Change-based Population Coding

by

Reza Moazzezi

Gatsby Computational Neuroscience Unit
University College London
17 Queen Square
London WC1N 3AR, United Kingdom

THESIS

Submitted for the degree of Doctor of Philosophy
University College London

2011

I, Reza Moazzezi confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

One standard interpretation of networks of cortical neurons is that they form dynamical attractors. Computations such as stimulus estimation are performed by mapping inputs to points on the networks' attractive manifolds. These points represent population codes for the stimulus values. However, this standard interpretation is hard to reconcile with the observation that the firing rates of such neurons constantly change following presentation of stimuli. Furthermore, these population codes are not robust to both dynamical noise and synaptic noise and learning the corresponding weight matrices has never been demonstrated which seriously limits the extent of their application. In this thesis, we address this problem in the context of an invariant discrimination task. We suggest an alternative view, in which computations that are performed over the course of the transient evolution of a recurrently-connected network are read out by monitoring the change in a readily computed statistic of the activity of the network. Such changes can be inherently invariant to irrelevant dimensions of variability in the input, a critical capacity for many tasks. We illustrate these ideas using a well-studied visual hyperacuity task, in which the computation is required to be invariant to the overall retinal location of the input. We show a class of networks based on a wide variety of recurrent interactions that perform nearly as well as an ideal observer for the task, and are robust to significant levels of noise. We also show that this way of performing computations is fast, accurate, readily learnable and robust to various forms of noise.

Acknowledgments

This research was supervised by Professor Peter Dayan. This thesis is a result of our wonderful scientific discussions. Peter was always around to answer questions and it was great that our scientific interactions started two years before I started my PhD.

I would also like to acknowledge my family for their support. This research was funded by Gatsby Charitable Foundation.

Contents

Chapter 2

Chapter 3

Chapter 4

Chapter 5

List of figures

List of Appendices

Chapter 1

Introduction

1.1 Introduction

Cortical computation consists of transformations whose ultimate goal is to extract the task relevant information from the stream of the incoming stimuli. These computations are carried out by cortical circuits and represented by the joint activity of the interconnected neurons (population code) building these circuits (Hubel and Wiesel, 1959; Snippe and Koenderink, 1992). The dynamical interactions among these recurrently connected nonlinear neurons are widely believed to play a central role in neural information processing (Douglas and Martin, 1989); One prominent theoretical account of these networks considers them in terms of attractors, with computation consisting in the mapping of inputs by recurrent dynamics into particular attractors, or particular locations on a continuous attractor. A read-out mechanism, which is often some form of feedforward network, then reports a characteristic of the final, attracted, state.

One class of such networks involves point attractors, which act as potential memories to be recalled from partial or noisy inputs (Hopfield, 1982). A class that has more recently been investigated, called "surface (line) attractor networks" (Seung, 1996; Zhang, 1996; Camperi and Wang, 1997; Pouget et al., 1998; Deneve et al., 2001; Wang, 2001; Wu et al., 2001; Renart et al., 2003; Wu and Amari, 2005), involves (null-stable) attractive manifolds which define population coded representations of continuous valued stimuli, such as the orientation of a visual bar or the direction the head of a rat is pointing in an environment. Attractor-based computation has been shown to perform nearly as well as is statistically possible on an important set of problems involving estimating such continuous-valued quantities (Pouget et al., 1998; Deneve et al., 2001).

However, there is little evidence that such attractor states exist in cortical networks. Neurons in sensory cortical areas rarely exhibit persistent activity in vivo (Vinje and Gallant, 2000; Reinagel, 2001), which suggests that their states do not converge to any attractor. Even neurons in prefrontal cortical areas that are well known for their persistent activity during delayed memory tasks, show systematic changes in their activity levels during delay periods (Rainer and Miller, 2002; Brody et al., 2003). Such changes in activity during persistent firing are inconsistent with the view that the state of the network has converged to a conventional attractor.

Besides this, to the best of our knowledge, the attractive manifold framework (line attractor framework) of information coding in recurrent networks has never been successfully applied to an important class of tasks involving discrimination in the presence of invariance like hyperacuity tasks (Zhaoping et al., 2003). Hyperacuity tasks have been proved to be useful for studying different aspects of neural coding (Poggio et al., 1992; Snippe and Koenderink, 1992) and information processing in recurrent networks (Zhaoping et al., 2003). In particular, Zhaoping et al (Zhaoping et al., 2003) studied an invariant discrimination task in the context of the Bisection hyperacuity task (Crist et al., 1997) and showed that although a recurrent network solution to this problem using the conventional line attractor-based approach performs better than the first and the second order approximations to the ideal observer model of the task, but the performance remained far from optimal. This sub-optimal performance level of the conventional line attractor-based approach and the assumptions it makes about the existence of the attractors which does not seem to be biologically plausible remain to be addressed.

In this thesis we address these problems by studying tasks involving invariant discrimination in the context of recurrent networks; similar to (Zhaoping et al., 2003), we largely consider variants of a visual psychophysical task, namely, bisection (Crist et al., 1997), which belongs to the class of perceptual learning hyperacuity tasks (Westheimer and McKee, 1977b). Rather than coding information by the attractor states of the recurrent network, we develop a new coding method that encodes information by the change of the neural activity pattern over time shortly after the stimulus onset. In particular, we show that coding information by the evolution of the pattern of the neural activity, and specifically by the change in a statistics of the neural activity during its early temporal evolution, allows optimal performance. Furthermore, since the information is coded by the early portion of the neural activity, this coding strategy, unlike the conventional attractor-based coding, does not depend on the attractor state of the network. This coding method extends population coding to the temporal domain. We refer to this new coding method as “change-based” coding. We will show that learning optimal recurrent weights is possible within the change-based framework and is fast and easy (chapter 5). The task and the model that we use allow us to study neural coding in a rather abstract way which in turn allows us to extend the temporal coding theory that we develop in this thesis to a broader class of tasks and computations (see chapter 5).

We will study the change-based processing mainly in the context of invariant discrimination which can be regarded as a specific form of the more general invariance problem that has been extensively studied in the context of object recognition (Logothetis and Sheinberg, 1996; Tanaka, 1996). Our visual system seems to recognize objects rapidly and effortlessly (Thorpe et al., 1996; Fabre-Thorpe et al., 1998); such an impressive capability is a result of both invariance and selectivity: the underlying computations are selective to those features that define the objects to be recognized while are invariant to others. Neurons showing both selectivity to complex stimuli like faces and invariance to transformations such as scale and position changes have been observed in areas like IT (Infrotemporal cortex) (Gross et al., 1972; Desimone et al., 1984; Desimone, 1991; Perrett et al., 1992; Perrett and Oram, 1993; Logothetis and Sheinberg, 1996; Tanaka, 1996). It is widely believed that such selectivity and invariance is achieved through progressive filtering of the visual information along the visual pathway. As a result, several hierarchical feedforward models have been suggested to account for the experimental observations but have achieved limited success (Fukushima, 1980; Riesenhuber and Poggio, 1999, 2000; Serre et al., 2007; Pinto et al., 2008). Invariant discrimination that we study here addresses the same two key computations needed for invariant object recognition: selectivity and invariance.

We will introduce change-based coding in chapter 2 and study its workings in chapter 3. We will study its relation to conventional attractor-based coding in chapter 4 and finally we will study learning in chapter 5. Chapters 2 and 4 of this thesis have been mainly taken from (Moazzezi and Dayan, 2008) and (Moazzezi and Dayan, 2010), respectively.

In this chapter, we will discuss and review the following: Since change-based coding is built up on population coding, we first discuss population coding and optimal decoding for both estimation and discrimination tasks within population coding framework (we are mainly interested in discrimination tasks but optimal discrimination is theoretically related to optimal estimation). We will also discuss network models for estimation. In particular, we will discuss the line attractor recurrent networks that have been shown to perform estimation tasks optimally (Pouget et al., 1998; Deneve et al., 2001). We develop change-based coding for recurrent networks in the context of the tasks that involve invariant discrimination; therefore we briefly discuss feedforward network models that have been suggested for invariant discrimination and recognition. We will discuss attractor recurrent network models that have been suggested for invariant discrimination at the beginning of chapter 2. Bisection hyperacuity task is the model task that we use to study the change-based coding and therefore we discuss this class of tasks (perceptual learning and hyperacuity tasks) too. Finally, we will discuss network models of perceptual learning.

1.2 Population Coding

It is widely believed that an accurate description of information processing by cortical circuits can be achieved by examining them at the level of their spiking activity. A huge wealth of work has been done to find out how information is coded and represented by the spiking activity of the cortical circuits. The nature of this representation and in particular the role of the temporal structure of the neural activity in it has been fervently debated. The proposed theories range from those that lay emphasis on the (relative) timing of the individual spikes (spike timing code), such as rank order coding (Thorpe et al., 1996; Van Rullen and Thorpe, 2001; VanRullen and Thorpe, 2001) and perhaps with sub-millisecond accuracy (Mainen and Sejnowski, 1995; Buracas et al., 1998; Reinagel and Reid, 2002), those that emphasize on averaging over time windows between a few hundreds of milliseconds to several seconds (Adrian, 1926; van Vreeswijk and Sompolinsky, 1996; Shadlen and Newsome, 1998) to those that emphasize on the temporal structure of the neural activity at the population level (Huber et al., 2008; Yang et al., 2008). For instance, some recent experiments performed *in vivo* in awake behaving animals substantiate the importance of the temporal structure of the neural activity in coding in a number of different modalities and brain areas (Houweling and Brecht, 2008; Huber et al., 2008; Yang et al., 2008). For example, Yang et al. (Yang et al., 2008) recently showed that a temporal delay as small as 5 ms between the activities of two nearby populations of cells in A1 (primary auditory cortex) has significant behavioural effect which supports the idea that the temporal structure of the neural activity (at the population level) might play a key role in cortical computation.

However, despite such huge differences in these theories, it is widely believed that the stimulus information is coded in the joint activity of populations of neurons (population coding (Hubel and Wiesel, 1959; Snippe and Koenderink, 1992)). Experiments show that population coding is extensively used by cortical circuits to represent information ranging from place cells in hippocampus for representing location of the rats (Hubel and Wiesel, 1959; O'Keefe and Dostrovsky, 1971), cells in primary visual cortex for representing orientation in a local patch of an image (Hubel and Wiesel, 1959), neurons in MT for representing direction of movement of oriented bars (Maunsell and van Essen, 1983; Maunsell and Van Essen, 1983b, a), neurons in primary motor cortex for representing direction of hand movements (Georgopoulos et al., 1986) to neurons in IT for representing complex objects (Tanaka et al., 1991).

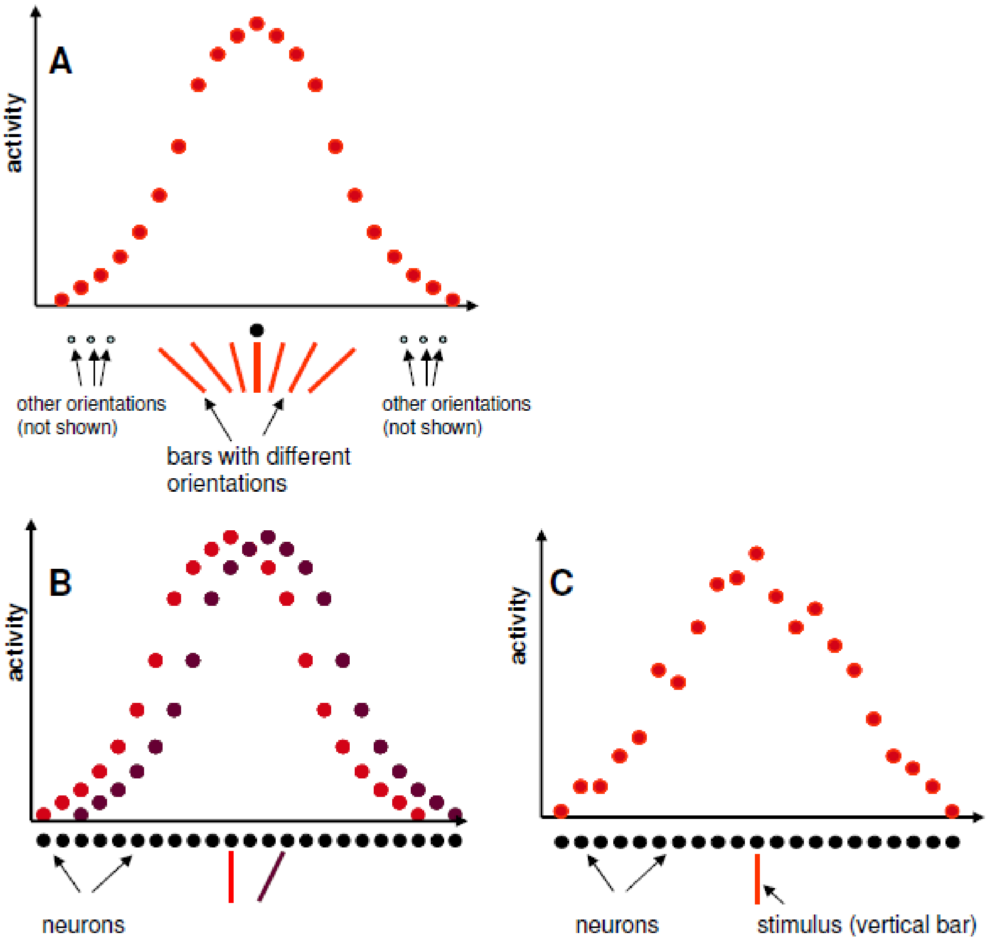
A
activity
other orientations (not shown)
bars with different orientations
other orientations (not shown)
B
activity
neurons
C
activity
neurons
stimulus (vertical bar)

Figure 1.1- Population coding. **A**) Orientation tuning curve. Cartoon representing the average response (average number of spikes emitted by a neuron over a long period of time, for instance, a few hundreds of milliseconds; red dots) of a neuron (black dot) to different orientations. The neuron responds optimally to vertical orientation and its response decreases monotonically as the orientation of the stimulus tilts away from vertical. **B**) Cartoon showing average population activity elicited by two different stimuli. It shows topographically arranged orientation selective neurons (black dots). Red dots represent average population activity elicited by a vertical bar (red bar) and purple dots represent population activity elicited by a bar tilted clockwise relative to vertical (purple bar). The bars are shown separately to the population but the responses are superimposed to illustrate the topographical arrangement of the orientation selective neurons. **C**) Noisy population response elicited by the vertical bar. The discrimination task involves estimating the orientation of the stimulus (which is vertical) from the pattern of noisy activities.

Von Neumann (Von Neumann, 1958) seem to be among the first to suggest population coding as he stated in his book “The Computer and the brain” that “It is therefore perfectly plausible that certain (statistical) relationships between such trains of pulses should also transmit information” (page 80).

Orientation representation in primary visual cortex (Hubel and Wiesel, 1959; Hubel and Wiesel, 1962) is perhaps the classical example of population coding in which the information about the orientation of a bar of light is coded by the joint activity of a population of orientation selective neurons (neurons’ activity is defined by their firing rate over a time window of a few hundreds of milliseconds) where each neuron responds to a range of orientations. Responses of V1 cells has been reported to evolve over time (Ringach et al., 1997) and the orientation selectivity of neurons in output layers of V1 has been shown to change during the temporal evolution of the neural activity. However, for simplicity, we only consider the spatial distribution of activity in primary visual cortex here. In addition, neurons are topographically arranged (and modelled by a one dimensional array of neurons) such that nearby cells have nearly similar preferences (figure 1.1). The average response of a neuron to bars with different orientations is characterized by its “tuning curve” (figure 1.1A; note that the average is over (ideally) infinite trials); the orientation to which it responds maximally is referred to as its “preferred orientation”. The tuning curve of orientation selective neurons is commonly modelled by a Gaussian:

$$f_i(\varphi) = ke^{\frac{-(\varphi-\theta_i)^2}{2\sigma^2}} \tag{1.1}$$

where i is the index of a (orientation selective) neuron on the one dimensional line (figure 1.1B), φ denotes the orientation of the bar of light, $f_i(\varphi)$ denotes the average response of neuron i to a bar of light with orientation φ, and θ_i is the preferred orientation of neuron i (i.e. the orientation to which it responds maximally).

We assume that all neurons have the same location preference and their only difference is in their orientation selectivity. We also assume that the shape of the tuning curves for different neurons (with different preferred orientations) are the same, just that they are shifted versions of each other; in other words, the shape of the orientation tuning curve is invariant to the preferred orientation of neurons.

However, because of noise, the response of the neurons to the same orientation is different on different trials. We model this probabilistically; the actual response r_i of neuron i to orientation φ at each trial can be modelled as follows:

$$r_i = f_i(\varphi) + \eta_i \tag{1.2}$$

where η_i denotes the noise which is drawn independently from a distribution with zero mean and finite variance and in general is represented by a probability distribution $P(\boldsymbol{\eta})$ (where $\boldsymbol{\eta} = (\eta_1, \eta_2, ..., \eta_n)$); we model $P(\boldsymbol{\eta})$ by Gaussian or Poisson distributions (Pouget et al., 1998; Deneve et al., 1999, 2001). Figure 1.1B cartoons the average response of a population of neurons to a stimulus with vertical orientation.

Figure 1.1C cartoons the response of the same population of neurons corrupted by noise. As we mentioned earlier, the task here is to estimate (decode) the orientation of the bar (estimated orientation is denoted by $\widehat{\varphi}$) from a vector of noisy population activity (denoted by $\boldsymbol{r} = (r_1, r_2, ..., r_n)$) generated by the stimulus. In the next section we explain optimal decoding methods that have been developed for decoding population activity.

1.2.1 Optimal decoding (estimation)

Before we discuss network models for decoding population activity, it is useful to study optimal decoding. Since this is a probabilistic inference problem, all the information about the orientation of the presented bar is included in the “posterior distribution over orientations given the observed noisy population activity” and can be obtained by applying the Bayes rule. This relies on two factors: the likelihood, that is, the probability of observing an activity pattern given the orientation of the stimulus, φ, which is denoted by $P(\boldsymbol{r} \mid \varphi)$. It is derived from equation 1.2 and depends on the distribution of the noise $P(\boldsymbol{\eta})$. The other factor is the prior over the orientation of the stimulus which is denoted by $P(\varphi)$. This reflects the probability of occurrence of each orientation. Since we are interested in the posterior $P(\varphi \mid \boldsymbol{r})$ we apply the Bayes rule:

$$P(\varphi \mid \boldsymbol{r}) = \frac{P(\boldsymbol{r} \mid \varphi)P(\varphi)}{P(\boldsymbol{r})} \tag{1.3}$$

where $P(\boldsymbol{r}) = \sum_{\varphi} P(\boldsymbol{r} \mid \varphi)P(\varphi)$; the optimal decoder is assumed to have access to both the posterior and the underlying cost function; for example, a common cost function is to minimize the variance of estimation, that is,

$$\mathrm{var} = \left\langle (\widehat{\varphi} - \varphi)^2 \right\rangle_{trials} \tag{1.4}$$

where the average $\langle . \rangle$ is over (ideally infinite) trials, the optimal estimate, $\widehat{\varphi}$, is the orientation that maximizes the posterior,

$$\widehat{\varphi} = \sum_{\varphi} \varphi P(\varphi \mid \mathbf{r}) \tag{1.5}$$

However, in the absence of any constraint, a natural choice for estimation is $\widehat{\varphi} = \underset{\varphi}{arg\,max}\ P(\varphi \mid \boldsymbol{r})$; This is referred to as maximum a posteriori (MAP) estimate. If the prior is flat, that is, $P(\varphi)$ in equation 1.3 is the same for all φ, then the estimate defined by equation 1.5 is referred to as maximum likelihood (ML) estimate. The variance of the ML estimate is related to the well-known Fisher Information, $I(\varphi)$:

$$I(\varphi) = -\left\langle \frac{\partial^2}{\partial \varphi^2} log\left(P(\boldsymbol{r} \mid \varphi)\right) \right\rangle_{\boldsymbol{r}} \tag{1.6}$$

Note that the average $\langle . \rangle$ here is over all realizations of $\boldsymbol{r}$. For Gaussian noise, where $P(\boldsymbol{\eta})$ is a multivariate Gaussian, the Fisher Information takes the following form:

$$I(\varphi) = \dot{\mathbf{f}}^T(\varphi)\mathbf{C}^{-1}\dot{\mathbf{f}}(\varphi) + \frac{1}{2} tr\left(\mathbf{C}^{-1}\dot{\mathbf{C}}(\varphi)\mathbf{C}^{-1}\dot{\mathbf{C}}(\varphi)\right) \tag{1.7}$$

where $\mathbf{C} = \mathbf{C}(\varphi) = \{C_{ij}(\varphi)\}$ denotes the noise correlation, $C_{ij} = \langle (r_i - f_i(\varphi))(r_j - f_j(\varphi)) \rangle_{trials}$, $\dot{\mathbf{C}}(\varphi) = \left\{ \frac{\partial C_{ij}(\varphi)}{\partial \varphi} \right\}$, and $\dot{\mathbf{f}}(\varphi) = \left(\frac{\partial f_1(\varphi)}{\partial \varphi}, \frac{\partial f_2(\varphi)}{\partial \varphi}, \ldots, \frac{\partial f_n(\varphi)}{\partial \varphi} \right)$. When the prior is flat, the minimum achievable variance for estimation is related to the Fisher Information as follows:

$$\sigma^2(\varphi) \geq \frac{1}{I(\varphi)} \tag{1.8}$$

Equation 1.8 is the well known Cramer-Rao bound that specifies the minimum achievable variance for unbiased estimators. Equation 1.6 offers a simple way of calculating the Fisher Information and therefore the minimum variance for estimation. Optimal decoding in discrimination tasks is also related to the Fisher information. We examine this relation in the next section.

1.2.2 Optimal decoding (discrimination)

In the previous section we discussed how to estimate a stimulus feature (for example, the orientation of a bar) from population activity. The Fisher Information not only is useful for studying optimal estimation, but also can be used to study optimal discrimination. To see this, consider a discrimination task in which two bars with orientations φ and $\varphi + \delta\varphi$ are shown to two identical networks, and the task is to decide whether the bar presented to one of the networks is tilted clockwise or counter-clockwise relative to the bar presented to the other. In other words, the task is to decide whether $\delta\varphi > 0$ or $\delta\varphi < 0$. For the case in which φ is fixed (i.e. the orientation shown to one of the networks is always the same) and $\delta\varphi$ changes over trials and the prior is flat, the optimal strategy for discrimination, i.e. the Maximum Likelihood (ML) discrimination, relies on the likelihood distribution given the response vectors. It can be shown that for small $\delta\varphi$ and large population of uncorrelated neurons, the probability of error for ML discrimination is (Seung and Sompolinsky, 1993):

P(error) = $\dfrac{1}{\sqrt{2\pi}}\displaystyle\int_{\frac{d'}{\sqrt{2}}}^{\infty} e^{-\frac{x^2}{2}}dx$ (1.9)

where $d'=|\delta\varphi|\sqrt{I(\varphi)}$. This means that higher Fisher Information (which is the inverse of the variance of estimation in estimation task (see equation (1.8))) implies lower probability of error for ML discrimination. Therefore, ML discrimination with lowest error probability corresponds to the highest Fisher Information. One way to see this intuitively is to imagine discrimination being performed by comparing the estimations of the two networks (figure 1.2) mentioned above (around orientation φ) which means that better estimation results in better discrimination which implies that the best discrimination is achieved through optimal estimation.

1.2.3 “Local discrimination” versus “discrimination in the presence of invariance”

In the following sections it is important to distinguish between two types of discrimination tasks: Local discrimination tasks and discrimination tasks in the presence of invariance. To define these two classes of tasks and see the difference between them, let’s take the discrimination task that we studied in the previous section. In both classes, the task is to decide whether the bar shown to one of the networks is tilted clockwise or counterclockwise relative to the bar shown to the other one. However, the first class, “local discrimination task”, involves tasks in which the orientation of the bar shown to one of the networks is fixed at all trials (φ is fixed; we refer to this as “reference orientation” or in general as “reference signal”) while the orientation of the bar shown to the other network (which is $\varphi+\delta\varphi$) changes over trials. In contrast, in the second class, “discrimination task in the presence of invariance”, the reference orientation φ is no longer fixed but rather takes a broad range of orientations and therefore the discrimination task is not local anymore. In this class of discrimination tasks the reference orientation φ is a nuisance parameter and carries no task relevant information; this is $\delta\varphi$ that is relevant to the task. In other words, the end result of the computations that are needed for optimal discrimination should only depend on $\delta\varphi$ and be invariant to φ.

In the next section, we discuss network models that have been proposed for both classes of discrimination tasks starting by local ones.

1.2.4 Local discrimination by a two-layer threshold-linear feedforward network

In feedforward models it is commonly assumed that the end result of computation is represented by the activity of the neurons at the top layer (also called the output layer). Each layer gets the output pattern of the previous layer, makes its own transformation, giving rise to its own output which is subsequently sent to the next layer and the readout decodes based on the activity pattern of the output layer. Network models for local discrimination have been previously studied (Seung and Sompolinsky, 1993) and a two-layer threshold-linear feedforward network have been shown to be optimal under certain conditions for the case in which the input noise comes from Poisson distribution. Here we study two-layer linear-threshold feedforward networks for the case in which the input noise is Gaussian and we show that they can optimally solve a local discrimination task provided that the noise is signal independent (covariance matrix does not depend on the mean). In particular, we show that such networks are sub-optimal when the noise distribution is signal dependent Gaussian.

To see this, consider a local discrimination task in which the network should decide if the orientation of a bar is tilted clockwise or counter clockwise relative to a vertical line ($\varphi = 90^\circ$, figure 1.2). We also assume that the orientation of the bar is always very close to the vertical (hence making it a challenging task). In the previous section we showed that how discrimination is related to estimation. Therefore, in order to show that a threshold-linear two-layer feedforward network optimally performs the discrimination task, all we need to show is that there exists a two-layer linear feedforward network that optimally performs the estimation task. To this end, we compute the variance of the best linear estimator (linear estimator is mathematically equivalent to a two-layer linear feedforward network) and compare it with the inverse of the Fisher Information. We show that they are exactly the same for local estimation task (under certain conditions that is mentioned above) which proves the optimality of the threshold-linear two-layer feedforward network for local discrimination task. To prove this, we assume that the stimulus driven activity pattern is denoted by $\boldsymbol{r}$ ($\boldsymbol{r} = (r_1, r_2, \dots, r_n)$) and the true orientation of the stimulus is φ. Linear readout estimates the orientation of the bar as follows:

$$\widehat{\varphi} = \sum_i w_i r_i \tag{1.10}$$

where $\hat{\varphi}$ denotes the estimation and w_i represents the weight associated with the activity of neuron i in the linear readout. The estimation has to be unbiased, therefore:

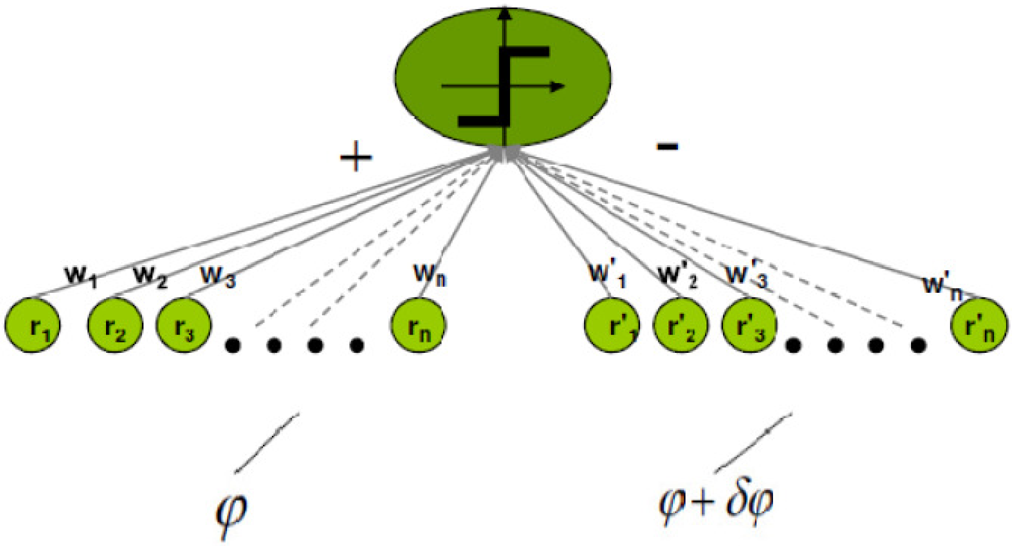

Figure 1.2- Discrimination versus estimation. Cartoon illustrating how to construct a threshold-linear network that discriminates based on the outputs of two linear networks that are supposed to estimate. The network shown on the left is presented by a bar with orientation φ that results in an activity pattern $\boldsymbol{r} = (r_1, r_2, \dots, r_n)$ in that network. The network shown on the right is presented by a bar with orientation $\varphi + \delta\varphi$ that results in an activity pattern $\mathbf{r}' = (r'_1, r'_2, \dots, r'_n)$ in that network. The network in the left is adapted to yield optimal estimates around orientation φ and the network on the right is adapted to yield optimal estimates around orientation $\varphi + \delta\varphi$. The two estimates of the networks are then subtracted and passed through a step nonlinearity to decide whether $\delta\varphi$ is positive or negative.

$$\varphi = \langle \hat{\varphi} \rangle = \left\langle \sum_i w_i r_i \right\rangle = \sum_i w_i \langle r_i \rangle = \sum_i w_i f_i(\varphi) \tag{1.11}$$

$\langle . \rangle$ denotes average over infinite trials and the last equality follows from equation 1.2. We rewrite the above equation in vector form:

$$\boldsymbol{w}^T \boldsymbol{f}(\varphi) = \varphi \tag{1.12}$$

where $\boldsymbol{w} = (w_1, w_2, ..., w_n)$ and $\boldsymbol{f}(\varphi) = (f_1(\varphi), f_2(\varphi), ..., f_n(\varphi))$. By taking the first derivative of the above equation with respect to φ we have:

$$\boldsymbol{w}^T \dot{\boldsymbol{f}}(\varphi) = 1 \tag{1.13}$$

where $\dot{\boldsymbol{f}}(\varphi) = \left(\frac{\partial f_1(\varphi)}{\partial \varphi}, \frac{\partial f_2(\varphi)}{\partial \varphi}, ..., \frac{\partial f_n(\varphi)}{\partial \varphi} \right)$. The variance of estimation, σ^2, is:

$$\sigma^2 = \left\langle (\hat{\varphi} - \varphi)^2 \right\rangle = \left\langle \left(\sum_i w_i r_i - \varphi \right)^2 \right\rangle = \left\langle \left(\sum_i w_i r_i - \sum_i w_i f_i(\varphi) \right)^2 \right\rangle$$
$$= \left\langle \left(\sum_i w_i (r_i - f_i(\varphi)) \right)^2 \right\rangle = \boldsymbol{w}^T \boldsymbol{C} \boldsymbol{w} \tag{1.14}$$

where $\boldsymbol{C} = \{C_{ij}\}$ and $C_{ij} = \left\langle (r_i - f_i(\varphi))(r_j - f_j(\varphi)) \right\rangle_{trials}$. Again, $\langle . \rangle$ denotes average over infinite trials. Here we assume that the noise in different neurons is independent, $C_{ij} = 0$ for $i \neq j$. We also assume that $\left. \frac{\partial \boldsymbol{C}(\varphi)}{\partial \varphi} \right|_{\varphi = 90^\circ} = \boldsymbol{0}$, i.e. correlation is stimulus independent. The objective is to minimize the variance of estimation (σ^2, see equation 1.14) with the constraint given by equation 1.13 which leads to the following Lagrangian:

$$L(\boldsymbol{w}) = \boldsymbol{w}^T \boldsymbol{C} \boldsymbol{w} - \lambda \left(\boldsymbol{w}^T \dot{\boldsymbol{f}}(\varphi) - 1 \right) \tag{1.15}$$

Minimizing $L(\boldsymbol{w})$ with respect to $\boldsymbol{w}$ leads to:

$$\frac{\partial L(\boldsymbol{w})}{\partial \boldsymbol{w}} = 0 \Rightarrow 2\boldsymbol{C}\boldsymbol{w} - \lambda\dot{\boldsymbol{f}}(\varphi) = 0 \Rightarrow 2\boldsymbol{C}\boldsymbol{w} = \lambda\dot{\boldsymbol{f}}(\varphi) \Rightarrow$$
$$\boldsymbol{w} = \frac{\lambda}{2}\boldsymbol{C}^{-1}\dot{\boldsymbol{f}}(\varphi) \tag{1.16}$$

From equations 1.16 and 1.13, we have:

$$\boldsymbol{w}^T\dot{\boldsymbol{f}}(\varphi) = 1 \Rightarrow \frac{\lambda}{2}\dot{\boldsymbol{f}}^T(\varphi)\boldsymbol{C}^{-1}\dot{\boldsymbol{f}}(\varphi) = 1 \Rightarrow \lambda = \frac{2}{\dot{\boldsymbol{f}}^T(\varphi)\boldsymbol{C}^{-1}\dot{\boldsymbol{f}}(\varphi)} \tag{1.17}$$

From equations 1.14, 1.16 and 1.17, the variance of the linear readout is:

$$\sigma^2 = \boldsymbol{w}^T\boldsymbol{C}\boldsymbol{w} = \frac{1}{\dot{\boldsymbol{f}}^T(\varphi)\boldsymbol{C}^{-1}\dot{\boldsymbol{f}}(\varphi)} \Rightarrow \frac{1}{\sigma^2} = \dot{\boldsymbol{f}}^T(\varphi)\boldsymbol{C}^{-1}\dot{\boldsymbol{f}}(\varphi) \tag{1.18}$$

Since we assume that $\dot{\boldsymbol{C}}(\varphi)\Big|_{\varphi=90^\circ} = \boldsymbol{0}$ equation 1.7 reduces to equation 1.18 and therefore the variance of the linear estimator (equation 1.18) and the variance of the optimal local estimator (equation 1.7) are equal. Thus the linear readout extracts all the Fisher Information. In the light of the relation between the Fisher Information and ML discrimination mentioned in the previous section (equation 1.9), we therefore conclude that the performance of the threshold-linear readout is optimal for discrimination task for the case in which the input noise comes from signal independent Gaussian noise.

However, comparing equations 1.18 and 1.7 indicates that the condition $tr\left(\mathbf{C}^{-1}\dot{\mathbf{C}}(\varphi)\mathbf{C}^{-1}\dot{\mathbf{C}}(\varphi)\right) = 0$ is necessary for the two-layer threshold-linear feedforward network to be optimal for the local discrimination task ($\dot{\boldsymbol{C}}(\varphi)\Big|_{\varphi=90^\circ} = \boldsymbol{0}$ is one way to satisfy this condition). If this condition doesn't hold, then even a local discrimination task can not be solved by a two-layer threshold-linear feedforward network. Similarly, discrimination tasks in the presence of invariance cannot be solved optimally by a two-layer threshold-linear feedforward structure. To address this class of tasks, a different network structure is required. Since better estimation results in better discrimination, we review optimal network models that have been suggested for estimation in the next section. We discuss network models that have been suggested for discrimination in the presence of invariance in the next chapter.

1.3 Recurrent networks for estimation

Here we review line attractor networks (Seung, 1996; Zhang, 1996; Pouget et al., 1998) as it is shown that they can perform some estimation tasks optimally (Pouget et al., 1998; Deneve et al., 1999, 2001). Line attractor networks are a subclass of attractor networks (Hopfield, 1982) in which the attractors can be described by a set of low dimensional manifolds in the state space defined by the joint activity of the neurons. One way to create such low dimensional manifolds is by introducing a recurring pattern of connectivity across the network. This is indeed a powerful way to address the complexities of the neocortical structure through the filter of a canonical microcircuit (Douglas and Martin, 1989, 1991). Figure 1.3 shows a cartoon of a one dimensional example of such a network with a recurring connectivity pattern. If, in this one dimensional network, we denote the connection from a neuron at location x_j to a neuron at location x_i by W_{ij}, then a recurring connectivity pattern emerges if $W_{ij} = W\left(x_i - x_j\right)$, i.e. if connection between two any neurons only depends on their distance. Figure 1.4 cartoons three example attractor states associated with this network. Note that since the connectivity is repeated across the network, the attractors can be divided into several groups, where the attractors within each group are similar in shape despite being at different locations (figure 1.4). When represented in the high dimensional state space of the network (figure 1.5), attractors that belong to the same group form a one dimensional manifold (in fact this is why these networks are referred to as “line attractor” networks).Figure 1.5 cartoons an example line attractor. Although each line attractor consists of individual attractor points sitting next to each other and therefore it is not a continuous one dimensional manifold, it better approximates the underlying manifold as the density of neurons increases in the network.

Simulations showed that the line attractor networks can perform a class of estimation tasks near optimally (Pouget et al., 1998). To see this, we illustrate the workings of the line attractor networks by studying an orientation estimation task similar to the one we discussed in the previous section. A bar is presented to the line attractor network with the task for the network and the attractor-based readout being to estimate its orientation (Ben-Yishai et al., 1995). This is a one dimensional network of V1 like (rate-based) neurons with a recurring structure that defines one dominant line attractor

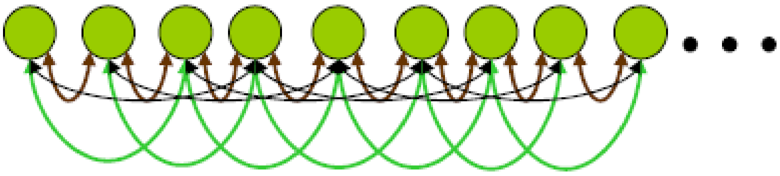

Figure 1.3- A one dimensional network with recurring connectivity pattern. The strength of the connection between any two neurons depends only on their distance from each other which is represented by different colors.

with a symmetric shape. Different neurons in the network are assumed to have the same location preference.

However, their orientation selectivity emerges from the dynamics and the connectivity of the network and can be approximated by a Gaussian. The stimulus elicits noisy activity, $\boldsymbol{r}$, according to equation 1.2 and this noisy activity serves as the input to the recurrent network; the network begins to evolve according to the following dynamics and initial condition:

$$\tau \frac{d\boldsymbol{u}_t}{dt} = -\boldsymbol{u}_t + \boldsymbol{W}\boldsymbol{g}(\boldsymbol{u}_t) \qquad (1.19)$$

$$\boldsymbol{u}_0 = \boldsymbol{r}$$

where $\boldsymbol{W} = \{W_{ij}\}$ is the weight matrix and $W_{ij} = W(|x_i - x_j|)$), $\boldsymbol{u}_t = (u_{1,t}, u_{2,t}, \ldots, u_{n,t})$ is the activity vector at time t and $\boldsymbol{g}(\boldsymbol{a})$ is defined as $\boldsymbol{g}(\boldsymbol{a}) = (g(a_1), g(a_2), \ldots, g(a_n))$ for an arbitrary vector $\boldsymbol{a} = (a_1, a_2, \ldots, a_n)$ where $g(.) : \Re \to \Re$ is a nonlinear activation function. Note that the input is provided transiently to the network as it appears in the form of the initial condition of the differential equation. The differential equation evolves and converges when $\frac{d\boldsymbol{u}_t}{dt} = \boldsymbol{0}$; the converged state, $\boldsymbol{u}_\infty = (u_{1,\infty}, u_{2,\infty}, \ldots, u_{n,\infty})$, satisfies the equation

$$\boldsymbol{u}_\infty = \boldsymbol{W}\boldsymbol{g}(\boldsymbol{u}_\infty) \qquad (1.20)$$

The form of an example converged state is cartooned in figure 1.6. As mentioned above, the readout mechanism employed here is attractor-based, i.e. decoding is based on $\boldsymbol{u}_\infty$. In this case, as we mentioned earlier, the weight matrix and the activation function are such that the converged state, $\boldsymbol{u}_\infty$, is approximately Gaussian; decoding consists of finding the parameter $\hat{\theta}$ for which we have:

$$u_{i,\infty} = ke^{\frac{-(\hat{\theta}-\theta_i)^2}{2\sigma^2}} \quad \text{for all } i \tag{1.21}$$

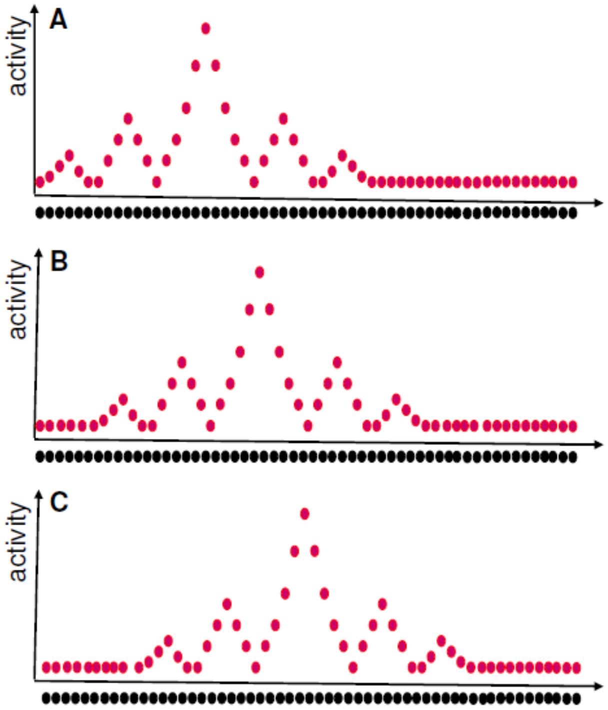

Figure 1.4- Line attractors in the network space**.** **A**, **B** and **C** cartoon three attractors (of a one dimensional network) that have the same shape but are at different locations. Neurons are represented by black circles and activity is represented by red circles. Here the attractors are symmetric but in general they do not need to be. The attractors of the networks with recurring connectivity like the one shown in figure 1.3 will have this property.

