Nature Reviews Neuroscience*, pp. 1–12, 2020.

[91] A. Saxe, S. Nelli, and C. Summerfield, "If deep learning is the answer, what is the question?" *Nature Reviews Neuroscience*, vol. 22, no. 1, pp. 55–67, 2021.

[92] P. Billingsley, *Probability and measure*. John Wiley & Sons, 2017.

[93] F. Mosteller, "On Some Useful "Inefficient" Statistics," *The Annals of Mathematical Statistics*, vol. 17, no. 4, pp. 377–408, 1946.

[94] C. H. Papadimitriou and S. S. Vempala, "Random projection in the brain and computation with assemblies of neurons," in *10th Innovations in Theoretical Computer Science Conference*, 2019.

[95] D. Mitropolsky, M. J. Collins, and C. H. Papadimitriou, "A biologically plausible parser," *To appear in TACL*, 2021.

[96] C. H. Papadimitriou, S. S. Vempala, D. Mitropolsky, M. Collins, and W. Maass, "Brain computation by assemblies of neurons," *Proceedings of the National Academy of Sciences*, vol. 117, no. 25, pp. 14 464–14 472, 2020.

[97] D. Mitropolsky, A. Ejaz, M. Shi, M. Yannakakis, and C. H. Papadimitriou, "Center-embedding and constituency in the brain and a new characterization of context-free languages," *arXiv preprint arXiv:2206.13217*, 2022.

[98] M. Dabagia, S. S. Vempala, and C. Papadimitriou, "Assemblies of neurons learn to classify well-separated distributions," in *Conference on Learning Theory*, PMLR, 2022, pp. 3685–3717.

[99] R. Axel, "Q & A," *Neuron*, vol. 99, pp. 1110–1112, 2018.

[100] G. Buzsáki, *The Brain from Inside Out*. Oxford University Press, 2019.

[101] S. T. Piantadosi, J. B. Tenenbaum, and N. D. Goodman, "The logical primitives of thought: Empirical foundations for compositional cognitive models.," *Psychological review*, vol. 123, no. 4, p. 392, 2016.

[102] R. Q. Quiroga, "Neuronal codes for visual perception and memory," *Neuropsychologia*, vol. 83, pp. 227–241, 2016.

[103] S. T. Piantadosi, J. B. Tenenbaum, and N. D. Goodman, "Bootstrapping in a language of thought: A formal model of numerical concept learning," *Cognition*, vol. 123, no. 2, pp. 199–217, 2012.

[104] G. Buzsáki, "Neural syntax: Cell assemblies, synapsembles, and readers," *Neuron*, vol. 68, no. 3, 2010.

[105] F. d'Amore, D. Mitropolsky, P. Crescenzi, E. Natale, and C. H. Papadimitriou, "Planning with biological neurons and synapses," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, 2022, pp. 21–28.

[106] P. Erdős and A. Rényi, "On the evolution of random graphs," *Publ. Math. Inst. Hung. Acad. Sci*, vol. 5, no. 1, pp. 17–60, 1960.

[107] R. Legenstein, W. Maass, C. H. Papadimitriou, and S. S. Vempala, "Long-term memory and the densest k-subgraph problem," in *Proc. of 9th Innovations in Theoretical Computer Science (ITCS) conference, Cambridge, USA, Jan 11-14. 2018*, 2018.

[108] A. Rangamani and A. Gandhi, "Supervised learning with brain assemblies," *Preprint, private communication*, 2020.

[109] L. G. Valiant, *Circuits of the mind*. Oxford University Press, 1994, ISBN: 978-0-19-508926-4.

[110] L. G. Valiant, "A neuroidal architecture for cognitive computation," *J. ACM*, vol. 47, no. 5, pp. 854–882, 2000.

[111] V. Feldman and L. G. Valiant, "Experience-induced neural circuits that achieve high capacity," *Neural Computation*, vol. 21, no. 10, pp. 2715–2754, 2009.

[112] T. P. Lillicrap, D. Cownden, D. B. Tweed, and C. J. Akerman, "Random synaptic feedback weights support error backpropagation for deep learning," in *Nature communications*, 2016.

