Dynamics and Learning in Recurrent Neural Networks

by

Xiaohui Xie

Submitted to the Department of Brain and Cognitive Sciences
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computational Neuroscience

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2002

Author .....
Department of Brain and Cognitive Sciences
May 10, 2002

Certified by ....
H. Sebastian Seung
Assistant Professor of Computational Neuroscience
Thesis Supervisor

Accepted by .....
Earl K. Miller
Chairman, Department Committee on Graduate Students

Dynamics and Learning in Recurrent Neural Networks

by

Xiaohui Xie

Submitted to the Department of Brain and Cognitive Sciences
on May 10, 2002, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computational Neuroscience

Abstract

This thesis is a study of dynamics and learning in recurrent neural networks. Many computations of neural systems are carried out through a network of a large number of neurons. With massive feedback connections among these neurons, a study of its dynamics is necessary in order to understand the network's function. In this thesis, I aim at studying several recurrent network models and relating the dynamics with the networks' computation. For this purpose, three systems are studied and analyzed in detail: The first one is a network model for direction selectivity; the second one is a generalized network of Winner-Take-All; the third one is a model for integration in head-direction systems.

One distinctive feature of neural systems is the ability of learning. The other part of my thesis is on learning in biologically motivated neural networks. Specifically, I study how the spike-time-dependent synaptic plasticity helps to stabilize persistent neural activities in the ocular motor integrator. I study the connections between back-propagation and contrastive-Hebbian learning, and show how backpropagation could be equivalently implemented by contrastive-Hebbian learning in a layered network. I also propose a learning rule governing synaptic plasticity in a network of spiking neurons and compare it with recent experimental results on spike-time-dependent plasticity.

Thesis Supervisor: H. Sebastian Seung
Title: Assistant Professor of Computational Neuroscience

Acknowledgments

This thesis would not have been possible without help and support from many people.

First of all, I thank my thesis advisor, Professor Sebastian Seung, for his constant guidance and support. He has been intensively involved in almost all work presented in this thesis, and constantly reminds me on how to do good science. His insight has inspired me in the past and will be with me in the future. He is not only a wonderful mentor in academics, but also a great friend in life. He has made a potentially painful process into an enjoyable experience of learning and exploration. I am indebted to him in many ways.

Thanks to my two wonderful collaborators, Richard Hahnloser and Martin Giese. Richard has been with me, sharing the same office, in all these years of graduate studies. Not only has he been very helpful in many aspects of researches, but also been a dear friend. Martin has collaborated with me on the recurrent network models of motion detection. I am impressed by his keen insight into problems. I also enjoyed a great deal of his friendship.

The members of the Seung Lab have also been very helpful. It is a great atmosphere for discussing scientific issues with such a diverse background of its members ranging from theoretical physicists and biologists to computer scientists. To Mark for helping with presentations and editing many of my papers. To Ila, Dezhe, Justin, Russ and Brett for many interesting discussions. To our new comers, Jen and Neville, for a lot of fun in the lab. To Ben for keeping our local network out of trouble. Mary has been extremely helpful in many aspects of the lab's daily life. I thank her and wish her and her baby the best luck.

I thank my thesis committee members, Professor Tommy Poggio, Professor Pawan Sinha and Professor Whitman Richards for their interest and encouragement.

Many thanks to my friends for their help and support. To Charles and Philip for making my life much easier and playing badminton with me. To Qusheng for his long-time friendship.

I reserve my final and greatest thanks to my parents. Without their support and

endless love, this would not have been possible. I dedicate this thesis to them.

Contents

List of Figures

Chapter 1

Preface

This thesis focuses on theoretical studies of neuroscience. The goal is to develop theoretical frameworks and computational models that elucidate principles governing the behavior of neural systems. This thesis is a study of dynamics and learning in biologically motivated recurrent neural networks.

Neural systems are extremely complex and possess a remarkable ability to learn from a constantly changing environment. Studies of neural systems have been further hampered by limited experimental data available. To deal with such situation, two principled approaches have been intensely used in the past. One is a constructionist approach: Given experimental facts about a particular neural system, construct a network that reproduces these properties and yields nontrivial and experimentally testable predictions. The other approach is to identify biologically plausible learning algorithms underlying the neural systems. Revealing these learning rules may be key to understanding the neural systems themselves.

These two approaches are very different from each other. In the constructionist approach, the network models constructed are usually much simplified for the purpose of analyzing and understanding the system. In contrast, the learning approach does not hand-wire the network structure, but rather learn it by implementing learning algorithms for a specific computational task. The advantage of the learning approach is that the synaptic connections are automatically learned and we do not need to concern about details of network connections. However, a disadvantage of learning

method is that the learned network may be too complicated to help understand basic mechanisms regarding how the computation is accomplished. A combination of both methods seems necessary to theoretical studies of neuroscience.

This thesis includes projects based on each of the two approaches outlined above. Unifying these projects is the attempt to understand how the brain computes and how the brain learns.

The layout of this thesis is as follows: In Chapter 2-4, we study three network models constructed for three different computational functions. Chapter 2 is a study of recurrent network models for direction selectivity. We suggest a type of neural activity patterns specific to the recurrent network models, which could be used to differentiate the recurrent network mechanism for accounting direction selectivity from others. In Chapter 3, we study a generalized Winner-Take-All network, and demonstrate how to wire network connections to mediate competitions between groups of neurons, rather than single neurons as in the traditional Winner-Take-All network. In Chapter 4, we propose a network model for integration in head-direction systems.

The next three chapters (5-7) investigate learning rules used for training biologically motivated neural networks. In Chapter 5, we study the spike-time-dependent synaptic plasticity and show that how it could be used to stabilize persistent neural activities in ocular motor integrator. In Chapter 6, we investigate the connections between backpropagation and contrastive Hebbian learning algorithms, and demonstrate how backpropagation could be equivalently implemented by contrastive Hebbian learning algorithm in a layered network. In Chapter 7, we introduce a synaptic learning rule by taking advantage of randomness on the spike trains of neurons.

In this thesis, each chapter is self-contained with introduction, main results and discussion included. Reading of each chapter does not need references from other chapters. To facilitate reading, I include a summary for each chapter in the following sections.

1.1 Dynamics

1.1.1 Nonlinear dynamics of direction-selective recurrent neural media

The direction selectivity of cortical neurons can be accounted for by asymmetric lateral connections. Such lateral connectivity leads to a network dynamics with characteristic properties which can be exploited for distinguishing in neurophysiological experiments this mechanism for direction selectivity from other possible mechanisms [1, 2, 3, 4, 5]. We present a mathematical analysis for a class of direction-selective neural models with asymmetric lateral connections. Contrasting with earlier theoretical studies which have analyzed approximations of the network dynamics by neglecting nonlinearities using methods from linear systems theory, we study the network dynamics with nonlinearity taken into consideration. We show that asymmetrically coupled networks can stabilize stimulus-locked traveling pulse solutions that are appropriate for the modeling of the responses of direction-selective neurons. In addition, our analysis shows that outside a certain regime of stimulus speeds the stability of these solutions breaks down, giving rise to lurching activity waves with specific spatio-temporal periodicity. These solutions, and the bifurcation by which they arise, can not be easily accounted for by classical models for direction selectivity.

1.1.2 A double-ring network model of the head-direction system

In the head-direction system, head direction of an animal is encoded internally by persistent activities of a pool of cells whose firing rates are tuned to the animal's directional heading [6, 7, 8, 9]. To maintain an accurate representation of the heading information when the animal moves, the system integrates horizontal angular head-velocity signals from the vestibular nuclei and yields an updated representation of the directional heading. Integration is a difficult computation, given that head-velocities can vary over a large range and the neural system is highly nonlinear. Previous

models of integration have relied on biologically unrealistic mechanisms, such as instantaneous change of synaptic strength, or very fast synaptic dynamics [10, 11, 12]. In this paper, we propose a new integration model with two populations of neurons, which performs integration based on the differential input of the vestibular nuclei to these two populations. We mathematically analyze dynamics of the model and demonstrate that with carefully tuned synaptic connections it can accurately integrate a large range of the vestibular input, with potentially slow synapses.

1.1.3 Selectively Grouping Neurons in Recurrent Networks of Lateral Inhibition

Winner-take-all networks have been proposed to underlie many of the brain's fundamental computational abilities [13, 14]. However, not much is known about how to extend the grouping of potential winners in these networks beyond single neuron or uniformly arranged groups of neurons. We show that competition between arbitrary groups of neurons can be realized by organizing lateral inhibition in linear threshold networks. Given a collection of potentially overlapping groups (with the exception of some degenerate cases), the lateral inhibition results in network dynamics such that any permitted set of neurons that can be coactivated by some input at a stable steady state are contained in one of the groups. The information about the input is preserved in this operation: The activity level of a neuron in a permitted set corresponds to its stimulus strength, amplified by some constant [15]. Sets of neurons that are not part of a group cannot be coactivated by any input at a stable steady state. We analyze the storage capacity of such a network for random groups, i.e., the number of random groups the network can store as permitted sets without creating too many spurious ones. In this framework we calculate the optimal sparsity of the groups (maximizing group entropy). We find that for dense inputs the optimal sparsity is unphysiologically small. However, when the inputs and the groups are equally sparse, we derive a more plausible optimal sparsity. We believe our results are the first steps toward attractor theories in hybrid analog-digital networks.

1.2 Learning

1.2.1 Spike-based learning rules and stabilization of persistent neural activity

We analyze the conditions under which synaptic learning rules based on action potential timing can be approximated by learning rules based on firing rates [16, 17]. In particular, we consider a form of plasticity in which synapses depress when a presynaptic spike is followed by a postsynaptic spike, and potentiate with the opposite temporal ordering. Such *differential anti-Hebbian plasticity* can be approximated under certain conditions by a learning rule that depends on the time derivative of the postsynaptic firing rate. Such a learning rule acts to stabilize persistent neural activity patterns in recurrent neural networks.

1.2.2 Equivalence of backpropagation and contrastive Hebbian learning in a layered network

Backpropagation [18, 19] and contrastive Hebbian learning [20, 21] are two methods of training networks with hidden neurons. Backpropagation computes an error signal for the output neurons and spreads it over the hidden neurons. Contrastive Hebbian learning involves clamping the output neurons at desired values, and letting the effect spread through feedback connections over the entire network. To investigate the relationship between these two forms of learning, we consider a special case in which they are identical, a multilayer perceptron with linear output units, to which weak feedback connections have been added. In this case, the change in network state caused by clamping the output neurons turns out to be the same as the error signal spread by backpropagation, except for a scalar prefactor. This suggests that the functionality of backpropagation can be realized alternatively by a Hebbian-type learning algorithm, which is suitable for implementation in biological networks.

1.2.3 A synaptic learning rules in networks of spiking neurons

In the past, many impressive learning algorithms have been proposed and shown great success in engineering problem solving. However, which learning scheme is used by our own brain is still largely unknown. In this project, we derive a synaptic plasticity rule based on reinforcement learning idea [25, 26]. Cortical neurons are known to fire highly irregular, roughly Poisson, spike trains. The fluctuation in firing rates of neurons correlated with a global reward signal could produce a learning rule that is easy to implement in neural systems and leads to spike-time-dependent plasticity, recently found in several neural domains [16, 17]. We show how this learning rule could be used for learning XOR computation in a network of spiking neurons.

Chapter 2

Nonlinear dynamics of direction-selective recurrent neural media

2.1 Introduction

Most classical models for the direction selectivity of cortical neurons have assumed feedforward mechanisms, such as multiplication or gating of afferent thalamo-cortical inputs (e.g. [1, 2, 27]), or linear spatio-temporal filtering followed by a nonlinear operation, like squaring (e.g. [3, 28]). More recently, the existence of strong lateral connectivity has motivated modeling studies that show that the properties of direction selective cortical neurons can also be reproduced by recurrent neural network models with asymmetric lateral excitatory or inhibitory connections [4, 5].

The relative contribution of feedforward and recurrent connectivity to the direction selectivity of cortical neurons remains an unresolved issue. In this paper we provide a different perspective by presenting a mathematical analysis of the nonlinear dynamics that arises in simple nonlinear neural networks with asymmetric recurrent connections that are driven by moving input stimuli. We show that such

[0]This chapter is based on the article with the same title published in *Physical Review E* by Xie and Giese, 051904 May 2002.

networks have a class of form-stable solutions, in the following signified as *stimulus-locked traveling pulses.* The amplitude of these traveling pulse solutions depends on the stimulus velocities because of the asymmetric recurrent interactions in the network, and therefore they are suitable for modeling the activity of direction selective neurons, as demonstrated by previous studies [29, 4, 5].

In contrast with these earlier studies, we are able to give an exact solution for the nonlinear network dynamics and to characterize the stability of the traveling pulse solutions. We find that the stability of such solutions depends on the stimulus speed, and can break down outside a certain regime of stimulus speeds. Outside this regime another class of solutions with characteristic spatio-temporal symmetry arises. Such solutions have been reported before in spiking networks [30, 31, 32, 33] and in brain slices [34, 35], and have been termed *lurching activity pulses.*

We find solutions with a similar spatio-temporal characteristics in the absence of any spiking mechanism, self-organized by the interplay between the network dynamics and the incoming time-dependent stimulus. This solution type was observed in our simulations for different types of threshold nonlinearities and over a regime of different parameters.

The bifurcation that underlies the transition between form-stable and lurching wave solutions results from the essentially nonlinear properties of the network dynamics. For this reason, it is crucial that in our mathematical analysis we take the threshold nonlinearity of the neurons into account. This contrasts our work with previous studies that have presented approximate analyses of similar recurrent network models by applying methods from linear systems theory [4, 29, 36].

Our mathematical analysis extends and combines methods that have been presented in the literature before [37, 38, 39, 40, 41], and applies them to a new solution class. The characteristic instability and lurching solutions seem to be difficult to account for on the basis of the classical models for direction selectivity. This leads us to conclude that the existence of lurching activity pulses provides an experimentally testable prediction that is very specific for the explanation of direction selectivity by asymmetric lateral connections.

2.2 Basic model

Dynamic neural fields have been repeatedly proposed as models for the average behavior of a large ensembles of neurons [42, 38, 43, 44, 39, 45]. The scalar neural activity distribution $u(x,t)$ characterizes the average activity at time t of an ensemble of functionally similar neurons that code for stimulus feature x. Using a continuous approximation of biophysically spatially discrete neuronal dynamics, it is in some cases possible to treat the nonlinear neural dynamics analytically.

The field dynamics of the neural activation variable $u(x,t)$ of our model is described by:

$$\tau\frac{\partial u(x,t)}{\partial t} + u(x,t) = \int_\Omega w(x-x')f(u(x',t))\mathrm{d}x' + b(x,t). \tag{2.1}$$

The left side of this equation models a leaky integrator with a total input that is given by the right hand side of the equation. This input signal includes a feedforward input term $b(x,t)$ and a feedback term that integrates the recurrent contributions from other laterally connected neurons. The *interaction kernel* $w(x-x')$ characterizes the average synaptic connection strength between the neurons coding position x' and the neurons coding position x. f is the *activation function* of the neurons. This function is nonlinear and monotonically increasing. It introduces the nonlinearity that makes it difficult to analyze the network dynamics.

In the following we consider stimuli with a constant activity profile that move at a constant velocity v. We study how the solutions of the network dynamics, and in particular how their stability changes when the stimulus speed v is varied.

In the presence of a stimulus that moves with a constant velocity v, the mathematical description of the dynamics can be simplified by using a moving frame of coordinates by changing the spatial variable to $\xi = x - vt$. In this new frame the stimulus is stationary: $B(\xi) = b(x,t)$. With the activity in the new frame $U(\xi,t) = u(x,t)$ the dynamics is

$$\tau\frac{\partial U(\xi,t)}{\partial t} - \tau v\frac{\partial U(\xi,t)}{\partial \xi} + U(\xi,t) = \int_\Omega w(\xi-\xi')f(U(\xi',t))\mathrm{d}\xi' + B(\xi). \tag{2.2}$$

A stationary solution in the moving frame has to satisfy

$$-\tau v\frac{\mathrm{d}U^*(\xi)}{\mathrm{d}\xi} + U^*(\xi) = \int_\Omega w(\xi-\xi')f(U^*(\xi'))\mathrm{d}\xi' + B(\xi). \tag{2.3}$$

$U^*(\xi)$ corresponds to a traveling pulse solution with velocity v in the original static coordinates. Therefore, the traveling pulse solution driven by the moving stimulus can be found by solving Eq. (2.3). The stability of the traveling pulse can be studied by perturbing the stationary solution in Eq. (2.2).

The neural field dynamics Eq. (2.2) is a nonlinear integro-differential equation. In most cases an analytic treatment of such equations is impossible. In this paper, we consider two biologically inspired special cases for which an analytical solution can be found. For this purpose we consider only one-dimensional neural fields and assume that the nonlinear activation function f is either a step function, or a linear threshold function.

2.3 Step activation function

We first consider the step activation function $f(z) = \Theta(z)$ where $\Theta(z) = 1$ when $z > 0$ and zero otherwise. This form of activation function approximates the activities of neurons which, by saturation, are either active or inactive. For the one-dimensional case, we assume that only a single stationary excited regime with $(U^*(\xi) > 0)$ exists and is located between the points (ξ_1^*, ξ_2^*). The validity of this assumption depends on the shape of the input $B(\xi)$ and the interaction kernel w [1]. Only neurons inside this regime contribute to the integral. Moreover, because f is constant in this regime this contribution only depends on the boundary values ξ_1^* and ξ_2^*. Accordingly, the spatial shape $U^*(\xi)$ of the stationary solution obeys the ordinary differential equation:

$$-\tau v\frac{\mathrm{d}U^*(\xi)}{\mathrm{d}\xi} + U^*(\xi) = W(\xi-\xi_1^*) - W(\xi-\xi_2^*) + B(\xi), \tag{2.4}$$

[1] Our analysis can be generalized to the case with multiple excited regimes resulting in more complex equations. Only neurons inside the excited regime contribute to the integral.

where the function $W(\cdot)$ satisfies $W'(x) = w(x)$. The solution of the last equation can be found by treating the boundaries ξ_1^* and ξ_2^* as fixed parameters and solving Eq. (2.4). To facilitate notation we define the following integral operator O with parameter $\alpha \neq 0$:

$$O[g(z);\alpha] \equiv \int_{z_0}^{z} g(m)\exp[(z-m)/\alpha]\mathrm{d}m, \tag{2.5}$$

where $z_0 = -\infty$ for $\alpha < 0$ and $z_0 = +\infty$ for $\alpha > 0$. Using this operator we define two functions $F(z) = O[W(z);\tau v]/(-\tau v)$ and $G(z) = O[B(z);\tau v]/(-\tau v)$. The solution of Eq. (2.4) can be written with these functions in the form:

$$U^*(\xi) = F(\xi - \xi_1^*) - F(\xi - \xi_2^*) + G(\xi). \tag{2.6}$$

For the boundary points, $U^*(\xi_1^*) = U^*(\xi_2^*) = 0$ must be satisfied, leading to the transcendental equation system:

$$-F(0) + F(\xi_1^* - \xi_2^*) = G(\xi_1^*) \tag{2.7}$$

$$F(0) - F(\xi_2^* - \xi_1^*) = G(\xi_2^*), \tag{2.8}$$

from which ξ_1^* and ξ_2^* can be determined. To be consistent with our initial assumption, it has to be verified that the solution $U^*(\xi)$ indeed has only one excited regime between ξ_1^* and ξ_2^*.

2.3.1 Stability of the traveling pulse solution

The stability of the traveling pulse solution can be analyzed by perturbing the dynamics around the stationary solution in the moving frame. To consider the step threshold nonlinearity in the dynamics, we perturb both the waveform and the boundary points. In addition, the perturbation of the boundary points can be related to the perturbation of the waveform at the boundary points. Based on this, we determine the

eigenvalue equation for the linearized perturbation dynamics,

$$[K(0) - c_1^*(1+\tau\lambda)][K(0) + c_2^*(1+\tau\lambda)] = K(\xi_1^* - \xi_2^*)K(\xi_2^* - \xi_1^*), \qquad (2.9)$$

where $c_i^* \equiv \mathrm{d}U^*(\xi_i)/\mathrm{d}\xi$ for $i = 1, 2$, and the function $K(\cdot)$ is defined as

$$K(z) \equiv O[w(z); \tau v/(1+\tau\lambda)](1+\tau\lambda)/(-\tau v).$$

From this equation eigenvalues λ can be found numerically. The traveling pulse solution is asymptotically stable only if the real parts of all eigenvalues λ are nonpositive. The detailed derivation of the eigenvalue equation is shown in Appendix 2.5.1.

2.3.2 Simulation results of step activation function model

In the previous analysis the only restriction for the interaction kernel was that it should allow solutions with a single excited regime. To test our mathematical results we simulated the model using an interaction function that was given by a difference of two exponential functions, simulating a receptive field with asymmetric local excitation and center-surround inhibition. Lateral connectivity of similar type, but typically symmetric with respect to the receptive field center, has been used in many models for short range interactions in the visual cortex. The advantage of using exponentials is that one can carry out the integration in Eq. (2.5) explicitly, which simplifies the subsequent calculations considerably.

We simulated the dynamics numerically and compared the results with the results from the mathematical analysis. The kernel had the following form

$$w(x) = a_e \exp(-k_e|x - x_0|) - a_i \exp(-k_i|x - x_0|),$$

where a_e and a_i are the amplitudes of excitation and inhibition. x_0 is an offset that causes the network to be asymmetric and induces the direction sensitivity.

As stimulus $b(x,t)$ we used a moving "bar" with constant width and amplitude. Fig. (2-1) plots a snapshot of the activity profile of $u(x,t)$ and stimulus $b(x,t)$ at

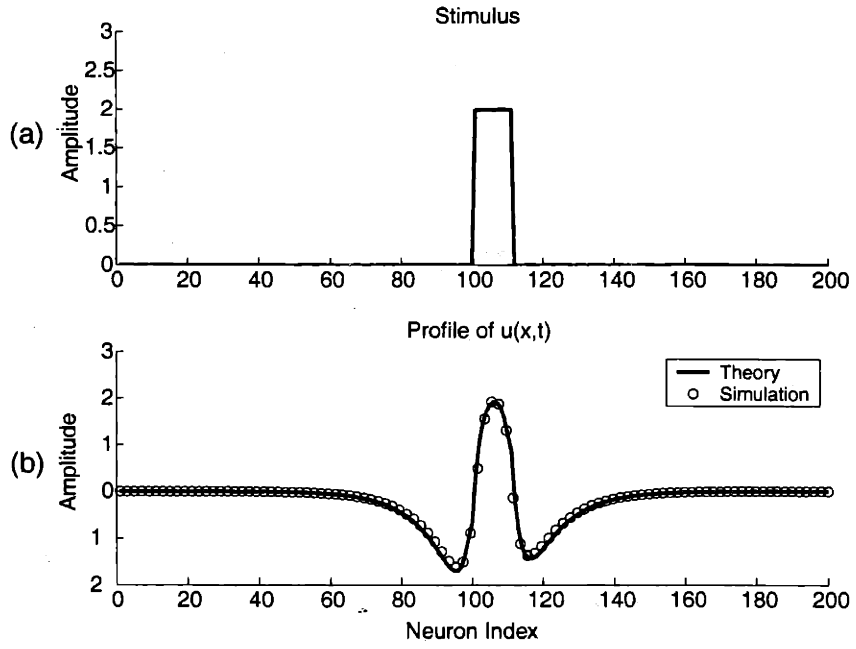

Figure 2-1: Stimulus and activity profile in the step activation function model. Panel (a) shows the stimulus, and Panel (b) the activity $m(x,t)$ at the time t for the traveling pulse solution. The solid line in (b) shows the result from the calculation, while the circles indicate the numerical simulation results. The interaction kernel used in this simulation was $w(x) = a_e \exp(-k_e|x - x_0|) - a_i \exp(-k_i|x - x_0|)$ with $a_e = 1, a_i = 5, k_e = 0.42, k_i = 0.1$ and $x_0 = 3$. The stimulus was a moving bar with width $d = 10$ and amplitude $h = 2$. Notice that the activity profile $u(x,t)$ has only a single excited regime.

a time t in the regime where the traveling pulse solution is stable. On top of the analytically calculated profile $u(x,t)$, we also plotted simulation results, which show good consistency with the theory.

We also determined the peak activities of $u(x,t)$ as function of the stimulus speed. The peak amplitude as a function of the speed is shown in Fig. (2-2). Panel (a) shows the speed tuning curve plotted as the dependence of the peak activity of the traveling pulse as a function of the stimulus velocity v. The solid line indicates the results from the theoretical solution and the dots indicate the simulation results. Panel (b) shows the maximum of the real parts of the eigenvalues obtained from Eq. (2.9). For stimulus velocities outside a certain range this maximum becomes positive indicating

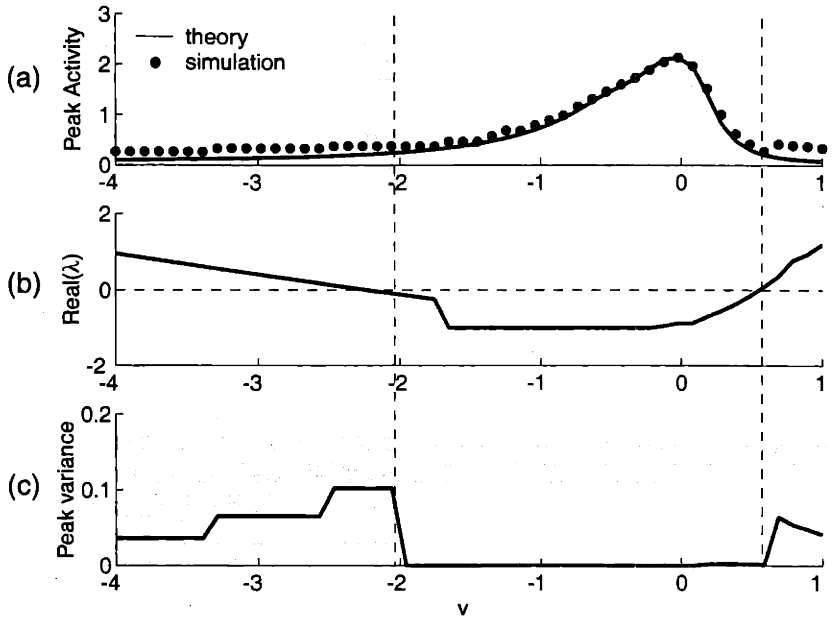

Figure 2-2: Traveling pulse solution and its stability in the step activation function model. Panel (a) shows the velocity tuning curves and the peak amplitude of the traveling pulse. The solid lines indicate the theoretical results, while the dots signify the numerical simulation results. The velocity v is normalized by the time constant of the dynamics in the unit of rad/τ. Panel (b) shows the largest real parts of the eigenvalue λ obtained by solving Eq. (2.9) numerically. Only solutions corresponding to the negative values of this function are form-stable. Panel (c) plots the variations of the peak amplitude of the pulse. A variance that deviates significantly from zero signifies a loss of stability of the traveling pulse solutions. the results are consistent with analysis of the eigenvalues in Panel (b). Also notice that in Panel (a) the theoretical peak amplitude fits well the simulation results only inside the stable regime.

a loss of stability of the form-stable solution. To verify this result we calculated also the variability of the peak activity over time after excluding the initial transients from the simulations. Panel (c) shows the variations as function of the stimulus velocity. At the velocities for which the eigenvalues indicate a loss of stability the variability of the amplitudes suddenly increases. This indicates that the stationary solution is not time-independent any more, consistent with our interpretation that the form-stable solution loses stability.

An interesting observation is illustrated in Fig. (2-3) that shows the space-time evolution of the activity. The left panel shows the propagation of the form-stable

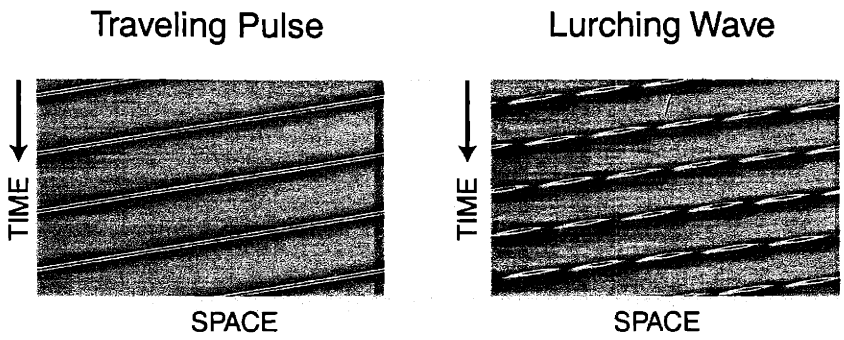

Figure 2-3: Traveling pulse and lurching wave in step activation function model. The color-coded plots show the spatial-temporal evolution of the activity $u(x,t)$. The left panel shows the propagation of the form-stable peak over time. The right panel shows the lurching activity wave that arises when stability is lost.

traveling pulse for medium stimulus speeds. The right panel shows the solution that arises when stability is lost. This solution is characterized by a characteristic spatio-temporal periodicity that is defined in the moving coordinate system by $U(y+mL_0, t+nT_0) = U(y,t)$, where L_0 and T_0 are constants that depend on the network dynamics. Solutions of similar type have been described before in different contexts, such as in brain slice experiments [34, 35] and in studies with spiking networks without time-dependent input signals. These solutions have been termed "lurching waves" because of the periodic discontinuity of the spatio-temporal evolution of the neural activity [46, 32, 31].

We have shown here only the comparison between theory and simulation for exponential interaction kernels and localized bar stimuli. However, we found in additional simulation studies that lurching activity waves arise very robustly for this type of networks also for other forms of interaction kernels or input signals. Further evidence for the robustness of the phenomenon of lurching waves is provided in the following by a demonstration that the same phenomenon arises also for another type of threshold function.

2.4 Linear threshold model

We also considered a model with an activation function f that had the form of a linear threshold, i.e. $f(z) = [z]^+ = \max\{z, 0\}$. Linear threshold models of similar type have been used before in a variety of neural modeling studies [43, 39, 47]. It has been argued that firing rates of neurons above threshold typical vary linearly with the stimulus strength. Moreover, neurons normally operate far below their saturation levels. Therefore, a linear threshold characteristic might approximate the activation function relatively well (cf. e.g. [48]). To further simplify the model, we consider a ring network with periodic boundary condition on the interval $\Omega = [-\pi, \pi)$.

The ring network dynamics can be written as

$$\tau \frac{\partial}{\partial t} m(\theta, t) + m(\theta, t) = \left[\int_{-\pi}^{\pi} w(\theta - \theta') m(\theta', t) \, (2\pi)^{-1} \mathrm{d}\theta' + b(\theta, t) \right]^+, \qquad (2.10)$$

where $b(\theta, t)$ is the time-dependent feedforward input.

The network in this form can be transformed to the network in the standard form that is given by Eq. (2.1) by a change of variables and by transforming the stimulus distribution. Defining the total network input $u(\theta, t)$ by

$$u(\theta, t) \equiv \int_{-\pi}^{\pi} w(\theta - \theta') m(\theta', t) \, (2\pi)^{-1} \mathrm{d}\theta' + b(\theta, t), \qquad (2.11)$$

we obtain the following dynamics for u

$$\tau \frac{\partial}{\partial t} u(\theta, t) + u(\theta, t) = \int_{-\pi}^{\pi} w(\theta - \theta') \, [u(\theta', t)]^+ \, (2\pi)^{-1} \mathrm{d}\theta' + \tilde{b}(\theta, t), \qquad (2.12)$$

where the transformed stimulus $\tilde{b}(\theta, t)$ obeys the partial differential equation: $\tilde{b}(\theta, t) = \tau \partial b(\theta, t)/\partial t + b(\theta$

For convenience, in the following discussions we use Eq. (2.10) for the analysis of the system dynamics. As in the previous model, the stimulus moves with a constant velocity $b(\theta, t) = B(\theta - vt)$. Again, we analyze traveling pulse solutions that are driven by the stimulus, and their stability.

2.4.1 General solutions and stability analysis

Because the activation function has linear threshold characteristics, inside the excited regime for which the total input ($u(\theta,t) > 0$) is positive the system is linear. One approach to solve this dynamics is therefore to find the solutions to the differential equation assuming the boundaries of the excited regime are given. The conditions at the boundaries lead to a set of self-consistent equations for the solutions to satisfy, from which the boundaries can be determined.

By denoting activities in moving coordinates as $M(\theta - vt, t) = m(\theta, t)$, the dynamics can be written as:

$$\tau\frac{\partial}{\partial t}M(\theta,t) - \tau v\frac{\partial}{\partial\theta}M(\theta,t) + M(\theta,t) = \left[\int_{-\pi}^{\pi} w(\theta-\theta')M(\theta',t)\,(2\pi)^{-1}\mathrm{d}\theta' + B(\theta)\right]^{+}. \tag{2.13}$$

Supposing the excited regime is $\theta \in (\theta_1(t), \theta_2(t))$, we solve the dynamics by Fourier transforming the above equation in the spatial domain $[-\pi, \pi)$. Let

$$\hat{m}_n(t) = \int_{-\pi}^{\pi} M(\theta,t)\exp(in\theta)\,(2\pi)^{-1}\mathrm{d}\theta \quad \text{and} \quad \hat{w}_n = \int_{-\pi}^{\pi} w(\theta)\exp(in\theta)\,(2\pi)^{-1}\mathrm{d}\theta,$$

Then in terms of these Fourier modes, the dynamics can be written as

$$\tau\dot{\hat{m}}_n + (1 + i\tau vn)\hat{m}_n = \sum_l C_{nl}\hat{m}_l + \hat{b}_n,$$

for $n = 0, \pm 1, ...,$ with

$$\begin{aligned} C_{nl} &= (2\pi)^{-1}\hat{w}_l[(\theta_2-\theta_1)\delta_{nl} - i(e^{i(n-l)\theta_2} - e^{i(n-l)\theta_1})(n-l)^{-1}(1-\delta_{nl})] \\ \hat{b}_n &= \int_{\theta_1}^{\theta_2} B(\theta)\exp(in\theta)\,(2\pi)^{-1}\mathrm{d}\theta. \end{aligned}$$

where δ_{nl} is the Kronecker delta defined as having the value one when $n = l$, and zero when $n \neq l$.