$\hat{\theta}$ is the estimated orientation of the bar by the attractor-based readout (note that such $\hat{\theta}$ does exist as the assumption is that the converged state is nearly Gaussian). In other words, the location of the converged state of the network on the low dimensional attractor manifold encodes and represents the orientation of the stimulus. Pouget et al showed that an appropriate choice of weight matrix and nonlinearity results in the attractor-based estimator performing as well as the ML estimator in this orientation estimation task (Pouget et al., 1998; Deneve et al., 1999, 2001); indeed the curve-fitting nature of the readout mechanism, and its near ML performance seemed to support the idea that the network is indeed performing ML estimation (Zemel et al., 1998): the network gets the noisy input and, similar to the ML estimator, network's processing aims to find the optimal fit to the noisy data.

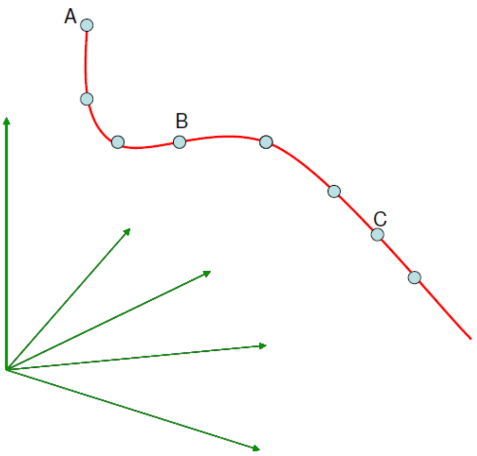

Figure 1.5 – Line attractors in activity space. A representation of the attractor states of the network in N dimensional space (N is the number of neurons). The activity of each neuron is represented along one of the coordinates (green lines). Each blue dot represents one of the attractor states of the network in this high dimensional space. Attractors that have the same form (figure 1.4) can be connected to form a line of attractors (called "line attractor") depicted by the red line. A, B and C cartoon the attractors shown in figure 1.4.

Despite its near optimality, there are at least three disadvantages associated with the attractor-based coding in estimation. Firstly, it should be noted that the attractor-based method performs near optimally in the absence of dynamical noise (noise that corrupts the neural activity during the evolution of the network). In its presence, the performance gets impaired as the state of the network begins to randomly walk on the one dimensional line attractor. Given that the information about the orientation is coded by the location of the converged state of the network on the line attractor, such noise driven random movements result in the loss of coded information.

Learning is another drawback for the attractor-based coding; none of the papers that have reported near optimality of the attractor-based coding have shown learning the corresponding recurrent weights (Pouget et al., 1998; Deneve et al., 1999, 2001). In all cases, the weights have been handcrafted. In fact, learning seems a big challenge for the attractor-based methods. Even powerful methods like Backpropagation Through Time (BPTT) (Rumelhart et al., 1986a; Rumelhart et al., 1986b) find it difficult to learn the weights for the attractor-based coding. Speed of convergence seems to play an important role here. We are not aware of any easy way to control and perhaps to increase it. As a result, it might take several iterations for the network to converge which increases the cost of learning exponentially. Besides, for the above estimation task, the Backprop method should improve the performance while making one dominant symmetric line attractor by the end of learning; again, this doesn't seem to be easy in the presence of the nonlinearity of the network.

Finally, being on the attractor state implies that neurons fire with constant firing rate over time. However, persistent activity of sensory neurons in the cortex has rarely been observed *in vivo*; rather, their activity has been reported as temporally sparse,(Reinagel, 2001) and, at least in primary visual cortex, has been related to the non-classical receptive field of neurons (Vinje and Gallant, 2000). It has also been proposed as a neural mechanism that seeks to minimize the metabolic cost of the spiking activity (Attwell and Laughlin, 2001). Persistent activity has been reported in head direction cells (Zhang, 1996), place cells (O'Keefe and Dostrovsky, 1971) and grid cells (Hafting et al., 2005). However, neurons in higher cortical areas that have been reported to fire persistently during the delay period in working memory tasks show systematic changes in their firing rate during this period (Brody et al., 2003); (but see (Zhang, 1996; O'Keefe and Dostrovsky, 1971; Hafting et al., 2005))) it has also been shown that the increase in activity during the delay period can be fairly small (Naya et al., 1996; Shafi et al., 2007) with some of the neurons becoming silent for several seconds before getting active again (Rainer and Miller, 2002). Such fluctuations are obviously not consistent with attractor-based theories of working memory (Amit, 1995; Camperi and Wang, 1997; Compte et al., 2000; Wang, 2001; Brunel, 2003; Renart et al., 2003; Machens et al., 2005).

In sum, the attractor-based coding faces several problems despite its near optimal performance in a class of estimation tasks. Besides, to the best of our knowledge, this method has not been shown to perform discrimination tasks optimally; attractor-based solution to this type of problems requires at least two line attractors with the readout deciding based on the line attractor to which the network converges. Optimality of such a solution has not been demonstrated for tasks involving discrimination in the presence of invariance.

1.4 Feedforward models for invariant recognition

1.4.1 Neocognitron

In the previous sections we showed that a two-layer threshold-linear feedforward network, despite solving a class of local discrimination tasks optimally, fails to solve discrimination tasks in the presence of invariance in an optimal way. We also showed that there are local discrimination tasks that can not be addressed by a two-layer linear threshold feedforward network. These results indicate that an optimal solution to the invariance problem by the feedforward architecture requires more than two layers of nonlinear elements (neurons). Neocognitron was the first feedforward architecture that was primarily proposed to address shift invariance problem (Fukushima, 1980; Riesenhuber and Poggio, 1999, 2000; Serre et al., 2007). It consists of two types of alternating layers referred to as “S” and “C” layers (figure 1.6A). Neurons in the “S” layer are linear and are supposed to extract features from their input activity. In contrast, neurons in the “C” layer are nonlinear and are supposed to be invariant to small perturbations (for example small shifts in location) introduced to features extracted in the immediately afferent “S” layer (figure 1.6B).

This architecture is mainly inspired by response properties and classical receptive field structure of Simple and Complex cells in the primary visual cortex (indeed, “S” and “C” stand for “Simple” and “Complex”, respectively). While simple cells respond to gratings with certain orientation and phase, complex cells’ response is phase invariant and is only orientation selective. Besides, simple cells have smaller receptive fields than complex cells. Hubel and Wiesel proposed a simple explanation for phase invariance of the complex cells suggesting that a typical complex cell gets its input from simple cells that have the same location and orientation preference but are selective to a range of different phases which results in the response of the complex cell to be phase invariant. This can be achieved through simple nonlinear operations like “Max operation” which selects the maximum activity over all the afferents.

In Neocognitron “C” and “S” layers are alternated and the neuronal responses are supposed to become more invariant to stimulus features as one proceeds to higher layers in the network. C-layer is believed to play a key role in the emergence of invariance in these networks, by a mechanism similar to that suggested by Hubel and Wiesel for phase invariance of complex cells. However, the invariance introduced by neurons at individual “C” layers needs to be small to ensure specificity is preserved (figure 1.6C). During learning, only the feedforward weights from “C” to “S” layers are trained; here, learning is in a winner-take-all fashion; only the synapses to a “S” neuron that has the highest activity are modified and the modification is proportional to their presynaptic activity. Neocognitron has also been shown to reproduce human-like behavior in object recognition, in particular rapid categorization (Serre et al., 2007;

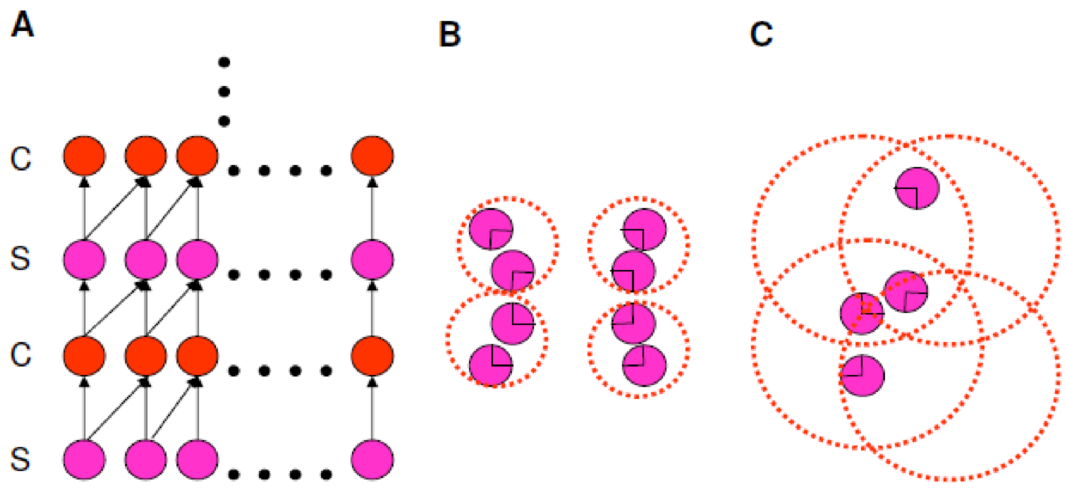

Figure 1.6- Neocognitron. **A**) A feedforward network of alternating C and S layers (see text for the definition of C and S layers). **B**) The receptive fields of the neurons of the S layer (small circles) and the efferent C layer (large dotted red circle) are shown schematically. For each S neuron, the feature to which the neuron is selective to is shown inside its receptive field. **C**) illustrates why the invariance introduced by each neuron at C layer needs to be small. If the task is to detect a rectangle, then the receptive field structure in (B) is more selective to a rectangular pattern than the receptive field structure in (C).

Pinto et al., 2008). However, it has been argued that the data sets on which this type of models have been tested consist of stimuli with limited variations and that increasing the variability impairs their performance level (Pinto et al., 2008).

1.4.2 Dynamical Routing

Dynamical routing is another proposed feedforward solution to the problem of invariant object recognition. (Anderson and Van Essen, 1987; Van Essen et al., 1992; Olshausen et al., 1993). The method which is also referred to as "shifter circuit" is based on template matching. The idea is to map the stimulus, where ever it is on the retina into a reference circuit that lies at several layers higher in the hierarchy. This requires a dynamic routing mechanism to dynamically map the incoming stimuli into the reference circuit.

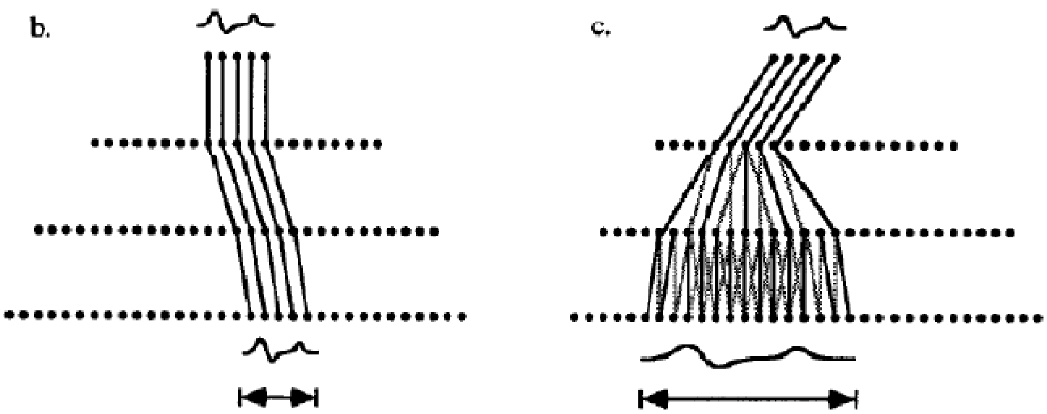

Figure 1.7- Dynamical routing. Left: The input is presented to the bottom layer and has to be routed to the reference circuit which is supposed to be at the top of the third layer. This kind of routing allows shift invariance. Right: The same mechanism as shown in left, just that the size of the stimulus is larger than the size of the reference circuit (the number of neurons representing the stimulus is more than the number of neurons in the reference circuit). This kind of routing results in size invariance. Figure adapted from (Olshausen et al., 1993).

Denoting the effective weight between input neuron i to reference neuron j by $w(i, j)$, then the routing mechanism is supposed to dynamically modify the weights $w(i, j)$ in order to appropriately map the stimulus into the reference circuit. Figure

1.7 Cartoons the idea for both shift and size invariance. In figure 1.7A the stimulus within the window of attention gets routed to the reference circuit through the connections marked. In this case, the size of the window of attention and the size of the reference circuit are the same; this is an example of how this approach solves shift invariance. Figure 1.7B shows its solution to the size invariance problem. The size of the stimulus here is larger than the size of the reference circuit which results in the effective connectivity to get dynamically modified in order to fit the reference circuit. This results in information loss as the resolution decreases but the image does not get deformed. However, appropriate changes in the effective weights require coordinated changes in the synapses of several layers in the brain; it is not clear how this would be achieved, both theoretically and experimentally.

1.5 Perceptual learning and hyperacuity

Perceptual learning is commonly referred to a class of psychophysical tasks that involve discriminating simple stimulus attributes (such as orientation, contrast, etc in vision) in which subjects' performance improves with practice (McKee and Westheimer, 1978; Fiorentini and Berardi, 1980; Karni and Sagi, 1991; Saarinen and Levi, 1995; Crist et al., 1997). However, these improvements have been shown to be highly specific to stimulus features (Ahissar and Hochstein, 1997); for instance, once the task is learned in one location, this learning does not transfer to other locations; in bisection task, in which three small parallel lines are presented and the subjects should decide to which outer bar the middle one is closer, it is shown that training improves the performance at training site and the degree to which it transfers to nearby locations decreases by the distance from the training site. It has also been shown that learning does not transfer to other orientations within the training location (Watt and Campbell, 1985). Interestingly, learning does not transfer from one eye to the other (Fahle et al., 1995). All these results seem to suggest that low level sensory areas like primary visual cortex - where neurons show selectivity to stimulus features like location, orientation, etc and have small receptive fields - might play an important role in perceptual learning; in other words, synaptic modifications, that are associated with performance improvement, are mainly at (low level) sensory areas. This has also been supported through neurophysiologic experiments; Schoups et al (Schoups et al., 2001) provided evidence that the orientation tuning curve of a subset of neurons in the primary visual cortex is modified by practice in a local orientation discrimination task. Interestingly they showed that this modification increased the slope of the tuning curves of these cells at the training site which in turn increased the Fisher Information.

However, the issue of the exact location and nature of the neural correlates of perceptual learning has been highly debated. It has been argued that learning is the result of modifications in higher cortical areas (Mollon and Danilova, 1996; Dosher and Lu, 1998; Lu and Dosher, 2004) and some experiments have reported modifications in extrastriate cortex after learning (Yang and Maunsell, 2004; Raiguel et al., 2006). Interestingly, trial by trial feedback is not necessary for performance improvement; performance improves even without any feedback about the correctness of the subjects' responses although this results in slower learning rate (Karni and Sagi, 1993). While early models of perceptual learning relied on feedback to improve their performance, later models (Weiss et al., 1993; Adini et al., 2002; Tsodyks et al., 2004) learned the task without feedback. In this framework, experiments seem to suggest that these modifications improve performance by increasing the strength of the internal signal rather than suppressing the internal noise (Gold et al., 1999). The high level modification framework, however, does not explain the specificities observed in perceptual learning tasks. A recent work (Xiao et al., 2008) has provided evidence that training can be transferred to other locations if those locations are primed during training at the target location. While such interesting results need more control experiments, it seems that, overall, these results support the role of both high and low level cortical areas in perceptual learning.

Hyperacuity (Westheimer and McKee, 1977a; Westheimer, 1981) is a term used for a class of visual acuity tasks in which human subjects demonstrate an impressive

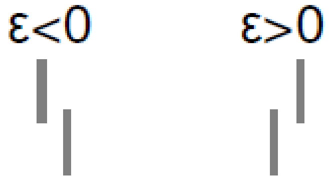

Figure 1.8- Vernier hyperacuity task. Two parallel lines are presented simultaneously and the subjects should decide if the bar at the top is on the left ($\varepsilon < 0$) or the right ($\varepsilon > 0$) side of the bar at the bottom.

performance level in resolving fine spatial details. Hyperacuity tasks have been subject to perceptual learning and have been shown to be specific to low level stimulus features similar to other perceptual learning tasks discussed above. An example is the Vernier task (figure 1.8) in which the subjects should report if the bar at the bottom is to the right or the left of the bar at the top. While the diameter of the photoreceptors in the fovea - where we have the highest spatial sensitivity across the retina - is in the range of 30 to 60 arc seconds, subjects have been reported to maintain visual acuity of about 5 arc seconds in a range of hyperacuity tasks including Vernier. Furthermore, neurons have been found in low level cortical areas like primary visual cortex whose sensitivity is below the spacing of the photoreceptors (Parker and Hawken, 1985; Swindale and Cynader, 1986); However, this isn’t surprising as Snipe and Koenderink (Snippe and Koenderink, 1992) were the first to point out that the behavioral threshold should be set by the information coded in the population level rather than by single cells. Encoding information redundantly through different neural pathways increases the signal to noise ratio and therefore improves visual acuity.

1.6 Network models of perceptual learning

Several network models have been suggested for perceptual learning. These models have been developed for different tasks ranging from Vernier hyperacuity task, orientation discrimination task, and contrast discrimination task to bisection hyperacuity task. In this section, we review the first three. We review the bisection task at the beginning of the next chapter.

1.6.1 Vernier Hyperacuity

Poggio and colleagues (Poggio and Girosi, 1990; Poggio et al., 1992) suggested a feedforward model of basis functions and showed how this model could reproduce some of the findings in the psychophysics of Hyperacuity tasks, in particular perceptual learning in Hyperacuity. The model, which is supposed to be task specific, is based on function approximation, where the function maps each stimulus to one class or the other (think of vernier acuity task and the offset which is either to the left or to the right). The network classifies the stimuli with a set of radial basis functions (RBF; figure 1.9). RBFs are defined as follows:

$$B(\mathbf{x},\mathbf{t}_i) = G\left((\mathbf{x}-\mathbf{t}_i)^T \mathbf{W}^T \mathbf{W}(\mathbf{x}-\mathbf{t}_i)\right) \tag{1.22}$$

where $\mathbf{x}$ is the new example shown to the network, $\mathbf{t}_i$ is the centre of the basis function and $\mathbf{W}$ is a weight matrix that is modified in order to classify the new examples correctly. G is a Gaussian. The readout is based on the sign of:

$$f(\mathbf{x}) = \sum_i \alpha_i B(\mathbf{x},\mathbf{t}_i) \tag{1.23}$$

Besides the weight matrix, α_i and $\mathbf{t}_i$ are also learned. When modifications fail to accommodate a new example, the new example becomes the centre of a new basis function which gets added to the above sum. The authors demonstrated that this model learns the task and its performance improves with practice and is specific to visual

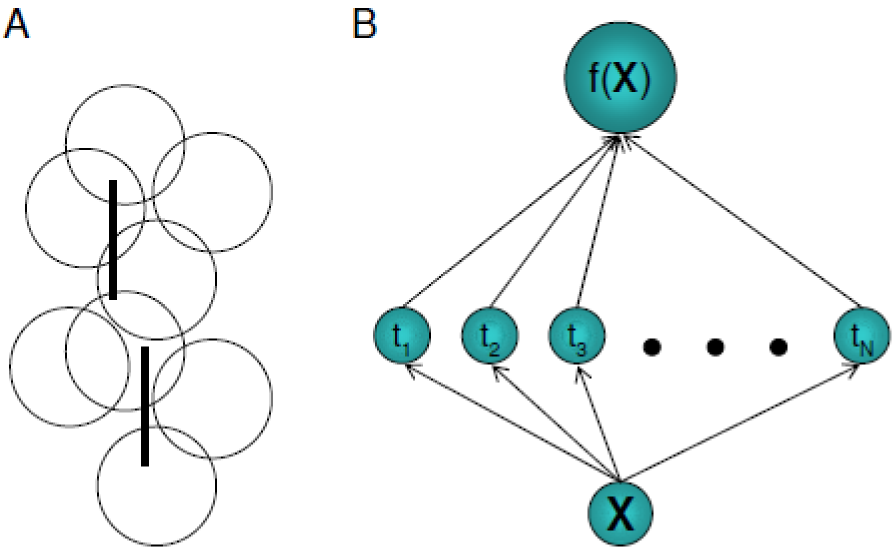

Figure 1.9 HyperBF network. **A**) Circles represent the receptive fields of the RBFs. The two parallel lines make the Vernier stimulus together. **B**) The input to the network $\mathbf{X}$ is convolved with a Gaussian (RBF) according to equation 1.22 and the results are linearly added to produce the output $f(\mathbf{x})$, (equation 1.23).

attributes of the stimuli; for example in the case of Vernier task, it does not get transferred to other orientations. However, they didn't do an ideal observer analysis to compare against the performance of the network. Besides, their feedforward model assumes no interaction between the neurons, while neurons in the cortex are highly interconnected. Their model requires feedback for training while as we mentioned above, it is shown that feedback is not necessary for training.

1.6.2 Contrast discrimination

Adini et al's network model of perceptual learning (Adini et al., 2002) was developed in the context of contrast discrimination task and was based on previous reports indicating that performance saturated after about 500 trials (that is, during fast learning processes (Karni and Sagi, 1993; Fahle et al., 1995)) and did not improve any more

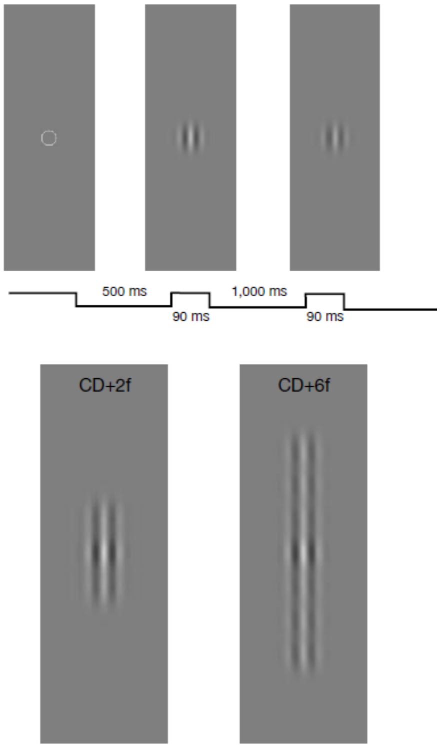

Figure 1.10 Contrast discrimination task. Top: Two gratings are presented successively and the subjects should decide which one has a higher contrast. Bottom: Contrast discrimination task in the presence of laterally placed gratings. Modifying the context in this way enables learning and results in performance improvement. Figure adapted from (Adini et al., 2002).

despite having several sessions of practice (Dorais and Sagi, 1997). Interestingly when subjects practiced the same task in a slightly modified context (figure 1.10), performance started to improve again and saturated at a higher level (Adini et al., 2002). To account for this context-dependent learning effect, they assumed that the task is mediated by a local network of inhibitory and excitatory neurons in the primary visual cortex. Assuming that the ratio between inhibition and excitation to this network is fixed (Adini et al., 1997), using linear approximation, they showed that the ratio of average firing rates of inhibitory and excitatory populations is independent of the contrast. Since the learning rule they used depended on this ratio, they concluded that a simple way to modify the weights is to introduce new stimuli to the visual context that change this ratio and lead to further modifications in synaptic efficacies, compatible with their observation. In their model learning is only activity dependent and does not rely on feedback which is consistent with the observation that feedback is not necessary for learning. However, their model does not address the specificities observed in perceptual learning tasks.

1.6.3 Orientation discrimination

Orientation discrimination has also been shown to improve with practice (Regan and Beverley, 1985; Gilbert, 1994; Schoups et al., 1995). Furthermore, Schoups et al. (Schoups et al., 2001) reported modifications in the tuning curves of the orientation selective neurons (whose receptive field overlaps with the training site) in the primary visual cortex as a result of training, suggesting a role for sensory areas in learning. In particular, they showed that the slope of the tuning curves around the training site increased after training (sharpening of orientation tuning curves). Teich and Qian (Teich and Qian, 2003) studied this in the context of recurrent network models of primary visual cortex. Since learning the recurrent weights has been proved difficult (Tsodyks and Gilbert, 2004), they modeled the end result of learning in the recurrent network and showed that reducing the strength of the excitatory synaptic connections to neurons around the trained orientation reproduces the tuning curve changes they observed in the primary visual cortex and improves the performance level of the network. Since adaptation is often regarded as a short term learning, they also showed that decreasing both excitatory and inhibitory synaptic weights to neurons around the adaptation site shifted the tuning curves away from the adaptation site and made them broader in line with the short-term adaptation-induced modifications observed in V1 tuning curves (Dragoi et al., 2000). However, they didn’t come up with a learning rule that reproduced the synaptic modifications they suggested to account for the observed changes in tuning curves.

Chapter 2

Change-based coding

2.1 Introduction

In the previous chapter we discussed optimal decoders performing discrimination on noisy population activity. We showed that there are specific noise models for which a two-layer threshold-linear feedforward network implements the optimal readout. However, we provided an example local discrimination task for which this architecture failed to extract the Fisher information. We also showed that this simple architecture is not optimal for tasks that involve discrimination in the presence of invariance. This raises the question of how to modify this architecture, including both the network and the readout, to improve its performance to the best possible level in tasks that involve discrimination in the presence of invariance.

To address this problem we study the hyperacuity bisection task as a paradigmatic example. This task has previously been studied exactly in terms of the invariance issues (Zhaoping et al., 2003) but within the attractor networks framework (where the end result of the network's computation is represented by the attractor states). The bisection task is a psychophysical problem used to measure perceptual learning (Crist et al., 1997). In it, subjects see three, small, nearly evenly-spaced bars, and have to decide whether the middle bar is closer to the one on the right or the one on the left (see Figure 2.1A). This decision should be invariant to the overall location on the retina of the stimulus array (represented by y) – for even if the experimenter always presents it at the same point on the screen, the observers' involuntary micro-saccades imply that it will always shift around and will be at slightly different locations at different trials. In other words, y changes on trial by trial basis, however, it is a nuisance variable that carries no task relevant information. Before we proceed to introduce our method, it is useful to study this problem within the attractor based paradigm. Under the standard line attractor view, this invariance problem could be solved by embedding two different surface attractors into the recurrent connections, one each for the central bar being left and right of the middle of the outer bars, and making a decision based on the one of these two at which the terminal state of the network resides (figure 2.2). While this is a possible solution, to the best of our knowledge, it has never been suggested to solve the problem of discrimination in the presence of invariance. Zhaoping et al's (2003) solution is perhaps the closest solution to this approach. However, its success was limited and we briefly review it here. They tested the performance of a two-layer threshold-linear feedforward network - that was optimal for local discrimination task (i.e. task in which $y = 0$ in all trials) - in a task that involved discrimination in the presence of invariance in which y was drawn randomly from $[-l \;\; l]$ at each trial (rather than being 0 at all trials) where l is a positive real number. The two-layer threshold-linear feedforward network's performance quickly reached chance level as they increased l.

To improve the performance, they introduced a recurrent network between the input and the threshold-linear feedforward readout (the neural activity elicited by the stimulus is referred to as "input" here). The recurrent network was supposed to take the input from the location (location is represented by y) at which the stimulus was presented and bring it back to a default location ($y = 0$) where the threshold-linear feedforward readout was adapted to. In other words, the recurrent network was supposed to preprocess the information for the threshold-linear readout by changing the location of the input pattern appropriately. For an optimal performance, it was vital for the recurrent network to shift the location of the input such that the information loss was minimized (here, "information" refers to the location of the middle bar relative to the middle of the two outer bars). While the combination of the recurrent network and the threshold-linear readout performed better than a quadratic readout – that was derived based on the second order Taylor approximation to the optimal readout - its performance level remained below optimal. They interpreted their recurrent solution in the context of the line attractor networks (Zhaoping et al, personal communication): the recurrent network creates two destabilized line attractors (one for left choice and one for right choice) in the form of point attractors; the threshold linear readout decodes the attractor to which the network is converged. However, as we mentioned, the performance of this network remained far from optimal.

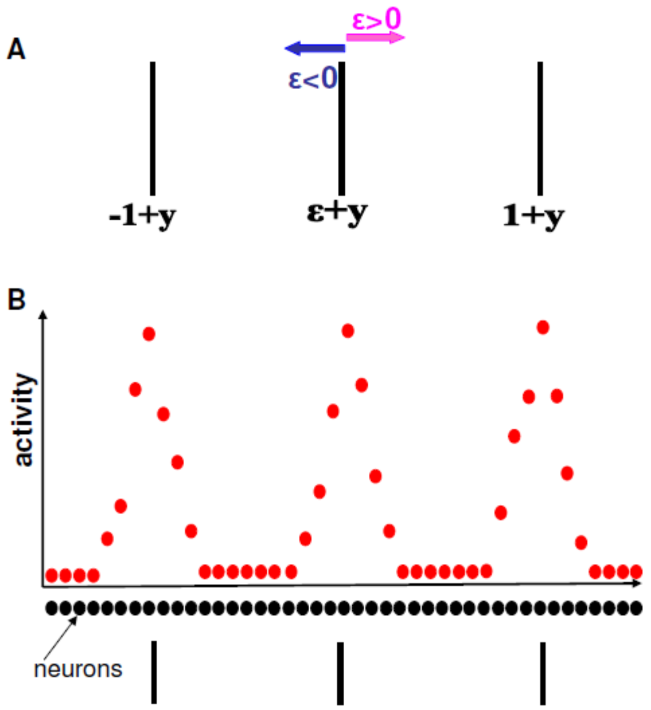

Figure 2.1 Bisection task and input representation **A**) Bisection task**.** Three parallel bars are presented, and subjects must decide whether the middle bar is closer to the left or right end bars. The distance between the two outer bars is always fixed at 2 units of distance. y represents the position of the whole array which changes from trial to trial because of eye movements between trials. ε represents the deviation of the middle bar relative to the middle of the two outer bars, so the task demands assessing whether $\varepsilon \leq 0$. **B**) The three bars mentioned in (A) elicit three (Poisson-distributed) noisy bumps of activity in a layer of units. The readout (potentially a member of another layer of units) reports on the sign of ε based on these activities.

In the previous chapter, we discussed the problems that are faced by the attractor-based estimation. In fact, some of those problems, such as learning the optimal weight matrix, also hold for the attractor-based discrimination in the presence of invariance. For instance, Zhaoping et al. (Zhaoping et al., 2003) handcrafted the recurrent weight they used for discrimination. In the absence of a powerful learning rule for training attractor-based networks, attractor-based methods seem to offer limited solutions to both estimation and discrimination tasks.

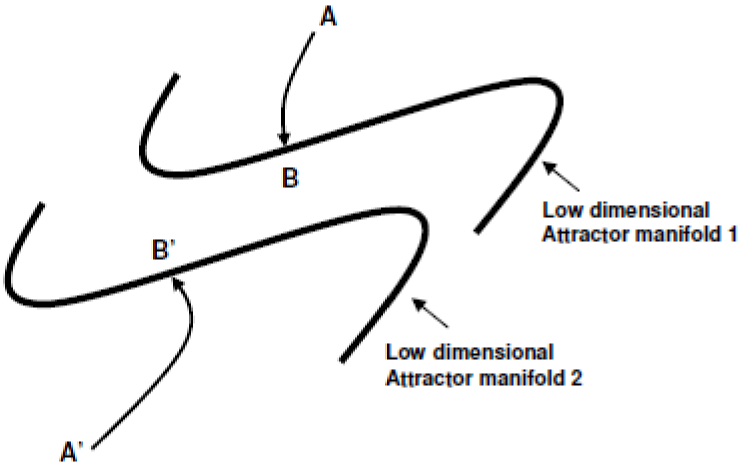

Figure 2.2 A potential solution to bisection task. There are two low dimensional attractors for two possibilities and each line is parameterized by the dimension to which the computation is supposed to be invariant.

In this chapter we introduce a new coding method that solves the problem of discrimination optimally in the presence of invariance. As we show in chapter 5, it also allows easy learning of the recurrent weights. We first explain the model, the task and the coding method. We then present the results.

2.2 Bisection hyperacuity task

2.2.1 Modeling Bisection Task

Figure (2.1A) illustrates the bisection task. It shows the three bars, at positions $-1+y$, $\varepsilon+y$ and $1+y$. The task is to determine the sign of ε, which indicates to which of the two outer bars, the inner bar is closer. As we mentioned above, the invariance problem comes in the form of the nuisance variable y, which quantifies the position of the overall array in the input, and whose value is irrelevant to the task.

Figure 2.1B shows the abstraction we employed for the coding of the input, which is based on that in Zhaoping et al. (2003). Here, there are N = 321 input neurons (for simplicity, we only consider units with vertical tuning which is the orientation of the bar). Unit i has preferred topographic position x_i along the line defined by the task. The distance between two neighboring units, $x_{i+1} - x_i$ is 0.05. Each single bar is assumed to drive the input units according to a Gaussian tuning curve; the mean feedforward input to unit i arising from the three bars is

$$\overline{a}_i = \lambda e^{-\frac{((-1+y)-x_i)^2}{2\sigma^2}} + \lambda e^{-\frac{((\varepsilon+y)-x_i)^2}{2\sigma^2}} + \lambda e^{-\frac{((1+y)-x_i)^2}{2\sigma^2}} \tag{2.1}$$

where σ, the width of tuning curve, is 0.1 and λ=20 Hz is the maximum height of each bump. We assume that the distance between the neighboring bars is much greater than the tuning width of the units and as a result each active unit is activated non-trivially by only one of the bars. The actual feedforward drive a_i associated with unit i follows a Poisson distribution with mean value $\overline{a}_i$

$$P(a_i \mid \overline{a}_i) = \frac{e^{-\overline{a}_i}}{a_i!} \overline{a}_i^{a_i} \tag{2.2}$$

with all the units being independent.

2.2.2 Dynamics of the recurrent network

We study this problem in the context of nonlinear rate-based units. Model neurons in the network follow rate-based dynamics for their firing rates; defining the vector of membrane potentials by $\mathbf{u}(t) = (u_1(t), u_2(t), \ldots, u_N(t))$ the dynamics of the network is governed by the following equation:

$$\tau \frac{du_i}{dt} = -u_i + C + \sum_{j=1}^{N} W_{ij} f(u_j) \tag{2.3}$$

$$u_i(0) = \eta a_i \qquad 1 \le i \le N$$

where the nonlinear activation function $f(.)$ is defined as $f(x) = x$ if $x > 0$ and $f(x) = 0$ if $x \le 0$; $\tau = 20ms$ is the time constant and is the same for all units, $\mathbf{W} = \{W_{ij}\}$ is a symmetric, translation invariant recurrent weight matrix (that is,

$W_{ij} = W_{|i-j|}$ for all i, j). We assume that the boundaries are very far from where the computations happen and therefore with good approximation we can assume that the line of neurons is infinite. $\mathbf{C} = (C, C, \ldots, C)$ is a uniform non-informative input vector to the network (it is non-informative because that's the same for all neurons) which maintains the activity of the net, and $\mathbf{a} = (a_1, a_2, \ldots, a_N)$ is the stimulus driven (informative) input activity (that initializes the differential equation). To summarize the parameters, we use σ=0.1, τ=20, λ=20 and $\frac{\eta}{C}$=7.5. Constant C is chosen to optimize the performance of the change-based readout. There is a broad range for the constant C that keeps the performance near optimal. The neurons are assumed to be initialized by upstream mechanisms to values reflecting $\mathbf{a}$, but then the activities evolve with only the uniform, uninformative, input $\mathbf{C}$.

2.3 Change-based Readout

Figure 2.3 cartoons our new method in the case that the network instantiated just a single line attractor. Note that the line attractor is parameterized by y (see appendix 2.B). In this case, it would be impossible to make any decision about the sign of ε based on the asymptotic activities because the stable pattern is completely invariant to ε. In other words, the initial pattern always converges to the same line attractor independent of the sign of ε and it would be impossible to disentangle the values of y and ε solely from the terminal position on the line.

Here, rather than trying to read out the final, converged, state of the network, we consider how a statistic of the activity ***changes*** over the evolution of the activity toward the attractor. The idea is that this *change* can be used to code information. For instance, for the bisection task, this requires designing a network (like the one shown in figure 2.3) that codes the information such that it shifts the overall activity towards right if $\varepsilon > 0$ and shifts it towards left if $\varepsilon < 0$. In particular this information can be encoded by the initial direction of the activity movement while the state of the network is far from the attractor. As we will see in the following chapters, this is very useful for learning as it allows methods like BackPropagation algorithm to easily learn the weight matrices associated with the optimal performance. In fact, all the weights that we handcrafted to solve this task by the change-based method turned out to code the information by the initial direction of the activity movement.

Coding task relevant information by the change of the early portion of the stimulus driven neural activity, introduces a new way of information processing by recurrent networks in which the temporal structure of the evolving neural activity plays a key role; this is in contrast to attractor-based models where the final converged state is supposed to capture all the task relevant information. Since the cortex is constantly bombarded with an endless stream of incoming stimuli, coding information by the early changes in the neural activity seems quite an effective way to make the speed of processing match the temporal statistics of the incoming information.

Here, we consider the population Center of Mass (CoM) as the statistic of the neural activity that encodes information by its evolution. The Center of Mass (CoM) of the neural activity is defined as follows:

$$\bar{\mu}(t) = \frac{\sum_{i=1}^{N} x_i u_i(t)}{\sum_{i=1}^{N} u_i(t)} \tag{2.4}$$

And the decision is based on whether the centre of mass increases or decreases over time. More formally, the sign of ε is estimated from $\bar{\mu}(t)$ at two different times $t = t_1$ and $t = t_2$ as follows:

$$\bar{\mu}(t_2) - \bar{\mu}(t_1) > 0 \quad \Rightarrow \quad \varepsilon > 0 \tag{2.5}$$

$$\bar{\mu}(t_2) - \bar{\mu}(t_1) < 0 \quad \Rightarrow \quad \varepsilon < 0 \tag{2.6}$$

Since the recurrent weights are translation invariant, this decision is completely invariant to y. In other words, the change-based readout is an immediate solution to the invariance problem (see appendix 2.B). Next, we examined how well it solved the discrimination task. Center of Mass estimator is similar to Population vector estimator that has been shown to encode direction of arm movement (Georgopoulos et al., 1986; Pouget et al., 1998).

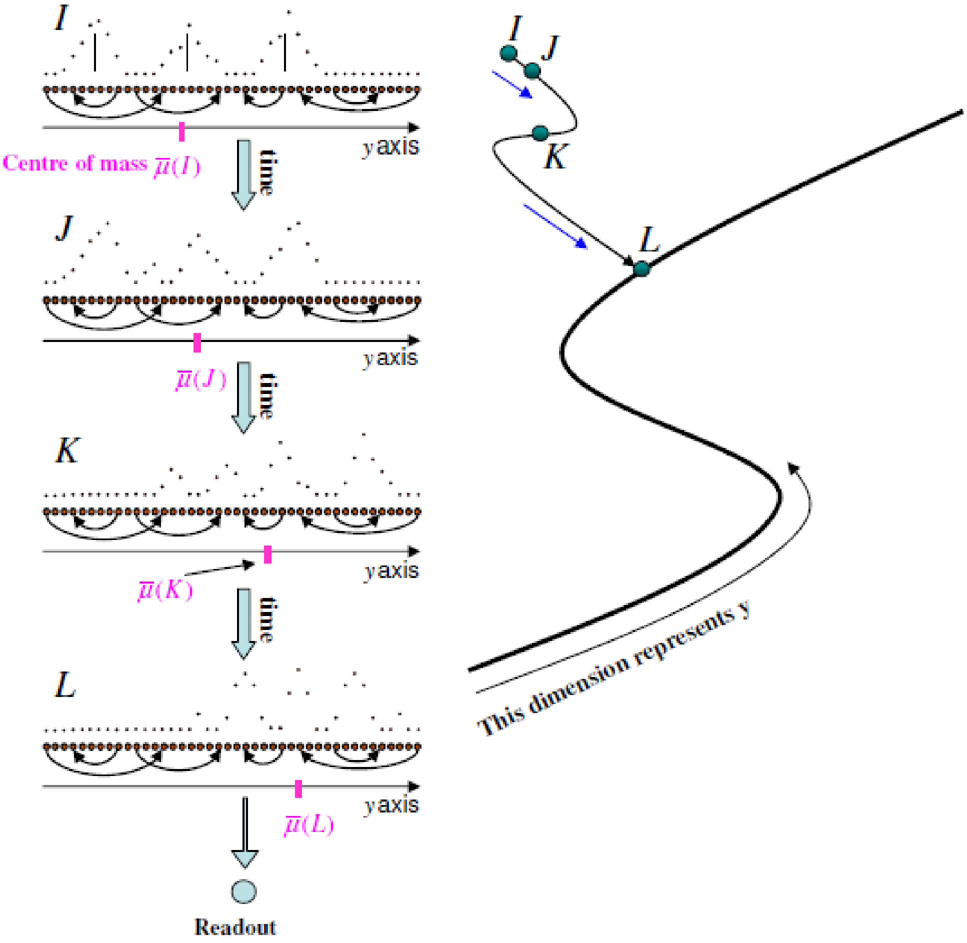

Figure 2.3. Recurrent processing and change based readout. Cartoons depicting the temporal readout mechanism for the case of the bisection task. In a recurrent paradigm, lateral connections between the units within a layer define a low dimensional point or line attractor structure in the high dimensional space of the activities of the units. As shown, attraction here is to a line of points defined across y for $\varepsilon = 0$. The initial activities (*I*) are mapped through attraction to activities *L* that represent a point on the line. Blue arrows show the direction of the evolution of the state of the network. Conventional recurrent readout is based on these activities. By contrast, the temporal readout is based on measuring the change as the activities relax towards the attractor in the centre of mass $\overline{\mu}(K) - \overline{\mu}(J)$ of the activities (magenta lines) between two times (*J* and *K*) close to the initial transient. This works as here even with only one line attractor that has no information about the sign of ε .