[113] J. Sacramento, R. P. Costa, Y. Bengio, and W. Senn, "Dendritic error backpropagation in deep cortical microcircuits," *arXiv preprint arXiv:1801.00062*, 2017.

[114] J. Guerguiev, T. P. Lillicrap, and B. A. Richards, "Towards deep learning with segregated dendrites," *ELife*, vol. 6, e22901, 2017.

[115] J. Sacramento, R. Ponte Costa, Y. Bengio, and W. Senn, "Dendritic cortical microcircuits approximate the backpropagation algorithm," *Advances in Neural Information Processing Systems*, vol. 31, pp. 8721–8732, 2018.

[116] J. C. Whittington and R. Bogacz, "Theories of error back-propagation in the brain," *Trends in cognitive sciences*, vol. 23, no. 3, pp. 235–250, 2019.

[117] J. C. Magee and C. Grienberger, "Synaptic plasticity forms and functions," *Annual review of neuroscience*, vol. 43, pp. 95–117, 2020.

[118] A. Payeur, J. Guerguiev, F. Zenke, B. A. Richards, and R. Naud, "Burst-dependent synaptic plasticity can coordinate learning in hierarchical circuits," *Nature neuroscience*, pp. 1–10, 2021.

[119] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," *Advances in neural information processing systems*, vol. 20, 2007.

[120] J. Sugar and M.-B. Moser, "Episodic memory: Neuronal codes for what, where, and when," *Hippocampus*, vol. 29, no. 12, pp. 1190–1205, 2019.

[121] J. L. Bellmund, I. Polti, and C. F. Doeller, "Sequence memory in the hippocampal–entorhinal region," *Journal of Cognitive Neuroscience*, vol. 32, no. 11, pp. 2056–2070, 2020.

[122] M. Sipser, "Introduction to the theory of computation," *ACM Sigact News*, vol. 27, no. 1, pp. 27–29, 1996.

[123] A. Sik, M. Penttonen, A. Ylinen, and G. Buzsáki, "Hippocampal ca1 interneurons: An in vivo intracellular labeling study," *Journal of Neuroscience*, vol. 15, no. 10, pp. 6651–6665, 1995.

[124] S. Jinno *et al.*, "Neuronal diversity in gabaergic long-range projections from the hippocampus," *Journal of Neuroscience*, vol. 27, no. 33, pp. 8790–8804, 2007.

[125] S. Zhang *et al.*, "Long-range and local circuits for top-down modulation of visual cortex processing," *Science*, vol. 345, no. 6197, pp. 660–665, 2014.

[126] L. Roux and G. Buzsáki, "Tasks for inhibitory interneurons in intact brain circuits," *Neuropharmacology*, vol. 88, pp. 10–23, 2015.

[127] V. Feldman and L. G. Valiant, "Experience-induced neural circuits that achieve high capacity," *Neural computation*, vol. 21, no. 10, pp. 2715–2754, 2009.

[128] T. P. Lillicrap, D. Cownden, D. B. Tweed, and C. J. Akerman, "Random synaptic feedback weights support error backpropagation for deep learning," *Nature communications*, vol. 7, no. 1, p. 13 276, 2016.

[129] C. Eliasmith *et al.*, "A large-scale model of the functioning brain," *science*, vol. 338, no. 6111, pp. 1202–1205, 2012.

[130] Y. Cui, S. Ahmad, and J. Hawkins, "Continuous online sequence learning with an unsupervised neural network model," *Neural computation*, vol. 28, no. 11, pp. 2474–2504, 2016.

[131] J. C. Whittington *et al.*, "The tolman-eichenbaum machine: Unifying space and relational memory through generalization in the hippocampal formation," *Cell*, vol. 183, no. 5, pp. 1249–1263, 2020.

[132] E. A. Maguire, E. R. Valentine, J. M. Wilding, and N. Kapur, "Routes to remembering: The brains behind superior memory," *Nature neuroscience*, vol. 6, no. 1, pp. 90–95, 2003.

[133] M. H. Kelly and S. Martin, "Domain-general abilities applied to domain-specific tasks: Sensitivity to probabilities in perception, cognition, and language," *Lingua*, vol. 92, pp. 105–140, 1994.

