Incorporating Biologically Realistic Neuron Models into the NEF

by

Peter Duggins

A thesis
presented to the University of Waterloo
in fullfilment of the
thesis requirements for the degree of
Master of Applied Science
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2017

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

Theoretical neuroscience is fundamentally concerned with the relationship between biological mechanisms, information processing, and cognitive abilities, yet current models often lack either biophysical realism or cognitive functionality. This thesis aims to partially fill this gap by incorporating geometrically and electrophisologically accurate models of individual neurons into the Neural Engineering Framework (NEF). After discussing the relationship between biologically complex neurons and the core principles/assumptions of the NEF, a neural model of working memory is introduced to demonstrate the NEF's existing capacity to capture biological and cognitive features. This model successfully performs the delayed response task and provides a medium for simulating mental disorders (ADHD) and its pharmacological treatments. Two methods of integrating more biologically sophisticated `NEURON` models into the NEF are subsequently explored and their ability to implement networks of varying complexity are assessed: the trained synaptic weights do realize the core NEF principles, though several errors remain unresolved. Returning to the working memory model, it is shown that bioneurons can perform the requisite computations in context, and that simulating the biophysical effects of pharmacological compounds produces results consistent with electrophysiological and behavioral data from monkeys.

Acknowledgements

I would like to thank

- **Chris Eliasmith**, for his leadership of the Computational Neuroscience Research Group, for his suggestions and encouragement through periods of scientific struggle, and for his astute navigation of friendship and advisorship.
- **Terrence C. Stewart**, for his help in designing the model in Chapter 2, developing the methods in Chapter 3, and refining various pieces of software and writing.
- **Aaron Voelker**, for his scientific and mathematical insights, software design, and patient guidance in developing the methods in Chapter 4.
- **Bryan Tripp, Daniel Stashuk, and Chris Eliasmith** for providing feedback on drafts of this thesis.
- My **housemates**, **bandmates**, **labmates**, and **teammates**, for being such supportive, engaging, and though-provoking people. You know who you are :).
- My parents, **Megan** and **David**, for basically everything.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Biological Plausibility and the NEF

1.1.1 Goals

Theoretical neuroscience seeks theories of information processing in the brain built upon techniques, mathematics, and data from numerous scientific disciplines. From a top-down perspective, the goal of computational neuroscience is to understand how cognition results from the computations performed in the brain and to explain how networks of neurons realize these functional capabilities. From a bottom-up perspective, the goal of computational neuronscience is to explain the functional relevance of particular neurobiological features and to describe how they interact to realize distributed computation. Although vast amounts of neural and behavioral data have been gathered in pursuit of these goals, the field still lacks broadly-accepted theories that consolidate this knowledge into a coherent description of the functional brain, leading some to describe the field as "data rich but theory poor" (Churchland and Sejnowski, 2016).

The Neural Engineering Framework, or **NEF** (Eliasmith and Anderson, 2003), is an attempt to unify neuroscientific data, engineering methods, and computational simulation into a coherent theory of brain function. The NEF provides tools that translate high-level, algorithmic descriptions of the brain's

computational processes into low-level neurobiological parameters. For instance, the NEF can be applied to translate the differential equations describing an **integrator**, a theoretical system that accumulates inputs and maintains the result over time, into synaptic weights of a recurrently-connected population of neurons. The resulting neural network realizes a **working memory**, a neural system that actively maintains information over short time periods and is capable of performing behavioral tasks.

The NEF also respects a number of biological constraints: it uses spiking neurons with limited firing rates and exponential synaptic filters; the number of neurons and their connectivity respect parameters of the relevant brain area(s); and learning rules rely on spike-timing dependent plasticity and/or error signals from perceptual systems. However, the NEF does not aim to explain all of the brain's biological features. This raises the question, "Which features are sufficiently (computationally) relevant to be included in the theory?" Although the features currently utilized by the NEF are generally regarded as functionally significant by the computational neuroscience community, some have argued that the NEF does not include enough biological detail to constitute a "brain model" (Sanders, 2013). Implicit in this statement is the assertion that NEF models lack certain biological mechanisms that are needed to understand the brain's cognitive capabilities or match certain classes of empirical data. This is a valid concern, but must be weighed against the dangers of introducing complexity for its own sake, i.e., incorporating features whose functional significance is unclear or whose mechanisms do not increase the theory's explanatory power.

1.1.2 Motivation

There are three principal reasons why the NEF will benefit from increased biological realism, specifically the development of tools to simulate biologically

realistic neurons. First, it is important to show that networks of complex artificial cells can represent information and perform computations on that information. Existing NEF models generally use analytically-tractable neuron models that reduce the complexity of biological cells down to a few essential equations, then utilize mathematical tools to implement representation and computation (two essential attributes of cognitive brains) by parameterizing neural networks populated with these cells. It is an open question whether these methods, and the models that have been built using them, will work when simple neuron models are replaced with biologically realistic ones. If the extensions introduced in this thesis are successful, it will advance the claim that the NEF is capable of engineering functional neural systems built from a biologically realistic neural substrate.

Second, explicitly modeling the geometry and electrophysiology of neurons will allow researchers to investigate how low-level biophysical and neural mechanisms/perturbations affect higher-level cognitive functions. In addition to being theoretically valuable, these tools can be used to simulate and study biophysically-grounded mental disorders such as Parkinson's Disease, which cause suffering for millions of people worldwide. Our current understanding of these disorders relies heavily on animal experiments which face severe scientific and ethical limitations. For instance, primate studies have small sample sizes, require expensive animal care, and take years to execute, while current technology limits simultaneous data collection at multiple scales. Computational neuroscience is a promising alternative. Simulations are quick, inexpensive, and reproducible, and the data they generate can be measured with arbitrary precision at multiple scales, allowing researchers to develop a more integrated picture of disorders. Furthermore, if computational models are sufficiently detailed to capture the underlying mechanisms of mental disorders (and have been convincingly validated), researchers can design and test drug treatments

by perturbing these models to simulate pharmacological or behavioral treatments. This methodology is potentially more ethically responsible and more experimentally targeted than clinical trials on animals. However, existing models rarely include both biological detail and functional capabilities, which makes them ill-suited for the multifaceted study of mental disorders. One goal of this extension is to develop models that capture a broader set of these low- and high-level features, including enough biophysical detail to simulate the underlying causes of mental disorders and drug treatments, enough neural detail to produce data that can be externally validated, and enough functional detail to conceptually describe why brain systems break down with mental disorders.

Finally, new classes of computation and behavior may become available with the use of biologically realistic cells. For example, it has been argued that nonlinear functions may be computed in dendritic trees, increasing the computational significance of individual neurons (London and Häusser, 2005). Similarly, **neuromodulators**, neurotransmitters that have regulatory effects on the post-synaptic cell, may allow novel forms of neural control, including the emotional modulation of cognition and behavior by structures such as the amygdala (Hermans et al., 2014; Phelps, 2004; Pessoa, 2008). Cognitive neuroscience currently lacks quantitative theories for how these biological features relate to these functional operations. Formulating and validating such a theory requires a simulation environment that connects the biophysical to the functional: the proposed extension to the NEF aims to fill this gap.

1.1.3 Outline

This thesis is structured as follows. The remainder of Chapter 1 is devoted to introducing the NEF, discussing its simplifying assumptions, and examining how neuron models built from `NEURON` can reintroduce biological complexity.

Chapter 2 explores how some of these biological features can be approximated using existing NEF methods (i.e., without introducing realistic neurons) using a model of working memory that reproduces electrophysiological and behavioral data from monkeys. Chapters 3 and 4 explore two methods for incorporating biologically realistic `NEURON` models (hereafter **bioneurons**) into the NEF, which involves training synaptic weights on the bioneurons' dendrites such that their resulting spiking behavior implements the core principles of the NEF. Finally, Chapter 5 applies these methods back to the working memory model of Chapter 2, discusses their limitations, and proposes extensions for future research.

1.2 Neural Engineering Framework (NEF)

1.2.1 Three Principles

In order to understand the brain from a functional or cognitive perspective, a theory must first describe how the brain internally represents information about the external world. The first principle of the NEF, **representation**, describes how neurons' spiking activity represents, or **encodes**, incoming information. To ensure a quantitative foundation for the theory, the NEF assumes that this information can be described in terms of an N-dimensional, time-varying, vector-valued signal $\mathbf{x}(t)$. Using vectors as a basis for representation has numerous strengths, including the general and flexible characterization of a representational hierarchy (i.e., the representation of quantities ranging from scalars and vectors through functions and vector fields (Eliasmith and Anderson, 2003)), the implementation of cognitive operations such as binding (Plate, 1995) and winner-take-all competition (Stewart and Eliasmith, 2011), and the application of powerful engineering techniques like control theory (DeWolf et al., 2016).

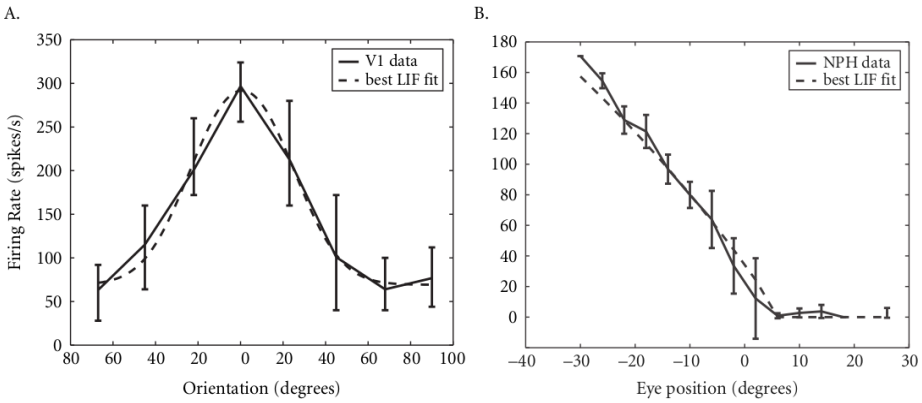

FIGURE 1.1: Example tuning curve showing the firing rate of single cells from (A) the primary visual cortex of a macaque monkey, as a function of the angle of rotation of a bar in the monkey's visual field; and (B) nuclei prepositus hypoglossi (a brain area that controls eye position) as a function of the position of the eye from midline. Error bars indicate standard deviation, and dotted lines are the best-fit of a LIF neuron model to the data. Figure reproduced from Eliasmith et al. (2012) with data from Dan Marcus and Kathleen E. Cullen.

To understand how neurons encode vector-valued information, the NEF utilizes neurons' **preferred direction vectors**. A neuron will respond (fire action potentials) most strongly when presented with this vector, and will respond less strongly to stimuli that are increasingly dissimilar from this vector. For example, a neuron that detects vertical bars in an animal's visual field will fire at its maximum rate when the animal perceives an image of a vertical line, and will fire less strongly as that line is rotated towards a horizontal orientation, as in Figure 1.1. A neuron's **tuning curve** quantifies this relationship by comparing a neuron's steady-state activity to the vector being represented. Preferred direction vectors and tuning curves have been widely used in the neuroscience literature as a useful way of characterizing neural representation (e.g., Georgopoulos et al., 1982).

In the NEF, each neuron i is assigned an N-dimension preferred direction vector, or **encoder**, $\mathbf{e}_i$. To produce a variety of tuning curves and firing rates that match the observed electrophysiological variance, neurons are also assigned a gain α_i and bias current β_i. These quantities determine how strongly

the incident vector $\mathbf{x}(t)$ drives the neuron:

$$I_{in}(t) = \alpha_i * (\mathbf{e_i} \cdot \mathbf{x}(t)) + \beta_i \tag{1.1}$$

where $I_{in}(t)$ is the current flowing into the neuron and $(\cdot)$ is the dot product between the encoder and input vector. The relationship between the neuron's firing rate and this input current is determined by the neuron model $G_i[\cdot]$, which describes the internal dynamics (of voltage, current, etc.) of the cell:

$$a_i(t) = G_i[I_{in}(t)]. \tag{1.2}$$

The NEF can be applied to a wide variety of neuron models, including sigmoid, rectified linear, rate- and spike-based leaky-integrate-and-fire neurons (**LIF**, discussed below), Izhikevich neurons, and others. So long as there is a well-defined relationship between the input current and the resulting firing rate, $G_i[\cdot]$ defines a tuning curve, the neuron's activity can be said to represent the vector $\mathbf{x}(t)$, and the NEF methods below can be applied. For spiking neurons, the output spikes are filtered using a lowpass filter $h(t)$, an operation that approximates the biophysical process in which incident spikes (individual receptor binding events) are translated to post-synaptic current (a continuous value):

$$a_i(t) = \delta_i(t - t') * h(t) \tag{1.3}$$

where $\delta_i(t - t')$ is a delta-function that rises (with unit area) each time t' that the neuron spikes, $(*)$ denotes convolution with the lowpass filter, and the filter is governed by a simple exponential decay:

$$h(t) = e^{(-t/\tau)} \quad (t > 0). \tag{1.4}$$

A distributed representation, or population encoding, extends this notion of

neural encoding: if the vector is fed into multiple neurons, each with a unique tuning curve, then each neuron will respond with a unique $a_i(t)$, and the collection of all activities will represent the signal $\mathbf{x}(t)$ with greater robustness and precision than a single neuron.

For neural encoding to be meaningful, there must be methods to recover, or **decode**, the original vector from the neurons' activities. The first principle of the NEF describes how to compute neural decoders $\mathbf{d}_i$ that perform this recovery. The second principle extends these methods to calculate decoders $\mathbf{d}_i^f$ that compute arbitrary functions of the vector, $f(\mathbf{x}(t))$. This functional decoding allows networks of neurons to **transform** the signal into a new state $\hat{\mathbf{x}}(t)$, which is essential for performing computational operations within a cognitive system. To do so, a linear decoding is applied to the activities of the neural population:

$$\hat{\mathbf{x}}(t) = \sum_{i}^{neurons} a_i(t) * \mathbf{d}_i^f, \tag{1.5}$$

where the hat notation indicates that the vector decoded from the neural activities is an imperfect estimate of the desired state due to various sources of noise (see Eliasmith and Anderson (2003)). The challenge now becomes finding decoders $\mathbf{d}_i^f$ that will compute the desired function. This can be framed as a least-squared optimization problem, with the objective of minimizing the error between the target value $\mathbf{x}(t)$ and the decoded estimate $\hat{\mathbf{x}}(t)$ over the signal.

$$E = \frac{1}{2}\int_0^T (\mathbf{x}(t) - \hat{\mathbf{x}}(t))^2 dt$$

Since a minimization over time is ill-defined (time-varying inputs can continuously change as $t \to \infty$), the NEF utilizes a rate-approximation when calculating decoders: if a constant input $\mathbf{x}$ is fed into a neuron, its activity will settle to a steady-state firing rate a_i. Using this assumption, the optimization

can be reframed as a minimization over all values of $\mathbf{x}$:

$$E = \frac{1}{2}\int_{x_{min}}^{x_{max}} (f(\mathbf{x}) - \hat{\mathbf{x}})^2 \, dx \tag{1.6}$$

$$= \frac{1}{2}\int_{x_{min}}^{x_{max}} (f(\mathbf{x}) - \sum_{i}^{neurons} a_i * \mathbf{d}_i^f)^2 \, dx. \tag{1.7}$$

Instead of integrating over all $\mathbf{x}$, the space is divided into S sample points, and a_i is computed at each value of $\mathbf{e_i} \cdot \mathbf{x}$ using the neurons' tuning curves. It can be shown (Eliasmith and Anderson, 2003) that the solution which minimizes E is given by

$$\mathbf{d}_i^f = \mathbf{\Gamma}^{-1}\mathbf{\Upsilon} \tag{1.8}$$

$$\mathbf{\Gamma} = \frac{1}{S}\mathbf{A}^T\mathbf{A} \tag{1.9}$$

$$\mathbf{\Upsilon} = \frac{1}{S}\mathbf{A}^T f(\mathbf{x}), \tag{1.10}$$

where $\mathbf{A}$ is the matrix of neural activities (the firing rate of each neuron at each sampled $\mathbf{x}$ value) and $f(\mathbf{x})$ is the function that the decoders should compute (to recover the input signal, $f(\mathbf{x})$ is the identity function).

In `nengo`, the software package used to build and simulate neural models specified using the NEF, the calculation of decoders using Equation 1.8 is handled using a `solver`. The `solver` takes as arguments the target values $f(\mathbf{x})$ and the associated activities $\mathbf{A}$ and returns the decoders. As discussed below, these target values and activities are by default the static sample points taken from the tuning curves, but the method also accepts time-varying signals $(f(\mathbf{x}(t)), \mathbf{A}(t))$ gathered from a previous simulation.

The third principle of the NEF, **dynamics**, allows neural networks to implement linear systems of the form $\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t)$, as well as nonlinear systems $\dot{\mathbf{x}} = F(\mathbf{x}, \mathbf{u}, t)$. This expands the scope of computations possible with NEF networks to include control-theoretic systems. To do so, the matrices A and B (not to be confused with the activities matrix $\mathbf{A}$ above) must be modified

to account for the dynamics that naturally occur when using neurons. If an input $\mathbf{u}(t)$ is fed into a neural population representing $\mathbf{x}(t)$, and this population is recurrently connected such that $\mathbf{x}(t)$ feeds back to itself, the dynamics of the represented state are

$$\dot{\mathbf{x}}(t) = A'\mathbf{x}(t) + B'\mathbf{u}(t) \tag{1.11}$$

$$s\mathbf{X}(s) = A'\mathbf{X}(s) + B'\mathbf{U}(s) \tag{1.12}$$

where A' and B' are the connection weight matrices on the feedforward and feedback connections, respectively, and the second line is the Laplace transform of the first. In the NEF, the dynamics coupling the inputs to the state change are governed by the synaptic filter, which is a lowpass filter by default, $H(s) = \frac{1}{1+s\tau}$. Substituting this into the above and equating with the non-neural form of the equation, A' and B' can be calculated using the simple transformation

$$A' = \mathbb{1} + \tau A \tag{1.13}$$

$$B' = \tau B. \tag{1.14}$$

When using `nengo` to construct a neural model that implements a desired dynamical system $\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t)$, one simply specifies the function $f(\mathbf{x}) = B'\mathbf{u}$ on the connection into a population of neurons (feedforward) and $f(\mathbf{x}) = A'\mathbf{x}$ on the recurrent connection (feedback). At build time, the `solver` computes the decoders $\mathbf{d}_i^f$ that implement these functional transformations. Nonlinear systems can be similarly handled: see Appendices E and F of Eliasmith and Anderson (2003).

1.2.2 LIF Neurons

Before incorporating biologically realistic neurons into the NEF, it will be helpful to review the leaky-integrate-and-fire (LIF) neuron model used in most

`nengo` models. The LIF neuron is designed to simulate neurons' internal voltage dynamics while producing artificial spikes with plausible statistics. When it comes to engineering neural systems, the LIF neuron is "a convenient and fruitful mixture of realism and simplicity" (Eliasmith and Anderson (2003)): it approximates the behavior of many types of biological neurons under a range of conditions and physiological parameters; it has been shown to be a limiting case for more complex neuron models such as the canonical Hodgkin-Huxley model (Partridge, 1966); and it goes beyond rate-based neuron models by capturing neural spikes, a critical feature of representation in biological brains.

LIF neurons are **point neurons**, a term used to describe artificial neurons which have no spatial extent. Point neurons have a single set of state variables, most importantly voltage $V(t)$, that represents the associated physical quantities across the entire cell. Point neurons can be contrasted with **compartmental neurons**, which (a) include numerous sections that each contain a set of state variables and (b) specify equations that govern how state variables in adjacent sections affect one another.

The LIF equations governing $V(t)$ describe voltage changes in the **soma**, the cell's central compartment, in response to inputs from presynaptic neurons and passive properties of the soma itself. These equations have three main features. The first is **integration**, which describes how neurons sum presynaptic inputs over time. These inputs may be **excitatory**, causing $V(t)$ to increase, or **inhibitory**, causing $V(t)$ to decrease. The second feature is **leak**, which describes the relaxation of $V(t)$ back to an equilibrium state, $V_{eq} \simeq -65\,\mathrm{mV}$. A variety of interconnected biophysical mechanisms contribute to this leak but are lumped into a single decay rate in the LIF approximation. The final feature is **firing**, the generation of a spike that is subsequently transmitted to other neurons across **synapses**, the intercellular structures that connect neurons. In the LIF model, when voltage accumulates to a **threshold** $V(t) \geq V_{thr}$, a delta function representing the spike is generated, and the voltage is immediately

reset to V_{reset} for some refractory period τ_{ref}. This procedure resembles the generation of (and recovery from) an **action potential**, a rapid rising and settling of somatic voltage that propagates down the **axon**, the elongated portion of a neuron that conducts electrical impulses towards the synapse. The complex biophysical kinetics of action potential generation are ignored in the LIF model in favor of simplicity and rapid simulation.

For a complete justification and derivation of the LIF equations, see Eliasmith and Anderson (2003). Under the assumption that $I_{in}(t)$ changes slowly compared to the interspike interval, the steady-state firing rate of a neuron a_i can be computed by passing the input through the neuron model $G[\cdot]$,

$$\begin{aligned} a_i &= G_i[J_{in}] \\ &= \frac{1}{\tau_{ref} - \tau_{rc}\log(1 - \frac{I_{thr}}{I_{in}})} \\ &= \frac{1}{\tau_{ref} - \tau_{rc}\log(1 - \frac{I_{thr}}{\alpha_i(\mathbf{x}\cdot\mathbf{e}_i)+\beta_i})}, \end{aligned} \tag{1.15}$$

where τ_{rc} is a time constant that accounts for the neuron's membrane resistance and capacitance. This equation is used to calculate the tuning curves for LIF neurons, which are in turn used to calculate decoders according to Equation 1.8.

1.2.3 Assumptions

The equations written above make several important assumptions about neurons that reduce biophysical complexity in favor of analytical tractability:

1. Multiple input signals $\mathbf{x}_0(t) \ldots \mathbf{x}_n(t)$ are linearly combined to compute the postsynaptic current that drives the neuron model (Equations 1.1 and 1.2)

2. Equivalent filters $h(t)$ are applied to each presynaptic spike when calculating the firing rate (Equation 1.3)

3. A neuron's steady-state firing rate is well defined for all values of a driving input signal $\mathbf{x}$ (Equation 1.6)

These assumptions hold true only for non-adapting point neurons: as discussed below, the spatial and electrical properties of biological neurons lead to nonlinear, time-dependent filtering of presynaptic inputs. One of the core questions of this thesis is whether the NEF can still be effectively applied if these assumptions are relaxed, i.e., if LIF neurons are replaced with biologically plausible neurons in which rate approximations, equivalent filters, and linear input summations are invalid.

1.3 Introduction to Biologically Realistic Neurons

This section demonstrates the biological realism of neuron models constructed using `NEURON` and introduces the challenges associated with their integration into the NEF.

