Artificial Neural Networks for Nonlinear System Identification of Neuronal Microcircuits

Thesis by
Dawna Paria Bagherian

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2021
Defended May 14, 2021

Dawna Paria Bagherian
ORCID: 0000-0003-4465-552X

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. 1745301. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. This work was also funded by a cloud computing grant from Amazon Web Services in collaboration with the Information Science and Technology initiative at the California Institute of Technology. This work was also supported by the Simons Collaboration on the Global Brain (grant 543015 to Markus Meister).

This work is dedicated to my parents, Saeedeh and Ali, my brother Pouya, my wonderful partner, James, and every single member of my enormous extended family. Their unconditional love and support through this PhD and my entire academic journey made all of it possible.

Thank you to my faculty committee: Markus Meister and Yisong Yue for their sage and patient advising on a project that took six long years to complete, and to Doris Tsao and Pietro Perona for offering guidance and asking thought-provoking questions as this work took shape.

I would like to acknowledge James Gornet and Jeremy Bernstein for their work on the theoretical exploration in Chapter 3, and Yu-Li Ni for his assistance with the retinal dissections that gave rise to the datasets in Chapter 5. Thank you to Dr. James Parkin for providing original illustrations of retinal neurons.

I would also like to thank my academic mentors from previous institutions whose belief in me propelled me to this point: Dr. Noah Prince, Dr. Erik van Erp, and Dr. Michael Sidorov. Your mentorship has molded how I think as a scientist, and I am so grateful for everything you taught me.

The Socialists of Caltech provided a home for me on campus as I completed the last year of my PhD. Thank you to every member of that organization for teaching me so much, and for all the ways they give themselves to their community every day. I couldn't have asked for a better group of role models than all of you. Ashay, Jane, Charles, Sean, Bee, Kriti, Zoya, Aaron, Arian, Peishi, Ollie, and everyone else in SoC, I will forever be working to live up to the generosity, dedication, brilliance, and compassion that you have displayed in the short time I've known you. Thank you for accepting me into the fold and bringing so much light into my life.

Finally, I want to thank my labmates. Dr. Kyu Hyun Lee, thank you for all of the pep talks and tough love. When I grow up, I want to be just like you. The soon-to-be Dr. Zeynep Turan, thank you for being the greatest friend a person could ask for. I still can't believe how lucky I was to sit next to you every day in the office. Alvita Tran, thank you for being so supportive and bringing so much fun to my time at Caltech. Sarah Sam, thank you for making me laugh and inspiring me to keep working to improve our community. I am grateful to all of you for making the past seven years such a blast, for supporting me through the darkest moments of this work, and for being my family away from home.

ABSTRACT

This thesis explores the application of artificial neural networks (ANNs) to nonlinear system identification. We use neuronal microcircuits in the retina as a testbed for our technique, which relies upon the marriage of partial anatomical information with large electrophysiological datasets. Rather than a typical application of machine learning, our primary goal is not to predict the output of retinal circuits, but rather to uncover their structure. We begin with a theoretical exploration in a toy problem and provide a proof of unique identifiability under a specific set of conditions. We then perform empirical simulations in a number of different circuit architectures and explore the space of constraints and regularizers to demonstrate that this technique is feasible in a hyperparametric regime that lends itself well to neuroscience datasets. We then apply the technique to mouse retinal datasets and show that we can both recover known biological information as well as discover new hypotheses for biological exploration. We end with an exploration of active stimulus design algorithms to distinguish between circuit hypotheses.

TABLE OF CONTENTS

Acknowledgements iii
Abstract v
Table of Contents vi
Chapter I: Introduction 1
1.1 Neurobiology of the retina 2
1.2 LN-LN cascade models can replicate the responses of retinal neurons 3
1.3 Past applications of deep learning to visual neuroscience 4
1.4 System identification and structure recovery 5
Chapter II: A method for nonlinear system identification of neuronal circuits using artificial neural networks 7
2.1 Biologically-inspired regularization of artificial neural networks . . . 7
Chapter III: Results I: A theoretical exploration of circuit identifiability under infinite data conditions 16
3.1 Related work 16
3.2 An identifiability theorem 17
3.3 Theoretical analysis 18
Chapter IV: Results II: System identification of simulated microcircuits . . . 22
4.1 Simulating biologically inspired feedforward circuits 22
4.2 Simulating retinal circuits 22
4.3 Case Study: The ON/OFF direction-selective ganglion cell 25
4.4 Implementing an “address book” constraint 53
4.5 Study of the parametric regime in which system identification is feasible 60
4.6 Sign constraints are necessary for system identification 67
4.7 Sparsity regularization is helpful for identification of some architectures 67
4.8 A heuristic for estimating the dimensionality of an unknown circuit . 72
Chapter V: Results III: Case study on data collected from mouse alpha retinal ganglion cells 78
5.1 The four subtypes of the alpha ganglion cell 78
5.2 Data collection using multi-electrode array 78
5.3 Data preprocessing 79
5.4 A system identification problem for alpha cell circuitry 80
5.5 Fitting an ANN to alpha cell data 84
5.6 Recovery of known biological information about the alpha cell 84
5.7 The ANN correctly identifies the circuitry of the transient OFF alpha cell and suggests an additional pathway 87
5.8 System identification as a tool for hypothesis selection 89
5.9 Summary 91
Chapter VI: Results IV: An exploration of algorithms for input selection in system identification 92

6.1 An adversarial stimulus generation algorithm 93
6.2 An algorithm to select between two classes of stimuli 95
6.3 An algorithm to maximize output of the circuit 97
6.4 An algorithm for optimal stimulus design given competing circuit hypotheses 102
6.5 Summary 103
Chapter VII: Conclusion and future directions 105
Bibliography 106

Chapter 1

INTRODUCTION

Much of modern biology is devoted to the art of untangling complex organic systems into clearly defined circuit diagrams. For instance, neuroscientists aim to understand how brain tissue computes in the same way we understand an electronic circuit [43, 71], and systems biologists study the cascading effects of gene expression or signal transduction networks by drawing molecular circuits [42]. In order to call a biological circuit "solved," we must uncover (1) the components of the circuit (neuronal types, genes or proteins), (2) how these components are connected (synapses or molecular interactions), and (3) how these components act on an input signal (intracellular processing and synaptic weights or molecular rate constants and binding affinities).

The field of biotechnology has exploded in recent years, yielding increasingly powerful methods for collecting neuronal data. These include genetic tools for labeling neurons [4], high throughput methods for gene product detection [81], dense electrode arrays for recording action potentials from living cells [37], genetically encoded fluorescent activity reporters [14], and complex surgical procedures to provide access to specific brain regions [21]. This progress has been swift and impressive, but it remains impossible to collect a complete dataset that can simultaneously measure the activity and the connectivity at every node in a biological system. Therefore, it falls to the biologists to carefully reason about incomplete (but increasingly rich) datasets and thereby attain a holistic understanding of the biological system.

In this work, we design and test a method for fine-grained neural system identification of biological circuits using rich but incomplete biological data. The technique aims to infer both the synaptic weights and the local computations performed by neurons within a feed-forward circuit by leveraging modern artificial neural networks (ANNs) and their associated optimization tools. For data collection, we assume the ability to apply stimuli to the sensory neurons in the input layer of the circuit, and measure the responses of the output neurons. We validate our approach in the mouse retina, where such data are readily available [56].

Our method accomplishes this using an overparameterized ANN that includes as

a sub-network an exact neuron-to-neuron and synapse-to-synapse correspondence to a mechanistic model of the true biological circuit. When fitting the input/output data collected from the biological circuit, this method operates much like standard supervised learning, thus benefiting from modern deep learning techniques. This technique also applies the standard sparsification technique of ℓ_1 regularization to prune the ANN and thereby match the synaptic patterns of the biological circuit. In order to do this, the method incorporates common forms of neuroscience domain knowledge in a systematic way to achieve more data-efficient learning by heavily constraining the space of circuits being searched.

We explore applications of this method to biological circuits both theoretically and empirically. We also showcase the practicality of the technique in a case study of circuits in the mouse retina. We measure responses to visual input from alpha retinal ganglion cells (alpha RGCs) in a live mouse retina using a multi-electrode array [56]. Using our method, we show that it is possible to (1) recover known results in retinal biology, and (2) guide experiment design to test biologically plausible hypotheses regarding the structure of an unknown retinal circuit.

1.1 Neurobiology of the retina

The retina is made up of five major classes of neurons: photoreceptors, horizontal cells, bipolar cells, amacrine cells, and ganglion cells. These are organized into five layers: the outer nuclear layer (ONL), outer plexiform layer (OPL), inner nuclear layer (INL), inner plexiform layer (IPL) and ganglion cell layer (GCL). The photoreceptors, which come in two major types (rods and cones), are densely packed together to form the outer nuclear layer (Fig.1.1). These are the light sensors of the retina. The rod type is mostly responsible for low-light vision, while the cone type is more active in bright light conditions. Most mammals have two types of cone photoreceptors, which are known as the s-cone and m-cone. These are sensitive to short wavelength (blue) light and medium wavelength (green) light respectively. Some primates, including humans, have a third cone type known as the l-cone, which is sensitive to long wavelength (red) light. The bipolar cell bodies live in the inner nuclear layer, and are the main excitatory interneuron of the retina (Fig.1.1). Every cone photoreceptor connects to every type of bipolar cell. The rod, however, piggybacks onto the cone circuitry via a single bipolar cell type known as the rod bipolar cell. Bipolar cells come in both ON and OFF varieties (which respond to light increments or decrements respectively) and release glutamate from their axons, which form excitatory synapses with both amacrine and ganglion cells.

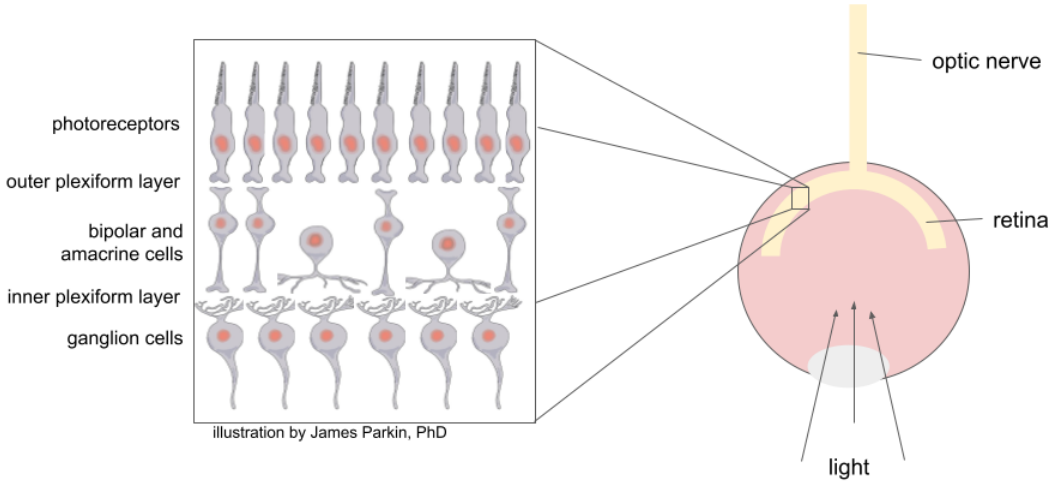

Figure 1.1: Illustration of the structure of the retina.

The horizontal cells have their somas in the inner nuclear layer, but receive signals from photoreceptors and provide lateral inhibition to both photoreceptors and bipolar cells. Amacrine cell bodies are also located in the inner nuclear layer, though their processes are located in the inner plexiform layer, and provide glycinergic and GABA-ergic inhibition (and in some rare cases cholinergic excitation) to the bipolar-ganglion synapse, as well as to each other. It is becoming more and more clear that amacrine cells already do a great deal of feature selection in the retina.

The ganglion cell layer is composed of ganglion cells, which are typically considered to be the "feature-detecting" cells, meaning that they typically respond selectively to specific features in the visual scene, due to the circuitry they form with upstream interneurons. The ganglion cell axons form the optic nerve, and transfer visual information to the brain (Fig.1.1). Ganglion cells fire action potentials, and encode visual information in both spike latency as well as firing rate.

1.2 LN-LN cascade models can replicate the responses of retinal neurons

Based on our understanding of the signal flow through the classes of neurons just described, it is well established that the activity of retinal ganglion cells can be well modeled with an LN-LN Cascade Model. LN stands for Linear-Nonlinear. A single LN unit takes the input and passes it through a linear convolutional filter, then through a static nonlinearity such as a ReLU or sigmoid [6]. The cascade model class combines two or more layers of LN model units to form a network. The first layer is often used to model the computation of the entire outer retina, including photoreceptors, horizontal cells, and bipolar cells, and the output of this

layer is meant to mimic the glutamate release of retinal bipolar cells [6]. The next layers model amacrine and ganglion cells. Stacking LN units in series can not only successfully predict the output of the ganglion cell layer, but also provides a mechanistic explanation for the type of feature selectivity often seen in retinal circuits [5, 25, 28, 33, 58, 60, 66].

1.3 Past applications of deep learning to visual neuroscience

While cascade models have worked well for quite some time and provided a language with which to describe myriad complex retinal computations, the recent explosion in machine learning research has naturally brought with it a burst of efforts to leverage the technology to model neuronal circuits. Neuroscience and deep learning have long been intertwined, each field alternately driving progress in the other. For instance, convolutional neural networks (CNNs) are originally inspired by biology [24], and were designed to have computational characteristics of the vertebrate visual system. More recently, researchers have endeavored to understand biological, and specifically neuronal, systems using artificial neural networks [1, 43, 53, 54, 70, 71, 80, 92]. Notable examples include work by McIntosh et al on training a standard CNN architecture on data collected from the output neurons of salamander retina. This group found that retinal interneuron-like characteristics emerged in intermediate layers [54]. The same group later determined that this artificial network employed circuit motifs like the ones in retina to perform complex computations, and was able to replicate many nonlinear retinal phenomena [51]. In 2019, Tanaka et al extended this work to extract mechanistic understanding from the trained CNN, which enabled the identification of connectivity motifs that underlie the network's ability to replicate a biological phenomenon [80]. The Yamins group has also studied ANNs as a model class for the visual system, asking whether training these networks on a visual behavioral task gives rise to the same types of computational strategies employed in mammalian cortex [91]. In 2018, Abbasi-Asl et al developed DeepTune, a framework to extract stimuli that demonstrate the tuning of individual units in an ANN model of primate visual cortex area V4 [1]. Such work has been largely restricted to coarse-grained analyses that characterize computation of regions of the brain, rather than that of individual neurons interacting within a circuit, or to the revelation of microcircuit motifs within trained CNNs that mirror those found in biology.

In contrast, this work focuses on the possibility of performing fine-grained circuit modeling where the neurons in the learned neural network have a one-to-one cor-

respondence with individual biological units. Preliminary efforts in this direction have been made for a smaller, more granular model of one layer of retinal synapses [70], but to the best of our knowledge, this work covers only parameter estimation, not hypothesis selection, and the general problem has not been studied theoretically or empirically via systematic simulations, nor has it been tested on real data from deeper circuits. The concept of using an artificial neural network as a 1:1 model for a retinal microcircuit started as a long-shot idea. As we tested it in simulation and were continually surprised by its success. In the work that follows, we will describe the evolution of this idea from a set of preliminary simulations in a toy model to a full-blown application of the technique to mouse retinal data. Along the way we will explore the theoretical underpinnings of the method and play with some interesting algorithms for optimal experimental design. We aim to convince the reader that ANNs can, in fact, be a useful tool to a circuit neuroscientist if partial prior knowledge is appropriately leveraged, and more generally, that the field of nonlinear system identification might benefit as a whole from further exploration of this technique.

1.4 System identification and structure recovery

Broadly speaking, system identification is the use of statistical methods to build mathematical models of systems from measured data [50]. Nonlinear system identification is a key tool in modeling dynamical systems, which includes early work on (coarse-grained) neural system identification for control systems [45]. A fundamental issue that arises is *identifiability* [11]—that is, when can one uniquely recover the true system. This affects the reliability of the results for downstream scientific analysis. Identifiability has been studied theoretically both in the context of nonlinear neural networks [2, 20, 67, 79, 87] and in "linear networks" in the context of matrix factorization [16, 34, 46]. In Section 3.3, we shall introduce and discuss some of the theoretical concepts that have a bearing on identifiability.

A related concept to system identification is support recovery, where the main goal is to discover which parameters of a model are non-zero [31, 48, 72, 75, 83], and which can be thought of as a subgoal of full system identification. Support recovery is commonly studied in sparse linear systems that have few non-zero parameters. Biological neural networks are also sparse [28, 57], but are nonlinear multi-layer models for which theoretical results in sparse linear support recovery do not directly apply. Nonetheless, our approach takes inspiration from the classic Lasso [83] for sparse linear regression in order to reliably estimate sparse neural networks

with limited training data; an interesting future direction would be to establish *sparsistency* results [48] for this type of approach.

Another related concept is structure discovery in (causal) graphical models [52, 69, 74, 84, 94]. These models are typically composed of nodes connected by edges. The edges can be directed (meaning that two nodes only communicate in a single direction) or undirected (meaning that signal flows bidirectionally between the two nodes.) A common problem focuses on a network with directed edges, which is acyclic, meaning that there are no "loops" created by the edges. This network is used to generate a dataset, and the problem is that of recovering the structure of the network (i.e., which edges are non-zero). This setting is very similar to ours with a few differences. First, the goal of structure discovery in graphical models is to recover the direction of the edges in addition to the weights, whereas in our setting all the edge directions are known a priori. Second, the training data for structure discovery is typically fully observed in terms of measuring every node in the network, whereas for our setting we only observe the inputs and final outputs of the network (and not the measurements of nodes in the hidden layers). Like in sparse system identification, most prior work in structure discovery of graphical models is restricted to the linear setting.

Chapter 2

A METHOD FOR NONLINEAR SYSTEM IDENTIFICATION OF NEURONAL CIRCUITS USING ARTIFICIAL NEURAL NETWORKS

2.1 Biologically-inspired regularization of artificial neural networks

Based on the huge amount of data collected about the retina over the past several decades, we designed a technique that marries incomplete anatomical data with large physiological datasets to generate microcircuit understanding of retinal circuits. Each form of biological knowledge is incorporated into the structure of an artificial neural network (ANN) which is designed so that every artificial neuron corresponds to a single biological neuron. The ANN is trained on electrophyisological datasets recorded from the output layer of the retina, the ganglion cell layer, in response to a set of visual stimuli. This chapter will describe each form of regularization or constraint in detail and lay out a road map for training these ANNs. The following chapters will demonstrate this technique in theory, simulation, and in application to real retinal data.

Artificial neurons as an analogy for biological neurons

Retinal neurons have been very thoroughly studied and modeled over the past several decades. One of the most commonly used models for single retinal neurons is called the Linear-Nonlinear (LN) model. This model consists of two steps of computation. The input to the neuron is convolved with a spatiotemporal filter, representing the linear receptive field of the neuron. This simulates the integration of input signal by the neuron. The second step is to pass the output of the convolution through a static nonlinearity (usually a ReLU or sigmoid). This simulates the spike generation or vesicle release process, which is nonlinear, and usually involves some type of thresholding [6, 22, 38]. A common variant on this model is the Linear-Nonlinear-Poisson (LNP) model, which uses a Poisson random variable to generate individual spike times from the output of the nonlinearity. This produces actual spike trains as the output of the model, rather than the continuous output from the LN model [38].

Figure 2.1 shows an example of the LN model for two simulated bipolar cell types. Based on the shape of the temporal convolutional filter used in the linear part of the model, we can simulate bipolar cells that have sustained or transient responses to

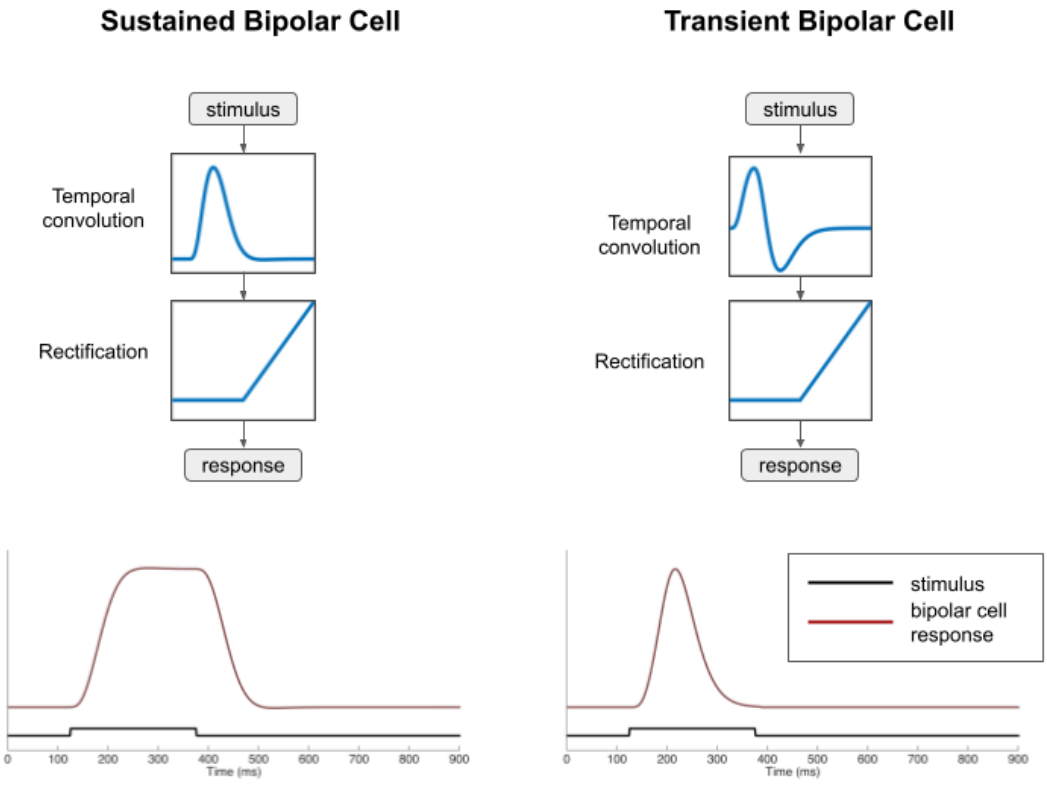

Figure 2.1: Illustration of the LN model for a retinal bipolar cell. The time-varying stimulus is convolved with a temporal filter and passed through a ReLU nonlinearity to produce a time-varying output. Based on the shape of the temporal filter, a sustained or transient response (red lines) can be generated to a 250 ms light flash stimulus (black lines).

light stimuli.

Biologically-derived LN model parameters for bipolar cells

Because the retina has been so extensively studied, many datasets of retinal neuron activity are publicly available. One of these, published by Franke et al in 2017, contains the responses of bipolar cells to a battery of visual stimuli. The responses are recorded as glutamate output rate, using the glutamate indicator iGluSnFr, which fluoresces to indicate the concentration of glutamate in the synapse. The neurons in the dataset were also labeled by type (there are 14 types of bipolar cells in the retina, and these are classified by function and anatomy) [22]. We were able to use this dataset to fit LN models to bipolar cell responses and average over many neurons within the same type, to generate a single set of LN model parameters for each of the fourteen bipolar cell types (Fig. 2.2.)

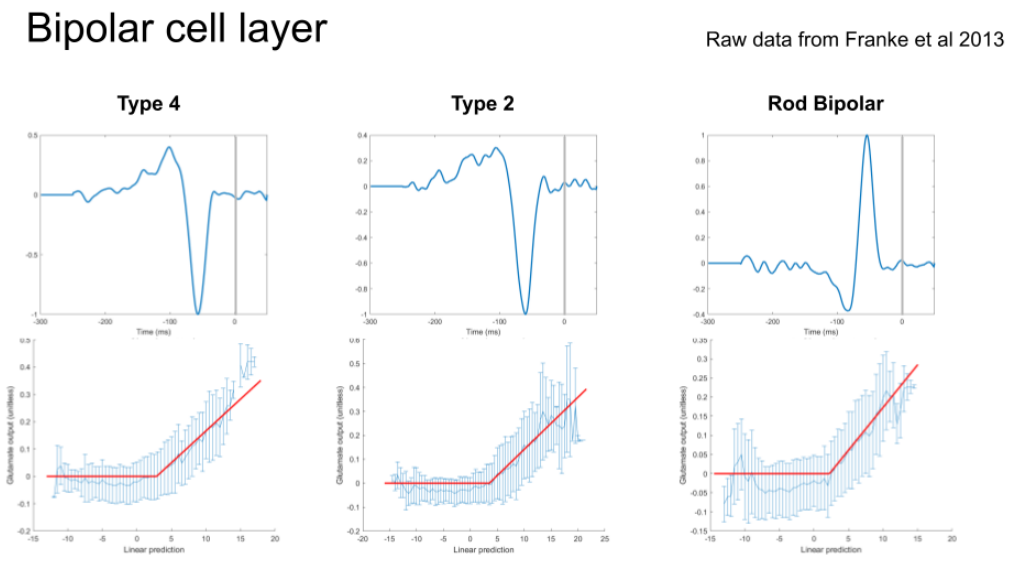

Figure 2.2: Example fits of LN models to three types of bipolar cell data from [22]. Spike-triggered averaging is used to compute the temporal filters and then a static ReLU nonlinearity is fit to the output. Each column is the fit to an individual neuron, whose type classification is listed at the top of the column.

In order to compute these models, we began by deconvolving the iGluSnFr traces with a kernel representing the time course of iGluSnFr fluorescence [36]. Thus, the deconvolved traces represented the actual glutamate release, without the dynamics of the glutamate indicator. These models could be plugged directly into an ANN and fixed in place or allowed to vary slightly over the course of training. This would depend on the assumption that the light conditions in the experiment that generated the training dataset matched the conditions in [22], and therefore the LN model parameters can be shared. It is known that due to changes in ambient light conditions, these parameters may change [6]. In the work that follows, these filters were never fixed in place in our ANNs, but they did occasionally serve to select initialization parameters for the bipolar cell layer.

Neural net layers as an analogy for classes of retinal neurons

Each layer of the ANN can be thought of as representing one of the five classes of retinal neurons. Photoreceptors and horizontal cells are not explicitly modeled in this technique. We consider them as reporting the visual scene in a linear fashion to bipolar cells, and therefore wrap them into the model of the bipolar cell as a linear-nonlinear unit. The first layer of the ANN after the stimulus represents the

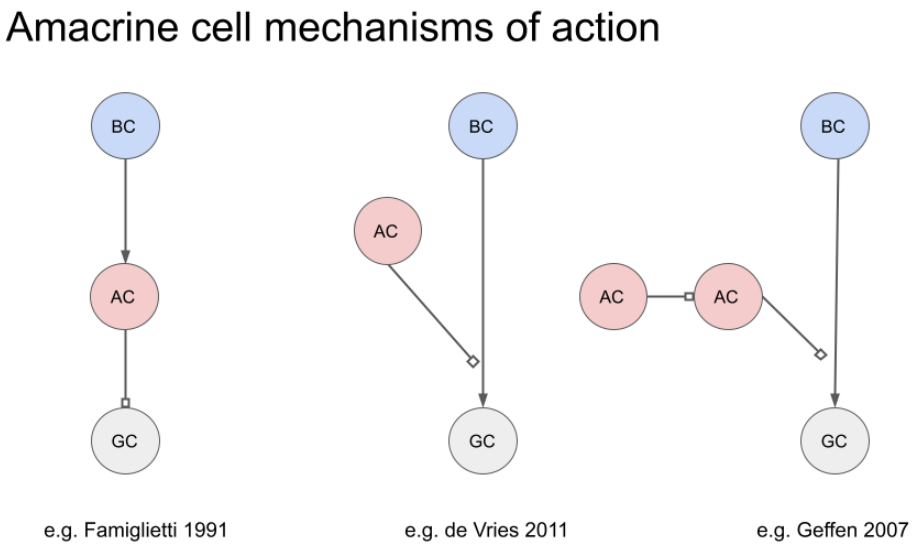

Figure 2.3: Schematics demonstrating three different mechanisms of action employed by amacrine cells in retinal circuits. Left: The amacrine cell is excited by a bipolar cell, and in turn inhibits a ganglion cell [19]. Center: An amacrine cell provides presynaptic inhibition at the bipolar cell to ganglion cell synapse [15]. Right: An amacrine cell provides lateral inhibition to another amacrine cell which presynaptically inhibtis a bipolar to ganglion cell synapse, as in [25].

bipolar cell layer. It is often subdivided into bipolar cells of various types. All cells belonging to a given type share some parameters such as their temporal receptive field or the threshold of their nonlinearity.

The second layer represents the amacrine cells, the inhibitory interneurons of the retina. Amacrine cells in the retina have various mechanisms of action (Fig. 2.3) including lateral reciprocal inhibition, and inhibition to the bipolar ganglion synapse either pre- or post-synaptically (e.g. [15, 19, 25]).

The use of an inner plexiform layer "address book" for connectivity between cell types as a constraint on the weights of the network

To further enhance reliable learning, we can constrain the weights of the network in two ways, both of which lead to a convex set of allowable weights W. First, existing domain knowledge constrains many connections to zero weight. In the retina, such connectivity constraints are implemented by the precisely organized anatomy of neurons. The so-called inner plexiform layer (IPL) is a meshwork of synapses between different unit types (bipolar, amacrine, and ganglion cells). Each

type sends its neural process into a distinct lamina of the IPL, and neurons may connect to each other only if they co-stratify in at least one lamina. This implements an "address book" of allowable connectivity between cell types in the network [73]. Using anatomical studies in the literature [7, 22, 26, 29, 40, 58, 85, 93], we have compiled such an address book for 36 retinal cell types (Fig. 2.4).

We constructed a "retinal address book" based on this idea. Because these anatomical data come from a number of studies with varying forms and thoroughness of anatomical data published, this address book was constructed by painstaking manual inspection. For example, different authors divided the IPL into different numbers of sublaminae, and set their boundaries in different locations.

We selected a sublamination scheme that was compatible with each of these. In this scheme, the mouse IPL was divided into six sublaminae: outer marginal (0.0-0.28 normalized depth), outer central (0.28-0.47 normalized depth), inner central (0.47-0.65 normalized depth), inner marginal (0.65-1.0 normalized depth), and the ON and OFF ChAT bands (defined by limits of choline acetyltransferase (ChAT) expression). With this sublamination scheme, we were able to neatly define a binary stratification profile for each cell type under study, and to therefore create a binary address book (Fig. 2.4).

One could imagine, however, a future in which stratification profiles are given in units of normalized IPL depth, with confidence intervals or some other measure of uncertainty, for every retinal cell type. From such a dataset, one could replace the binary entries of this table with a continuously varying measure of co-stratification and uncertainty.

Regularization to constrain the sign of weights of excitatory and inhibitory synapses

Another strong constraint on the weights regards their sign. Each synapse in the network can be identified via prior knowledge to be inhibitory (nonpositive) or excitatory (nonnegative) [28, 65, 88]. For example, bipolar cells can send only excitatory input to amacrine and ganglion cells. In fact, for the retinal circuits we study in in this work, we have sufficient knowledge to sign constrain every weight. More generally, in other domains such as genetic circuits, the current scientific understanding includes information about whether specific classes of genes up- or down-regulate other genes [3].

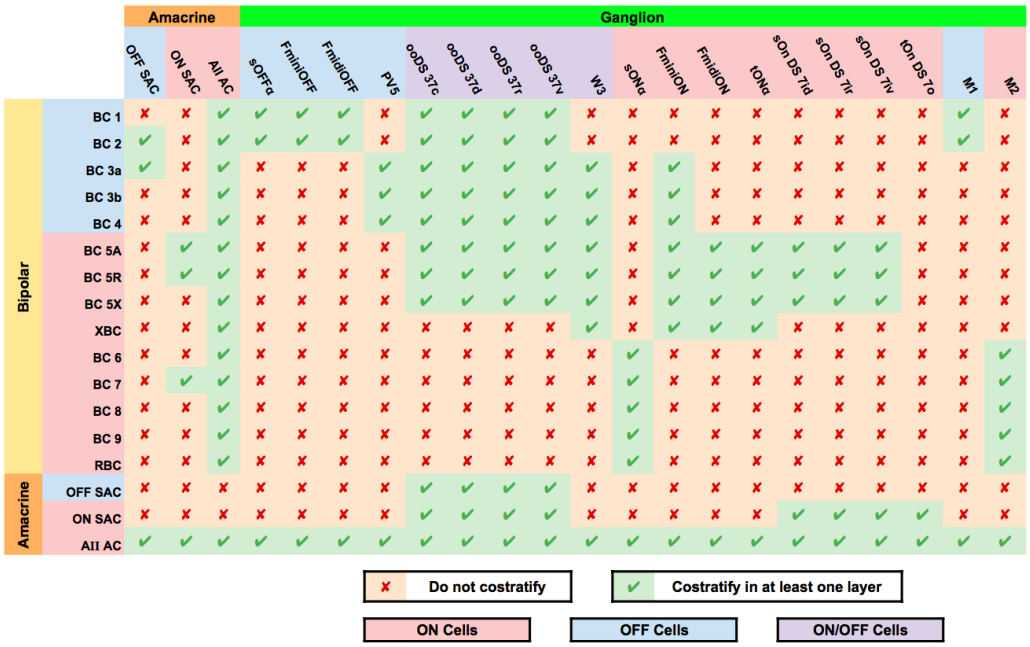

Figure 2.4: "Address book" of mouse retinal inner plexiform layer, showing costratification of 36 cell types. This table was constructed with data from [7, 22, 26, 29, 40, 58, 85, 93].

Low dimensional parameterization of convolutional filters

Throughout this work, we reduce the number of free parameters in the trained ANN by reparameterizing the first convolutional layer. This layer takes a spatiotemporal convolution of the stimulus with a bank of biologically inspired filters. These filters have a stereotyped form, and can be parameterized in many ways. The first way is to describe the filter as a "two-bump function" of the form

$$\begin{aligned} f(t) &= f_+(t) - f_-(t) \\ &= \left(c_+(t+a_+)^{n_+} e^{-b_+(t+a_+)}\right) - \left(c_-(t+a_-)^{n_-} e^{-b_-(t+a_-)}\right) \end{aligned} \tag{2.1}$$

An example of a subspace of filters spanned by this parameterization is shown in Fig. 2.5. A similar parameterization is used in [66]. In this work, we often fix a_+, a_-, b_+, and b_- in place, and ask the network to only learn c_+ and c_-. This is done in the following way: Let s represent a stimulus. Let $f = af_+ - bf_-$ be a "two-bump" filter as described above. Then, note that by linearity of the convolution operator, we can write

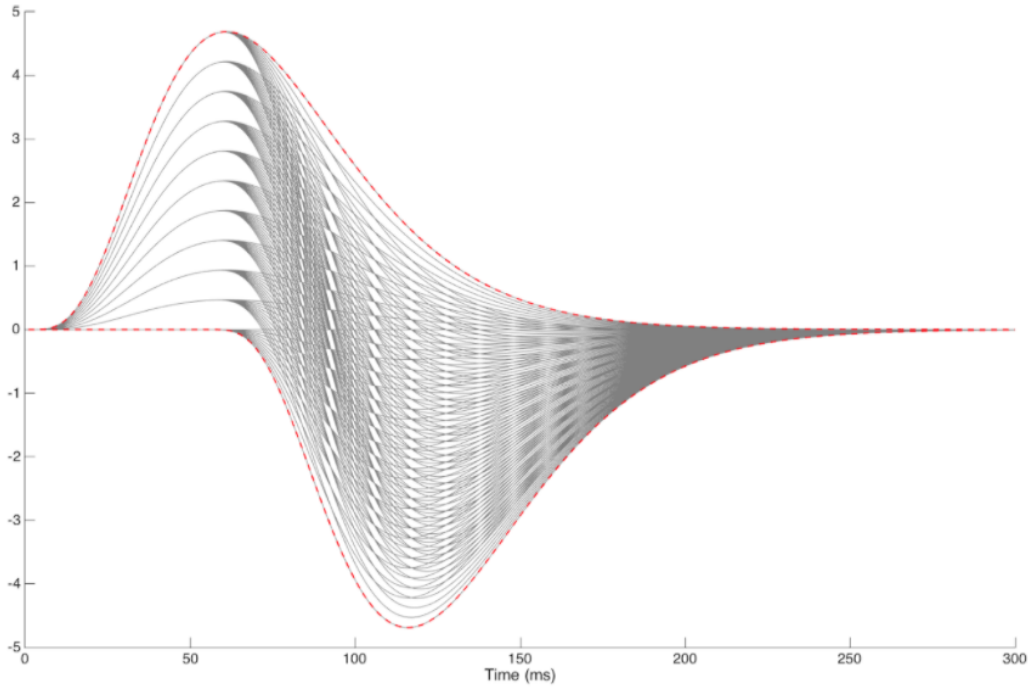

Figure 2.5: Illustration of the two-bump function convolutional filter basis. Red dashed lines depict each of the two-bump functions, while black lines show various linear combinations of these basis functions to illustrate the range of shapes that can be achieved by varying only two parameters.

$$s * f = a(s * f_+) - b(s * f_-). \tag{2.2}$$

Let $y = F(s)$ represent the output of a network, F. Let $\{f_i\}_{i=1}^{N}$ be a more general "basis set" of convolutional filters.

Then we can describe the network by rewriting

$$y = F(s) = F'(\{s * f_i\}_{i=1}^{N}). \tag{2.3}$$

That is, we can make the input to the network the convolution of the stimulus with our set of basis filters, which do not change during training, rather than the raw stimulus. Then, all we need to learn is a set of weights on these basis convolutions. Therefore, there are N free parameters representing the input convolution, rather than having to learn the entire convolutional filter from scratch.

Another set of basis filters used frequently in this work (shown in Fig. 2.6) is taken from [38] and takes the form:

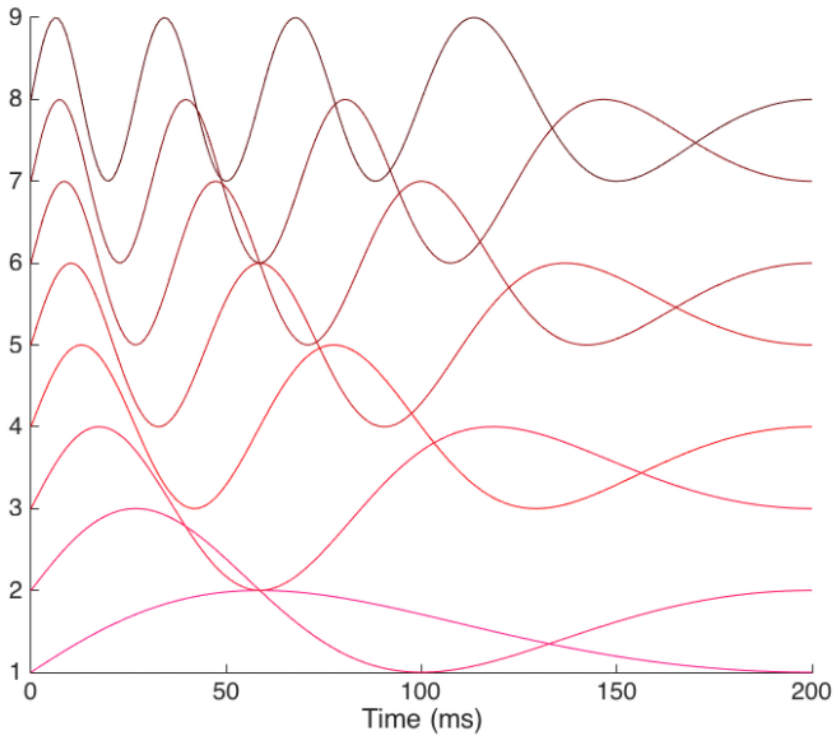

Figure 2.6: Illustration of the stretched sinusoid filters described in Eq 2.4. Each successive basis function is vertically shifted for visual clarity.

$$f_j(t) = \begin{cases} \sin\left(\pi j \left(2\frac{t}{\tau} - \left(\frac{t}{\tau}\right)^2\right)\right) & \text{for } 0 \leq t \leq \tau \\ 0 & \text{otherwise} \end{cases} \tag{2.4}$$

for $j = 1, ..., N$. When this filter basis is used, we typically let $N = 16$.