Therefore, the stationary solution in moving coordinates is

$$\hat{\mathbf{m}}^* = (I + i\tau vK - C)^{-1}\hat{\mathbf{b}}, \tag{2.14}$$

where matrix K is defined as the diagonal matrix $K \equiv \mathrm{diag}([0, 1, -1, 2, -2, ...])$. The components of the vector $\hat{\mathbf{m}}$ are $\hat{m}_n$, and those of $\hat{\mathbf{b}}$ are $\hat{b}_n$. The total input for the stationary solution in the moving frame can then be written as

$$U^*(\theta) = \sum_n \exp(-in\theta) \sum_l C_{nl} \hat{m}_l^* + B(\theta), \tag{2.15}$$

which has to satisfy the two boundary conditions $U^*(\theta_1) = U^*(\theta_2) = 0$. From these two equations the stationary values of θ_1 and θ_2 can be determined.

The stability of this traveling pulse solution can be analyzed by linear perturbation theory. Note that the perturbations of the boundary points will not contribute to the linearized perturbed dynamics because the contribution from this perturbation is $\delta\theta_i \, U^*(\theta_i) = 0$ for $i = 1, 2$. Therefore, the linearized perturbation dynamics can be fully characterized by the perturbed Fourier modes with fixed boundaries. Hence, the stability of the traveling pulse solution is determined by the eigenvalues of the matrix $A = -(I + i\tau v K - C)$. If the maximum of the real parts of the eigenvalues of A is negative, then the stimulus locked traveling pulse is stable.

2.4.2 Linear threshold network with simple kernels

The general solution introduced above requires the solution of a system of equations. In practice, the Fourier series has to be truncated in order to obtain a finite number of Fourier components at the expense of an approximation error.

Next we use a simple model that contains only the first two Fourier components in both the interaction kernel and the input distribution. We modify the model by introducing asymmetry into the interaction kernel, and study how the network activity changes as a function of the stimulus velocity. For this model, a closed form solution and stability analysis are presented that provides an insight into some rather general properties of linear threshold networks.

The interaction kernel and feedforward input are taken to be of the following form:

$$w(\theta) = J_0 + J_1 \cos(\theta + \beta) \tag{2.16}$$

$$b(\theta, t) = C(1 - \epsilon + \epsilon \cos(\theta - \theta_0(t))) - T, \tag{2.17}$$

where the variable β makes the interaction asymmetric. In the input, the threshold term T is subtracted, and $\theta_0(t) = vt$ is the input's peak location. This model was introduced by Hansel and Sompolinsky in their model of cortical orientation selectivity [39], with $w(\theta)$ being symmetric and b being static.

Since the interaction kernel and feedforward input involve only the first two Fourier components, the Fourier transform method presented in the previous section can be simplified significantly. As a consequence, the dynamics of the network can be studied in terms of the first two Fourier components of $M(\theta, t)$, namely, $\hat{m}_0(t)$ and $\hat{m}_1(t)$. Next we present the analysis, following similar treatments of Hansel and Sompolinsky [39].

The first Fourier component $\hat{m}_0(t)$ is a real number representing the mean of the neural activities, which is denoted by $r_0(t)$ in the following. The second Fourier component $\hat{m}_1(t)$ is a complex number. Let's denote the amplitude of $\hat{m}_1(t)$ by $r_1(t)$. Therefore, in summary we have

$$r_0(t) = \hat{m}_0(t) = \int_{-\pi}^{\pi} m(\theta, t)\, (2\pi)^{-1} \mathrm{d}\theta \tag{2.18}$$

$$r_1(t) = |\hat{m}_1(t)| = \int_{-\pi}^{\pi} m(\theta, t) \exp[i(\theta - \Psi(t))]\, (2\pi)^{-1} \mathrm{d}\theta, \tag{2.19}$$

where the phase $\Psi(t)$ is used to make the right hand side of the equation being a real number.

In terms of the Fourier components, the total input in Eq. (2.10) can be written as

$$\begin{aligned} I(\theta, t) &= \int_{-\pi}^{\pi} w(\theta - \theta') m(\theta', t)\, (2\pi)^{-1} \mathrm{d}\theta' + b(\theta - \theta_0(t)) \\ &= I_0(t) + I_1(t) \cos(\theta - \Phi), \end{aligned} \tag{2.20}$$

where $I_0(t)$ and $I_1(t)$ are defined as:

$$I_0(t) = C(1-\epsilon) + J_0 r_0(t) - T \tag{2.21}$$

$$I_1(t) = \epsilon C \cos(\theta_0(t) - \Phi) + J_1 r_1(t) \cos(\Psi - \Phi - \beta). \tag{2.22}$$

Here, the phase variable $\Phi(t)$ represents the location for the peak of the total input, that is, $\Phi(t) = \text{argmax}_\theta I(\theta, t)$, which should satisfy

$$\epsilon C \sin(\Psi - \theta_0(t)) + J_1 r_1 \sin(\Phi - \Psi + \beta) = 0. \tag{2.23}$$

Fig. (2-4) shows a snapshot of the network activity $m(\theta, t)$, the total input $I(\theta, t)$, and the stimulus $b(\theta, t)$ at the time t. Three phase variables are indicated in the figure, with θ_0, Ψ, and Φ being the peak location of the input, the first Fourier mode, and the total input $I(\theta, t)$ respectively.

To write down the dynamics in terms of these Fourier components, we need one more step to take care of the rectification nonlinearity. Suppose there is only a single excited interval $\theta \in (\Phi - \theta_c, \Phi + \theta_c)$ in which the total input $I(\theta, t)$ is positive. From Eq. (2.20), the critical width can be determined as $\theta_c = \arccos(-I_0/I_1)$. Using θ_c, the dynamics can be rewritten as

$$\tau \frac{\partial}{\partial t} m(\theta, t) + m(\theta, t) = I_1(t) \left[\cos(\theta - \Phi) - \cos(\theta_c)\right]^+ . \tag{2.24}$$

Fourier transforming the above equation, we derive the dynamics of the Fourier components:

$$\tau \dot{r}_0 = -r_0 + I_1(t) f_0(\theta_c) \tag{2.25}$$

$$\tau \dot{r}_1 = -r_1 + I_1(t) f_1(\theta_c) \cos(\Phi - \Psi) \tag{2.26}$$

$$\tau r_1 \dot{\Psi} = I_1(t) f_1(\theta_c) \sin(\Phi - \Psi), \tag{2.27}$$

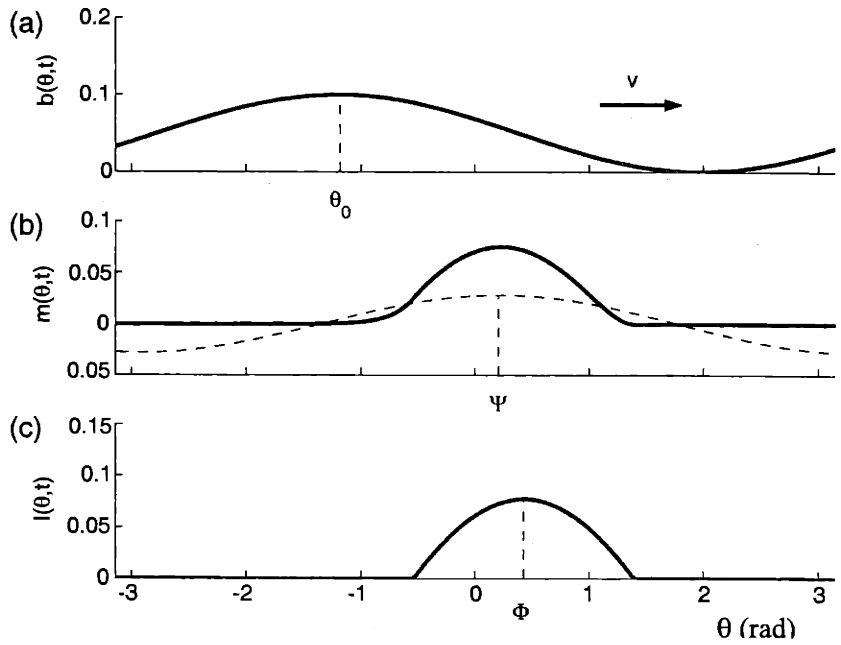

Figure 2-4: Traveling pulse for the linear threshold model with a simple periodic kernel (Eq. (2.16)). Panel (a) shows the stimulus with a moving peak centered at θ_0. The activation profile $m(\theta, t)$ is shown in Panel (b). The dashed line indicates its first order Fourier component with a maximum at Ψ. Panel (c) shows the profile of the total input $I(\theta, t)$. The phase variable Φ is defined by the peak location of the total input.

where two functions $f_0(\theta_c)$ and $f_1(\theta_c)$ are defined as

$$\begin{aligned} f_0(\theta_c) &= \pi^{-1}[\sin(\theta_c) - \theta_c \cos(\theta_c)] \\ f_1(\theta_c) &= (2\pi)^{-1}[\theta_c - \sin(2\theta_c)/2]. \end{aligned}$$

Interestingly, introducing the time-dependent input and asymmetric connections does not change the principle form of the Fourier component dynamics compared with the case with static inputs and symmetric connections [39]. Instead, the changes only appear inside $I_1(t)$ (see Eq. (2.22)). This property is very helpful for the analysis of the dynamics of this system.

Similarly, we can derive the dynamics of the Fourier components with orders higher than two. But fortunately, the dynamics in Eq. (2.25-2.27) is independent of these higher order components. Moreover, it can be shown that if the dynamics in Eq. (2.25-2.27) is stable, the dynamics of the higher order Fourier components is stable as well. Therefore, the stability of these three dimensional dynamics fully characterizes that of the neural field Eq. (2.24).

2.4.3 Traveling pulse solutions

A traveling pulse solution corresponds to a stationary solution in the moving frame. Therefore, $\dot{r}_0 = \dot{r}_1 = 0$ and $\dot{\Psi} = v$, which lead to

$$\begin{aligned} r_0 &= I_1 f_0(\theta_c) \\ r_1 &= I_1 f_1(\theta_c) \cos(\Phi - \Psi) \\ \tau v &= \tan(\Phi - \Psi). \end{aligned}$$

Suppose that θ_c is given. From the above equations, the Fourier components r_0 and r_1 can be derived as

$$r_0 = [(1-\epsilon)C - T] f_0(\theta_c)[-J_0 f_0(\theta_c) - \cos(\theta_c)]^{-1} \quad (2.28)$$

$$r_1 = [(1-\epsilon)C - T] \cos(\Delta) f_1(\theta_c)[-J_0 f_0(\theta_c) - \cos(\theta_c)]^{-1}, \quad (2.29)$$

where the variable $\Delta \equiv \Phi - \Psi = \text{atan}(\tau v)$. Subsequently, I_0 and I_1 can be determined from Eq. (2.21-2.22). Substituting them into Eq. (2.23) leads to

$$1 - \Gamma^{-1} = [J_0 f_0(\theta_c) + \cos(\theta_c)][J_1^2 f_1^2(\theta_c)\cos^2(\Delta) - 2J_1 f_1(\theta_c)\cos(\Delta)\cos(\Delta+\beta) + 1]^{-1/2}, \tag{2.30}$$

where $\Gamma \equiv \epsilon C/(C-T)$ represents the contrast of the stimulus. From this equation, the critical width θ_c can be found, using numerical methods. Consequently, the values of r_0 and r_1 can be determined.

2.4.4 Existence of traveling pulse solutions

The critical width θ_c must satisfy Eq. (2.30). The existence of traveling pulse solutions depends on whether θ_c exists for a given stimulus velocity v. It is possible that θ_c does not exist for a particular range of v. Next we characterize the conditions on v for the existence of a traveling pulse solution.

Let $B = [J_0 f_0 + \cos(\theta_c)]\Gamma/(\Gamma - 1)$. Then, Eq. (2.30) can be rewritten as

$$[J_1 f_1 \cos(\Delta) - \cos(\Delta+\beta)]^2 = B^2 - \sin^2(\Delta+\beta).$$

Therefore, for a solution θ_c to exist, we must have $|\sin(\Delta+\beta)| \leq B$. Dividing both sides by $\cos(\beta)\cos(\Delta)$, we derive the condition that v has to satisfy for the existence of θ_c

$$|v - v^*| \leq \frac{\sqrt{1+\tau^2 v^2}}{\tau\cos(\beta)} B, \tag{2.31}$$

where $v^* \equiv -\tan(\beta)/\tau$.

The above equation can not be used to determine the v for which a traveling pulse solution arises, since the right hand side of the equation depends on the unknown variable θ_c. However, it gives some general characterizations about the admissible range of v.

For example, the limit for the stimulus contrast $\Gamma \to 0$ implies the only admissible $v = v^*$, which means that the traveling pulse solution has a unique velocity v^* that is independent from the stimulus, and determined only by the network dynamics. So-

lutions of this type have been analyzed before for networks with saturating threshold functions in [10]. In this case the traveling pulse solution is caused purely by the asymmetric structure of the network, parametrized here by the variable β. When the stimulus is not uniform, the traveling pulse solution exists only when the stimulus velocity is not too different from the intrinsic velocity v^*. The smaller the contrast B, the smaller is the range of stimulus speeds v for which a traveling pulse solution exists. This range is also influenced by the time constant τ. Smaller τ lead to a larger velocity range.

2.4.5 Optimal velocity

The network presented here is asymmetric, and has its own intrinsic velocity v^* determined by the asymmetry parameter β. When the network is driven by the stimulus moving at different velocities the amplitude of the solution is modulated as a function of the velocity. This dependency defines the *velocity tuning curve*, which can be measured in physiological experiments. To characterize the velocity tuning curve fully in this network is not easy since θ_c can only be determined numerically. We focus, therefore, on finding the optimal stimulus velocity that leads to the maximal mean activity r_0.

Note that r_0 in Eq. (2.28) only depends on θ_c, but not directly on v. Furthermore, r_0 depends on θ_c only through $\cos(\theta_c)/f_0(\theta_c)$ as

$$r_0(\theta_c) = [(1-\epsilon)C - T][-J_0 - \cos(\theta_c)/f_0(\theta_c)]^{-1}.$$

For $\theta_c \in [0, \pi]$ it is easy to check that $f_0(\theta_c)$ is monotonically increasing, and consequently $\cos(\theta_c)/f_0(\theta_c)$ is monotonically decreasing. Overall, $r_0(\theta_c)$ is a monotonically decreasing function of θ_c. Therefore, the optimal velocity v^m for which r_0 is maximal corresponds to the smallest value of θ_c in Eq. (2.30), that is $v^m \equiv \text{argmax}_v r_0(\theta_c(v)) = \text{argmin}_v \theta_c(v)$.

Taking the derivative with respect to v on both sides of Eq. (2.30) and using the

condition $\mathrm{d}\theta_c(v^m)/\mathrm{d}v = 0$ yields

$$v^m = -\frac{J_1 f_1 \sin(\beta)}{\tau(1-B^2)}.$$

When the stimulus contrast is small, $\Gamma \ll 1$, we have $B \ll 1$ and $J_1 f_1 \approx 1/\cos(\beta)$. Substituting this result into the above equation, we find for weak stimulus contrast $v^m \approx v^*$. This implies that the optimal velocity for which the mean activity is maximal is the intrinsic velocity. This is a nice property in the sense that it relates the optimal stimulus velocity to the network structure. By changing the asymmetry parameter β, the network can have different preferred velocities. Notice that the approximate equality between the optimal v^m and the intrinsic v^* holds only if the stimulus contrast is low.

2.4.6 Stability analysis of the traveling pulse

A stability analysis can be carried out by perturbing the dynamics of the Fourier components in Eq. (2.25-2.27). The final linearized perturbation dynamics is shown in Appendix 2.5.2. In the case when $\epsilon C \ll 1$, the perturbed dynamics can be simplified into

$$\tau \delta\dot{r}_0 = (\pi^{-1} J_0 \theta_c - 1)\, \delta r_0 + \pi^{-1} J_1 \sin(\theta_c)\, \delta r_1 \tag{2.32}$$

$$\tau \delta\dot{r}_1 = \pi^{-1} \cos(\beta) \sin(\theta_c) J_0\, \delta r_0 + [-1 + (2\pi)^{-1} J_1 (\theta_c + \sin(2\theta_c)/2) \cos(\beta)]\, \delta r_1 \tag{2.33}$$

2.4.7 Simulation results for the linear threshold model

Fig. (2-5) shows the comparison between the results from the mathematical analysis and the simulations. Panel (a) shows the speed tuning curve plotted as values of r_0 and r_1 with respect to different stimulus velocities v. The solid and dashed lines indicate calculation results, and the dotted lines represent those from numerical simulations. Panel (b) shows the largest real part of the eigenvalues of the stability matrix obtained by linearizing the three dimensional Fourier component dynamics around

the stationary solution as described in the previous section. For stimulus velocities outside a certain range, the maximum of the real parts of the eigenvalues becomes positive indicating a loss of stability of the form-stable solution. To verify this result we calculated the variations of r_0 and r_1 over time in the simulation. Panel (c) shows the variations as a function of the stimulus velocity. At the velocities for which the eigenvalues indicate a loss of stability the variations of r_0 and r_1 suddenly increase, consistent with our interpretation.

Like the results shown before for the step function model (Fig. (2-3)), Fig. (2-6) illustrates the space-time evolution of the activity. The left panel shows the propagation of the form-stable peak over time, whereas the right panel shows the solution that arises when stability is lost. Like those in the model with a step threshold, lurching activity pulses arise for a whole regime of different parameters for networks that show substantially direction selective behavior.

The phase diagram of the form-stable traveling pulse solution is plotted in Fig. (2-7), where we show the range of stimulus velocity for a stable traveling pulse as the asymmetry parameter β, and consequently the intrinsic velocity ($v^* = -\tan(\beta)/\tau$), changes. The stable region for v is typically located around the intrinsic velocity v^*.

So far, we have shown the traveling pulse and lurching wave solutions in models with step threshold and linear threshold activation functions. The development of direction selectivity of the travel pulse solutions among certain velocity range and loss of stability when outside the range are not confined only to these two types of models. To demonstrate this, we simulate the dynamics Eq. (2.1) with a sigmoidal shaped activation function and an asymmetric interaction kernel. Again, we observe the tuning of neural activities to input velocities, and the bifurcation of traveling pulse solutions to lurching waves when the velocity of the input is outside a certain range (Fig. (2-8)).

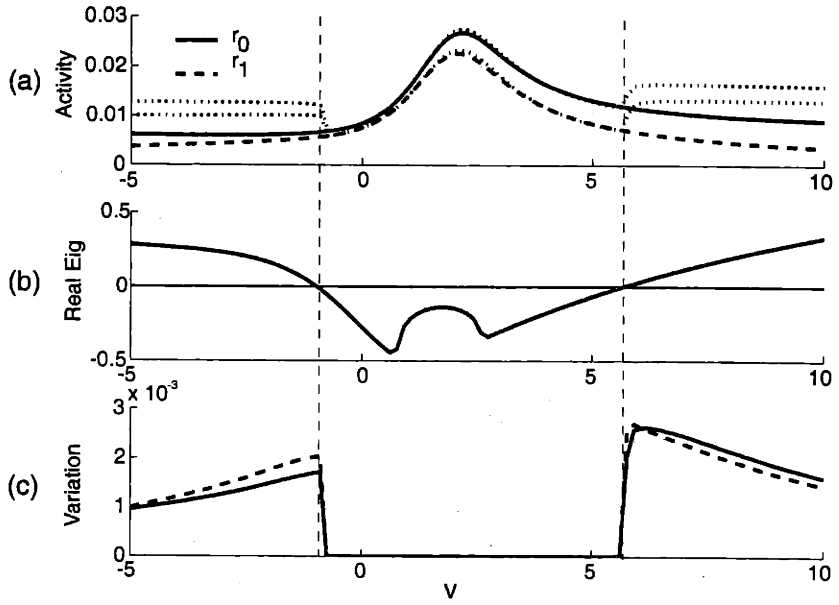

Figure 2-5: Traveling pulse solution and its stability in the linear threshold model. Panel (a) shows the velocity tuning curves of r_0 and r_1. The dotted lines indicate numerical simulation results, while solid and dashed lines are the results from our analytical solution. The theoretical results fit well the simulation results in the range of velocity between the two vertical dashed lines. Panel (b) shows the maximum of the real parts of eigenvalues of the stability matrix obtained by perturbing the dynamics around the stationary solution. For stimulus velocities outside a certain range this value becomes positive, indicating a loss of stability of the form-stable solution. Panel (c) shows the variations of r_0 (solid curve) and r_1 (dashed curve) over time determined from the simulation. A nonzero variance signifies a loss of stability for the traveling pulse solution, consistent with the eigenvalue analysis in Panel (b). The velocity v is normalized by the time constant of the dynamics in the unit of rad/τ. Parameters used are $C = 5$, $\epsilon = 0.01$, $T = 4.9$, $J_0 = -9.8$, $J_2 = 13.5$, and $\beta = 0.46$.

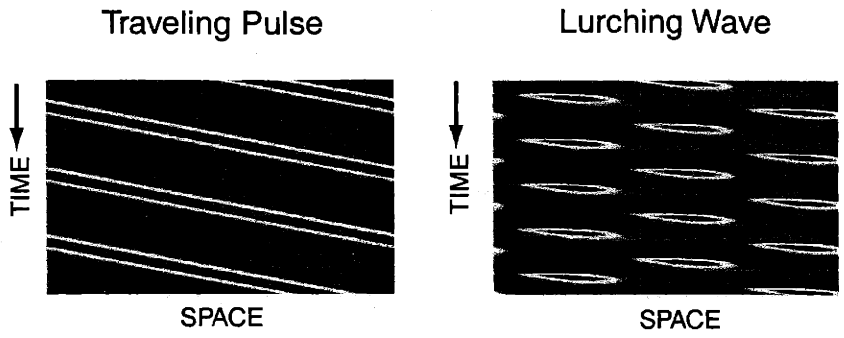

Figure 2-6: Traveling pulse and lurching wave in the linear threshold model. Shown here is a color-coded plot of spatial-temporal evolution of the activity $m(x,t)$. The left panel shows the propagation of the form-stable peak over time, whereas the right panel shows the lurching activity wave that arises when stability is lost.

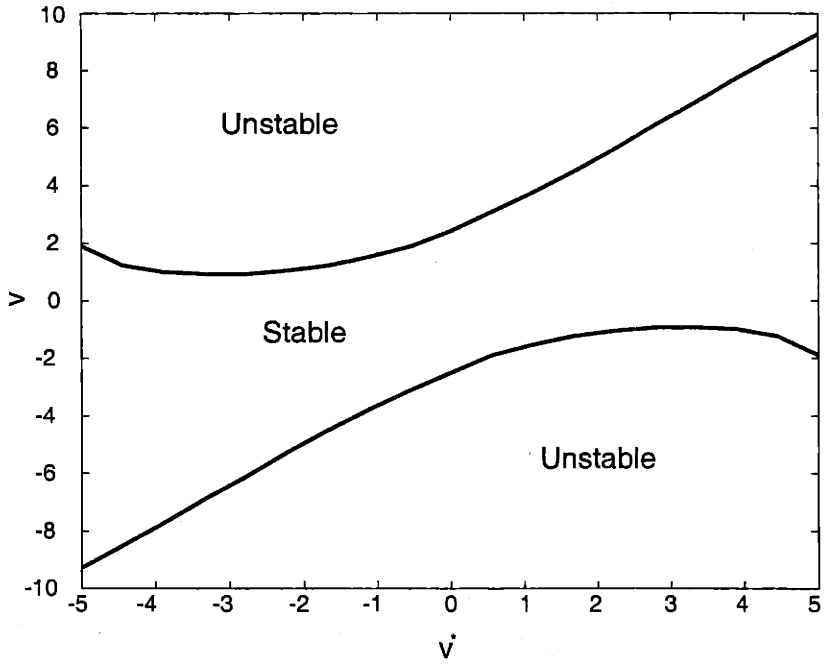

Figure 2-7: Stable regime of traveling pulse solutions. Shown here is the regime velocities v for which a stable traveling pulse solution arises as the intrinsic velocity v^* changes. The intrinsic velocity v^* depends on the the asymmetry variable β of the interaction kernel.

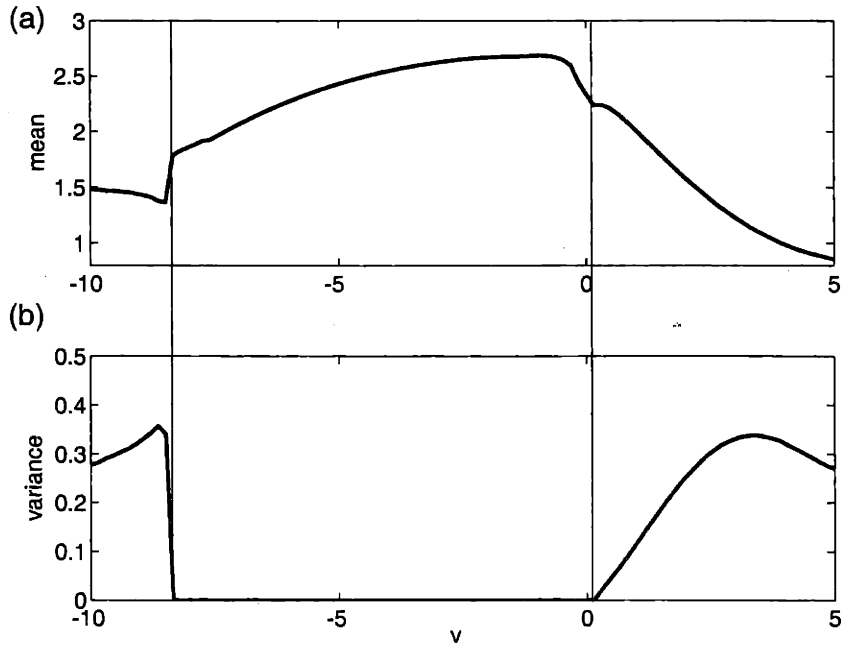

Figure 2-8: Traveling pulse solution and its stability with a sigmoidal shaped activation function. Panel (a) shows that mean peak activity of the moving solutions and Panel (b) plots the variations of the solution averaged over time. The traveling pulse solution is stable only for velocities between the two vertical lines. The velocity v is normalized by time constant τ in the unit of rad/τ. The activation function used is $f(x) = 1/(1 + \exp(-2x))$. The interaction kernel is the difference of two Gaussian functions, but with the center shifted, $w(x) = A_e \exp(-(x-\mu)^2/(2\sigma_e^2)) - A_i \exp(-(x-\mu)^2/(2\sigma_i^2))$ with $\sigma_e = 0.08$, $\sigma_i = 1$, $A_e = 62$, $A_i = 37$, and $\mu = 0.05$. The input used is a rectified bump $b(\theta, t) = 7[\exp(\cos(\theta - vt) - 2]^+$.

2.5 Conclusion

In this paper we have presented a mathematical analysis of a class of models that account for the direction selectivity by asymmetric lateral connections between cortical neurons. Given the large number of recurrent connections in the visual cortex, it seems plausible that lateral connections play an important role for the realization of direction selectivity [4, 5]. Contrasting with earlier works on such models [29, 36], we have presented a mathematical analysis of the full nonlinear dynamics of such networks that takes the nonlinear response functions of the neurons into account.

One result from our analysis is that such recurrent models, for a certain regime of stimulus speeds, have traveling pulse solutions that are form-stable and move with the same speed as the stimulus. We have termed such solutions *stimulus-locked traveling pulses*. In the stationary state, these solutions have space-time characteristics that is also compatible with other models for direction selectivity, e.g. motion energy models with feed-forward structure, or models with linear feedback. In particular, the recurrent mechanism that we analyzed can account for biologically realistic degrees of velocity tuning of cortical neurons [29]. The preferred speed of the neurons in such recurrent models is determined by the network structure. For example, we show that for the model with linear threshold activation function, the preferred speed for input signals with small contrast is close to the equilibrium speed of the self-generated traveling pulse solution in the absence of a time-dependent stimulus. The speed tuning in the nonlinear model that we analyzed arises because, for sufficiently strong interaction, the network tends to stabilize a traveling peak solution that "locks" to the moving activity peak of the stimulus. This solution becomes unstable if this locking is lost.

Our stability analysis shows that the traveling pulse solution is stable only within a certain regime of stimulus speeds. At the borders of this regime a bifurcation arises and the stimulus-locked solution becomes unstable. Such speed-dependent bifurcations can not arise in the classical feed-forward models, and in networks with linear feedback. For such networks the solutions are either always stable, or the

network is unstable.

An important observation in our simulations is that the loss of stability of the stimulus-locked solution is frequently accompanied by the formation of *lurching activity pulses.* Lurching activity has been described by different other authors in brain slices [34, 35], and in artificial spiking networks without time-dependent inputs [32, 33]. Our simulation results imply that spiking neurons are not necessary for the generation of lurching activity waves if a moving stimulus is present. Such lurching waves cannot be produced by a feedforward network, in which the output of the network is always phase-locked to the stimulus. Moreover, there is no stability issue in feedforward networks. Therefore, the bifurcation observed in recurrent networks can not appear in feedforward networks. In models with linear feedback, oscillations of the activity could potentially be obtained, e.g. if the network contains multiple neuron populations that are connected by excitatory connections. Still it would be difficult to account for the speed-dependence of the bifurcation.

With respect to the mathematics, we have tried to characterize a class of solutions of spatially continuous neural networks that is different from solutions have been analyzed in previous work that apply similar mathematical methods. By the presence of a time-dependent stimulus, the stimulus-locked traveling pulse solution is different from the stable stationary solutions of networks with static inputs that have been repeatedly analyzed in the literature (e.g. [42, 38, 37, 49, 40]). The stimulus-locked solution is also different from self-generated traveling waves or pulses that have been studied in different contexts [10]. For such solutions the pulse propagates with an equilibrium speed that is specified by the network dynamics, whereas for the stimulus-locked traveling pulse solution the propagation speed is given by the stimulus. At least for the linear threshold model with small contrast, the speed regime for which a stimulus-locked traveling pulse solution exists is, however, in a neighborhood of the optimal speed with which a self-generated pulse would propagate in the absence of a time-dependent stimulus. The proposed recurrent mechanism for direction selectivity exploits a kind of "resonance" between the tendency of the network to stabilize a traveling pulse solution with characteristic speed and the incoming time-dependent

stimulus activity.

We conclude from our analysis that the observation of lurching activity waves in populations of direction-selective neurons in the visual cortex would be a strong indicator for the relevance of the recurrent mechanism for direction selectivity that we discussed in this paper. Lurching waves and the related bifurcations might be experimentally observable by recording from populations of direction selective neurons. The neurons first would have to be clustered according to their speed selectivity and the centers of their receptive fields. The responses would have to be time-aligned with respect to the stimulus. Then activity waves could potentially be observed either by simple histogramming within the different spatio-temporal bins, or by using more sophisticated techniques for interpolation, either based on standard regularization or Bayesian techniques [50, 51, 52]. A potential complication in the visual cortex might be that multiple populations of neurons with different speed selectivity might inhibit each other [29]. The same mechanism, however, might be relevant in other cortical areas as well, that are experimentally easier to access. One example is the direction-selective place cells in the hippocampus that have the advantage that multi-unit recordings with more than 100 electrodes are possible [53].

Appendix: Stability analysis

2.5.1 Stability of the traveling pulse solution in the step threshold model

The stability of the traveling pulse solution is analyzed by perturbing the stationary solution in the moving coordinate system. Let $\delta U(\xi, t)$ be a small perturbation of $U^*(\xi)$. The linearized perturbation dynamics reads

$$\tau \frac{\partial \delta U}{\partial t} - \tau v \frac{\partial \delta U}{\partial \xi} + \delta U(\xi, t) = -w(\xi - \xi_1^*)\, \delta\xi_1 + w(\xi - \xi_2^*)\, \delta\xi_2, \qquad (2.34)$$

where $\delta\xi_i$ $(i = 1, 2)$ are the perturbations of the boundary points of the exited regime from the stationary values of ξ_i^* with $\xi_i = \xi_i^* + \delta\xi_i$ satisfying $U(\xi_i, t) = 0$. Note that

$\delta\xi_i$ is not independent of $\delta U(\xi,t)$, and the dependence can be found through

$$U(\xi_i^* + \delta\xi_i, t) = U(\xi_i^*, t) + \frac{\partial U(\xi_i^*, t)}{\partial \xi}\delta\xi_i + O(\delta\xi_i^2) = 0.$$

Since $U(\xi_i^*, t) = \delta U(\xi_i^*, t)$, to the first order we have

$$\delta\xi_i = -\delta U(\xi_i^*, t)/c_i^*,$$

where $c_i^* \equiv \mathrm{d}U^*(\xi_i)/\mathrm{d}\xi$. Substituting this back into the perturbed dynamics, we derive the perturbed dynamics with perturbations in U only:

$$\tau\frac{\partial \delta U}{\partial t} - \tau v\frac{\partial \delta U}{\partial \xi} + \delta U(\xi,t) = \frac{w(\xi-\xi_1^*)}{c_1^*}\delta U(\xi_1^*,t) - \frac{w(\xi-\xi_2^*)}{c_2^*}\delta U(\xi_2^*,t).$$

To check its stability, we substitute a solution of the form $\delta U(\xi,t) = e^{\lambda t}Y(\xi)$ into the above dynamics and derive the equation for $Y(\xi)$:

$$-\tau v Y'(\xi) + (1+\tau\lambda)Y(\xi) = \frac{w(\xi-\xi_1^*)}{c_1^*}Y(\xi_1^*) - \frac{w(\xi-\xi_2^*)}{c_2^*}Y(\xi_2^*).$$

We solve this equation by first assuming that $Y(\xi_1^*)$ and $Y(\xi_2^*)$ are constant, and afterwards we give self-consistent conditions for the solutions at ξ_1^* and ξ_2^* to satisfy. The solution of the above equation is

$$Y(\xi) = \frac{K(\xi-\xi_1^*)}{c_1^*(1+\tau\lambda)}Y(\xi_1^*) - \frac{K(\xi-\xi_2^*)}{c_2^*(1+\tau\lambda)}Y(\xi_2^*), \tag{2.35}$$

The solution $Y(\xi)$ in Eq. (2.35) has to satisfy two self-consistency equations for the solutions at ξ_1^* and ξ_2^*:

$$\begin{aligned} Y(\xi_1^*) &= \frac{K(0)}{c_1^*(1+\tau\lambda)}Y(\xi_1^*) - \frac{K(\xi_1^*-\xi_2^*)}{c_2^*(1+\tau\lambda)}Y(\xi_2^*) \\ Y(\xi_2^*) &= \frac{K(\xi_2^*-\xi_1^*)}{c_1^*(1+\tau\lambda)}Y(\xi_1^*) - \frac{K(0)}{c_2^*(1+\tau\lambda)}Y(\xi_2^*). \end{aligned}$$

For the above equations to have a solution, the transcendental Eq. 2.9 has to be sat-

isfied. From this equation the eigenvalues λ can be found numerically. The traveling pulse solution is asymptotically stable only if the real parts of all eigenvalues λ that solve Eq. (2.9) are nonpositive.