1.3.1 NEURON

The `NEURON` simulation package is designed for modeling individual neurons and networks of neurons in a numerically sound, computationally efficient, and empirically constrained manner. It is particularly well-suited to building models that involve cells with complex anatomical and biophysical properties (Carnevale and Hines, 2006). `NEURON` simulates compartmental neurons in which each section's variables are governed by biophysically accurate differential equations such as the **cable equation**, which describes how current passes between spatial sections (e.g., down the axon) based on their relative voltages

and geometries. NEURON permits the meticulous construction of neural morphology, allowing the user to build cells that closely resemble those taken from particular brain regions (e.g., layer 5 pyramidal neurons). Geometry becomes particularly important when considering the interactions of synaptic inputs, which must be filtered through **dendrites**, the highly-branched neuronal structures onto which presynaptic neurons connect, before reaching the soma. This filtering causes nonlinear summation of presynaptic inputs, violating the LIF assumption that inputs can be added together using a weighted sum.

NEURON also simulates **ion channels**, structures within the membranes of biological cells that permit the influx and efflux of positively charged molecules such as sodium, postassium, and calcium. In biological cells, ion channels are **ligand-gated**, meaning that they open and close when presynaptic neurotransmitters bind to receptors on the postsynaptic cell. Ion channels are also **voltage-gated**, meaning that they open and close in response to changes in the cell's voltage. The nonlinear interaction between ion channels' kinetics produces action potentials when excitatory input is sufficiently large (relative to inhibitory inputs and leak channels). However, the threshold for spike generation is not as clear-cut as the LIF model would suggest: it depends on the cell's ion channel composition its recent spiking activity, a phenomenon referred to as **adaptation** that violates the rate-mode approximation. NEURON allows the user to specify the distribution of ion channels within each section of a cell; it also simulates **conductance-based synapses**, which approximate changes in a cell's membrane conductances induced by receptor-binding and thereby determine the kinetics of the postsynaptic current. To the extent that these synapses, and the dendrites on which they reside, vary between neurons, the NEF assumption of equivalent filtering is also violated.

1.3.2 Realistic NEURON Models

Taken together, these features permit the creation and simulation of biologically realistic neurons. Perhaps the best examples of this are the `NEURON` models created by the Human Brain Project, a large European collaboration that simulates sections of cortical tissue with unprecedented biological accuracy. The HBP uses cellular and synaptic experimental data to reconstruct detailed anatomy, electrophysiology, and connectivity of 55 morphological and 207 electrical neuron subtypes (Markram et al., 2015). When simulated, these neurons reproduce an array of *in vivo* experiments; they also permit the investigation of relationships between biophysical and network properties, such as the spontaneous emergence of neural synchrony as a function of calcium levels and network connectivity. The HBP's `NEURON` models are available online through the NMC portal.

The extreme realism of the HBP's `NEURON` models comes at the expense of simulation speed and neuron-type generality, motivating the selection of a reduced geometry `NEURON` model developed by Bahl et al. (2012) for this thesis. This model was developed using a three-step algorithmic strategy for optimizing compartmental neurons, reducing the number of parameters needed to simulate dendritic morphology while maintaining key aspects of geometry, voltage responses, dendritic calcium spikes, and somatic-dendritic coupling. The result is a reduced model of layer 5 pyramidal neurons that closely reproduces experimental data. It includes 7 key anatomical sections (soma, basal dendrites, apical dendrite, apical dendritic tuft, axon initial segment, axon hillock, and axon) simulated using 20 connected compartments whose geometries are derived from a detailed reconstruction by Zhu (2000). See Figure 1.2 for a diagram of the reduced model's geometry. The model also includes 9 ion channels (hyperpolarization-activated cation; transient and persistent sodium; fast, slow, muscarinic, and calcium-dependent potassium;

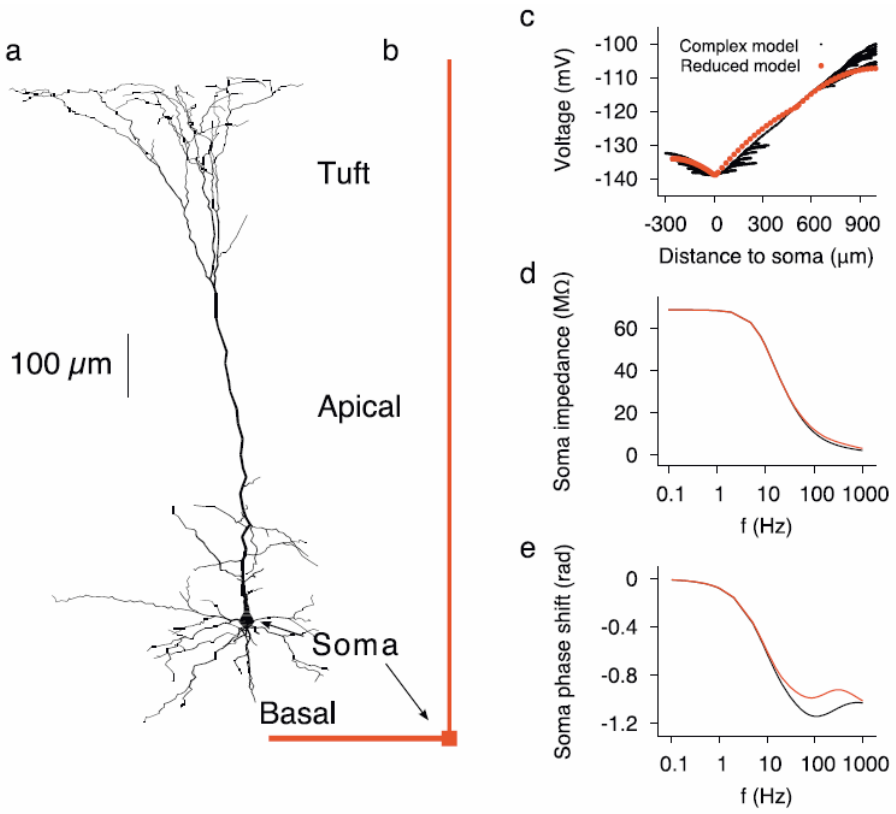

FIGURE 1.2: Morphology and passive electrophysiology for a detailed reconstruction of a layer 5 pyramidal neuron (a, Stuart and Spruston (1998)) and the reduced model (b, Bahl et al. (2012)). The remaining figures demonstrate that the reduced model reproduces the complex model's internal responses to various applied current injections, including (c) the steady-state voltage at different locations in response to a constant current injected at the soma, (d) the somatic impedance (a function of oscillations in cellular currents and potentials) in response to an oscillatory current injected at the soma, and (e) the somatic potential's phase shift in response to the same oscillator input. Reproduced from Bahl et al. (2012).

slow and pumped calcium), which were selected and distributed based on experimental findings and modeling studies. See (Bahl et al., 2012) for a more detailed description of geometric and electrophysiological matching with experimental data. The optimized reduced neuron model(s) are available for download on ModelDB.

1.3.3 Integration with the NEF

In order to successfully achieve integration with the NEF, bioneurons must meet two general requirements. First, the spiking activity of a population of bioneurons must encode information that can subsequently be decoded. More specifically, bioneurons should implement some form of tuning curve: when fed a vector-valued input, the bioneurons' resulting spike rate must vary over this vector space. Preferably, this encoding would be (a) spike-agnostic, such that identical state-space inputs encoded in distinct presynaptic spike trains would produce similar bioneuron responses; (b) continuous, such that nearby points in state-space would produce similar bioneuron responses; and (c) heterogeneous, such that presynaptic inputs would produce a different response in each bioneuron. If these criteria are met, the bioneurons' activites will be said to represent the information. Second, the tools used to decode the state-space information from the bioneuron activities must be able to compute arbitrary functions from these spikes. In the standard NEF, this is achieved by manipulating the decoders using a least-squares optimal method, but these techniques may not be as straightforward for bioneurons. If such decoding is discovered, it will be possible to implement functional transformations and dynamical systems using the bioneurons.

A number of theoretical challenges for NEF incorporation are immediately obvious. One difficulty is that bioneurons built using `NEURON` may only take spikes as inputs. While this is clearly a biological necessity, neural communication in `nengo` typically utilizes a form of dimensionality reduction in which decoding, filtering, summing, and encoding happen in a particular sequence. As depicted in Figure 1.3 (top), `nengo` models first encode the **state-space** signal $\mathbf{x}(t)$ into a **neuron-space** signal $\sum_{i,t'} \delta_i(t - t')$, then decode a new state-space estimate $\hat{\mathbf{x}}(t)$ by filtering, weighting, and summing these spikes, and

finally feed this vector into the encoders of neurons in the postsynaptic population. Essentially, for every connection, a full encoding and decoding between state-space and neuron-space occurs, such that each postsynaptic neuron has access to the state-space estimate recovered from *all* the presynaptic activities at $t = t'$. In contrast, the synapse objects `ExpSyn` in `NEURON` models receive unweighted spikes, then perform weighting and filtering in a single step, and finally send the resulting spikes to other bioneurons without explicit decoding. Because each `ExpSyn` receives input from only a single presynaptic neuron, the state-space representation $\hat{\mathbf{x}}(t)$ is not directly available to the neurons. Essentially, bioneurons operate entirely in neuron-space, see Figure 1.3 (bottom).

This difference is crucial because it is no longer possible to calculate firing rate from state-space input using a formula like Equation 1.15 that utilizes a time-invariant $\mathbf{x}$: instead, firing rate depends on the relative arrival times of input spikes, which change with the properties of the presynaptic neurons (i.e., gain, bias, seed, etc.). As mentioned above, `nengo` calculates decoders using `solvers` that take as input a sample of state-space values and corresponding firing rates from the neural population. For LIF neurons, these samples are static **evaluation points** that tile the N-dimensional state-space, and the firing rates are analytically calculated at each evaluation point using Equation 1.15. In essence, this method uses the fixed tuning curves of a LIF population to calculate the decoders before the simulation proper begins. With bioneurons, the notion of state-space values and instantaneous firing rate are more subtle due to adaptation and spiking inputs. Instead of using fixed evaluation points, the state-space samples $\mathbf{x}_{target}(t)$ can only be realized in comparison to a time-varying input signal $\mathbf{u}(t)$. Similarly, the instantaneous firing rate must be approximated by feeding the input (which has somehow been converted to spikes) to the bioneurons then collecting and smoothing their spikes. However, to collect targets and bioneuron spikes, the network must be simulated for a

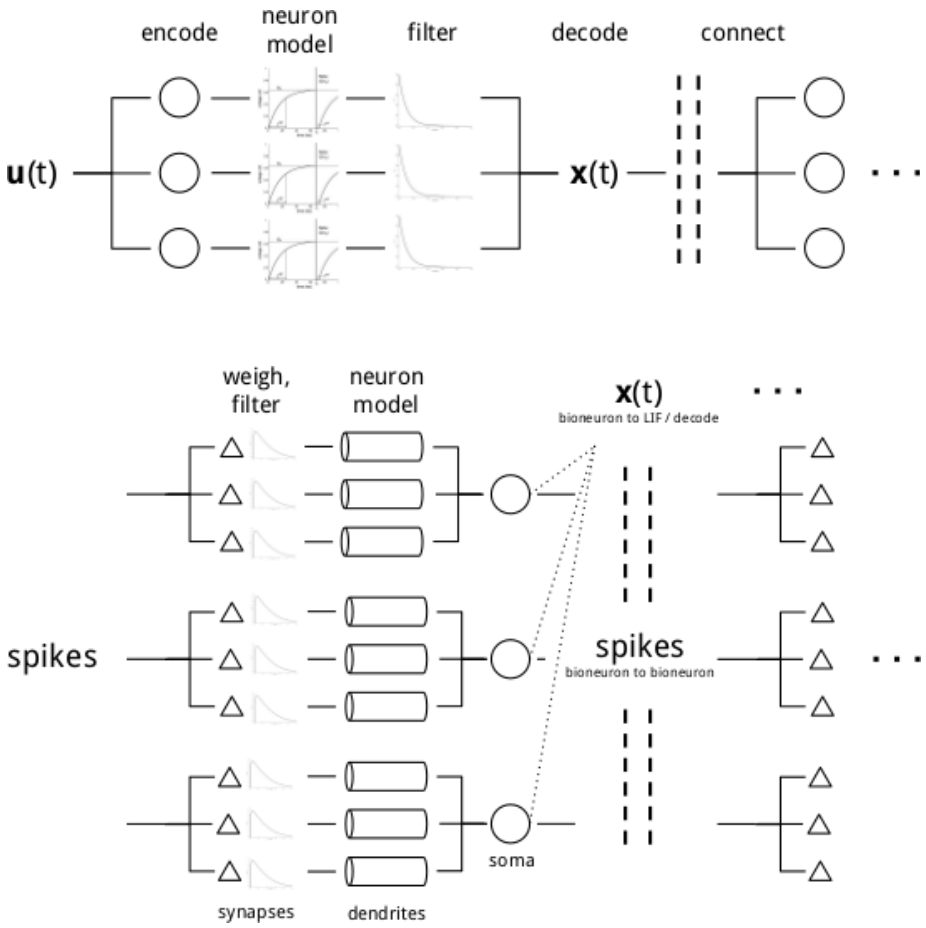

FIGURE 1.3: Order of operations for encoding, filtering, and decoding in nengo (top) vs NEURON (bottom) networks. Each circle represents an individual neuron or bioneuron soma and the dashed vertical lines represent a connection between populations. **Top**: the equations governing the nengo progression are Equations 1.1 (encoding), 1.15 (LIF neuron model), 1.3 (filtering), and 1.5 (decoding). Note that all neurons in this scheme receive state-space inputs $\mathbf{u}(t)$, and that the state-space estimate $\hat{\mathbf{x}}(t)$ is decoded from the neuron space before projection to the following neural population. **Bottom**: the equations governing the NEURON progression include a conductance-based exponential synapse model that simultaneously weights and filters input spikes (Equations 3.1–3.2), a multi-channel, multi-compartment neuron model that induces dendritic filtering, and adaptive somatic dynamics. Note that bioneurons only send and receive spikes, and hence operate entirely in the neuron space: Equations 1.3 and 1.5 are only used to decode the state-space estimate when projecting to LIF populations or reading out the state.

period of time before the simulation proper begins. To ensure good representation, this period must be long enough that (a) $\mathbf{u}(t)$ covers the entire state space, (b) the smoothed bioneuron activity begins to coalesce for each region of the state space, even after accounting for (c) adaptation and nonlinear dendritic filtering. The following chapters explore a number of techniques for overcoming these challenges, then examine the fidelity of bioneurons' representation and dynamics in canonical NEF networks.

Chapter 2

Neurobiology, Working Memory, and the NEF

2.1 Introduction

This chapter applies standard NEF techniques to a create a spiking neural network model of working memory (WM) and action selection applied to a mnemonic cognitive test, the spatial delayed response task (DRT). In doing so, it demonstrates the NEF's ability to capture a broad set of low- and high-level features: the model approximates enough neural detail to describe the underlying causes of mental disorders and typical treatments, enough electrical detail to respect biological constraints and produce data that can be externally validated, and enough functional detail to provide a conceptual description of WM systems and their disorder-induced deficits. After describing this model and its results, the integration of biologically detailed neurons into the NEF is reconsidered. In addition to familiarizing the reader with NEF models, this chapter thus lays the groundwork for Chapter 5, which incorporates the bioneurons developed in Chapters 3 and 4 into the model and compares its performance with the aforementioned approximations.

2.2 Working Memory

Working memory (WM) is a central component of cognitive systems that is required for temporary information storage during the execution of complex tasks. It is an ideal example system for this thesis because it draws upon several brain areas, coordinates complex cognitive functions, and has been the subject of numerous quantitative studies at multiple levels of analysis. Furthermore, WM is impaired by a variety of mental disorders including Attention Deficit Hyperactivity Disorder (ADHD) (Scahill et al., 2014), raising questions about how the system's neurobiological processes are disrupted by WM disorders (Avery et al., 2000) and how they can be treated through pharmacological interventions acting on the brain's neurobiological substrate.

2.2.1 Previous Models

Although computational models are well-suited to the task of simulating WM, existing models rarely provide both biological detail and a functional architecture capable of generating behavioral predictions. For example, models such as CoJACK (Dancy et al., 2015) and Gunzelmann et al. (2009) are concerned with how high-level cognitive abilities like mental arithmetic, perception, and tactical planning relate to low-level details like caffeine or sleep loss, but must implement these low-level details through the models' symbolic plans and production rules rather than through neural perturbations. ACT-R/ϕ (Ritter et al., 2012) also investigates low-level details (e.g., epinephrine levels) using a mathematical model of physiology, but does not yet simulate neurons explicitly. On the other hand, the Human Brain Project (Markram et al., 2015) simulates cortical microcircuits with unprecedented biological accuracy, but lacks a theoretical framework that relates model activity to high-level cognitive abilities like perception, decision-making, and WM. New theories and models

are needed to unify these approaches and characterize the complex relationships between the pharmacological, neurobiological, and cognitive aspects of WM.

2.2.2 Neurology

WM is at least partly realized in the prefrontal cortex (PFC), a brain region whose prominent size in primates suggests its importance in complex cognitive tasks that require a flexible mental workspace. The PFC represents information that is temporarily held in mind, used to guide behavior and decision-making, and maintained through recurrent excitatory connections between neurons with similar tuning properties (Goldman-Rakic, 1995). Computationally, this recurrence realizes an extended temporal integration that preserves the represented item without external stimulation (Singh and Eliasmith, 2006). Therefore, the core requirement in a neural model of WM is that a population of neurons can maintain its state over time. That is, given a brief input, the internal connectivity should cause the neural activity pattern that results from that input to persist after the input has stopped. This persistence will not be perfect: over time the neural activity will drift away from its initial value.

2.2.3 Delayed Response Task

A standard behavioral test of working memory is called the spatial delayed response task (DRT). In this task, a monkey fixates on a point in the center of the screen, then is briefly presented a visual cue on the left or right (cue period, $1\,\mathrm{s}$). The cue is removed, then comes a delay period ($2\,\mathrm{s}$, $4\,\mathrm{s}$, $6\,\mathrm{s}$, or $8\,\mathrm{s}$), during which the monkey has to represent and maintain the cue's location in working memory. After the delay period, the monkey recalls the cue's location and responds by pressing a button on the left or right.

2.2.4 Disorders and Pharmacological Treatments

The stable representation of items stored in WM is sensitive to the synaptic connections of intra-PFC loops and the biochemical environment of PFC neurons. Impairments in the dopamine and norepinephrine system are closely associated with WM disorders such as ADHD (Arnsten and Lombroso, 2000; Chandler, Waterhouse, and Gao, 2014), and the drugs used to treat them target these impaired systems biophysically (Avery et al., 2000; Scahill et al., 2014). Specifically, drugs prescribed for ADHD affect PFC neurons that express Hyperpolarization-activated Cyclic Nucleotide-gated (HCN) ion channels (Franowicz et al., 2002). HCN channels are located on neurons' dendritic spines and are open at rest, shunting synaptic input by permitting nonspecific cations to flow out of the cell, as shown in Figure 2.1. These channels control the excitability of pyramidal neurons by modulating dendritic summation and the cells' resting potentials (Magee, 1999; Poolos, Migliore, and Johnston, 2002); when the neuromodulator norepinephrine binds to the α2A-adrenoreceptor (α2A-AR), it activates a cAMP-mediated intracellular signalling cascade that ultimately closes HCN channels. The result is reduced shunting and increased excitability of the neuron.

The drugs guanfacine (GFC) and phenylephrine (PHE) are an agonist and an antagonist of the α2A-AR respectively; GFC is prescribed to alleviate WM deficits in patients with ADHD (Scahill et al., 2014) while PHE reproduces many of the disorder's symptoms (Levy, 2008; Arnsten and Leslie, 1991). A study by Wang et al., 2007 showed that GFC increased (and a compound similar to PHE decreased) the firing rate of PFC neurons with weak mnemonic tuning in the direction of the cue presented in the DRT, while having no effect on cells tuned in the opposite direction, Figure 2.2 (top). These results are consistent with monkeys' increased (decreased) performance on DRT when injected with GFC (or PHE) (Mao, Arnsten, and Li, 1999; Ramos et al., 2006),

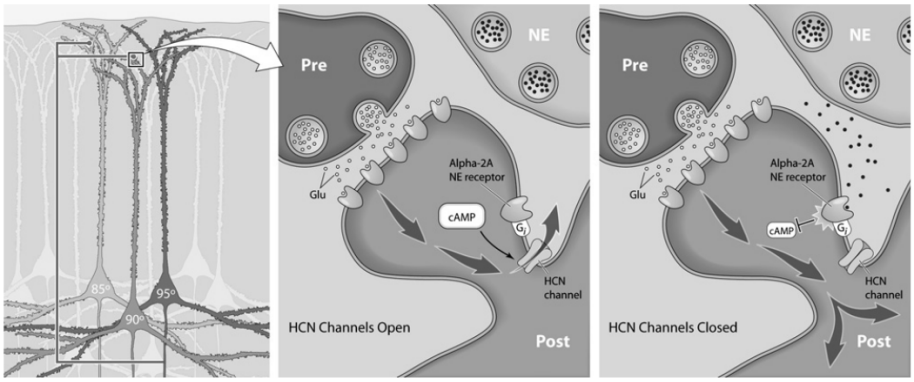

FIGURE 2.1: Diagram of GFC's biophysical interactions with PFC neurons. Excitatory post-synaptic currents induced by presynaptic glutamate release are shunted from dendritic spines via open HCN channels, leading to minimal postsynaptic potentiation. When norepinephrine or its agonist GFC binds the α2A-AR, HCN channels close, increasing the efficacy of cortical inputs. Image reproduced from Wang et al. (2007).

Figure 2.2 (bottom). A reasonable hypothesis is that GFC raises the firing rate of neurons with cue-aligned encoders, slowing the decay of information stored in the PFC neural integrator and increasing performance on the DRT.

