

A Spiking Neural Model of Episodic Memory Encoding and Replay in Hippocampus

by

Oliver Trujillo

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Math
in
Computer Science

Waterloo, Ontario, Canada, 2014

© Oliver Trujillo 2014

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

As we experience life, we are constantly creating new memories, and the hippocampus plays an important role in the formation and recall of these episodic memories. This thesis begins by describing the neural mechanisms that make the hippocampus ideally suited for memory formation, consolidation and recall. We then describe a biologically plausible spiking-neuron model of the hippocampus' role in episodic memory. The model includes a mechanism for generating temporal indexing vectors, for associating these indices with experience vectors to form episodes, and for replaying the original experience vectors in sequence when prompted. The model also associates these episodes with context vectors using synaptic plasticity, such that it is able to retrieve an episodic memory associated with a given context and replay it, even after long periods of time. We demonstrate the model's ability to experience sequences of sensory information in the form of semantic pointer vectors and replay the same sequences later, comparing the results to experimental data. In particular, the model runs a T-maze experiment in which a simulated rat is forced to choose between left or right at a decision point, during which the neural firing patterns of the model's place cells closely match those found in real rats performing the same task. We demonstrate that the model is robust to both spatial and non-spatial data, since the vector representation of the input data remains the same in either case. To our knowledge, this is the first spiking neural hippocampal model that can encode and recall sequences of both spatial and non-spatial data, while exhibiting temporal and spatial selectivity at a neural level.

Acknowledgements

Thank you to my supervisors, Chris Eliasmith and Jeff Orchard, who have gone above and beyond to make me feel comfortable at the University of Waterloo. Their guidance and insights have made writing this thesis possible. Thanks also go to my readers, Matthijs van der Meer and Dan Lizotte, for volunteering their time into this research and giving useful feedback.

I would like to thank all my colleagues in the Computational Neuroscience Research Group, particularly Terry Stewart, Xuan Choo, Trevor Bekolay, Travis DeWolf, Aaron Voelker, Aziz Hurzook, Eric Crawford, and Eric Hunsberger for their help in getting me up to speed with Nengo and the NEF, and for their collaborations and company over the past two years.

I am extremely grateful for all the friendships I've made during my time in Waterloo. (In no particular order) Stephen Kiazzyk, Hella Hoffmann, Dean Shaft, Valerie Sugarman, Cecylia Bocovich, Andrew Arnold, Jack Thomas, David Dufour, Marianna Rapoport, Brent Komer, Rhiannon Rose, Aaron and Christina Moss, Kyle Menegaldo, and the members of the David R. Cheriton School of Hockey Superfriends, thank you all for making Waterloo an amazing and fun place.

Finally, I'd like to thank my mom Sandra, my dad Julio, and my wife Laura for their unending love and support. Laura, you are my biggest supporter and best friend.

Dedication

I dedicate this thesis to my mom, who despite my best efforts to throw things away, has kept files of all my schoolwork going back to kindergarten. I hope this thesis is almost as entertaining to read as my first grade journal entries.

Table of Contents

List of Figures	ix
List of Tables	xi
1 Introduction	2
1.1 Thesis Organization	3
2 Episodic Memory and the Hippocampus	5
2.1 Episodic memory	5
2.2 Hippocampal Anatomy	6
2.2.1 Entorhinal Cortex	7
2.2.2 Dentate Gyrus	8
2.2.3 CA3	8
2.2.4 CA1	8
2.2.5 Subiculum	9
2.2.6 Medial Septum	9
2.3 Hippocampal Phenomena	9
2.3.1 Grid Cells and Place Cells	9
2.3.2 Remapping	11
2.3.3 Time cells and other hippocampal cell types	11
2.3.4 Replay	12
2.3.5 Theta Rhythm	14

3	Background Methods	16
3.1	Representing Concepts as Vectors	16
3.1.1	Semantic Pointers	17
3.2	The NEF	18
3.2.1	Vector Representation	18
3.2.2	Vector Transformation	23
3.2.3	Population Dynamics	24
3.2.4	Memory in the NEF	25
3.2.5	Learning, Memory, and Synaptic Plasticity	27
3.3	Nengo	29
4	A Neural Model of the Hippocampus	31
4.1	Motivation	31
4.2	Model Description	32
4.2.1	Model Overview	32
4.2.2	Sparsification (DG)	34
4.2.3	Index Generation (CA3)	34
4.2.4	Associative Memory (CA3)	37
4.2.5	Unbinding (CA1)	39
4.2.6	Learning for Consolidation (CA3)	40
4.2.7	Functional Overview	43
4.2.8	Neural Parameters	44
5	Simulations and Results	45
5.1	Replay of Concept Vectors	45
5.1.1	Replay from Neural Activity	45
5.1.2	Replay from Connection Weights	46
5.2	Replay of a Spatial Trajectory	49
5.3	Time Cells	50
5.4	Theta	54

6 Discussion and Future Work	56
6.1 Contribution	56
6.2 Potential Model Extensions	57
6.3 Further Discussion	57
References	59

List of Figures

2.1	Taxonomy of memory systems	6
2.2	Connectivity in the Hippocampus	7
2.3	Grid cell and place cell spatial tuning	10
2.4	Place cell activity	10
2.5	Place cell remapping	12
2.6	Time cell activity	13
2.7	Forward replay of place cells	14
3.1	Encoding in the NEF	20
3.2	1D LIF tuning curves	21
3.3	2D LIF tuning curve	22
3.4	Stable point of a 1D integrator	25
3.5	Gated Difference Unit Architecture	26
3.6	Gated Difference Unit Results	27
4.1	Model overview	33
4.2	Sparsification in DG	35
4.3	CA3 Indexing Architecture	38
4.4	Counting in CA3	39
4.5	CA3 Memory Architecture	40
4.6	CA1 Unbinding Architecture	41

4.7	CA3 Learning Architecture	42
4.8	Functional Model Architecture	43
5.1	Basic replay in model	46
5.2	Context switching replay	47
5.3	Spike-trains showing forward replay in model vs rats	48
5.4	Sequence length vs. accuracy	48
5.5	Simulated T maze	50
5.6	T maze results	51
5.7	T maze position reconstruction	52
5.8	Time cells in model vs rats	53
5.9	Theta oscillations in model vs rats	55

List of Tables

4.1 Neural parameters	44
---------------------------------	----

“I read that it’s important to sleep. While you sleep, the hippocampus in your brain replays things that happened during the day, e.g. what you studied. So therefore it remembers it for you.” - Jaclyn Moriarty, *The Ghosts of Ashbury High*

Chapter 1

Introduction

The hippocampus is one of the most highly studied parts of the brain, yet there is still a surprising amount of debate as to its exact function. Many researchers over the years have performed studies on both humans and animals in attempts to discover its function. In particular, two separate views of hippocampal function have gained prominence. In humans, the hippocampus was found to be important to *episodic memory*, which is the memory of events or ‘episodes,’ while other groups of researchers found evidence linking the hippocampus to cognitive mapping and spatial learning, particularly in rats.

In 1957, the hippocampus was implicated in episodic memory in a famous study when Henry Molaison (until recently known as patient HM to preserve his anonymity while he was alive) had both hippocampi removed in an operation to relieve seizures [90]. Neuroscientists then had the opportunity to perform tests on HM, finding that his short-term working memory functioned as normal, but he was unable to form new episodic memories. Any previous memories he had acquired before the surgery were retained, but any new experiences from after the surgery were unable to be consolidated into long-term memory.

In particular, when introduced to a person, he would say “hello,” but not “nice to meet you.” [47]. He had become accustomed to not acknowledging whether or not he knew a person, since he could not remember whether he was just meeting them for the first time or if he had known them for many years. This loss of ability for HM to form new memories naturally supported the hypothesis that the hippocampus is involved in episodic memory formation.

Later, in 1971, Nobel Prize winning neuroscientist John O’Keefe and Jonathon Dostrovsky [78] were developing an alternate theory of hippocampal function. They discovered *place cells*, which are neurons in the hippocampus whose firing rates are correlated with a

specific location, or place, in an environment. This led to the theory that the hippocampus plays the role of a *cognitive map* [79], by which people and animals maintain maps of environments. This hypothesis, along with Morris' later findings that the hippocampus is involved in spatial learning [72], seemingly conflicted with the episodic memory evidence found in HM.

Further studies on patient HM [70] [19] and other patients like him [86] contributed to the prevalence of the episodic memory theory, eventually leading to a more general theory of the hippocampus encoding both spatial and non-spatial information [29] [66] [26] [64]. This view stated that the hippocampus does indeed help us form cognitive maps, but that it does so as part of a more general memory formation process [30], and that in addition to place cells, it also represents higher level conceptual memories.

The purpose of this thesis is to detail a neural model of episodic memory formation and recall in hippocampus. As we will go on to describe, the *neural engineering framework*, or NEF [33], gives us a natural way of representing either spatial or non-spatial information in a vector space to represent episodic memories in a biologically plausible way. We also use *semantic pointers*, or SPs [32] to represent concepts as HRR vectors in spiking neural networks. We will give the background required to construct the model and an overview of the results of running simulations on the model.

1.1 Thesis Organization

Chapter 2, Episodic Memory and the Hippocampus, goes into more detail about the definition of episodic memory and how it differs from other types of memory. It also gives details about how episodic memory functionality is thought to map onto mammalian neural anatomy, particularly the anatomy of the hippocampal formation, and it describes some phenomena associated with episodic memory in the hippocampus.

Chapter 3, Background Methods, gives an overview of the background methods used in the model, including concept representation with SPs, vector representation in spiking neurons with the NEF, memory in the NEF, and the Nengo modelling software.

Chapter 4, A Neural Model of the Hippocampus, describes the steps taken to construct the neural model of episodic memory formation, and each of the functional elements in detail.

Chapter 5, Simulations and Results, details the simulations performed, a variety of hippocampal experiments, and gives comparisons between our simulation results and those experiments.

Finally, **Chapter 6, Discussion and Future Work**, gives some comments on our results and some options for future extensions of the project.

Chapter 2

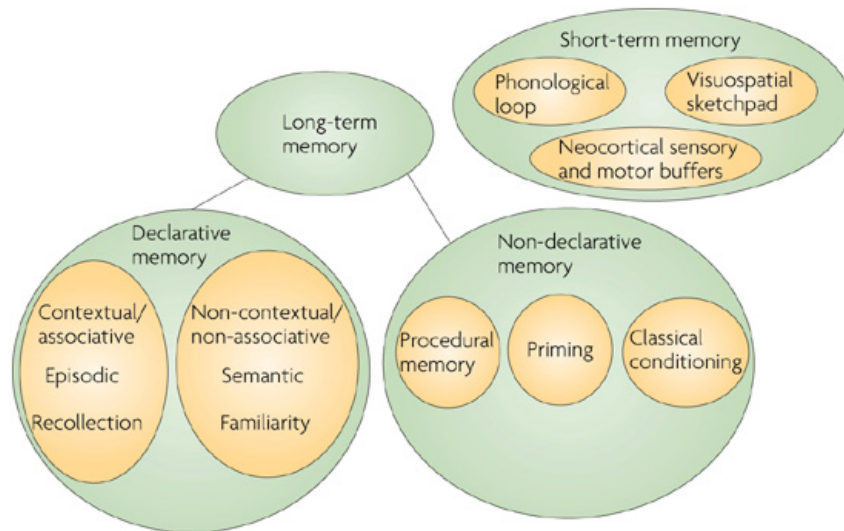
Episodic Memory and the Hippocampus

2.1 Episodic memory

Cognitive science has given way to several distinct ways of describing what we colloquially refer to as ‘memory.’ The *episodic memory* system, one particular form of memory, was first defined by Endel Tulving in 1972 [99], as a system that “receives and stores information about temporally dated episodes or events.” This definition by Tulving can be contrasted to *semantic memory*, which is general knowledge about the world [98]. For example, remembering the events that happened during a walk home from work the previous day is a kind of episodic memory, while remembering that Ottawa is the capital of Canada is a semantic memory.

The consensus among memory researchers points to the existence of multiple memory systems in our brains [100], often distinguishing between *working memory*, which is what we use to actively keep recent information in our minds [5], and long-term memory, which is further subdivided into episodic memory, semantic memory, and *procedural memory* [16], which is knowledge about how perform a task, such as riding a bicycle. There is still no clear agreement as to where to draw the lines between the various memory systems in our brains, but this breakdown fits well with both subjective experience and neurological evidence [30]. Figure 2.1 shows a subdivision of the brain’s memory systems.

Since Tulving’s original definition, the concept of episodic memory has evolved to include details about context and recall that have since been observed [47]. It is generally



Nature Reviews | Neuroscience

Figure 2.1: Taxonomy of memory systems. Long-term memory is divided into *declarative* and *non-declarative* memory, while declarative is further subdivided into semantic and episodic memory. Non-declarative memory includes procedural memory, classical conditioning, which is a more instinctual form of memory related to utility, and priming, which is an unconscious recognition of words or objects that is separate from semantic memory. Short-term memory includes temporary storage in the sensory and motor areas of cortex, as well as a phonological loop (working memory rehearsal) and a visuospatial sketchpad, which is thought of as a short-term storage space for image ‘snapshots.’ Adapted from [10].

accepted that retrieval of an episodic memory will answer the questions, “What was observed at time T in context C?” As we will now describe, the anatomy of the hippocampus makes it well-suited to encode information in a way that will allow us to answer queries of this form.

2.2 Hippocampal Anatomy

The hippocampus is a seahorse-shaped part of the forebrain, located in the medial temporal lobe. Extensive research has been performed on the hippocampus determining its critical role in declarative, and specifically episodic memory along with the entorhinal cortex [101]

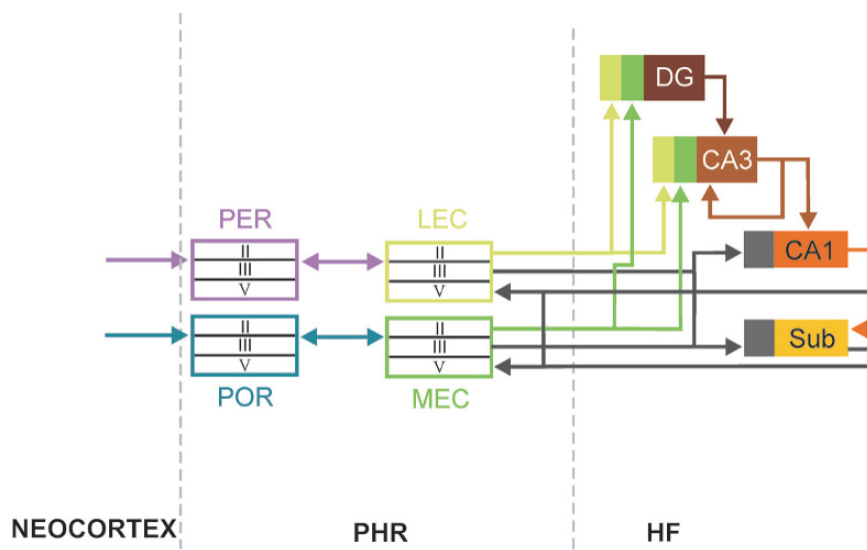


Figure 2.2: Diagram showing neural connectivity from neocortex to entorhinal cortex (PHR or parahippocampal region) to hippocampus (HF or hippocampal formation). The perirhinal (PER) and postrhinal (POR) cortices provide parallel input streams to the lateral (LEC) and medial (MEC) entorhinal cortex. Layers II and III of LEC and MEC project to all hippocampal subregions via the perforant path, while regions CA1 and SUB project back to layer V of LEC and MEC. From [21].

[94] [6]. It is strongly connected to the entorhinal cortex (EC) and is further subdivided into four primary regions: dentate gyrus (DG), CA3, CA1, and subiculum (SUB). Information flows roughly in a loop through the hippocampal subregions, following the order $EC \rightarrow DG \rightarrow CA3 \rightarrow CA1 \rightarrow SUB \rightarrow EC$. Figure 2.2 gives a connectivity diagram between the various subregions of the hippocampal formation and its cortical connections.

2.2.1 Entorhinal Cortex

Entorhinal cortex (EC) is the primary input to and output from hippocampus, and is often considered a part of the hippocampal formation. It can be thought of as a bridge between the neocortex and the hippocampus proper. There is a distinction made between the medial (MEC) and lateral (LEC) parts of EC, with the lateral part receiving input from the perirhinal and olfactory cortices and amygdala, and the medial part having projections from the postrhinal and visual association cortices [107]. There are projections from EC

to all parts of the hippocampus through the perforant path, and projections from areas CA1 and SUB back to EC's 'output' pathways.

The entorhinal cortex is thought to contain a representation of highly processed sensory information, with a division between spatial information in MEC and non-spatial information in LEC [27]. A number of functionally specialized cell types exist in EC. In addition to grid cells in MEC, which we will go on to describe below, neurons have been found that encode the velocity of the animal and the direction the animal is facing (head direction) [89]. It has recently been shown that these functionally specialized neurons such as grid cells do in fact project directly into hippocampus [109].

2.2.2 Dentate Gyrus

The dentate gyrus is a part of the hippocampal formation containing small granule cells. It receives input from the entorhinal cortex (LEC II and MEC II) and projects into CA3 via mossy fibres. It is one of only two brain areas in which adult *neurogenesis* takes place [1]. It is thought that these newly formed neurons help the dentate gyrus perform its theorized function of pattern separation, or sparsification [106] [7] [25]. That is, it has a large number of sparsely firing neurons that are sensitive to very particular patterns, that can for example allow us to differentiate between two similar but not identical environments.

2.2.3 CA3

CA3 is the 'deepest' part of the hippocampus, in that it is located furthest from the entorhinal cortex. It is composed of pyramidal cells similar to those found in neocortex. CA3 is distinguished from the other hippocampal subregions by its strongly recurrent connections, which are not present elsewhere in the hippocampus [48]. It projects into area CA1 through the Schaffer collaterals. The recurrent connections, along with considerable evidence that CA3 pyramidal cells exhibit synaptic plasticity in the form of *long-term potentiation* (LTP) and *long-term depression* (LTD) [61] [24], suggest that CA3 is heavily involved in the learning and formation of new episodic memories. The many forms of synaptic plasticity, including LTP and LTD are described in more detail in section 3.2.5.

2.2.4 CA1

Next we have CA1, which is located in between CA3 and SUB. It takes input from the CA3 Schaffer collaterals and from EC via the perforant path [54], and along with CA3 is thought

to play a role in the formation and recall of episodic memories [74]. It is also sometimes theorized to perform a comparison function between the EC input and the CA3 input [49] [95], possibly restricting incomplete or incorrect memory retrieval from propagating back to EC. *Place cells* (see section 2.3.1) are found both here and in CA3.

2.2.5 Subiculum

The subiculum is located between area CA1 and layer V of entorhinal cortex. Its function is not well understood, other than that it is involved in spatial navigation along with CA3 and CA1 [81], possibly performing path integration [91]. Because of this, its only function in our model is that of a communication channel between CA1 and the EC output, but see [18] for a related model of path integration in the NEF.

2.2.6 Medial Septum

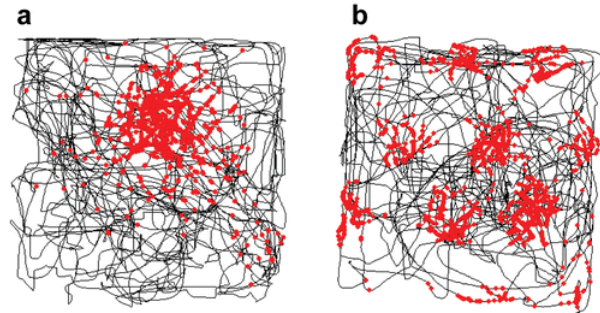
While not actually part of the hippocampal formation, the medial septum merits mention because of its cholinergic and GABAergic projections into hippocampus. These excitatory and inhibitory connections are thought to regulate the hippocampal theta rhythm and possibly provide control over learning and memory encoding in the hippocampal network [85].

