Examining the Dynamics of Biologically Inspired Systems Far From Equilibrium

Jacob A. Carroll

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Physics

Uwe C. Täuber, Chair
Michel Pleimling
Shengfeng Cheng
Eric Sharpe

February 20, 2019
Blacksburg, Virginia

Keywords: Non-equilibrium systems, Surface plasmon resonance, Neural avalanches, Neural networks

Examining the Dynamics of Biologically Inspired Systems Far From Equilibrium

Jacob A. Carroll

(ABSTRACT)

Non-equilibrium systems have no set method of analysis, and in this work we present three very different non-equilibrium models inspired by biological systems and phenomena that we analyze through computational means.

In the first system, a surface plasmon resonance (SPR) cell, a mean-field analysis is commonly used to extract binding rates between chemicals. We simulate the SPR cell and compare computational results with a mean-field approximation, and find that such a simplification is only valid for low Damköhler number, the ratio between the transport and reaction time-scales, after which the system becomes diffusion-limited.

The second system is an avalanching neural network that models cascading neural activity observed across various species. We used a model devised by Lombardi, Herrmann, de Arcangelis et al. to simulate this system and characterized its behavior as the fraction of inhibitory neurons changed. At low fractions of inhibitory neurons we observed epileptic-like behavior in the systems' power spectral density (PSD), and extended tails in the avalanche strength and duration distributions, which dominate the system. We also observed how the connectivity of these networks evolved under the effects of different inhibitory fractions p_{inh}, and found high fractions p_{inh} cause networks to evolve more sparsely, while networks with low fractions maintain their initial connectivity. We demonstrated two strategies to control the extreme avalanches present at low inhibitory fractions through the random or targeted disabling of neurons.

The final system is a sparsely encoding convolutional neural network, a computational system inspired by the human visual cortex that has been engineered to reconstruct inputted images via a series of "patterns" learned from previous images as basis elements. The network attempts to do so "sparsely," so that the fewest number of neurons are used. Such systems are used for denoising tasks, where noisy or fragmented images are reconstructed. We observed a minimum in this denoising error as the fraction of active neurons was varied, and observed the depth and location of this minimum to obey finite-size scaling laws that suggest the system is undergoing a second-order phase transition.

Examining the Dynamics of Biologically Inspired Systems Far From Equilibrium

Jacob A. Carroll

(GENERAL AUDIENCE ABSTRACT)

Non-equilibrium systems have no set method of analysis, and a wide array of dynamics can be present in such systems. In this work we present three very different non-equilibrium models, inspired by biological systems and phenomena, that we analyze through computational means to showcase both the range of dynamics encompassed by these systems, as well as various techniques used to analyze them.

The first system we model is a surface plasmon resonance (SPR) cell, a device used to determine the binding rates between various species of chemicals. We simulate the SPR cell and compare these computational results with a mean-field approximation, and find that such a simplification fails for a wide range of reaction rates that have been observed between different species of chemicals. Specifically, the mean-field approximation places limits on the possible resolution of the measured rates, and such an analysis fails to capture very fast dynamics between chemicals.

The second system we analyzed is an avalanching neural network that models cascading neural activity seen in monkeys, rats, and humans. We used a model devised by Lombardi, Herrmann, de Arcangelis et al. to simulate this system and characterized its behavior as the fraction of inhibitory neurons was changed. At low fractions of inhibitory neurons we observed epileptic-like behavior in the system, as well as extended tails in the avalanche strength and duration distributions, which dominate the system in this regime. We also observed how the connectivity of these networks evolved under the effects of different inhibitory fractions, and found the high fractions of inhibitory neurons cause networks to evolve more

sparsely, while networks with low fractions maintain their initial connectivity. We demonstrated two strategies to control the extreme avalanches present at low inhibitory fractions through either the random or targeted disabling of neurons.

The final system we present is a sparsely encoding convolutional neural network, a computational system inspired by the human visual cortex that has been engineered to reconstruct images inputted into the network using a series of "patterns" learned from previous images as basis elements. The network attempts to do so "sparsely," so that the fewest number of neurons are used. Such systems are often used for denoising tasks, where noisy or fragmented images are reconstructed. We observed a minimum in this denoising error as the fraction of active neurons was varied, and observed the depth and location of this minimum to obey finite-size scaling laws that suggest the system is undergoing a second-order phase transition. We can use these finite-size scaling relations to further optimize this system by tuning it to the critical point for any given system size.

Dedication

To my Mother, Father, Sister and Partner.

Acknowledgments

I would like to thank my advisor Professor Uwe Täuber, for his continued support, advice, and mentorship throughout my graduate career. He has pushed me to be the best researcher and scientist I could be, and I have greatly valued my time as his student.

I would also like to thank Professor Michel Pleimling for his support and guidance throughout my time at Virginia Tech, as well as for many valuable discussions about my work.

I also want to thank Dr. Garret Kenyon, of Los Alamos National Lab, for his mentorship during my Summers there.

Finally, I would to like thank my remaining committee members Professor Shengfeng Cheng and Professor Eric Sharpe, who in addition to serving on my committee and helping to steer my growth in the PhD. program, were also truly excellent teachers.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The vast majority of real-world systems operate far from thermodynamic equilibrium, as most macroscopic physical systems are driven by one or more external forces. This accounts for any biological or biologically inspired systems, which require constant external stimuli to operate or else they equilibrate towards death [1].

Unfortunately, exact analytical solutions to non-equilibrium systems are rare, and in many cases the only way to properly analyze a sufficiently complex system far from equilibrium is to construct a model that captures its relevant dynamics and run many simulations at extensive computational cost. However, analytical solutions can be found in non-equilibrium systems undergoing continuous (second-order) phase transitions close to critical points in the system's phase space.

A phase transition is a transformation of a system's macroscopic dynamics through the change of some external control parameter. Phase transitions can be classified through the behavior of their "order parameter," which is a macroscopic observable chosen to distinguish between phases of the system. Examples of order parameters include the net magnetization of a ferromagnet, or the difference in density between the solid and liquid phases of solid and liquid matter. Order parameters are chosen to be zero in one of the phases, typically the high-temperature, disordered phase, and change to a non-zero value outside of that phase, though their behavior as they pass through the point of transition between phases can vary. Some systems will have a discontinuous change in their order parameter as they

transition between phases, such as the solid to liquid and liquid to solid transitions, and these systems are said to undergo a "first-order" phase transition. Other systems' order parameter will change continuously as they transition, such as the transition from ferromagnetism to paramagnetism in some magnetic materials as temperature is increased. This type of transition is a "second-order" or continuous phase transition.

The correlation length (i.e. the length-scale in which elements of the system can be correlated) of first-order phase transitions is finite. In contrast, in second-order phase transitions' correlation lengths become infinite (in infinitely sized systems) at the "critical point," the point of a continuous transition between two (or more) phases, and wash away any short-scale system-specific behavior. Additionally, the relaxation time of the system (i.e. the time it takes for the system to relax to a steady state) diverges at critical points during a continuous phase transition.

This divergence of correlation length and relaxation time causes these critical systems to display scale-free behavior where their statics and dynamics can be reduced to a set of power laws that themselves are governed by a set of "critical exponents." Additionally, system-specific details of critical systems are destroyed by these long-range correlations, and systems near a critical point can display universal behavior where the statics and dynamics of very different systems are governed by the same set of critical exponents [2, 3]. Systems that share the same set of critical exponents governing their statics belong to a "static universality class." The dynamics of systems in static universality classes can still display different dynamical behavior, and can be further separated into "dynamical universality classes" that share the critical exponents governing their dynamics as well.

The scale free behavior governed by these critical exponents often persists even in systems that are driven far from equilibrium [4], which allows the machinery developed for the analysis of continuous phase transitions to be applied to non-equilibrium systems that are sufficiently

close to a critical point as well as other systems displaying generic scale invariance. However, even these critical non-equilibrium systems with possible analytical descriptions often require simulations to be performed in order to be uncover their critical exponents.

While the vast majority of real-world systems are not in equilibrium, not all non-equilibrium systems need model some real-world dynamics. Indeed, in some computational cases the model is the system itself, such as with machine learning focused neural networks, where the goal of such a system is not to model or capture the dynamics of some physical phenomena, but to solve certain computational problems. Neural networks are systems constructed by connecting models of biological neurons together in a system specific manner. There are many different models of neurons used to construct such networks, but all of them share certain characteristics, such as integrating incoming signals into an internal potential stored inside the neuron, which mimics a biological neuron's ability to store information in the potential difference across its cell wall [5, 6], and transmitting an output that is dependent in some generally non-linear way on the strength of its internal potential. Such systems are of great importance to the fields of computer science and machine learning, where neural networks have been devised to perform such tasks as image recognition, image reconstruction, and data analysis in an unsupervised setting [7], i.e. these networks solve these problems without being specifically programmed to do so.

The initial resemblance of machine learning focused neural networks to any physical system has often been engineered away, with the topology of such networks being very finely tuned for a specific problem, but the machinery and techniques used in the analysis of non-equilibrium systems can still be borrowed to optimize and improve these systems.

In this work we present three different models of biologically inspired systems far from equilibrium: a surface plasmon resonance cell, an avalanching neural network, and a sparsely encoding convolutional neural network, and demonstrate approaches on how to analyze, and

in the last two cases control, their dynamics.

1.1 Surface plasmon resonance

The first system, detailed in Chapter 2, is a Surface Plasmon Resonance (SPR) cell. This chapter is based on the work J. Carroll et al (2016) [8] published in Physical Biology.

An SPR cell is an experimental system used to determine the binding dynamics between different species of chemicals. The cell is constructed as a flow channel with a gold substrate on the bottom. One of the species of chemicals is bound to the gold substrate at a certain density and form the "receptors" of the system. The second species of chemical, referred to as the "ligands," is mixed with some solvent at a known concentration. This solution is then driven through the SPR cell and over the receptor surface at a constant current. The current that drives the ligand solution over the receptor surface forces the system out of equilibrium. As the ligands are transported to the receptors they bind and unbind according to the dynamics of their interactions. The index of refraction of the gold substrate changes as ligands bind to receptors, and this can be used to extract the bound ligand-receptor density as a function of time [9, 10].

Experimentally this bound ligand-receptor density is used to extract the association and dissociation rates between the ligand and receptor species, through the use of a mean-field approximation which ignores all spatial and temporal correlations and fluctuations. In Chapter 2 we model a SPR cell and simulate its relevant dynamics to show how the rates extracted through this simple mean-field approximation can differ from the true rates by several orders of magnitude due to the ignored spatio-temporal correlations and fluctuations, and demonstrate a regime in which this mean-field analysis fails.

This work demonstrates a common failing in the analysis of non-equilibrium systems: which

is to ignore any notion of space in the system and assume a perfect and universal ability of all components of a system to interact with each other as entailed in the mean-field approximation. Such an approximation ignores any spatio-temporal correlations, which in the case of non-equilibrium systems are often of great importance to the system dynamics.

1.2 Avalanching neural networks

The second system, an avalanching neural network, is a model of a purely biological system and differs greatly from the first SPR model. This system and the work we performed analyzing it is detailed in Chapter 3. Specifically, we studied the collective dynamics of large numbers of modeled biological neurons, connected to each other in a time and activity dependent manner. This chapter is based on the work J. Carroll et al. (2019) [11], which is currently in review for publication in Physical Review E.

Biological neurons operate through the transmission and reception of electro-chemical signals. These signals can have varying strengths, and upon reception are integrated and added to an internal potential of the receiving neuron. Once this internal potential exceeds a threshold value, the neuron "fires" and transmits its own signal to other neurons, proportional in strength to the internal potential it held at the time of firing. This "integrate and fire" behavior allows neurons transmitting strong signals to immediately exceed the firing potential of the receiving neurons, causing them to transmit their own signals [5, 6]. When this happens repeatedly a cascade or "avalanche" of activity can propagate throughout this neural network from only a single initial stimulus.

In Chapter 3 we used a model originally devised by Lombardi, Herrmann, de Arcangelis et al. [12, 13, 14, 15] to simulate this avalanching behavior of biological neural networks, in addition to their other relevant dynamics, and characterize the network behavior as the

inhibitory neuron fraction of the network was changed. In this context, inhibitory neurons are neurons that send negative signals to other neurons, i.e. they decrease the internal potential of neurons that receive their signals, "inhibiting" them.

We observed epileptic-like behavior in networks with very low fractions of inhibitory neurons, in which these networks sustained incredibly powerful and long lasting avalanches that dominated the networks. This corroborates results obtained by Lombardi et al [15]. We then demonstrate how this "epileptic" behavior can be curtailed and controlled through both the targeted disabling of highly connected excitatory neurons (i.e. neurons that increase internal potentials of other neurons), and the disabling of random excitatory neurons.

Additionally, we show how the connectivity of the network evolves as a result of the inhibitory fraction, with high inhibitory fractions creating more sparsely connected networks, and low inhibitory fractions generating networks that maintain their initial strong connectivity. This inverse relationship between network connectivity and inhibitory fractions helps to reinforce the emergence of extreme avalanches in networks with low inhibitory fractions, as it is easier for signals to propagate in a strongly connected network.

The complexity of this system makes analytical tools intractable, and well demonstrates how non-equilibrium systems often require a computational approach.

1.3 Sparsely encoding convolutional neural networks

Finally the third system, detailed in Chapter 4, represents another example of a neural network, though this time developed for use in the task of machine learning. This type of neural network is called a sparsely encoding convolution neural network, and has been heavily optimized for the task of pattern recognition in images, such that it is only loosely inspired by biological systems as opposed to the avalanching neural networks described in

Chapter 3. In this case, the model and the system of interest are one and the same, and we are not concerned with validly modeling a physical system in order to understand its dynamics. Rather, we want to use machinery created for the analysis of non-equilibrium systems to control and optimize this computational system.

The work in Chapter 4 is based on work done with Dr. Garrett T. Kenyon at Los Alamos National Laboratory, J. Carroll et al. (2017) [16], which was presented as conference proceedings at the Modeling and Learning Interactions from Complex Data workshop in the 2017 Neural Information Processing Systems (NIPS) conference. This work has been cleared for unrestricted release under LA-UR-17-26726.

A sparsely encoding convolutional neural network is a system that attempts to reconstruct visual input (in our case, 32x32 thumbnail images taken from the CIFAR-10 dataset [17]) as a linear combination of neuron external potentials and weight vectors as coefficients and basis vectors respectively. The system does this by minimizing an energy function that is composed of both an "error term": the difference between the original input and the network's reconstruction, and a "sparsity term": the sum of each neurons' external potential. Minimizing this energy enforces that the reconstruction generated by the network be sparse, i.e. that most of the neurons' external potentials be zero.

The neurons in this system are "leaky integrators," such that they integrate the weighted sum of their inputs and add the result to an internal potential, which will decay over time. Additionally each neuron will inhibit each other neuron proportional to the overlap of their weight vectors. Each neuron receives the same input, and only interacts with other neurons through the aforementioned inhibitions [18].

The weight vectors of each neuron are chosen in a very specific manner. A select number of neurons are initialized with a very sparse weight vector, corresponding spatially to a small

"patch" of the input image. This patch and the weights inside are convolved over the entire image by copying the original neuron and spatially shifting the location of this patch of weights until there is a neuron for every possible position of this patch on the input. In this way the original neuron's weight vector has been convolved over the entire image. Each neuron in a single "convolutional group" is collected together and forms a sub-layer of the network's neural layer, which is made of many of these sub-layers.

The patches of weights shared among each sub-layer can be iteratively trained to represent patterns common across many images. As the patches of weights converge to various patterns, the network can reconstruct images more accurately and more sparsely, as the weight vectors it uses as basis elements more accurately represent portions of the image it is reconstructing.

We performed this training process and copied the resulting weights to a network that is exposed to noisy images. As this new network tries to reconstruct noisy images using the previously trained weights, it must choose between accurately reconstructing the entire noisy input at the cost of sparsity, or reconstructing what it can while maintaining a sparse output. We observed that there is a particular value of sparsity for a given network size at which the network will most accurately denoise the noisy input images, and that this value is always at a minimum of reconstruction error versus sparsity.

We analyzed how the depth and location of this minimum changed as we varied the number of neurons in our network, and observed it to display finite-size scaling behavior, as the depth and location of this minimum followed power law relationships with the network's system size (in our case the number of neurons). We extracted exponents $\bar{\nu}$ and $\bar{\gamma}$ that describe the minimum's power law scaling relationships. Such power laws scaling relations are characteristic of critical systems, so the existence of this finite scaling behavior suggests that the system is undergoing a continuous (second-order) phase transition as the sparsity

of the system is varied.

Additionally, the finite-size scaling relations enable us to predict the optimum level of sparsity for a given network size, allowing us to using the machinery developed for analyzing critical non-equilibrium systems to optimize and control the dynamics of this sparsely encoding convolutional neural network.

Chapter 2

Ligand-receptor binding dynamics in surface plasmon resonance cells

The following chapter was adapted with minor modifications, with permission from Physical Biology, from our publication:

J. Carroll, M. Raum, K. Forsten-Williams, and U. C. Täuber. Ligand-receptor binding kinetics in surface plasmon resonance cells: a Monte Carlo analysis. Physical Biology 13, 066010 (2016).

2.1 Introduction

The accurate measurement of the reaction rates between different species of chemicals is a crucial component in the process of understanding and manipulating the biochemical processes which perpetuate or extinguish life [19, 20].

A common method of measuring these rates is via surface plasmon resonance (SPR) [21, 22]. SPR allows the binding dynamics between two species of chemicals to be measured in real time, and is performed by binding one of the two chemical species to a substrate (the receptor species), and then measuring the change in index of refraction as the other chemical species (the ligand species) flows over the substrate and the two chemicals interact [23, 24, 25]. See Fig. 2.1 for a schematic of the experimental setup.

Ideally, the data from this experiment allows for the easy extraction of the binding and unbinding rates. However, in SPR cells the rates of transport to the reaction surface can be quite slow relative to the reaction rates, i.e., it may take much longer to diffusively transport down to the receptor surface than it does to bind to that surface, so the well-mixed assumption of first-order reaction kinetics may not necessarily be valid. The rate of transport to the reaction surface combines with the intrinsic reaction rates to create the effective reaction rates that are measured in an SPR assay. In order to determine the intrinsic reaction rates the influence of the transport rate must be properly accounted for [26].

Most of the attempted approaches to the problem of decoupling the transport and reaction rates model the system with a deterministic process, where the dependence on the parameters of the SPR system is governed by a set of coupled differential partial rate equations [9, 27, 28]. Simulations for SPR systems are often derived from numerical solutions to these PDEs, but these solutions often fail to capture the spatial and temporal correlations between the ligands and the receptors as they interact, and ignore statistical fluctuations [29].

Monte Carlo simulations are a computational tool developed to numerically solve the basic master equation for stochastic processes, and faithfully encode account for the presence of fluctuations and correlations in the modeled system. Monte Carlo methods have found widespread application in the modeling of physical, chemical, and biological systems. Since we cannot provide a comprehensive overview of Monte Carlo techniques in this brief chapter, we refer the reader to Ref. [30] as a recent review of stochastic modeling for biological systems.

In this chapter, we present results from Monte Carlo simulations of SPR cells for a broad range of binding and unbinding rates that allow for the observation of how the presence of correlations and fluctuations influence SPR data. Reaction rates derived from the standard mean-field model of the reaction kinetics[1] will be compared with known intrinsic reaction

[1]The mean-field model of reaction kinetics is physics nomenclature for the well-mixed assumption of the

rates used in the simulations in order to determine the degree to which spatio-temporal correlations and fluctuations are important to the dynamics of the system.

2.2 Surface plasmon resonance

2.2.1 The structure of the surface plasmon resonance cell

The structure of the surface plasmon resonance cell is discussed in more detail in literature [9, 10], but the following section will attempt to give a brief overview.

A surface plasmon resonance cell (schematically detailed in Fig. 2.1) is constructed by embedding a gold substrate into the bottom a of flow cell with linear dimensions on the order of millimeters. Two chemical species are chosen with the goal of determining the binding dynamics between them. One of these species is designated the receptors, and the other the ligands. The receptors are typically distributed randomly along the gold substrate and fixed in place, creating the receptor surface. A non-reactive solvent has a predetermined concentration of ligands dissolved into it, and this solution is allowed to flow over the receptor surface at a constant flow velocity.

The ligands in the solution are transported diffusely down to the receptor surface, where they bind and unbind to the receptors according to their respective dynamics. The binding and unbinding of the ligands to the receptors cause the resonance energy of the surface plasmon waves in the gold substrate to change [9, 21, 33]. This change in the energy of the waves can be measured by shining a p-polarized beam of light onto the substrate through a prism. The prism allows the momentum of the incident beam to be varied, and when the momentum of the incident beam and the surface plasmons of the gold substrate are the same, the beam

law of mass action (i.e., physical and temporal correlations are ignored). The term 'mean-field' will be used to refer to this model throughout this chapter, but the two terms are equivalent [31, 32].

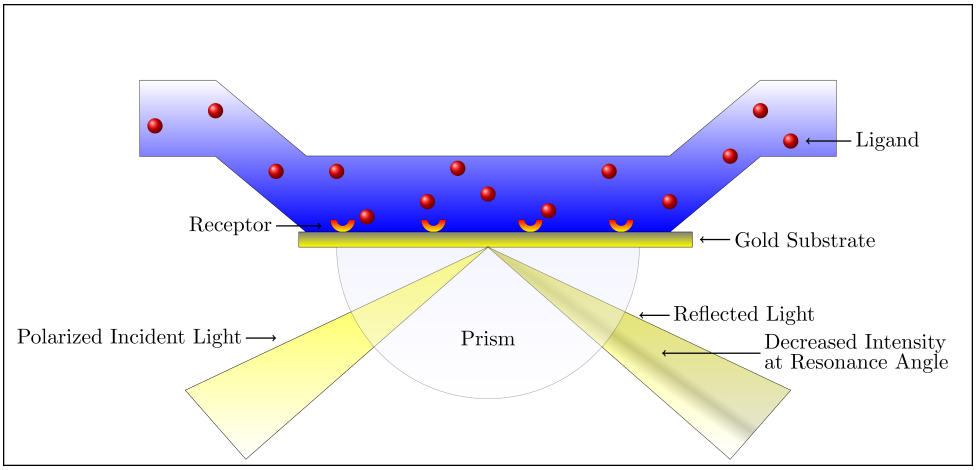


Figure 2.1: Surface plasmon resonance cell (schematic). A gold substrate is embedded at the bottom of a flow cell. Receptors (red-orange half circles) are distributed evenly across the substrate. Ligands (red spheres) are dissolved in a non-reactive solvent, and allowed to flow across the receptor surface with a constant concentration and flow rate. Ligands will be transported down to the receptor surface where they bind and unbind to the receptors according to their dynamics. Incident p-polarized light is shown through a prism onto the receptor surface. The angle at which resonance between the incident beam and the standing waves of electrons (plasmons) in the gold substrate can be measured by recording the angle at which the reflected light has a decreased intensity [9, 21, 33]. The extracted data of this resonance angle as a function of time can be rescaled to indicate the bound ligand density (i.e. the number of bound ligands normalized by the concentration of ligands in the flow cell) as a function of time [34].

and plasmons couple and create a surface plasmon polariton in the gold substrate. This coupling results in a decrease in energy of the reflected beam of light, and the momentum at which this occurs can be measured by recording the angle where the resonance between the incident beam and the surface plasmon appears. The change in resonance angle as a function of time can then be rescaled into a plot of bound ligand-receptor pairs as a function of time [34].

2.2.2 Stages of the surface plasmon resonance experiment

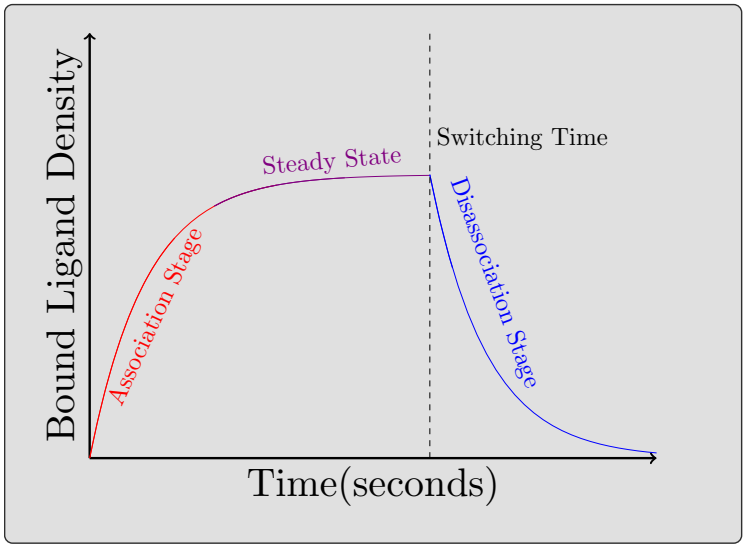


Figure 2.2: Stages of an SPR experiment. The red line indicates the association state of the SPR experiment, where a solution of ligands flows over the receptor surface with a constant concentration and fixed flow velocity. The bound concentration reaches a steady state (depicted in purple), at which point the concentration of incoming ligands is cut off, letting the bound ligands decay off the receptors in the dissociation stage, represented by the blue line.

The experimental process of surface plasmon resonance is typically performed in two stages. First, the solution of ligands is allowed to flow over the receptor surface with a constant concentration of ligands and fixed flow velocity. The system is allowed to evolve in this state until a steady-state concentration of bound ligands is observed. This stage of the experiment

is referred to as the association stage. Subsequently, the concentration of incoming ligands is cut off, and the number of bound ligand-receptor pairs is allowed to decay away, as the ligands gradually unbind. This stage of the experiment is known as the dissociation stage. The concentration of bound ligands is measured throughout both stages, and data similar to the kind depicted in Fig. 2.2 is generated. This chapter aims to replicate both stages via Monte Carlo simulations, in order to determine the role that the spatio-temporal correlations induced by diffusion-limited association and repeated ligand rebinding processes play in the dynamics of the SPR cell.

2.3 SPR cell model

2.3.1 Cell geometry

We model the SPR cell as a rectangular lattice, with lattice spacing of 10nm. The lattice is constructed with maximum dimensions of L_x, L_y, L_z on the x, y, and z axes, which correspond to the laboratory dimensions of the SPR chip. Periodic boundary conditions are imposed on the z axis, and a reflective boundary condition imposed along the $y = L_y$ top of the y axis. Ligands are introduced at the $x = 0$ surface, and perform a random walk to adjacent lattice sites until they encounter the $x = L_x$ surface, at which point they are removed from the lattice.