2.4 Results

2.4.1 Performance of the change-based readout

The curves in figure 2.4A show examples of the activity of the net 20 ms and 250 ms after stimulus onset, and the asymptotic activities of the network (the corresponding weight matrix is shown in figure 2.4b). These figures indicate that the state of the network is far from equilibrium 250 ms after the activity onset, and in fact, it only reaches asymptote at around 1000 ms. As can be seen from this figure, the converged state of the network is symmetric. In fact, this network has only one dominant line attractor whose form is the one that is shown in figure 2.4A. Therefore, this is an interesting case where the line attractor to which the network converges is not informative about the task relevant signal, i.e. $\mathcal{E}$, at all and therefore the converged state of the network can not be used for discrimination.

Also note that as the network evolves, new bumps appear which is the effect of the threshold non-linearity: as time evolves, new neurons that were not super-threshold at the beginning become active, while some of those that were active at the beginning become silent. The weight matrix in figure 2.4B was handcrafted, starting originally from the design in Zhaoping et al (2003). We took the following considerations into account in its design: We thought that a weight matrix with optimal performance required at least three excitatory bumps to allow crosstalk between units that were activated by different bars. Having five rather than three excitatory bumps allowed those bumps to be at positions that the Fisher information (which depends on the slope of the hill of activity) was highest. However, as we will see in the next chapter there are more fundamental reasons for the successful performance of the change-based method.

Figure 2.4C shows the discrimination performance of the single line attractor network (associated with the weights in figure 2.4B) based on comparisons between the activity at 20 ms and 150 ms (magenta), along with the ideal observer (black). Change-based readout performs very much better than the previous model (Li & Dayan, 2001), which is based on a static readout method that decides after the convergence of the network (red). As can be seen, the change-based method performs the task near optimally and at the same time is completely invariant to the location of the stimulus. Moreover, the performance of the attractor-based readout used in

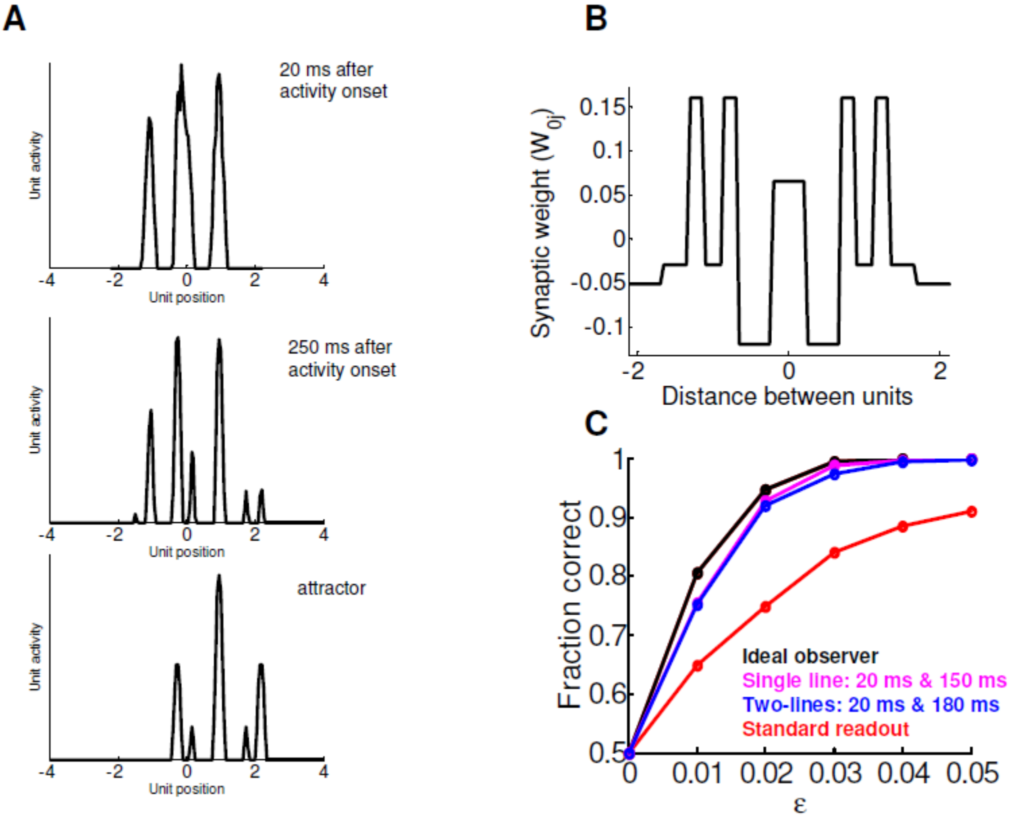

Figure 2.4. Performance of the network which has a single line attractor. **A**) Normalized neural activities on the line attractor (the associated weight matrix is shown in B which has only one line attractor), and an example of the activity of the network after 20 and 250 ms. **B**) Central row of the symmetric position invariant synaptic weight matrix ($W(i,j) = W(|i-j|)$) for the case of only one line attractor. For graphical convenience, we represent the weights (and associated attractors) as being continuous, even though only they are really discrete. **C**) The performance of the ideal observer (black), previous recurrent network (Zhaoping et al., 2003) for a range of y between -.6 and .6 (red) and the performance of the new line attractor network with only one line attractor (shown in (B)) with temporal readout (magenta). The performance of the latter is based on the comparing the centre of mass of the neural activity at 20 and 150 ms after the onset of neural activity. Also, blue shows the performance of the network with two line attractors (figure 2.7A) based on the comparing the centre of mass of the neural activity at 20 and 180 ms after the onset of neural activity. Performances are based on averaging 400 trials (the standard deviations of the estimates here and in subsequent graphs are less than 0.025). Note that the performance of the previous recurrent network (which is based on standard readout) decreases as the range of y increases while the performance of the change based readout is independent of the range of y.

(Zhaoping et al., 2003) goes to chance level as the range of variations of y increases. We derive the equations of the ideal observer for Poisson noise that is shown in figure 2.4C in the appendix 2.A of this chapter.

Next, we examined how the performance level depends on the delay between the first and the second measurement of the centre of mass. Figure 2.5A shows the discrimination performance of the single line attractor network - associated with weight matrix shown in figure 2.4B - based on comparisons between the activity at 1 ms and 20 ms (red); 70 ms (blue) along with the ideal observer (black). The network's

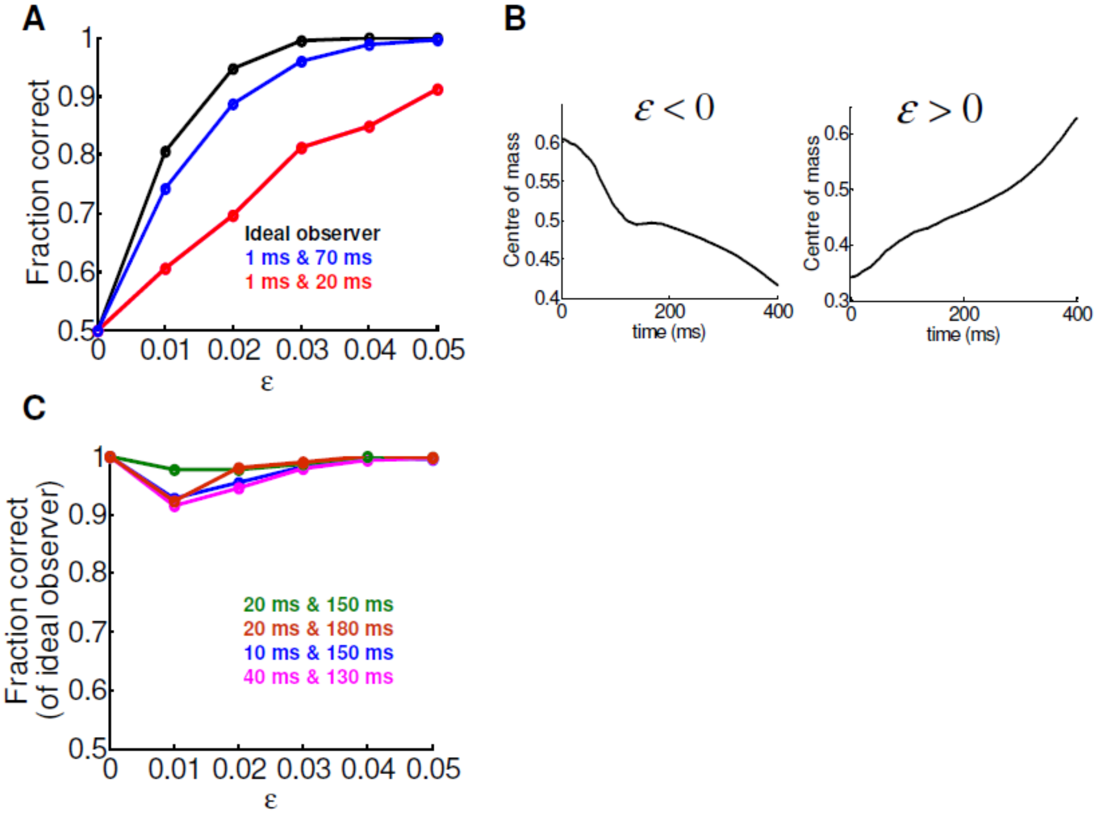

Figure 2.5. Performance improvement during evolution of neural activity of the network. **A**) Performance based on the movement of the centre of mass of the neural activity immediately after (1 ms after) the activity onset versus t ms after the onset of neural activity. $t = 20$ ms (red), $t = 70$ ms (blue) and ideal observer (black). **B**) Two example trajectories of the centre of mass for the network in figure 2.4B. Left: $\varepsilon < 0$. Right: $\varepsilon > 0$ **C**) Performance of the same network comparing centre of mass 20 ms and 150 ms (green), 20 and 180 ms (brown), 40 and 130 ms (magenta) and 10 and 150 ms (blue) after the neural activity onset as a fraction of that of ideal observer.

performance dramatically improves within 70 ms after the activity onset, and is quickly near optimal. Thus, in a network of units of 20 ms time constant, the direction that the centre of mass moves is highly informative about the sign of ε within 70 ms after activity onset.

We next measured the sensitivity of the performance level to the times at which the measurements are made. Figure 2.5C shows that the change-based readout performs well for different times of measurement of the CoM (performance shown as a proportion of that of the ideal observer in figure 2.5a). The main constraints are that the timing of the two measurements of the CoM be sufficiently far apart to have a measurable evolution between them, and, at least for the single attractor network, the measurement times be sufficiently close to the onset transient so that the critical information has not been suppressed through the process of attraction. These are key dimensions of robustness of the network's performance.

Figure 2.5B shows two examples of the evolution of the centre of mass for this network; change-based readout discriminates the sign of ε effectively while the recurrent network is far from the attractor. Evidence supporting the existence of such a distance from the attractor is that digital selection (Hahnloser et al., 2000) between the different units in the network is still occurring, i.e. the set of units that are supra-threshold is still changing.

2.4.2 Robustness to dynamical noise

So far, we have assumed deterministic, noise-free dynamics, and thus avoided the potential sensitivity of the change-based read-out to perturbative dynamics. Here, we introduce noise to the dynamics of the network to measure its sensitivity. To test the effects of noise, we considered a stochastic difference equation version of the dynamics, using Gaussian (rather than Poisson) noise, but making the variance of this noise equal to the mean firing rate.

We used a forward Euler discretization (with $\Delta t = .2ms$) of the resulting stochastic differential equation:

$$\tau \Delta u_i(t) = -u_i(t)\Delta t + C\Delta t + \sum_{j=1}^{N} W_{ij}\left(f(u_j(t))\Delta t + \sqrt{f(u_j(t))}z_j(t)\sqrt{\Delta t}\right) \qquad (2.7)$$

$$u_i(0) = \eta a_i \qquad 1 \le i \le N$$

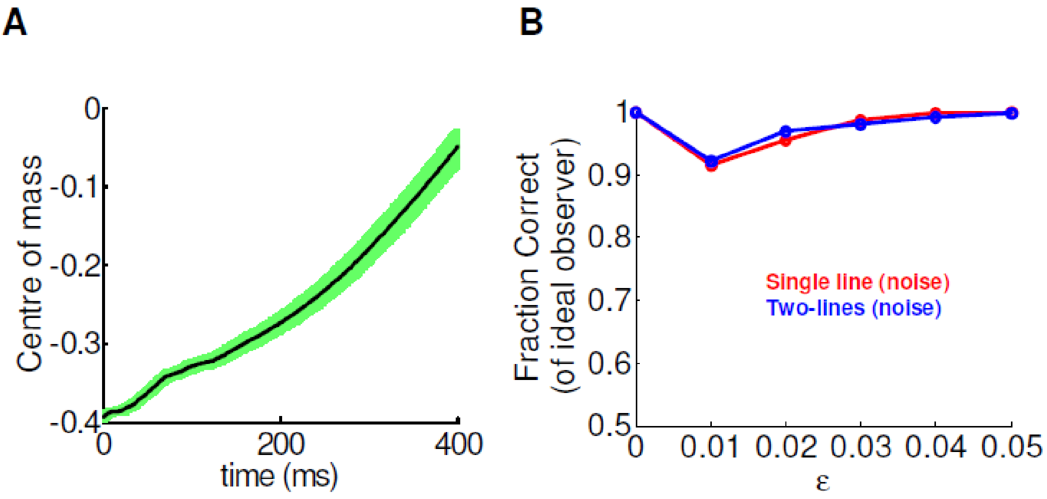

Figure 2.6. Effect of noise and persistent input during the dynamical evolution of the network. **A**) An example of the trajectory of the centre of mass with no dynamical noise (black) and the standard deviation (green) from this mean trajectory as a result of Gaussian mean dependent noise added to the output of active units during the evolution of the network for the same input as for the case with no dynamical noise. Standard deviations are computed based on 200 trials. **B**) Performance as a fraction of that of ideal observer of the one line attractor network (red) and two line attractor network (blue) which is shown in figure 2.7A in the presence of Gaussian mean dependent noise with variance equal to the mean. Performances are based on averaging 400 trials. Performances are based on averaging 400 trials.

where $z_i(t) \sim Normal(0,1)$ was an (independent) random number generated from a Gaussian distribution with mean zero and variance 1, so that the variance of the added noise was $f(u_i(t))$, the same as the mean output. Here, $\Delta u_i(t) = u_i(t+\Delta t) - u_i(t)$.

Figure 2.6A illustrates an example of the rather limited effect of noise on the early evolution of the centre of mass. Although the effect accumulates as time progresses, the net influence remains quite small over the first few hundred milliseconds of activity that are used by the change-based readout, and therefore its overall performance is not greatly affected (red curve in figure 2.6B). However, the noise effect gets stronger as the state of the network evolves towards the attractor, as is clear from figure (2.6A).

This is quite an important result since the noise added to the dynamics has the same mean and variance, i.e. the perturbations induced by dynamical noise are strong.

Note that figure 2.6B shows that the performance of the network in the presence of dynamical noise is nearly as well as the ideal observer that only takes the initial input noise in to account and is unaware of the dynamical noise.

Such a powerful noise would make serious problems for the attractor-based methods. It would result in a highly diffusive random walk of the neural activity on the line attractor which would “blur” the shape of the attractor and would make it difficult (if not impossible) and computationally more expensive for the (attractor-based) readout to identify it.

2.4.3 Robustness to variations in weight matrix and number of line attractors

Change-based readout not only is effective in the particular case of a single line attractor (figure 2.4) that most clearly demonstrates its unique computational properties but also works well in more general cases, including those involving multiple line attractors. As an example, figure 2.7 shows another example recurrent weight associated with two main mirror line attractor patterns (figure 2.7), for which change-based readout also performs near optimally (figure 2.2; blue). As for the case of the single line attractor, this network is also robust to high levels of noise (figure 2.6 blue).

Figure 2.8 shows the robustness in another way. Figure 2.8A depicts a further set of weights that perform the bisection task near optimally with and without noise; we were able to find many such. Figure 2.8B illustrates the fact that even some large changes to the weights can also preserve overall performance. In this case, some of the synapses (marked by the arrows) are modified by about 50% and yet the performance is near optimal. This robustness is important since synaptic weights in the brain have been reported to be highly unreliable. However, changes in recurrent weights can significantly impair the performance level of the attractor based method since even a small perturbation of the recurrent weight matrix might significantly perturb the form of the converged states (line attractors) and even their number

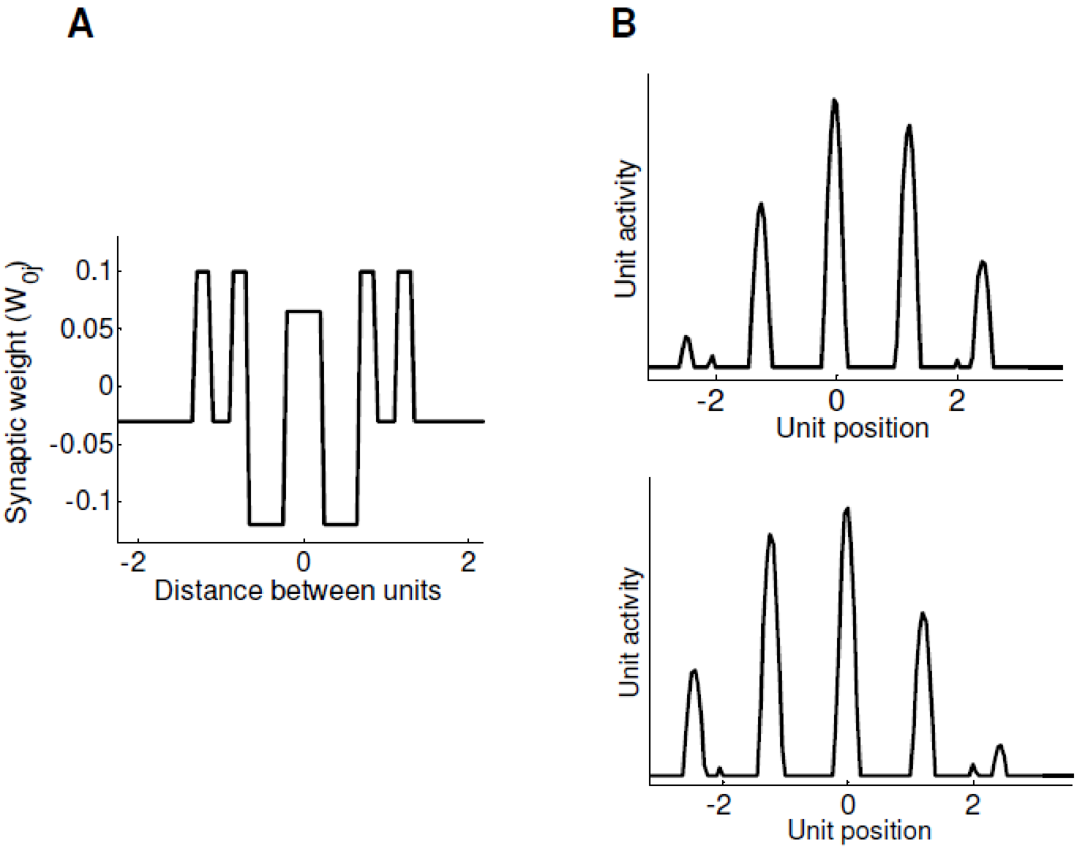

Figure 2.7. Recurrent network with two dominant line attractors. **A**) Central row of the symmetric position invariant recurrent weight matrix with two line attractors. **B**) The form of the two dominant line attractors of the network shown in (A).

(figure 2.9). Assuming that the perturbations are noise driven (rather than systematic) implies that the decoder would not recognize the new converged state (even in an entirely noiseless dynamics condition) since the modifications in its form are noise driven.

As we discussed in the section concerning dynamical noise, this would be even worse in the presence of such a noise as it results in an additional random walk on the line attractor. It is important to note that while these handcrafted weights are not robust to change in the spacing of the outer bars, it is possible to learn the network for different spacings using BPTT algorithm (we will show the results in the last chapter which is about learning the weights). Recently Herzog and his colleagues (Parkosadze et al., 2008) demonstrated that it is possible for human subjects to learn different spacings (for the Bisection task) although that requires much more trials (they report it to be around 18000 trials!).

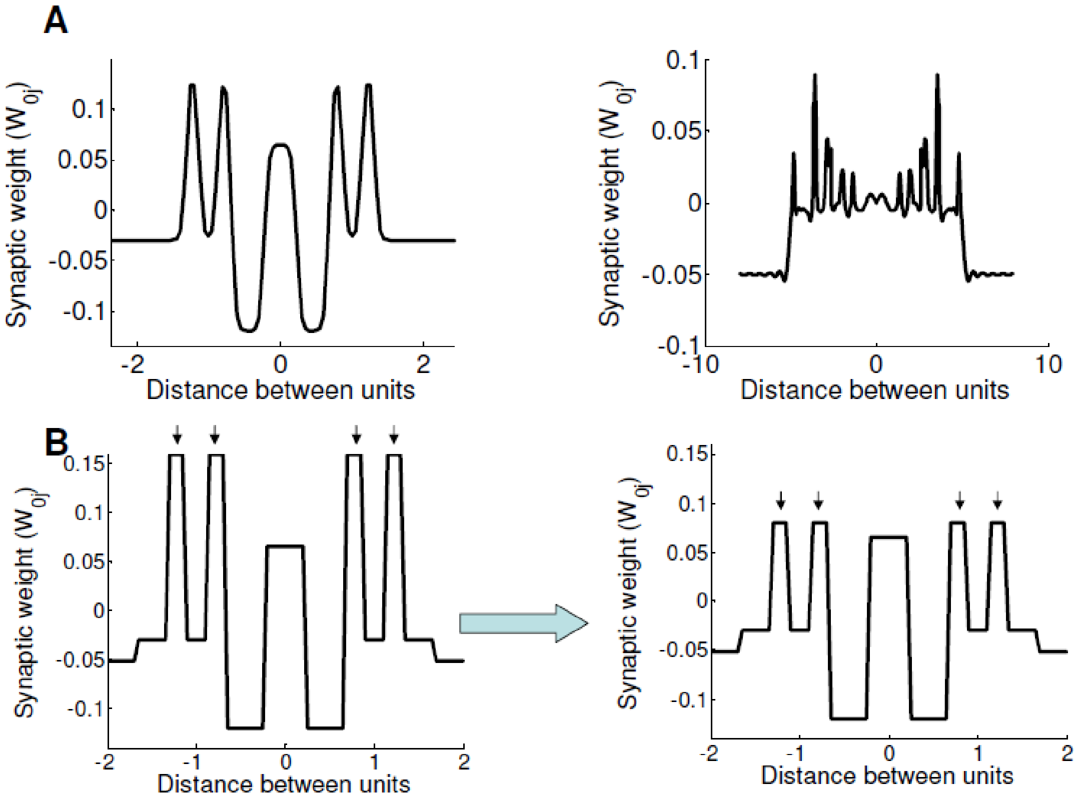

Figure 2.8. Robustness to variations in weight matrix. **A**) Two example weight matrices with different number of line attractors that perform the task near optimally. **B**) The two weight matrices shown are similar except that the long range excitation (indicated by the black arrows) of the right one is about 50% of that of the left one. Nevertheless, the performance levels of both recurrent weights are near optimal.

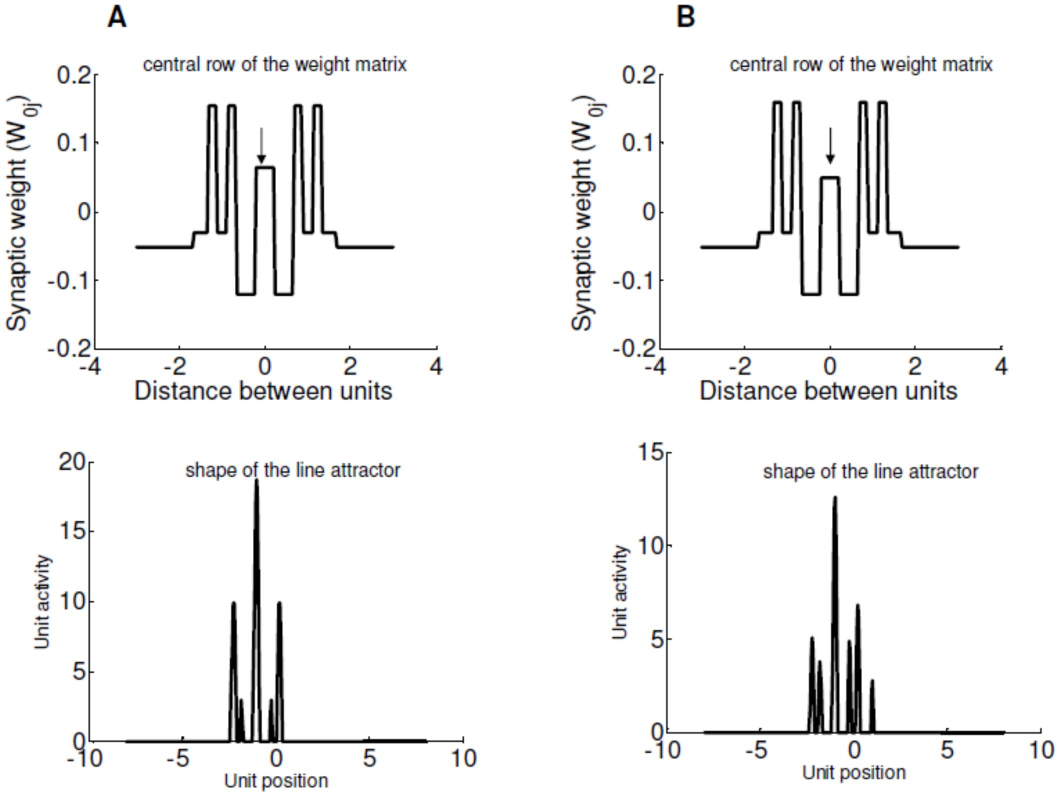

Figure 2.9 Small modifications of the weight matrix result in large changes in the shape of the attractors. **A**) A weight matrix with one dominant line attractor (top) along with the corresponding line attractor (bottom). **B**) The middle bump of the weight matrix shown in (A) is slightly modified; however, the form of the dominant attractor (bottom) is dramatically changed.

2.4.4 Density of the neurons

Here we show that increasing the number of neurons per unit distance improves the performance of the network along with the performance of the ideal observer. This is an important control to make sure that the performance is optimal and does not depend on the specific spacing of the neurons. We increased the number of neurons from 20 per unit distance (where the distance between nearby neurons is 0.05) to 100. This improves the performance of the ideal observer as more neurons respond to the stimulus. To build a new weight matrix for the dense network based on the previous one, we considered the smooth version of the weight matrix (shown in figure (2.4)) and we sampled more densely. As can be seen from figure 2.10, this also improves the performance of the network to near optimal level (red). This example indicates that the network with 20 neurons per unit distance has already reached its characteristic length scale and that the performance is optimal irrespective of the density of the neurons.

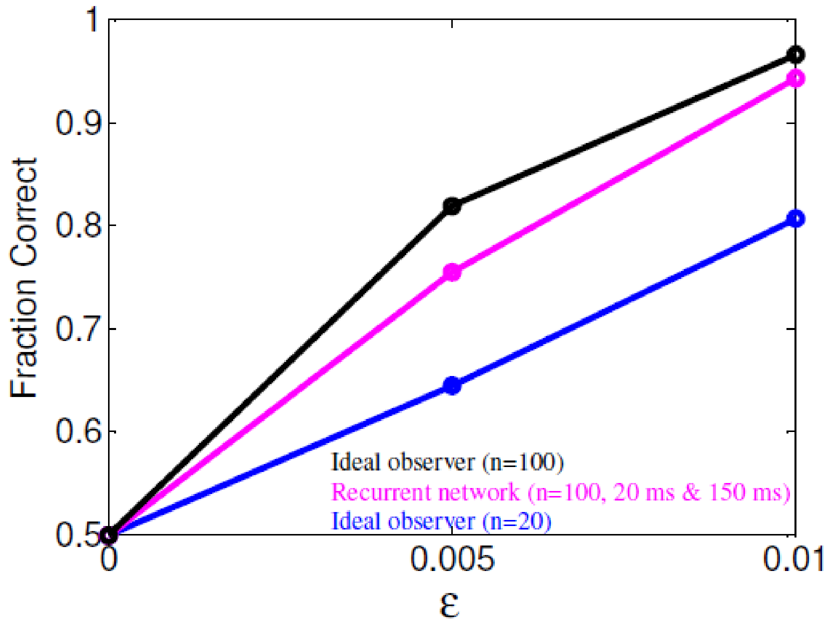

Figure 2.10 Density of the neurons. Blue curve shows the performance level of the ideal observer for the case in which there are 20 units (neurons) per unit distance. Black curve shows the performance level of the ideal observer for the case in which there are 100 units (neurons) per unit distance. Magenta shows the performance level of the temporal readout (comparing centre of mass 20 ms and 150 ms after activity onset) for the recurrent weight matrix shown in figure 2.4B for a network with 100 units (neurons) per unit distance.

2.5 Discussion

Nonlinear dynamical systems are powerful candidates as neural information processors. The conventional view about such networks is that they operate as forms of point, surface or line attractors, representing information according to which attractor, or where on the continuous attractor, their activity state converges (Zhang 1996; Seung 1996; Pouget et al. 1998; Deneve et al. 2001; Wu et al. 2001; Wu & Amari 2005; Wang 2001; Camperi & Wang 1997; Compte et al. 2000; Renart et al.

2003). Here, we studied a different way in which such networks might compute far from the attractor. Precedents for this view come from the locust antennal lobe (Mazor and Laurent 2005). In this system, it has been shown that signal amplification is optimal at a point at which the state of the projection cells that form the output from the lobe is far from a fixed point. Further, the Kenyon cells, that read out the state of the projection cells, are not responsive during the time the state of the projection cells is at a fixed point. This has been shown both when the stimulus has been presented briefly or persistently.

In our case, using the visual rather than the olfactory modality, we examined a specific computational advantage associated with off-attractor computations, namely the possibility of building invariance to a task-irrelevant dimension directly into the structure of the dynamics and the readout. Discrimination associated with one quantity in the face of invariance to others is a very common requirement for recognition; we used the bisection task as a paradigmatic example. In this task, we showed that by assessing how a statistic of the neural activity (the centre of mass of the underlying population code) changes over time during the evolution of the state of a translation invariant recurrent network, we could perform inferences about the structure of the input pattern (i.e. about the sign of ε) in a way that was invariant to the translation of the whole input array along the retina (i.e. the value of y).

We showed that the change-based readout led to near optimal behaviour which was robust against noise corrupting the ongoing activity of the network, noise in the synaptic weights (figure 2.8 and 2.9 are two example random changes to the weight. We have tested this for a broad range of changes and performance has remained near optimal; indeed that performance varied smoothly with the weights permitted gradient-based learning to work well), different times at which the centre of mass was compared, and also the persistence or otherwise of the external input. We argued that these characteristics weigh in favour of change-based readout in comparison with the classical view of attractor-based computations.

There are at least three such classical solutions for the bisection task against which change-based readout might be compared. One solution is to use two separate line attractor networks, one for $\varepsilon > 0$, a second for $\varepsilon < 0$, both parameterized continuously by the value of y. The trouble with this is that it is difficult to decide to which line attractor the network has converged in the face of varying y. In response to this, a second idea, employed in the work that inspired ours, was to destabilize the two line attractors so that they converge to single points each (Li & Dayan 2001; Zhaoping et al., 2003), effectively attached to one single, special, value of y. In turn, the trouble with this is that as the initial value of y strays further from this special value, the network performs less well. Indeed, the performance reported by Li et al. (Li and Dayan 2001; Zhaoping et al., 2003) is much further away from the ideal observer than is that of our change-based readout method.

A third idea [Latham & Beck, personal communication], based on the recent work on probabilistic population codes by Ma et al. and Beck et al. (Ma et al., 2006; Beck et al., 2007) would be to build a two dimensional, surface, attractor that represents the distribution of values of both y and ε at the attractor itself, and then marginalize across y to work out the likely value of ε. The operations to do this involve a form of divisive normalization. Under certain circumstances, at least for very small amounts of noise, this can be proved to work exactly as well as an ideal observer. This method is inferentially viable, but poses a harder learning problem, requires longer for inference for each case, cannot cope with persistent input, and is likely to be more sensitive to the effects of dynamical and weight-based noise.

In addition, an alternative idea to ours which is more closely associated with transience is to use the dynamics of an amorphously structured network to create a wealth of non-linear temporal and conjunctive filters over the input information (Maass et al., 2002; Jaeger and Haas, 2004). The unstructured nature of such networks has the advantage of permitting pluripotentiality; but at the likely expense of chaotic dynamics. By contrast, the more restricted dynamics of our networks are better adapted to the particular computations required.

One obvious empirical direction in which to test temporal readout is its likely greater sensitivity to transient changes in the input. For instance, presenting parts of the pattern at slightly different times could be illuminating. It would also be worthwhile to study the effect of presenting patterns with different luminosities for the different components. Change-based readout emphasizes the early portion of the trajectory of the state of the network following presentation of an input. There is experimental evidence that substantial stimulus information is available during this time, and can be extracted from the network (Hung et al., 2005; Mazor and Laurent, 2005; Ganguli et al., 2008). Indeed, it has been suggested that the stochastically stable state that ultimately ensues after a substantial delay is informationally impoverished compared with the early states (Mazor and Laurent, 2005). Further, there are recent data suggesting that relatively fine-temporal scale aspects of activity, even in early sensory cortical areas, can significantly influence the course of decision-making (Yang et al., 2008); this is another facet of change-based readout.

Conversely, the sensitivity of change-based readout to transient activity suggests one route towards a psychophysical test of the idea based on noise images (Dosher and Lu, 1999; Mareschal et al., 2006). Consider presenting patterns corrupted by explicit spatio-temporal noise, and eliciting moderately fast responses from subjects in the task we considered here. By correlating their ultimate decisions with the patterns of noise presented on each trial, it would be possible to examine whether and how they are affected by different epochs of noise (Wong et al., 2007; Nienborg and Cumming, 2009). We may expect change-based inference to be particularly affected by the noise in earlier epochs (whereas attractor-based inference of the sort considered by Li & Dayan (Li and Dayan, 2000) would be more influenced by explicit noise in later epochs).

We should also note that the change-based method requires the information gathered by the first measurement to be kept in the memory for the subsequent comparison with the second measurement. In general, stable patterns of activity are the exception in the brain rather than the rule, and so it is important to understand the range of possible computations created by the transient evolution of neural activities. Recurrent networks offer an attractive metaphor for many neural computations. By showing a different way in which they can be seen as processing information, we hope to open up a wealth of new possibilities.

2.6 Summary

In this chapter we introduced a new way of coding information by recurrent neural networks. We showed that it is possible to encode information optimally by the temporal evolution of the neural activity. Our results indicate that even the simplest case in which the recurrent network encodes information by the change in its state between two time points allows optimal performance. By taking the bisection task as a paradigmatic example, we showed that the change-based method can optimally solve the tasks that involve discrimination in the presence of invariance (in chapter 5 we show that change based coding allows learning which in turn allows the change-based method to be applied to a broader range of invariant discrimination tasks). We also demonstrated its robustness to different sources of noise including dynamical and synaptic noise.

Appendix 2.A: Ideal observer's interpretation for Poisson noise

Here we derive the equations of the ideal observer for the case in which the input noise comes from the Poisson distribution. Based on the Bayes rule we have:

$$P(\varepsilon, y \mid \mathbf{a}) = \frac{P(\mathbf{a} \mid \varepsilon, y) P(\varepsilon, y)}{P(\mathbf{a})} \tag{2.8a}$$

Note that the decision of the ideal observer is based on the following rule:

If the sum $\sum_{\varepsilon > 0, y} P(\varepsilon, y \mid \mathbf{a}) > .5$ then the decision is $\varepsilon > 0$ (2.8b)

(2.8b)

If the sum $\sum_{\varepsilon > 0, y} P(\varepsilon, y \mid \mathbf{a}) < .5$ then the decision is $\varepsilon < 0$ (2.8c)

The location and deviation of the middle bar from the middle point are independent. If we assume a uniform prior over both of them we have:

$$P(\varepsilon, y \mid \mathbf{a}) \propto P(\mathbf{a} \mid \varepsilon, y) \tag{2.9}$$

and since the noise is Poisson, we have:

$$P(\mathbf{a} \mid \varepsilon, y) \propto \prod_i \overline{a}_i^{a_i} \Rightarrow \log\left(P(\mathbf{a} \mid \varepsilon, y)\right) \propto \sum_i a_i \log(\overline{a}_i) \tag{2.10}$$

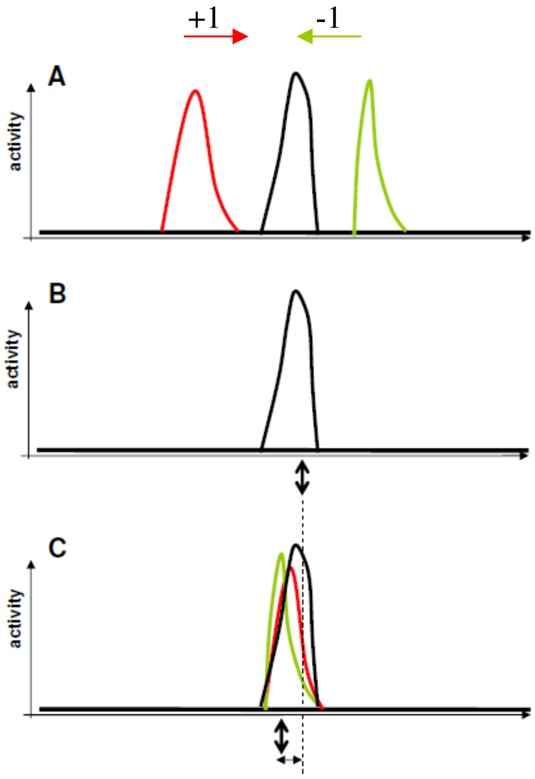

Figure 2.11 An interpretation of the ideal observer for the Poisson noise model. **A**) The three bars and as a result the three hills of activity patterns are assumed to be far from each other. The ideal observer identifies these three bumps. **B**) In the next step the ideal observer measures the CoM of the central bump alone (shown by the arrow). **C**) Then it shifts the outer bumps towards the center by one unit distance (+/-1) and linearly combines them with the central one and measures the CoM of the new activity pattern (shown by the vertical arrow). Finally, it decides based on the comparison between these two CoMs (B versus C; horizontal arrow).

Because in the bisection task, the three bars are assumed to be far enough from each other such that we can assume that each unit is activated only by one of the bars, we therefore divide the neurons in to three groups, neurons that are activated by the left bar,(L), neurons that are activate by the middle bar (M) and neurons that are activated by the right bar (R).

$$log\left(P(\mathbf{a} \mid \varepsilon, y)\right) \propto \sum_i a_i \, log(\bar{a}_i) = \\ \sum_i a_i^L (x_i - y_L)^2 + \sum_i a_i^M (x_i - y_M)^2 + \sum_i a_i^R (x_i - y_R)^2 \tag{2.11}$$

From figure 2.2 we have that: $y_L = -1 + y$, $y_M = \varepsilon + y$ and $y_R = 1 + y$.

$$\log\left(P(\mathbf{a}\,|\,\varepsilon,y)\right)\propto$$
$$\sum_i a_i^L(x_i-(-1+y))^2+\sum_i a_i^M(x_i-(\varepsilon+y))^2+\sum_i a_i^R(x_i-(1+y))^2 \tag{2.12}$$

This is a two dimensional Gaussian with respect to ε and y. Taking the first derivative with respect to both and setting it equal to zero yields:

$$\varepsilon\propto\frac{\sum_i a_i^M x_i}{\sum_i a_i^M}-\frac{\sum_i a_i^M x_i+\sum_i a_i^L(x_i+1)+\sum_i a_i^R(x_i-1)}{\sum_i a_i} \tag{2.13}$$

Since the distribution is Gaussian, the decision rule mentioned above is equivalent to the decision rule that is based on whether the above equation is positive or negative. In other words, the ideal observer calculates the CoM of the middle bump, then it shifts the two outer bumps towards the middle bump and calculates the CoM of this combination and eventually compares them to decide about the sign of epsilon (figure 2.11).