Removing degeneracy by fixing biases

In order to remove some degeneracy from the space of circuit models being searched, it was, in some cases, necessary to fix the biases in place during training. The toy example in figure 2.7 illustrates the need for this. In order to efficiently compute the structure recovery score, R, we needed to remove this type of degeneracy from the space of circuit structures being searched. When b is fixed in place during training, this degeneracy is removed.

Training procedure

The ANN is constructed using whichever constraints are relevant from the list above. Then, all weights are randomly initialized, and training proceeds with the standard

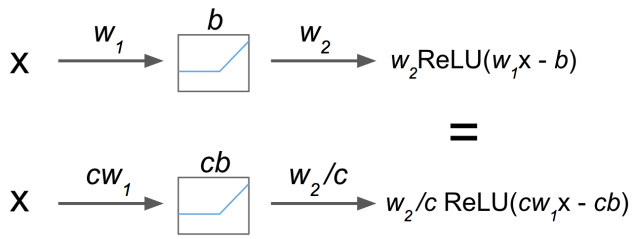

Figure 2.7: Two toy networks with differing parameter sets which produce the same output for an input, x.

squared loss function. That is, given training data pairs $(\mathbf{x}, \mathbf{y})$, and an ANN $f(\mathbf{x}; \mathbf{W})$, the loss function can be written as:

$$L(\mathbf{x}, \mathbf{y}, \mathbf{W}) = \sum_{(\mathbf{x},\mathbf{y})\in S} \|f(\mathbf{x}; \mathbf{W}) - \mathbf{y}\|^2. \tag{2.5}$$

In what follows, the optimization algorithms used for training were either Adam [41], Momentum [64], or standard gradient descent. An ℓ_1 regularizer was used unless otherwise indicated. Thus the objective function can be written as:

$$O(\mathbf{x}, \mathbf{y}, W, \lambda) = \sum_{(\mathbf{x},\mathbf{y})\in S} \|f(\mathbf{x}; \mathbf{W}) - \mathbf{y}\|^2 + \lambda|\mathbf{W}|. \tag{2.6}$$

Algorithm 1: An algorithm for fine-grained neural system identification

input: $S = \{(\mathbf{w}, y)\}$; // training set
λ ; // ℓ^1 regularization
$f(\cdot; \mathbf{W})$; // neural circuit
W ; // allowable weights
for $k = 1, \ldots, K$ **do**
Initialize a $\mathbf{W}$ randomly in W Solve $\mathbf{W}_k \leftarrow_{\mathbf{W}\in W} \sum_{(\mathbf{x},y)\in S} \|f(\mathbf{x}; \mathbf{W}) - y\|^2 + \lambda|\mathbf{W}|$;
// from the random initialization
return: $\mathbf{W}_1, \ldots, \mathbf{W}_K$

Algorithm 1 describes this process. The ANN is trained multiple times with multiple random initializations to produce a set of hypothesis circuit structures. Based on this set, the experimenter can confirm or rule out their hypotheses, and can make decisions about the allocation of future experimental resources.

Chapter 3

RESULTS I: A THEORETICAL EXPLORATION OF CIRCUIT IDENTIFIABILITY UNDER INFINITE DATA CONDITIONS

3.1 Related work

System identification and structure recovery

Broadly speaking, system identification is the use of statistical methods to build mathematical models of systems from measured data [50]. System identification is a key tool in modeling dynamical systems, which includes early work on (coarse-grained) neural system identification for control systems [45]. A fundamental issue that arises is *identifiability* [11]—that is, when can one uniquely recover the true system. This affects the reliability of the results for downstream scientific analysis.

Identifiability has been studied theoretically both in the context of nonlinear neural networks [2, 20, 67, 70, 79, 87] and in "linear networks" in the context of matrix factorization [16, 34, 46]. In Section 3.3, we introduce and discuss some of the theoretical concepts that have a bearing on identifiability. As stated before, our main goal is to provide a thorough theory-to-practice investigation grounded in real neuroscience modeling challenges.

A related concept is support recovery, where the main goal is to discover which parameters of a model are non-zero [31, 48, 72, 75, 83]. Support recovery can be thought of as a subgoal of full system identification. It is commonly studied in sparse linear systems that have few non-zero parameters. While biological neural networks are also sparse [28, 57], they are nonlinear multi-layer models for which theoretical results in sparse linear support recovery do not directly apply. Nonetheless, we show that, under suitable conditions, one can employ ℓ_1-regularized regression (that is commonly used for estimating sparse linear models [83]) to reliably estimate sparse neural networks with limited training data; an interesting future direction would be to establish *sparsistency* guarantees [48].

Another related concept is structure discovery in (causal) graphical models [52, 69, 74, 84, 94]. A typical setting is to recover the structure of a directed acyclic network (i.e., which edges are non-zero) that forms the causal or generative model of the data. This setting is very similar to ours with a few differences. First, the goal of structure discovery in graphical models is to recover the direction of the edges in

addition to the weights, whereas in our setting all the edge directions are known a priori. Second, the training data for structure discovery is typically fully observed in terms of measuring every node in the network, whereas for our setting we only observe the inputs and final outputs of the network (and not the measurements of nodes in the hidden layers). Like in sparse system identification, most prior work in structure discovery of graphical models is restricted to the linear setting.

3.2 An identifiability theorem

We formulate the problem of system identification in nonlinear feed-forward networks as follows. Consider a function $f(\mathbf{x}; \mathbf{W}, \mathbf{b})$ known as the *network*. The network is parameterized by an unknown weight vector $\mathbf{W}$ and bias $\mathbf{b}$. We assume that we can query the network's nonlinear input-output mapping:

$$\mathbf{x} \mapsto f(\mathbf{x}; \mathbf{W}, \mathbf{b}) \equiv \mathbf{y}. \tag{3.1}$$

That is, we may apply input stimuli $\mathbf{x}$ to the network and record the output. See Chapter 5 for details on collecting such data from the mouse retina.

Our goal is to recover the true $\mathbf{W}$ and $\mathbf{b}$ given such $(\mathbf{x}, \mathbf{y})$ queries. A first question that arises is whether recovering the true $\mathbf{W}$ and $\mathbf{b}$ is possible, even with infinite data—i.e., whether $\mathbf{W}$ and $\mathbf{b}$ are uniquely identifiable. Other questions include how to accurately recover $\mathbf{W}$ and $\mathbf{b}$ (or at least their non-zero support) given finite and noisy data, which data points to query, and what types of domain knowledge can aid in this process.

In practice, estimating the true $\mathbf{W}$ and $\mathbf{b}$ given training data is tackled as a regression problem. We are particularly interested in the case where $\mathbf{W}$ is sparse. The conventional way to encourage sparsity is to use ℓ_1-regularization [83], which we will also employ. We discuss in Chapter 4 practical considerations through extensive evaluation of simulated biological circuits.

Summary of Domain Knowledge. Our goal is to not only establish conditions where system identification of nonlinear feedforward networks is possible, but also that those conditions be practically relevant. Guided by neuroscience domain knowledge, we study circuits with the following properties:

(i) The nonlinearity of individual neurons is well understood, and can be well modeled using ReLUs or leaky ReLUs [6].

(ii) The dominant computation of the neural circuit is feedforward, which is true for circuits found in the retina [28, 88].

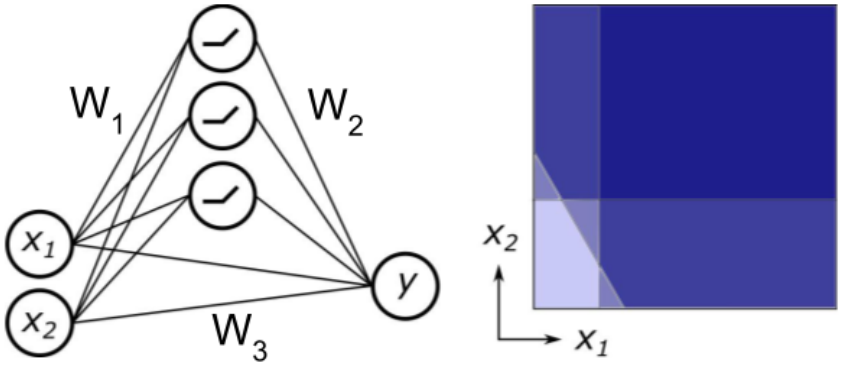

Figure 3.1: A one hidden layer ReLU network with skip connection (Eq 3.2). The nonlinearity transitions occur on hyperplanes in input space.

(iii) Many potential weights can be preemptively set to zero, due to the implausibility of neurons in various spatial configurations being connected to one another [17, 73].

(iv) All weights can be sign constrained (either non-negative or non-positive), due to knowing the inhibitory or excitatory behavior of each neuron [28, 65, 88].

Items (i) and (ii) above imply knowing the functional form (i.e., multi-layer perceptron with known number of layers, maximal number of neurons in each layer, and form of the nonlinearity). Item (iii) implies that one can reduce the number of free parameters in the model, thus easing the burden of learning (although still requiring high-dimensional sparse estimation). Item (iv) is perhaps the most interesting property, as it effectively constrains the parameters to be within a single known orthant. We show in Section 3.3 that the sign constrained condition is an important sufficient condition for proving identifiability, and we show empirically in Chapter 4 that ℓ_1-regularized regression can succeed in system identification on sign constrained networks and fail on unconstrained networks.

3.3 Theoretical analysis

Motivated by the neuroscience domain knowledge discussed Chapter 2, we now theoretically establish factors that govern fine-grained identifiability of neural networks—a topic that has been studied since the 1990s [2, 20, 79]. To develop the core ideas, it will help to consider the following neural network:

$$f(\mathbf{x}) := \mathbf{W_2} \max(\mathbf{0}, \mathbf{W_1 x} + \mathbf{b_1}) + \mathbf{W_3 x} + \mathbf{b_2}, \tag{3.2}$$

for weights $\mathbf{W_1} \in \mathbb{R}^{n_1 \times n_0}$, $\mathbf{W_2} \in \mathbb{R}^{n_2 \times n_1}$, $\mathbf{W_3} \in \mathbb{R}^{n_2 \times n_0}$ and biases $\mathbf{b_1} \in \mathbb{R}^{n_1}$, $\mathbf{b_2} \in \mathbb{R}^{n_2}$.

There are three reasons why this is a sensible network to consider. First, the network's relatively simple structure facilitates theoretical insight. Second, this network bears a resemblance to neural (sub-)circuits found in the retina (see Figure 4.47). And third, if one ignores the skip connections, biases, and nonlinearity, then the identification problem reduces to factorization of $\mathbf{W_2W_1}$, allowing us to compare to results in the matrix factorization literature [16, 34, 46].

We shall now provide a result on identifiability for this class of nonlinear networks that is somewhat typical of the results in this literature. For example, in 1992 Sussmann [79] proved a similar result for a network with one hidden layer, tanh nonlinearity, and no skip connections. In contrast, our result applies to networks with skip connections and ReLU nonlinearity. The proof of Theorem 1 is given below.

Theorem 1 *Suppose that network (3.2) satisfies the following three conditions:*

(i) no column of $\mathbf{W_2}$ *or row of* $\mathbf{W_1}$ *is entirely zero;*

(ii) no two rows of $\mathbf{W_1}$ *are collinear;*

(iii) $\mathbf{W_1}$ *is nonnegative.*

Let $\mathbf{P}$ *denote an unknown permutation matrix and* $\mathbf{D}$ *an unknown positive diagonal matrix. Then, by input-output queries of the form (3.1), we may recover* $\mathbf{DPW_1}$, $\mathbf{DPb_1}$, $\mathbf{W_2P^{-1}D^{-1}}$, $\mathbf{W_3}$, *and* $\mathbf{b_2}$.

The appearance of permutation matrix $\mathbf{P}$ reflects the invariance of a two-layer network to permutations of its hidden units. Also, by positive homogeneity of the max function, the output synapses of a hidden unit (columns of $\mathbf{W_2}$) may be scaled up by some $\alpha > 0$ provided that the input synapses (rows of $\mathbf{W_1}$) are scaled down by $1/\alpha$. This gives rise to the diagonal matrix $\mathbf{D}$. These symmetries are innate to two-layer systems—the same issue is present in matrix factorization [34, Definition 4].

Discussion of preconditions and assumptions

Condition (i) imposes that every hidden unit must be connected to both the input and output of the circuit, and condition (ii) imposes that no hidden unit is computationally redundant with another. These conditions are intuitively important for identifiability. The most substantive preconditon is condition (iii), which imposes a sign constraint on the synapses at the first layer. In the neuroscience context, this would correspond

to knowing that all of the synapses in the first layer are excitatory as is the case for the outgoing synapses of bipolar cells in the retinal circuit (see Fig. 4.47). This demonstrates that neuronal cell types can simplify both the theoretical analysis as well as the recovery of the neural network's weights.

Our theoretical analysis has some notable limitations. For instance, our result assumes that we can query the network on inputs $\mathbf{x}$ lying in the negative orthant, which is often not biologically plausible. Additional constraints on the bias $\mathbf{b_1}$ would be required to guarantee identifiability given only non-negative input queries. Furthermore, our result makes no comment on sample complexity or dealing with non-convexity of the underlying optimization problem, which are all interesting directions for future work. We do, however, provide an extensive empirical study of such questions in Chapter 4.

Connection to identifiability in other systems

The preconditions of Theorem 1 are mild in comparison to results in the nonnegative matrix factorization literature, which require strong sparsity conditions on $\mathbf{W_1}$ to guarantee identifiability of $\mathbf{W_2 W_1}$ [16, 34, 46]. Theorem 1 suggests that—surprisingly—the presence of nonlinearities can make system identification *easier*. This is because the location of the nonlinear thresholds in input space is what reveals the entries of $\mathbf{b_1}$ and $\mathbf{W_1}$ up to a scaled permutation. We illustrate this in Fig. 3.1.

Proof of Theorem 1

In this section, we prove the identifiability result of Section 3.3. See [2, 20, 67, 79, 87] for related prior work. Recall the definition of network (3.2):

$$f(\mathbf{x}) := \mathbf{W_2} \max(\mathbf{0}, \mathbf{W_1 x} + \mathbf{b_1}) + \mathbf{W_3 x} + \mathbf{b_2},$$

for weights $\mathbf{W_1} \in^{n_1 \times n_0}$, $\mathbf{W_2} \in^{n_2 \times n_1}$, $\mathbf{W_3} \in^{n_2 \times n_0}$ and biases $\mathbf{b_1} \in^{n_1}$, $\mathbf{b_2} \in^{n_2}$.

To begin, observe that each component of the vector output $f(\mathbf{x})$ is a piecewise linear function of $\mathbf{x}$. We will first show that the transitions of the max function may be identified with the transitions between linear regions, thereby allowing identification of $\mathbf{W_1}$ and $\mathbf{b_1}$.

Since condition (i) excludes the possibility that any row of $\mathbf{W_1}$ is entirely zero, every row of $\mathbf{W_1}$ will cause a transition of the max function. For the jth row of $\mathbf{W_1}$, the transition occurs on a hyperplane H_j in the input domain:

$$H_j := \left\{ \mathbf{x} : \sum_{k=1}^{n_0} \mathbf{W_1}^{(jk)} \mathbf{x}^{(k)} + \mathbf{b}^{(j)} = 0 \right\} \subset^{n_0} .$$

The transition of the max function corresponding to the jth row of $\mathbf{W_1}$ will only affect output component $f^{(i)}$ if $\mathbf{W_2}^{(ij)} \neq 0$. We know by condition (i) that such an index i exists since the jth column of $\mathbf{W_2}$ is not entirely zero. Therefore all rows of $\mathbf{W_1}$ correspond to a nonlinear transition in at least one output component of f. Since the rows of $\mathbf{W_1}$ are not collinear by condition (ii), we are now sure that the boundaries between linear regions of f correspond to n_1 distinct hyperplanes that partition input space. Since the formula of a hyperplane is unique up to scalings, we may recover $\mathbf{b_1}$ and the rows of $\mathbf{W_1}$ up to scalings. Since we do not know in which order the hyperplanes should be listed, the recovery is only unique up to scalings *and* permutations.

Since we have identified $\mathbf{W_1}$ and $\mathbf{b_1}$ up to scaled permutations of the rows of $\mathbf{W_1}$, we may now query f in the region of input space where all components of max return zero. This region surely exists since by conditions (i) and (iii) combined, all rows of $\mathbf{W_1}$ point into the positive orthant. Therefore, for $\mathbf{x}$ sufficiently far into the negative orthant, we have that:

$$f(\mathbf{x}) = \mathbf{W_3}\mathbf{x} + \mathbf{b_2}.$$

The gradient $\nabla_{\mathbf{x}} f(\mathbf{x})$ in this region identifies $\mathbf{W_3}$, at which point $\mathbf{b}_2$ may be identified via:

$$\mathbf{b_2} = f(\mathbf{x}) - \mathbf{W_3}\mathbf{x}.$$

All that remains is to identify $\mathbf{W}_2$. Consider a point $\mathbf{x}^*$ on the jth hyperplane H_j that is far away from the other hyperplanes H_{-j}. Such a point exists by condition (ii). Let $\mathbf{x}^+$ and $\mathbf{x}^-$ denote points in the local neighborhood of $\mathbf{x}^*$ but on the positive and negative side of H_j, respectively. Observe that:

$$\frac{\partial f^{(i)}}{\partial \mathbf{x}^{(k)}}(\mathbf{x}^+) - \frac{\partial f^{(i)}}{\partial \mathbf{x}^{(k)}}(\mathbf{x}^-) = \mathbf{W_2}^{(ij)}\mathbf{W_1}^{(jk)}.$$

Therefore $\mathbf{W}_2^{(ij)}$ may be identified by measuring the change in gradient of $f^{(i)}$ across the jth hyperplane H_j. Of course since the rows of $\mathbf{W_1}$ are known only up to a permutation and scale, we will be unsure of the columns of $\mathbf{W_2}$ up to the same symmetries.

With $\mathbf{W_3}$ and $\mathbf{b_2}$ identified exactly, and $\mathbf{W_2}$, $\mathbf{W_1}$ and $\mathbf{b_1}$ identified up to a permutation and scale, we are done.

Chapter 4

RESULTS II: SYSTEM IDENTIFICATION OF SIMULATED MICROCIRCUITS

4.1 Simulating biologically inspired feedforward circuits

The work presented in this section rests on the assumption that cascade models are ideal for representing retinal circuits at a level of abstraction that is useful to a circuit neuroscientist. Though some models of individual ganglion cell circuits rest on finer resolution models (e.g. [32]), this assumption generally holds for the circuits that follow. Unless otherwise specified, the simulated retinal circuits described in this chapter were designed based on the following principles:

1. The photoreceptor layer is modeled as linear. Contrast adaptation is not modeled as stimuli are designed to be presented at a constant contrast level.
2. The output of the bipolar cell layer is modeled as a linear temporal or spatiotemporal convolution of a filter representing photoreceptor and bipolar cell computation with the stimulus image or video, followed by a half-wave rectifying (ReLU) nonlinearity.
3. Any amacrine cells are modeled as simply taking a linear combination of bipolar cell outputs and passing them through a ReLU nonlinearity, with no further modification unless otherwise specified.
4. There is a single ganglion cell at the output of the network which takes a linear combination of bipolar and amacrine cell outputs and passes them through a final ReLU or sigmoid nonlinearity to produce the output.

4.2 Simulating retinal circuits

The W3 circuit

The W3 retinal ganglion cell is known for its small and densely packed receptive field. It provides high-resolution information about the visual scene, and relative to other cell types, there are more of retinal ganglion cells of this type. However, the W3 cell does more than just pass a high-resolution representation of the visual scene to the brain. It is also an object-motion detector, meaning that it responds to small moving objects on a featureless or stationary background [93].

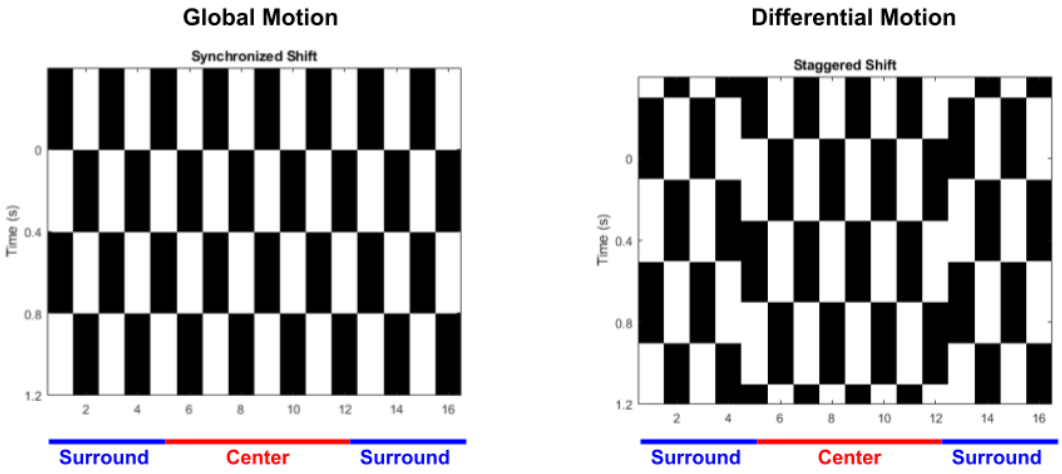

Figure 4.1: Illustration of global and differential motion stimuli. In each plot, the vertical axis represents time, while the horizontal represents space. The stimulus is spatially divided between the center and surround of the receptive field (blue and red lines). In the global motion stimulus (left) the gratings in the center and surround move simultaneously. In the differential motion stimulus (right), the center and surround are temporally out of phase.

Detailed biological study of the W3 ganglion reveals that it pools input from both ON and OFF bipolar cells, each of which is half-wave rectified. The W3 also receives lateral inhibition via wide field amacrine cells whose receptive fields are located far outside of the receptive field center of the W3 ganglion cell. Thus, local motion excites the ganglion cell, while distant motion inhibits it, leading to selectivity for local motion on a stationary background [93].

This mechanism lends itself well to the cascade model class. I simulated the circuit described above and selected a set of parameters that led to strong object motion selectivity. Fig. 4.2 shows the responses of each unit type in the simulated circuit when a grating moves back and forth (Fig. 4.1.) When coordinated motion occurs simultaneously in both the center and surround of the receptive field, excitation from the center bipolar cells occurs simultaneously with inhibition from the widefield amacrine cell, leading to no response from the ganglion cell. In contrast, when the motion in the center and motion in the surround are out of phase with one another, the excitation and inhibition are also out of phase, meaning that the ganglion cell can be excited by the bipolar cells and respond to every shift in the grating in the center of the receptive field.

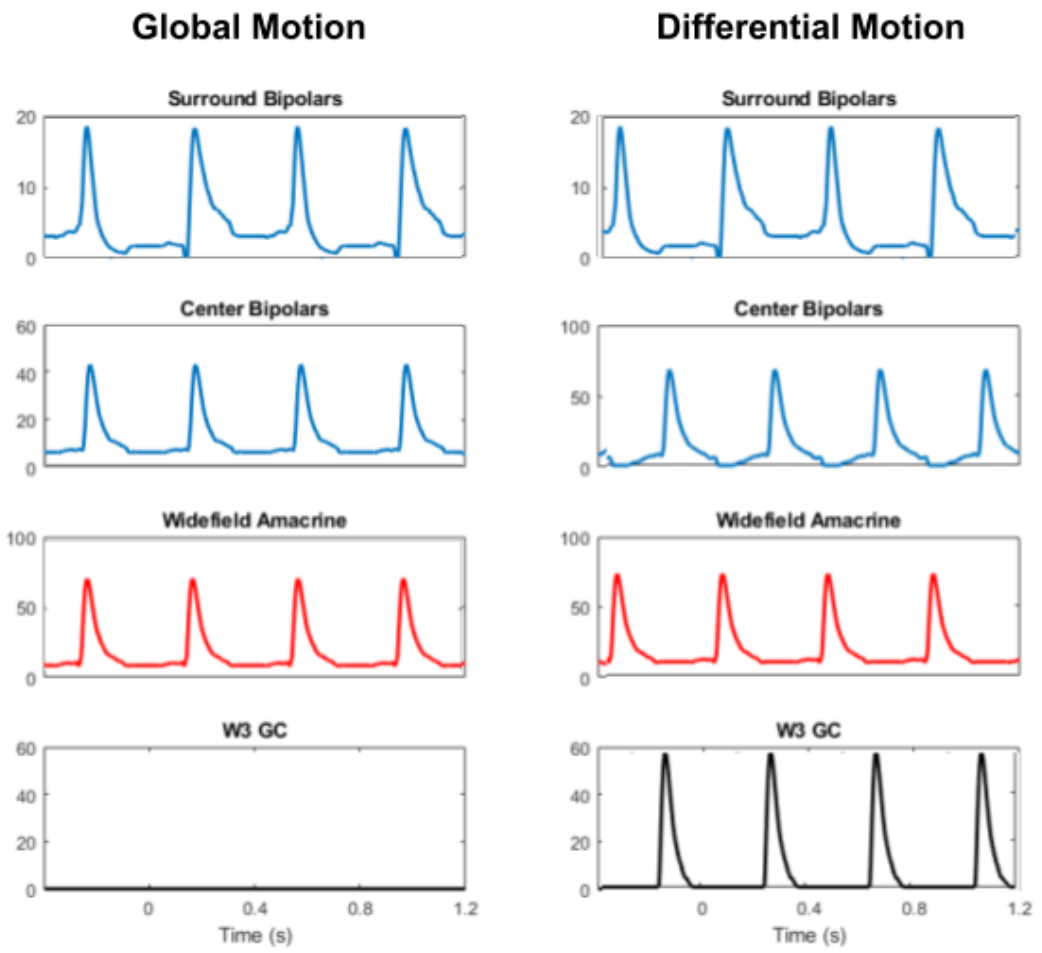

Figure 4.2: Time-varying responses of each neuron type in a simulated W3 circuit to global and differential motion stimuli (shown in Fig. 4.1).

The PV-5 circuit

The PV-5 retinal ganglion cell, one of 7 Parvalbumin positive retinal ganglion cell types is also known as the transient OFF alpha cell. This ganglion cell type is sensitive to approaching motion stimuli, but not to lateral motion. This is accomplished via a push-pull mechanism. OFF bipolar cells excite the ganglion cell, while ON bipolar cells inhibit the ganglion cell via the AII amacrine cell [58].

This cell, too, can be well modeled by a cascade model. I implemented such a model and was able to replicate the response of the neuron to an approaching and lateral motion. The approaching motion stimulus works as follows: a dark spot appears on the screen and remains stationary for two seconds. The ganglion cell responds to the appearance of the spot, and then adapts and goes silent. At that point, the

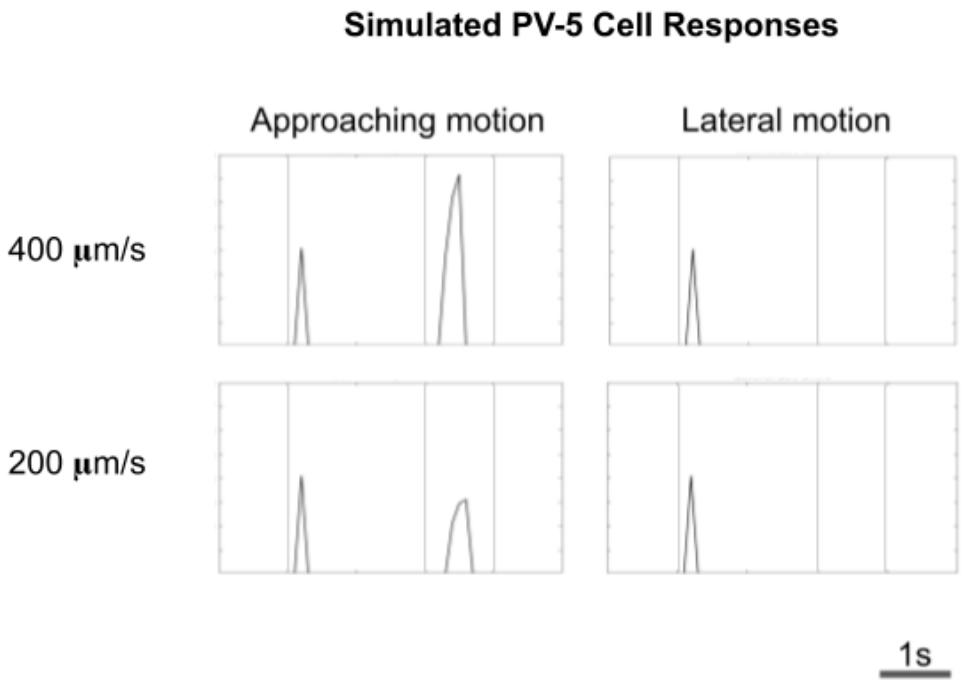

Figure 4.3: Simulation of the PV-5 circuit replicates its approach sensitivity. Approaching or laterally moving dark spots are presented to the PV-5 neuron. In each case, the stimulus appears on the screen (first vertical line), then either expands or moves laterally at either 400 or 200 μm per second (second-third vertical lines). The PV-5 retinal ganglion cell fires action potentials in response to the appearance of the spot in both cases, but only responds strongly to approaching motion, and not to lateral motion. Our model matches the results recorded from real PV-5 cells in [58].

spot begins to either expand or to move laterally at either 200 or 400 μm/second. In the case of the expanding stimulus, the ganglion cell fires again during this second phase of the stimulus. However, when the spot moves laterally, the ganglion cell is silent during this second phase (Fig. 4.3).

4.3 Case Study: The ON/OFF direction-selective ganglion cell

The ON/OFF direction-selective (ooDS) ganglion cell poses an interesting challenge for system identification. It is a retinal circuit that responds to motion in a preferred direction, but not to motion in the opposite (null) direction. Direction selective ganglion cells were first reported by Barlow and Levick in rabbit retina in 1965 [9]. They come in ON, OFF, and ON-OFF varieties. As indicated by its name, the ON-OFF direction selective cell responds both to dark and light moving objects.

Distinguishing between current circuit hypotheses

The mechanism for this computation is still under debate. Several models have been proposed at varying levels of abstraction [13, 18, 23, 29, 32, 39, 47, 61, 77, 86]. These models each point to the starburst amacrine cell (SAC) as the main driver behind the direction selectivity. The SAC is an inhibitory interneuron whose dendrites fan out in all directions, giving it a "starburst"-like appearance. The SAC may form both excitatory and inhibitory synapses with nearby direction selective ganglion cells (DSGCs). It is known to release two types of neurotransmitters: GABA as well as acetylcholine (ACh). Its key computational unit is a single dendrite. When the dendrite is stimulated by motion in the centrifugal (null) direction, it inhibits DSGCs that synapse onto that dendrite. When it is stimulated by motion in any direction, it actually excites DSGCs via ACh. Thus the DSGC receives directional inhibition and symmetric excitation from the SAC [82].

The mechanism for this phenomenon at the level of the SAC dendrite is what remains under debate. A wide variety of hypotheses have been suggested including but not limited to:

1. Asymmetry of electrotonic conduction along the length of the dendrite [32];
2. Differences in the kinetics of inputs from bipolar cells along the dendrite [23, 29, 39];
3. Lateral inhibition between SACs [47];
4. Thresholding at the SAC output synapse [86];
5. Asymmetric chloride gradient along the length of the dendrite [18];
6. Asymmetric distribution of potassium channels along the length of the dendrite [61].

It has also been hypothesized that dendritic spiking in the DSGC enhances the direction selectivity of the cell [59].

One interesting seeming contradiction in this vast body of work arose regarding item 2. In [39], it was hypothesized that bipolar cells with slower and/or more sustained kinetics excite the proximal end of the SAC dendrite, while faster and/or more transient bipolar cells excite the distal end. This asymmetry in combination with sharp thresholding at the output of the SAC would give rise to a centrifugally selective

release of GABA. However, Stincic et al later measured excitatory postsynatptic potentials (EPSCs) along the dendrite of the SAC and did not find such an asymmetry [77]. This was later refuted by Fransen et al who did measure temporally diverse EPSCs along the length of the SAC dendrite [23], seeming to confirm the hypothesis set forth in [39].

Thus, the ooDS circuit represents a canonical, well-studied retinal computation which is truly complex, and whose precise mechanism is still hotly debated. It therefore presented the perfect testing ground for a system identification technique. Since careful biological experiments seem to produce contradicting evidence, perhaps a computational system identification method might one day settle this question.

As a test of the idea of using over-connected neural networks to do this type of system identification, we focused on one detail of the models under dispute and asked whether our technique could resolve the discrepancies in the literature in simulation. Specifically, we focused on the discrepancy between [77] and [23], as to the spatial distribution of bipolar cells with different temporal dynamics onto the SAC.

We began by assuming the perspective of Stincic et al [77] with some adjustments. Namely, that there are at least two types of bipolar cells that provide input to the SAC (one sustained type and one transient type (Fig. 2.1)), but that these two types synapse onto the SAC dendrite with an even spatial distribution from the proximal end to the distal tip. In that case, the spatial asymmetry must arise in some other form in order to generate direction selectivity. One idea, put forth by Alvita Tran (a former Meister lab member), posited that the sustained and transient cells connected to the SAC dendrite in pairs, and that each pair underwent its own half wave rectification within a small segment of the dendrite. Biologically, this would equate to a physiological restriction on the ability of one depolarized membrane segment to pass that depolarization along to the next segment unless some threshold was met. While this does not translate readily to the standard cable theory, it is not unheard of for individual segments of dendrites to behave in this manner, due to, for instance, the presence of voltage-gated ion channels [78, 89].

By pairing spatially adjacent sustained and transient bipolar cells in this way, one can devise a situation in which each segment of the SAC dendrite is direction selective, and the output at the distal tip of the dendrite is simply a reflection of the summed activity of all the segments. Henceforth, we shall refer to this as the Tran model of direction selectivity.

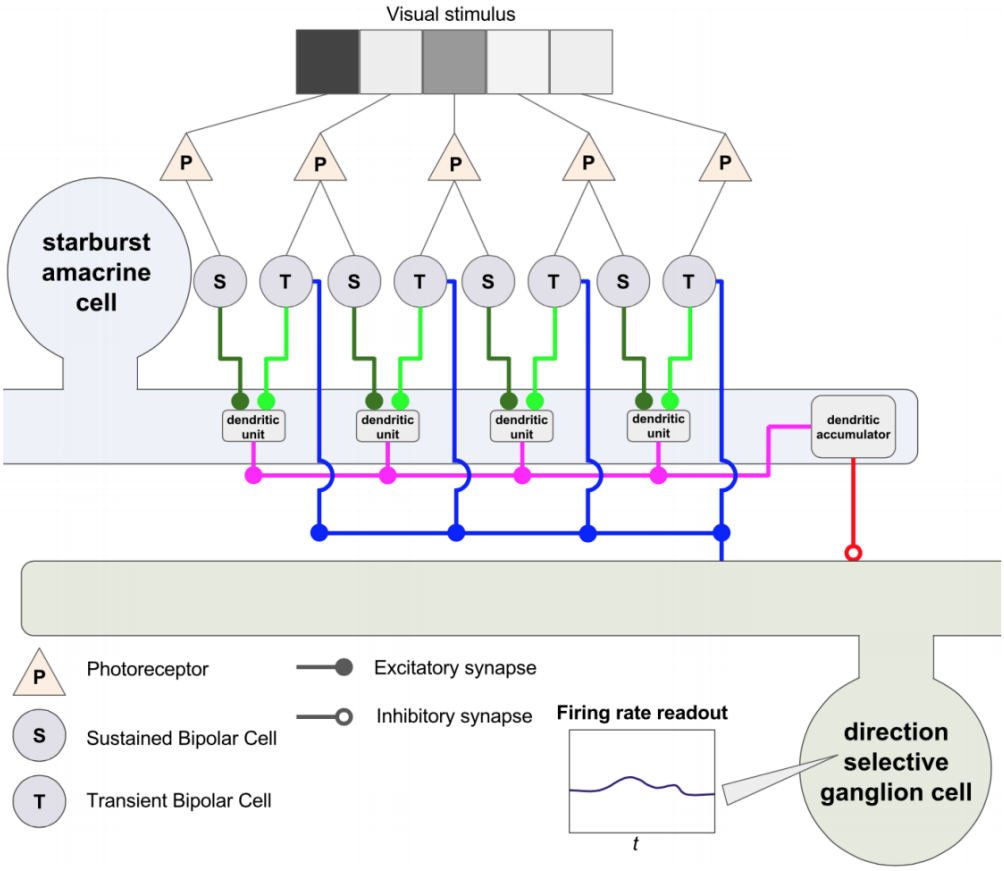

Figure 4.4: Schematic of Tran model of direction selectivity. Adjacent sustained and transient bipolar cells are paired at a single segment of SAC dendrite. The output of the SAC is the sum of the output of each nonlinear segment, passed through a final layer of nonlinearity. The transient bipolar cells also excite the ganglion cell.

Test of method in small circuit

We generated data from a simulated direction-selective cell with known circuit connectivity, modeled in the form of a CNN. Each neuron performed a convolution in time and passed the output through a static nonlinearity to produce an activation (Meister and Berry 1999). The convolutional kernels were temporal filters chosen to match the recorded flash responses of retinal cell types (Baccus and Meister, 2002).