[134] B. J. Knowlton, L. R. Squire, and M. A. Gluck, "Probabilistic classification learning in amnesia.," *Learning & Memory*, vol. 1, no. 2, pp. 106–120, 1994.

[135] J. R. Saffran, E. L. Newport, and R. N. Aslin, "Word segmentation: The role of distributional cues," *Journal of Memory and Language*, vol. 35, no. 4, pp. 606–621, 1996.

[136] L. J. Batterink, K. A. Paller, and P. J. Reber, "Understanding the neural bases of implicit and statistical learning," *Topics in Cognitive Science*, vol. 11, no. 3, pp. 482–503, 2019.

[137] L. Hasher and R. T. Zacks, "Automatic processing of fundamental information: The case of frequency of occurrence.," *American Psychologist*, vol. 39, no. 12, p. 1372, 1984.

[138] N. B. Turk-Browne, J. A. Jungé, and B. J. Scholl, "The automaticity of visual statistical learning.," *Journal of Experimental Psychology: General*, vol. 134, no. 4, p. 552, 2005.

[139] R. Frost, B. C. Armstrong, N. Siegelman, and M. H. Christiansen, "Domain generality versus modality specificity: The paradox of statistical learning," *Trends in Cognitive Sciences*, vol. 19, no. 3, pp. 117–125, 2015.

[140] R. Joober and S. Karama, "Randomness and nondeterminism: From genes to free will with implications for psychiatry," *Journal of Psychiatry & Neuroscience: JPN*, vol. 46, no. 4, E500, 2021.

[141] T. Branco and K. Staras, "The probability of neurotransmitter release: Variability and feedback control at single synapses," *Nature Reviews Neuroscience*, vol. 10, no. 5, pp. 373–383, 2009.

[142] R. C. Cannon, C. O'Donnell, and M. F. Nolan, "Stochastic ion channel gating in dendritic neurons: Morphology dependence and probabilistic synaptic activation of dendritic spikes," *PLoS Computational Biology*, vol. 6, no. 8, e1000886, 2010.

[143] D. J. Tolhurst, J. A. Movshon, and A. F. Dean, "The statistical reliability of signals in single neurons in cat and monkey visual cortex," *Vision Researchesearch*, vol. 23, no. 8, pp. 775–785, 1983.

[144] R. Azouz and C. M. Gray, "Cellular mechanisms contributing to response variability of cortical neurons in vivo," *Journal of Neuroscience*, vol. 19, no. 6, pp. 2209–2223, 1999.

[145] M. Rudolph and A. Destexhe, "Do neocortical pyramidal neurons display stochastic resonance?" *Journal of Computational Neuroscience*, vol. 11, pp. 19–42, 2001.

[146] J. S. Anderson, I. Lampl, D. C. Gillespie, and D. Ferster, "The contribution of noise to contrast invariance of orientation tuning in cat visual cortex," *Science*, vol. 290, no. 5498, pp. 1968–1972, 2000.

[147] P. Hoyer and A. Hyvärinen, "Interpreting neural response variability as monte carlo sampling of the posterior," *Advances in Neural Information Processing Systems*, vol. 15, 2002.

[148] N. Daw and A. Courville, "The pigeon as particle filter," *Advances in Neural Information Processing Systems*, vol. 20, pp. 369–376, 2008.

[149] L. Shi, N. H. Feldman, and T. L. Griffiths, "Performing bayesian inference with exemplar models," in *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 30, 2008.

[150] R. Levy, F. Reali, and T. Griffiths, "Modeling the effects of memory on human online sentence processing with particle filters," *Advances in Neural Information Processing Systems*, vol. 21, 2008.

[151] S. Gershman, E. Vul, and J. Tenenbaum, "Perceptual multistability as markov chain monte carlo inference," *Advances in Neural Information Processing Systems*, vol. 22, 2009.

[152] L. Buesing, J. Bill, B. Nessler, and W. Maass, "Neural dynamics as sampling: A model for stochastic computation in recurrent networks of spiking neurons," *PLoS Computational Biology*, vol. 7, no. 11, e1002211, 2011.