2.3 Hippocampal Phenomena

2.3.1 Grid Cells and Place Cells

The most well-known type of neuron to exist in the hippocampus is the place cell, first discovered in rats by O'Keefe and Dostrovsky [78]. They were first discovered in CA1, but later found to exist in CA3, DG and subiculum as well [3]. Place cells are neurons that exhibit spatially correlated firing patterns. That is, they are neurons that are tuned to fire only when an animal is in a particular region of two-dimensional space called a *place field*. Their tuning curves typically look like two-dimensional Gaussian functions. The existence of place cells were the foundation of the belief that the hippocampus is used to form a spatial map of the environment.

Another cell type was discovered in 2005 that came to be known as grid cells [45] [13]. Grid cells are found in the entorhinal cortex, and like place cells exhibit spatially correlated




 Moser EI, et al. 2008.
Annu. Rev. Neurosci. 31:69–89.

Figure 2.3: Recordings from a place cell and a grid cell as a rat runs freely around a square environment. The lines represent the rat’s position in the environment, while the red dots represent a spike from the neuron being recorded from. a) The place cell fires most frequently in a single location and less frequently as the position moves away from this location. b) The grid cell fires most frequently in a hexagonal grid pattern within the environment.

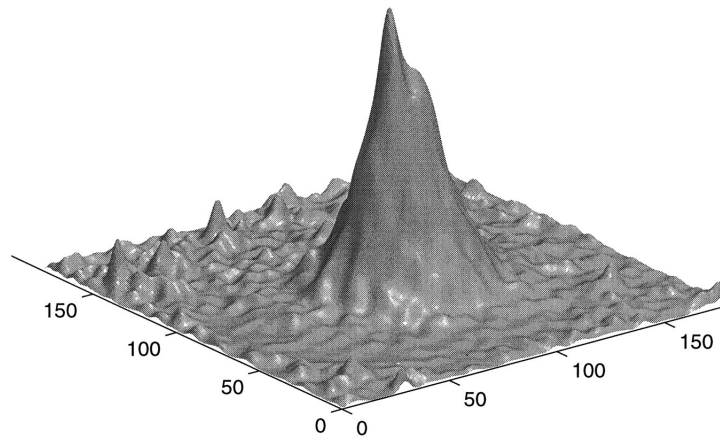


Figure 2.4: Activity packet of a group of place cells arranged by spatial tuning. The data was taken from a rat randomly foraging for food in a square environment. The cells tuned to the center of the environment are currently firing, indicating the rat’s current position. From [88].

firing patterns. However, grid cells fire in multiple locations within an environment in a hexagonal grid pattern. Different grid cells have different distances between coordinates in the grid. Several models have proposed explanations for how grid cells can be used to encode spatial maps and combine to create place cells [104] [82]. Figure 2.3 shows an example of the firing patterns of a place cell and a grid cell in a rat as it explores a test environment [73].

2.3.2 Remapping

One potential problem when dealing with episodic memory is that of memory interference. It is desirable that two separate but similar memories occupy distinct representational space. Navigation can pose a similar concern, as navigating through the same environment on two different occasions can be extremely different. For example, navigating a car through a large city can be very different depending on whether or not it is rush hour. Likely for this reason, hippocampal place cells are *remapped* depending on context [17], firing at different locations for different environments, or even the same environment under different conditions.

Two place cells do not remain consistent from one environment to the next. In 2005, Leutgeb et al performed an experiment recording from the same place cells while rats were placed into different environments [60]. In one set, they varied the colour of the environment while keeping the location the same (variable-cue constant-place), and in another set, they kept the colour the same while varying the location (constant-cue variable-place). They found that in the variable-cue case, the locations of the place cells remained the same but their firing rates differed, while in the variable-place case, the locations of the place cells changed completely. These two cases are known as *rate remapping* and *global remapping* (see figure 2.5).

2.3.3 Time cells and other hippocampal cell types

Since the discovery of place cells, several other types of cells have been discovered in the hippocampus, including head direction cells [96], border cells [92], odor sensitive cells [26], and time cells [64], which are neurons that fire periodically independent of location or behaviour. These time cells are evidence of temporal coding in the hippocampus, which might provide a mechanism for encoding the sequential nature of episodic memories. Figure 2.6 shows evidence of time cells in hippocampal recordings from rats.

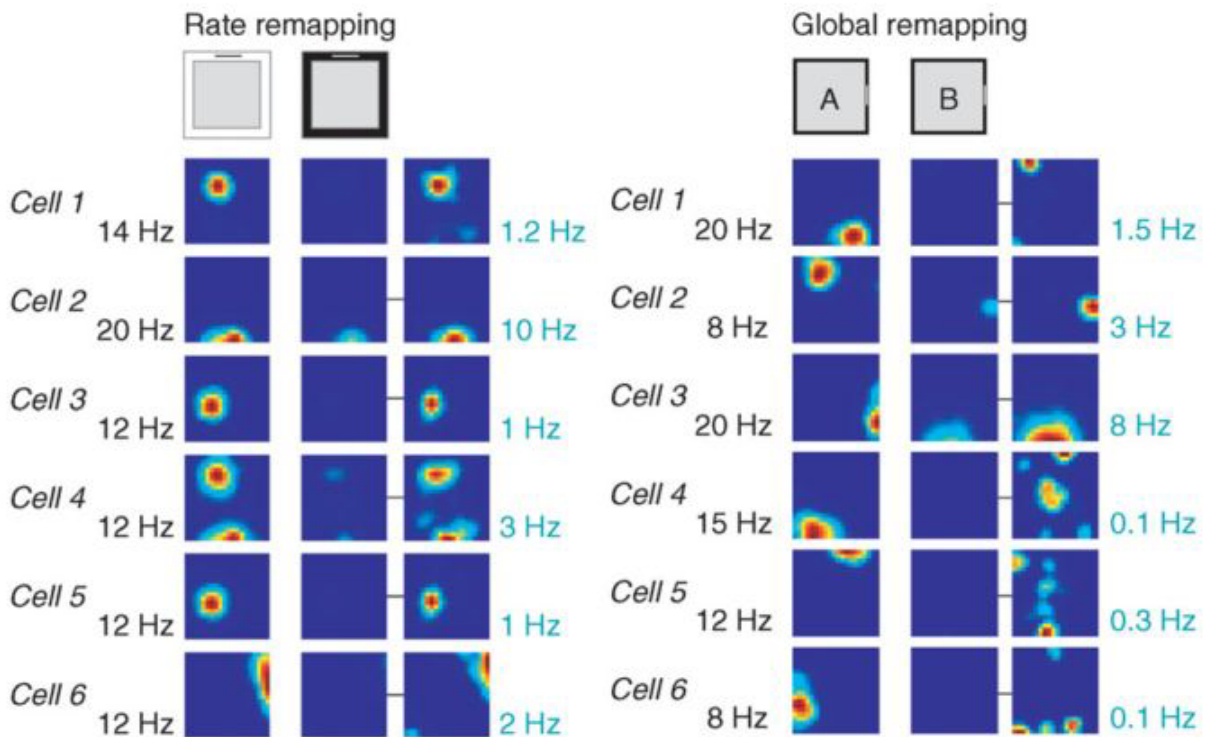


Figure 2.5: Rate remapping (left) and global remapping (right). The left columns represent firing rates in the first environment and the center columns represents rates in the second environment. The right columns are versions of the center columns that have been scaled for their new mean firing rates. The black numbers represent the mean firing rates of the left column, and the blue numbers the mean firing rates of the center/right columns. In the variable-cue case, the place cell locations remain constant but scaled (rate remapping). In the variable-location case the place cell locations have changed completely (global remapping). From [60].

2.3.4 Replay

Episodic replay, or just *replay* is the phenomenon of hippocampal cells firing in a certain pattern once during an experience, and then firing again in the same sequence at a later time. This replay effect has been found to occur in rats in anticipation of an experiment [28], during slow-wave sleep [105], during an experiment at a decision point [56], and even at seemingly unexplained times [44]. Although replay is often linked to immediate experience, it can also occur minutes or even hours after the original experience. It has been observed

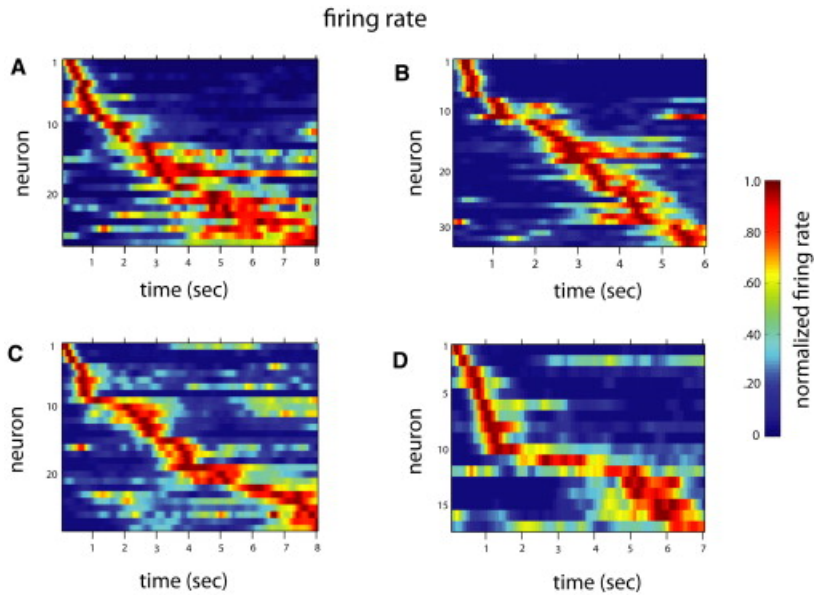


Figure 2.6: Time cell recordings exhibiting temporally correlated firing patterns taken from neurons in four separate rats [64].

in cells in both CA3 and CA1. The fact that replay occurs during sleep is thought to be important for consolidation of episodic memories into long-term memory storage.

The majority of replay-focused studies have involved recordings from rats in a maze. The rat performs the experiment while LFP recordings are taken from place cells, then using statistical inference the replayed firing patterns can be used to reconstruct the spatial trajectory that the rat took during the experiment, showing evidence that the rat is recalling a memory of that sequence. Figure 2.7 shows results from one such experiment. There have also been studies on humans showing similar effects, where recall of a specific memory triggers activation of sequences of hippocampal cells in the order that they were active during the original experience [38]. This evidence leads to the belief that hippocampal replay takes place during memory recall.

In addition to forward replay, *reverse replay* of firing patterns is also a commonly observed occurrence [35]. In rats, it often (but not always [44]) occurs immediately after a reward, and could arise from persistent neural activity, rather than any long-term memory consolidation. Although our model’s representation of memory is compatible with reverse replay, for simplicity’s sake we focus on modelling forward replay only, with reverse replay left as a possible extension to the model.

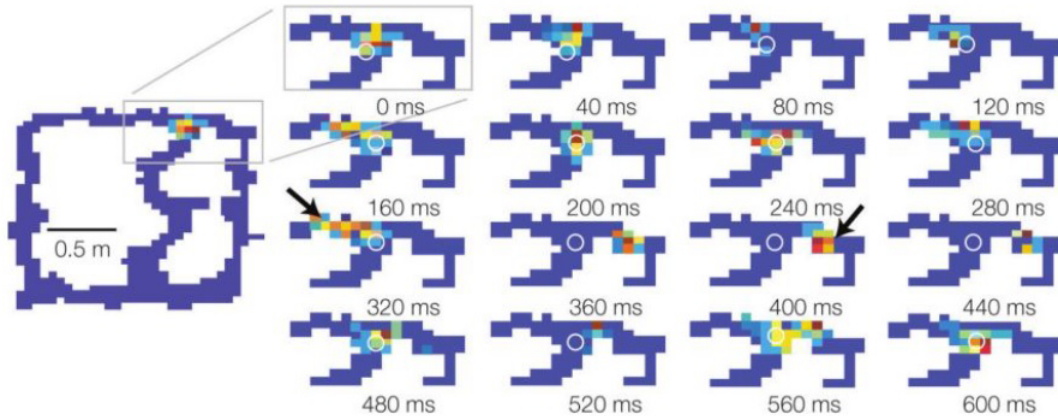


Figure 2.7: Place cell recordings showing lookahead (replay) at the decision point of a T-maze, particularly visible at 160 and 240 ms. The position reconstructions during this period are obtained from statistical inference on data from previous runs through the maze. The rat pauses for approximately 600 ms at the decision point and recalls its previous experiences down both paths. From [102].

2.3.5 Theta Rhythm

In 1954, Green and Arduini [43] found evidence of large-amplitude oscillations in the overall ambient voltage (local field potential) of the mammalian hippocampus using EEG recordings. In rats, theta rhythm is strongly associated with movement, and has a frequency of between 6 and 10 Hz [12], while in other mammals it is thought to occur when the animal is focused on sensory input, such as stalking prey or avoiding predators [87]. It is slower in larger mammals, oscillating at between 4 and 6 Hz. There is also a considerable amount of evidence linking theta rhythm to learning and time-dependant memory encoding in both humans and other animals [41]. In working memory tasks performed on both rats and humans, a theta phase reset has been found to occur immediately after the presentation of a stimulus, indicating the involvement of theta in encoding of memories related to sensory input [97]. In humans, theta effects have been found to occur strongly during recall of and navigation through complex environments [57], suggesting theta rhythm’s involvement in regulating the dynamics of memory encoding and retrieval [46].

In 1993, O’Keefe and Recce [80] made a discovery that linked theta rhythm in rats to place cells. They found that when a rat first entered a place field, the place cell

would fire late in the theta cycle, but as the rat remained in the same place field, the cell would fire progressively earlier. This phenomenon is known as *theta phase precession*, and is important in that it shows evidence of temporal coding in place cells. It also provides evidence for a continuous spatial code, since the firing patterns of place cells relative to theta oscillations differ depending on the rat's location within the place field at the time.

Chapter 3

Background Methods

This chapter gives an overview of the computational methods that were used in our model. The majority of the chapter focuses on detailing the NEF, a framework that allows us to perform computation with simulated spiking neurons. We detail the basics of representation, transformation, and dynamics in the NEF, and discuss methods for implementing memory and learning, two techniques that our model makes use of. First, we discuss vector symbolic architectures and semantic pointers, which we use to represent concepts and episodic memories in our model.

3.1 Representing Concepts as Vectors

Because the hippocampus deals with forming associations and structure from compressed sensory information, we need methods for representing and giving structure to high-level concepts. Vector Symbolic Architectures (VSAs) are a class of methods that allow us to do just that, providing the ability to represent single or multiple concepts as vectors and associatively bind concepts together with a set of vector operations. VSAs have been shown to fulfill Jackendoff’s linguistic challenges [37], proving to be sufficient for concept representation in a set of non-trivial tasks important to both linguistics and cognitive neuroscience as a whole. In addition, a method has recently been proposed for learning VSAs [36], and it has been shown that the binding operation can be learned by spiking neural networks [8], giving credence to the plausibility of their use in the brain.

The particular VSA implementation we will use is Holographic Reduced Representations (HRRs) [83]. HRRs use vector addition as an *addition* operator, and circular convo-

lution as a *binding* operator. Both of these operations have nice properties, which will be explained in further detail below.

3.1.1 Semantic Pointers

Semantic pointers, or SPs [32] are the implementation of HRR vectors employed by the NEF to represent concepts as vectors using spiking neurons. We will show how we can implement the convolution operation as a vector transformation and the addition operation as a memory module in sections 3.2.2 and 3.2.4. For our purposes, we will use the term semantic pointer to refer to any high-dimensional HRR vector representation of a concept.

SPs represent each atomic concept by a different D -dimensional vector in a vocabulary. In practice, we usually randomly generate these vectors, making sure that no two are within some similarity threshold. Assuming the constraint that all vectors are normalized (so that their Euclidean length is 1) and that there is a minimum distance between any two concepts in our vocabulary, our maximum vocabulary size increases exponentially with D , since the surface area of a D -dimensional sphere is also exponential in D . Thus, a large enough D will allow us to represent an enormous vocabulary with minimal overlap between concepts. For now, for illustrative purposes however, we will let $D = 3$. Then, let us define the following concepts:

$$\begin{aligned} \textit{blue} &= [0.0, 1.0, 0.0] \\ \textit{red} &= [0.0, 0.0, 1.0] \\ \textit{circle} &= [0.8, 0.6, 0.0] \\ \textit{square} &= [0.3, 0.3, 0.91] \end{aligned}$$

In this case, we could represent multiple concepts, such as “A circle and a square” as the Euclidean norm of the vector $\textit{circle} + \textit{square}$, or $[0.65, 0.54, 0.54]$, which ends up being a vector somewhat similar to both \textit{circle} and \textit{square} .

The true strength of SPs is that they allow us to represent structure in a vector encoding. HRRs and SPs, in particular, ensure that the dimensionality of the vector space does not increase with the structure. This ensures efficient use of resources, but also results in a ‘lossy’ encoding. Suppose we want to represent the concept “A blue circle and a red square.” Simply adding the four concept vectors together will result in ambiguity. If we are given the vector $(\textit{blue} + \textit{red} + \textit{circle} + \textit{square})$, how do we know what colour the circle was? So instead, we use a *binding*, or circular convolution [83] operator and represent the

above relations as the vector $(blue \otimes circle + red \otimes square)$, forming an *association* between *blue* and *circle* and between *red* and *square*.

Note that these operations are also sufficient for representing ordered sequences. Given a set of concepts C_0, C_1, \dots , we can define vectors for position indices P_0, P_1, \dots , and thus represent the sequence with

$$(P_0 \otimes C_0) + (P_1 \otimes C_1) + \dots$$

This will form the basis for our representation of an episodic memory.

3.2 The NEF

Now that we’ve seen how to represent sequences of concepts, we need a way of representing these vectors with populations of spiking neurons. For this purpose, we employ a set of modeling techniques collectively called the *Neural Engineering Framework*, or *NEF* [33]. The NEF proposes a set of methods for performing large-scale computations using populations of spiking neurons. It allows for encoding of vector values by populations of neurons, and computation of optimal decoders that allow the approximation of linear or nonlinear functions between neural ensembles. This allows us to perform arbitrary computations on vector or scalar quantities using simulated neurons. The following paragraphs go on to describe our computational methods and the NEF in more detail.

3.2.1 Vector Representation

Many empirical studies have found that neural populations in mammalian brains can encode real-world stimuli [51] [39] [23]. The NEF allows us to replicate this effect by encoding vector-valued stimuli with populations of simulated neurons, or neural *ensembles*.

Encoding

In the NEF, every neuron in an ensemble responds most strongly to a particular stimulus vector \mathbf{e}_i , known as that neuron’s *preferred direction* or *encoding vector*. Each neuron will receive more ionic current J when responding to its encoding vector, and receive less current the further away the stimulus vector \mathbf{x} is from \mathbf{e}_i . So given a vector stimulus as

input to a population, $\mathbf{x} = (x_1, x_2, \dots, x_n)$, we can relate the activity of a single neuron a_i in the ensemble to the stimulus by

$$a_i(x) = G_i \left[J(x) \right] \tag{3.1}$$

$$= G_i \left[\alpha_i (\mathbf{e}_i \cdot \mathbf{x}) + J_i^{bias} \right] \tag{3.2}$$

where G_i is the nonlinear (spiking or non-spiking) function specific to our neuron model, α_i is a scalar gain factor, and J_i^{bias} is a background bias current. The dot product $\mathbf{e}_i \cdot \mathbf{x}$ ensures the activity of neuron i is correlated with the similarity of that neuron’s encoding vector to its input \mathbf{x} .

Figure 3.1 shows a population of neurons encoding a sine wave, and gives a spike raster showing the firing of each individual neuron in the population over time.

Decoding

In addition to being able to encode stimulus values across neural ensembles, we also would like to be able to recover the original stimulus given an ensemble’s firing pattern. This allows us to build a representation (both encoding and decoding) for arbitrary stimuli using neural ensembles. The simplest way to do this is to make the assumption that the stimulus is a linear combination of the neural activities, which turns out to be quite accurate if we have enough¹ neurons in our representation. That is, we assume our stimulus vector $\hat{\mathbf{x}}$ can be represented by

$$\hat{\mathbf{x}} = \sum_{i=1}^N a_i \mathbf{d}_i \tag{3.3}$$

with N being the number of neurons in the ensemble and \mathbf{d}_i being a vector of decoding weights for neuron i . If we know \mathbf{x} , it is possible to find the optimal set of linear decoders \mathbf{d} that minimize the squared error between \mathbf{x} and $\hat{\mathbf{x}}$. This is a common problem in linear algebra, and can be solved as follows:

$$\mathbf{d} = \Gamma^{-1} \mathbf{v} \tag{3.4}$$

$$\Gamma_{ij} = \sum_x a_i a_j$$

$$\mathbf{v}_j = \sum_x a_j f(\mathbf{x}).$$

¹representation error decreases at a rate of approximately $1/N$, where N is the number of neurons [33].