2.5.2 Stability of the traveling pulse solution in the linear threshold model

The stability analysis is carried out by perturbing the dynamics of the Fourier components in Eq. (2.25-2.27). The general procedure is to perturb the dynamics first, which involves the perturbation of terms such $\delta\theta_c$, $\delta\Phi$ and $\Delta I_1(t)$. To determine these terms, we subsequently perturb Eqs. (2.21, 2.22, 2.23). Defining $\tilde{\Phi} \equiv \Phi - \theta_0$ and $\tilde{\Psi} \equiv \Psi - \theta_0$, the perturbed linearized dynamics can be summarized as follows:

$$
\begin{aligned}
\tau\delta\dot{r}_0 &= (\frac{J_0\theta_c}{\pi} - 1)\,\delta r_0 + J_1\cos(\tilde{\Phi} - \tilde{\Psi} + \beta)\frac{\sin(\theta_c)}{\pi}\,\delta r_1 - \epsilon C\sin(\tilde{\Phi})\frac{\sin(\theta_c)}{\pi}\,\delta\tilde{\Psi} \\
\tau\delta\dot{r}_1 &= \cos(\tilde{\Phi} - \tilde{\Psi})\frac{\sin(\theta_c)}{\pi}J_0\,\delta r_0 + \{-1 + J_1[\frac{\theta_c}{2\pi}\cos(\beta) + \frac{\sin(2\theta_c)}{4\pi} \\
&\times \cos(2(\tilde{\Phi} - \tilde{\Psi}) + \beta)]\}\,\delta r_1 - \epsilon C\{\frac{\theta_c}{2\pi}\sin(\tilde{\Psi}) + \frac{\sin(2\theta_c)}{4\pi}\sin(2\tilde{\Phi} - \Psi)\}\,\delta\tilde{\Psi} \\
\tau r_1\delta\dot{\tilde{\Psi}} &= \sin(\tilde{\Phi} - \tilde{\Psi})\frac{\sin(\theta_c)}{\pi}J_0\,\delta r_0 + [-\tau v - J_1(\frac{\theta_c}{2\pi}\sin(\beta) - \frac{\sin(2\theta_c)}{4\pi} \\
&\times \sin(2(\tilde{\Phi} - \tilde{\Psi}) - \beta)]\,\delta r_1 - \epsilon C\{\frac{\theta_c}{2\pi}\cos(\tilde{\Psi}) - \frac{\sin(2\theta_c)}{4\pi}\cos(2\tilde{\Phi} - \Psi)\}\,\delta\tilde{\Psi}\ .
\end{aligned}
$$

To determine the stability of the traveling pulse solution, we have to analyze the dynamics of these three coupled differential equations. If $\epsilon C \ll 1$, then the dynamics of $\delta\dot{\tilde{\Psi}}$ is decoupled from that of δr_0 and δr_1 and the stability condition can be approximated by the stability of the two dynamics Eqs. (2.32,2.33).

Chapter 3

Selectively Grouping Neurons in Recurrent Networks of Lateral Inhibition

3.1 Introduction

It has long been known that lateral inhibition in neural networks can lead to winner-take-all competition, so that only a single neuron is active at a steady state [13, 14, 54, 55, 56, 57]. When used for unsupervised learning, such winner-take-all networks enforce grandmother-cell representations as in vector quantization [58]. Recently many research efforts have focused on unsupervised learning algorithms for sparsely distributed representations [59, 60]. These algorithms lead to representations where multiple neurons participate in the encoding of an object and so are more distributed than vector quantization. Therefore, it is of interest to find ways of using lateral inhibition to mediate winner-take-all competition between groups of neurons, enforcing the sparse representation at a network level.

Competing groups of neurons are the essence of attractor models of associative memory. Selectively grouped neurons correspond to patterns that are stored as at-

[0]This chapter is based on an article to appear in *Neural Computation* by Xie, Hahnloser and Seung.

tractors in the network, with only one of these patterns retrieved at a steady state [61, 62, 63]. In this case, the input to the network is represented in the initial conditions of the dynamic system, and the winning group is the resulting steady state. However, the binary behavior of an individual neuron in associative memory models is much different and computationally less informative than a biophysical neuron, whose firing rate encodes information on the signal it is processing. Although there have been extensions of these discrete and digital attractor networks to networks with graded [64, 65] or stochastic neurons [66], the behavior of the individual neuron tends to be inactive or saturate and thus remains binary in essence.

In this paper, we show how winner-take-all competition between groups of neurons can be realized in networks of non-binary, analog neurons. In a network model to be introduced later, neurons at a steady state can be either active or inactive, which form a binary pattern representing a permitted grouping of the neurons. At the same time, the activated neurons carry analog values resulting from computations implemented by the network.

We present a natural way of wiring the network to selectively group neurons by adding strong lateral inhibition between them. Given a collection of potentially overlapping groups, the inhibitory connectivity is set by a simple formula that can be interpreted as arising from an online learning rule. To show that the resulting network functions as "group winner-take-all", we perform a stability analysis. If the strength of inhibition is sufficiently great and the group organization satisfies certain conditions, one and only one group of neurons can be activated at a stable steady state. In general, the identity of the winning group depends on the network inputs, and also the initial conditions of the dynamics.

We characterize the storage capacity, the maximum number of groups the network can mediate to produce winner-take-all competitions, for random sparse groups in which each neuron has the probability p to be included in each group. Let n be the total number of neurons in the network. We determine the optimal sparsities p that maximize group entropy in two cases: (1) When the input is dense, the optimal sparsity scales as $\ln(n)/n$, and (2) when the inputs are of equal sparsity as the groups

themselves, the optimal sparsity scales as $\sqrt{\ln(n)/n}$. In the first case, the storage capacity roughly scales as n^2, and in the second case, the storage capacity scales as $n/\ln(n)$.

3.2 Basic definitions

Let m groups of neurons be given, where the group membership of the ath group is specified by

$$\xi_i^a = \begin{cases} 1 & \text{if the } i\text{th neuron is in the } a\text{th group} \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

for $i = 1, ..., n$.

We will assume that every neuron belongs to at least one group[1], and that every group contains at least one neuron. A neuron is allowed to belong to more than one group, so that the groups are potentially overlapping. The inhibitory synaptic connectivity of the network is defined in terms of the group membership,

$$J_{ij} = \prod_{a=1}^{m} (1 - \xi_i^a \xi_j^a) = \begin{cases} 0 & \text{if } i \text{ and } j \text{ both belong to a group} \\ 1 & \text{otherwise.} \end{cases} \tag{3.2}$$

The synaptic matrix J basically states that a connection between neuron i and j is established only if they do not belong to any of the same groups. This pattern of connectivity could arise from a simple learning mechanism. Suppose that all elements of J are initialized to be unity, and the groups are presented sequentially as binary vectors $\xi^1, \ldots, \xi^m$. The ath pattern is learned through the update

$$J_{ij} \leftarrow J_{ij}(1 - \xi_i^a \xi_j^a). \tag{3.3}$$

In other words, if neurons i and j both belong to pattern a, then the connection between them is removed. After presentation of all m patterns, this leads to equa-

[1]This condition can be relaxed, but is kept for simplicity.

tion 3.2. At the start of the learning process, the initial state of J corresponds to uniform inhibition, which is known to implement winner-take-all competition between individual neurons. It will be seen that, as inhibitory connections are removed during learning, the competition evolves to mediate competition between groups of neurons rather than individual neurons.

Let x_i be the activity of neuron i. The dynamics of the network is given by

$$\frac{dx_i}{dt} + x_i = \left[b_i + \alpha x_i - \beta \sum_{j=1}^{n} J_{ij} x_j \right]^+, \tag{3.4}$$

for all $i = 1, ..., n$, where $[z]^+ \equiv \max\{z, 0\}$ denotes rectification, $\alpha > 0$ is the strength of self-excitation, and $\beta > 0$ is the strength of lateral inhibition. b_i is the external input. Equivalently, the dynamics can be written in matrix-vector form as

$$\dot{\boldsymbol{x}} + \boldsymbol{x} = [\boldsymbol{b} + W\boldsymbol{x}]^+, \tag{3.5}$$

where $W = \alpha I - \beta J$ includes both self-excitation and lateral inhibition. The state of the network is specified by the vector $\boldsymbol{x} = [x_1, ..., x_n]^T$, and the external input by the vector $\boldsymbol{b} = [b_1, ..., b_n]^T$. Recurrent networks with linear threshold units have been used in a variety of neural modeling studies [39, 44, 67, 15].

A vector $\boldsymbol{v}$ is said to be nonnegative, $\boldsymbol{v} \geq 0$, if all of its components are nonnegative. The nonnegative orthant is the set of all nonnegative vectors. Notice that in equation 3.4, $\dot{x}_i \geq 0$ whenever $x_i = 0$. Moreover, the linear threshold function is obviously Lipschitz continuous. These two properties are sufficient to guarantee that the nonnegative orthant is a positive invariant set of the dynamics, that is, any trajectory of equation 3.4 starting in the nonnegative orthant remains there [68]. Furthermore, even if the initial state of $\boldsymbol{x}$ is negative, it will become nonnegative after some transient period. Therefore, for simplicity we consider trajectories that are confined to the nonnegative orthant $\boldsymbol{x} \geq 0$. However, we consider input vectors $\boldsymbol{b}$ whose components are of arbitrary sign.

3.3 Network performance

Next, we briefly state some of the properties of the network. The detailed analysis is deferred to later sections.

We start with a simple case with n different groups each containing one of the n neurons, which is the traditional winner-take-all network. Suppose $k > 1$ neurons are active initially. After proper ordering, the interaction matrix between these k active neurons is $W = (\alpha + \beta)I - \beta \mathbf{1}\mathbf{1}^T$, where $\mathbf{1}$ is the column vector consisting of all ones. One eigenvector of W is $\mathbf{1}$ with eigenvalue $\alpha - (k-1)\beta$. The other $k-1$ eigenvectors are differential modes whose components sum to zero, with eigenvalue $\alpha + \beta$. If the inhibition strength is strong enough, $\beta > 1 - \alpha$, the differential modes are unstable. Hence the network cannot have more than one neuron active at a steady state. Moreover, the network is guaranteed to converge to a steady state provided that $\alpha < 1$. Under these conditions, we can conclude that for all $\boldsymbol{b}$ and initial conditions of $\boldsymbol{x}$, the network always converges to one of the given groups.

For the general case with arbitrary group membership matrix ξ, the above conclusion still holds true, except in some degenerate cases (which will be described in the next section). If the lateral inhibition is strong enough ($\beta > 1 - \alpha$) as in the previous case, any steady state with two active neurons not contained in the same group is unstable. If $\alpha < 1$, the network is again guaranteed to converge to a steady state. Therefore, one and only one of the given groups can be active at each steady state.

Which groups could potentially be the winner is specified by the input $\boldsymbol{b}$. In the case of nonoverlapping groups, the potential winners are determined by the aggregate positive input $B^a = \sum_{i=1}^{n} [b_i]^+ \xi_i^a$ that each group receives. Any group with $B^a \geq (1-\alpha)\beta^{-1} b_{\max}$ could end up as the winning group, where $b_{\max} \equiv \max_i \{b_i\}$. Which groups wins in the end depends on the initial conditions. It is possible for a specific group to win for all initial conditions if its inputs are sufficiently large.

The synaptic connections between neurons within a group are restricted to self-excitation. This causes the activities of winning neurons to be equal to their rectified

input, amplified by a gain factor $1/(1-\alpha)$. Thus the network implements a form of hybrid analog-digital computation, selectively amplifying activities in only one group of neurons.

3.4 Analysis of the network dynamics

3.4.1 Convergence to a steady state

This section characterizes the steady state responses of the network equation 3.4 to an input $\boldsymbol{b}$ that is constant in time. For this to be a sensible goal, we need some guarantee that the dynamics converges to a steady state, and does not diverge. This is provided by the following theorem.

Theorem 1 *Consider the network equation 3.4. The following statements are equivalent:*

1. *For any input $\boldsymbol{b}$, the network state $\boldsymbol{x}$ converges to a steady state that is stable in the sense of Lyapunov, except for initial conditions in a set of measure zero consisting of unstable equilibria.*

2. *The strength α of self-excitation is less than one.*

Proof: To prove (2) $\Rightarrow$ (1), if $\alpha < 1$, the function

$$E(\boldsymbol{x}) = \frac{1}{2}(1-\alpha)\boldsymbol{x}^T\boldsymbol{x} + \frac{\beta}{2}\boldsymbol{x}^T J\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{x} \tag{3.6}$$

is bounded below and radially unbounded in the nonnegative orthant. Furthermore, E is nonincreasing following the dynamics

$$\begin{aligned} dE/dt &= -((I-W)\boldsymbol{x} - \boldsymbol{b})^T(\boldsymbol{x} - [W\boldsymbol{x} + \boldsymbol{b}]^+) \\ &= -\sum_{i\in M}((I-W)\boldsymbol{x} - \boldsymbol{b})_i^2 - \sum_{i\notin M}(x_i^2 - (W\boldsymbol{x} + \boldsymbol{b})_i\, x_i) \\ &\leq -\sum_{i\in M}((I-W)\boldsymbol{x} - \boldsymbol{b})_i^2 - \sum_{i\notin M} x_i^2 \\ &\leq 0 \end{aligned}$$

where $M \equiv \{i \mid (W\boldsymbol{x} + \boldsymbol{b})_i > 0, \forall i = 1, ..., n\}$. The notation $(\boldsymbol{z})_i$ denotes the ith component of the vector $\boldsymbol{z}$.

Equality above holds if and only if $\boldsymbol{x}$ is at the steady state. Therefore, $E(\boldsymbol{x})$ is a Lyapunov-like function assuring convergence to a stable steady state, except for initial conditions in a set of measure zero.

To prove (1) $\Rightarrow$ (2), let us suppose that (2) is false. If $\alpha \geq 1$, choose $\boldsymbol{b} = (1, 0, ..., 0)^T$ and initial conditions $\boldsymbol{x}(0) = (1, 0, ..., 0)^T$ so that the dynamics of the first neuron is reduced to $\dot{x}_1 + x_1 = [\alpha x_1 + 1]^+ \geq x_i + 1$, in which x_1 diverges. In addition, x_1 diverges for initial conditions in a set of nonzero measure, so (1) is contradicted. Therefore, $\alpha < 1$ is the both necessary and sufficient condition for convergence to a stable steady state. □

In the following, we restrict the network to $\alpha < 1$.

3.4.2 Permitted and forbidden sets

In general, the network may have many fixed points. However, only those that are stable are typically observed at a steady state. We will call a set of neurons that can be coactivated by some input at a stable (in the sense of Lyapunov) steady state a *permitted set.* Otherwise, it is termed a *forbidden set.*

For a set of neurons to be a permitted set, two conditions have to be satisfied: first, its neurons have to be steadily coactivated by some input; second, the steady coactivation must be stable. For the network we are considering, it is always possible to choose an input that realizes a steady coactivation of the given set of neurons. Hence, the first condition is readily satisfied. Consequently, whether a set is permitted or forbidden depends only on its stability, which is determined by the the synaptic connection matrix between the coactivated neurons. If the largest eigenvalue of that matrix is less than unity, then the set is a permitted. Otherwise, it is forbidden.

One special property of the permitted and forbidden sets is that any superset of a forbidden set is forbidden, and any subset of a permitted set is permitted [15]. An intuitive understanding of this property is that by inactivating a neuron its feedback is removed. Because the connections in a symmetric network form effectively positive

feedback loops, in the form of mutual excitation or disinhibition, removing feedbacks increases stability of the network. Similarly, adding positive feedbacks decreases stability, in agreement with the property that any superset of a forbidden set is forbidden.

The above property adds convenience for verifying whether a set is permitted or forbidden. For example, if we know a subset of a set is forbidden, then the set itself is forbidden. We will use this property in the following sections.

3.4.3 Relationship between groups and permitted sets

The network in equation 3.4 is constructed to make the groups and their subgroups the only permitted sets of the network. To determine whether this is the case, we must answer two questions. First, are all groups and their subgroups permitted? Second, are all permitted sets contained in the given groups? The first question is answered by the following Lemma.

Lemma 1 *All groups and their subgroups are permitted.*

Proof: If a set is contained in a group, then there is no lateral inhibition between the neurons in the set. Provided that $\alpha < 1$, all eigenvalues of the interaction matrix between neurons in the group are less than unity, so the set is permitted. □

The answer to the second question, whether all permitted sets are contained in the groups, is not necessarily affirmative. For example, consider the network defined by the group membership matrix $\xi = \{(1,1,0),(0,1,1),(1,0,1)\}$. Since every pair of neurons belongs to some group, there is no lateral inhibition ($J = 0$), which means that there are no forbidden sets. As a result, $(1,1,1)$ is a permitted set, but obviously it is not contained in any group.

Let us define a *spurious permitted set* to be a permitted set that is not contained in any group. For example, $(1,1,1)$ is a spurious permitted set in the above example. To eliminate all the spurious permitted sets in the network, certain conditions on the group membership matrix ξ have to be satisfied.

Definition 1 *The membership matrix ξ is* degenerate *if there exists a set of $k \geq 3$ neurons that is not contained in any group, but all of its subsets with $k-1$ neurons belong to some group. Otherwise, ξ is called* nondegenerate.

For example, $\xi = \{(1,1,0),(0,1,1),(1,0,1)\}$ is degenerate. Using this definition, we can formulate the following theorem.

Theorem 2 *The neural dynamics equation 3.4 with $\alpha < 1$ and $\beta > 1-\alpha$ has a spurious permitted set if and only if ξ is degenerate.*

To prove the theorem, we need the following lemma.

Lemma 2 *If $\beta > 1-\alpha$, any set containing two neurons not in any same group is forbidden under the neural dynamics equation 3.4.*

Proof: We start by analyzing a very simple case where there are two neurons belonging to two different groups. Let the group membership be $\{(1,0),(0,1)\}$. In this case, $W = \{(\alpha,-\beta);(-\beta,\alpha)\}$. This matrix has eigenvectors $(1,1)^T$ and $(1,-1)^T$ with eigenvalues being $\alpha-\beta$ and $\alpha+\beta$ respectively. Since $\alpha < 1$ for convergence to a steady state and $\beta > 0$ by definition, the $(1,1)^T$ mode is always stable. But if $\beta > 1-\alpha$, the $(1,-1)^T$ mode is unstable. This means that it is impossible for the two neurons to be coactivated at a stable steady state. Since any superset of a forbidden set is also forbidden, the result generalizes to more than two neurons. $\square$.

Now we are ready to prove Theorem (2) by using Lemma (2).

Proof of Theorem 2:

If ξ is degenerate, there must exist a set $k \geq 3$ neurons that is not contained in any group, but all of its subsets with $k-1$ neurons belong to some group. There is no lateral inhibition between these k neurons, since every pair of neurons belongs to some group. Thus the set containing all k neurons is permitted and spurious.

On the other hand, if there exists a spurious permitted set P, we need to prove that ξ must be degenerate. We will prove this by contradiction and induction. Let's assume ξ is nondegenerate.

P must contain at least 2 neurons since any one neuron subset is permitted and not spurious. By Lemma 2, these 2 neurons must be contained in some group, or else it is forbidden. Thus P must contain at least 3 neurons to be spurious, and any pair of neurons in P belongs to some group by Lemma 2.

If P contains at least k neurons and all of its subsets with $k-1$ neurons belong to some group, then the set with these k neurons must belong to some group, otherwise ξ is degenerate. Thus P must contain at least $k+1$ neurons to be spurious, and all its k subsets must belong to some group.

By induction, this implies that P must contain all neurons in the network, in which case, P is either forbidden or nonspurious. This contradicts the assumption that P is a spurious permitted set. □

Remark. The group winner-take-all competition described above holds only for the case of strong inhibition $\beta > 1-\alpha$. If β is small, the competition will be weak and may not result in group winner-take-all. In particular, if $\beta < (1-\alpha)/\lambda_{max}(-J)$, where $\lambda_{max}(-J)$ is the largest eigenvalue of $-J$, then the set of all neurons is permitted. Since every subset of a permitted set is permitted, this means that there are no forbidden sets and the network is monostable. Hence, group winner-take-all does not hold. In the intermediate regime, $(1-\alpha)/\lambda_{max}(-J) < \beta < 1-\alpha$, the network has forbidden sets, but the possibility of spurious permitted sets cannot be excluded.

3.5 The potential winners

We have seen that if ξ is nondegenerate, any stable coactive set of neurons must be contained in a group, provided that lateral inhibition is strong ($\beta > 1-\alpha$). The group that contains the coactive set is the "winner" of the competition between groups. The identity of the winner depends on the input $\boldsymbol{b}$, and also on the initial conditions of the dynamics.

Suppose the ath group is the winner. For all neurons not in this group to be

inactive, the self-consistent condition should read

$$\sum_i [b_i]^+ \xi_i^a J_{ij} \geq (1-\alpha)\beta^{-1}[b_j]^+, \tag{3.7}$$

for all $j \notin a$. If the group a contains the neuron with the largest input, this condition is always satisfied. Hence any group containing the neuron with the largest input is always a potential winner.

In the case of nonoverlapping groups, the condition in equation 3.7 can be simplified as

$$\sum_i [b_i]^+ \xi_i^a \geq (1-\alpha)\beta^{-1} \max_{j \notin a} \{[b_j]^+\}, \tag{3.8}$$

and therefore potential winners are determined by the aggregate *group inputs* $B^a = \sum_i [b_i]^+ \xi_i^a$. Denote the largest input as $b_{\max} = \max_i\{b_i\}$ and assume $b_{\max} > 0$. Only those groups whose aggregate inputs are not smaller than $(1-\alpha)\beta^{-1}b_{\max}$ can win, with the exact winner identity determined by the initial conditions of the dynamics.

3.6 An example – the ring network

In this section, we take the ring network as an example to illustrate several results we have obtained so far. Let n neurons be organized into a ring, and let every set of d contiguous neurons form a group. Thus in total there are n patterns to be stored. In the special case $d = 1$, this network becomes a traditional winner-take-all network.

In the case $d > 1$, the groups are overlapping and ξ could be degenerate, In fact, it can be shown that ξ becomes degenerate when $d \geq n/3 + 1$. This is illustrated in Figure 3-1, which shows the permitted sets of a ring network with 15 neurons. If the group width is $d = 5$ neurons, there are no spurious permitted sets (Figure 3-1A-C). However, when the group width is 6, the network contains 5 spurious permitted sets (Figure 3-1F).

Figure 3-2 shows the effect of changing the strength of lateral inhibition. When the strength of inhibition is strong ($\beta > 1 - \alpha$), there are no spurious permitted sets provided that ξ is nondegenerate (Figure 3-2D). In the other extreme, when

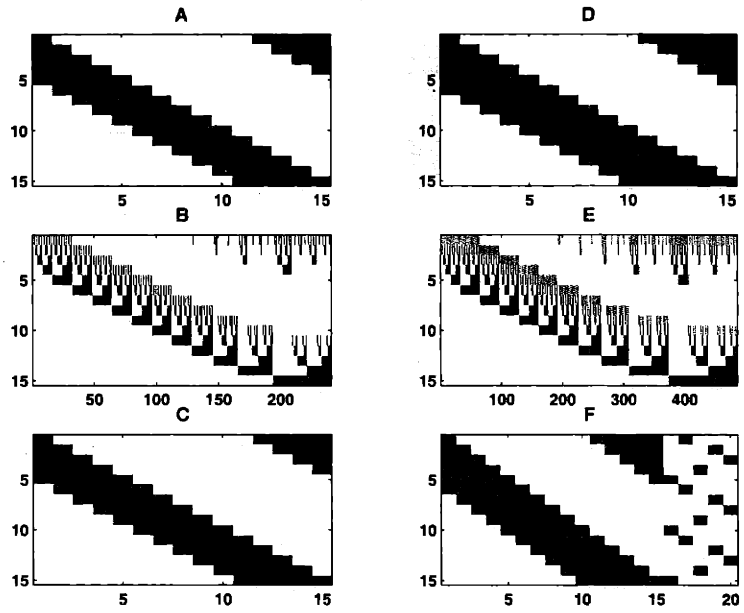

Figure 3-1: Permitted sets of the ring network. The ring network consists of 15 neurons with $\alpha = 0.4$ and $\beta = 1$. In panels A and D, the 15 groups are represented by columns. Black refers to active neurons and white refers to inactive neurons. (A) 15 groups of width $d = 5$. (B) All permitted sets corresponding to the groups in A. (C) The 15 permitted sets in B that have no permitted supersets. They are the same as the groups in A. (D) 15 groups with width $d = 6$. (E) All permitted sets corresponding to groups in D. (F) There are 20 permitted sets in E that have no permitted supersets. Note that there are 5 spurious permitted sets.

$\beta < (1-\alpha)/\lambda_{max}(-J)$, there is no unstable differential mode in the network. All neurons could potentially be active at a stable steady state, given a suitable input (Figure 3-2A). Between these two critical values ($1-\alpha < \beta < (1-\alpha)/\lambda_{max}(-J)$), there exist both unstable differential modes and spurious permitted sets (Figure 3-2C).

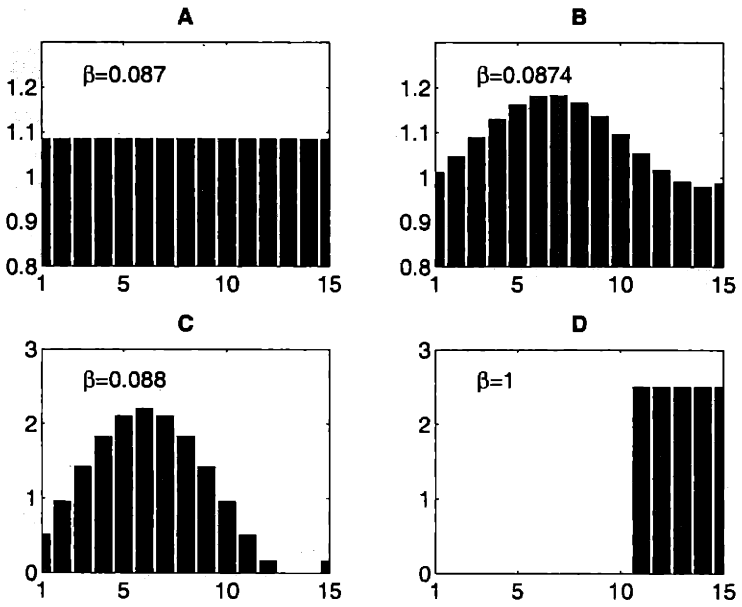

Figure 3-2: Lateral inhibition strength β determines the behavior of the network. The network is a ring network of 15 neurons with width $d = 5$ and where $\alpha = 0.4$, and input $b_i = 1$ for all i. The panels show the steady state activities of the 15 neurons. (A) There are no forbidden sets. (B) The marginal state $\beta = (1-\alpha)/\lambda_{max}(-J) = 0.874$, in which the network forms a continuous attractor. (C) Forbidden sets exist, and so do spurious permitted sets. (D) Group winner-take-all case, no spurious permitted sets.

3.7 Storage capacity for random sparse groups

An important characterization of any attractor network is its storage capacity for random patterns, i.e., random groups [69, 65]. In our case, as the number of groups gets larger, the probability of the groups being degenerate increases. We call the probability of error the probability that a neuron outside a group is activated by

mistake.

We choose random sparse groups; $p \ll 1$ is the probability that a particular neuron is part of a particular group. The storage capacity is defined to be the maximum number of groups the network can store, such that the error probability remains smaller than a given bound. After constructing the synaptic weight matrix, we present random inputs to the network. We assume that each component of the input $\boldsymbol{b}$ has the probability q of being positive. The expected number of neurons receiving positive inputs is $\bar{n} = nq$. Since a neuron receiving a nonpositive input can never become active in our network, the error probability is effectively determined by the network of the $\bar{n}$ neurons receiving positive inputs.

Under the randomness in both the groups and the inputs, the expected number of coactive neurons in a stable steady state is $c = \bar{n}p = npq$. Next, we assume that c neurons are coactivated, and calculate the probability $\mathcal{P}_\mathcal{E}$ of mistakenly activating any of the other $\bar{n} - c$ neurons.

We use $X(i,j)$ to denote the existence of synaptic inhibition between neurons i and j, which in our network implies that neurons i and j are not contained in any same group (See Figure 3-3). According to Lemma 2, $X(i,j)$ also represents mutual exclusion of neuron i and j at any stable steady state.

Without loss of generality, we index the c active neurons from 1 to c. For neuron j within the other $\bar{n} - c$ neurons to be inactive, it must make an inhibitory connection with at least one of the c neurons. The probability of this to happening is $\Pr\{\bigvee_{i=1}^{c} X(i,j)\}$, where $\bigvee$ represents logical "or". Extending this to all the other $\bar{n} - c$ neurons, we derive the probability for all the $\bar{n} - c$ neurons being inactive as follows:

$$\mathcal{P}_\mathcal{C} = \Pr\{\bigwedge_{j=1}^{\bar{n}-c} \bigvee_{i=1}^{c} X(i,j)\}, \tag{3.9}$$

where $\bigwedge$ represents logical "and". The error probability $\mathcal{P}_\mathcal{E}$ for at least one neuron being mistakenly activated is then

$$\mathcal{P}_\mathcal{E} = 1 - \mathcal{P}_\mathcal{C} = \Pr\{\bigvee_{j=1}^{\bar{n}-c} \overline{\bigvee_{i=1}^{c} X(i,j)}\}, \tag{3.10}$$

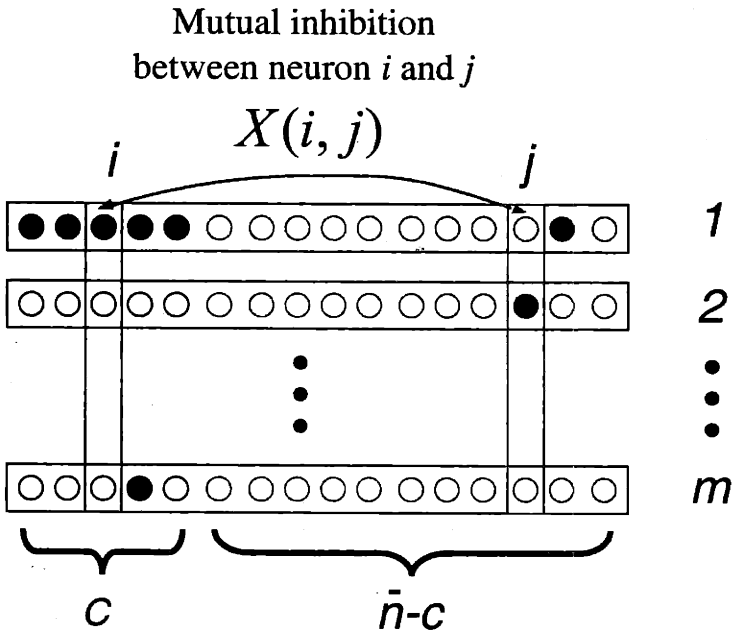

Figure 3-3: Diagram of m random groups. Filled circles represent active neurons. The first c neurons in the group 1 are coactivated. For a perfect retrieval, all the other $\bar{n} - c$ neurons must be inactive, i.e., all of them must be inhibited by at least one of the c active neurons. The error probability $\mathcal{P}_\mathcal{E}$ is the probability that at least one of the $\bar{n} - c$ neurons is active.

where the over-line denotes logic complement. Next we find an upper bound on $\mathcal{P}_\mathcal{E}$ and use it to estimate the capacity of the network.

3.7.1 Capacity

The error probability is upper bounded by

$$\mathcal{P}_\mathcal{E} \leq (\bar{n}-c)\Pr\{\overline{\bigvee_{i=1}^{c} X(i,j)}\} = (\bar{n}-c)(1-\Pr\{\bigvee_{i=1}^{c} X(i,j)\}), \tag{3.11}$$

where $\Pr\{\bigvee_{i=1}^{c} X(i,j)\}$ can be exactly calculated using the inclusion-exclusion principle [70] as follows:

$$\begin{aligned}
\Pr\{\bigvee_{i=1}^{c} X(i,j)\} &= \sum_{k=1}^{c}(-1)^{k+1}\binom{c}{k}\Pr\{\bigwedge_{i_1,i_2,\ldots,i_k} X(i_k,j)\} &(3.12)\\
&= \sum_{k=1}^{c}(-1)^{k+1}\binom{c}{k}\left[1-p+p(1-p)^k\right]^{m-1} &(3.13)
\end{aligned}$$

In the above equation, the term $1-p+p(1-p)^k$ represents the probability that neuron j does not coexist with other k neurons. This can happen in two cases: with neuron j being inactive (with probability $1-p$), or neuron j being active but all other k neurons being inactive (with probability $p(1-p)^k$).

Equation 3.13 can be further simplified by

$$\begin{aligned}
\Pr\{\bigvee_{i=1}^{c} X(i,j)\} &= 1+\sum_{k=0}^{c}(-1)^{k+1}\binom{c}{k}\left[1-kp^2+O(p^3)\right]^{m-1} &(3.14)\\
&\approx 1-\sum_{k=0}^{c}(-1)^{k}\binom{c}{k}\exp(-kmp^2) &(3.15)\\
&= 1-[1-\exp(-mp^2)]^c. &(3.16)
\end{aligned}$$

We have made two approximations in the above calculation. To derive equation 3.14 we have assumed that cp is sufficiently small, implying sparse groups. In the approximation made in equation 3.15 we have assumed $m(cp^2)^2 \to 0$ in the large n limit, i.e., the number of groups m should scale in order less than $1/(cp^2)^2$. We will see later,

after deriving the capacity m, that these assumptions are indeed satisfied.

By substituting equation 3.16 into equation 3.11, we derive the upper bound on the error probability

$$\mathcal{P}_{\mathcal{E}} \leq (\bar{n} - c)[1 - \exp(-mp^2)]^c. \tag{3.17}$$

We observed close tightness of this bound when compared to the true error probability from numerical simulations of random groups (see Figure 3-4).