Appendix 2.B: Line attractor is parameterized by the nuisance parameter

Figure 2.12 points out that the single line attractor that we used for the temporal read-out is not a possible solution, since the converged point on the attractor represents y and not ε. This is evident in the fact that an input shifted precisely in y, m-→n, leads to an output shifted by precisely the same amount in y, m'-→n'), and therefore can not support any calculation of the sign of ε.

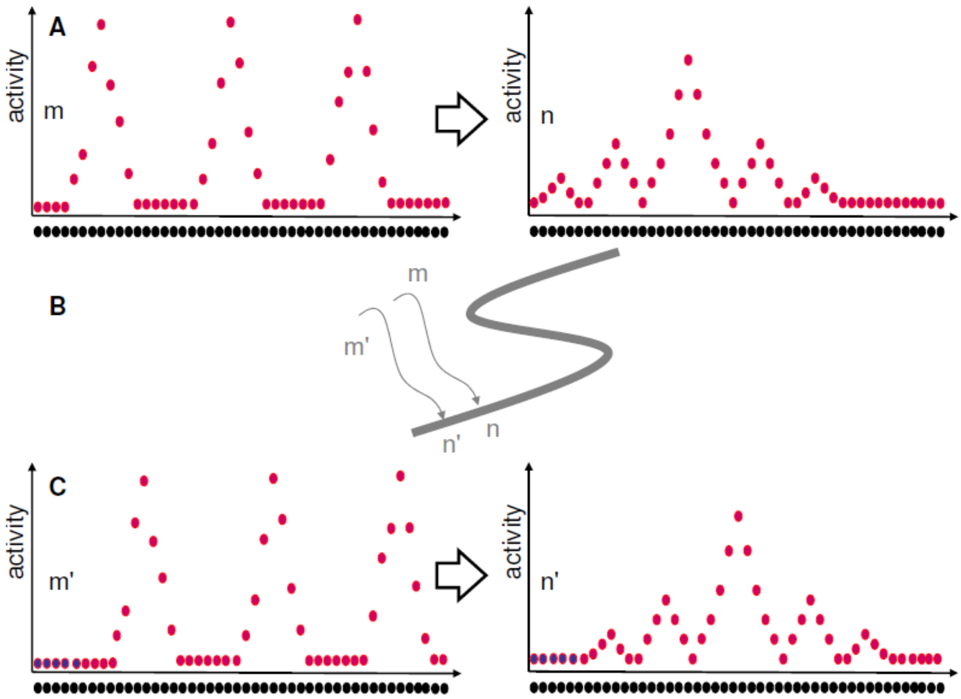

Figure 2.12 Line attractors: transformation from the network space to activity space. **A**) Pattern m converges to pattern n. **C**) Pattern m' is shifted version of pattern m (by five units to the right). Pattern n' to which pattern m' converges will also be a shifted version of the pattern n (again by five units to the right). **B**) Given that both patterns n and n' are on the line attractor, therefore the line attractor is parameterized by y.

Appendix 2C: Change-based performance and the form of the attractors

Figure 2.13A and 2.13B shows the attractor states of the weights shown in figure 2.8B to show that they have different attractors despite having similar performance level. Figure 2.13C shows the performance of weight shown in figure 2.9B to show that the performance of this weight matrix is near optimal despite having a different attractor from the weight shown in figure 2.9A.

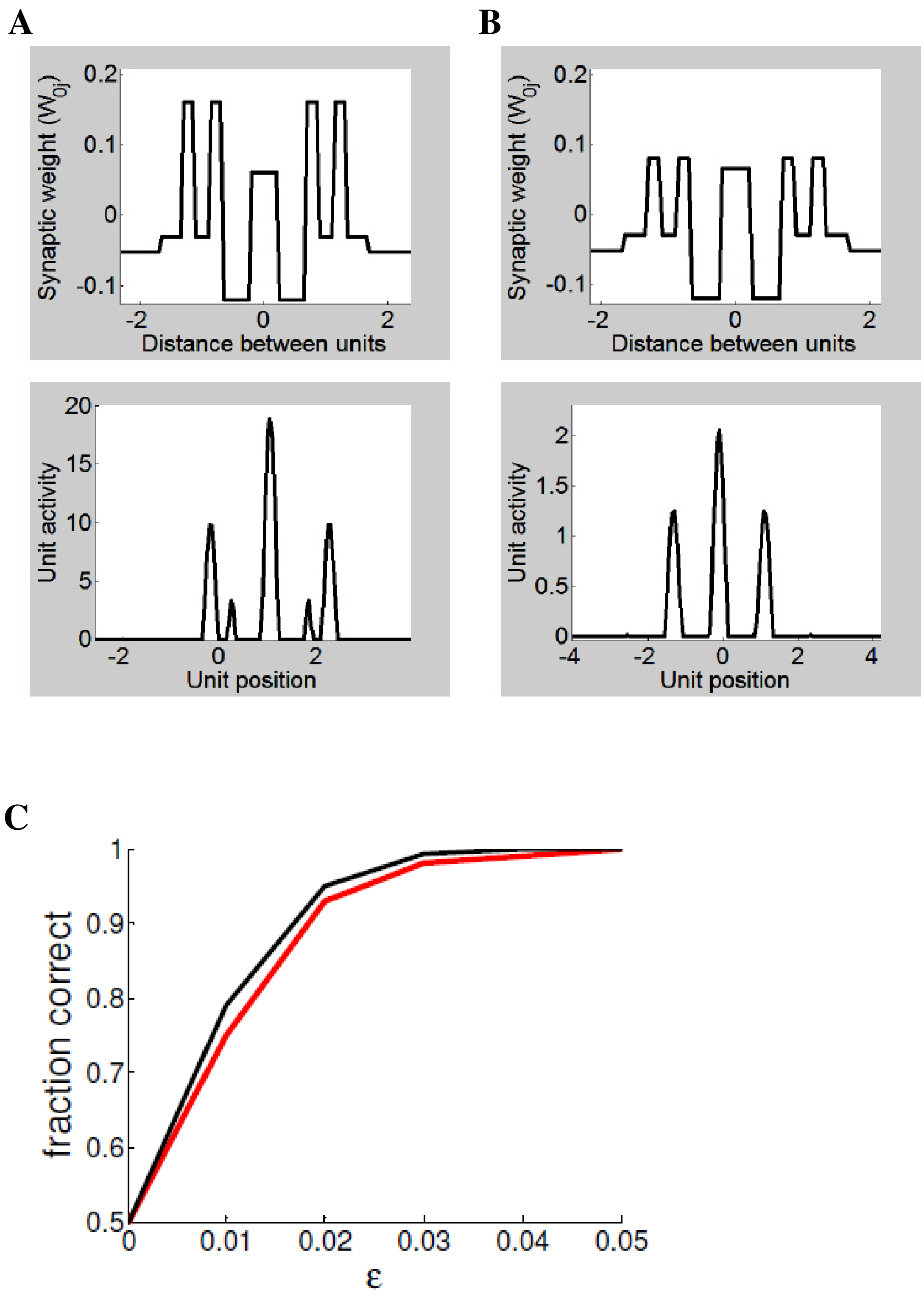

Figure 2.13: Attractors and change-based performance. **A**, **B**) Form of the attractors of the weights shown in figure 2.8. **C**) Performance of the weight shown in figure 2.9B.

Chapter 3

Change-based processing mechanism

3.1 Introduction

We introduced the change-based method in the previous chapter, and studied its performance under different conditions. We showed that the change-based method offers an excellent solution to the problem of invariant discrimination (we only studied the bisection task in chapter two; as we will see in chapter 5, optimal weights can be learned to solve a class of invariant discrimination tasks), has a very high processing speed and is robust to significant levels of both dynamical and synaptic noise. In addition, the change-based method encodes the task relevant information by early changes in the neural activity which, as we will see in the last chapter, allows us to easily learn optimal recurrent weights for a variety of tasks.

In this chapter, we aim to understand the workings of the change-based method and in particular, the mechanisms underlying its optimality. We seek to understand why in the bisection task the initial direction of the movement of the CoM is highly informative about the deviation of the middle bar from the middle point and how the network extracts the task relevant information from the neural activity; this is a challenging task: for example, in the bisection task, the stimulus driven activity carries information about both the nuisance parameter (denoted by y) and the task relevant signal (denoted by ε) and is also corrupted by the noise. In order to solve the task, the network and the readout need to separate the signal information from both the information about the nuisance parameter and the noise.

To address this, we study the change-based method under a new stimulus presentation paradigm that proves useful in understanding its workings. In particular, we slightly modify the stimulus presentation paradigm that we considered in chapter 2; unlike the previous chapter in which the information was presented to the network all at once (represented by the initial condition of the differential equation 2.2) – here the information is progressively made available to the network over a period of time. This way, one can fix the signal (ε) and change the duration of stimulus presentation (or in general its temporal structure) to address this problem. This temporally extended paradigm is extensively used in designing experiments that address perceptual decision making (Britten et al., 1996; Roitman and Shadlen, 2002). We describe the paradigm and the results in the next section.

3.2 Variation of the stimulus location over different time scales

In the previous chapter, we assumed that the location of the bisection stimulus varied over trials but was kept fixed during each trial; in other words the nuisance parameter, i.e. the location of the stimulus y, varied over a longer time scale than that of a single trial. Here, we consider two types of positional variation; in addition to the trial by trial variability of the stimulus location, we assume that the stimulus location also varies during each trial. In fact, both types of positional variation are motivated by the experimental data regarding fixational eye movements, namely, micro-saccades, drifts and tremors (Alpern, 1972; Steinman et al., 1973). While inter-trial variations of the stimulus location models micro-saccades, intra-trial variation of the stimulus location models drifts and tremors that occur during a trial.

Similar to the previous chapter we continue to study the change-based processing in the context of the bisection task but under slightly different stimulus presentation paradigm. To illustrate this, figure 3.1 cartoons two different trials of the temporally extended paradigm. Here, once the stimulus appears on the screen, it stays there for a duration, say 10 ms, and then moves to a slightly different location; then it stays there for another 10 ms and then moves again to another nearby location and so on and so forth until the stimulus disappears. We assume that the intra-trial movements of the stimulus during a trial (that model drift and tremor) are small in magnitude. In contrast, the inter-trial movements (that model micro-saccades) can be large. We also assume that the micro-saccades are suppressed during each trial and they only occur during inter trial intervals. As can be seen from figure 3.1, the stimulus appears at a different location at the second trial (due to micro-saccades that take place in between trials) and begins to jitter around this new location until it disappears. As we mentioned above, we assume that the amplitude of the jitter is small and of the same order as the signal (ε).

As we discussed in previous chapters, since the discrimination task is not local, it can not be solved by a two-layer threshold-linear feedforward network. On the other hand, we showed that the change-based method offers an excellent solution to a task of discrimination in the presence of invariance. It remains unclear whether the change-based method would be able to "extract" and "combine" the task relevant information in the presence of the jitter. In the next section, we challenge the change-based method with this new paradigm.

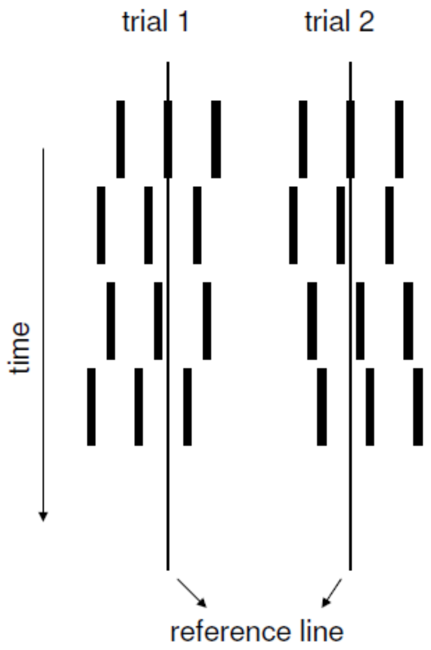

Figure 3.1 Temporally extended stimulus presentation paradigm. Two different trials of the simulations are cartooned here. The stimulus jitters at each trial but the magnitude of the jitter is relatively small. This jitter can be regarded as a model for drifts and tremors during fixation. The overall location of the stimulus (represented by the reference lines) is different at different trials and this difference is normally much larger than the jitter during a trial. Note that the experimenter presents the bisection stimulus at the same location on all trials; the change in the stimulus location from reference line 1 to reference line 2 is due to (and can be regarded as a model of) micro-saccades that occur during inter trial intervals. The reference lines are imaginary and just for illustrative purposes.

3.3 Change-based method with this new paradigm

3.3.1 Model

The network model that we use in this chapter is the same as the one we used in the previous chapter. Time is discretized into 1 ms bins and the differential equation governing the dynamics of the recurrent network is as follows:

$$\tau\Delta u_i(t) = -u_i(t)\Delta t + C\Delta t + \eta a_i^t \delta^t(s)\Delta t + \sum_{j=1}^{N} W_{ij} f(u_j(t))\Delta t \tag{3.1}$$

$$u_i(0) = \eta a_i^0 \delta^0(s) \qquad 1 \le i \le N$$

where $u_i(t)$ is the network driven input to neuron i at time t, $\Delta u_i(t) = u_i(t+\Delta t) - u_i(t)$, $\delta^t(s) = 1$ if stimulus is presented at time t and $\delta^t(s) = 0$ otherwise, and a_i^t is the stimulus driven input to neuron i at time t. We study this network under different stimulus conditions. In the first condition, the stimulus is presented continuously and the network gets independent samples of the stimulus every ΔT ms until the stimulus disappears. In the second condition, the stimulus is presented intermittently every ΔT_1 ms for a duration of ΔT_2 ms. Figure 3.2 illustrates an example temporal interaction between independent samples presented to the network within a trial.

3.3.2 Results

Figure 3.3A shows an example trajectory of the CoM for the case in which the stimulus is presented persistently for 80 ms and the network receives independent

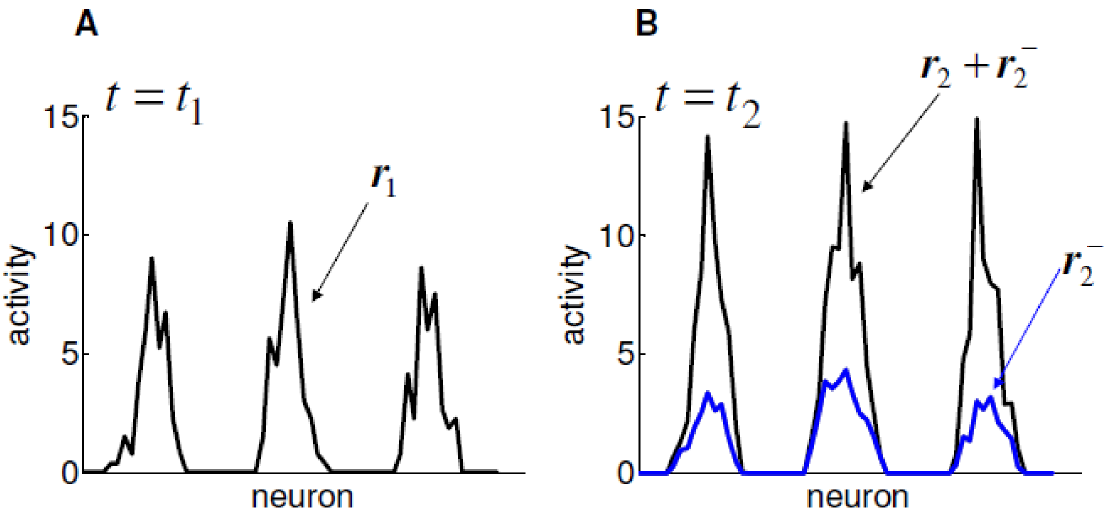

Figure 3.2 Temporal interaction of the temporally distributed samples. **A**) The noisy response pattern elicited by the stimulus at location y_1 (signal strength is ε for all samples) that is presented at time t_1. **B**) The network begins to evolve until time t_2 at which the second sample is presented. $\mathbf{r}_2^-$ represents the activity pattern of the network just before the presentation of the second sample. The second sample elicits response $\mathbf{r}_2$ that is linearly combined by $\mathbf{r}_2^-$ and the network continues to evolve until the third sample is presented.

samples of the stimulus every 1 ms (as we will see later, the performance is not sensitive to the sampling rate as long as the rate is within a range which is determined by the network's dynamics explained in details in the next section). The network gets 80 independent samples of the stimulus (every 1 ms for 80 ms) during this period. As can be seen, the CoM fluctuates during the stimulus presentation period which continues for a while after the stimulus disappears and eventually starts to deviate towards right.

Figure 3.3B shows the performance of the change-based method along with that of the ideal observer. Because there are 80 evidences, the performance of the ideal observer saturates at much lower signal levels than when there is only one evidence, the case that we studied in the previous chapter.

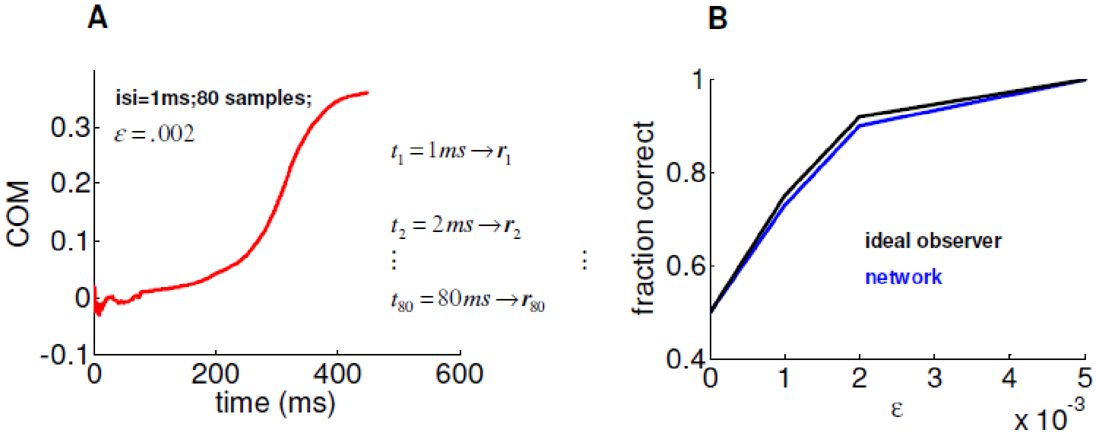

Figure 3.3 Evolution of the CoM under extended stimulus presentation paradigm (continuous). **A**) An example case in which the stimulus is shown for 80 ms (from time 1 to 80). We assume that the network gets a new independent sample every 1 ms. This is a correct trial for a signal level of .002. **B**) Performance of the network as a function of the signal's strength for both the change-based method and the ideal observer. Note that the ideal observer considers all the independent samples separately to make a decision while the neural activity elicited by different samples get linearly added up in the network and therefore the network does not get a separate copy of each independent sample, yet its performance is near optimal.

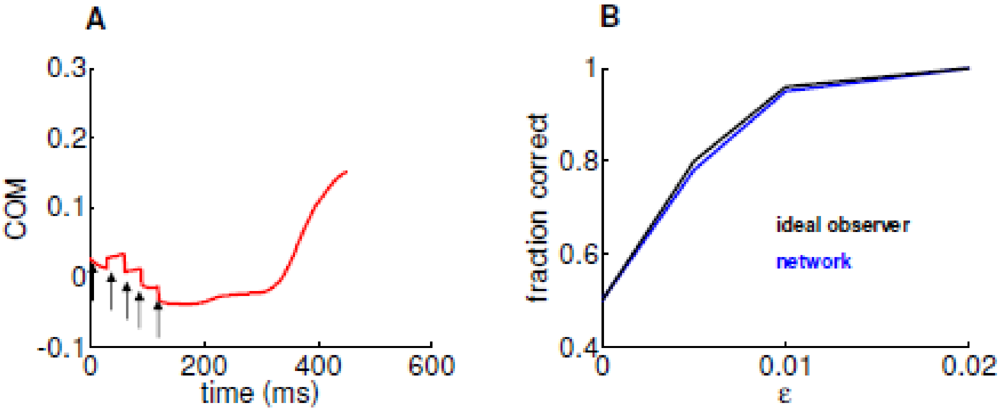

Figure 3.4 Evolution of the CoM under extended stimulus presentation paradigm (intermittent). **A**) An example case in which 5 samples are presented every 30 ms. Arrows indicate the timings of the stimulus presentations. This is a correct trial for $\mathcal{E} = .005$. **B**) Performance of the network as a function of the signal's strength for both the change-based method and the ideal observer.

Figure 3.4A shows another example case where 5 samples are presented every 40 ms and for a 5 ms duration each. Again, the performance of the change-based method follows the ideal observer and remains near optimal (figure 3.4B). Note that in both this case and the previous case mentioned in the above paragraph, the two times at which the CoM is measured are after the time at which the latest sample is presented to the network.

These results indicate that the change-based method is not only entirely invariant to the location of the stimulus on trial by trial basis but also extracts the task relevant information successfully in the presence of a low amplitude jitter. From a computational standpoint, to solve the bisection task optimally, the network needs to extract the information about the signal from each sample, keep them in a memory and combine them and eventually translate the end result into a change of CoM to express its decision. The results we have got so far indicate that the network and the change-based readout accomplish all these steps successfully. But how?

3.4 Analysis

3.4.1 Linearization far from the attractor

Since in all cases that we have seen so far, the change-based method performs optimally while the state of the network is far from the attractors and close to the initial state, it is natural to address the change-based method by analyzing it in the regime where it operates, i.e. while the state of the network is far from the attractors. Linearization techniques have been previously used to study the performance of the attractor-based method within line-attractor networks. However, as we discuss in appendix 4.C, the linearization in line attractor networks has been around the converged state of the network where the computations are believed to be accomplished. In contrast, to analyze the change-based method, we linearize the network around its initial state since the computations are carried out during the early portion of the neural activity. Figure 3.5 illustrates the idea behind our analysis. This analysis is for the threshold-linear activation function that is commonly used to study neural dynamics and computation (for instance, see Hahnloser et al., 2000). It approximates the sigmoidal nonlinearity, that is commonly used in neural modeling (Haykin, 1998), by assuming that neurons operate in the non-saturated regime (informative regime) of the sigmoidal activation function. Imagine that the state of the network immediately after the presentation of the very last sample (just after the stimulus disappears) is the one depicted in figure 3.5. We linearize the network around this state and study the resulting linear system.

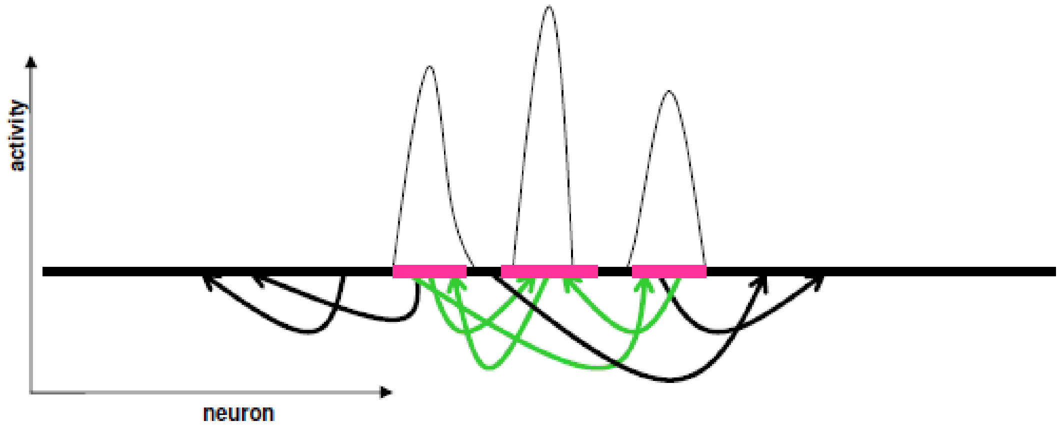

Figure 3.5 Linear analysis of the change-based method. The noisy activity pattern (three hills of activity) cartoons the activity of the network immediately after the end of the stimulus presentation. One possibility is to linearize the network around this state. As we discuss in the paper, this would make it difficult to compare different trials for different trials might be associated with slightly different initial states. Alternatively, one could think of an “average” trial that could be regarded as an initial state for all trials. Such an average trial (defined in the text) selects out neurons marked by red color. The synapses that connect the red neurons are shown with green color and the set of green synapses are called the reference weight matrix. The reference weight matrix is not only a good approximation to different initial states associated with different trials but also allows us to better understand the underlying mechanisms of change-based processing.

If we denote the recurrent input to the network just after the very last sample by $\tilde{\mathbf{u}} = (\tilde{u}_1, \tilde{u}_2, ..., \tilde{u}_N)$, and the output of the network by $\tilde{\mathbf{a}} = f(\tilde{\mathbf{u}})$ then linearization approximates the network as follows:

$$\tau \Delta u_i(t) = -u_i(t)\Delta t + \sum_{j=1}^{N} W_{ij} \frac{\partial f(\tilde{u}_j)}{\partial \tilde{u}_j} \Delta u_j \tag{3.2a}$$

$$u_i(0) = \tilde{u}_i \qquad 1 \le i \le N$$

Which, in vector form, is represented as follows:

$$\tau\Delta\mathbf{u}(t) = -\mathbf{u}(t)\Delta t + \sum_{j=1}^{N}\mathbf{W}\frac{\partial\mathbf{f}(\tilde{\mathbf{u}})}{\partial\tilde{\mathbf{u}}}\Delta\mathbf{u} \qquad (3.2b)$$

$$\mathbf{u}(0) = \tilde{\mathbf{u}} \qquad 1 \le i \le N$$

where the term $\frac{\partial\mathbf{f}(\tilde{\mathbf{u}})}{\partial\tilde{\mathbf{u}}}$ is the Jacobian of the network around the initial state $\tilde{\mathbf{u}}$ and the term $\mathbf{W}\frac{\partial\mathbf{f}(\tilde{\mathbf{u}})}{\partial\tilde{\mathbf{u}}}$ is the weight matrix associated with the linearized system. We call this the “effective weight matrix”. However, as can be seen from the equation of the effective weight matrix, linearizing around the initial state has the drawback that the initial state changes on trial by trial basis as it depends on the temporal distribution of the evidences and the noise at each trial. Therefore, linearizing around the initial state according to equation 3.2b results in having different linear systems for different trials and makes it difficult to compare different trials with each other. Therefore, for a fixed location, we thought about an initial state that would be the average of all the initial states associated with that location. Therefore the linear equations describing the linear approximation to the system would be as follows:

$$\tau\Delta\mathbf{u}(t) = -\mathbf{u}(t)\Delta t + \mathbf{W}\frac{\partial\mathbf{f}(\overline{\mathbf{u}})}{\partial\overline{\mathbf{u}}}\Delta\mathbf{u} \qquad (3.3)$$

$$\mathbf{u}(0) = \overline{\mathbf{u}}$$

where $\overline{\mathbf{u}}$ represents the average initial state (in contrast to $\tilde{\mathbf{u}}$ that represents the initial state corresponding to a single trial). Such an initial state selects a subset of the neurons that would be on average active if the stimulus is presented at a certain location. These neurons are represented by red color in figure 3.5. Linear analysis selects out these neurons and the synapses that connect them (the effective weight matrix here is represented by $\mathbf{W}\frac{\partial\mathbf{f}(\overline{\mathbf{u}})}{\partial\overline{\mathbf{u}}}$) and assumes that the network consisting of these neurons and synapses is linear. Therefore every trial is analyzed assuming that $\mathbf{W}\frac{\partial\mathbf{f}(\overline{\mathbf{u}})}{\partial\overline{\mathbf{u}}}$ is the effective weight matrix. We call it “reference weight matrix” since it allows us to compare different trials. All trials are linearly analyzed with respect to the reference weight matrix.

As we mentioned earlier the idea of linearizing the network around the reference weight matrix is primarily to make it possible to compare different trials. However, one might argue that the reference weight matrix $\mathbf{W}\frac{\partial \mathbf{f}(\bar{\mathbf{u}})}{\partial \bar{\mathbf{u}}}$ is different from the actual effective weight matrix $\mathbf{W}\frac{\partial \mathbf{f}(\tilde{\mathbf{u}})}{\partial \tilde{\mathbf{u}}}$ at different trial. In fact, we are approximating the effective weight matrix at each trial by the reference weight matrix. This approximation has a very negligible effect on the results since the two activity patterns $\tilde{\mathbf{u}}$ and $\bar{\mathbf{u}}$ differ only on the neurons that are at the edges of the three activity bumps and therefore their activity is comparably small (compared to the central part of the activity bumps) and as a result their effect on the dynamics is negligible. Analyzing around the reference weight matrix rather than the effective weight matrix makes it much easier to understand the workings of the change-based method.

It is helpful to divide the information processing of the network in to two stages. In the first stage, which we call it "loading", the information is presented to the network. That is the duration over which the stimulus is on. In the second stage, the stimulus is off and the network begins to shift the center of mass to express its decision. We call this "unloading" stage.

3.4.2 Linear sub network

Figure 3.6A shows an example weight matrix which, similar to those in the previous chapter, is symmetric and translation invariant. Figure 3.6C shows the reference weight matrix (cartooned as green connections associated with red neurons in figure 3.5; note that the other connections (those that are not green) are set to zero in figure 3.6C). We get the weight matrix shown in figure 3.6D by only considering the green connections of the recurrent weight matrix. Finally, figure 3.6E shows the eigenvalue spectrum (eigenvalues associated with the eigenvectors) of the reference weight matrix shown in figure 3.6D.

Among the eigenvectors of the reference weight matrix, as we will see later, the one with highest eigenvalue has interesting properties and plays a key role in change-based processing.

We divide the set of eigenvectors of the reference weight matrix into even (symmetric around the center) and odd (not symmetric) subset of eigenvectors. (an even function is a function where f(x)=f(-x) and an odd function is a function where f(x)=-f(-x). every function can be written as a sum of an odd and an even function. By an odd eigenvector we refer to the odd component of that eigenvector).

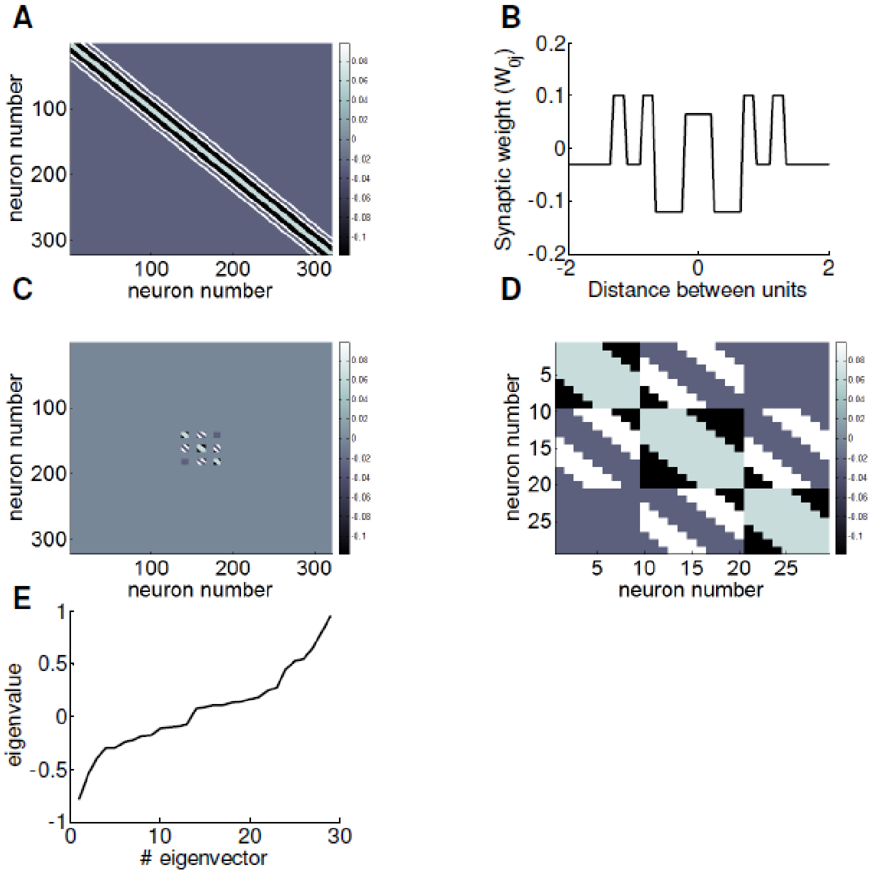

Figure 3.6 Reference weight matrix. **A**) Synaptic weight matrix of the recurrent network which is symmetric and position invariant. **B**) Central row of the weight matrix shown in (A). **C**) Reference weight matrix associated with green synapses in figure 3.5. Synapses that are not green are set to zero. **D**) Reference weight matrix in a new space where only the linear network is considered and the rest of the network is ignored. **E**) The eigenspectrum of the reference weight matrix.

Any shift of the CoM is not caused by the symmetric eigenvectors because they are symmetric and can not favor one direction to another. Therefore the movement of the CoM towards right or left is caused by the odd eigenvectors. In fact, the eigenvector with highest eigenvalue is an odd eigenvector which is shown in figure 3.7A. As we mentioned above, this eigenvector has very interesting properties which helps us understand the change-based method. These properties are as follows: Orthogonality to the invariance, sensitivity to the location of the middle bar and controlling the direction of the movement of the CoM. We will explain each of these in the following.

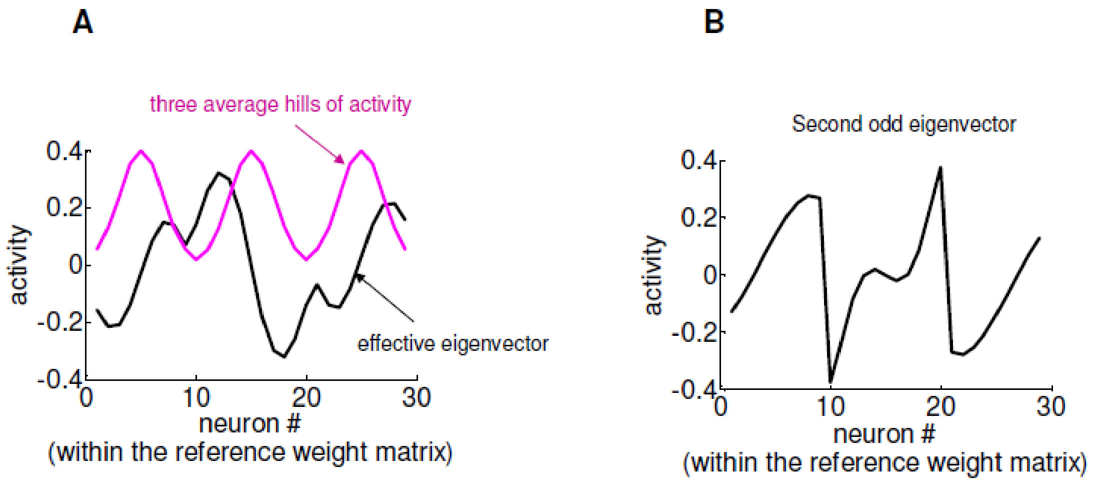

Figure 3.7 Effective eigenvector. **A**) Effective eigenvector of the weight matrix shown in figure 3.6A superimposed by the average activity for the case with zero signal. The effective eigenvector consists of three parts that are very similar to the first derivative of a Gaussian. **B**) The second odd eigenvector of the weight matrix shown in figure 3.6A.

3.4.3 Orthogonality to the invariance

Figure 3.7A depicts the effective eigenvector and the average activity elicited by the bisection stimulus, both rescaled. They are both represented in the new space in which only the sub-network associated with the reference weight matrix is considered (figure 3.6D). The jitter in the stimulus location shifts the average activity randomly towards left or right; since the magnitude of the jitter is small, it can be represented in the state-space in a direction defined by the change in the average activity as a result of a small change in the stimulus location. In other words, given that the magnitude of the jitter is small, the direction to which the discrimination should be orthogonal is the first (spatial) derivative of the average activity. Formally, if the dependence of the average activity $\bar{\mathbf{a}}$

on the location x is denoted by $\bar{\mathbf{a}}_x$, then the direction, $\mathbf{I}$, to which the inference should be invariant is computed as follows:

$$\mathbf{I} = \bar{\mathbf{a}}_{x+\delta x} - \bar{\mathbf{a}}_x = \frac{\bar{\mathbf{a}}_{x+\delta x} - \bar{\mathbf{a}}_x}{\delta x}\delta x = \frac{\partial \bar{\mathbf{a}}_x}{\partial x}\delta x \qquad (3.4)$$

where δx is a small displacement (jitter) of the stimulus location. Figure 3.8 shows the eigenvector superimposed on the direction specified by $\mathbf{I}$ (again, both are rescaled).

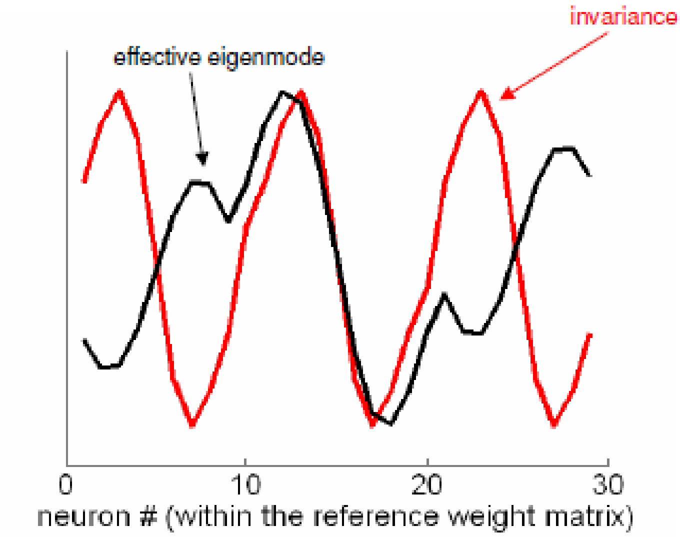

Figure 3.8 Effective eigenvector is orthogonal to the invariance. Effective eigenvector (black) of the weight matrix shown in figure 3.6A, superimposed on the first derivative of the average input (invariance; red). The inner product of the two vectors is zero.

The inner product of the effective eigenvector and $\mathbf{I}$ is zero. This orthogonality means that the effective eigenvector does not “see” the jitter at all; it is entirely invariant to it.

3.4.4 Sensitivity to the location of the middle bar

The mechanism that underlies the optimality of the change-based method not only needs to be invariant to the jitter in the location of the stimulus but also needs to be sensitive to the deviation of the middle bar from the middle of the two outer bars. The average activity generated by the middle bar (as is shown in figure 3.7A) is a smooth hill at the center and any deviation of the middle bar from the middle point only affects the central hill of activity and not the two outer ones. If we denote the average hill of activity generated by the central bar at location x by $\overline{\mathbf{a}}_x^m$ (where m stands for “middle”), then the change in $\overline{\mathbf{a}}_x^m$ as a function of change in the location of the central bar is:

$$\overline{\mathbf{a}}_{x+\delta x}^m - \overline{\mathbf{a}}_x^m = \frac{\overline{\mathbf{a}}_{x+\delta x}^m - \overline{\mathbf{a}}_x^m}{\delta x}\delta x = \frac{\partial \overline{\mathbf{a}}_x^m}{\partial x}\delta x \tag{3.5}$$

Which is proportional to the first (spatial) derivative of the middle bump. Figure 3.9 shows the effective eigenvector superimposed by the first derivative of the middle bump of the activity and as can be seen, they are highly similar. This indicates that the effective eigenvector is very sensitive to the deviations of the middle bar and is therefore able to detect and measure the task relevant signal. In general, the best way to quantify the sensitivity of the effective eigenvector to signal is to measure its performance in the presence of noise as is shown in figure 3.13 (see section 3.4.6). Being orthogonal to the direction specified by the jitter and sensitive to the task relevant signal are two key properties that make the effective eigenvector an ideal candidate for accumulating the temporally distributed information. Therefore, we examined the range of time over which this eigenvector is able to accumulate the incoming task relevant signals.

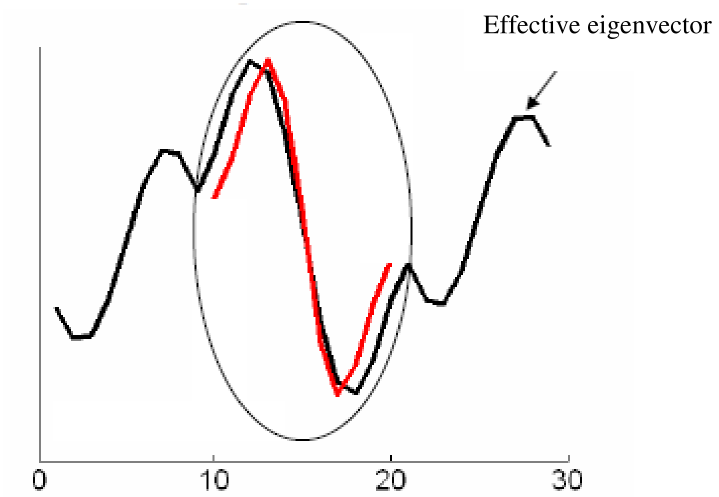

Figure 3.9 Sensitivity to the location of the middle bar. Effective eigenvector (black) of the weight matrix shown in figure 3.6A is superimposed by the first derivative of the middle hill of the average activity (that carries the signal; red curve). The central part of the effective eigenvector (marked by an ellipse) is very similar to the first derivative of the central hill of the average activity, indicating that the effective eigenvector is sensitive to the deviations of the middle bar from the middle of the two outer bars.