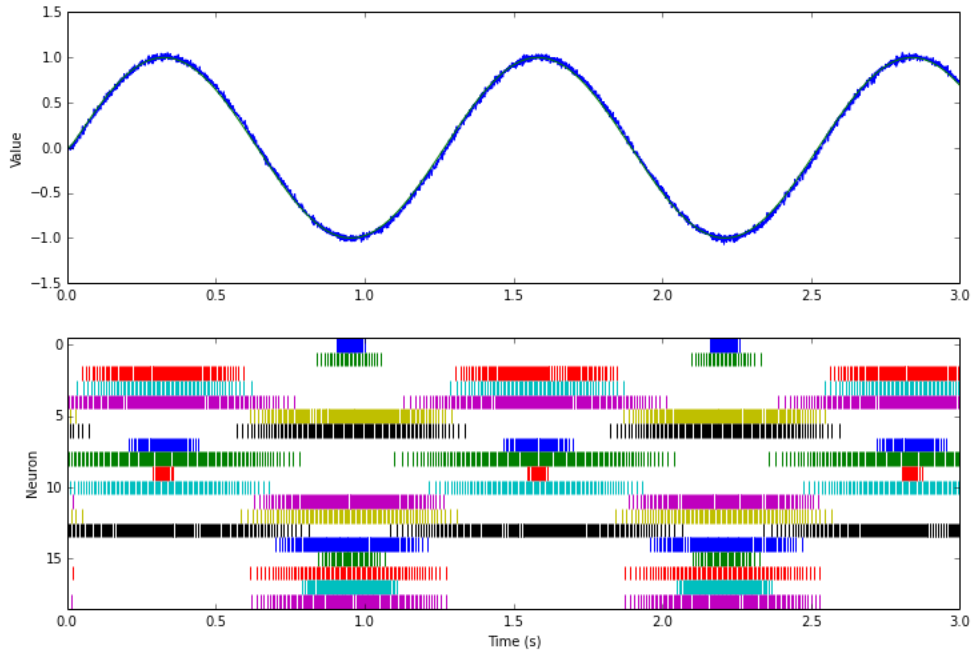


Figure 3.1: Plots showing a neural ensemble representing a sine wave ($f(x) = \sin(5x)$). Graphs show the decoded value from the ensemble (top) and a spike raster showing when each neuron in the population spikes (bottom). In the 1-dimensional case, each neuron’s encoding vector is either 1 or -1, so approximately half of the neurons in the population spike most frequently when the input value is close to 1, and the other half spike most frequently when the input is close to -1.

where the inverse is a Moore-Penrose pseudo-inverse. Solving for the optimal linear decoders, \mathbf{d} , allows us to recover an estimate of the original stimulus vector given a neural ensemble’s activity. To do this, we simply set $f(\mathbf{x}) = \mathbf{x}$ in the bottom equation, although it is also possible to decode functions of \mathbf{x} . As we will see, this decoding allows us to directly compute the neural connection weights in order to perform a computation between two ensembles.

Neuron Model

Our model uses the leaky integrate-and-fire (LIF) neuron as our single cell model. Figure 3.3 shows the tuning curve of an example LIF neuron. The behaviour of a LIF neuron (with parameters τ_{RC} and R) with current membrane potential V and input current J is

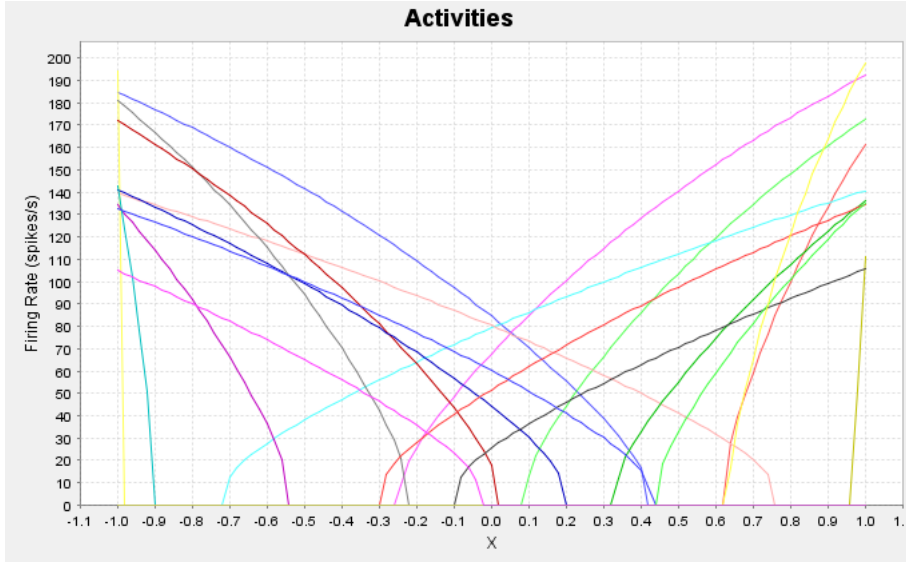


Figure 3.2: Tuning curves of 20 one-dimensional LIF neurons. Each neuron’s preferred direction vector is either 1 or -1. The curves’ x-intercepts are determined by the J^{bias} parameter, and the steepness of the curves are determined by the α (gain) parameter.

governed by the following differential equation.

$$\frac{dV}{dt} = \frac{-1}{\tau_{RC}}(V - JR) \quad (3.5)$$

This behavior holds up until the neuron hits a threshold voltage V_{th} (and corresponding threshold current J_{th}), at which point it generates a spike. In the LIF model, we take a spike to be a discrete event, after which the membrane potential drops and the neuron enters a refractory period for τ_{ref} seconds. This distills the concept of a spiking neuron down to the basic idea of a cell that emits spikes based on input voltage, but abstracts away details of biological neurons such as ion channel currents. The NEF is compatible with any neuron model, and it has been shown that comparable results can be achieved with a wide variety of neuron models [33].

Under a constant input current J , the *activity* of a LIF neuron $a_i(J)$ can be thought of as its steady state firing rate, and is given by

$$a_i(J) = \left[\tau_{ref} - \tau_{RC} \ln \left(1 - \frac{J_{th}}{J} \right) \right]^{-1} \quad (3.6)$$

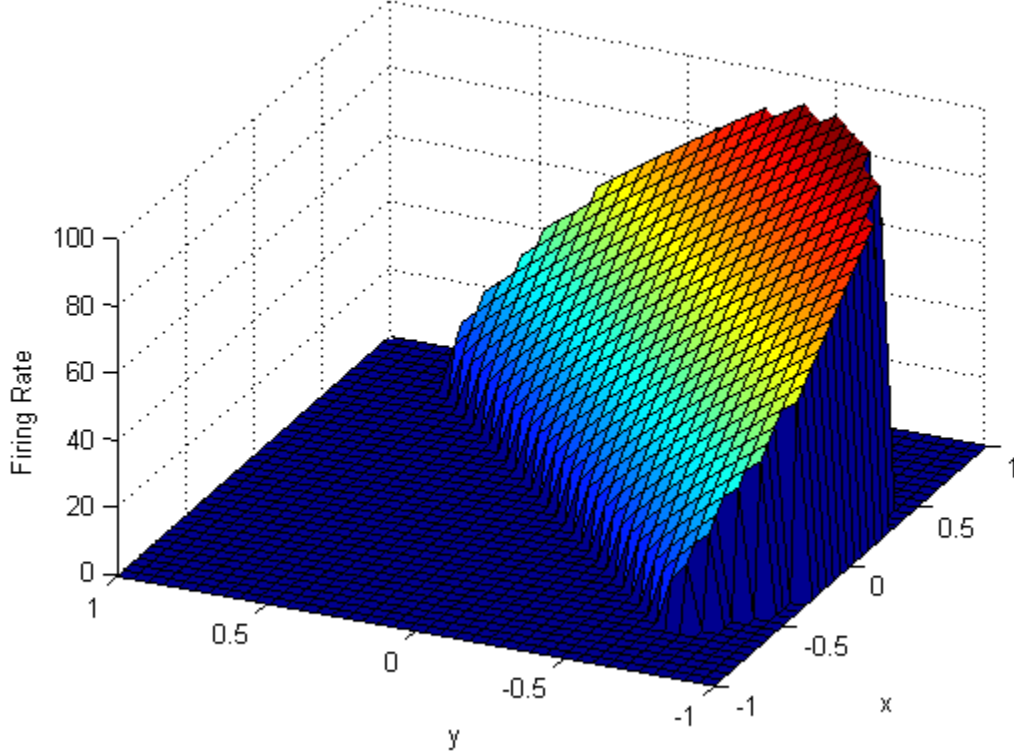


Figure 3.3: Tuning curve of a two-dimensional LIF neuron. Firing rate increases as the x component of the input increases, and the y component decreases. This neuron’s preferred direction vector would be $\|(1, -1)\|$. The firing rates at the corners of the figure are undefined because the neuron is set to represent vectors with radius less than or equal to one, so the value $(1, 1)$ would fall outside of the representational space. Adapted from [108].

where J_{th} is the threshold current of the neuron, τ_{ref} is the refractory period for the neuron, and τ_{RC} is the membrane time constant for the neuron. Using these in conjunction with equation 3.1 fully specifies our system of encoding and decoding.

Temporal Representation

Note that thus far, our model of neural activity $a_i(x)$ has referred to a neuron’s firing rate. In order to run simulations of spiking networks in the NEF in real time, we need a model of synaptic dynamics. When decoding from a spiking population, we apply a model of post-synaptic current, or a *post-synaptic filter* $h(t)$ to each spike by convolving it² with $a_i(\mathbf{x}(t))$, where \mathbf{x} is now time-dependant. Our decoding equation then becomes

$$\hat{\mathbf{x}}(t) = \sum_{i=1}^N h(t) * a_i(\mathbf{x}(t)) \mathbf{d}_i \quad (3.7)$$

For $h(t)$, we use a post-synaptic current curve, which is meant to model the sharp rise and exponential decay in neural voltage caused by the release of a neurotransmitter when a neuron spikes.

$$h(t) = \frac{1}{\tau_{\text{PSC}}} e^{-t/\tau_{\text{PSC}}} \quad (3.8)$$

Here τ_{PSC} is a post-synaptic time constant controlling the speed at which current is delivered. This allows our model to exhibit biologically realistic dynamics, and hence empirically constrained timing data.

3.2.2 Vector Transformation

Now that we have defined a way of encoding and decoding stimulus values, we will describe how to perform computations between neural ensembles using our encoding and decoding vectors. Suppose we want to have an ensemble B encode some function of the value another ensemble A is encoding, \mathbf{x} . i.e. $\mathbf{y} = f(\mathbf{x})$, where \mathbf{y} is the value encoded by ensemble B. We simply compute the decoders for \mathbf{x} as in equation 3.4, setting $f(\mathbf{x})$ to our desired function instead of just using \mathbf{x} when computing \mathbf{v}_j . Then in order to encode our desired function, we multiply our new functional decoding weights \mathbf{d} by our encoding weights for population \mathbf{y} , yielding a new set of weights between the populations that generate the desired transformation,

$$\omega_{ij} = \alpha_j(\mathbf{d}_i \cdot \mathbf{e}_j) \quad (3.9)$$

²using linear convolution, not circular convolution

where α_j is a gain term associated with neuron j . Once we have our connection weights, we can use them to encode the desired transformation in population B, determining the neural activity of population B, $b_j(\mathbf{x}(t))$, given input $\mathbf{x}(t)$ to population A.

$$b_j(\mathbf{x}(t)) = G_j \left[\sum_{i=1}^N h(t) * a_i(\mathbf{x}(t)) \omega_{ij} + J_j^{bias} \right] \quad (3.10)$$

Note that this technique works well for nonlinear functions as well as linear ones, as we are in effect projecting into a higher dimensional space than our representation, effectively turning a nonlinear function into one that is linear in the weight space. The only difference for nonlinear functions is that the population computing the function in its decoders must encode each argument to the function. So if, for example, we wanted to perform multiplication of two scalar values ($f(x, y) = xy$), we would need to first project x and y into an intermediary two-dimensional population, and have that population's decoders compute the multiplication.

3.2.3 Population Dynamics

The NEF also defines a way of computing functions defined over time, or *dynamic* functions. Incorporating time-dependance is important in understanding and modelling neural responses, since in the real world, neural activity is of course taking place in real-time. In general, we describe a linear dynamic function by

$$\dot{x}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) \quad (3.11)$$

where $\mathbf{x}(t)$ is the value currently being represented, and $\mathbf{u}(t)$ is an input value from another ensemble.

One useful example of such a function is an *integrator*, defined simply by $A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $B = \begin{pmatrix} \tau & 0 \\ 0 & \tau \end{pmatrix}$, where τ is the integrator's post-synaptic time constant. This system adds whatever input value it receives to its currently stored value, integrating the total input over time. τ controls the speed of this integration. When given no input, it simply holds its current value, acting as a simple form of memory. To have an ensemble exhibit this behavior, we simply define a *recurrent* connection from a population to itself in the same manner as if we were defining a normal connection between two populations.

3.2.4 Memory in the NEF

As seen above, one way of storing values over time in the NEF is through the use of neural integrators, a type of recurrently connected neural network. Integrators attempt to hold their value over time when not given any additional input, but are subject to neural dynamics. A recurrently connected population of LIF neurons is an attractor network, and attractor networks have stable fixed points [31]. For a one-dimensional integrator, a fixed point will occur whenever the decoded estimate is exactly equal to the target value being represented, i.e $\hat{x} = x$. Of these fixed points, exactly half will be stable fixed points, depending on which direction the estimate crosses over the target value. The number of these stable points will depend on how well our ensemble estimates the target value, which depends on the number of neurons used in the representation.

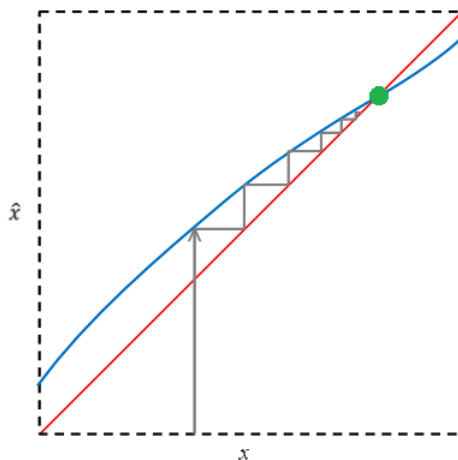


Figure 3.4: Graph of the original input value x vs. an integrator's decoded estimate of the input \hat{x} . Errors in the representation will cause drift over time, as the recurrent connection is fed back into the integrator as input, until the representation reaches a stable point (green dot).

When we move into higher dimensional space however, our representational space becomes a curved surface in hyperspace rather than a line. There are very few points (or none) at which our decoding is perfect, and thus the integrator's ability to hold its value suffers. For this reason, whenever we are working with an integrator as a memory for a high-dimensional semantic pointer, we will use multiple one-dimensional integrators instead of one multi-dimensional one.

One other useful feature we want to add to an integrator is the ability to load and unload values from it. For example, say we have a one-dimensional integrator currently holding the value 0.5, and we want to set it to 0.2. We could give the integrator the value -1 for exactly 0.3 seconds, but this reliance on precise timing will inevitably cause drift. Instead, we can use a *gated difference unit*, which is a convenient method for loading and storing values. This is an integrator paired with another ensemble. There is a connection from the integrator to the difference population, and another connection from the difference ensemble back to the integrator with weight -1. This difference ensemble is *gated* with inhibitory connections from a third gate population, which prevent neurons in the difference population from firing when the gate population is active. This way we can keep the gating population active most of the time, which will cause the unit to remember its current value, and deactivate the gate when we want to load a new value into the integrator. Figure 3.5 shows the populations and connections in a gated difference unit, and figure 3.6 shows the results of using one to store values.

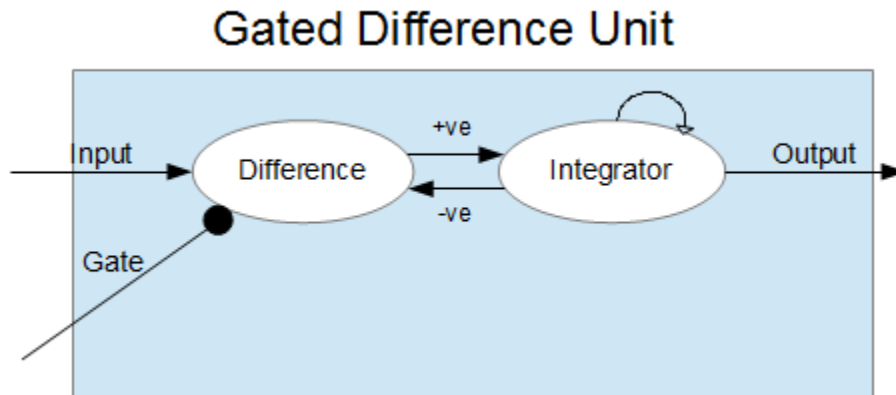


Figure 3.5: Architecture of a gated difference unit. When the gate signal is high, the difference population is inhibited and the unit retains its value because of the recurrent connections in the integrator. When the signal is low, the Input value is loaded into the Integrator and the negative projection from the Integrator back into the Difference population cancels out the integrator's recurrent connection. Circles represent inhibitory connections.

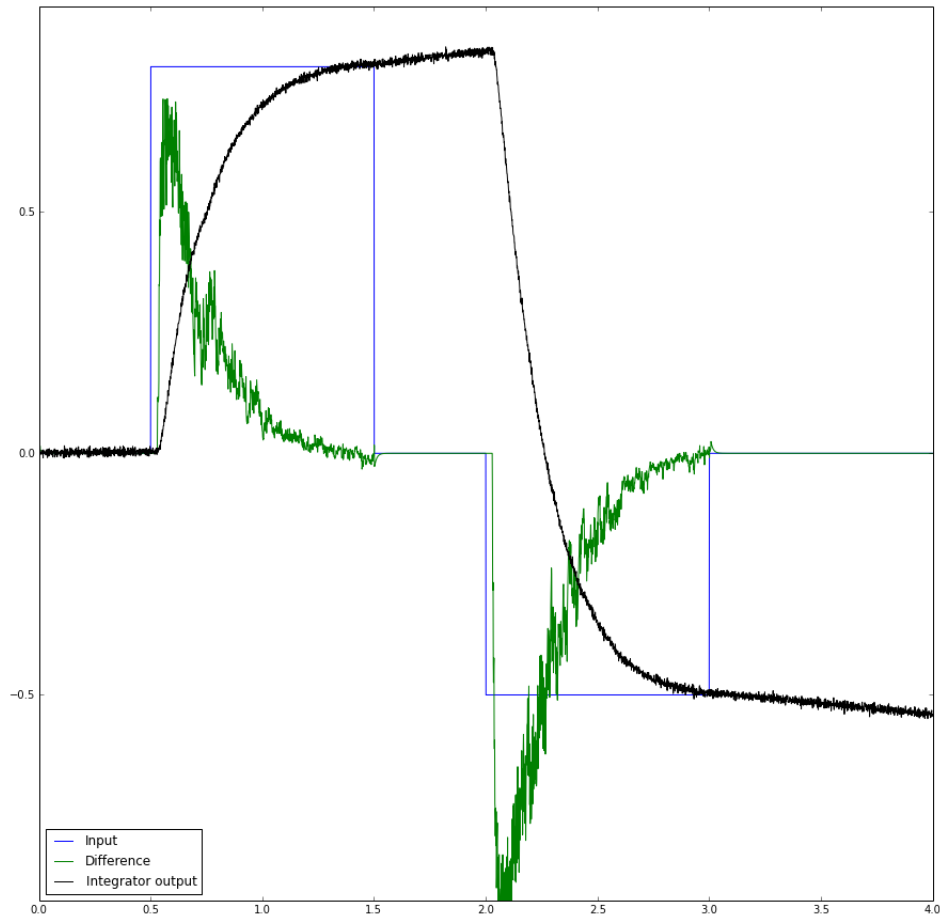


Figure 3.6: Plot showing results of using a gated difference unit. An input value of 0.8 is given at $t = [0.5, 1.5]$ and an input of -0.5 is given at $t = [2.0, 3.0]$. This value is loaded into the integrator, which holds its value when no input is given. Notice the small amount of drift in intervals $t = [1.5, 2.0]$ and $t = [3.0, 4.0]$.