Given some small number d, the error probability $\mathcal{P}_{\mathcal{E}}$ is guaranteed not to exceed this number, provided that $m < m^*(d)$, where

$$\begin{aligned} m^*(d) &= -p^{-2} \ln \left\{1 - [d/(\bar{n} - c)]^{1/c}\right\} && (3.18) \\ &\approx -p^{-2} \ln \left\{1 - [d/(nq)]^{1/(npq)}\right\}, && (3.19) \end{aligned}$$

where (3.19) follows from $c \ll \bar{n}$.

Given n, p and q, using equation 3.19, we can estimate the maximum number of random groups the network can store in such a way that the probability of incorrect retrieval remains smaller than d.

3.7.2 Optimal sparsity

How sparse should the random groups optimally be? We define the optimal sparsity p^* as the sparsity p that maximizes the information capacity of the network. We measure the information capacity I by the normalized entropy of the m^* random groups,

$$I = -m^* n \ [p \log_2 p + (1 - p) \log_2 (1 - p)]/n^2. \tag{3.20}$$

In other words, I is the entropy of the m^* binary words with length n with probability p of 1, normalized by the number of synaptic connections.

The denominator n^2 corresponds to the total entropy the binary synaptic weight matrix J can hold. Thus I is expressed as the fraction of the possible entropy of J, used to store groups. The optimal sparsity p^* is given by $p^* = \text{argmax}_p \{I\}$. To calculate p^*, we first have to choose some value for q, determining the probability

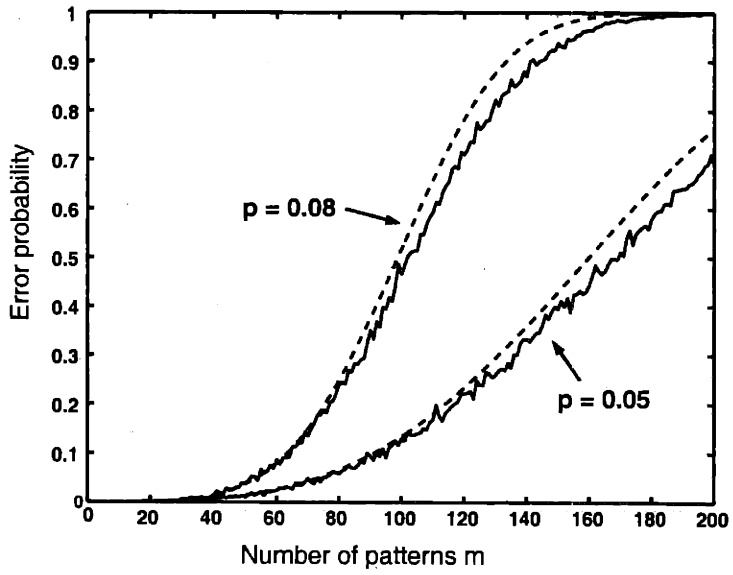

Figure 3-4: The error probability $\mathcal{P}_\varepsilon$ is plotted as a function of the number of groups m. Here $q = 1$ and the number of neurons $n = 100$. The solid curves show the results from numerical simulations, and the dashed curves are the upper bounds calculated by equation 3.16. Two different sparsities p are used.

that a neuron receives an excitatory input. We consider two different cases. First, q is independent of p, and without loss of generality we take $q = 1$, which corresponds to the case where the inputs are excitatory and non-sparse. Second, q depends on p, and for simplicity we choose $q = p$, which is the case where the inputs are of equal sparsity as the groups.

The optimal sparsity calculations for both cases are derived in the Appendix. Here we only state the results:

$$p^* \approx \begin{cases} \log_2(n)/n & \text{when } q = 1 \\ \sqrt{k \ln(n)/n} & \text{when } q = p, \end{cases} \tag{3.21}$$

where $k = 2.86$ is a constant. The approximation becomes exact when the number of neurons goes to infinity. This result shows that to achieve the maximum information capacity, p^* should scale as $\ln(n)/n$ when $q = 1$ and as $\sqrt{\ln(n)/n}$ when $q = p$. Correspondingly, the average number of neurons in each pattern scales as $\ln(n)$ for $q = 1$ and $\sqrt{n \ln(n)}$ for $q = p$.

By substituting p^* into equation 3.19, we derive the the storage capacity for these optimal sparsities,

$$m^* \approx \begin{cases} \alpha n^{2-1/c} & \text{when } q = 1 \\ k_m n / \ln(n) & \text{when } q = p, \end{cases} \tag{3.22}$$

where $c \approx \log_2(n)$, $\alpha = d^{1/c}/\log_2^2(n)$ and $k_m = -\ln[1 - \exp(-1/(2k^2))]/k^2 \approx 0.35$. Since $\ln(n)$ hardly increases for large n, the capacity in the $q = 1$ case roughly scales as n^2 and in the $q = p$ case it roughly scales as n.

In section (3.7.1), equation 3.15 is derived under the assumption that $m(cp^2)^2 \to 0$ in the large n limit. Now we check the validity of this assumption. Self-consistently, in the case $q = 1$ we find $m^*(cp^{*2})^2 \sim 1/n^2$, and in the case $q = p$, $m^*(cp^{*2})^2 \sim 1/n$. Both of them approach zero in the large n limit.

3.8 Discussion

We have presented a network that uses structured lateral inhibition to mediate winner-take-all competition between potentially overlapping groups of neurons. Our construction utilizes the distinction between permitted and forbidden sets of neurons, and identifies the allowed groupings as permitted sets inherent in the network.

Our capacity calculation in the $q = 1$ case reveals similarity with the Willshaw model [63]: We find that the optimal sparsity scales as $\ln(n)/n$, e.g., for a network of 10^{10} neurons an optimal group consists of less than 30 neurons and is thus unrealistically small. In the case where inputs are sparse, $q = p$, we find that the optimal sparsity scales roughly as $\sqrt{n}$ and is thus within the realm of real networks.

A distinct feature of our generalized winner-take-all network is the coexistence of discrete pattern selection and analog computation. We use strong lateral inhibitory interactions to constrain certain groupings of neurons, but leave the analog values of the active neurons unconstrained, except by the input. It might be interesting to apply our principle of how to constrain active groups to the problem of data reconstruction using a constrained set of basis vectors. The constraints on the linear combination of basis vectors could for example implement sparsity or nonnegativity constraints [60].

The coexistence of analog filtering with logical constraints on neural activation represents a form of hybrid analog-digital computation that may be especially appropriate for perceptual tasks. Using this network model for object recognition, the perception of an object could be represented by the set of active neurons, while activities of these neurons correspond to continuous instantiations of the object such as viewpoint, illumination, and scale [71]. In addition, this type of network may constitute a neural mechanism for feature binding and sensory segmentation problems, as suggested by Wersing et al. [72, 73]. In the domain of olfactory perception, recent experimental data on odor evoked population responses in the olfactory bulb also show some promising applications of our model [74, 75, 76].

As we have shown, there are some degenerate cases of overlapping groups, to which

our method does not apply. It is an interesting open question whether there exists a general way of translating arbitrary groups of coactive neurons into permitted sets without involving spurious permitted sets. There are several possible approaches. For example, we could use a more sophisticated interaction matrix, including both lateral inhibition and excitation. For instance, in the three neuron degenerate example given earlier, if we choose the interaction matrix $W = a\boldsymbol{v}\boldsymbol{v}^T$ with $\boldsymbol{v} = [1, -1, 1]^T$ and $1/3 < a < 1/2$, then the spurious set $(1, 1, 1)$ is forbidden, whereas its subsets are still permitted. Another possible approach would be to use higher order interactions. Take again the three neuron degenerate case for example. If we added quadratic interactions into the dynamics, $\dot{x}_i + x_i = [b_i + \alpha x_i - \beta \sum_j J_{ij} x_j - \gamma \sum_{j,k} x_j x_k]^+$, it would follow that for large enough inputs and suitable parameters the set $(1, 1, 1)$ would not be permitted but its subsets would. One more possible approach would be to use hierarchical networks with inter-layer excitation and intra-layer inhibition.

In the past, a great deal of research has been inspired by the idea of storing memories as fixed-point attractors in neural networks with a fixed input. Our model suggests an alternative viewpoint, which is to regard permitted sets as memories latent in the synaptic connections, while the fixed points corresponding to permitted sets can continuously change depending on the input. From this viewpoint, the contribution of the present paper is a method of storing and retrieving memories as permitted sets in neural networks.

Appendix

Calculation of the optimal sparsity for random patterns

Start from the information capacity of the network, given by

$$\begin{aligned} I &= -m^* n[p\log_2 p + (1-p)\log_2(1-p)]/n^2 \\ &\approx \log_2(p)(pn)^{-1}\ln\{1-[(nq)^{-1}d]^{1/(npq)}\}. \end{aligned}$$

Here, m^* is from equation 3.19 and the approximation is made in the small p limit. Next we consider two cases for choosing the value of q and find the optimal $p^* = \text{argmax}_p\ \{I\}$ for these two cases respectively. The calculation is done under the condition that the number of neurons n is sufficiently large.

3.1.1 Dense inputs, $q = 1$

The information capacity I can be written as $I = c^{-1}\ln(1-(d/n)^{1/c})\log_2(c/n)$, where $c = pn$. By setting the derivative of I with respect to c equal to zero, we find

$$\ln(d/n)^{1/c}\ln(c/n) + [(d/n)^{-1/c} - 1]\ln[1-(d/n)^{1/c}]\ [1-\ln(c/n)] = 0.$$

Let $z = (d/n)^{1/c}$. Then we have

$$\ln z\ln(c/n) + (z^{-1}-1)\ln(1-z)\ [1-\ln(c/n)] = 0.$$

Under sparsity assumption, $p = c/n \ll 1$, we have $|\ln(c/n)| \gg 1$. Hence, the above equation can be simplified to

$$(1-z)\ln(1-z) = z\ln(z). \tag{3.23}$$

The solutions of the above equation are $z = 0$, $1/2$, and 1. Given a fixed n, c can only be a finite number. Therefore, the solution $z = 1$ is impossible. The other two solutions lead to $c = 0$ or $c = \log_2(n)$. Correspondingly, $p = 0$ or $p =$

$\log_2(n)/n$. Substituting the value of $p = 0$ into I, we find that $p = 0$ corresponds to a local minimum. Furthermore, the boundary value $p = 1$ also corresponds to a local minimum. From these, we conclude that the optimal probability is given by $p^* = \log_2(n)/n$. Notice that it satisfies sparsity assumption ($p^* \ll 1$).

3.1.2 Sparse inputs, $q = p$

The derivative of I with respect to p is

$$\begin{aligned} I'(p) &= \{\ln(1-t)[1-\ln(p)] + t(1-t)^{-1}\ln(p)/(np^2)[1+2\ln(d/(pn))]\}/(np^2\ln 2) \\ &\approx -[1-2k\ln(pn)/(np^2)]\ln(p)\ln(1-t)/(np^2\ln 2), \end{aligned}$$

where $k \equiv -t[(1-t)\ln(1-t)]^{-1}$ and $t \equiv [d/(pn)]^{1/(p^2n)}$. To derive the above equation, we have neglected small terms by assuming that n is sufficiently large.

By setting $I'(p^*) = 0$ we find that p^* obeys,

$$\frac{\ln(p^*n)}{np^{*2}} = \frac{1}{2k}.$$

Deriving the exact form of p^* as a function of n from the above equation is not easy. However, when n is sufficiently large, we can simplify the calculation by assuming that k is independent of n. Under this ansatz, we derive p^* to scale as

$$p^* = \sqrt{k\ln(n)/n}. \tag{3.24}$$

Next we need to self-consistently verify our ansatz still holds by replacing p^* into the definition of t,

$$\ln t = \frac{\ln(d) - \ln[kn\ln(n)]/2}{k\ln(n)} \approx -\frac{1}{2k}. \tag{3.25}$$

The approximation becomes exact as n goes to infinity. Thus, we have verified that t is approximately constant, equal to $\exp(-1/(2k))$. This completes our ansatz.

We still need to determine the value of k. Substituting equation 3.25 into the

definition of k, we derive that

$$\frac{1-t}{2t} = \frac{\ln(t)}{\ln(1-t)}. \tag{3.26}$$

The root of this algebra equation can be found numerically. The final result is $t = 0.8396$ and $k = 2.86$. We can further check that the boundary values p at 0 or 1 only lead to local minima of I. Therefore, we conclude that p^* in equation 3.24 is the optimal sparsity.

Chapter 4

A double-ring network model of the head-direction system

4.1 Introduction

In the rat head-direction system, head-velocity inputs from the vestibular nuclei are integrated to yield a neural representation of the current directional heading with respect to the external environment. Neurons of this system, called head-direction cells, fire maximally when the rat faces one particular direction [6, 7, 8, 9]. These cells usually have different preferred directions, and a population of them encodes the rat's directional heading.

Previously, several network models have been proposed to emulate the properties of head-direction cells [10, 11, 12]. The work by Zhang focuses on modeling persistent activity of head-direction cells during stationary head states [10]. The persistent neural activity is generated in a ring-attractor network with symmetric excitatory and inhibitory synaptic connections. Independently, he and Redish et al. showed that integration is possible by adding asymmetrical connections to the attractor network [11]. The strength of these connections is modulated by head-velocity. When the rat moves its head to the right, the asymmetrical feedback loops between neurons

[0]This is a collaborative work with Richard H.R. Hahnloser and H. Sebastian Seung.

are biased toward the right-hand side and so induces a rightward shift of the activity in the attractor network. However, the integration mechanism by instantaneous changing synaptic strength is biologically unrealistic. A more plausible model without multiplicative modulation of connections has been studied recently by Goodridge and Touretzky [12]. There, the head-velocity input has a modulatory influence on the firing of intermittent neurons with spatially offset connections rather than on their connection strengths. However, to achieve an accurate integration in this model, the head-velocity input has to be transformed with some nonlinear function before acting on the network. This nonlinear function was obtained by curve-fitting the simulation with the desired result. It is unclear whether such nonlinear transformations actually exist in the head-direction system. Moreover, in this model, to achieve good integration for large head velocities, very fast synapses (less than 1ms for [12]) had to be assumed.

In this paper, we propose a new model with two populations of neurons. It integrates the head velocity signal directly based on the differential vestibular input to these two populations, with potentially slow synapses such as NMDA and GABAb. In our model, the connections made by one ring are responsible for rightward turns and the connections made by the other ring are responsible for leftward turns. We mathematically analyze the dynamics of the network, and find that with carefully chosen synaptic parameters, the network is able to achieve integration with high precision.

Although our network is conceptually simpler than previous models, we show that using two simple read-out methods, averaging and extracting the maximum, it is possible to approximate head-velocity independent tuning curves as observed in the Postsubiculum (PoS) and anticipatory responses in the anterior dorsal thalamus (ADN) [7, 77].

4.2 Definition of the model

We model the head-direction system with two populations of neurons, each of which is organized into a ring network structure. We assume the population size in each ring is sufficiently large, so that activities of neurons sharing similar properties in each ring can be averaged, resulting in a continuous approximation of the discrete neuronal dynamics,

$$\tau \frac{\partial s_l(\theta,t)}{\partial t} + s_l(\theta,t) = f_l(\theta,t) \tag{4.1a}$$

$$\tau \frac{\partial s_r(\theta,t)}{\partial t} + s_r(\theta,t) = f_r(\theta,t), \tag{4.1b}$$

which are leaky integrators that model the dynamics of synapses with time constant τ. $s_l(\theta,t)$ and $s_r(\theta,t)$ represent synaptic activation (e.g., neurotransmitter concentration) indexed by θ at time t in the left and right ring respectively. $f_l(\theta,t)$ and $f_r(\theta,t)$ denote the activities of neurons in the left and right ring respectively, which are determined by the feedforward inputs and the recurrent synaptic inputs weighted by synaptic connection strengths,

$$f_l = \left[\int_{-\pi}^{\pi} [W_s(\theta-\theta'-\phi)s_l(\theta',t) + W_d(\theta-\theta'+\psi)s_r(\theta',t)](2\pi)^{-1}\mathrm{d}\theta' + b_l\right]^+ \tag{4.2a}$$

$$f_r = \left[\int_{-\pi}^{\pi} [W_d(\theta-\theta'-\psi)s_l(\theta',t) + W_s(\theta-\theta'+\phi)s_r(\theta',t)](2\pi)^{-1}\mathrm{d}\theta' + b_r\right]^+ \tag{4.2b}$$

where $[x]^+ \equiv \max(0,x)$ denotes a rectification nonlinearity. b_l and b_r are the vestibular feedforward inputs that differentially signal head movements with a common baseline (....?). For simplicity, we take $b_l = b_0 - \Delta b$ and $b_r = b_0 + \Delta b$ where $\Delta\hat{b} \equiv \Delta b/b_0$ is proportional to angular head velocity. The function W_s represents the synaptic connection profile between neurons on the same ring and W_d between neurons on different rings. The phase variable ϕ is the intra-ring connection offset and ψ is the inter-ring connection offset. The two rings form mirror-symmetric copies of each other.

Because of the rotational symmetry of the ring network structure, function W_d and W_s can be decomposed into sums of Fourier series. For simplicity of mathematical

treatment, we approximate each of them by the first two Fourier components,

$$W_s(\theta) = J_0 + J_1 \cos\theta \qquad W_d(\theta) = K_0 + K_1 \cos\theta, \tag{4.3}$$

where J_0, J_1, K_0 and K_1 are synaptic connection parameters determining the connection strength of the intra- and inter-ring connections.

4.3 Integration

Depending on parameters chosen, the network Eq. 4.1 may exhibit different dynamic behavior. We model the head-direction system with an appropriately chosen parameter regime, under which the activities in each ring converges to a stationary bump when $\Delta b = 0$ (Fig. 4-1a) , and generates a traveling bump with constant form when $|\Delta b| > 0$ (Fig. 4-1b).

The stationary bump is used to model persistent activities of the head-direction cells when the animal is not moving. Based on these persistent neural activities, current head-direction can be "read out" using methods such as population vector [78, 79]. Because of the rotation-symmetry of the ring network, this stationary bump can be located at any position, and thus is able to represent an arbitrary head-direction. The symmetry also implies that the tuning curves of individual head-direction cells with respect to the head-direction is the same as the profile of the stationary bump. Therefore, properties of the bump can be used to predict those of the tuning curves, which can be measured in experiments by single-unit recordings.

When Δb is nonzero, the stationary bump starts to move with a velocity depending on Δb. The moving of the bump models the integration process. To achieve an accurate integration, several issues need to be addressed.

First, the angular velocity v of the traveling pulse should be the same as the angular head velocity. In our model, we assumed that $\Delta\hat{b}$ is proportional to the head angular velocity. Therefore, for a perfect integration, it is required that v be linearly related to $\Delta\hat{b}$ with a gain inverse to the gain between $\Delta\hat{b}$ and the head angular velocity.

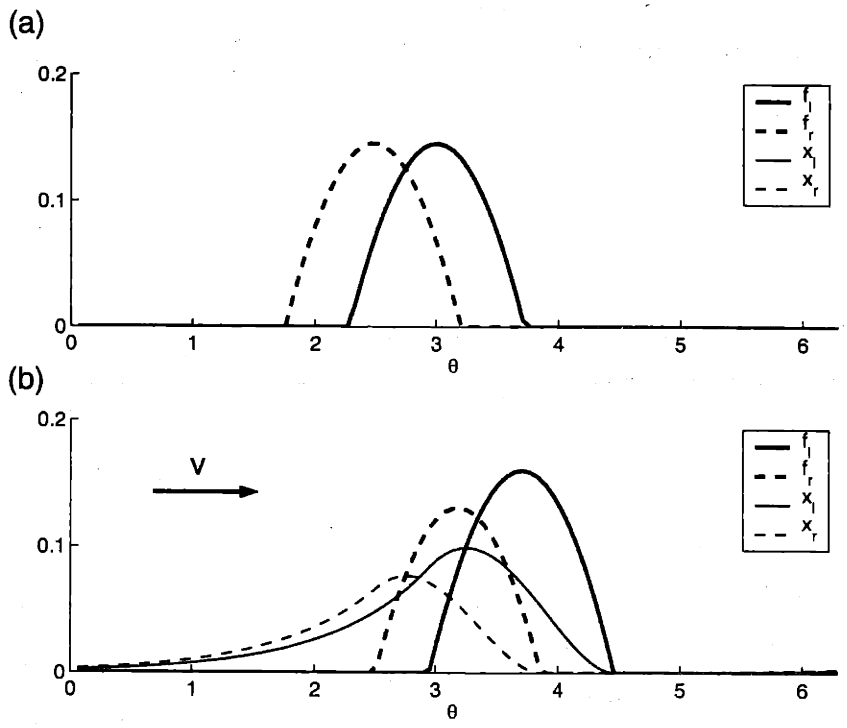

Figure 4-1: Neural activity and synaptic activation profiles of two rings in the stationary (a) and the moving state (b). The dashed and solid lines indicates those from the left and right ring respectively. In (a), $\Delta\hat{b} = 0$; in (b) $\Delta\hat{b} = -0.2$. Other parameters $J_0 = -60$, $K_0 = -5$, $J_1 = K_1 = 80$, $\phi = 80^o$, $\psi = 50^o$, and $\tau = 80$ms.

Second, the linearity should extend over a large range of Δb. Since the vestibular input is excitatory, $b_l, b_r \geq 0$, we consider the range $|\Delta\hat{b} \leq 1|$. Third, animals can keep track of head direction even at very high head velocities (e.g., up to $700^o/s$ in mice). This requires that the network be able to produce traveling pulses with high velocities. This can be easily achieved by using a fast time constant τ. However, with slow synapses such as NMDA or GABAb, a large-range and precise integration imposes several constraints on the synaptic connection parameters.

Next, we analyze dynamics of the network, and demonstrate that the above requirements on accurate integration can be achieved with carefully chosen synaptic parameters. We start from solving stationary solutions when $\Delta b = 0$, and then characterize the functional relationship between v and Δb.

4.3.1 Stationary solution

When the head is not moving ($\Delta b = 0$), both rings receive the same feedforward input. Let us assume that the synaptic connection parameters are chose such that each ring forms a stationary bump, which can be written, according to the symmetry, in the form

$$s_l^*(\theta) = [A\cos(\theta - \theta_0) - C]^+ \quad \text{and} \quad s_r^*(\theta) = [A\cos(\theta - \theta_0 + \beta) - C]^+, \tag{4.4}$$

where θ_0 represents the current head direction, β is the offset between the two bumps, and $*$ denotes steady states. Substituting this equation into the steady state equations of 4.1 $s_l^*(\theta) = f_l^*(\theta)$ and $s_r^*(\theta) = f_r^*(\theta)$, we derive that the parameters A, C and the offset β should satisfy

$$\beta = \arcsin(J_1/K_1 \sin\phi) - \psi \tag{4.5}$$

$$A = b_0[-(J_0 + K_0)f_0(\theta_c) - \cos\theta_c]^{-1} \tag{4.6}$$

$$1 = f_1(\theta_c)\left[J_1\cos\phi + (K_1^2 - J_1^2\sin^2\phi)^{1/2}\right], \tag{4.7}$$

where the functions f_0 and f_2 are given by

$$f_0(\theta_c) = \frac{1}{\pi}(\sin\theta_c - \theta_c\cos\theta_c), \quad f_1(\theta_c) = \frac{1}{2\pi}[\theta_c - \frac{1}{2}\sin(2\theta_c)].$$

Here θ_c is the critical half-width beyond which $f_l(\theta)$ is zero, that is, $A\cos\theta_c = C$ (The half-width is the same for both rings when $\Delta b = 0$.) The above set of equations fully characterizes the stationary bump solution. Eq. 4.5 determines the offset β between the two rings, Eq. 4.7 determines the threshold θ_c, and Eq. 4.6 determines the amplitude A. Once θ_c and A are known, C can be computed accordingly.

Because of the rotational symmetry inherent in the network, the stationary bump is marginally stable. Its location is not specified by the steady state equations, but rather by the initial conditions of the dynamics. The stationary bump is maintained due to the balance in the interaction received from each ring. When $\Delta b \neq 0$, the neural activities in one ring increase at the expense of the other ring, which causes an imbalance in the interaction between two rings and drives the activity bump to move around. The dependence of the traveling velocity v on Δb could be complicated because of the nonlinearity of the dynamics. We can characterize the dependence of v on $\Delta\hat{b}$ when $\Delta\hat{b}$ is small by perturbation.

4.3.2 Small head-velocity approximation

To study the traveling bump solution with velocity v, we transform the coordinate into a moving frame attached to the pulses traveling at velocity v. After the change of variables $S_l(\theta,t) = s_l(\theta - vt, t)$, the original traveling bump corresponds to a stationary solution in the new coordinates, satisfying

$$-\tau v S_l^{*\prime}(\theta) + S_l^*(\theta) = F_l(\theta) \quad \text{and} \quad -\tau v S_r^{*\prime}(\theta) + S_r^*(\theta) = F_r(\theta), \tag{4.8}$$

where $F_l(\theta) = f_l^*(\theta - vt, t)$ and $F_r(\theta) = f_r^*(\theta - vt, t)$.

When $\Delta\hat{b}$ is small, $\Delta b/b_0 \ll 1$, inside the excited regime ($F_l(\theta) > 0$, $F_r(\theta) > 0$), the solutions $S_l^*(\theta)$ and $S_r^*(\theta)$ can be viewed as perturbations of $s_l^*(\theta)$ and $s_r^*(\theta)$ of

Eq. 4.4 respectively,

$$S_l^*(\theta) = (A + \delta A_l)\cos(\theta - \theta_0) - (C + \delta C_l) \quad (4.9)$$

$$S_r^*(\theta) = (A + \delta A_r)\cos(\theta + \beta - \theta_0) - (C + \delta C_r). \quad (4.10)$$

Substitute Eqs. 4.9,4.10 into Eq. 4.8 and linearize the dynamics Eq. 4.8 around the solution Eq. 4.4. From this we find a linear equation for v with solution

$$v = J_1 \sin\phi(2\pi\tau A)^{-1}[(\theta_c + \sin(2\theta_c)/2)\delta A - 2\sin\theta_c \delta C], \quad (4.11)$$

where $\delta A \equiv \delta A_r - \delta A_l$ and $\delta C \equiv \delta C_r - \delta C_l$. To determine δA and δC, we linearize the dynamics of the variable $S_r^*(\theta - \beta) - S_l^*(\theta) = \delta A \cos(\theta - \theta_0) - \delta C$. After solving the linearized differential mode dynamics, we find

$$\delta A = 2[(k_3\theta_c - 1)k_2 - k_3 \sin\theta_c]^{-1}\Delta b \quad (4.12)$$

$$\delta C = 2k_2[(k_3\theta_c - 1)k_2 - k_3 \sin\theta_c]^{-1}\Delta b, \quad (4.13)$$

where $k_1 = [J_1 \cos\phi - K_1 \cos(\psi + \beta)]^{-1}$, $k_2 = [\theta_c + \sin(2\theta_c)/2 - 2\pi k_1](2\sin\theta_c)^{-1}$, and $k_3 = (J_0 - K_0)/\pi$. By substituting δA and δC into Eq. (4.11), we find

$$v = 2k_1 J_1 \sin\phi(\tau A)^{-1}[(k_3\theta_c - 1)k_2 - k_3 \sin\theta_c]^{-1}\Delta b. \quad (4.14)$$

Equation (4.14) relates the velocity v of the two bumps to the differential vestibular input Δb when $\Delta\hat{b} \ll 1$. This linear v-Δb relationship is plotted in Fig. 4-2 for various synaptic parameters, and is compared with the results obtained from numerical simulations. There is a good agreement of Eq. 4.14 only in a small region around the origin. As we have discussed, the desired v-Δb curve should be linear over entire range of $\Delta\hat{b}$ ($|\Delta\hat{b} \leq 1|$). Such a large-range linear regime is shown in Fig. 4-2d. We will address in section 4.4.1 on how to choose synaptic parameters to achieve this result.

One observation of the v-Δb curve is that v saturates in both ends when $|\Delta b|$ is

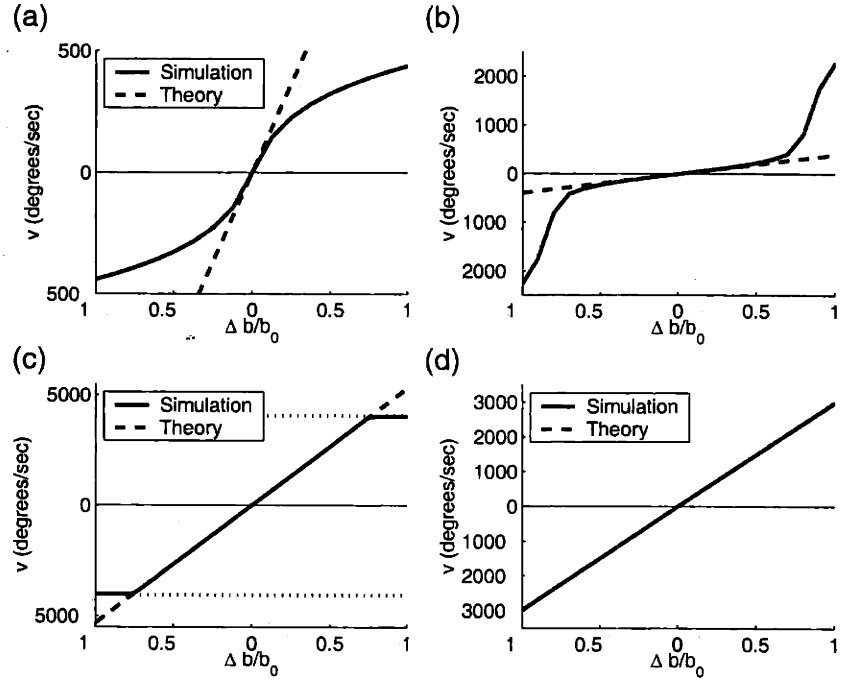

Figure 4-2: Moving bump velocity v as a function of the input Δb for different synaptic parameters. The slope is indicated as the dashed lines. All these curves saturate when $|\Delta b|$ is large, with the saturating velocity indicated by the dotted line. Panel (a) shows the result when $K_1 < J_1$ ($K_1 = 70$,$\phi = 60$). Panel (b) shows the result when $K_1 > J_1$ ($K_1 = 100$). The good linearity of v-Δb curve is achieved when $K_1 = J_1$ as indicated in Panel (c) and (d). The range of linearity is related to the synaptic variable K_0 as shown in Panel (c) for $K_0 = -20$ and Panel (d) for $K_0 = -5$. If not otherwise stated, the parameters used are $J_0 = -60$, $J_1 = 80$, $\phi = 80^o$, $\psi = 50^o$, and $\tau = 80$ms.

sufficiently large. The saturation velocity sets a limit on the largest velocity at which the bump can move. Next we calculate this saturation velocity, and discuss how to choose synaptic parameters such that the saturation velocity is above the highest head rotating velocity that an animal can produce.

4.3.3 Saturating velocity

When Δb is sufficiently large, at some point, the left ring becomes inactive. In this case, the network dynamics are determined only by neurons in the right ring. This still leads to a traveling bump solution in the right ring due to the asymmetric recurrent connections. However, here the moving velocity is fixed, independent of the exact value of Δb, as long as it is above a threshold value. This happens because the intra-ring synaptic connection strength can be separated into two terms: one is symmetric and the other one is anti-symmetric as follows:

$$\begin{aligned} W(\theta) &= J_0 + J_1 \cos(\theta - \phi) \\ &= J_0 + J_1 \cos\phi \cos\theta + \tan\phi \sin\theta \\ &= \tilde{W}_S(\theta) - \tan\phi \; \tilde{W}'_S(\theta), \end{aligned}$$

where $\tilde{W}_S(\theta) = J_0 + J_1 \cos\phi \cos\theta$. Now, let $s^*(\theta)$ be the steady solution of a ring network with symmetric connections $\tilde{W}_S(\theta)$. By differentiating, it follows that $s^*(\theta - \tan\phi/\tau t)$ is the solution of a ring network with connections $W(\theta)$. Hence, the saturating velocity v_{sat} is given by

$$v_{sat} = \tan\phi/\tau. \tag{4.15}$$

To make the saturation velocity high, we can use a small synaptic time constant τ, or choose the phase variable ϕ to be close to $\pi/2$, which seems necessary if slow synapses are involved in the integration of the head-direction system. For the parameters used in Fig. 4-2c,d , we use $\phi = 80^o$ and $\tau = 80$ms, and find $v_{sat} = 4062^o$/sec.

Similarly, when Δb is large negative over some threshold value, the right ring is

inactivated, and the traveling bump in the left ring moves in an opposite direction with the speed saturated at v_{sat} as well (See Fig. 4-2c).

So far, our characterization of the v-Δb relationship has been based on the special cases. Next, we analyze the dynamics of the double-ring network more systematically by Fourier transforming the original continuous field dynamics into the one described by a set of order parameters. Based on these order parameter dynamics, we analyze the solution for traveling bumps, and discuss how to choose synaptic parameters to achieve a good linearity like the one plotted in Fig. 4-2d.

4.4 Analysis in terms of Fourier modes

The double-ring network has two special properties that aids mathematical treatments. First, it is translation-invariant with period 2π. Second, the synaptic interaction function only involves the first two Fourier components. Therefore, we can simplify the original dynamics significantly by performing Fourier transform. Next we present the analysis, following similar treatments of Hansel and Sompolinsky [39].

Let us define the order parameters

$$r_i^0(t) = \int_{-\pi}^{\pi} s_i(\theta, t)\,(2\pi)^{-1} \mathrm{d}\theta \tag{4.16}$$

$$r_i^1(t) = \int_{-\pi}^{\pi} s_i(\theta, t) \exp[i(\theta - \Psi_i(t))]\,(2\pi)^{-1} \mathrm{d}\theta, \tag{4.17}$$

where $i = l, r$ is the index of the left and right rings. r_i^0 represents the mean synaptic activation of neurons in each ring. The phase $\Psi_i(t)$ is used to make $r_i^1(t)$ always being a real number, or in other words, r_i^1 is the amplitude of the second Fourier modes and $\Psi_i(t)$ is the phase (Fig. 4-3).