A subsection of the $y = 0$ surface is selected to model the receptor surface, from $x = x_0$ to $x = x_1$. Receptors are distributed evenly over this subsection with density R_0, and the receptors are modeled such that if a ligand is directly adjacent above the receptor, the ligand can bind to the receptor with a probability $\widetilde{k_+}$. Once the ligand is attached to a receptor it can no longer move, but can unbind from the receptor with probability $\widetilde{k_-}$. Ligands are assumed to be small enough that they do not interact in the lattice, and a receptor that is

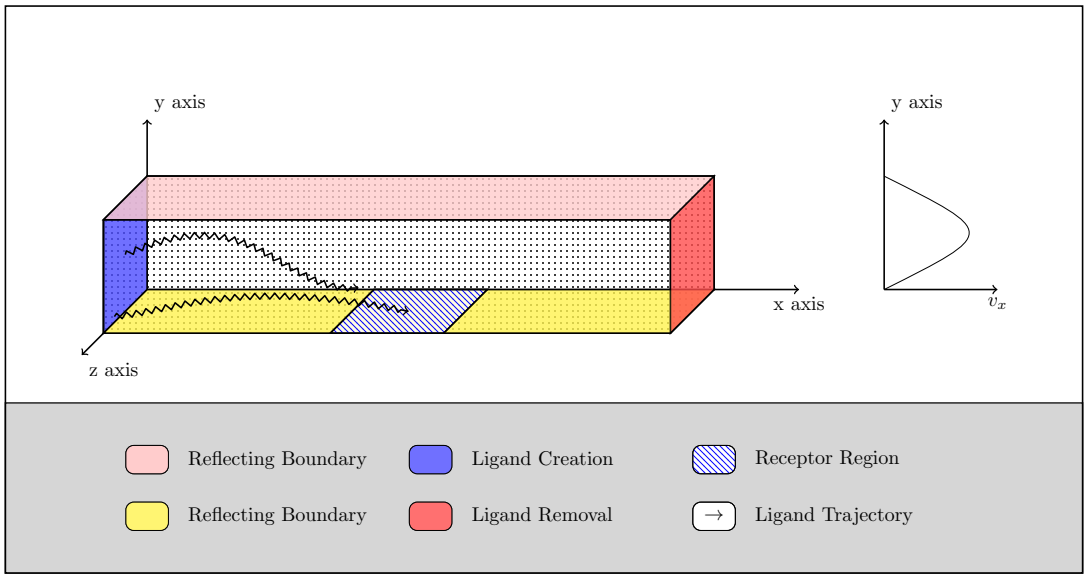


Figure 2.3: The discretized model of the SPR cell. Ligands are introduced at the ligand creation region depicted in solid blue, and perform a random walk through the lattice. This random walk is biased to create a parabolic flow profile (shown in the plot to the right of the schematic) that would be expected in the regime of laminar flow typical of SPR cells. The top and bottom planes of the lattice (depicted by the solid pink and solid yellow planes) form reflecting boundaries for the ligands. Receptors are evenly distributed in the receptor region (the dashed blue plane), and any ligand directly adjacent above a receptor has a chance to bind to it according to the association rate $\widetilde{k_+}$. Ligands perform their random walk through the lattice sites until they encounter the ligand removal region (depicted by the solid red plane), where they are removed from the simulation. In the association stage of the simulation, a ligand is immediately introduced at the ligand creation region to keep the concentration of ligands in the SPR cell constant, while in the dissociation stage of the simulation, the ligands are removed and not reintroduced, to allow the concentration of bound ligands to decay.

bound to a ligand cannot bind to another ligand until the first ligand unbinds.

A summary of the laboratory parameters of the SPR chip is given in Tab. 2.1, and a schematic representation of the simulation cell shown in Fig. 2.3.

While SPR regions are three-dimensional, the dynamics themselves are captured sufficiently in a two-dimensional representation if enough simulations are performed. Thus, only the x and y dimensions of the SPR chip are of concern for the model. The laboratory parameters

Table 2.1: The laboratory parameters of a surface plasmon resonance chip.

Parameter	Description	Value	
L_x	Lattice size along x axis	4.80	mm
L_y	Lattice size along y axis	0.0500	mm
v	Mean flow velocity	1.33	mm/s
D	Diffusion coefficient	30.0	$\mu m^2/s$
R_0	Receptor concentration	5000	μm^{-2}
C_0	Ligand concentration	100	nM
k_+	Association rate	—	$M^{-1}s^{-1}$
k_-	Dissociation rate	—	s^{-1}
x_0	Start of SPR scanning region	2.9	mm
x_1	End of SPR scanning region	4.3	mm

are then discretized using the lattice constants detailed in Tab. 2.2, which give the SPR model parameters listed in Tab. 2.3.

Table 2.2: The lattice constants used to discretize the SPR model.

Parameter	Description	Value	
λ	Lattice size constant	10	nm
δt	Time step	1.51×10^{-6}	s

2.3.2 Ligand movement

Surface plasmon resonance cells are small, on the order of millimeters. This results in SPR cells having very small Reynolds numbers [35]. This in turn means that SPR cells reside in the regime of almost ideal laminar flow, so the movement of ligands in our simulation is biased to reflect this laminar transport.

The movement of the ligands through the lattice is modeled via a biased random walk, where the probabilities of moving parallel to the flow velocity are adjusted to create a parabolic flow profile as is expected in the case of laminar flow. The first moment of the ligand position

Table 2.3: The discretized parameters of the surface plasmon resonance model.

Parameter	Relation to lab param.	Value
$\widetilde{L_x}$	L_x/λ	4.80×10^5
$\widetilde{L_y}$	L_y/λ	5×10^3
$\widetilde{v}$	$v \cdot (\delta t/\lambda)$	200.8
$\widetilde{D}$	$D \cdot (\delta t/\lambda^2)$	0.453
$\widetilde{R_0}$	$R_0 \cdot \lambda^2$	0.5
$\widetilde{C_0}$	$C_0 \cdot N_A \cdot \lambda^3$	6.022×10^{-5}
$\widetilde{k_+}$	$k_+ \cdot \delta t/(N_A \cdot \lambda^3)$	—
$\widetilde{k_-}$	$k_- \cdot \delta t$	—
$\widetilde{x_0}$	x_0/λ	2.90×10^5
$\widetilde{x_1}$	x_1/λ	4.30×10^5

is taken from the flow velocity in that direction,

$$p_\mu^+ - p_\mu^- = \widetilde{v}_\mu \,, \tag{2.1}$$

The second cumulant of the ligand position is taken from diffusion in the fluid,

$$(p_\mu^+ + p_\mu^-) - (p_\mu^+ - p_\mu^-)^2 = \widetilde{D_\mu} = \widetilde{D}/3 \,. \tag{2.2}$$

The probabilities of ligand movement can be extracted from these conditions along with a normalization condition:

$$p_0 + \sum_\mu p_\mu^\pm = 1 \,. \tag{2.3}$$

Here p_0 is the probability of staying still, $p_\mu^\pm$ respectively denote the probability of moving in the positive or negative μ direction; v_μ and D_μ are the flow velocity and diffusion constant in the μ direction, where μ can be either x, y, or z. Diffusion in the system is isotropic while

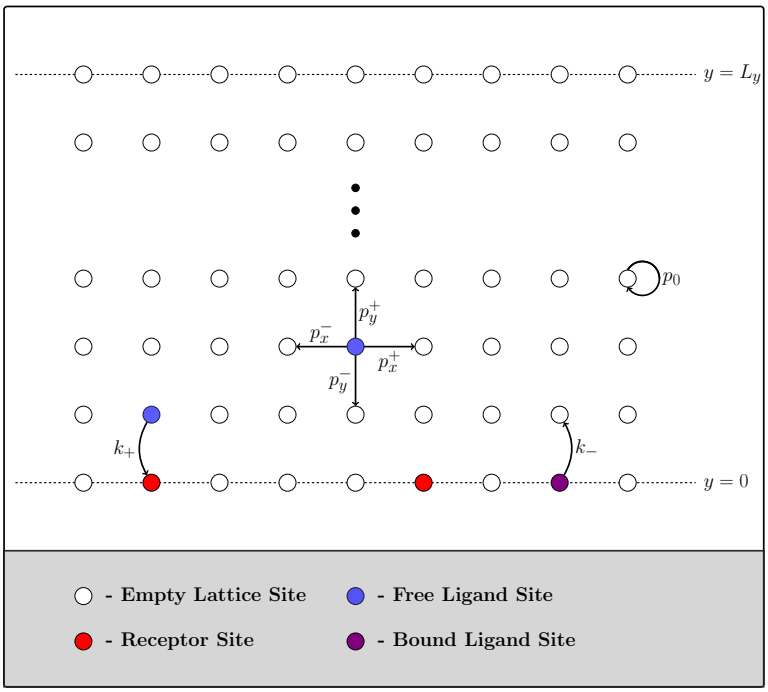


Figure 2.4: The two-dimensional dynamics of the SPR model. This plot shows all the possible actions that a ligand might take as it moves through the lattice sites. A ligand has probabilities $p_\mu^\pm$ of moving forward (+) or backwards (-) in the $\mu \in \{x, y, z\}$ direction, and a probability p_0 of stay put. Additionally, if a ligand is above a receptor it has probability $\widetilde{k_+}$ of binding to the receptor (independent of the probabilities of movement; for the purposes of the simulation the ligands are stepped in a direction determined by the movement probabilities, and then check if they could bind to a receptor). If a ligand is bound, it can no longer move, but has a probability $\widetilde{k_-}$ to unbind. Once unbound, the ligand continues the random walk through the lattice.

the following bias velocities are chosen to model laminar flow:

$$\widetilde{v_y} = \widetilde{v_z} = 0\,, \tag{2.4}$$

$$\widetilde{v_x} = \frac{6\widetilde{v}y(\widetilde{L_y} - y)}{\widetilde{L_y}^2}\,. \tag{2.5}$$

The probabilities of movement perpendicular to the flow velocity are unchanged. The parameters with a '$\sim$' superscript are dimensionless simulation parameters related to the physical parameters of the SPR chip via Tab. 2.3. The dimensional mean flow velocity v is related

to the pressure gradient ΔP across the system as well as the viscosity η [36] via

$$v = -\frac{L_y^2 \Delta P}{12 \eta L_x}\,. \tag{2.6}$$

As the ligands propagate through the lattice and encounter receptors in the receptor surface on the lattice floor, some percentage of the ligand population will bind to the receptors. This percentage is measured every time step for both the association and dissociation stages of the simulation. An example of these results is shown in Fig. 2.6. A brief summary of the algorithm used for the Monte Carlo simulations is given in Appendix 2.C.

2.3.3 Analysis

The system described in Tab. 2.3 was then simulated, with the parameters scaled by a factor of $\alpha = 0.025$ as described in Appendix 2.B. Nine different association rates and two different dissociation rates were selected from the range of known values (detailed in Fig. 2.5, with values ranging from $10^3 M^{-1}s^{-1}$ to $10^7 M^{-1}s^{-1}$ and $10^{-2}s^{-1}$ to $10^{-3}s^{-1}$ respectively).

All possible pairs of these association and dissociation rates where then simulated giving eighteen different simulations. In order to obtain statistically significant results, each of these eighteen simulations was performed five hundred times (each time the simulation is independent of all others), with new random initial conditions for each realization of the simulation. The number of realizations of each simulation was chosen to be five hundred in order to shrink the associated error while still being computationally feasible. Figure 2.6 shows example results of an averaged set of five hundred runs of an association-dissociation rate pair simulation. The example simulation data in Fig. 2.6 displays fits for both the association stage (red circles), and the dissociation stage (blue triangles). The mean-field prediction of the dissociation phase is represented by the (green) dashed line with square

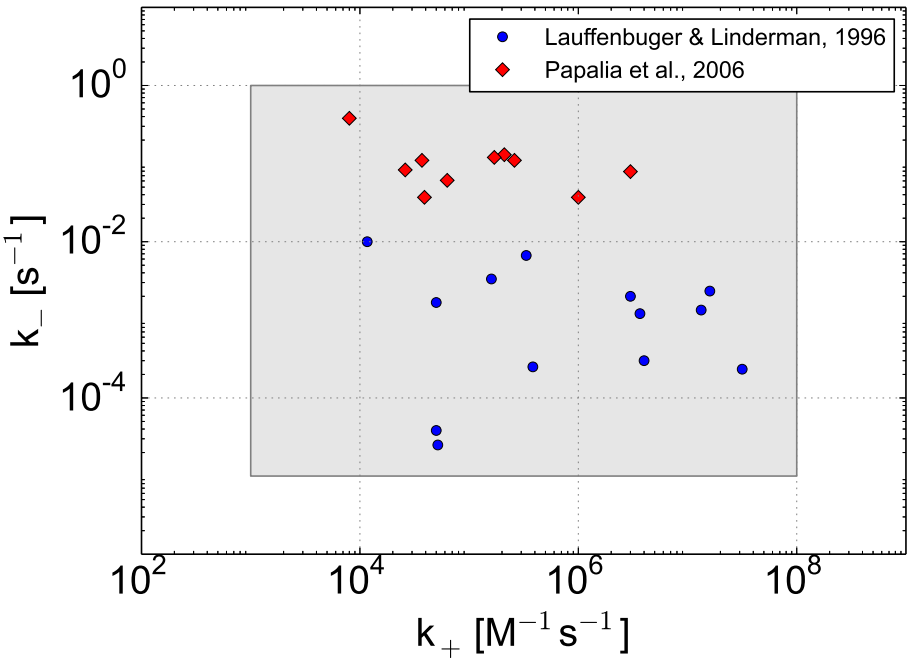


Figure 2.5: The range of experimentally determined reaction rates between pairs of different chemical species. The dissociation rate k_- of the chemical pair is plotted against the association rate k_+ on a log-log plot in order to give a representation of the range of values that these rates can take. The blue circles represent pairs recorded by Papalia et al. [37], while the red diamonds represent pairs recorded by Lauffenburger and Linderman [38]. The shaded region represents the regime of typical association and dissociation rates.

markers. The error bars are not included because they are the same size as the (gray) data points. The inset in Fig. 2.6 highlights the non-exponential behavior of the dissociation phase, by showing a logarithmic plot of the dissociation stage of Fig. 2.6. The (blue) line with triangular markers is the non-exponential fit of the (gray) data points, and the (green) dashed line with square markers is the mean-field prediction. Again, error bars are excluded because they are the same size as the (gray) data points. This plot of a high association rate is chosen to showcase the non-exponential behavior of the dissociation stage at high Da. This behavior does not coincide with the prediction of the mean-field analysis, and will be discussed in Sec. 2.4.

2.3.4 Mean-field approximation

The mean-field rate equation for the SPR system is given by the first-order differential equation for the bound ligand concentration p[2],

$$\dot{p} = C_0 k_+(\gamma - p) - k_- p\,, \tag{2.7}$$

where C_0, k_+, and k_- are described in Tab. 2.1 and $n_l = C_0(x_1 - x_0)L_y L_z$ and $n_r = R_0(x_1 - x_0)L_z$ are the number of ligands and receptors in the SPR scanning region, respectively. The factor γ is the ratio of the number of ligands in the volume of the SPR cell bounded by the receptor surface, to the number of receptors on the receptor surface: $\gamma = n_l/n_r$.

The mean-field association and dissociation rates were extracted via several parameters (summarized in Tab. 2.4) that are easily extracted from the numerical data. These values are often employed in the analysis of sensogram[3] data [39, 40]. The mean-field model, Eq.

[2]In this case p is defined as the number of bound ligand-receptor pairs normalized by the number of ligands in the volume of the SPR cell bounded by the receptor surface. This number of ligands has a value of: $n_l = C_0(x_1 - x_0)L_y L_z$.

[3]A sensogram is a plot of SPR data vs. time. Figure 2.6 is an example sensogram, generated via

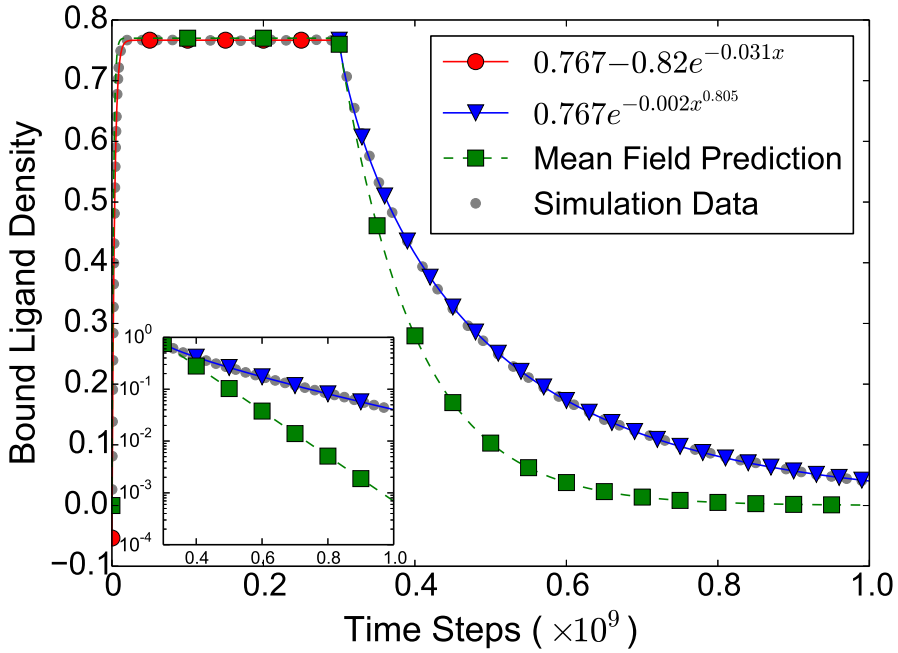


Figure 2.6: An example of simulation data for an association rate of $10^6 M^{-1}s^{-1}$ and a dissociation rate of $10^{-3}s^{-1}$. Error bars are the same size as the data points, and are thus excluded. The simulation results for the density of bound ligands is represented by the (gray) dots. A subset of the simulation data points is shown to ensure that the data points do not overlap and are easily visible. The fit of the association stage of the simulation is represented by the (red) line with circular markers, and the fit of the dissociation stage is represented by the (blue) line with triangular markers. For comparison, the mean-field prediction for a dissociation rate of $10^{-3}s^{-1}$ is shown by the (green) squares. The inset is a logarithmic (base ten) plot of the dissociation data of the main panel, again plotting the bound ligand density versus simulation time steps. The (blue) line marked with triangles is the stretched exponential fit of the data, represented by the (gray) dots, and the mean-field prediction is represent by the (green) squares. This particular rate pair was selected because it demonstrates the non-exponential behavior of the dissociation phase at high Da. This is easily seen in the form of the fit for the dissociation phase, which is a stretched exponential (i.e., $p(t) \propto e^{-\alpha t^\beta}$ for $\alpha, \beta \in \mathbb{R}$) rather than simple exponential (i.e., $p(t) \propto e^{-\alpha t}$ for $\alpha \in \mathbb{R}$). This contradicts the predictions of the mean-field analysis, and will be discussed in more detail in Sec. 2.4.

(2.7), provides predictions for these parameters which are summarized in Eqs. (2.8)-(2.12) below. Specifically, the parameters listed in Tab. 2.4 are: the time derivative $f_0 = \dot{p}(0)$ of the bound ligand concentration at the initial time[4]; f_∞, which is the change in the time derivative $\dot{p}$ with respect to the bound ligand concentration p at the switching time between the association and dissociation stages; the change r_0 in $\ln(p)$ with respect to time at the switching time; the change r_∞ in $\ln(p)$ with respect to time as time goes to infinity; and the saturation concentration p^* of bound ligands as they reach a steady state in the association phase:

$$f_0 = \gamma k_+ C_0 \,, \tag{2.8}$$

$$f_\infty = k_+ C_0 + k_- \,, \tag{2.9}$$

$$p^* = \frac{\gamma k_+ C_0}{k_+ C_0 + k_-} \,, \tag{2.10}$$

$$r_0 = k_- \,, \tag{2.11}$$

$$r_\infty = k_- \,. \tag{2.12}$$

Table 2.4: The sensogram metrics.

Parameter	Definition
f_0	$\dot{p}(0)$
f_∞	$-\lim_{p\to p^*}(\frac{\partial^2}{\partial p \partial t} p)$
r_0	$-\frac{\partial}{\partial t}\ln p(t)\vert_{t=t_{\text{switch}}}$
r_∞	$-\frac{\partial}{\partial t}\ln p(t)\vert_{t=t_\infty}$
p^*	$p(t_{\text{switch}})$

To measure the association and dissociation rates, f_0, f_∞, and r_0 were used. These param-

simulations.

[4]Because the concentration of ligands in the flow cell is not constant at the beginning of the simulation, the time used to calculate this was not $t = 0$, but instead the time when the concentration began to behave like an exponential.

eters were chosen because they are easily extracted from the numerical data, and provide simple relations to the association and dissociation rates. The numerical values of each of the three parameters was taken from the simulation data for each of the rate pairs, and the association rates and dissociation rates were solved for twice, namely via

$$k_+ = \frac{f_0}{\gamma C_0}\,, \tag{2.13}$$

or

$$k_+ = \frac{f_\infty - r_0}{C_0}\,. \tag{2.14}$$

In each case the dissociation rate of the system is

$$k_- = r_0\,. \tag{2.15}$$

The two different association rates k_+ are paired with the one dissociation rate k_-, and compared with the actual input simulation values of these rates.

2.4 Results

The comparison of the simulation rates and the rates extracted from the data by applying the mean-field analysis can be seen in Fig. 2.7. The true simulation rates are denoted by the (blue) circles, the rates extracted using f_0 and r_0, Eqs. (2.13) and (2.15), are denoted by the (green) triangles, and the rates extracted by f_∞ and r_0, Eqs. (2.14) and (2.15), are indicated by the (red) squares. The (gray) dashed lines connect the mean-field rates with the corresponding simulations that they were extracted from. The dotted lines denote different values of constant $Da = k_+ R_0 (L_x L_y / 6vD^2)^{1/3}$. The solid (black) line labeled k_{+max} marks a theoretical maximum that the mean-field theory can predict, which will be discussed below.

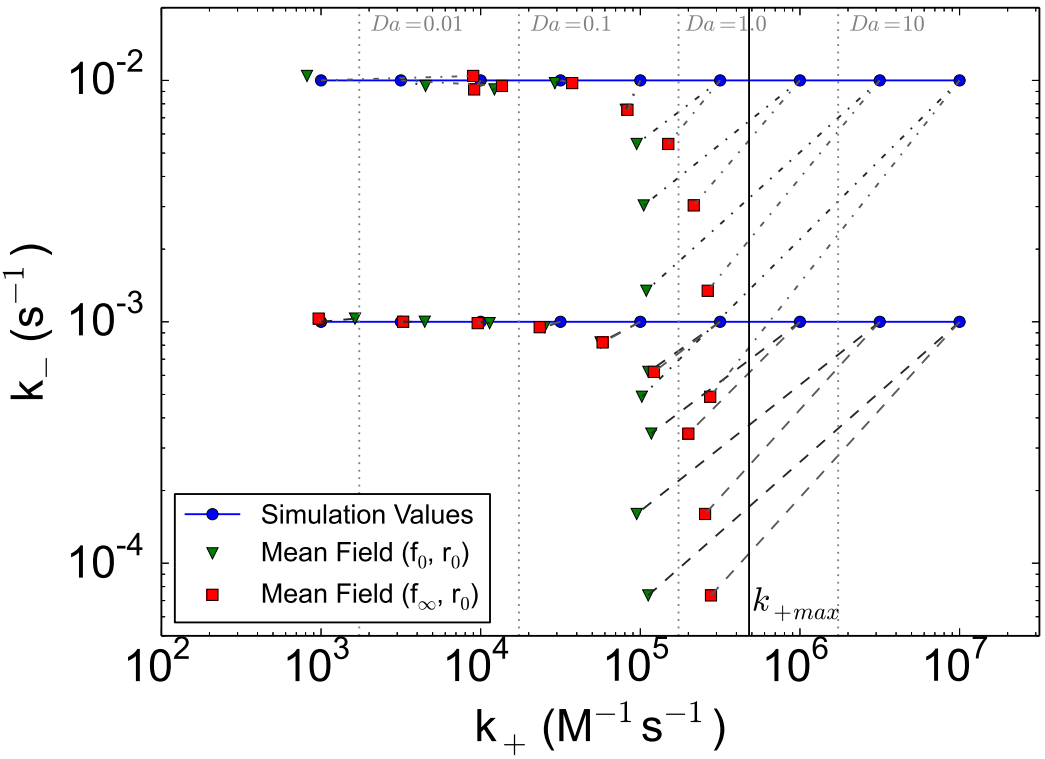


Figure 2.7: The comparison of extracted and simulation association and dissociation rates. The plot shows the dissociation rates k_- plotted against the association rates k_+ on a log-log scale for the eighteen different simulated pairs of association and dissociation rates. The intrinsic simulation rates are denoted by the (blue) circles, the rates extracted using the f_0 and r_0 sensogram metrics, Eqs. (2.13) and (2.15), are denoted by the (green) triangles, and the rates extracted by f_∞ and r_0 sensogram metrics, Eqs. (2.14) and (2.15), are indicated by the (red) squares. The (gray) dashed lines connect the mean-field rates with the corresponding simulations from which they were extracted from. The dotted lines denote different values of constant $Da = k_+ R_0 (L_x L_y / 6vD^2)^{1/3}$. The solid (black) line labeled k_{+max} represents a theoretical maximum that can be extracted from the mean-field theory for this particular system. Note that the highest value of k_+ that can be accurately predicted is much lower, and occurs around $Da \sim 0.1$.

These results were replicated with various values of the lattice spacing constant λ and time step Δt in order to ensure these results are independent of the discretization of the system. The values used in this chapter were chosen because they accurately model the average receptor size and binding timescale of a SPR cell.

It is immediately apparent from Fig. 2.7 that the extracted mean-field rates diverge rapidly from the simulation values as Da increases, though it is interesting to note that the mean-field measurements of k_+ using f_0 and r_0 are better than those using f_∞ and r_0 for $Da < 0.1$ and high k_-, while the predictions of f_∞ and r_0 are slightly more accurate for $Da > 0.1$ than those of f_0 and r_0. The better predictive abilities of (f_0, r_0) at low Da and high k_- are due to the high sensitivity of the association rate k_+ to the sensogram metric f_∞ at low Da and high k_-.