3.2.5 Learning, Memory, and Synaptic Plasticity

While the brain uses neural activity as a form of working memory [42], in order to create more permanent memories information must be stored in neural connection weights [68]. The brain accomplishes this through *synaptic plasticity*, which is the modification of synaptic strengths between neurons according to their activity. The first mathematical model of synaptic plasticity was based off of a theory by Donald Hebb in 1949 [50]. He proposed

that neurons that are connected by a synapse and are active together will increase their connection strength. More formally:

$$\Delta\omega_{ij} = \nu a_i a_j$$

where ω_{ij} is the connection weight between presynaptic neuron i and postsynaptic neuron j , a_i and a_j are some measure of activity of neurons i and j (e.g. spike rate), and ν is a learning rate parameter. A more stable version of this rule was later formulated [58], taking the derivative of a_j instead:

$$\Delta\omega_{ij} = \nu a_i \dot{a}_j$$

Despite a lack of experimental evidence supporting the theory at the time, it eventually became evident that Hebb's postulate was a reasonable model for synaptic modification in the brain, and it became known as *Hebbian learning*.

The first experimental evidence of synaptic plasticity was *long-term potentiation*, discovered by Bliss and Lømo in 1973 [11]. They discovered that by stimulating neurons in the rabbit hippocampus and recording from downstream neurons, the correspondence between firing rates grew over time. A few years later an opposite effect, *long-term depression*, was found. Lynch et al found that downstream neurons could also become less correlated with firing rates of upstream neurons over time [63]. Finally, a temporal correlation was discovered in 1983 by Levy and Stewart in the hippocampus [61], that was later described by Markram as *spike-time dependant plasticity*, or STDP [67] and found to occur throughout most brain areas. Between a presynaptic and postsynaptic pair of neurons, A and B, STDP causes the connections between A and B to strengthen when A fires immediately before B, and it causes the connections to weaken when B fires immediately before A. In all cases, there is evidence that synaptic plasticity is the brain's form of long-term storage for memory and learning [71].

In 1989, Erkki Oja proposed a different formalization of Hebbian learning, which also does a good job of explaining synaptic modifications in parts of the brain [76]. The formula for modification of connection weights between neurons i and j , called Oja's rule, is

$$\Delta\omega_{ij} = \nu a_j (a_i - a_j \omega_{ij}) \tag{3.12}$$

This rule is useful in that it is stable, meaning connection strengths will not continue to increase or decrease indefinitely, and in that it allows a neuron to learn to better compute the principal component of its input over time.

Synaptic plasticity in the NEF

In the NEF, we model synaptic plasticity as changes in connection weights (the ω matrix) over time based on the activities of the neurons involved. MacNeil and Eliasmith suggested a simple error-modulated Hebbian learning rule that allows populations of neurons to change their connection weights to minimize an error signal [65], thus learning a transformation. If we take \mathbf{E} to be the difference between the represented value \hat{x} and the desired value x , then the change in decoder values of neuron i at each timestep is

$$\Delta \mathbf{d}_i = \nu \mathbf{E} a_i \tag{3.13}$$

To express this rule in terms of connection weights we simply multiply by the encoders of the postsynaptic neuron and use equation 3.9.

$$\begin{aligned} \Delta \mathbf{d}_i \cdot \mathbf{e}_j \alpha_j &= \nu \mathbf{E} a_i \cdot \mathbf{e}_j \alpha_j \\ \Delta \omega_{ij} &= \nu \mathbf{E} a_i \cdot \mathbf{e}_j \alpha_j \end{aligned} \tag{3.14}$$

We also have a way of representing a vectorized version of Oja’s rule in the NEF, thanks to recent work done by Voelker [103]. This is useful particularly in building associative memories, since it allows populations to modify their encoding vectors on the fly to better represent the input.

These learning rules allow us to incorporate synaptic plasticity in our simulation, allowing for a model of long-term memory storage and learning.

3.3 Nengo

Our model was developed using version 2.0 of the Nengo neural modelling software [9]. Nengo is a software package that implements the algorithms in the NEF and gives an interface between the model specification and the low level implementation details. For example, using the Nengo GUI, a modeller can drag and drop populations into a model, connect them, specify a function for them to compute, and connect an input signal to them. Nengo would then take care of generating each individual neuron, calculating the optimal decoders as in equation 3.4, and calculating connection weights between populations. Nengo also provides a python scripting interface that allows for programmatic model

generation. Finally, Nengo allows for the simulation of our neural models, generating and plotting spikes and decoded values as the network runs. More information about Nengo can be found at <http://nengo.ca>.

Chapter 4

A Neural Model of the Hippocampus

4.1 Motivation

While there exist many neural memory models of hippocampus already [2, 49, 75, 69], to the best of our knowledge none of them have both the functional power to imitate real-world human memory behaviour, while still being biologically plausible and replicating experimentally observed neural phenomena. One notable hippocampal model that simulates replay through biologically plausible learning rules is that of Levy et al [62]. Their model also includes mechanisms for both online and offline sequence compression, modelling the consolidation of memories from the hippocampus into long-term cortical storage. However, their model does not provide a mechanism to explain context switching, and does not generalize their sequence representation to multi-modal (both spatial and non-spatial) information. More recently, Hasselmo’s comprehensive hippocampal memory model [47] provides an explanation for how multi-modal information can be stored in memory, representing an episodic memory as a spatial trajectory and also unifying experimental data on place cells and theta phase precession. While his model focuses on providing biologically realistic representations of neural phenomena in the hippocampus, it does not attempt to use a high-level unified representation scheme for episodic memories as ours does. Finally, Wu [108] used the NEF and Hessian-free learning to train a hippocampal model that could perform sequence learning and both forward and reverse replay. While powerful, the Hessian-free learning algorithm used in his model is not biologically plausible.

Our goal is to create a functional hippocampal model that can encode and replay sequences of information, while obeying experimentally observed constraints to maintain biological plausibility. We also want to use the semantic pointer architecture in order to be

compatible with Spaun, the recently published large-scale brain model by Eliasmith et al [34]. Spaun included a working memory module in order to store sequential information for tasks such as serial-order recall, but did not include any method for longer term memory storage of episodic memories. We want our simulation to give a more complete picture of episodic memory storage and recall, while still being compatible and useful to integrate with the existing SPA architecture used in Spaun (i.e. visual system, reinforcement learning, etc).

4.2 Model Description

Our Nengo model is engineered primarily with functionality in mind. We want our model to be able to take a context and sequences of data as input, remember them, and associate them correctly such that it can later recreate the same sequence of data for the given context. We have built our model with biological constraints in mind, but choose to adhere to functionality over biological accuracy when a tradeoff is necessary.

4.2.1 Model Overview

Our model takes 4 signals as input. First of all, we have two sensory experience signals coming from EC, context and experience. These are semantic pointers, or in our case, 64 dimensional vectors. These signals represent high-level (compressed) sensory information coming from various parts of the brain. Note that we take this to include both spatial and non-spatial information. Our model does not differentiate between spatial and conceptual information, but later we will show spatially correlated neural firing patterns (place cells) result from interpreting these input signals in a spatial manner.

The other two input signals are control signals from the medial septum. Both are 1-dimensional on/off toggles, one of which switches the network between encoding and recall mode, while the other is a ‘reset’ signal, which resets the network back to its initial state and clears the currently stored memory by resetting the neural activity in our memory integrator.

The model gives only a single output signal, a result signal. During encoding, this signal should simply mimic the sensory input signal, but during retrieval it should match whatever was previously stored in memory for the given context and current ordinal index.

The model has 5 primary components:

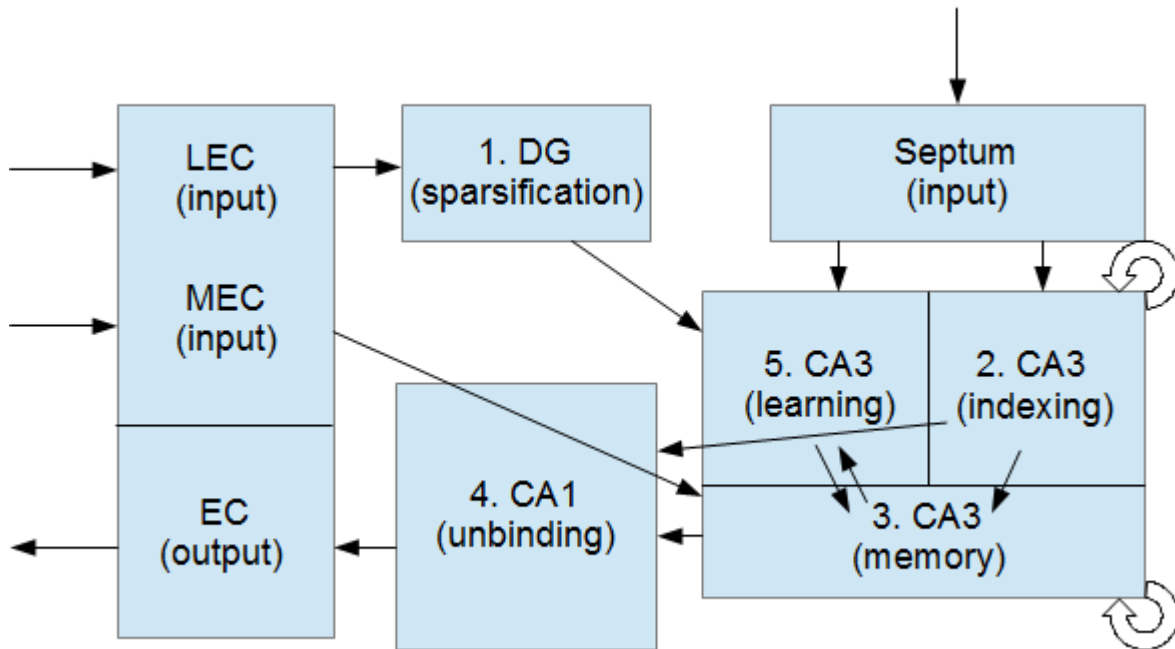


Figure 4.1: A high level overview of our hippocampal model, showing connectivity and function of each part of the model.

1. DG performs pattern separation on the context signal.
2. Then we have sequence generation in CA3. A neural oscillator is used as a kind of ‘clock’, and inhibitory connections allow a pair of difference units to count through a sequence of unitary *position* vectors.
3. Next we have an associator, which binds the current sensory experience with the index coming out of CA3. This is theorized to take place in a different region of CA3, getting the sensory input from EC and the position/index input from CA3. These associations are then stored in an integrator, or working memory, which stores a sum of the bound sensory experiences.
4. In CA1 we have a deconvolution network, which deconvolves whatever is currently stored in the working memory with the index from CA3, giving a result. During encoding, this result should just be the same as the sensory input, but during recall, this result population should cycle through the same series of sensory vectors as it

was originally given, giving a *forward replay* effect.

5. Finally, we have a second associative memory, this one making use of rapidly learned connection weights. This memory associates the current context (from DG) with the complete memory sequence (in CA1/EC). This association from a context to a compressed version of a complete memory episode is what allows for later recall of previously stored memories and eventual memory consolidation into cortex.

We will go on to describe each component of our model in more detail.

4.2.2 Sparsification (DG)

The dentate gyrus part of our model takes the context signal from EC and represents it with a separate neural population. This connection is simply a communication channel, representing the same value as the input signal. The difference is that the neurons in the DG population are tuned to represent their value more sparsely. We do this by setting the distribution of intercepts for the neurons in this population uniformly in the interval $[0.3, 1.0]$, as opposed to $[-1.0, 1.0]$. This means neurons in this population will only fire for input values whose dot product is very close to their preferred direction vector.

This sparse representation helps to differentiate similar contexts. The DG population will have far fewer neurons firing at any time, but each neuron will be more selective, only firing for inputs that it is tuned to. Although our model does not simulate neurogenesis, we theorize that newly formed neurons in DG can be tuned to uniquely identify newly experienced contexts, whether they be new environments or new experiences (episodes) in known environments. Figure 4.2 shows the difference in representational sparsity between EC and DG.

4.2.3 Index Generation (CA3)

The sequence generation part of our model relies on recurrent connections, and thus is theorized to reside in CA3. The purpose of this network is to generate a temporal indexing system, similar to hypothesized function of the time cells described by MacDonald et al in 2011 [64].

First we build a 2D neural oscillator to act as a timer for the network. Oscillators in the NEF are simply dynamical systems defined by a recurrent connection and a transform matrix, and can be implemented similar to the way that integrators were described in

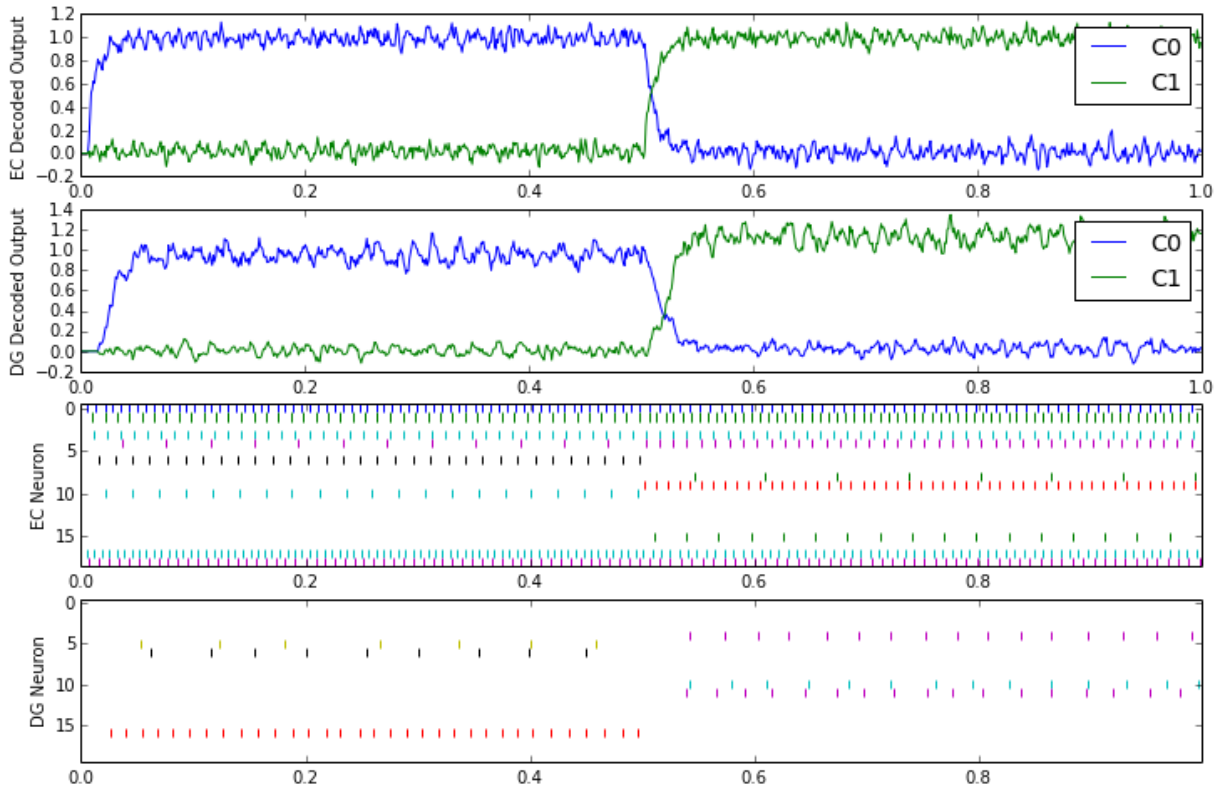


Figure 4.2: Representation of context and neural spike trains in EC vs. DG. Two context vectors, C0 and C1, are given as input for 0.5 seconds each. Both EC and DG populations represent the context vectors equally well (top two graphs), but the DG population does so more sparsely, with neurons tuned to represent a single context. In EC there are many neurons that fire for both contexts, but neurons in DG fire for only one or the other (bottom two graphs).

3.2.3. The oscillator is by default tuned to oscillate with a frequency of 2Hz, making it a recurrently connected population with transform matrix $A = \begin{pmatrix} 1 & -2\tau \\ 2\tau & 1 \end{pmatrix}$, where τ is a time constant (we use $\tau = 0.05$). We also add a *control* input to the oscillator, allowing for control over its oscillation frequency. This input simply adds the transform $\begin{pmatrix} 0 & -x \\ x & 1 \end{pmatrix}$ to the recurrent A matrix. One dimension of the oscillator is then connected to a separate *clock* ensemble, which is simply a step function, 0 if the oscillator value is negative and 1 if the oscillator value is positive. This clock will serve as a timing mechanism for two inhibitory gating connections that will act as a flip-flop, as we will see.

The rest of the CA3 circuit is responsible for the temporal indexing functionality of the model, which is accomplished by sequentially counting through list indices. We define semantic pointers representing the concepts *ONE* and *ADD1*, allowing us to compute $TWO = ONE \otimes ADD1$, $THREE = TWO \otimes ADD1$, etc. We have two gated difference integrators storing these indices, a *current* and a *next* population. The output of the *current* unit is connected to the input of the *next* unit with a transform matrix computing the transformation defined by convolving with the *ADD1* vector. The *next* unit is then connected back to the *current* unit via a communication channel (identity transformation). We have a population, *one*, that simply always stores the value *ONE*. To begin the simulation, and every time the *reset* input coming from medial septum is on, the *one* population projects into both *current* and *next*, erasing their current value and loading the value *ONE* into both integrators. When the reset signal is off, an inhibitory gating signal is sent to these projections, turning them off and allowing the difference units to function as integrators. In addition to the *reset* signal, our *clock* population serves as an additional control, toggling the gated inhibition on and off between the main two difference units. This allows the network to initialize the sequence to *ONE* when necessary, and then when left alone, begin to count through the sequence of indices at a frequency equal to that of the oscillator. The circuit with the gated difference units connected to each other recurrently is based on a similar sequential memory circuit described and implemented in the NEF by Choo [14].

Although the circuit works as described, the representation accuracy of the counting degrades due to integrator drift and representation error. To solve this problem, we use a cleanup memory [93]. We attach a cleanup memory to each of the connections between the gated difference units, such that the connections become $current \rightarrow cleanup1 \rightarrow next \rightarrow cleanup2 \rightarrow current$. The cleanup memory serves to compare the dot product of the vector fed in as input to a predefined vocabulary, in our case, the set *ONE*, *TWO*, *THREE*, ..., and outputs the value of the closest match in a winner-take-all. If there are no matches close enough past some threshold (we set it to 0.5), the zero vector is output. These cleanup memory units work by using one population per vector in the vocabulary to calculate the dot product between the input vector and that population's vocabulary vector. Once this is done, we approximate the nonlinear function

$$\begin{aligned} f(x) &= x : x \geq t \\ f(x) &= 0 : x < t \end{aligned}$$

where t is the threshold over which we want an output value. Finally, to allow for a winner-take-all output, we include inhibitory connections from each population to each other population, allowing for only one population to be active at a time. The addition of

the cleanup memory networks allows the circuit to function as intended, counting through indices without any decay or drift.

As can be seen in figure 4.3, this part of the model makes use of recurrence, both in the dynamical systems used in the oscillator and integrator (difference unit) components and in the overall cyclical nature of the network. The output of this subnetwork is a semantic pointer indicating the current index in a sequence. The inputs are control signals from the medial septum: a *reset* signal telling the network to reset back to *ONE*, and a frequency signal controlling the speed of the oscillator. This frequency controlled oscillator, previously implemented in the NEF in [53] and [82], provides a mechanism for speeding up and slowing down the sequence indexing to simulate the temporal compression effect seen in rat replay [4]. Not included, but implementable would be the ability to reverse the sequencing, allowing the rest of the network to exhibit reverse replay [35]. See chapter 6 for further discussion on reverse replay. Figure 4.4 shows the activity of different populations in this subnetwork during sequence generation.