In terms these order parameters, the neural activities can be written as

$$f_l(\theta, t) = \left[I_l^0 + I_l^1 \cos(\theta - \Phi_l)\right]^+ \tag{4.18}$$

$$f_r(\theta, t) = \left[I_r^0 + I_r^1 \cos(\theta - \Phi_r)\right]^+, \tag{4.19}$$

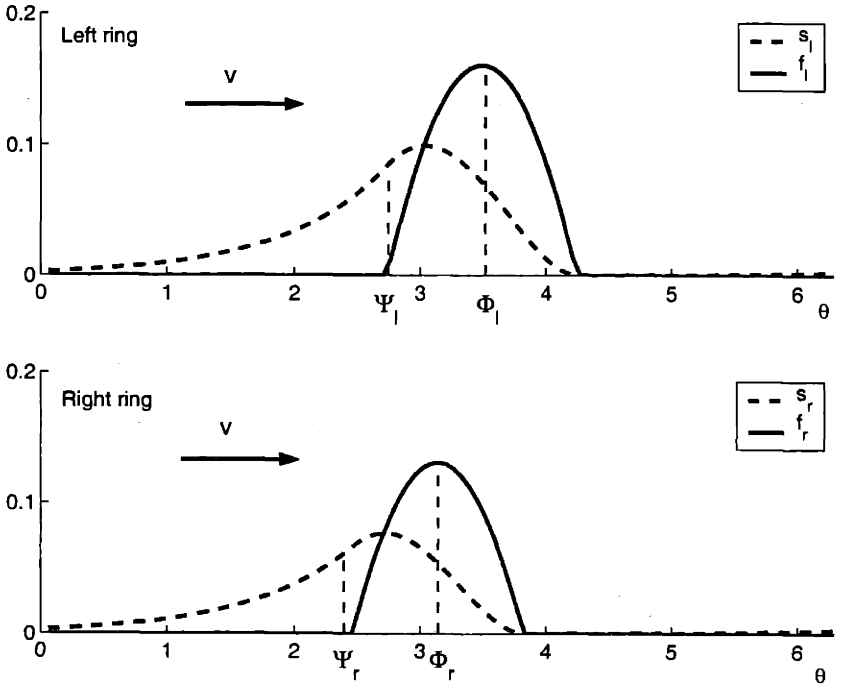

Figure 4-3: A snapshot of the traveling bumps in two rings. The dashed lines denote the synaptic variables and the solid lines represent the firing rate. Ψ_l and Ψ_r are the phases of the second order Fourier components of $s_l(\theta)$ and $s_r(\theta)$ respectively. The phase variable Φ_l is the defined by the peak location of $f_l(\theta)$, and ϕ_r is the peak location of $f_r(\theta)$.

where

$$I_l^0 = J_0 r_l^0 + K_0 r_r^0 + b_l \tag{4.20}$$

$$I_l^1 = J_1 r_l^1 \cos(\Phi_l - \Psi_l - \phi) + K_1 r_r^1 \cos(\Phi_l - \Psi_r + \psi) \tag{4.21}$$

$$I_r^0 = K_0 r_l^0 + J_0 r_r^0 + b_r \tag{4.22}$$

$$I_r^1 = K_1 r_l^1 \cos(\Phi_r - \Psi_l - \psi) + J_1 r_r^1 \cos(\Phi_r - \Psi_r + \phi). \tag{4.23}$$

The phase variables Φ_l and Φ_r are the peak of the neural activities in the left and right ring respectively (See Fig. 4-3). They satisfy

$$J_1 r_l^1 \sin(\Phi_l - \Psi_l - \phi) + K_1 r_r^1 \sin(\Phi_l - \Psi_r + \psi) = 0 \tag{4.24a}$$

$$K_1 r_l^1 \sin(\Phi_r - \Psi_l - \psi) + J_1 r_r^1 \sin(\Phi_r - \Psi_r + \phi) = 0. \tag{4.24b}$$

Let the half-width of positive $f_l(\theta, t)$ domain to be θ_l^c, and that of positive $f_r(\theta, t)$ domain to be θ_r^c. If f_i is a rectified bump, we have $\theta_i^c = \arccos(-I_i^0/I_i^1)$ for $i = l, r$. Next we assume θ_l^c and θ_r^c are given, in which case the original dynamics can be view as a linear one. Based on this, we perform Fourier transform of the network dynamics Eq. 4.1a-4.1b and derive the dynamics of the order parameters as follows

$$\tau \dot{r}_i^0 = -r_i^0 + I_i^1(t) f_0(\theta_i^c) \tag{4.25a}$$

$$\tau \dot{r}_i^1 = -r_i^1 + I_i^1(t) f_1(\theta_i^c) \cos(\Phi_i - \Psi_i) \tag{4.25b}$$

$$\tau r_i^1 \dot{\Psi}_i = I_i^1(t) f_1(\theta_i^c) \sin(\Phi_i - \Psi_i), \tag{4.25c}$$

where $i = l, r$. Similarly we can also write down the dynamics of the high order Fourier components. However, the above set of dynamics is decoupled from the higher order components, and can be solved independently. In particular, we are interested in the

traveling bump solutions in both rings, which can be described by

$$r_i^0 = I_i^1(t) f_0(\theta_i^c) \tag{4.26}$$

$$r_i^1 = I_i^1(t) f_1(\theta_i^c) \cos(\Phi_i - \Psi_i) \tag{4.27}$$

$$\tau v = \tan(\Phi_i - \Psi_i) \tag{4.28}$$

where v is the velocity of the traveling bumps. Given the vestibular input Δb, the moving velocity of the bumps can be determined numerically from the above set of algebra equations. Let $\bar{\beta} = \Phi_r - \Psi_l - \psi$, and $\theta = \Phi_r - \Psi_r = \Phi_l - \Psi_l = \text{atan}(\tau v)$. These set of algebra equations can be further simplified into a set of self-consistent four equations with θ, $\bar{\beta}$, θ_l^c, and θ_r^c being unknown variables:

$$K_1 r_l^1 \sin(\bar{\beta}) + J_1 r_r^1 \sin(\theta + \phi) = 0 \tag{4.29a}$$

$$J_1 r_l^1 \sin(\theta - \phi) + K_1 r_r^1 \sin(2\theta - \bar{\beta}) = 0 \tag{4.29b}$$

$$I_r^1 = K_1 r_l^1 \cos(\bar{\beta}) + J_1 r_r^1 \cos(\theta + \phi) \tag{4.29c}$$

$$I_l^1 = J_1 r_l^1 \cos(\theta - \phi) + K_1 r_r^1 \cos(2\theta - \bar{\beta}) \tag{4.29d}$$

where I_r^1, I_l^1, r_l^1, and r_r^1 can be written as functions of θ_l^c and θ_r^c.

$$I_r^1 = [b_r z_l - b_l K_0 f_0(\theta_l^c)][K_0^2 f_0(\theta_r^c) f_0(\theta_l^c) - z_l z_r]^{-1} \tag{4.30a}$$

$$I_l^1 = [b_l z_r - b_r K_0 f_0(\theta_r^c)][K_0^2 f_0(\theta_l^c) f_0(\theta_r^c) - z_l z_r]^{-1}, \tag{4.30b}$$

with $z_i = J_0 f_0(\theta_i^c) + \cos\theta_i^c$ for $i = l, r$, and the order parameter r_i^1 can be derived from Eq. 4.27. By using numerical methods, solutions of Eq. 4.29 can be determined.

4.4.1 Linearity when $J_1 = K_1$

One critical requirement on our model to achieve accurate integration is that the velocity of the traveling bumps should be linearly proportional to the vestibular input over all possible range of $\Delta\hat{b}$. Our simulation study shows that the network achieves an excellent linearity when $J_1 = K_1$ (See Fig. 4-2 for different choices of K_1 and J_1).

One typical result is shown in Fig. 4-2d. Next we present the analysis to justify this result.

As we have stated, the v-$\Delta\hat{b}$ curve typically follows sigmoidal shape, saturating in both ends when $|\Delta\hat{b}|$ is larger than a certain critical value $\Delta\hat{b}_c$. To measure the quality of the linearity over the nonsaturated regime of $\Delta\hat{b}$, we can simply compare the difference between the slope at the origin and the velocity-input ratio at the critical point, $v_{sat}/\Delta\hat{b}_c$.

First, we determine the critical $\Delta\hat{b}_c$ that gives rise to the saturation velocity. At the critical $\Delta\hat{b}_c$, one ring becomes inactive, and with loss of generality we assume the left ring is inactive, that is, $r_l^1 = 0$, $\theta_l^c = 0$. From Eq. 4.29a, we have $\theta = -\phi$ and $\bar{\beta} = 2\theta$, and therefore $I_r^1 = J_1 r_r^1$ and $I_l^1 = K_1 r_r^1$. Substituting these into Eq. 4.30, we find $\Delta\hat{b}_c$ to be

$$\Delta\hat{b}_c = [(K_0 - J_0)f_0(\theta_r^c) + K_1/J_1 - \cos\theta_r^c][(J_0 + K_0)f_0(\theta_r^c) + K_1/J_1 + \cos\theta_r^c]^{-1}, \quad (4.31)$$

where θ_r^c satisfies $J_1 f_1(\theta_r^c)\cos\phi = 1$, from which θ_r^c can be determined.

Typically θ_r^c can only be solved by numerical methods. However, an approximated value of θ_r^c can be found by asymptotic expansion of the function $f_1(x) \approx x^3/(3\pi)$. Similarly, function $f_0(x)$ can be approximated by $x^3/(3\pi)$. Therefore, $f_0(\theta_r^c) \approx 1/(J_1\cos\phi)$. Substituting this into Eq. 4.31, we find

$$\Delta\hat{b}_c \approx [K_0 - J_0 + J_1\cos\phi(K_1/J_1 - \cos\theta_r^c)][J_0 + K_0 + J_1\cos\phi(K_1/J_1 + \cos\theta_r^c)]^{-1}.$$

In our network, the phase variable ϕ is chosen to be close to $\pi/2$ to get a large saturation velocity, and J_0 is large negative to guarantee the stability of the network. Therefore, the terms multiplied $\cos\phi$ in the above is small and can be neglected in an approximation. After these considerations, the ratio between the velocity and the differential input at the critical value can be approximated by

$$v_{sat}/\Delta\hat{b}_c \approx \tan\phi(J_0 + K_0)[\tau(K_0 - J_0)]^{-1}. \quad (4.32)$$

On the other hand, when $J_1 = K_1$, the slope at the origin derived from Eq. 4.14 is

$$\begin{aligned} S_0 &= 2J_1 \sin\phi \sin\theta_c \tau^{-1}[-(J_0+K_0)f_0(\theta_c) - \cos\theta_c][\pi - \theta_c(J_0-K_0)]^{-1} \\ &\approx \tan\phi \sin\theta_c \tau^{-1}[-(J_0+K_0)\tan\phi][\pi + (J_0-K_0)\theta_c]^{-1} \\ &\approx \tan\phi(J_0+K_0)[\tau(K_0-J_0)]^{-1}. \end{aligned}$$

In the above approximation, we have used the asymptotic expansion of $f_0(\theta_c) \approx (2J_1\cos\phi)^{-1}$ and assumed $(J_0 - K_0)\theta_c \gg \pi$. The above result indicates that in an approximation, the ratio of the velocity-input at the critical value is the same as the slope S_0 at the origin. This explains the supreme linearity achieved when $J_1 = K_1$.

Another requirement on the integration of the head-direction system is that the linearity should cover all possible range of $\Delta\hat{b}$. From Eq. 4.32, we have the threshold differential input $\Delta\hat{b}_c \approx |(K_0 - J_0)/(K_0 + J_0)|$. Provided that J_0 is large negative, to guarantee $\Delta\hat{b}_c > 1$, K_0 has to be small negative or positive, which implies that the inter-ring global interaction should be weak inhibitory or excitatory. If K_0 is large negative, the saturation happens at a point when $\Delta\hat{b} < 1$ (Fig. 4-2c).

In the following, we consider dynamics in the case of $J_1 = K_1$.

4.4.2 Solution of the network when $J_1 = K_1$

When $J_1 = K_1$, the set of equations used to determine the thresholds and velocity can be simplified. After reordering and simplifying Eq. 4.29, we find that θ_l^c, θ_r^c and θ are determined by:

$$f_1(\theta_l^c)\sin(\theta-\phi) + f_1(\theta_r^c)\sin(\theta+\phi) = 0 \tag{4.33}$$

$$J_1 f_1(\theta_l^c)\cos\theta\cos(\theta-\phi) + K_1 f_1(\theta_r^c)\cos\theta\cos(\theta+\phi) = 1 \tag{4.34}$$

$$\begin{aligned} &[(J_0+K_0)(f_0(\theta_l^c)+f_0(\theta_r^c)) + \cos\theta_r^c + \cos\theta_l^c]\Delta\hat{b} = \\ &(J_0-K_0)(f_0(\theta_r^c) - f_0(\theta_l^c)) + \cos\theta_r^c - \cos\theta_l^c \end{aligned} \tag{4.35}$$

Variable θ can be solved from the Eq. 4.33, with a dependence on θ_l^c and θ_r^c,

$$v = \tan\theta/\tau = \tan\phi[f_1(\theta_l^c) - f_1(\theta_r^c)][\tau(f_1(\theta_l^c) + f_1(\theta_r^c))]^{-1}. \tag{4.36}$$

Substituting this into Eq. 4.34, we derive

$$J_1^2[f_1^2(\theta_l^c)+f_1^2(\theta_r^c)+2f_1(\theta_r^c)f_1(\theta_l^c)\cos(2\phi)] = \tan^2\phi[f_1(\theta_l^c)-f_1(\theta_r^c)]^2[f_1(\theta_l^c)+f_1(\theta_r^c)]^{-2}+1.$$

The above and Eq. 4.35 consist of two self-consistent equations, from which the threshold widths θ_l^c and θ_r^c can be determined for each differential input $\Delta\hat{b}$. Using Eq. 4.36, we can then determine the velocity of the moving bumps for each $\Delta\hat{b}$.

The results of this calculation are shown in Fig. 4-4a and are compared with the simulation results, which show a good consistency between the theoretical and simulation results.

The threshold width θ_l^c and θ_r^c characterize the width of the tuning curves of the head-directions. Its dependence on the head moving velocity can be measured experimentally. We plot the change of them as a function of $\Delta\hat{b}$ in Fig. 4-4b obtained from the above calculation. Besides threshold width, another characterization of the neural responses is the peak firing rates of the traveling bumps in each ring, which can be written as

$$P_r = I_r^1(1 - \cos\theta_r^c) \tag{4.37}$$

$$P_l = I_l^1(1 - \cos\theta_l^c), \tag{4.38}$$

for the right and left ring respectively. Here,

$$I_r^1 = I_l^1 = 2\Delta b\{(K_0 - J_0)[f_0(\theta_r^c) - f_0(\theta_l^c)] + \cos\theta_l^c - \cos\theta_r^c\}^{-1}.$$

The result is plotted in Fig. 4-4c,d. It shows that P_r and P_l is linearly related to the differential input $\Delta\hat{b}$.

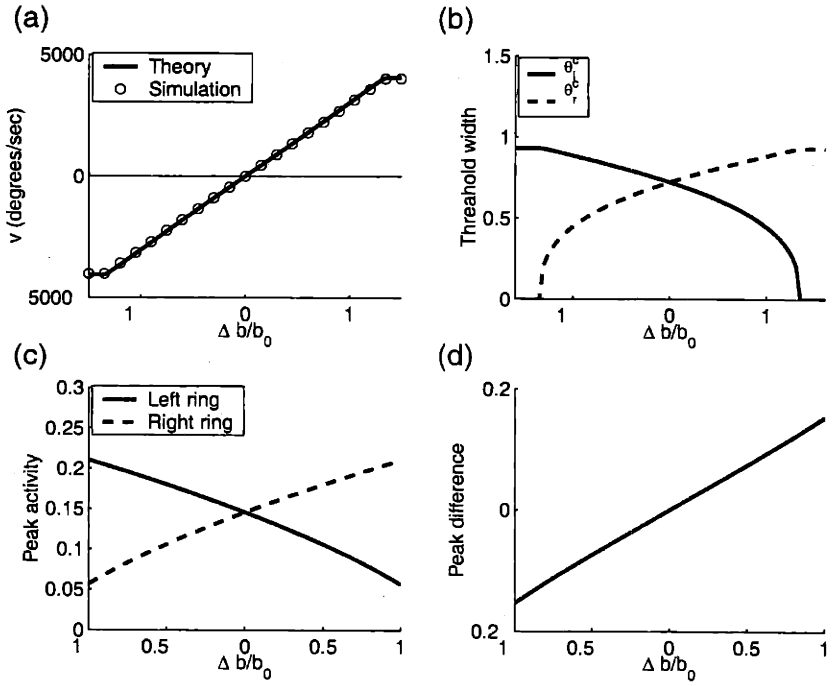

Figure 4-4: Results from the theoretical calculations when $J_1 = K_1$. The velocity of the traveling bumps as a function of the input is plotted in (a), which shows an excellent consistency between the theoretical and the simulation results. The width of the tuning curves are shown as a function of the input in (b). The peak firing rate of two bumps are modulated by Δb, with the peak rate for each ring shown in (c) and the difference between them shown in (d).

4.5 Stability

The network is translation-invariant. Potentially there are homogeneous solution to the dynamics. Assume $s_l(\theta) = u_l$ and $s_r(\theta) = u_r$, and substitute them into the Eq. 4.1. We find $u_l = J_0 u_l + K_0 u_r + b_l$ and $u_r = J_0 u_r + K_0 u_l + b_r$, which leads to the homogeneous solutions

$$u_r = b_0(1 - J_0 - K_0)^{-1} + \Delta b(1 - J_0 + K_0)^{-1} \tag{4.39}$$

$$u_l = b_0(1 - J_0 - K_0)^{-1} - \Delta b(1 - J_0 + K_0)^{-1}. \tag{4.40}$$

If $J_0 + K_0 < 1$, then u_r and u_l are both positive and so the homogeneous solutions exist.

Out network works in a regime where stationary bump develops when $\Delta b = 0$ and moves when $\Delta b \neq 0$. To guarantee the network work in such a regime, we have to choose synaptic parameters such that the homogeneous solution becomes unstable.

The stability of the homogeneous solution can be easily characterized by perturbing the dynamics around the homogeneous solution. In terms of first two Fourier components, the perturbed dynamics can be written as

$$\tau \delta \dot{r}_l^0 = (J_0 - 1)\delta r_l^0 + K_0 \delta r_r^0 \tag{4.41}$$

$$\tau \delta \dot{r}_r^0 = (J_0 - 1)\delta r_r^0 + K_0 \delta r_l^0, \tag{4.42}$$

for the first Fourier component, and

$$\tau \delta \dot{r}_l^1 = (J_1/2\cos\phi - 1)\delta r_l^1 + K_1/2\cos\phi \delta r_r^1 \tag{4.43}$$

$$\tau \delta \dot{r}_r^1 = (J_1/2\cos\phi - 1)\delta r_r^1 + K_1/2\cos\phi \delta r_l^1, \tag{4.44}$$

for the second Fourier component. For Eq. 4.41,4.42 to be stable, $J_0 + |K_0| < 2$. This is typically satisfied if we choose J_0 to be large negative. For r_l^1 and r_r^1 to be stable, it requires that $J_1 \cos\phi < 1$, provided that $J_1 = K_1$ as we have constrained. Therefore, to beak the stability of the homogeneous solution, we can choose $J_1 > 1/\cos\phi$.

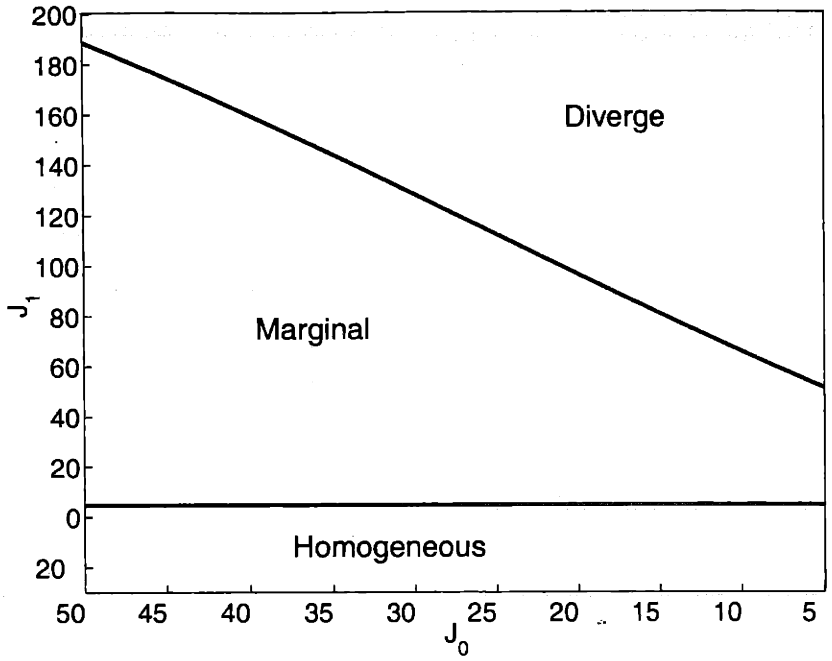

Figure 4-5: Phase diagram when $\Delta b = 0$. $K_1 = J_1$, and other parameters $K_0 = -5$, $\phi = 80^o$ and $\psi = 50^o$. The marginal phase is the desired parameter regime, in which a stationary bump develops, although the location of the bump can be arbitrary.

The stability of the stationary bump or traveling bumps can be analyzed by perturbing the dynamics Eq. 4.25 around the stationary or traveling bump solutions. However, the perturbed dynamics involves too many terms, and the details are not included in this paper. The phase diagram of the stationary bump obtained from simulation is shown in Fig. 4-5. This phase diagram does not include all possible solutions among all range of synaptic parameters. For example, if we we choose parameters such that $K_1 < J_1 \sin\phi$, the stationary bump solution will not exist and the network will yield a lurching bump solution whose shape keeps changing in time. For the stability of the traveling bump solutions, the bifurcation line from the homogeneous solution is the same as the one shown in Fig. 4-5, but the boundary where the traveling bump solution diverge is different for different $\Delta\hat{b}$.

4.6 ADN and POs neurons

Goodridge and Touretzky's integrator model was designed to emulate details of neuronal tuning as observed in the different areas of the head-direction system. Wondering whether the simple double ring studied here can also reproduce multiple tuning curves, we analyze simple read-out methods of the firing rates f_l and f_r. What we find is that two read-out methods can indeed approximate response behavior resembling that of ADN and POs neurons.

ADN neurons: By reading out firing rates using a maximum operation (See Fig. (4-6)), $z(\theta) = \max(f_r(\theta), f_l(\theta))$, anticipatory head-direction tuning arises due to the fact that there is an activity offset β between the two rings, equation (4.5). When the head turns to the right, the activity on the right ring is larger than on the left ring and so the tuning of $z(\theta)$ is biased to the right. Similarly, for left turns, $z(\theta)$ is biased to the left. Thus, the activity offset between the two rings leads to an anticipation time T for ADN neurons, see Fig. (4-6). Because, by assumption β is head-velocity independent, it follows that T is inversely proportional to head-velocity (assuming perfect integration), $T = \beta/(2v)$. In other words, the anticipation time tends to be smaller for fast head rotations and larger for slow head rotations.

POs neurons: By reading out the double ring activity as an average (See Fig. (4-6)), $y(\theta) = 1/2(f_r(\theta) + f_l(\theta))$, neurons in POs do not have any anticipation time: because averaging is a symmetric operation, all information about the direction of head rotations is lost.

4.7 Discussion on synaptic parameters

Here we discuss how the various connection parameters contribute to the double-ring network to function as an integrator. In particular we discuss how parameters have to be tuned in order to yield an integration that is large in Δb and in v.

- τ: By assumption the synaptic time constant τ is large. τ has the simplest effect of all parameters on the integrator properties. According to equation (4.14), τ

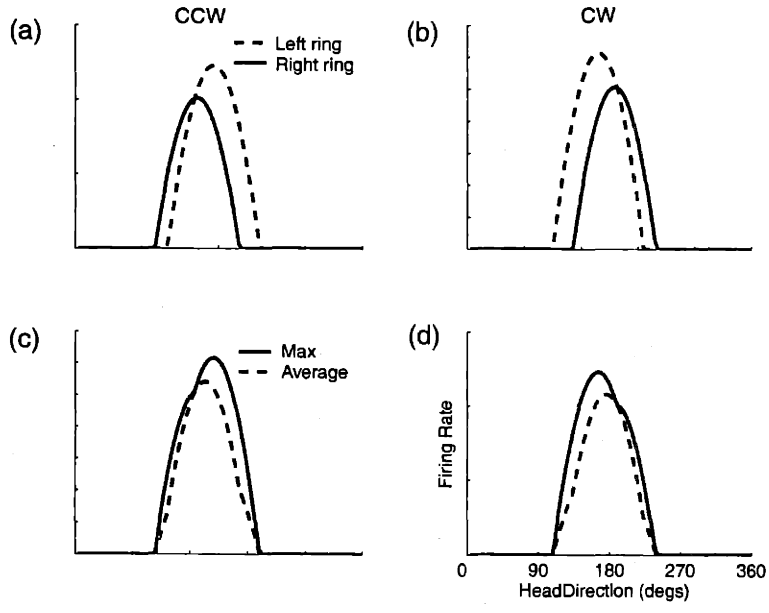

Figure 4-6: Snapshots of the activities on the two rings for counter-clock-wise head rotation (a) and clock-wise rotation (b) respectively. Reading out the activities by averaging and by a maximum operation (c, d).

scales the range of v. Notice that if τ were small, a large range of v could be trivially achieved. The art here is to achieve this with small τ.

- ϕ: The connection offset ϕ between neurons receiving similar vestibular input is the sole parameter besides τ that determines the saturating head-velocity, beyond which integration is impossible. According to equation (4.15), the saturating velocity is large if ϕ is close to 90^o (we want the saturating velocity to be large). In other words, for good integration, excitatory connections should be strongest (or inhibitory connections weakest) for neuron pairs with preferred head-directions differing by a little less than 90^o.

- ψ: The connection offset ψ between neurons receiving different vestibular input determines the anticipation time T of thalamic neurons. If ψ is large, then β, the activity offset in equation (4.5) is large. And, because β is proportional to T (assuming perfect integration), we conclude that ψ should preferentially be large (close to 90^o) if T is to be large. Notice that by equation (4.14), the range of v is not affected by ψ.

- K_0 and K_1: The inter-ring connections should be mainly weak inhibitory, or excitatory, which implies that K_0 should not be too negative. The intuitive reason is the following. We want the integration to be as linear in Δb as possible, which means that we want our linear expansions (4.9) and (4.10) to deviate as little as possible from (4.4). Hence, the differential gain between the two rings should be small, which is the case when the two rings excite each other. The inter-ring excitation makes sure, even for large values of Δb, that there are comparable activity levels on the two rings. This is one of the main points of this study.

- J_0 and J_1: The intra-ring connections should be mainly inhibitory, which implies that J_0 should be strongly negative. The reason for this is that inhibition is necessary to result in proper and stable integration. Since inhibition cannot come from the inter-ring connections, it has to come from J_0. Notice also that

according to equation (4.7), J_1 cannot be much larger than K_1. What would happen is that the persistent activity in the no head-movement case would become unstable.

4.8 Conclusion and remarks

We have proposed a new model for integration in the head-direction system with potentially slow synapses. The model is essentially a push-pull model with two populations of neurons receiving differential vestibular inputs. The difference in input breaks the balance between the interactions of two rings that is maintained during stationary head states, and causes activity bump developed in both rings to move around. With carefully chosen synaptic parameters, we demonstrate that the integration can be achieved with high precision.

We abstract the integration mechanism in the head-direction system with two rings. It is a convenient abstraction for mathematical treatments. It does, however, necessarily imply that there exists two nuclei of ring structures in the rat's brain. In fact, the network described by Goodridge and Touretzky [12] could, in a broad sense, be viewed as a double-ring network as well.

Our model requires specially organized network structures and carefully tuned synaptic parameters. Ideally the network structure and synaptic parameters should be able to self-organized and constantly corrected by some learning mechanism. We are currently investigating learning rules that give rise to this kind of synaptic connections. The results will be reported elsewhere.

Chapter 5

Spike-based learning rules and stabilization of persistent neural activity

Recent experiments have demonstrated types of synaptic plasticity that depend on the temporal ordering of presynaptic and postsynaptic spiking. At cortical [16] and hippocampal [17] synapses, long-term potentiation is induced by repeated pairing of a presynaptic spike and a succeeding postsynaptic spike, while long-term depression results when the order is reversed. The dependence of the change in synaptic strength on the difference $\Delta t = t_{post} - t_{pre}$ between postsynaptic and presynaptic spike times has been measured quantitatively. This *pairing function*, sketched in Figure 5-1A, has positive and negative lobes correspond to potentiation and depression, and a width of tens of milliseconds. We will refer to synaptic plasticity associated with this pairing function as differential Hebbian plasticity—*Hebbian* because the conditions for potentiation are as predicted by Hebb [80], and *differential* because it is driven by the difference between the opposing processes of potentiation and depression.

The pairing function of Figure 5-1A is not characteristic of all synapses. For example, an opposite temporal dependence has been observed at electrosensory lobe

[0]This chapter is based on the article with the same title in *Advances in Neural Information Processing Systems* 12, 199-205 (2000) by Xie and Seung.

synapses of electric fish [81]. As shown in Figure 5-1B, these synapses depress when a presynaptic spike is followed by a postsynaptic one, and potentiate when the order is reversed. We will refer to this as differential anti-Hebbian plasticity.

According to these experiments, the maximum ranges of the differential Hebbian and anti-Hebbian pairing functions are roughly 20 and 40 ms, respectively. This is fairly short, and seems more compatible with descriptions of neural activity based on spike timing rather than instantaneous firing rates [82, 83]. In fact, we will show that there are some conditions under which spike-based learning rules can be approximated by rate-based learning rules.

5.1 Introduction

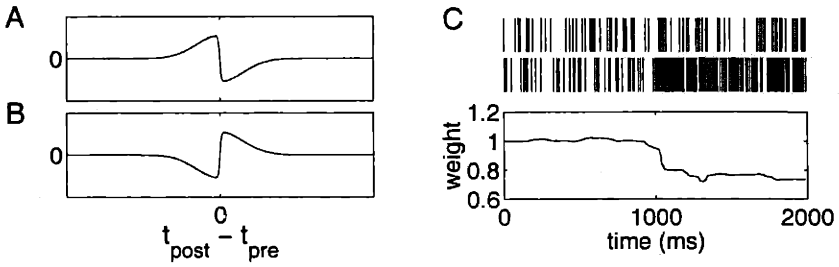

Figure 5-1: (A) Pairing function for differential Hebbian learning. The change in synaptic strength is plotted versus the time difference between postsynaptic and presynaptic spikes. (B) Pairing function for differential anti-Hebbian learning. (C) Differential anti-Hebbian learning is driven by changes in firing rates. The synaptic learning rule of Eq. (5.1) is applied to two Poisson spike trains. The synaptic strength remains roughly constant in time, except when the postsynaptic rate changes.

The pairing functions of Figures 5-1A and 5-1B lead to rate-based learning rules like those traditionally used in neural networks, except that they depend on temporal derivatives of firing rates as well as firing rates themselves. We will argue that the differential anti-Hebbian learning rule of Figure 5-1B could be a general mechanism for tuning the strength of positive feedback in networks that maintain a short-term memory of an analog variable in persistent neural activity[84]. A number of recurrent

network models have been proposed to explain memory-related neural activity in motor [85] and prefrontal[86] cortical areas, as well as the head direction system [10] and oculomotor integrator[87, 88, 89]. All of these models require precise tuning of synaptic strengths in order to maintain continuously variable levels of persistent activity. As a simple illustration of tuning by differential anti-Hebbian learning, a model of persistent activity maintained by an integrate-and-fire neuron with an excitatory autapse is studied.

5.2 Spike-based learning rule

Pairing functions like those of Figure 5-1 have been measured using repeated pairing of a single presynaptic spike with a single postsynaptic spike. Quantitative measurements of synaptic changes due to more complex patterns of spiking activity have not yet been done. We will assume a simple model in which the synaptic change due to arbitrary spike trains is the sum of contributions from all possible pairings of presynaptic with postsynaptic spikes. The model is unlikely to be an exact description of real synapses, but could turn out to be approximately valid.

We will write the spike train of the ith neuron as a series of Dirac delta functions, $s_i(t) = \sum_n \delta(t - T_i^n)$, where T_i^n is the nth spike time of the ith neuron. The synaptic weight from neuron j to i at time t is denoted by $W_{ij}(t)$. Then the change in synaptic weight induced by presynaptic spikes occurring in the time interval $[0, T]$ is modeled as

$$W_{ij}(T + \lambda) - W_{ij}(\lambda) = \int_0^T dt_j \int_{-\infty}^{\infty} dt_i \, f(t_i - t_j) s_i(t_i) \, s_j(t_j) \tag{5.1}$$

Each presynaptic spike is paired with all postsynaptic spikes produced before and after. For each pairing, the synaptic weight is changed by an amount depending on the pairing function f. The pairing function is assumed to be nonzero inside the interval $[-\tau, \tau]$, and zero outside. We will refer to τ as the *pairing range.*

According to our model, each presynaptic spike results in induction of plasticity only after a latency λ. Accordingly, the arguments $T + \lambda$ and λ of W_{ij} on the left hand side of the equation are shifted relative to the limits T and 0 of the integral on

the right hand side. We will assume that the latency λ is greater than the pairing range τ, so that W_{ij} at any time is only influenced by presynaptic and postsynaptic spikes that happened before that time, and therefore the learning rule is causal.

5.3 Relation to rate-based learning rules

The learning rule of Eq. (5.1) is driven by correlations between presynaptic and postsynaptic activities. This dependence can be made explicit by making the change of variables $u = t_i - t_j$ in Eq. (5.1), which yields

$$W_{ij}(T+\lambda) - W_{ij}(\lambda) = \int_{-\tau}^{\tau} du f(u) C_{ij}(u) \tag{5.2}$$

where we have defined the cross-correlation

$$C_{ij}(u) = \int_0^T dt\, s_i(t+u)\, s_j(t)\ . \tag{5.3}$$

and made use of the fact that f vanishes outside the interval $[-\tau, \tau]$. Our immediate goal is to relate Eq. (5.2) to learning rules that are based on the cross-correlation between firing rates,

$$C_{ij}^{rate}(u) = \int_0^T dt\, \nu_i(t+u)\, \nu_j(t) \tag{5.4}$$

There are a number of ways of defining instantaneous firing rates. Sometimes they are computed by averaging over repeated presentations of a stimulus. In other situations, they are defined by temporal filtering of spike trains. The following discussion is general, and should apply to these and other definitions of firing rates.