The training data for each machine learning experiment consisted of a set of visual stimuli and the associated network output of the true retinal model. Stimuli were $k \times T$ movies (T frames of k pixels). Network outputs were $1 \times T$ sequences of ganglion cell activation, which took a value between 0 and 1 at each time point $t \leq T$. Visual stimuli were chosen from three classes: moving dot stimuli (Fig. 4.7a), static random patterns (Fig. 4.7b) and moving random patterns (Fig. 4.7c). Each pixel of

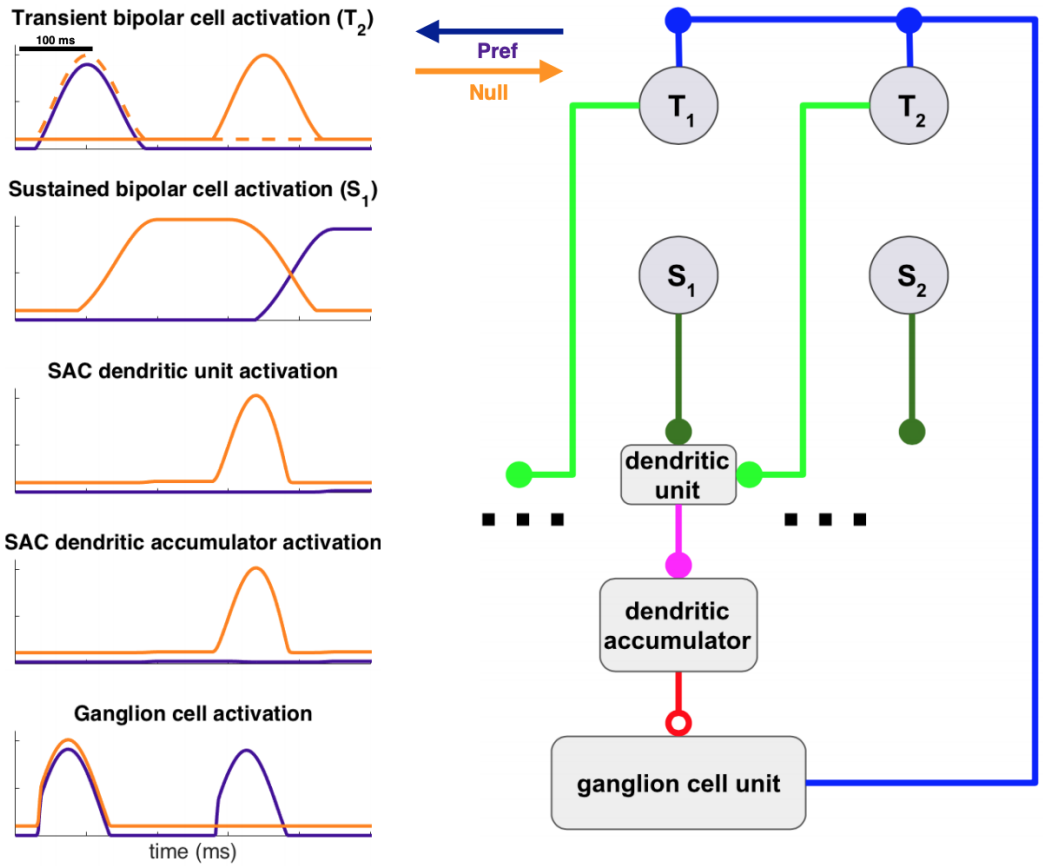

Figure 4.5: Right: Tran model circuit motif. "S" and "T" refer to sustained and transient type bipolar cells. Left: each model neuron's activations for apparent motion moving dot stimulus in the preferred and null directions. Null direction responses are vertically offset for visibility. Dotted orange line represents activation of T1.

the stimulus corresponded to the spatial receptive field of a single simulated bipolar cell. The duration of each stimulus was chosen based on the length of the temporal convolutional filters used in the first hidden layer. This layer modeled the retina's bipolar cell layer (Fig. 4.6), which is known to comprise the slowest stage in retinal processing after the photoreceptor layer (Baccus and Meister, 2002).

The simulated DSGC had a preferred direction of stimulus motion, defined by the fact that the network was always activated by a white moving dot in one direction and silent for a moving dot in the opposite (null) direction. This selectivity was accomplished as follows: In the null direction, the white dot passed first over a slow bipolar cell's receptive field, before passing over the fast bipolar cell to its left. The sustained response of the slow bipolar cell therefore overlapped in time with the

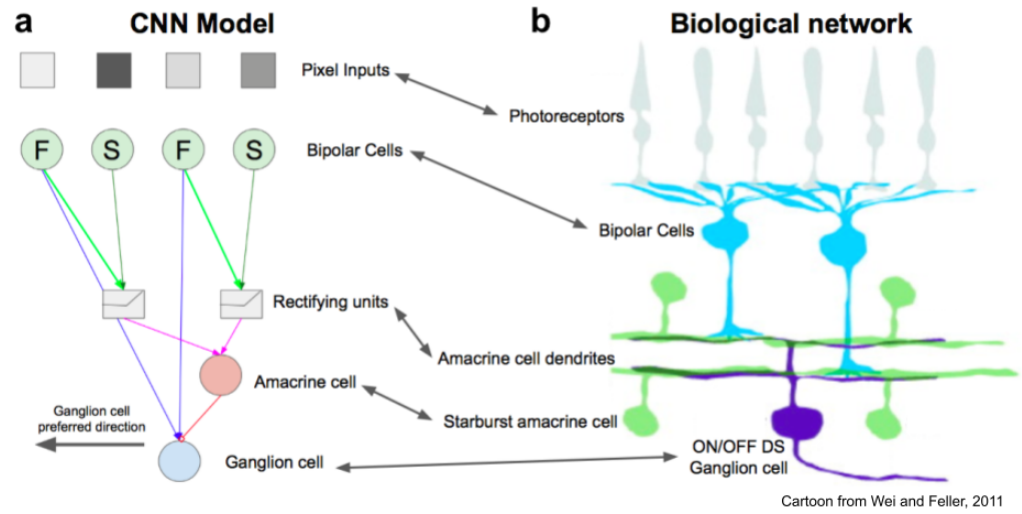

Figure 4.6: The CNN model's layers correspond to the layers of the retina. Individual artificial neurons model biological neurons as linear-nonlinear units with a temporal convolutional filter and a static nonlinearity. Biological network cartoon taken from Wei and Feller, 2011.

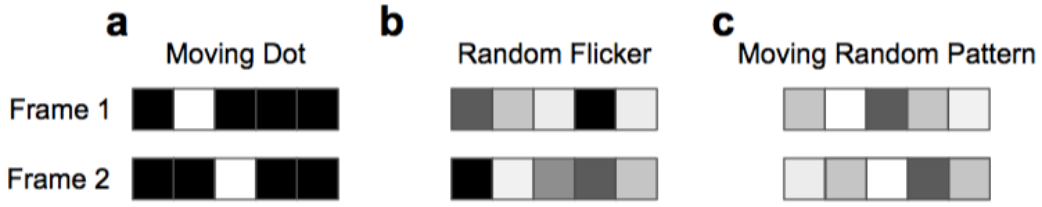

Figure 4.7: Examples of stimuli used to generate training data. Each stimulus was 500 ms long, consisting of two 250 ms frames. **a**: Moving dot: white dot on a black background that shifts one pixel to the left or to the right. **b**: Static random pattern: randomly generated pattern that turns on and off. **c**: Moving random pattern: randomly generated pattern that shifts one pixel to the left or to the right.

more transient response of the fast bipolar cell. These two bipolar cell outputs were then summed and the output was rectified by the second hidden layer, representing nonlinear input integration at the amacrine cell dendrite (Fig. 4.6). Because of the temporal overlap of the two bipolar cell responses, their sum crossed the rectifier's threshold, creating a nonzero rectifier output. The amacrine cell became activated as a result and provided an inhibitory input to the ganglion cell. This inhibition cancelled the excitatory input from the bipolar cells to the ganglion cell, silencing the ganglion cell output (Fig. 4.5).

In the preferred direction, the fast and slow bipolar cell responses occur in the opposite order, and therefore do not overlap in time. Thus the input to the rectifying unit is not sufficient to cross the threshold) and the rectifying unit output is zero. The bipolar cell excitatory input to the ganglion cell is therefore unimpeded by the amacrine cell input, leading to activation of the ganglion cell (Fig. 4.5).

Using these simulated training data, the problem of algorithmic modeling was approached in three stages. In phase 1 we constructed a deep net with the data-generating model and trained it in order to learn only the synaptic weights of the original model. In phase 2 we increased the complexity of the learning problem by constructing a deep net with the same structural motif, but more neurons than the original model, and trained with ℓ_1-regularization to encourage pruning away of extra synapses and the recovery of the true model. Finally, in phase 3 we constructed a deep net that also included alternative structural motifs and again used regularized training in a way that encouraged the selection of model structure in addition to the learning of synaptic weights.

Phase 1

In this phase, we attempted to learn only the synaptic weights of a retinal model. We generated training data by presenting moving dot and static moving pattern stimuli to a network model. We then constructed an identical network with randomly initialized synaptic weights, and trained it to learn the synaptic weights of the true model. We used basic gradient descent with a strong momentum parameter on a dataset of about two thousand visual stimulus examples. The learning problem can be expressed as:

$$_{w_j}\left(\sum_{i=1}^{N}\left(y_{\text{trained}}(x_i)-y_{\text{true}}(x_i)\right)^2\right),$$

where w_j were the synaptic weights of the model, x_i was the ith visual stimulus, $y_{\text{true}}(x_i)$ was the output of the simulated retinal ganglion cell for stimulus x_i, and $y_{\text{trained}}(x_i)$ was the output of the trained CNN for stimulus x_i.

We observed that regardless of random initialization, training consistently proceeded in two stages. At first the synapses of the excitatory pathway (Fig. 4.8b, blue) were learned within a few training iterations. Additionally, many inhibitory pathways quickly moved from their random initializations to the neighborhood of the correct value (Fig. 4.8b, green, red). After this, there was a slower stage in which the synapses between the amacrine cell rectifying units and the amacrine cell convolu-

tional unit were learned (Fig. 4.8b, magenta). During this stage, fine tuning of other inhibitory pathway synapses also took place. It is interesting to note that in this phase, the training set of visual stimuli consisted of about a thousand static random patterns and about a thousand moving white dots. Using fewer moving dot stimuli precluded the network from learning the correct values of the inhibitory pathway synapses (Fig. 4.8c).

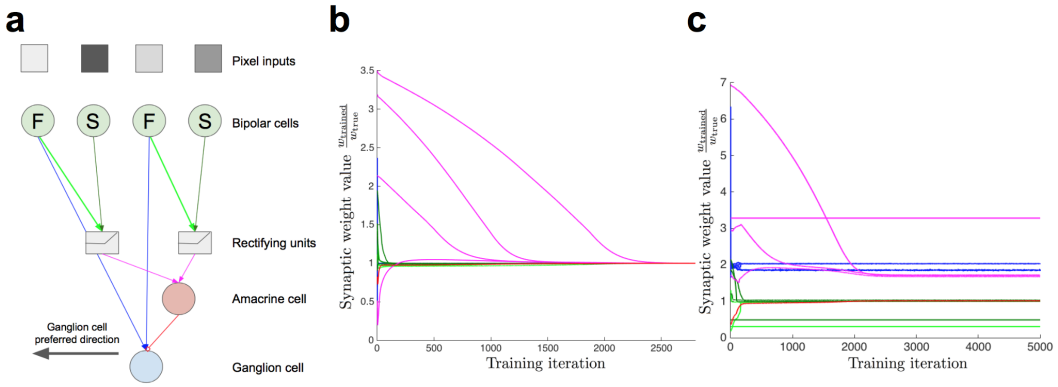

Figure 4.8: Phase 1: learning the synaptic weights of a direction selective model retinal ganglion cell. **a**: True network structure. **b**: Synaptic weights in the trained model converged to their true value (this is indicated by $\frac{w_{\text{trained}}}{w_{\text{true}}} = 1$). **c**: Using fewer moving pattern examples (184 instead of 984) led to an inability to learn the correct synaptic weights. Curves are colored to match their corresponding synapses in the diagram.

Phase 2

In phase 2, we trained a network to learn the spatial receptive field of the true network. In this stage, the true model had sixteen bipolar cells and seven amacrine cell rectifying units, while the trained model was initialized with twenty-four bipolar cells and eleven amacrine cell rectifying units. We generated training data with moving random pattern and static random pattern stimuli. The basic connectivity motif of the true model was conserved in the trained model initialization (Fig. 4.9).

A small amount of ℓ_1-regularization was required in order to eliminate unnecessary synapses. The learning problem thus becomes:

$$w_j \left(\sum_{i=1}^{N} \left(y_{\text{trained}}(x_i) - y_{\text{true}}(x_i) \right)^2 + \lambda \sum_j |w_j| \right),$$

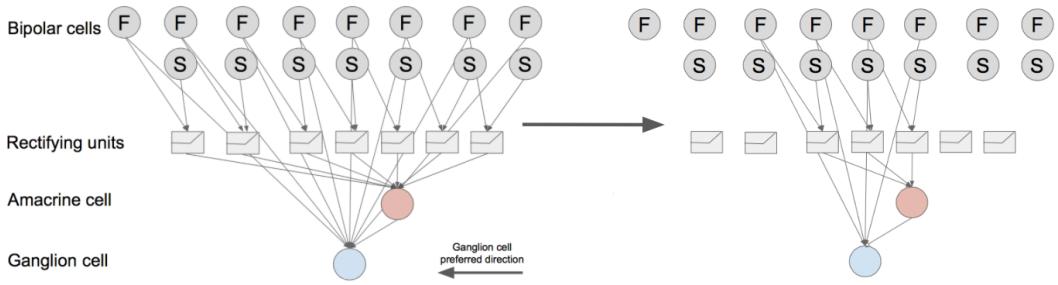

Figure 4.9: Phase 2: learning a spatial receptive field. The network on the left was pruned to the structure on the right.

where λ is a tunable parameter controlling the strength of the regularization term during training. Without this regularization, extra neurons often maintained small synaptic weights along the inhibitory pathway that contributed very little to overall squared error on the training set (Fig. 4.10a). With regularization, these synaptic weights often decayed linearly and slowly to zero, indicating that a large portion of the gradient with respect to these synapses came from the regularizer rather than the loss term of the objective function (Fig. 4.10b).

While phase 1 learning occurred using training data generated from moving dot and static random pattern stimuli, phase 2 training could not be successfully completed with these training data (Fig. 4.10c). Including moving pattern stimuli (Fig. 4.7c) in the training set rather than static patterns led to successful learning in this phase, even with half as much training data (Fig. 4.10b).

Phase 3

In phase 3 we aimed to learn not only the simulated ganglion cell's spatial receptive field and preferred direction but also its underlying circuit motif. In this case the trained network was initialized with additional neurons and synapses such that the overall network included multiple hypothesized computational pathways. This time, "pruning" led to not only the elimination of unnecessary neurons but also the selection of a model structure to reproduce the training data. This was the most direct simulation of algorithmic modeling—an automated method to select model structures and fit their synaptic weights to recorded data simultaneously.

In this case, we allowed for two potential inhibitory pathways to produce the phenomenon of direction selectivity. First, we connected all bipolar cells directly to the amacrine cell output node (the third hidden layer), simulating linear input integration

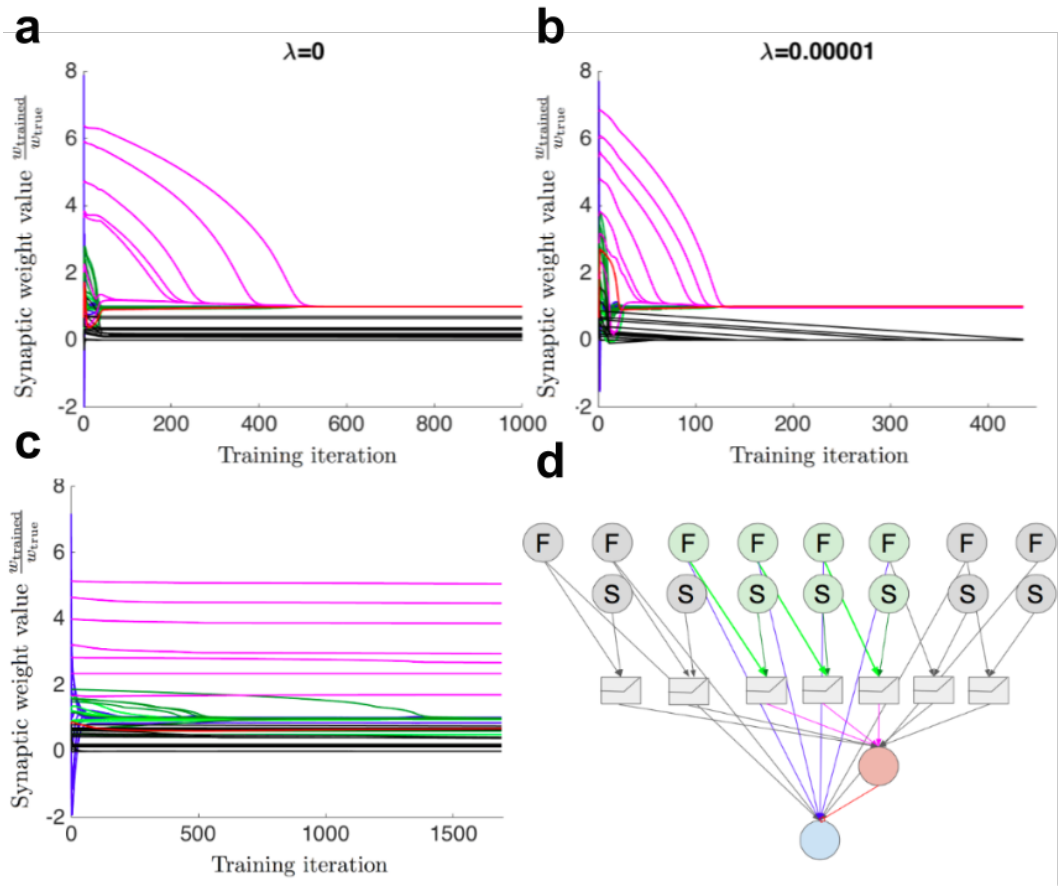

Figure 4.10: Evolution of synaptic weights over time. Curves are colored to match their corresponding synapses in **d**. For synapses whose true value is zero, w_{trained} is plotted instead of $\frac{w_{\text{trained}}}{w_{\text{true}}}$, and these curves are colored black. **a**: Without regularization, synapses included in the true model converged to their correct value (this is indicated by $\frac{w_{\text{trained}}}{w_{\text{true}}} = 1$). However, unnecessary synapses remained at their initialized value. **b**: With regularization, all synapses converged to their true value and pruning was successful. **c**: Without moving pattern, but with twice as many moving dot and static pattern data examples, synaptic weights failed to converge to their true value.

by the amacrine cell dendrite (via summation and temporal convolution) and a single output nonlinearity [39]. Second, we again included a pathway that paired adjacent fast and slow bipolar cells and passed their summed outputs through rectifiers before being convolved with the amacrine cell temporal filter and passing through the output nonlinearity (again allowing for either choice of preferred direction.) The latter mechanism, with two nonlinear processing stages at the amacrine cell, was used to generate the training data, so we expected that all synapses corresponding only to the former circuit as well as the latter circuit with opposite preferred direction

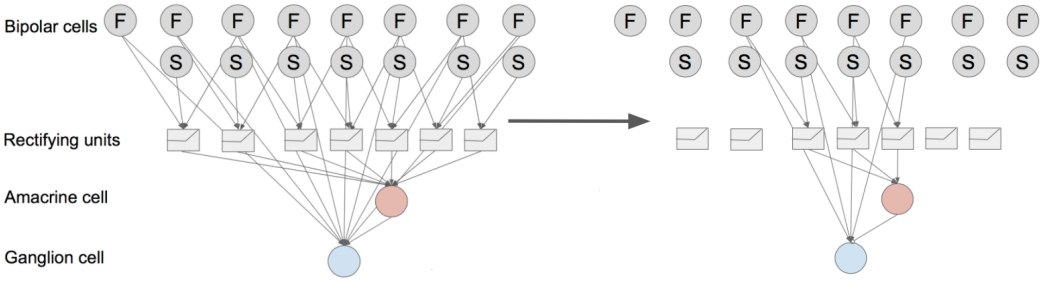

Figure 4.11: Phase 2: learning a spatial receptive field and preferred direction. In the network initialization, the rectifying units pair each slow bipolar cell with both a fast bipolar cell on the left and one on the right to allow for either preferred direction to be learned. During training, the network on the left is pruned to the structure on the right.

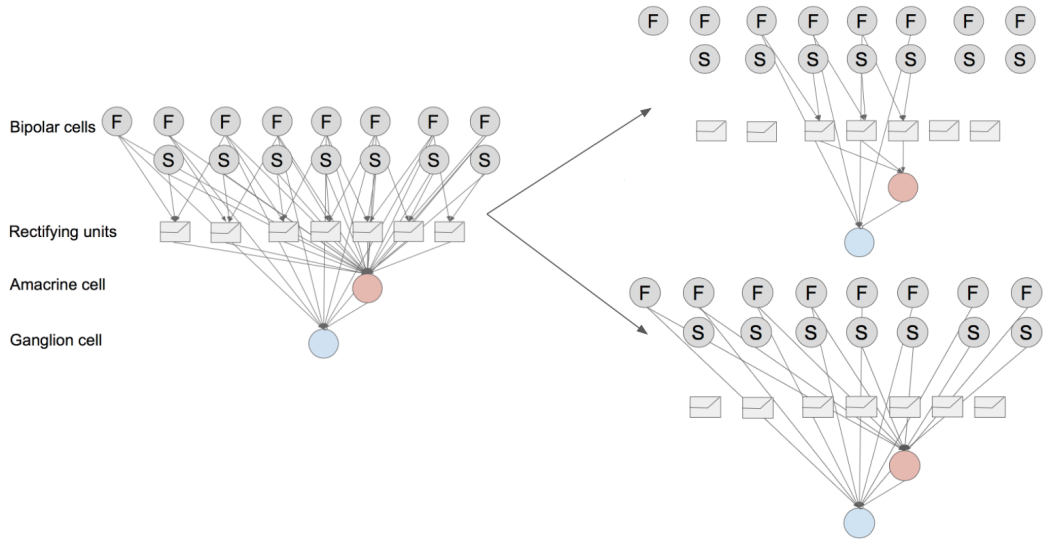

Figure 4.12: Phase 3: learning a network architecture. The network on the left is pruned to one of the two candidate structures on the right.

would be eliminated during training, though we allowed the trained network to use either mechanism or even a combination of the two (Fig. 4.12). Again, we were able to train the network to correctly learn both the synaptic weight values of the true network and to prune away the unnecessary neurons.

Phase 4

In phase 4, we extended the work in phase 3 to a larger network that includes both ON and OFF pathways in order to recover the full, biologically plausible circuitry of a simulated ON/OFF direction selective ganglion cell. Once again the network

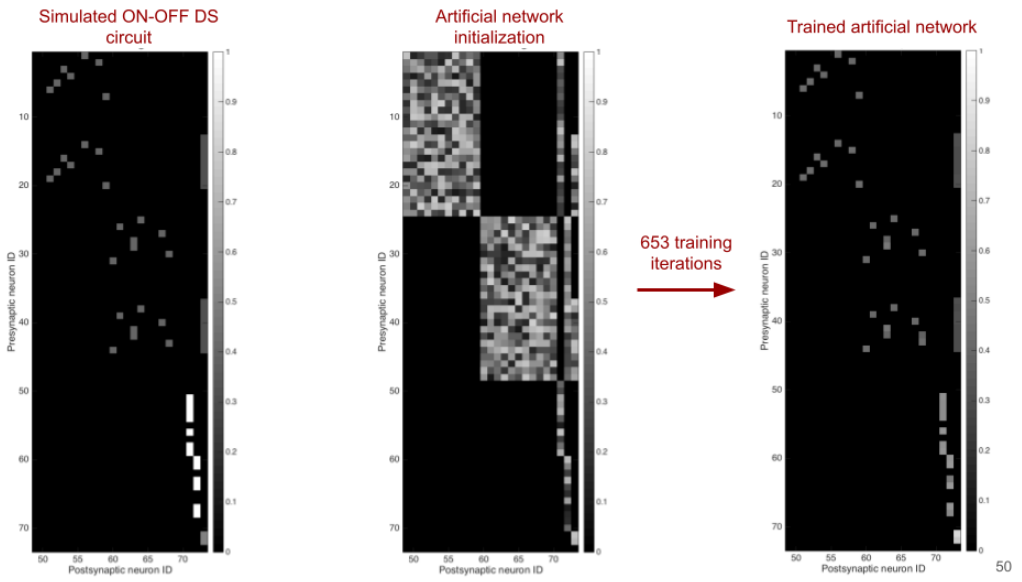

Figure 4.13: Left: Weight matrix of a simulated ON-OFF DS circuit. Entry (i, j) represents the magnitude weight between neuron i and neuron j. Center: Weight matrix of initialized ANN. Right: Weight matrix of trained ANN is nearly identical to the simulated circuit, with some small differences.

was initialized as overconnected, with too many neurons and synapses, and this time included cells initialized as both OFF and ON neurons. The trained network was able to recover the circuitry of the simulated ooDS cell almost perfectly, and this was robust to random initialization. We randomly initialized all the synaptic weights in the network 10 times, and were able to nearly perfectly reconstruct the circuitry each time. This was the most direct simulation of algorithmic modeling—an automated method to select model structures and fit their synaptic weights to recorded data simultaneously.

Quantifying system identification

To better quantify the success of this system identification technique, we needed to develop some type of heuristic or score. The first version of this was dubbed the "projection score" and was computed by simply taking the normalized vector projection of the learned synaptic weight matrix onto the true synaptic weight matrix.

Let the *projection score* be defined on the vectorized weight matrices as:

$$S = \frac{\mathbf{W}_{\text{learned}} \cdot \mathbf{W}_{\text{oracle}}}{|\mathbf{W}_{\text{oracle}}|^2}. \tag{4.1}$$

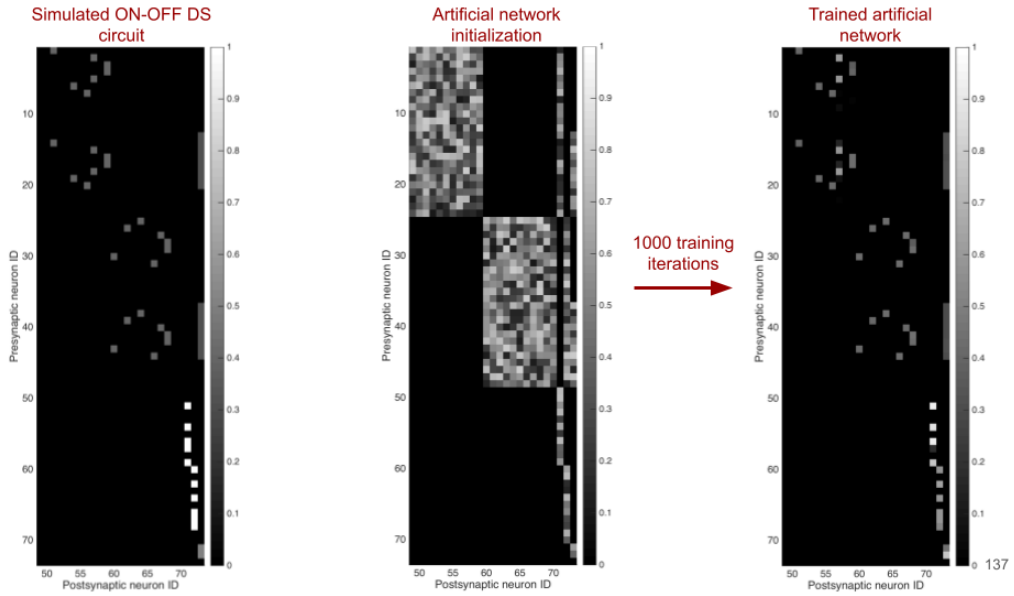

Figure 4.14: Same as Fig. 4.13, but with a different random initialization.

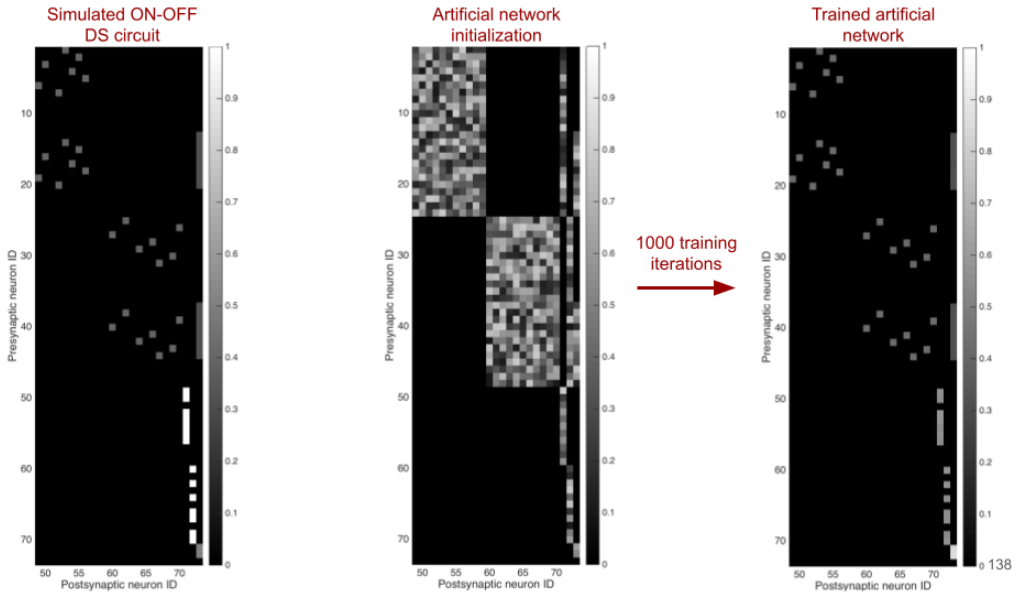

Figure 4.15: Same as Fig. 4.13, but with a different random initialization.

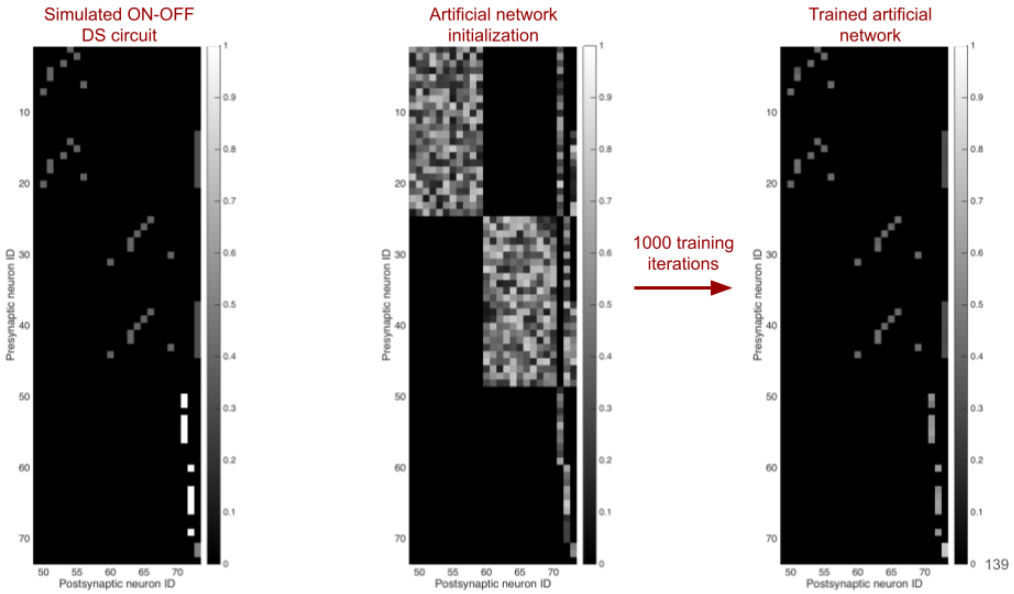

Figure 4.16: Same as Fig. 4.13, but with a different random initialization.

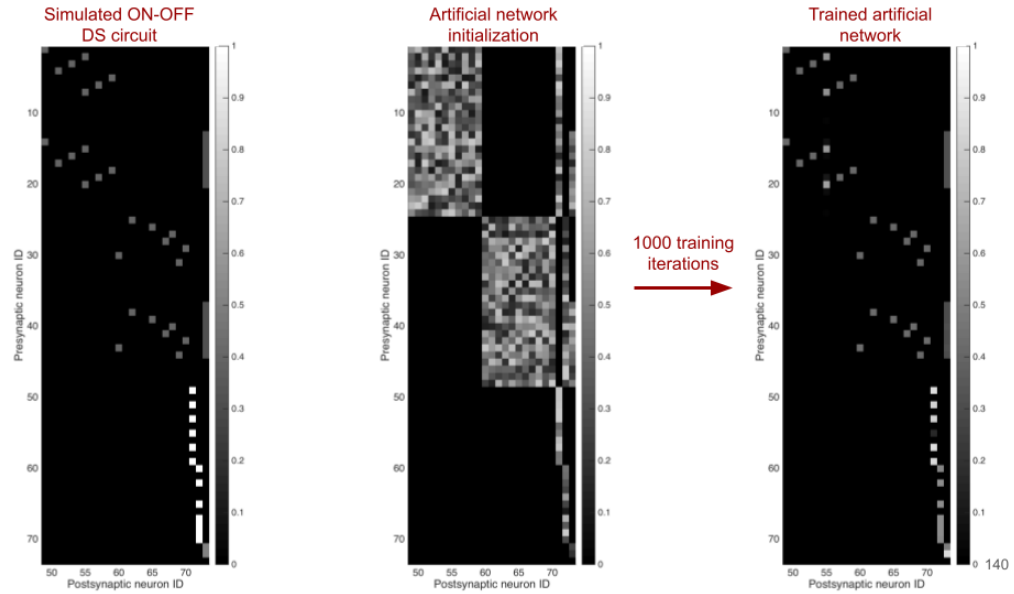

Figure 4.17: Same as Fig. 4.13, but with a different random initialization.

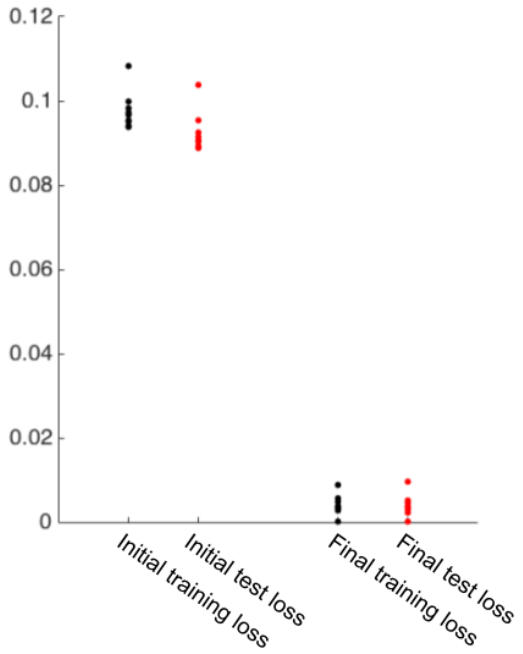

Figure 4.18: Summary of initialized and final training and test loss for the ooDS simulations above.

We can now use this score to understand the success of system identification over the course of training. Figure 4.27 shows one way of doing this. In it, we keep track of the projection of the learned network onto both the true circuit (the Tran circuit) as well as an alternative hypothesis, the Kim circuit [39] described in Section 4.3. By doing this, we are simulating the scenario in which the researcher trains the ANN with both of these hypotheses in mind. Note that the initialized ANN structure in all the experiments described so far can accommodate either hypothesis, depending on which synapses are pruned away during training. The researched in this scenario would compare the projection score of the learned ANN with each of these two hypotheses at the end of training to determine which is more likely to have generated the data.

Replicating the results of a classical circuit dissection experiment

Barlow and Hill discovered direction selective retinal ganglion cells in the rabbit in 1963 [8]. Upon this discovery, they set to work devising possible circuit architectures that could perform such a computation. Two years later, Barlow and Levick proposed two alternative models: the excitatory and the inhibitory model. Each version employed a two subunits, one with a delay line, joined by a coincidence detector. The excitatory model joined the two subunits with an AND gate, while the inhibitory

model used an AND NOT gate [9]. In 1976, Wyatt and Daw set out to determine which model was in use in rabbit retina. To understand whether inhibition was at play in the direction selective retinal circuit, they applied pictrotoxin, an antagonist of $GABA_A$ receptors. They found that this eliminated the direction selectivity phenomenon and concluded that an inhibitory mechanism was in use in this circuit [90]. The Tran circuit described above is somewhat different from the one that Barlow originally drew. The excitatory mechanism in Barlow's paper describes how direction selectivity arises in the Starburst amacrine cell, but the SAC then inhibits the ganglion cell, so that it does not fire when motion occurs in the null direction.

Can we re-discover this finding using a purely computational method? If so, this would be an indicator that such a method can be extremely useful for circuit neuroscientists. To simulate this, I initialized an ANN containing all the neurons and synapses necessary for both an excitatory and an inhibitory version of the Tran circuit (Fig. 4.19). This network was then trained on data generated by the Tran circuit, and converged to an inhibitory circuit with one small difference. A few of the dendritic units directly inhibit the ganglion cell, rather than going through the dendritic accumulator unit (Fig. 4.20). However, the ultimate result is still an inhibitory circuit with two circuit motifs, one of which is the correct motif that generated the data. Thus, the network was able to correctly replicate the result of Wyatt and Daw's experiment in simulation and identify that direction selectivity in the retina arises via an inhibitory mechanism, and to partially recover the structure of the ooDS circuit.

Intelligent choice of stimuli is necessary for circuit recovery

In studies of the ooDS and other direction selective cells, a commonly used stimulus is the moving spot or moving bar [8, 9, 90]. In this stimulus, a spot or bar appears on a plain, contrasting background and then moves in various directions (Fig. 4.22a.) Typically the background is kept at a gray level to which the retina has adapted, and the spot or bar is presented in black or white depending on the polarity of the cell. This stimulus helps to identify direction selective cells, and can even be used to do in real time during the recording. The moving spot can be thought of as the simplest possible stimulus that gets at the heart of the computation done by the DS cell. However, the amount of information one can learn by presenting moving spot stimuli is limited. While the question of whether or not a neuron is direction selective can be answered, it would be hard to estimate the weights of the network based only on the response to this stimulus.

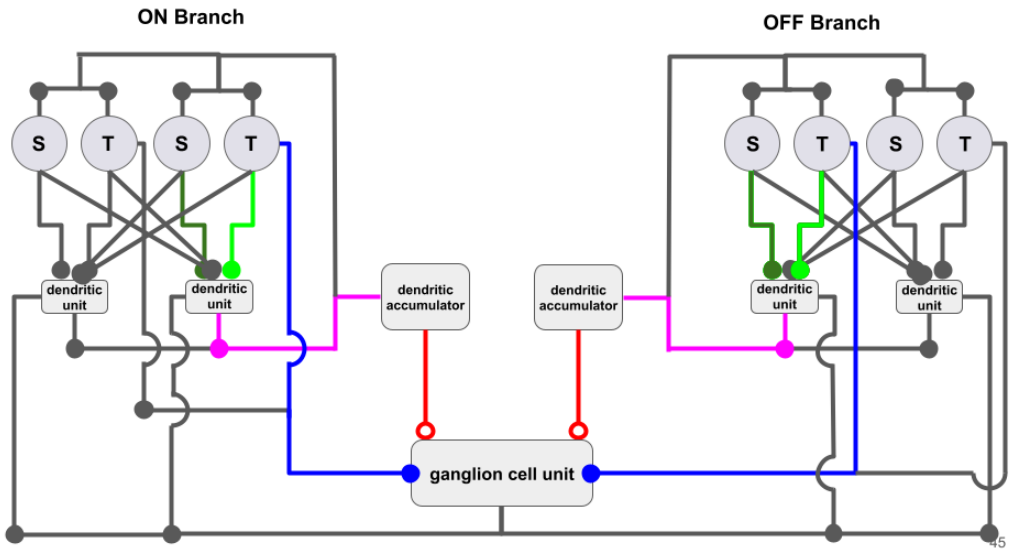

Figure 4.19: Initialized ANN structure used to replicate experimental result in [90]. This ANN contains all the same elements as previous versions for the ooDS cell, however, it also includes an excitatory pathway directly from the dendritic units to the ganglion cell unit. These are meant to simulate an excitatory pathway.