4.2.4 Associative Memory (CA3)

Now that we have a mechanism for counting through sequences, we're able to bind these sequence indices with the sensory input coming from EC. We do this with a circular convolution network, taking *index* from CA3 and *sensory* from EC as inputs. To implement circular convolution with neurons, we note that the convolution operation $A \circledast B$ is equivalent to the element-wise product of A and B in the frequency domain, so we can take the Discrete Fourier Transform (DFT) of A and B, element-wise multiply them, and perform an inverse DFT on the result. A DFT can be performed as a linear transformation, defined by the W matrix.

$$W_{ab} = \cos \frac{-2\pi ab}{d} + i \sin \frac{-2\pi ab}{d} \quad (4.1)$$

So to compute this value in neurons, we set the decoders of a population to compute the above transformation¹. We take the resulting frequency domain vectors and multiply their results using a second population, and finally use a third population to compute the inverse DFT of the result.

The output of the circular convolution network leads to a recurrently connected *memory* population acting as an integrator that stores the current memory sequence. Given a

¹For efficiency reasons, the computation is divided into computing both the real and imaginary terms of this equation and adding them together in a separate population

CA3 (Indexing)

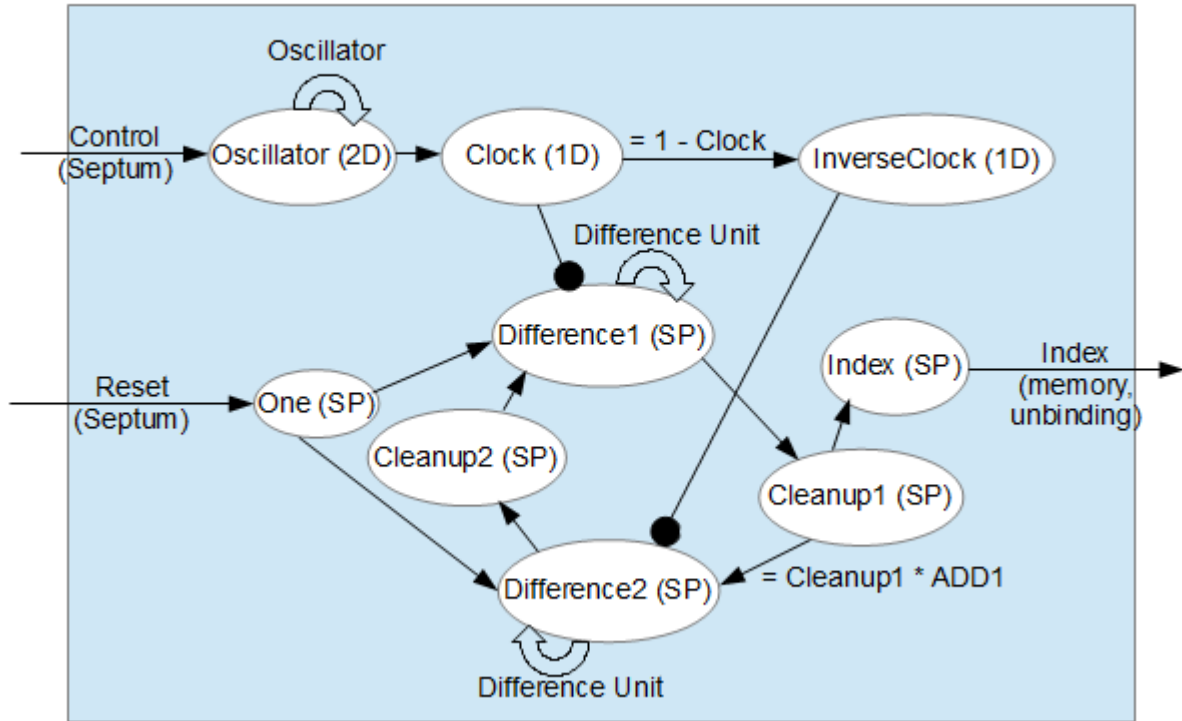


Figure 4.3: CA3 indexing architecture. The network counts through a sequence of unitary vectors in the recurrent loop $\text{Difference1} \rightarrow \text{Cleanup1} \rightarrow \text{Difference2} \rightarrow \text{Cleanup2} \rightarrow \text{Difference1}$. A clock signal governs the temporal frequency of the network. Circles represent inhibitory connections.

sequence of sensory inputs S_0, S_1, S_2, \dots , (and a sequence of position vectors P_0, P_1, P_2, \dots from CA3) the integrator stores the convolved output along with whatever was previously stored. After a sequence has been experienced, the value stored in this population will be the HRR vector $(P_0 \otimes S_0) + (P_1 \otimes S_1) + (P_2 \otimes S_2) + \dots$. The population can also be inhibited by a *reset* signal coming from septum, allowing for this memory to be cleared as necessary. Figure 4.5 shows the neural populations and connections in this subnetwork.

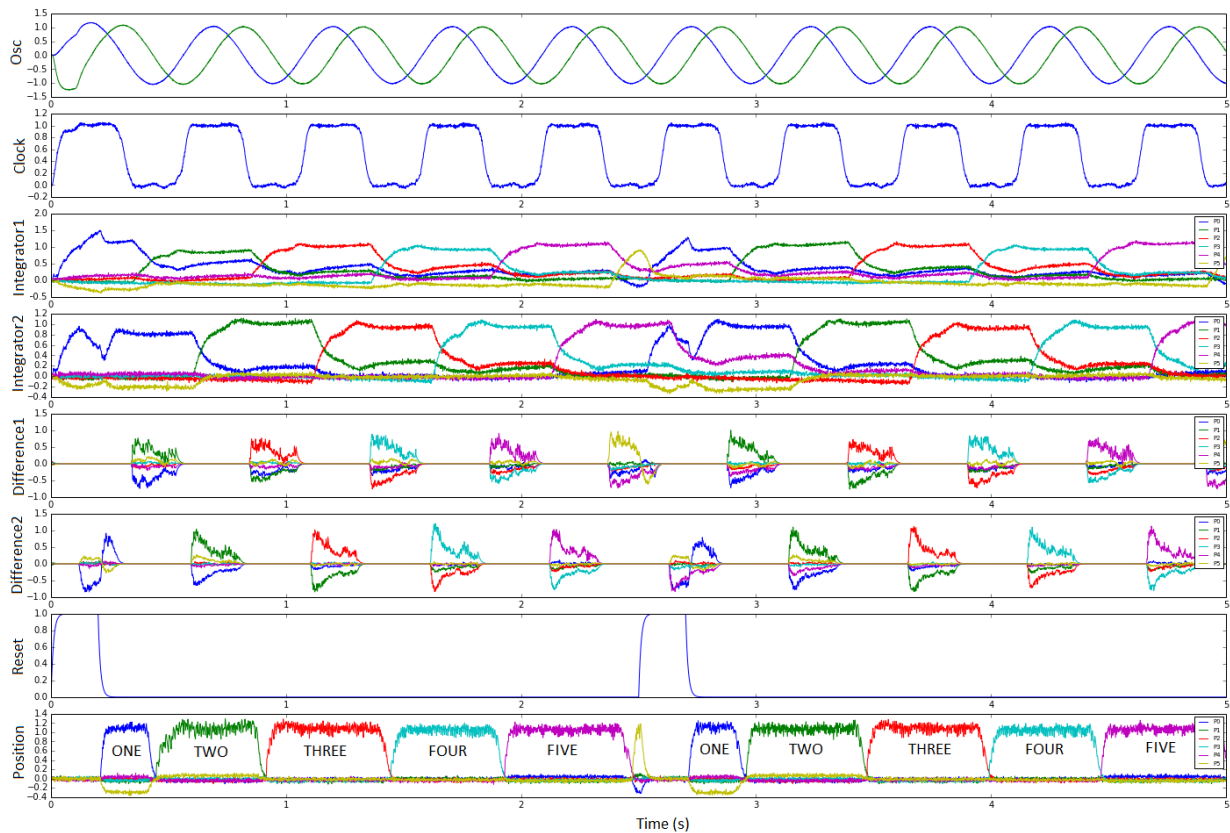


Figure 4.4: Counting through a sequence in the CA3 indexing network. Graphs show the decoded value being represented by each population over time. For the populations representing semantic pointers, values shown are the comparisons (dot products) between the decoded value and the vectors in the vocabulary. The network is being driven by an oscillator with frequency 2 Hz. The Reset signal is high at time $t = 0$ and $t = 2.5$, resetting the index stored in the integrators to P0. Position is the output from this part of the model.

4.2.5 Unbinding (CA1)

The deconvolution part of our model takes the current sequence in memory and convolves it (via circular convolution) with the inverse (involution [83]) of the current *position* vector. This convolution is done in the exact same manner as the above binding convolution,

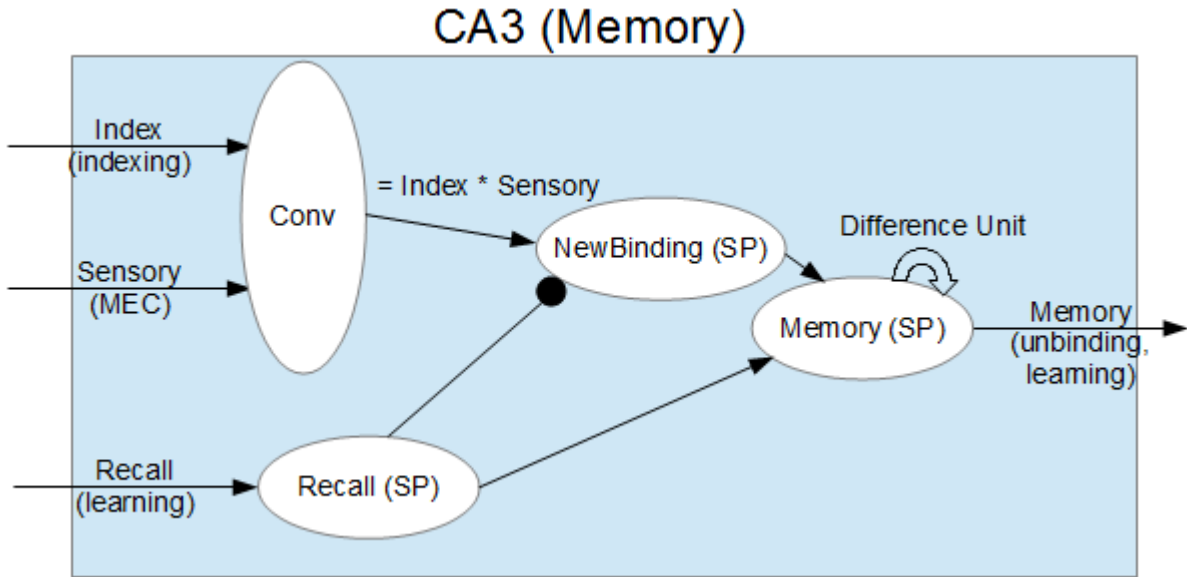


Figure 4.5: CA3 memory architecture. The index signal from the indexing part of the model is convolved with the sensory input and added into the memory integrator. If we instead are in recall mode, this new binding is inhibited, and memory is loaded directly from the signal coming from the learning part of the model. Circles represent inhibitory connections.

except that we invert one input before the convolution is performed. This allows us to reconstruct the originally encoded sensory input for that *position* in a *result* population. For example, if our current *position* input is $P1$ and the *memory* populations contain the sequence $M = (P0 \otimes S0) + (P1 \otimes S1) + (P2 \otimes S2) + \dots$, then $M \otimes P1^{-1} = S1 + noise$.

We also employ a cleanup memory here, implemented in the same manner as the one used in CA3. The cleanup memory outputs a winner-take-all between the dot product of the possible concept vectors and the resulting value is stored in the result population, allowing us to retrieve a cleaned up version of the result vector. Figure 4.6 shows the neural populations and connections in this subnetwork.

4.2.6 Learning for Consolidation (CA3)

Up to this point, we've described how our network can count through a sequence, bind indices in this sequence with sensory inputs, store them in a working memory, and recall

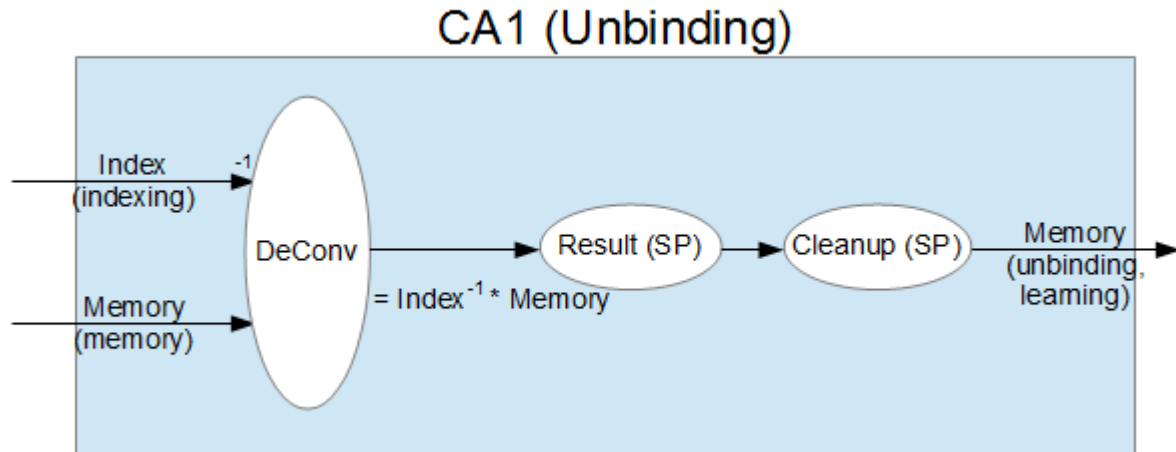


Figure 4.6: CA1 unbinding architecture. The current index is deconvolved with whatever is loaded into memory, giving back a sensory vector. The vector is then fed through a cleanup memory, yielding the original vector that was bound with that index, in that memory.

and replay them. However we have not yet detailed how our model would consolidate these memories into longer-term storage in connection weights rather than neural activity. The synaptic modification part of our model does this.

We implement a hetero-associative memory as described in [103]. This subnetwork works by employing the vectorized Oja rule to allow a population to learn to sparsely represent a set of *keys*. The encoders in this population start out uniformly distributed around the space, but as it receives input, any neurons that fire strongly (their encoders are in a vector space close to the input vector), they will shift to better represent that input vector. This allows the population to uniquely represent a set of discrete keys. We can think of these keys as indices in a map that get associated with corresponding *values*.

The reason that we need learning here, rather than using a circuit similar to the cleanup memory, is that we want to be able to learn associations involving context and sensory vectors, which are not part of some predefined vocabulary, they are inputs given at ‘runtime.’ At the same time as the encoder learning, a standard error-modulated learning rule is used to shift the population’s decoders to compute the transform *key* \rightarrow *value*.

We use this associative memory subnetwork to associate the context signal coming from DG with the value stored in the *memory* population, allowing us to perform context

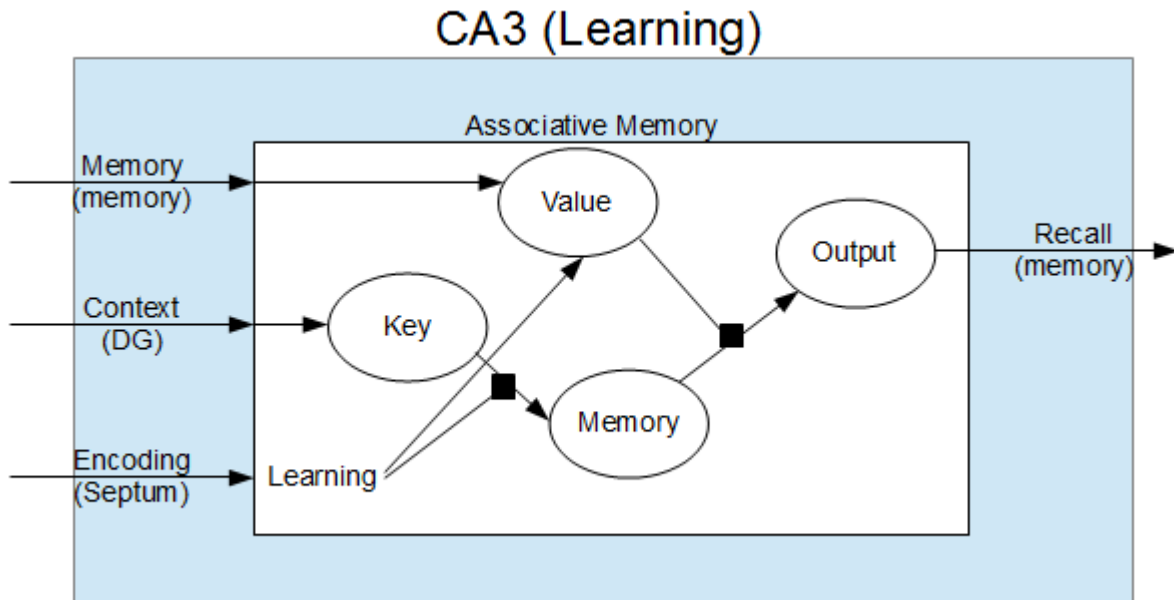


Figure 4.7: CA3 learning architecture. The context signal from DG is associated with the current memory. Encoders in the Memory population use Oja’s rule to learn to better represent the input, while decoders in the memory population learn the transformation between Key and Value via error modulated learning. The Learning input is taken from Septum, and can just be the inverse of the Recall signal mentioned in figure 4.5 (i.e. when Recall is 1, Learning is 0 and vice verse). The Recall output gets sent to the CA3 Memory module to be loaded into memory, if necessary. Squares represent modulatory connections.

switching, associating multiple different contexts with different memory sequences associated with those contexts. We also have a control signal coming from medial septum to toggle this part of the model between encoding and recall mode. During encoding mode, learning is turned on and the connections between the convolution and memory populations are enabled, allowing the memory to be built up and learned as we encounter new experiences. Conversely, during recall mode, learning in the heteroassociative memory is turned off (the neurons in the error modulating population are inhibited), the connections between the convolution and memory populations are disabled, and a new connection is enabled between the output of the heteroassociator and the memory population. This allows the memory corresponding to the current context to be loaded and replayed. All toggling of connections is accomplished through the use of inhibitory gating signals. Figure 4.7 shows the neural populations and connections in this subnetwork.

4.2.7 Functional Overview

Figure 4.8 gives a simplified view of the populations and connections that comprise the entire hippocampal model.