The "rate correlation" is commonly subtracted from the total correlation to obtain the "spike correlation" $C_{ij}^{spike} = C_{ij} - C_{ij}^{rate}$. To derive a rate-based approximation to the learning rule (5.2), we rewrite it as

$$W_{ij}(T+\lambda) - W_{ij}(\lambda) = \int_{-\tau}^{\tau} du\, f(u) C_{ij}^{rate}(u) + \int_{-\tau}^{\tau} du\, f(u) C_{ij}^{spike}(u) \tag{5.5}$$

and simply neglect the second term. Shortly we will discuss the conditions under

which this is a good approximation. But first we derive another form for the first term by applying the approximation $\nu_i(t+u) \approx \nu_i(t) + u\dot{\nu}_i(t)$ to obtain

$$\int_{-\tau}^{\tau} du f(u) C_{ij}^{rate}(u) \approx \int_0^T dt[\beta_0 \nu_i(t) + \beta_1 \dot{\nu}_i(t)]\nu_j(t) \tag{5.6}$$

where we define

$$\beta_0 = \int_{-\tau}^{\tau} du f(u) \qquad \beta_1 = \int_{-\tau}^{\tau} du\, u f(u) \tag{5.7}$$

This approximation is good when firing rates vary slowly compared to the pairing range τ. The learning rule depends on the postsynaptic rate through $\beta_0\nu_i + \beta_1\dot{\nu}_i$. When the first term dominates the second, then the learning rule is the conventional one based on correlations between firing rates, and the sign of β_0 determines whether the rule is Hebbian or anti-Hebbian.

In the remainder of the paper, we will discuss the more novel case where $\beta_0 = 0$. This holds for the pairing functions shown in Figures 5-1A and 5-1B, which have positive and negative lobes with areas that exactly cancel in the definition of β_0. Then the dependence on postsynaptic activity is purely on the time derivative of the firing rate. Differential Hebbian learning corresponds to $\beta_1 > 0$ (Figure 5-1A), while differential anti-Hebbian learning leads to $\beta_1 < 0$ (Figure 5-1B). To summarize the $\beta_0 = 0$ case, the synaptic changes due to rate correlations are approximated by

$$\dot{W}_{ij} \propto \dot{\nu}_i \nu_j \quad \text{(diff. Hebbian)} \qquad \dot{W}_{ij} \propto -\dot{\nu}_i \nu_j \quad \text{(diff. anti-Hebbian)} \tag{5.8}$$

for slowly varying rates. These formulas imply that a constant postsynaptic firing rate causes no net change in synaptic strength. Instead, changes in rate are required to induce synaptic plasticity.

To illustrate this point, Figure 5-1C shows the result of applying differential anti-Hebbian learning to two spike trains. The presynaptic spike train was generated by a 50 Hz Poisson process, while the postsynaptic spike train was generated by an inhomogeneous Poisson process with rate that shifted from 50 Hz to 200 Hz at 1

sec. Before and after the shift, the synaptic strength fluctuates but remains roughly constant. But the upward shift in firing rate causes a downward shift in synaptic strength, in accord with the sign of the differential anti-Hebbian rule in Eq. (5.8).

The rate-based approximation works well for this example, because the second term of Eq. (5.5) is not so important. Let us return to the issue of the general conditions under which this term can be neglected. With Poisson spike trains, the spike correlations $C_{ij}^{spike}(u)$ are zero in the limit $T \to \infty$, but for finite T they fluctuate about zero. The integral over u in the second term of (5.5) dampens these fluctuations. The amount of dampening depends on the pairing range τ, which sets the limits of integration. In Figure 5-1C we used a relatively long pairing range of 100 ms, which made the fluctuations small even for small T. On the other hand, if τ were short, the fluctuations would be small only for large T. Averaging over large T is relevant when the amplitude of f is small, so that the rate of learning is slow. In this case, it takes a long time for significant synaptic changes to accumulate, so that plasticity is effectively driven by integrating over long time periods T in Eq. (5.1).

In the brain, nonvanishing spike correlations are sometimes observed even in the $T \to \infty$ limit, unlike with Poisson spike trains. These correlations are often roughly symmetric about zero, in which case they should produce little plasticity if the pairing functions are antisymmetric as in Figures 5-1A and 5-1B. On the other hand, if the spike correlations are asymmetric, they could lead to substantial effects[83].

5.4 Effects in recurrent network dynamics

The learning rules of Eq. (5.8) depend on both presynaptic and postsynaptic rates, like learning rules conventionally used in neural networks. They have the special feature that they depend on time derivatives, which has computational consequences for recurrent neural networks of the form

$$\dot{x}_i + x_i = \sum_j W_{ij}\sigma(x_j) + b_i \tag{5.9}$$

Such classical neural network equations can be derived from more biophysically realistic models using the method of averaging[90] or a mean field approximation[91]. The firing rate of neuron j is conventionally identified with $\nu_j = \sigma(x_j)$.

The cost function $E(\{x_i\};\{W_{ij}\}) = \frac{1}{2}\sum_i \dot{\nu}_i^2$ quantifies the amount of drift in firing rate at the point $x_1, \ldots, x_N$ in the state space of the network. If we consider $\dot{\nu}_i$ to be a function of x_i and W_{ij} defined by (5.9), then the gradient of the cost function with respect to W_{ij} is given by $\partial E/\partial W_{ij} = \sigma'(x_i)\dot{\nu}_i\nu_j$. Assuming that σ is a monotonically increasing function so that $\sigma'(x_i) > 0$, it follows that the differential Hebbian update of (5.8) increases the cost function, and hence increases the magnitude of the drift velocity. In contrast, the differential anti-Hebbian update decreases the drift velocity. This suggests that the differential anti-Hebbian update could be useful for creating fixed points of the network dynamics (5.9).

5.5 Persistent activity in a spiking autapse model

The preceding arguments about drift velocity were based on approximate rate-based descriptions of learning and network dynamics. It is important to implement spike-based learning in a spiking network dynamics, to check that our approximations are valid. Therefore we have numerically simulated the simple recurrent circuit of integrate-and-fire neurons shown in Figure 5-2. The core of the circuit is the "memory neuron," which makes an excitatory autapse onto itself. It also receives synaptic input from three input neurons: a tonic neuron, an excitatory burst neuron, and an inhibitory burst neuron. It is known that this circuit can store a short-term memory of an analog variable in persistent activity, if the strengths of the autapse and tonic synapse are precisely tuned[92]. Here we show that this tuning can be accomplished by the spike-based learning rule of Eq. (5.1), with a differential anti-Hebbian pairing function like that of Figure 5-1B.

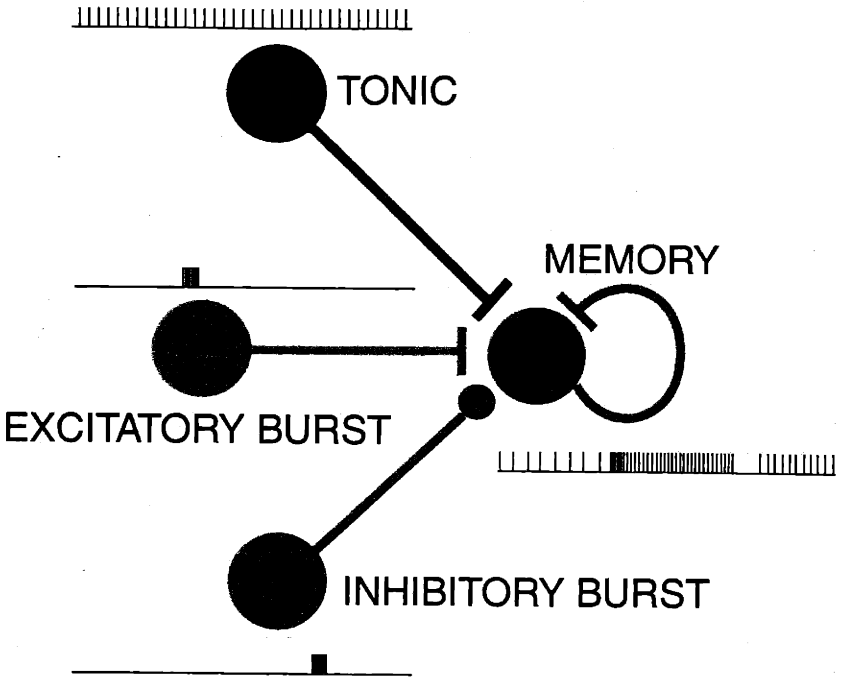

Figure 5-2: Circuit diagram for autapse model

The memory neuron is described by the equations

$$C_m \frac{dV}{dt} = -g_L(V - V_L) - g_E(V - V_E) - g_I(V - V_I) \tag{5.10}$$

$$\tau_{syn} \frac{ds}{dt} + s = \alpha_s \sum_n \delta(t - T_n) \tag{5.11}$$

where V is the membrane potential. When V reaches V_{thres}, a spike is considered to have occurred, and V is reset to V_{reset}. Each spike at time T_n causes a jump in the synaptic activation s of size α_s/τ_{syn}, after which s decays exponentially with time constant τ_{syn} until the next spike.

The synaptic conductances of the memory neuron are given by

$$g_E = Ws + W_0 s_0 + W_+ s_+ \qquad g_I = W_- s_- \tag{5.12}$$

The term Ws is recurrent excitation from the autapse, where W is the strength of the autapse. The synaptic activations s_0, s_+, and s_- of the tonic, excitatory burst,

and inhibitory burst neurons are governed by equations like (5.10) and (5.11), with a few differences. These neurons have no synaptic input; their firing patterns are instead determined by applied currents $I_{app,0}$, $I_{app,+}$ and $I_{app,-}$. The tonic neuron has a constant applied current, which makes it fire repetitively at roughly 20 Hz (Figure 5-3). For the excitatory and inhibitory burst neurons the applied current is normally zero, except for brief 100 ms current pulses that cause bursts of action potentials.

As shown in Figure 5-3, if the synaptic strengths W and W_0 are arbitrarily set before learning, the burst neurons cause only transient changes in the firing rate of the memory neuron. After applying the spike-based learning rule (5.1) to tune both W and W_0, the memory neuron is able to maintain persistent activity. During the interburst intervals (from λ after one burst until λ before the next), we made synaptic changes using the differential anti-Hebbian pairing function $f(t) = -A\sin(\pi t/\tau)$ for spike time differences in the range $[-\tau, \tau]$ with $A = 1.5 \times 10^{-4}$ and τ=λ=120 ms. The resulting increase in persistence time can be seen in Figure 5-4A, along with the values of the synaptic weights versus time.

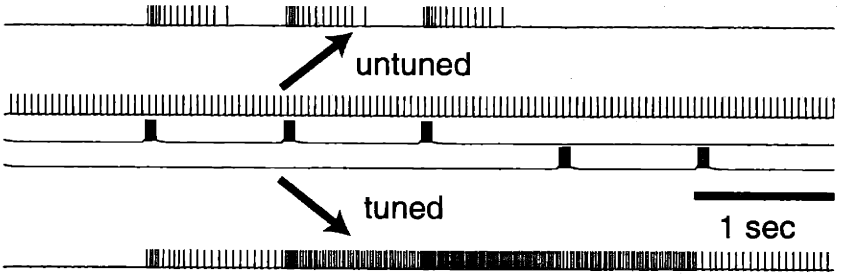

Figure 5-3: Untuned and tuned autapse activity. The middle three traces are the membrane potentials of the three input neurons in Figure 5-2 (spikes are drawn at the reset times of the integrate-and-fire neurons). Before learning, the activity of the memory neuron is not persistent, as shown in the top trace. After the spike-based learning rule (5.1) is applied to the synaptic weights W and W_0, then the burst inputs cause persistent changes in activity. $C_m = 1$ nF, $g_L = 0.025$ μS, $V_L = -70$ mV, $V_E = 0$ mV, $V_I = -70$ mV, $V_{thres} = -52$ mV, $V_{reset} = -59$ mV, $\alpha_s = 1$, $\tau_{syn} = 100$ ms, $I_{app,0} = 0.5203$ nA, $I_{app,\pm} = 0$ or 0.95 nA, $\tau_{syn,0} = 100$ ms, $\tau_{syn,+} = \tau_{syn,-} = 5$ ms, $W_+ = 0.1$, $W_- = 0.05$.

To quantify the performance of the system at maintaining persistent activity, we

determined the relationship between $d\nu/dt$ and ν using a long sequence of interburst intervals, where ν was defined as the reciprocal of the interspike interval. If W and W_0 are fixed at optimally tuned values, there is still a residual drift, as shown in Figure 5-4B. But if these parameters are allowed to adapt continuously, even after good tuning has been achieved, then the residual drift is even smaller in magnitude. This is because the learning rule tweaks the synaptic weights during each interburst interval, reducing the drift for that particular firing rate.

Autapse learning is driven by the autocorrelation of the spike train, rather than a cross-correlation. The peak in the autocorrelogram at zero lag has no effect, since the pairing function is zero at the origin. Since the autocorrelation is zero for small time lags, we used a fairly large pairing range in our simulations. In a recurrent network of many neurons, a shorter pairing range would suffice, as the cross-correlation does not vanish near zero.

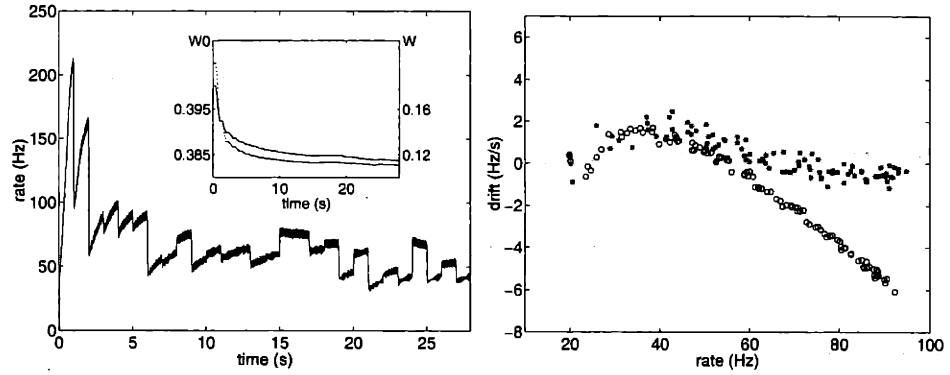

Figure 5-4: Tuning the autapse. (A) The persistence time of activity increases as the weights W and W_0 are tuned. Each transition is driven by pseudorandom bursts of input (B) Systematic relationship between drift $d\nu/dt$ in firing rate and ν, as measured from a long sequence of interburst intervals. If the weights are continuously fine-tuned ('*') the drift is less than with fixed well-tuned weights ('o').

5.6 Discussion

We have shown that differential anti-Hebbian learning can tune a recurrent circuit to maintain persistent neural activity. This behavior can be understood by reducing the spike-based learning rule (5.1) to the rate-based learning rules of Eqs. (5.6) and (5.8). The rate-based approximations are good if two conditions are satisfied. First, the pairing range must be large, or the rate of learning must be slow. Second, spike synchrony must be weak, or have little effect on learning due to the shape of the pairing function.

The differential anti-Hebbian pairing function results in a learning rule that uses $-\dot{\nu}_i$ as a negative feedback signal to reduce the amount of drift in firing rate, as illustrated by our simulations of an integrate-and-fire neuron with an excitatory autapse. More generally, the learning rule could be relevant for tuning the strength of positive feedback in networks that maintain a short-term memory of an analog variable in persistent neural activity[84]. For example, the learning rule could be used to improve the robustness of the oculomotor integrator[87, 88, 89] and head direction system[10] to mistuning of parameters. In deriving the differential forms of the learning rules in (5.8), we assumed that the areas under the positive and negative lobes of the pairing function are equal, so that the integral defining β_0 vanishes. In reality, this cancellation might not be exact. Then the ratio of β_1 and β_0 would limit the persistence time that can be achieved by the learning rule.

Both the oculomotor integrator and the head direction system are also able to integrate vestibular inputs to produce changes in activity patterns. The problem of finding generalizations of the present learning rules that train networks to integrate is still open[84].

Chapter 6

Equivalence of backpropagation and contrastive Hebbian learning in a layered network

6.1 Introduction

Backpropagation and contrastive Hebbian learning (CHL) are two supervised learning algorithms for training networks with hidden neurons. They are of interest, because they are generally applicable to wide classes of network architectures. In backpropagation [18, 19], an error signal for the output neurons is computed and propagated back into the hidden neurons through a separate teacher network. Synaptic weights are updated based on the product between the error signal and network activities. CHL updates the synaptic weights based on the steady states of neurons in two different phases – one with the output neurons clamped to the desired values and the other one with the output neurons free [20, 93]. Clamping the output neurons causes the hidden neurons to change their activities, and this change constitutes the basis for the CHL update rule.

CHL was originally formulated for the Boltzmann machine [21], and was extended

[0]This is a collaborative work with H. S. Seung.

later to deterministic networks [94, 95], in which case it can be interpreted as a mean-field approximation of the Boltzmann machine learning algorithm. However, this interpretation is not necessary, and CHL can be formulated purely for deterministic networks [20, 93]. Compared to backpropagation, CHL appears to be quite different. Backpropagation is typically implemented in feedforward networks, whereas CHL is implemented in networks with feedback. Backpropagation is an algorithm driven by error, whereas CHL is a Hebbian-type algorithm, with update rules based on the correlation of pre- and post-synaptic activities. There has been some work to relate CHL to the general framework of backpropagation [96, 97, 98]. However, a direct link between them has been lacking.

To investigate the relationship between these two algorithms, we consider a special network for which CHL and backpropagation are equivalent. This is a multilayer perceptron to which weak feedback connections have been added and with output neurons that are linear. The equivalence holds because in CHL clamping the output neurons at their desired values causes the hidden neurons to change their activities, and this change turns out to be equal to the error signal spread by backpropagation, except for a scalar factor.

6.2 The learning algorithms

In this section, we describe the backpropagation and CHL algorithms. Backpropagation is in the standard form, implemented in a multilayer perceptron [18]. CHL is formulated in a layered network with feedback connections between neighboring layers of neurons. It is an extension of the typical CHL algorithm formulated for recurrent symmetric networks [20].

6.2.1 Backpropagation

Consider a multilayer perceptron with $L+1$ layers of neurons and L layers of synaptic weights (Figure 6-1A). The activities of the kth layer of neurons are denoted by the vector x_k, their biases by the vector b_k, and the synaptic connections from layer $k-1$

to layer k by the matrix W_k. All neurons in the kth layer are assumed to have the same transfer function f_k, but this transfer function may vary from layer to layer. In particular, we will be interested in the case where f_L is linear, though the other transfer functions may be nonlinear. In the basic definition, f_k acts on a scalar and returns a scalar. However, we will generally use it to act on a vector, in which case it returns a vector, operating component by component. f_k' is the derivative of f_k with respect to its argument. Similar to f_k, when f_k' acts on a vector, it returns a vector as well. We assume that f_k is monotonically increasing.

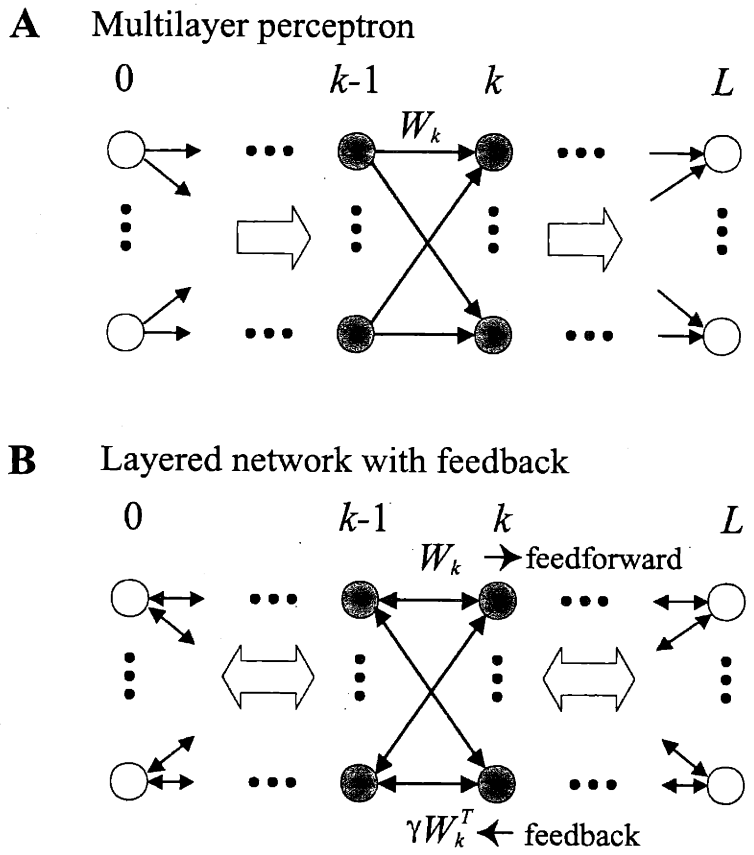

Figure 6-1: Diagram on the network structures of the multilayer perceptron (A) and the layered network with feedback connections (B). Layer 0 is the input, layer L is the output, and the others are hidden layers. The forward connections are the same for both networks. In (B), there exist feedback connections between neighboring layers of neurons.

Backpropagation learning is implemented by repeating the following steps for each example in a training set of input-output pairs.

1. In the forward pass,

$$x_k = f_k(W_k x_{k-1} + b_k) \tag{6.1}$$

is evaluated for $k = 1$ to L, thereby mapping the input x_0 to the output x_L.

2. The desired output d of the network, provided by some teacher, is compared with the actual output x_L to compute an error signal

$$y_L = D_L(d - x_L)$$

The matrix $D_k \equiv \text{diag}\{f_k'(W_k x_{k-1} + b_k)\}$ is defined by placing the components of the vector $f_k'(W_k x_{k-1} + b_k)$ in the diagonal entries of a matrix.

3. The error signal is propagated backwards from the output layer by evaluating

$$y_{k-1} = D_{k-1} W_k^T y_k$$

for $k = L$ to 2.

4. The weight update

$$\Delta W_k = \eta y_k x_{k-1}^T \tag{6.2}$$

is made for $k = 1$ to L, where $\eta > 0$ is a parameter controlling the learning rate.

6.2.2 Contrastive Hebbian learning

To formulate CHL, we consider a modified network, in which in addition to the feedforward connections from layer $k-1$ to layer k, there are also feedback connections between neighboring layers (Figure 6-1B). The feedback connections are assumed to be symmetric with the feedforward connections, except that they are multiplied by a

positive factor γ. In other words, the matrix γW_k^T contains the feedback connections from layer k back to layer $k-1$.

CHL is implemented by repeating the following steps for each example of the training set.

1. The input layer x_0 is held fixed, and the dynamical equations

$$\frac{dx_k}{dt} + x_k = f_k(W_k x_{k-1} + \gamma W_{k+1}^T x_{k+1} + b_k) \tag{6.3}$$

for $k = 1$ to L are run until convergence to a fixed point. The case $k = L$ is defined by setting $x_{L+1} = 0$ and $W_{L+1} = 0$. Convergence to a fixed point is guaranteed under rather general conditions, to be shown later. This is called the free state of the network, and is denoted by $\check{x}_k$ for the kth layer neurons.

2. The anti-Hebbian update

$$\Delta W_k = -\eta\gamma^{k-L}\check{x}_k\check{x}_{k-1}^T$$

is made for $k = 1, ..., L$.

3. The output layer x_L is clamped at the desired value d, and the dynamical equation 6.3 for $k = 1$ to $L-1$ is run until convergence to a fixed point. This is called the clamped state, and is denoted by $\hat{x}_k$ for the kth layer neurons.

4. The Hebbian update

$$\Delta W_k = \eta\gamma^{k-L}\hat{x}_k\hat{x}_{k-1}^T$$

is made for $k = 1, ..., L$.

Alternatively, the weight updates could be combined, and made after both clamped and free states are computed,

$$\Delta W_k = \eta\gamma^{k-L}(\hat{x}_k\hat{x}_{k-1}^T - \check{x}_k\check{x}_{k-1}^T). \tag{6.4}$$

This form is the one used in our analysis.

This version of CHL should look familiar to anyone who knows the conventional version, implemented in symmetric networks. It will be derived in section 6.4, but first we prove its equivalence to the backpropagation algorithm.

6.3 Equivalence in the limit of weak feedback

Next, we prove that CHL in equation 6.4 is equivalent to the backpropagation algorithm in equation 6.2, provided that the feedback is sufficiently weak and the output neurons are linear.

In notation, x_k, $\hat{x}_k$, and $\check{x}_k$ represent the kth layer activities of the feedforward network, the clamped state and the free state respectively. We consider the case of weak feedback connections, $\gamma \ll 1$, and use "$\approx$" symbol to mean that terms of higher order in γ have been neglected and "$\sim$" to denote the order.

The proof consists of the following four steps:

1. Show that the difference between the feedforward and free states is of order γ,

$$\delta\check{x}_k \equiv \check{x}_k - x_k \sim \gamma \tag{6.5}$$

   for all $k = 1, ..., L$.

2. Show that in the limit of weak feedback, the difference between the clamped and free states satisfies the following iterative relationship,

$$\delta x_k \equiv \hat{x}_k - \check{x}_k \approx \gamma D_k W_{k+1}^T \delta x_{k+1}, \tag{6.6}$$

   for $k = 1, ..., L-1$, and $\delta x_L = d - \check{x}_L$.

3. Show that if the output neurons are linear, δx_k is related to the error signal in backpropagation through

$$\delta x_k \approx \gamma^{L-k} y_k. \tag{6.7}$$

4. Finally, show that the CHL update can be approximated by

$$\Delta W_k \approx \eta y_k x_{k-1}^T. \tag{6.8}$$

In the CHL algorithm, clamping the output layer causes changes in the output neurons to spread backward to the hidden layers, because of the feedback connections. Hence, the new clamped state differs from the free state over the entire network, including the hidden neurons. Equation 6.6 states that δx_k decays exponentially with distance from the output layer of the network. This is because the feedback is weak, so that δx_k is reduced from δx_{k+1} by a factor of γ.

Remarkably, as indicated in equation 6.7, the difference between the clamped and free states is equivalent to the error signal y_k computed in the backward pass of backpropagation, except for a factor of γ^{L-k}, when the output neurons are linear. Moreover, this factor annihilates the factor of γ^{k-L} in the CHL rule of equation 6.4, resulting in the update rule equation 6.8.

6.3.1 Proof

To prove the first step, we start from the steady state equation of the free phase,

$$\check{x}_k = f_k(W_k \check{x}_{k-1} + b_k + \gamma W_{k+1}^T \check{x}_{k+1}),$$

for $k = 1, ..., L-1$. Subtracting this equation from equation 6.1 and performing Taylor expansion, we derive

$$\begin{aligned} \delta\check{x}_k &\equiv \check{x}_k - x_k \\ &= f_k(W_k \check{x}_{k-1} + b_k + \gamma W_{k+1}^T \check{x}_{k+1}) - f_k(W_k x_{k-1} + b_k) \\ &\approx D_k W_k \delta\check{x}_{k-1} + \gamma D_k W_{k+1}^T \check{x}_{k+1}, \end{aligned}$$

for all hidden layers, and $\delta\check{x}_L \approx D_L W_L \delta\check{x}_{L-1}$ for the output layer. Since the zeroth layer is fixed with the input, $\delta\check{x}_0 = 0$, under the above iterative relationships, we

must have $\delta\check{x}_k \sim \gamma$ for all $k = 1, ..., L$.

To prove equation 6.6, we compare the fixed point equations of the clamped and free states,

$$\begin{aligned} f^{-1}(\check{x}_k) &= W_k\check{x}_{k-1} + b_k + \gamma W_{k+1}^T\check{x}_{k+1} \\ f^{-1}(\hat{x}_k) &= W_k\hat{x}_{k-1} + b_k + \gamma W_{k+1}^T\hat{x}_{k+1}, \end{aligned}$$

for $k = 1, ..., L-1$. Subtract them and perform Taylor expansion around $\check{x}_k$. Recall the definition of $\delta x_k \equiv \check{x}_k - \hat{x}_k$. We have

$$\begin{aligned} W_k\delta x_{k-1} + \gamma W_{k+1}^T\delta x_{k+1} &= f^{-1}(\hat{x}_k) - f^{-1}(\check{x}_k) \\ &\approx J_k\delta x_k, \end{aligned} \tag{6.9}$$

where the matrix $J_k \equiv \text{diag}\{\partial f^{-1}(\check{x}_k)/\partial\check{x}_k\}$. Since $\check{x}_k - x_k \sim \gamma$, to the leading order in γ, matrix J_k can be approximated by $J_k \approx \text{diag}\{\partial f^{-1}(x_k)/\partial x_k\} = D_k^{-1}$. Substitute this back to equation 6.9. We get

$$\delta x_k \approx D_k(W_k\delta x_{k-1} + \gamma W_{k+1}^T\delta x_{k+1}).$$

Assume that δx_k is of order γ^{L-k}. Then $W_k\delta x_{k-1}$ is of higher order in γ than $\gamma W_{k+1}^T\delta x_{k+1}$. Therefore,

$$\delta x_k \approx \gamma D_k W_{k+1}^T\delta x_{k+1},$$

with $\delta x_L = d - \check{x}_L \approx d - x_L$. This equation indicates that $\delta x^k \sim \gamma\delta x^{k+1}$, which implies that $\delta x_k \sim \gamma^{L-k}$. Therefore, it completes our assumption, and equation 6.6 is proved.

If the output neurons are linear, then $y_L \approx \delta x_L$. Consequently, $\delta x_k \approx \gamma^{L-k}y_k$ for all $k = 1, ..., L$.

Finally, the weight update rule of CHL follows

$$\begin{aligned}
\Delta W_k &= \eta\gamma^{k-L}(\hat{x}_k\hat{x}_{k-1}^T - \check{x}_k\check{x}_{k-1}^T) \\
&= \eta\gamma^{k-L}\delta x_k\check{x}_{k-1}^T + \eta\gamma^{k-L}\check{x}_k\delta x_{k-1}^T + \eta\gamma^{k-L}\delta x_k\delta x_{k-1}^T \\
&\approx \eta\gamma^{k-L}\delta x_k\check{x}_{k-1}^T \\
&\approx \eta y_k x_{k-1}^T.
\end{aligned}$$

The last approximation is made because $\check{x}_{k-1} - x_{k-1} \sim \gamma$. This result shows that the CHL algorithm in the layered network with linear output neurons is identical to the backpropagation as $\gamma \to 0$.

6.4 Contrastive Function

The CHL algorithm stated in section 6.2.2 can be shown to perform gradient descent on a contrastive function that is defined as the difference of the network's Lyapunov functions between clamped and free states [20, 93].

Suppose $E(x)$ is a Lyapunov function of the dynamics in equation 6.3. Construct the contrastive function $C(W) \equiv E(\hat{x}) - E(\check{x})$, where $\hat{x}$ and $\check{x}$ are steady states of the whole network in the clamped and free phase respectively, and $W \equiv \{W_1, ..., W_L\}$. For simplicity, let us first assume that $E(x)$ has a unique global minimum in the range of x and no local minima. According to the definition of Lyapunov functions, $\check{x}$ is the global minimum of E, and so is $\hat{x}$, but under the extra constraints that the output neurons are clamped at d. Therefore, $C(W) = E(\hat{x}) - E(\check{x}) \geq 0$ and achieves zero if and only if $\check{x} = \hat{x}$, that is, when the output neurons reach the desired values. Performing gradient descent on $C(W)$ leads to the CHL algorithm. On the other hand, if $E(x)$ does not have a unique minimum, $\check{x}$ and $\hat{x}$ may only be local minima. However, the above discussion still holds, provided that $\hat{x}$ is in the basin of attraction of $\check{x}$ under the free phase dynamics. This imposes some constraints on how to reset the initial state of the network after each phase. One strategy is to let the clamped phase settle to the steady state first, and then run the free phase without resetting

hidden neurons. This will guarantee that $C(W)$ is always nonnegative and constitutes a proper cost function.

Next we introduce a Lyapunov function for the network dynamics in equation 6.3,

$$E(x) = \sum_{k=1}^{L} \gamma^{k-L}[1^T \bar{F}_k(x_k) - x_k^T W_k x_{k-1} - b_k^T x_k], \tag{6.10}$$

where function $\bar{F}_k$ is defined so that $\bar{F}_k'(x) = f_k^{-1}(x)$, which is the inverse of f_k. $x \equiv \{x_1, ..., x_L\}$ represents the states of all layers of the network.

For $E(x)$ to be a Lyapunov function, it must be nonincreasing under the dynamics equation 6.3. This can be shown by

$$\begin{aligned}
\dot{E} &= \sum_{k=1}^{L} \left(\frac{\partial E}{x_k}\right)^T \dot{x}_k \\
&= \sum_{k=1}^{L} \gamma^{k-L}[f_k^{-1}(x_k) - W_k x_{k-1} - \gamma W_{k+1}^T x_{k+1} - b_k]^T \dot{x}_k \\
&= \sum_{k=1}^{L} -\gamma^{k-L}[f_k^{-1}(x_k) - f_k^{-1}(\dot{x}_k + x_k)]^T [x_k - (\dot{x}_k + x_k)] \\
&\leq 0,
\end{aligned}$$

where the last inequality holds because f_k is monotonic as we have assumed. Therefore, $E(x)$ is nonincreasing following the dynamics, and stationary if and only if at the fixed points. Furthermore, with appropriately chosen f_k, such as sigmoid functions, $E(x)$ is also bounded below, in which case $E(x)$ is a Lyapunov function.

Given the Lyapunov function, we can form the contrastive function $C(W)$ and derive the gradient descent algorithm on C accordingly.

The derivative of $E(\hat{x})$ with respect to W_k is

$$\begin{aligned}
\frac{dE(\hat{x})}{dW_k} &= \frac{\partial E}{\partial W_k} + \sum_k \frac{\partial E}{\partial \hat{x}_k} \frac{\partial \hat{x}_k}{\partial W_k} \\
&= \frac{\partial E}{\partial W_k} \\
&= -\gamma^{k-L} \hat{x}_k \hat{x}_{k-1}^T,
\end{aligned}$$

where the second equality holds because $\partial E/\partial \hat{x}_k = 0$ for all k at the steady states. Similarly, we derive

$$\frac{dE(\check{x})}{dW_k} = -\gamma^{k-L}\check{x}_k\check{x}_{k-1}^T.$$

Combining the above two equations, we find the derivative of $C(W)$ with respect to W_k shall read

$$\frac{dC}{dW_k} = \frac{dE(\hat{x})}{dW_k} - \frac{dE(\check{x})}{dW_k} = -\gamma^{k-L}(\hat{x}_k\hat{x}_{k-1}^T - \check{x}_k\check{x}_{k-1}^T).$$

With a suitable learning rate, gradient descent on $C(W)$ leads to the CHL algorithm in equation 6.4.