2.3 Model

This model, reported previously in Duggins et al. (2017), extends existing NEF models of WM to perform the DRT using the architecture shown in Figure 2.3. The location of the DRT cue, $\mathbf{x}(t) \in \{-1, +1\}$ for left and right respectively, is fed into the network and represented by a population of neurons labeled `input` (100 neurons). Each neuron i has a preferred direction, in this case the cue location $\mathbf{e}_i \in (-1, 1)$, for which it fires most strongly. When driven by $\mathbf{x}(t)$, each neuron's firing rate $a_i(t)$ represents the cue's location; these activities can be decoded with the least-squares optimal decoders $\mathbf{d}_i$ to recover an estimate of the cue's location, $\hat{\mathbf{x}}(t)$. `input` connects feedforward to a working memory population labeled `WM` (100 neurons) that represents the same information. However, the recurrent connections within `WM` must dynamically stabilize this representation by maintaining the neural activities over time, such that when

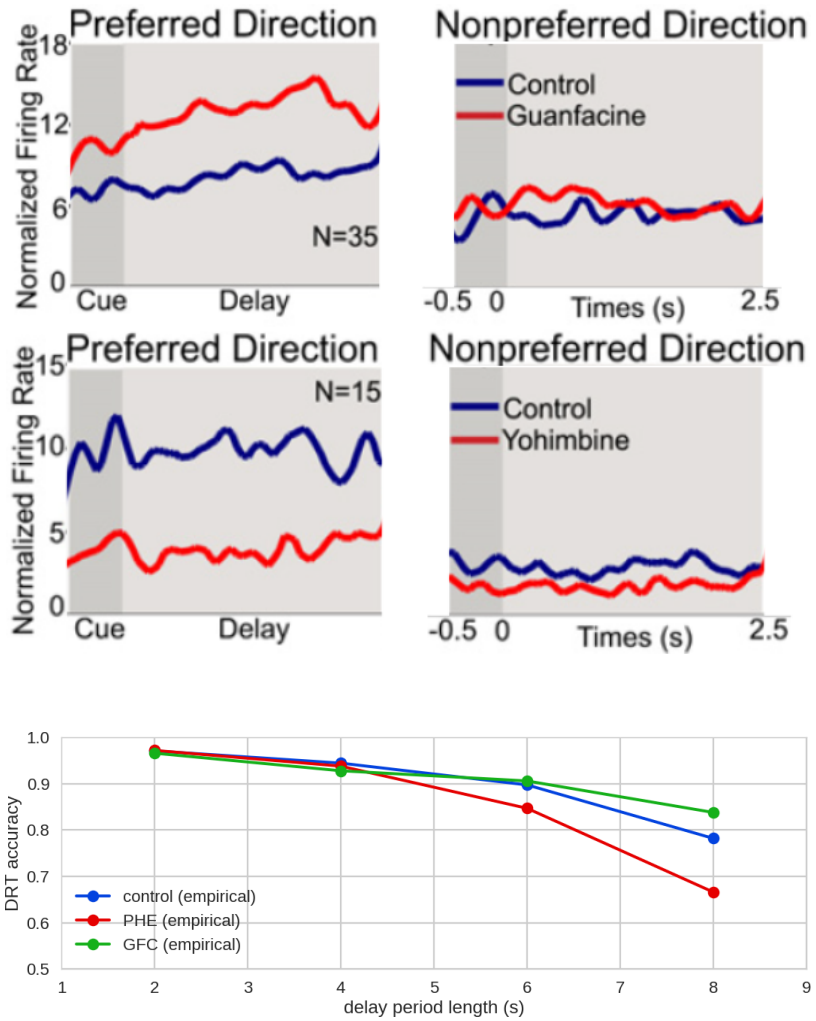

FIGURE 2.2: **Top**: Guanfacine increases (Yohimbine decreases) the activity of neurons that are spatially tuned to the cue's location in the DRT, but have a negligible effect on nonpreferred direction neurons. Data were obtained from neurons in area 46 of dorsolateral PFC that displayed spatial tuning during control conditions. GFC and Yohimbine were applied iontophoretically to the monkeys during DRT performance (Wang et al., 2007). Yohimbine and PHE are both antagonists for the α2A-AR, so their neural and behavioral effects are expected to be similar. **Bottom**: DRT accuracy vs. delay period length in monkeys injected with saline, GFC, and PHE (Mao, Arnsten, and Li, 1999). The blue line represents the baseline forgetting curve, a measure of how quickly the monkey forgets the cue's location. The outlier datapoint, GFC at $t = 4\,\mathrm{s}$, probably arises from the small samplesize of the dataset: a single (unique) monkey was used for each experimental condition, though each line represents 800 to 1200 DRT trials from that animal. Error in the original data were negligible so are not plotted here.

the cue input $\mathbf{x}(t)$ is removed, the WM representation remains. Using the dynamics principle of the NEF, this neural integrator is build using $A' = (\mathbb{1} + \tau)$ and $B' = \tau$. The result is a population of spiking neurons that maintains its activity over time, and has been the basis of multiple WM models (Singh and Eliasmith, 2006; Choo and Eliasmith, 2010).

To simulate a "forgetful" WM population, a second dimension is incorporated into the input vector $\mathbf{x}(t)$: the first dimension $\mathbf{x}_0(t)$ is the value (cue location) to be remembered, and the second dimension $\mathbf{x}_1(t)$ is the amount of time it has been remembered for. Empirical and modeling evidence are consistent with the claim that PFC neurons explicitly encode the passage of time (Lewis and Miall, 2006; Bekolay, Laubach, and Eliasmith, 2014). For example, some PFC neurons start firing only after a given amount of time has passed, while others gradually decrease their firing rate over time (Romo et al., 1999). These positive monotonic and negative monotonic neurons are sensitive to both the value being represented and the amount of time the memory has been held; in other words, these are **spatial mnemonic** neurons with large values in the first and second dimensions of $\mathbf{e}_i$. Other neurons may only be sensitive to one or the other dimension (i.e., have small $\mathbf{e}_i$ values for one of those two dimensions). This variability in $\mathbf{e}_i$ matches well to the observed variability in WM tuning curves (Singh and Eliasmith, 2006).

During the simulation, the cue's location is fed as a stimulus through `input` into `WM` for the duration of the cue period ($1\,\mathrm{s}$) then is removed; after this, the memory must be maintained by feedback activity. Two sources of *instability* are introduced to simulate forgetting during the delay period. First, external noise approximating the stochastic variability found in the brain is injected into `WM` neurons using a bias current. Second, the constant passage of time, encoded as $\mathbf{x}_1(t)$, steadily increases the firing rate of positive monotonic `WM` neurons until they saturate. Once a significant portion of the neurons saturate, decoding the cue value from the population's activity becomes noisy and inaccurate.

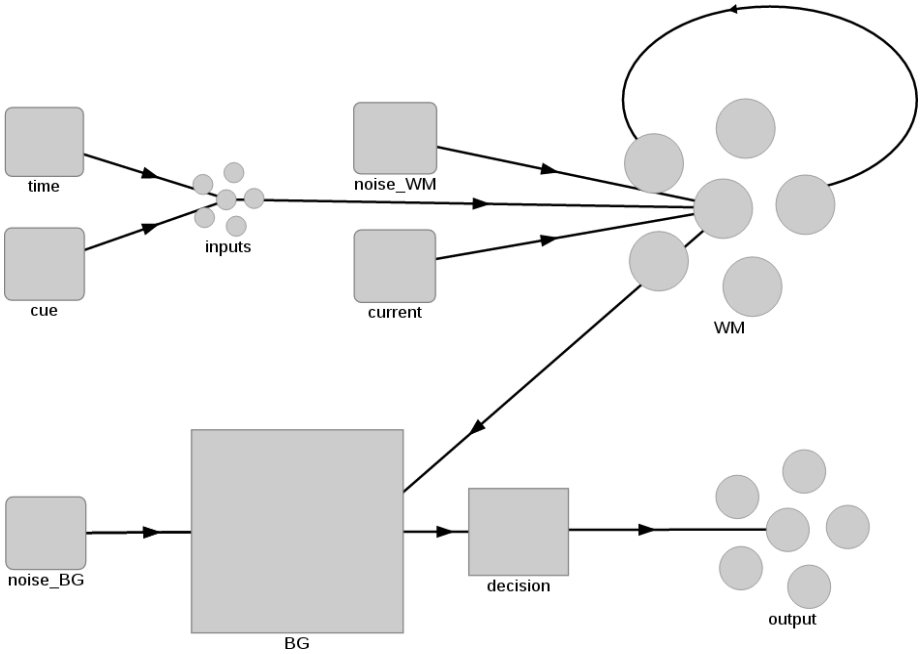

FIGURE 2.3: Schematic of the spiking neuron model of the delayed response task. Circles represent neural populations, rounded boxes represent nodes that output vectors, and square boxes represent subnetworks containing several connected neural populations.

Without these instabilities, the information stored in `WM` is stable for a very long time (minutes to hours), but when they are present, the information decays over tens of seconds, consistent with decay rates of human WM (Choo and Eliasmith, 2010).

To produce a response, the model must access the value stored in `WM` and produce one of two $outputs \in \{-1, +1\}$. A basal ganglia subnetwork `BG` decodes the cue information from the neural activity of `WM` neurons, then decides whether the result is most similar to -1 or 1. This basal ganglia model (Stewart, Choo, and Eliasmith, 2010; Stewart and Eliasmith, 2011) has previously been used to simulate several cognitive tasks that require reading information stored in working memory, such as action selection and procedure following. Furthermore, the basal ganglia's structure, function, and parameters are biologically plausible. When the value stored in `WM` become indistinguishable from 0 to this noisy decision procedure, the model randomly outputs -1 or 1, leading to low DRT accuracy.

There are four important free parameters in this model: k_{time} controls the rate of interference due to elapsed time; $noise_{WM}$ interferes with the cue representation and $noise_{BG}$ controls the accuracy of the decision procedure; and the $prob_{mis}$ gives the likelihood that the model fails to perceive the cue in the first place ($\mathbf{x}_0(t) = 0$). The two noise parameters were fixed biologically-plausible values, $noise_{WM} = 0.005$ and $noise_{BG} = 0.025$, then remaining parameters were tuned until the baseline forgetting curve was approximated, $prob_{mis} = 0.05$ and $k_{time} = 0.4$. The fit reported in Duggins et al. (2017) is tighter due to a greater number of neurons and an alternative decision procedure; it is likely that additional fine-tuning of model parameters (noise, neuron tuning properties, etc.) could further improve the fit. However, because the goal of this chapter is to demonstrate that the drug perturbations affect a baseline forgetting curve in a qualitatively accurate (rather than quantitatively exact) manner, this fine-tuning is left as an exercise for future work.

2.4 Results

This section describes three drug simulations, each of which approximates the effects of GFC and PHE on the model at a different scale. These approximations perturb standard NEF quantities such as encoders, decoders, gains, and biases to emulate biological mechanisms that (until Chapter 5) are too low-level to simulate directly. At the highest level, it explores how the drugs functionally alter forgetting rates in `WM` by manipulating the recurrent connection weights. Next, it investigates how injecting a constant current into all `WM` neurons biases their resting states, changing their firing rates and the population's ability to maintain information. Finally, it examines the underlying causes of these firing rate changes by manipulating the LIF neurons' inherent properties (α_i and β_i) to approximate the effects of α2A-AR (in)activation.

2.4.1 Functional Simulation

To simulate the high-level, functional effects of GFC and PHE on working memory, the weights in the `WM` recurrent connection are multiplied by a constant k_f, with the expectation that $k_f > 1$ will increase feedback and promote remembering, whereas $k_f < 1$ will increase decay and promote forgetting. Under normal conditions, as the model forgets the original stimulus, the cue value of $\hat{\mathbf{x}}_0(t)$ decoded from `WM` decays exponentially. When the strength of the recurrent connection is increased ($k_f = 1.03$), a higher value of $\hat{\mathbf{x}}_0(t)$ is fed back as input to `WM`, increasing the firing rate of cue-aligned neurons and more strongly encoding the cue's location. As shown in Figure 2.4 (top), the cue representation rises and its decay lengthens compared to control. This makes it easier for the decision procedure to distinguish the decoded cue location from noise, which shifts the forgetting curve up, Figure 2.4 (bottom). Conversely, weakening the recurrent connection ($k_f = 0.985$) quickens the decay rate and shifts the forgetting curve down. The model's response qualitatively matches the forgetting curves of monkeys injected with these drugs (Mao, Arnsten, and Li, 1999). Reported results were averaged over 1000 model realizations with randomized cues, neuron properties, and noise.

2.4.2 Electrical simulation

Although the functional simulation is conceptually simple and produces a decent empirical match, it is unrealistic because GFC and PHE do not physically transform the synaptic connections between neurons. One hypothesis is that these drugs alter the firing rate of PFC neurons in a way that later manifests functionally as improved or impaired forgetting. This experiment introduces a global increase (decrease) in somatic current to all `WM` neurons through `current`: $I_{GFC} = 0.5$ and $I_{PHE} = -0.2$. Importantly, even though Wang et al., 2007 showed that, *in vivo*, an increase in activity was only observed for neurons

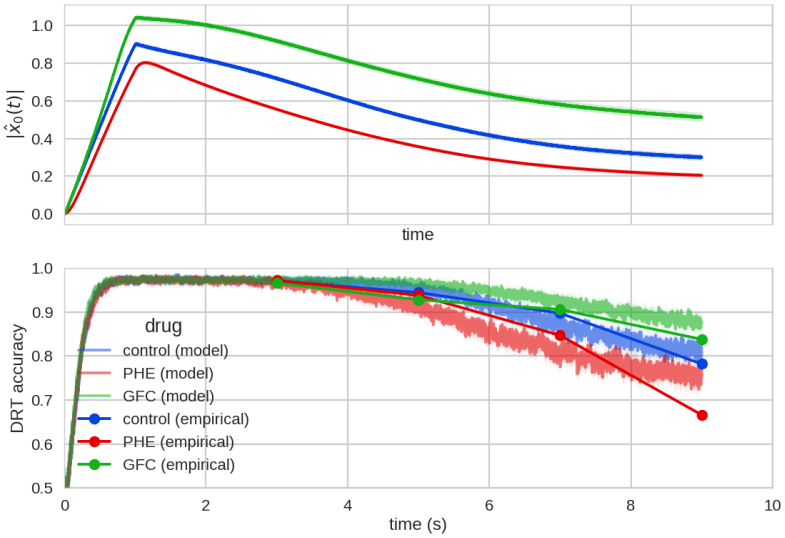

FIGURE 2.4: Empirical and model data for the functional drug simulation. **Top**: the cue location decoded from the spiking activity of WM neurons. **Bottom**: the forgetting curve, which describes the percentage of correct responses as a function of delay period length. Both for monkeys and the model, accuracy decreases steadily from $2\,\mathrm{s}$ to $6\,\mathrm{s}$ then drops sharply at $8\,\mathrm{s}$. Consistent with behavioral data from monkeys performing the DRT, applied GFC increases task accuracy while PHE decreases it. Gray regions represent $95\,\%$ confidence intervals.

whose preferred direction was aligned with the stimulus being remembered, this simulated drug effect is applied to *all* the neurons in `WM`, then afterwards the neurons with encoders within a certain range are observed. While this seems counter-intuitive, the network effects of the recurrent connections are sufficient to cause the differential response observed by Mao, Arnsten, and Li, 1999. In other words, GFC initially increases the firing rate of all neurons, but only cue-aligned neurons that are already firing (because they are representing the cue location) will feedback excitation to themselves and effectively multiply this increase in activity.

Figure 2.5 shows the normalized firing rate of neurons before and after the simulated application of GFC and PHE. As with the empirical data, the electrical drug simulation for GFC increases (PHE decreases) the firing rate of simulated preferred-direction neurons while having little effect on neurons in the nonpreferred direction. This differential activation of preferred direction neurons in turn allows the integrator to maintain a coherent representation of the cue's location for a longer duration, shifting the forgetting curve up, Figure 2.6. These electrophysiological and behavioral results are consistent both with the functional drug simulation and with empirical data, Figure 2.2.

2.4.3 Neural Simulation

The final experiment approximates GFC and PHE at the individual neuron level by altering the inherent properties of model neurons. At rest, HCN channels allow positive ions to flow into the cell, so closing HCN effectively induces a negative current, lowering the resting membrane potential. This effect is here modeled by lowering the bias current β_i of each LIF neuron in `WM`. Additionally, closing HCN channels modulates neurons' dendritic summation such that small, desynchronized dendritic spikes more strongly influence the somatic membrane potential. This effectively increases neurons' responses to

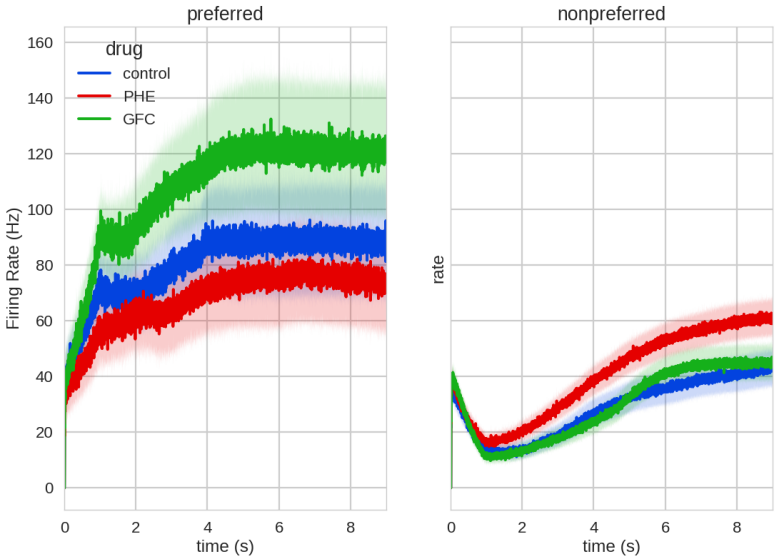

FIGURE 2.5: Firing rate of simulated neurons with encoders in the preferred vs. nonpreferred directions in response to injected current. The neurons chosen for plotting were tuned to the preferred direction during control conditions, as per their hypothesized importance in representing the cue's location during the delay period. Wang et al., 2007 failed to provide a precise definition of "weak spatial mnemonic tuning" or their procedure for choosing such neurons, so model neurons were selected based on the magnitude of their encoders ($0.3 < |\mathbf{e}_i| < 0.6$). They also did not discuss their method of calculating "normalized firing rate," so the WM neurons' properties were *not* fine-tuned to match the absolute rates reported in Figure 2.2

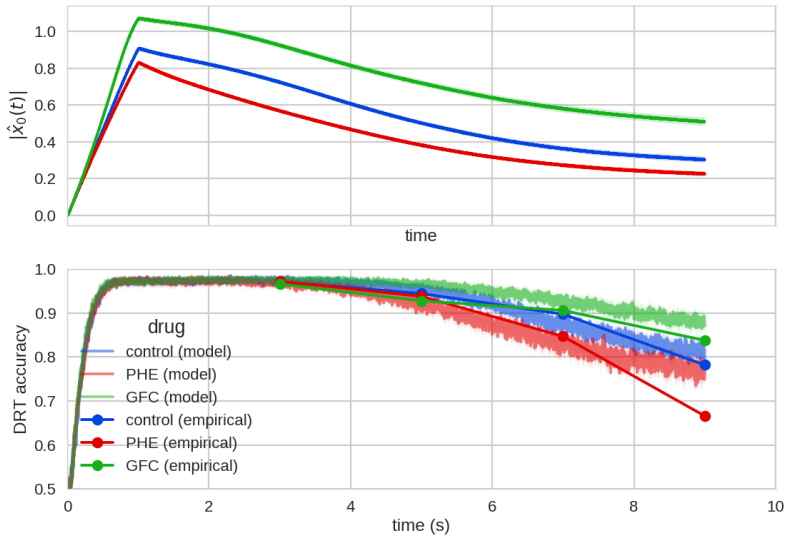

FIGURE 2.6: Cue representation and forgetting curve for the electrical drug simulation.

a given synaptic input, which is here modeled by increasing the gain α_i of each neuron. This approximation was validated using a study in mice (Nolan et al., 2004) that showed that closing HCN channels decreases neurons' resting membrane potentials and increases their gains in the subthreshold regime, Figure 2.7.

After initializing the neural model to perform least-squares optimal integration (i.e., distributing neurons' initial gains, biases, encoders, and decoders), the gains of all `WM` neurons are shifted by adding a constant $\delta_\alpha = 0.1$ for GFC ($\delta_\alpha = -0.1$ for PHE), and the biases of all WM neurons by $\delta_\beta = -0.025$ ($\delta_\beta = 0.025$). The impact of these perturbations on the equilibrium membrane potential and steady-state firing rate of an example LIF neuron is depicted in Figure 2.8: with a constant input current, increases (decreases) in gains overwhelm decreases (increases) in biases, causing GFC-affect neurons to exhibit hiring firing rates for all values in the state space. (Note that this validation was qualitative; reproducing the equilibrium state of a complex biological mouse neuron by fitting non-adapting LIF neuron with solely the gain and bias

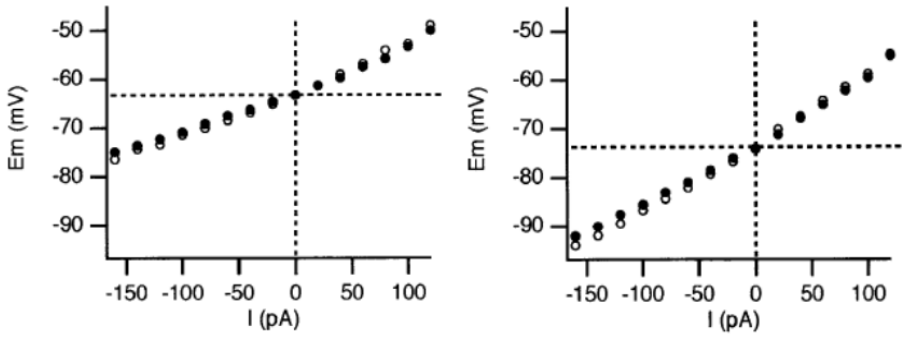

FIGURE 2.7: Subtheshold resting membrane potential as a function of applied current for normal mice (left) vs. mice with HCN channel genes artificially inactivated (right). Closing HCN channels lowers the neuron's resting potential (lower value of E_m at $I = 0$) while increasing the neuron's response to subsequent input (higher slope of E_m vs. I). Image reproduced from Nolan et al. (2004).

parameters is not feasible.)

In the context of recurrently connected WM neurons, simulated GFC should therefore increase neurons' overall activity, while simulated PHE should do the opposite. Figure 2.9 confirms that this neural simulation reproduces the empirical drug-induced change in PFC neurons' activities, though the effect is less pronounced than in Figure 2.5 or in Duggins et al. (2017). Again, this simulation is applied to all neurons in WM population, so the network effects from the recurrent connection are responsible for the differential response of preferred vs. nonpreferred direction neurons. The neural intervention also alters cue encodings in WM and shifts the forgetting curve in a manner consistent with the behavioral data and the previous drug simulations, Figure 2.10.