2.4.1 Sensitivity

Sensitivity in this context means the ratio of relative change in the extracted rate to the relative change in the sensogram metrics. To clarify, if $y = f(x)$, then the sensitivity S_y, of y to x is defined by the relation $dy/y = S_y dx/x$. Thus $S_y(x) = (x/f(x))df/dx$. The sensitivity of k_+ to f_0 and f_∞ is given by the equations

$$S_{k_+}(f_0) = 1\,, \tag{2.16}$$

$$S_{k_+}(f_\infty) = \frac{f_\infty}{f_\infty - r_0} = \frac{C_0 k_+ + k_-}{C_0 k_+} = 1 + K\,. \tag{2.17}$$

For extraction of rate constants, the ideal value for sensitivity is 1; sensitivities $\ll 1$ would indicate that the rate constants are independent of the sensogram metrics, while sensitivities $\gg 1$ indicate that small errors in the measurement of sensogram metrics will be amplified into large errors in the interpreted rate constants. The sensitivities are plotted in Fig. 2.8

for the range of k_+ values used in the simulations, as well as both values of k_-. The (green) dashed line is the sensitivity of k_+ to f_∞ with a constant $k_- = 0.01s^{-1}$, the (red) dashed-dotted line is the sensitivity of k_+ to f_∞ with a constant $k_- = 0.001s^{-1}$, and the solid (blue) line is the sensitivity of k_+ to f_0 for all values of k_-. As can be seen, in the regime where k_+ is relatively low and therefore $Da < 1$, k_+ is less sensitive to changes in the sensogram metric f_0 than f_∞. The results extracted from the (f_0, r_0) interpretation therefore predict the rates more accurately in this regime. Additionally, k_+ is approximately an order of magnitude less sensitive to f_∞ for the smaller k_- at low Da, and so the predictions of the (f_∞, r_0) metric at $k_- = 0.001s^{-1}$ are more accurate than those of the same interpretation at $k_- = 0.01s^{-1}$ for low Da.

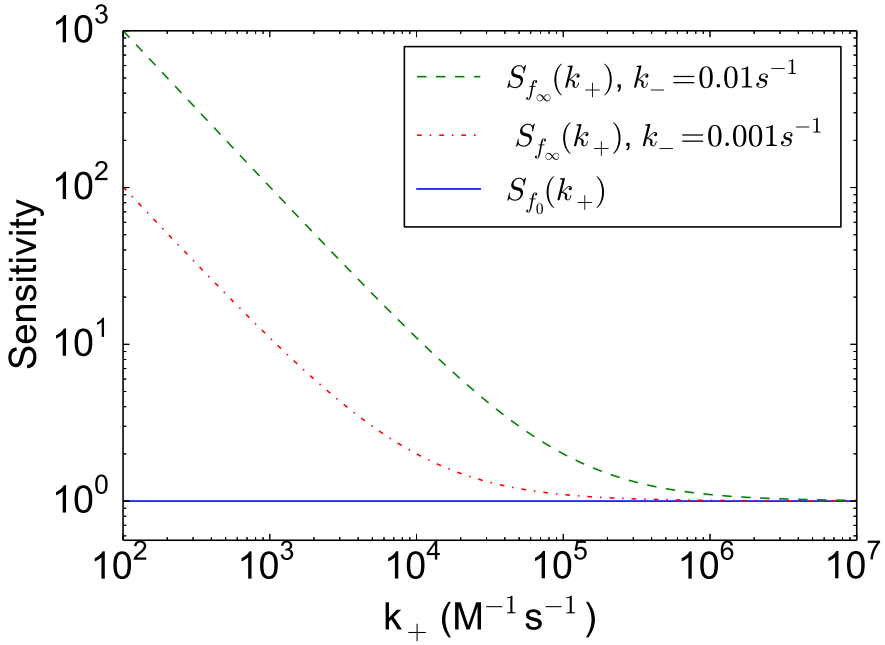


Figure 2.8: A log-log plot of the sensitivity of the attachment rate to the sensogram metrics f_0 and f_∞, Eqs. (2.8) and (2.9), as a function of k_+. The (green) dashed line is the sensitivity of k_+ to f_∞ for $k_- = 0.01s^{-1}$, the (red) dashed-dotted line is the sensitivity of k_+ to f_∞ for $k_- = 0.001s^{-1}$, and the (blue) solid line is the sensitivity of f_0 to k_+ for all values of k_-. The concentration of ligands C_0 was taken to be 100nM.

2.4.2 The diffusion-limited regime

In the regime of high Da, f_∞ becomes the more accurate of the metrics. This (as noted in Ref. [39]) is because f_∞ is less affected by the transport of ligands, since it is extracted from later parts in the experiment, where most of the ligands in the system are near the binding surface. There is still a qualitative increase in the error of the sensogram metrics' predictions as Da increases. One cause of this deviation is the effect of diffusive transport on the ligands. As k_+ increases, the average time for a ligand to bind to a receptor begins to be dominated by the time it takes for a ligand to be transported to the receptor surface [39]; however, at low association rates, $C_0 k_+ < D/(L_y/2)^2$, the time delay an average ligand will experience before binding will be due to the association rate. As the association rate increases into the regime of $C_0 k_+ > D/(L_y/2)^2$, the time delay will not be due to the association rate, but instead will be dominated by the much longer time it takes to be diffusely transported to the receptor.

The mean-field approximation can only interpret the time spent before binding as being due to the association rate, and so the time scale it takes to diffusely transport ligands to the receptor surface gives a theoretical maximum on the association rate that the mean-field theory can predict,

$$k_{+max} \approx \frac{1}{C_0} \frac{D}{(L_y/2)^2} \,. \tag{2.18}$$

This value is marked with a (black) solid line in Fig. 2.7. In this figure the asymptotic approach of the (f_∞, r_0) prediction comes close to this value as Da increases, while the prediction of (f_0, r_0) approaches an asymptote at a lower value because it is more sensitive to the diffusive transport in the system.

2.4.3 Ligand-receptor rebinding events

The remaining effect to mention is that of ligand rebinding, which is assumed not to happen in the mean-field dissociation phase of the SPR experiment. However, the ligands may still perform random walks back to the receptor surface after they have unbound. As the association rate increases, the likelihood of a ligand rebinding to a receptor increases. This causes ligands to on average stay on the receptor surface longer. The mean-field interpretation of this is a lowered dissociation rate, which is why the extracted dissociation rate decreases as the simulation association rate increases.

Additionally, it was predicted by Gopalakrishnan et al. [26] that ligand dissociation from a surface with uniform receptor density R_0 into a semi-infinite domain in the absence of advective transport results in non-exponential late time dissociation of the form $p(t) \propto e^{ct}\text{erfc}(ct)$ where c is a parameter that depends on the density of receptors and the dissociation rate, and $\text{erfc}(z) = 2/\sqrt{\pi}\int_z^\infty e^{-x^2}dx$. As seen in Fig. 2.6, the dynamics of the dissociation phase are indeed non-exponential for high Da, but are stretched exponentials (i.e. $p(t) \propto e^{-\alpha t^\beta}$ for $\alpha, \beta \in \mathbb{R}$) instead of error functions. This difference from the predictions of Ref. [26] is likely due to the presence of advective transport in the SPR cell. For low Da, the behavior of the late-time dissociation corresponds to exponential kinetics, as the effects of the temporal correlations of ligand-receptor rebinding and diffusion are negligible compared to the time it takes for association. This exponential behavior at low Da corresponds to the agreement between the simulation rates and the mean-field predictions at low Da, as seen in Fig. 2.7.

2.5 Conclusion

These Monte Carlo simulations of ligand-receptor binding kinetics in SPR cells provide a testing ground for different analysis techniques. They were used in this chapter to determine

the regime in which a mean-field analysis of SPR is applicable. The system in Tab. 2.1 was modeled using these methods, and the dynamics of many ligand-receptor species with differing association and dissociation rates were simulated. The sensogram metrics defined in Tab. 2.4 were employed to relate the mean-field approximation of the system to parameters easily extracted from the simulation data.

The predictions of the sensogram metric were close to the actual simulation values for $Da < 0.1$, but after that point the association rate begins to get large enough that diffusive transport begins to dominate the time scale on which ligands interact with receptors, and the probability of ligand rebinding events becomes very high. By ignoring these two temporal correlations, the mean-field predictions begin to drastically differ from the simulation parameters, and within a factor ten increase in the association rate, the error between the mean-field predictions and the simulation parameters increased by a factor of one hundred. Thus, these simulations show that a mean-field analysis of surface plasmon resonance is only valid for small values of $Da < 0.1$, due to the importance of the diffusive and ligand-rebinding temporal correlations. Further work could be done on looking at the effects of the ligand-rebinding correlations on different receptor topologies. In biological systems, such as cells, receptors are not evenly distributed like those on the bottom of the SPR flow cell, but appear in clusters on the cell surface. This clustering could increase the likelihood that a ligand rebinding event occurs, allowing ligands to remain on the cell surface longer than would strictly be predicted from their binding rates, c.f. Ref. [31]. This would further distance the dynamics of these biological systems from mean-field predictions.

2.A Reaction-diffusion-advection PDE

This appendix is added to present a model of the SPR system described by Tab. 2.1, and to show that this can be reduced to a system of three dimensionless parameters Da, D_D, K,

and a time scale τ.

The simplification of the advection-diffusion PDE follows from a derivation performed by Ref. [9]. We start with the PDE for ligand concentration in a flow cell with a receptor surface on the $y = 0$ plane,

$$C_t = D(C_{xx} + C_{yy}) - (\frac{6v}{L_y^2})y(L_y - y)C_x \,, \tag{2.19}$$

where subscripts on C denote differentiation with respect to the subscript. Eq. (2.19) can be recast in terms of the scaled variables $\hat{x} = x/L_x$, $\hat{y} = y/L_y$, $\hat{z} = z/L_z$ and $\hat{t} = 6vt/L_x$,

$$C_{\hat{t}} = Pe^{-1}(\varepsilon^2 C_{\hat{x}\hat{x}} + C_{\hat{y}\hat{y}}) - \hat{y}(1 - \hat{y})C_{\hat{x}} \,, \tag{2.20}$$

where $\varepsilon = L_y/L_x$ is a dimensionless parameter, and Pe denotes the Peclét number

$$Pe = \frac{6vL_y^2}{DL_x} \,, \tag{2.21}$$

which represents the ratio of the advective transport rate to the diffusive transport rate. The surface density of bound receptors ($R(\hat{x}, \hat{t})$) evolves according to the reaction rate equation

$$R_{\hat{t}}(\hat{x}, \hat{t}) = k_+ C(\hat{x}, 0, \hat{t})(R_0 - R) - k_- R \,, \tag{2.22}$$

and the boundary condition for the receptor surface is given by

$$C_{\hat{y}}(\hat{x}, 0, \hat{t}) = \frac{Pe}{L_y} R_{\hat{t}}(\hat{x}, \hat{t}) \,. \tag{2.23}$$

SPR systems typically have a Peclét number on the order of 100.

Now we can show that for systems with large Peclét numbers, close to the receptor surface (2.20) simplifies and Pe becomes irrelevant. First we redefine the $\hat{y}$ and $\hat{t}$ variables to a more

useful form:

$$\eta = Pe^{\alpha}\hat{y}\,, \qquad \tau = Pe^{\beta}\hat{t}\,, \tag{2.24}$$

where α and β are quantities that will be determined later. Using these substitutions, Eq. (2.20) becomes

$$C_\tau = Pe^{-(\alpha+\beta)}(\varepsilon^2 C_{\hat{x}\hat{x}} + Pe^{2\alpha}C_{\eta\eta}) - (Pe^{-(\alpha+\beta)}\eta + Pe^{-(2\alpha+\beta)}\eta^2)C_{\hat{x}}\,. \tag{2.25}$$

If we require the Péclet coefficients on $C_{\eta\eta}$ and $\eta C_{\hat{x}}$ to be unity, the exponents α and β must be $\alpha = 1/3$ and $\beta = -1/3$. Eq. (2.25) then reduces to

$$C_\tau = Pe^{-2/3}\varepsilon^2 C_{\hat{x}\hat{x}} + C_{\eta\eta} - (\eta - Pe^{-1/3}\eta^2)C_{\hat{x}}\,. \tag{2.26}$$

Because η is a rescaling of $\hat{y}$, the only part of Eq. (2.26) that determines the binding dynamics is the region where $\eta \to 0$. In this limit (2.26) simplifies to

$$C_\tau = Pe^{-2/3}\varepsilon^2 C_{\hat{x}\hat{x}} + C_{\eta\eta} - \eta C_{\hat{x}}\,. \tag{2.27}$$

Then, in the regime where $Pe^{-2/3}\varepsilon^2$ is small, the ligand concentration is governed by the reduced equation

$$C_\tau = C_{\eta\eta} - \eta C_{\hat{x}}\,. \tag{2.28}$$

Finally, the ligand and receptor concentrations can be rendered dimensionless by the transformation

$$\begin{aligned} c(\hat{x},\eta,\tau) &= C(\hat{x},\eta,\tau)/C_0\,, \\ r(\hat{x},\eta,\tau) &= R(\hat{x},\eta,\tau)/R_0\,. \end{aligned} \tag{2.29}$$

Under this transformation, the boundary conditions on the receptor surface given by Eqs. (2.23) and (2.22) become

$$\begin{aligned} c_\eta(\hat{x},0,\tau) = &\quad D_D^{-1} r_\tau(\hat{x},\tau)\,, \\ r_\tau(\hat{x},\tau) = &\quad Da D_D\{c(\hat{x},0,\tau)(1-r) - Kr\}\,, \end{aligned} \tag{2.30}$$

where Da, D_D, K, and τ are defined in Eqs. (2.31)–(2.34).

2.B Scaling Method for Simulation Parameters

Taking the laboratory parameters from Tab. 2.1 and converting them into simulation parameters as listed in Tab. 2.3 yields values too large to simulate in a reasonable amount of time. Therefore, it is necessary to find a method of scaling that can shrink this dynamical system down to an equivalent simulation cell.

There are four parameters that characterize the system [9]. These are derived in Appendix 2.A, and are summarized below. These are τ, the time scale of the diffusive reactive system:

$$\tau = \left(\frac{6v}{L_x}\right)^{2/3}\left(\frac{D}{L_y^2}\right)^{1/3} t\,. \tag{2.31}$$

The Damköhler number Da is the ratio of the rate of ligand binding action at the receptor surface to the rate of transport to that surface:

$$Da = k_+ R_0 \left(\frac{L_x L_y}{6vD^2}\right)^{1/3}. \tag{2.32}$$

D_D is the ratio at which ligands diffuse across the vertical axis of the lattice, to the rate of transport to the receptors:

$$D_D = \frac{C_0}{R_0}\left(\frac{L_x L_y D}{6v}\right)^{1/3}. \tag{2.33}$$

Finally, K represents the equilibrium dissociation constant for the reaction, normalized by the ligand concentration:

$$K = \frac{k_-}{C_0 k_+}\,. \tag{2.34}$$

Any method of scaling that preserves the dynamics of the system must keep these values

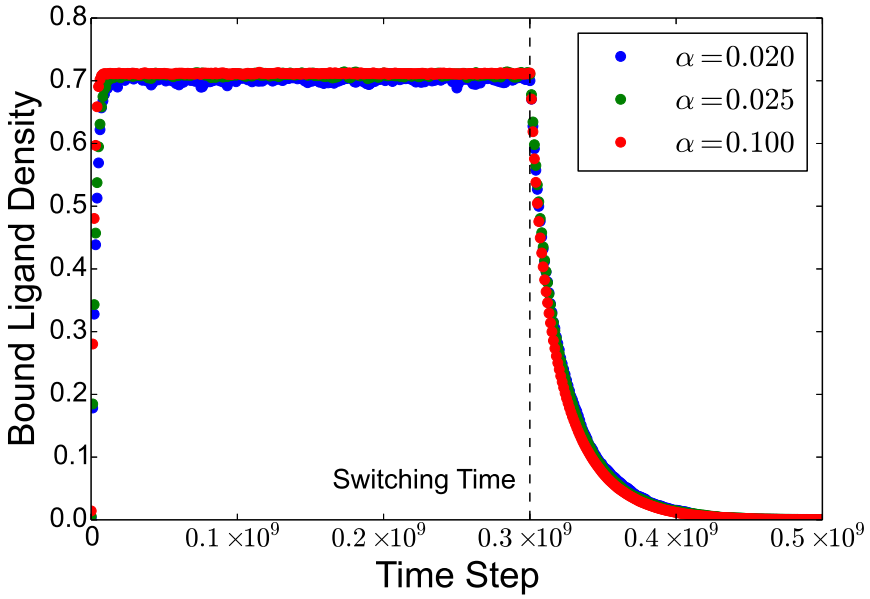


Figure 2.9: An example of scaling for various values of α. The density of bound ligands is plotted against the unscaled Monte Carlo time step for three realizations of the system described in Tab. 2.3 with association rate $k_+ = 10^6 M^{-1}s^{-1}$ and dissociation rate $k_- = 10^{-2}s^{-1}$. The simulations were performed using three different values of the scaling parameter α. Note that changing the values of α by a factor of 5 implies a rescaling of the system length in the x direction and of the overall time scales by a factor of 25. The results of these simulations were unscaled by multiplying by the reciprocal of the scaling factors when needed, and plotted versus the unscaled time steps. The unscaled concentration of ligands is 100nM for each simulation. This concentration is held constant for the duration of the association phase, which lasts until the 0.3×10^9 time step. At this point, marked by the (black) dashed line and labeled as the 'switching time', the concentration of incoming ligands is set to zero, to initiate the dissociation phase. The three data sets are represented by the (blue, red, and green) dots, and as expected, each of the three sets of data coincide.

unchanged. We may hence scale each of the physical parameters in these four values by a

scale parameter α specified such that the values Da, D_D, and K remain fixed:

$$\begin{aligned} L_x &\to \alpha^{\gamma_x} L_x\,, & k_+ &\to \alpha^{\gamma_+} k_+\,, & v &\to \alpha^{\gamma_v} v\,, \\ L_y &\to \alpha^{\gamma_y} L_y\,, & k_- &\to \alpha^{\gamma_-} k_-\,, & D &\to \alpha^{\gamma_D} D\,, \\ C &\to \alpha^{\gamma_C} C\,, & R &\to \alpha^{\gamma_R} R\,, & t &\to \alpha^{\gamma_t} t\,, \end{aligned}$$

where the constant α is a positive real number. We choose the exponents such that

$$\begin{aligned} 0 = &\quad \gamma_t + \frac{1}{3}(\gamma_D + 2\gamma_v - 2\gamma_x - 2\gamma_y)\,, \\ 0 = &\quad \gamma_+ + \gamma_R + \frac{1}{3}(\gamma_x + \gamma_y - \gamma_v - 2\gamma_D)\,, \\ 0 = &\quad \gamma_C - \gamma_R + \frac{1}{3}(\gamma_x + \gamma_y + \gamma_D - \gamma_v)\,, \\ 0 = &\quad \gamma_- - \gamma_C - \gamma_+\,. \end{aligned} \tag{2.35}$$

The above requirements ensure that none of the four parameters are affected by this scaling. At this point any exponents that satisfy the above requirements can be chosen. For simplicity's sake, the exponents of v, D, and R were chosen to be zero. γ_y and γ_x were chosen to be 1 and 2 respectively. This yields the following definitions

$$\begin{aligned} \gamma_x &= 2\,, & \gamma_t &= 2\,, & \gamma_v &= 0\,, \\ \gamma_y &= 1\,, & \gamma_+ &= -1\,, & \gamma_D &= 0\,, \\ \gamma_C &= -1\,, & \gamma_- &= -2\,, & \gamma_R &= 0\,. \end{aligned} \tag{2.36}$$

Fig. 2.9 shows the results of simulations of the system described in Tab. 2.3 with association rate $k_+ = 10^6 M^{-1} s^{-1}$ and dissociation rate $k_- = 10^{-2} s^{-1}$ scaled with various scaling constants α. The range of values of α shown here is actually representative of a whole order of magnitude of values after α has been raised to the appropriate exponents. Note that

the coincidence of the differently scaled simulation results confirms the assertion that the results of scaled simulations of the system described in Tab. 2.3 will accurately represent the dynamics of the unscaled system.

2.C Algorithm for Monte Carlo Simulation

A summary of the algorithm used for the Monte Carlo simulation is as follows.

1) Select a random ligand and generate a random number r uniformly distributed between zero and one.

2) If the ligand is not bound to a receptor:

   a) If $r < p_0$ the ligand remains at the same location.

   b) If instead $r < p_0 + p_x^+$ the ligand is stepped in the positive x direction.

      i) If the ligand encounters the end of the SPR cell ($x = \widetilde{L_x}$), remove the ligand.

      ii) If the simulation is in the association phase, introduce a new ligand at the $x = 0$ plane to maintain ligand concentration.

   c) If instead $r < p_0 + p_x^+ + p_x^-$ the ligand is stepped in the negative x direction.

      i) If the ligand encounters the beginning of the SPR cell ($x = 0$), do not move the ligand.

   d) If instead $r < p_0 + p_x^+ + p_x^- + p_y^+$, step the ligand in the positive y direction. Otherwise if $r < p_0 + p_x^+ + p_x^- + p_y^+ + p_y^-$, step the ligand in the negative y direction.

      i) If the ligand encounters either the top or bottom planes of the SPR cell (i.e. $y = 0$ or $y = \widetilde{L_y}$), reflect the ligand back one lattice spacing into the lattice to ensure reflective boundary conditions.

e) If instead $r < p_0 + p_x^+ + p_x^- + p_y^+ + p_z^+$, step the ligand in the positive z direction. Otherwise if $r < p_0 + p_x^+ + p_x^- + p_y^+ + p_y^- + p_z^+ + p_z^-$, step the ligand in the negative z direction.

i) If the ligand moves past either of the z axis boundaries of the SPR cell (i.e. $z = 0$ or $z = \widetilde{L_z}$), place the ligand on the opposite boundary to create periodic boundary conditions.

f) After the ligand is stepped, if it is one lattice site above an empty receptor, generate a random number q evenly distributed between zero and one.

i) If $q < \widetilde{k_+}$, bind ligand and receptor, and set ligand position to receptor position.

3) If the ligand is bound to a receptor, check if $r < \widetilde{k_-}$. If it is, unbind the ligand.

4) Repeat the above process n times every time step, where $n = \widetilde{C_0} \cdot (\widetilde{L_x} \cdot \widetilde{L_y} \cdot \widetilde{L_z})$ is the number of ligands in the SPR cell.

5) Count the number of bound ligand receptor pairs and divide by the number of ligands in the volume of the SPR cell bounded on the bottom by the receptor surface during the association phase to retrieve the bound ligand density. Record this every time step.

6) After a steady-state concentration of bound ligand-receptor pairs has been reached, change from the association stage to the dissociation stage.

Chapter 3

The effects of inhibitory and excitatory neuron fractions on the dynamics and control of avalanching neural networks

The following chapter is adapted with minor modifications, with the permission from Physical Review E, of our publication:

J. Carroll. A. Warren, and U. C. Täuber. The effects of inhibitory and excitatory neurons on the dynamics and control of avalanching neural networks. Physical Review E, (under review).

This research was sponsored by the Army Research Office and was accomplished under Grant Number W911NF-17-1-0156. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

3.1 Introduction

Neurons are the ubiquitous cells found in all intelligent animals, whose operations and dynamics are responsible for cognition. Individually the dynamics of neurons are well known [5, 6, 42, 43, 44], but neurons are connected dynamically in ensembles of billions, and the collective behavior of these networks of neurons that gives rise to high-level cognitive functions such as memory and consciousness is only partially understood.

A common method for analyzing the collective behavior of biological neural networks is through the study of neural avalanches. An avalanche is a period of continuous neural activity in the network where signals are continually transmitted from neuron to neuron. Distributions of various quantities of these neural avalanches, such as the avalanche strength, duration, and power spectral density, are observed to follow power laws, and the exponents of these power laws appear to govern the govern the proper functioning of the network [45, 46, 47].

In this chapter we discuss a model based on the work of Lombardi, Herrmann, De Arcangelis et al. [12, 13, 14, 15] of an avalanching neural network, that correctly reproduces the power law behavior of the avalanche strength distribution, duration distribution, and power spectral density of neuron activity. We show how these power laws are affected by modifying the fraction of inhibitory neurons, neurons that serve to suppress signals in the network, and observe intriguing extended power law behavior indicative of criticality in the avalanche strength and duration distributions, as well as exponents suggestive of epileptic behavior in the power spectral density at low inhibitory fractions. We also monitor how the outgoing connectivity distribution of the network evolves under the effects of different inhibitory fractions.

Finally, we present two distinct strategies to control and remove the extended tails of the

avalanche strength and duration distributions in networks with low inhibitory fractions through the disabling of either random or highly connected neurons. Removing these extended tails serves to protect these networks from the extreme avalanches that occur in these extended distributions.

3.1.1 Neurons

Biological neural networks are composed of individual neurons connected to each other by synapses. Synapses are small gaps between neighboring neurons where neurons can release and receive neurotransmitters: chemicals that cause the receiving neuron to open or close ion gates and pumps in order to increase or decrease its membrane potential, depending on the neurotransmitter received. Some neurons solely release inhibiting neurotransmitters, and will be referred to as "inhibitory neurons." These neurons serve to suppress signals in brain, and make up 20-30% of the neurons in the human cortex [15, 48]. As a matter of definition, the neuron releasing the neurotransmitters will be referred to as the "pre-synaptic" neuron, and the neuron receiving the neurotransmitters will be referred to as the "post-synaptic" neuron [5]. A schematic of a synapse is shown in Fig. 3.1.

If the membrane potential of the post-synaptic neuron is increased beyond a threshold value then the post-synaptic neuron generates a spiking potential that travels down the length of the axon. Synapses that this signal reaches will release neurotransmitters of their own, to be picked up by other neurons. This signal is referred to as an "action potential," and this process will summarily be called "firing." It is important to note that the connections between neurons are not symmetric. It is not necessarily true that if neuron A can transmit to neuron B, then neuron B can transmit to neuron A [5].