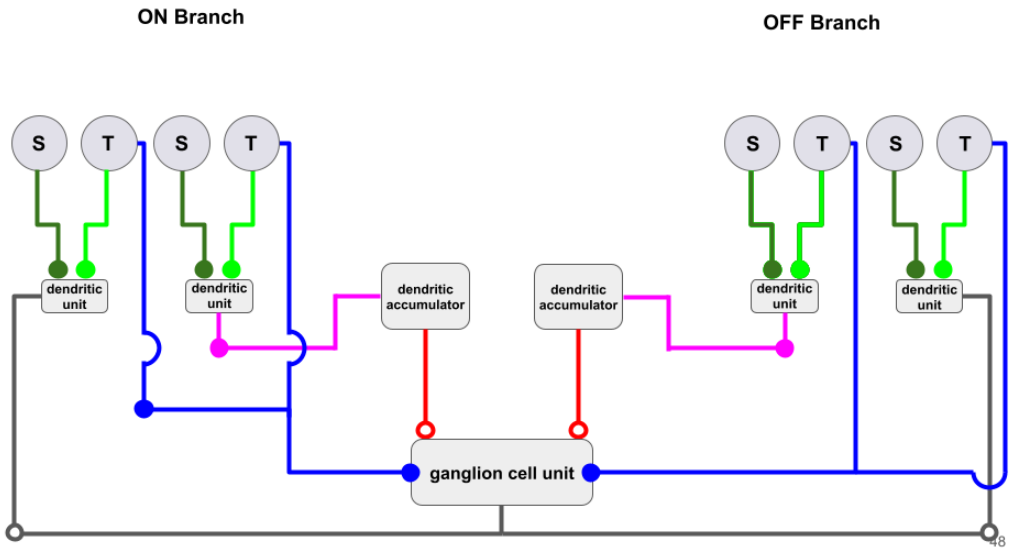

Figure 4.20: After training the ANN in figure 4.19, the resulting structure is the correct ooDS circuit that generated the training data, with one difference. The dendritic units are directly inhibiting the ganglion cell unit, in addition to doing so via the dendritic accumulator. Thus we can replicate the results of Wyatt and Daw and prove that direction selectivity in the retina uses an inhibitory mechanism.

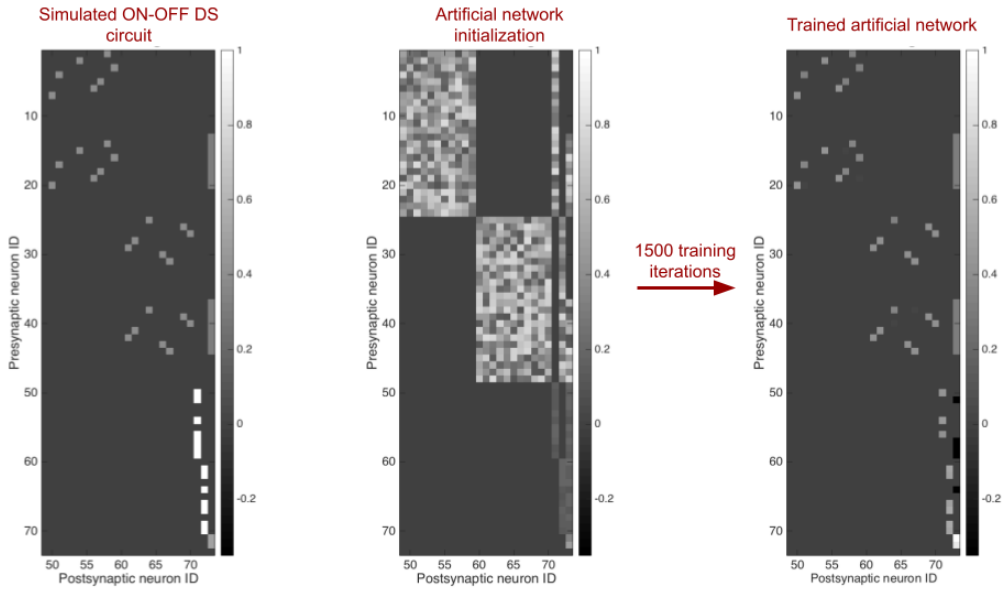

Figure 4.21: Left: Weight matrix of a simulated ON-OFF DS circuit. Entry (i, j) represents the magnitude weight between neuron i and neuron j. Center: Weight matrix of initialized ANN. Right: Weight matrix of trained ANN is nearly identical to the simulated circuit, with some small differences.

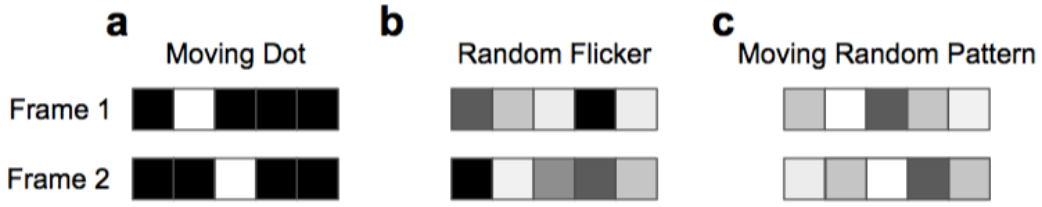

Figure 4.22: Three types of visual stimuli. These stimuli have one spatial dimension and are time varying. **a** In the moving dot stimulus, a dot appears on a blank background and moves left or right. **b** In the random flicker stimulus, each frame is a new set of checkers with random intensities. **c** The moving random pattern stimulus begins by presenting a random pattern on the screen and then shifts that pattern either left or right.

For parameter estimation, a standard stimulus such as random flicker (Fig. 4.22b) might be more useful. Random flicker stimuli are created by dividing the visual field into 1D bars or 2D "checkers," and giving each a random intensity. The bars or checkers can be binary (black and white), or their intensities can be drawn from a continuous grayscale. The intensities of each of the bars or checkers are randomly drawn from a uniform or Gaussian distribution, and is redrawn in every frame of the stimulus, which typically varies at 50 or 60 Hz. This stimulus is frequently used for spike-triggered average and covariance analysis, and is designed to span the space of spatial and temporal frequencies as much as possible during a time-limited neural recording. Thus, the neuron's response to this stimulus yields a great deal of information and can be used for continuous parameter estimation in many cases.

However, it is true that not all neurons actually fire in response to random flicker stimulus. Depending on the size of the bars/checkers and contrast of the stimulus, it may not be sufficient to drive some neurons with highly nonlinear circuitry. While direction-selective retinal ganglion cells usually do fire during random flicker stimulus, we know that it is unlikely to drive the cells in the salient regime of visual space that includes motion stimuli. Random flicker, while designed to be agnostic and highly informative, does not do a great job of simulating motion, and therefore may not activate the direction selective circuit in the relevant regime for system identification.

To address this, we combined the moving spot stimulus with the random flicker to produce what we call the "moving random pattern" stimulus (Fig. 4.22c). In this condition, a random flicker-type pattern appears on the screen. But instead of the pattern randomly changing every frame, this single pattern moves across the screen in various directions. Thus, we are able to drive the neuron across a wide range of stimulus space while still incorporating motion, which we know will activate the circuitry relevant to our study.

Indeed, we found that data collected using only the moving dot stimulus is insufficient for system identification (Fig. 4.23). Under this condition, most neurons in the network are silent most of the time. Only a single subunit is being driven at any given time, when the spot is over its receptive field. When the predominant signal contributing to the loss function is silence, the sparsity regularizer dominates the objective function, and almost all the synapses in the network are pruned to zero. Thus, even though the moving dot stimulus would alert a human experimenter to the fact that the neuron is direction selective almost immediately, it is not sufficient for

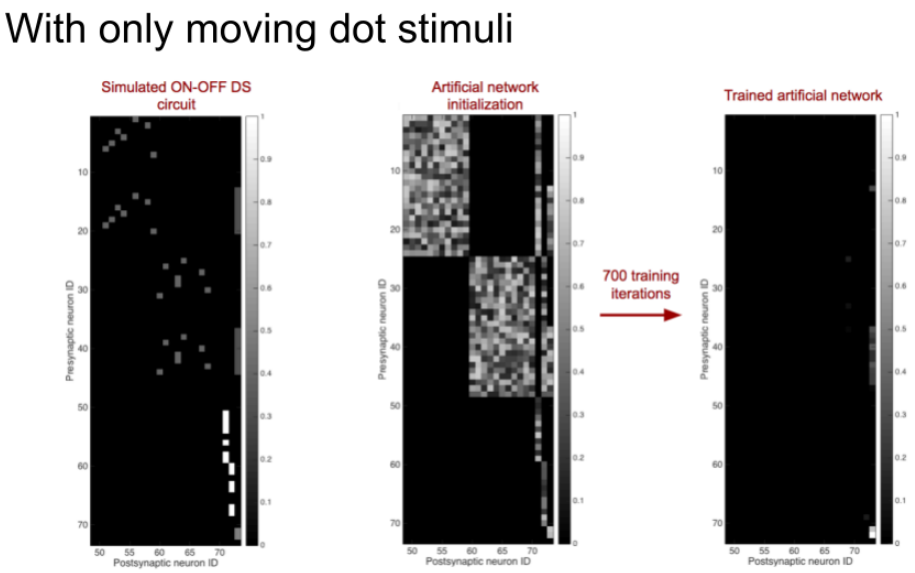

Figure 4.23: Left: Weight matrix of a simulated ON-OFF DS circuit. Entry (i, j) represents the magnitude weight between neuron i and neuron j. Center: Weight matrix of initialized ANN. Right: Weight matrix of trained ANN. When only moving dot stimuli are used to generate training data, almost all synapses are pruned out of the network.

the artificial network to recover its structure. Moving random pattern, on the other hand, is informative enough and drives the circuit well enough to do near-perfect system identification (Fig. 4.13).

We also found that training the network with moving random pattern stimuli only led to better structure recovery than splitting the training dataset between moving random patterns and random flicker (Fig. 4.24,) indicating that some information about this motion-sensitive circuit simply cannot be conveyed by its responses to the random flicker stimulus. However, with both training sets, the task of selecting the Tran circuit over the Kim circuit is easily accomplished (Fig. 4.24).

The dynamics of learning

The network is set two tasks during training: prune away unnecessary synapses and learn the correct value of each synaptic weight. Interestingly, these tasks are performed in a stereotyped order. Training can reliably be divided into two phases. The first phase is the structure learning phase, in which many synapses are quickly pruned out of the network by the sparsity regularizer. Once the correct structure has been learned, the synaptic weights must be approximated. We call this the fine-

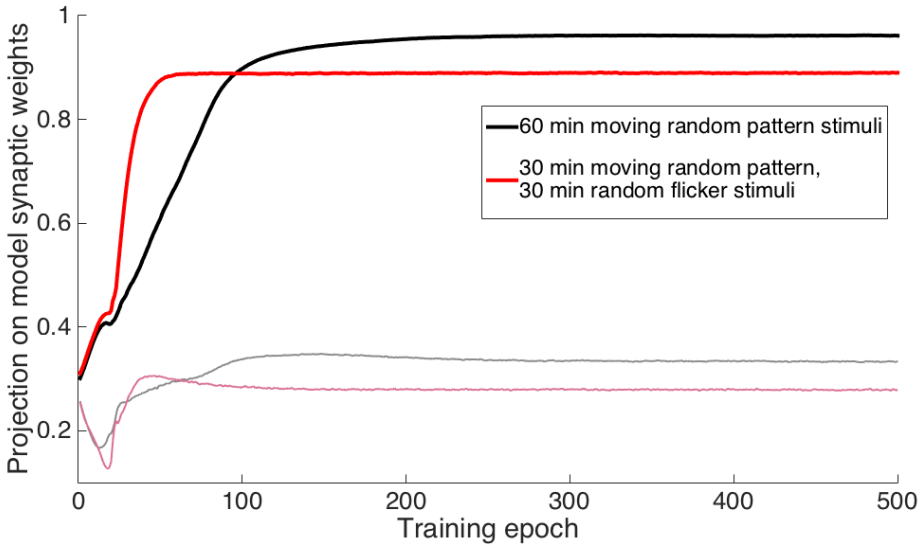

Figure 4.24: Projection score over the course of training for two trained ANNs with different training sets. Thick lines represent projection onto ground truth circuit structure. Thin lines represent projection onto alternative hypothesis. Black lines represent ANN trained on a dataset of one hour of moving random pattern stimuli. Red lines represent the same ANN trained on a dataset of 30 minutes of moving random patterns and 30 minutes of random flicker. The purely moving random pattern dataset leads to better final projection score, but slower convergence.

tuning phase of learning, which proceeds much more slowly and is driven purely by the loss component of the objective function. Note that in these experiments, the relative strength of the loss and regularizer was held constant throughout the course of training, so this phasic behavior arises organically, and not due to a changing objective function. One example is shown in Fig. 4.25, where a change in the derivative of the loss function is also clearly visible at the epoch when the final unnecessary synapse is pruned to zero.

A sparsity regularizer is necessary for recovery of this circuit

This approach rests on the idea of creating an overconnected ANN with too many neurons and too many synapses, and pruning it down to only the necessary neurons and connections needed to recreate the training data. To do so, we employ the standard ℓ_1-regularizer, which is commonly used to encourage sparsity both in neural networks, but more famously in linear models [83]. But is this regularizer truly necessary? After all, selecting the exact circuit that generated the data should result in a loss of 0, so shouldn't the loss function be sufficient to encourage pruning?

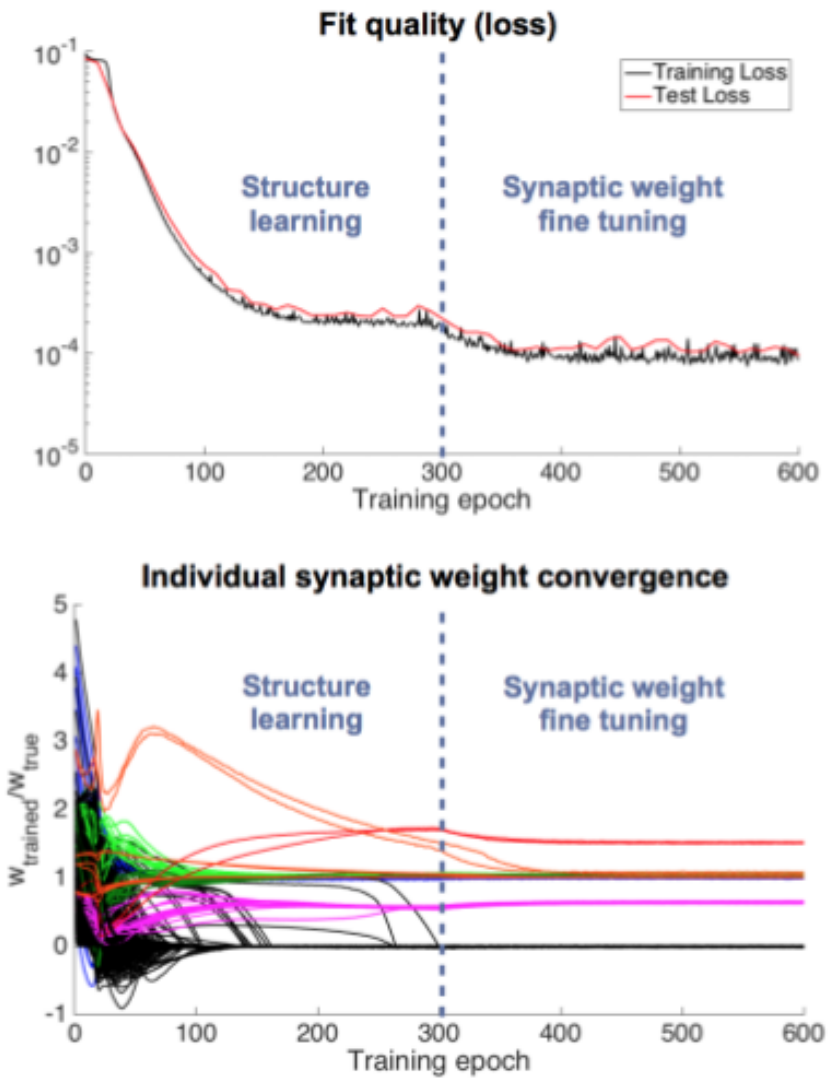

Figure 4.25: Top: Training and test loss over the course of training for ooDS circuit. After 300 epochs, there is an escape from a local minimum to a structure that produces lower loss. Bottom: $\frac{w_{\text{trained}}}{w_{\text{trained}}}$ plotted for weights of different types in the trained ANNs. Black lines represent weights that do not belong in the true circuit. All other curves are colored according to the color scheme in Fig. 4.4. At 300 epochs, the last unecessary synapses are pruned to zero, and this corresponds to the shift in the loss function to a better solution. We therefore refer to the first half of training, prior to this shift, as the "structure learning" phase, and the latter half as the "fine tuning" phase, in which the synaptic weights of the remaining synapses are adjusted to their final values.

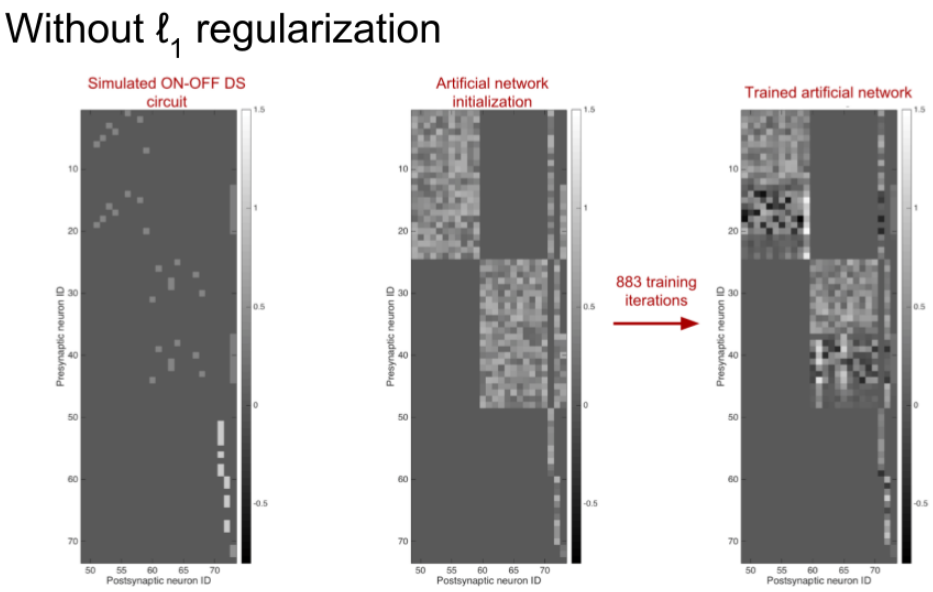

Figure 4.26: Left: Weight matrix of a simulated ON-OFF DS circuit. Entry (i, j) represents the magnitude weight between neuron i and neuron j. Center: Weight matrix of initialized ANN. Right: Weight matrix of trained ANN. When ℓ_1-regularizer is not used, the structure of the ground truth circuit is not recovered. Many extra synapses remain in the trained ANN.

Or, are there so many local minima and such complexity to the error surface being searched that a regularizer is required to help the network identify the correct, sparse solution? To test this, we re-trained the network without the ℓ_1-regularization. We found that in this case, system identification failed. The network converged to a local minimum with higher loss, which contained many "extra" synapses that did not correspond to anything in the true circuit (Fig. 4.26). Later in this work, we study this idea systematically, for a variety of circuit architectures with various amounts of training data and free parameters to understand whether it holds universally.

To summarize, Fig. 4.27 shows that training the ANN without regularization leads to a poor projection score for both hypothesis circuits. Training it with only moving dot stimuli also leads to poor projection scores for both hypotheses, and in fact, the incorrect hypothesis has a higher projection score with the learned ANN at the end of training. However, when sparsity regularization is implemented and the more informative moving random pattern stimulus is used, the true circuit has a much higher projection score with the learned ANN, close to 1, and the incorrect hypothesis has a low projection score, around 0.3. Thus this experiment would indicate strongly to the hypothetical researcher that the circuit uses the Tran structure

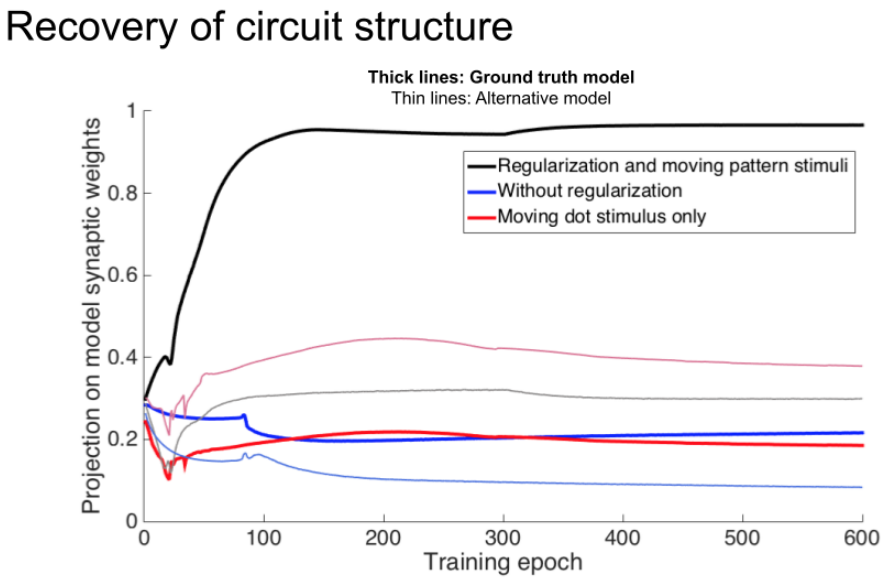

Figure 4.27: Projection score over the course of training for three trained ANNs. Thick lines represent projection onto ground truth circuit structure. Thin lines represent projection onto alternative hypothesis. Black lines represent ANN trained on a dataset of one hour of moving random pattern stimuli with an ℓ_1-regularizer. Red lines represent the same ANN trained on moving dot stimuli only, with an ℓ_1-regularizer. Blue lines represent the same ANN trained on moving random pattern stimuli without an ℓ_1-regularizer.

and not the Kim structure. This is the correct conclusion. This gives us confidence that given an appropriate choice of stimulus and regularizers, this method can be quite effective at correctly distinguishing between competing circuit hypotheses.

The quantity of training data required for circuit recovery

Biological experiments are time-limited, especially when live tissue is involved. Most electrophysiological techniques for the retina involve enucleating the eye, removing the retina from the eye cup, and keeping it alive in an oxygenated nutrient bath during the recording. It is quite difficult to record action potentials from retinal ganglion cells in an intact animal. Despite the experimenter's best efforts to keep the retina comfortable in the nutrient bath, the trauma of removal from the animal as well as the lack of bloodflow to the tissue ultimately results in death of the retinal neurons. However, a skilled experimenter can keep the retina alive outside of the animal for a few hours before this happens. During this time, it is crucial to present stimuli efficiently in order to extract as much information as possible about the retinal circuitry. One might wonder whether sufficient data can be recorded during

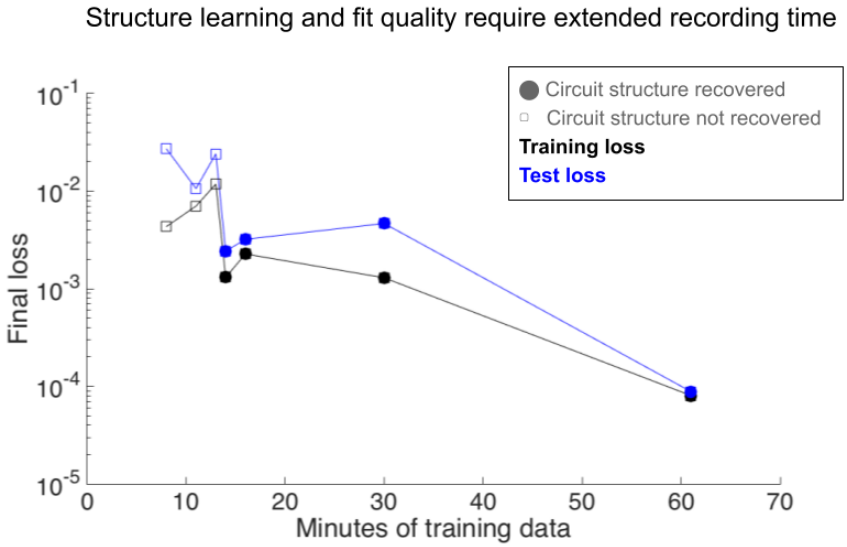

Figure 4.28: Final training and test loss is plotted for ANNs trained on varying quantities of training data. Final learned structures were examined by eye and qualitatively assigned to the "circuit structure recovered" (filled circles) and "circuit structure not recovered" (open squares) categories. Circuit recovery is possible with at least 15 minutes of training data or more, and this corresponds also to a sharp decrease in the final train and test loss. Final loss decreases when more training data are used.

this time to do something as intensive as train a convolutional neural network, which are notorious for requiring large amounts of data. To check whether this would be in the realm of possibility, we titrated the quantity of data provided to the ANN and measured the final loss of the network (Fig. 4.28) and found, as expected, that larger amounts of training data resulted in a smaller loss. However, we also inspected the final learned structures and checked that they matched or nearly matched the structure of the true circuit (qualitatively by eye). We found that given about 15 minutes of training data, the network was able to perform near-perfect system identification. Below that, system identification was not possible. Later in this work we quantify the success of system identification using a system recovery score, and perform this study more rigorously in multiple circuit architectures.

Incorporating noisy data

So far the simulations I have described all utilized noise-free data. However, retinal ganglion cell spike trains, and in fact, all biological datasets, include noise. Surprisingly, retinal ganglion cell spike trains are quite reliable [12]. RGC firing rates

display Poisson statistics, and there are multiple places in the retinal circuit where noise is introduced. Under scotopic conditions, when photons are scarce, noise in the retinal spike trains is dominated by photon noise (noise inherent to the detection of quanta of light) as well as photoreceptor noise [10]. That is because, at this light level, little averaging occurs at the photoreceptor-horizontal cell gap junctions, in order that the retina can respond to single quanta of light. However, other sources of noise, which become more prominent under brighter light conditions, include synaptic transmission noise in both the outer and inner plexiform layers, as well as noise generated by the ganglion cell itself during the spike-generation process [63].

To understand whether neural system identification using ANNs is still possible even under noisy conditions, I added two sources of noise to my simulated retinal ganglion cell circuit. First, I inserted additive Gaussian noise at the input to the network to model photon and photoreceptor noise. I also included additive Gaussian noise at the level of the ganglion cell generator potential, just before the nonlinearity for spike generation (Fig. 4.29.) This noise created some variability in the firing rate of the simulated ganglion cell. I then provided the network with two types of stimuli: 50 Hz random flicker and 4 Hz random flicker. The effect of the noise was more pronounced for the faster stimulus, as firing events were denser and lower in amplitude (Fig. 4.30). Under the 4 Hz stimulus condition, each firing event is very pronounced and there are only a few "false positive" events where noise pushes the cell over the firing threshold. However, under the 50 Hz condition, since all the events are small and fast, it would be difficult to distinguish by eye a noise even from a genuine stimulus-triggered firing event.

When the ANN is trained on these data, it predictably does a worse job at both response prediction and system identification. First, the final training and test loss is higher for the ANN trained on noisy data (Fig. 4.32). Additionally, the projection score for this ANN maxes out at around 0.75, while the noise-free ANN reaches about 0.9 projection score (Fig. 4.31). However, even with noise in the data, the projection score onto the true circuit is well separated from the projection score onto the alternate hypothesis circuit (0.75 vs. 0.3.) Thus, the task of distinguishing between the two circuit hypotheses can still be accomplished, even with noisy data.

Conclusion

Using a simulated ooDS circuit as a test case, we found that, surprisingly, near-perfect system identification is attainable using a biologically constrained neural network

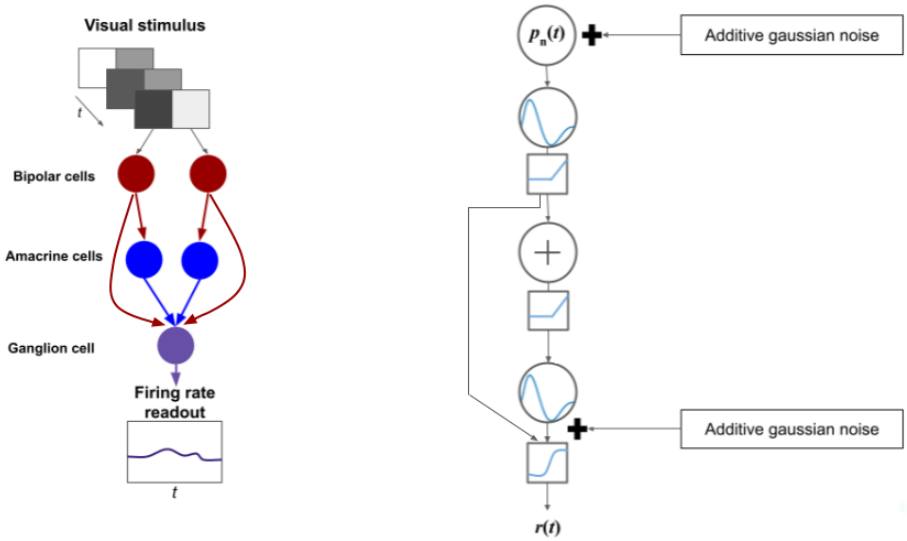

Figure 4.29: Left: Schematic of a generic simulated retinal circuit. The visual stimulus is presented to the bipolar cells, which pass the signal through the bipolar and amacrine cells to produce a firing rate readout. Right: The same circuit redrawn with computational units to demonstrate where the addition of noise occurs. Noise is added both at the stimulus as well as just before the output nonlinearity.

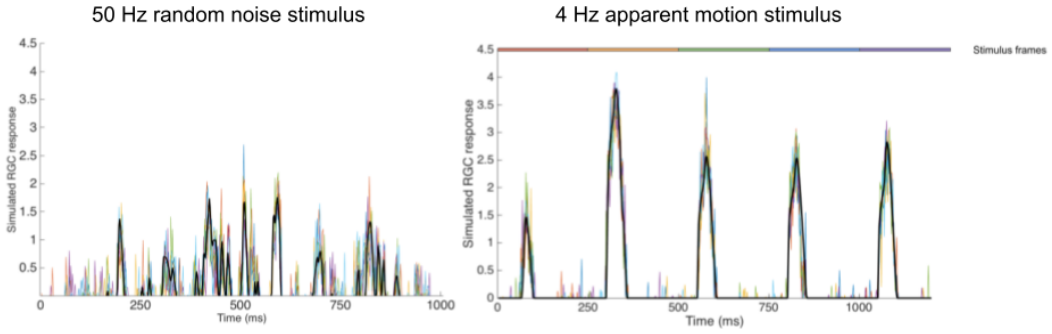

Figure 4.30: Responses of the ooDS circuit to two types of stimuli when noise is modeled as described.

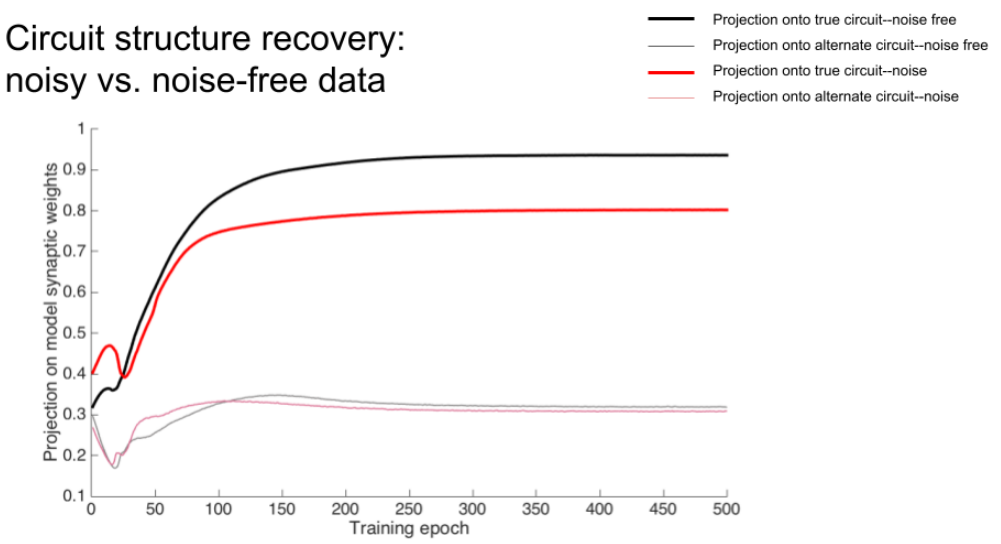

Figure 4.31: Projection score over the course of training for two trained ANNs. Thick lines represent projection onto ground truth circuit structure. Thin lines represent projection onto alternative hypothesis. Black lines represent ANN trained on a dataset without noise. Red lines represent an ANN trained on noisy data. The noise-free ANN attains a higher projection score by the end of training.

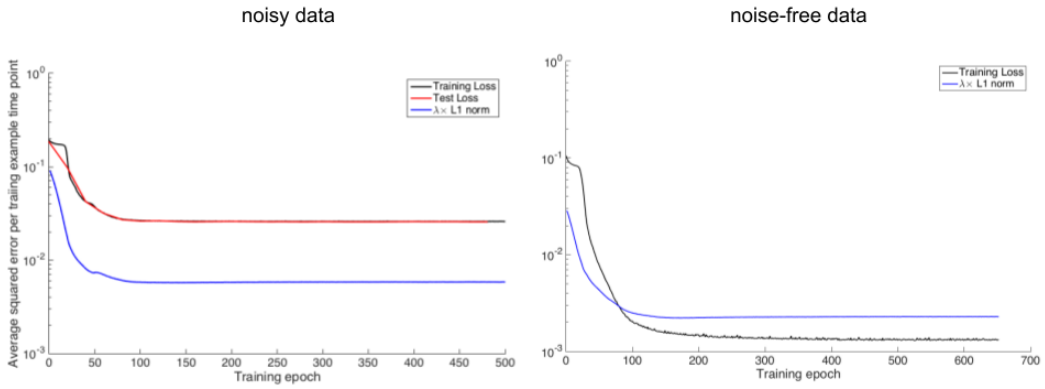

Figure 4.32: Training loss, test loss, and ℓ_1 norm are plotted over the course of training for the two ANNs, one trained on noisy data and one trained on noise-free data. The noise-free ANN converges to a lower final error rate.

with a very basic training algorithm. This is not an intuitive result. Nonlinear system identification is thought to be difficult, and the ooDS cell circuit is highly nonlinear, and performs quite a complex visual computation. What makes this possible? We hypothesized that due to the heavy constraints on the ANN derived from biological understanding of the retina, the space being searched was so small that there is really only one global minimum of the objective function in parameter space, and that this corresponds to the correct solution, and we would go on to test this idea in simulation and in theory. But many questions remained. Was this simply an idiosyncrasy of the ooDS circuit? Was it possible to do system identification more generally with other circuit architectures? We found that the ℓ_1-regularizer was a necessary constraint for system identification. Which other constraints were vital, and could we gain a more general understanding of the hyperparametric regime in which this type of system identification is feasible? The remaining sections of this chapter describe our deep dive into the more general problem of nonlinear system identification using ANNs. We maintain a connection to biology in the generic architectures of the circuits we will study, but we try to attain a more general understanding so that our findings might transfer to other applications as well. If the success we had with the ooDS circuit is reflective of the general ease and success of this method, this could present a powerful and broadly applicable new direction in which nonlinear system identification will move in the future. As deep learning progresses technologically and new, more effective algorithms and regularizers are designed, this general framework can be adapted to myriad applications. But first, a rigorous understanding of its success and failure modes is vital.

4.4 Implementing an "address book" constraint

One interesting potential biological constraint pertains to the genetic "address book" described in Section 2.1. To test this constraint, we create a "toy retina" setting and began exploring its utility. The first toy retina was designed to include three bipolar cell types, called B1, B2, and B3, and three ganglion cell types: G1, G2, and G3. B1 cells provide excitatory input only to the G1 ganglion cell, B2 to G2, and B3 to G3 (Fig. 4.33). This is the ground truth circuit. It was simulated and used to generate a training data set. Each bipolar cell in the circuit was a linear-nonlinear unit, which took a temporal convolution of a constrained spatial region of the input stimulus with a temporal filter derived directly from real bipolar cells in mouse retina [22]. The output of this convolution was then passed through a ReLU nonlinearity, multiplied by a synaptic weight matrix, and passed to the ganglion cell layer. Each ganglion

cell layer simply took its weighted input and passed it through an additional ReLU nonlinearity to produce a time-varying output in response to the stimulus video.

We then imagined a fictional anatomical study, in which an experimenter performed a careful genetic dissection of the inner plexiform layer (IPL) of this toy retina. The results of this fictional study are drawn in figure 4.35 left. This toy retina IPL contains five distinct laminae. Based on the stratification profile of each of the cell types, a putative connectivity matrix is drawn in figure 4.35 right. Note that every square of this table is filled in, meaning that every bipolar cell type could potentially connect to every ganglion cell type. This is, therefore, the IPL address book for this retina with the least possible information, and the fewest possible constraints on connectivity. It should therefore be the hardest problem to solve, and we planned to show that it was difficult and then make it progressively easier by using sparser and sparser "address books." However, as we will show, this problem was already very easy for the network to solve and near-perfect system identification was achieved without addtional constraints (Fig. 4.37).

From this IPL address book, we then derived the initialization for the ANN (Fig. 4.36 left), which meant creating and randomly initializing synapses from all three bipolar cell types to all three ganglion cells. The task of training, then, was to prune this fully connected network down to one with much sparser connectivity (Fig. 4.36 right). Fig. 4.37 illustrates this process. The trained synaptic weight matrix is almost identical to the true weight matrix, the only difference being very, very slight variation in the precise values of the synaptic weights. This tiny variation did not affect the output fit in any visible manner (Fig. 4.38.) Thus under these conditions, using the very broadest possible address book constraint (essentially a null constraint) the circuit could still be easily recovered. We therefore re-designed the learning problem somewhat in order to make it more difficult and see if the address book constraint would be more useful in that scenario.

The second toy retina included more cell types, specifically it also included two types of amacrine cells, modeled with ReLU output nonlinearities, which we call A1, and A2 (not related in any way to the AII amacrine cell type.) These two amacrine cell types differ in that A1 is wide-field, with a large spatial receptive field which sums over many bipolar cells, whereas A2 is narrow field, and each amacrine cell of this type only connects to a single bipolar cell (Fig. 4.40.) The IPL stratification profile for this toy retina and the putative connectivity matrix to which it gives rise are shown in Fig. 4.35. This connectivity matrix is sparse. Many entries of the table

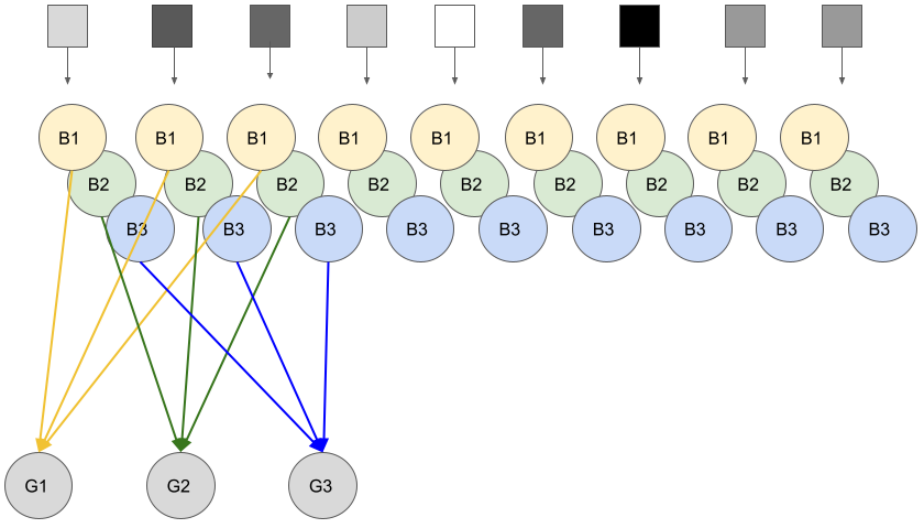

Figure 4.33: Circuit diagram of a network with three bipolar cell types and three ganglion cell types. Greyscale squares represent stimulus pixels. Each bipolar cell unit has a single-pixel spatial receptive field.