2.5 Discussion

This chapter presented a spiking neuron model of WM and the DRT, then used this model to investigate the underlying causes of WM disorders and their treatments through the simulated application of GFC and PHE. The model extends classical works on WM dynamics (Brunel and Wang, 2001) by

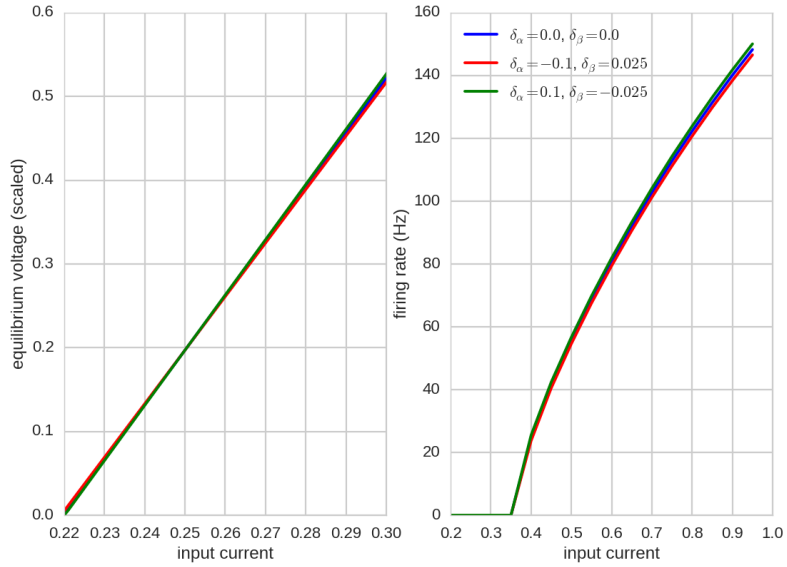

FIGURE 2.8: Subtheshold resting voltage (left) and steady-state firing rate (right) as a function applied current for a LIF neuron subjected to the neural perturbation. Simulated GFC (green) decreases the bias, shifting the equilibrium curve right, but also increases the gain, increasing the curve's slope. The result is a more depolarized resting state for currents greater than 0.25, which translates to greater firing rates with larger effective inputs. Simulated PHE (red) does the opposite.

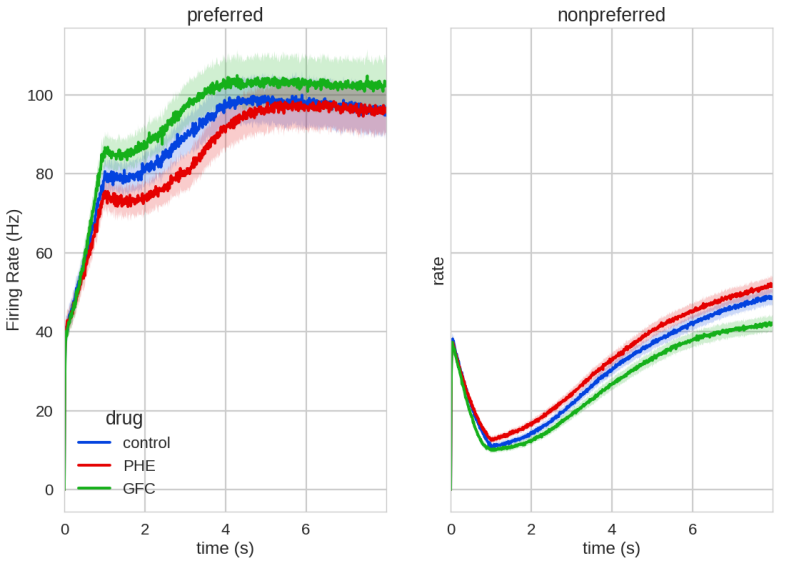

FIGURE 2.9: Firing rate of simulated neurons for the neural drug simulation.

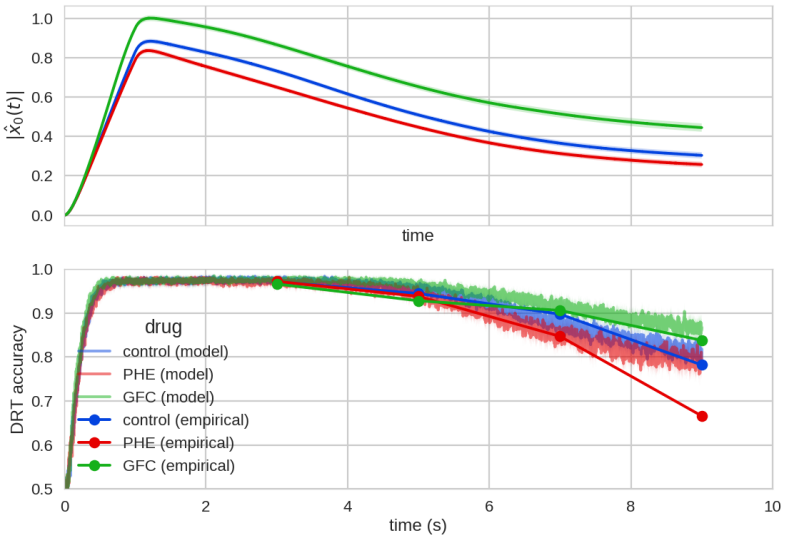

FIGURE 2.10: Cue representation and forgetting curve for the neural drug simulation.

incorporating the NEF, an approach that allows for (a) the principled encoding and decoding of information in large-scale spiking neural networks, and (b) the manipulation of these networks at levels ranging from the neural to the functional. The model further utilized the NEF to investigate interactions between WM and two drugs that reduce and enhance WM deficits in ADHD, showing that these interactions could be explained from a functional, electrical, or neural perspective. Three distinct drug simulations, each computationally realizing one of these perspectives by perturbing a different part of the model, all produced similar, and empirically accurate, effects on electrophysiology and task performance. This result unifies these seemingly-disparate descriptions of the drugs' interaction with WM systems.

However, the model and experiments relied heavily on the use of LIF neurons. Although LIF neurons are a widely accepted approximation of neural behavior across multiple contexts and brain areas, they limit the biological plausibility of the model and constrain the bottom-up investigation of biophysical disorders and pharmacological treatments. It is unclear whether the NEF

can engineer functional neural systems (e.g., compute recurrent connection weights that maintain WM representations) if the simulated neurons have electrophysiological adaptation or presynaptic connections spread across a dendritic tree. Furthermore, to simulate ADHD and GFC/PHE application, the model relied on assumptions about the relationship between HCN channels and LIF activities. This was possible largely because existing work had already classified the relationship between GFC application and firing rate (Wang et al., 2007). Though this chapter's approximations were successful in reproducing the empirical data, this approach cannot necessarily be generalized to simulate other bottom-up biophysical manipulations.

A more robust approach for incorporating low-level biological features into NEF models is to replace LIF neurons with biologically detailed neurons. As discussed in Chapter 1, simulated bioneurons include explicit ion channels that can be directly manipulated by drugs, expanding the range of biochemical processes that can be simulated without detailed foreknowledge. Chapters 3 and 4 present two methods of resolving the challenges associated with incorporating bioneurons into the NEF, then Chapter 5 applies these methods with the goal of incorporating bioneurons into the WM model explored above.

Chapter 3

Bioneurons with Spike-Match Training

3.1 Introduction

Recall that, in order to achieve neural representation using the NEF, there must be a well-defined relationship between the information fed into bioneurons and their resulting activities. The approach introduced in this chapter manipulates bioneurons' parameters until their responses align with predefined tuning curves. Although any parameters accessible through `NEURON` could in principle be altered to reach this end, many of the geometric and electrophysiological parameters are fixed by the Bahl et al. (2012) `NEURON` model, and changing them would compromise aspects of its biological realism.

Instead, this thesis trains the **synaptic weights** that connect spikes sent from presynaptic neurons to the bioneurons' dendrites. Synaptic weight refers to a collection of biophysical features that vary between neurons and over time, including the number of presynaptic neurotransmitter vesicles, the density of postsynaptic neurotransmitter receptors, the surface area of the postsynaptic dendrites, and more. Because these quantities are too low-level even for simulation in `NEURON`, they are lumped together into the abstract quantity "weight". Manipulating synaptic weights is a reasonable method for training bioneurons

because (a) synapses link incoming information directly to bioneurons' post-synaptic responses and indirectly to their firing rate, (b) connection weights have traditionally been used to compute functions in the NEF and deep neural networks, and (c) synaptic plasticity is considered one of the brain's primary learning mechanisms (Abbott and Nelson, 2000; Bekolay and Eliasmith, 2011).

What defines an "ideal" tuning curve, the desired target of synaptic weight training? There are two potential criteria. The first criteria is functional: bioneurons must be able to dynamically represent and transform information. For a population of bioneurons to accurately represent an input, each bioneuron's firing rate should vary discernibly over the state-space, and the collection of all bioneurons' tuning curves should cover the state-space evenly (with sufficient redundancy for the accurate representation of the encoded vector). The second criteria is electrophysiological: the voltage and spike dynamics exhibited by bioneurons should agree with recordings taken from biological neurons. The vast diversity of spiking behaviors across neuron types means that many potential electrophysiological targets exist; choosing among them will depend on the goals of the model and the morphology and electrophysiology of neurons in the brain area being modeled.

The methods in this chapter compromise functional and electrophysiological criteria by choosing the activity of a spiking LIF population as the target behavior. LIFs are functionally desirable because their tuning curves (a) can be computed analytically and adjusted with a few theoretically-motivated parameters and (b) have successfully represented information in numerous NEF models. LIFs are also electrophysiologically desirable for the reasons highlighted in Chapter 1: they approximate the firing rate of neurons found in various brain areas, are the limiting case of more complex neuron models, and capture spiking, the critical neuron nonlinearity. LIFs are widely used in computational neuroscience and represent a solid baseline from which more complex behaviors, such as bursting and adaptation, can be added as

extensions. However, it should be stressed that the methods introduced in this chapter can be applied to any electrophysiological target $a_{ideal}(t)$, be it another neuron model (e.g., Izhikevich neurons (Izhikevich, 2003)) or spike trains measured from biological cells using standard neuroscientific techniques (e.g., single-cell recordings under specific experimental conditions).

3.2 Methods

The bioneurons' spike-match training proceeds as follows. First, the `NEURON` objects are initialized. The cell geometry is created according to the template provided by Bahl et al. (2012), and voltage probes are placed in the somatic compartment to record the initiation of action potentials (defined as the voltage rising above $V_{thr} = -20\,\mathrm{mV}$). Connections from the presynaptic neuron to the bioneuron utilize `NEURON`'s conductance-based exponential synapse `ExpSyn` with event-based spike delivery. The current flowing into the postsynaptic neuron $I_{in}^{syn}(t)$ from a single synapse, or postsynaptic current (**PSC**) is given by

$$I_{in}^{syn}(t) = g_{syn}(t)(V_{syn}(t) - E_{syn}) \tag{3.1}$$

where $V_{syn}(t)$ is the membrane voltage at the synapse's location and E_{syn} is the synapse's reversal potential (E_{syn}=$-80\,\mathrm{mV}$ for inhibitory synapses and E_{syn}=$0\,\mathrm{mV}$ for excitatory synapses). $g_{syn}(t)$ is a time-varying synaptic conductance the represents the opening and closing of ligand-gated ion channels (those that respond when a neurotransmitter binds to a receptor on the postsynaptic dendrite). When an `ExpSyn` receives a spike, g_{syn} instantly rises by the (trained) synaptic weight w_{syn}, then decays according to the exponential

$$\dot{g_{syn}}(t) = -\frac{g_{syn}(t)}{\tau_{syn}} \tag{3.2}$$

where τ_{syn} is the synaptic time constant, which represents the rate at which neurotransmitters undergo reuptake (unbinding from receptors and reabsorbtion into the cell) and thereby return the ligand-gated channels to rest ($g_{syn}(t) \Rightarrow 0$). The bioneuron's `ExpSyn` are distributed randomly along the length of the apical dendrite. Synaptic weights are randomly chosen in the range $-1 \times 10^{-3}\,\mu\text{S}$ to $1 \times 10^{-3}\,\mu\text{S}$ and rarely stray outside this range during training.

Figure 3.1 depicts the network used to train the bioneurons and calculate readout decoders. A state-space input signal $\mathbf{u}(t)$ originates in a `node` labeled `stim` and is fed into a `pre` ensemble consisting of 100 LIF neurons. $\mathbf{u}(t)$ may either be **sinusoid** with frequency $f_{sin} = 2\pi f$ and an amplitude of $amp = 1.0$, or an **white noise** signal, which has equal power at all frequencies below the cutoff frequency $f_{max} = 5\,\text{Hz}$ and an amplitude of $\text{RMS} = 0.5$. Both equal-power and prime-sinusoid have the advantage of spanning the state-space while incorporating a range of smooth dynamics.

The spikes produced by `pre` are then fed into the `ideal` population, which consists of LIF neurons whose individual properties, including gains α, biases β, and encoders e, determine their tuning curves. The spikes from `ideal` are recorded, then smoothed with a lowpass filter to produce the activities $a_{ideal}(t)$ corresponding to the state-space input $\mathbf{u}(t)$. This is the electrophysiological target. The `pre` spikes are also fed into the bioneuron ensemble `bio`, where they are transmitted to the weighted synapse objects of each cell. `NEURON` handles the internal dynamics of the bioneurons and records the spikes, which are passed back to `nengo` and smoothed using the same lowpass filter as was used for `ideal` spikes, resulting in $a_{bio}(t)$ corresponding to $\mathbf{u}(t)$. This is the electrophysiological result.

To train the synaptic weights such that $a_{bio}(t) \rightarrow a_{ideal}(t)$, several stochastic optimization algorithms were employed, including Hyperopt (Bergstra, Yamins, and Cox, 2013) and a "$1 + \lambda$ evolutionary strategy", with the root-mean-squared-error between $a_{bio}(t)$ and $a_{ideal}(t)$ as the loss metric. Both these

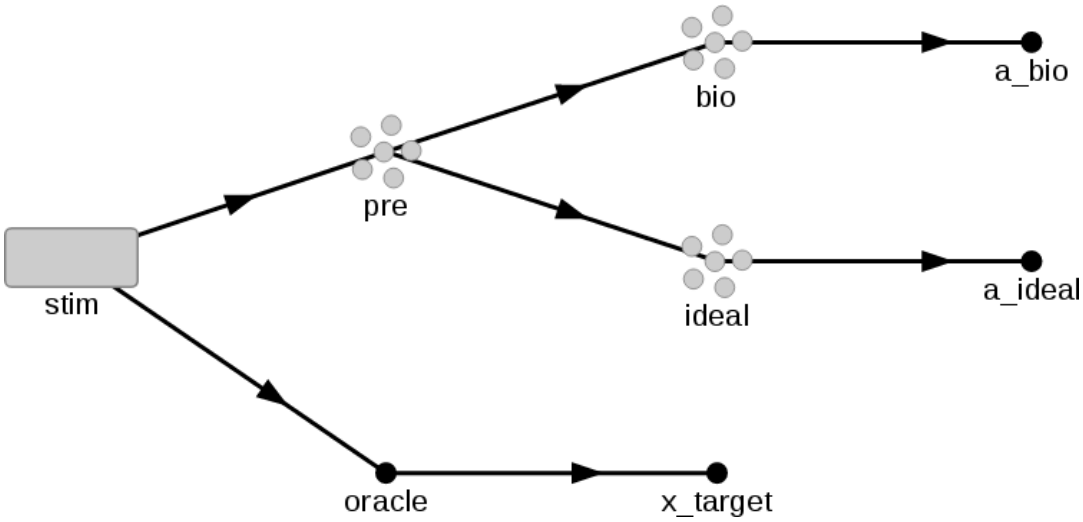

FIGURE 3.1: Schematic of the network used to train the bioneuron spikes $a_{bio}(t)$ to match the ideal LIF tuning curves $a_{ideal}(t)$.

strategies achieved reasonable spike matching within 100 evaluations for most experiments; reported below are the results from the evolutionary strategy.

Finally, $\mathbf{u}(t)$ is also fed into a special `node` referred to as `oracle`, which analytically computes the function specified on the connection between `pre` and `bio` (and between `pre` and `ideal`) and outputs the state-space target $\mathbf{x}_{target}(t)$. Using a standard `nengo` least-squares decoder, the readout decoders for `bio` are computed from $\mathbf{x}_{target}(t)$ and $a_{bio}(t)$. This method for computing decoders can be called the **oracle method**: it extends the NEF notion of static evaluation points and associated rate-mode-approximated firing rates to bioneurons while addressing many of the challenges introduced in Chapter 1. Notably, the oracle method accounts for spiking inputs, which take place in continuous time, and is more robust to strange electrophysiological dynamics like bursting and adaptation. Once decoders have been calculated for connections out of `bio`, the state-space representation of the bioneurons can be read out and assessed for accuracy, as well as transmitted onward to other LIF or bioneuron ensembles.

3.3 Results

This section applies the spike-match training described above to a population of 100 bioneurons and examines the resulting differences between $a_{bio}(t)$ and $a_{ideal}(t)$. It then explores the accuracy of the bioneurons' state-space representation in various networks where the connections into and out of `bio` compute the types of functions that are needed to construct the WM network presented in Chapter 2.

3.3.1 Attenuation and Nonlinearities

Before attempting to train synaptic weights, it is worth investigating how PSCs interact as they travel to the soma. Though Figure 1.2 and other figures in (Bahl et al., 2012) explore questions related to direct current injection, they do not address the interaction of PSCs induced by spikes at different points in the apical dendrite. Figure 3.2 shows the peak change in somatic voltage (from rest) induced by a single spike fed to a synapse at a distance d from the soma. This feedforward attenuation is normalized by division with $\Delta V(d = 0)$ and is well fit by a linear regression, demonstrating that the somatic voltage induced by a single PSC scales approximately linearly with synaptic distance from the soma.

However, this result is a misleading indicator of dendritic nonlinearities, which result primarily from the interaction of multiple PSCs induced by synapses at different locations and at different times. To demonstrate this, bioneurons are initialized with synaptic weights equal to the standard NEF connection weights, $w_{ij} = \mathbf{d}_j \cdot \mathbf{e}_i$. To account for the feedforward voltage attenuation reported in Figure 3.2, these weights are then scaled by location using the linear fit. Figure 3.3 shows the tuning curves of two bioneurons and their ideal LIF counterparts in response to a sinusoidal input. The slopes and intercepts of these curves are qualitatively similar, indicating the bioneurons'

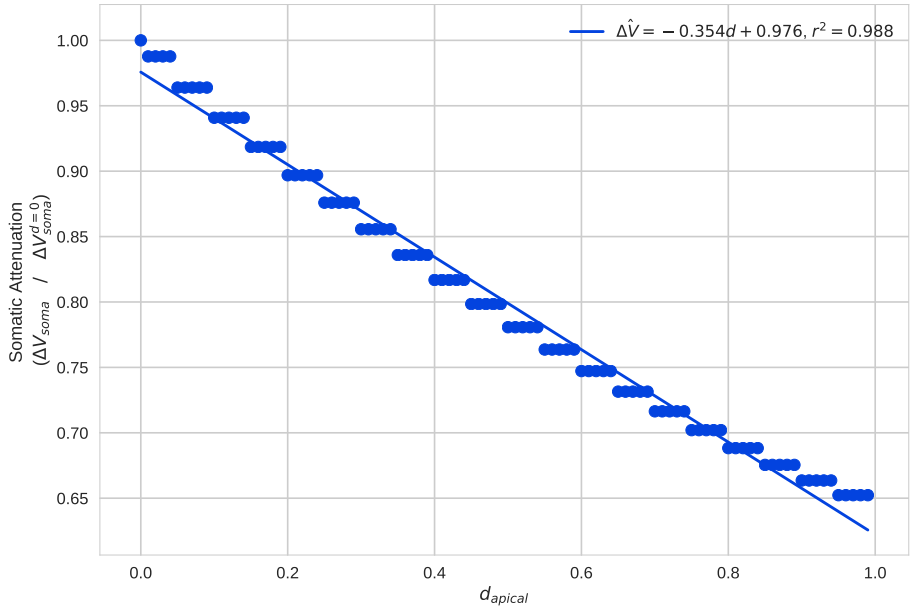

FIGURE 3.2: Somatic voltage as a function of synaptic location (distance from soma) when a single spike is fed into the synapse. Although the distance attenuation likely follows some exponential curve, a linear regression is a reasonable fit for this NEURON model, indicating linear encoding and decoding methods may be applied with minimal errors. The stepped nature of the simulated voltage ratios results from the apical dendrite being divided into $nseg = 20$ discrete compartments.

encoders, gains, and biases are working as expected, but the curves are quantitatively dissimilar, differing in firing rate by $100\,\text{Hz}$ or more for some regions of state space. Adding more synapses, additional input connections, or more dimensions exacerbates these differences. The mismatch between the ideal LIF and untrained bioneuron tuning curves supports the claim that nonlinear interaction of postsynaptic currents occurs in the dendrites, and motivates a training procedure that fine-tunes synaptic weights to give a more exact match to the ideal tuning curves.

3.3.2 Spike Matching

The simplest bioneuron network consists of a stimulus node `stim` that outputs a sinusoidal signal to `pre`, whose LIF neurons convert the signal into spikes. These spikes are transmitted to `ideal` and `bio`, whose activities are probed

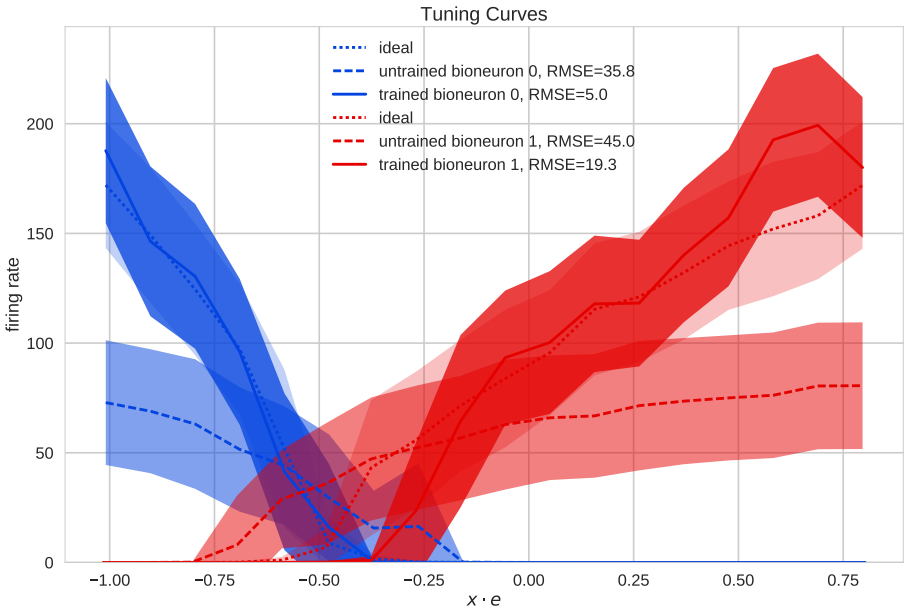

FIGURE 3.3: Comparison of the ideal LIF tuning curves to the untrained and trained bioneurons. Tuning curves are calculated from each population's time-varying inputs and activities: the one-dimensional state-space is divided into 20 bins, and the times when $\mathbf{u}(t)$ falls within each bin are recorded. The mean and standard deviation of the neural activities at each time within each bin are calculated, giving an indication of the range of activities associated with each stimulus value. Accuracy improves with number of synapses, training generations, and training time.

and compared. Figure 3.3 shows approximate tuning curves of two neurons from `ideal` and `bio` before training and after 20 generations of training using the $1 + \lambda$ evolutionary algorithm. A visual comparison confirms the significant drop in RMSE between $a_{bio}(t)$ and $a_{ideal}(t)$ as a result of training. This indicates the spike-match training is capable of tuning bioneurons' synaptic weights to reproduce simple electrophysiological targets. The training takes about $1\,\mathrm{h}$ on a standard workstation.