Pre-synaptic neurons that have just fired lock down their ion channels until they can recoup the resources used in generating an action potential and releasing neurotransmitters, for a

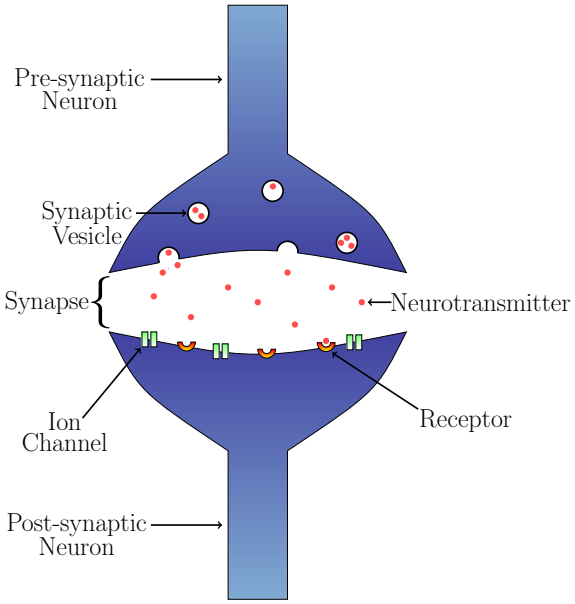


Figure 3.1: A schematic of a synapse. The upper half of the image represents the pre-synaptic neuron, while the lower half represents the post-synaptic neuron. The gap between the pre- and post-synaptic neurons is the synapse. Neurotransmitters (small red circles) are collected in the synaptic vesicles (circular cavities in the pre-synaptic neuron). When the neuron fires, the synaptic vesicles are transported to the surface of the neuron at the synapse and expel their neurotransmitters. The neurotransmitters propagate through the synapse and bind to receptors on the surface of the post-synaptic neuron (red-orange half-circles). These receptors activate ion channels (paired green rectangles) on the post-synaptic neuron which cause the internal potential of the post-synaptic neuron to change as ions are transported into or out of the post-synaptic neuron's cell body [5, 6].

period of 3-4ms [6]. During this period, the neuron cannot respond to any received neurotransmitter, and is for our purposes dormant. This period is referred to as the "refractory period."

The strength of signals transmitted across synapses is determined by the combination of several different factors from both the pre- and post-synaptic neurons, such as the quantity of neurotransmitters produced by the pre-synaptic neuron, and the number of neurotransmitter receptors available on the post-synaptic neuron's surface. The strength of these signals can change over time [42, 43, 44].

Synapses that successfully perpetuate signals, i.e., when the pre-synaptic neuron causes the post-synaptic neuron to fire, increase their efficiency through the increased production of neurotransmitters in the pre-synaptic neuron [42] or an increase in the number of receptors on the post-synaptic neurons [43]. Synapses that do not perpetuate signals have their neurotransmitter production or receptor number reduced [44]. This activity-dependent change in synapse strength is known as Hebbian learning [49], and is a feature the model was devised in Refs. [12, 13, 14, 15] to capture.

3.1.2 Avalanches

In the context of neural networks, avalanches are defined as a period of continuous neural activity. The name is chosen in analogy to Abelian sandpile models, where a column of sand can topple sending sand down onto lower columns causing them to topple and so on, creating a literal avalanche of sand [50]. The same process can happen among neurons where a single neuron fires, causing other neurons to fire, et cetera, until the network has temporarily exhausted its resources. The period of continuous neuron firing is an avalanche.

The dynamics of neural avalanches were studied by Beggs and Plenz in rat cortex cultures,

and various properties of avalanches in these cultures were found to follow power law distributions [45]. The observables investigated included the avalanche strength: the sum of all signals sent by firing neurons; and the avalanche duration: the length of time that the avalanche persists. These results for avalanche strength have since in 2014 been replicated in macaque monkeys [47].

Additionally, power law behavior has been recorded in the power spectral density of neural activity in humans using electroencephalography and electrocorticography [46], and avalanche models have been shown to replicate this behavior, even matching exponents seen experimentally [12].

3.2 Neural network model

Our numerical model is based on the work of Lombardi, Herrmann, De Arcangelis et al. [12, 13, 14, 15], and in addition to modeling key features of biological neural networks; namely firing at a threshold potential, refractory periods, and Hebbian learning [49], the model accurately reproduces several experimentally determined distributions related to avalanches of neural activity in biological neural networks [45, 46].

An extension to this model developed by Lombardi, Herrmann, De Arcangelis et al. which in addition recreates avalanche waiting time distributions is described in Appendix 3.A.

The following simplified model variant however does not attempt to recreate these waiting time distributions, and this extension is excluded to reduce the complexity of the system. Table 3.1 lists the various parameters used in our model.

Table 3.1: The network parameters used in the avalanching neural network model.

Parameter	Description	Value
N	Number of neurons in the network	64000
t	Time step	$10ms$
p_{inh}	Fraction of inhibitory neurons in the network	—
$J_{ij}(t)$	Weight of connection from i^{th} to j^{th} neuron	—
$g_{ij}(t)$	Weight scaled by degrees of connectivity	—
J_{min}	Minimum weight strength	0.001
J_{max}	Maximum weight strength	2
$n_i(t)$	Potential of i^{th} neuron	—
$s_i(t)$	Action potential of i^{th} neuron	—
$n_{\max}$	Threshold potential	-55mV
$k_{in_i}(t)$	Number of incoming connections to i^{th} neuron	—
$k_{out_i}(t)$	Number of outgoing connections from i^{th} neuron	—

3.2.1 Neuron dynamics

The model consists of a number of neurons N, each neuron i defined by its potential n_i. At the beginning of the simulation, each neuron is randomly designated as inhibitory, with probability p_{inh}, or excitatory with probability $1 - p_{inh}$. This will determine whether signals from this neuron increase (excitatory) or decrease (inhibitory) the potentials of other neurons.

The neuron is then randomly assigned an out-degree k_{out_i} from a truncated power law distribution, formed such that $P(k_{out}) \sim k_{out}^{-2}$ for $k_{out} \in [2, 100]$. This range was chosen to mimic the distribution of connectivity experimentally observed in human cortices, which demonstrates power law behavior across two decades of connectivity, following an exponent of -2 [51]. Additionally, the truncated nature of the power law also allows this distribution to be normalized. The k_{out_i} neurons are then chosen from a uniform distribution, and connections between them and the i^{th} neuron are established by assigning each synapse an initial weight J_{ij} uniformly distributed on the interval $(0, 1)$. Connections from a neuron to itself are not allowed in the model. Once the initial network topology is established, the in-degree k_{in_i} is

tabulated for each neuron.

The network is initialized such that the potential of each neuron is set to 90% of the threshold potential $n_{max} \sim -55mV$ to facilitate and accelerate the initial building-up of network activity.

During each time step t, any neuron whose potential has increased past the system's firing threshold, $n_i(t) \geq n_{max}$, fires, sending an action potential $s_i(t)$ to each of the $k_{out_i}(t)$ connected neurons. If the potential of the i^{th} neuron is not above the threshold potential, the action potential is zero:

$$s_i(t) = \begin{cases} 0, & n_i(t) < n_{max} \ , \\ n_i(t), & n_i(t) \geq n_{max} \ . \end{cases} \tag{3.1}$$

After a neuron has fired, its potential is set to zero for one time step, during which it cannot receive signals from other neurons. This mimics the refractory period of real neurons [5]. The signal received by the post-synaptic neurons not in a refractory period is proportional to the action potential of the pre-synaptic neuron. This behavior is summarized in Eq. (3.2),

$$n_j(t+1) = \begin{cases} 0, & s_j(t) > 0 \ , \\ n_j(t) \pm g_{ij}(t) s_i(t), & s_j(t) = 0 \ , \end{cases} \tag{3.2}$$

where the upper and lower signs are for i excitatory and inhibitory, respectively, and $g_{ij}(t)$ controls how much the j^{th} neuron is affected by signals from the i^{th} neuron:

$$g_{ij}(t) = \frac{k_{out_i}(t)}{k_{in_j}(t)} \frac{J_{ij}(t)}{\sum_k J_{ik}(t)} \ . \tag{3.3}$$

While the strength of the synaptic signal is proportional to the pre-synaptic neuron's action potential potential, this signal is scaled by a factor $J_{ij}(t)/\sum_k J_{ik}(t)$ that determines the rel-

ative strength of the connection between the i^{th} and j^{th} neurons compared to all connections from the i^{th} neuron. This is then in turn scaled by the factor $k_{out_i}(t)/k_{in_j}(t)$.

While $J_{ij}(t)/\sum_k J_{ik}(t)$ determines the strength of the $i \to j$ connection relative to all of the i^{th} neuron's connections, the factor $k_{out_i}(t)/k_{in_j}(t)$ rescales each of these connections relative to their importance to the network. This is necessary in order to properly compare the strength of signals from different neurons.

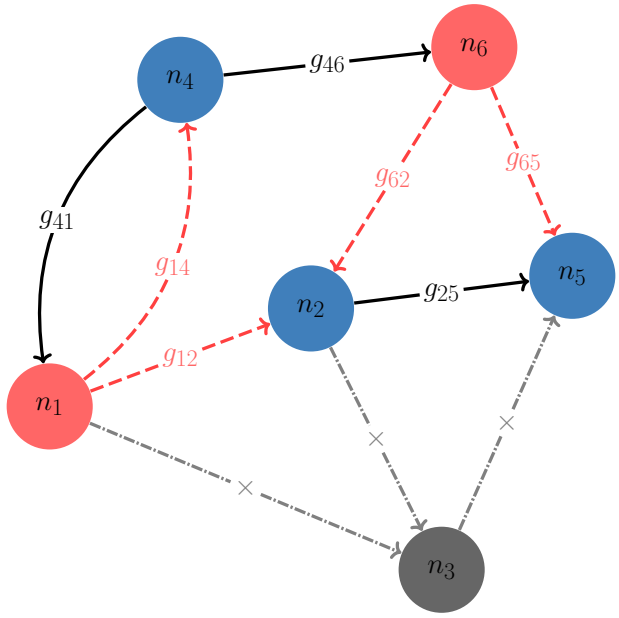


Figure 3.2: An example network of six neurons showcasing the various possible interactions between neurons. The colored circles represent six different neurons, labeled n_i, while the arrows connecting them represent the synaptic connections between neurons with the corresponding elements of the weight matrix g centered in each line. The (blue) neurons n_2, n_4, and n_5 have zero action potential and send no signals through their connections (the black lines) but receive signals from firing neurons. The (red) neurons n_1 and n_6 are firing, sending their non-zero action potentials through their outgoing connections (the red dashed lines). The (grey) neuron n_3 is in a refractory period, and can neither send or receive signals through its synaptic connections (the grey dash-dotted lines).

A neuron with many outgoing connections will be more important to the network than a neuron with few, and so its outgoing signals will be scaled by the neuron's number of outgoing connections, $k_{out_i}(t)$. A neuron that receives signals from many different neurons

will be less excited by any one connection, so any incoming signals to it will be scaled by its number of incoming connections, $k_{in_j}(t)$. Indeed, this factor $k_{out_i}(t)/k_{in_j}(t)$ is responsible for the power law distribution of the avalanche strengths, though it has no bearing on the waiting time or duration distributions. The change in potential in Eq. (3.2) is referred to as the depolarization of j due to i for this time step.

A series of successive time steps during each of which at least one neuron fires constitutes an avalanche.

Figure 3.2 shows a schematic of a simple, six neuron network with examples of different possible connections and neuron behavior.

3.2.2 Hebbian learning and pruning

After each avalanche, the strength of connections between neurons is adjusted according to Hebbian-like rules [49], and then pruned (set to zero) if the strength of connection drops below a threshold J_{min}.

Due to the variable length of each avalanche, it is convenient to index avalanches on a separate variable τ where each avalanche has beginning and ending times $t_i(\tau)$ and $t_f(\tau)$. At the end of each τ^{th} avalanche, we implement Hebbian-like plasticity rules in parallel across the synaptic connections between all neurons. The strength of each synapse J_{ij} is increased proportional to the sum of all signals sent through the synapse during the avalanche, and decreased by the average increase in synaptic strength. We cap each J_{ij} to a maximum value of J_{max} in order to ensure stability in the network. The change in synaptic strength is summarized in Eq. (3.4):

$$J_{ij}(\tau) = J_{ij}(\tau - 1) + \frac{\delta n_{ij}(\tau)}{n_{max}} - \Delta J(\tau) \ , \tag{3.4}$$

where $\delta n_{ij}(\tau)$ is the sum of magnitudes of all signals sent from neuron i to neuron j during the τ^{th} avalanche,

$$\delta n_{ij}(\tau) = \sum_{t=t_i(\tau)}^{t_f(\tau)} \frac{k_{out_i}(t)}{k_{in_j}(t)} \frac{J_{ij}(t)}{\sum_k J_{ik}(t)} s_i(t) \ , \tag{3.5}$$

and $\Delta J(\tau)$ is the average increase in connection strength after the τ^{th} avalanche,

$$\Delta J(\tau) = \frac{1}{N_C(\tau)} \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{\delta n_{ij}(\tau)}{n_{max}} \ . \tag{3.6}$$

Here, $N_C(\tau)$ is the number of non-zero connections in the network,

$$N_C(\tau) = \sum_{i,j} \Theta(J_{ij}(\tau)) \ ,$$

where Θ represents Heaviside's step function. The competition between the second and third terms of Eq. (3.4) causes used connections to increase in strength, while unused connections weaken. If any connection is lowered below a threshold J_{min}, the connection is permanently removed as that element J_{ij} is set to zero. This removal of weak connections is called pruning.

To prevent over-pruning, we impose an upper bound of J_{max} on the strength of any given connection. This strengthening procedure can saturate this bound, but not exceed it. This rule forces the network to prioritize those connections which are most often used, as in biological neural networks [49].

Between avalanches, the system is stimulated via small ($\sim 1\%$ of the threshold potential) constant noise applied to randomly chosen neuron potentials. This noise tends to drive the system towards another avalanche.

3.3 Distributions of avalanche parameters

Several different parameters of neural avalanches can be studied through statistical analysis, and have been shown to be important to the proper operation of biological neural networks [45, 46, 47, 52]. These observables are the avalanche strength, the avalanche duration, and the power spectral density of neuron activity. In this section we describe each of these quantities in turn, and our methods for modeling and recording them.

3.3.1 Avalanche strength distribution

The strength of the τ^{th} avalanche is defined to be the sum of absolute values of all signals sent between neurons during the avalanche.

Biologically, this is the sum of all neuron action potentials. This has been recorded experimentally through the careful placement of electrodes on both *in-vivo* and *in-vitro* neural networks [45, 47]. The strength of many different avalanches can be collected to form a probability distribution describing the likelihood of a given avalanche having a certain strength P_S, where S is the strength of a given avalanche. This distribution P_S has been found to follow a power law of $P_S(S) \sim S^{-1.5}$ [45, 47].

We calculate the total avalanche strength by summing the strength of each signal sent between neurons during an avalanche. If each avalanche has beginning and ending times $t_i(\tau)$ and $t_f(\tau)$ respectively, we define the strength of the avalanche as:

$$S(\tau) = \sum_{t=t_i(\tau)}^{t_f(\tau)} \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{k_{out_i}(t)}{k_{in_j}(t)} \frac{J_{ij}(t)}{\sum_k J_{ik}(t)} s_i(t) \ . \tag{3.7}$$

3.3.2 Avalanche duration distribution

The duration of an avalanche is the length of time that the avalanche persists. This has been recorded experimentally in the same manner as the avalanche strength, and has been collected into a distribution P_D describing the likelihood of a given avalanche duration. Experimentally, this distributions has been observed to follow a power law, $P_D(D) \sim D^{-2.0}$ followed by an exponential cut-off [45].

The duration of the τ^{th} avalanche, $D(\tau)$, is taken to be the number of time steps that the avalanche persists for. This can be written as

$$D(\tau) = t_f(\tau) - t_i(\tau) \ , \tag{3.8}$$

where $t_i(\tau)$ is the initial time step of the τ^{th} avalanche, and $t_f(\tau)$ is the final time step of the τ^{th} avalanche.

3.3.3 Avalanche power spectral density

The power spectral density (PSD) of a signal describes the distribution of power in the signal as a function of frequency. This is a common form of analysis done on the measurements of *in-vivo* neural activity via techniques such as electroencephalography and electrocorticography. Electroencephalography and electrocorticography both measure the action potentials of neurons firing in living brains through the placement of electrodes either outside (electroencephalography) or inside (electrocorticography) the skull. The time series measurement of electrical activity can be decomposed into their power spectral density and has been suggested as a means to diagnose epilepsy [46, 52]. Healthy non-epileptic brains have a PSD exponent in the range of $(-1.5, -0.8)$ [15], while brains undergoing epileptic events have

been recorded with PSD exponents in the range of $(-2.2, -1.8)$ [15, 46].

In our simulations the sum of all depolarizations is calculated after each time step in an avalanche. This sum is then appended to a time series as x_n, the n^{th} sum of depolarizations. We perform this summation for each time step in every avalanche until the series $\{x_n\}$ contains the sum of depolarizations for every time step of every avalanche.

The power spectral density of time series data can be determined by computing this series' discrete Fourier transform. The average of the square of the contribution of each frequency in the discrete Fourier transform gives the power spectral density of that frequency,

$$PSD(f) = \frac{1}{N}\left|\sum_{n=1}^{N} x_n e^{-ifn}\right|^2, \tag{3.9}$$

where PSD is the power spectral density as a function of the frequency f, x_n is the sum of depolarizations in the n^{th} time step, and N is the total number of time steps that these depolarizations were recorded for. We compute the PSD across the entire frequency range using Welch's method [53] and record it in a histogram using logarithmic binning to smooth out fluctuations that occur at the lower frequency ranges due to their limited occurrence in our data. This limited occurrence is due to the power law nature of the model's PSD (see Fig. 3.6).

3.4 Results

For each of the following results, with the exception of the finite-size effects plot shown in Fig. 3.4, we simulated 100 different networks of 64,000 neurons, each randomly initialized according to the methods described in Sec. 3.2.1. Each network was allowed to operate for $10,000 - 100,000$ separate avalanches, after which the various distributions described in Sec. 3.3 were calculated and averaged across the 100 separate networks. The averaged

distributions are recorded below.

3.4.1 Avalanche strength distribution

Using the network described in Sec. 3.2.1 and Eq. (3.7) we measured the distribution histogram of avalanche strengths. Figure 3.3 shows the avalanche strength distributions for two different networks, each made of 64,000 neurons. The two networks differ only in their inhibitory fraction p_{inh}. The (blue) dots represent the avalanche strength distribution of a network with $p_{inh} = 0.04$ and the (orange) triangles represent the avalanche strength distribution of a network with $p_{inh} = 0.10$. The (green) dashed line indicates a power law with exponent -1.55. In both cases the early avalanche strength distributions ($S \in [10^1, 10^5]$) follow a power law of $P_S \sim S^{-1.55}$, which agrees well with the distributions found in rat and macaque monkey cortices [45, 47].

For larger inhibitory fractions, the power law behavior reaches an exponential cut-off at high avalanche strengths as the network is unable to sustain the activity necessary for massive avalanches. Again, as seen in the avalanche duration distributions, as the inhibitory fraction decreases the network becomes better able to sustain increasingly larger avalanches, and at a inhibitory fraction of $p_{inh} = 0.04$ we see the exponential cut-off disappear as the power law behavior is extended for several more decades. This extension of the power law behavior due to these massive avalanches suggests the system is approaching a critical regime as the inhibitory fraction is lowered.

The hump displayed in both sets of data is a finite-size effect related to the total number of neurons in the network. Because the network is stimulated by small constant noise between avalanches, every neuron in the network will, on average, be very close to firing when an avalanche begins. This allows avalanches to initially propagate more easily through the network, until every neuron has fired at least once. After this, this initial "supply" of

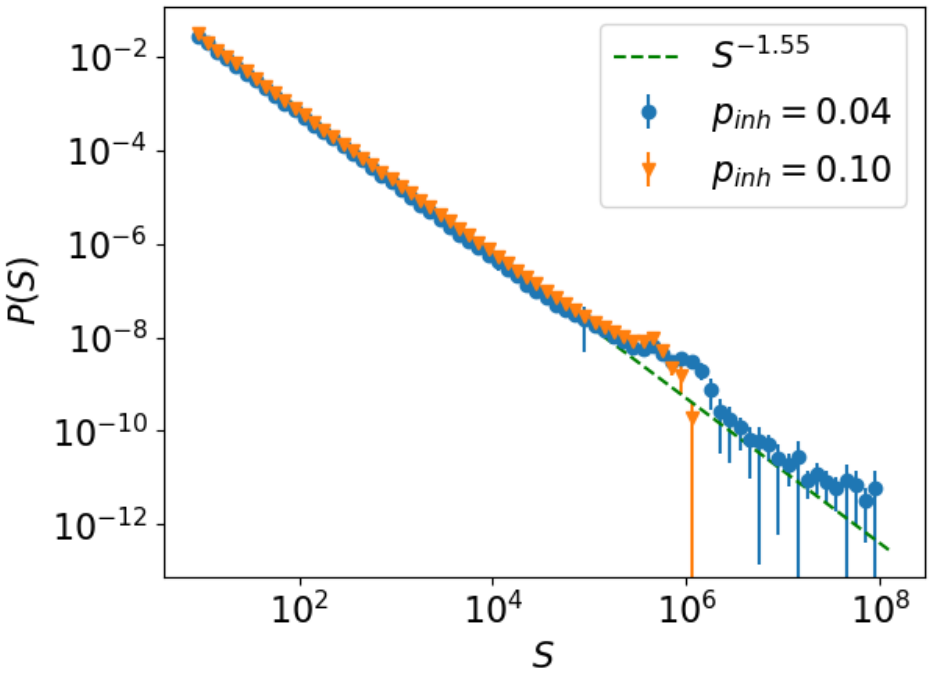


Figure 3.3: Avalanche strength distributions for two 64,000 neuron networks with differing inhibitory fractions p_{inh}. All other network parameters are as described in Sec. 3.2.1. The x axis represents the avalanche strength as defined in Sec. 3.3.1. The y axis represents the probability of an avalanche occurring with that strength. The (blue) dots are the distribution of avalanche strength for a network with $p_{inh} = 0.04$ inhibitory fraction. The (orange) triangles are the distribution of avalanche strength at $p_{inh} = 0.1$. The (green) dashed line indicates a power law with exponent -1.55. Note that for $p_{inh} = 0.04$ the exponential cut-off seen at $S = 5 \cdot 10^5$ in the $p_{inh} = 0.1$ data disappears, and the power law behavior extends to much larger avalanche strengths. Avalanches were recorded with strengths up to $S \sim 10^{10}$, but these were excluded from the figure due to poor statistics. The bump seen at $S \sim 10^6$ in both distributions is a finite-size effect that is proportional to the simulation value of the neuron firing threshold multiplied by the number of neurons in the network. This relation is shown in greater detail in Fig. 3.4.

neuron potential has been exhausted, and avalanches must be self sustaining to continue past this point. Many avalanches are not strong enough to continue propagating without many neurons in the network having highly elevated potentials, and so many avalanches end with this value of strength, creating a hump in the distribution. The dependence on the location of this hump to system size is shown in Fig. 3.4.

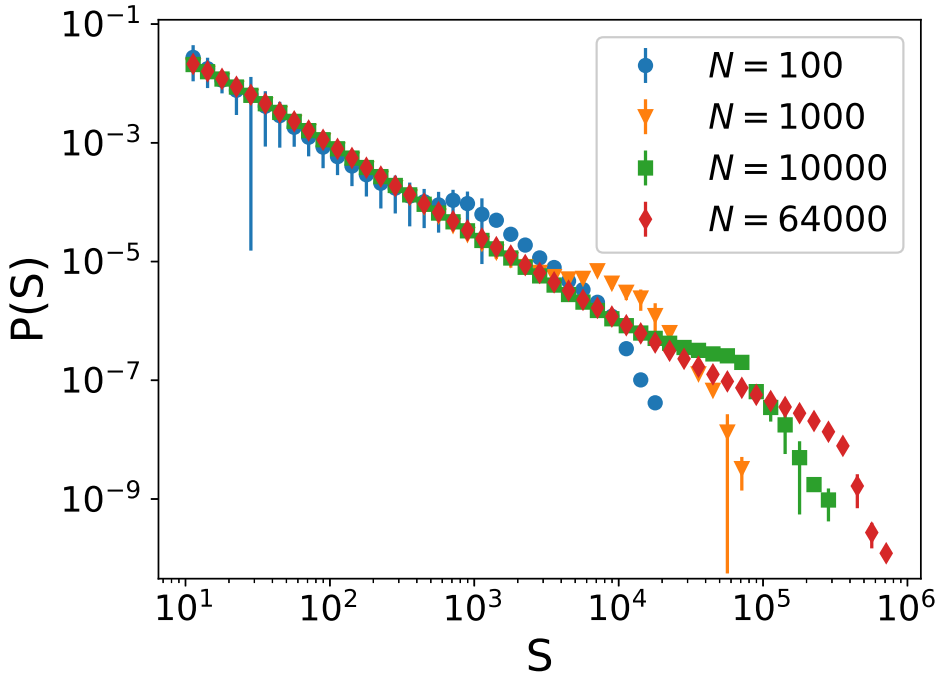


Figure 3.4: The avalanche strength distribution for four different system sizes, each with inhibitory fraction of 0.09. The data for each system size was averaged from fifty different randomly initialized instances of the network. The (blue) circles, (yellow) inverted triangles, (green) squares, (red) diamonds each represent the distribution of avalanche strength for networks with $N = 100$, $N = 1000$, $N = 10000$, and $N = 64000$ neurons respectively. Each distribution follows a power law before reaching a "hump" that is followed by a cut-off. In each case the location of this hump is very close to the system size (number of neurons) multiplied by the simulation value of neuron threshold potential. These humps form because in between avalanches we repeatedly stimulate randomly chosen neurons with a small increase in the neuron's internal potential until a neuron fires. This small stimulation guarantees that on average, when the first neuron in an avalanche fires a neuron fires many neurons will have an internal potential close to the threshold potential. These stimulated internal potentials will help sustain the avalanche until either the avalanche ends through some fluctuation in the system, or until all neurons have on average fired once.

This finite-size effect is also apparent in the avalanche duration distribution, Fig. 3.5.

3.4.2 Avalanche duration distribution

Using the network described in Sec. 3.2.1 and Eq. (3.8) we measured the distribution histogram of avalanche durations. Figure 3.5 shows the avalanche duration distributions for two networks of 64,000 neurons. The (blue) dots represent the distribution of a network with inhibitory fraction $p_{inh} = 0.10$, while the (orange) triangles represent the distribution of a network with inhibitory fraction $p_{inh} = 0.04$. The (green) dashed line indicates a power law with exponent -2.1.

For short avalanche durations both distributions agree well with the experimental results for the avalanche duration distribution which follows a power law with exponent of -2.0 followed by an exponential cut-off [45]. The distribution with higher inhibitory fraction, $p_{inh} = 0.10$, matches the experimental results very well, while the distribution from the network with low inhibitory fraction $p_{inh} = 0.04$ does not show the experimentally found exponential cut-off, and instead displays continued power law behavior at the same exponent for several more decades before statistics of the measured events becomes too poor. This increase in available avalanche durations is due to the inability of the network to suppress signals because there are few inhibitory neurons in the network. This generates very long lasting avalanches that in turn give rise to the extended power law regime seen in the avalanche strength distribution shown in Fig. 3.3. These long-lasting avalanches ultimately dominate the dynamics of the network, because while they are rare, these avalanches can be up to 10^4 time steps longer than avalanches seen in a network with a higher inhibitory fraction.