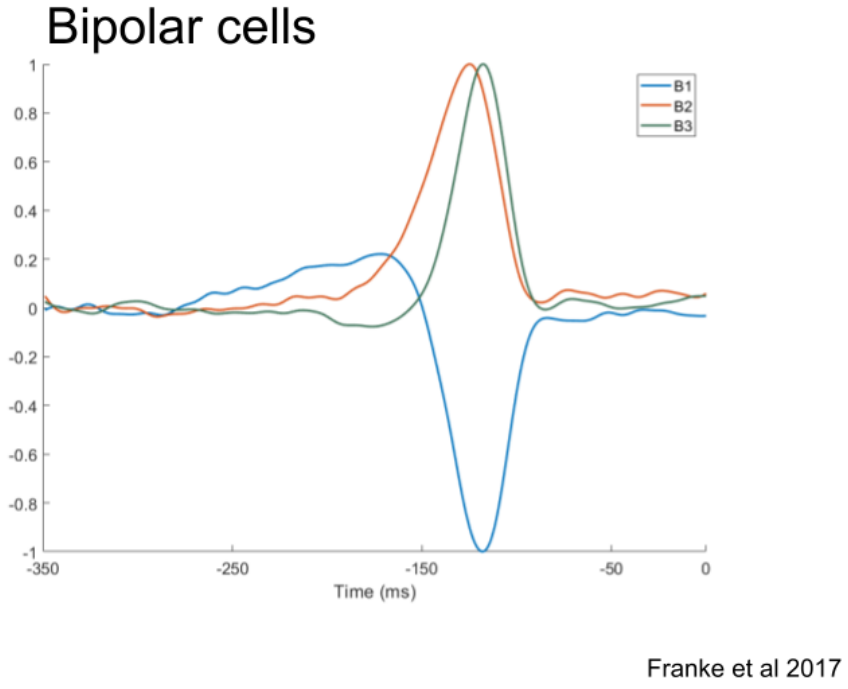

Figure 4.34: Temporal convolutional filters assigned to each of the three bipolar cell types in Fig. 4.33.

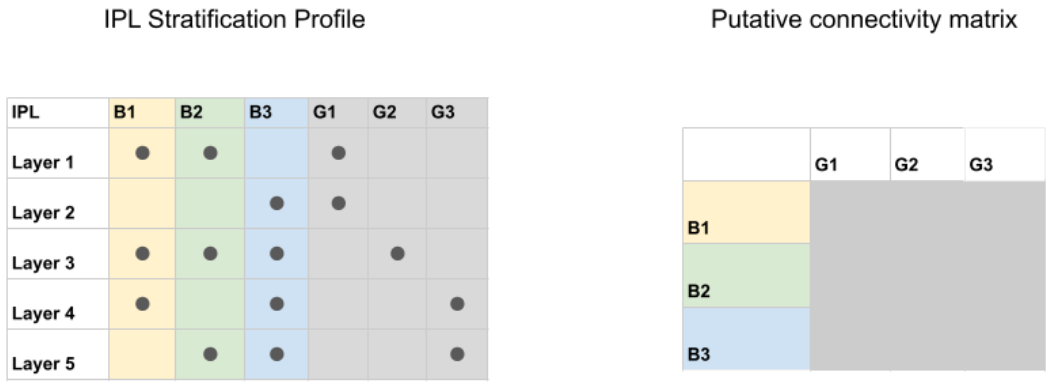

Figure 4.35: Left: Fake inner plexiform layer used in this experiment. Black dots represent layers in which each cell type stratifies. Right: Putative connectivity matrix based on this fake IPL. Filled in entries represent possible connections. Every entry is filled in, meaning that every bipolar cell type could potentially connect to every ganglion cell type.

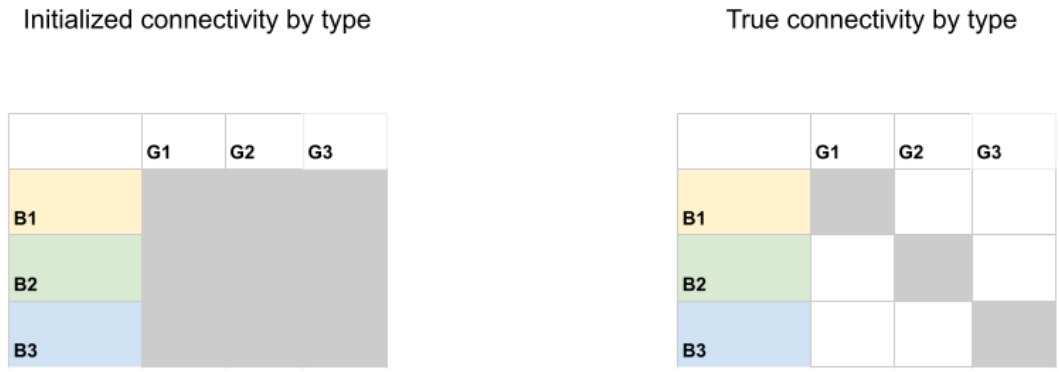

Figure 4.36: Left: Connectivity by type in the initialized ANN. Filled in squares represent nonzero connections in the ANN. Right: True connectivity by type, derived from Fig. 4.33

.

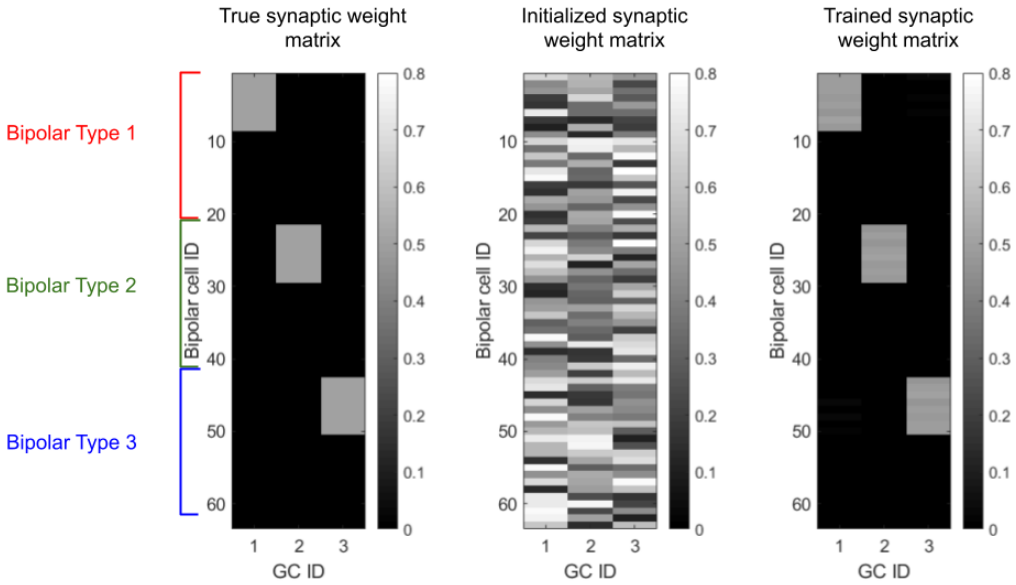

Figure 4.37: Left: Synaptic weight matrix for true circuit. Center: Initialized synaptic weight matrix for trained ANN. Right: Weight matrix of trained ANN after training is completed. The correct structure and weights are recovered.

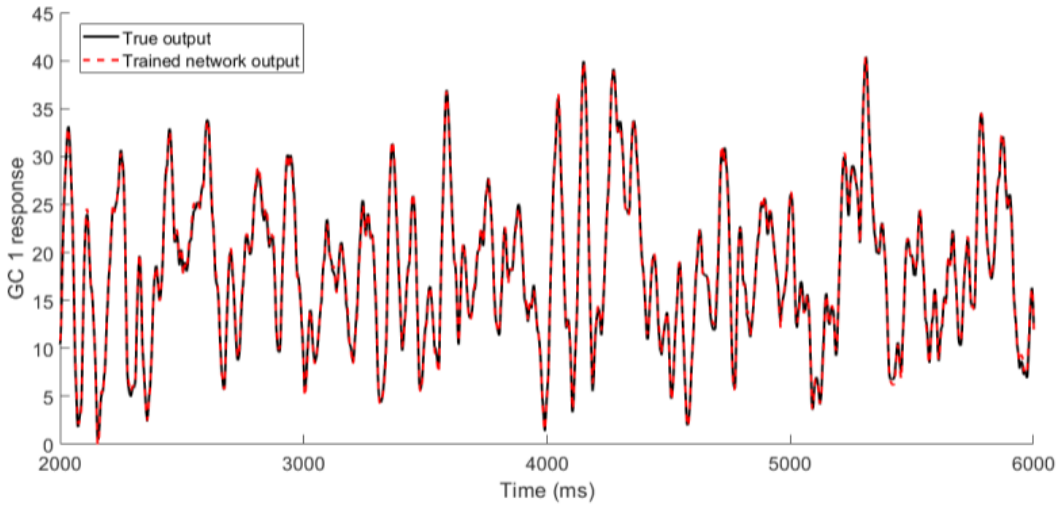

Figure 4.38: Fit of the trained ANN output to the test data is virtually perfect.

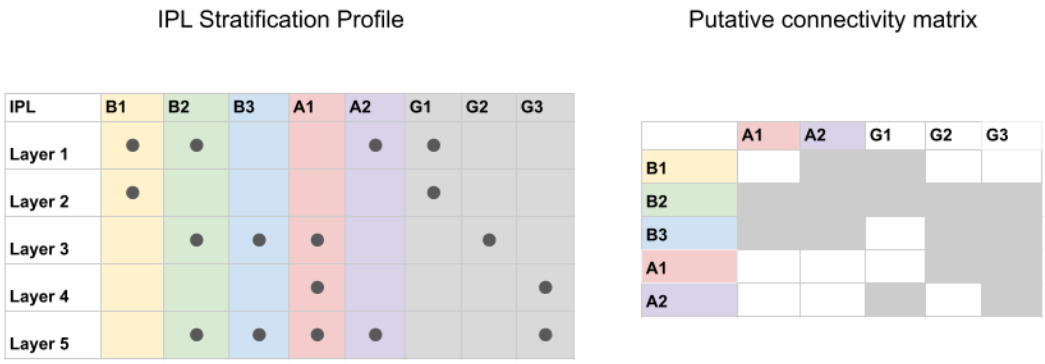

Figure 4.39: Left: Fake inner plexiform layer used in this experiment. Black dots represent layers in which each cell type stratifies. Right: Putative connectivity matrix based on this fake IPL. Filled in entries represent possible connections.

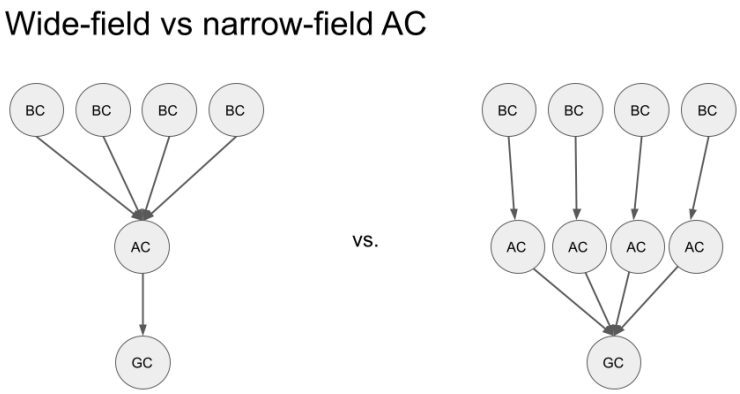

Figure 4.40: Schematic diagrams representing the two types of amacrine cells used in this experiment. The wide-field amacrine cell pools over many bipolar cells, while the narrow-field only gets input from a single bipolar cell.

are not filled in, indicating that based on the IPL stratification profile, it would be impossible for those two cell types to connect to one another. Thus, the ANN can be initialized without any of these synapse types included, reducing the number of free parameters and therefore the complexity of the learning problem.

Fig. 4.41 shows the initialized connectivity matrix and also the true connectivity matrix, corresponding to the "ground truth" circuit which was simulated and from which a training dataset was generated. The synaptic weight matrix for this circuit is shown in Fig. 4.42 on the left. The ANN is then initialized to include all synapses considered legal under Fig. 4.39 with randomly initialized weights (Fig.

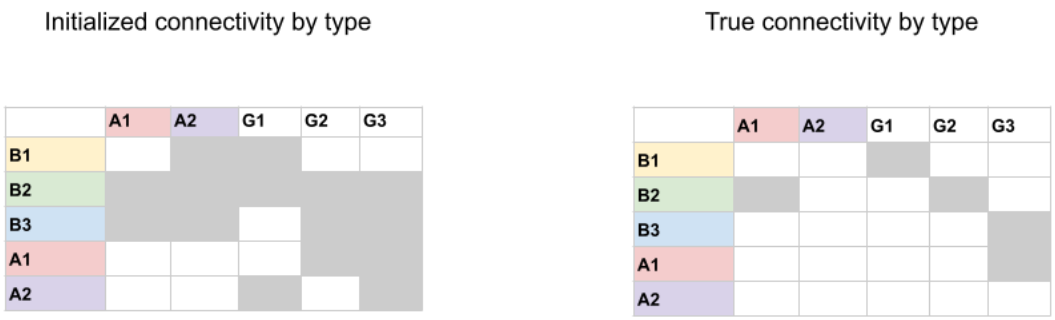

Figure 4.41: Left: Connectivity by type in the initialized ANN. Filled in squares represent nonzero connections in the ANN. Right: True connectivity by type.

4.42 middle.) This ANN was randomly initialized and trained three separate times, and the results are shown in the rightmost panels of Figs. 4.42, 4.43, and 4.44. These results are summarized in Fig. 4.46, focusing only on the third ganglion cell type, since the other two circuits were correctly identified in every case. To the fake experimenter studying this toy retina, this is the set of possible circuits giving rise to the data collected from G3. As shown, in the first two trainings, the correct circuit structure for G3 is identified by the ANN. However, in the third training, the learned circuit is slightly different from the true circuit. The structure learned in this case is shown in Fig. 4.45. The difference in this circuit is that one set of bipolar cells, rather than inhibiting the ganglion cell via a widefield amacrine cell, connects directly to the ganglion cell with negative weight.

This raises an interesting point. We, as retinal neuroscientists, know that such a circuit is "illegal." That is, the bipolar cell to ganglion cell synapse is always an excitatory one. Therefore, the circuit learned in this third round of training is biologically implausible and will be tossed out of the hypothesis set by the experimenter. But since we knew this information about bipolar cells in advance, could we have prevented the ANN from ever even providing such a hypothesis? The answer is yes, and it requires nothing more than a simple sign constraint on each of the weights. In the retina, we can classify almost all synapses as either "excitatory" or "inhibitory" with confidence. We know, for example, that almost all synapses between amacrine and ganglion cells are GABA- or glycinergic, and inhibitory. In contrast, bipolar cell to ganglion cell synapses are glutamatergic and excitatory. Therefore, we can constrain the space searched over the course of training to only circuits which follow these rules. This is done in later experiments by simply putting an absolute value around the synaptic weight variables in the Tensorflow code.

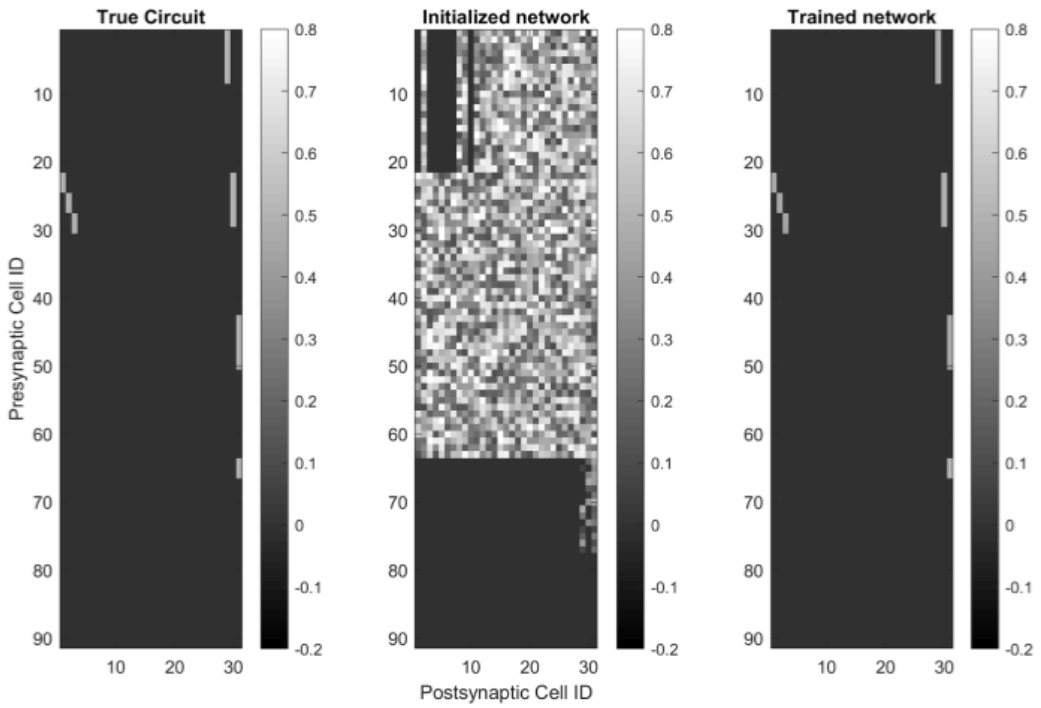

Figure 4.42: Left: Synaptic weight matrix for true circuit. Center: Initialized synaptic weight matrix for first trained ANN. Right: Weight matrix of trained ANN after training is completed. The correct structure and weights are recovered.

The sign constraint aside, this experiment with the toy retina is a successful example of how an IPL constraint can be employed in a real biological situation. IPL stratification profiles like those invented in Figs. 4.35 and 4.39 have already been elucidated for many retinal cell types [7, 27, 58, 85, 93], and can be put to use for algorithmic system identification in this way.

4.5 Study of the parametric regime in which system identification is feasible

In this section, we empirically analyze conditions under which ℓ_1-regularized regression can lead to successful fine-grained identification of neural circuits, and cross-reference with the theoretical results from Section 3.3 where appropriate. Practical system identification of neural networks can be challenging even with significant constraints imposed (e.g., many weights preemptively constrained to zero). Most applications of artificial neural networks (ANNs) are not concerned with identification of the "true" network, but rather focus on predictive performance (which can be achieved by many parameterizations). Therefore, it is important to understand, operationally, when system identification can be reliably performed.

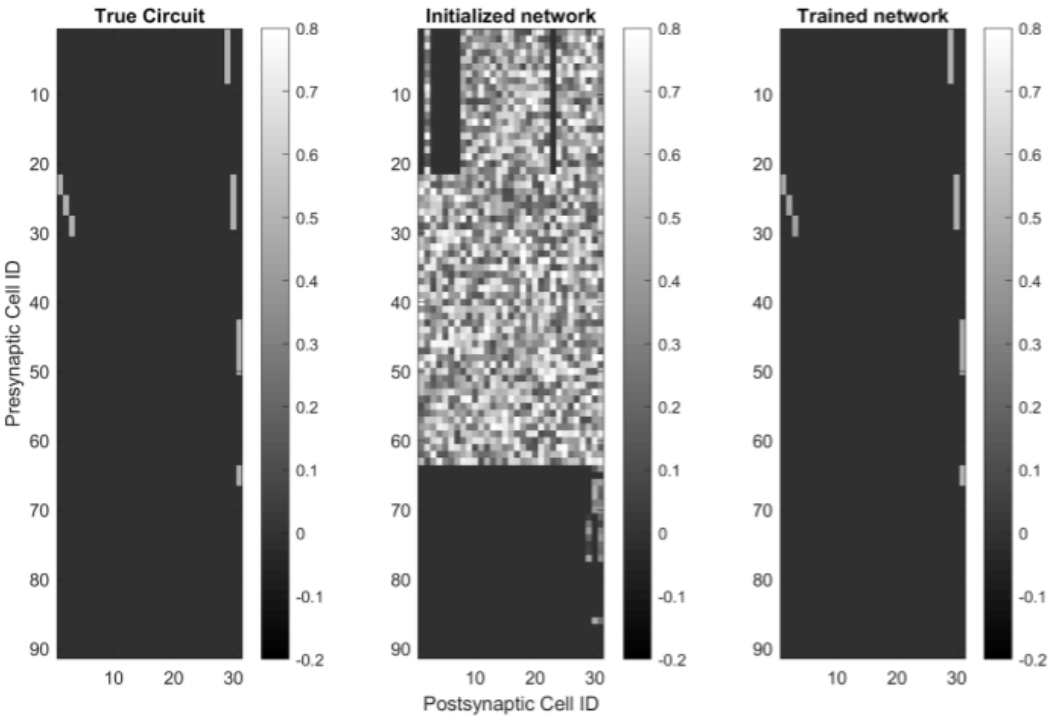

Figure 4.43: Left: Synaptic weight matrix for true circuit. Center: Initialized synaptic weight matrix for second trained ANN. Right: Weight matrix of trained ANN after training is completed. The correct structure and weights are recovered.

Since we assume that the true neural circuit is sparse, we employ ℓ_1-regularized regression. Given a training set $S = \{(x, y)\}$, tuning parameter for ℓ_1-regularization λ, neural circuit $f(\cdot; \mathbf{W})$, and set of allowable weights , the optimization problem under study can be described as:

$$\mathbf{W}_k \leftarrow_{\mathbf{W}\in} \sum_{(\mathbf{x},y)\in S} \|f(\mathbf{x}; \mathbf{W}) - y\|^2 + \lambda|\mathbf{W}|. \tag{4.2}$$

Due to the nonconvex nature of this optimization, it is preferable to run it with multiple (K) random initializations, thereby producing a set of solutions $\mathbf{W}_1, \ldots, \mathbf{W}_K$.

Using controlled simulated settings, we performed a systematic study to understand the dependence of successful system identification on the following factors:

(i) The design of the function class $f(\cdot; \mathbf{W}, \mathbf{b})$, and in particular the presence of skip connections in the architecture.

(ii) The number of non-zero weights $\mathbf{W} \in$, via preemptively setting many connections to zero.

(iii) The use of sign constraints on the weights $\mathbf{W} \in$.

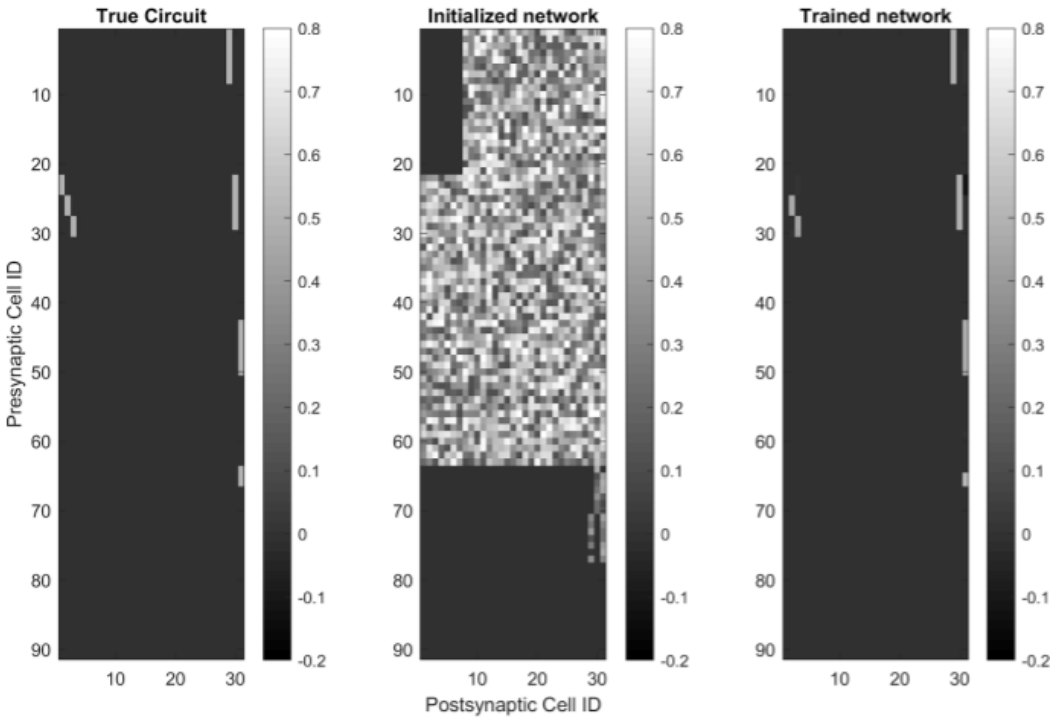

Figure 4.44: Left: Synaptic weight matrix for true circuit. Center: Initialized synaptic weight matrix for third trained ANN. Right: Weight matrix of trained ANN after training is completed. The correct structure and weights are almost perfectly recovered, with some differences.

(iv) The use of ℓ_1 regularization to encourage weight sparsity.

(v) The design of the training dataset, i.e., which data points to query.

In each simulated experiment, we created a sparse oracle network, and then trained a second network to recover that structure via the optimization problem in Eq. 4.2, while varying the constraints, regularizers, and datasets.

We define two measures of success: [System Recovery Score] Let the system recovery score be defined on the vectorized weight matrices as:

$$R_{\text{sys}} = 2 \times \frac{\mathbf{W}_{\text{learned}} \cdot \mathbf{W}_{\text{oracle}}}{|\mathbf{W}_{\text{learned}}|^2 + |\mathbf{W}_{\text{oracle}}|^2}. \tag{4.3}$$

[Support Recovery Score] We will denote by $\overline{\mathbf{W}}$ the vector of the same size as $\mathbf{W}$ where $\overline{\mathbf{W}}_i = 0$ if $\mathbf{W_i} = 0$ and 1 otherwise. Let the support recovery score be defined on the vectorized and binarized weight matrices as:

$$R_{\text{supp}} = 2 \times \frac{\overline{\mathbf{W}}_{\text{learned}} \cdot \overline{\mathbf{W}}_{\text{oracle}}}{|\overline{\mathbf{W}}_{\text{learned}}|^2 + |\overline{\mathbf{W}}_{\text{oracle}}|^2}. \tag{4.4}$$

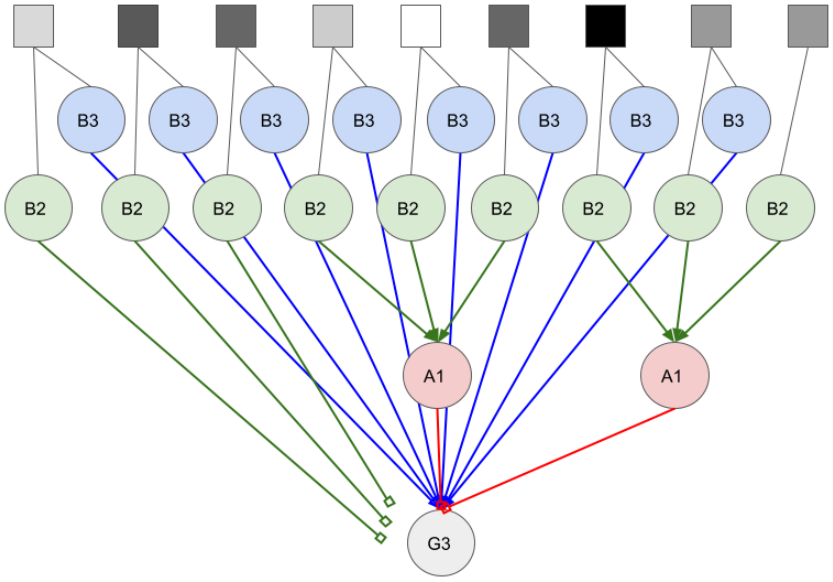

Figure 4.45: Circuit diagram representing the structure learned in the experiment shown in Fig. 4.44. Everything matches the true circuit, except that some of the B2 bipolar cells directly inhibit G3, rather than doing so via an amacrine cell, which is biologically implausible.

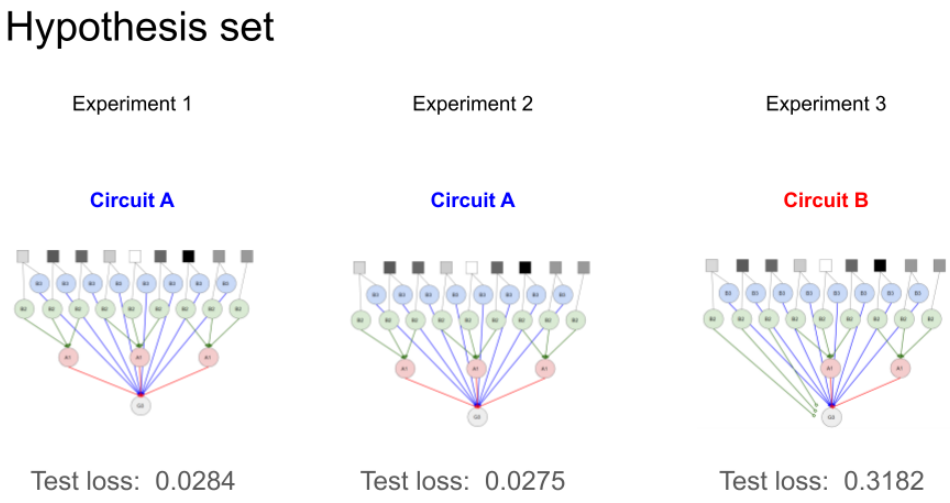

Figure 4.46: The three training experiments give rise to three circuit hypotheses. The first two are the same and are labeled Circuit A. This turns out to be the correct circuit. The third is slightly different and is labeled Circuit B. Final test loss is listed in each case. Note that experiment 3, which leads to an incorrect circuit structure, also has a higher final test loss.

Both measures are related to cosine similarity and the Jaccard coefficient. The System Recovery Score achieves perfect performance when the exact system is recovered with the exact weights. The Support Recover Score achieves perfect performance whenever the non-zero support (i.e., the structure) is recovered exactly.

Summary of findings

Fig. 4.48 shows our main findings on the practicality of the approach. For all simulated circuits, we are able to reliably recover most of the structure and parameters using ℓ_1-regularized regression, when there are around 500 non-zero, sign-constrained free parameters in the network. As we will see in Chapter 5, the biological problems we are interested in have a few hundred free parameters, and so are closely modeled by the black curves in Figs. 4.47 and 4.53.

Fig. 4.53 shows that successful recovery can be contingent on the use of ℓ_1-regularization as well as sign constraints. In particular, we see that ℓ_1-regularization is especially crucial when the circuit has skip connections (as is common in the retina), and that the estimation problem fails completely without sign constraints. The former finding potentially suggests skip connections as an important factor for finite-sample analysis of system identification of nonlinear circuits, and the latter finding supports our theoretical finding that sign constraints are an important condition for guaranteeing identifiability.

Network architecture

We focus on the most generic network architectures that are prevalent in retinal biology—though similar architectures arise in gene regulatory networks [52]. We call these networks:

- Circuit 1: Three-layer networks whose first layer is convolutional and fixed, and whose output layer is nonlinear (Fig. 4.47a).
- Circuit 2: Three-layer networks whose first layer is convolutional and fixed, whose output layer is linear and which also include skip connections (Fig. 4.47b).
- Circuit 3A: Circuit 2 with the addition of a low-threshold ReLU output nonlinearity (Fig. 4.47b).

- Circuit 3B: Circuit 2 with the addition of a high-threshold ReLU output nonlinearity (Fig. 4.47b).

Note that Circuit 2 is, in fact, the circuit studied theoretically in Section 3.3, but with a restricted input domain. In the retina, one can think of the output of the first layer as modeling bipolar cell glutamate release rates, the second layer as modeling amacrine cells, and the third as modeling ganglion cells, though this "feedforward loop" circuit motif also appears in other domains, most prominently in transcriptional regulatory networks [3, 30].

For each of these four architectures, we constructed a specific oracle circuit by selecting a fixed set of weights. Circuit 1 had 36 nonzero synapses. Circuits 2, 3A, and 3B had 96 nonzero synapses. For each of these four circuits, we generated a training dataset by presenting images to the circuit and recording the responses of the output units. The rest of Chapter 4 will analyze recovery performance of these architectures while varying the other factors of interest.

Non-zero connection constraints are helpful for identification of some architectures

We first considered varying the number of free parameters in the ANN. Intuitively, the more free parameters there are, the harder it will be to recover the true network. However, constraining the network to fewer parameters would require increasing amounts of domain knowledge.

Existing neuroscience domain knowledge constrains many connections to zero weight. In the retina, such connectivity constraints are implemented by the precisely organized anatomy of neurons. The so-called inner plexiform layer (IPL) is a meshwork of synapses between different unit types (bipolar, amacrine, and ganglion cells). Each type sends its axons and dendrites into a distinct lamina of the IPL, and neurons may connect to each other only if they co-stratify in at least one lamina. This implements an "address book" of allowable connectivity between cell types in the network [73]. Using anatomical studies in the literature [7, 22, 26, 29, 40, 58, 85, 93], we have compiled such an address book for 36 retinal cell types (Fig. 2.4). A strong address book constraint translates to a small number of free parameters.

For convenience, the first layer of weights (W_1) were also constrained so that each amacrine cell analogue in layer 2 only received input from a restricted spatial region of the output of layer 1. This is to remove permutation symmetry of W_1 and W_2, and

it is also realistic, since it is known that amacrine cells in the retina typically pool inputs over a small region in space, not the entire visual field. Scaling symmetry of the weights was removed by fixing biases in place during training.

Fig. 4.48 shows our main results, where we also employ sign constraints (we later conduct an ablation study). We see that all circuits can be (approximately) identified given enough training data and if sufficiently constrained in the number of free parameters (around 500). Circuit 1, which is the only circuit without skip connections, always yields high system and support recovery scores (given enough training data) irrespective of the number of free parameters. However, the successful recovery of Circuits 2, 3A, and 3B exhibits a strong dependence on the number of free parameters, where recovery can be unsuccessful despite a large amount of training data. Our theory directly implies that Circuit 2 should be identifiable, which suggests that the non-convex optimization landscape of ANN training and the restricted input domain may be a significant factor here.

Test loss is not always useful heuristic for structure recovery

Figures 4.49, 4.50, 4.51, and 4.52 display the final training and test losses computed for the experiments plotted in figure 4.48. We can see that in the case of Circuit 1, higher structure and system recovery scores seem to correlate with lower final training and test losses (Fig. 4.49). However, the same does not hold for Circuit 2. In this case, even though smaller numbers of free parameters led to significantly better system identification, the final training and test loss seems nearly identical for all experiments where training data exceeded 1000 examples (Fig. 4.50). In circuits 3A and 3B, there was yet another interesting behavior, in which all final losses appear to be identical except for in the case of the smallest number of free parameters, in which case the final loss was lower by multiple log units (Figs 4.51 and 4.52). This means that in the case of those two circuits, having near perfect system identification also translated to a significant decrease in the loss. However, differences between experiments where the system and structure recovery scores were below one were not resolvable just by examining the final test loss. This gives rise to the following heuristic: For three layer circuits without skip connections, final test loss serves as a decent proxy for system and structure recovery scores. For circuits with skip connections but no output nonlinearity, when there are more free parameters in the ANN then there are in the true circuit, this does not hold. Just because two ANNs fit the data equally well, that does not mean that they approximate the structure equally well. For circuits that include skip connections and an output nonlinearity, in the

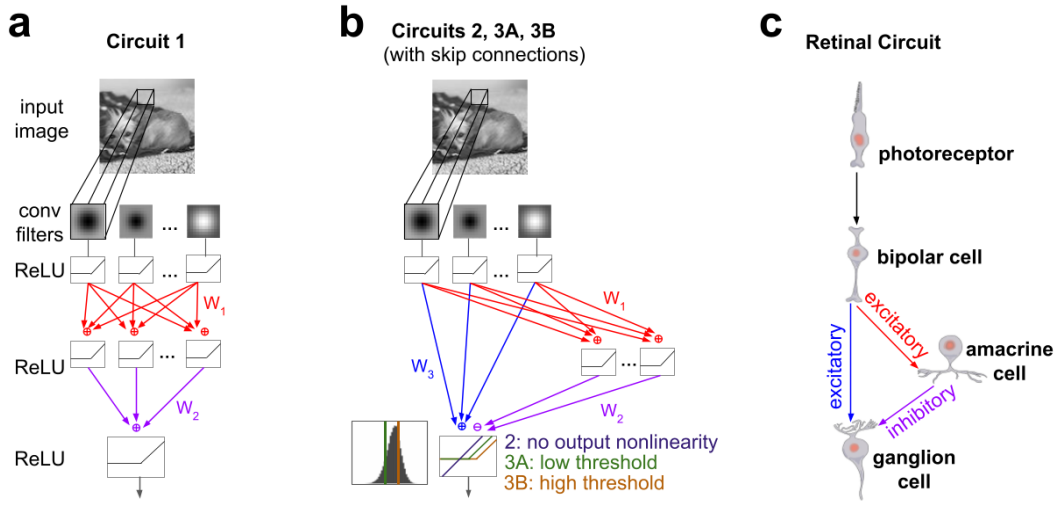

Figure 4.47: Networks studied in simulation. **(a)** Simulated circuit. **(b)** Three simulated circuits with skip connections, which vary in their use of output nonlinearity. **(c)** The retinal circuit that inspired the models in (a) and (b). The photoreceptor layer is treated as linear, and the convolutional layers in (b) and (c) represent the combined computation of photoreceptors and bipolar cells.

regime where the number of free parameters is less than or equal to ~5x the number of true parameters test loss will drop with better system identification.

4.6 Sign constraints are necessary for system identification

Another way to constrain the network is to use sign constraints, due to abundant neuroscience domain knowledge about whether certain neurons are either excitatory or inhibitory [28, 65, 88]. In fact, for the retina circuits we study in Chapter 5, we have sufficient knowledge to sign constrain every weight. Fig. 4.53 c and d shows the results of an ablation study that removes the sign constraints. We see a sharp contrast compared to Fig. 4.48 in that removing the sign constraints causes system identification via (4.2) to fail. This finding is consistent with our theoretical exploration and Theorem 1, where the sign constraint was a crucial element in identifying the half planes generated by the ReLU units.