[153] M. Dabagia, C. Papadimitriou, and S. Vempala, "Computation with sequences of assemblies in a model of the brain," in *International Conference on Algorithmic Learning Theory*, PMLR, 2024, pp. 499–504.

[154] E. Oja, "Simplified neuron model as a principal component analyzer," *Journal of Mathematical Biology*, vol. 15, pp. 267–273, 1982.

[155] W. Simpson and J. F. Voss, "Psychophysical judgments of probabilistic stimulus sequencies.," *Journal of Experimental Psychology*, vol. 62, no. 4, p. 416, 1961.

[156] D. E. Erlick, "Absolute judgments of discrete quantities randomly distributed over time.," *Journal of Experimental Psychology*, vol. 67, no. 5, p. 475, 1964.

[157] G. F. Pitz, "Response variables in the estimation of relative frequency," *Perceptual and Motor Skills*, vol. 21, no. 3, pp. 867–873, 1965.

[158] G. F. Pitz, "The sequential judgment of proportion," *Psychonomic Science*, vol. 4, no. 12, pp. 397–398, 1966.

[159] E. Brunswik, "Probability as a determiner of rat behavior.," *Journal of Experimental Psychology*, vol. 25, no. 2, p. 175, 1939.

[160] W. K. Estes, "Probability learning," in *Categories of Human Learning*, Elsevier, 1964, pp. 89–128.

[161] M. E. Bitterman, "Phyletic differences in learning.," *American Psychologist*, vol. 20, no. 6, p. 396, 1965.

[162] C. R. Gallistel, *The Organization of Learning.* The MIT Press, 1990.

[163] J. R. Saffran, R. N. Aslin, and E. L. Newport, "Statistical learning by 8-month-old infants," *Science*, vol. 274, no. 5294, pp. 1926–1928, 1996.

[164] M. D. Hauser, E. L. Newport, and R. N. Aslin, "Segmentation of the speech stream in a non-human primate: Statistical learning in cotton-top tamarins," *Cognition*, vol. 78, no. 3, B53–B64, 2001.

[165] J. Fiser and R. N. Aslin, "Unsupervised statistical learning of higher-order spatial structures from visual scenes," *Psychological Science*, vol. 12, no. 6, pp. 499–504, 2001.

[166] N. Furl, S. Kumar, K. Alter, S. Durrant, J. Shawe-Taylor, and T. D. Griffiths, "Neural prediction of higher-order auditory sequence statistics," *Neuroimage*, vol. 54, no. 3, pp. 2267–2277, 2011.

[167] E. Paraskevopoulos, A. Kuchenbuch, S. C. Herholz, and C. Pantev, "Statistical learning effects in musicians and non-musicians: An meg study," *Neuropsychologia*, vol. 50, no. 2, pp. 341–349, 2012.

[168] T. Daikoku, Y. Yatomi, and M. Yumoto, "Implicit and explicit statistical learning of tone sequences across spectral shifts," *Neuropsychologia*, vol. 63, pp. 194–204, 2014.

[169] T. Daikoku, Y. Yatomi, and M. Yumoto, "Statistical learning of music-and language-like sequences and tolerance for spectral shifts," *Neurobiology of Learning and Memory*, vol. 118, pp. 8–19, 2015.

[170] K. McNealy, J. C. Mazziotta, and M. Dapretto, "Cracking the language code: Neural mechanisms underlying speech parsing," *Journal of Neuroscience*, vol. 26, no. 29, pp. 7629–7639, 2006.

[171] T. Cunillera *et al.*, "Time course and functional neuroanatomy of speech segmentation in adults," *Neuroimage*, vol. 48, no. 3, pp. 541–553, 2009.

[172] E. A. Karuza, E. L. Newport, R. N. Aslin, S. J. Starling, M. E. Tivarus, and D. Bavelier, "The neural correlates of statistical learning in a word segmentation task: An fmri study," *Brain and Language*, vol. 127, no. 1, pp. 46–54, 2013.