The inhibitory fraction at which we see these long lasting avalanches occur is much lower than the value observed in human cortices, which is around $0.2 - 0.3$ [15, 48]. Milton et al. [57] suggest that the exponential cut-offs seen in the avalanche distributions exist to protect

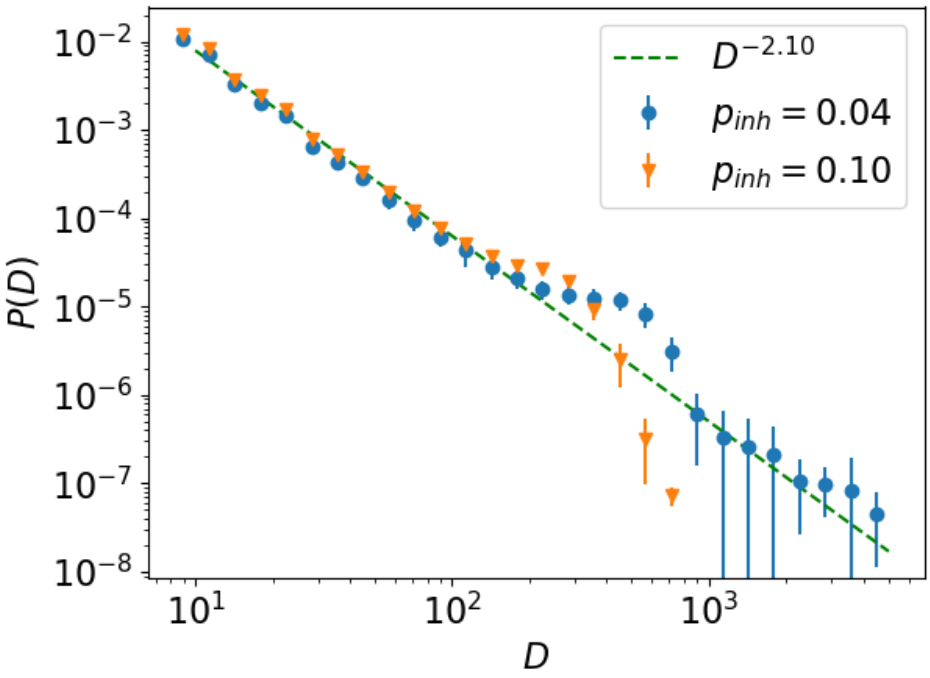


Figure 3.5: Avalanche duration distributions for two 64,000 neuron networks with different inhibitory fractions p_{inh}. The x axis represents the duration D of an avalanche in time steps of $10ms$. The y axis represents the probability of this avalanche duration. The (blue) dots represent the distribution of avalanche durations with $p_{inh} = 0.04$. The (orange) triangles represent the distribution of avalanche durations with $p_{inh} = 0.1$. The (green) dashed line indicates a power law with exponent -2.1. As with the avalanche strength distributions (see Fig. 3.3) the exponential cut-off seen at $D = 3 \cdot 10^2$ time-steps when $p_{inh} = 0.1$ disappears and the power law behavior persists for several decades at the same exponent, before the statistics of the events become to poor. This extended tail for the lower inhibitory fraction ultimately dominates the dynamics of the system, as the avalanches in this regime last for orders of magnitude more time steps than the original regime. This tail continues for several more decades, but these data points have been trimmed from the plot due to poor statistics. The hump seen in both the high and low inhibitory fraction data around $D \sim 5 \cdot 10^2$ is a finite-size effect related to the duration necessary for all neurons in the network to fire, on average, once.

the brain from runaway avalanches; our results corroborate this idea that the brain might operate away from this regime in order to not be dominated by the incredibly long lasting avalanches present at low inhibitory fractions, so the brain actually ultimately avoids truly critical behavior.

Additionally, both the avalanche strength and duration distributions were found to be very stable with respect to the noise strength, minimum and maximum weight strengths, as well as the threshold potential, with changes in these parameters resulting in little to no modifications in their dynamics.

3.4.3 Power spectral density

Using Eq. (3.9) we constructed the power spectral density of our networks of 64,000 neurons for different inhibitory fractions. Figure 3.6 shows two different network power spectral densities for inhibitory fractions 0.04 and 0.30.

The PSD data for the $p_{inh} = 0.04$ network is represented by the (blue) circles, while the data for the $p_{inh} = 0.30$ network is represented by the (red) triangles. Both PSDs displayed in Fig. 3.6 show two distinct dynamical regimes: a power law regime at mid to low frequencies, and a semi-constant regime at high frequencies. The PSD's power law regime is due to long-range temporal correlations present between neuron firings. It is an assumption of this model that neurons firings are uncorrelated across different avalanches, so this power law regime becomes more pronounced as the inhibitory fraction decreases, due to the increasing accessibility of long duration avalanches. This behavior corroborates results shown by Lombardi et al. [15]. We also observe the range of power law behavior to shrink as the inhibitory fraction is increased. While the network with a lower inhibitory fraction shows a very clear power law following an exponent around -2.04, (shown in Fig. 3.6 as the blue dashed line), the higher inhibitory network seems to at most follow a power law with exponent around -1.0

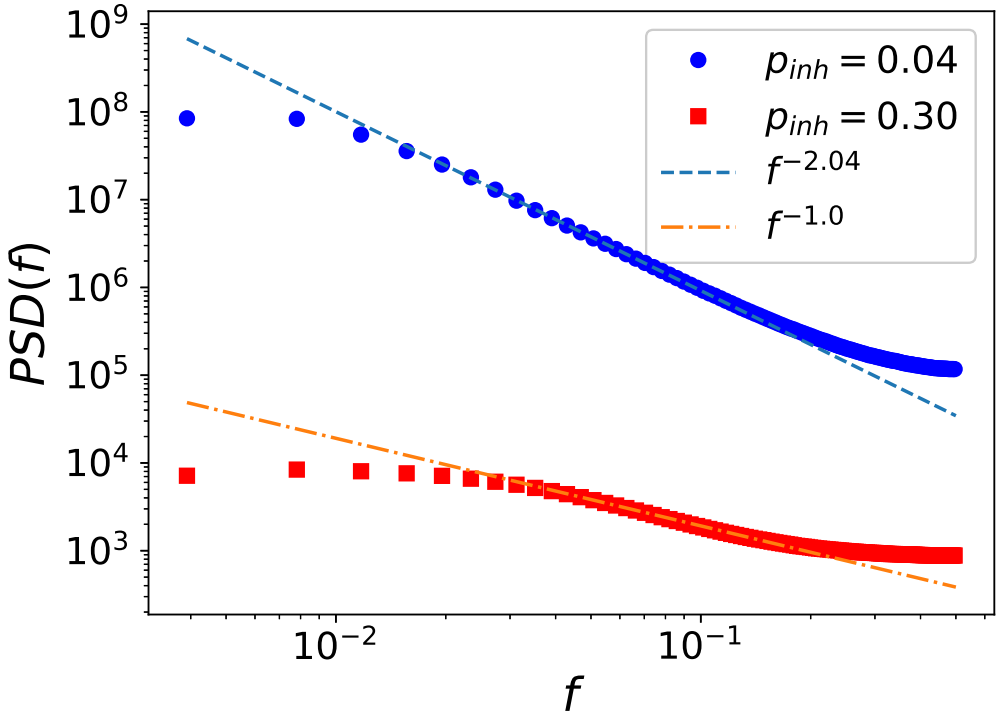
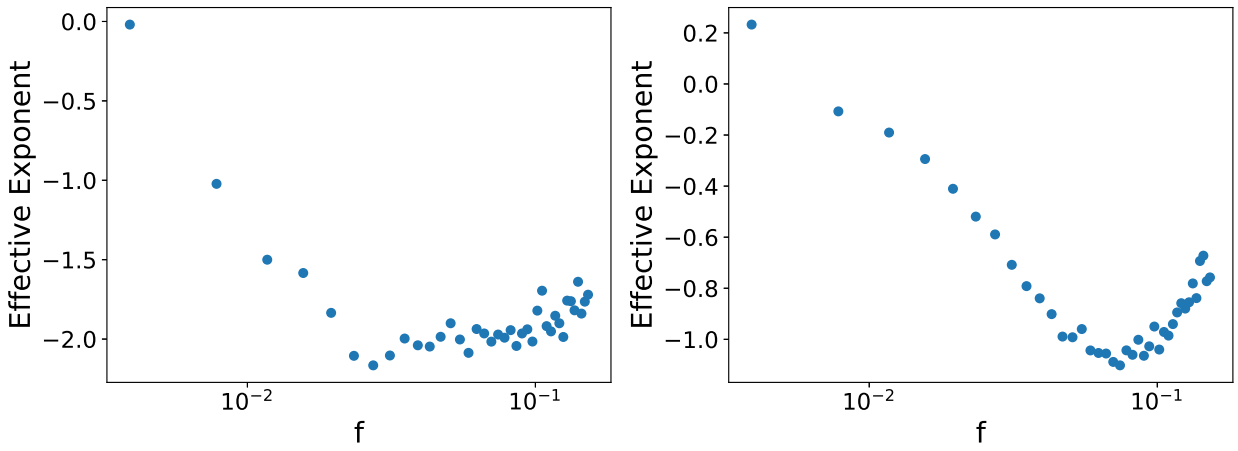


Figure 3.6: Two power spectral densities of 64,000 neuron networks with different inhibitory fractions p_{inh}. The (blue) dots represent the power spectral density of the network with $p_{inh} = 0.04$. The (red) squares represent the power spectral density of the network with $p_{inh} = 0.30$. The (blue) dashed line indicates a power law with exponent -2.04, an exponent in the regime of epileptic behavior seen in humans [15, 46]. The (orange) dash-dotted line represents a power law with exponent -1.0, which is in the range of exponents observed in "healthy" human brains [15, 46]. While there appears to be a well defined power law for the $p_{inh} = 0.04$ data, the power law regime shrinks as we raise the inhibitory fraction to $p_{inh} = 0.30$, and we at most observe a brief period of power law behavior around $f \sim 10^{-1}$ where there is approximately a power law with an exponent close to -1. To more effectively interpret this data, we calculated the local derivative between each consecutive data points and plotted the effective exponents of each of the two PSDs in Fig. 3.7(a) and 3.7(b).



(a) (b)

Figure 3.7: The log-log plot of the local derivative between consecutive pairs of points for the PSD of the (a) $p_{inh} = 0.04$ network and (b) $p_{inh} = 0.30$ network shown above in Fig. 3.6. The y-axes display the effective exponent of any power law behavior observed across the two data sets, while the x-axes represent the average frequency between the pairs of PSD data points that were used to calculate each local derivative. True power law behavior in the PSDs will then be displayed on this graph as a plateau in the effective exponent across many different frequencies. This behavior is clearly visible in (a), where the effective exponent converges to a value around ~ -2 for much of the frequency range between 10^{-2} and 10^{-1}. In contrast we see at most very short-lived power law behavior in (b). The effective exponent displays a short-lived plateau around an exponent of approximately ~ -1, but this behavior ends very quickly.

(represented in Fig. 3.6 by the orange dash-dotted line) for only a small portion of its frequency range, if at all.

This decrease in the power law regime is most likely due to the suppression of highly correlated neuron firing events by the increased fraction of inhibitory neurons. To investigate the true extent of the power law regime we extract the effective exponents of each of the PSDs shown in Fig. 3.6 and plot the effective exponents as a function of frequency in Figs. 3.7(a) and 3.7(b).

The effective exponents shown in Fig. 3.7(a) showcase very clear power law behavior for the low inhibitory network with exponent around -2.0. This value of the exponent is in the

regime of experimentally observed epileptic behavior seen in humans [15, 46] and we observe this behavior for the same values of inhibitory fractions were we observed the extended avalanche strength and duration distributions.

Figure 3.7(b) displays no true power law behavior, with only a small possible plateau of the effective exponent around -1.0. While this value is in the regime of normal operating brain behavior for human PSDs [15, 46], and a transition of exponents from a value of -1.0 to -2.0 as the inhibitory fraction of these networks is lowered was reported by Lombardi et al. [15], we cannot claim to see true power law behavior at biologically relevant inhibitory fractions of $p_{inh} = 0.30$.

3.4.4 Neuron connectivity distribution

In addition to the various avalanche distributions, we can also observe how the connectivity of the network evolves over time for different inhibitory fractions. Figure 3.8 shows an example of the initial distribution of outgoing connectivity for our networks. The horizontal axis represents the degree of outgoing connections k_{out} for individual neurons, while the vertical axis represents the number of neurons measured with a particular value of k_{out}. The (blue) circles represent distribution of connectivity that was measured from our network upon its creation. The (orange) line is the best fit of a power law to the measured distribution, which follows a power law with exponent very close to -2.0.

The data very clearly displays the behavior described in Sec. 3.2, which requires that the initial degrees of outgoing connectivity be drawn from a truncated power law distribution with exponent -2.0, constrained such that $k_{out} \in [2, 100]$. While the following plots refer to specific inhibitory fractions, Fig. 3.8 serves as an example for all inhibitory fractions, as all networks are initialized in the same manner.

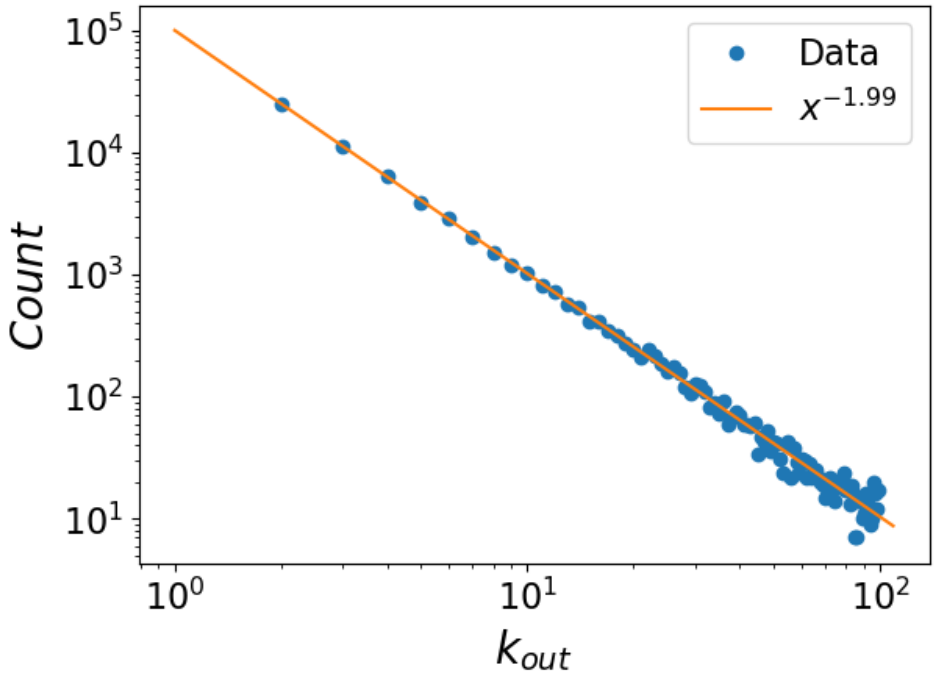
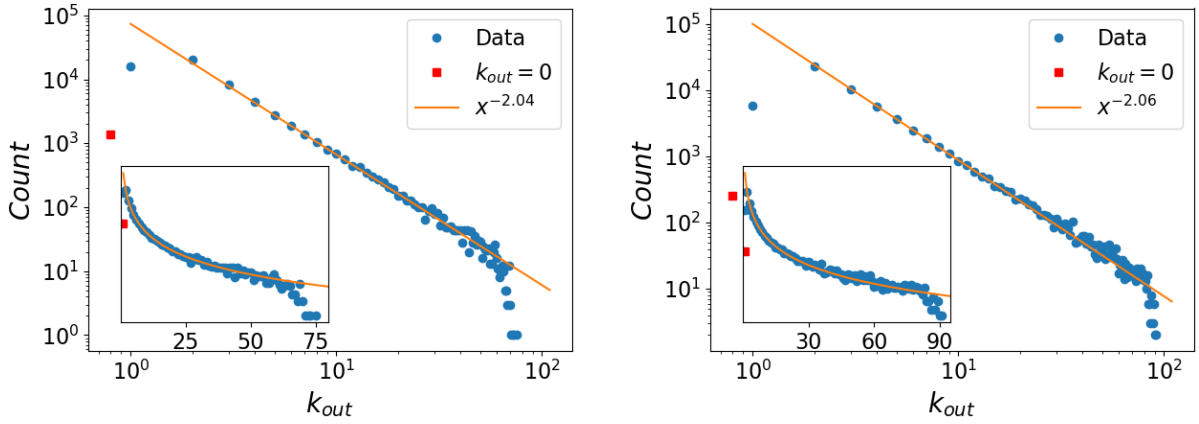


Figure 3.8: An example of the initial k_{out} degree distribution for our simulations. The horizontal axis represents the number of outgoing connections k_{out} a neuron has, while the vertical axis represents the number of neurons in the network with a particular k_{out}. The (blue) circles represent measured degrees of connectivity for all the neurons in a network upon initialization, while the (orange) line displays the best fit of a power law to the data. As described in Sec. 3.2, the outgoing degrees of connectivity k_{out} are chosen from a power law distribution $P(k_{out}) \sim k_{out}^{-2}$, truncated such that $k_{out} \in [2, 100]$.

As avalanches occur in the network, the Hebbian rules defined in Sec. 3.2.2 will cause the connections between neurons to change according to their use. Frequently used connection will be strengthened, and infrequently employed connections will be weakened or even "pruned" (i.e., removed) if the connections become too small. We are interested in observing how the distribution of outgoing connectivity evolves as function of time and inhibitory fraction under these effects.

Figure 3.9 shows the distribution of outgoing degrees of connectivity for two networks with (a) $p_{inh} = 0.30$, and (b) $p_{inh} = 0.04$ after $45,000$ avalanches. The horizontal axis again represents the degrees of outgoing connectivity k_{out} of individual neurons, while the vertical axis displays the number of observed neurons with a particular k_{out}. The (blue) circles represent the number of neurons measured with all degrees of connectivity except zero, and the (red) square represents the measured number of neurons with zero outgoing connections. The (red) square is placed with some positive offset on the horizontal axis in order to display this data on a double-logarithmic plot. The (orange) line represents the best fit of a power law to the data, which follows an exponent very close to -2.0 in both situations. The inset is a log-linear graph of the same data, replotted to highlight the maximum degree of connectivity in the network, $k_{out} = 75$ for $p_{inh} = 0.04$ (a), and $k_{out} = 91$ for $p_{inh} = 0.04$ (b). While the power law behavior of the majority of the distribution is unchanged, there are striking differences in the head and tail of the distribution after the Hebbian rules have been applied for $45,000$ avalanches. Many of the high degrees of connectivity have been pruned out of the distribution, and many more neurons retain either only one or zero outgoing connections; additionally, no neurons have more than 75 and 91 outgoing connections, respectively, for these two inhibitory fractions. The capability of the model to prune away all connections of a neuron is a noticeable departure from biological relevancy, and we find that the Hebbian rules detailed in Sec. 3.2.2 have a tendency to "over-prune" due to the inability for new

connections to be formed, or for old zero-strength "dead" connections to be reestablished. Future iterations of this model could be improved by allowing new or dead connections to become established between neurons through some probabilistic rate during the network updates that take place after each avalanche.



(a) (b)

Figure 3.9: The k_{out} degree distribution for networks with an inhibitory fraction of (a) $p_{inh} = 0.30$ and (b) $p_{inh} = 0.04$ after 45,000 avalanches. The horizontal axis (k_{out}) represents the number of outgoing connections individual neurons have. The vertical axis represents the number of neurons that have a particular value of k_{out}. The (blue) circles are the data points we measured and averaged from 100 networks with (a) $p_{inh} = 0.30$, and (b) $p_{inh} = 0.04$ after $45,000$ avalanches, and represent the number of neurons with each degree of connectivity except zero. The (red) squares represents the data points for the number of neurons with zero outgoing connections. These data points are plotted with some offset along the horizontal axis in order to display it on a double-logarithmic plot. The (orange) line is the best fit to the power law regime of the distribution, with exponents -2.04 (a) and -2.06 (b). The inset represents a log-linear graph of the same data, replotted to highlight the maximum value of k_{out} for this network, which is 75 (a) and 91 (b), respectively.

Even with the general decrease in connectivity, the lower inhibitory fraction network is inherently more strongly connected than the higher inhibitory fraction network after a long simulation time allowing for $45,000$ avalanches. The neural network with smaller p_{inh} displays a higher maximum value of k_{out}, and the number of neurons with only one or zero outgoing connections is reduced by almost an order of magnitude as compared with the

network with larger p_{inh}.

This difference in connectivity evolution results from the distinct inhibitory fractions of the networks as follows: A stronger inhibitory network will result in weaker signals being transmitted through it. Weaker signals in turn cause weaker connections between neurons, and hence an increase in the number of neuron connections pruned due to the Hebbian rules that govern the system. In comparison, the network with a lower inhibitory fraction cannot suppress the signals in the network as strongly as the network with a higher p_{inh}. The network with a lower inhibitory fraction will sustain stronger connections and will consequently not prune away neuron links as drastically. In general we observe that the pruning mechanism preserves the power law structure of the initial distribution, and the effects of the pruning are only visible in the very head and tails of the distribution.

3.4.5 Control of avalanche distributions

Figure 3.3 shows how the avalanche strength distribution changes as the inhibitory fraction of the network is varied. The plot shows two regimes of activity: At higher inhibitory fractions the distribution follows a power law behavior terminating in an exponential cut-off; at very low inhibitory fractions the algebraic decay extends further and the exponential cut-off is shifted to very high avalanche strengths. These extended power laws dominate the dynamics of the networks they occur in. In the following subsection we draw inspiration from work previously done on the robustness of scale-free networks [54, 55] and propose two different control strategies to remove these extended power law tails in networks with low inhibitory fractions through the disabling of either (1) randomly picked or (2) specifically selected highly connected excitatory neurons.

Control through disabling random excitatory neurons

The first strategy we implemented is disabling randomly chosen excitatory neurons. Disabling a neuron means that the internal potential of the neuron is permanently set to zero from the time step it is disabled. We choose excitatory neurons only because the goal is to prevent the large avalanches occurring in the extended tail of this network's normal avalanche strength distribution. An active inhibitory neuron is more effective at stopping these avalanches than a disabled neuron, so we only pick excitatory neurons. The neurons are selected with equal probability from all excitatory neurons. We tested several different fractions of excitatory neurons to disable randomly, and only observed the extended tail present in the avalanche strength distribution to disappear for disabling fractions greater or equal to 0.30.

Figure 3.10(a) shows the avalanche strength distribution of a network with inhibitory fraction of $p_{inh} = 0.04$ after 30% of the excitatory neurons were randomly selected and disabled, i.e. $n_i(t)$ was held at zero for all disabled neurons in the subsequent evolution. The network was allowed to evolve unperturbed for $60,000$ avalanches before the excitatory neurons were disabled. The data shown was averaged over 100 realizations of the network. The extended tail seen in avalanche strength distributions with this inhibitory neuron fraction has disappeared, due to the fragmentation of the network caused by disabling so many neurons. The network is no longer able to sustain the large network-wide avalanches necessary for the extended power law tail in the avalanche strength distribution. Disabling fractions of random excitatory neurons less than 30% does not remove the extended tail. The system is thus quite robust against random disablings, which is not surprising given its scale-free structure. Scale-free networks are known to be quite stable against the random removal of nodes, which is analogous to our disabling of neurons. In order to prevent a signal from being propagated across a generic scale-free network of the same size as our network, more

than 90% of the nodes must be randomly removed [54, 55]. We only need to disable a much lower fraction of random neurons to see the extended power law disappear because we are not attempting to disrupt the entire network activity, but only curtail the power law regime of these extended avalanches.

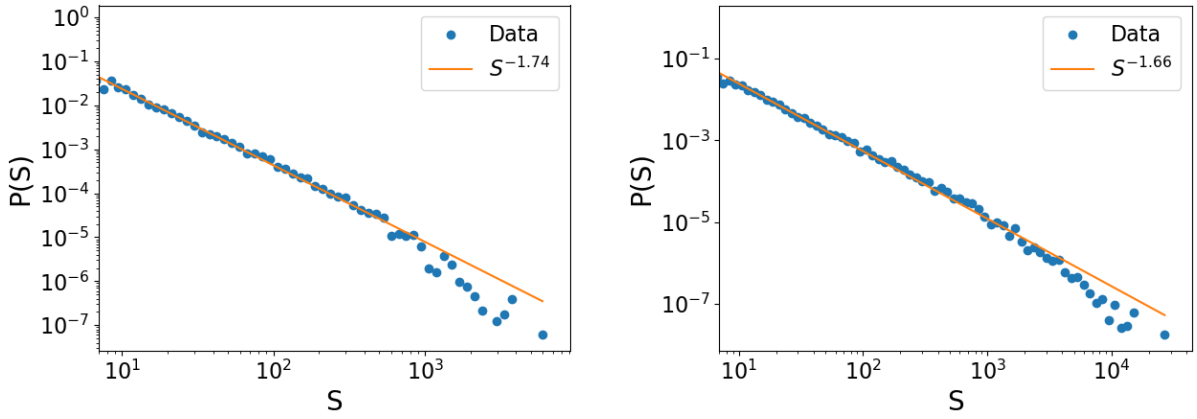
However, even with the network's robustness, disabling this many neurons from the network is destructive to the normal dynamics of the network, and we observe a change in exponent of the power law behavior of the distribution from -1.5 to -1.74. This lower exponent results in a much lower probability of strong avalanches, and we see the cut-off appear several decades below the threshold in networks with higher inhibitory fractions.

Disabling random neurons consequently is an ultimately successful strategy for removing the long power law tail in this avalanche distribution; however, it requires a significant portion of the network's neurons be disabled, which is certainly not ideal, and very likely quite detrimental in any biological neural network.

Control through disabling highly connected neurons

The second strategy we implemented is the disabling of the most highly connected excitatory neurons. The neurons were chosen based on their degree of outgoing connectivity k_{out}. We tested several different fractions of the most connected excitatory neurons, and observed that only the top 1% of highly connected neurons need to be disabled in order to stop these incredibly large avalanches.