4.7 Sparsity regularization is helpful for identification of some architectures

We next investigate the benefits of using ℓ_1-regularization in practice. Fig. 4.53 a and b shows the results of an ablation study that omits the use of ℓ_1-regularization. Compared to Fig. 4.48, we see that regularization never hurts system identification, and sometimes appears to be crucial in practice. In particular, one can still perform identification without ℓ_1-regularization for Circuit 1, which has no skip connections,

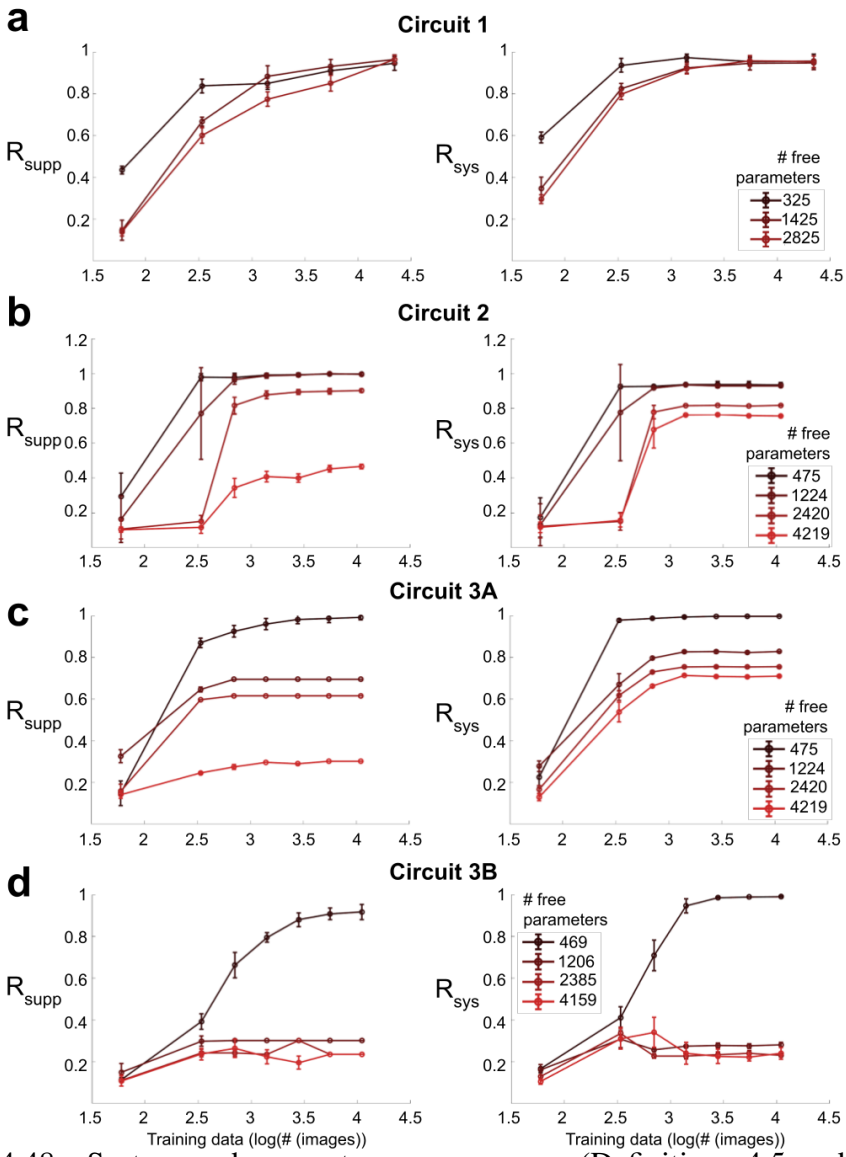

Figure 4.48: System and support recovery scores (Definitions 4.5 and 4.5) for Circuits 1, 2, 3A, and 3B respectively, using ℓ_1-regularization and sign constraints. Each curve represents an ANN with 325–4219 free parameters (see legend). The ANN was given varying quantities of free parameters and training data. For each circuit structure, a single training dataset was generated, then networks with varying numbers of free parameters were given access to some fraction of the training set and trained. System and support recovery scores were computed in each case. Mean and standard deviation of these scores are plotted for 10 training runs, each with a different random initialization. We see that one can reliably recover most of the system structure and parameters when there are around 500 free parameters in the system.

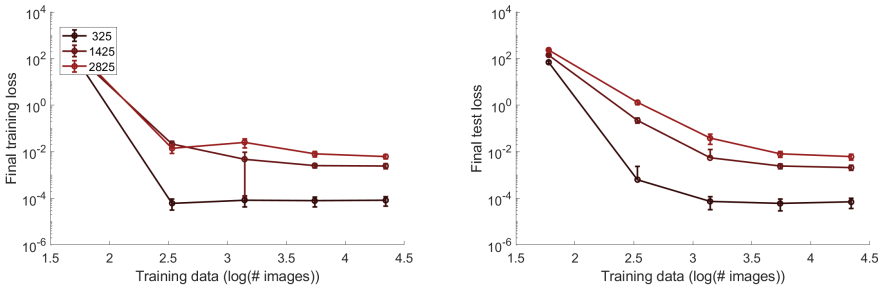

Figure 4.49: Final training and test losses plotted for the training experiments on Circuit 1 shown in figure 4.48a. Curve color represents number of free parameters, as in figure 4.48.

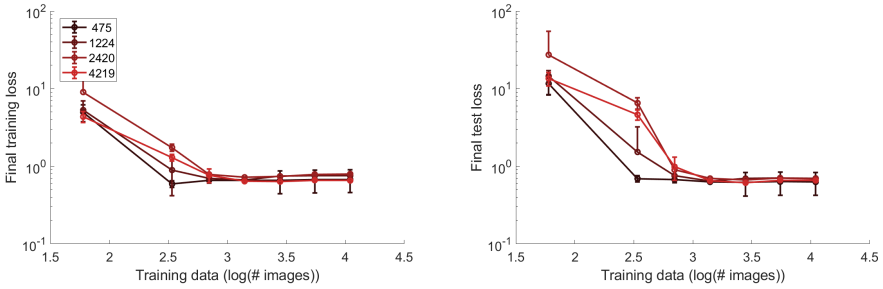

Figure 4.50: Final training and test losses plotted for the training experiments on Circuit 2 shown in figure 4.48b. Curve color represents number of free parameters, as in figure 4.48.

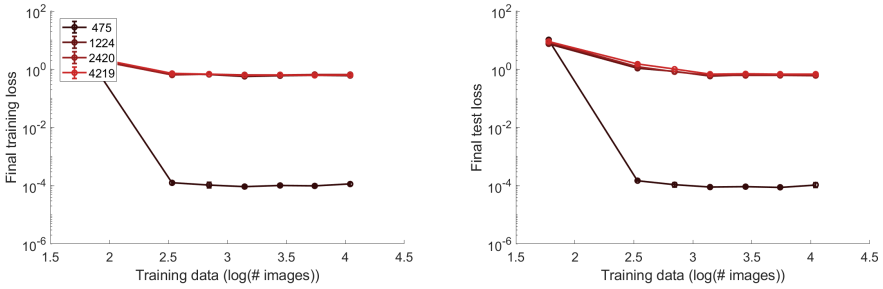

Figure 4.51: Final training and test losses plotted for the training experiments on Circuit 3A shown in figure 4.48c. Curve color represents number of free parameters, as in figure 4.48.

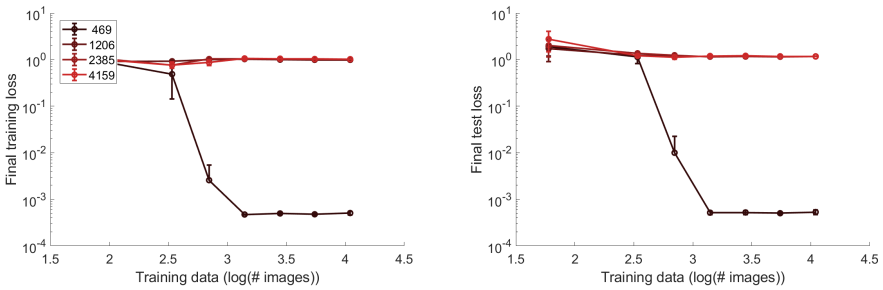

Figure 4.52: Final training and test losses plotted for the training experiments on Circuit 3B shown in figure 4.48d. Curve color represents number of free parameters, as in figure 4.48.

but not for Circuit 2, which has skip connections. We hypothesize that a theoretical characterization of sample-efficient system identification of neural circuits could depend in a non-trivial way on the presence or absence of skip connections.

The regularizer is controlled by a strength parameter, λ, which is selected via cross-validation. How sensitive is this approach to the strength of the sparsity regularizer? It turns out that there is a wide range of λ that leads to a similar level of success for system identification in a circuit with skip connections. In fact, that range spans five log units, which means that this method is not very sensitive to the value of the λ, though outside of that five log unit range, the structure recovery score does drop off steeply (Fig. 4.54). Thus, a simple cross validation should be sufficient to select an appropriate value of lambda.

Dataset design matters when data is scarce

In visual neuroscience experiments one must choose among many possible images or movies with which to query the neural circuit to collect training data. Practically, biological experiments are time-limited, and therefore, the training dataset size is limited. Unsurprisingly, system identification of all four circuits showed a strong dependence on the size of the training dataset (Fig. 4.48a-d).

Our final analysis in this section is to compare the efficacy of two very different data collection approaches: white-noise images vs. natural photographs. White noise has a long history for system identification in engineering, whereas natural images better reflect the signal domain that the retina evolved to handle. White-noise stimuli were generated as random checkerboards. Each image was 100×100 pixels, and consisted of 5×5 grayscale checkers of random intensity. Natural grayscale images were taken from the COCO (Common Objects in Context) dataset

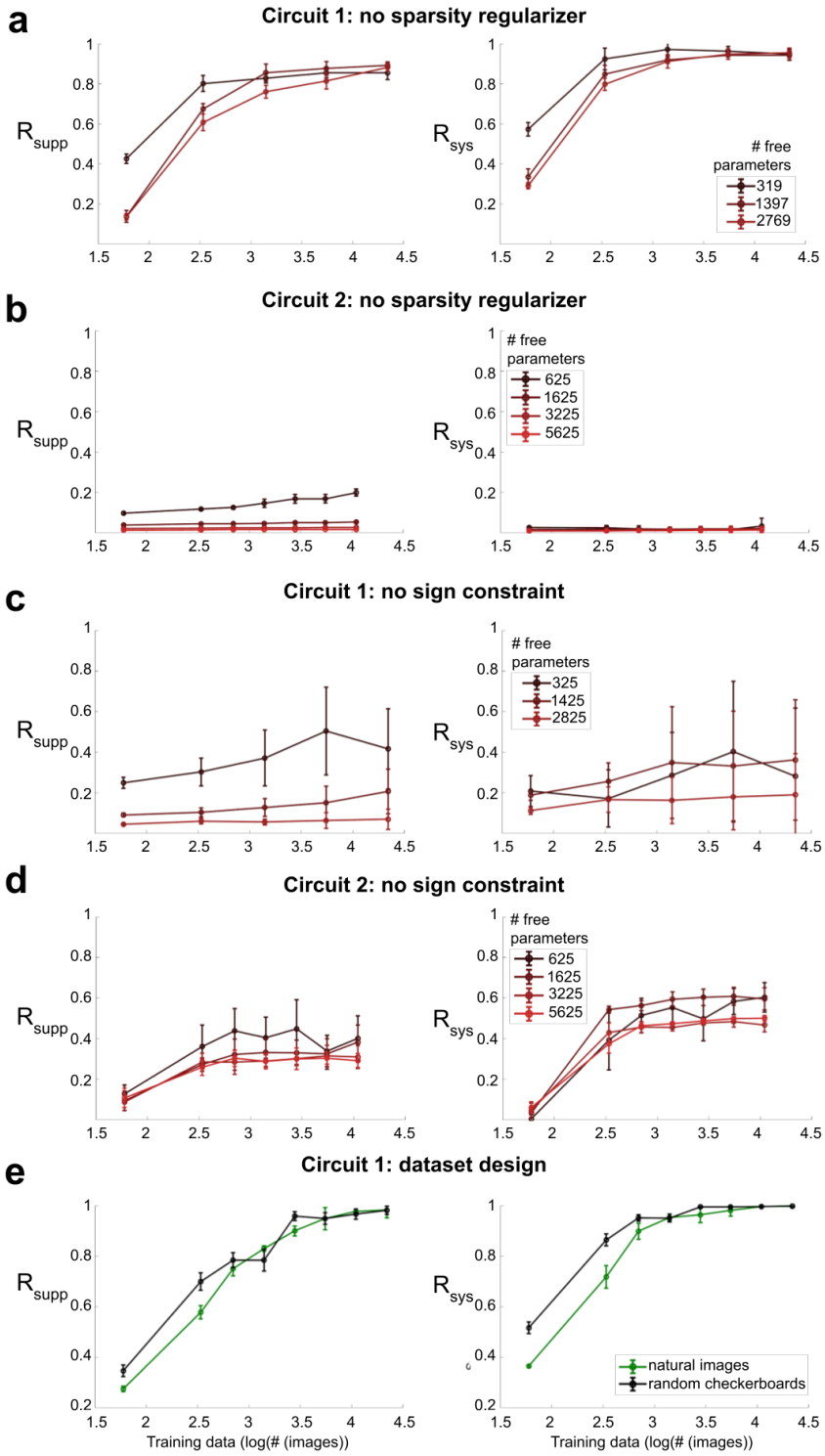

Figure 4.53: Results of ablation studies. Successful system identification can be contingent on regularization, sign constraint and dataset type. For simulated retina-style circuits: (a) System identification of Circuit 1 proceeds as normal without ℓ_1-regularization; (b) System identification of Circuit 2 is impaired without ℓ_1-regularization; (c) System identification of Circuit 1 is impaired without the sign constraint; (d) System identification of Circuit 2 is impaired without the sign constraint; (e) random noise images lead to better system identification of Circuit 1 than natural images when data is scarce.

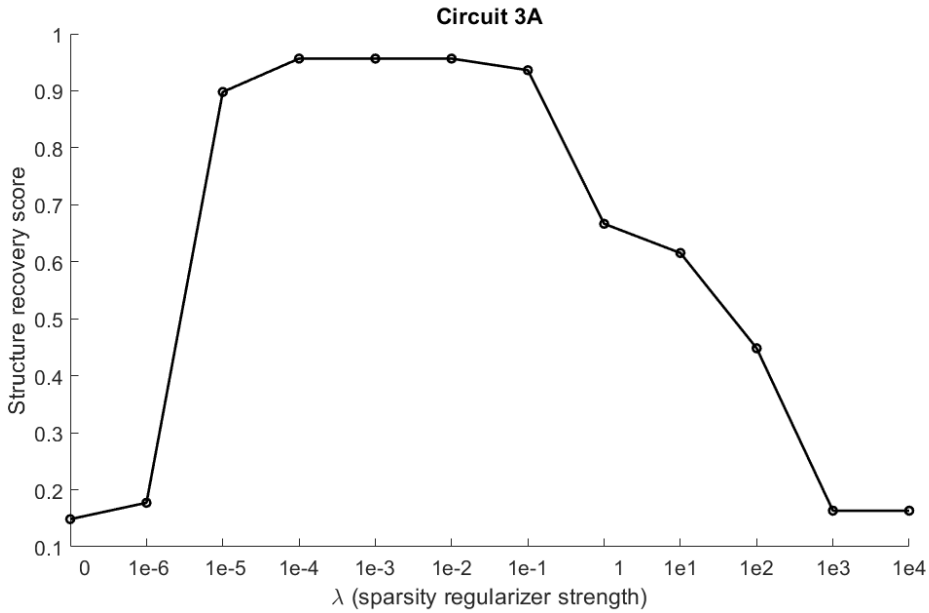

Figure 4.54: Structure recovery score for an ANN trained on data generated by Circuit 3A. The random initialization remains the same in each case, but the value of λ, which controls the strength of the sparsity regularizer, is titrated on a log scale. There is a wide regime of λ for which the structure recovery score is close to 1.

[49]. An overparameterized artificial neural network with architecture matching Circuit 1 (Fig. 4.47a) was trained with varying quantities of data from both sets. When training data were plentiful, performance on the two datasets was comparable. However, when data were more scarce, random checkerboard stimuli led to better system identification (Fig. 4.53e). Interpolating, we find that a system recovery score of at least $R_{\text{sys}} = 0.9$ can be achieved with about 400 checkerboard images, but requires 630 natural images.

4.8 A heuristic for estimating the dimensionality of an unknown circuit

The number of synapses in a given retinal circuit can be fairly well estimated based on the field's extensive understanding of synaptic convergence and divergence in the retina [13, 28, 76]. However, this does not hold in other, less understood neural systems, or in other domains. So, it would be nice to have a heuristic way to understand how many synapses to put in the initialized ANN.

The most natural place to start searching for such a heuristic is by examining the readout most readily available to a hypothetical future research: the validation loss: that is, the error computed on a held-out dataset. One might begin one's search by

initializing several ANNs with a large range in the number of free parameters, fitting each, and computing the validation loss in each case, then selecting the ANN with the lowest validation loss.

It turns out that this straightforward solution actually proves fruitful. We tested this approach on Circuit 1 described above, which has 36 synapses within it. We initialized ANNs with anywhere from 30 to 2825 free parameters, randomly initialized each one five times, and plotted the final loss on a validation set in each case. As one might expect, the loss is the lowest for the network initialized with slightly more parameters than the true circuit (1.22x). As one increases the number of free parameters in the ANN, the loss slowly increases. However, if one decreases the number of free parameters in the ANN by just 15% below the true number (a different random group of synapses deleted in each initialization), the loss shoots up dramatically, in this case by eight log units (Fig. 4.56).

How does system identification itself fare under these conditions? As described in previous sections, for Circuit 1, system identification is near perfect for all the ANNs initialized with 36 or more free parameters. However, for the network initialized with 30 free parameters, system identification obviously breaks down. We see that the structure recovery score drops to about 85% which makes sense, given that about 15% of the necessary synapses are missing. However, the system recovery score drops to an average of about 68% and takes on a huge variance (Fig. 4.55). This indicates that the network converges to a wildly different set of weight values for each random initialization, which is not the case in any of the other experiments when the ANN is provided with sufficiently many free parameters.

Thus, our advice to future experimenters who do not know the number of synapses in the system they are studying is as follows: Perform the cross validation described above, and look for this point of fragility. If you initialize your ANN at or to the right of the minimum of this curve, you are likely to perform successful system identification. The minimum of this curve likely represents a tight upper bound on the number of synapses in the system you are studying. Initializing with too few synapses in your ANN, as one would expect, leads to a total breakdown of this technique.

One might ask, however, if the eight log unit difference in the loss is truly catastrophic in terms of the goodness of fit of the ANN initialized with 15% too few free parameters? Would an experimenter know, even without performing the full cross validation, that the ANN was lacking crucial dimensionality? The answer to this

question is no. In examining the actual output of the 30- and 36 free parameter trained ANNs and comparing them to the data, one can see that the 36 free parameter network attains a virtually perfect fit to the data, while the 30-free parameter network misses the mark quite frequently, as it is not expressive enough to attain this perfect fit (Fig. 4.57).

However, computing the correlation coefficient between the ANN outputs and the validation data (which in this case is equivalent to computing explained variance, since the data are noiseless) shows that the 30-free parameter ANN is still able to attain between a 73% to 98% correlation with the data depending on the random initialization. So while for some initializations there is a performance difference of nearly 30%, there are some "lucky" initializations that only fall 2% short of the ANN with exactly the right number of free parameters (Fig. 4.58). Therefore, it is not enough to rely solely on the error from a single ANN. One must perform the cross-validation described above to be sure that one is using the appropriate number of free parameters for system identification.

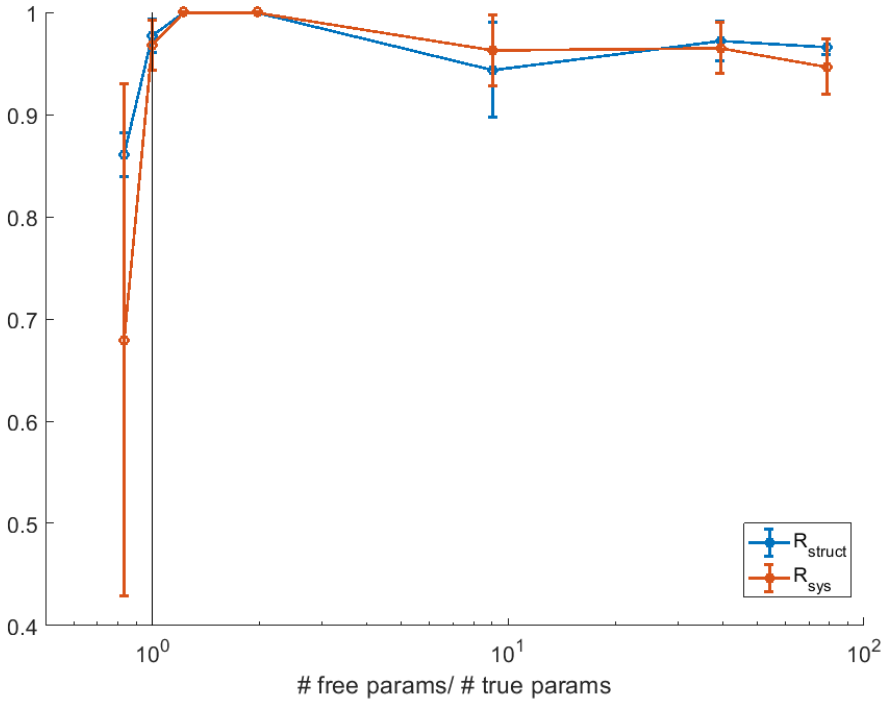

Figure 4.55: System and structure recovery scores plotted for ANNs trained on data from Circuit 1 which have been initialized with different numbers of free parameters. When the number of free parameters is less than the number of true parameters, there is a severe breakdown in system identification. Structure recovery score is around 85% regardless of random initialization, but the system recovery score varies widely depending on random initialization. System identification is much more successful and consistent when the number of free parameters is greater than or equal to the number of true parameters.

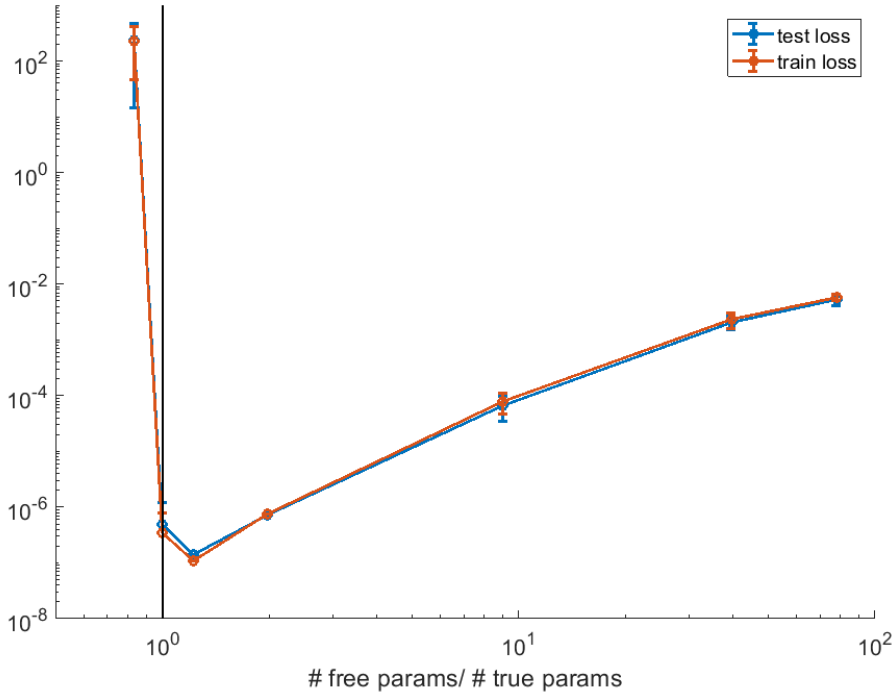

Figure 4.56: Final training and test loss plotted for ANNs trained on data from Circuit 1 which have been initialized with different numbers of free parameters. When the number of free parameters is less than the number of true parameters by just 15%, there is a very high loss. Loss declines sharply when the number of free parameters equals the number of true parameters and rises gradually as the number of free parameters is increased, indicating that a researcer without knowledge of the size of the system under study could use validation loss as a heuristic to choose the size of their initialized network.

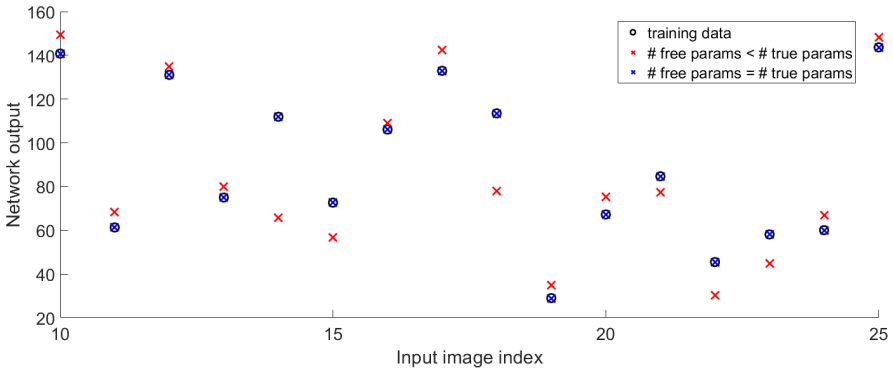

Figure 4.57: Test data and output of two ANNs trained with different numbers of free parameters.

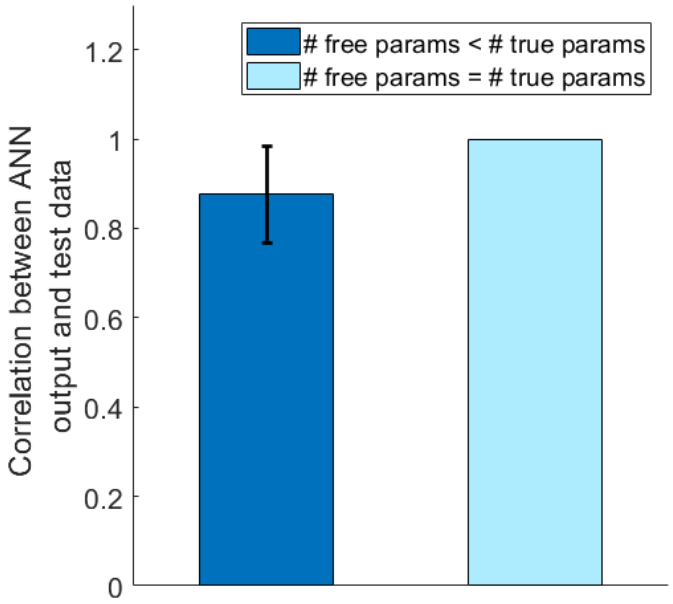

Figure 4.58: Correlation between ANN output and test data for two ANNs, one which is initialized with too few free parameters and one which has the exact same number of free parameters as in the true circuit. Correlation drops by about 10% for the underparameterized network, while it is virtually perfect regardless of random initialization when the network is initialized correctly.

Chapter 5

RESULTS III: CASE STUDY ON DATA COLLECTED FROM MOUSE ALPHA RETINAL GANGLION CELLS

5.1 The four subtypes of the alpha ganglion cell

The alpha retinal ganglion cell is characterized by its large dendritic tree and cell body, as well as its thick axon. Alpha cells can be divided into four subtypes: transient ON, transient OFF, sustained ON, and sustained OFF. These are named for their response dynamics and polarity. All of these types have a large receptive field center, and display weak surround suppression and no direction selectivity. They are present in many mammalian species, including both mice and humans [44]. Their precise microcircuitry, however, is not yet fully understood.

5.2 Data collection using multi-electrode array

Due to their large size, alpha cells lend themselves well to extracellular electrophysiological recording. Perhaps the simplest tool with which to do this is the multi-electrode array [55]. This array is constructed on a sheet of glass, into which a grid of 16×16 titanium nitride electrodes. A plastic bucket is glued to the glass surrounding the array, which can be filled with a nutrient bath to keep the retina alive. Two glass pipettes allow oxygenated solution to flow over the tissue.

To prepare the tissue for recording, a mouse is dark adapted, and then anesthetized with cervical dislocation, to avoid introducing any drugs into the bloodstream that may affect neuronal responses. The eye is enucleated and placed immediately in an oxygenated nutrient bath. Then, the retina is dissected out from the eye and placed ganglion-cell-layer-side down onto the array (Fig. 5.1), which is then connected to an amplifier and recording begins. The recording starts with an additional 15-30 minute light adaptation period wherein the retina is simply exposed to a blank gray screen. After this point, stimuli can be shown.

In most recordings, the following stimuli are presented to the retina and responses to each are recorded:

1. **Random flicker:** The screen is divided into a set of bars. The width of the bars is a parameter chosen by the experimenter. In our experimenters we used

widths between 13 and 95 microns. These bars are colored in black or white randomly. The pattern is randomly selected in each frame, and the frames flicker at 60 Hz.

2. **Fullfield flash:** The screen starts out as gray. The entire screen is turned black (or white) for one second, and then turns back to gray for one second. This repeats several times.

3. **Moving bars:** The screen starts out as gray. A bar (whose width and angular orientation is selected by the experimenter) is moved across the screen in one direction, and then the opposite direction.

4. **Switching gratings:** A black and white grating appears on the screen. The grating reverses sign every second. The spatial frequency of the grating is chosen by the experimenter (Fig. 5.2).

5. **Barcode:** A single 20 second, 20 Hz random flicker with 95 μm bars is shown. The same random flicker stimulus is repeated five times. This is presented between other stimuli throughout the duration of the recording. Each neuron's stereotyped response to repetitions of the same stimulus helps keep track of the health of the neuron throughout the duration of the recording.

5.3 Data preprocessing

After recording, spike sorting is performed using Kilosort [62]. Once the voltage traces have been separated into spike trains attributed to single cells, the spike trains are smoothed into continuous firing rate traces via convolution with a Gaussian filter. At this point, various analyses are implemented in order to classify neurons as alpha or non-alpha. This pipeline begins with a basic spike-triggered average analysis in order to obtain an estimation of the receptive field center size. Alpha cells are identified based on two properties: 1) receptive field center larger than a fixed threshold and 2) lack of direction selectivity. They are then sorted into subtypes based on polarity and decay time of their responses to the fullfield flash stimulus. Over the course of three recording days, we recorded from a total of 190 neurons. From among those, we were able to identify 2 sOFFα cells, 4 sONα cells, 14 tOFFα cells, and 1 tONα cell for which the quality of the data was sufficient for further analysis.

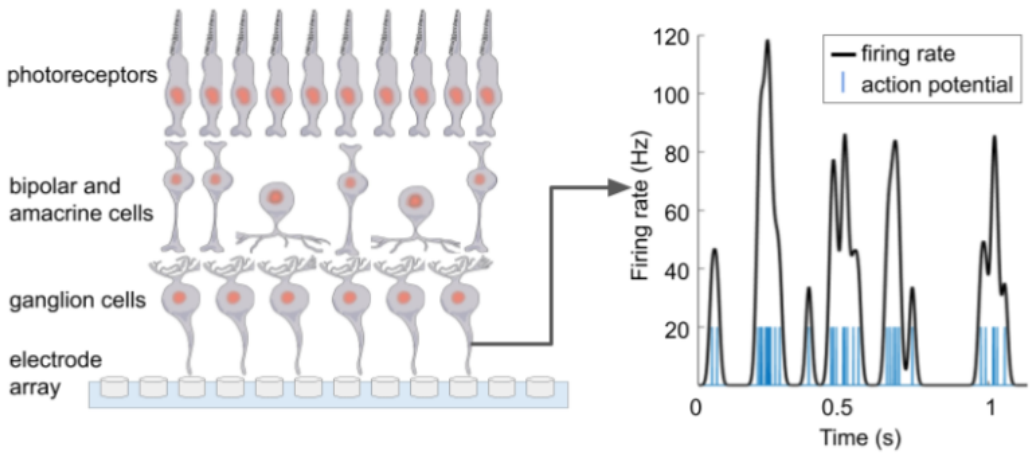

Figure 5.1: Left: Schematic of mouse retina on microelectrode array. The retina is positioned so that the ganglion cell layer is flush with the array of electrodes. Signals from hundreds of ganglion cells can be simultaneously recorded. Right: Example of signal from a single mouse retinal ganglion cell. Blue ticks represent single action potentials. Black curve shows the smoothed firing rate of the neuron over time.

5.4 A system identification problem for alpha cell circuitry

While the four alpha cell subtypes share similar morphology, close inspection reveals some functional differences outside of just their polarity or sustained/transient response dynamics. One such difference can be easily identified using the switching grating stimulus. Krieger et al displayed a switching grating stimulus in a circular mask centered over the receptive field. For each of the alpha cells, when the grating displayed was very fine, there was no response. Once the spatial frequency of the grating crossed a certain threshold, the neuron began to fire in response to every switch of the grating. Then once the the grating became coarse enough, the neuron would only respond to every other switch of the grating [44].

This behavior can be explained using the well-known Y-Cell model of retinal computation. In this model, the ganglion cell receptive field is divided into several "subunits," representing bipolar cells, each of which responds to only a small portion of the receptive field, and whose response is half-wave rectified (Fig. 5.3b.) Thus, switching gratings of an appropriate size elicits a response with every switch, even though the overall intensity in the ganglion cell receptive field remains constant. Using the switching grating stimulus, one can approximate the size of these subunits by finding the finest grating for which the neuron fires with every switch. Krieger et al showed that the subunit size of the sOFFα cell was significantly larger than the

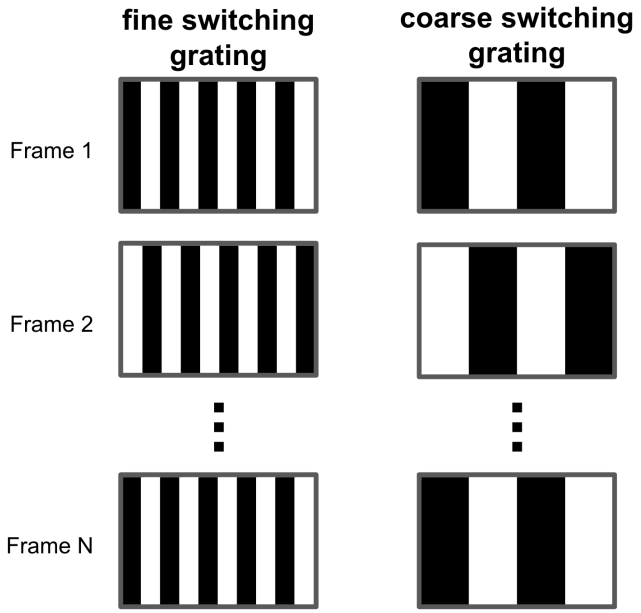

Figure 5.2: Contrast-reversing grating stimulus

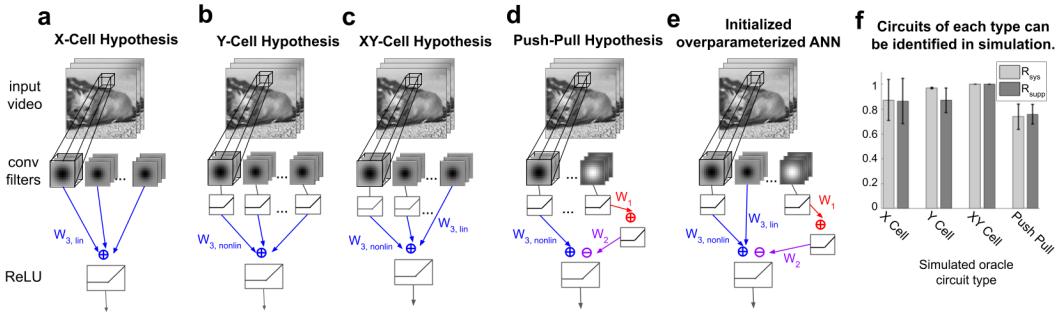

Figure 5.3: Hypothesized circuit architectures for the alpha ganglion cell. Each structure convolves the input video with a bank of 3D filters in the first layer. (a) X-Cell circuit: Utilizes only linear units in the first layer. (b) Y-Cell circuit: Utilizes only nonlinear units in the first layer. (c) Y-Cell circuit: Utilizes both linear and nonlinear units in the first layer. (d) Push-pull circuit: Utilizes nonlinear units in the first layer. Includes an additional inhibitory pathway. Structurally analogous to circuits 3A and 3B in Fig. 4.47b. (e) The ANN is initialized to include all components and connections used in all four hypotheses. (f) Structure and support recovery scores when the ANN in (e) is trained on data generated by simulated circuits with architecture matching (a)-(d).

other three types (about 200 μm compared to about 25 μm) [44]. We recorded a similar phenomenon in our recordings of alpha cells, though the difference was far less pronounced. In our case, the sOFFα subunit size was approximated to be about 150 μm (Fig. 5.4 top left) while the tOFFα subunit size was approximated as about 95 μm (Fig. 5.5). This discrepancy is likely due to the fact that the switching grating stimulus we presented was not fullfield, not presented solely over the receptive field center. Since we performed simultaneous recordings from around 50 or so cells, there was not sufficient recording time to isolate each receptive field individually and display the masked version of the stimulus used in [44].

200 μm is much larger than any known bipolar cell receptive field size. What is the circuitry that underlies this phenomenon? One simple idea is that the sOFFα cell is not a Y-Cell, it is an X-Cell. The X-Cell model is the same as the Y-Cell model, except that the subunits are not half-wave rectified (Fig. 5.3a.) This means that the ganglion cell will not respond to moderately sized switching gratings, because the intensity in the entire receptive field remains constant throughout the stimulus.

While this is the simplest explanation, it is not the only possibility. The sOFFα cell might also utilize a push-pull type model, which includes rectified subunits, but those subunits are inhibited by another set of subunits with the opposite polarity, via an amacrine cell (Fig. 5.3d.). Or it may simply be an XY-Cell, which is what we decided to name a circuit hypothesis that includes both rectified and non-rectified subunits (Fig. 5.3c.). We simulated each of these models and showed that they could replicate the phenomenology of the sOFFα cell response to switching gratings (Fig. 5.4).

And, while the Y-Cell hypothesis might seem the most plausible for the other three subtypes of alpha cell, they could also possibly be described with an XY-Cell or Push-Pull model. As a test of our system identification technique, we decided to design an ANN with the necessary components for all four of these hypotheses (Fig. 5.3e.), and to train it in turn on data from each alpha cell subtype to see if we could identify any circuit differences between them.