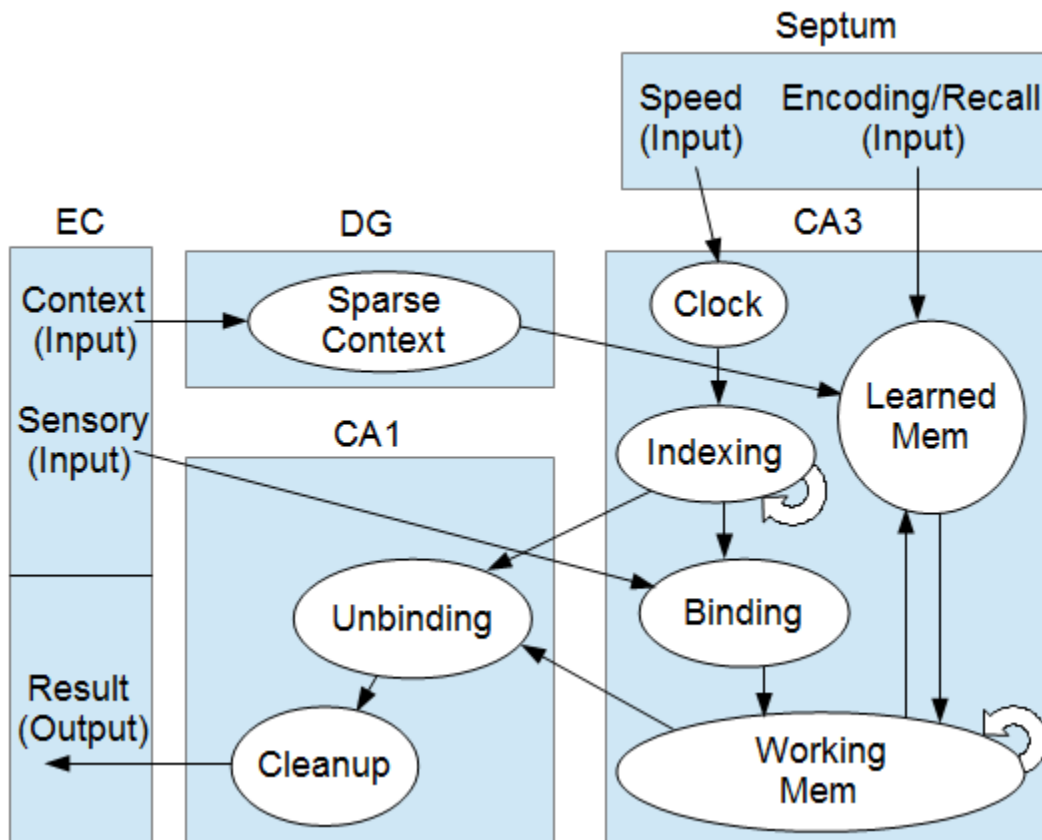


Figure 4.8: Simplified functional overview of hippocampal model. During encoding, the sensory input from EC is bound with the current index and stored in the working memory integrator. At the same time, the contents of the integrator is associated with the sparsified context input from EC (via DG) in the learned memory subnetwork. During recall, the working memory is loaded from the learned memory network, and this signal is unbound with the current index and output through CA1. The septum inputs provide control for the network, regulating the clock speed and controlling whether the network is in encoding or recall mode.

4.2.8 Neural Parameters

Table 4.1 outlines the various parameters used in different parts of the model.

Ensemble	Num neurons	Dimensions	Max firing rates (Hz)	Intercepts	PSTC (s)
1) ECContext	3200	64	[100, 200]	[-1.0, 1.0]	0.01
1) DGContext	12800	64	[100, 200]	[0.3, 1.0]	0.01
2) Oscillator	2000	2	[200, 400]	[-1.0, 1.0]	0.05
2) Clock	200	1	[200, 400]	[-1.0, 1.0]	0.01
2) One	3200	64	[200, 400]	[-1.0, 1.0]	0.01
2) Difference1	6400	64	[200, 400]	[-1.0, 1.0]	0.1
2) Difference2	6400	64	[200, 400]	[-1.0, 1.0]	0.1
2) Cleanup1	500	64	[200, 400]	[-1.0, 1.0]	0.01
2) Cleanup2	500	64	[200, 400]	[-1.0, 1.0]	0.01
2) Index	3200	64	[200, 400]	[-1.0, 1.0]	0.01
3) Conv	12800	64	[200, 400]	[-1.0, 1.0]	0.01
3) NewBinding	3200	64	[200, 400]	[-1.0, 1.0]	0.01
3) Recall	3200	64	[200, 400]	[-1.0, 1.0]	0.01
3) Memory	19200	64	[200, 400]	[-1.0, 1.0]	0.1
4) DeConv	12800	64	[200, 400]	[-1.0, 1.0]	0.01
4) Result	3200	64	[200, 400]	[-1.0, 1.0]	0.01
4) CleanResult	3200	64	[200, 400]	[-1.0, 1.0]	0.01
5) Key	3200	64	[200, 400]	[-1.0, 1.0]	0.01
5) Value	3200	64	[200, 400]	[-1.0, 1.0]	0.01
5) Memory	1000	64	[200, 400]	[-1.0, 1.0]	0.01
Total	103400				

Table 4.1: Neural parameters used in the model. The number before each ensemble refers to the numbered subnetworks in figure 4.1. Intercepts refers to the distribution of x-intercepts in the neurons’ LIF tuning curves (see figure 3.3). For recurrently connected populations, the recurrent PSTC is given.

Chapter 5

Simulations and Results

This chapter details the simulations we ran on our model and the resulting neural spike and decoder data gathered from Nengo. Our aim is to show that the model can encode and recall sequences of both spatial and non-spatial data, and do so in a biologically plausible manner that generates similar neural phenomena to those found in experimental research.

5.1 Replay of Concept Vectors

5.1.1 Replay from Neural Activity

First we demonstrate the basic functionality of the memory part of the model by encoding a sequence, resetting the indexing, and replaying that sequence in the absence of sensory input. Figure 5.1 shows the results of running a replay simulation. The model is run for 5.5 seconds. From $t = 0$ to $t = 2.5$ seconds, a sequence of five 46-dimensional sensory input vectors, A through E, are shown for 0.5 seconds each. At the same time, the indexing part of the model is counting through a sequence, also with period 0.5s. During this time, the sensory input is being convolved with the index and added into our memory integrator. For $t = 2.5$ to $t = 3.0$, the *reset* input is high, causing the indexing to reset. Then from $t = 3.0$ to $t = 5.5$, the index resumes counting through the sequence of position vectors, the contents of the memory are deconvolved with the current position vector, and the result is fed through a cleanup memory to reconstruct the original input sequence.

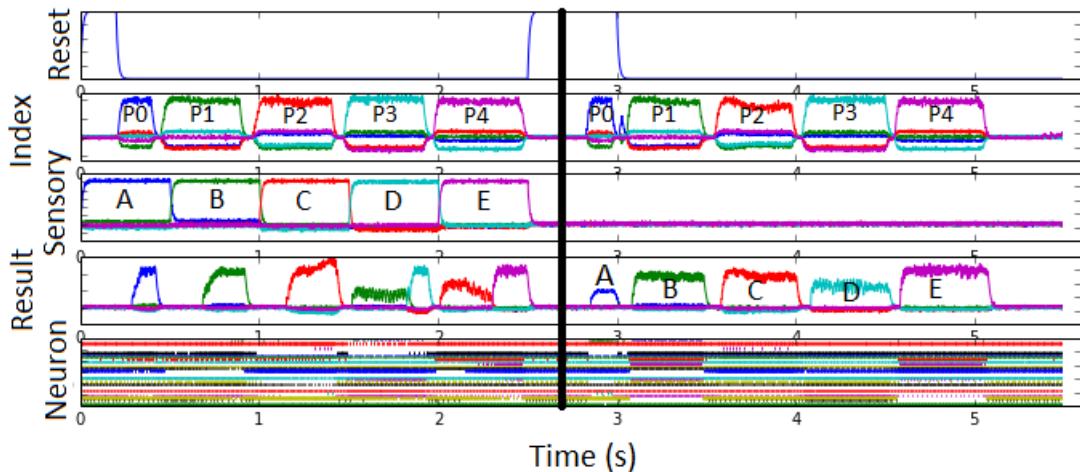


Figure 5.1: The original input sequence is replayed by the *Result* and *CleanResult* populations at time $t = 3.0$ to 5.5 . Graphs show the decoded value being represented by each population over time. For the populations representing semantic pointers, values shown are the comparisons (dot products) between the decoded value and the vectors in the vocabulary. *Position* is the output of the CA3 indexing part of the model. Also shown is a spike raster for 30 neurons in the *Result* population. Note the similarity between spikes when the same value is being represented, e.g. $t = 0.5$ to 1.0 and $t = 3.5$ to 4.0 .

5.1.2 Replay from Connection Weights

Next we demonstrate the functionality of the learning part of the module, which learns to associate contexts with whole memories. We run a simulation where two sequences are given as input consecutively with two different context vectors also being given as input. The learning part of the model modifies its connection weights to learn the association between each context and the entire sequence in memory at the time. The two sequences are of length 5 and 7, and are presented for 2.5s and 3.5s respectively, then the *recall* input is turned on and the model replays the two sequences. The results are shown in figure 5.2, in this case making one error in the length 7 sequence. This shows the model’s ability to switch back and forth between contexts, as has been demonstrated in rats [60].

We also run a simulation of *remote replay*, which is the replay of a sequence long after the original experience. Figure 5.3 shows spike trains from our model compared to spike trains taken from rats during sleep in an experiment performed by Ji et al [55].

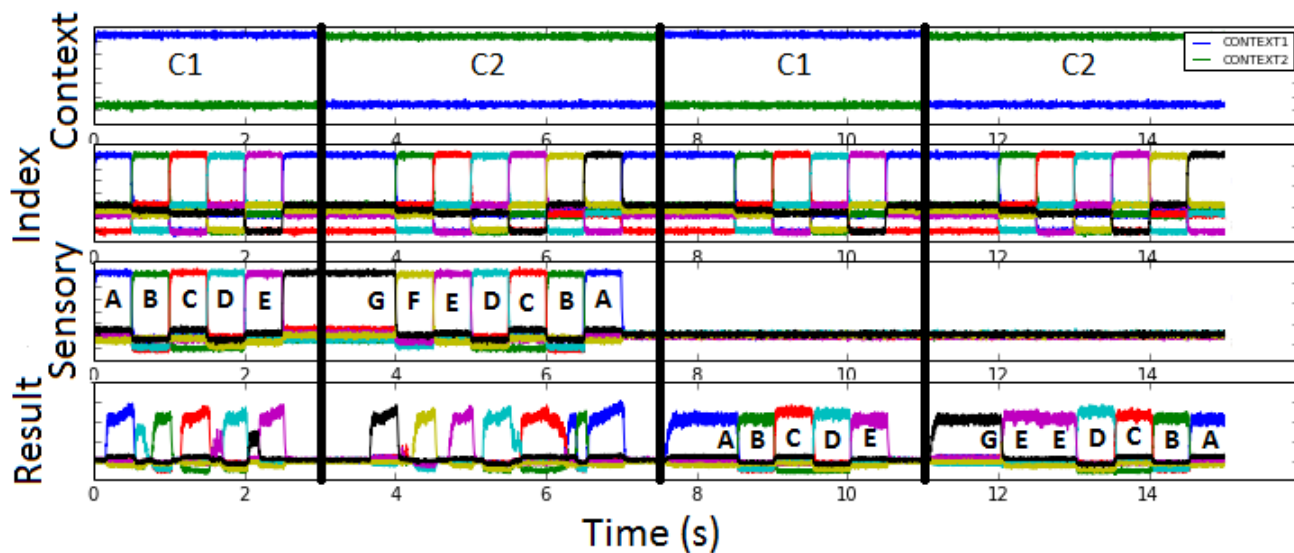


Figure 5.2: Two input sequences are experienced with two different contexts. Graphs show the decoded value being represented by each population over time. For the populations representing semantic pointers, values shown are the comparisons (dot products) between the decoded value and the vectors in the vocabulary. For the first 2.5 seconds the model is given CONTEXT1 and [A, B, C, D, E] as input, then for 3.5 seconds it is given CONTEXT2 and [G, F, E, D, C, B, A] as input. The model is then given CONTEXT1 and *Recall* as input for 2.5 seconds, and the sequence [A, B, C, D, E] is replayed. Then it is given CONTEXT2 and *Recall*, and it replays the sequence [G, E, E, D, C, B, A] (one error). A one second gap is left between each phase to reset the position input and clear the memory integrator.

Accuracy vs. Sequence Length

As the length of the sequences increase, the performance of the model degrades, as is to be expected. This error is caused primarily by the noise introduced by adding so many bound vectors together into one integrator. When the deconvolution occurs to get our *Result* vector, if there is too much noise, the model will be unable to clean up the result vector accurately. To test how representation accuracy varies with sequence length, we perform 10 simulations each of sequences of length 3 to 8, clearing the memory after the sequence input and recalling from the learning part of the model each trial. Figure 5.4 shows the resulting tradeoff between sequence length and recall accuracy in our model.

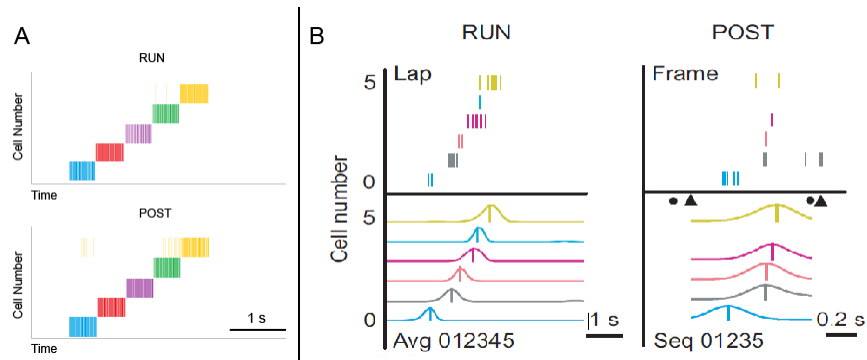


Figure 5.3: Spike train data taken from *Result* population in model (A) vs. data taken from rats in [55] (B). In the rats, the replay takes place during sleep, at a remote instance in time. In the model, we simulate this by clearing the working memory and running the model for 20 seconds on different input. The replay event is triggered by setting the *recall* signal to true and giving the same *context* input as the original experience.

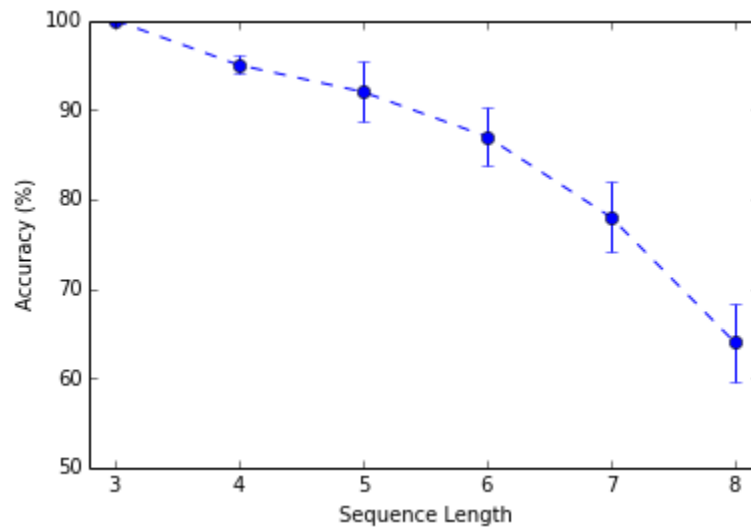


Figure 5.4: Graph of sequence length vs. recall accuracy. Accuracy is calculated per element in a sequence, so if 4/5 elements were recalled correctly, that would correspond to 80%.

5.2 Replay of a Spatial Trajectory

We have now shown that our model can encode and recall sequences of concept vectors, but what about spatial trajectories? The hippocampus has been shown to be important in spatial navigation, so our model needs a way of representing positional information. It turns out that because of our system of vector representation, we don't need to change anything in our model for these input vectors to represent spatial information vs. conceptual information. As we will show, we can simply interpret some of the dimensions of our vectors as spatial data.

The critical difference between representing spatial data and representing high-level conceptual data is the number of dimensions (64 vs 2). So, we can have a subset of neurons in a population only represent two dimensions by setting their encoding vectors to 0 in dimensions 2-64, effectively splitting the population up into *place cells* and *concept cells*. We can do this with every population in the model that represents either sensory or contextual information (i.e. the inputs from EC), allowing us to effectively build a 2D spatial circuit in parallel to our 64D conceptual one, linking spatial and conceptual vectors together in the model. In fact, we can represent both concepts and spatial information in a single 64D signal by simply setting the first two dimensions of the vector to represent spatial information scaled down to a value in the interval $[-0.1, 0.1]$. The information loss from the two dimensions is small enough that our recalled concept vectors will still be robust when fed back through a cleanup memory.

The following experiment similar to work done by Johnson et al [56] simulates a rat navigating through a simple T-maze (figure 5.5). The semantic pointers to be used as input are chosen such that they have a spatial component in their first two dimensions. Three trials are performed, in which the rat chooses the left path, the right path, and then pauses at the decision point evaluating both paths. In the first trial (the left path), for $t = 0$ to 3.0, the context input is RUN, followed by CHOICE1, for 1.5 seconds each. The sequence of sensory vectors experienced is A, B, C, then C, D, E for 0.5 seconds each. In the second trial, for $t = 4.0$ to 7.0, the context input is RUN and CHOICE2, and the sensory input is A, B, C, then C, F, G. In the third trial (the right path), for $t = 8.0$ to 11.0, the context is RUN, CHOICE1, CHOICE2, and the *recall* input is turned on at $t = 9.5$, to allow the model to replay the sequences learned for the contexts CHOICE1 and CHOICE2. At this point, the indexing circuit is sped up, allowing for temporal compression when recalling the learned sequences. The memory integrator is reset in between trials, so results are not dependant on time between trials. Experimental results are shown in figure 5.6.

By plotting the decoded output from our spatially sensitive neurons, we can obtain a reconstruction of the rat's position while the recall input is turned on. Figure 5.7 shows this

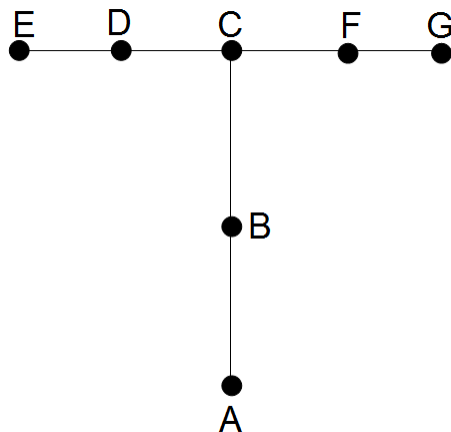


Figure 5.5: Structure of the simulated T maze. The rat starts at point A $(0, 1)$, and moves upward to the decision point C $(0, -1)$. It then makes a decision to go left or right, ending the run at either point E $(-1, -1)$ or point G $(1, -1)$.

reconstructed position over time during the period when the rat is paused at the decision point, in comparison with experimental results from [56].

5.3 Time Cells

There are neurons in the CA3 indexing part of our model that exhibit temporal selectivity similar to the time cells described by MacDonald et al in [64]. They found that these time cells would fire in sequence both during memory tasks and in delay periods between tasks in rats. The indexing model's sequence generation populations are temporally cycling through their sequence of unitary vectors, so individual neurons in the model will fire at a point in the sequence when a vector similar to their encoder is being represented. We record the spikes from neurons in the *Position* population in the CA3 indexing part of the model in a 5 second simulation and compare them to the time cells found in rats. The results are shown in figure 5.8.

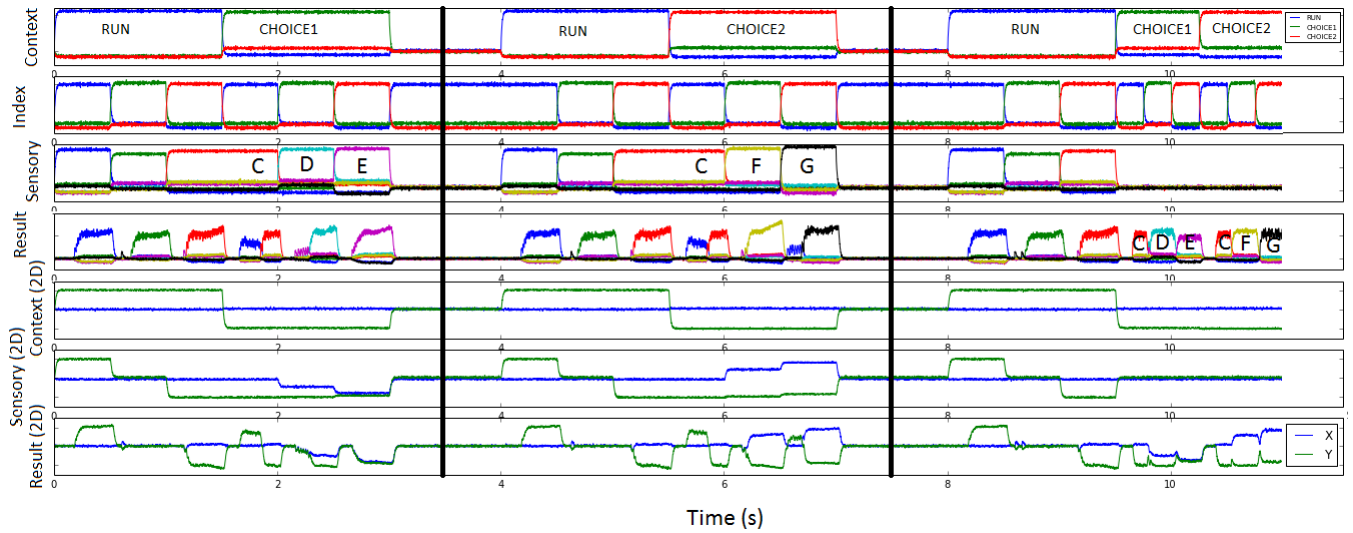


Figure 5.6: Results of running the simulated T maze experiment. During the recall period from $t = [9.5, 11.0]$, the Result population replays the sequences previously experienced at $t = [1.5, 3]$ and $t = [5.5, 7]$. Graphs show the decoded value being represented by each population over time. For the populations representing semantic pointers, values shown are the comparisons (dot products) between the decoded value and the vectors in the vocabulary. The bottom three graphs show the 2D component of the input signals: Context and Sensory, and the output signal: Result.