Figure 3.10(b) depicts the avalanche strength of a network with inhibitory fraction of $p_{inh} = 0.04$ with the top 1% of the most highly connected neurons disabled. The network was again allowed to evolve for $60,000$ avalanches before the highly connected neurons were disabled. The (blue) circles are the averaged data points of the avalanche strength distributions from 100 different network realizations. The (orange) line is the best fit of a power law to the



(a) (b)

Figure 3.10: The avalanche strength distribution of a network with an inhibitory fraction of $p_{inh} = 0.04$ after (a) randomly selected 30% of the excitatory neurons, and (b) the top 1% of highly connected excitatory neurons have been disabled. In either case, the network was allowed to evolve naturally for the duration of $60,000$ avalanches before the excitatory neurons were disabled. The (blue) circles represent the probability of an avalanche having a given avalanche strength S, and the (orange) line is the best fit of a power law to the data. The extended tail present in a normal $p_{inh} = 0.04$ network has disappeared after disabling (a) a large fraction, (b) a minor fraction of excitatory neurons. For case (a), disabling 30% of inhibitory of neurons is very destructive to the dynamics of the network, and a marked change in the power law behavior of the avalanche strength distribution is observed: This network follows a power law with exponent ~ -1.74, lower than the normal exponent of -1.5. This is the result of signals dying out more quickly in this heavily diluted network. Disabling random fractions less than 0.30 of the excitatory neurons retained the power law tails typically seen in the avalanche strength distributions of these low inhibitory fraction networks; indeed, 0.30 is the lowest fraction of random excitatory neurons that must be disabled to curtail these extended power law tails. For (b), the extended tail present in a normal $p_{inh} = 0.04$ network has disappeared after disabling only a minor fraction of the excitatory neurons. Due to the power law structure of the network connectivity, these top 1% of the excitatory neurons are much more important to the network dynamics than the vast majority of all other neurons. Disabling these neurons is still destructive to the network dynamics, though less so than the random disabling case shown in (a), and a change in the power law behavior of the network is observed. This network follows a power law with exponent ~ -1.66, again lower than the normal exponent of -1.5. This is the result of signals being unable to propagate through the most heavily connected neurons, which play a large role in the network's transmission capability. 0.01 is the required lowest fraction of highly connected excitatory neurons that needs to be disabled in order to effectively remove these extended power law tails. Data averaged over 100 independent network realizations.

data. This model is very sensitive to disabling highly connected neurons, which is also to be expected given its scale-free structure. In addition to being very robust against random disablings, scale-free networks are highly susceptible to "targeted" disablings, where the most highly connected nodes are removed [54, 55]. A generic scale-free network of the same size and exponent of the connectivity distribution needs approximately the top 3% of highly connected nodes disabled to completely fragment the network and destroy any long-range connectivity [54, 55, 56]. In our system, we need only disable the top 1% of the highly connected neurons because we do not aim to completely destroy the network dynamics, yet merely wish prevent the occurrence of exceedingly strong avalanches.

Permanently disabling these highly connected neurons does still considerably affect the network dynamics. The avalanche strength distribution follows a power law with exponent -1.66 instead of the typical -1.5. This is due to the signals being unable to propagate as strongly through the network after the highly connected neurons have been disabled. These weaker avalanches are more likely to die out earlier than their counterparts in an unsuppressed network, resulting in a lower exponent and an earlier cut-off in the avalanche strength distribution.

Disabling only the highly connected neurons is hence demonstrably a very successful control strategy for removing these extended power law tails that dominate the network. However, this approach does require significant knowledge about the structure of the network and is still destructive to the network dynamics because, by definition, these highly connected neurons are very important to signal propagation through the system.

The disabled fraction of connections

We can also compare the fraction of connections between neurons that are effectively removed from our networks due to each of these two strategies, in addition to the fraction of neurons.

We observed the process of randomly selecting a certain fraction of neurons in our networks to select the same fraction of the connections. Thus when 30% of the excitatory neurons are disabled we also disable approximately 30% of the network's connections. In contrast, when we target the highly connected neurons we sample the tail of our connectivity distribution, and we observe that we disable approximately 10% of the network's connections even though we disable only 1% of the network's neurons.

The discrepancy in the fraction of disabled connections between these two strategies is due to the power law nature of the network's connectivity distribution, as well as the self reinforcement of connection strength that is driven by the Hebbian learning rules detailed in Sec. 3.2.2. With a sufficiently large network, randomly choosing neurons will access the same fraction of connections as neurons, but due to the power law connectivity distribution most of these neurons have a very small degree of connectivity and are relatively unimportant to the network dynamics. The Hebbian learning rules enforce that connections that are not often used will grow weaker than those that are used frequently, and neurons with low numbers of connections will not use those connections very often. Thus you need to disable a large number of these weak connections that make up the majority of the network's connectivity distribution in order to dramatically effect the network dynamics. Conversely, the connections from highly connected neurons will get used quite often and will be inherently stronger than the connections to neurons with lower k_{out}. In addition to the highly connected neurons serving as "hubs" that connect many otherwise poorly connected parts of the network together, their inherently stronger connections will sustain signals more easily in the network and will be more important to the propagation of extremely large avalanches. Thus only approximately 10% connections need be removed from the network when taken from the tail of the connectivity distribution through the disabling of the most highly connected neurons.

3.5 Discussion

Operating our model with the parameters described in Sec. 3.2.1 and an inhibitory fraction of $p_{inh} = 0.30$, we observe that the avalanche strength distribution of our model follows a power law of $P_S(S) \sim S^{-1.55}$, and that the avalanche duration distribution of our model obeys a power law of $P_D(D) \sim D^{-2.1}$, both of which agree well with experimental results [45, 47] and reproduce the results shown by Lombardi, Herrmann, De Arcangelis et al. As we lower the inhibitory fraction of our network towards an inhibitory fraction of 0.04, we intriguingly find behavior suggestive of criticality as the exponential cut-off present previously in the avalanche strength and duration distributions disappears, and these distributions continue to follow power laws for several more decades. At this value of inhibitory fraction, the network becomes dominated by long-lasting avalanches that persist for orders of magnitude more time steps than avalanches in the cut-off distributions. The particular value of inhibitory fraction at which we see this extension of the distributions is far below the fraction found in human cortices, which is closer to $0.2 - 0.3$ [15, 48].

Additionally, the power spectral density of our network at low inhibitory fractions ($p_{inh} = 0.04$) behaves similarly to power spectral densities of epileptic humans by following a power law with exponent -2.0. As the inhibitory fraction of the network is increased to a more biologically relevant value of 0.3 the period of power law behavior in the PSD shrinks, but we still see a brief regime of possible power law activity with the exponent -1.0, which matches the observed exponent of healthy human brains [46]. The transition of the exponent from an "epileptic" value of -2.0 to a "healthy" value of -1.0 reproduces results observed by Lombardi, Herrmann, De Arcangelis et al. [15].

Low inhibitory fractions allow the network to access much higher avalanche durations and corresponding avalanche strengths, because even though the underlying power law distri-

bution of these quantities does not change, the exponential cut-off disappears allowing the power law distributions to extends into regimes of greatly increased duration and strength. The incredibly large "black swan events" that the network can access have correspondingly low probabilities due to the power law distribution, but because they are so large, they dominate the network for many orders of magnitude more time steps than avalanches from a "healthy" distribution, once they occur.

The exponential cut-offs protect the network from these events, and human cortices may naturally operate at higher inhibitory fractions in order to avoid a truly critical point, yet still benefit from wide distributions at lower intensity avalanche events.

This corroborates the idea proposed by Milton et al. [57], that critical behavior in the brain, though long sought after, might be destructive as the long-range correlations introduced by approaching a critical point could destroy and dominate the short-range interactions necessary for the proper operation of the brain.

We also observe how the outgoing connectivity distribution changes as the network evolves under the Hebbian learning rules described in Sec. 3.2.2 after $45,000$ avalanches have run through the system.

Networks with a high inhibitory fraction ($p_{inh} = 0.30$) prune away many connections, as the system is unable to propagate avalanches strong enough to sustain all of the links. This results in the tail of the connectivity distribution being truncated with ultimately no neurons maintaining more than 75 outgoing connections, while the head of the distribution becomes inflated, as many neurons end up having only one or zero outgoing connections. Networks with a lower inhibitory fraction ($p_{inh} = 0.04$), prune away fewer connections than networks with a higher inhibitory fraction, as they are able to sustain stronger avalanches in the network. This results in a extended connectivity distribution tail, with some neurons having

as many as 91 outgoing connections after $45,000$ avalanches. Additionally these networks display an order of magnitude fewer neurons with zero or merely one connection than the networks endowed with a higher inhibitory fraction.

The combination of the inhibitory neurons and the Hebbian rules of the system cause networks with high inhibitory fractions to evolve into more sparsely connected networks than networks with a low inhibitory fraction. These differences in connectivity reinforce the networks' ability to sustain or disrupt very large avalanches. Networks with high inhibitory fractions will display weaker avalanches, causing them to be less connected, which in turn further weakens them in their capability to sustain large-scale avalanches. Networks with lower inhibitory fractions will on occasion go through massive avalanches allowing them to remain more connected, which hence will assist these systems in permitting further strong avalanches.

Finally we investigate two different strategies to remove these exceedingly large avalanches from networks with low inhibitory fraction through either the disabling of randomly selected or carefully chosen highly connected excitatory neurons, respectively. In order to curtail these large events through random disablings, 30% of the networks excitatory neurons must be disabled. This strategy is therefore ultimately effective, but would be quite destructive to any biological neural networks. In contrast, switching off highly connected neurons proves to be a much more effective strategy, as only the top 1% of these prominently connected excitatory neurons need to be disabled in order to prevent such large-scale avalanche events. Both of these strategies provide a means to circumvent the inherent occurrence of incredibly large "epileptic" avalanches in systems with very low inhibitory neuron fraction.

3.A Extended model to reproduce waiting time distribution

The model described in section 3.2 accurately recreates the avalanche strength distribution, the avalanche duration distribution, and the power spectrum distribution observed in rat and human cortices [45, 52]. In order to also generate experimentally detected waiting time distributions, the model must be extended. This extended model is detailed by Lombardi, Herrmann, de Arcangelis et al. [12, 13, 14, 15] and is briefly summarized here.

The waiting time between avalanches is the time between the end of the last avalanche and the beginning of the next. Figure 3.11 shows experimentally determined waiting time of seven different slices of rat cortex. This figure was taken with permission from Lombardi, Herrmann, de Arcangelis et al.'s paper *The balance between excitation and inhibition controls the temporal organization of neuronal avalanches* [12].

The experimentally determined waiting time distributions [12] shown in Fig. 3.11 display bimodal behavior with an initial power law regime followed by a "bump."

This bimodal behavior requires that avalanches which occur within short waiting times should be highly correlated to previous avalanches in order to reproduce the power law behavior at low waiting times. In comparison, avalanches with long waiting times need to be uncorrelated to reproduce the exponential behavior seen at longer waiting times.

These distinct features can be reproduced by introducing two network-wide macro-states: "up" and "down." The up state is defined as a period of high network activity, during which many neurons are close to firing potential. Anytime an avalanche occurs, the system transitions to or remains in the up state.

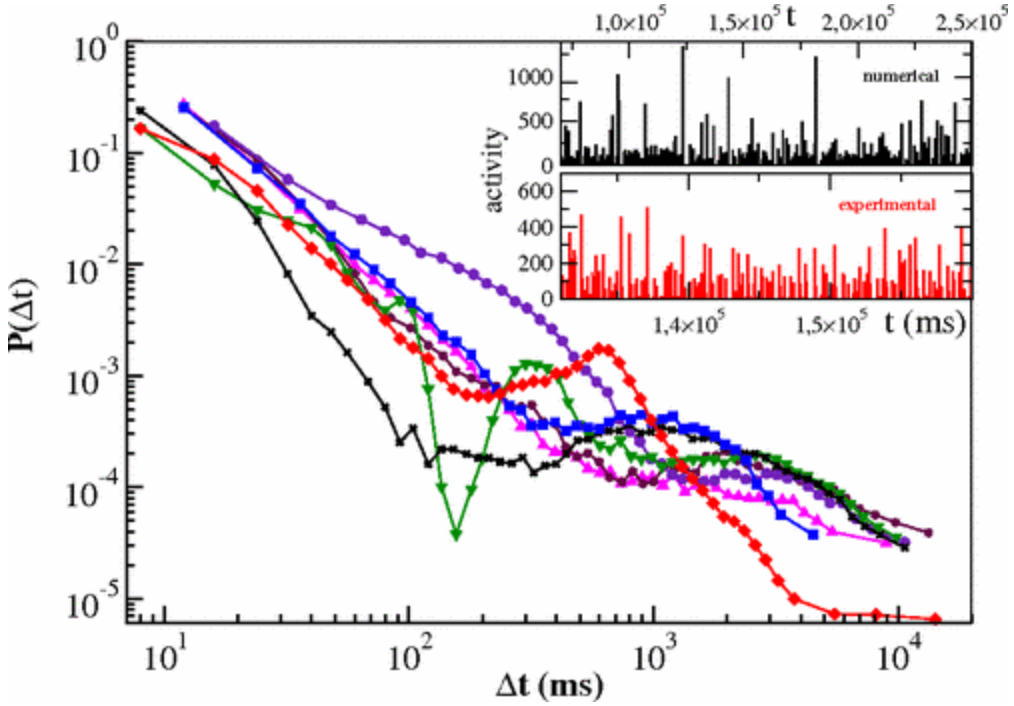


Figure 3.11: Figure reproduced with permission from Ref. [12]. This plot shows the avalanche waiting time distributions for seven different slices of rat cortex. The inset shows two examples of temporal neural sequences. The majority of the waiting time distributions display bimodal behavior, with an initial power law at low waiting times, followed by a bump at higher waiting times.

During an up state, the noise driving the system is drawn from the distribution $(0, S_{max}/S(\tau)]$, where S_{max} is the avalanche strength threshold, and $S(\tau)$ is the strength of the last avalanche. Additionally, when an avalanche ends in the up state, each neuron in the network is reset to be close to the neuron firing threshold.

$$n_i \to n_{max}(1 - S(\tau)/S_{max}) \ . \tag{3.10}$$

The resetting in Eq. (3.10) ensures with high probability that avalanches in the up state will have short waiting times, and the correlations introduced from the up state noise distribution produces the power law behavior seen in Fig. 3.11.

If the strength of the last avalanche exceeds the avalanche strength threshold S_{max}, the system transitions to the down state. The down state is characterized as a period of no activity in the network in which the system is slowly brought back to firing. When the network transitions from the up state to the down state, each neuron is drastically polarized in opposition of the previous behavior,

$$n_i \to n_i - h\Delta n_i \ , \tag{3.11}$$

where Δn_i is the sum of depolarizations during the last avalanche in the up state, and h is a system parameter introduced to control the strength at which the neuron is anti-polarized. During the down state, the network is driven by small ($\sim 0.01 \cdot n_{max}$) constant noise.

The hyperpolarization of neurons coupled with the small constant noise ensures that the waiting times during the down state will be very long, and will produce an approximately Gaussian distribution due to the central limit theorem.

Chapter 4

Phase transitions in image denoising via sparsely coding convolutional neural networks

The following chapter has been heavily extended and adapted, with permission from the workshop, from a four-page conference proceedings that was presented at the NIPS 2017 workshop Advances in Modeling and Learning Interactions from Complex Data:

J. Carroll, N. Carlson, and G. T. Kenyon. Phase Transitions in Image Denoising via Sparsely Coding Convolutional Neural Networks. NIPS 2017 workshop on Advances in Modeling and Learning Interactions from Complex Data, arXiv 1710.09875 (2017).

The work this chapter is based on was performed at Los Alamos National Laboratory, and has been cleared for unrestricted release under LA-UR-17-26726.

4.1 Introduction

The system detailed in the following chapter, a sparsely encoding convolutional neural network, differs greatly from the neural network detailed in Chapter 3. That network is based on reproducing the macroscopic avalanching behavior seen in biological neural networks (i.e. real brains), and exists in both structure and analysis in close analogy to sandpile models.

The following network is based in part on the dynamics of a real network [18, 58], but ultimately is designed to solve computational problems, and so has been heavily modified and optimized such that it no longer resembles a biological network.

Specifically we used PetaVision [59], a high performance neural simulation toolbox designed by Dr. Garrett Kenyon of Los Alamos National Lab, to simulate these sparsely encoding convolutional neural networks, and looked at the finite-size scaling of these networks as they denoised small 32x32 thumbnail images taken from the CIFAR-10 dataset [17].

The image denoising error of these networks was observed to have a minimum at a particular fraction of active neurons, i.e. the number of neurons with a non-zero external potential, and the location and depth of this minimum were observed to follow power law relations as the system size of the network (in this case the number of neurons) was changed. These algebraic relationships are a sign of criticality, and suggest that the network is undergoing a second-order phase transition as the fraction of active neurons is changed.

4.2 Sparsely encoding convolutional neural network

Sparsely encoding convolutional neural networks are a class of neural networks devised by Rozell et al. [18] to mimic behavior observed in the human visual cortex by reconstructing inputs to the network sparsely, i.e. by using the fewest number of neurons possible [58]. These artificial neural networks do this by minimizing the following energy function:

$$E(t) = \|\mathbf{s}(t) - \widehat{\mathbf{s}}(t)\| + \lambda \sum_m C(a_m(t)) \ , \tag{4.1}$$

where $\mathbf{s}(t)$ is the input vector given to the network, $\widehat{\mathbf{s}}(t)$ is the network's reconstruction of the input, a_m is the output or "activation" of the m^{th} neuron, C is some cost function defined to be non-negative, and λ is a system parameter that scales the importance of the second term

in the energy. The choice of C is dependent on the dynamics of individual neurons, and is described in Sec. 4.2.1; however, for this system C ends up being the l_1 norm. This make the energy function of this system:

$$E(t) = \|\mathbf{s}(t) - \widehat{\mathbf{s}}(t)\| + \lambda \sum_m |a_m(t)| \,. \tag{4.2}$$

This energy function then has two competing terms: the first "error" term that is the l_2 norm of the difference between the network's input and its reconstruction of that input, and the "cost of activity" term that is the scaled l_1 norm of the outputs of all neurons in the system. In order to minimize this energy, the system must balance properly reconstructing its input with using as few neurons as possible, i.e. its output is "sparse" [18, 60]. In the following subsection we describe the specific model of neurons that have been devised to minimize this energy.

4.2.1 Neurons

In the system we are using, neurons are modeled as leaky integrators that inhibit other "similar" neurons. This model was devised by Rozell et al. [18].

Specifically, each m^{th} neuron in this system is modeled as having some internal potential $u_m(t)$, a vector of inputs $\boldsymbol{s}(t)$ that is the same for all neurons, some vector of weights $\boldsymbol{\Phi}_m(t)$ that governs the strength of its inputs, and an output or "activation" $a_m(t)$.

The input vector and weight vectors of all neurons are all of length 3072. Specifically, the inputs for all networks described in this chapter were all 32x32 RGB thumbnail images[1] that have been transformed into 3072 length vectors, all taken from the CIFAR-10 image set [17].

[1]The 32x32 RGB images are 32x32x3 matrices, but we transform them into a 3072 length vector in order to serve as the input to our neurons. When we refer to spacial locations and sections of images throughout this chapter we are implicitly referring to the elements of this 3072 length vector that can be transformed back to those spatial locations.

The pixel values of these images were "normalized" for each image such that they had an average of zero and a standard deviation of one.

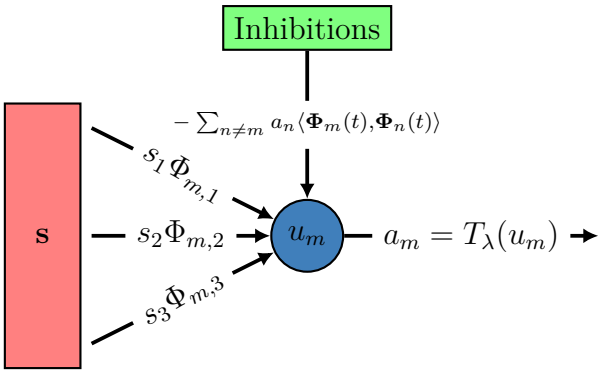


Figure 4.1: A schematic of the m^{th} sparsely encoding convolutional neuron. The (red) input vector $\mathbf{s}$ is dotted with the neuron's weight vector $\mathbf{\Phi}_m$. The result of this dot product is added to the internal potential u_m represented by the (blue) circle. Additionally, inhibitions from neurons with similar weight vectors (represented by the green rectangle) will be subtracted from internal potential u_m. Though not displayed, the internal potential will also decay proportional to its value each time step. Finally after the weighted inputs and inhibitions have been added to the internal potential, it is passed through a thresholding function (see Eq. (4.11)) to become the output a_m of the neuron. The competition between the inhibitions, inputs, and internal decay of the neuron enforce that neurons with weaker inputs will eventually have zero output, while neurons with strong inputs will have proportionally stronger outputs.

Additionally, the weight vectors $\mathbf{\Phi}_m(t)$ of each neuron were chosen in a very specific fashion. A subset of the input was chosen for each m^{th} neuron, and all elements of the weight vector that did not share the same indices as this subset of the input vector were set to zero. In the case of the thumbnail images used for this network this subset of the input was simply a 8x8 subsection of one color channel of the image, with different neurons being assigned a different 8x8 subsection of the input image. The non-zero elements of each neuron's weight vector are initialized such that many neurons will share spatially shifted copies of the same weight vector in an attempt to "convolve" a single weight vector over the entire input. The exact form of the initialization of the weight vectors is described in detail in Sec. 4.2.2.

The neuron performs a dot product between the input vector and its weight vector and adds

the result to its internal potential u_m. The internal potential will decay with some half-life τ. The combination of the integration of the weighted inputs into the internal potential and its subsequent decay results in the "leaky integrator" part of the neuron model, so the internal potential must be constantly increased through the integration of the weighted inputs or it will decay to zero.

The output of the m^{th} neuron, $a_m(t)$, is some thresholding function T of its internal potential:

$$a_m(t) = T(u_m(t)) \ . \tag{4.3}$$

The form of T for this model will be derived further down, but the only restriction placed on the general form of the thresholding function is that it is monotonic in u_m.

Because the thresholding function is monotonic, the output of the neuron is representative of how much the weight vector of each neuron overlaps with the input vector. In our system this indicates how similar the non-zero portion of the neuron's weight vector is to the originally chosen 8x8 subsection of the image.

If the output of a particular neuron is high then we say it "represents" that 8x8 portion of the image well. That is to say, its weight vector is a good approximation of that section of the input image. Indeed we define the reconstruction of the input image $\hat{\boldsymbol{s}}$ as:

$$\hat{\boldsymbol{s}} = \sum_n a_n \boldsymbol{\Phi_n} \ , \tag{4.4}$$

so that the reconstruction of the input is simply a linear combination of the each neuron's weight vectors using the respective output of that neuron as the coefficient. Again we can transform a 3072 length vector to a 32x32 RBG image, so we can transform this output back into an image.

However, neurons in this model will inhibit, or decrease the internal potential of, other neurons proportional to their output multiplied by the overlap of the weight vectors of the neurons. If the inner product between two neurons' weight vectors is large then they must share both spatially similar 8x8 sections of the input image, and have similar non-zero elements in their weight vectors. Because the goal of this network is to approximate the input image using a linear combination of each neuron's weight vectors with the smallest number of active neurons, this degree of similarity is not wanted because the information present in both weight vectors is just as well represented in a single one.

This forces neurons with both similar 8x8 chosen subsection of the input and similar non-zero elements of their respective weight vectors to compete.

The total behavior of a neuron's internal potential can be collected into one differential equation:

$$\dot{u}_m(t) = \frac{1}{\tau}(\langle \mathbf{\Phi}_m(t), \mathbf{s}(t) \rangle - \sum_{n \neq m} a_n \langle \mathbf{\Phi}_m(t), \mathbf{\Phi}_n(t) \rangle - u_m(t)), \tag{4.5}$$

where the first term represents the integration of the neuron's weighted inputs, the second term represents the inhibitions the neuron receives due to its weight vector's overlap with similar neurons, and the third term causes the internal potential to decay exponentially.

The derivation of the cost function C and the thresholding function T

The minimization of the energy defined in Eq. (4.2) is done by making an ansatz of quasi-gradient descent of the internal potentials of the neurons:

$$\dot{u}_m \propto -\frac{dE}{da_m} \, . \tag{4.6}$$

This relation can be used with Eqs. (4.1) and (4.5) to get a differential equation relating the yet unspecified C and T:

$$\lambda \frac{dC(a_m)}{da_m} = u_m - T_\lambda(u_m) \ . \tag{4.7}$$

At this point we pick a general form of T,

$$T_{(\alpha,\gamma,\lambda)}(u_m) = \frac{u_m - \alpha\lambda}{1 - e^{\gamma(u_m - \alpha)}} \ , \tag{4.8}$$

which is chosen because it is monotonic and easily numerically integrable [18]. We can insert this into Eq. (4.7) and numerically integrate it while taking the limit as $\gamma \to \infty$ to get C:

$$C_{(\alpha,\infty,\lambda)}(a_m) = \frac{(1-\alpha)^2\lambda}{2} + \alpha|a_m| \ . \tag{4.9}$$

Finally setting $\alpha = 1$ gives

$$C(a_m) = |a_m| \ , \tag{4.10}$$

and

$$T_\lambda(u_m) = \begin{cases} 0, & u_m < 1 \ , \\ u_m - \lambda, & u_m \geq 1 \ . \end{cases} \tag{4.11}$$

The forms of C and T shown in Eqs. (4.10) and (4.11) thus guarantee that this system will minimize the energy function from Eq. (4.2) while obeying the dynamics governed by Eq. (4.5).

Discretization and restructuring of the model

In order to actually simulate these networks, each neuron must check each other neuron in order to determine if it inhibits it. This is very computationally expensive, but thankfully the differential equation shown in Eq. (4.5) can be restructured to change the local inhibitions between neurons to a global difference between the input to the network and the network's reconstruction, which is much easier computationally.

To show this, consider the second term in Eq. (4.5) that is responsible for the local inhibitions:

$$\begin{aligned}\sum_{n\neq m} a_n \langle \mathbf{\Phi}_m(t), \mathbf{\Phi}_n(t)\rangle &= \langle \mathbf{\Phi}_m, \sum_{n\neq m} a_n \mathbf{\Phi}_n\rangle \ , \\ &= \langle \mathbf{\Phi}_m, \sum_{n} a_n \mathbf{\Phi}_n\rangle - \langle \mathbf{\Phi}_m, a_m \mathbf{\Phi}_m\rangle \ , \\ &= \langle \mathbf{\Phi}_m, \widehat{\mathbf{s}}\rangle - a_m \ .\end{aligned}$$

Replacing terms allows Eq. (4.5) to be rewritten as:

$$\dot{u}_m(t) = \frac{1}{\tau}(\langle \mathbf{\Phi}_m(t), \mathbf{s}(t) - \widehat{\mathbf{s}}(t)\rangle - u_m(t) + a_m(t)) \ . \tag{4.12}$$

This changes the neuron from taking the inner product of its weights and its input, to the inner product of its weights and the difference between the input and the network's reconstruction of the input. This is functionally identical to the previous form of the equation, but is much easier to perform computationally, and for this reason this is the form of the dynamics that is followed in the actual simulations.