.

The effective weight matrix is symmetric (since we assume that the nonlinearity is threshold-linear nonlinearity), and as a result, the eigenvectors are orthogonal to each other and the eigenvalues are all real. We decompose the linearized differential equation 3.3 along each eigenvector; in particular, decomposing it along the effective eigenvector yields:

$$\tau\Delta\alpha_{eff}(t) = -\alpha_{eff}(t)\mathbf{v}_{eff}\Delta t + \lambda_{eff}\alpha_{eff}(t)\mathbf{v}_{eff}\Delta t \qquad (3.6)$$

This equation can be rewritten as follows:

$$\begin{aligned} &\frac{\tau}{\lambda_{eff}-1}\Delta\alpha_{eff}(t) = \alpha_{eff}(t)\mathbf{v}_{eff}\Delta t \\ &\underset{\Delta t\to 0}{\Rightarrow} \frac{\tau}{1-\lambda_{eff}}\frac{\partial\alpha_{eff}(t)}{\partial t} = -\alpha_{eff}(t)\mathbf{v}_{eff} \end{aligned} \qquad (3.7)$$

Thus, the effective time constant for this eigenvector is $\dfrac{\tau}{1-\lambda_{eff}}$. Given that τ is 20 ms and the effective eigenvalue is .93, the effective time constant is about 300 ms.

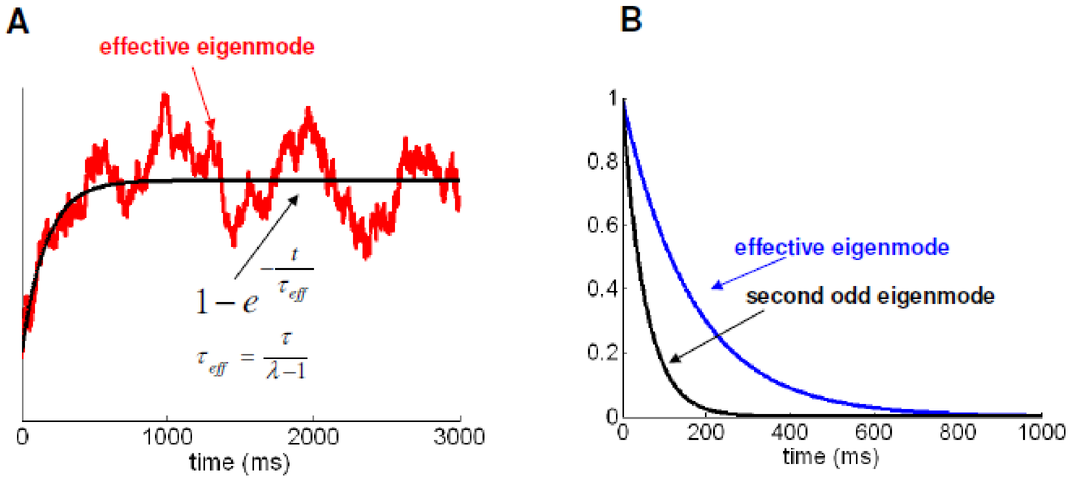

Figure 3.10 Effective eigenvector has the highest time constant. **A**) Evolution of the coefficient of the effective eigenvector as the network samples every 1 ms for a duration of 3000 ms (red). Black curve reflects the theoretical prediction. **B**) The time course of decay for both effective eigenvector and the second odd eigenvector.

Figure 3.10A shows how the effective eigenvector accumulates the information about $\mathcal{E}$; it saturates at around 500 ms, in line with the theoretical prediction (black line). This predicts that if the inter stimulus interval is more than about 500 ms, then the effective eigenvector would not be able to combine them, since by the time the next sample arrives, the impact of the previous one would already have decayed.

3.4.5 Unloading

We have shown that the effective eigenvector accumulates evidences (samples) optimally and is orthogonal to the direction defined by the jitter. However, it is not clear if the information in this eigenvector is the basis for the subsequent movement of the CoM. For example, it might be that a different mechanism also gathers the incoming information similar to the effective eigenvector and in addition controls the movement of the CoM.

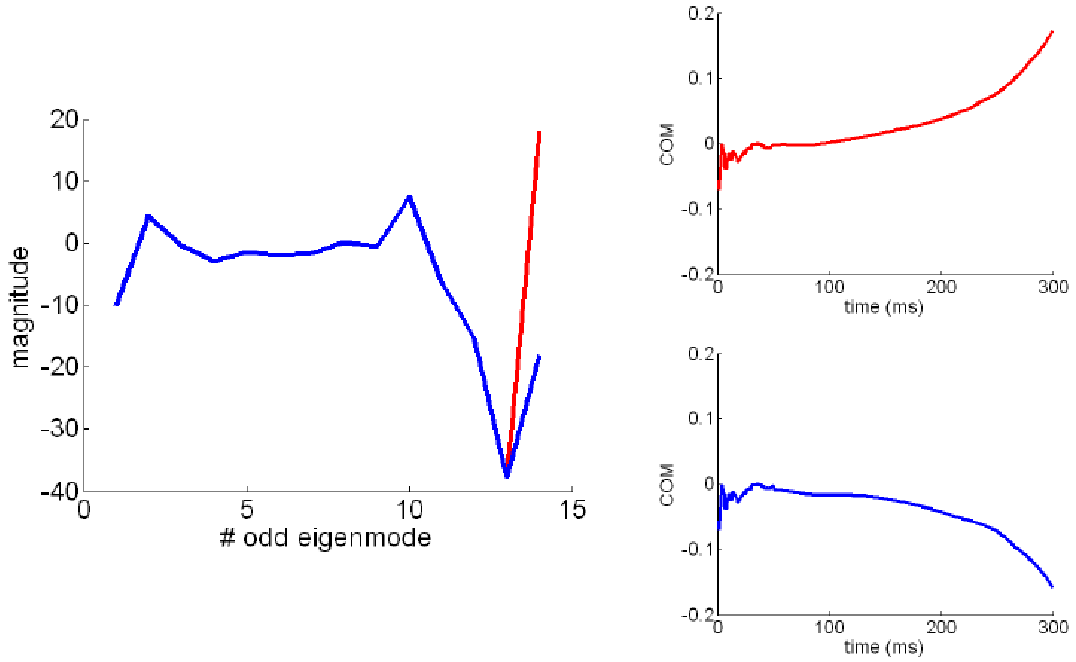

Figure 3.11) Decomposition of the neural activity into eigenvectors of the reference weight matrix. Left: Blue curve indicates the magnitude of each odd eigenvector after activity decomposition according to equation 3.8. Right, Bottom) The CoM shifts towards left in this case. Right, Top) By changing the sign of the coefficient of the effective eigenvector (Red curve in the figure on the left) while keeping others unchanged, the CoM starts to shift towards right.

To test this, we changed the sign of the effective eigenvector and studied its effect on the evolution of the CoM. To this end, we first decomposed the networks activity (immediately after the stimulus disappeared) in to the eigenvectors of the reference weight matrix as follows:

$$\mathbf{a} = \sum_i \alpha_i \mathbf{v}_i \tag{3.8}$$

where $\mathbf{a}$ is the activity immediately after the stimulus disappears, $\mathbf{v}_i$ are the eigenvectors of the reference weight matrix and α_i are the coefficients after decomposing activity in to the eigenvectors of the reference weight matrix. The blue curve in Figure 3.11 Left shows an example in which the coefficients of the decomposition for only the odd eigenvectors are shown. In this case, the CoM decreases and the pattern moves towards left (figure 3.11 Right, Bottom).

Next we changed the coefficient of the effective eigenvector from negative to positive (figure 3.11 Left, red curve) while we kept the other coefficients the same. As can be seen, CoM increased and the pattern moved towards right (figure 3.11 Right, Top). It seems that the future direction of the movement of the CoM is controlled by the coefficient of the effective eigenvector. Next we measured the correlation between the coefficient of the effective eigenvector and the change of the CoM.

Figure 3.12B shows that the sign of the coefficient of the effective eigenvector is perfectly correlated with the future direction of the movement of the CoM. If it is positive then the CoM moves towards right and if it is negative then the CoM moves towards left. Therefore the same eigenvector that optimally accumulates the temporally distributed samples also controls the future evolution of the CoM and plays a fundamental role in change-based processing. Figure 3.12B combines all trials in which ε is randomly drawn from the range -.05 to .05. To control for this, we also considered the case in which there is no signal (ε is zero) and therefore the CoM is only driven by the noise. Again, as can be seen in figure 3.12D, the sign of the effective eigenvector is highly correlated with the sign of the change of the CoM. Why the coefficient of the effective eigenvector predicts the evolution of the center of mass of the neural activity so accurately? As we showed in figure 3.6E, the effective eigenvector has the largest eigenvalue and therefore, according to equation 3.7, has the slowest decay. Figure 3.10B shows how quickly the effective eigenvector and the second slowest eigenvector decay to zero. Note that all other odd eigenvectors decay even faster than the second slowest one. As can be seen, all odd eigenvectors vanish around 200 ms while the effective eigenvector is still way above zero at that time. Therefore, the effective eigenvector takes full control of the CoM because other odd eigenvectors have already vanished and this is the only operating odd eigenvector. As a result, the effective eigenvector fully controls the subsequent evolution of the center of mass.

3.4.6 Neural noise

As we showed in figure 3.12B, the change of the CoM is reliably predicted by the

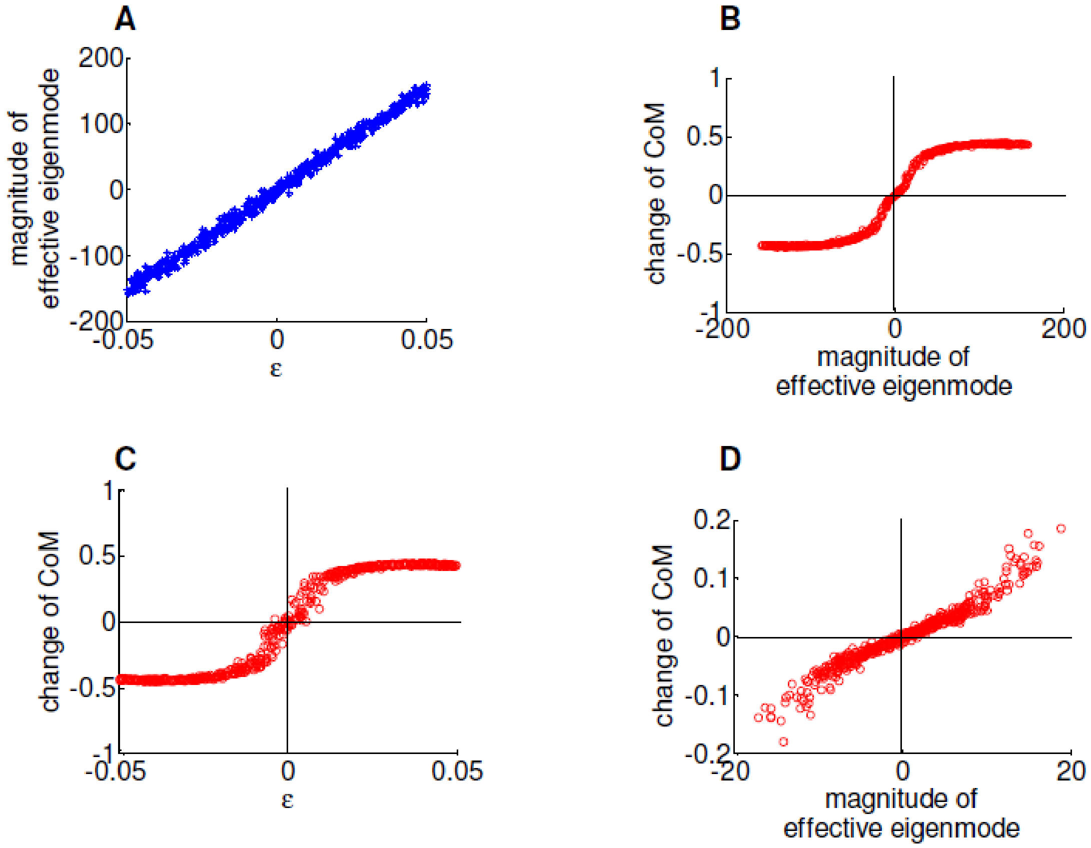

Figure 3.12 Change of the CoM versus effective eigenvector. **A**) The linear relation between the signal, $\mathcal{E}$, and the coefficient of the effective eigenvector. **B**) The change of the CoM between times 120 and 250 ms as a function of the coefficient of the effective eigenvector for a range of signal, $\mathcal{E}$, between -0.05 to 0.05. **C**) Change of the CoM as a function of $\mathcal{E}$. **D**) The change of the CoM between times 120 and 250 ms as a function of the coefficient of the effective eigenvector for the case in which the signal is zero and the change in the CoM is entirely noise driven.

coefficient of the effective eigenvector. Since the performance, which is based on the change of the CoM, is optimal, we conclude that the performance is also optimal based on the coefficient of the effective eigenvector. To test the effect of the noise on the effective eigenvector directly, we assume a hypothetical readout that has access to the coefficient of the effective eigenvector and decides based on that (or more accurately, its sign). As can be seen from figure 3.13, the performance of this readout is also optimal, indicating that the effective eigenvector is minimally affected by the noise.

Note that there exists a class of vectors that are orthogonal to the jitter. There also exists another class of vectors that have a very high signal to noise ratio, i.e. its members are maximally sensitive to the signal and minimally sensitive to the noise. The effective eigenvector is indeed a simulation-based proof that there exists a vector that is orthogonal to the jitter and has the highest possible to signal to noise ratio, i.e. it belongs to both classes. In addition, its effective time constant allows optimal evidence accumulation over a range of time. In chapter 5, we will discuss signal to noise ratio in more detail and we will show that we can learn the optimal weights by simply maximizing the signal to noise ratio.

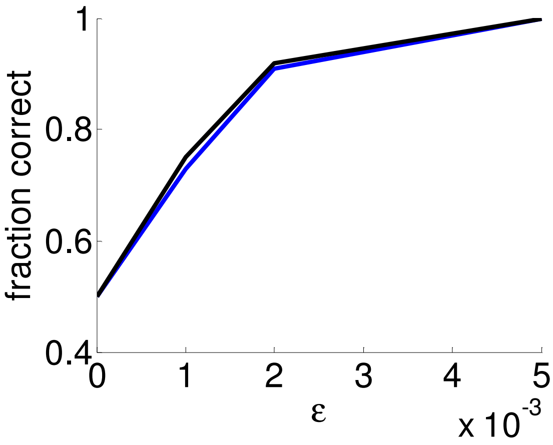

Figure 3.13 Effect of noise on the effective eigenvector. The performance of a hypothetical readout that has access to the coefficient of the effective eigenvector immediately after the last sample is shown (blue). Black curve indicates the performance of the ideal observer.

3.5 Discussion

3.5.1 Uncertainty

Several studies have shown that subjects are able to perform a range of psychophysical tasks near optimally (Jacobs, 1999; Ernst and Banks, 2002; Alais and Burr, 2004; Körding and Wolpert, 2004). Other studies have also shown that some visual illusions can also be accounted for by assuming that the subjects perform Bayesian computations (Weiss et al., 2002). It has been suggest that the neural circuits might represent probability distributions and perform Bayesian computations (Hinton et al., 1995; Rao, 1999); however, there is no direct experimental evidence that proves that the neural circuits represent probability distributions at every point in time. In fact, optimality could be achieved without representing the distributions. For instance, consider the ideal observer model for the bisection task that we discussed in appendix 2.A. There, one way to solve the task optimally is to perform Bayesian computation specified in equation 2.8a and then make the decision based on 2.8b and 2.8c. Alternatively, one could make the decision based on equation 2.13. The two are equivalent but the first case requires direct representation of the probability distributions while the second case specifies a mapping from stimulus to decision. One could imagine a system that learns this mapping without representing the full distributions. Such a system would be optimal but does not explicitly represent the probability distributions.

In this regard, we point to an interesting aspect of the change-based method. From figure 3.12A we learned that the magnitude of the effective eigenvector is linearly correlated with the amount of evidence provided to the network. We also learned from figure 3.12B that the sign of the effective eigenvector predicts the change of the CoM. Combining these two figures (figure 3.12C), we see that there is a range over which the change of the CoM is linearly related to the amount of evidence provided to the network. In other words, larger change of the CoM reflects larger evidence in favor of the corresponding decision. Therefore the same variable that reflects the decision (i.e. change of CoM) also reflects the uncertainty about the decision as well. As can be seen in figures 3.12B and 3.12C, it saturates once the amount of incoming evidence gets beyond a certain level. From network's standpoint, the interpretation is that the change of the CoM saturates because of the nature of the connectivity and the form of the attractors of the network.

3.5.2 change-based method and perceptual decision making

We developed the change-based method primarily to solve the problem of invariant discrimination, and the paradigms that we developed and studied in this chapter helped us understand the workings of the changed-based processing; however, the change-based method also suggests a solution to the problem of evidence accumulation over time which is a central issue in perceptual decision making. In this section we discuss the difference between the change-based processing and more conventional theories of perceptual decision making. We also discuss experimental data that are at odds with the conventional theories of perceptual learning to provide a better understanding of the conventional theories.

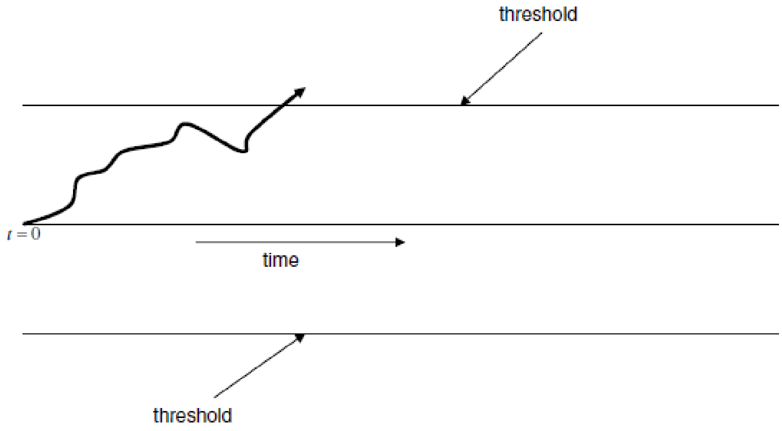

Figure 3.14 Cartoon illustrating the accumulation to bound theory. The two thresholds represent two different options. At each point in time, an evidence supporting one of the options is provided. A decision variable represents this by moving towards the threshold that is supported by the evidence. The option associated with the threshold that the decision variable crosses first is more likely to be supported by the evidences and is selected.

Figure 3.14 cartoons “Accumulation to bound” theory of perceptual decision making. Figure 3.14 illustrates this for the case in which there are two choices (similar to the bisection task) and an agent is tasked to choose the one that is supported by the incoming samples. The accumulation to bound theory assumes that there is a decision variable and two thresholds representing the two choices. Every new sample shifts the decision variable towards the threshold associated with the choice that it supports. The magnitude of the shift reflects the extent to which the sample supports the corresponding choice. The agent chooses the option that is associated with the threshold that the decision variable reaches first. A class of neurons have been found in area LIP whose activity resembles the temporal evolution of the decision variable in the accumulation to bound theory (Roitman and Shadlen, 2002).

However, despite several studies reporting on neurons in LIP to support the accumulation to bound theory of decision making, there are experiments in which it is difficult to interpret the LIP activity as such. Figure 3.15A is taken from (Roitman and Shadlen, 2002) where they showed the similarity between the LIP activity and the predictions of the accumulation to bound theory. However, in a control experiment (figure 3.15B), the monkey was not allowed to express its decision until a go signal appeared on the screen after at least 1 second after the stimulus (which was presented for a second) disappeared. As can be seen, the well known ramping activity, that has been regarded as an evidence for the accumulation to bound theory of the perceptual decision making, is well correlated with the reaction time again which in this case depends on the go signal too. However, for strong signal levels, as it is shown in figure 3.15A, the activity crosses the threshold within a few hundreds of milliseconds after the stimulus onset which begs the question of why the decision making in the second experiment has to wait until the go signal comes up? If the ramping activity predicts the decision, then why the activity does not cross the threshold much earlier even for high signal levels? Monkeys could have made decision much earlier and kept it (only one bit of information) in memory until the go signal appeared on the screen. Therefore, although this theory is partly supported by the data, there are experiments that the theory needs to address.

In addition, this theory does not specify how the raw data (samples) are processed to provide a scalar that decides whether the decision variable should move towards one threshold or the other. It assumes that this transformation from the raw data to scalar has already been accomplished. It doesn’t address what mechanism(s) are responsible for keeping the earlier information in the memory nor does it address how they combine with the information that arrive later. In contrast, the change-based method addresses the problem of how these operations can be neurally implemented

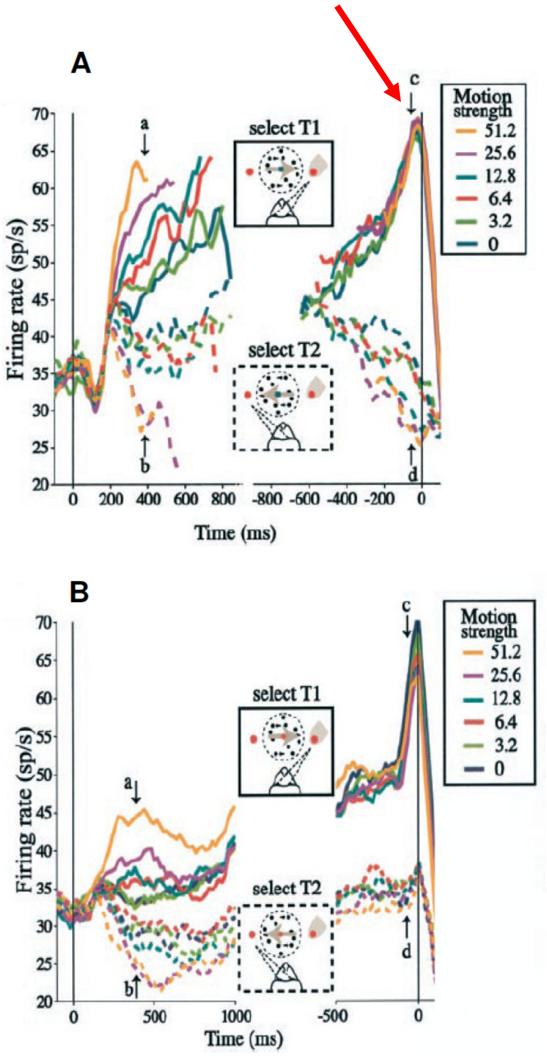

Figure 3.15 LIP activity and drift-diffusion model. **A**) Neural activity of a typical LIP neuron in a reaction time task for different signal levels. A random dot stimulus appears on the screen at time zero and a fraction of dots move coherently in one direction (signal) while the rest of the dots move randomly (noise). The duration of the stimulus is one second and monkeys are allowed to make a saccade any time after the stimulus onset. Monkeys are trained to make a saccade towards right or left depending on the direction of the motion of the signal as quickly as possible. Curves on the left show the activity of an LIP neuron for different levels of signal aligned by the stimulus onset while curves on the right are aligned by reaction time. It seems that decision is made when the activity crosses a threshold marked by a red arrow. **B**) Neural activity of a typical neuron in a fixed duration paradigm. This is similar to the experiment mentioned in (A) except that the monkey is not allowed to make a saccade until a go signal

appears on the screen at least one second after the stimulus disappears. adapted from (Roitman and Shadlen, 2002).

by a recurrent network. Machen et al (Machens et al., 2005) addressed this problem within the attractor-based framework; however, they needed the network to be fine tuned to perform the task and furthermore the task they addressed was to combine only two scalars (representing the frequency of two successively presented tactile stimuli). Change-based coding approaches the decision making problem from a population coding point of view and suggests a new way for a population of neurons in an interconnected network to jointly make a decision. Therefore change-based method seems to address different aspects of decision making than those that are addressed by accumulation to bound theory.

Change-based method is perhaps the simplest way of taking the temporal structure of the neural activity in to account in order to process information and represent decisions. We have shown that even such a simple theory successfully addresses several key problems, like discrimination in the presence of invariance and accumulation of evidence over time. We have used threshold-linear activation function for the model neurons. However, for general nonlinearity, since both the recurrent weight and the Jacobian are centrosymmetric, the effective weight matrix (which is their multiplication) is also centrosymmetric and the eigenvectors of the centrosymmetric matrices are always either odd or even (Andrew, 1973). Although the change-based method works optimally with threshold-linear activation function, it is interesting from a theoretical standpoint to study the range of nonlinearities with which the change-based method works optimally and also the validity of the linear analysis in those cases (to see whether this is the first order term (linear term) in the Taylor expansion that encodes the relevant information or higher order terms). This remains an interesting future direction.

3.6 Summary

In this chapter we studied the change-based method from a theoretical point of view. We linearized the network around its initial state and studied the properties of the resulting linear system. Our goal was to understand why the change-based method solves the bisection task. We studied the weight matrix associated with the linear sub-network and we found that one of its eigenvectors (we referred to it as effective eigenvector) plays a key role in change-based processing. In fact, it accomplishes both steps necessary for optimal performance; not only the effective eigenvector accumulates the temporally distributed information optimally, but also it controls the initial movement of the CoM. Its time constant is the highest among all the eigenvectors of the reference weight matrix and combines the samples that are tens of milliseconds away from each other in an optimal way. Its orthogonality to the small random jitters makes it invariant to such perturbations. And finally, since its time constant is the highest among all the odd eigenvectors, it controls the initial movement of the center of mass of the neural activity. In sum, the linear analysis reveals the key concepts underlying the optimality of the change-based processing which is important for extending this method to broader class of computations.

Appendix 3.A: Another example weight matrix with near optimal performance

Figure 3.16 shows another weight with optimal performance along with its reference weight matrix and eigenspectrum and effective eigenvector.

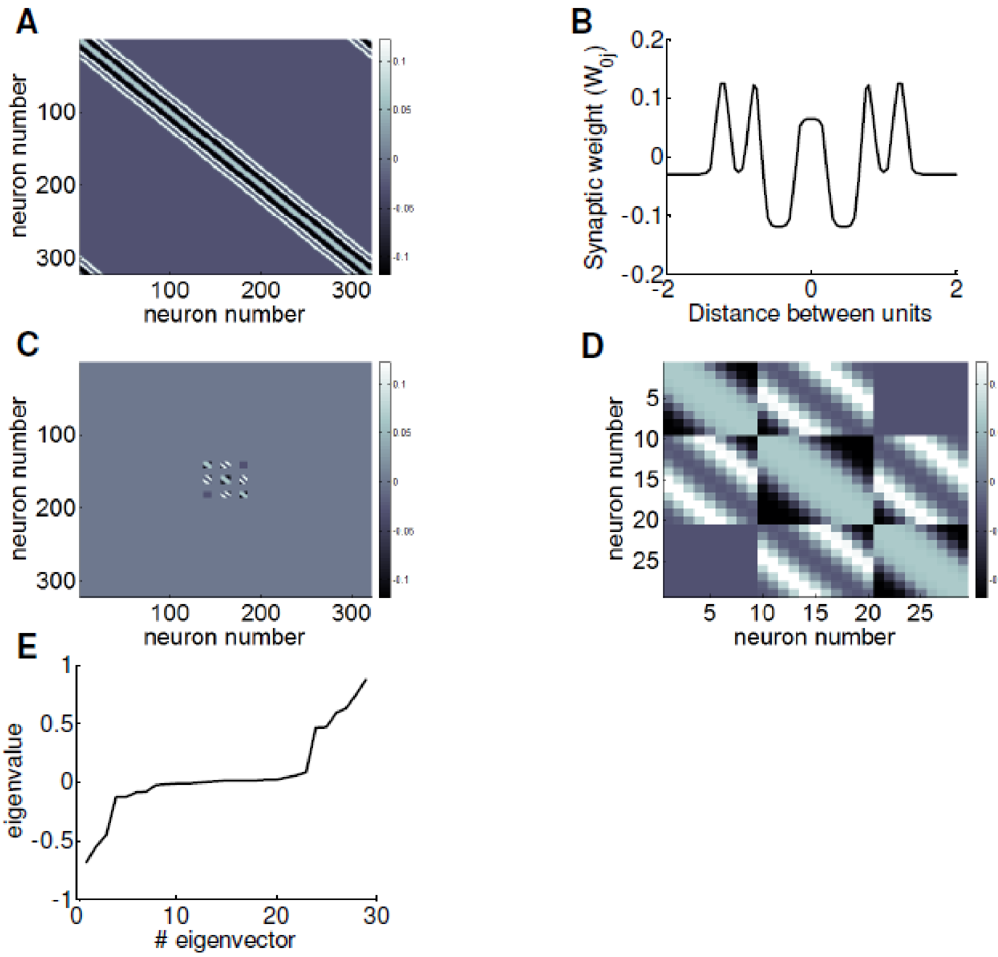

Figure 3.16 Effective eigenvector of another optimal weight matrix. **A**) Synaptic weight matrix of another recurrent network which is symmetric and position invariant. **B**) Central row of the weight matrix shown in (A). **C**) Reference weight matrix associated with green synapses in figure 3.4. Synapses that are not green are set to zero. **D**) Reference weight matrix in the new space where only the linear network is considered and the rest of the network is ignored. **E**) The eigenspectrum of the reference weight matrix

Chapter 4

Change-based versus attractor-based inference

4.1 Introduction

In the previous chapter, we studied the change-based method by linearizing network activity around the initial state of the network immediately after stimulus onset. This was motivated by the fact that change-based method operates optimally while the state of the network is far from the attractor and near its initial state. On the other hand, as we discussed in the previous chapters, the attractor-based method has been shown to operate optimally for a class of estimation tasks near attractor manifolds. In this chapter, we address the relation between the change-based and the attractor-based methods. Change-based method performs a broad class of discrimination tasks in the presence of the invariance optimally (as we will see in chapter 5, learning is possible for the change-based method and we show other example tasks that can be learned) and the attractor-based method is shown to perform an orientation estimation task near ideal observer level. It is natural to ask if there is any relation between the two methods since one is about estimation and the other one is about discrimination.

However, since in all the example that we have seen and also according to the linear analysis that we presented in the previous chapter, the change-based method works far from attractors while the attractor-based method works near attractors, it is not clear how the two methods can be compared. To this end, in this chapter, we take a first step towards analyzing the change-based method near the attractor by considering a subtly different discrimination task from the one we considered in the previous chapters. In the new task, change-based readout operates near-optimally in the linear regime near the network's line attractor and since this very same, linearizable, regime has previously been used to elucidate the workings of estimation in conventional attractor-based readout (Pouget et al., 1998; Deneve et al., 2001), we would be able to examine the relation between change-based and attractor-based methods. The similarity between the discrimination task that we consider here and the conventional estimation task used to demonstrate the optimality of the conventional attractor-based method, together with the fact that both readout methods (change-based and attractor-based) here operate in the linear regime, allows us to relate change-based discrimination and attractor-based estimation precisely. We show that change-based discrimination can only be successful near the attractor since attractor-based estimation is slightly flawed, an analysis that we confirm directly by exhibiting a trade-off between the two. In other words, in the near attractor regime, if the attractor-based estimation had been perfect for estimation task, then the change-based method would have performed the discrimination task at chance level. The excellent performance of change-based discrimination shows that this is not the case. We also show that the linear analysis is able to predict the performance of a broader class of networks accurately.

In the following, we describe discrimination and estimation tasks, the recurrent network, and change- and attractor-based inference in detail. Then we provide simulation results that confirm the near-optimality of our network at both estimation and discrimination in linearizable and non-linear regimes. Next, we analyze the network in the linear regime and show how change-based computation works. Finally, we consider broader issues raised by our analyses.

4.2 Methods

4.2.1 The model and the task

The recurrent network that we consider is an abstract model of topographically arranged processing in a retinotopic early visual area. It is a (non-circular) version of one of the best studied line attractor networks which has been shown to perform estimation from noisy input (in this case of retinotopic location) at a level near to that of an ideal observer (Pouget et al., 1998). Recurrent interactions include topographically-local interactions, and divisive normalization (Carandini and Heeger, 1994; Pouget et al., 1998).

The network consists of N units arranged topographically on a line; we consider stimuli in just the central portion, and thereby avoid edge effects. These units form a population code for visual inputs. For simplicity, we consider all the inputs to be short vertical bars, but inference is about the absolute or relative locations of the bars, and not about their orientations (thus we assume that all units are vertically tuned). Units are selective to different locations in the input, with the preferred location of unit i being denoted by x_i (we also use x_i to identify unit i in formulae and plots).

We consider two different tasks for the network. The main task involves discriminating between two classes of input, and is solved by change-based readout. For this, two very nearby bars are presented anywhere on the line, one having a lower contrast than the other. The task is to decide whether the low contrast bar is to the left or the right of the high contrast bar (see figure 4.1A). This first task is solved using change-based readout. The second task provides the analytical backdrop for the first task. In this, only the high contrast bar is presented and the problem is to estimate its location on the topographic array. This is solved using conventional, attractor-based readout. Note that the inputs for both tasks are presented to the very same network, it is just that depending on the task, different readout mechanisms are involved. Also note the relationship between the two tasks: the variable that must be estimated in the estimation task (location of the high contrast bar) is the invariant dimension in the discrimination task. In other words, the variable that is important for the discrimination task is the relative location of the two bars and not the absolute location of the high contrast bar.

4.2.2 Discrimination task

4.2.2.1 Task

The visual discrimination task involves reporting whether a low contrast bar, which we refer to as the signal, is slightly to the left or the right of a high contrast bar, which we call the carrier (see figure 4.1A). The carrier is presented at location y and the signal

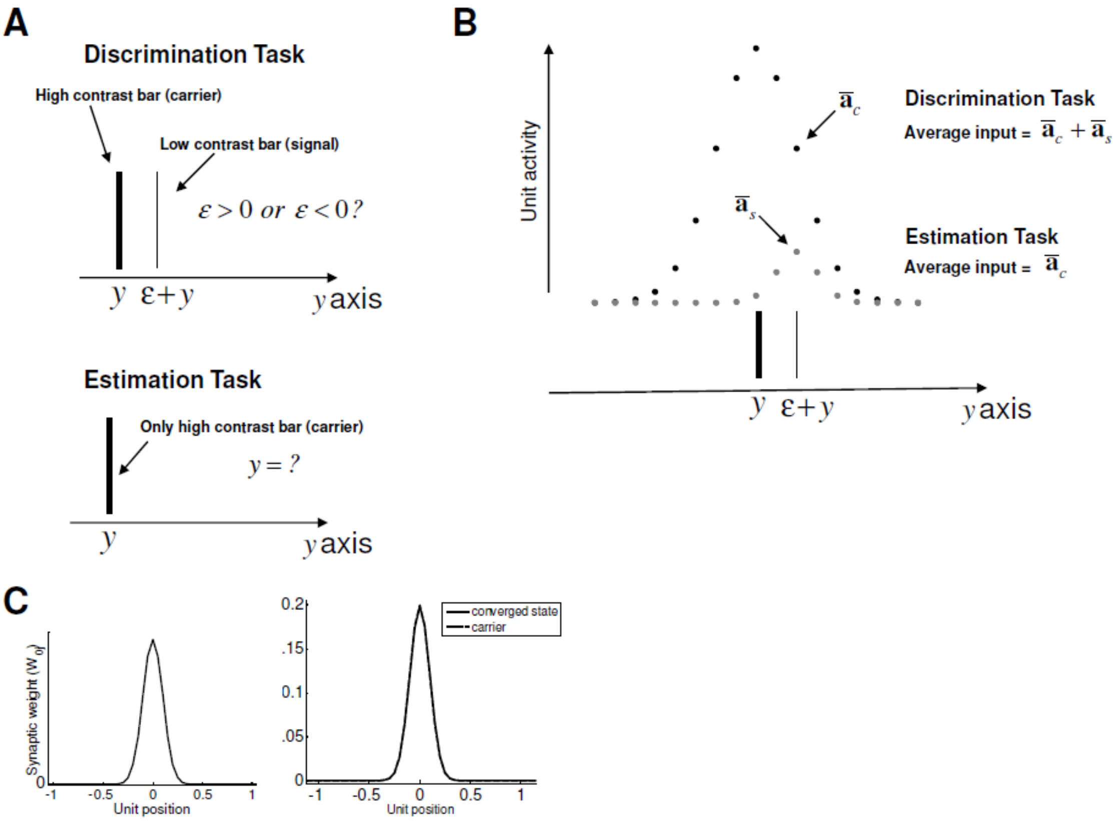

Figure 4.1. The discrimination and estimation tasks. **A**) Top: Discrimination Task. Two bars are presented simultaneously to the network, one with high contrast (called the carrier) and one with low contrast (the signal). The task is to decide if the signal is to the left or to the right of the carrier. Bottom: Estimation Task. Only the high contrast bar is presented to the network and the task is to estimate its location. **B**) On average, each bar generates a Gaussian hill of activity with different heights reflecting their different contrasts. Note that units are arranged on the y axis. The high contrast bar (carrier) generates the average hill of activity $\bar{a}_c$ and the low contrast bar (signal) generates the average hill of activity $\bar{a}_s$. For discrimination task, on each trial, the actual input is a noisy version of the linear sum of the two hills of activity $\bar{a}_c + \bar{a}_s$. For estimation task, on each trial, the actual input to the network is a noisy version of the average hill of activity elicited by the carrier, $\bar{a}_c$. (C) Left: The synaptic weight between the unit at position zero and other units. The weight matrix is translation invariant (away from the boundary). Right: the carrier and the converged state of the network (note that both patterns have very nearly the same form).

is presented at location $\varepsilon + y$, and so the task is to decide if $\varepsilon > 0$ or $\varepsilon < 0$ (see figure 4.1A). Note that y represents the invariant dimension.

4.2.2.2 Model

We assume that the bars activate the units additively. If the carrier is presented by itself to the network at location y, then the mean activity of unit i is:

$$\overline{a}_{ic} = \frac{e^{-\frac{(y-x_i)^2}{2\sigma^2}}}{H} \tag{4.1}$$

where in $\overline{a}_{ic}$, the first index, i, represents the unit number (between 1 and N), and the second index, c, stands for "carrier"; σ is the width of the smooth hill activated by the carrier; and $H = \sum_i e^{-\frac{(y-x_i)^2}{2\sigma^2}}$, is the summed weight of this hill. H normalizes the mean activity associated with the carrier so that the mean activity associated with the carrier is very similar to the form of the converged state of the recurrent network (as we explain below). Similarly, the average activity received by unit i when the signal is presented by itself at location $\varepsilon + y$ is:

$$\overline{a}_{is} = \frac{\zeta e^{-\frac{((\varepsilon+y)-x_i)^2}{2\eta^2\sigma^2}}}{H} \tag{4.2}$$

where in $\overline{a}_{is}$, the first index, i, represents the unit number (between 1 and N), and the second index, s, stands for "signal", η scales the width, and ζ scales the contrast or strength of the signal. We mostly use a very small value for ζ (i.e., a weak signal) to make discrimination challenging; this also permits linearization of the network around the activity associated with the carrier. Since we assume that the effects of the bars sum linearly, the overall average activity received by neuron i is:

$$\overline{a}_i = \overline{a}_{ic} + \overline{a}_{is} \tag{4.3}$$

These average activities are cartooned in figure 4.1B. We consider two noise models: scaled Poisson noise and Gaussian mean dependent noise. For both noise models, the actual input to the network ($\boldsymbol{a} = (a_1, a_2, ..., a_N)$), is generated in two stages. The first stage controls the strength of the noise while the second stage is necessary in order to have $\langle \boldsymbol{a} \rangle = \overline{\boldsymbol{a}}_c$ (in the absence of the signal) where $\langle \boldsymbol{a} \rangle$ denotes average over infinite random draws of $\boldsymbol{a}$ and $\overline{\boldsymbol{a}}_c = (\overline{a}_{1c}, \overline{a}_{2c}, ..., \overline{a}_{Nc})$. This way of generating the inputs is for analytical convenience, and it also allows a more straightforward comparison between the two noise models. Although we scale the strength of the noise (to limit its non-linear effects), the discrimination and estimation performance of the networks are always compared with those of ideal observers experiencing exactly the same patterns.