[173] N. B. Turk-Browne, B. J. Scholl, M. M. Chun, and M. K. Johnson, "Neural evidence of statistical learning: Efficient detection of visual regularities without awareness," *Journal of Cognitive Neuroscience*, vol. 21, no. 10, pp. 1934–1945, 2009.

[174] E. A. Karuza, L. L. Emberson, M. E. Roser, D. Cole, R. N. Aslin, and J. Fiser, "Neural signatures of spatial statistical learning: Characterizing the extraction of structure from complex visual scenes," *Journal of Cognitive Neuroscience*, vol. 29, no. 12, pp. 1963–1976, 2017.

[175] C. M. Conway and M. H. Christiansen, "Modality-constrained statistical learning of tactile, visual, and auditory sequences.," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 31, no. 1, p. 24, 2005.

[176] N. Siegelman and R. Frost, "Statistical learning as an individual ability: Theoretical perspectives and empirical evidence," *Journal of Memory and Language*, vol. 81, pp. 105–120, 2015.

[177] C. M. Conway and M. H. Christiansen, "Statistical learning within and between modalities: Pitting abstract against stimulus-specific representations," *Psychological Science*, vol. 17, no. 10, pp. 905–912, 2006.

[178] N. B. Turk-Browne, B. J. Scholl, M. K. Johnson, and M. M. Chun, "Implicit perceptual anticipation triggered by statistical learning," *Journal of Neuroscience*, vol. 30, no. 33, pp. 11 177–11 187, 2010.

[179] A. C. Schapiro, L. V. Kustner, and N. B. Turk-Browne, "Shaping of object representations in the human medial temporal lobe based on temporal regularities," *Current biology*, vol. 22, no. 17, pp. 1622–1627, 2012.

[180] N. V. Covington, S. Brown-Schmidt, and M. C. Duff, "The necessity of the hippocampus for statistical learning," *Journal of Cognitive Neuroscience*, vol. 30, no. 5, pp. 680–697, 2018.

[181] M. E. Reid and S. S. Vempala, "The k-cap process on geometric random graphs," in *Proceedings of Thirty Sixth Conference on Learning Theory*, G. Neu and L. Rosasco, Eds., ser. Proceedings of Machine Learning Research, vol. 195, PMLR, 2023, pp. 3469–3509.

[182] D. Mitropolsky and C. H. Papadimitriou, "The architecture of a biologically plausible language organ," *arXiv preprint arXiv:2306.15364*, 2023.

[183] R. P. Rao, B. A. Olshausen, and M. S. Lewicki, *Probabilistic Models of the Brain: Perception and Neural Function*. MIT Press, 2002.

[184] K. Doya, *Bayesian Brain: Probabilistic Approaches to Neural Coding*. MIT press, 2007.

[185] T. S. Lee and D. Mumford, "Hierarchical bayesian inference in the visual cortex," *JOSA a*, vol. 20, no. 7, pp. 1434–1448, 2003.

[186] D. Kersten, P. Mamassian, and A. Yuille, "Object perception as bayesian inference," *Annu. Rev. Psychol.*, vol. 55, pp. 271–304, 2004.

[187] K. P. Körding and D. M. Wolpert, "Bayesian integration in sensorimotor learning," *Nature*, vol. 427, no. 6971, pp. 244–247, 2004.

[188] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for Boltzmann machines," *Cognitive Science*, vol. 9, no. 1, pp. 147–169, 1985.

[189] A. Pouget, P. Dayan, and R. Zemel, "Information processing with population codes," *Nature Reviews Neuroscience*, vol. 1, no. 2, pp. 125–132, 2000.

[190] N. A. Lynch, "Multi-neuron representations of hierarchical concepts in spiking neural networks," *arXiv preprint arXiv:2401.04628*, 2024.

[191] B. de Boysson-Bardies and M. M. Vihman, "Adaptation to language: Evidence from babbling and first words in four languages," *Language*, vol. 67, no. 2, pp. 297–319, 1991.

[192] D. Kahneman, *Thinking, Fast and Slow*. MacMillan, 2011.

[193] P. K. Kuhl, "Early language acquisition: Cracking the speech code," *Nature Reviews Neuroscience*, vol. 5, no. 11, pp. 831–843, 2004.