Then, we can discretize this differential equation into time steps of length $\tau = 1$. The update

rule for the internal potential of the m^{th} neuron becomes:

$$\Delta u_m(t+1;t) = \langle \mathbf{\Phi}_m(t), \mathbf{s}(t) - \widehat{\mathbf{s}}(t) \rangle - u_m(t) + a_m(t) \ , \tag{4.13}$$

where $\Delta u_m(t+1;t)$ is the difference in u_m between the t and $t+1$ time-steps:

$$\Delta u_m(t+1;t) = u_m(t+1) - u_m(t) \ .$$

4.2.2 Network structure

Each neural network described in this chapter is structured into layers. Most of these layers have no neurons in them, and exist only to either store the inputs and outputs of the network, or to perform some manipulation on the input or outputs. Neurons only exist in a single layer, the "V1" layer, so named after the V1 visual cortex in human brains [18].

Every neuron in the V1 layer is capable of inhibiting every other neuron in the V1 layer, based on the overlap of their weight vectors and their outputs, respectively.

The V1 layer is composed of sublayers of neurons, shown in Fig. 4.2. Each of these sublayers has three groups of neurons, each group of which is composed of neurons with weight vectors that have only non-zero elements corresponding to one color channel of the input images. i.e., each neuron in the "blue" group of the sublayer shown in Fig. 4.2 has weight vectors that will only overlap with elements of the input image in the blue color channel (which again, is transformed to some subset of a 3072 length vector before the inner product of the weight vectors and the input is performed).

These groups are formed by creating one neuron per color channel, and assigning it some 8x8 square of the 32x32 input images. The location of this 8x8 square will never change.

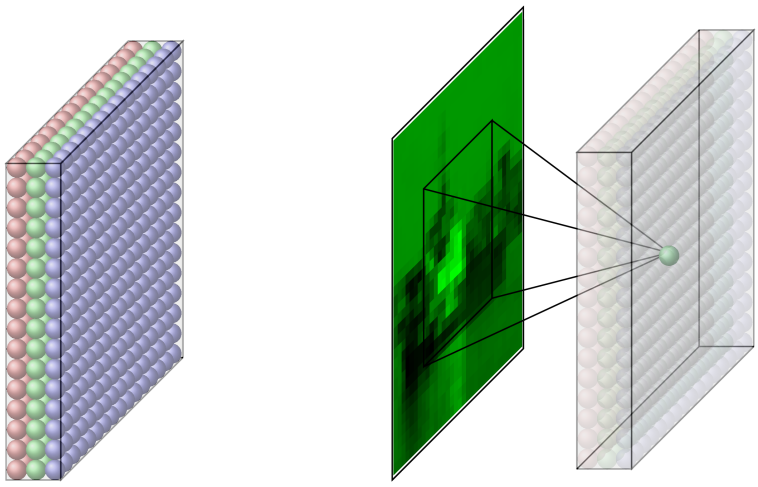


Figure 4.2: A single sublayer of the V1 layer (left), and a individual neuron with the spacial representation of its weight vector highlighted (right). The sublayer (left) is composed neurons, represented by the multicolored spheres, collected into three sections of neurons, each colored to represent which color channel of the input image its weights will overlap. Each section is created by creating a single neuron, with a randomly initialized 8x8 patch of non-zero elements in its weight vector, and then repeatedly copying this neuron and translating its 8x8 patch of non-zero weights until there is a neuron for every possible 8x8 patch of the assigned color channel. This corresponds to convolving the original 8x8 patch over the entire color channel of the image. Different neurons are created and convolved for each color channel. The right image shows the portion of an input image that the center green neuron will perform an inner product with.

For the purpose of this explanation, consider the top-leftmost 8x8 corner of a 32x32 image's blue color channel. This is visually represented in Fig. 4.3.

Every element of the neuron's weight vector that does not share indices with this 8x8 block are permanently set to zero. The elements of the neuron's weight vector which do share indices have their values randomly initialized from a Gaussian distribution with mean zero and standard deviation one, and are then normalized.

Figure 4.3: A visual representation of the elements accessible to a single neuron's weight vector. The image above is a 32x32 pixel thumbnail image of a ship, with an example neuron's assigned region of the picture's blue color channel highlighted in blue. This neuron has been assigned the top left most corner of the blue color channel. No other elements of the neuron's weight vector that do not share indices with the highlighted region are allowed to be non-zero.

We then construct the entire blue section of this sublayer by copying this neuron and spatially shifting the 8x8 patch of non-zero weights over by single pixel values until the entire image is covered by massively overlapping 8x8 patches of neurons' weight vectors. This is essentially a convolution of the original neuron's 8x8 patch of weights over the entire image. This process of copying and shifting neurons to convolve a single weight vector over an input is the "convolutional" part of this sparsely encoding convolutional neural network.

In order to completely cover the image with the convolution of the original weight vector, 256 neurons need to be created, which can be organized into a 16x16 square of neurons.

This process is repeated with a neuron for the green and red channels, creating a 16x16

square of neurons for each color. The weight vectors are not copied across color channels, and a new 8x8 patch of weights is initialized and convolved for each color channel.

Thus a sublayer is composed of a block of 16x16x3 neurons.

The V1 layer is composed of many of these sublayers. The number of sublayers in the V1 layer of our network will not change after the creation of a network, and the total number of neurons N in each network is:

$$N = S \cdot 16 \cdot 16 \cdot 3 \ ,$$

where S is the number of sublayers in the V1 layer.

The finite-size scaling analysis performed later on will require changing the system size of the network, which we consider to be the total number of neurons in the network. We change N across different simulations by changing the number of sublayers S each network is initialized with.

Adjustment to the weight vectors

The purpose of a convolutional neural network is to "learn" commonly occurring patterns that can be observed across many different images. This learning process will be performed by updating the weights of the neurons making up the neural network as it is iteratively exposed to images. The learning mechanism itself borrows from biological processes, where the strength of synapses grow proportionally to the strength of signals sent through them, and how strongly the receiving neuron fires [5, 6].

In our model this will take the form:

$$\Delta\Phi_{m,i} \propto \begin{cases} a_m \cdot s_i, & \Phi_{m,i} \text{ is allowed to change ,} \\ 0, & \text{else ,} \end{cases} \tag{4.14}$$

where $\Delta\Phi_{m,i}$ is the update to i^{th} element of the m^{th} neuron's weight vector. However, in addition to adding $\Delta\Phi_{m,i}$ to $\Phi_{m,i}$, $\Delta\Phi_{m,i}$ also gets added to the elements of the weight vectors of every neuron in the same color channel of the same sublayer as the m^{th} neuron. Specifically, $\Delta\Phi_{m,i}$ is added to all weight vector elements that are the convolution of the same element in the original 8x8 patch. In this way, the 8x8 patches of all neurons in a sublayer's respective color channels are updated together. When any one of them would increase, they all increase. After updating all weights, the weight vectors are renormalized.

The combination of this convolutional learning and renormalization forces the weights of neurons to converge to patterns common to many different images [61], and we will use this to learn a quasi-basis of these patterns that will be used for image reconstruction.

4.2.3 The training network

In order for the weight update rule described in Eq. (4.14) to force the neuron's weight vectors to converge to patterns common to many images, the V1 layer described in the previous section must be exposed to many different images, and exposed to each image sufficiently long enough for the learning rule to be effective. We created a specific network, named the training network, to train the weight vectors by exposing the network to images for 500 time steps, for 50,000 different images.

A schematic of the training network is shown in Fig. 4.5. The training network is operated by initializing the input layer with an image taken from the CIFAR-10 image set. The input error layer, V1 layer, and input reconstruction layer are all initialized as either a vector of

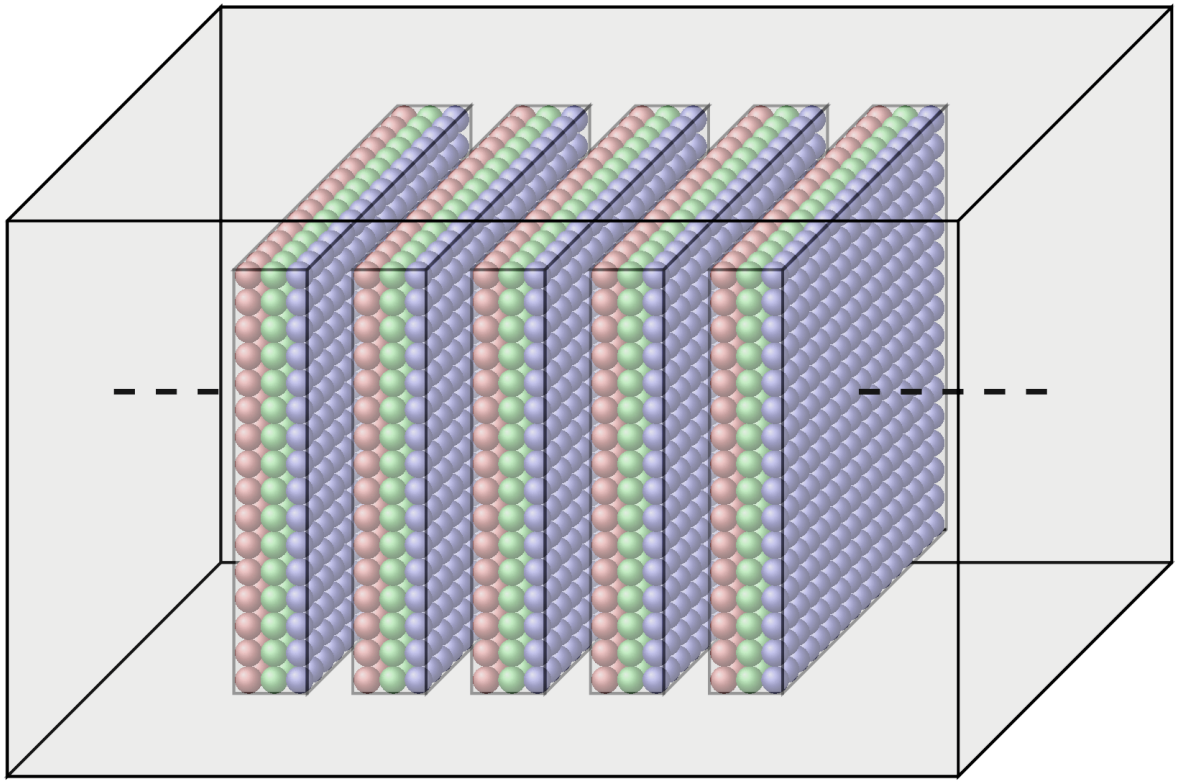


Figure 4.4: A schematic of the V1 layer of the neural network. The layer is made of many different sublayers, each composed of three different sets of convolutional neurons, one for each color channel of the input images.

zeros for the input error and input reconstruction layers, or with all zero internal potentials for the neurons of the V1 layer. Once the input layer has been initialized with the RGB values of the initial image, the difference between the input layer and input reconstruction layer is calculated and stored in the input error layer. For the first time step, the difference is the entire input image. The input error is then given as the input to the V1 layer. The V1 layer was initialized with random weights as described above, with some number of sublayers S. Many different networks with varying numbers of sublayers were trained for the subsequent system size scaling analysis.

The V1 layer performs dot products between the weight vectors of its neurons and the input error layer, and adds the results of the dot product to the internal potentials, a_m. Because the input to the network is the difference between the original input and the network's reconstruction (which is initially nothing), the inhibitions between neurons do not need to be calculated and each neuron can immediately calculate its external potential u_m using Eq. (4.11). After all external potentials have been calculated, the reconstruction of the input can be calculated from Eq. (4.4), which is done by computing the linear combination of each neuron's weight vectors and external potentials. The result of this is stored in the input reconstruction layer.

The weights of each neuron are then updated according to Eq. (4.14). The previous steps account for all the actions taken during a single time step.

During the next time step, the difference between the input and input reconstruction is calculated and stored in the input error layer, and the input error layer is again given to the V1 layer neurons as their input. The V1 layer will create a new reconstruction of the image, and will have their weights updated.

This process will continue for 500 time steps, after which the input will be swapped for a

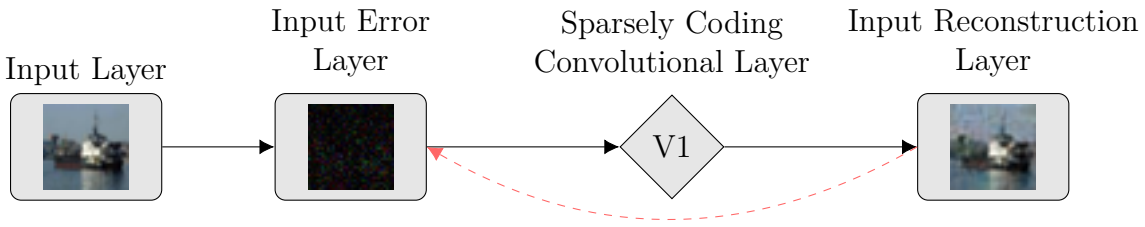


Figure 4.5: A schematic of the neural network used to train the neurons' weights to recognized common patterns across thumbnail images. The network is composed of four layers: the input layer, the input error layer, the V1 layer, and the input reconstruction layer. Only the V1 layer contains neurons, all other layers are computational objects needed to hold and manipulate the inputs and outputs of the network. The input layer stores the current input image to the network. The input error computes and stores the difference between the current input, held in the input layer, and current reconstruction of the input, held in the input reconstruction layer. The input error layer serves as the input to the V1 layer. The V1 layer is the layer of convolutional neurons described previously. The neurons in the V1 layer perform an inner product between their weight vectors and the vector stored in the input error layer, following the update rule described in Eq. (4.13). After every time step, the reconstruction of the input is recalculated (see Eq. (4.4)). The result is stored in the input reconstruction layer. Additionally, at the end of every time step the weight vectors of the neurons in the V1 layer are updated according to Eq. (4.14). The same input image was given to the network for a period of 500 time steps, at which point it was replaced with a different thumbnail image from the CIFAR-10 dataset. This was repeated 50,000 times for 50,000 unique images. The weight update rules governed by Eq. (4.14) will copy portions of input error into the neurons weight vector proportional to how much that neuron was activated (i.e. how large a_m is). The renormalization of the weight vectors after this, followed by the exposure of the neuron to many different images effectively averages away any image specific patterns that the weight vectors "learn," leaving only patterns that are common across many images.

different image from the CIFAR-10 dataset. In this fashion, the network will be exposed to 50,000 images.

After this has been performed the weights of the neurons in the V1 layer will have converged to patterns common across the 50,000 images [61]. The constant renormalization of the weights serves to average away patterns common to only few images.

The weights of neurons in a sublayer of the V1 layer can be represented visually, by combining the 8x8 block of weights from each color channel and combining them into an 8x8 RGB image. An example of the patterns "learned" from a network with $S = 64$ sublayers, $N = 49,152$ neurons is shown in Fig. 4.6.

Once the weights of the V1 layer had been trained, we copied those weights to the V1 layer of a denoising network.

4.2.4 The denoising network

The denoising network was used to reconstruct images that had noise artificially added to them. This was performed by replacing the inputs to the network with noisy[2] 32x32 RGB images, while using the same set of weight vectors that was trained in the training network for the same value of S. The weight update rules were turned off for all denoising networks we used.

The denoising network is structurally very similar to the training network, with an extra layer in between the input and the input error layers: the noise layer. A schematic of the denoising network is shown in Fig. 4.7.

As with the training networks, the noisy inputs to the denoising network are maintained for

[2]Gaussian noise with mean zero and standard deviation 0.5 was added to each pixel value of the normalized images.

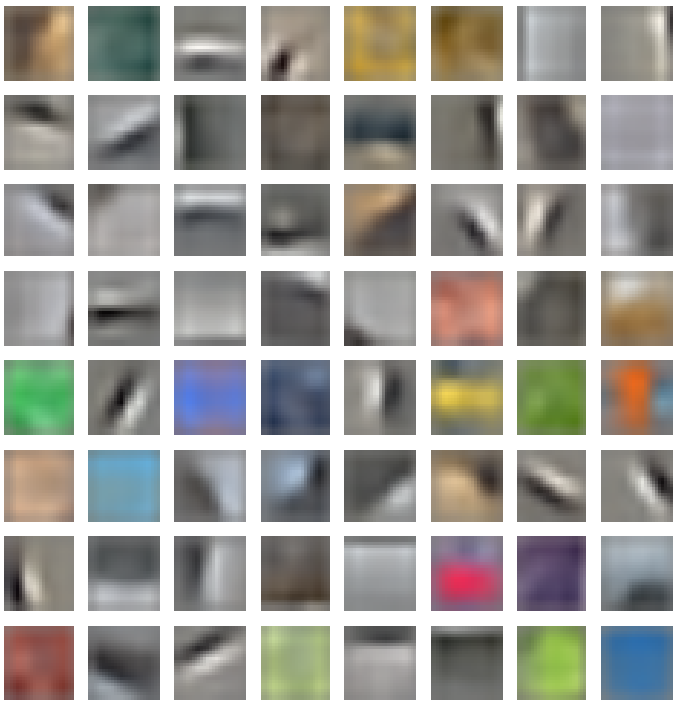


Figure 4.6: Visual representations of the patterns learned by a training network with $S = 64$ sublayers. Each 8x8 pixel image is the combination of the three 8x8 weight blocks from each sublayer's color channel. These 8x8 pictures form the basis elements that the networks attempts to reconstruct the input image from.

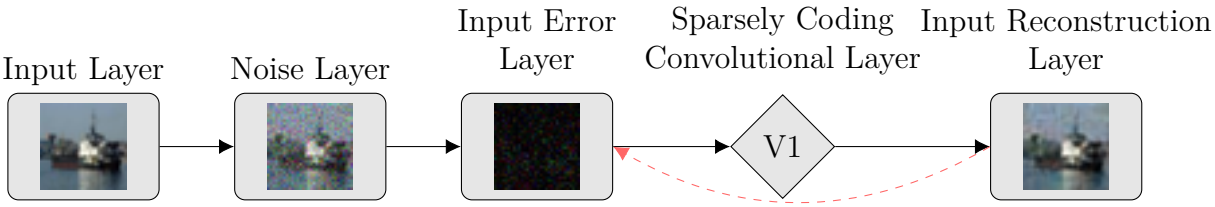


Figure 4.7: A schematic of the denoising neural network. The denoising network is structurally almost identical to the training network, with the only difference being the inclusion of the noise layer. Images from the CIFAR-10 image set are introduced into the input layer, which are then passed into the noise layer. The noise layer adds Gaussian noise with mean zero and standard deviation 0.5 to every pixel value, and presents that as the input to the input error layer. The input error layer calculates the difference between the noise layer and the input reconstruction layer, and presents that as the input to the V1 layer. The weights of the V1 layer are initialized from the a training networks final set of weight vectors. This requires that the number of sublayers S be the same in both the training and denoising network. Additionally, we varied λ across several different networks. For each different value of λ we used, a new set of weights was trained with that value. The weight update rules are turned off for the denoising network, so the weights will remain the values that they converged to in the training network. Again, the V1 layer will be exposed to the same input image with the same noise added to it for 500 time steps. The reconstruction of the image at the 500^{th} time step will be used to calculate the average reconstruction error and fraction of active neurons, Eqs. (4.15) and (4.16) respectively. After the 500 time steps the input image is replaced with another from the CIFAR-10 dataset and a new noise mask is generated by the noise layer. This process is repeated for 10,000 separate images. The 10,000 images used by the denoising network are new images, and were not used to train the network.

500 time steps before being replaced with a new noisy input. The final reconstruction of each input is the 500^{th} reconstruction of the given input. This reconstruction is used to calculate the average percent reconstruction error P_{err} and the average fraction of active neurons F_a, Eqs. (4.15) and (4.16) respectively.

$$P_{err} = \frac{1}{10,000} \sum_{i=1}^{10,000} \frac{\|\mathbf{s}_i - \widehat{\mathbf{s}}_i\|^2}{\|\mathbf{s}_i\|^2}, \tag{4.15}$$

where P_{err} is the average percent reconstruction error, $\mathbf{s}_i$ is the i^{th} original image before it has Gaussian noise added to it, $\widehat{\mathbf{s}}_i$ is the i^{th} reconstruction of the noised image taken from the sparsely coding convolutional layer [18, 59, 65]. The error is averaged over all 10,000 reconstruction of the noisy images given to the network.

It is energetically favorable for the network to have as few neurons with non-zero external potential, a_m, as possible in the V1 layer. This forces the network to balance between reconstructing the entire noisy input and activating additional neurons. Because the Gaussian noise we added to the image is hard to reproduce with the weight vector basis elements taken from the training network, the denoising network reconstructs the image without the noise. Its ability to do so accurately is dependent on the strength of the noise, and the strength of the sparsity term in the energy (see the second term of Eq. (4.2)), which is governed by λ.

The parameter λ can be used to change the fraction of active neurons used in the networks final reconstruction of the image. The fraction of active neurons is the fraction of neurons in the network that have non-zero a_m. We are interested in the average fraction of active neurons of all reconstructions of the 10,000 noisy images,

$$F_a = \frac{1}{10,000} \sum_{i=1}^{10,000} \frac{1}{N} \sum_{m=1}^{N} \Theta(a_m) \ , \tag{4.16}$$

where F_a is the average fraction of active neurons, Θ is Heaviside's step function. The fraction is averaged over all 10,000 reconstructions of the noisy images.

We observed a minimum in P_{err} as F_a was change by varying λ. Because minima and maxima in a system parameter can be indicative of a phase transition, and because phase transitions have been observed in other neural networks [63], we investigated the existence of a phase transition by observing how this minimum changed location and depth as the system size of the network, N, was changed.

4.3 Phase transitions and finite-size scaling

A phase of a system is defined as a subspace of the microscopic system parameters where the system's dynamics obey the same macro-scale laws and relations everywhere in that subspace. The space of system parameters can have many phases, and the system can transition between them as system control parameters change.

In a system undergoing a phase transition, there is a given "order parameter," chosen from the systems macroscopic quantities, that specifies the phase in which the system is in. An order parameter will be chosen such that it is zero in one of the phases of the system, typically the disordered, high-temperature phase, outside of which it will change to a non-zero value. The behavior of the order parameter as the system crosses between phases can be used to characterize a given phase transition into two different classifications. If the order parameter jumps discontinuously when the system transitions between phases, then the system is said to undergo a first-order phase transition. However, if the order parameter changes continuously then the system is said to undergo a second-order, or continuous, phase transition.

In first-order phase transitions the correlation length of the system is generally finite, but diverges in systems undergoing second-order phase transitions with infinite system size.

Specifically this divergence, and the continuous phase transition, occurs at a "critical point" in the system parameters. This divergence of the correlation length washes away any short-scale dynamics, and causes scale-free behavior to emerge in most of the system's statics and dynamics in sufficiently long time-scales. This scale-free behavior is characterized by power laws in the system which are in turn governed by a set of "critical exponents." Because much of the system-specific behavior is destroyed by these diverging correlation lengths, disparate systems can display the same critical behavior sufficiently close to their respective critical points by sharing the same set of critical exponents. Systems which have identical critical exponents are said to belong to the same "universality class."

In addition to driving the statics and dynamics of the system to a set of power laws, the divergence of the correlation length can cause other system parameters to diverge as well. Again this divergence can only occur in systems with infinite system sizes, but even in finite-sized systems these points should still show up as extrema in the system which should diverge as the system size is increased [2, 3]. Specifically, the shift in the location of these extrema from their location in an infinite system as well their height should change according to a set of power laws governed by the critical exponents ν, and γ. Equations (4.17) and (4.18) show how a minimum in the system would shift in location and depth as governed by these critical exponents:

$$\text{Shift of minimum relative to infinite system} \sim L^{-1/\nu} \,, \tag{4.17}$$

$$\text{Height of minimum} \sim L^{-\gamma/\nu} \,, \tag{4.18}$$

where L is the linear system size. This behavior is known as finite-size scaling [2, 3], and is indicative of criticality. The meaning of these exponents is well defined: ν controls the divergence of the correlation length in critical systems approaching the critical point, while γ controls the divergence of the susceptibility of the system. The meaning of the correlation

length and the susceptibility are not well understood in this sparsely encoding convolutional neural network, and may have no true meaning in this system. Additionally, L in Eqs. (4.17) and (4.18) is the linear system size, analogous to the linear width or height of a d-dimensional volume, while the dimensionality of our neural network system is not immediately clear. Thus we cannot use Eqs. (4.17) and (4.18) as a measure of this system's criticality, but we can use a set of paired equations that are analogous to them:

$$\text{Shift of minimum relative to infinite system} \sim N^{-1/\bar{\nu}} \,, \tag{4.19}$$

$$\text{Height of minimum} \sim N^{-\bar{\gamma}/\bar{\nu}} \,. \tag{4.20}$$

System quantities diverging algebraically with system size is still a symptom of criticality, and we can still measure how the minimum in reconstruction error shifts in depth and location as we vary the number of neurons in our system. We take N, the total number of neurons in our network, as the analogy to L, the linear system size, because the dimensionality of these networks is unknown and N is the simplest measure of system size. Additionally we replace ν and γ with the variables $\bar{\nu}$ and $\bar{\gamma}$. These exponents only describe power law relations between the minimum and the system size, and cannot be said to be related to the critical ν and γ due to the unclear existence of correlation length and susceptibility in this system. However, Eqs. (4.19) and (4.20) will still act as a measure of critical behavior in this system, and we can probe the existence of critical phenomena in this system by varying the number of neurons the networks are initialized with and observing how the position and depth of the minimum in reconstruction error changes.

4.4 Results

The parameters of the system that we are interested in are the fraction of active neurons and the average percent reconstruction error of our noisy images:

The fraction of active neurons is controlled by a parameter λ, as described above [18], that behaves monotonically with the sparsity of active neurons and inversely with the fraction of active neurons. Through λ we can control the fraction of active neurons and observe how the average percent reconstruction error behaves as the fraction of active neurons is varied. We observed a minimum in P_{err} occur as we varied the fraction of active neurons for many different system sizes. These results are summarized in Fig. 4.8.

We measured the shift in height and location of the minima in P_{err} as the system size was varied, and plot each on a log-log plot (see Figs. 4.9 (a), and 4.9 (b)).

Table 4.1: The finite-size scaling exponents.