For the Poisson noise model, we first generate vector $\boldsymbol{a}' = (a'_1, a'_2, ..., a'_N)$ according to a Poisson distribution with mean $q\lambda\overline{\mathbf{a}}$ (the reason to decompose the coefficient in to q and λ is to simplify the comparison between Poisson and Gaussian noise models):

$$P(a'_i \mid q\lambda\overline{a}_i) = \frac{e^{-q\lambda\overline{a}_i}}{(a'_i)!}(q\lambda\overline{a}_i)^{a'_i} \tag{4.4}$$

where $1/q$ controls the strength of the noise. Note that the elements of vector $\boldsymbol{a}'$ are mutually independent. The mean and standard deviation of a'_i are $q\lambda\overline{a}_i$ and $\sqrt{q\lambda\overline{a}_i}$ respectively. Thus, the signal to noise ratio is $\sqrt{q\lambda\overline{a}_i}$, which increases as q increases.

Since the dynamics of the network in any case involve normalization (Carandini and Heeger, 1994), the second step is consider the actual input to the network to be

$$\boldsymbol{a} = \frac{\boldsymbol{a}'}{q\lambda} \tag{4.5}$$

For the Gaussian noise model, in the first stage we generate a vector $\boldsymbol{a}' = (a'_1, a'_2, ..., a'_N)$ from a Gaussian distribution with mean $\lambda\overline{\boldsymbol{a}}$ and variance $\frac{\lambda\overline{\boldsymbol{a}}}{q}$:

$$P\left(a'_i \,|\, \frac{\lambda\overline{a}_i}{q}\right) = \frac{1}{\sqrt{2\pi\left(\frac{\lambda\overline{a}_i}{q}\right)}} exp\left(-\frac{(a'_i - \lambda\overline{a}_i)^2}{2\left(\frac{\lambda\overline{a}_i}{q}\right)}\right) \tag{4.6}$$

where $\frac{1}{q}$ determines the strength of the noise. Again, the elements of vector $\boldsymbol{a}'$ are independent. For the Gaussian noise case, in the second step the actual input to the network is

$$\boldsymbol{a} = \frac{\boldsymbol{a}'}{\lambda} \tag{4.7}$$

As we mentioned above, defining the input to the network by equations 4.5 (for Poisson noise) and 4.7 (for Gaussian noise) ensures that $\langle \boldsymbol{a} \rangle = \overline{\boldsymbol{a}}_c$ (in the absence of the signal (the low contrast bar)) for both noise models. Note that the scaled Poisson and Gaussian noise models have the same signal to noise ratio. We consider the following discrete dynamics for the network:

$$\boldsymbol{u}^{n+1} = \boldsymbol{f}(\boldsymbol{W}\boldsymbol{u}^n) \tag{4.8}$$

where n indicates the iteration number, $\boldsymbol{u}^{(n)} = \left(u_1^{(n)}, u_2^{(n)}, ..., u_N^{(n)}\right)$ and $\boldsymbol{u}^{(0)} = \boldsymbol{a}$ defines the input to the network (equations 4.5 and 4.7 for Poisson and Gaussian noise, respectively). We used discrete dynamics here because the goal here was to find a (handcrafted) weight matrix $\boldsymbol{W}$ that resulted in a converged state that had a form very similar to the carrier and it also performed both estimation and discrimination tasks optimally. The discrete dynamics made it possible for us to find such a weight matrix. The weights $\boldsymbol{W}$ are translation invariant and their form (see appendix 4.A) is shown in figure 4.1C left. We define the weights $\boldsymbol{W}$ and the dynamics of the network $\boldsymbol{f}(.)$ such that if $\boldsymbol{u}^{(0)} = \overline{\boldsymbol{a}}_c$ then the converged state of the network, $\boldsymbol{u}^{(\infty)}$, and the carrier, $\overline{\boldsymbol{a}}_c$, are nearly the same (figure 4.1C left). In general, the weights $\boldsymbol{W}$ and the dynamics of the network $\boldsymbol{f}(.)$ are defined such that $\boldsymbol{u}^{(\infty)}$ and $\overline{\boldsymbol{a}}_c$ have nearly the same form (figure 4.1C left), but they might not be at the same location, because of the discrimination signal and the noise that corrupt $\overline{\boldsymbol{a}}_c$.

We follow (Deneve et al., 2001) in defining the network's non-linear activation function $\boldsymbol{f}(.) = (f_1(.), f_2(.), ..., f_N(.))$ as the squaring, normalizing non-linearity:

$$f_i(\boldsymbol{r}) = \frac{r_i^2}{\sum_j r_j^2} \tag{4.9}$$

where $\boldsymbol{r} = (r_1, r_2, ..., r_N)$ is an arbitrary vector. Normalization has been shown to realize a particularly convenient (Deneve et al., 1999; Wu and Amari, 2005) and neurobiologically relevant (Heeger et al., 1996; Carandini et al., 1997) form of attractor network. However, change-based readout does not depend on this; in our earlier paper (Moazzezi and Dayan, 2008), we considered a threshold linear hinge function. Since the weights are symmetric and translation invariant (at least in the network's central regime), there is at least one, one-dimensional, line of attractor states. These states are smooth unimodal bumps, which have very nearly the same form as the mean activity associated with the carrier (figure 4.1C right).

4.2.2.3 Change-based discrimination

According to change-based readout, the decision about the location of the signal relative to the carrier depends on whether the Centre of Mass (CoM) of the neural activity in the network moves to the left or to the right. The Centre of Mass (CoM) of the neural activity $\boldsymbol{u}^{(n)}$at iteration n is defined as:

$$\mu\left(\boldsymbol{u}^{(n)}\right)=\frac{\sum_i u_i^{(n)} x_i}{\sum_i u_i^{(n)}} \tag{4.10}$$

For this discrimination task the decision about the sign of ε (which is the distance between signal and carrier, figure 4.1A) is based on the sign of $\mu\left(\boldsymbol{u}^{(\infty)}\right)-\mu\left(\boldsymbol{u}^{(0)}\right)$ (which is equal to $\mu\left(\boldsymbol{u}^{(\infty)}\right)-\mu(\boldsymbol{a})$ given $\boldsymbol{u}^{(0)}=\boldsymbol{a}$). This is an example of change-based readout. Compared with our previous application of this method, which we described in the introduction, here we measure the change between start and end, rather than two intermediate times. However, this does not materially affect the network's performance on the task.

4.2.3 Estimation task

In the estimation task, the carrier is presented by itself, and the task is to estimate its location, y. Therefore the average activity is generated from the carrier only, with the signal being absent, that is $\overline{a}_i = \overline{a}_{ic}$ for all i. The noisy input activity $\boldsymbol{a}$ is generated from $\overline{\boldsymbol{a}}$, as for discrimination (equations 4.4 and 4.5 for Poisson noise and 4.6 and 4.7 for Gaussian noise). The network performs the task of estimating y from this noisy input, by mapping the input to the line attractor and reporting the location on the attractor of the converged state. As mentioned above, we call this attractor-based readout. Since the converged state is smooth, noise-free and unimodal, its location can be accurately characterized by its centre of mass, i.e. $\mu\left(\boldsymbol{u}^{(\infty)}\right)$.

4.2.4 Parameters

The main parameters were set to $x_{i+1}-x_i=.05$, $N=81$, $\sigma=.1$, $\lambda=20H$, $\eta=.5$ creating a smooth population code. We explored two regimes for the tasks: one which enabled linearization, with very small signal and noise, ($\zeta=.1, q=100$); and the other to show that performance remains good in a more realistic, non-linear, regime, with large signal and noise ($\zeta = 1, q = 1$). The network is found to have converged by 10 iterations.

4.3 Results

4.3.1 Performance of the network and the ideal observer

We measured the performance of the ideal observer and change-based readout for the discrimination task for weak (scaled) Poisson and Gaussian noise. Since the signal is also weak in this regime, the ideal observer, which knows the height and width of both the carrier and the signal and the full distributions of likelihoods and priors (though not y), was not perfect. Figures 4.2A and 4.2B compare the performance levels of the network for the discrimination task for both noise models with those of the corresponding ideal observers for a range of ε. The network is evidently near optimal. Doubling the number of units by increasing their density twofold led to a performance that improved in line with that of the ideal observer. This indicates that high quality inference is a stable characteristic of the network.

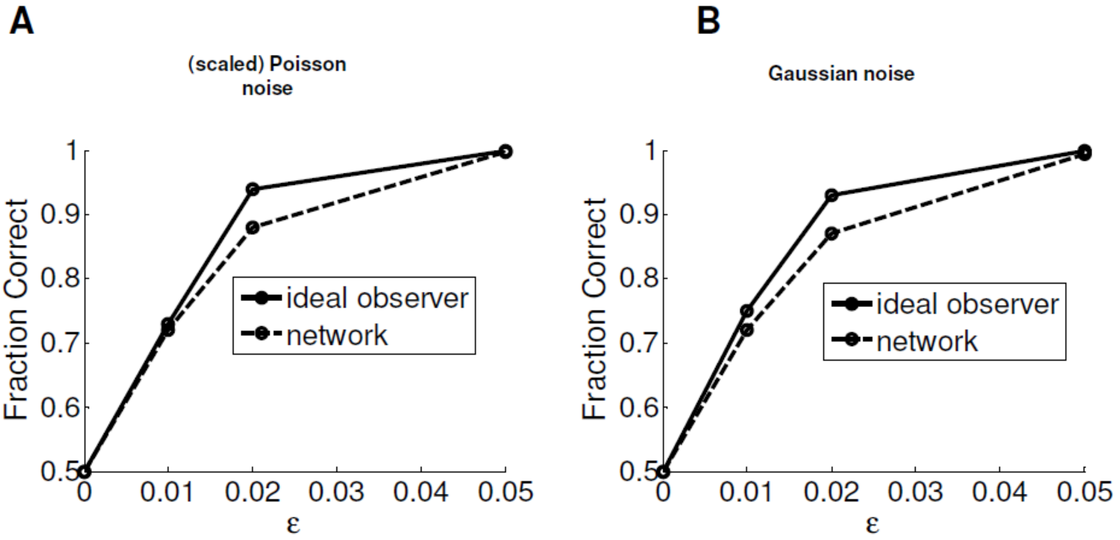

Figure 4.2. Performance of the network and the ideal observer. **A**) Performance level of the ideal observer and the network as a function of ε for Poisson noise. **B**) same as (A) except that the noise is mean dependent Gaussian. In both cases the network's performance is near optimal.

Attractor-based readout was also near optimal at estimation for both Poisson and Gaussian noise, having a standard deviation of only about 1.1 times that of the ideal observer. This same ratio was preserved when we increased the density of the units in the population code eightfold, indicating, as expected from previous studies of estimation, that such good performance is a stable characteristic of such networks. Note that for the Poisson noise, the ideal observer has a simple interpretation as the CoM of the input to the units ($\boldsymbol{a}$). This immediately follows from the fact that the form of the carrier is Gaussian. The same is approximately true for low levels of Gaussian noise.

4.3.2 Linearization of the network

Figure 4.3A shows the carrier, an example input ($\boldsymbol{a}$) generated from the carrier plus signal (equations 4.4 and 4.5 for Poisson noise and 4.6 and 4.7 for Gaussian noise) and the output of the network after convergence. Figure 4.3B shows that the difference between the converged state and the example input is very small relative to their magnitudes, and motivates an approximation to the action of the network about a single point in state space (the carrier at one particular location) in terms of a linear filter $\boldsymbol{J}_{cb}$. This filter, which is shown in Figure 4.3C, predicts the change $\delta\mu_{cb}$ in the CoM of the neural activity as:

$$\delta\mu_{cb} = \mu\left(\boldsymbol{u}^{(\infty)}\right) - \mu(\boldsymbol{a}) \approx \boldsymbol{J}_{cb}^{T}.(\boldsymbol{a} - \overline{\boldsymbol{a}}_{c}) \tag{4.11}$$

where $\overline{\boldsymbol{a}}_c = (\overline{a}_{1c}, \overline{a}_{2c}, ..., \overline{a}_{Nc})$ is the carrier, $\boldsymbol{a} = (a_1, a_2, ..., a_N)$ is the input to the network and $\boldsymbol{u}^{(\infty)} = \left(u_1^{(\infty)}, u_2^{(\infty)}, ..., u_N^{(\infty)}\right)$ is the converged state of the network. Note that the difference $\boldsymbol{a} - \overline{\boldsymbol{a}}_c$ in equation 4.11 includes the signal ($\overline{\boldsymbol{a}}_s$) for the discrimination task, where $\overline{\boldsymbol{a}}_s$ is defined as: $\overline{\boldsymbol{a}}_s = (\overline{a}_{1s}, \overline{a}_{2s}, ..., \overline{a}_{Ns})$.

We can understand equation 4.11 intuitively (figure 4.3D). $\delta\mu_{cb}$ is the amount the CoM changes as the network evolves from its initial state $\boldsymbol{u}^{(0)} = \boldsymbol{a}$ to its final state $\boldsymbol{u}^{(\infty)}$. The change in CoM of the network activity between the input and the converged

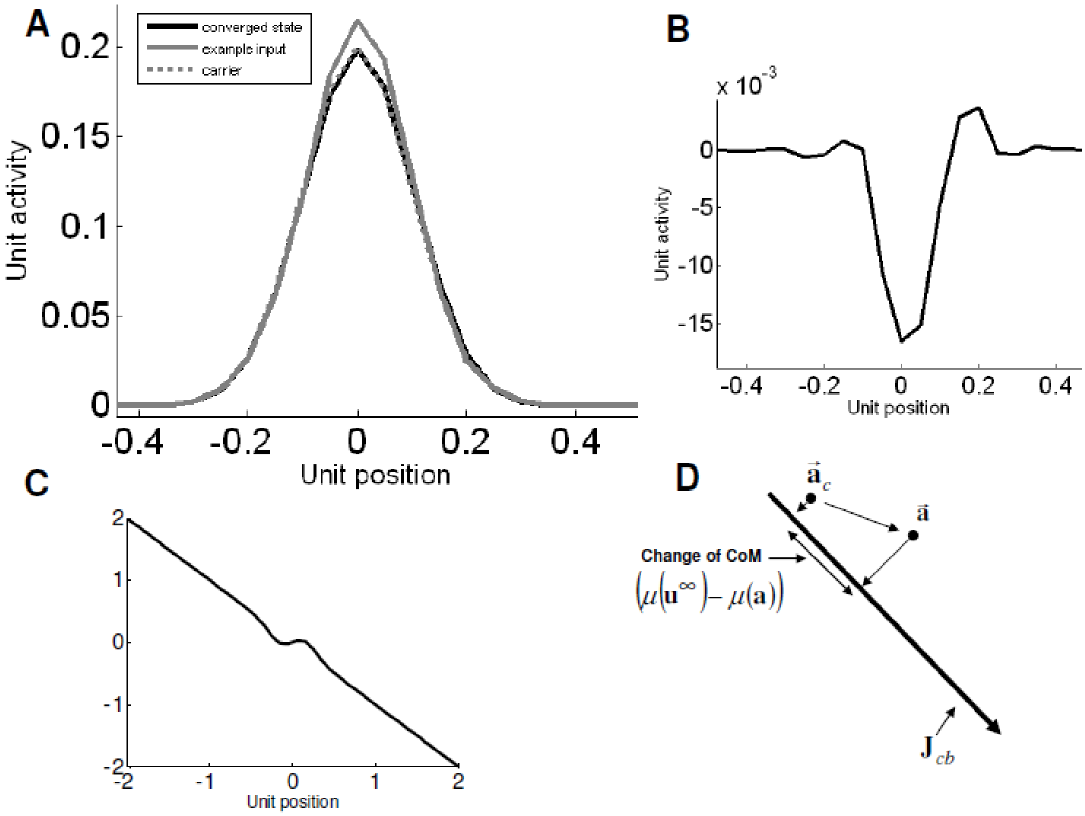

Figure 4.3 Network attractor and linearization. **A**) Carrier (dashed grey line), an example input (solid grey line) and output (solid black line) of the network. **B**) The difference between the input and the output of the network, which is very small relative to the size of the input. **C**) The equivalent linear filter J_{cb} that predicts the change in the centre of mass. This was obtained by linearizing about the carrier (equation 4.11). Unit 0 on the X axis represents the center of the carrier and the Y axis represents J_{cb} for different units (equation 4.11). **D**) Schematic description of equation 4.11. The change in CoM of the network activity between the input and the converged state $(\mu(u^{\infty}) - \mu(a))$ is predicted based on the difference of the input pattern and the activity associated with the carrier $(a - \bar{a}_c)$. The difference of the carrier ($\bar{a}_c$) and example input (a) is projected on to the change-based (linear) filter (J_{cb}) and its magnitude and sign represents the magnitude and sign of the change in CoM which is $(\mu(u^{\infty}) - \mu(a))$.

state $\left(\mu\left(\boldsymbol{u}^{(\infty)}\right)-\mu(\boldsymbol{a})\right)$ is predicted based on the difference of the input pattern and the mean activity pattern associated with the carrier, i.e. $(\boldsymbol{a}-\overline{\boldsymbol{a}}_c)$. $\boldsymbol{a}$ is a noisy

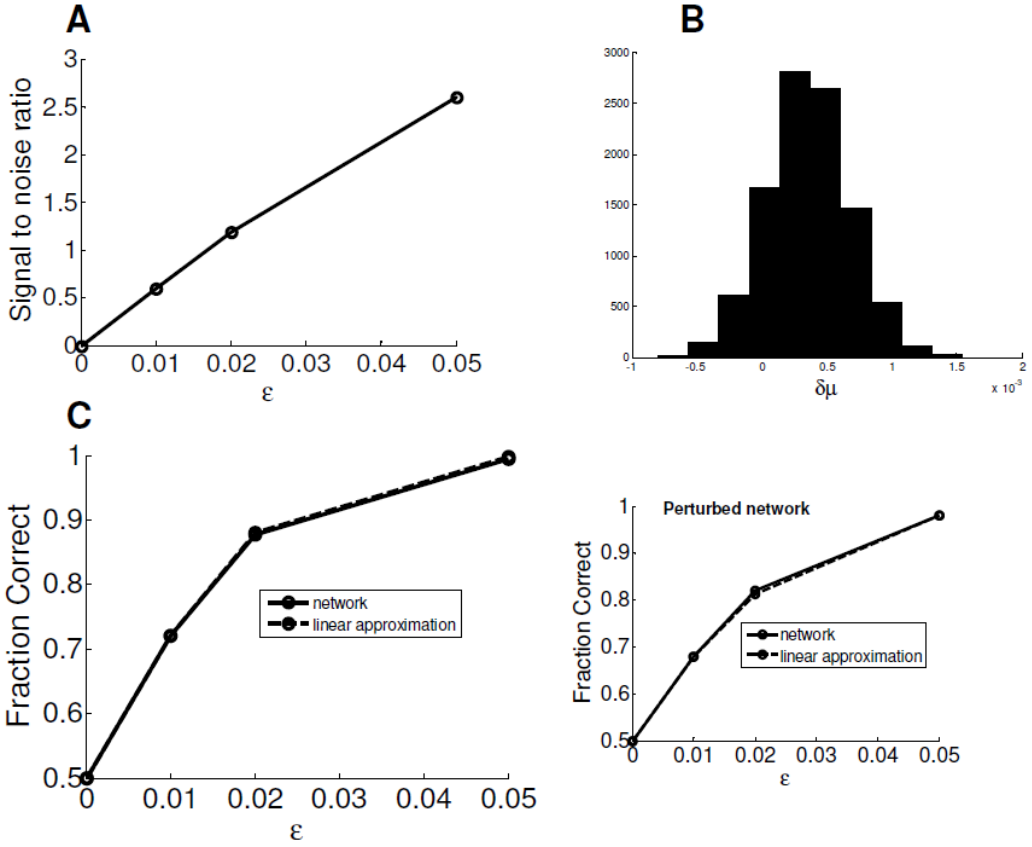

Figure 4.4. Linear performance. **A**) The signal to noise ratio (in the change of CoM) predicted by the linear filter that was shown in figure 4.3C as a function of ε. **B**) An example histogram for the linear prediction of the change in the CoM for ε =.02. This is approximately Gaussian because it can be written as a sum of a moderate number of independent random variables. **C**) Performance level of the network predicted by the linear approximation and the actual performance level of the network. The inset shows the same quantities for the perturbed network (see figure 4.7) showing that the linear approximation works very well even when the initial state of the network is far from the attractor and therefore the input-output transformation is nonlinear.

version of the underlying, smooth, mean activity pattern associated with the carrier $\overline{\boldsymbol{a}}_c+\overline{\boldsymbol{a}}_s$, and the change in CoM $\left(\mu\left(\boldsymbol{u}^{(\infty)}\right)-\mu(\boldsymbol{a})\right)$ is approximated as being linearly related to the difference of the patterns of the input and the carrier. Note that for the case that the attractor states and the carrier have the same form, it is straightforward to derive the linear filter $\boldsymbol{J}_{cb}$ from the Jacobian of the network (see appendix 4.C). We define the signal to noise ratio associated with the filter as

$$S/N = \frac{\left|\boldsymbol{J}_{cb}^{T}\overline{\boldsymbol{a}}_{s}\right|}{\sqrt{\boldsymbol{J}_{cb}^{T}\boldsymbol{\Sigma}\boldsymbol{J}_{cb}}} \tag{4.12}$$

where $\boldsymbol{\Sigma}$ is the noise covariance matrix of $\boldsymbol{a}$. Note that $\overline{\boldsymbol{a}}_{s}$ is a function of ε. Figure 4.4A shows the predicted signal to noise ratios for the filter of Figure 4.3C over the relevant range of ε. Figure 4.4B shows an example histogram for the changes of the CoM predicted by the linear filter. Since it is the sum of independent random variables, the central limit theorem suggests that the distribution of the change will be roughly Gaussian. One can therefore use the S/N to predict the performance of change-based readout. Figure 4.4C confirms the empirical rectitude of this analysis.

4.3.3 Discrimination and estimation

As well as predicting its performance correctly, the linearization also clarifies the relationship between change-based discrimination and attractor-based estimation (figure 4.5). This is predicated on the facts that the CoM itself is the optimal estimator for the case of Poisson noise, and the approximately optimal estimator in the low noise regime for Gaussian noise (see appendix 4.B). We will show that, in a formal sense, change-based discrimination exactly exploits the sub-optimality of attractor-based estimation.

First, consider the case that $y = 0$ and the carrier is presented by itself (the estimation task). This implies that $\overline{\boldsymbol{a}} = \overline{\boldsymbol{a}}_{c}$, and therefore that we can write $\boldsymbol{a} = \overline{\boldsymbol{a}}_{c} + \boldsymbol{n}$ where $\boldsymbol{n}$ is either Poisson or Gaussian noise. Again, the optimal estimator of y can be linearized as:

$$\mu_{ml} \approx \boldsymbol{J}_{ml}^{T}\left(\boldsymbol{a} - \overline{\boldsymbol{a}}_{c}\right) = \boldsymbol{J}_{ml}^{T}\left(\boldsymbol{n}\right) \tag{4.13}$$

where μ_{ml} is the optimal estimator of y given the input pattern $\boldsymbol{a}$ and equation 4.13 predicts this optimal estimation as a linear function of the difference of the input and carrier patterns. The index "ml" stands for "maximum likelihood" because the optimal estimator is the maximum likelihood estimator. The linear filter $\boldsymbol{J}_{ml}$ is derived in appendix 4.B (dashed curve in figure 4.6A). As we mentioned in section 4.3.1, for Poisson noise and low levels of Gaussian noise μ_{ml} is the CoM of input pattern $\boldsymbol{a}$.

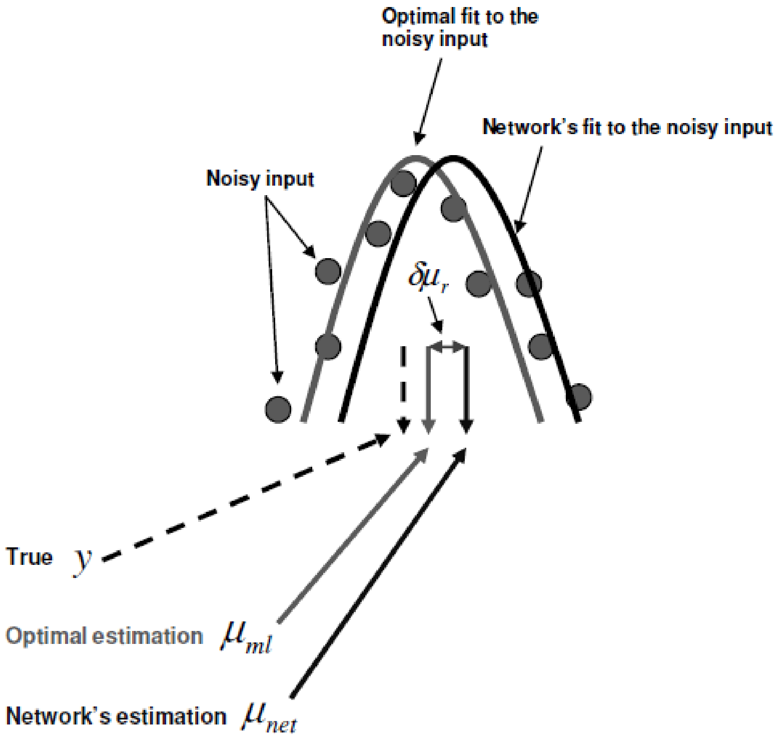
Optimal fit to the noisy input
Network's fit to the noisy input
Noisy input
$\delta\mu_r$
True y
Optimal estimation μ_{ml}
Network's estimation μ_{net}

Figure 4.5 Change-based and attractor-based readout. This schematic shows a case in which noisy activity (circles) came from a true value of $y = 0$. The optimal ML estimate of y is μ_{ml} , which amounts to fitting the shape of the mean activity associated with the carrier to the actual activity (weighting errors by an amount that has to do with the noise model). The network's estimate is μ_{net} , whose position is determined by relaxation to the attractor. The difference $\delta\mu_r = \mu_{net} - \mu_{ml}$ underlies change-based readout.

If we denote the linear filter associated with attractor-based method by $\boldsymbol{J}_{net}$ (the dotted curve in figure 4.6A), including the read-out, then the network's estimate of y is

$$\mu_{net} \approx \boldsymbol{J}_{net}^{T}(\boldsymbol{a} - \bar{\boldsymbol{a}}_c) = \boldsymbol{J}_{net}^{T}(\boldsymbol{n}) \tag{4.14}$$

Note that the network's estimate of y is the CoM of the converged state of the network. Equation 4.14 predicts the network's estimate as a linear function of the difference between the input and carrier patterns. Since the final pattern (converged state) is symmetric, μ_{net} (figure 4.5) can also be interpreted as the CoM of the converged state of the network. Attractor-based readout would therefore be optimal if $\boldsymbol{J}_{net} = \boldsymbol{J}_{ml}$, i.e. if the optimal estimate and the network's estimate are the same at every trial ($\mu_{net} = \mu_{ml}$). Given that $\boldsymbol{J}_{net} \neq \boldsymbol{J}_{ml}$ (figure 4.6A) there is a difference between the network's estimate and the optimal estimate $\delta\mu_r = \mu_{net} - \mu_{ml}$. We can therefore write $\delta\mu_r \approx \boldsymbol{J}_r^T \boldsymbol{n}$ where $\boldsymbol{J}_r = \boldsymbol{J}_{net} - \boldsymbol{J}_{ml}$.

This difference is crucial for change-based discrimination. Consider what happens when the signal is presented along with the carrier (for the discrimination task). Since μ_{ml} is the CoM of the input pattern and μ_{net} is the CoM of the converged state of the network, we have that $\delta\mu_r$ is the change in the CoM of the neural activity, which is exactly what change-based readout measures. Therefore, $\boldsymbol{J}_r$ is the linear filter associated with the change in CoM, that is, $\boldsymbol{J}_r = \boldsymbol{J}_{cb}$. If the network was a perfect estimator ($\boldsymbol{J}_{net} = \boldsymbol{J}_{ml}$) then $\boldsymbol{J}_{cb} = \boldsymbol{J}_r = \boldsymbol{J}_{net} - \boldsymbol{J}_{ml} = \boldsymbol{0}$ and therefore by equation 4.11, $\delta\mu_{cb} = 0$ and the performance of the change-based method for discrimination would be at chance level.

Another way of understanding this result is that if the network were to be an optimal estimator, its converged output state would have to have the same CoM as its input, since maximum likelihood estimation is based on the input CoM. However, if this were true, then change-based readout would report 0, since the CoM would not move. Thus, a necessary condition for change-based readout to work is that $\boldsymbol{J}_{net} \neq \boldsymbol{J}_{ml}$. In appendix 4.B we derive the form of the optimal filter $\boldsymbol{J}_{dis-opt}$ for discrimination. According to equations 4.22 and 4.23, this is the form of the carrier and the signal that sets the length scale of the optimal discrimination filters in figures 4.3, 4.6 and 4.7 and this is the form of the carrier that sets the length scale of estimation filters in those figures.

We can also compare the forms of the actual and optimal linear filters for both tasks. Figure 4.6B presents this comparison for the case of discrimination (also showing the weight matrix, the carrier and the converged state of the network); figure 4.6A presents the comparison for estimation. The filter for estimation is close to the optimal form in its

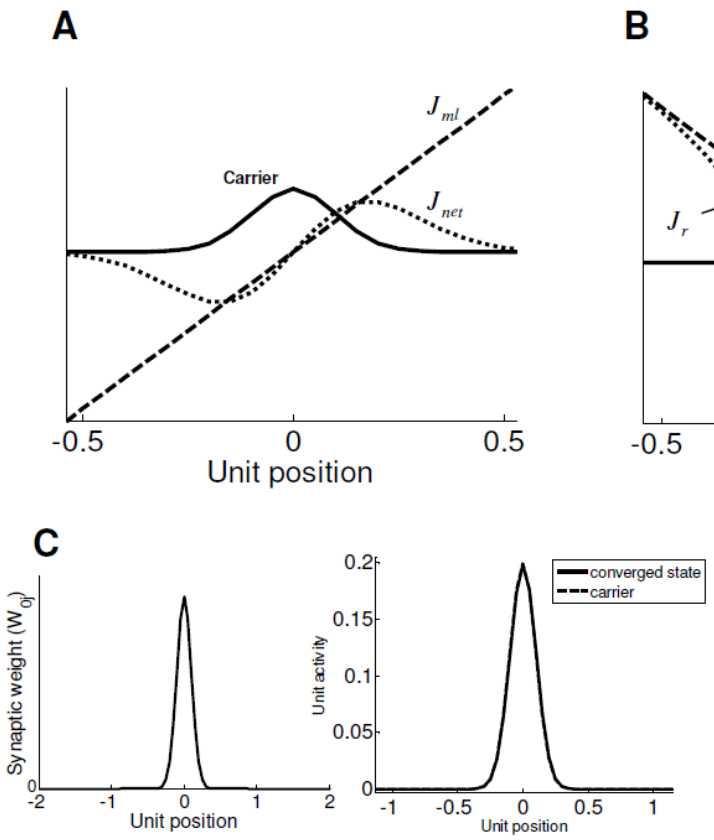

Figure 4.6. Optimal and network filters. **A**) Estimation task. The dashed line shows the optimal filter for estimation. The dotted line shows the network's filter for estimation (J_{net}). The solid line shows the carrier. **B**) Discrimination task. The dashed line shows the optimal filter for discrimination. The dotted line shows the network's filter (J_r) for the case of Poisson noise. The solid line shows the carrier (scaled to fit the graph) indicating where the input is substantial. For both (A) and (B), unit 0 on the X axis represents the center of the carrier and the Y axis represents the magnitude of different filters for different units. **C**) Central portion of the translation-invariant weights (left) and the carrier and converged state of the network (right), same as 4.1C (both patterns have very nearly the same form).

central regime, but then deviates substantially. However, the input is near zero where the carrier is near zero, and therefore this does not disturb the performance of the network. Note that the overall scale of the discrimination filter is arbitrary, since this does not affect its performance. Of course, the point of linearization varies with the location of the carrier, so the linear filters 'track' the actual value of y.

Finally, it is possible to change the weights in the network to improve its performance at one of these two tasks, albeit to the possible detriment of the other. Figure 4.6A shows that the width of the network's linear filter is slightly too narrow for estimation. Figure 4.7C shows a slightly broader set of weights, together with the broader converged state of the network. Although this is no longer the same as the input, it is still possible to linearize the network and calculate its various linear filters. As for the previous case, linearization accurately predicts the performance level of the network. Figure 4.7A shows that the linear filter for estimation is now even closer to that of the optimal linear estimator, and indeed the ratio of their standard deviations is now 1.005. However, figure 4.7B shows that the discrimination filter is now much flatter in the crucial regime (although still having the appropriate sign), and indeed its signal to noise ratio is worse than that with the previous weights. Figure 4.7D compares the performance levels of the two sets of weights.

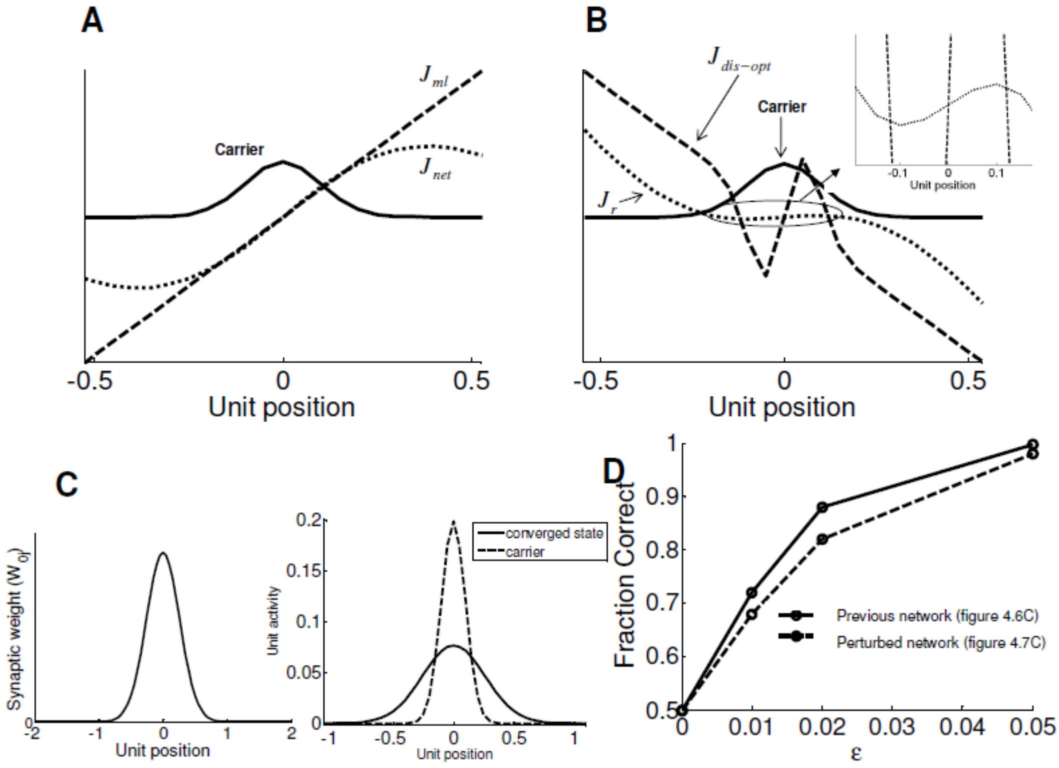

Figure 4.7. The filters of a perturbed network. The same convention is used for all the plots as in figure 4.6. **A**) Estimation task. The extra breadth makes the network's filter a better match to the optimal estimation filter. **B**) Discrimination task. The broader weights lead to a pattern of activity of the units on the attractor being quite different. This makes the network's change-based filter very shallow. For both (A) and (B), unit 0 on the X axis represents the center of the carrier and the Y axis represents the magnitude of different filters for different units. **C**) Central portion of the translation-invariant weights (left) and the carrier and converged state of the network (right). **D**) Performance levels of the weight matrices shown in 4.6C and 4.7C for discrimination task.

Note that the linear approximation predicts the performance of the perturbed network accurately (figure 4.4C inset) even though the widths of the average input and output patterns are different (figure 4.7C, right). As we mentioned earlier in the paper, we considered the case where the widths are the same mainly to simplify the comparison between the change-based and attractor-based methods.

For the case of figure 4.6C, the input is very close to the line attractor. By contrast, for the case of figure 4.7C, the input is far from the line attractor. Nevertheless, the linear analysis accurately predicts the performance of the network (figure 4.4C inset) and therefore the same trade-off still holds between the change-based and the attractor-based methods. As might be expected, as long as the input-output transformation is such that the first order Taylor expansion accurately predicts the effect of the first order perturbations, the linear approximation will accurately predict the performance of the network for low levels of noise.

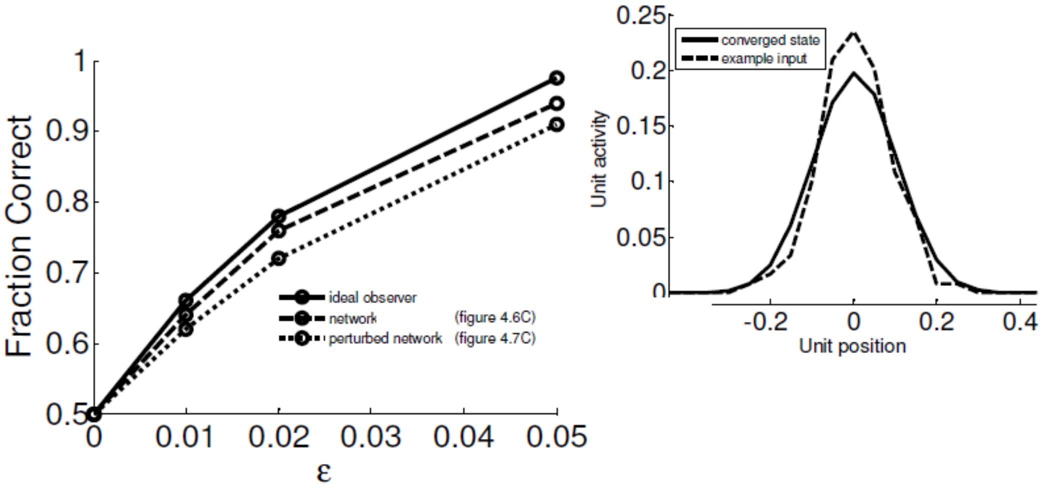

Figure 4.8. High noise regime. Left: Performance levels of the ideal observer, the network shown in 4.6C and the perturbed network shown in 4.7C for the discrimination task in the face of high noise. Right: shows an example input and the average input to the network for Poisson noise. They are now substantially different.

We have effectively shown a limiting trade-off between change-based discrimination and attractor-based estimation. If the latter was perfect, then the former would be at chance. Of course, we also showed that it is possible to have both methods operating near-optimally simultaneously (figure 4.2 and section 4.3.1). The reason is that we allowed change-based inference to work with very small changes in the CoM. If one constrains these changes to be not smaller than a threshold (for instance to overcome noise in the actual read-out itself), then this will force a decrease in the performance of attractor-based estimation and therefore there will be a trade-off between the performance of the change-based readout for discrimination task and the performance of the attractor-based readout for estimation task.

For the convenience of the analysis, we have so far considered low levels of input noise (and signal). Figure 4.8 shows that the network is also near optimal when signal and noise are both substantially stronger. However, in this regime, linearization is not such a good approximation, making it hard to understand the provenance of the network's performance based on linearization around the attractor. However, one could still linearize the network around the initial state similar to the analysis that we presented in the previous chapter to understand the near optimality of the network's performance. Note that the performance level of the ideal observer changes with the high levels of signal and noise, and the network's performance follows the performance level of the ideal observer.

It is interesting to note that although the linearization fails in the high noise regime, increasing the breadth of the weights (as in figure 4.7) has the same effect as in the low noise case of improving estimation and impairing discrimination (figure 4.8).

4.4 Discussion

We had previously observed superior performance at invariant pattern discrimination for the change-based method (Moazzezi and Dayan, 2008) and provided a complete analysis of the workings of the method by linearizing around the initial state of the network (chapter 3). In this chapter, we showed that the change-based method is also able to perform optimally near the attractor and we analyzed the method in that regime. In particular, we considered a new task for which it was possible to take advantage of the advanced state of understanding of the near-attractor dynamics of line attractor networks in the context of stimulus estimation (Pouget et al., 1998). In this regime, such a network can be successfully linearized; we used the results of this approximation to show the close relationship between optimal and change-based inference. One striking finding was that change-based inference only performs well since the ability of the network to perform estimation has been compromised, so that the signal for discrimination is not treated as noise for estimation. We analyzed the resulting trade-off.

We employed a task that, to the best of our knowledge, has never been the subject of experimental test. Nevertheless, it has some interesting properties that make it an attractive target – for instance, performance in our model is surprisingly sensitive to the relative widths of the population hills of activity induced by the bars. Even when the heights of carrier and signal are so dissimilar that there can be no confusion between them, inference is easier when their widths are different. Though this particular prediction might not be fully robust to factors such as multi-scale receptive fields, it would be interesting to study the dependence of error rate on this factor.

In previous work on estimation, linear analysis was only applied when the network's state started and remained near its attractor (Pouget et al., 1998). We showed that the performance level remains optimal (for both estimation and discrimination tasks) when the network's state starts so far from the attractor that the approximation fails (because of noise or a mismatch between a large signal and the network's attractor state). The workings of the network in these regimes (for the discrimination task), can be understood by linearizing the state of the network around the initial state rather than the network's final, attractor state as we proved it in chapter 3 for the bisection task as a paradigmatic example. It is interesting that the performance of the attractor-based readout (for estimation) also remains optimal for the case that the initial state is far from the attractor despite the fact that, as mentioned above, the linear analysis around the attractor state that has been suggested to explain its optimal performance for estimation tasks (Pouget et al., 1998) fails in this case.