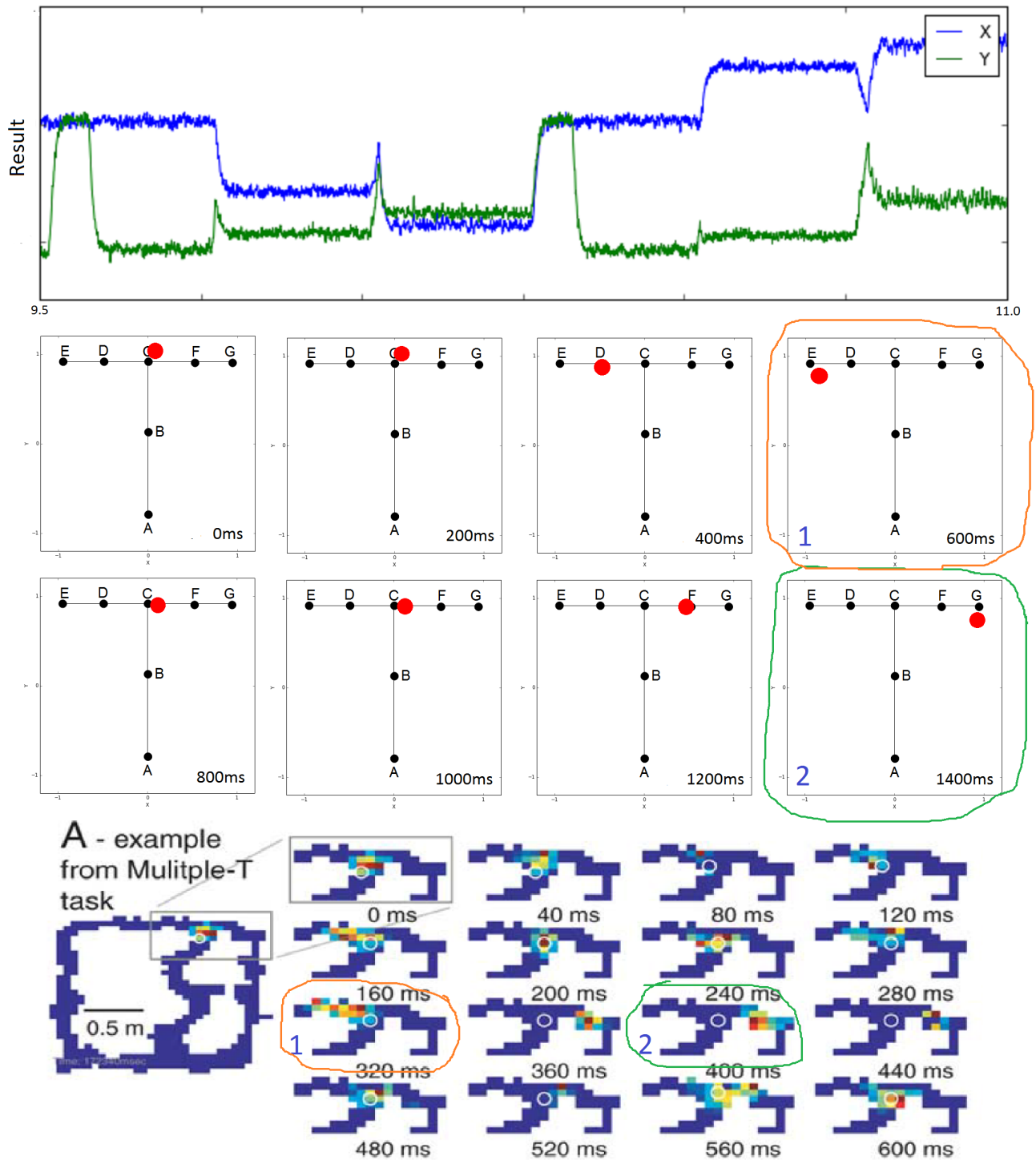


Figure 5.7: Reconstructed position in both rat (bottom, from [56]) and model (middle) at the decision point of a T maze. Although the time scale is different, forward replay effects are visible in both experimental results and model as the rat ponders which branch of the maze to take. Blue numbers label comparable position reconstructions between rat and model.

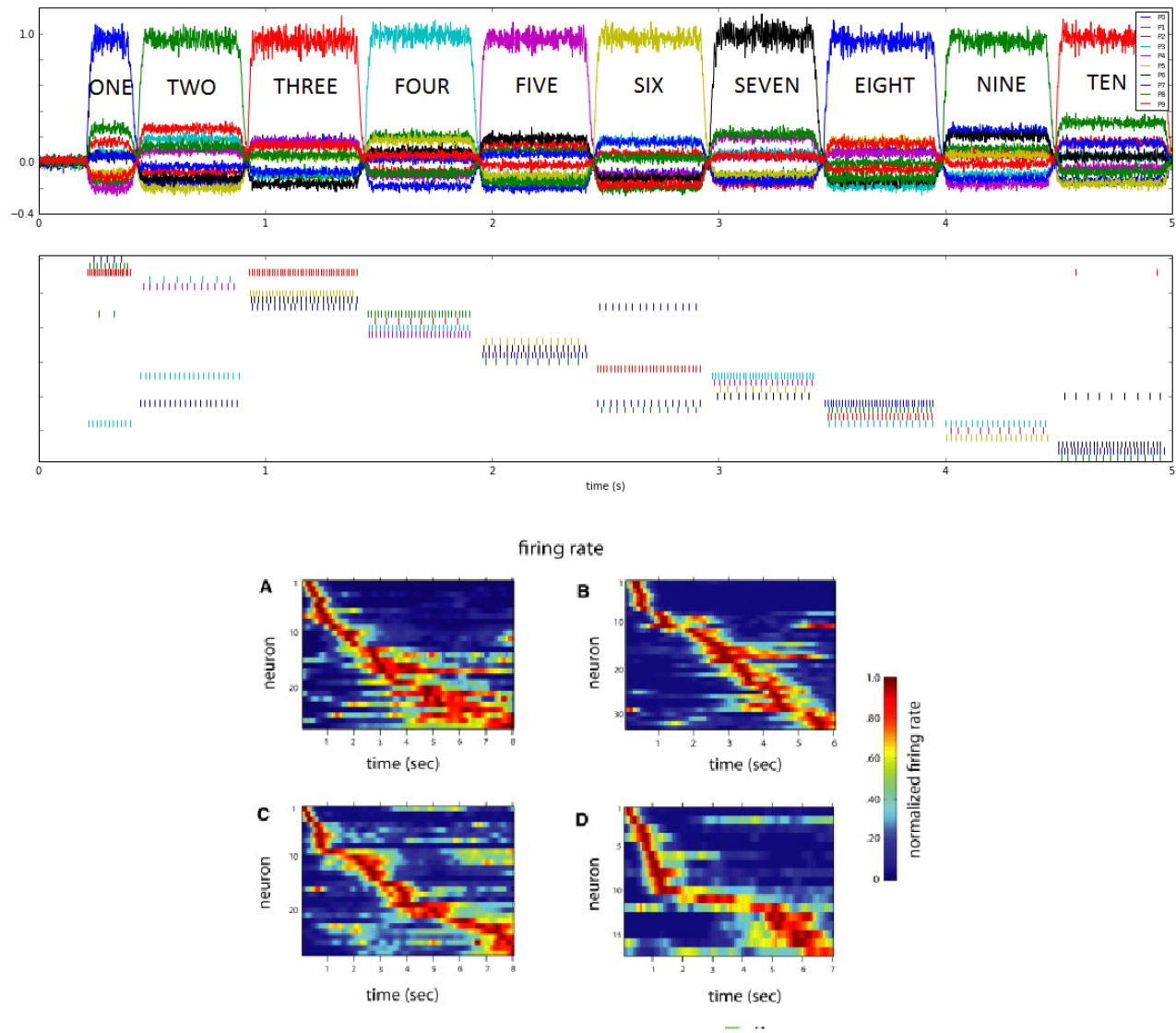


Figure 5.8: Temporally correlated firing patterns in the *Position* population of the model vs. that found in rats. The top plot shows the decoded value of this population in the model. The middle plot shows spike trains of individual neurons in the population, and the bottom 4 plots are heat maps of firing rates found in the hippocampus of rats. The model was initialized with a *reset* signal for 0.2 seconds, then left to run with no further input for 5 seconds. Rat data is taken from [64]. For the rat data, panels A-D each represent a different rat. Neurons are sorted by latency of firing rate.

5.4 Theta

The oscillator serving as a neural ‘clock’ controls the temporal dynamics of the model, leading to theta-like oscillations in CA3. The oscillations arise from inhibitory connections in the indexing part of the model, suppressing the firing of certain populations when the clock signal is on or off. In effect, these oscillations determine the time scale for both encoding and recall: as our clock frequency increases, the model encodes or recalls sensory experiences more quickly.

In rats, theta rhythm is correlated with movement, increasing in frequency when the rat moves more quickly [47]. In other animals, including humans, theta is associated with sensory stimuli [87] and spatial memory [57]. While the exact function of biological theta rhythm is still unclear, there is enough evidence to suggest that just like in our model, it is involved in (or arises from) temporal control of memory encoding and recall. Figure 5.9 gives a comparison between theta rhythm in our model vs. that found in experimental studies in rats during exploration.

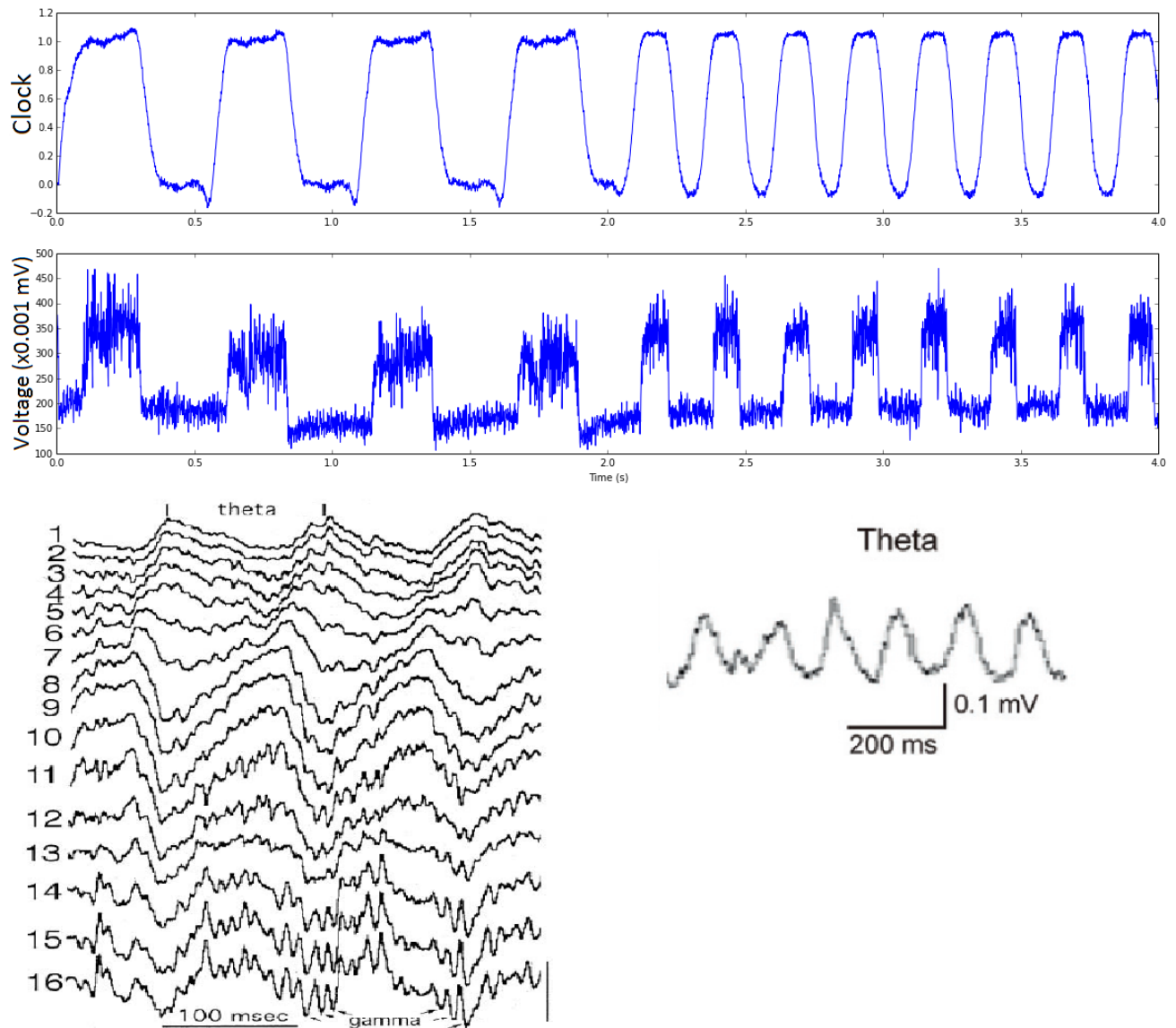


Figure 5.9: Comparison of theta oscillations in CA3 indexing populations (difference units) in model (top) vs. that found in rats (bottom-left [12] and bottom-right [40]). Model data is collected from voltage probes of neurons in the difference units in the CA3 indexing part of the model. Frequency of theta oscillations in the model fluctuate depending on the frequency control input to the oscillator. Shown are oscillations of 2 Hz ($t = [0.0, 2.0]$) and 4 Hz ($t = [2.0, 4.0]$).

Chapter 6

Discussion and Future Work

6.1 Contribution

To the best of our knowledge, this is the first biologically plausible spiking neural model to take into account both timing and sensory tuning of hippocampal cells, while exhibiting the ability to recall previously experienced sequences. In addition, it operates on arbitrary sensory vectors as input, thus not constraining the model to spatial or non-spatial information and allowing it to be extended to perform spatial navigation tasks. While other neural hippocampal models exist, they either do not provide a consistent representation for multi-modal data [62, 47], or are not biologically plausible [108].

The primary contribution of this thesis is to provide a ‘first attempt’ at creating a modular hippocampal model that performs episodic memory encoding and recall. We use the NEF as a way of unifying low-level neural phenomena with high-level behaviour, and semantic pointers as a convenient vector representation of sensory data. We hope the model can be extended and integrated with other NEF models in the future to give a more complete picture of the hippocampus’ role in the brain.

Rasmussen’s recent NEF model of hierarchical reinforcement learning [84] is one potential integration point for our model. In a Morris water maze environment [72], for example, our hippocampal model could be used for navigation, remembering sequences of paths and whether or not there was a reward at the end of them, while the reinforcement learning model determines which path to take. Another interesting integration point is Hunsberger’s visual system model [52], which could be used to provide more realistic sensory input to our model, instead of the randomly generated input vectors we use in testing.

6.2 Potential Model Extensions

While our model is able to encode and recall sequences and replicate some phenomena found in the hippocampus, by no means is it complete. We do not attempt to model reverse replay; our version of theta oscillations are not quite the same as those found in the brain, and omitted from the model is the extraction of spatial data from entorhinal grid cells. This section outlines some of these possible extensions to the model.

One issue that our model leaves unaddressed is the fact that the cleanup memory used to disambiguate our *result* vector requires the possible sensory vectors to be known beforehand. For sensory experiences that have been previously experienced, this isn't a problem, as something like Crawford's NEF knowledge representation model [20] could be used to disambiguate concepts. But what about novel sensory experiences that are being encoded for the first time? We hypothesize that neurogenesis in the dentate gyrus could be involved in creating neural representations of novel experiences that could be added to our knowledge base and thus used in cleanup memory, but exactly how this is done is outside the scope of our model.

We also do not attempt to explain the mechanisms by which place cells extract spatial data from entorhinal grid cells; that is left to other models. Orchard's recent NEF model is a good example of this [82], and could be combined with our model to provide a more biologically realistic place cell representation for both the sensory and context inputs.

In addition to the forward replay effects that we see in our model, experimental evidence also points to the existence of reverse replay in the hippocampus, often seen in rats immediately following a reward [35]. In order to implement reverse replay in our model, we would need a way of allowing our indexing circuit to count backwards in addition to counting forwards. Mathematically, this would simply mean inverting the linear transformation performed by convolving with the *ADD1* vector in the CA3 indexing part of the model. Implementing this would require providing a second route through which the two difference units in that part of the model could interact, and inhibiting one route or the other based on a control signal.

6.3 Further Discussion

One of the most interesting implications of our model is the use of theta rhythm as a control signal for encoding/recall speed. We use this as a method for performing temporal compression during recall, as well as for controlling the model's memory encoding speed.

We can even slow our clock’s frequency to zero, which turns off theta oscillations completely and tells the model to not encode anything. This is consistent with theta rhythms observed in rats, where theta increases in speed as the animal moves around its environment more quickly, and disappears completely during periods of inactivity such as grooming, where presumably no information about the environment would need to be encoded. Our model would make the prediction that theta (or some sort of network oscillation) occurs during replay as well as during encoding, which seems to be true in humans [57] and REM sleep in rats [59], but not necessarily for normal recall events in rats [22]. In any case, the use of theta as a control signal for our model seems to be an incomplete explanation, and is something to keep in mind when discussing extensions to the model. We also do not include explanations for theta phase precession [80], or for gamma oscillations or ripple events in hippocampus.

When we think about episodic memories intuitively, we often make unconscious distinctions between higher-level and lower-level memories. For example, one can think about their memory of getting ready for work in the morning. The memory could be the sequence: get out of bed, shower, eat breakfast, leave the house. Each of these elements of the memory would be associated with a location in space. However, each of these elements could also be broken down into sub-elements. Eat breakfast, for example, could be composed of the sequence: walk to kitchen, fill bowl with cereal, pour milk in bowl, consume cereal. Each of these sub-components could be broken down into even smaller sub-sub-components, with spatial information about the exact position in the kitchen that the event took place. This suggests that the process of episodic memory encoding and recall (in humans at least) could be a hierarchy, with complete low-level memories as elements of higher-level memories. If we make this assumption, our model’s compressed representation of memories as vectors with dimensionality equal to that of their components provides a natural way of re-using memories as modular components of other memories. We believe that this is a strong argument in favour of our representation of memories as semantic pointer vectors.

Memories are an integral part of our humanity. Although we may never fully understand exactly how our brains form and recall memories, we hope that our work here has provided a small step towards a better understanding of memory, and towards the greater goal of understanding the human brain.