Before applying the technique to real data, we simulated each of the four circuit hypotheses and generated training data sets from each. We then trained the initialized ANN depicted in Fig. 5.3e on each dataset and confirmed that the correct circuitry could be recovered in each case. Fig. 5.3f shows the system and structure recovery scores for 10 random initializations of each training. Each circuit could be correctly identified and scores ranged between 0.8-1.0, lending confidence that the application

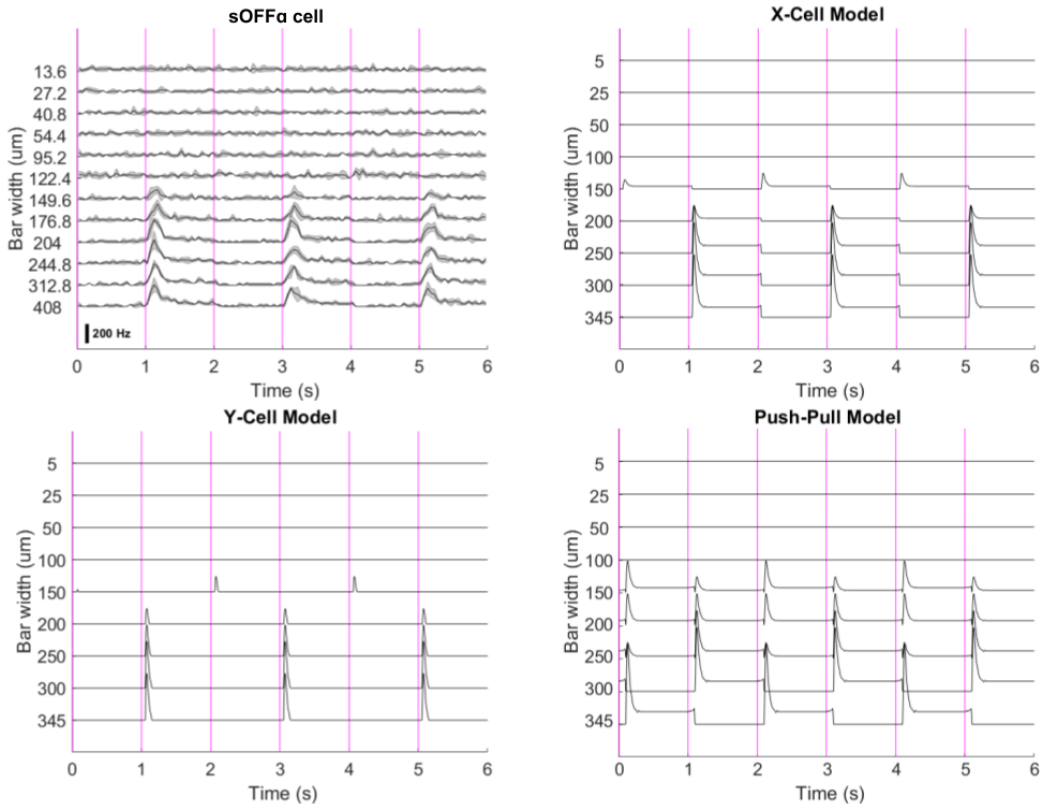

Figure 5.4: Response of sOFFα cell and three models to switching gratings of increasing size. We simulated each of the three models with parameter sets chosen such that each responded only to large switching gratings. Note that the exact timing and phase of these responses depends on precise choice of parameters, as well as the spatial phase of the stimulus, and therefore we cannot rule out any hypotheses based on this. Pink vertical lines represent times of grating switch.

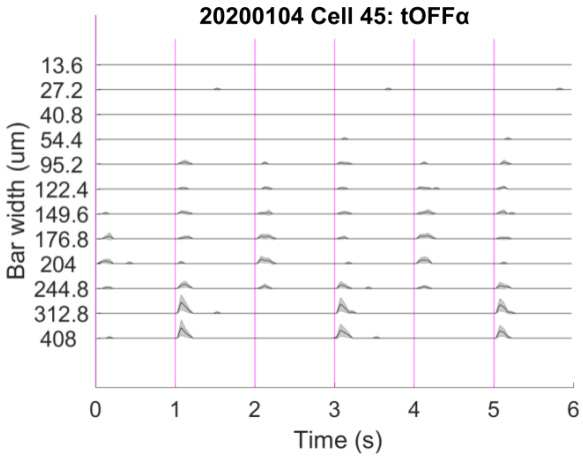

Figure 5.5: Response of a tOFFα cell and three models to switching gratings of increasing size.

	X-Cell	Y-Cell	XY-Cell	Push-Pull
X-Cell	1.00	0.00	0.80	0.00
Y-Cell	0.00	1.00	0.40	0.07
XY-Cell	0.80	0.40	1.00	0.04
Push-Pull	0.00	0.07	0.04	1.00

Figure 5.6: For each simulated circuit used in the simulations in figure 5.3f, the R_{sys} was computed with every other circuit hypothesis. This serves as a "floor" to better contextualize the R_{sys} of the trained ANN's in each case.

to real data would provide meaningful results. For comparison, Fig. 5.6 shows R_{sys} scores computed between each of the four models.

5.5 Fitting an ANN to alpha cell data

The ANN shown in Fig. 5.3f is trained on data from each individual alpha cell separately. The responses to approximately 1-1.5 hours of random flicker stimuli with varying spatial frequencies were used as the training dataset in each case. Over the course of training, we tracked the fraction of variance explained by the ANN model using a held-out validation set. The variance in the alpha cell data was computed using the repetitions of the barcode stimulus. A stopping condition was imposed to stop training when the fraction of explained variance either exceeded 1.0 (overfitting) or 200 training epochs had elapsed, whichever came first. At this point, we analyzed the structure of the trained ANN.

Fig. 5.7 shows one example of the quality of the fit of the ANN to the training data for a single ganglion cell, and also summarizes the fraction of explained variance of the final trained model, separated out by alpha cell subtype. While in many cases, the explained variance is quite high, we know from our experience with the simulations in Chapter 4 that this does not necessarily indicate that the circuitry learned was correct.

5.6 Recovery of known biological information about the alpha cell

Our confidence in the learned circuitry of the ANN is bolstered by the simulations summarized in Fig. 5.3, but we were also able to identify some known circuit properties of the alpha cell subtypes that were recovered by the ANN. First, the ANN was able to recover the spatial receptive field of the neuron in each case. One example is shown in Fig. 5.8.

Additionally, the ANN was able to independently re-classify each of the neurons

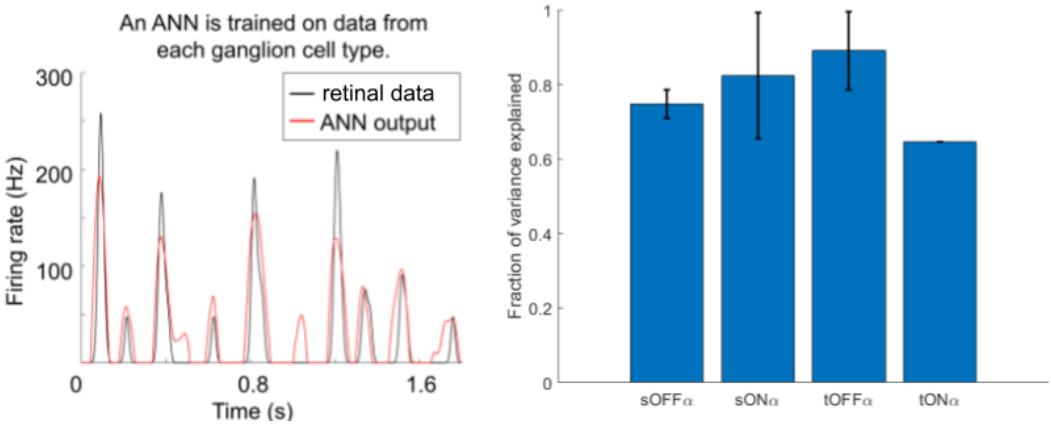

Figure 5.7: Left: Example of quality of fit to ganglion cell firing rate. Right: Fraction of variance in the retinal data explained by the trained ANN, separated by cell type.

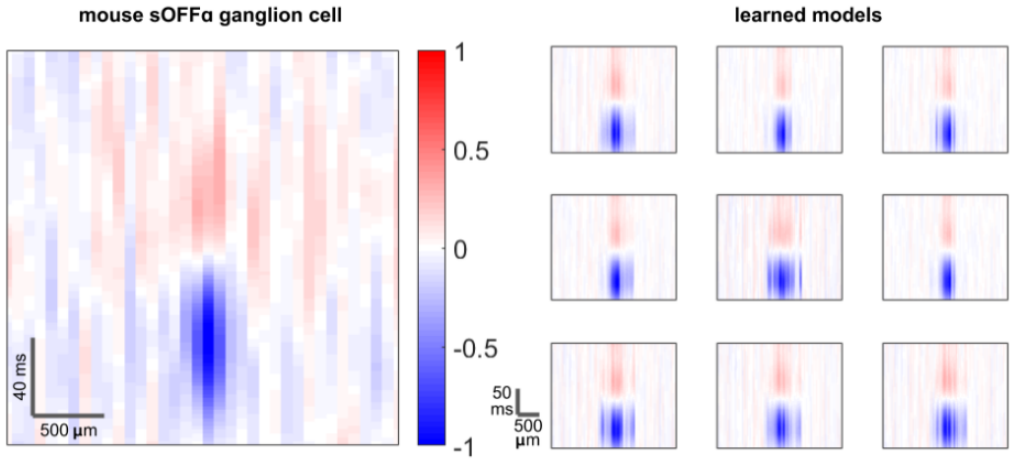

Figure 5.8: The trained ANN can replicate the spatiotemporal receptive field of an sOFFα ganglion cell. Receptive fields were computed using reverse correlation on a held out test set.

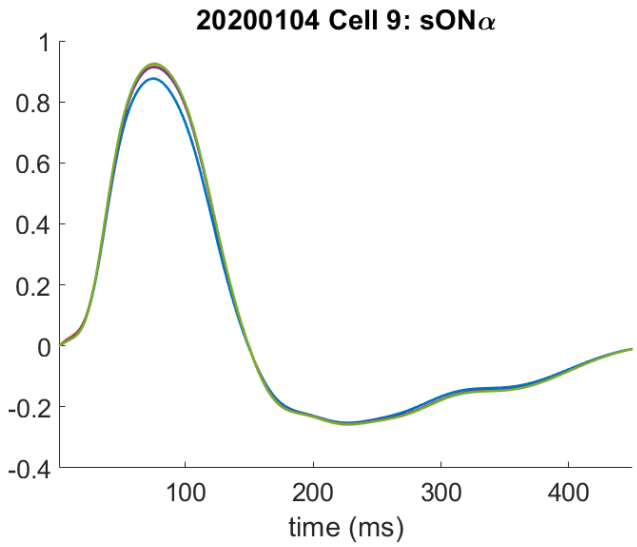

Figure 5.9: Temporal filters in the bipolar cell layer of an ANN trained on data from an sONα cell with varying random initializations.

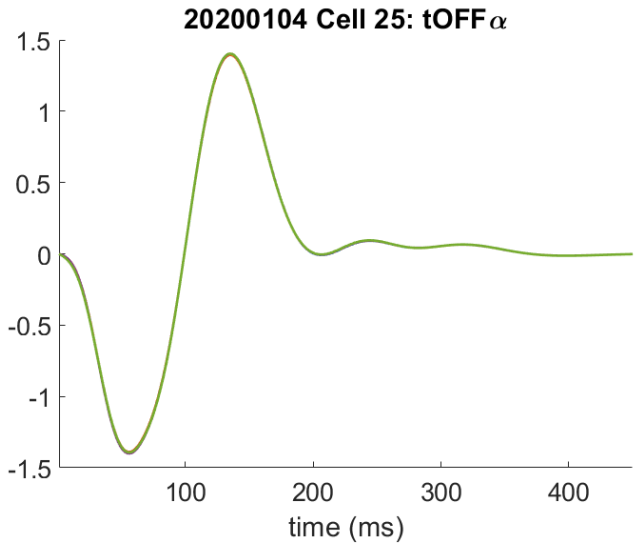

Figure 5.10: Temporal filters in the bipolar cell layer of an ANN trained on data from an tOFFα cell with varying random initializations.

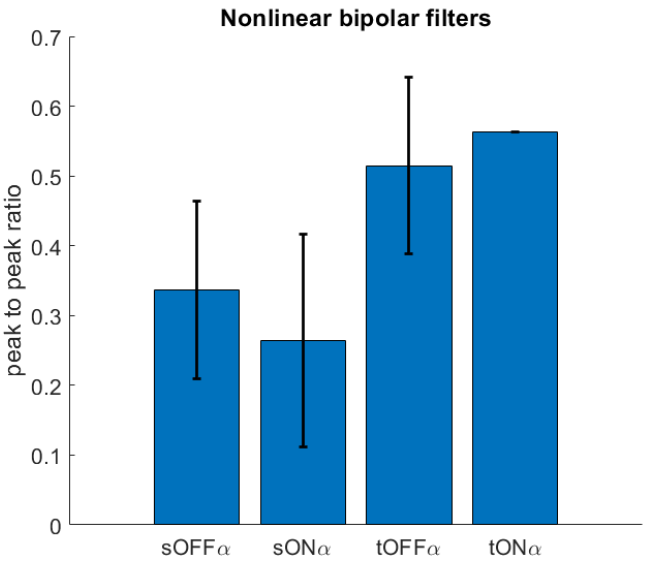

Figure 5.11: Ratio of magnitude of second peak to first peak in each set of learned temporal filters for nonlinear bipolar cells in the trained ANNs, separated by cell type. ANNs trained on data from sustained type cells have a smaller peak-to-peak ratio.

by subtype. The spatiotemporal filters of the bipolar cell subunits in each trained ANN reflected either the sustained or transient and OFF or ON property of each neuron correctly. The difference between sustained and transient temporal filters can be quantified by measuring the ratio of amplitudes of the positive and negative peaks of the filter. For an OFF cell, a high amplitude negative peak, followed by a low amplitude positive peak, as in the example in Fig. 5.9, leads to a much more sustained response than when the two peaks are comparably sized (Fig. 5.10.) This phenomenon is demonstrated in Fig. 2.1. Quantifying the peak to peak ratio illustrates that when trained on data from sustained alpha cell types, the ANN subunits developed much more sustained-type filters in their nonlinear subunits (Fig. 5.11). However, the same did not hold for the linear subunits (Fig. 5.12). Thus, the ANN has successfully recovered known information about each of the four alpha cell circuit subtypes.

5.7 The ANN correctly identifies the circuitry of the transient OFF alpha cell and suggests an additional pathway

We next examined the circuit structures to identify which of the four hypotheses in Fig. 5.3 best described each of the four cell subtypes. Strikingly, the four subtypes

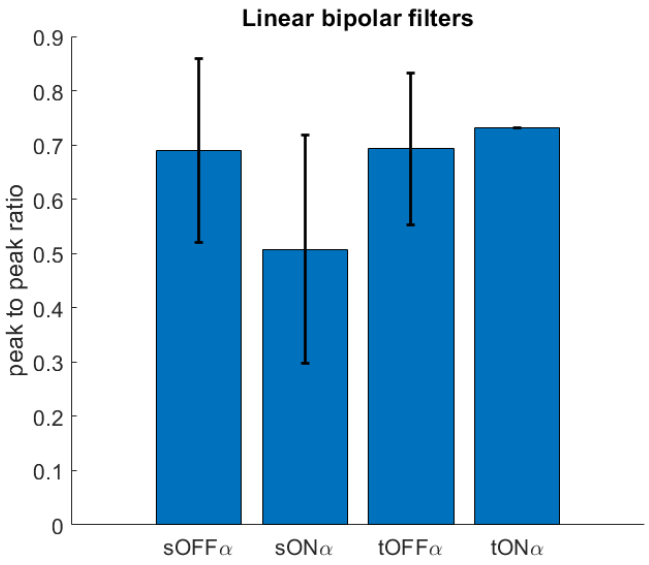

Figure 5.12: Ratio of magnitude of second peak to first peak in each set of learned temporal filters for linear bipolar cells in the trained ANNs, separated by cell type. ANNs trained on data from sustained type cells have a smaller peak-to-peak ratio.

were each assigned a circuit which we will call XY + Push-Pull by the ANN. All four synapse types remained in the final trained ANN for all four alpha cell subtypes (Fig. 5.13).

In the case of the transient OFF subtype, this finding confirms something which has already been elucidated via painstaking biological circuit dissection experiments [58]. The tOFFα cell is also known as the PV-5 ganglion cell [68], and has been shown to use a Push-Pull circuit mechanism. Based on recordings of excitatory and inhibitory currents in the PV-5 ganglion cell, it is known that the PV-5 cell is excited by nonlinear OFF bipolar cells, and inhibited by ON bipolar cells via the AII amacrine cell [58]. With our technique, however, we find that an additional component is in use. In the trained ANN, there are linear bipolar cells which also excite the ganglion cell.

This finding is exciting, because it proves that our system identification technique can recover the correct circuitry of a real retinal circuit. But it also provides a path forward for future experimentation. The inclusion of this linear excitatory pathway suggests that previous studies may have missed a crucial component of the circuit. Now that our system identification technique has made this hypothesis, future researchers can direct resources towards confirming or ruling out the existence

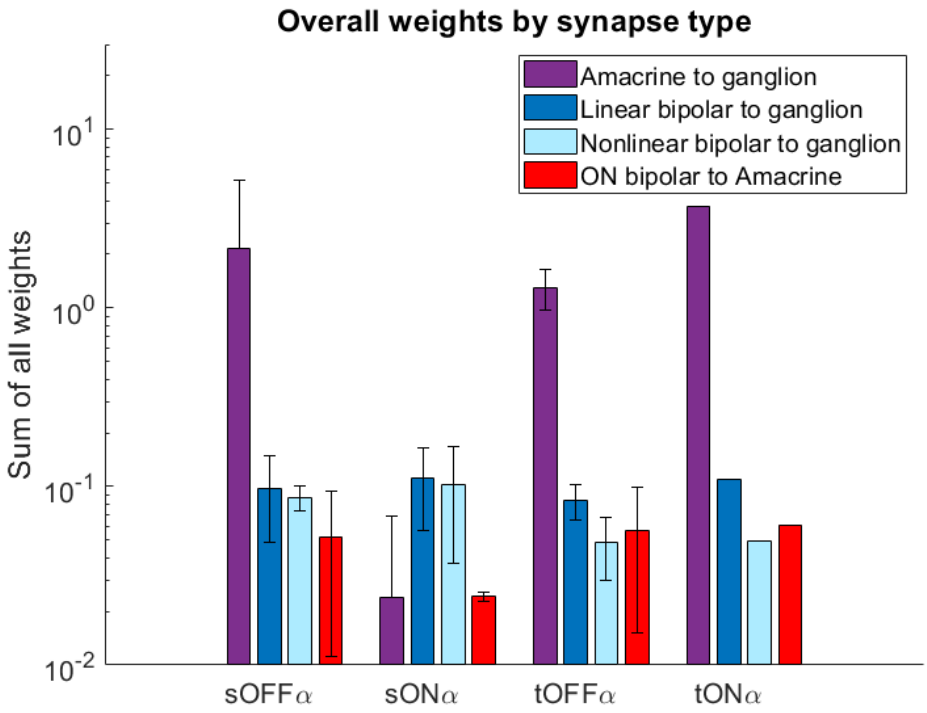

Figure 5.13: Sum of all weights in the trained ANNs, separated by synapse type, in order to show relative contributions of the various pathways learned for each alpha cell subtype.

of these putative synapses.

5.8 System identification as a tool for hypothesis selection

We next looked at the circuitry proposed by the ANN for the other three alpha cell subtypes. While there were 14 transient OFF cells in the dataset, there were only 2 sustained OFF, 4 sustained ON, and 1 transient ON cell in the dataset. Thus, the findings we present for these neurons are preliminary and warrant further exploration. As mentioned above, our working hypothesis was that the sOFFα type used an X-Cell circuit, while the other three types used a Y-Cell circuit. We found that this was not, in fact, the case, according to the ANN. All four cell types were classified as XY + Push-Pull Cells. The sustained ON cell appeared to have a relatively weak inhibitory pathway compared to the other three types (Fig. 5.13).

Our next question was: does the sOFFα employ more linear subunits than the other three types? Perhaps that could account for its more X-Cell type behavior in [44]. In other words, perhaps the relative strength of $\mathbf{W}_{3,\,\mathrm{lin}}$ and $\mathbf{W}_{3,\,\mathrm{nonlin}}$ differed for the

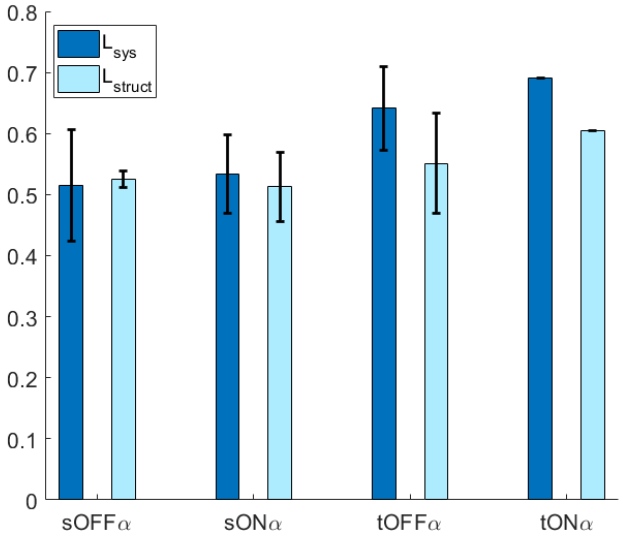

Figure 5.14: Comparing the strength of linear and nonlinear pathways in the learned alpha cell models. L_{sys} and L_{supp} (Eq. 5.1 & 5.2) for artificial neural networks trained on datasets recorded from each of the four alpha ganglion cell types in mouse retina.

sOFFα cell.

To quantify this, we defined the following quantities:

$$L_{\text{sys}} = \frac{\sum_{i,j} \mathbf{W}^{ij}_{3,\text{ lin}}}{\sum_{i,j} \mathbf{W}^{ij}_{3,\text{ lin}} + \sum_{i,j} \mathbf{W}^{ij}_{3,\text{ nonlin}}}, \tag{5.1}$$

and

$$L_{\text{supp}} = \frac{\sum_{i,j} \overline{\mathbf{W}}^{ij}_{3,\text{ lin}}}{\sum_{i,j} \overline{\mathbf{W}}^{ij}_{3,\text{ lin}} + \sum_{i,j} \overline{\mathbf{W}}^{ij}_{3,\text{ nonlin}}}, \tag{5.2}$$

where $\overline{\mathbf{W}}_{3,\text{ lin}}$ and $\overline{\mathbf{W}}_{3,\text{ nonlin}}$ represent the binarized versions of $\mathbf{W}_{3,\text{ lin}}$ and $\mathbf{W}_{3,\text{ nonlin}}$.

Our analysis showed that for all four cell types, $L_{\text{sys}} > 0.5$. L_{supp}, meanwhile, was also equal to or slightly larger than 0.5 in most cases. Thus, in all four subtypes, the contribution of the linear subunits is larger than that of the nonlinear subunits. This matches our expectation based on the analysis of the neurons' responses to the switching grating stimulus. Many of them did not respond to switching gratings smaller than 95 μm and the average bipolar cell receptive field is approximately 60 μm in diameter [22]. This is a decidedly different finding than that presented in [44], though as mentioned, the switching grating stimulus in [44] was restricted to

the receptive field center, while ours was fullfield, and this alone could account for the difference.

However, this may not be totally attributable to activation of surround suppression. The amacrine cells available to the network were fully pruned out, when they could have been used to model a suppressive surround. Another thing to note is that the explaned variance of the trained ANNs varies between about 60-90%. Thus, there is another 10-30% of the variance that was not accounted for by our ANNs. This points to a second possibility: perhaps the difference in circuitry between the sOFFα cell and the other three subtypes is due to a circuit mechanism that was not included in the initialized ANN. Perhaps it is not a matter of simple X-Cell vs. Y-Cell circuitry, but something more complex. There could be recurrence in the alpha cell circuit for which we have not accounted, or lateral inhibition between bipolar cells may play an active role. We should also note that our study was underpowered in all subtypes except for the tOFFα cell which was overrepresented in the dataset and accounted for 2/3 of the cells studied. Future work is needed to advance this technique to the point that it can fully account for the phenomenon observed in [44].

5.9 Summary

We have shown that when applied to data from a real retinal ganglion cell, our system identification technique can recover known information about the circuits under study. This provides us with the highest form of confirmation of the utility of this technique that is currently avaiable to us. The technique also presented several interesting hypotheses regarding the various alpha cell types that could be directions for research. This work provides a highly encouraging jumping off point for myriad applications of ANNs for nonlinear system identification in neuronal circuits.

Chapter 6

RESULTS IV: AN EXPLORATION OF ALGORITHMS FOR INPUT SELECTION IN SYSTEM IDENTIFICATION

After our system identification method has been applied, the researcher is meant to be left with a set of potential circuit hypotheses, and must then generate targeted biological experiments to confirm or reject each one. One way of doing this is by using targeted visual stimuli to maximally distinguish between alternate hypotheses. One could even imagine extending this idea so that the initial ANN training is done online during the recording from the neuron. In that case, one could select the next stimulus to display based on the current partially trained ANN, in order to maximally resolve some form of uncertainty. This would be a form of optimal experimental design (OED) or "active learning."

This is particularly intriguing because this application provides us with a lot of flexibility in terms of dataset design. In most applications of machine learning, the dataset is viewed as something static, which is handed to the researcher incomplete, and to which very few if any changes can be made. In contrast, in retinal neuroscience, one can show virtually any visual stimulus to the retina by simply altering the pixels on a screen. And, as described in previous sections, we have already spent some time thinking about the design of the dataset.

While input generation is easy, output collection (electrophysiological recording) is expensive. This is due to the limited recording time when the retina is removed from the animal. While the tissue can be kept alive in an oxygenated nutrient bath, this is only feasible for up to six hours at best. Thus, it is crucial to choose stimuli wisely, as each stimulus presentation takes up part of this limited time window. The idea of targeted stimulus design to distinguish between hypotheses or resolve uncertainty is therefore well suited to system identification of retinal circuits.

Optimal experimental design of the type described above may have any of several possible goals. Some possibilities include:

1. Minimizing uncertainty about the network structure;

2. Minimizing uncertainty about the network's output;

3. Minimizing output error.

We spent some time investigating these possibilities in simulation.

6.1 An adversarial stimulus generation algorithm

The first and possibly most naive stimulus generation algorithm is as follows:

Algorithm 2: An adversarial stimulus generation algorithm

Result: Trained ANN

Initialize an overconnected ANN as described in previous sections;

Prepare a predefined set of visual stimuli, A;

Prepare retinal recording;

while *Retina alive AND stopping condition not met* **do**

Display a batch of stimuli, $B_i \subset A$ and record retinal response; Run ANN forward and select $x_i \in B_i$ with maximum loss;

Generate a set of similar stimuli by taking small, random steps in stimulus space. call this set X_i;

Create a new batch of stimuli, B_{i+1}, containing X_i, along with some additional stimuli from A and repeat

end

This simple algorithm can take on many variations depending on how the stimulus space is parameterized. For example, steps can be taken in the high dimensional space where every pixel is its own axis (demonstrated in Fig. 6.1), or steps can be taken along axes corresponding to salient stimulus features, such as motion speed or contrast.

We applied the simplest version of this algorithm in the case where the stimuli were moving random patterns, and the circuit under study was the simulated ooDS network described in previous sections. These stimuli were 16x1 random patterns which shifted across the screen over the course of 12 individual frames, where each frame was presented for 250 ms. This is known as an apparent motion stimulus. We therefore parameterized stimulus space as 12×16 dimensional and took random steps in this space rather than raw pixel space. We tracked the projection score over the course of training to understand whether this algorithm could improve system identification. However, we found that it actually led to poorer system identification

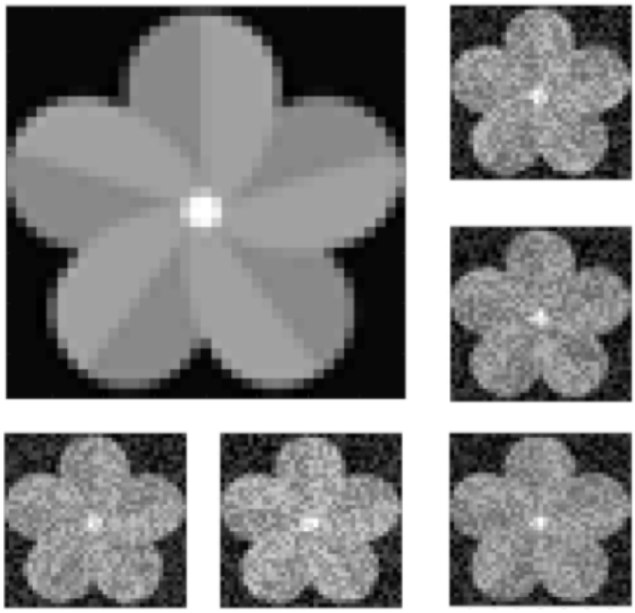

Figure 6.1: Cartoon demonstrating stimulus generation method. If a flower is given as the seed stimulus, taking random steps in stimulus space will give rise to several noisy-looking pictures of the same flower.

than a passive algorithm which simply trained on moving random pattern stimuli (Fig. 6.2.) While in both cases, the true circuit was easily distinguished from the alternate model, active stimulus selection did not provide the boost in performance we were expecting. This is likely due to the fact that the steps taken in stimulus space were too small, and therefore the batches of stimuli were too similar and did not contain enough information to perform accurate system identification. This highlights the tradeoff between emphasizing certain stimuli where performance is poor and lack of variety in the dataset.

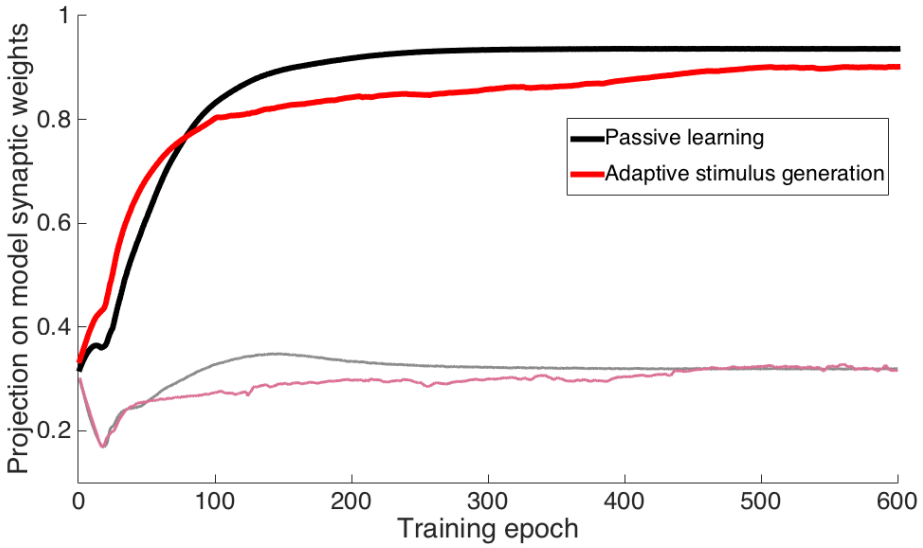

Figure 6.2: Projection score over the course of training for two trained ANNs with different training algorithms. Thick lines represent projection onto ground truth circuit structure. Thin lines represent projection onto alternative hypothesis. Black lines represent ANN trained with standard "passive" optimization. Red lines represent the same ANN trained using Algorithm 2. Passive learning results in better system identification.

6.2 An algorithm to select between two classes of stimuli

Algorithm 3: A second adversarial stimulus generation algorithm

Result: Trained ANN

Initialize an overconnected ANN as described in previous sections;

Prepare a predefined set of visual stimuli, A;

Prepare retinal recording;

while *Retina alive AND stopping condition not met* **do**

Display a batch of stimuli, $B_i \subset A$ and record retinal response; Run ANN forward and select a set of $x_{i,j} \in B_i$ with maximum loss. Call this set X_i;

Create a new batch of stimuli, B_{i+1}, containing X_i, along with some additional stimuli from A and repeat

end

This algorithm, rather than generating new similar stimuli to the highest error stimulus at each step, simply repeats training on the high error stimuli, and slowly

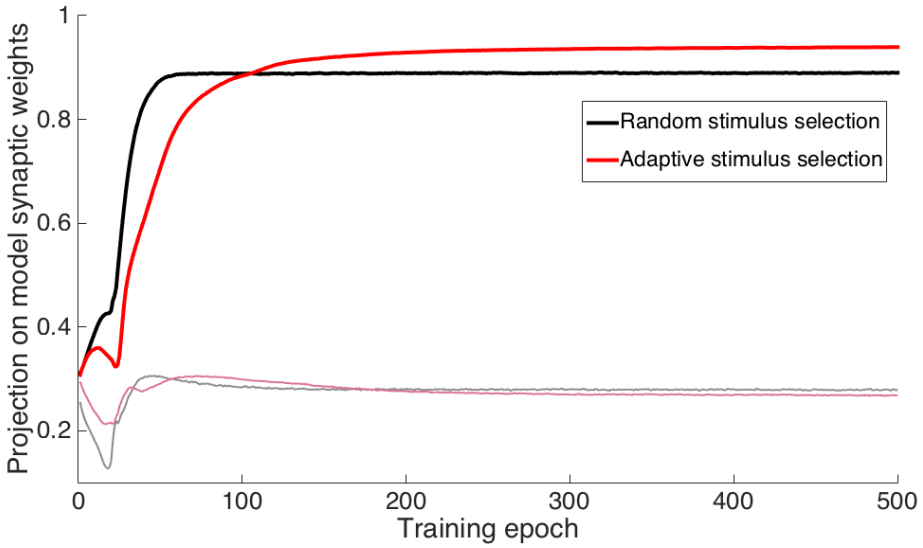

Figure 6.3: Projection score over the course of training for two trained ANNs with different training algorithms. Thick lines represent projection onto ground truth circuit structure. Thin lines represent projection onto alternative hypothesis. Black lines represent ANN trained with standard "passive" optimization. Red lines represent the same ANN trained using algorithm 3. This time, adaptive stimulus selection leads to better system identification, though convergence is slower.

adds more random stimuli to the training set. We applied this algorithm to the stimulus distribution A that contained both random flicker and moving random pattern stimuli in equal measure. This meant that during the course of training, the algorithm actually favored one of these two stimulus types depending on where the largest errors were being made. So, inadvertently, this algorithm provided us with information about which stimulus type is most useful during specific phases of training.

Optimal choice of training stimulus varies over the course of training

This algorithm did provide a boost in system identification, and improved the final projection score by about 0.05, though it did also slow convergence somewhat (Fig. 6.3.) But the most interesting result of this experiment was the choices made by the algorithm during training. Each new batch of stimuli, B_i, is composed of some moving random pattern stimuli and some random flicker stimuli, and the ratio of these depends on which stimuli produce the highest error at this phase of training. Fig. 6.4 shows the fraction of moving random pattern stimuli in each B_i over the course of training. Early in training, during what we call the "structure learning"

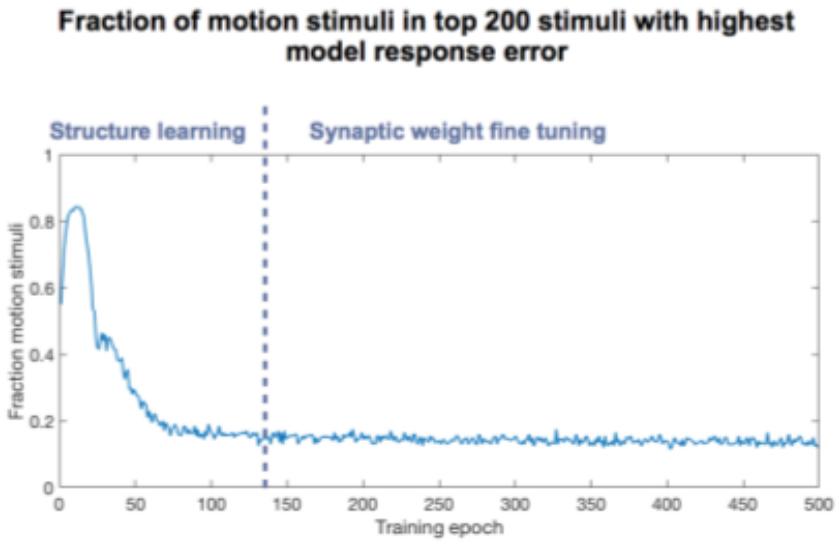

Figure 6.4: Fraction of motion stimuli selected by Algorithm 3 over the couse of training. Motion stimuli are preferred early in training, while random flicker stimuli are preferred later in training.

phase, much of the synaptic pruning is taking place. We found that early in the structure learning phase, the algorithm strongly prefers moving random pattern stimuli to random flicker stimuli (80% motion stimuli.) During the "fine tuning" phase, after all the unnecessary synapses have been pruned out, the network is simply adjusting the synaptic weights of the remaining connections. During this phase, random flicker stimuli make up about 80% of each batch, B_i (Fig. 6.4.)

6.3 An algorithm to maximize output of the circuit

Another way to guide stimulus design is to try to select for stimuli that maximize the firing rate of the circuit. We implemented a simple version of this in the following way: We began by constructing two artificial networks which shared weights. The first was the trained ANN. The second had the same structure and shared the same weights, but was used to generate stimuli (Fig. 6.5.) In this scheme, data are collected from the retina and used to train the trained network. After partial training, the weights are passed to the stimulus-generating network, which uses the current estimate of the weights to produce a new set of stimuli that should maximize the firing of the retinal ganglion cell. These stimuli are presented to the retina and the recorded responses are used to train the trained network. This process repeats until convergence.

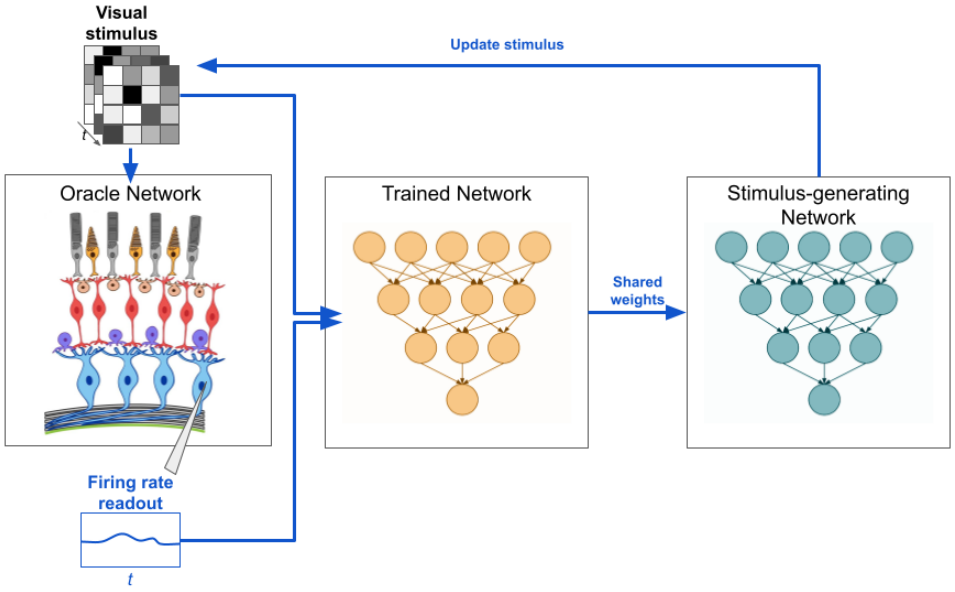

Figure 6.5: Schematic of the active stimulus-generation algorithm.