The change-based method decides based on the change of a statistic of the neural activity. Clearly the choice of statistic is influenced by the nature of the task. For the bisection task we employed a linear statistic of the normalized neural activity (i.e., the CoM) and we observed that one could extract nearly all the information from the change in this. For the task we considered in this chapter we used the same statistic (CoM). Its suitability is evident in the near optimal performance of the task.

An alternative potential solution to the task we consider in this chapter is to design two line attractor networks, for which both get the very same input (a linear sum of signal, carrier, and noise) but whereas one of them extracts the location of the signal, the other extracts the location of the carrier. The decision would then be based on comparing these two estimated locations. To the best of our knowledge, such an approach has not previously been attempted, perhaps for the very good reason that the signal is swamped by the noise, and so it is not clear how the first of these networks could function. Even if this idea could be made to work, the computational demands of this approach would be substantially greater than for change-based discrimination, by virtue of requiring two separate networks rather than just one. Further, we showed in the previous chapters that change-based inference is very robust to substantial levels of synaptic noise. By contrast, line attractor networks are extremely sensitive to noise, because of the requirement for, and nature of, null stability. Noise will have a significantly deleterious effect by itself in each of the two networks; and will then pose an even more drastic problem for a computation based on the difference between quantities estimated from each.

Appendix 4.A: The form of the translation invariant weight matrix:

The weights have the form of a truncated circular Gaussian:

$$\mathbf{W}_{ij} \propto \begin{cases} e^{\frac{cos(\pi|i-j|/N)-1}{\gamma^2}} & if\ |i-j| < N/2 \\ 0 & otherwise \end{cases} \tag{4.15}$$

The central row of this translation invariant weight matrix is shown in figures 4.1C, 4.6C and 4.7C. We used $\gamma = 0.078$ for weights shown in figures 4.1C and 4.6C and $\gamma = .2$ for weight shown in figure 4.7C.

Appendix 4.B: Linearized form of the optimal discrimination and estimation filters

In this appendix, we derive the linearized form of the optimal discrimination and estimation filters, under the assumption that the signal and noise are both small compared with the carrier and (scaled) Poisson or Gaussian noise. The signal to noise ratio of this filter controls the optimal performance.

Consider the case that the carrier is at y^*, giving rise to mean activity for unit i of $f_i\left(y^*\right)$, and the signal is at ε^*, with activity $s_i\left(\varepsilon^*, y^*\right)$. The total mean activity is

$$h_i\left(\varepsilon^*, y^*\right) = f_i\left(y^*\right) + s_i\left(\varepsilon^*, y^*\right) \qquad (4.16)$$

whence, assuming that noise is small, the actual activity of unit i, a_i is

$$a_i = h_i\left(\varepsilon^*, y^*\right) + \delta a_i \qquad (4.17)$$

We write $h_i^{\varepsilon}\left(\varepsilon, y\right) = \dfrac{\partial h_i\ (\varepsilon, y)}{\partial \varepsilon}$, and similarly for the other partial derivatives.

For scaled Poisson noise, the log likelihood associated with activities a_i is

$$log\ P(\boldsymbol{a}, \varepsilon, y) = \phi \sum_i a_i\ log\ h_i(\varepsilon, y) + K$$

where ϕ and K are independent of ε, y. Maximum likelihood inference determines ε and y as the solutions of

$$\frac{\partial\, log\ P(\boldsymbol{a}, \varepsilon, y)}{\partial \varepsilon} = \frac{\partial\, log\ P(\boldsymbol{a}, \varepsilon, y)}{\partial y} = 0 \qquad (4.18)$$

We consider this to first order, where δa_i, ε, ε^* and $\delta y = y - y^*$ are very small. Further, we consider the translation invariant limit, for which $\sum_i h_i\left(\varepsilon, y\right)$ is independent of ε and y, and so $\forall \varepsilon, y$:

$$\sum_i h_i^{\varepsilon}\left(\varepsilon, y\right) = \sum_i h_i^{y}\left(\varepsilon, y\right) = \sum_i h_i^{\varepsilon\varepsilon}\left(\varepsilon, y\right) = \sum_i h_i^{\varepsilon y}\left(\varepsilon, y\right) = 0 \qquad (4.19)$$

If we expand the first part of equation (4.18), we get

$$\sum_i \frac{a_i}{h_i\ \left(\varepsilon, y\right)} h_i^{\varepsilon}\left(\varepsilon, y\right) = 0$$

which, to first order about $\varepsilon = 0$, $\delta y = 0$, writing h_i for $h_i\left(0, y^*\right)$ and similarly for the other derivatives for convenience, gives

$$\sum \left(1 + \frac{\delta a_i}{h_i}\right)\left(1 - \varepsilon \frac{h_i^{\varepsilon}}{h_i} - \delta y \frac{h_i^{y}}{h_i}\right)\left(h_i^{\varepsilon} + \varepsilon h_i^{\varepsilon\varepsilon} + \delta y h_i^{y\varepsilon}\right) = 0$$

And so, taking advantage of equation (4.19), we get

$$\varepsilon\left(\sum_i \frac{\left(h_i^{\varepsilon}\right)^2}{\left(h_i\right)}\right)+\delta y\left(\sum_i \frac{\left(h_i^{\varepsilon}\right)\left(h_i^{y}\right)}{\left(h_i\right)}\right)=\sum_i \delta a_i \frac{h_i^{\varepsilon}}{h_i} \tag{4.20}$$

and similarly

$$\varepsilon\left(\sum_i \frac{\left(h_i^{\varepsilon}\right)\left(h_i^{y}\right)}{\left(h_i\right)}\right)+\delta y\left(\sum_i \frac{\left(h_i^{y}\right)^2}{\left(h_i\right)}\right)=\sum_i \delta a_i \frac{h_i^{y}}{h_i} \tag{4.21}$$

If we denote $\alpha=\sum_i \frac{\left(h_i^{\varepsilon}\right)^2}{\left(h_i\right)}$, $\beta=\sum_i \frac{h_i^{\varepsilon}\left(h_i^{y}\right)}{\left(h_i\right)}$ and $\gamma=\sum_i \frac{\left(h_i^{y}\right)^2}{\left(h_i\right)}$ then, solving the simultaneous equations (4.20) and (4.21) for ε gives

$$\varepsilon=\frac{1}{\alpha\gamma-\beta^2}\sum_i\left(\frac{\gamma h_i^{\varepsilon}-\beta h_i^{y}}{h_i}\right)\delta a_i$$

And thus the linearized ML discrimination filter has i th component

$$J^i_{dis-opt}=\frac{1}{\alpha\gamma-\beta^2}\frac{\gamma h_i^{\varepsilon}-\beta h_i^{y}}{h_i} \tag{4.22}$$

The same procedure can be followed for the case of scaled Gaussian noise, and, given the way we defined it, leads to the same expression as in equation (4.22).

The procedure to derive the optimal estimator is similar to above except that the goal is to estimate δy for the case that $\varepsilon=0$. This, therefore involves only equation (4.21) which imposing $\varepsilon=0$ gives:

$$\delta y=\frac{1}{\gamma}\sum_i \frac{h_i^{y}}{h_i}\delta a_i$$

And therefore the linearized ML estimation filter has i th component

$$J^i_{est-opt}=\frac{1}{\gamma}\frac{h_i^{y}}{h_i} \tag{4.23}$$

Appendix 4.C: Linear filter and how it relates to the Jacobian of the network

Here we represent the linear filter for the network for which the form of the carrier and the converged states are the same in terms of the Jacobian of the network. We define $\boldsymbol{v}^n = \boldsymbol{u}^n - \overline{\boldsymbol{a}}_c$ where $\boldsymbol{u}^0 = \boldsymbol{a}$. If we denote the Jacobian matrix by $\boldsymbol{K}$ (since the network dynamics in (Pouget et al., 1998) minimized a bounded Lyapunov function, all the eigenvalues of the Jacobian had a magnitude less than 1 and therefore $\mathbf{K}^{\infty}$ existed) then we have (Pouget et al., 1998):

$$\boldsymbol{K}^n \boldsymbol{v}^0 = \boldsymbol{v}^n$$

The above equation implicitly assumes that the Jacobian does not change during the evolution of the neural activity. Note that $\boldsymbol{u}^0 = \overline{\boldsymbol{a}}_c + \overline{\boldsymbol{a}}_s + \boldsymbol{n}$ and that $\overline{\boldsymbol{a}}_c$ is a function of y ($\overline{\boldsymbol{a}}_c = \overline{\boldsymbol{a}}_c(y)$) and $\overline{\boldsymbol{a}}_s$ is a function of both ε and y , ($\overline{\boldsymbol{a}}_s(\varepsilon, y)$). We are also assuming that the converged state of the network has the same form as the carrier but is at a different location $y + \delta y$ and therefore is represented by $\overline{\boldsymbol{a}}_c(y + \delta y)$.

$$\mathbf{v}^{\infty} = \mathbf{K}^{\infty}\mathbf{v}^0 = \overline{\mathbf{a}}_c(y + \delta y) - \overline{\mathbf{a}}_c(y) = \delta y \frac{\partial \overline{\mathbf{a}}_c(y)}{\partial y}$$

(4.24)

Therefore we have:

$$\mathbf{K}^{\infty}(\overline{\mathbf{a}}_s + \mathbf{n}) = \delta y \frac{\partial \overline{\mathbf{a}}_c(y)}{\partial y} \Rightarrow$$

$$\left[\frac{\partial \overline{\mathbf{a}}_c(y)}{\partial y}\right]^T \mathbf{K}^{\infty}(\overline{\mathbf{a}}_s + \mathbf{n}) = \delta y \left[\frac{\partial \overline{\mathbf{a}}_c(y)}{\partial y}\right]^T \frac{\partial \overline{\mathbf{a}}_c(y)}{\partial y}$$

(4.25)

And therefore the network's filter for estimation is:

$$\boldsymbol{J} = \frac{\left[\frac{\partial \bar{\boldsymbol{a}}_c(y)}{\partial y}\right]^T \boldsymbol{K}^{\infty}}{\left[\frac{\partial \bar{\boldsymbol{a}}_c(y)}{\partial y}\right]^T \frac{\partial \bar{\boldsymbol{a}}_c(y)}{\partial y}} \tag{4.26}$$

Chapter 5

Learning

5.1 Introduction

We showed optimal performance for the change-based method in performing perceptual learning hyperacuity tasks like the bisection task. However, the weight matrices in all cases were handcrafted and not learned. Learning recurrent models of perceptual learning has proved difficult and is still an unsolved problem (Tsodyks and Gilbert, 2004).

Learning optimal recurrent weights (weights that allow optimal performance) is a necessary step to prove that the optimal performance of the change-based method that we showed in previous chapters is not specific to a particular task (like the bisection task) and it is able to perform a broader class of computations optimally. In this chapter, we first show that the well-known BackPropagation Through Time (BPTT) algorithm (Rumelhart et al., 1986a; Rumelhart et al., 1986b) is able to learn the optimal recurrent weights within the change-based framework to solve a class of tasks including the bisection task that we studied in previous chapters. Although BPTT is not a biologically plausible learning algorithm, it has never been demonstrated to learn the optimal weights for recurrent models of perceptual learning. In addition to BPTT, we also develop another algorithm that directly maximizes the first order signal to noise ratio and we show that it also learns the optimal recurrent weights.

Since learning has never been demonstrated in recurrent network models of perceptual learning before, our goal in this chapter is mainly to prove that learning is possible by the change-based method. We show that learning is easy and fast within the change-based framework.

We should also note that learning recurrent weights has never been demonstrated within the conventional line attractor-based approach. This makes it difficult to generalize the attractor-based method to a broader class of tasks. For BPTT, the cost of learning within the change-based framework is much less than that of the attractor-based framework since the computations of the change-based method are performed during the early portion of the neural activity (within the first few iterations of the recurrent activity) and the error does not need to backpropagate all the way from the attractor to the initial state .

5.2 BPTT (Back Propagation Through Time)

5.2.1 BPPT for change-based method

BPTT is a version of the Back Propagation (BP) method designed for training recurrent networks. BP has been applied to train small feedforward networks. The conventional BP defines a cost function that depends on the difference between the actual output at the top layer of the feedforward network and the desired output (Rumelhart et al., 1986a; Rumelhart et al., 1986b). It aims to learn the weights that minimize this difference, averaged over all the inputs in the training set. BPTT for recurrent networks is based on the idea that unrolling a recurrent network over time turns it in to a multiple layered feedforward network; BPTT for recurrent networks is equivalent to BP for unrolled feedforward counterpart (figure 5.1). The only difference is

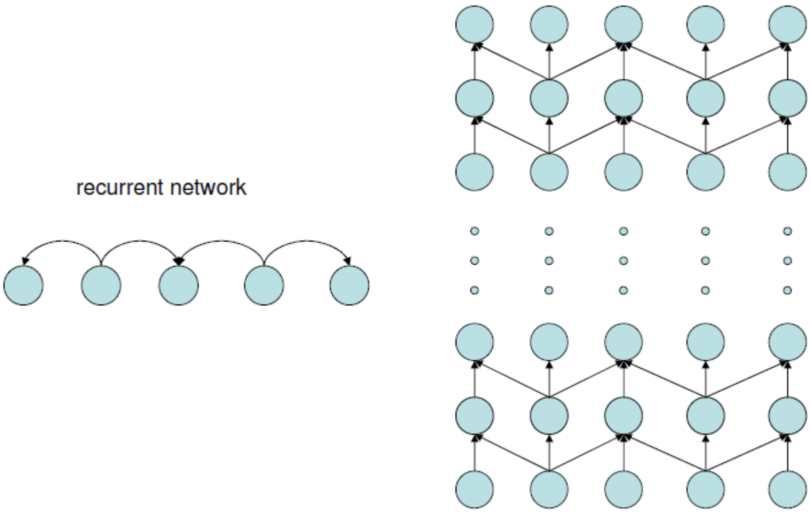

Figure 5.1 A recurrent network and its equivalent unfolded feedforward network. Every iteration of the recurrent network is equivalent to one layer of the feedforward network.

that the feedforward weights connecting successive layers in the feedforward network are constrained to be the same.

Here we briefly describe the BPTT method that we employed to train recurrent networks within the change-based framework. The cost function, depends both on the correct and incorrect trials during training and on the evolution of the CoM of the neural activity; in particular, it is based on the difference of the center of mass (CoM) of the activity at two layers of the (equivalent) feedforward network. For simplicity we assume that the change in the CoM is measured between the first and the last layers of the feedforward network:

$$\Delta = \frac{\sum_i x_i O_i^T}{\sum_i O_i^T} - \frac{\sum_i x_i O_i^0}{\sum_i O_i^0} \tag{5.1}$$

where x_i is the location of unit i and O_i^t denotes its output activity at time t in the recurrent network or at layer t in the equivalent feedforward network. We denote the input to the network at time t by U_i^t (U_i^0 is the input to the network). We define the cost function for the BP algorithm to be:

$$\begin{aligned} C &= \log\left(\frac{1}{1+e^{-\gamma\Delta}}\right) & & if\ (\varepsilon > 0) \\ C &= \log\left(1 - \frac{1}{1+e^{-\gamma\Delta}}\right) & & if\ (\varepsilon < 0) \end{aligned} \tag{5.2}$$

The BPTT method modifies the weights at each trial according to the following equation:

$$\delta W_{ij}^t \propto -\frac{\partial C}{\partial W_{ij}^t} \tag{5.3}$$

where W_{ij}^t denotes the weight from neuron j at layer $t-1$ to neuron i at layer t. We consider the following dynamics for the network:

$$O_i^n = f\left(U_i^n\right) \tag{5.4}$$

$$U_i^t = \sum_j W_{ij}^t O_j^{t-1} \tag{5.5}$$

where t indicates the iteration number (or layer number in the equivalent feedforward network). Since the feedforward network is an unfolded version of the recurrent network, we have:

$$W^t = W \qquad \forall t \tag{5.6}$$

In other words, the weights that connect different successive layers are the same. By expanding equation 5.3 using the chain rule we get the following update rule for the weights that connect to the top layer:

$$\delta W_{ij}^T \propto -\frac{\partial C}{\partial W_{ij}^T} = -\frac{\partial C}{\partial \Delta} * \frac{\partial \Delta}{\partial O_i^T} * \frac{\partial O_i^T}{\partial U_i^T} * \frac{\partial U_i^T}{\partial W_{ij}^T} \tag{5.7}$$

The first term in the right hand side of equation 5.7 is (using equation 5.2):

$$\frac{\partial C}{\partial \Delta} = 1 - \frac{1}{1+e^{-\gamma\Delta}} \qquad if\ \ (\varepsilon > 0) \tag{5.8}$$

$$\frac{\partial C}{\partial \Delta} = \frac{1}{1+e^{-\gamma\Delta}} \qquad if\ \ (\varepsilon < 0) \tag{5.9}$$

And the second term in the right hand side of equation 5.7 is (using equation 5.1):

$$\frac{\partial \Delta}{\partial O_i^T} = \frac{x_i}{\sum_j O_j} - \frac{1}{\sum_j O_j} \frac{\sum_j x_j O_j}{\sum_j O_j} \tag{5.10}$$

The third term can be directly computed from equation 5.4 and the last term $\frac{\partial U_i^T}{\partial W_{ij}^T}$ is (using equation 5.5):

$$\frac{\partial U_i^T}{\partial W_{ij}^T} = O_j^{T-1} \tag{5.11}$$

The change in the weights in other layers can be computed in a similar way to equation 5.7 to calculate the terms δW_{ij}^{T-1}, δW_{ij}^{T-2}, etc. For example,

$$\delta W_{ij}^{T-1} \propto -\frac{\partial C}{\partial W_{ij}^{T-1}} =$$

$$-\frac{\partial C}{\partial \Delta} * \sum_m \left(\frac{\partial \Delta}{\partial O_m^T} * \frac{\partial O_m^T}{\partial U_m^T} * \frac{\partial U_m^T}{\partial O_i^{T-1}} * \frac{\partial O_i^{T-1}}{\partial U_i^{T-1}} * \frac{\partial U_i^{T-1}}{\partial W_{ij}^{T-1}} \right)$$

Since we assume that the weight matrices that connect all successive layers represent the same underlying weight matrix (equation 5.6), the change in the recurrent weight matrix δW_{ij} would be:

$$\delta W_{ij} = \frac{1}{T}(\delta W_{ij}^T + \delta W_{ij}^{T-1} + \delta W_{ij}^{T-2} + ...) \tag{5.12}$$

We used the following activation function for individual units in the network:

$$f(x) = x/(1+e^{-\beta x}) \tag{5.13}$$

Where $\beta = 50$. We imposed two other constraints during training: that the recurrent weight matrix is position invariant $W_{ij} = W_{|i-j|}$ and that it is symmetric. Therefore, if the weight matrix is of size $(2l+1)*(2l+1)$, then the two constraints mentioned above reduce the number of the free parameters that describe the weight matrix to $l+1$. The weight matrix can be reconstructed by using these $l+1$ elements. We refer to this as the “generating vector” of the weight matrix. One can reconstruct the central row of the weight matrix using this vector and the whole weight matrix by shifting the central row appropriately to generate the other rows.

To learn the weight matrix, we used both gradient descent and conjugate gradient methods. Both methods successfully learned the optimal weights, though, conjugate gradient learned faster (with fewer training examples). In the following we only report on the results of the conjugate gradient method. Figure 5.2A shows the central row of an example learned weight matrix that performs nearly as well as ideal observer after only one iteration (figure 5.2B magnifies its central part). Figure 5.2C shows its performance level. We introduced long range inhibition to the initial weight matrix (upon which BPTT learned the optimal one) to ensure stability. Each element of the excitatory part of the generating vector of the weight matrix, was randomly and independently

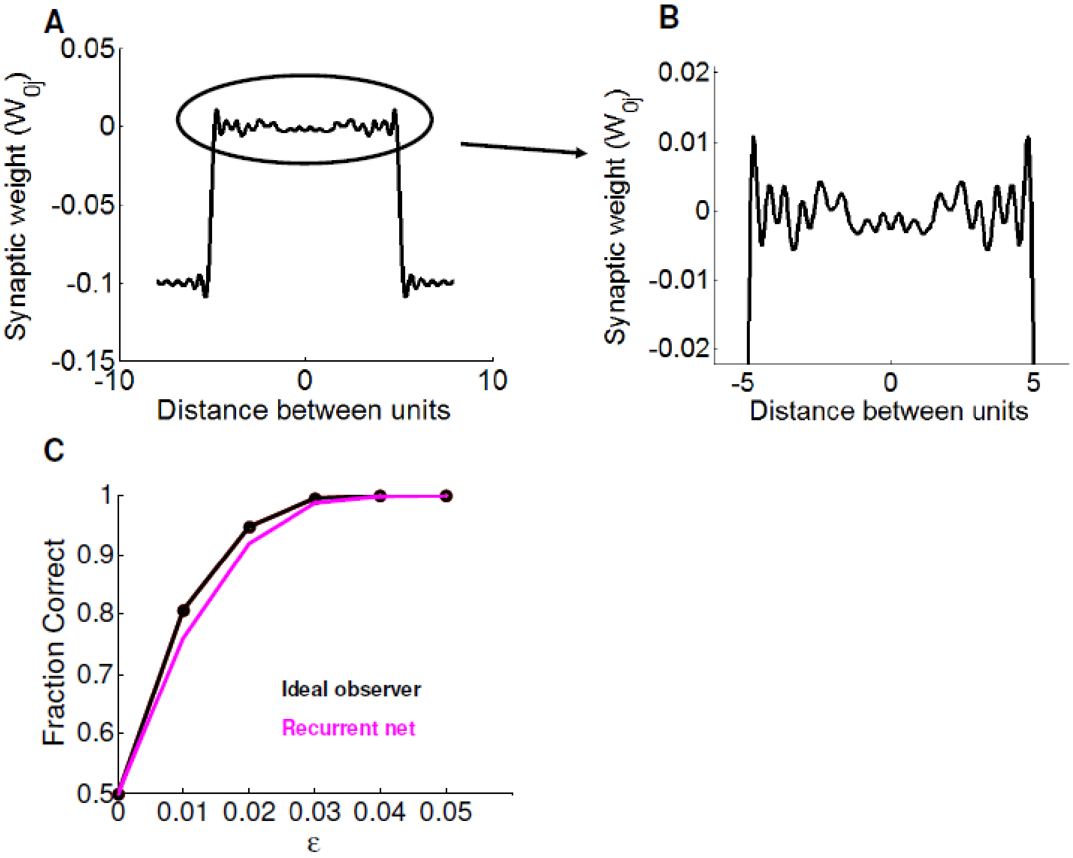

Figure 5.2 A learned weight via BPTT. **A**) The central row of a learned weight matrix using BPTT method for a two layer feedforward network which is equivalent to a recurrent network with just one iteration. **B**) magnifies the central part of the weight shown in (A). **C**) Performance level for the weight matrix shown in (A) along with the performance level of the ideal observer.

drawn from a Gaussian distribution with mean 0.1 and standard deviation of 0.05. Next, we generated the central row of the weight matrix and decomposed it into its $l+1$ (which is 161 here) Fourier components. We selected the components associated with the lowest K frequencies and ignored the rest. We linearly added these K components to generate the central row of the initial weight matrix. Conjugate gradient based BPTT learned from almost all initial conditions described above. Figure 5.3 shows a few more learned example weight matrices with near optimal performances. The examples shown in figures 5.2 and 5.3 are learned for only one iteration of the recurrent network (that is, one time step of the network or an equivalent two-layered feedforward network). Figure 5.4 shows more examples that are learned for 3 and 5 iterations of the recurrent network and they all perform near optimally. Therefore, learning is not only possible within the change-based framework but also it is very easy in the sense that a broad range of initial conditions result in near optimal performance. In light of our findings in chapter 3, the near optimal performance observed here is a result of BPTT algorithm learning effective eigenvectors.

Next we addressed the role of different frequencies of the generating vector of the weight matrix in the performance level of the recurrent network.

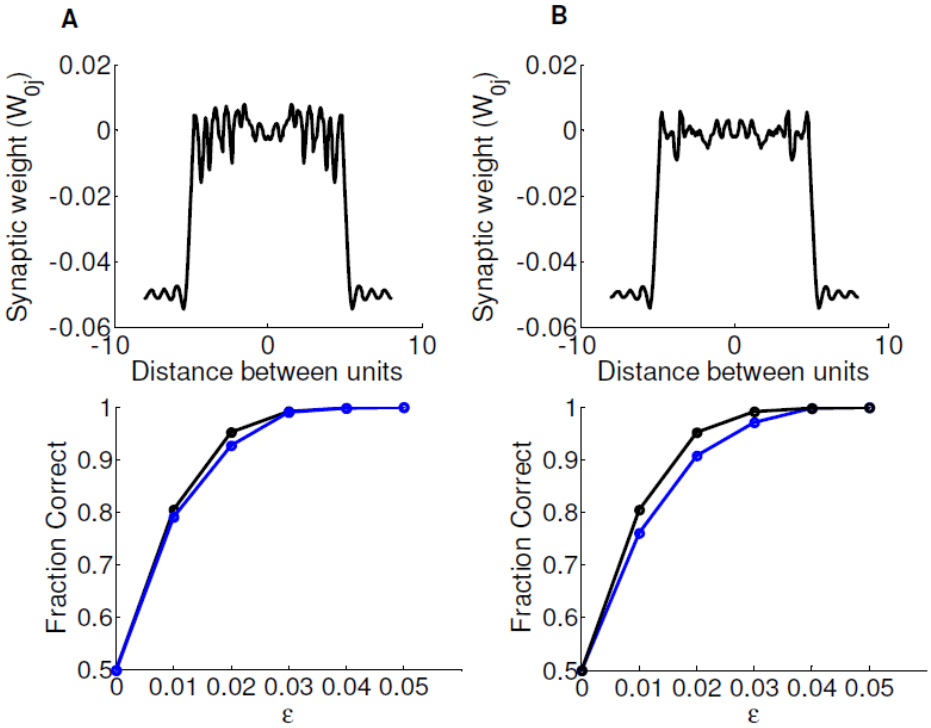

Figure 5.3 More examples of learned weight matrices for recurrent networks with only one iteration. **A**) and **B**) Top row shows the central row of two weight matrices that are learned to perform bisection task optimally after only one iteration. Bottom row shows the performance levels of the weights (blue) along with the performance level of the ideal observer (black).

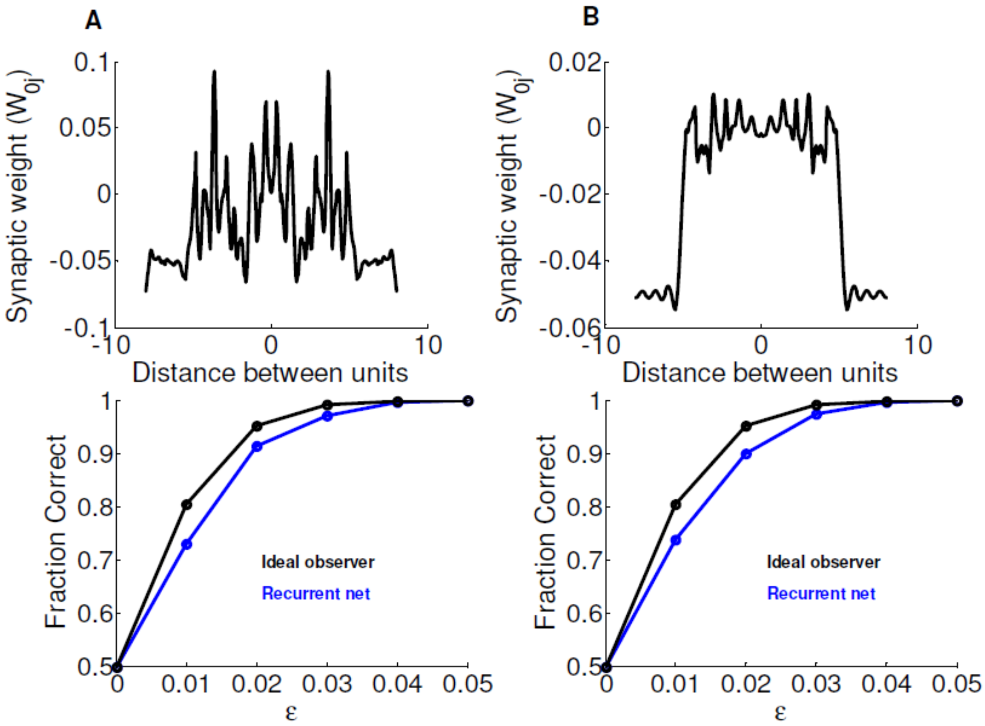

Figure 5.4 Examples of learned weight matrices for recurrent networks with multiple iterations. **A**) Top row shows the central row of a weight matrix that is learned to perform bisection task optimally after three iteration. Bottom row shows the performance level of the weight (blue) along with the performance level of the ideal observer (black). **B**) Top row shows the central row of a weight matrix that is learned to perform bisection task optimally after five iteration. Bottom row shows the performance level of the weight (blue) along with the performance level of the ideal observer (black).

5.2.2 Role of different frequencies

Once the network was trained and reached optimal performance, we looked for the minimal number of spatial frequencies needed for the learned weight matrix to maintain its near optimal performance. In other words, the question was what is the minimal K for which the first K lowest spatial frequencies of the trained weight matrix are sufficient for optimal performance. The answer is different for different learned weights. Therefore we measured the statistics. Note that the maximum number of spatial frequencies is $l+1$ which is 161 here. To statistically measure this effect, we learned

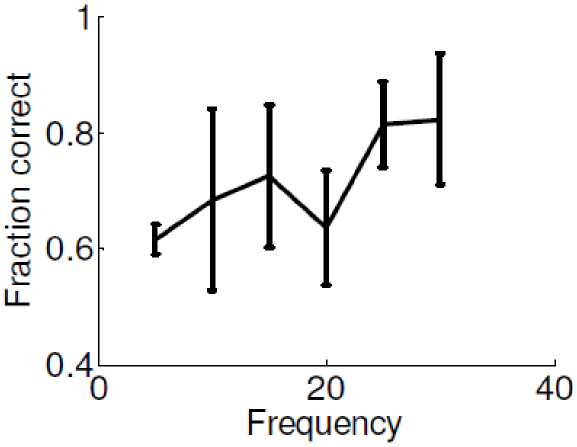

Figure 5.5 Role of different frequencies in performance level. Some frequencies improve the performance level more than others. Adding frequencies between 20 and 25 here results in the highest increase in the average performance. All the weight matrices included in this figure had a performance level of 84% or more for $\mathcal{E}$ =.02. The ideal observer's performance level for $\mathcal{E}$ =.02 is about 95%.

a set of weight matrices with optimal performances and measured their average performance in the presence of their first 5, 10, 15, 20, 25 and 30 frequencies (figure 5.5). The performance improved by adding the first 15 frequencies. However, adding the next 5 frequencies impaired the performance. Interestingly adding more frequencies reversed this and improved the performance again and the first 30 frequencies were sufficient for near optimal performance. In this thesis we will only focus on BPTT and maximum linear signal to noise ratio method for learning; nevertheless, these results reveal which frequencies carry more information and therefore can be used to design even more efficient learning algorithms.

5.2.3 Performance as a function of the number of examples

Next we asked how many examples are needed for the performance to reach optimal level? This is important since perceptual learning has been reported to be fast (Poggio et al., 1992). Figure 5.6 shows the performance as a function of the number of examples for the case in which the initial conditions are generated by keeping the first 40 frequencies. As can be seen, the performance increases to near optimal level by

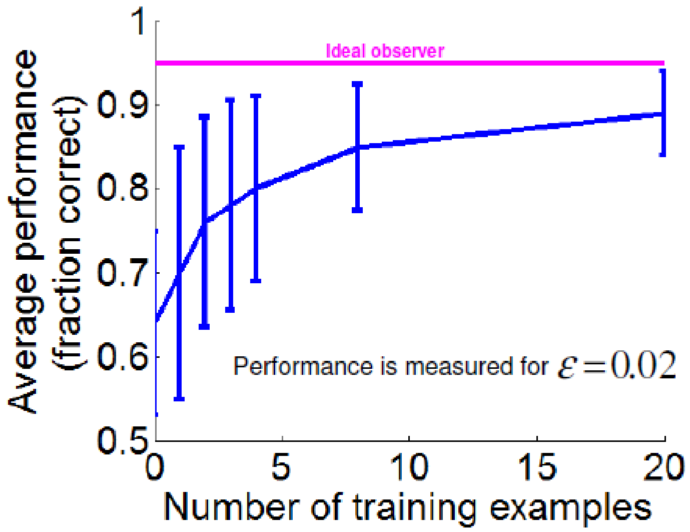

Figure 5.6 Performance as a function of the number of training examples. Performance for ε =.02 as a function of the number of training examples. Performance quickly improves with increasing the number of examples and reaches near optimal level with only 10 examples. The performance level without training here is about 65%.

about 10 examples. Such fast learning clearly demonstrates the ease by which the change-based method learns the bisection task. As can be seen from figure 5.6, keeping the first 40 frequencies of the generating vector of the weight matrix leads to a performance level that is much higher than chance even without training.

Next, we show that the fast learning observed here is not specific to the recurrent network with only 1 iteration. To this end, I measured the speed of learning for 2 and 3 iterations as well and as can be seen from figure 5.7 performance quickly improves and the speed of learning is not different from the 1 iteration case. However, for 3 iterations, conjugate gradient method performed 8 line searches while for 2 and 1 iterations, it only performed 3 line searches. Despite the difference in the number of line searches, the number of examples needed for near optimal performance remained the same for all three cases.

Obviously increasing the number of iterations makes it more and more difficult for BPTT to learn the weights which is one of the major problems with attractor-based method and one of the reasons why learning has never been demonstrated for neither estimation tasks nor discrimination tasks in the presence of invariance within attractor-based framework. Change-based method learns the optimal weights using BPTT exactly because it operates within the first few iterations of the recurrent network.

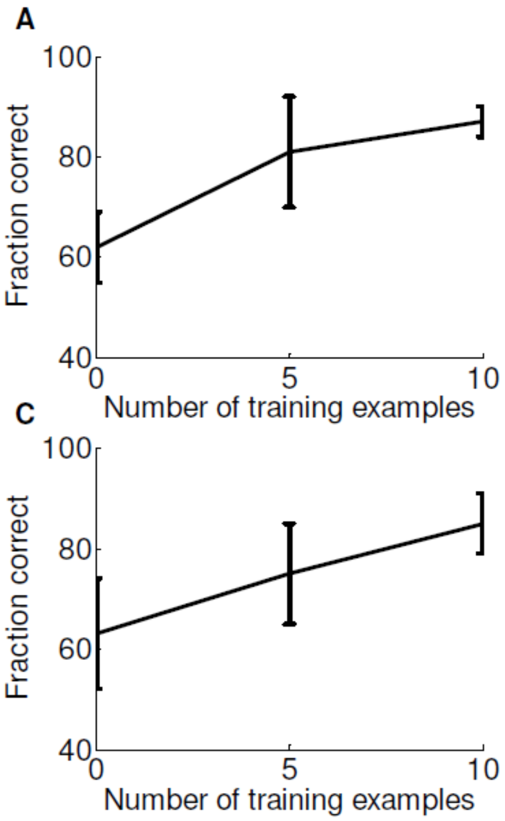

Figure 5.7 Performance as a function of the number of the training examples for different number of iterations. **A**) one iteration, **B**) two iterations and **C**) three iteration. The speed of learning is nearly the same in all three cases.

5.2.4 Different distances of the outer bars in the bisection task

Since learning is possible with change-based method, we tried a range of different distances of the outer bars for the bisection task. Figure 5.8 shows (the central row of) an example learned weight matrix for a distance twice the one we have considered so far for the bisection task (distance here is 2 units of length or equivalently 40 neurons. It was 20 neurons in previous examples). The performance level is near optimal.

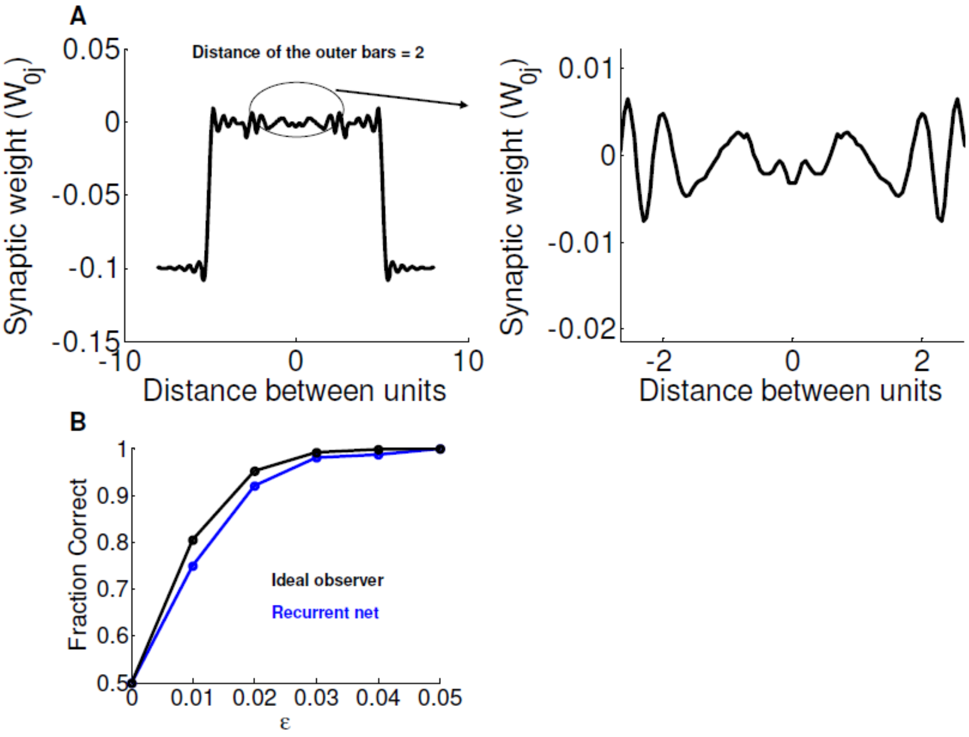

Figure 5.8 Training network for a larger distance between the outer bars. Training bisection task for the case in which the distance between outer bars is twice the one we have been studying so far. Performance is near optimal. **A**) Central row of a learned weight (figure on the right magnifies the central part of the weight). **B**) Performance of the weight matrix along with that of the ideal observer.

We also tried training for different distances within the same training session; the task here is again the bisection task but the distance of the outer bars is either 1 unit of length or 2 units of length at different trials. Training improved the performance to near optimal level for both distances of the outer bars (Figure 5.9). It is interesting to note that Herzog and colleagues tested this on human subjects and found that this dramatically decreased the learning speed and about 18000 trials were needed to reach the hyperacuity level (Parkosadze et al., 2008). Learning different distances both separately and simultaneously encourages us to try BPTT on other tasks.