References

- [1] Joseph Altman. Are new neurons formed in the brains of adult mammals? *Science*, 135(3509):1127–1128, 1962.
- [2] Pablo Alvarez and Larry R Squire. Memory consolidation and the medial temporal lobe: a simple network model. *Proceedings of the National Academy of Sciences*, 91(15):7041–7045, 1994.
- [3] Per Andersen, Richard Morris, David Amaral, Tim Bliss, and John O’Keefe. *The hippocampus book*. Oxford University Press, 2006.
- [4] David A August and William B. Levy. Temporal sequence compression by an integrate-and-fire model of hippocampal area CA3. *Journal of computational neuroscience*, 6(1):71–90, 1999.
- [5] Alan D Baddeley and Graham Hitch. Working memory. *Psychology of learning and motivation*, 8:47–89, 1974.
- [6] Heather R Bailey, Jeffrey M Zacks, David Z Hambrick, Rose T Zacks, Denise Head, Christopher A Kurby, and Jesse Q Sargent. Medial temporal lobe volume predicts elders’ everyday memory. *Psychological science*, 24(7):1113–1122, 2013.
- [7] Arnold Bakker, C Brock Kirwan, Michael Miller, and Craig EL Stark. Pattern separation in the human hippocampal CA3 and dentate gyrus. *Science*, 319(5870):1640–1642, 2008.
- [8] Trevor Bekolay. Learning in large-scale spiking neural networks. Master’s thesis, University of Waterloo, Waterloo, ON, 2011.
- [9] Trevor Bekolay, James Bergstra, Eric Hunsberger, Travis DeWolf, Terrence C Stewart, Daniel Rasmussen, Xuan Choo, Aaron Russell Voelker, and Chris Eliasmith.

- Nengo: a python tool for building large-scale functional brain models. *Frontiers in neuroinformatics*, 7, 2013.
- [10] Chris M Bird and Neil Burgess. The hippocampus and memory: insights from spatial processing. *Nature Reviews Neuroscience*, 9(3):182–194, 2008.
 - [11] Tim VP Bliss and Terje 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*, 232(2):331–356, 1973.
 - [12] György Buzsáki. Theta oscillations in the hippocampus. *Neuron*, 33(3):325–340, 2002.
 - [13] Zhe Chen, Stephen N Gomperts, Jun Yamamoto, and Matthew A Wilson. Neural representation of spatial topology in the rodent hippocampus. *Neural computation*, 26(1):1–39, 2014.
 - [14] Feng-Xuan Choo and Chris Eliasmith. A spiking neuron model of serial-order recall. In *Proceedings of the 32nd Annual Conference of the Cognitive Science Society*, pages 2188–93, 2010.
 - [15] John Cohen and Howard Eichenbaum. *Memory, amnesia, and the hippocampal system*. MIT press, 1993.
 - [16] Neal J Cohen and Larry R Squire. Preserved learning and retention of pattern-analyzing skill in amnesia: Dissociation of knowing how and knowing that. *Science*, 210(4466):207–210, 1980.
 - [17] Laura Lee Colgin, Edvard I Moser, and May-Britt Moser. Understanding memory through hippocampal remapping. *Trends in neurosciences*, 31(9):469–477, 2008.
 - [18] John Conklin and Chris Eliasmith. A controlled attractor network model of path integration in the rat. *Journal of computational neuroscience*, 18(2):183–203, 2005.
 - [19] Suzanne Corkin, David G Amaral, R Gilberto González, Keith A Johnson, and Bradley T Hyman. Hms medial temporal lobe lesion: findings from magnetic resonance imaging. *The Journal of Neuroscience*, 17(10):3964–3979, 1997.
 - [20] Eric Crawford, Matthew Gingerich, and Chris Eliasmith. Biologically plausible, human-scale knowledge representation. In *35th Annual Conference of the Cognitive Science Society*, pages 412–417. Cognitive Science Society, 2013.

- [21] Vassilis Cutsuridis, Bruce Graham, Stuart Cobb, and Imre Vida. *Hippocampal microcircuits*. Springer, 2010.
- [22] Thomas J Davidson, Fabian Kloosterman, and Matthew A Wilson. Hippocampal replay of extended experience. *Neuron*, 63(4):497–507, 2009.
- [23] Peter Dayan and Laurence F Abbott. *Theoretical neuroscience*. Cambridge, MA: MIT Press, 2001.
- [24] Dominique Debanne, Beat H Gähwiler, and Scott M Thompson. Long-term synaptic plasticity between pairs of individual CA3 pyramidal cells in rat hippocampal slice cultures. *The Journal of Physiology*, 507(1):237–247, 1998.
- [25] Wei Deng, James B Aimone, and Fred H Gage. New neurons and new memories: how does adult hippocampal neurogenesis affect learning and memory? *Nature Reviews Neuroscience*, 11(5):339–350, 2010.
- [26] Sachin S Deshmukh and Upinder S Bhalla. Representation of odor habituation and timing in the hippocampus. *The Journal of neuroscience*, 23(5):1903–1915, 2003.
- [27] Sachin S Deshmukh and James J Knierim. Representation of non-spatial and spatial information in the lateral entorhinal cortex. *Frontiers in behavioral neuroscience*, 5, 2011.
- [28] Kamran Diba and György Buzsáki. Forward and reverse hippocampal place-cell sequences during ripples. *Nature neuroscience*, 10(10):1241–1242, 2007.
- [29] Howard Eichenbaum. Hippocampus: Mapping or memory? *Current Biology*, 10(21):R785–R787, 2000.
- [30] Howard Eichenbaum and Neal J Cohen. *From conditioning to conscious recollection: Memory systems of the brain*. Oxford University Press, 2001.
- [31] Chris Eliasmith. A unified approach to building and controlling spiking attractor networks. *Neural computation*, 17(6):1276–1314, 2005.
- [32] Chris Eliasmith. *How to build a brain: A neural architecture for biological cognition*. Oxford University Press, 2013.
- [33] Chris Eliasmith and Charles H Anderson. *Neural engineering: Computation, representation, and dynamics in neurobiological systems*. MIT Press, 2004.

- [34] Chris Eliasmith, Terrence C Stewart, Xuan Choo, Trevor Bekolay, Travis DeWolf, Yichuan Tang, and Daniel Rasmussen. A large-scale model of the functioning brain. *science*, 338(6111):1202–1205, 2012.
- [35] David J Foster and Matthew A Wilson. Reverse replay of behavioural sequences in hippocampal place cells during the awake state. *Nature*, 440(7084):680–683, 2006.
- [36] Stephen I Gallant and T Wendy Okaywe. Representing objects, relations, and sequences. *Neural computation*, 25(8):2038–2078, 2013.
- [37] Ross W Gayler. Vector symbolic architectures answer Jackendoff’s challenges for cognitive neuroscience. *arXiv preprint cs/0412059*, 2004.
- [38] Hagar Gelbard-Sagiv, Roy Mukamel, Michal Harel, Rafael Malach, and Itzhak Fried. Internally generated reactivation of single neurons in human hippocampus during free recall. *Science*, 322(5898):96–101, 2008.
- [39] Apostolos P Georgopoulos, Andrew B Schwartz, and Ronald E Kettner. Neuronal population coding of movement direction. *Science*, 233(4771):1416–1419, 1986.
- [40] MJ Gillies, RD Traub, FEN LeBeau, CH Davies, T Gloveli, EH Buhl, and MA Whittington. A model of atropine-resistant theta oscillations in rat hippocampal area CA1. *The Journal of physiology*, 543(3):779–793, 2002.
- [41] Bennet S Givens and David S Olton. Cholinergic and gabaergic modulation of medial septal area: effect on working memory. *Behavioral neuroscience*, 104(6):849, 1990.
- [42] PS Goldman-Rakic. Cellular basis of working memory. *Neuron*, 14(3):477–485, 1995.
- [43] John D Green and Arnaldo A Arduini. Hippocampal electrical activity in arousal. *J. neurophysiol*, 17(6):533–557, 1954.
- [44] Anoopum S Gupta, Matthijs AA van der Meer, David S Touretzky, and A David Redish. Hippocampal replay is not a simple function of experience. *Neuron*, 65(5):695–705, 2010.
- [45] Torkel Hafting, Marianne Fyhn, Sturla Molden, May-Britt Moser, and Edvard I Moser. Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436(7052):801–806, 2005.

- [46] Michael E Hasselmo. What is the function of hippocampal theta rhythm? Linking behavioral data to phasic properties of field potential and unit recording data. *Hippocampus*, 15(7):936–949, 2005.
- [47] Michael E Hasselmo. *How we remember: brain mechanisms of episodic memory*. MIT press, 2012.
- [48] Michael E Hasselmo, Eric Schnell, and Edi Barkai. Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region CA3. *The Journal of neuroscience*, 15(7):5249–5262, 1995.
- [49] Michael E Hasselmo and Bradley P Wyble. Free recall and recognition in a network model of the hippocampus: simulating effects of scopolamine on human memory function. *Behavioural brain research*, 89(1):1–34, 1997.
- [50] Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. Wiley, 1949.
- [51] Donald R Humphrey, EM Schmidt, and WD Thompson. Predicting measures of motor performance from multiple cortical spike trains. *Science*, 170(3959):758–762, 1970.
- [52] Eric Hunsberger, Peter Blouw, James Bergstra, and Chris Eliasmith. A neural model of human image categorization. In *35th Annual Conference of the Cognitive Science Society*, pages 633–638. Cognitive Science Society, 2013.
- [53] Aziz Hurzook, Oliver Trujillo, and Chris Eliasmith. Visual motion processing and perceptual decision making. In *35th Annual Conference of the Cognitive Science Society*, pages 2590–2595. Cognitive Science Society, 2013.
- [54] Tim Jarsky, Alex Roxin, William L Kath, and Nelson Spruston. Conditional dendritic spike propagation following distal synaptic activation of hippocampal CA1 pyramidal neurons. *Nature neuroscience*, 8(12):1667–1676, 2005.
- [55] Daoyun Ji and Matthew A Wilson. Coordinated memory replay in the visual cortex and hippocampus during sleep. *Nature neuroscience*, 10(1):100–107, 2006.
- [56] Adam Johnson and A David Redish. Neural ensembles in CA3 transiently encode paths forward of the animal at a decision point. *The Journal of neuroscience*, 27(45):12176–12189, 2007.

- [57] Michael J Kahana, Robert Sekuler, Jeremy B Caplan, Matthew Kirschen, and Joseph R Madsen. Human theta oscillations exhibit task dependence during virtual maze navigation. *Nature*, 399(6738):781–784, 1999.
- [58] Christoph Kolodziejcki, Bernd Porr, and Florentin Wörgötter. On the asymptotic equivalence between differential hebbian and temporal difference learning. *Neural computation*, 21(4):1173–1202, 2009.
- [59] Lai-Wo Stan Leung. Theta rhythm during REM sleep and waking: correlations between power, phase and frequency. *Electroencephalography and clinical neurophysiology*, 58(6):553–564, 1984.
- [60] Stefan Leutgeb, Jill K Leutgeb, Carol A Barnes, Edvard I Moser, Bruce L McNaughton, and May-Britt Moser. Independent codes for spatial and episodic memory in hippocampal neuronal ensembles. *Science*, 309(5734):619–623, 2005.
- [61] WB Levy and O Steward. Temporal contiguity requirements for long-term associative potentiation/depression in the hippocampus. *Neuroscience*, 8(4):791–797, 1983.
- [62] William B Levy, Ashlie B Hocking, and Xiangbao Wu. Interpreting hippocampal function as recoding and forecasting. *Neural Networks*, 18(9):1242–1264, 2005.
- [63] Gary S Lynch, Thomas Dunwiddie, and Valentin Gribkoff. Heterosynaptic depression: a postsynaptic correlate of long-term potentiation. 1977.
- [64] Christopher J MacDonald, Kyle Q Lepage, Uri T Eden, and Howard Eichenbaum. Hippocampal “time cells” bridge the gap in memory for discontinuous events. *Neuron*, 71(4):737–749, 2011.
- [65] David MacNeil and Chris Eliasmith. Fine-tuning and the stability of recurrent neural networks. *PloS one*, 6(9):e22885, 2011.
- [66] Ludise Malkova and Mortimer Mishkin. One-trial memory for object-place associations after separate lesions of hippocampus and posterior parahippocampal region in the monkey. *The Journal of neuroscience*, 23(5):1956–1965, 2003.
- [67] Henry Markram, Joachim Lübke, Michael Frotscher, and Bert Sakmann. Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science*, 275(5297):213–215, 1997.

- [68] James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3):419, 1995.
- [69] Michael J Milford, Gordon F Wyeth, and David Prasser. Ratslam: a hippocampal model for simultaneous localization and mapping. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 1, pages 403–408. IEEE, 2004.
- [70] Brenda Milner, Suzanne Corkin, and H-L Teuber. Further analysis of the hippocampal amnesic syndrome: 14-year follow-up study of HM. *Neuropsychologia*, 6(3):215–234, 1968.
- [71] RG Morris. Synaptic plasticity and learning: selective impairment of learning rats and blockade of long-term potentiation in vivo by the n-methyl-d-aspartate receptor antagonist AP5. *The Journal of neuroscience*, 9(9):3040–3057, 1989.
- [72] Richard Morris. Developments of a water-maze procedure for studying spatial learning in the rat. *Journal of neuroscience methods*, 11(1):47–60, 1984.
- [73] Edvard I Moser, Emilio Kropff, and May-Britt Moser. Place cells, grid cells, and the brain's spatial representation system. *Neuroscience*, 31(1):69, 2008.
- [74] Kazu Nakazawa, Michael C Quirk, Raymond A Chitwood, Masahiko Watanabe, Mark F Yeckel, Linus D Sun, Akira Kato, Candice A Carr, Daniel Johnston, Matthew A Wilson, et al. Requirement for hippocampal CA3 NMDA receptors in associative memory recall. *Science*, 297(5579):211–218, 2002.
- [75] Kenneth A Norman and Randall C O'Reilly. Modeling hippocampal and neocortical contributions to recognition memory: a complementary-learning-systems approach. *Psychological review*, 110(4):611, 2003.
- [76] Erkki Oja. Neural networks, principal components, and subspaces. *International journal of neural systems*, 1(01):61–68, 1989.
- [77] John O'Keefe. Do hippocampal pyramidal cells signal non-spatial as well as spatial information? *Hippocampus*, 9(4):352–364, 1999.
- [78] John O'Keefe and Jonathan Dostrovsky. The hippocampus as a spatial map. preliminary evidence from unit activity in the freely-moving rat. *Brain research*, 34(1):171–175, 1971.

- [79] John O’Keefe and Lynn Nadel. *The hippocampus as a cognitive map*, volume 3. Clarendon Press Oxford, 1978.
- [80] John O’Keefe and Michael L Recce. Phase relationship between hippocampal place units and the eeg theta rhythm. *Hippocampus*, 3(3):317–330, 1993.
- [81] Shane M O’Mara, Sean Commins, Michael Anderson, and John Gigg. The subiculum: a review of form, physiology and function. *Progress in neurobiology*, 64(2):129–155, 2001.
- [82] Jeff Orchard, Hao Yang, and Xiang Ji. Does the entorhinal cortex use the Fourier transform? *Frontiers in computational neuroscience*, 7, 2013.
- [83] Tony A Plate. Holographic reduced representations. *Neural networks, IEEE transactions on*, 6(3):623–641, 1995.
- [84] Daniel Rasmussen and Chris Eliasmith. A neural model of hierarchical reinforcement learning. In *Proceedings of the 36th Annual Conference of the Cognitive Science Society*, pages 1252–1257. Cognitive Science Society, 2014.
- [85] JNP Rawlins, J Feldon, and JA Gray. Septo-hippocampal connections and the hippocampal theta rhythm. *Experimental Brain Research*, 37(1):49–63, 1979.
- [86] Nancy L Rempel-Clower, Stuart M Zola, Larry R Squire, and David G Amaral. Three cases of enduring memory impairment after bilateral damage limited to the hippocampal formation. *The Journal of Neuroscience*, 16(16):5233–5255, 1996.
- [87] Robert S Sainsbury, Arnold Heynen, and Christopher P Montoya. Behavioral correlates of hippocampal type 2 theta in the rat. *Physiology & behavior*, 39(4):513–519, 1987.
- [88] Alexei Samsonovich and Bruce L McNaughton. Path integration and cognitive mapping in a continuous attractor neural network model. *The Journal of neuroscience*, 17(15):5900–5920, 1997.
- [89] Francesca Sargolini, Marianne Fyhn, Torkel Hafting, Bruce L McNaughton, Menno P Witter, May-Britt Moser, and Edvard I Moser. Conjunctive representation of position, direction, and velocity in entorhinal cortex. *Science*, 312(5774):758–762, 2006.
- [90] William Beecher Scoville and Brenda Milner. Loss of recent memory after bilateral hippocampal lesions. *Journal of neurology, neurosurgery, and psychiatry*, 20(1):11, 1957.

- [91] Patricia E Sharp. Subicular cells generate similar spatial firing patterns in two geometrically and visually distinctive environments: comparison with hippocampal place cells. *Behavioural brain research*, 85(1):71–92, 1997.
- [92] Trygve Solstad, Charlotte N Boccara, Emilio Kropff, May-Britt Moser, and Edward I Moser. Representation of geometric borders in the entorhinal cortex. *Science*, 322(5909):1865–1868, 2008.
- [93] Terrence C Stewart, Yichuan Tang, and Chris Eliasmith. A biologically realistic cleanup memory: Autoassociation in spiking neurons. *Cognitive Systems Research*, 12(2):84–92, 2011.
- [94] Khena M Swallow, Deanna M Barch, Denise Head, Corey J Maley, Derek Holder, and Jeffrey M Zacks. Changes in events alter how people remember recent information. *Journal of Cognitive Neuroscience*, 23(5):1052–1064, 2011.
- [95] Hiroto Takahashi and Jeffrey C Magee. Pathway interactions and synaptic plasticity in the dendritic tuft regions of CA1 pyramidal neurons. *Neuron*, 62(1):102–111, 2009.
- [96] Jeffrey S Taube, Robert U Muller, and James B Ranck. Head-direction cells recorded from the postsubiculum in freely moving rats. i. description and quantitative analysis. *The Journal of Neuroscience*, 10(2):420–435, 1990.
- [97] CD Tesche and J Karhu. Theta oscillations index human hippocampal activation during a working memory task. *Proceedings of the National Academy of Sciences*, 97(2):919–924, 2000.
- [98] E Tulving. Elements of episodic memory, 1983.
- [99] Endel Tulving. Episodic and semantic memory 1. *Organization of Memory. London: Academic*, 381:e402, 1972.
- [100] Endel Tulving. How many memory systems are there? *American psychologist*, 40(4):385, 1985.
- [101] Endel Tulving and Hans J Markowitsch. Episodic and declarative memory: role of the hippocampus. *Hippocampus*, 8(3), 1998.
- [102] Matthijs AA Van Der Meer and A David Redish. Expectancies in decision making, reinforcement learning, and ventral striatum. *Frontiers in neuroscience*, 4, 2010.

- [103] Aaron Russell Voelker, Eric Crawford, and Chris Eliasmith. Learning large-scale heteroassociative memories in spiking neurons. In *13th International Conference, UCNC*, 2014.
- [104] Adam C Welday, I Gary Shlifer, Matthew L Bloom, Kechen Zhang, and Hugh T Blair. Cosine directional tuning of theta cell burst frequencies: evidence for spatial coding by oscillatory interference. *The Journal of neuroscience*, 31(45):16157–16176, 2011.
- [105] Matthew A Wilson and Bruce L McNaughton. Reactivation of hippocampal ensemble memories during sleep. *Science*, 265(5172):676–679, 1994.
- [106] Laurenz Wiskott, Malte J Rasch, and Gerd Kempermann. A functional hypothesis for adult hippocampal neurogenesis: avoidance of catastrophic interference in the dentate gyrus. *Hippocampus*, 16(3):329–343, 2006.
- [107] Menno P Witter, Floris G Wouterlood, Pieterke A Naber, and Theo Van Haeften. Anatomical organization of the parahippocampal-hippocampal network. *Annals of the New York Academy of Sciences*, 911(1):1–24, 2000.
- [108] Yan Wu. In search of lost time: A computational model of the representations and dynamics of episodic memory. Master’s thesis, University of Waterloo, Waterloo, ON, 2012.
- [109] Sheng-Jia Zhang, Jing Ye, Jonathan J Couey, Menno Witter, Edvard I Moser, and May-Britt Moser. Functional connectivity of the entorhinal–hippocampal space circuit. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369(1635):20120516, 2014.
- [110] Eric A Zilli and Michael E Hasselmo. Coupled noisy spiking neurons as velocity-controlled oscillators in a model of grid cell spatial firing. *The Journal of neuroscience*, 30(41):13850–13860, 2010.