3.3.3 Representation

To assess the bioneurons' representational capacity, the synaptic weights between `pre` and `bio` are trained using the spike-match approach, then the represented state is estimated using readout decoders. The connection from

`stim` to `oracle` in Figure 3.1 is set to compute the identity function, which is often referred to as a feedforward **communication channel**. Figure 3.4 reports four estimates of the target signal calculated with different bioneuron/LIF activities and oracle/static decoders.

- $\hat{\mathbf{x}}_{bio,\ oracle}(t)$: $a_{bio}(t)$ decoded with $\mathbf{d}_{oracle}$. To prevent overfitting and ensure that the oracle decoders generalize to new signals, the synaptic weights are first trained as above, then the network is stimulated with a training signal, the activities and targets are collected, and the oracle decoders are computed. This extra step is required to perform real-time oracle decoding (i.e., to transmit $\hat{\mathbf{x}}_{bio}(t)$ to other populations during the test simulation), since $\mathbf{d}_{oracle}$ are only available after a simulation has completed. If the chosen training signal activates the appropriate dynamics and covers the state space, these decoders should perform almost as well as oracle decoders generated on the test signal.

- $\hat{\mathbf{x}}_{bio,\ static}(t)$: $a_{bio}(t)$ decoded with $\mathbf{d}_{ideal}$, the readout decoders from the `ideal` LIF population. "Static" refers to the static evaluation points used by the `solver` to calculate $\hat{\mathbf{x}}_{lif,\ static}(t)$, as in Equation 1.8. If the spike-match training perfectly reproduced the LIF tuning curves, this estimate would be equivalent to $\hat{\mathbf{x}}_{lif,\ static}(t)$; as $a_{bio}(t)$ diverges from $a_{lif}(t)$ (due to ineffective training), this decoding will become increasingly inaccurate.

- $\hat{\mathbf{x}}_{lif,\ oracle}(t)$: $a_{lif}(t)$ decoded with $\mathbf{d}_{oracle}$. The oracle decoders for the LIF population are calculated using the same procedure as for $\hat{\mathbf{x}}_{bio,\ oracle}(t)$, but with $a_{lif}(t)$ replacing $a_{bio}(t)$. These two oracle estimates should be compared to assess the representational capacity of bioneurons and LIFs when applying this novel decoding scheme.

- $\hat{\mathbf{x}}_{lif,\ static}(t)$: $a_{lif}(t)$ decoded with $\mathbf{d}_{ideal}$. This is the standard NEF decoding and default reference for representational accuracy in this thesis.

As expected, given imperfect spike-match training and the oracle's extra information about $\hat{\mathbf{x}}_{target}(t)$, the bioneuron oracle decoding $\hat{\mathbf{x}}_{bio,\ oracle}(t)$ is slightly more accurate than $\hat{x}_{bio,\ static}(t)$. In contrast $\hat{\mathbf{x}}_{lif,\ static}(t)$ and $\hat{\mathbf{x}}_{lif,\ oracle}(t)$ are equally accurate, indicating that the oracle method can decode novel signals about as effectively as standard NEF decoding methods. Most importantly, the bioneuron decodings have higher error than the LIF decodings, indicating that representation is more difficult in bioneurons than in LIF neurons. However, the bioneuron representations are still quite accurate, demonstrating that the spike-match training can effectively encode information in bioneuron spike trains and that various decoding schemes can be used to recover this information. Therefore, bioneurons can realize the NEF-style representation.

It is important to note that the bioneuron decodings have a systematic error that does not appear in the LIF decodings: $\hat{\mathbf{x}}_{bio,\ oracle}(t)$ and $\hat{\mathbf{x}}_{bio,\ static}(t)$ are **phase-shifted** left and fail to reach the peak magnitude of $\hat{\mathbf{x}}_{target}(t)$. The source of this error is related to the bioneurons' adaptive properties as is explored in Chapter 4.

3.3.4 Computing Linear and Nonlinear Functions

In standard NEF connections, functions and transformations are computed by multiplying decoders, altering the state-space representation that is fed into the postsynaptic population. This is problematic in bioneurons because these state-space scaling methods may translate nonlinearly to postsynaptic changes in bioneurons. To circumvent these issues, the desired function is applied to the connection from `pre` to `ideal` and from `stim` to `oracle`, such that $a_{ideal}(t)$ and $\mathbf{x}_{target}(t)$ account for the transformed signal, even though the spike train fed into `bio` remains unchanged. The same spike-training and oracle-decoding method are then applied to this network.

Figure 3.5 shows the decoded outputs of `bio` and `ideal` when a linear

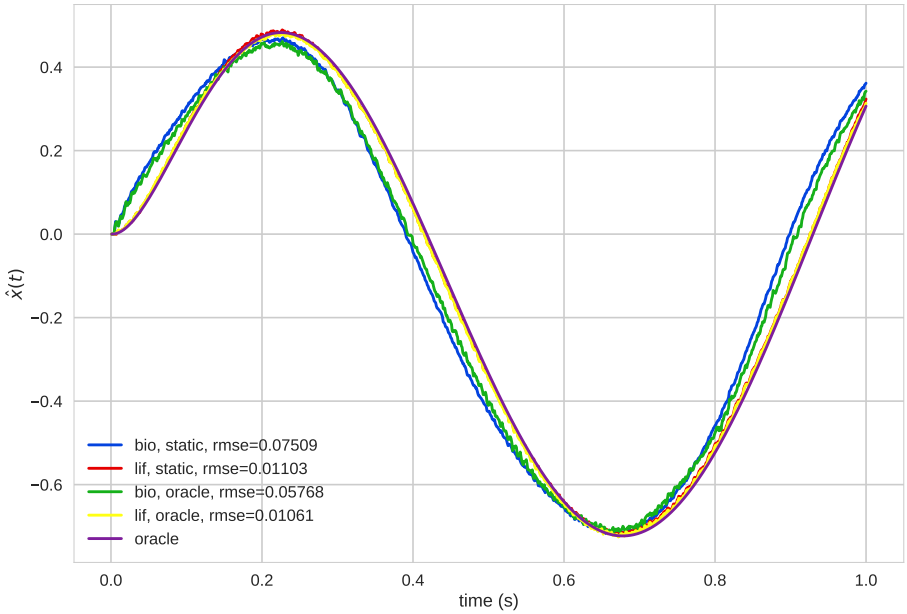

FIGURE 3.4: Comparison of the target signal $\hat{\mathbf{x}}_{target}(t)$ to estimates obtained from decoding bioneuron and LIF spikes using oracle and static decoders in a feedforward communication channel. The training and testing signals are identical for this preliminary plot: $\mathbf{u}_{train}(t) = \mathbf{u}_{test}(t) = \cos(2\pi t)$.

transform of $T = -0.5$ is applied to these connections, and Figure 3.6 shows the decoded outputs when a nonlinear function, the Legendre Polynomials of order 1 to 4, are applied to these connections. The resulting RMSEs for $\hat{\mathbf{x}}_{bio}(t)$ are comparable to those of the ideal ensemble for the linear transform and lower-order polynomials, but diverge for higher-order polynomials. This demonstrates that the spike-match training can effectively compute linear and simple nonlinear functions on connections, but that the technique fails to capture complex nonlinearities. What's more, training and testing on different signals exacerbates the systematic error mentioned above: the phase-shift and magnitude errors of the bioneuron estimates are more pronounced.

Multiple inputs presents another potential difficulty for bioneurons, as the PSCs induced from two presynaptic spike trains may add together nonlinearly. The following experiment adds `stim2` and `pre2` to the network shown in Figure 3.1, then connects both `pre` and `pre2` to `bio` and `ideal`. The connection weight matrices are trained sequentially. For comparison, `stim` and `stim2` are also

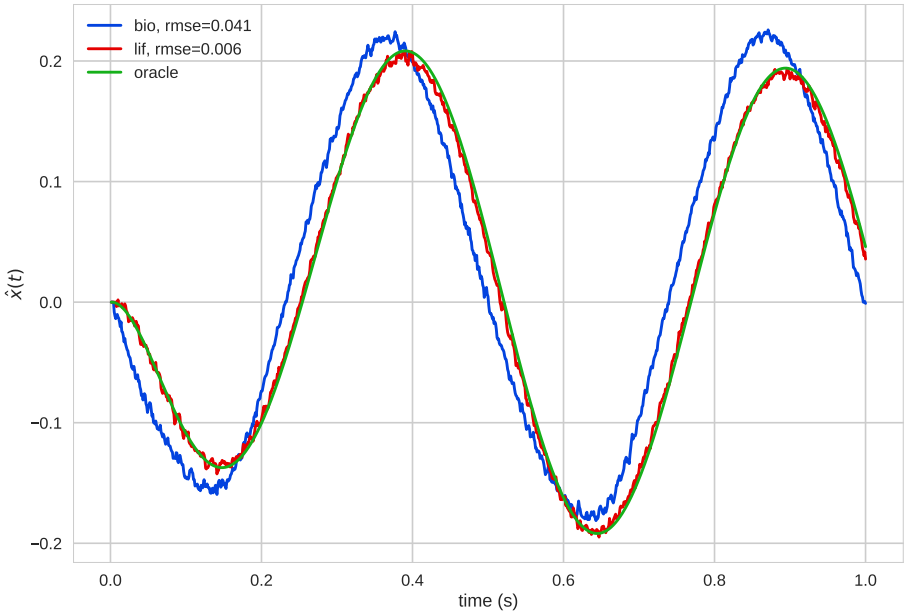

FIGURE 3.5: Comparison of the target signal, bioneuron estimate, and LIF estimate when computing a linear transform of $T = -0.5$. Here, and in future plots, the training and testing signals differ to ensure that the trained weights and decoders generalize: $\mathbf{u}_{train}(t) = \cos(2\pi t)$, $\mathbf{u}_{test}(t) = \cos(4\pi t)$.

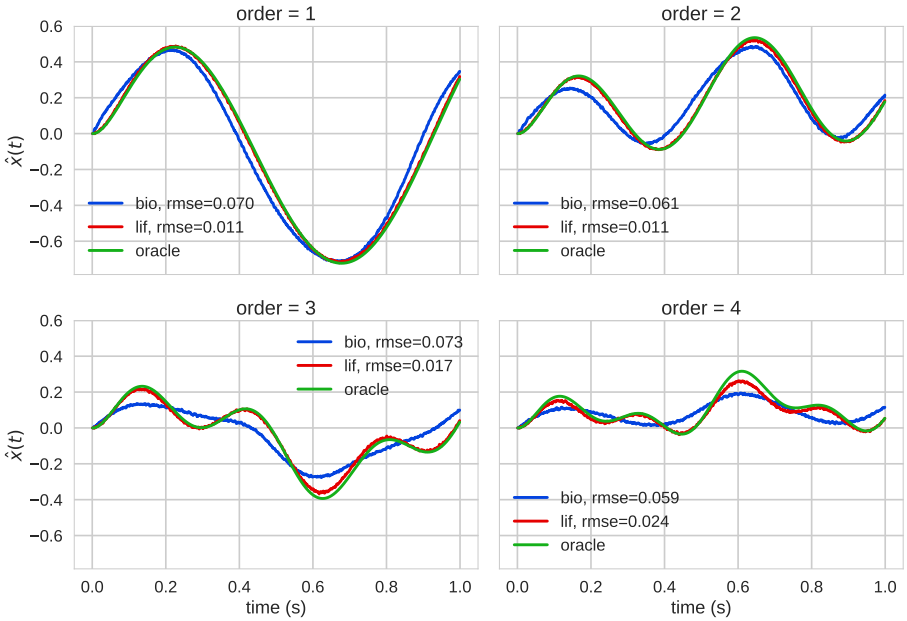

FIGURE 3.6: Comparison of the target signal, bioneuron estimate, and LIF estimate when computing a nonlinear function, the n^{th} order Legendre Polynomial. Given equal training time, RMSE increases with the nonlinearity of the computed function.

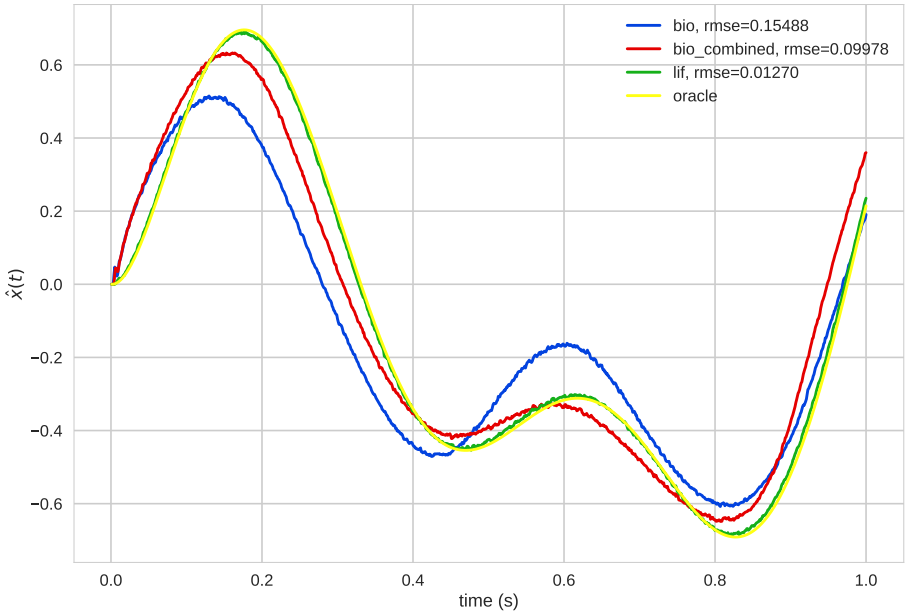

FIGURE 3.7: Comparison of the target signal, bioneuron estimate, and LIF estimate when two inputs, $\mathbf{u}_1(t) = \cos(2\pi t)$, $\mathbf{u}_2(t) = \cos(4\pi t)$, are fed into the network. State-space additivity is only partially preserved: feeding two spike trains into the bioneurons results in greater error than feeding in one spike train representing the combined inputs.

fed into `pre_combined`, which is fed into a separate `bio_combined`. Figure 3.7 shows that the trained weights implement feedforward communication, but that feeding two spike trains (which represent the individual inputs) into `bio` is not identical to feeding in a single spike train (which represents the combined inputs) into `bio_combined`. The greater phase and magnitude errors in the former case indicate that the spike-matching method only partially preserves state-space additivity.

3.3.5 Dynamical Systems and the Neural Integrator

To confirm that bioneuron-to-bioneuron connections work as expected, two bioneuron populations are connected in a feedforward communication channel. This preliminary experiment ensures that the spikes coming from one bioneuron population can viably be used to train a second bioneuron population before introducing the complex dynamics inherent in recurrent connections.

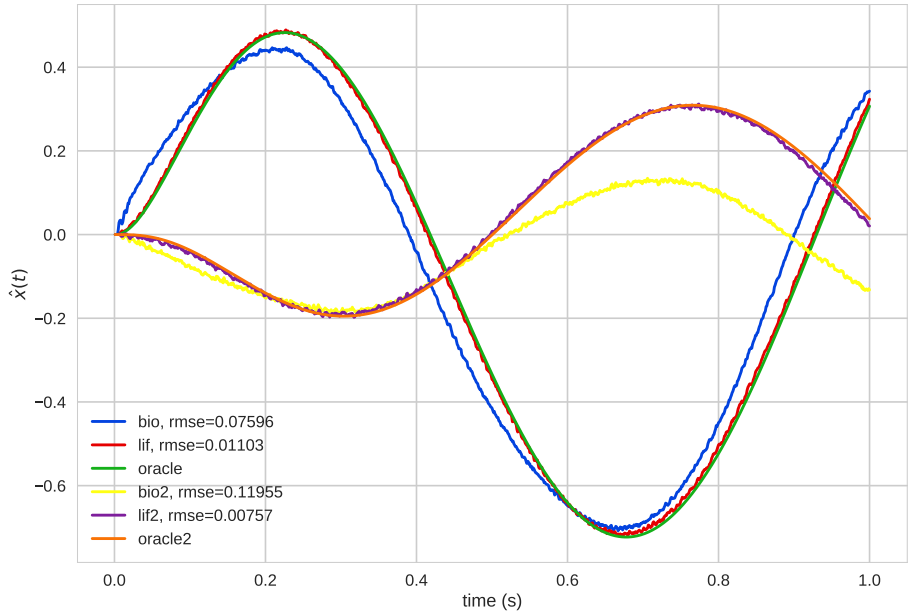

FIGURE 3.8: Comparison of the target signal, bioneuron estimate, and LIF estimate from the second population in a feedforward chain. A transform of $T = -0.5$ is applied to the connection between the first and second ensembles to ensure bioneuron-to-bioneuron connections can correctly compute functions.

As expected, Figure 3.8 shows that the spike matching approach can compute functions on bioneuron-to-bioneuron connections, though errors are more significant in the `bio2` decode.

The final network puts all these features together to construct an integrator, in which a recurrently connected `bio` accumulates feedforward stimulus from `pre` while maintaining its currently represented value through feedback connections. As shown in Figure 3.9, the bioneurons' decoded spikes approximate the integrator, but the representational error is significantly higher than for the LIF neurons, with the decoded signal drifting away from the ideal signal over time. Using the oracle method to compute readout decoders instead of using the ideal LIF decoders partially cancels this drift, reducing this error by a factor of 3, but leads to other inaccuracies (not shown). The recurrent training also requires a greater number of synapses and longer training times than the feedforward training to find suitable synaptic weights.

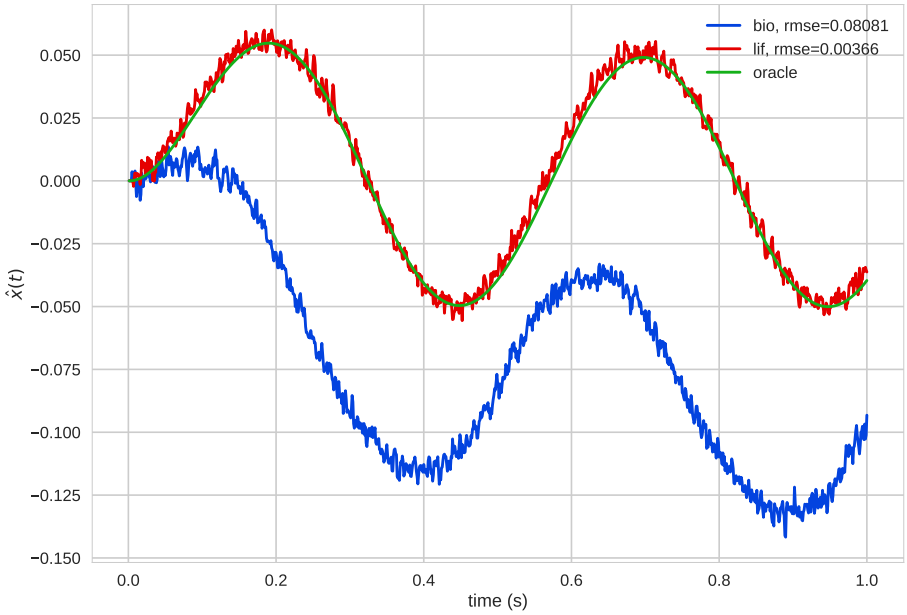

FIGURE 3.9: Comparison of the target signal, bioneuron estimate, and LIF estimate for a recurrently-connected integrator network. As in Chapter 2, the required transforms are $A' = \mathbb{1}$ and $B' = \tau$, but the dynamics introduced by the recurrent connection pose an additional difficulty for the spike-match training, leading to poor estimates. Using oracle decoders to estimate the bioneuron's representation reduces the error by a factor of 3 (not shown).

3.3.6 Scaling

The last two experiments assess how the bioneurons' representational accuracy scales with the number of neurons and the training time. In both experiments, RMSE between the neural decode and $\mathbf{x}_{target}(t)$ was calculated in a feedforward network. Figure 3.10 shows that RMSE falls with the number of generations in the $1+\lambda$ ES. Future work seeking to improve the spike-matching training algorithm should compare this training course with other stochastic optimization methods. Figure 3.11 shows that the RMSEs for the bioneuron and LIF estimates scale comparably with the number of neurons, supporting the claim that bioneurons can serve as an effective representational substrate in large-scale neuron models. However, for $n_{neurons} > 30$, the bioneuron decode consistently has an RMSE approximately 7 times larger than the LIF decode due to the phase shift.

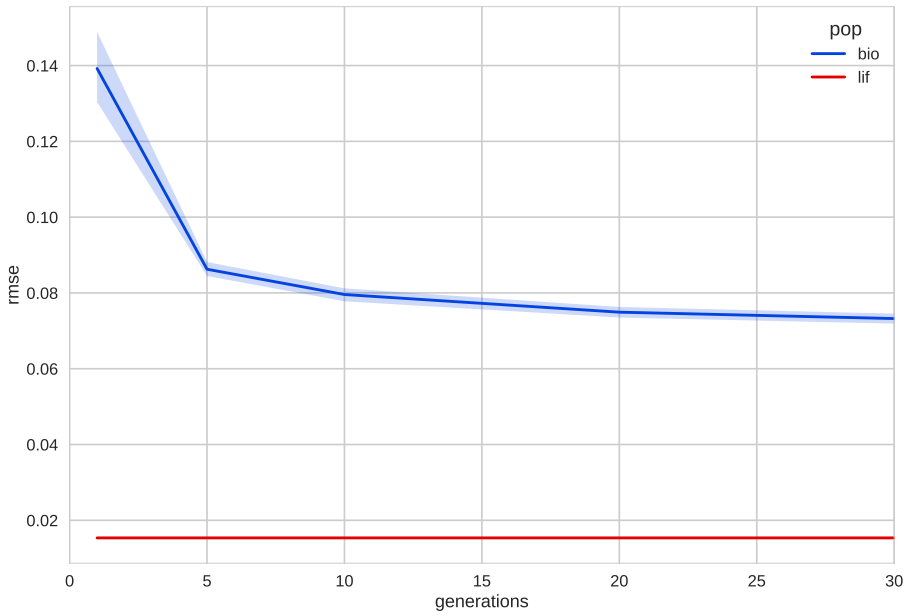

FIGURE 3.10: Root-mean-squared-error between the bioneuron representation and the target decoding as a function of training time (number of evolutionary generations). Shaded areas represent $95\,\%$ bootstrapped confidence intervals over 10 realizations.