6.5 Equivalence of cost functions

In section 6.3, we proved that the CHL algorithm in the layered network with linear output neurons is equivalent to backpropagation in the weak feedback limit. Since both algorithms perform gradient descent on some cost function, the equivalence in the update rule implies that their cost functions should be equal, up to a multiplicative or an additive constant difference. Next, we demonstrate this directly by comparing the cost functions of these two algorithms.

The backpropagation learning algorithm is gradient descent on the squared difference, $\|d - x_L\|^2/2$, between the desired and actual outputs of the network.

For the CHL algorithm, the cost function is the difference of Lyapunov functions between the clamped and free states, as shown in the previous section. After reordering, it can be written as

$$C = \sum_{k=1}^{L} \gamma^{k-L}[1^T(\bar{F}_k(\hat{x}_k) - \bar{F}_k(\check{x}_k)) - \delta x_k^T(W_k\check{x}_{k-1} + b_k) - \delta x_{k-1}^T W_k^T\hat{x}_k].$$

Recall that $\delta x_k \sim \gamma^{L-k}$. Therefore, the δx_k term above multiplied by the factor γ^{k-L} is of order 1, whereas the δx_{k-1} multiplied by the same factor is of order γ, and

thus can be neglected in the leading order approximation. After this, we get

$$C \approx \sum_{k=1}^{L} \gamma^{k-L}[1^T(\bar{F}_k(\hat{x}_k) - \bar{F}_k(\check{x}_k)) - \delta x_k^T(W_k\check{x}_{k-1} + b_k)].$$

If the output neurons are linear ($f_L(x) = x$), then $\bar{F}_L(x) = x_L^T x_L/2$ and $W_L\check{x}_{L-1} + b_L = \check{x}_L$. Substituting them into C and separating terms of the output and hidden layers, we derive

$$\begin{aligned} C &\approx \frac{1}{2}[(\hat{x}_L^T\hat{x}_L - \check{x}_L^T\check{x}_L) - \delta x_L^T\check{x}_L] + \sum_{k=1}^{L-1} \gamma^{k-L}\delta x_k^T[f_k^{-1}(\check{x}_k) - W_k\check{x}_{k-1} - b_k] \\ &\approx \frac{1}{2}\|d - x_L\|^2, \end{aligned}$$

where the second term with the sum vanishes because of the fixed point equations.

In conclusion, to the leading order in γ, the contrastive function in CHL is equal to the squared error cost function of backpropagation. The demonstration on the equality of the cost functions provides another perspective on the equivalence between these two forms of learning algorithms.

So far, we have always assumed that the output neurons are linear. If this is not true, how different is the cost function of CHL from that of backpropagation? Repeating the above derivation, we get the cost function of CHL for nonlinear output neurons,

$$\begin{aligned} C &\approx 1^T\bar{F}_L(\hat{x}_L) - 1^T\bar{F}_L(\check{x}_L) - \delta x_L^T f_L^{-1}(\check{x}_L) \\ &\approx \frac{1}{2}\delta x_L^T D_L^{-1}\delta x_L \end{aligned} \tag{6.11}$$

Since D_L^{-1} is a positive definite diagonal matrix, the cost function of CHL for nonlinear output neurons is the weighted sum of the square errors for each output neuron.

6.6 Generalization

In the preceding sections, we studied CHL in a layered recurrent network with feedback connections, and showed that in the limit of weak feedbacks, CHL becomes identical to backpropagation. Next, we consider of applying CHL to a class of more general networks, which subsume fully connected symmetric networks and layer recurrent networks considered above.

The generalized network dynamics is defined to be

$$\dot{x}_i + x_i = f_i(\sum_{j=1}^{n} \alpha_i W_{ij} \beta_j x_j + b_i), \tag{6.12}$$

where $W_{ij} = W_{ji}$ and $\alpha_i, \beta_i > 0$ for all $i, j = 1, ..., n$ with n being the total number of neurons. This network is a generalization of the traditional symmetric networks, with the parameters α_i and β_j introduced to form potentially asymmetric synaptic interactions. It subsumes both the symmetric networks and the layer networks. For example, in the special case of $\alpha_i = 1$ and $\beta_i = 1$ for all i, the network becomes a symmetric network. On the other hand, the layer networks can be instantiated by arranging the network into L layers and choosing α_i in the kth layer to be $\gamma^{(1-k)/2}$, and β_i in the kth layer to be $\gamma^{k/2}$.

We show in the Appendix that the dynamics in Eq. (6.12) is guaranteed to converge to a fixed point under some very general conditions, in which case the network has a Lyapunov function

$$E(x) = \sum_{i=1}^{n} \beta_i \alpha_i^{-1} (\bar{F}_i(x_i) - b_i x_i) - \frac{1}{2} \sum_{i.j=1}^{n} \beta_i x_i W_{ij} x_j \beta_j. \tag{6.13}$$

6.6.1 The learning algorithm in the generalized network

CHL algorithm in the generalized network follows similar steps as in layered networks: first, clamp the output neurons with the desired value d and let the network converge to the steady state, denoted as $\hat{x}_i$ for the ith neuron; second, in the free phase, let the network converge without clamping output neurons, with the steady state denoted

by $\check{x}_i$. The CHL algorithm updates the synaptic weight W_{ij} with

$$\Delta W_{ij} = \eta \beta_i \beta_j (\hat{x}_i \hat{x}_j - \check{x}_i \check{x}_j). \tag{6.14}$$

This generalized CHL algorithm can be shown to perform gradient descent on a contrastive function defined to be the difference of Lyapunov function between the clamped and free states. We detail this in Appendix.

When $\delta x \equiv \hat{x} - \check{x}$ is small, the CHL algorithm can be approximated, in the leading order, by

$$\Delta W_{ij} \approx \eta \beta_i \beta_j (\delta x_i \check{x}_j + \check{x}_i \delta x_j). \tag{6.15}$$

Backpropagation algorithm can also be formulated under the network structure in Eq. (6.12), in which case it is often called recurrent backpropagation since it is implemented in a network with dynamics. The recurrent backpropagation algorithm performs gradient descent on the cost function defined to be the square error between the desired and the actual steady state output. The derivation of the algorithm is shown in Appendix by using the Lagrange multiplier method. It takes the following form

$$\Delta W_{ij} = -\eta \partial L / \partial W_{ij} = \eta \beta_i \beta_j (\tilde{u}_i x_j + \tilde{u}_j x_i), \tag{6.16}$$

where $\tilde{u}$ is the error signal, spread back from the error in the output neurons. The exact form of $\tilde{u}$ is shown in Appendix.

This update rule for recurrent backpropagation is in the same form as the CHL in Eq. (6.15), except that $\tilde{u}$ is the propagated error signal, whereas δx in CHL is the difference between the clamped and free steady state.

The close connections between backpropagation and CHL algorithm originate from the similar forms of update rules on one hand, and the relationship between the error signal and δx on the other hand. In layered networks with weak feedbacks, the error signal becomes the same as the δx, leading to the equivalence between these two forms of learning algorithms. In the generalized network, this is generally not true any more. However, δx and $\tilde{u}$ are within the ninety degrees (proved in Appendix),

that is, $\delta x^T \tilde{u} \geq 0$.

This does not necessarily imply that the update rules for recurrent backpropagation and the generalized CHL are within ninety degrees. A counter example is given in the next section, which shows that in some update steps CHL may actually increase the square error, minimized in every step of backpropagation.

6.6.2 CHL step does not always decrease the square error: an example

In the layered network with weak feedbacks, we demonstrate that the CHL algorithm decreases the square error cost function in every update step. However, for the generalized network, this is not always true. We give a counter example here in the context of symmetric networks, that is, taking $\alpha_i = 1$ and $\beta_i = 1$ for all i in Eq. (6.12).

The fixed point of the symmetric network of Eq. (6.12) is $f^{-1}(x) = Wx + b$. Let's consider one particular learning step, in which the weight is changed by δW, the resulting fixed point changes, denoted by δx, shall satisfy

$$D\delta x = W\delta x + \delta W x, \tag{6.17}$$

where the diagonal matrix $D \equiv \text{diag}\{df^{-1}(x)/dx\}$. From the above equation, we derive $\delta x = (D - W)^{-1}\delta W x$.

We further assume that the network has no hidden neurons. Then, after one CHL step, following the CHL update rule, the change in weight reads

$$\delta W = \eta(dd^T - xx^T) = \eta(\epsilon x^T + x\epsilon^T + \epsilon\epsilon^T),$$

where d is the desired output and $\epsilon = d - x$. From this, we derive the change in fixed point δx becomes $\delta x = \eta(D - W)^{-1}[(x^T x)\epsilon + (\epsilon^T x)(x + \epsilon)]$.

To test if δx goes along the direction of decreasing the square error function, we take the inner product between ϵ and δx, and check its sign. For simplicity, next

we consider the case when the function $f(z) = z$, and take an example W to be $W = [1 - 1/k, 0; 0, 0]$. In this case, matrix $(I - W)^{-1} = [k, 0; 0, 1]$, and the inner product between ϵ and δx becomes

$$\epsilon^T \delta x = \|x\|^2(k\epsilon_1^2 + \epsilon_2^2) + \epsilon^T x(k\epsilon_1 x_1 + k\epsilon_1^2 + x_2\epsilon_2 + \epsilon_2^2). \tag{6.18}$$

We find that when k is large, that is, when matrix $(D-W)^{-1}$ is strongly anisotropic, there exist a ϵ and a x such that $\epsilon^T x < 0$. In other words, in this case, the CHL update direction increases square error function. An example is $k = 100$, $\epsilon = [-0.174, -0.985]$, and $x = [0.655, -0.756]$, in which case, $\epsilon^T x = -0.209$.

In summary, this example demonstrates that the square error cost function used in backpropagation is not equivalent to the cost function minimized by the generalized CHL. In particular, some steps in the generalized CHL may actually increase the square error. However, despite this, we expect that in the long trend the accumulative effect of the CHL steps shall decrease the square error.

6.6.3 Cost function for the generalized CHL

Since both backpropagation and CHL are gradient descent algorithms on some cost functions, the relationship between them can be examined by comparing their cost functions. Next, we derive the cost function for the CHL algorithm.

Start from the contrastive function, defined as the difference of the Lyapunov function between the clamped and free states as follows

$$\begin{aligned} C &= E(\hat{x}) - E(\check{x}) \\ &= \sum_i \beta_i \alpha_i^{-1} [\bar{F}_i(\hat{x}_i) - \bar{F}_i(\check{x}_i) - b_i(\hat{x}_i - \check{x}_i)] - \frac{1}{2} \sum_{ij} \beta_i \beta_j (\hat{x}_i W_{ij} \hat{x}_j - \check{x}_i W_{ij} \check{x}_j). \end{aligned}$$

Often the performance of an iterative learning algorithm critically depends on its performance close to the optimal solution [99]. Next, we will examine the form of the contrastive function when the difference between the clamped and free states, $\delta x \equiv \hat{x} - \check{x}$, is small, in which case we can carry out the Taylor expansion of the

contrast function C with δx being the small variable. Up to the second order, C can be approximated by

$$C \approx \frac{1}{2}\delta x^T(\tilde{D} - \tilde{K})\delta x, \tag{6.19}$$

where matrix $\tilde{D} \equiv \Lambda D$ with diagonal matrices $D \equiv \text{diag}\{df^{-1}(\check{x})/dx\}$, and $\Lambda \equiv \text{diag}\{\beta_i/\alpha_i\}$. Matrix $\tilde{K}$ is defined to be $\tilde{K} = \Lambda K$ with matrix $K_{ij} \equiv \alpha_i\beta_j W_{ij}$. Matrix $\tilde{K}$ is symmetric, because $\tilde{K}_{ij} = \beta_i\beta_j W_{ij} = \tilde{K}_{ji}$.

The approximated contrastive function takes the quadratic form, involving both output and hidden neurons. This is different from standard cost functions, which is usually a function of output neurons. However, in the CHL algorithm, δx of hidden neurons depend on those of output neurons. Next, we give the relationship between them.

For simplicity of notation, let's denote output neurons with subscript 1 and hidden neurons with 2. Thus $\check{x}_1$ represents the steady states of output neurons and $\check{x}_2$ those of hidden neurons in the free phase; δx_1 represents the difference between clamped and free steady state of output neurons and δx_2 those of hidden neurons. Define diagonal matrices $D_1 \equiv \text{diag}\{df^{-1}(\check{x}_1)/d\check{x}_1\}$, $D_2 \equiv \text{diag}\{df^{-1}(\check{x}_2)/d\check{x}_2\}$. Similarly, matrix Λ_1 is defined to be $\Lambda_1 \equiv \text{diag}\{\alpha_i/\beta_i\}$ for all output neuron i, and Λ_2 for hidden neurons. Four submatrices of W are listed as W_{11} being connections between output neurons, W_{12} connections between output and hidden neurons, and so on for W_{21} and W_{22}. Similarly, submatrices of K are defined.

The relationship between δx_2 and δx_1 can be found by examining the fixed point equations of hidden neurons in both the clamped and the free phase

$$\begin{aligned} f^{-1}(\hat{x}_2) &= K_{21}\hat{x}_1 + K_{22}\hat{x}_2 + b_2 \\ f^{-1}(\check{x}_2) &= K_{21}\check{x}_1 + K_{22}\check{x}_2 + b_2. \end{aligned}$$

Subtract and perform Taylor expansions around the free phase fixed point equations. We derive

$$(D_2 - K_{22})\delta x_2 \approx K_{21}\delta x_1. \tag{6.20}$$

Multiple both side by matrix Λ_2. We find the relationship $\delta x_2 = (\tilde{D}_2 - \tilde{K}_{22})^{-1}\tilde{K}_{21}\delta x_1$, where $(\tilde{D}_2 - \tilde{K}_{22})^{-1}$ exists because $\tilde{D}_2 - \tilde{K}_{22}$ is a positive definite matrix, as shown in Appendix. Substitute this back into the contrastive function C. We derive

$$C \approx \frac{1}{2}\delta x_1^T Q \delta x_1, \tag{6.21}$$

where matrix $Q \equiv \tilde{D}_1 - \tilde{K}_{11} - \tilde{K}_{12}(\tilde{D}_2 - \tilde{K}_{22})^{-1}\tilde{K}_{21}$.

We show in Appendix that Q is positive definite. Therefore, Eq. (6.21) is a well defined cost function in a generalized quadratic form. If we check back for the layered recurrent network in the $\gamma \to 0$ limit, the matrices $W_{21} \to 0$, $W_{11} = 0$ and matrix $Q = I$ if output neurons are linear. In this case, the error function Eq. (6.21) becomes the square error function the same as in backpropagation, which is consistent with the previous results.

If the matrix Q is not isotropic, the decrease in $\delta x_1^T Q \delta x_1$ does not necessarily mean the decrease in $\delta x_1^T \delta x_1$ as well. Therefore, it is possible that some CHL steps may actually increase the square error cost function, as indicated in our counter example. However, despite the fluctuations in the individual steps, the accumulative effect of the CHL steps will decrease the square error cost function.

6.7 Discussion

We have shown that backpropagation in multilayer perceptrons can be equivalently implemented by the CHL algorithm if weak feedback is added. This is demonstrated from two different perspectives: evaluating the two algorithms directly, and comparing their cost functions. The essence behind this equivalence is that CHL effectively extracts the error signal of backpropagation from the difference between the clamped and free steady states.

We further generalize the CHL in a class of networks, subsuming both the layered networks and the symmetric networks. In this case, the change in steady state caused by clamping output neurons is not the same as the error signal spread by recurrent backpropagation, but they are within ninety degrees. When the network is close to

the optimal solution, we derive the cost function for the CHL, which is in the form of a generalized quadratic function, compared to the square error function as in recurrent backpropagation.

The investigation on the relationship between backpropagation and CHL is motivated by the researches looking for biologically plausible learning algorithms. It is believed by many that backpropagation is not biologically realistic. However, in an interesting study done by Zipser and Andersen on coordinate transform in posterior parietal cortex of monkeys, they show that hidden neurons in a network model trained by backpropagation share very similar properties to real neurons recorded from that area [100]. This work has prompted the search for a learning algorithm, which has similar functionality as backpropagation [101, 102], and at the same time is biologically plausible. CHL is a Hebbian-type learning algorithm, relying only on pre- and post-synaptic activities. The implementation of backpropagation equivalently by CHL suggests that CHL could be a candidate solution to this problem.

Mazzoni et al. also proposed a biologically plausible learning rule as an alternative to backpropagation [102]. Their algorithm is a reinforcement type learning algorithm, which is usually slow, has large variance, and depends on global signals. In contrast, the CHL algorithm is a deterministic algorithm, which could be fast. However, a disadvantage of CHL is its dependence on special network structures, such as the layered network in our case. Whether either algorithm is used by biological systems is an important question, which needs further investigations in both experiments and theory.

Appendix

6.7.1 Lyapunov function and CHL in the generalized network

Under some very general conditions, the network Eq. (6.12) always converges to a fixed point. We demonstrate this by using Lyapunov theory. First, let's prove the following result:

For the network dynamics defined in Eq. (6.12), the following function

$$E(x) = \sum_{i=1}^{n} \beta_i \alpha_i^{-1} (\bar{F}_i(x_i) - b_i x_i) - \frac{1}{2} \sum_{i,j=1}^{n} \beta_i x_i W_{ij} x_j \beta_j \tag{6.22}$$

is nonincreasing and stationary if and only if at the fixed points.

Proof: Taking derivative of E with respect to x and substituting the network dynamics lead to

$$\begin{aligned} \partial E/\partial x_i &= \beta_i/\alpha_i (f_i^{-1}(x_i) - b_i) - \sum_j W_{ij} \beta_i \beta_j x_j \\ &= \beta_i/\alpha_i \, [f_i^{-1}(x_i) - \sum_j K_{ij} x_j - b_i] \\ &= \beta_i/\alpha_i \, [f_i^{-1}(x_i) - f_i^{-1}(\dot{x}_i + x_i)], \end{aligned}$$

where $K_{ij} \equiv \alpha_i \beta_j W_{ij}$ is the effective interaction matrix.

Therefore, the time derivative of E is

$$dE/dt = \sum_i \partial E/\partial x_i \; \dot{x}_i = - \sum_i \beta_i/\alpha_i \, [f_i^{-1}(x_i) - f_i^{-1}(\dot{x}_i + x_i)] \; \dot{x}_i \leq 0.$$

The final inequality holds because function $f_i(\cdot)$ is monotonic. Therefore, function E is nonincreasing following the network dynamics. Moreover, it is stationary if and only if at the fixed point. □

With an appropriately chosen f_i, such as the sigmoid function, E will be lower bounded and radially unbounded, in which case it is a Lyapunov function, and the network will always converge to a fixed point.

Define the contrastive function $C(W) \equiv E(\hat{x}) - E(\check{x})$, which is nonnegative and equal to zero when optimal solution is achieved. Taking derivative of $C(W)$ with respect to W, we have the CHL as follows

$$\Delta W_{ij} \propto \beta_i \beta_j (\hat{x}_i \hat{x}_j - \check{x}_i \check{x}_j). \tag{6.23}$$

The layered recurrent network can be instantiated with the connections from the

layer $k-1$ to layer k being expressed as $c_{k+1}d_kW_k$, and the connections from the layer k to $k-1$ bing $c_kd_{k+1}W_k^T$ with $c_k = \gamma^{(1-k)/2}$ and $d_k = \gamma^{k/2}$.

For symmetric networks, take $\alpha_i, \beta_i = 1$ for all $i = 1, ..., n$. Consequently, the CHL algorithm is simply $\Delta W \propto \hat{x}\hat{x}^T - \check{x}\check{x}^T$.

6.7.2 Matrix Q is positive definite

Proof: Recall that matrices $\tilde{D} = \Lambda D$ and $\tilde{K} = \Lambda K$. Matrix $\tilde{D} - \tilde{K}$ is the Hessian of Lyapunov function Eq. (6.22) at the steady state of free step, thus it is positive definite. For any vector $x = [x_1, x_2]^T$, we must have

$$x_1^T(\tilde{D}_1 - \tilde{K}_{11})x_1 + x_2^T(\tilde{D}_2 - \tilde{K}_{22})x_2 - 2x_1^T\tilde{K}_{12}x_2 \geq 0$$

Take $x_2 = (\tilde{D}_2 - \tilde{K}_{22})^{-1}\tilde{K}_{21}x_1$ and substitute back. We find

$$x_1^T[(\tilde{D}_1 - \tilde{K}_{11}) - \tilde{K}_{12}(\tilde{D}_2 - \tilde{K}_{22})^{-1}\tilde{K}_{21}]x_1 \geq 0$$

which holds for any vector x_1. Therefore matrix Q is positive definite. □

6.7.3 Backpropagation algorithm in the generalized network

The recurrent backpropagation algorithm [103] for the generalized network can be derived using the Lagrangian multiplier method. First, we construct a Lagrangian function as follows

$$L = \frac{1}{2}\|d - x_1\|^2 + u^T(f^{-1}(x) - Kx - b),$$

where vector u is the Lagrange multiplier. The update for weight W can be found by taking derivative of L with respect to W,

$$\Delta W_{ij} = -\eta\partial L/\partial W_{ij} = \eta\beta_i\beta_j(\tilde{u}_i x_j + \tilde{u}_j x_i),$$

where $\tilde{u} = \Lambda u$. This update rule is in the same form as in contrastive Hebbian learning, except that $\tilde{u}$ is the propagating error signal, whereas δx in contrastive Hebbian learning is the difference between the clamped and free steady state.

$\tilde{u}$ can be found by taking derivative of L with respect to x as follows,

$$\begin{aligned} -Q\delta x_1 + (\tilde{D}_1 - \tilde{K}_{11})^T \tilde{u}_1 - \tilde{K}_{12}\tilde{u}_2 &= 0 \\ -\tilde{K}_{21}\tilde{u}_1 + (\tilde{D}_2 - \tilde{K}_{22})\tilde{u}_2 &= 0. \end{aligned}$$

From the above equation, we derive $\tilde{u}$ to be

$$\begin{aligned} \tilde{u}_1 &= Q^{-1}\delta x_1 \\ \tilde{u}_2 &= M\tilde{u}_1, \end{aligned}$$

where matrix $M \equiv (\tilde{D}_2 - \tilde{K}_{22})^{-1}\tilde{K}_{21}$.

Because matrix Q is positive definite, therefore we have

$$\begin{aligned} \delta x_1^T \tilde{u}_1 &= \tilde{u}_1^T Q \tilde{u}_1 \geq 0 \\ \delta x_2^T \tilde{u}_2 &= \tilde{u}_1^T M^T M Q \tilde{u}_1 \geq 0 \end{aligned}$$

Therefore, although the difference between the clamped and free steady states is not the same as backpropagation error signals, they are within ninety degrees.

Chapter 7

A Synaptic Learning Rule in Networks of Spiking Neurons

7.1 Introduction

Neurons in vivo typically fire highly irregular spike trains. There have been studies regarding how spikes trains are generated randomly [104]. However, why neurons fire irregular spike trains and what is the benefit the brain derives from it still remain unclear. In this chapter, we suggest that the irregular spiking could be used as a mechanism for learning.

The interspike interval distribution of cortical neurons is roughly exponential and the coefficient of variation of the interspike interval distribution is close to 1 [105]. This suggests that the spike train of cortical neurons is roughly Poisson. In this chapter, we derive an algorithm for a network of spiking neurons that fire Poisson spike trains, based on the REINFORCE learning idea [25, 26]. The derived learning algorithm takes a form that is based on the correlation between fluctuation of postsynaptic firing and presynaptic EPSP, modulated by a global reward signal.

To test whether this algorithm could be applied to real neurons whose's spike trains may not be exactly Poisson, we simulate a network of integrate-and-fire neurons with

[0]This is a collaborative work with H. S. Seung.

white noise injected, and apply the learning rule to learn XOR computation.

7.2 Poisson Neurons

7.2.1 Basic definition

In this section, we derive a learning rule for a network of neurons that fire Poisson spike trains. Let the state of a neuron denoted by a binary variable $\sigma_i(t)$ for the ith neuron at time t. $\sigma_i(t) = 1$ denotes spiking of the neuron, and $\sigma_i(t) = 0$ denotes nonspiking. For a network of n neurons we will also use vector $\sigma(t) = [\sigma_1(t), \sigma_2(t), ..., \sigma_n(t)]$ to denote the state of the network at time t. Suppose spiking of the neurons is an inhomogeneous Poisson process. Let the instantaneous rate of the ith neuron be $\lambda_i(t)$, which is determined by

$$\lambda_i(t) = f_i\left(\sum_{j=1}^{n} W_{ij} h_j(t)\right), \tag{7.1}$$

where W_{ij} is the synaptic weight from neuron j to neuron i, and $f_i(\cdot)$ is the transfer function determining the firing rate of neuron i based on total presynaptic inputs. $h_j(t)$ is the activation variable that represents the receptor dynamics due to the spiking of presynaptic neuron j and is modeled by

$$\tau_s \dot{h}_j(t) + h_j(t) = \sum_a \delta(t - T_j^a) \tag{7.2}$$

where T_i^a is the ath spiking time of neuron j. This equation models that the activation variable experiences an instantaneous jump when a spike comes and slowly decays with a time constant τ_s when there are no spikes.

7.2.2 Episodic learning

We first consider an episodic version of the algorithm. Suppose the network is run between time 0 and T. At the end of each episode, the overall performance of the

network is evaluated by a reward function R that depends on the state $\sigma(t)$ from $t = 0$ to T. The objective of the learning is to update W_{ij} such that the expected reward R is increased.

The spike generating process is a continuous process. To facilitate derivation of the algorithm, we discretize the time with sufficiently small interval Δt. Therefore, the overall state of network between time 0 and T can be denoted by a vector $[\sigma(0), \sigma(\Delta t), ..., \sigma(T)]$. The expected reward is then

$$\langle R \rangle = \sum_{\sigma(0),\sigma(\Delta t),...,\sigma(T)} P(\sigma(0), \sigma(\Delta t), ..., \sigma(T)) R(\sigma(0), \sigma(\Delta t), ..., \sigma(T))$$

where $P(\sigma(0), \sigma(\Delta t), ..., \sigma(T))$ is the probability for the state $[\sigma(0), \sigma(\Delta t), ..., \sigma(T)]$, and the summation is over all possible states.

Next we derive a learning algorithm that update synaptic weights by performing gradient descent on the expected reward. We first compute the gradient of $\langle R \rangle$ with respect to weight W_{ij}, which can be written as

$$\begin{aligned} \frac{\partial \langle R \rangle}{W_{ij}} &= \sum_{\sigma(0),\sigma(\Delta t),...,\sigma(T)} \frac{\partial}{\partial W_{ij}} P(\sigma(0), \sigma(\Delta t), ..., \sigma(T)) R(\sigma(0), \sigma(\Delta t), ..., \sigma(T)) \\ &= \langle e_{ij} R(\sigma(0), \sigma(\Delta t), ..., \sigma(T)) \rangle \end{aligned} \tag{7.3}$$

where $e_{ij} = \partial \ln P(\sigma(0), \sigma(\Delta t), ..., \sigma(T))/\partial W_{ij}$ is the eligibility trace, which records the states of neurons and will be used in updating synaptic weights.

At the end of each episode, the reward function is evaluated and synaptic weights are updated according to the rule:

$$\Delta W_{ij} = \eta R(\sigma(0), \sigma(\Delta t), ..., \sigma(T)) e_{ij} \tag{7.4}$$

where η is a positive number determining learning rate. From the above derivations, on average this learning rule will perform gradient descent on the expected reward. This form of learning rule, called REINFORCE learning, is introduced by Williams [25], and extended later by others [106].

The activation variable $h_i(t)$ is deterministic, depending only on spikes of neuron i happened before time t. Given the Poisson spike train assumption, the probability $P(\sigma(0), \sigma(\Delta t), ..., \sigma(T))$ can then be expressed as

$$P(\sigma(0), \sigma(\Delta t), ..., \sigma(T)) = \prod_{t=0}^{T} P(\sigma(t)|\sigma(t-1), ..., \sigma(0))$$

which is the product of the probability at each time step conditioned on previous states. Furthermore, at any same time the spiking of each neuron is conditionally independent of each other. Therefore, the logarithm of the probability $P(\sigma(0), \sigma(\Delta t), ..., \sigma(T))$ can be written as

$$\ln P(\sigma(0), \sigma(\Delta t), ..., \sigma(T)) = \sum_{t=0}^{T} \sum_{i=1}^{n} \ln P(\sigma_i(t)|\sigma(t-1), ..., \sigma(0))$$

From this, we derive the eligibility trace as follows

$$e_{ij} = \sum_{t=0}^{T} \frac{\partial}{\partial W_{ij}} \ln P(\sigma_i(t)|\sigma(t-1), ..., \sigma(0))$$

Given the Poisson spiking neurons we are considering, the probability for neuron i to fire a spike or not during the interval $[t, t+\Delta t)$ is determined by

$$\sigma_i(t) = \begin{cases} 1 & \text{with probability } p_i(t) = \lambda_i(t)\Delta t \\ 0 & \text{with probability } 1 - p_i(t) \end{cases}$$

where $\lambda_i(t)$ is the instantaneous rate given by Eq. (7.1). The above holds when Δt is sufficiently small. Based on the above formula, the eligibility trace can be further

simplified to

$$e_{ij} = \sum_{t=0}^{T} \sigma_i(t) \frac{\partial}{\partial W_{ij}} \ln p_i(t) + (1 - \sigma_i(t)) \frac{\partial}{\partial W_{ij}} \ln(1 - p_i(t)) \tag{7.5}$$

$$= \sum_{t=0}^{T} \left[\frac{\sigma_i(t)}{p_i(t)} - \frac{1 - \sigma(t)}{1 - p_i(t)} \right] \frac{\partial p_i(t)}{\partial W_{ij}} \tag{7.6}$$

$$= \sum_{t=0}^{T} \frac{\sigma_i(t) - \lambda_i(t)\Delta t}{\lambda_i(t)\Delta t(1 - \lambda_i(t)\Delta t)} \frac{\partial \lambda_i(t)}{\partial W_{ij}} \Delta t \tag{7.7}$$

$$= \int_0^T \frac{f_i'(t)}{f_i(t)} [s_i(t) - f_i(t)] h_j(t) dt \tag{7.8}$$

where $s_i(t) = \sum_a \delta(t - T_i^a)$ is a series of delta functions representing spiking of neuron i in continuous time. Eq. (7.8) is derived by taking the limit of Δt to zero.

Substituting the eligibility trace e_{ij} into Eq. (7.4), we derive an episodic version of the learning rule for Poisson spiking neurons:

$$\Delta W_{ij} = \eta R \int_0^T \frac{f_i'(t)}{f_i(t)} [s_i(t) - f_i(t)] h_j(t) dt \tag{7.9}$$

7.2.3 Online learning

The episodic version of the algorithm is nice in that on average it is guaranteed to perform gradient descent on the expected reward function. However, in real biological systems, most learning problems cannot be strictly separated into each episode with a fixed duration. With some heuristic arguments, however, we can extend the episodic version of the algorithm into an online algorithm. Rather than integrating over whole duration to get the eligibility trace as in Eq. (7.8), we integrate over only a short time period over the past as

$$\tau_e \dot{\bar{e}}_{ij} + \bar{e}_{ij} = \frac{f_i'(t)}{f_i(t)} [s_i(t) - f_i(t)] h_j(t) \tag{7.10}$$

where the time constant τ_e determines the integration time period. The online version of the learning algorithm is then formulated as

$$\dot{W}_{ij} = \eta R \bar{e}_{ij}(t). \tag{7.11}$$

The online learning rule can be understood as a Hebbian-type learning rule based on presynaptic and postsynaptic activities, but modulated by a global reward signal. However, our synaptic update rule is based on fluctuation on the postsynaptic activity. Moreover, the update rule is spike-dependent. More specifically, when the reward signal is positive and presynaptic neuron j fires a spike causing a jump in $h_j(t)$, a synapse will be strengthened only if the postsynaptic neuron fires a spike subsequently, in a time window during which $h_j(t)$ has not decayed to zero. Therefore, this learning rule naturally leads to the potentiation part of the spike-time-dependent synaptic plasticity, and predicts that the time window for potentiation to happen is determined by the time constant for receptor dynamics τ_s. Conversely, if the neuron does not fire a spike, the synapse is depressed.

7.2.4 Simulation results

Next, we apply the learning rule Eq. (7.11) in a network of Poisson neurons to learn the representation of XOR. The network consists of two input neurons, ten hidden neurons and one output neuron. The training data are four binary patterns $\{(1,0),(0,1),(1,1),(0,0)\}$ with the desired outputs to be $\{1,1,0,0\}$ respectively.

During training, each pattern is presented to the input for about 500 ms. The reward function is evaluated based on activities of the output neuron. Let the binary variable $\sigma_o(t)$ denote the state of the output neuron. We use the reward $R(t) = 0.5 - |\sigma_o(t) - d(t)|$, where $d(t)$ is the desired output.

The firing rates of the output neuron corresponding to input patterns are shown in Fig. (7-1). Initially, the output neuron fires at high rates only when both inputs are 1. However, by the end of training, the output neuron fires at high rates only when one input neuron receives 1, but not both. The learning curve for XOR learning

is shown in Fig. (7-2).

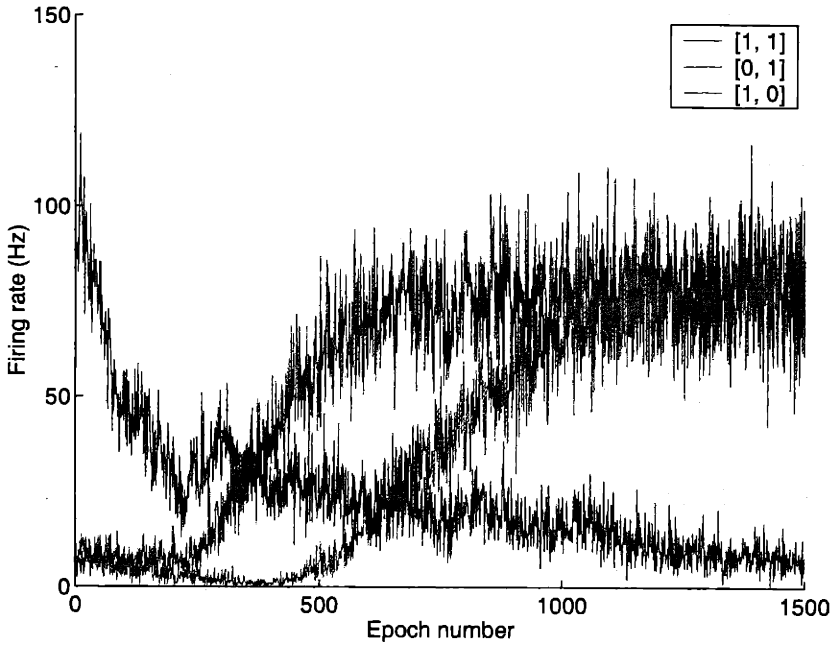

Figure 7-1: Firing rates of the output neuron plotted as a function of epochs during training. Three curves correspond to three input patterns $\{(1,0),(0,1),(1,1)\}$. The pattern $(0,0)$ does not drive hidden neurons to spike, and therefore, firing rate of the output neuron is always near zero and is not shown here.