Exponent	Value
$\bar{\nu}$	1.32 ± 0.04
$\bar{\gamma}$	0.0099 ± 0.0095

We observe power law behavior in both the location and height of the minima as the system size is varied in the manner defined by the finite-size scaling relationships Eqs. (4.19) and (4.20). This finite-size scaling behavior suggests the system is undergoing a continuous phase transition is occurring as the sparsity of the network is varied.

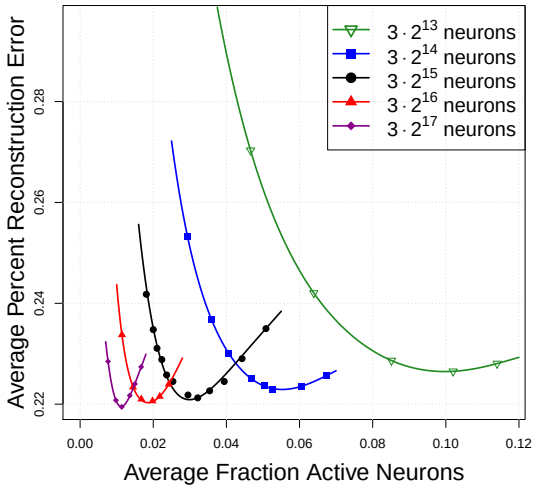
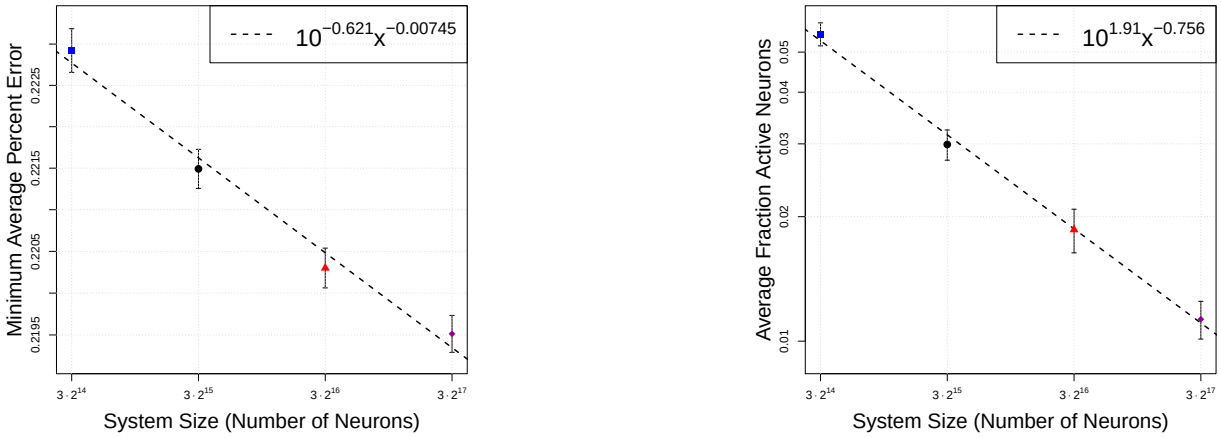


Figure 4.8: A plot of the average reconstruction error vs. the average fraction of active neurons for five different network sizes, and many different average fractions of active neurons. Each data point represents results from a unique neural network that was first trained on unnoised images, as described in Section 4.2.4 for a particular system size and value of λ. Each neural network was exposed to 10,000 noisy thumbnail images for 500 time steps each, after which the final reconstruction was taken. After all 10,000 images had been reconstructed, the average reconstruction error and average fraction of active neurons were calculated via Eqs. (4.15) and (4.16). The (green) upside down triangles represent networks with $S = 32$ sublayers and $N = 3 \cdot 2^{13}$ neurons. The (blue) squares represent networks with $S = 64$ sublayers and $N = 3 \cdot 2^{14}$ neurons. The (black) circles represent networks with $S = 128$ sublayers and $N = 3 \cdot 2^{15}$ neurons. The (red) triangles represent networks with $S = 256$ sublayers and $N = 3 \cdot 2^{16}$ neurons. The (purple) diamonds represent networks with $S = 512$ sublayers and $N = 3 \cdot 2^{17}$ neurons. For each system size there is a characteristic value of average fraction of active neurons for which the average reconstruction error is the lowest. As the system size is increase the locations of these minima shift and decrease in depth. The relationships between system size and the depth and location of these minima are shown in Fig. 4.9 (a) and (b), respectively.



(a) Height of minima vs. system size

(b) Location of minima vs system size

Figure 4.9: The power law behavior of the minimum average percent reconstruction error (a), and the fraction of active neurons at that minimum (b). Both (a) and (b) are plotted on log-log scales, and are plotted against the network system size. The best fit of the data is plotted on both as a (black) dashed line. The best fit of the minimum average percent reconstruction error in (a) follows a power law with exponent -0.00745. The best fit of the average fraction active neurons in (b) follows a power law with exponent -0.756. We then extracted exponents from these fits via Eqs. (4.19) and (4.20). These extracted exponents are summarized in Tab. 4.1 .

4.5 Discussion

The existence of phase transitions in neural networks is not unique to this sparsely coding convolutional system. The auto-associative network proposed by [64] was shown by [63] to display a first-order phase transition in its memory capacity. If the number of patterns recorded by the network exceeds a "critical fraction" of the network size, the output of the network is maximally disordered [63].

We propose a similar mechanism is responsible for the observed continuous phase transition of our sparsely coding convolution network, where the fraction of active neurons is analogous to the "critical fraction" of learned patterns in the auto-associative network. If our network's fraction of active neurons is too far above the "critical fraction," the network will have the freedom to reconstruct the noise in the image, while if the fraction of active neurons is too low, the network will only reconstruct image components for which it has learned strong priors. These two different regions of dynamics form our "phases." The existence of a phase transition in the average percent reconstruction error of the network as the fraction of active neurons is varied guarantees the persistence of the power law behavior seen in Fig. 4.9 (b). This power law behavior allows us to predict the optimal fraction of active neurons for any system size, which in turn can be tuned to through the parameter λ, as described by [18], to ensure that any sparsely coding convolutional network is operating at the optimal level of sparsity.

The critical behavior of the network allows us to always achieve the minimum denoising error by operating the network at this critical value of sparsity.

Chapter 5

Conclusions

In this thesis we have presented three separate systems that demonstrate a wide range of non-equilibrium dynamics and serve as examples in the analysis of such non-equilibrium systems.

In Chapter 2, we modeled a surface plasmon resonance cell using Monte Carlo simulations to evaluate the applicability of a simplified mean-field approximation of the system. We simulated the system using the parameter values listed in Tab. 2.1, along with many different values of association and dissociation rates. We then compared the rates extracted via the sensogram metrics defined in Tab. 2.4 using a simple mean-field approximation with the actual simulation values.

The predictions of the sensogram metrics were close to the actual simulation values for Damköhler number $Da < 0.1$; yet beyond $Da \gtrsim 0.1$ the system becomes diffusion-limited as the association rate becomes sufficiently large that diffusive transport begins to dominate the time scale on which ligands interact with receptors, and the probability of ligand rebinding events becomes very high. By ignoring these spatio-temporal correlations, the mean-field predictions start to drastically differ from the simulation parameters; in some cases by several orders of magnitude.

Further work in this project would entail investigating the effects of different topologies on the ligand-receptor rebinding correlations. The SPR cell is constructed with a uniform

receptor density on the bottom of its flow cell, but many biological systems such as cells display different receptor topologies, as receptors often appear in clusters on cell walls. Such clustering should increase the likelihood of ligand-receptor rebinding events, such that the ligand would remain on the receptor surface for longer periods of time than would be predicted from their binding rates. The extended interactions caused by these spatio-temporal correlations between the ligands and receptors should further distance the physical behavior of these systems from a simple mean-field prediction.

This work demonstrates a common failing in the analysis of non-equilibrium systems via a mean-field approximation. Such an approximation will ignore spatio-temporal correlations and fluctuations, which can be very important to the dynamics of non-equilibrium systems. Oftentimes the only way to fully capture these dynamics is to numerically simulate the system.

In Chapter 3 we describe biological neural networks using a model originally proposed by Lomardi, Herrmann, de Arcangelis et al. to characterize the dynamics of these networks as the inhibitory neuron fraction is varied.

We observed the avalanche strength and duration distributions for networks at a biologically accurate inhibitory fraction $p_{inh} = 0.30$ follow power laws $P_S(S) \sim S^{-1.55}$ and $P_D(D) \sim D^{-2.1}$ respectively, which corroborates experimental results, as well as previous observations of Lomardi, Herrmann, de Arcangelis et al. As we lowered the inhibitory fraction of our network towards an inhibitory fraction of 0.04, we observed behavior suggestive of criticality as the exponential cut-offs present previously in the avalanche strength and duration distributions disappeared, and these distributions continued to follow power laws for several more decades. At this low value of inhibitory fraction the network becomes dominated by incredibly strong and long-lasting avalanches that persist for several orders of magnitude more time steps than avalanches in a network with a higher inhibitory fraction. Intriguingly, the particular value

of inhibitory fraction at which we see this extension of the distributions is far below the fraction found in human cortices, which is closer to $0.2 - 0.3$ [15, 48].

Additionally, the power spectral density of our network at low inhibitory fractions ($p_{inh} = 0.04$) behaves similarly to power spectral densities of epileptic humans by following a power law with exponent -2.0. However, as the inhibitory fraction of the network is increased to a more biologically relevant value of 0.3 the power law regime of the PSD shrinks, and there is only a small regime of possible power law behavior with exponent -1.0, which is in the regime of observed exponents for healthy human brains [46]. The transition of the exponent between "epileptic" and "healthy" regimes reproduced results previously observed by Lombardi, Herrmann, De Arcangelis et al.

The exponential cut-offs present at "healthy" fractions of inhibitory neurons ($\sim 0.20 - 0.30$) protect the network from these incredibly large avalanches, and human cortices may naturally operate at higher inhibitory fractions in order to avoid a truly critical point, yet still benefit from wide distributions at lower intensity avalanche events.

We also observed how the outgoing connectivity distribution of our networks changed as the networks evolved under the Hebbian learning rules. After $45,000$ avalanches, networks with a high inhibitory fraction ($p_{inh} = 0.30$) prune away many connections, as the system is unable to propagate avalanches strong enough to sustain all of the links. This results in a truncation of the tail of the connectivity distribution, while the head of the distribution becomes inflated, as many neurons end up having only one or zero outgoing connections. Networks with a lower inhibitory fraction ($p_{inh} = 0.04$) prune away fewer connections than networks with a higher inhibitory fraction, and retain more of their initial connectivity, because these networks are able to sustain stronger avalanches. Additionally, because these networks prune their connections less intensely, they display an order of magnitude fewer neurons with zero or only one connection than the networks that evolved under the effects

of a higher inhibitory fraction.

The combination of the suppressive abilities of inhibitory neurons and the Hebbian rules of the system induce networks with high inhibitory fractions to evolve into more sparsely connected networks than networks with a low inhibitory fraction. These differences in connectivity reinforce the networks' ability to sustain or disrupt very large avalanches.

Finally we investigated two strategies to curtail and control the extremely large avalanches present in the low inhibitory fraction networks, through either the disabling of randomly selected or carefully chosen highly connected excitatory neurons, respectively.

In order to curtail these large events through random disablings, we found it necessary to disable at least 30% of the network's excitatory neurons. This strategy is ultimately effective at stopping these large avalanche events, but is quite destructive to the normal operation of the network in the process, and would be devastating to any biological neural network. In contrast, targeting and disabling the most highly connected excitatory neurons proved to be a much more efficient and less destructive strategy, as only the top 1% of these prominently connected excitatory neurons need to be disabled in order to prevent such large-scale avalanche events. Both of these strategies provide a means to circumvent the inherent occurrence of incredibly large "epileptic" avalanches in systems with very low inhibitory neuron fraction.

The next step in this work will involve investigating the efficacy of time-dependent or periodic disablings. The afore mentioned results are taken from networks where excitatory neurons are disabled permanently and are effectively "killed." This effectively controls extreme avalanche events, but is destructive to the rest of the system's dynamics. A better solution would be to disable these neurons only for as long as necessary to curtail these extreme avalanches, and then re-enable them. The cut-off present in "healthy" networks' duration distributions

at $D = 3 \cdot 10^2$ time steps might provide the time scale for a periodic disabling of neurons, in which a fraction of random or targeted excitatory neurons are disabled and re-enabled. Ideally this will have the same effectiveness in controlling the extreme avalanche events in these systems, while being less destructive to the entire system's dynamics as portions of the networks will never be permanently removed.

Finally, in Chapter 4 we used the neural network software package PetaVision [59] to simulate a sparsely encoding convolutional neural network and denoise very noisy images. We observed a minimum in the plot of denoising error vs. fraction of active neurons in the network and investigated the criticality of this point by performing a finite-size scaling analysis.

We denoised 10,000 noisy images with many different networks each trained at a particular value of fraction active neurons and system size, and observed the depth and location of this minimum to decrease according to two respective power laws described in Eqs. (4.19) and (4.20). We extracted finite-size scaling exponents $\bar{\nu} = 1.32 \pm 0.04$ and $\bar{\gamma} = 0.0099 \pm 0.0095$, which govern how the depth and location of the minimum reconstruction error change with the network's system size.

This power law behavior is suggestive of a second-order phase transition that the system goes through as the fraction of active neurons is varied.

Additionally, because the critical point is the point of highest denoising accuracy of the network, we can use the finite-size scaling laws inherent to this critical behavior to tune a network with any given system size to operate with maximal efficiency.

In this way we can use the machinery developed to analyze critical phenomena in non-equilibrium systems to optimize and control this computational system.

Further work on this project would involve the investigation of correlation lengths in this system. As systems undergoing continuous phase transitions approach a critical point, cor-

relation lengths diverge. In this sparsely encoding convolutional neural network, neurons inhibit each other in their attempts to compete for representation of a certain spatial patch of their input. As they compete neurons that strongly represent their given patch will set up an "exclusion zone" around them where other similar neurons will be suppressed. The size of this exclusion zone should correspond to some length scale of the system, which perhaps is related to the correlation length. If this system is truly displaying critical behavior then this correlation length should diverge as it approaches its critical point.

Non-equilibrium systems are near-omnipresent in physics, and no one framework exists for their analysis. At best there are regimes in systems which display universal behavior, but even their identification often requires a computational analysis.

These three very different models demonstrate, but do not encompass, the range of possible dynamics in non-equilibrium systems as well as common failings and methods in their analyses. In some systems, such as the surface plasmon resonance cells, mean-field approximations are commonly applied, which ignore significant and highly relevant portions of the systems' dynamics outside of certain regimes. In other cases even a mean-field analysis is intractable, and a computational approach must be taken to analyze system dynamics. And in certain purely computational systems, machinery developed for non-equilibrium systems can be applied to optimize these systems.

Bibliography

[1] S. Ornes. How nonequilibrium thermodynamics speaks to the mystery of life. PNAS 114, 423 (2017).

[2] U. C. Täuber. Critical dynamics: a field theory approach to equilibrium and non-equilibrium scaling behavior. Cambridge: Cambridge University Press, (2014). ISBN 9780521842235.

[3] J. Cardy. Scaling and renormalization in statistical physics. Cambridge: Cambridge University Press, (1996). ISBN 0521499593.

[4] H. Hinrichsen. Non-equilibrium phase transitions. Physica A: Statistical Mechanics and its Applications 369, 1 (2006).

[5] G. L. Fain. Molecular and Cellular Physiology of Neurons, Second Edition. Cambridge: Harvard University Press, (2014). ISBN 0674599217.

[6] R. F. Schmidt and G. Thews. Human Physiology, Second Edition. Berlin: Springer, (1989). ISBN 9783642738319.

[7] K.-L. Du and M. N. S. Swamy. Neural Networks and Statistical Learning. London: Springer-Verlag (2014). ISBN 9781447155706.

[8] J. Carroll, M. Raum, K. Forsten-Williams, and U. C. Täuber. Ligand-receptor binding kinetics in surface plasmon resonance cells: a Monte Carlo analysis. Physical Biology 13, 066010 (2016).

[9] D. Edwards. Estimating rate constants in a convection-diffusion system with a boundary reaction. IMA Journal of Applied Mathematics 63, 89 (1999).

[10] R. Schasfoort (ed.) and A. Tudos (ed.). Handbook of Surface Plasmon Resonance. Cambridge: The Royal Society of Chemistry, (2008). ISBN 1782627308.

[11] J. Carroll. A. Warren, and U. C. Täuber. The effects of inhibitory and excitatory neurons on the dynamics and control of avalanching neural networks. Physical Review E, (under review).

[12] F. Lombardi, H. J. Herrmann, C. Perrone-Capano, D. Plenz, and L. De Arcangelis. Balance between Excitation and Inhibition Controls the Temporal Organization of Neuronal Avalanches. Physical Review Letters 108, 228703 (2012).

[13] L. De Arcangelis. Are dragon-king neuronal avalanches dungeons for self-organized brain activity? The European Physical Journal Special Topics 205, 243 (2012).

[14] F. Lombardi and L. De Arcangelis. Temporal organization of ongoing brain activity. The European Physical Journal Special Topics 223, 2119 (2014).

[15] F. Lombardi, H. J. Herrmann, and L. De Arcangelis. Balance of excitation and inhibition determines 1/f power spectrum in neuronal networks. Chaos: An Interdisciplinary Journal of Nonlinear Science 27, 047402 (2017).

[16] J. Carroll, N. Carlson, and G. T. Kenyon. Phase Transitions in Image Denoising via Sparsely Coding Convolutional Neural Networks. NIPS 2017 workshop on Advances in Modeling and Learning Interactions from Complex Data, arXiv 1710.09875 (2017).

[17] A. Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[18] C. J. Rozell, D. H. Johnson, R. G. Baraniuk, and B. A. Olshausen. Sparse coding via thresholding and local competition in neural circuits. Neural Computation, 20, 2526 (2008).

BIBLIOGRAPHY

[19] D. Nelson and M. Cox. Lehninger Principles of Biochemistry, Fourth Edition. New York: W.H. Freeman and Company (2004). ISBN 9780716743392.

[20] D. Voet and J. Voet. Biochemistry, Fourth Edition. Hoboken, New Jersey: John Wiley & Sons, Inc. (2011). ISBN 0470570954.

[21] N. de Mol (ed.) and M. Fischer (ed.). Surface Plasmon Resonance. Berlin: Springer-Verlag (2010). ISBN 1607616696.

[22] E. M. Phizicky and S. Fields. Protein-protein interactions: methods for detection and analysis. Microbiological Reviews 59, 94 (1995).

[23] R. Rich and D. Myszka. Survey of the year 2005 commercial optical biosensor literature. J. Mol. Recognit. 19, 478 (2006).

[24] R. Rich and D. Myszka. Survey of the year 2006 commercial optical biosensor literature. J. Mol. Recognit. 20, 300 (2007).

[25] R. Rich and D. Myszka. Survey of the year 2007 commercial optical biosensor literature. J. Mol. Recognit. 21, 355 (2008).

[26] M. Gopalakrishnan, K. Forsten-Williams, T. Cassino , L. Padro, T. Ryan and U. C. Täuber. Ligand rebinding: self-consistent mean-field theory and numerical simulations applied to surface plasmon resonance studies. European Biophysics Journal 34, 943 (2005).

[27] D. G. Myszka, T. A. Morton, M. L. Doyle and I. M. Chaiken. Kinetic analysis of a protein antigen-antibody interaction limited by mass transport on an optical biosensor. Biophysical Chemistry. 64, 127 (1997).

[28] D. G. Myszka, X. He, M. Dembo, T. A. Morton, and B. Goldstein. Extending the range of rate constants available from BIACORE: interpreting mass transport-influenced binding data. Biophysical Journal 75, 583 (1998).

[29] G. Hu, Y. Gao, and D. Li. Modeling micropatterned antigen–antibody binding kinetics in a microfluidic chip. Biosensors and Bioelectronics 22, 1403 (2007).

[30] D. Schnoerr, G. Sanguinetti, and R. Grima. Approximation and inference methods for stochastic biochemical kinetics - a tutorial review. `arXiv:1608.06582` (2016).

[31] M. Gopalakrishnan, K. Forsten-Williams, M. A. Nugent, and U. C. Täuber. Effects of Receptor Clustering on Ligand Dissociation Kinetics: Theory and Simulations. Biophysical Journal 89, 3686 (2005).

[32] H. Motulsky and L. Mahan. The Kinetics of Competitive Radioligand Binding Predicted by the Law of Mass Action. Molecular Pharmacology 86, 592 (2014).

[33] S. Zeng, X. Yu, W. Law, Y. Zhang, R. Hu, X. Dinh, H. Ho, and K. Yong. Size dependence of Au NP-enhanced surface plasmon resonance based on differential phase measurement. Sensors and Actuators B: Chemical 176, 1128 (2013).

[34] T. Davis and W. Wilson. Determination of the refractive index increments of small molecules for correction of surface plasmon resonance data. Analytical Biochemistry 284, 348 (2000).

[35] M. Zourob (ed.), S. Elwary (ed.), and A. Turner (ed.). Principles of Bacterial Detection: Biosensors, Recognition Receptors and Microsystems. New York: Springer, (2008). ISBN 9780387751139.

[36] L. D. Landau and E. M. Lifshitz. Fluid Mechanics, Second Edition. Oxford: Butterworth-Heinemann, (1998). ISBN 0750627670.

[37] G. Papalia, S. Leavitt, M. Bynum, P. Katsamba, R. Wilton, H. Qiu, M. Steukers, S. Wang, L. Bindu, S. Phogat, A. Giannetti, T. Ryan, et al. Comparative analysis of 10 small molecules binding to carbonic anhydrase II by different investigators using Biacore technology. Analytical Biochemistry 359, 94 (2006)

[38] D. A. Lauffenburger and J. J. Linderman. Receptors: Models for Binding, Trafficking, and Signaling. New York: Oxford University Press (1993). ISBN 0195106636.

[39] R. W. Glaser. Antigen-Antibody Binding and Mass Transport by Convection and Diffusion to a Surface: A Two-Dimensional Computer Model of Binding and Dissociation Kinetics. Analytical Biochemistry 213, 152 (1993).

[40] P. Schuck and A. Minton. Analysis of Mass Transport-Limited Binding Kinetics in Evanescent Wave Biosensors. Analytical Biochemistry 240, 262 (1996).

[41] J. M. Oliver and R. Berlin. Distribution of receptors and functions on cell surfaces: Quantitation of ligand-receptor mobility and a new model for the control of plasma membrane topography. Philosophical Transactions of the Royal Society of London. B, Biological Sciences 299, 215 (1982).

[42] J. L. Gaiarsa, O. Caillard, and Y. Ben-Ari. Long-term plasticity at GABAergic and glycinergic synapses: mechanisms and functional significance. Trends in Neurosciences 25, 564 (2002).

[43] K. Gerrow and A. Triller. Synaptic stability and plasticity in a floating world. Current Opinion in Neurobiology 20, 631 (2010).

[44] R. C. Malenka and M. F. Bear. LTP and LTD: an embarrassment of riches. Neuron 44, 5 (2004).

[45] J. M. Beggs and D. Plenz. Neuronal Avalanches in Neocortical Circuits. Journal of Neuroscience 23, 11167 (2003).

[46] J. Yan, Y. Wang, G. Ouyang, T. Yu, Y. Li, A. Sik, and X. Li. Analysis of electrocorticogram in epilepsy patients in terms of criticality. Nonlinear Dynamics 83, 1909 (2016).

[47] S. Yu, A. Klaus, H. Yang, and D. Plenz. Scale-Invariant Neuronal Avalanche Dynamics and the Cut-Off in Size Distributions. PLOS ONE 9, 1 (2014).

[48] S. Sahara, Y. Yanagawa, D. D. M. O'Leary, and C. F. Stevens. The Fraction of Cortical GABAergic Neurons Is Constant from Near the Start of Cortical Neurogenesis to Adulthood. Journal of Neuroscience 32, 4755 (2012).

[49] S. J. Cooper. Donald O. Hebb's synapse and learning rule: a history and commentary. Neuroscience & Biobehavioral Reviews 28, 851 (2005).

[50] V. B. Priezzhev, D. V. Ktitarev, and E. V. Ivashkevich. Formation of Avalanches and Critical Exponents in an Abelian Sandpile Model. Physical Review Letters 76, 2093 (1996).

[51] V. M. Eguíluz, D. R. Chialvo, G. A. Cecchi, M. Baliki,and A. V. Apkarian. Scale-Free Brain Functional Networks. Physical Review Letters 94, 018102 (2005).

[52] M. H. Myers, E. Jolly, Y. Li, A. de Jongh Curry, and H. Parfenova. Power Spectral Density Analysis of Electrocorticogram Recordings during Cerebral Hypothermia in Neonatal Seizures. Annals of Neurosciences 24, 12 (2017).

[53] P. D. Welch. The use of fast Fourier transforms for the estimation of power spectra: A method based on time averaging over short modified periodograms. IEEE Transactions on Audio and Electroacoustics 15, 70 (1967).

[54] R. Albert, H. Jeong, and A. Barábasi. Error and attack tolerance of complex networks. Nature 406, 378 (2000).

[55] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. Reviews of Modern Physics 74, 47 (2002).

[56] D. S. Callaway, M. E. J. Newman, S. H. Strogatz, and D. J. Watts. Network Robustness and Fragility: Percolation on Random Graphs. Physical Review Letters 85, 5468 (2000).

[57] J. G. Milton. Neuronal avalanches, epileptic quakes and other transient forms of neurodynamics. European Journal of Neuroscience 36, 2156 (2012).

[58] B. A. Olshausen and D. J. Field. Sparse Coding with an Overcomplete Basis Set: A Strategy Employed by V1? Vision Research 37, 3311 (1997).

[59] Petavision. URL github.com/PetaVision/OpenPV.

[60] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Structured Sparsity through Convex Optimization. Statistical Science 27, 450 (2012).

[61] Y. Watkins, A. Thresher, D. Mascarenas, G. T. Kenyon. Sparse Coding Enables the Reconstruction of High-Fidelity Images and Video from Retinal Spike Trains. Proceedings of the International Conference on Neuromorphic Systems (ICONS), 8 (2018).

[62] V. Dotsenko. An introduction to the theory of spin glasses and neural networks. World Scientific Lecture Notes in Physics. Hackensack, New Jersey: World Scientific Publishing Company, (1995). ISBN 9810218737.

[63] J. A. Hertz, A. S. Krogh, and R. G. Palmer. Introduction to the theory of neural computation. Boston: Addison-Wesley Publishing Company, (1991). ISBN 0201515601.

[64] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences 79, 2554 (1982).

[65] P. F. Schultz, D. M. Paiton, W. Lu, and G. T. Kenyon. Replicating kernels with a short stride allows sparse reconstructions with fewer independent kernels. arXiv:1406.4205, (2014).