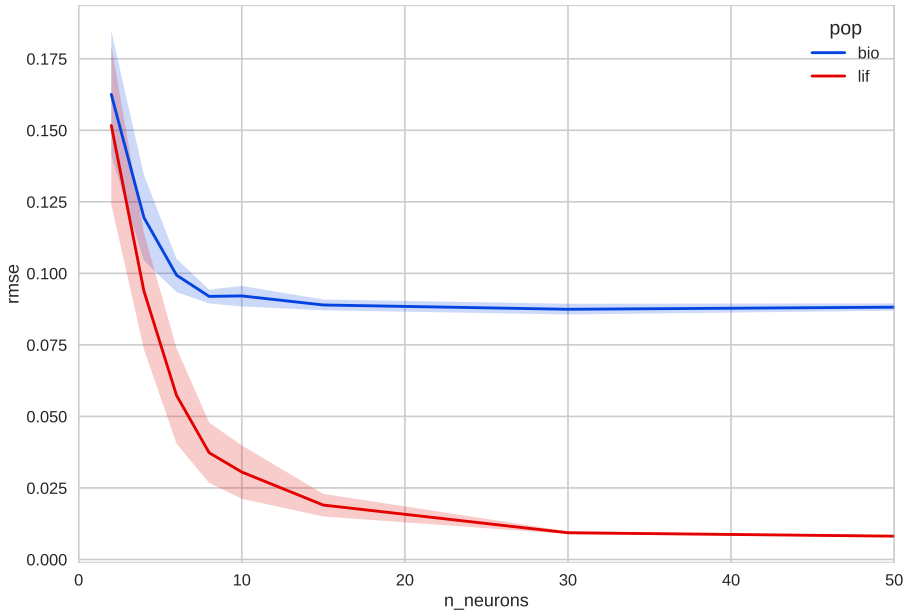

FIGURE 3.11: Root-mean-squared-error between the bioneuron/LIF representations and the target decoding as a function of the number of neurons in the respective populations. Shaded areas represent $95\,\%$ bootstrapped confidence intervals over 10 realizations.

3.4 Discussion

This chapter sought to integrate bioneurons into the NEF by training their synaptic weights until the cells' responses matched a set of ideal spike trains. This approach has two strengths. First, any empirical or simulated spike train can be used as the target of training, allowing the bioneurons to, in principle, exhibit any electrophysiological behavior (though different `NEURON` models reflecting different morphologies would likely be necessary). Second, any method for optimizing weights can be applied to achieve this goal. The choice to match the spiking behavior of LIF neurons was motivated by their simplicity, generality, and representational ability. It was relatively easy to reproduce LIF spikes using a simple evolutionary algorithm, though training time could undoubtedly be shortened, and generalizability improved, using more advanced techniques.

Bioneurons trained using this method were also able to accurately represent and transform state-space information in feedforward networks. However, the training was less successful at finding synaptic weights on recurrent connections, as evidenced by the bioneurons' inability to integrate an incoming signal over time. These shortcomings likely result from a failure of the trained weights to generalize to novel signals or dynamics, a problem that may be fundamental to the spike-matching approach itself. This method may also be criticized on the grounds that it (a) requires long and irregular training regimes, (b) necessitates the choice of ideal spiking behavior, which may constrain the range of electrophysiological behaviors that the bioneurons can exhibit, and (c) does not fully utilize the NEF notion of encoders and decoders to help structure the training.

Chapter 4 addresses this last concern by developing a new training method. This technique more fully incorporates the NEF by decomposing the synaptic weights into decoders and encoders, then using the least-squares `solvers`, in

conjunction with the oracle method and optimized readout filters, to train the weights with the explicit goal of minimizing representational error in dynamical systems. Although this method does not explicitly address electrophysiological realism, it does perform NEF-inspired neural computation more effectively.

Chapter 4

Bioneurons with Oracle Training

4.1 Introduction

This chapter develops an alternative method for training bioneurons which more fully utilizes the theoretical tools available with the NEF. Unlike the spike-matching approach, which prioritizes the neuron-space, the goal here is to specify synaptic weights and readout filters such that bioneurons optimally perform representations, transformations, and dynamics in the state-space.

When training with the oracle method, synaptic weights are decomposed into encoders and decoders, which are handled separately during initialization and training:

$$w_{ij} = \mathbf{d}_i \cdot \mathbf{e}_j \tag{4.1}$$

where w_{ij} is the weight on the `ExpSyn` connecting presynaptic neuron i to bioneuron j, $\mathbf{d}_i$ is the presynaptic neuron's decoder for the `pre` to `bio` connection, and $\mathbf{e}_j$ is the bioneuron's encoder. The $\mathbf{d}_i$'s are computed using standard NEF techniques, while the $\mathbf{e}_j$'s are chosen randomly, with the constraint that the resulting w_{ij} produce a heterogeneous spiking response (to different points in state-space and across the population) when a time-varying input is fed to the bioneurons. Once the input signal is encoded in the bioneurons' spikes, the burden of training falls on the oracle method, which is applied to calculate

the output, or **readout**, decoders that minimize the bioneurons' representational error. After discussing the new methods for setting bioneurons' synaptic weights, this chapter assesses the state-space accuracy of bioneurons trained with the oracle method on all the networks introduced in Chapter 3.

4.2 Methods

The initialization of `NEURON` objects proceeds as before, except for a few differences related to the synaptic weight decomposition. Each bioneuron is assigned an encoder $\mathbf{e}_j$, gain α_j, and bias β_j from uniform distributions (the same distributions used to choose these parameters for the ideal LIF neurons in Chapter 3). Each synaptic weight is computed according to Equation 4.1: for LIF-to-bio connections, $\mathbf{d}_i$ is calculated using a `nengo` least-squares solver (which minimizes the representational error when computing the function specified on the `pre` to `bio` connection); for bio-to-bio connections, $\mathbf{d}_i$ is calculated using the oracle method described below. The weights are then multiplied by α_j and added with a bias term w_{i,β_j}.

The bias term emulates baseline bioneuron activity by converting the β_j LIF parameter, which is equivalent to a current injected directly into the soma, into equivalent perturbations of the synaptic weight matrix. To realize this perturbation in a more biologically-plausible manner, a constant bias is decoded off of `pre`'s activities. The necessary decoders are calculated using a `nengo` solver, where the input activities are `pre`'s tuning curve rates and the targets are the bias value. These decoders essentially decode a constant value off of any activity coming from `pre`: as long `pre` is connected to `bio`, this extra component of the synaptic weight matrix should effectively inject a constant current β_j into bioneuron j.

Although choosing $\mathbf{e}$, α, and β in this manner does not ensure LIF-like tuning curves, these parameters play the same theoretical roles as in LIF

neurons (i.e., α controls magnitude of response to a fixed input and β controls baseline activity), and their distribution guarantees that each bioneuron will have a unique spiking response to inputs (see Figure 4.1).

As in Chapter 3, a white noise or sinusoidal signal $\mathbf{u}(t)$ is sent to (a) an `oracle` that analytically computes the connection's desired function $\mathbf{x}_{target}(t)$ and (b) a `pre` population of LIF neurons that generates spikes representing that signal. These spikes are transmitted to the synapses of `bio`, whose weights have been initialized using the decoders implementing the desired function. Importantly, even though weights are decomposed into decoders and encoders, the signal from `pre` is *not* decoded to state-space then encoded back to neuron-space: as in Chapter 3, spikes are transmitted directly from `pre` to `bio` (recall the bottom of Figure 1.3). `NEURON` simulates the internal dynamics of the bioneurons, whose spiking output is recorded and smoothed with a filter, producing the activities $a_{bio}(t)$ corresponding to $\mathbf{x}_{target}(t)$. A lowpass filter, the `nengo` standard for converting spikes to activities, is initially used to calculate $a_{bio}(t)$. Finally, the oracle method is applied to compute the bioneuron decoders from $a_{bio}(t)$ and $\mathbf{x}_{target}(t)$. These decoders are used to estimate (readout) the current state-space representation in `bio`, $\hat{\mathbf{x}}_{bio}(t)$, and to calculate the synaptic weights on bioneuron-to-bioneuron connections.

4.2.1 Tuning Curves and Representation

In order for the oracle method to successfully implement the three principles of the NEF, bioneurons must have heterogeneous tuning curves. Using the same feedforward communication channel as in Chapter 4, and drawing the bioneuron parameters $\mathbf{e}$, α, and β from standard `nengo` distributions, the effective tuning curves in response to a white noise input are shown in Figure 4.1. Although the synaptic weights are no longer trained to ensure LIF-like tuning, these curves nonetheless display heterogeneity in preferred direction,

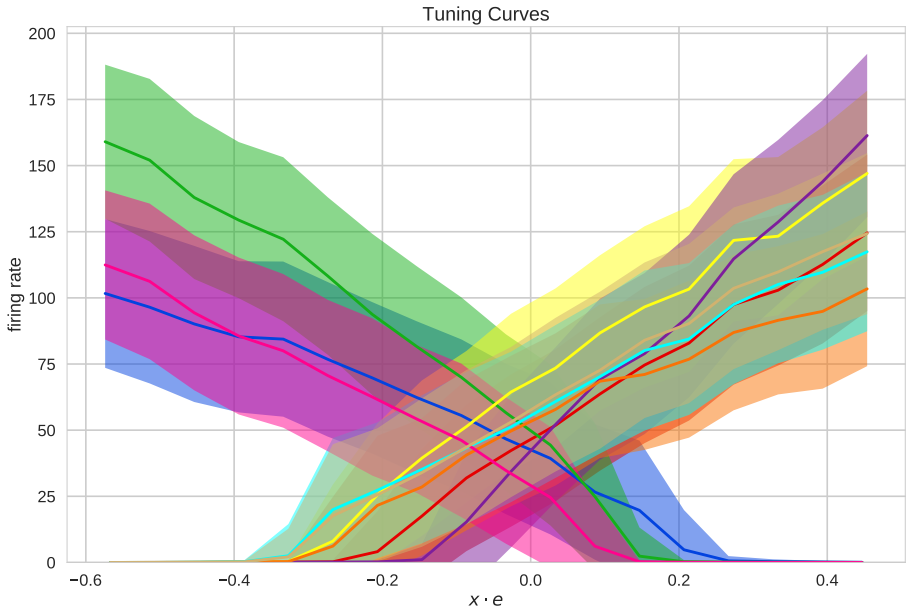

FIGURE 4.1: Bioneuron tuning curves with randomly distributed encoders, gains, and biases. Tuning curves are calculated from each population's time-varying inputs and activities: the one-dimensional state-space covered by $-1 < \mathbf{u}(t) < 1$ is divided into $n = 20$ bins, and the times when $\mathbf{u}(t)$ falls within each bin are recorded. The mean and standard deviation of the neural activities at each time within each bin are calculated, giving an indication of the range of activities associated with each stimulus value.

slope, and intercept. This establishes the necessary variety in $a_{bio}(t)$ for decoder calculation using the oracle method. That said, it appears that varying the synaptic weights tends to produce tuning curves centered around one of two prototypical curves (x-intercept around ± 0.2 and pseudo-linear slope). Future work could benefit from incorporating more variance into the bioneurons themselves, presumably through varying their geometric or electrophysiologic parameters in `NEURON`, so as to manifest more diverse tuning curves.

4.2.2 Representation

The same network as shown in Figure 3.1 is used to examine basic representation with the oracle method. To demonstrate generality, the training and testing signals are switched from the sinusoids used in Chapter 3 to white noise

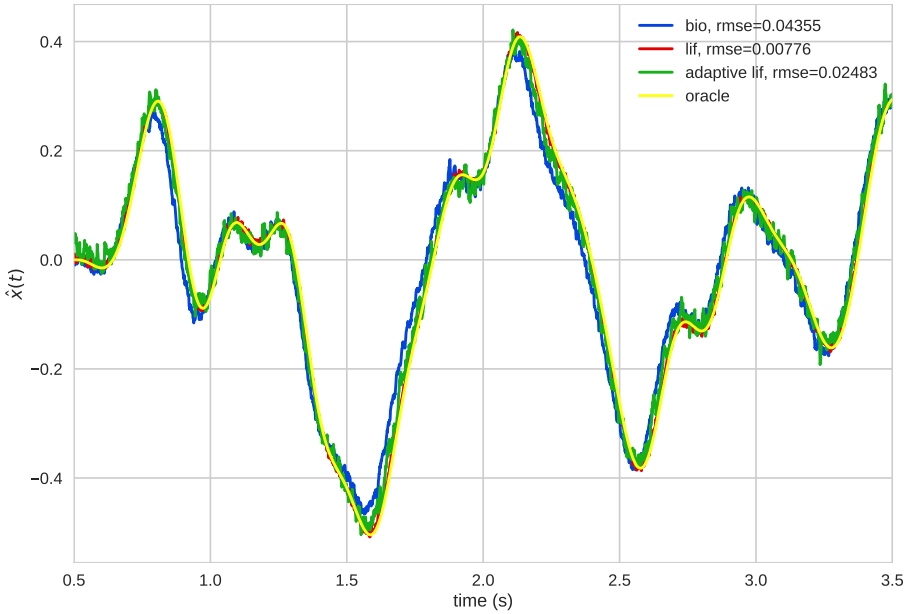

FIGURE 4.2: Comparison of the state-space estimates $\hat{\mathbf{x}}_{bio}(t)$, $\hat{\mathbf{x}}_{ideal}(t)$, and $\hat{\mathbf{x}}_{adapt}(t)$ to the target signal $\mathbf{x}_{target}(t)$ for a feed-forward communication channel when training synaptic weights using the oracle method. Both the bioneurons and adaptive LIF neurons exhibit phase-shift errors to signals (especially those with higher frequency), indicating that the neurons' internal adaptive dynamics may be responsible.

signals with unique training and testing seeds (but identical cutoff frequencies, starting values, and RMS). Figure 4.2 shows the oracle-decoded output of `bio`. The estimate $\hat{\mathbf{x}}_{bio}(t)$ is a reasonable representations of the target signal $\mathbf{x}_{target}(t)$, but the phase and magnitude errors remain. The oracle method's ability to decode bioneuron activity is unsurprising given the heterogeneity of $a_{bio}(t)$ demonstrated in Figure 4.1 and the decoding abilities of the oracle method established in Chapter 3. Still, the persistence of the phase shift under this decoding scheme indicates that these errors originate in the bioneurons' internal dynamics, rather than the synaptic weight distribution or readout decoding.

4.2.3 Eliminating the Phase Shift

Additional methods must be applied to address this error. Interestingly, adaptive LIF neurons (ALIF, green in Figure 4.2) exhibit the same phase shift phenomenon; their simplicity may illuminate sources of the problem and suggest solutions that can be applied to the bioneurons. ALIFs' effective firing rates change over time in a manner that approximates biological neurons' **adaptation**, the changes in firing rate in response to a constant input that result from ion channel interactions over longer timescales. ALIFs function like their LIF counterparts except that an adaptation state n is subtracted from the input current. n is incremented every time the neuron spikes and its dynamics are given by

$$\dot{n} = -n/\tau_n. \tag{4.2}$$

In a feedforward network, this adaptation effectively causes ALIF neurons to begin firing sooner than their LIF counterparts, but afterwards exhibit very similar rate dynamics. Neither the static nor the oracle decoders are capable of linearly weighting these activities in a manner that aligns with the target signal: the best they can do (in terms of minimizing RMSE between the estimate and the target) is to produce an estimate that has shape and magnitude similar to the target, but is aligned with the earlier onset of the ALIF spikes, producing a phase shift to the left.

4.2.4 Additional Inputs

The adapting neurons appear to respond more quickly to changes in the target signal by approximating the signal's derivative and using it to "predict" future input values. One potential solution to the phase shift problem is to drive the adapting neurons with additional inputs in order to align their spikes with the signal, in some senses canceling the adaptation. To achieve this, the dimension of the bioneuron and ALIF populations is increased by 1, and the derivative of

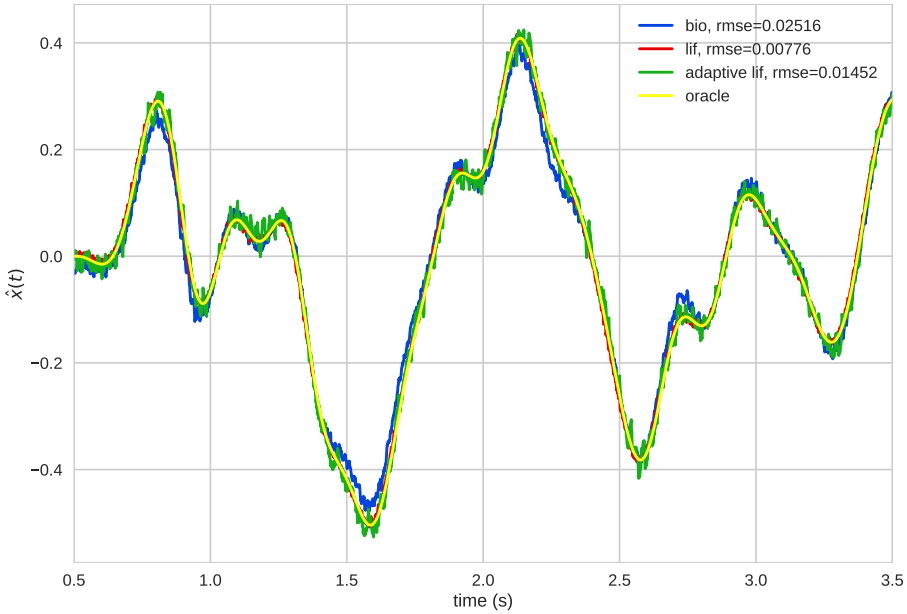

FIGURE 4.3: Comparison of the state-space estimates $\hat{\mathbf{x}}_{bio}(t)$, $\hat{\mathbf{x}}_{ideal}(t)$, and $\hat{\mathbf{x}}_{adapt}(t)$ to the target signal $\mathbf{x}_{target}(t)$ for a feedforward communication channel when $\dot{u}(t)$ is fed into the bioneurons and ALIF neurons. This extra information reduces phase-shift errors significantly.

the input signal $\dot{\mathbf{u}}(t)$ is fed into this new dimension. The motivation behind this choice is that feeding this information into the neurons should oppose the neurons' predictive tendencies through interaction with the `ExpSyn` filter, the nonlinear dendritic filters, and the system dynamics. Figure 4.3 shows that the oracle method finds decoders that better estimate the signal when neurons have access to this extra information. Although various combinations of $\mathbf{u}(t)$'s derivatives and integrals (orders 0 to 3) were shown to improve the fit, an exploration of how the bioneurons' inherent dynamics interact with these inputs is beyond the scope of this thesis (though potentially a fruitful avenue for future work).

4.2.5 Training Readout Filters

Another potential solution is to utilize more complex filters to readout the state represented by the bioneuron spikes. Up until this point, a lowpass filter,

whose transfer function is given by

$$H(s) = \frac{1}{1+\tau s}, \tag{4.3}$$

has been used to smooth spikes into neural activities. The persistence of the phase shift with both static and oracle decoders indicates that no linear combination of activites smoothed this way will successfully estimate the target. With this new method, both the decoders *and* the readout filter are adapted to account for the bioneuron/ALIF internal dynamics. To do this, the $1+\lambda$ evolutionary algorithm is applied to optimize the parameters governing an arbitrary filter, which can be written in the standard transfer-function notation as

$$H(s) = \frac{b_m s^m + b_{m-1}s^{m-1} + \cdots + b_1 s + b_0}{a_n s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0} \tag{4.4}$$

or expressed in pole-zero notation as

$$H(s) = \frac{N(s)}{D(s)} = K\frac{(s-z_1)(s-z_2)\ldots(s-z_{m-1})(s-z_m)}{(s-p_1)(s-p_2)\ldots(s-p_{n-1})(s-p_n)} \tag{4.5}$$

where $z_{1\ldots m}$ are the roots of the numerator and the **zeros** of the transfer function, $p_{1\ldots m}$ are the roots of the denominator and the **poles** of the transfer function, and K is the gain. The optimization proceeds by choosing zeros, poles, and gain, simulating the network, collecting `bio` spikes, smoothing them into $a^*_{bio}(t)$ with the chosen filter, calculating oracle decoders $\mathbf{d}^*_{bio}$ from $a^*_{bio}(t)$ and $\mathbf{x}_{target}(t)$, and estimating the state using equation 1.5 with $\mathbf{d}^*_{bio}$ and $a^*_{bio}(t)$. Figure 4.4 shows that the evolved filters and computed decoders together reduce the phase shift and faithfully estimate the target signal; Table 4.1 reviews the relative accuracies of the four training methods. Figure 4.5 shows that the impulse response of the evolved filters (both for the bioneurons and the ALIFs) bears a strong resemblance to the lowpass filter. Testing indicates that combining both the derivative and filtering techniques does not lead to

TABLE 4.1: Relative accuracies of the bioneuron training methods. Reported RMSEs reflect error in a feedforward network when trained and tested on the same pair of white noise signals (training and testing seed differ).

Method	RMSE
spike match	0.0795
vanilla oracle	0.0435
derivative oracle	0.0251
filter oracle	0.0096

compounded improvements, so only the filtering method is utilized to eliminate the phase shift for the remainder of this chapter.

4.3 Results

4.3.1 Computing Linear and Nonlinear Functions

A linear and nonlinear transform are applied to the `pre-to-bio` connection to establish functional decoding. Because synaptic weights are composed of encoders and decoders in the oracle training, standard NEF techniques can be used to compute the $\mathbf{d}_i$ that optimally implement these transforms, and these functional decoders can be used to calculate synaptic weights. As shown in Figure 4.6 and 4.7, this technique successfully computes the desired functions. Two inputs are also fed into `bio1` to ensure linear addition of state-space inputs; Figure 4.8 demonstrates the oracle method again successfully finds readout decoders. In all three cases, the RMSE is proportionally smaller than the spike-match decodes in Chapter 3 (and absolutely smaller when training and testing on sinusoids, not shown).

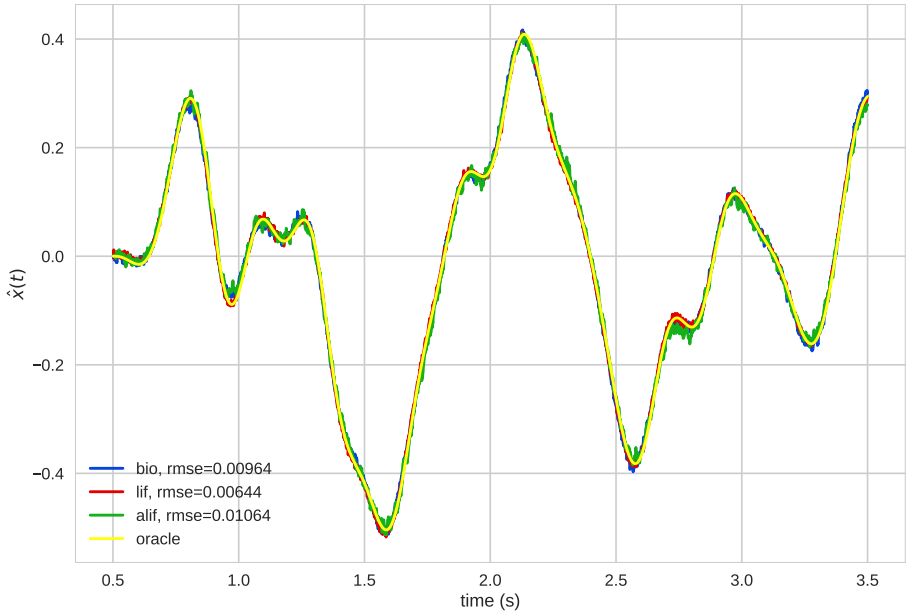

FIGURE 4.4: Comparison of the state-space estimates $\hat{\mathbf{x}}_{bio}(t)$, $\hat{\mathbf{x}}_{ideal}(t)$, and $\hat{\mathbf{x}}_{adapt}(t)$ to the target signal $\mathbf{x}_{target}(t)$ for a feedforward communication channel when an evolved filter with 1 zero ($z_0 = -41.8$) and 2 poles ($p_0 = -26.8,\ p_1 = -8.9$) is used to decode the bioneuron and ALIF spikes. This specialized filtering, in conjunction with the oracle decoders, totally eliminates the phase shift.