The network learns to represent XOR computation by balancing excitation and inhibition. In Fig. (7-3) we plot the synaptic weights before and after learning. The synaptic weights are initialized randomly. After learning, each hidden neuron is excited by one input neuron and inhibited by the other one. Therefore, if both input neurons are activated, the hidden neurons become inactive because the total synaptic inputs are small.

7.3 Integrate-and-Fire Neurons

So far, our derivation of the algorithm has been based on the Poisson spiking assumption. For real neurons, the spike train will not exactly like Poisson. Can the learning rule Eq. (7.11) still work for those neurons? We address this issue in this section by

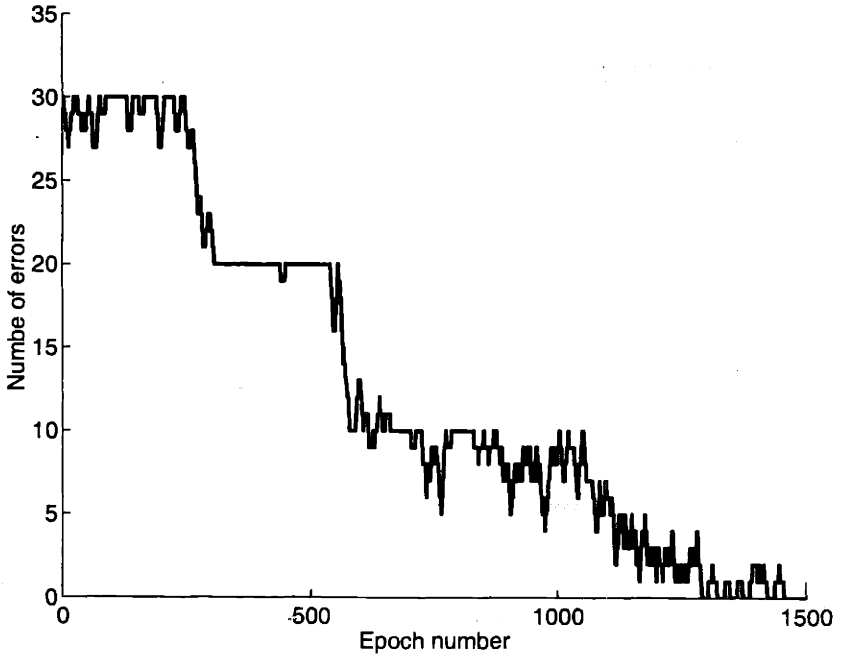

Figure 7-2: Learning curve for XOR learning. The error shown here is the summed errors over 10 repetition of three input patterns.

simulating a network of integrate-and-fire neurons. We inject white noise to those neurons to emulate random inputs neurons receive. The model neuron is described by

$$\tau_m \frac{dV_i}{dt} = -V_i + V_{rest} + I_i(t) + \xi_i(t), \tag{7.12}$$

where V_i is the membrane potential for neuron i, τ_m is the membrane time constant, V_{rest} is the resting potential, and $I_i(t)$ is the total synaptic input. $\xi_i(t)$ is the white noise:

$$\langle \xi_i(t) \rangle = 0 \qquad \langle \xi_i(t)\xi_j(t') \rangle = \sigma^2/\tau_m \delta_{ij}\delta(t-t'), \tag{7.13}$$

for all $i, j = 1, ..., n$. When membrane potential V_i reaches a threshold V_{th}, a spike is generated and V_i is reset to V_r. The parameters we use are $\tau_m = 20$ ms, $V_{th} = -54$ mV, $V_r = -60$ mV, $V_{rest} = -74$ mV, and $\sigma = 5.6$.

The firing rate vs. current relationship can be calculated explicitly when white

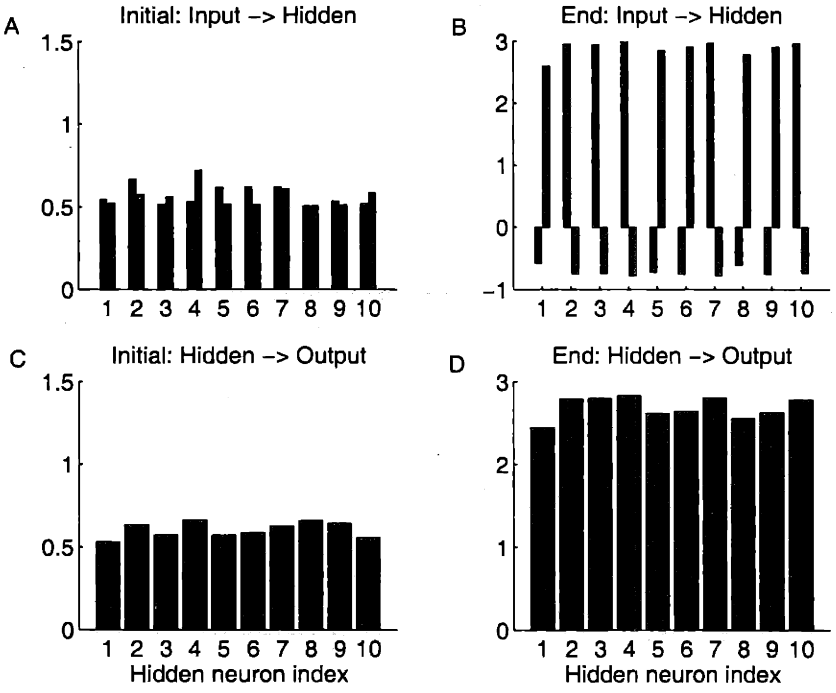

Figure 7-3: Synaptic weights before and after learning. Panel A and B are synaptic connections from two input neurons to 10 hidden neurons. The blue and brown bars represent the connections from two input neurons. In the bottom two panels plotted are the synaptic weights from hidden to output neurons.

noise is injected. The firing rate is described by [107]

$$f_i(I_i) = \left[\tau_m \int_0^\infty e^{-u^2}(e^{2y_{th}u} - e^{2y_r u})/u du\right]^{-1}, \tag{7.14}$$

where $y_{th} = (V_{th} - V_{rest} - I)/\sigma$ and $y_r = (V_r - V_{rest} - I_i)/\sigma$. The synaptic input $I_i(t) = \sum_j W_{ij} h_j(t)$, where $h_j(t)$ is the synaptic activation variable, which is modeled by Eq. (7.2).

We apply learning rule Eq. (7.11) to learn the same XOR problem described in Section (7.2.4). The result is shown in Fig. (7-4), which demonstrates that the learning rule could still be used for learning XOR representation. The results are very similar to those in the preceding section with Poisson spiking neurons.

7.4 Discussion

In this chapter, we propose a synaptic leaning rule for a network of spiking neurons. The learning rule takes advantage of the fact that neurons fire highly irregular spike trains. The fluctuation in the activities of those neurons is essentially a mechanism for exploring different states of neurons and thus beneficial to learning if synaptic update rules are based on this fluctuation and correlated with some global reward signals.

We derive our learning rule by making the assumption that neurons fire Poisson spike trains. After that, we apply the algorithm to a network of integrate-and-fire neurons to test if the derived learning rule could still work when the spike trains of neurons are not exactly Poisson. Our simulation results show that XOR computation could be still be learned in such networks.

The learning rule is consistent with recent experimental findings on spike-time-dependent synaptic plasticity [16, 17] . However, contrasting with the traditional Hebbian learning idea, an essential element of our learning algorithm is the global reward signal. Investigating the existence of such reward signals is an interesting challenge for future researches on synaptic plasticity.

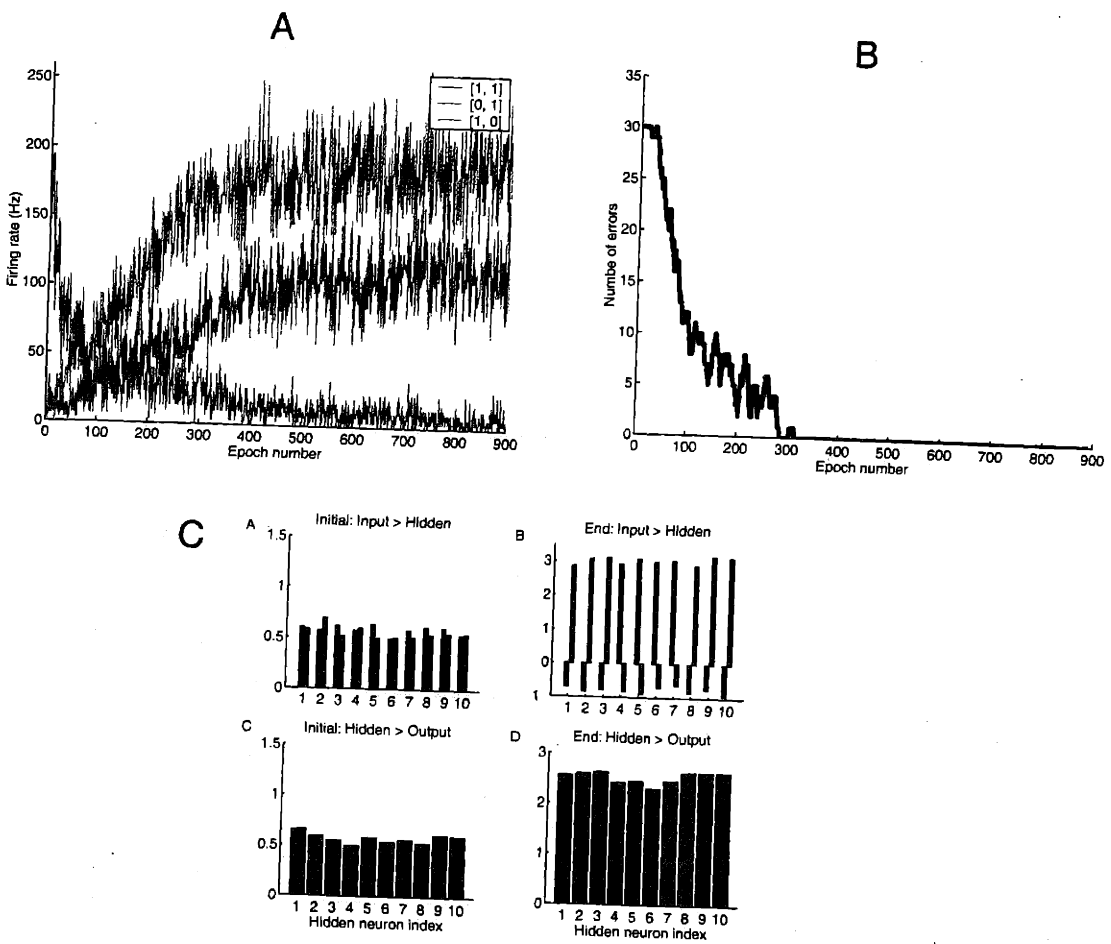

Figure 7-4: Learning XOR in a network of integrate-and-fire neurons. Panel A plots the firing rate of the output neuron over training epochs, with different colors representing different input patterns. Panel B is the learning curve. Panel C is the synaptic weights before and after learning.

Chapter 8

Conclusion

In this thesis, we have studied dynamics and learning in recurrent network models. In Chapter 2, we analyze the dynamics of recurrent network models for direction selectivity. We find that depending on stimulus velocity, the spatiotemporal patterns of neural activities in recurrent networks can change dramatically, bifurcating at some critical velocities from form-stable traveling pulse solutions to form-unstable lurching waves. Since lurching wave solutions cannot arise through feedforward mechanisms, observation of the lurching wave spatiotemporal patterns in experiments could act as a strong indicator for the involvement of recurrent network mechanism for direction selectivity.

In Chapter 3, we introduce a way of wiring synaptic connections to mediate competitions between groups of neurons, which is a generalization of the traditional Winner-Take-All network. From another perspective, the model we propose can be viewed as an associative memory model in a network of analog neurons. In traditional Hopfield-type associative memory model, neurons are essentially binary, which is different from real neurons, which can have a continuous representation of information through firing rates. In our model, neurons can be either active or inactive, and thus form a binary pattern. However, the activities of the active neurons are analog with exact values determined by the computation carried by the network. Our network demonstrates the coexistence of digital selection and analog computation, which has been argued as a mechanism for cortical computation.

In Chapter 4, we propose a double-ring network model for integration in the head-direction system. Two theoretical issues are addressed. First, when the head is still, how does the neural system keep a short-term memory of the current directional heading? Second, when the head starts to move, how does the system integrate the angular head velocity signals to get an updated direction? For the first question, our model keeps the short-term memory through persistent activities of neurons, which is maintained through tuned recurrent feedback. To address the second question, we propose that the integration is carried out through the interaction of two populations of neurons, each receiving a differential input from vestibular nuclei.

The second part of the thesis is regarding learning in biological motivated networks. In Chapter 5, we study the synaptic plasticity that depends critically on the temporal ordering of the pre- and postsynaptic spiking times. What kind of computation could this kind of synaptic plasticity produce? Through mathematical analysis and numerical simulations, we find that under certain conditions, when reduced to a rate-based learning rule, the spike-time dependent plasticity produces a differential form of Hebbian learning rule that depends on the time derivative of the postsynaptic firing rate. We show that a learning rule of this form could act to stabilize persistent neural activity patterns in recurrent neural network.

In Chapter 6, we study the relationship between backpropagation and contrastive-Hebbian learning, two methods used to train networks. Both of them are of interest, because they could be applied to a wide-class of network architectures. Contrastive-Hebbian learning updates synaptic weights based on pre- and postsynaptic activities only, and therefore is suitable for implementation in biological networks. In contrast, backpropagation has been argued as implausible for biological networks, though it is very powerful and has been widely used in engineering problems. In this Chapter, we establish an equivalence between these two algorithms, when implemented in a layered network. This suggests that the functionality of backpropagation can be realized in biological networks, by using contrastive-Hebbian learning.

In Chapter 7, we introduce a synaptic learning rule for a network of spiking neurons. We derive the algorithm base on the assumption that neurons fire Poisson

spike trains. After that, we implement this algorithm in a network of integrate-and-fire neurons with white noise injected. We show that the learning rule could be used to learning XOR computation in such networks. The learning rule we suggest is very different from traditional Hebbian learning idea. The rule takes a form that depends on the correlation between fluctuation of postsynaptic firing and presynaptic EPSP, modulated by a global reward signal.

The network models we study in Chapters 2-4 are rate-based. To what extent these rate-based models can describe behaviors of real neurons is a critical question. In the past, people have studied how to reduce real spiking dynamics to rate-based models using the method of average or mean field approximation. However, all these studies depends on strong assumptions, which may not hold for real neuronal networks. In fact, rate-based models ignore all temporal information of spike trains, which has been argued recently as a method for coding information other than rates. Therefore, in the future, studies in dynamics directly in spiking networks seem important.

Understanding learning may be key to understanding the neural systems. Studies of dynamics have relied on a strong assumption that essential components of neural computation are simple and could be studied through simplification and approximation. This assumption, however, may not be true for some neural systems in which no simple law or principles can be found. Indeed, some people in neuroscience believe that neural systems are like complex machines consisting of large number of components, assembled together through evolution. This school of thoughts believes that there is no simple law underlying neural systems. If this is true, to understand neural systems by analyzing dynamics or understanding how each component works seems hopeless. Rather, a more sensible approach seems to be related to learning. It seems critical to understand learning rules that are used to change synaptic strength. Maybe in neural systems, there are only a small number of learning rules. If we can understand these learning rules, we can then train networks performing similar computations. As a consequence, we only need to look at the input-output relationships with no needs to knowing the details on how the computations are done.

Despite many years of researches, people still know little about how the brain

learns. In machine learning research, people have proposed many impressive algorithms. However most of these algorithms, which aimed at solving engineering problems, have little relevance to learning in neural systems. This contrasts with researches in synaptic physiology, where biophysical learning rules have been proposed, but have little computational power. It seems important to combine these two types of researches.

The hedonistic neuron idea presented in Chapter 7 is interesting in several aspects. It is both biophysically plausible and computationally powerful. It can also be directly tested in experiments. Investigating this type of learning rules in experiments is a great challenge to experimental neuroscientists.

Bibliography

[1] W. Reichardt. Autocorrelation: a principle for the evaluation of sensory information by the central nervous system. In A. Rosenblith, editor, *Sensory Communication*, pages 303–317. MIT Press and John Wiley and Sons, New York, 1961.

[2] C Koch and T Poggio. The synaptic veto mechanism: does it underlie direction and orientation selectivity in the visual cortex. In D Rose and V G Dobson, editors, *Models of the Visual Cortex*, pages 15–34. John Wiley, 1989.

[3] E. H. Adelson and J. R. Bergen. Spatiotemporal energy models for the perception of motion. *J. Opt. Soc. Am. A*, 256:284–99, 1985.

[4] H. Suarez, C. Koch, and R. Douglas. Modeling direction selectivity of simple cells in striate visual cortex within the framework of the canonical microcircuit. *J. Neurosci.*, 15:6700–19, 1995.

[5] R. Maex and G. A. Orban. Model circuit of spiking neurons generating directional selectivity in simple cells. *J. Neurophysiol.*, 75:1515–45, 1996.

[6] H.T. Blair, P.E. Sharp, and J. Cho. The anatomical and computational basis of the rat head-direction cell signal. *Trends in Neurosciences*, 24(5):289–294, 2001.

[7] H.T. Blair and P.E. Sharp. Anticipatory head direction signals in anterior thalamus: evidence for a thalamocortical circuit that integrates angular head

motion to compute head direction. *The Journal of Neuroscience*, 15(9):6260–6270, 1995.

[8] H.T. Blair, J. Cho, and P.E. Sharp. Role of the lateral mammillary nucleus in the rat head direction circuit: A combined single unit recording and lesion study. *Neuron*, 21:1387–1397, 1998.

[9] R.W. Stackman and J.S. Taube. Firing properties of lateral mammilary single units: head direction, head pitch and angular head velocity. *The Journal of Neurophysiology*, 18(21):9020–9037, 1998.

[10] Kechen Zhang. Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: A theory. *J. Neurosci.*, 16(6):2112–2126, 1996.

[11] A.D. Redish, A. N. Elga, and D.S. Touretzky. A coupled attractor model of the rodent head direction system. *Network: Computation in Neural Systems*, 7:671–685, 1996.

[12] J.P. Goodridge and D.S. Touretzky. Modeling attractor deformation in the rodent head-direction system. *The Journal of Neurophysiology*, 83:3402–3410, 2000.

[13] S. Amari and M. A. Arbib. Competition and cooperation in neural nets. In J. Metzler, editor, *Systems Neuroscience*, pages 119–165. Academic Press, 1977.

[14] J. Feng and K.P. Hadeler. Qualitative behaviour of some simple networks. *J. Phys. A:*, 29:5019–5033, 1996.

[15] R. H. Hahnloser, R. Sarpeshkar, M. Mahowald, R. J. Douglas, and H. S. Seung. Digital selection and ananlog amplification coexist in a silicon circuit inspired by cortex. *Nature*, 405:947–51, 2000.

[16] H. Markram, J. Lubke, M. Frotscher, and B. Sakmann. Regulation of synaptic efficacy by coincidence of postsynaptic aps and epsps. *Science*, 275(5297):213–5, 1997.

[17] G. Q. Bi and M. M. Poo. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *J Neurosci*, 18(24):10464–72, 1998.

[18] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back–propagating errors. *Nature*, 323:533–536, 1986.

[19] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. I: Foundations*, pages 318–362. Bradford Books/MIT Press, Cambridge, MA., 1986.

[20] J. Movellan. Contrastive hebbian learning in the continuous hopfield model. In D. Touretzky, J. Elman, T. Sejnowski, and G. Hinton, editors, *Proceedings of the 1990 Connectionist Models Summer School (pp. 10–17). San Mateo, CA.* Morgan Kaufmann., 1990.

[21] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski. A learning algorithm for Boltzmann Machines. *Cognitive Science*, 9:147–169, 1985.

[22] P. Dayan and L. Abbott. *Theoretical Neuroscience.* MIT Press, Cambridge, MA, 2001., 2001.

[23] A. Pouget, P. Dayan, and R. Zemel. Information processing with population codes. *Nat Rev Neurosci*, 1:125–32, 2000.

[24] M. A. Paradiso. A theory for the use of visual orientation information which exploits the columnar structure of striate cortex. *Biol Cybern*, 58:35–49, 1988.

[25] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

[26] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, 1998.

[27] J. P. van Santen and G. Sperling. Elaborated reichardt detectors. *J. Opt. Soc. Am. A*, 256:300–21, 1985.

[28] A. B. Watson and A. J. Ahumada. Model of human visual-motion sensing. *J. Opt. Soc. Am. A*, 256:322–41, 1985.

[29] P. Mineiro and D. Zipser. Analysis of direction selectivity arising from recurrent cortical interactions. *Neural Comput.*, 10:353–71, 1998.

[30] D. Golomb, X. J. Wang, and J. Rinzel. Propagation of spindle waves in a thalamic slice model. *J. Neurophysiol.*, 75:750–69, 1996.

[31] J. Rinzel, D. Terman, X. Wang, and B. Ermentrout. Propagating activity patterns in large-scale inhibitory neuronal networks. *Science*, 279:1351–5, 1998.

[32] D. Golomb and G. B. Ermentrout. Continuous and lurching traveling pulses in neuronal networks with delay and spatially decaying connectivity. *Proc. Natl. Acad. Sci. USA*, 96(23):13480–13485, 1999.

[33] D. Golomb and G. B. Ermentrout. Bistability in pulse propagation in networks of excitatory and inhibitory populations. *Physical Review Letters*, 86:4179–4182, 2001.

[34] M. von Krosigk, T. Bal, and D. A. McCormick. Cellular mechanisms of a synchronized oscillation in the thalamus. *Science*, 261:361–4, 1993.

[35] T. Bal, M. von Krosigk, and D. A. McCormick. Synaptic and membrane mechanisms underlying synchronized oscillations in the ferret lateral geniculate nucleus in vitro. *J. Physiol.*, 261:641–63, 1995.

[36] S. P. Sabatini and F. Solari. An architectural hypothesis for direction selectivity in the visual cortex: the role of spatially asymmetric intracortical inhibition. *Biol. Cybern.*, 80:171–83, 1999.

[37] G. B. Ermentrout and J. B. McLeod. Existence and uniqueness of travelling waves for a neural network. *Proc. Roy. Soc. Edinburgh Sect. A*, 123:461–478, 1993.

[38] S. Amari. Dynamics of pattern formation in lateral-inhibition type neural fields. *Biol. Cybern.*, 27(2):77–87, 1977.

[39] D. Hansel and H. Sompolinsky. Modeling feature selectivity in local cortical circuits. In C. Koch and I. Segev, editors, *Methods in Neuronal Modeling*, pages 499–567. MIT Press, Cambridge, Massachusetts, 1998.

[40] D. J. Pinto and G. B. Ermentrout. Spatially structured activity in synaptically coupled neuronal networks: I. traveling fronts and pulses. *SIAM J. Appl. Math.*, 62:206–225, 2001.

[41] D. J. Pinto and G. B. Ermentrout. Spatially structured activity in synaptically coupled neuronal networks: II. lateral inhibition and standing pulses. *SIAM J. Appl Math.*, 62:226–243, 2001.

[42] H. R. Wilson and J. D. Cowan. A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue. *Kybernetik*, 13(2):55–80, 1973.

[43] R Ben-Yishai, R L Bar-Or, and H Sompolinsky. Theory of orientation tuning in visual cortex. *Proc. Natl. Acad. Sci. USA*, 92:3844–3848, 1995.

[44] E. Salinas and L. F. Abbott. A model of multiplicative neural responses in parietal cortex. *Proc. Natl. Acad. Sci. USA*, 93:11956–11961, 1996.

[45] M A Giese. *Dynamic Neural Field Theory of Motion perception.* Kluwer Academic Publishers, Dordrecht, 1999.

[46] D. Golomb and G. B. Ermentrout. Effects of delay on the type and velocity of travelling pulses in neuronal networks with spatially decaying connectivity. *Network: Comput. Neural Syst.*, 11:221–246, 2000.

[47] R Hahnloser, R J Douglas, M Mahowald, and K Hepp. Feedback interactions between neuronal pointers and maps for attentional processing. *Nature Neuroscience*, 2(8):746–752, 1999.

[48] M Carandini and D Ferster. Membrane potential and firing rate in cat primary visual cortex. *J. Neurosci.*, 20:470–484, 2000.

[49] D Pelinovsky and V.G. Yakhno. fronts of multiple transitions in neural network media. *Neural Network World*, 4:443–456, 1993.

[50] D Jancke, W Erlhagen, H R Dinse, A C Akhavan, M Giese, A Steinhage, and G Schöner. Parametric population representation of retinal location: neuronal interaction dynamics in cat primary visual cortex. *J. Neurophysiol.*, 19(20):9016–9028, 1998.

[51] Pouget A Zemel R S, Dayan P. Probabilistic interpretation of population codes. *Neural Comput.*, 10(2):403–430, 1998.

[52] K Zhang, I Ginzburg, B L McNaughton, and T J Sejnowski. Interpreting neuronal population activity by reconstruction: unified framework with application to hippocampal place cells. *J. Neurophysiol.*, 79(2):1017–1044, 1998.

[53] M R Mehta, M C Quirk, and M A Wilson. Experience-dependent asymmetric shape of hippocampal receptive fields. *Neuron*, 25(3):707–725, 2000.

[54] R. H. Hahnloser. About the piecewise analysis of networks of linear threshold neurons. *Neural Networks*, 11:691–697, 1998.

[55] John P. F. Sum and Peter K. S. Tam. Note on the maxnet dynamics. *Neural Computation*, 8(3):491–499, 1996.

[56] R. Coultrip, R. Granger, and G. Lynch. A cortical model of winner-take-all competition via lateral inhibition. *Neural Networks*, 5:47–54, 1992.

[57] W. Maass. On the computational power of winner-take-all. *Neural Comput*, 12:2519–35, 2000.

[58] T. Kohonen. *Self-Organization and Associative Memory.* Springer-Verlag, Berlin, 3 edition, 1989.

[59] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

[60] D. D. Lee and H. S. Seung. Learning the parts of objects by nonnegative matrix factorization. *Nature*, 401:788–91, 1999.

[61] J.J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA*, 79, 1982.

[62] D. J. Willshaw, Buneman, and H. C. Longuet-Higgins. Nonholographic associative memory. *Nature*, 222:960–962, 1969.

[63] D. Willshaw and H. Longuet-Higgins. Associative memory models. *Machine Intelligence*, 351, 1970.

[64] J. J. Hopfield. Neurons with graded response have collective properties like those of two-state neurons. *Proc. Natl. Acad. Sci. USA*, 81:3088–3092, 1984.

[65] D. A. Miller and S. W. Zucker. Computing with self-excitatory cliques: A model and an application to hyperacuity-scale computation in visual cortex. *Neural Comput*, 11:21–66, 1999.

[66] D. Golomb, N. Rubin, and H. Sompolinsky. Willshaw model: Associative memory with sparse coding and low firing rates. *Physical Review A*, 41:1843–1854, 1990.

[67] Heiko Wersing, Wolf-Jurgen Beyn, and Helge Ritter. Dynamical stability conditions for recurrent neural networks with unsaturating piecewise linear transfer functions. *Neural Computation*, 13(8):1811–1825, 2001.

[68] H. K. Khalil. *Nonlinear systems.* Prentice Hall, NJ, 2 edition, 1996.

[69] D. J. Amit, H. Gutfreund, and H. Sompolinsky. Statistical mechanics of neural networks near saturation. *Ann. Phys. (New York)*, 173:30, 1985.

[70] R. P. Stanley. *Enumerative Combinatorics*, volume 1. Cambridge University Press, 1999.

[71] H. S. Seung and D. D. Lee. Cognition the manifold ways of perception. *Science*, 290:2268–9, 2000.

[72] H. Wersing, J. J. Steil, and H. Ritter. A competitive-layer model for feature binding and sensory segmentation. *Neural Comput*, 13:357–387, 2001.

[73] H. Wersing. Learning lateral interactions for feature binding and sensory segmentation. *Advances in Neural Information Processing Systems*, 2001.

[74] J. G. Hildebrand and G. M. Shepherd. Mechanisms of olfactory discrimination: converging evidence for common principles across phyla. *Annu Rev Neurosci*, 20:595–631, 1997.

[75] T. A. Christensen, V. M. Pawlowski, H. Lei, and J. G. Hildebrand. Multi-unit recordings reveal context-dependent modulation of synchrony in odor-specific neural ensembles. *Nature Neuroscience*, 12:927–31, 2000.

[76] K. Mori, H. Nagao, and Y. Yoshihara. The olfactory bulb: coding and processing of odor molecule information. *Science*, 286:711–5, 1999.

[77] H.T. Blair, B.W. Lipscomb, and P.E. Sharp. Anticipatory time intervals of head-direction cells in the anterior thalamus of the rat: implications for path integration in the head-direction circuit. *The Journal of Neurophysiology*, 78:145–159, 1997.

[78] A. P. Georgopoulos, J. F. Kalaska, R. Caminiti, and J. T. Massey. On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *J Neurosci*, 2:1527–37, 1982.

[79] E. Salinas and L. F. Abbott. Vector reconstruction from firing rates. *J Comput Neurosci*, 1:89–107, 1994.

[80] D. O. Hebb. *Organization of behavior.* Wiley, New York, 1949.

[81] C. C. Bell, H. Z. Han, Y Sugawara, and K. Grant. Synaptic plasticity in a cerebellum-like structure depends on temporal order. *Nature*, 387:278–81, 1997.

[82] W. Gerstner, R. Kempter, J. L. van Hemmen, and H. Wagner. A neuronal learning rule for sub-millisecond temporal coding. *Nature*, 383(6595):76–81, 1996.

[83] L. F. Abbott and S. Song. Temporally asymmetric hebbian learning, spike timing and neuronal response variability. *Adv. Neural Info. Proc. Syst.*, 11, 1999.

[84] H. S. Seung. Learning to integrate without visual feedback. *Soc. Neurosci. Abstr.*, 23(1):8, 1997.

[85] A. P. Georgopoulos, M. Taira, and A. Lukashin. Cognitive neurophysiology of the motor cortex. *Science*, 260:47–52, 1993.

[86] M. Camperi and X. J. Wang. A model of visuospatial working memory in prefrontal cortex: recurrent network and cellular bistability [in process citation]. *J Comput Neurosci*, 5(4):383–405, 1998.

[87] S. C. Cannon, D. A. Robinson, and S. Shamma. A proposed neural network for the integrator of the oculomotor system. *Biol. Cybern.*, 49:127–136, 1983.

[88] H. S. Seung. How the brain keeps the eyes still. *Proc. Natl. Acad. Sci. USA*, 93:13339–13344, 1996.

[89] H. S. Seung, D. D. Lee, B. Y. Reis, and D. W. Tank. Stability of the memory of eye position in a recurrent network of conductance-based model neurons. *Neuron*, 2000.

[90] B. Ermentrout. Reduction of conductance-based models with slow synapses to neural nets. *Neural Comput.*, 6:679–695, 1994.

[91] O. Shriki, H. Sompolinsky, and D. Hansel. Rate models for conductance based cortical neural networks. *preprint*, 1999.

[92] H. S. Seung, D. D. Lee, B. Y. Reis, and D. W. Tank. Short-term analog memory storage by a conductance-based model neuron with an excitatory autapse. *Journal of computation neuroscience*, 2000.

[93] P. Baldi and F. Pineda. Contrastive learning and neural oscillator. *Neural Computation*, 3:526–545, 1991.

[94] C. Peterson and J. Anderson. A mean field theory learning algorithm for neural networks. *Complex Systems*, 1:995–1019, 1987.

[95] G. E. Hinton. Deterministic boltzmann learning performs steepest descent in weight-space. *Neural Computation*, 1:143–150, 1989.

[96] J. J. Hopfield. Learning algorithms and probability distributions in feed-forward and feed-back networks. *Proc. Natl. Acad. Sci. USA*, 84:8429–8433, Dec 1987.

[97] R. O'Reilly. Biologically plausible error-driven learning using local activation differences: The generalized recirculation algorithm. *Neural Computation*, 8:895–938, 1996.

[98] G. E. Hinton and J. McClelland. Learning representations by recirculation. In D. Z. Anderson, editor, *Neural Information Processing Systems*, New York, NY, 1988. American Institute of Physics.

[99] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 1999.

[100] D. Zipser and R. A. Andersen. A back-propagation programmed network that simulates response properties of a subset of posterior parietal neurons. *Nature (London)*, 331:679–684, 1988.

[101] F. Crick. The recent excitement about neural networks. *Nature*, 337:129–132, 1989.

[102] P. Mazzoni, R. A. Andersen, and M. I. Jordan. A more biologically plausible learning rule for neural networks. *Proc. Natl. Acad. Sci. USA*, 88:4433–4437, May 1991.

[103] F. J. Pineda. Generalization of backpropagation to recurrent neural networks. *Physical Review Letters*, 18:2229–2232, 1987.

[104] TW Troyer and KD Miller. Physiological gain leads to high isi variability in a simple model of a cortical regular spiking cell. *Neural Comput*, 9(5):971–83, Jul 1 1997.

[105] WR Softky and C Koch. The highly irregular firing of cortical cells is inconsistent with temporal integration of random epsps. *J Neurosci*, 13(1):334–50, Jan 1993.

[106] J. Baxter and P.L. Bartlett. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:319–350, 11 2001.

[107] N Brunel and V Hakim. Fast global oscillations in networks of integrate-and-fire neurons with low firing rates. *Neural Comput*, 11(7):1621–71, Oct 1 1999.