Algorithm 4: A third adversarial stimulus generation algorithm

Result: Trained ANN

Initialize an overconnected ANN as described in previous sections;

Prepare a predefined set of visual stimuli, A;

Prepare retinal recording;

while *Retina alive AND stopping condition not met* **do**

Display a batch of stimuli, $B_i \subset A$ and record retinal response; Run ANN forward and select a set of $x_{i,j} \in B_i$ with maximum loss. Call this set X_i;

Create a new batch of stimuli, B_{i+1}, containing X_i, along with some additional stimuli from A and repeat

end

How should we generate stimuli to increase the firing rate of the output neuron? Our method was inspired by [35]. The stimulus generating network is given an approximation of the structure and weights of the network from the trained ANN. To generate a new stimulus to show to the retina, we fix those weights in place. We then backpropagate from the output of the stimulus generation network all the way to the stimulus (Fig. 6.6.) We are therefore able to compute the partial derivative of the network output with respect to the stimulus image, and use gradient ascent to

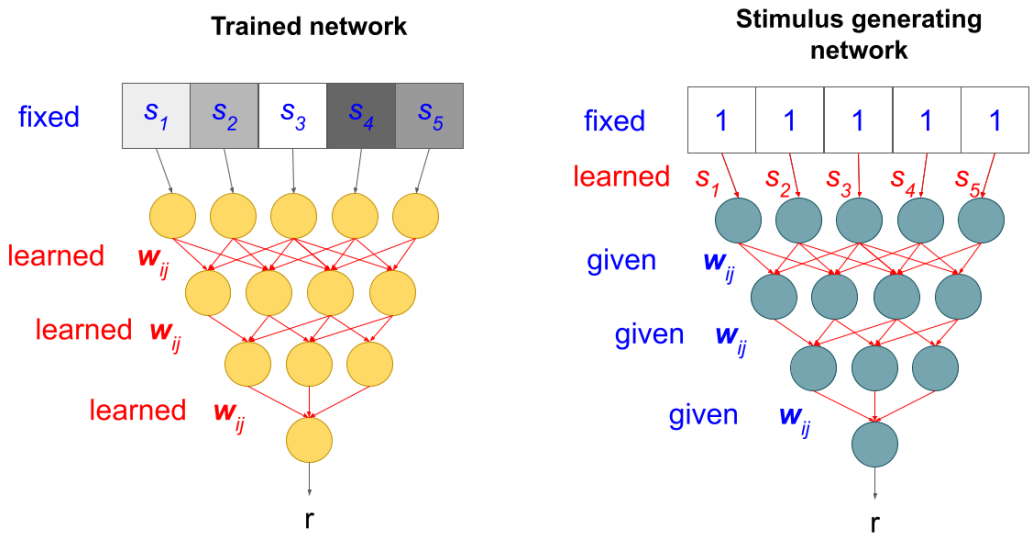

Figure 6.6: Stimulus generation method schematic.

find more salient stimuli.

To test this algorithm, we simulated a very simple circuit, made up of two LN bipolar cells. Since the algorithm was quite complex, we wanted to start very small in order to understand the behavior of the algorithm. They each had adjacent and isolated 1-pixel spatial receptive fields. The bipolar cells took a convolution of the stimulus video with a temporal filter and passed the output through a ReLU nonlinearity. The ganglion cell unit simply took a weighted sum of the output of these two bipolar cells and passed it through a second temporal convolution and ReLU nonlinearity, to produce a firing rate output (Fig. 6.7). Each temporal filter in the bipolar cell layer was parameterized as a two-bump function, as described in previous chapters. The two bipolar cells shared these parameters, but with signs reversed. Thus we were simulating one ON bipolar cell and one OFF bipolar cell, each connected to the same ganglion cell.

We then initialized an ANN with the exact same structure, but with random weights and temporal filter parameters. This meant there was a total of four free parameters in the ANN. No pruning was necessary, but these weights were randomly initialized needed to be adjusted to match the simulated retinal circuit.

We first trained the ANN passively, by simply presenting a series of random flicker stimuli to the simulated circuit, recording the output, and presenting these data to the

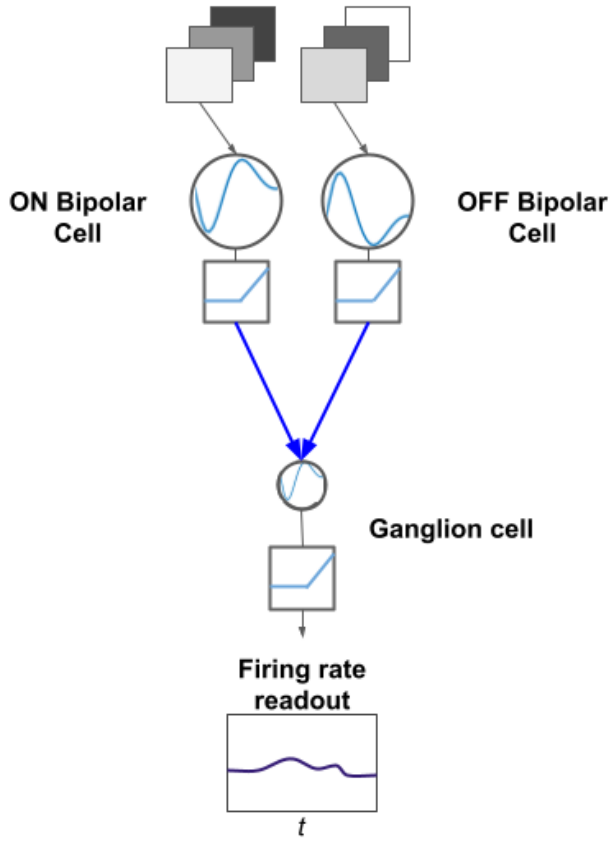

Figure 6.7: Three neuron simulated circuit used in simulations of Algorithm 4.

ANN for gradient descent. As expected, after some training, the ANN approximated the weights almost perfectly (Fig. 6.8). This is not surprising, since the network is quite simple compared to previous circuits we have simulated and whose structure we have successfully recovered with this technique.

Next, we applied Algorithm 6.4, and attempted to recover the weights once more. This time, the weights were not learned correctly, though they came close and certainly improved on the initialization (Fig. 6.9.) To understand this better, we examined the stimuli that were generated by the stimulus-generating network over the course of training.

The seed stimuli used in this case were random flicker stimuli where the pixels took random intensities on a continuous scale from 0 to 1. Figure 6.10 shows the seed stimuli on the left and stimuli generated by the stimulus-generating network after 500 epochs of backpropagation on the right. It is apparent that the gradient ascent algorithm converges to high contrast stimuli which are simultaneously white in the

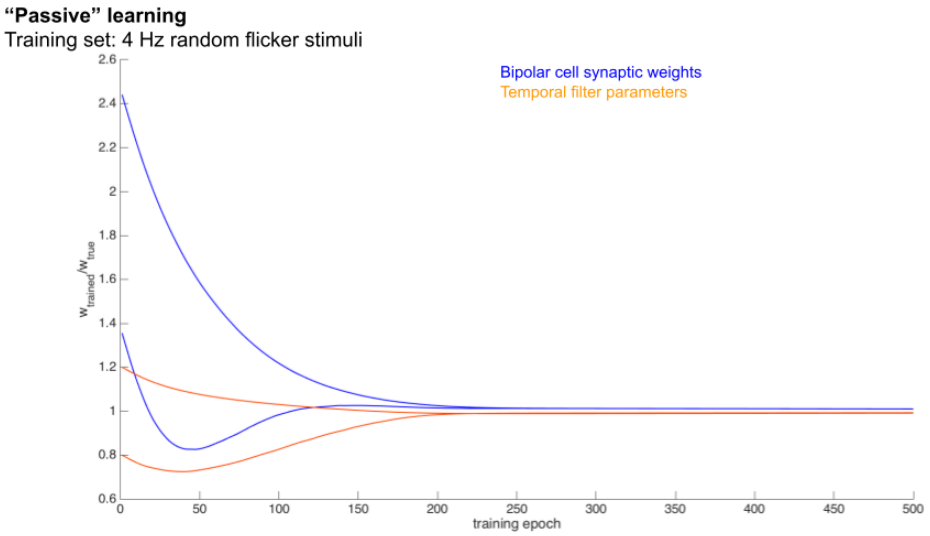

Figure 6.8: $\frac{w_{\text{trained}}}{w_{\text{true}}}$ plotted for the weights and filter parameters in the circuit in Fig. 6.7 over the course of training when passive learning is used.

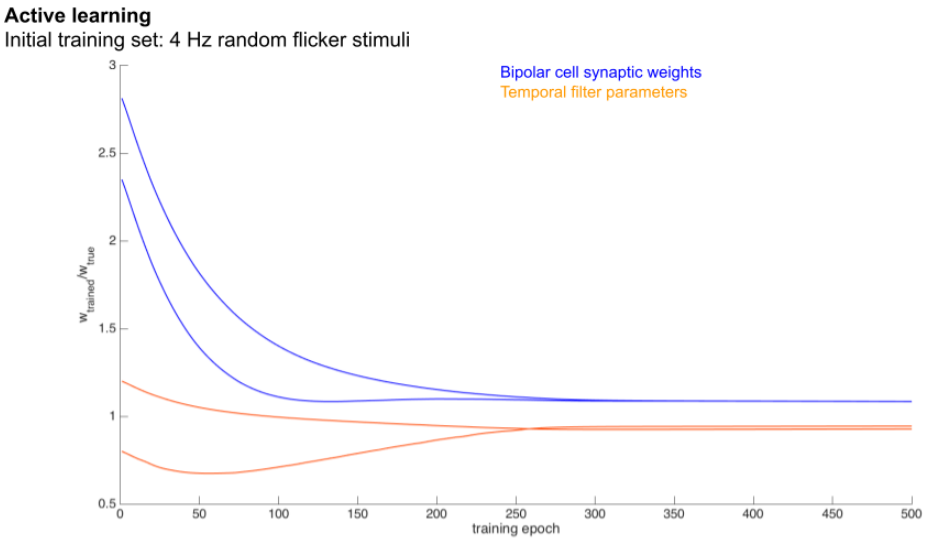

Figure 6.9: $\frac{w_{\text{trained}}}{w_{\text{true}}}$ plotted for the weights and filter parameters in the circuit in Fig. 6.7 over the course of training when Algorithm 4 is used.

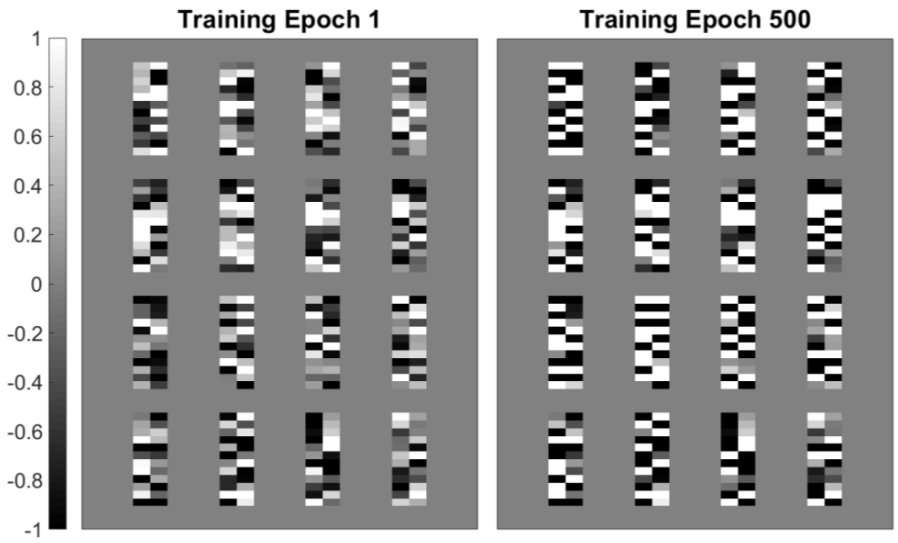

Figure 6.10: Batch of 16 stimuli generated by Algorithm 4 in epoch 1 and epoch 500. Each stimulus is 2 pixels by 12 temporal frames.

left pixel (exciting the ON bipolar cell) and black in the right pixel (exciting the OFF bipolar cell). This is, intuitively, a very straightforward stimulus to select if one wants to maximize the firing of the ganglion cell in Fig. 6.7.

However, because there was no mechanism to encourage the algorithm to maintain diversity in the stimulus set, as the stimulus generating network converges to this high contrast "checkerboard" stimulus, the stimulus batches used to train the trained network become very monotonous. This may account for the failure of the algorithm to perfectly recover the synaptic weights of the circuit.

6.4 An algorithm for optimal stimulus design given competing circuit hypotheses

Once system identification has produced a set of circuit hypotheses to describe a biological system, how can one further narrow down this set? One option is to design a stimulus that can optimally distinguish between these hypotheses, then present that stimulus to the biological system, and use the response to further rule out hypotheses. As a direction for future research, we propose an algorithm inspired by work done by Hwang et al in 1991 [35]. This proposed algorithm would work by creating an ANN whose output node is simply the squared difference between the output of two circuit hypotheses. The weights of these hypotheses are fixed in

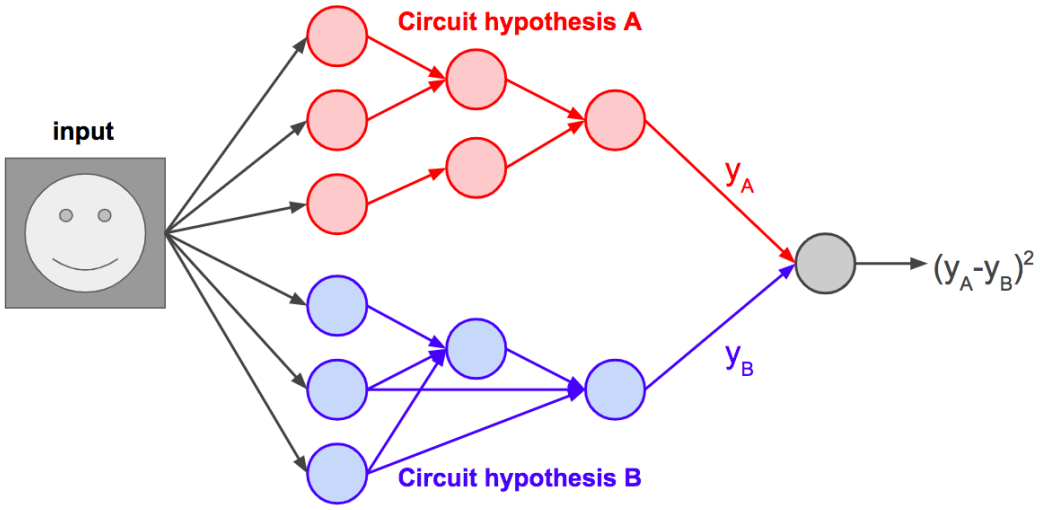

Figure 6.11: Schematic of proposed stimulus design algorithm.

place, and gradient ascent is run to find a stimulus which maximizes the output of the network (Algorithm 6.4.) This stimulus, therefore, maximizes the difference in output of the two circuit hypotheses. It can then be applied to the biological system in question to obtain ground truth, and provide evidence for or against each hypothesis. Of course, many such stimuli can be generated from a set of randomly chosen or carefully designed seed stimuli.

Algorithm 5: Stimulus design algorithm for distinguishing between circuit hypotheses

input: $\mathbf{s_0}$; `// seed stimulus`
, $f_A(s; \mathbf{W_A})$; `// circuit hypothesis A`
, $f_B(s; \mathbf{W_B})$; `// circuit hypothesis B`
Initialize an ANN whose output is the squared difference between the output of network A, and that of network B (Fig. 6.11).
fix $\mathbf{W_A}$ and $\mathbf{W_B}$ in place
Solve $\mathbf{s}^* \leftarrow \text{argmax}_{\mathbf{s}}(f(s; \mathbf{W_A}) - f(s; \mathbf{W_B}))^2$; `// from the random initialization`
return: $\mathbf{s}^*$

6.5 Summary

In this chapter, we have simulated several different active stimulus generation networks. Algorithm 3 provided an improvement on system identification. Though the others did not, they do demonstrate useful information about how training proceeds in these ANNs. For example, we learned that different types of stimuli are preferable

in early vs. late training, and our simulations also emphasized the importance of having a diverse stimulus set.

Chapter 7

CONCLUSION AND FUTURE DIRECTIONS

In this thesis, we have thoroughly explored the application of ANNs to system identification of neuronal microcircuits in the retina. We have demonstrated that theoretically, a retina-like toy model can be shown to be uniquely identifiable. Importantly, we saw that this is reliant on a sign constraint on at least one set of weights. We saw that the necessity of sign constraints carried over to our empirical simulations, in which we showed that nonlinear system identification with ANNs is feasible for multiple different retinally-inspired architectures in a parametric regime that is relevant to retinal experimental conditions. We also derived a heuristic by which an experimenter working in a new system can decide how many free parameters the ANN should have. We applied this technique to real data from alpha retinal ganglion cells and confirmed known results while also suggesting future directions for biological research. We also simulated several active stimulus generation algorithms and provided a basis for future research in this area.

Many questions still remain to be untangled. The simulations can be extended to further circuit architectures with more layers or with the addition of recurrence. The alpha cell work can be extended to a larger dataset, and biological circuit dissection experiments can be undertaken to confirm the hypotheses presented in this work. The active stimulus generation techniques can be refined and even tested on live retinal circuits. And this technique shows great promise for extension to other systems. Within this work we have mentioned gene circuits as one possible target, but another logical next step would be to move one synapse down in the visual system, and to apply this technique to neurons in superficial superior colliculus or lateral geniculate nucleus. One could even go further, into deeper colliculus or the visual cortex. As mentioned in the introduction, this technique is applicable to any feedforward circuit for which partial foreknowledge exists and for which the input and output can be easily accessed. The possibilities are endless, and my hope is that this work can open the door to more play and exploration of this type of nonlinear system identification using the ever-improving tools of deep neural networks.

BIBLIOGRAPHY

[1] Reza Abbasi-Asl, Yuansi Chen, Adam Bloniarz, Michael Oliver, Ben DB Willmore, Jack L Gallant, and Bin Yu. The deeptune framework for modeling and characterizing neurons in visual cortex area v4. *bioRxiv*, page 465534, 2018.

[2] Francesca Albertini, Eduardo D. Sontag, and Vincent Maillot. Uniqueness of weights for neural networks. 1993.

[3] Uri Alon. *An introduction to systems biology: design principles of biological circuits*. CRC Press/Taylor & Francis Group, 2006.

[4] Andrew, Xu, Jian, Deniz, Cherian, Steven H., Zhu, Yongling, and Suraj. Intersectional strategies for targeting amacrine and ganglion cell types in the mouse retina, Aug 2018. URL `https://www.frontiersin.org/articles/10.3389/fncir.2018.00066/full`.

[5] S. A. Baccus, B. P. Olveczky, M. Manu, and M. Meister. A retinal circuit that computes object motion. *Journal of Neuroscience*, 28(27):6807–6817, Feb 2008. doi: 10.1523/jneurosci.4206-07.2008.

[6] Stephen A. Baccus and Markus Meister. Fast and slow contrast adaptation in retinal circuitry. *Neuron*, 36(5):909–919, 2002. doi: 10.1016/s0896-6273(02)01050-4.

[7] J. Alexander Bae, Shang Mu, Jinseop S. Kim, Nicholas L. Turner, Ignacio Tartavull, Nico Kemnitz, Chris S. Jordan, Alex D. Norton, William M. Silversmith, Rachel Prentki, and et al. Digital museum of retinal ganglion cells with dense anatomy and physiology. *Cell*, 173(5):1293–1306, 2017. doi: 10.1101/182758.

[8] H. B. Barlow and R. M. Hill. Selective sensitivity to direction of movement in ganglion cells of the rabbit retina. *Science*, 139(3553):412–412, 1963. doi: 10.1126/science.139.3553.412.

[9] H B Barlow and W R Levick. The mechanism of directionally selective units in rabbit's retina. *J. Physiol*, 178:477–504, 1965.

[10] H.b. Barlow, W.r. Levick, and M. Yoon. Responses to single quanta of light in retinal ganglion cells of the cat. *Vision Research*, 11:87–101, 1971. doi: 10.1016/0042-6989(71)90033-2.

[11] Ror Bellman and Karl Johan Åström. On structural identifiability. *Mathematical biosciences*, 7(3-4):329–339, 1970.

[12] Michael J Berry, David K Warland, and Markus Meister. The structure and precision of retinal spike trains. 94:5411–5416, 1997. URL www.pnas.org.

[13] Kevin L. Briggman, Moritz Helmstaedter, and Winfried Denk. Wiring specificity in the direction-selectivity circuit of the retina. *Nature*, 471:183–190, 3 2011. ISSN 00280836. doi: 10.1038/nature09818.

[14] Tsai-Wen Chen, Trevor J. Wardill, Yi Sun, Stefan R. Pulver, Sabine L. Renninger, Amy Baohan, Eric R. Schreiter, Rex A. Kerr, Michael B. Orger, Vivek Jayaraman, and et al. Ultrasensitive fluorescent proteins for imaging neuronal activity, Jul 2013. URL https://www.nature.com/articles/nature12354.

[15] Saskia E. J. de Vries, Stephen A. Baccus, and Markus Meister. The projective field of a retinal amacrine cell. *Journal of Neuroscience*, 31(23):8595–8604, 2011. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.5662-10.2011. URL https://www.jneurosci.org/content/31/23/8595.

[16] David Donoho and Victoria Stodden. When Does Non-Negative Matrix Factorization Give a Correct Decomposition into Parts? In S. Thrun, L. K. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 1141–1148. MIT Press, 2004.

[17] Felice A. Dunn and Rachel O. L. Wong. Wiring patterns in the mouse retina: collecting evidence across the connectome, physiology and light microscopy. *The Journal of Physiology*, 592(22):4809–4823, 2014. doi: 10.1113/jphysiol.2014.277228.

[18] Germán A. Enciso, Michael Rempe, Andrey V. Dmitriev, Konstantin E. Gavrikov, David Terman, and Stuart C. Mangel. A model of direction selectivity in the starburst amacrine cell network. *Journal of Computational Neuroscience*, 28(3):567–578, 2010. doi: 10.1007/s10827-010-0238-3.

[19] E V Famiglietti. Synaptic organization of starburst amacrine cells in rabbit retina: Analysis of serial thin sections. *J Comp Neurol*, 309(1):40–70, Jul 1991.

[20] Charles Fefferman. Reconstructing a neural net from its output. *Revista Matematica Iberoamericana*, 10:507–555, 1994.

[21] Evan H. Feinberg and Markus Meister. Orientation columns in the mouse superior colliculus, Dec 2014. URL https://www.nature.com/articles/nature14103.

[22] Katrin Franke, Philipp Berens, Timm Schubert, Matthias Bethge, Thomas Euler, and Tom Baden. Inhibition decorrelates visual feature representations in the inner retina. *Nature*, 542(7642):439–444, 2017. doi: 10.1038/nature21394.

[23] James W. Fransen and Bart G. Borghuis. Temporally diverse excitation generates direction-selective responses in on- and off-type retinal starburst amacrine cells. *Cell Reports*, 18:1356–1365, 2 2017. ISSN 22111247. doi: 10.1016/j.celrep.2017.01.026.

[24] K Fukushima. Neocognitron: a self organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, 1980. URL https://www.ncbi.nlm.nih.gov/pubmed/7370364/.

[25] Maria Neimark Geffen, Saskia E J de Vries, and Markus Meister. Retinal ganglion cells can rapidly change polarity from off to on. *PLoS Biology*, 5(3), 2007.

[26] Krishna K. Ghosh, Sascha Bujan, Silke Haverkamp, Andreas Feigenspan, and Heinz Wässle. Types of bipolar cells in the mouse retina. *Journal of Comparative Neurology*, 469(1):70–82, 2003. doi: 10.1002/cne.10985.

[27] Krishna K. Ghosh, Sascha Bujan, Silke Haverkamp, Andreas Feigenspan, and Heinz Wässle. Types of bipolar cells in the mouse retina. *Journal of Comparative Neurology*, 469:70–82, 1 2004. ISSN 00219967. doi: 10.1002/cne.10985.

[28] Tim Gollisch and Markus Meister. Eye smarter than scientists believed: Neural computations in circuits of the retina. *Neuron*, 65(2):150–164, 2010. doi: 10.1016/j.neuron.2009.12.009.

[29] Matthew J. Greene, Jinseop S. Kim, and H. Sebastian Seung. Analogous convergence of sustained and transient inputs in parallel on and off pathways for retinal motion computation. *Cell Reports*, 14(8):1892–1900, 2016. doi: 10.1016/j.celrep.2016.02.001.

[30] Rong Gui, Quan Liu, Yuangen Yao, Haiyou Deng, Chengzhang Ma, Ya Jia, and Ming Yi. Noise decomposition principle in a coherent feed-forward transcriptional regulatory loop. *Frontiers in Physiology*, 7, 2016. doi: 10.3389/fphys.2016.00600.

[31] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical learning with sparsity: the lasso and generalizations*. CRC press, 2015.

[32] Susanne E. Hausselt, Thomas Euler, Peter B. Detwiler, and Winfried Denk. A dendrite-autonomous mechanism for direction selectivity in retinal starburst amacrine cells. *PLoS Biology*, 5:1474–1493, 7 2007. ISSN 15449173. doi: 10.1371/journal.pbio.0050185.

[33] S Hochstein and R M Shapley. Linear and nonlinear spatial subunits in y cat retinal ganglion cells. *The Journal of Physiology*, 262(2):265–284, 1976. doi: https://doi.org/10.1113/jphysiol.1976.sp011595. URL https://physoc.onlinelibrary.wiley.com/doi/abs/10.1113/jphysiol.1976.sp011595.

[34] K. Huang, N. D. Sidiropoulos, and A. Swami. Non-negative matrix factorization revisited: Uniqueness and algorithm for symmetric decomposition. *IEEE Transactions on Signal Processing*, 62(1):211–224, 2014.

[35] J.-N. Hwang, J.j. Choi, S. Oh, and R.j. Marks. Query-based learning applied to partially trained multilayer perceptrons. *IEEE Transactions on Neural Networks*, 2(1):131–136, 1991. doi: 10.1109/72.80299.

[36] Ben James, Léa Darnet, José Moya-Díaz, Sofie-Helene Seibel, and Leon Lagnado. An amplitude code increases the efficiency of information transmission across a visual synapse. *bioRxiv*, 2018. doi: 10.1101/328682. URL https://www.biorxiv.org/content/early/2018/05/30/328682.

[37] James J. Jun, Nicholas A. Steinmetz, Joshua H. Siegle, Daniel J. Denman, Marius Bauza, Brian Barbarits, Albert K. Lee, Costas A. Anastassiou, Alexandru Andrei, Çağatay Aydın, and et al. Fully integrated silicon probes for high-density recording of neural activity, Nov 2017. URL https://www.nature.com/articles/nature24636.

[38] Justin Keat, Pamela Reinagel, R.clay Reid, and Markus Meister. Predicting every spike: A model for the responses of visual neurons. *Neuron*, 30(3): 803–817, 2001. doi: 10.1016/s0896-6273(01)00322-1.

[39] Jinseop S. Kim, Matthew J. Greene, Aleksandar Zlateski, Kisuk Lee, Mark Richardson, Srinivas C. Turaga, Michael Purcaro, Matthew Balkam, Amy Robinson, Bardia F. Behabadi, Michael Campos, Winfried Denk, and H. Sebastian Seung. Space-time wiring specificity supports direction selectivity in the retina. *Nature*, 509:331–336, 2014. ISSN 14764687. doi: 10.1038/nature13240.

[40] Jinseop S. Kim, Matthew J. Greene, Aleksandar Zlateski, Kisuk Lee, Mark Richardson, Srinivas C. Turaga, Michael Purcaro, Matthew Balkam, Amy Robinson, Bardia F. Behabadi, and et al. Space–time wiring specificity supports direction selectivity in the retina. *Nature*, 509(7500):331–336, 2014. doi: 10.1038/nature13240.

[41] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. URL http://arxiv.org/abs/1412.6980. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

[42] Hiroaki Kitano, Akira Funahashi, Yukiko Matsuoka, and Kanae Oda. Using process diagrams for the graphical representation of biological networks. *Nature biotechnology*, 23(8):961–966, 2005.

[43] David Klindt, Alexander S Ecker, Thomas Euler, and Matthias Bethge. Neural system identification for large populations separating "what" and "where". In *Advances in Neural Information Processing Systems*, pages 3506–3516, 2017.

[44] Brenna Krieger, Mu Qiao, David L. Rousso, Joshua R. Sanes, and Markus Meister. Four alpha ganglion cell types in mouse retina: Function, structure, and molecular signatures. *PLoS ONE*, 12, 7 2017. ISSN 19326203. doi: 10.1371/journal.pone.0180091.

[45] John G Kuschewski, Stefen Hui, and Stanislaw H Zak. Application of feed-forward neural networks to dynamical system identification and control. *IEEE Transactions on Control Systems Technology*, 1(1):37–49, 1993.

[46] Hans Laurberg, Mads Græsbøll Christensen, Mark D. Plumbley, Lars Kai Hansen, and Søren Holdt Jensen. Theorems on Positive Data: On the Uniqueness of NMF. *Computational Intelligence and Neuroscience*, 2008:788 – 791, 2008. doi: 10.1155/2008/764206.

[47] Seunghoon Lee and Z. Jimmy Zhou. The synaptic mechanism of direction selectivity in distal processes of starburst amacrine cells. *Neuron*, 51:787–799, 9 2006. ISSN 08966273. doi: 10.1016/j.neuron.2006.08.007.

[48] Yen-Huan Li, Jonathan Scarlett, Pradeep Ravikumar, and Volkan Cevher. Sparsistency of ℓ_1-regularized M-estimators. In *AISTATS*, 2015.

[49] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. *Computer Vision – ECCV 2014 Lecture Notes in Computer Science*, page 740–755, 2014. doi: 10.1007/978-3-319-10602-1_48.

[50] Lennart Ljung. System identification. *Wiley encyclopedia of electrical and electronics engineering*, pages 1–19, 1999.

[51] Niru Maheswaranathan, David B. Kastner, Stephen A. Baccus, and Surya Ganguli. Inferring hidden structure in multilayered neural circuits. *bioRxiv*, 3 2017. doi: 10.1101/120956.

[52] Calvin McCarter and Seyoung Kim. On sparse gaussian chain graph models. *Advances in Neural Information Processing Systems*, 27:3212–3220, 2014.

[53] Lane McIntosh and Niru Maheswaranathan. A deep learning model of the retina, 2015.

[54] Lane McIntosh, Niru Maheswaranathan, Aran Nayebi, Surya Ganguli, and Stephen Baccus. Deep learning models of the retinal response to natural scenes. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 1369–1377. Curran Associates, Inc., 2016. URL `http://papers.nips.cc/paper/6388-deep-learning-models-of-the-retinal-response-to-natural-scenes.pdf`.

[55] Markus Meister, Jerome Pine, and Denis A Baylor. Multi-neuronal signals from the retina: acquisition and analysis. *Journal of Neuroscience Methods*, 51:95–106, 1994. doi: 10.1016/0165-0270(94)90030-2.

[56] Markus Meister, Jerome Pine, and Denis A. Baylor. Multi-neuronal signals from the retina: acquisition and analysis. *Journal of Neuroscience Methods*, 51(1):95–106, 1994. doi: 10.1016/0165-0270(94)90030-2.

[57] A. Mizrahi. Synaptogenesis in the adult cns–olfactory system. *Cellular Migration and Formation of Neuronal Connections*, page 739–755, 2013. doi: 10.1016/b978-0-12-397266-8.00112-5.

[58] Thomas A Münch, Rava Azeredo Da Silveira, Sandra Siegert, Tim James Viney, Gautam B Awatramani, and Botond Roska. Approach sensitivity in the retina processed by a multifunctional neural circuit. *Nature Neuroscience*, 12 (10):1308–1316, Jun 2009. doi: 10.1038/nn.2389.

[59] Nicholas Oesch, Thomas Euler, and W. Rowland Taylor. Direction-selective dendritic action potentials in rabbit retina. *Neuron*, 47(5):739–750, 2005. doi: 10.1016/j.neuron.2005.06.036.

[60] Bence Olveczky, Stephen Baccus, and Markus Meister. Segregation of object and background motion in the retina. *Nature*, 423:401–8, 06 2003. doi: 10.1038/nature01652.

[61] Ander Ozaita, Jerome Petit-Jacques, Béla Völgyi, Chi Shun Ho, Rolf H. Joho, Stewart A. Bloomfield, and Bernardo Rudy. A unique role for kv3 voltage-gated potassium channels in starburst amacrine cell signaling in mouse retina. *Journal of Neuroscience*, 24:7335–7343, 8 2004. ISSN 02706474. doi: 10.1523/JNEUROSCI.1275-04.2004.

[62] Marius Pachitariu, Nicholas A Steinmetz, Shabnam N Kadir, Matteo Carandini, and Kenneth D Harris. Fast and accurate spike sorting of high-channel count probes with kilosort. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4448–4456. Curran Associates, Inc., 2016.

[63] Christopher L. Passaglia and John B. Troy. Impact of noise on retinal coding of visual signals. *Journal of Neurophysiology*, 92(2):1023–1033, 2004. doi: 10.1152/jn.01089.2003.

[64] Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12(1):145–151, 1999. ISSN 0893-6080. doi: https://doi.org/10.1016/S0893-6080(98)00116-6. URL `https://www.sciencedirect.com/science/article/pii/S0893608098001166`.

[65] Elio Raviola and Giuseppina Raviola. Structure of the synaptic membranes in the inner plexiform layer of the retina: A freeze-fracture study in monkeys and

rabbits. *The Journal of Comparative Neurology*, 209(3):233–248, 1982. doi: 10.1002/cne.902090303.

[66] Esteban Real, Hiroki Asari, Tim Gollisch, and Markus Meister. Neural circuit inference from function to structure. *Current Biology*, 27(2):189–198, 2017. doi: 10.1016/j.cub.2016.11.040.

[67] David Rolnick and Konrad P. Kording. Identifying weights and architectures of unknown re{lu} networks, 2020. URL `https://openreview.net/forum?id=HklFUlBKPB`.

[68] Botond Roska and Markus Meister. *The Retina Dissects the Visual Scene into Distinct Features*, page 163–182. MIT Press, 2014.

[69] Basil Saeed, Snigdha Panigrahi, and Caroline Uhler. Causal structure discovery from distributions arising from mixtures of dags. In *International Conference on Machine Learning*, pages 8336–8345. PMLR, 2020.

[70] Cornelius Schröder, David Klindt, Sarah Strauss, Katrin Franke, Matthias Bethge, Thomas Euler, and Philipp Berens. System identification with biophysical constraints: A circuit model of the inner retina. 2020. doi: 10.1101/2020.06.16.154203.

[71] Katja Seeliger, Luca Ambrogioni, Yağmur Güçlütürk, Umut Güçlü, and Marcel AJ van Gerven. End-to-end neural system identification with neural information flow. *bioRxiv*, page 553255, 2019.

[72] Jie Shen and Ping Li. On the iteration complexity of support recovery via hard thresholding pursuit. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3115–3124. JMLR. org, 2017.

[73] Sandra Siegert, Brigitte Gross Scherf, Karina Del Punta, Nick Didkovsky, Nathaniel Heintz, and Botond Roska. Genetic address book for retinal cell types. *Nature Neuroscience*, 12(9):1197–1204, Feb 2009. doi: 10.1038/nn.2370.

[74] Kyung-Ah Sohn and Seyoung Kim. Joint estimation of structured sparsity and output structure in multiple-output regression via inverse-covariance regularization. In *Artificial Intelligence and Statistics*, pages 1081–1089. PMLR, 2012.

[75] Raghav Somani, Chirag Gupta, Prateek Jain, and Praneeth Netrapalli. Support recovery for orthogonal matching pursuit: upper and lower bounds. In *Advances in Neural Information Processing Systems*, pages 10814–10824, 2018.

[76] P Sterling, MA Freed, and RG Smith. Architecture of rod and cone circuits to the on-beta ganglion cell. *Journal of Neuroscience*, 8(2):623–642, 1988. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.08-02-00623.1988. URL `https://www.jneurosci.org/content/8/2/623`.

[77] Todd Stincic, Robert G. Smith, and W. Rowland Taylor. Time course of epscs in on-type starburst amacrine cells is independent of dendritic location. *Journal of Physiology*, 594:5685–5694, 10 2016. ISSN 14697793. doi: 10.1113/JP272384.

[78] Greg J. Stuart and Bert Sakmann. Active propagation of somatic action potentials into neocortical pyramidal cell dendrites. *Nature*, 367(6458):69–72, 1994. doi: 10.1038/367069a0.

[79] Héctor J. Sussmann. Uniqueness of the weights for minimal feedforward nets with a given input-output map. *Neural Networks*, 1992.

[80] Hidenori Tanaka, Aran Nayebi, Niru Maheswaranathan, Lane McIntosh, Stephen Baccus, and Surya Ganguli. From deep learning to mechanistic understanding in neuroscience: the structure of retinal prediction. In *Advances in Neural Information Processing Systems 32*, pages 8537–8547. Curran Associates, Inc., 2019.

[81] Fuchou Tang, Catalin Barbacioru, Yangzhou Wang, Ellen Nordman, Clarence Lee, Nanlan Xu, Xiaohui Wang, John Bodeau, Brian B Tuch, Asim Siddiqui, and et al. mrna-seq whole-transcriptome analysis of a single cell, Apr 2009. URL `https://www.nature.com/articles/nmeth.1315`.

[82] W.r. Taylor and R.g. Smith. The role of starburst amacrine cells in visual signal processing. *Visual Neuroscience*, 29(1):73–81, 2012. doi: 10.1017/s0952523811000393.

[83] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[84] Robert E Tillman and Frederick Eberhardt. Learning causal structure from multiple datasets with similar variable sets. *Behaviormetrika*, 41(1):41–64, 2014.

[85] Yoshihiko Tsukamoto and Naoko Omi. Classification of mouse retinal bipolar cells: Type-specific connectivity with special reference to rod-driven aii amacrine pathways. *Frontiers in Neuroanatomy*, 11, 2017. doi: 10.3389/fnana.2017.00092.

[86] John J. Tukker, W. Rowland Taylor, and Robert G. Smith. Direction selectivity in a model of the starburst amacrine cell. *Visual Neuroscience*, 21:611–625, 7 2004. ISSN 09525238. doi: 10.1017/S0952523804214109.

[87] Verner Vlacic and Helmut Bölcskei. Neural network identifiability for a family of sigmoidal nonlinearities. *ArXiv*, abs/1906.06994, 2019.

[88] H. Wassle and B. B. Boycott. Functional architecture of the mammalian retina. *Physiological Reviews*, 71(2):447–480, 1991. doi: 10.1152/physrev.1991.71.2.447.

[89] Charles D. Woody, Daniel L. Alkon, James L. Macgaugh, Wilfrid Rall, and Idan Segev. *Dendritic Spine Synapses, Excitable Spine Clusters, and Plasticity*. Plenum Press, 1989.

[90] Harry J. Wyatt and Nigel W. Daw. Specific effects of neurotransmitter antagonists on ganglion cells in rabbit retina. *Science*, 191:204–205, 1976. ISSN 00368075. doi: 10.1126/science.1857.

[91] Daniel L K Yamins, Ha Hong, Charles F Cadieu, Ethan A Solomon, Darren Seibert, and James J DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, 111:8619–8624, 2014.

[92] Daniel LK Yamins, Ha Hong, Charles F Cadieu, Ethan A Solomon, Darren Seibert, and James J DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, 111(23):8619–8624, 2014.

[93] Yifeng Zhang, In-Jung Kim, Joshua R. Sanes, and Markus Meister. The most numerous ganglion cell type of the mouse retina is a selective feature detector, Sep 2012. URL `https://www.pnas.org/content/109/36/E2391`.

[94] Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. Dags with no tears: Continuous optimization for structure learning. *Advances in Neural Information Processing Systems*, 31:9472–9483, 2018.