4.3.2 Dynamical Systems and the Neural Integrator

Next, two bioneuron populations are connected in a feedforward communication channel. To compute the synaptic weights that connect `bio1` and `bio2`, the oracle method must be used to calculate the decoders out of `bio1` with the appropriate functional transform (in this case, identity). This requires two successive simulations of the network. In the first, $a_{bio1}(t)$ and $\mathbf{x}_{target1}(t)$ are gathered and the $\mathbf{d}_{bio1}$ are computed using the oracle method. In the second, the synaptic weights $w_{bio1-bio2} = \mathbf{d}_{bio1} \cdot \mathbf{e}_{bio2}$ are computed, then the simulation is run, $a_{bio2}(t)$ and $\mathbf{x}_{target2}(t)$ are gathered, and the representation in `bio2` is calculated from $\mathbf{d}_{bio2}$ (oracle method again). Figure 4.9 shows that $\mathbf{d}_{bio1}$ can be used, in conjunction with $\mathbf{e}$, α, and β, to compute synaptic weights on bioneuron-to-bioneuron connections in a manner that preserves (and transforms) the state-space representation in `bio1`.

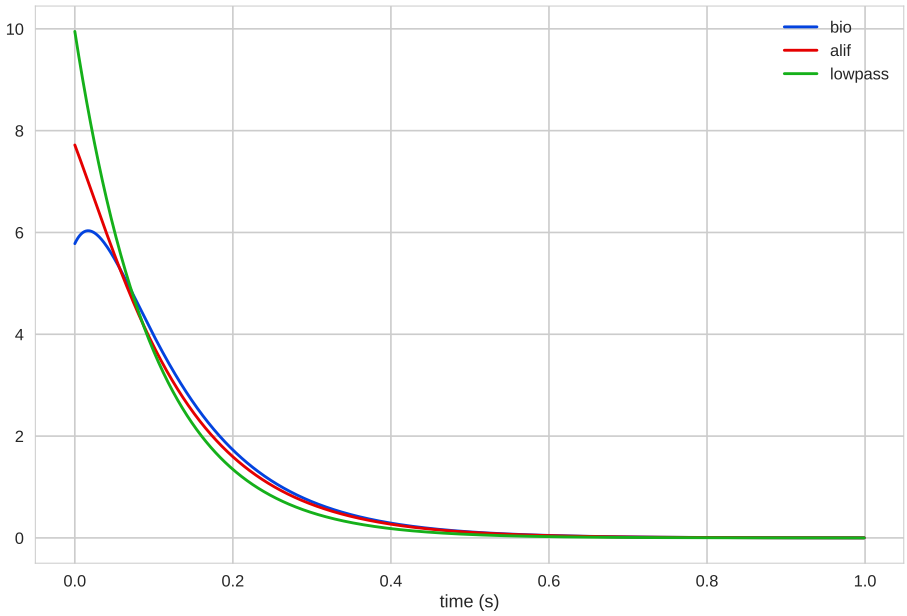

FIGURE 4.5: Impulse response of a typical filter evolved using the revised oracle method. Preliminary testing indicated that transfer functions with one zero and two poles were a decent compromise between filter complexity (ability to smooth the bioneuron spikes to match arbitrary target signals) and biological realism (similarity to a lowpass filter, a standard approximation for synaptic filtering). The plotted filter is the same used in Figure 4.4.

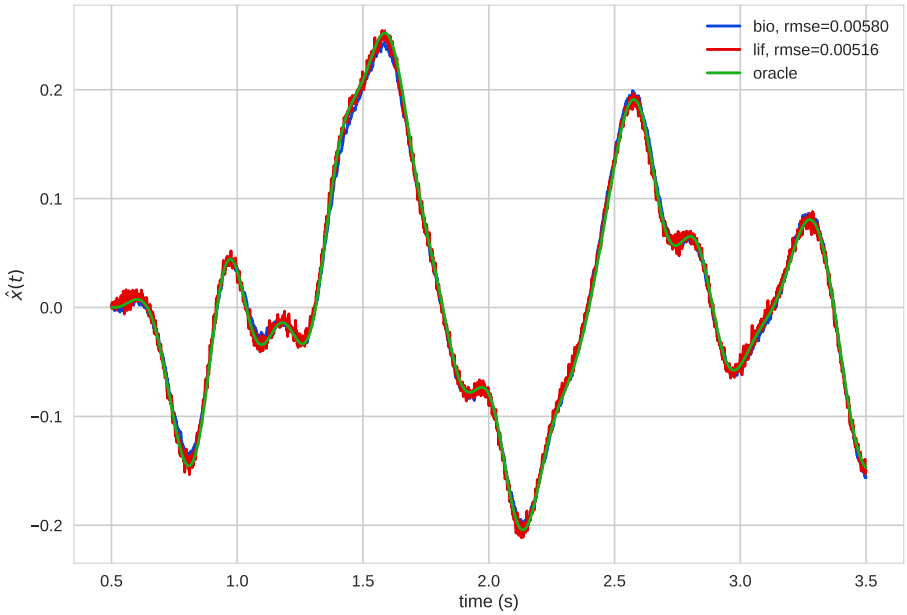

FIGURE 4.6: Comparison of the target signal, bioneuron estimate, and LIF estimate when computing a linear transform of $T = -0.5$.

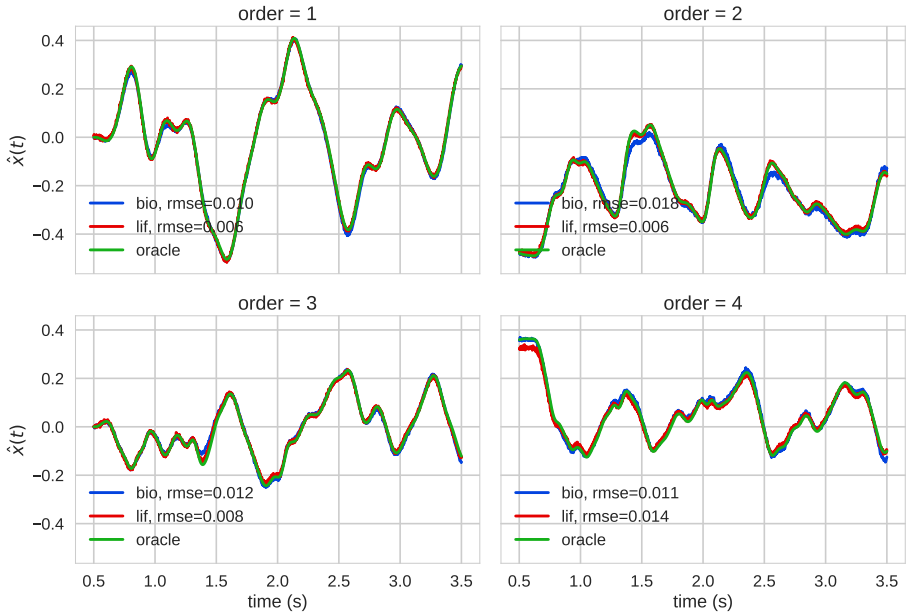

FIGURE 4.7: Comparison of the target signal, bioneuron estimate, and LIF estimate when computing a nonlinear function, the n^{th} order Legendre Polynomial. RMSEs of the bioneuron estimates are smaller than the spike-match trained decodes by a factor of 4 to 7.

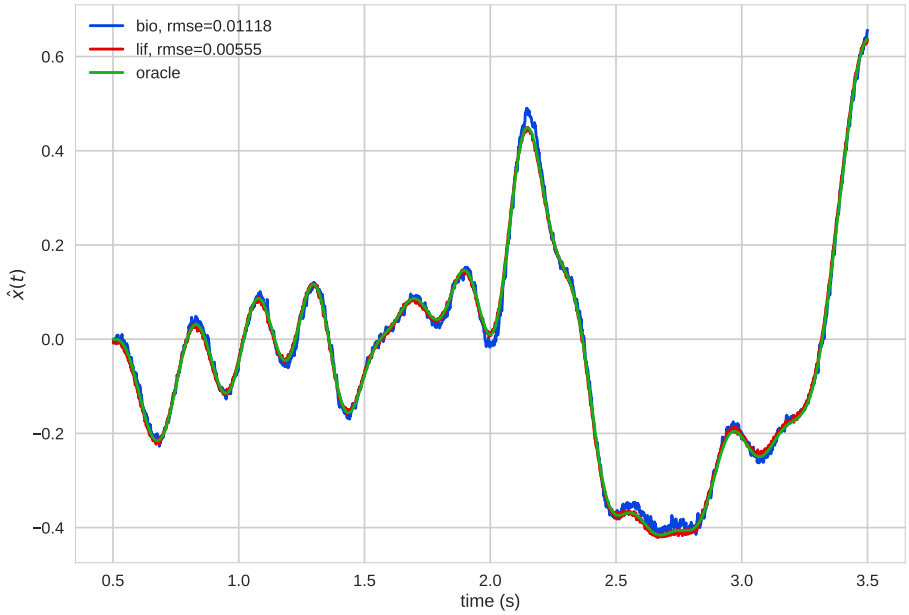

FIGURE 4.8: Comparison of the target signal, bioneuron estimate, and LIF estimate when two inputs (two white noise signals) are fed into the network. This decode does not exhibit problems with state-space additivity as did the spike match approach, Figure 3.7

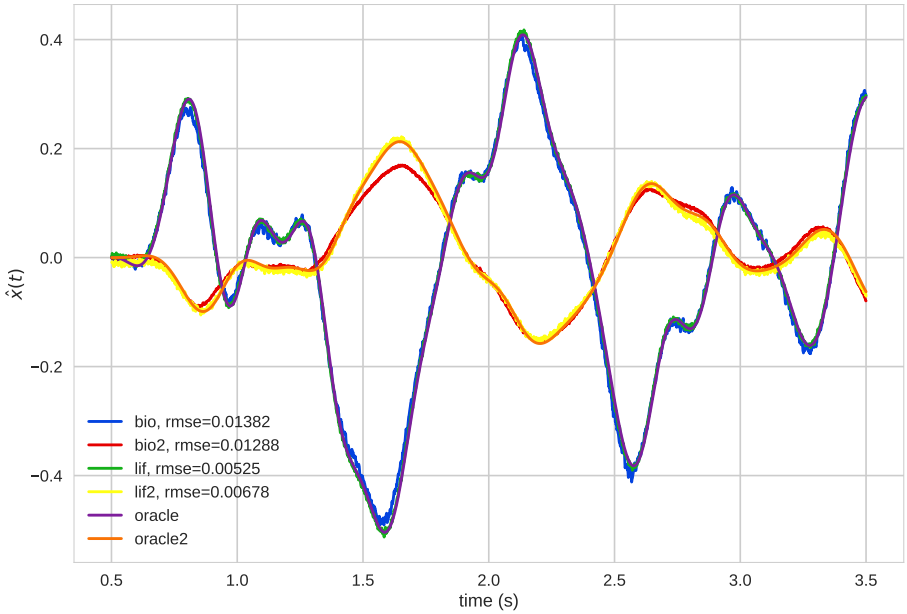

FIGURE 4.9: Comparison of the target signal, bioneuron estimate, and LIF estimate from the second population in a feedforward chain. A transform of $T = -0.5$ is applied to the connection between the first and second ensembles to ensure bioneuron-to-bioneuron connections can correctly compute functions.

Finally, `bio` is recurrently connected to construct an integrator. However, several extra steps are needed to compute the decoders and filters for this feedback connection. Because the choice of decoder now affects the bioneurons' activities (through Equation 4.1, which in turn affects what the oracle method computes as the optimal decoder ($\mathbf{d}_{recurrent} = \texttt{solver}(a_{bio}(t), \mathbf{x}_{target}(t))$), it is no longer possible to do one-shot learning like in the feedforward networks.

Oracle Spike Feedback

As a first attempt to train the recurrent bioneuron decoders, the spikes from `bio` are not transmitted directly back to `bio`. Instead, during training, $\mathbf{x}_{target}(t)$, is used as the ideal state-space recurrent signal. Because bioneurons may only receive spikes, this ideal signal must first be translated to spikes by passing it through an intermediary population `inter`, which plays an analogous role to `pre` for the feedback signal. In the training step, the direct `bio`-to-`bio` connection is removed so that the bioneurons only see the "ideal" spikes

as recurrent inputs. To ensure the spikes from `inter` (during training) will align as closely as possible with the recurrent `bio` spikes (during testing), the evolved readout filter is applied on the connection into `inter`. The intention here is to filter $\mathbf{x}_{target}(t)$ as it passes into `inter` in a manner resembling the bioneurons' dendritic filters. The final training network is thus `ideal` $\rightarrow$ (evolved filter) `inter` $\rightarrow$ (lowpass) `bio`, where the terms in parentheses are the applied presynaptic filters.

The training proceeds as follows. First, a feedforward pass of the network is simulated (`inter-bio` and `bio-bio` connections removed, and $\mathbf{x}_{target}(t) = \mathbf{u}(t)$), and the readout filters are evolved. Recall that these filters translate the phase-shifted `bio` spikes back to an aligned state-space representation. In the second step of training, these filters are applied on the `ideal-inter` connection, with the intention that the signal from `ideal` will be similarly translated, and that the inputs to `inter` will resemble (in state-space) the misaligned recurrent spikes that `bio` will produce during testing. Because `ideal` feeds `inter` with $\mathbf{x}_{target}(t)$ and `inter` connects to `bio` during this training stage, `bio` will receive training spikes that resemble the ideal state-space feedback signal *and* account for the bioneurons' nonlinearities. In the final training step, the above network is simulated, the spike from `bio` are collected, and a new set of $\mathbf{d}^*_{bio}$ and readout filters are evolved. During testing, the `inter-bio` connection is removed, and $\mathbf{d}^*_{bio}$ is used to calculate the synaptic weights on the recurrent `bio-bio` connection, while the new readout filters and decoders are used to filter spikes and estimate the bioneuron's output state, $\hat{\mathbf{x}}_{bio}(t)$. The number of neurons was increased to 300 for the integrator network.

Figure 4.10 compares the bioneuron and LIF state estimates to the ideal value for the integrator network. Although the above training method was quite successful in decoding the training signal ($\mathrm{RMSE} = 0.03$ for bioneurons and ALIF, not shown), it was only somewhat successful when decoding novel signals: both the ALIF and bioneurons capture many dynamical features of the

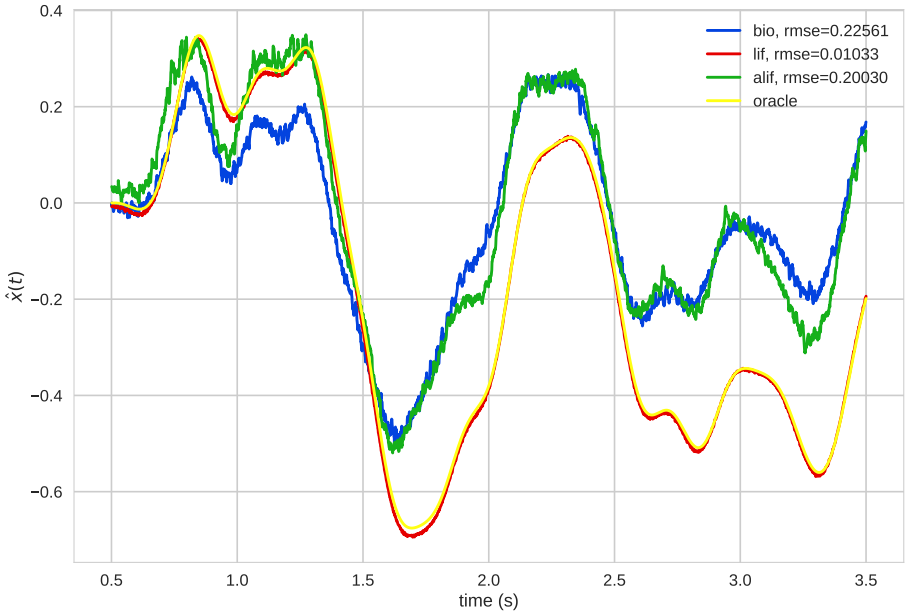

FIGURE 4.10: Comparison of the target signal, bioneuron estimate, ALIF estimate, and LIF estimate for a recurrently-connected integrator network trained using spiking feedback from the oracle. This training method finds recurrent synaptic weights and readout decoders/filters that approximate the desired dynamical system for the adapting neurons, but longer training is needed to reduce the bioneurons' error to acceptable levels.

ideal integrator, but frequently drift away from the target signal and display incorrect magnitudes. Test accuracy did not improve significantly with additional training, suggesting that some aspect of this training method hampers generalization. Unfortunately, time constraints prohibited the exploration of training regimes based on this method; with luck, future work can reduce the remaining discrepancies with more exhaustive training.

Evolved Decoders Feedback

As an alternative method to train the recurrent decoders, the oracle feedback through `inter` is removed, and the recurrent decoders are optimized separately using the usual evolutionary approach. In the first pass of training, readout filters and decoders are evolved for a feedforward communication channel (no recurrent connection on `bio`, $\mathbf{x}_{target}(t) = \mathbf{u}(t)$). In the second step, a new

evolutionary algorithm trains the recurrent bioneuron decoders: fitness is determined by simulating the network with the specified decoders, collecting the bioneuron spikes, estimating the state using the previously optimized readout decoders and filters, and computing the RMSE between the estimate and the target. Figure 4.11 shows that this method successfully computes recurrent decoders in a network populated with ALIF neurons, and that this method generalizes to the test signal better than the previous method ($\mathrm{RMSE}_{\mathrm{inter}} = 0.0346$ vs. $\mathrm{RMSE}_{\mathrm{evolve}} = 0.2003$ for ALIFs, despite similar RMSEs on the training signals). Interestingly, initializing the evolutionary population by mutating copies of the readout decoders led to significantly faster evolution than when using uniformally-distributed random decoders. This suggests that feeding back a neuron-space signal which resembles (in terms of weighting) the target state-space signal produces the desired integration (as we would expect from the NEF), but that slight modifications to the decoders are necessary to account for adaptation and bioneuron nonlinearities. Future work should compare the properties of NEF, readout, and recurrent decoders, looking especially at differences in the state-space estimates they produce.

4.4 Discussion

This chapter sought to integrate bioneurons into the NEF by decomposing synaptic weights into decoders, encoders, gains, and biases, then using least squares solvers to compute the optimal decoders while filtering the bioneuron spikes with an evolved readout filter. This approach has two strengths. First, encoding and decoding on bioneurons' spikes permits straightforward translation between neuron space and state space. Second, the training is focused on state-space accuracy, which results in better implementation of the three NEF principles: representation, transformation, and dynamics. For these reasons, the oracle method is chosen to be the training method in Chapter

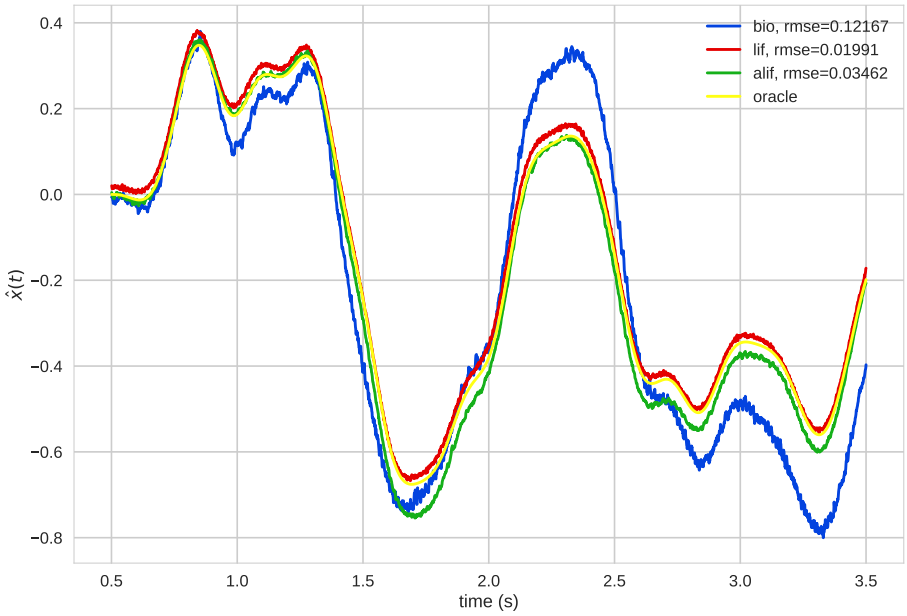

FIGURE 4.11: Comparison of the target signal, bioneuron estimate, ALIF estimate, and LIF estimate for a recurrently-connected integrator network trained using evolved recurrent decoders. This training method also finds recurrent synaptic weights and readout decoders/filters that implement the desired dynamical system for the adapting neurons. However, longer training and larger bioneuron ensembles (here, 30 bioneurons for 1 generation) are needed to ensure this result is robust.

5, which returns to the WM model presented in Chapter 2 and applies this method when substituting bioneurons into the WM population. It then presents several experiments that simulate ADHD and its pharmacological treatments by biophysically perturbing the neurons in a manner that was not possible with LIF neurons.

Chapter 5

Applications and Conclusions

5.1 Applying Bioneurons to the WM Model

The previous chapters demonstrated that populations of bioneurons could represent and dynamically transform information with reasonable accuracy, indicating their suitability for use in arbitrary NEF models. However, the additional challenges imposed by substitution into interconnected networks, including increased noise and poorly-defined target behavior, make this final test of neural capability far from trivial. To assess the bioneurons' functionality within the context of a larger brain model, the 100 LIF neurons in the WM population of the Chapter 2 model were replaced with bioneurons; all other aspects of the model remained unchanged.

5.1.1 Training the Bioneuron WM

The major difficulty with applying bioneurons to the WM model is training the recurrent and readout decoders. Recall that Figure 4.10 and 4.11 showed that the bioneuron integrator often has significant generalization errors on differently-seeded white noise signals. Even worse, the testing signals in the DRT model are either constant or zero in their respective dimensions, meaning their statistics and dynamics differ significantly from the signals previously used for training. To minimize these difficulties, an "extra information" training

regime is used to give the bioneuron WM the best possible opportunity to function in context. This training signal is very similar to the test signal: $\mathbf{u}_{train}(t) = [cue(t), 0]$ and $\mathbf{u}_{test}(t) = [cue(t), k_{time}]$. This produces a target signal whose first dimension represents perfect integration of the cue stimulus, $\mathbf{x}_{target}(t) = \mathbf{x}_0(t)$, but whose second "time" dimension, $\mathbf{x}_1(t)$, is not trained at all. Futhermore, the training signal only lasts $4\,\mathrm{s}$, meaning that the final $5\,\mathrm{s}$ of the test simulation is unfamiliar to the WM bioneurons: they must rely on the trained recurrent decoders to keep the representation active. The logic behind this training regime is that (a) the bioneurons will have enough training data to store and maintain the cue location, but not enough to perfectly decode the cue location over the whole testing period, and (b) feeding an input into the untrained time dimension will destructively interfere with the cue representation.

5.1.2 Simulating the DRT and Biophysical Perturbations

The recurrent and readout decoders are trained as above, then the model is initialized with bioneurons populating the WM population. To simulate the effects of GFC and PHE in a biophysically accurate manner, the conductance of the I_h channel in the `NEURON` model, $\bar{g}_{I_h}$, is multiplied by a constant k_g. Recall from Chapter 2 that these channels, which correspond to the HCN channels in pyramidal PFC neurons, are open at rest, and that agnositic binding by GFC closes them (antagonistic binding by PHE opens them further), altering the cell's excitability. Preliminary tests indicated that the default $\bar{g}_{I_h}$ corresponds to nearly all HCN channels being open, making it impossible to test the effects of further channel opening via PHE. To compensate for this, the default conductance is increased by multiplying $\bar{g}_{I_h}$ with $k_g = 500$, which effectively closes a larger portion of the I_h channels when the cell is at rest (control condition). $k_g = 100$ is chosen for the PHE condition and $k_g = 1000$ for the GFC condition.

Figure 5.1 shows the WM representation and DRT accuracy as a function of delay period length for the biophysical perturbation. As with the figures in Chapter 2, the WM stores and maintains the cue location at the start of the delay period, but as noise and imperfect recurrence take their toll, the representation decays to zero, introducing errors into the decision procedure. The shape of the bioneuron WM decay differs from previous results, with the representation remaining near $|\hat{\mathbf{x}}_0(t)| = 1$ until around $4\,\mathrm{s}$, then decaying linearly rather than exponentially. This is likely related to the training period duration of $4\,\mathrm{s}$ and to the different sources of interference. Nonetheless, the biophysical GFC and PHE perturbations had the expected effects, leading to different WM decay curves. Unfortunately, time constraints prevented a full round of model simulations (50 bioneuron realizations compared to 1000 LIF realizations), making it difficult to assess the accuracy of the forgetting curves. Given the differences in the WM decay curves between control, GFC, and PHE, it is reasonable to suspect that the forgetting curves would follow the trends of Chapter 2, given more realizations.

Simulated GFC and PHE also differentially altered the firing rates of bioneurons with cue-aligned encoders in a manner qualitatively similar to the electrical and neural perturbations of LIF neurons, Figure 5.2. This is interesting given that bioneurons' encoders only act indirectly through the decomposition of synaptic weights, rather than directly transforming the state space inputs as with LIF neurons.

5.1.3 Discussion

The results of applying bioneurons to the WM memory model are not incontrovertible: specialized training was required to capture the desired WM decay, and insufficient model data exists to make statistically confident conclusions. An optimistic outlook is that with additional training and further refinements

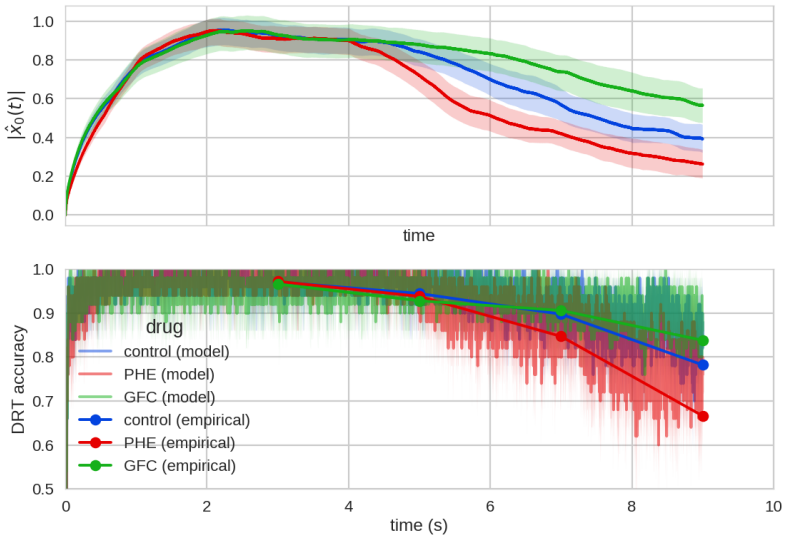

FIGURE 5.1: Cue representation and forgetting curve for the bioneuron WM under the biophysical drug simulation. The noise in the bottom plot reflects the low number of realizations simulated with the bioneurons (50) due to time constraints.

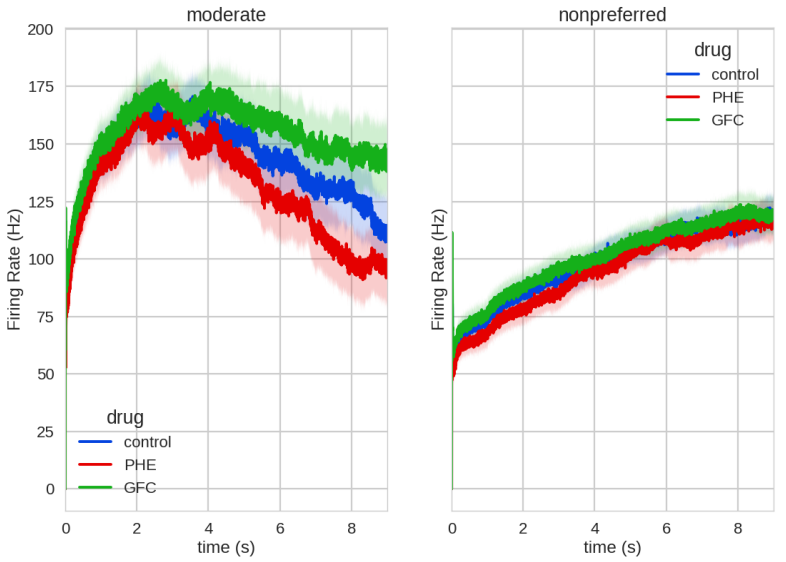

FIGURE 5.2: Firing rate of simulated bioneurons for the biophysical drug simulation.

of the Chapter 4 methods, the bioneurons will more robustly perform integration and instantiate an effective WM. Ideally, this would demonstrate more conclusively that bioneurons' have cognitive potential in the context of larger models. Efforts towards meeting this objective are currently underway.

5.2 Conclusions

Engineering artificial neural systems that both respect biology and behave cognitively is a difficult task. Recent work has shown that networks of neuron-like objects can be networked together and trained to perform sophisticated functions: Google DeepMind's "AlphaGo" has mastered the game of "Go" with deep neural networks (Silver et al., 2016), while the Computational Neuroscience Research Group's "Spaun" has performed symbolic pattern matching and memory tasks using the Neural Engineering Framework (Eliasmith et al., 2012). However, for reasons of computational and analytical tractability, these simulations ignore many neurobiological features, raising questions about their biological plausibility. Recent work has also shown that our understanding of neuroscience is now sufficiently deep to digitally reconstruct sections of brains that, when simulated, produce data that are consistent with electrophysiological, geometric, and connectomic experiments (Markram et al., 2015). However, these reconstructions have not yet been applied to functional tasks that require the representation and manipulation of information, raising questions about their cognitive plausibility.

This thesis attempted to unify these two domains by building neurocomputational systems that can be more widely acknowledged as "brain models". The basic approach was to incorporate biologically realistic neuron models into the NEF; that is, to take biophysically complex models of individual neurons and substitute them for the simplified neuron models hitherto used in a functional architecture. Although `nengo`, the software package written to implement

NEF models, supports the substitution of arbitrarily complex neuron models, there are several underlying assumptions in the NEF about neural connectivity that make this incorporation theoretically difficult. Chapter 1 discussed these limitations by introducing the NEF, discussing the LIF neuron model typically used to construct NEF models, and contrasting it with the biologically realistic `NEURON` models that were used through the remainder of the thesis.

Before tackling the theoretical and practical difficulties with NEF/`NEURON` integration, Chapter 2 concretely motivated the importance of biological plausibility by introducing a spiking neural model of working memory, the delayed response task, and related mental disorders/pharmacological treatments. The DRT is an exemplar task because (a) the neurons that implement the underlying computations must perform all three NEF principles, (b) DRT performance is quanitifiably affected by ADHD and the drugs used to treat it, and (c) these pharmacological perturbations affect the relevant brain areas at a biophysical level that is normally inaccessible to cognitive models. The DRT model, built using LIF neurons and other standard NEF techniques, replicated electrophysiological and behavioral data from monkeys performing this task. When perturbed using simulated interventions that approximated the functional, electrical, and neural aspects of drug treatments, the model data continued to agree with empirical data. This suggested a multi-level theory of WM that coherently connects biological, electrical, and functional descriptions. However, the model was built upon on biological simplifications and expert knowledge that will not generally be available when constructing biologically realistic models, motivating a more accurate and generalizable approach to biological integration.

Chapters 3 and 4 comprised the main theoretical contribution of this thesis, presenting two alternative methods for incorporating `NEURON` models into NEF networks. In each case, the main challenge was to find connection weights for synapses distributed randomly along the bioneurons' dendrites such that,

when the cells were driven with input spike trains representing vector-valued information, the NEF could be used to accurately represent and dynamically transform that information.

In the first method, synaptic weights were trained so that the bioneurons' effective tuning curves matched the tuning curves of an `ideal` LIF population, whose representational and dynamical properties were known to be compatible with NEF methods. Chapter 3 showed that a simple evolutionary strategy could achieve a decent match to the behavior of spiking LIFs – good enough that the decoders calculated for the `ideal` population could reliably convert from the neuron space back to accurate state space estimates. However, this decoding became problematic when bioneurons were recurrently connected, since any discrepancy between the `ideal` and bioneuron spike trains became amplified in a feedback network.

Problems with feedforward phase shifts and feedback drift motivated the methods in Chapter 4, which trained synaptic weight and readout filters with the explicit goal of maximizing representational and dynamical accuracy. This chapter advanced the oracle method, which utilized `solvers` to calculate decoders that minimized the error between a state space representation decoded from bioneuron activities and the target state. Though these tools are usually applied to static evaluation points, this chapter used time-varying spike trains and target signals to compute decoders for both reading out the bioneurons' state space representation and for connecting bioneurons to one another. These methods met with greater success than the spike-match trained bioneurons, requiring shorter training times and producing lower representational error. Because this method was tied more closely with the NEF notion of encoders and decoders, other NEF tools were also leveraged to boost accuracy. Notably, the use of evolved readout filters to decode the bioneurons spikes was essential in eliminating the phase and magnitude errors. Unfotunately, even the refined oracle method currently struggles to produce decoders which give accurate

estimates of novel signals in recurrently connected systems.

To complete the investigation of biologically realistic neuron models and the NEF, Chapter 5 returned to the WM model of Chapter 2, substituting bioneurons for LIF neurons by applying the methods of Chapter 4. Given the aforementioned difficulties with the bioneuron integrator, several liberties were taken with training this network to maximize the likelihood that the bioneurons would function properly in context. The bioneurons were able to store and maintain the location of the DRT cue in WM over an extended period, and the model appeared to replicate the baseline forgetting curve, though more realizations are required to make definite conclusions. Next, the application of GFC and PHE were induced in the most biologically realistic manner yet: by manipulating the conductance constant for the hyperpolarization-activated cation channel in the `NEURON` model. As expected, this biophysical manipulation induced a differential change in the firing rate of cue-aligned neurons, which in turn altered the representation stored in the bioneuron WM and led to changes in the forgetting curve. The consistency between these results and the experiments in Chapter 2 makes a strong case for the biophysical, electrophysiological, and functional realism of this model. Furthermore, the successful application of bioneurons to a standard NEF model demonstrates that the theory advanced in Chapters 3 and 4 can be fruitfully applied to existing models, not only to support claims of biological realism, but to investigate a host of low-level phenomenon previously outside the perview of the NEF.

Future work may take this research in several directions. Most obviously, the training regime for the bioneuron integrator should be further developed to improve generalizations to novel signals. In general, it would be useful to begin explicitly utilizing nonlinearities within the bioneurons to help with representation, computation, and dynamics rather than try to cancel them using linear decoding. Another avenue of extension is to experiment with

other NEURON models, either by introducing cell-to-cell variance with the current NEURON model, or by investigating new models that have fully complex geometries. New strategies for synaptic weight training may be necessary in highly branched dendrites, and further revisions of the oracle method may be necessary to account for more extreme attenuation and adaptation. A more user-friendly software interface would also make training and perturbing NEF models built with bioneurons more accessible.

Unsurpisingly, millions of years of evolution have settled upon a better neurocompuational system than the one presented here. Still, this thesis represents an important first step in the ongoing effort to build biologically plausible, cognitively capable brain models. The author believes that this methodology profitably unifies bottom-up and top-down approaches to the central questions of computational neuroscience, and hopes that its extensions will prove fruitful in expanding our understanding of the human mind.

Bibliography

Abbott, Larry F and Sacha B Nelson (2000). “Synaptic plasticity: taming the beast”. In: *Nature neuroscience* 3, pp. 1178–1183.

Arnsten, AFT and FM Leslie (1991). “Behavioral and receptor binding analysis of the α 2-adrenergic agonist, 5-bromo-6 [2-imidazoline-2-YL amino] quinoxaline (UK-14304): evidence for cognitive enhancement at an α 2-adrenoceptor subtype”. In: *Neuropharmacology* 30.12, pp. 1279–1289.

Arnsten, Amy FT and Paul J Lombroso (2000). “Genetics of childhood disorders: XVIII. ADHD, Part 2: Norepinephrine has a critical modulatory influence on prefrontal cortical function”. In: *Journal of the American Academy of Child & Adolescent Psychiatry* 39.9, pp. 1201–1203.

Avery, Robert A et al. (2000). “The alpha-2A-adrenoceptor agonist, guanfacine, increases regional cerebral blood flow in dorsolateral prefrontal cortex of monkeys performing a spatial working memory task”. In: *Neuropsychopharmacology* 23.3, pp. 240–249.

Bahl, Armin et al. (2012). “Automated optimization of a reduced layer 5 pyramidal cell model based on experimental data”. In: *Journal of Neuroscience Methods* 210.1, pp. 22–34.

A general error-modulated STDP learning rule applied to reinforcement learning in the basal ganglia (2011).

Bekolay, Trevor, Mark Laubach, and Chris Eliasmith (2014). “A spiking neural integrator model of the adaptive control of action by the medial prefrontal cortex”. In: *The Journal of Neuroscience* 34.5, pp. 1892–1902.

Bergstra, James, Dan Yamins, and David D Cox (2013). "Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms". In: *Proceedings of the 12th Python in Science Conference*. Citeseer, pp. 13–20.

Brunel, Nicolas and Xiao-Jing Wang (2001). "Effects of neuromodulation in a cortical network model of object working memory dominated by recurrent inhibition". In: *Journal of Computational Neuroscience* 11.1, pp. 63–85.

Carnevale, Nicholas T and Michael L Hines (2006). *The NEURON book*. Cambridge University Press.

Chandler, Daniel J, Barry D Waterhouse, and Wen-Jun Gao (2014). "New perspectives on catecholaminergic regulation of executive circuits: evidence for independent modulation of prefrontal functions by midbrain dopaminergic and noradrenergic neurons". In: *Frontiers in Neural Circuits* 8.

Choo, Feng-Xuan and Chris Eliasmith (2010). "A spiking neuron model of serial-order recall". In: *Proceedings of the 32nd Annual Conference of the Cognitive Science Society*, pp. 2188–93.

Churchland, Patricia S and Terrence J Sejnowski (2016). *The computational brain*. MIT press.

Dancy, Christopher L et al. (2015). "Using a cognitive architecture with a physiological substrate to represent effects of a psychological stressor on cognition". In: *Computational and Mathematical Organization Theory* 21.1, pp. 90–114.

DeWolf, Travis et al. (2016). "A spiking neural model of adaptive arm control". In: *Proceedings of the Royal Society B* 283.48. DOI: 10.1098/rspb.2016.2134. URL: http://dx.doi.org/10.1098/rspb.2016.2134.

Duggins, Peter et al. (2017). "The effects of guanfacine and phenylephrine on a spiking neuron model of working memory". In: *Topics in cognitive science* 9.1, pp. 117–134.

Eliasmith, Chris and Charles H Anderson (2003). *Neural engineering: Computation, Representation, and Dynamics in Neurobiological Systems*. MIT press.

Eliasmith, Chris et al. (2012). "A large-scale model of the functioning brain". In: *Science* 338.6111, pp. 1202–1205.

Franowicz, Jenna S et al. (2002). "Mutation of the α2A-adrenoceptor impairs working memory performance and annuls cognitive enhancement by guanfacine". In: *The Journal of Neuroscience* 22.19, pp. 8771–8777.

Georgopoulos, Apostolos P et al. (1982). "On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex". In: *The Journal of Neuroscience* 2.11, pp. 1527–1537.

Goldman-Rakic, PS (1995). "Cellular basis of working memory". In: *Neuron* 14.3, pp. 477–485.

Gunzelmann, Glenn et al. (2009). "Sleep deprivation and sustained attention performance: Integrating mathematical and cognitive modeling". In: *Cognitive Science* 33.5, pp. 880–910.

Hermans, Erno J et al. (2014). "How the amygdala affects emotional memory by altering brain network properties". In: *Neurobiology of learning and memory* 112, pp. 2–16.

Izhikevich, Eugene M (2003). "Simple model of spiking neurons". In: *IEEE Transactions on neural networks* 14.6, pp. 1569–1572.

Levy, Florence (2008). "Pharmacological and therapeutic directions in ADHD: Specificity in the PFC". In: *Behavioral and Brain Functions* 4.1, p. 1.

Lewis, Penelope A and R Chris Miall (2006). "Remembering the time: a continuous clock". In: *Trends in Cognitive Sciences* 10.9, pp. 401–406.

London, Michael and Michael Häusser (2005). "Dendritic computation". In: *Annu. Rev. Neurosci.* 28, pp. 503–532.

Magee, Jeffrey C (1999). "Dendritic Ih normalizes temporal summation in hippocampal CA1 neurons". In: *Nature Neuroscience* 2.6, pp. 508–514.

Mao, Zheng-Mei, Amy FT Arnsten, and Bao-Ming Li (1999). "Local infusion of an α-1 adrenergic agonist into the prefrontal cortex impairs spatial working memory performance in monkeys". In: *Biological Psychiatry* 46.9, pp. 1259–1265.

Markram, Henry et al. (2015). "Reconstruction and simulation of neocortical microcircuitry". In: *Cell* 163.2, pp. 456–492.

Nolan, Matthew F et al. (2004). "A behavioral role for dendritic integration: HCN1 channels constrain spatial memory and plasticity at inputs to distal dendrites of CA1 pyramidal neurons". In: *Cell* 119.5, pp. 719–732.

Partridge, Lloyd D (1966). "A possible source of nerve signal distortion arising in pulse rate encoding of signals". In: *Journal of theoretical biology* 11.2, pp. 257–281.

Pessoa, Luiz (2008). "On the relationship between emotion and cognition". In: *Nature Reviews Neuroscience* 9.2, pp. 148–158.

Phelps, Elizabeth A (2004). "Human emotion and memory: interactions of the amygdala and hippocampal complex". In: *Current opinion in neurobiology* 14.2, pp. 198–202.

Plate, Tony A (1995). "Holographic reduced representations". In: *IEEE Transactions on Neural networks* 6.3, pp. 623–641.

Poolos, Nicholas P, Michele Migliore, and Daniel Johnston (2002). "Pharmacological upregulation of h-channels reduces the excitability of pyramidal neuron dendrites". In: *Nature Neuroscience* 5.8, pp. 767–774.

Ramos, Brian P et al. (2006). "α2A-adrenoceptor stimulation improves prefrontal cortical regulation of behavior through inhibition of cAMP signaling in aging animals". In: *Learning & Memory* 13.6, pp. 770–776.

Ritter, Frank E et al. (2012). "CoJACK: A high-level cognitive architecture with demonstrations of moderators, variability, and implications for situation awareness". In: *Biologically Inspired Cognitive Architectures* 1, pp. 2–13.

Romo, Ranulfo et al. (1999). “Neuronal correlates of parametric working memory in the prefrontal cortex”. In: *Nature* 399.6735, pp. 470–473.

Sanders, Laura (2013). “Mind & brain: model brain mimics human quirks: computer simulation turns decisions into plans for action”. In: *Science News* 183.1, pp. 13–13.

Scahill, Larry et al. (2014). “A placebo-controlled study of guanfacine in the treatment of children with tic disorders and attention deficit hyperactivity disorder”. In: *American Journal of Psychiatry*.

Silver, David et al. (2016). “Mastering the game of Go with deep neural networks and tree search”. In: *Nature* 529.7587, pp. 484–489.

Singh, Ray and Chris Eliasmith (2006). “Higher-dimensional neurons explain the tuning and dynamics of working memory cells”. In: *The Journal of Neuroscience* 26.14, pp. 3667–3678.

Stewart, Terrence C, Xuan Choo, and Chris Eliasmith (2010). “Dynamic behaviour of a spiking model of action selection in the basal ganglia”. In: *Proceedings of the 10th international conference on cognitive modeling*. Citeseer, pp. 235–40.

Stewart, Terrence C and Chris Eliasmith (2011). “Neural cognitive modelling: A biologically constrained spiking neuron model of the Tower of Hanoi task”. In: *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*. Citeseer, pp. 656–61.

Stuart, Greg and Nelson Spruston (1998). “Determinants of voltage attenuation in neocortical pyramidal neuron dendrites”. In: *Journal of Neuroscience* 18.10, pp. 3501–3510.

Wang, Min et al. (2007). “α2A-adrenoceptors strengthen working memory networks by inhibiting cAMP-HCN channel signaling in prefrontal cortex”. In: *Cell* 129.2, pp. 397–410.

Zhu, J Julius (2000). “Maturation of layer 5 neocortical pyramidal neurons: amplifying salient layer 1 and layer 4 inputs by Ca2+ action potentials in adult rat tuft dendrites”. In: *The Journal of Physiology* 526.3, pp. 571–587.