5.2.5 Other tasks

Next we tested learning on a slightly different task in which four parallel bars are presented and the task is to decide if the middle of the two inner bars is to the right or to the left of the middle of the two outer bars. Figure 5.10 shows an example learned weight matrix and its performance level along with the performance of the ideal observer. Optimal performance is evident. We also tried BPTT for the task that we studied in chapter 4. Figure 5.11 shows an example learned weight and the corresponding performance level which is near optimal. These examples show that the change-based method enables learning and can be used for a much wider range of tasks including the bisection task. To appreciate the significance of these results, consider the line attractor network designed by Pouget et al (1998) that performed the task of orientation estimation optimally. A bar of light elicits a noisy hill of activity (which on average has a Gaussian shape) and the network estimates the orientation of the

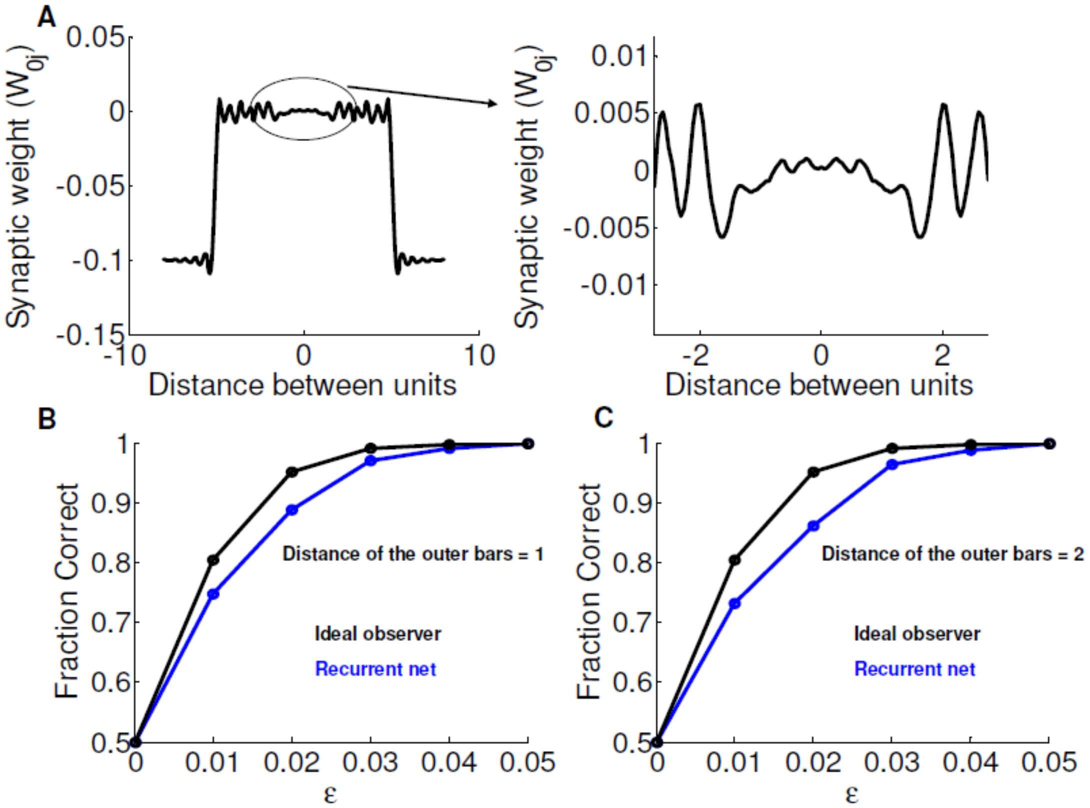

Figure 5.9 Simultaneous training for two distances of the outer bars. The distance between the outer bars in the bisection task is either 1 unit (20 neurons) or 2 units (40 neurons). **A**) Central row of a trained weight matrix, **B**) Performance of the weight matrix shown in (A) for the case in which the distance of the outer bars is 1 unit and **C**) Performance of the weight matrix shown in (A) for the case in which the distance of the outer bars is two units.

bar. Since the weight matrix for this task is handcrafted, it is not clear how the weight needs to be modified if we change the representation of the bars from Gaussian to another form. In fact, designing optimal recurrent weights within attractor-based framework has never been demonstrated for a broader range of tasks and we are not aware of any easy way to design such weights. The weight matrices associated with a few other tasks that has been addressed by the line attractor method have all been handcrafted and no learning algorithm has ever been shown to successfully learn within the attractor-based framework.

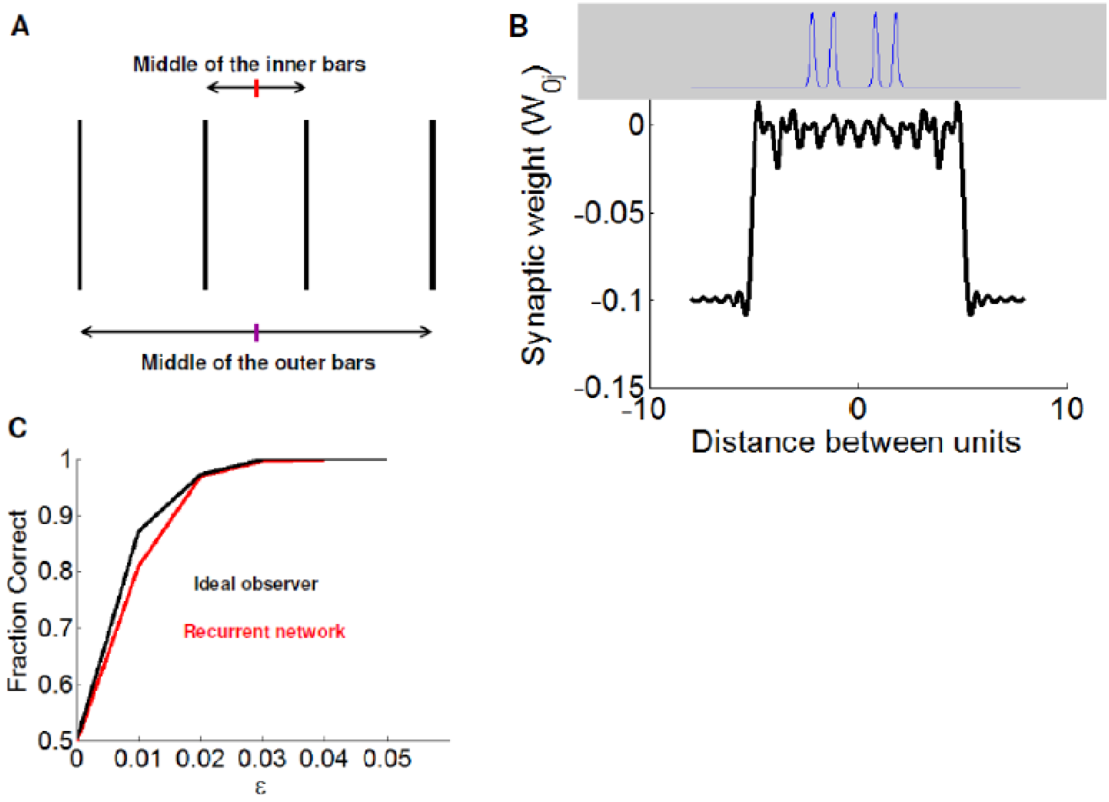

Figure 5.10 Learning another task. **A**) The task is decide whether the middle of the two inner bars is to the right or to the left of the middle of the two outer bars. **B**) The central row of a weight matrix trained to perform the task explained in (A) along with the average activity elicited by the four bars. **C**) Performance of the weight shown in (B) along with that of the ideal observer for this task.

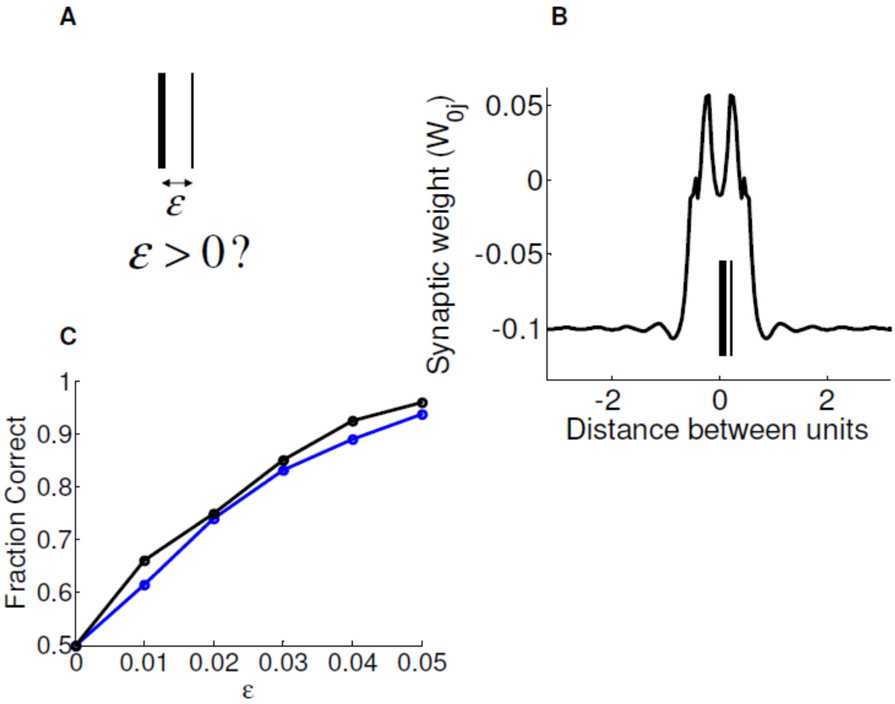

Figure 5.11 Learning the task studied in chapter 4. **A**) The task is to decide whether the low contrast bar (thickness of the lines represents their contrast) is to the left or to the right of the high contrast bar (exactly the same task that we considered in chapter 4). **B**) The central row of an example learned weight matrix using BPTT which performs this task optimally along with the low and high contrast bars. **C**) Performance of the network (blue) along with the performance of the ideal observer (black).

5.3 Linear analysis

5.3.1 Analysis

As we showed in chapter 3, a linear analysis around the initial state of the network provides an accurate description of the performance of the change-based method. Therefore we asked if it is possible to learn the weights by maximizing the signal to noise ratio derived from the first order linear analysis. In this section we derive the first order signal to noise ratio and apply it to the bisection task and we show that maximizing it results in learning optimal weights. Furthermore, linear analysis also helps us to understand how BPTT learns the optimal weights; linear analysis indicates that the performance of almost all weights learned by BPTT learning rule is predicted by the first order Taylor expansion which is in line with our results and the linear analysis that we presented in chapter 3.

We consider the simplest case in which the recurrent network performs only one iteration or the equivalent feedforward network has only two layers. Similar to the previous section, we model the task by rate based units with the following activation function:

$$f(x) = x/(1+e^{-\beta x})$$

where $\beta = 50$. We consider a recurrent network with one iteration:

$$\mathbf{u}^1 = f(W\mathbf{u}^0) \tag{5.14}$$

where $\mathbf{u}^0 = \mathbf{a}$ is the input to the network and $\mathbf{u}^1 = f(W\mathbf{a})$ is the output of the network after one iteration. We denote the centre of mass function by $g(.)$; $g(\mathbf{a})$ is the centre of mass of (the activity) vector $\mathbf{a}$. Linear analysis assumes that the vector $\mathbf{a}$ is the sum of three terms:

$$\mathbf{a} = \overline{\mathbf{a}} + \varepsilon\mathbf{S} + \mathbf{N} \tag{5.15}$$

where $\mathbf{S}$ (or perhaps better to say $\varepsilon\mathbf{S}$) is the signal vector and $\mathbf{N}$ is the noise vector. We are interested in the centre of mass of vector $\mathbf{a}$:

$$g(\mathbf{a}) = g(\overline{\mathbf{a}} + \varepsilon\mathbf{S} + \mathbf{N}) \tag{5.16}$$

The first order Taylor expansion around $\mathbf{a} = \overline{\mathbf{a}} + \varepsilon\mathbf{S} + \mathbf{N}$ yields:

$$g(\mathbf{a}) = g(\overline{\mathbf{a}}) + (\varepsilon\mathbf{S})^T \dot{g}(\overline{\mathbf{a}}) + (\mathbf{N})^T \dot{g}(\overline{\mathbf{a}})) \tag{5.17}$$

We can compute the centre of mass of the output of the network after one iteration in the same way:

$$g(\mathbf{f}(W\mathbf{a})) = g(\mathbf{f}(W\overline{\mathbf{a}} + \varepsilon W\mathbf{S} + W\mathbf{N}))$$

and its first order Taylor expansion is:

$$g(\mathbf{f}(W\mathbf{a})) = g(\mathbf{f}(W\overline{\mathbf{a}})) + (\varepsilon W\mathbf{S})^T (Q\dot{\mathbf{g}}(\mathbf{f}(W\overline{\mathbf{a}}))) + (W\mathbf{N})^T (Q\dot{\mathbf{g}}(\mathbf{f}(W\overline{\mathbf{a}}))) \tag{5.18}$$

where $Q_{ij} = \dot{\mathbf{f}}_i(W\overline{\mathbf{a}})\delta_{ij}$. We also have $g(\mathbf{f}(W\overline{\mathbf{a}})) = g(\overline{\mathbf{a}})$, i.e. the average activity with no signal results in no change in the CoM which ensures that the discrimination is unbiased, Comparing equations 5.17 and 5.18 yields:

$$g(\mathbf{f}(W\mathbf{a}))-g(\mathbf{a})=\left[g(\mathbf{f}(W\overline{\mathbf{a}}))-g(\overline{\mathbf{a}})\right]+$$
$$\left\lfloor(\varepsilon W\mathbf{S})^T(Q\dot{\mathbf{g}}(\mathbf{f}(W\overline{\mathbf{a}})))-(\varepsilon\mathbf{S})^T\dot{g}(\overline{\mathbf{a}})\right\rfloor+\left\lfloor(W\mathbf{N})^T(Q\dot{\mathbf{g}}(\mathbf{f}(W\overline{\mathbf{a}})))-(\mathbf{N})^T\dot{g}(\overline{\mathbf{a}})\right\rfloor=$$
$$0+signal+noise$$

where

$$signal=(\varepsilon W\mathbf{S})^T(Q\dot{\mathbf{g}}(\mathbf{f}(W\overline{\mathbf{a}})))-(\varepsilon\mathbf{S})^T\dot{g}(\overline{\mathbf{a}})$$

and

$$noise=(W\mathbf{N})^T(Q\dot{\mathbf{g}}(\mathbf{f}(W\overline{\mathbf{a}})))-(\mathbf{N})^T\dot{g}(\overline{\mathbf{a}}).$$

These equations represent the first order approximations to the signal and the noise in a single trial. However, to learn an optimal weight by the signal to noise analysis, we need to compute the variance of the noise and maximize the first order signal to noise ratio with respect to the weight. We can rewrite the signal and the noise terms as follows:

$$signal=\sum_j\sum_i\frac{\partial f_i(\vec{x})}{\partial x_i}\dot{g}_i(\vec{f}(W\vec{\bar{a}}))W_{ij}S_j-\sum_i S_i\dot{g}_i(\vec{\bar{a}}) \tag{5.19}$$

$$noise=\sum_j\sum_i\frac{\partial f_i(\vec{x})}{\partial x_i}\dot{g}_i(\vec{f}(W\vec{\bar{a}}))W_{ij}N_j-\sum_i N_i\dot{g}_i(\vec{\bar{a}}) \tag{5.20}$$

Since we have independent noise among units, we can write:

$$Var(noise)=<noise>^2=\sum_j\left(\sum_i\frac{\partial f_i(\vec{x})}{\partial x_i}\dot{g}_i(\vec{f}(W\vec{\bar{a}}))W_{ij}-\dot{g}_j(\vec{\bar{a}})\right)^2\sigma_j^2$$

Therefore, the signal to noise ratio that is maximized is:

$$(S/N)^2=\frac{\left(\sum_j\left(\sum_i\frac{\partial f_i(\vec{x})}{\partial x_i}\dot{g}_i(\vec{f}(W\vec{\bar{a}}))W_{ij}-\dot{g}_j(\vec{\bar{a}})\right)S_j\right)^2}{\sum_j\left(\sum_i\frac{\partial f_i(\vec{x})}{\partial x_i}\dot{g}_i(\vec{f}(W\vec{\bar{a}}))W_{ij}-\dot{g}_j(\vec{\bar{a}})\right)^2\sigma_j^2} \tag{5.21}$$

The goal is to find a weight matrix W that maximizes the above ratio. The performance of the resulting weight matrix is measured by the change-based method to see whether the resulting weights are optimal only to the first order or are optimal with respect to the change-based readout. To maximize equation 5.21 we used conjugate gradient method again.

5.3.2 Results

In this section, we first examine how well the linear analysis predicts the performance of the weight matrices learned via BPTT (discussed in the previous section) and then we present the results on learning the weights by maximizing the first

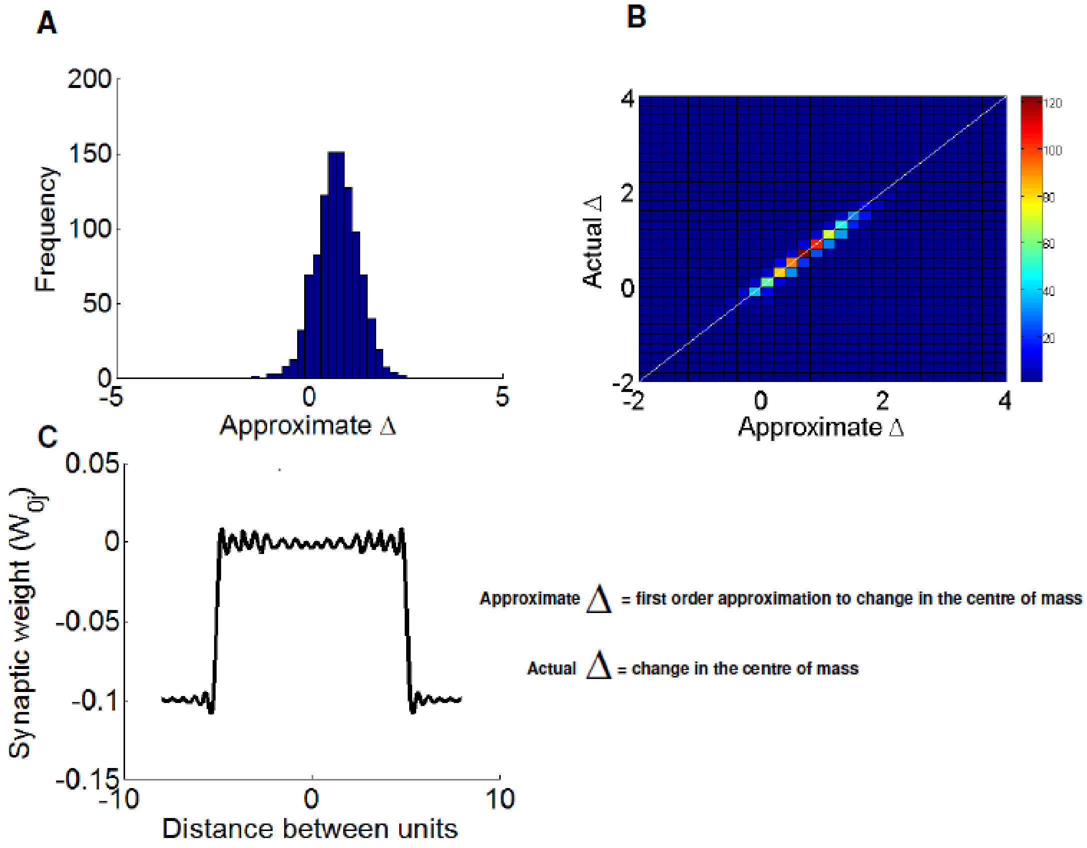

Figure 5.12 First order approximation accurately predicts the shift in the CoM for weights learned by BPTT. **A**) Histogram of the change in the CoM predicted by linear approximation for a weight learned by BPTT (shown in (C)). This is for the case where for $\mathcal{E}$ =.02. **B**) Scatter plot compares the change in the CoM predicted by the linear approximation against the actual change of the CoM and they are nearly identical for the weight matrix shown in (C). **C**) The central row of the trained weight matrix using BPTT.

order signal to noise ratio. Figure 5.12C shows an example learned weight matrix (using BPTT) and figure 5.12A shows the histogram of the change of the CoM predicted by the first order approximation for $\mathcal{E}$=.02. The first order approximation to change in the CoM at each trial is the sum of signal (equation 5.19) and noise (equation 5.20) in that trial. The predicted performance from linear analysis is not different from the optimal performance (figure 5.12A). Figure 5.12B shows a scatter plot that compares the real change in the CoM and the one that is predicted by the first order Taylor expansion (signal plus noise; equations 5.19 and 5.20). As can be seen, the predicted and the actual changes in the CoM are nearly identical. This holds for almost all weight matrices that we examined providing further support for our results in chapter 3 that the first order term in the Taylor expansion is capable of encoding almost all the task relevant information.

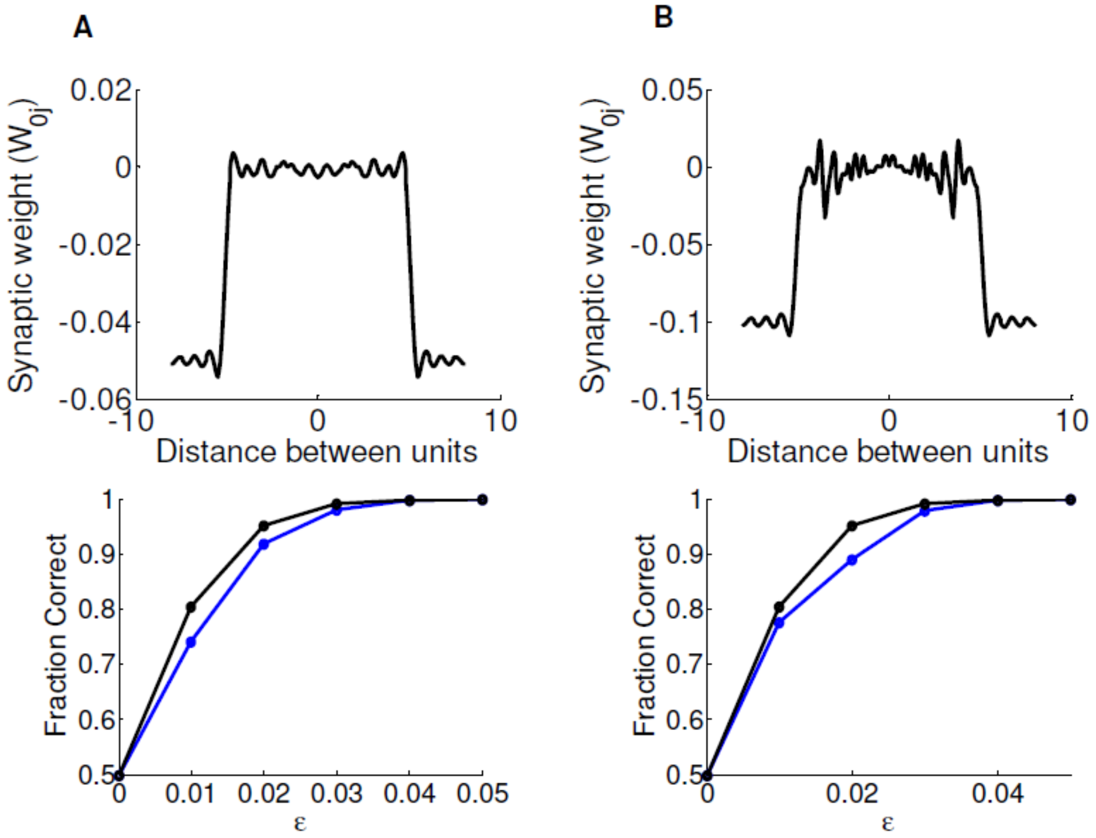

Figure 5.13 Learning by maximizing the signal to noise ratio. **A**) and **B**) Top row shows the central row of two weight matrices that are learned to perform bisection task by directly maximizing the signal to noise ratio (equation 5.21). Bottom row shows the performance levels of the weights (blue) along with the performance level of the ideal observer (black).

Next we examined learning by maximizing the signal to noise ratio. Figure 5.13 shows two example learned weights and their performance levels along with the optimal performance level for the bisection task. Both example weights perform optimally. Similar to BPTT, we added inhibition to the initial weights to ensure stability. We find that a majority of such initial conditions allow learning optimal weights.

These results indicate that learning based on maximizing the signal to noise ratio is robust to initial conditions and does not require fine tuning of the initial weights. In addition, the learned weights are highly diverse in shape which implies that the optimal weights do not need to be fine tuned too.

5.4 Summary

In this chapter we showed that training nonlinear recurrent networks to perform benchmark tasks like the bisection hyperacuity task is possible within the change-based coding framework. We also examined this over a range of different tasks and showed that learning them is also possible and in fact is easy and fast within the change-based framework. To the best of our knowledge, this is for the first time that learning recurrent models of perceptual learning tasks is demonstrated. The computational cost of BPTT increases exponentially with the number of iterations of the recurrent network. Learning by BPTT within the change-based framework became possible since it operates on early iterations. Learning allows us to apply the change-based method to a broader range of computations and tasks and we showed examples of that in this chapter.

We also developed another learning rule that maximized the first order signal to noise ratio and we showed that it also learned the optimal weights, indicating that the first order terms are sufficient to learn the task. In sum, the results that we discussed in this chapter show that the change-based method can be applied to a broader class of computations and provide further theoretical support that coding information by the temporal structure of the neural activity is able to solve an important class of computational problems that have not been solved previously by the conventional methods.

Contributions and future directions

The major accomplishments of this thesis are as follows:

1- We introduced the notion of change-based coding for nonlinear recurrent networks. According to this idea, networks encode information by the way their state changes over time. We studied the case in which information is coded by the change in a statistic of the population activity (the Center of Mass of the neural activity in this case), and showed that this coding strategy is able to solve a class of invariant discrimination tasks near-optimally. We studied the bisection task as a particular example of this. The best previous performance based on conventional attractor-based processing had been reported in (Zhaoping et al., 2003), however, this was far from optimal. We also discussed the major problems of the attractor-based approach in solving invariant discrimination tasks.

2- We showed that the change-based coding is very robust to high levels of noise corrupting the ongoing activity of the network. In contrast, attractor-based methods are sensitive to dynamical noise due to the random walk such networks perform in their null stable attractor, which results in the loss of encoded information.

3- We also showed that the change-based method is robust to noise added to the recurrent weights. This is important since synapses have been shown to be noisy and unreliable, and it is also known that the temporal structure of spike patterns dynamically modifies their efficacy (Markram and Tsodyks, 1996).

4- To understand the workings of the change-based method, we linearized the network around its initial state and showed that there is an eigenvector of the subweight matrix associated with the active neurons that plays a key role in change-based processing. This eigenvector is orthogonal to the required invariance; therefore it is not affected by the jitter inherent in the task. The eigenvector is also sensitive to the signal, detecting it as well as the ideal observer. The time constant associated with this eigenvector is of the order of a few hundreds of milliseconds; this allows information that is presented within this range of time to be combined optimally. Finally, the sign of the coefficient of this eigenvector determines whether the initial movement of the Center of Mass will be leftwards or rightwards.

5- We showed that the same variable that is the basis for decision in change-based processing (change of CoM here), also reflects (linearly) the amount of evidence gathered in favour of the decision (at least for a range of evidence strengths). In other words, the change-based method is able to represent uncertainty about decoding too.

6- We also studied the change-based method in the context of the conventional attractor-based method and showed a trade-off between the two. To this end, we first showed that it is possible to perform change-based coding near the attractor associated with conventional attractor-based coding. Next we showed that in this regime, the change-based method works because the attractor-based method performs sub-optimally.

7- Finally, we studied learning within change-based coding framework. We used the BPTT algorithm to learn the optimal weights for change-based coding, showing not only that it is possible to learn the optimal weights, but also that it is fast and requires no fine tuning of the initial condition. Learning allowed us to apply the change-based method to a broader range of invariant discrimination tasks. We applied it to several example tasks and we showed that learning optimal weights is possible and easy and fast.

One interesting future direction is to find other learning rules that are even faster and computationally less expensive than BPTT or the linearized signal to noise technique that we used. We showed that on average, some frequencies in the Toeplitz structure of the connection matrix played a more important role than others. It would be interesting to see how learning could be sped up through concentrating learning on those frequencies that carry more information. In light of our analysis in chapter 3, it would be interesting to design the optimal weights according to the shape of the effective eigenvector. That would requite designing a weight matrix for which the effective eigenvector has the highest eigenvalue (and therefore is the slowest one) and the weight matrix is translation invariant.

Another interesting future direction would be to code more than one variable by different statistics at the same time. For instance, imagine that in the bisection task, the two outer bars have the same contrast but the middle bar has a higher or lower contrast relative to the outer bars. The task is to decide if the middle bar is closer to the right or to the left bar and also whether its contrast is higher or lower than the outer bars. The CoM statistics can be used to solve the bisection. We could conceive of other statistics of the network activity that measure whether the activity shrinks or expands (spatially) during the early evolution of the network, and use it to encode contrast information. In particular, the breadth of activity might decrease if the contrast is higher and increase if it is lower. Since very many weights seem to perform bisection task optimally, it is plausible that a set could be found that performs both tasks as well as their corresponding ideal observers.

References

Adini Y, Sagi D, Tsodyks M (1997) Excitatory-inhibitory network in the visual cortex: psychophysical evidence. Proc Natl Acad Sci U S A 94:10426-10431.

Adini Y, Sagi D, Tsodyks M (2002) Context-enabled learning in the human visual system. Nature 415:790-793.

Adrian E (1926) The impulses produced by sensory nerve endings: Part I. J Physiol 61:49-72.

Ahissar M, Hochstein S (1997) Task difficulty and the specificity of perceptual learning. Nature 387:401-406.

Alais D, Burr D (2004) The ventriloquist effect results from near-optimal bimodal integration. Curr Biol 14:257-262.

Alpern M (1972) Eye movements. In: Handbook of Sensory Physiology (Jameson D, Hurvich L, eds). Berlin Springer

Amit D (1995) The Hebbian paradigm reintegrated: Local reverberations as internal representations. Behavioral and Brain Sciences 18:617-657.

Anderson C, Van Essen D (1987) Shifter circuits: a computational strategy for dynamic aspects of visual processing. Proc Natl Acad Sci U S A 84:6297-6301.

Andrew A, L (1973) Eigenvectors of Certain Matrices. In: Linear Algebra and its Applications, pp 151-162.

Attwell D, Laughlin S (2001) An energy budget for signaling in the grey matter of the brain. J Cereb Blood Flow Metab 21:1133-1145.

Beck J, Ma W, Latham P, Pouget A (2007) Probabilistic population codes and the exponential family of distributions. Prog Brain Res 165:509-519.

Ben-Yishai R, Bar-Or R, Sompolinsky H (1995) Theory of orientation tuning in visual cortex. Proc Natl Acad Sci U S A 92:3844-3848.

Britten K, Newsome W, Shadlen M, Celebrini S, Movshon J (1996) A relationship between behavioral choice and the visual responses of neurons in macaque MT. Vis Neurosci 13:87-100.

Brody C, Hernández A, Zainos A, Romo R (2003) Timing and neural encoding of somatosensory parametric working memory in macaque prefrontal cortex. Cereb Cortex 13:1196-1207.

Brunel N (2003) Dynamics and plasticity of stimulus-selective persistent activity in cortical network models. Cereb Cortex 13:1151-1161.

Buracas G, Zador A, DeWeese M, Albright T (1998) Efficient discrimination of temporal patterns by motion-sensitive neurons in primate visual cortex. Neuron 20:959-969.

Camperi M, Wang X-J (1997) Modeling delay-period activity in the prefrontal cortex during working memory tasks In: Computational Neuroscience: Trends in Research (Bower J, ed), pp 273-279. New York & London Plenum Press.

Carandini M, Heeger D (1994) Summation and division by neurons in primate visual cortex. Science 264:1333-1336.

Carandini M, Heeger D, Movshon J (1997) Linearity and normalization in simple cells of the macaque primary visual cortex. J Neurosci 17:8621-8644.

Compte A, Brunel N, Goldman-Rakic P, Wang X (2000) Synaptic mechanisms and network dynamics underlying spatial working memory in a cortical network model. Cereb Cortex 10:910-923.

Crist R, Kapadia M, Westheimer G, Gilbert C (1997) Perceptual learning of spatial localization: specificity for orientation, position, and context. J Neurophysiol 78:2889-2894.

Deneve S, Latham P, Pouget A (1999) Reading population codes: a neural implementation of ideal observers. Nat Neurosci 2:740-745.

Deneve S, Latham P, Pouget A (2001) Efficient computation and cue integration with noisy population codes. Nat Neurosci 4:826-831.

Desimone R (1991) Face-selective cells in the temporal cortex of monkeys. In, pp 1-8: Journal of Cognitive Neuroscience

Desimone R, Albright TD, Gross CG, Bruce C (1984) Stimulus-selective properties of inferior temporal neurons in the macaque. J Neurosci 4:2051-2062.

Dorais A, Sagi D (1997) Contrast masking effects change with practice. Vision Res 37:1725-1733.

Dosher B, Lu Z (1998) Perceptual learning reflects external noise filtering and internal noise reduction through channel reweighting. Proc Natl Acad Sci U S A 95:13988-13993.

Dosher B, Lu Z (1999) Mechanisms of perceptual learning. Vision Res 39:3197-3221.

Douglas R, Martin K (1989) A canonical microcircuit for neocortex. Neural Computation 1:480–488.

Douglas R, Martin K (1991) A functional microcircuit for cat visual cortex. J Physiol 440:735-769.

Dragoi V, Sharma J, Sur M (2000) Adaptation-induced plasticity of orientation tuning in adult visual cortex. Neuron 28:287-298.

Ernst M, Banks M (2002) Humans integrate visual and haptic information in a statistically optimal fashion. Nature 415:429-433.

Fabre-Thorpe M, Richard G, Thorpe S (1998) Rapid categorization of natural images by rhesus monkeys. Neuroreport 9:303-308.

Fahle M, Edelman S, Poggio T (1995) Fast perceptual learning in hyperacuity. Vision Res 35:3003-3013.

Fiorentini A, Berardi N (1980) Perceptual learning specific for orientation and spatial frequency. Nature 287:43-44.

Fukushima K (1980) Neocognitron: a self organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol Cybern 36:193-202.

Ganguli S, Bisley J, Roitman J, Shadlen M, Goldberg M, Miller K (2008) One-dimensional dynamics of attention and decision making in LIP. Neuron 58:15-25.

Gilbert C (1994) Early perceptual learning. Proc Natl Acad Sci U S A 91:1195-1197.

Gold J, Bennett P, Sekuler A (1999) Signal but not noise changes with perceptual learning. Nature 402:176-178.

Gross CG, Rocha-Miranda CE, Bender DB (1972) Visual properties of neurons in inferotemporal cortex of the Macaque. J Neurophysiol 35:96-111.

Hafting T, Fyhn M, Molden S, Moser MB, Moser EI (2005) Microstructure of a spatial map in the entorhinal cortex. Nature 436:801-806.

Hahnloser R, Sarpeshkar R, Mahowald M, Douglas R, Seung H (2000) Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. Nature 405:947-951.

Haykin S (1998) Neural Networks: A Comprehensive Foundation, 2 Edition: Prentice Hall.

Heeger D, Simoncelli E, Movshon J (1996) Computational models of cortical visual processing. Proc Natl Acad Sci U S A 93:623-627.

Hinton G, Dayan P, Frey B, Neal R (1995) The "wake-sleep" algorithm for unsupervised neural networks. Science 268:1158-1161.

Hopfield J (1982) Neural networks and physical systems with emergent collective computational abilities. Proc Natl Acad Sci U S A 79:2554-2558.

Houweling A, Brecht M (2008) Behavioural report of single neuron stimulation in somatosensory cortex. Nature 451:65-68.

Hubel D, Wiesel T (1962) Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. J Physiol 160:106-154.

Hubel DH, Wiesel TN (1959) Receptive fields of single neurones in the cat's striate cortex. J Physiol 148:574-591.

Huber D, Petreanu L, Ghitani N, Ranade S, Hromádka T, Mainen Z, Svoboda K (2008) Sparse optical microstimulation in barrel cortex drives learned behaviour in freely moving mice. Nature 451:61-64.

Hung C, Kreiman G, Poggio T, DiCarlo J (2005) Fast readout of object identity from macaque inferior temporal cortex. Science 310:863-866.

Jacobs R (1999) Optimal integration of texture and motion cues to depth. Vision Res 39:3621-3629.

Jaeger H, Haas H (2004) Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. Science 304:78-80.

Karni A, Sagi D (1991) Where practice makes perfect in texture discrimination: evidence for primary visual cortex plasticity. Proc Natl Acad Sci U S A 88:4966-4970.

Karni A, Sagi D (1993) The time course of learning a visual skill. Nature 365:250-252.

Körding K, Wolpert D (2004) Bayesian integration in sensorimotor learning. Nature 427:244-247.

Li Z, Dayan P (2000) NIPS 2000. In: Position variance, recurrence and perceptual learning (TK L, TG D, V T, eds), pp 31–37. Cambridge, MA: MIT Press.

Logothetis NK, Sheinberg DL (1996) Visual object recognition. Annu Rev Neurosci 19:577-621.

Lu Z, Dosher B (2004) Perceptual learning retunes the perceptual template in foveal orientation identification. J Vis 4:44-56.

Ma W, Beck J, Latham P, Pouget A (2006) Bayesian inference with probabilistic population codes. Nat Neurosci 9:1432-1438.

Maass W, Natschläger T, Markram H (2002) Real-time computing without stable states: a new framework for neural computation based on perturbations. Neural Comput 14:2531-2560.

Machens C, Romo R, Brody C (2005) Flexible control of mutual inhibition: a neural model of two-interval discrimination. Science 307:1121-1124.

Mainen Z, Sejnowski T (1995) Reliability of spike timing in neocortical neurons. Science 268:1503-1506.

Mareschal I, Dakin S, Bex P (2006) Dynamic properties of orientation discrimination assessed by using classification images. Proc Natl Acad Sci U S A 103:5131-5136.

Markram H, Tsodyks M (1996) Redistribution of synaptic efficacy between neocortical pyramidal neurons. Nature 382:807-810.

Maunsell J, van Essen D (1983a) The connections of the middle temporal visual area (MT) and their relationship to a cortical hierarchy in the macaque monkey. J Neurosci 3:2563-2586.

Maunsell J, Van Essen D (1983b) Functional properties of neurons in middle temporal visual area of the macaque monkey. II. Binocular interactions and sensitivity to binocular disparity. J Neurophysiol 49:1148-1167.

Maunsell J, Van Essen D (1983c) Functional properties of neurons in middle temporal visual area of the macaque monkey. I. Selectivity for stimulus direction, speed, and orientation. J Neurophysiol 49:1127-1147.

Mazor O, Laurent G (2005) Transient dynamics versus fixed points in odor representations by locust antennal lobe projection neurons. Neuron 48:661-673.

McKee S, Westheimer G (1978) Improvement in vernier acuity with practice. Percept Psychophys 24:258-262.

Moazzezi R, Dayan P (2008) Change-based inference for invariant discrimination. Network 19:236-252.

Moazzezi R, Dayan P (2010) Change-based inference in attractor nets: linear analysis. Neural Comput 22:3036-3061.

Mollon J, Danilova M (1996) Three remarks on perceptual learning. Spat Vis 10:51-58.

Naya Y, Sakai K, Miyashita Y (1996) Activity of primate inferotemporal neurons related to a sought target in pair-association task. Proc Natl Acad Sci U S A 93:2664-2669.

Nienborg H, Cumming BG (2009) Decision-related activity in sensory neurons reflects more than a neuron's causal effect. Nature 459:89-92.

O'Keefe J, Dostrovsky J (1971) The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. Brain Res 34:171-175.

Olshausen B, Anderson C, Van Essen D (1993) A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information. J Neurosci 13:4700-4719.

Parker A, Hawken M (1985) Capabilities of monkey cortical cells in spatial-resolution tasks. J Opt Soc Am A 2:1101-1114.

Parkosadze K, Otto T, Malania M, Kezeli A, Herzog M (2008) Perceptual learning of bisection stimuli under roving: slow and largely specific. J Vis 8:5.1-8.

Perrett D, Oram M (1993) Neurophysiology of shape processing. In, pp 317-333: *Image* Vision *Computing*

Perrett DI, Hietanen JK, Oram MW, Benson PJ (1992) Organization and functions of cells responsive to faces in the temporal cortex. Philos Trans R Soc Lond B Biol Sci 335:23-30.

Pinto N, Cox D, DiCarlo J (2008) Why is real-world visual object recognition hard? PLoS Comput Biol 4:e27.

Poggio T, Girosi F (1990) Regularization Algorithms for Learning That Are Equivalent to Multilayer Networks. Science 247:978-982.

Poggio T, Fahle M, Edelman S (1992) Fast perceptual learning in visual hyperacuity. Science 256:1018-1021.

Pouget A, Zhang K, Deneve S, Latham P (1998) Statistically efficient estimation using population coding. Neural Comput 10:373-401.

Raiguel S, Vogels R, Mysore S, Orban G (2006) Learning to see the difference specifically alters the most informative V4 neurons. J Neurosci 26:6589-6602.

Rainer G, Miller E (2002) Timecourse of object-related neural activity in the primate prefrontal cortex during a short-term memory task. Eur J Neurosci 15:1244-1254.

Rao R (1999) An optimal estimation approach to visual perception and learning. Vision Res 39:1963-1989.

Regan D, Beverley K (1985) Postadaptation orientation discrimination. J Opt Soc Am A 2:147-155.

Reinagel P (2001) How do visual neurons respond in the real world? Curr Opin Neurobiol 11:437-442.

Reinagel P, Reid RC (2002) Precise firing events are conserved across neurons. J Neurosci 22:6837-6841.

Renart A, Song P, Wang X (2003) Robust spatial working memory through homeostatic synaptic scaling in heterogeneous cortical networks. Neuron 38:473-485.

Riesenhuber M, Poggio T (1999) Hierarchical models of object recognition in cortex. Nat Neurosci 2:1019-1025.

Riesenhuber M, Poggio T (2000) Models of object recognition. Nat Neurosci 3 Suppl:1199-1204.

Ringach DL, Hawken MJ, Shapley R (1997) Dynamics of orientation tuning in macaque primary visual cortex. Nature 387:281-284.

Roitman J, Shadlen M (2002) Response of neurons in the lateral intraparietal area during a combined visual discrimination reaction time task. J Neurosci 22:9475-9489.

Rumelhart D, Hinton G, Williams R (1986a) Learning internal representations by error propagation. In: *Parallel Distributed Processing*. Cambridge, MA: MIT Press.

Rumelhart DE, Hinton GE, Williams RJ (1986b) Learning representations by back-propagating errors. Nature 323:533-536.

Saarinen J, Levi D (1995) Perceptual learning in vernier acuity: what is learned? Vision Res 35:519-527.

Schoups A, Vogels R, Orban G (1995) Human perceptual learning in identifying the oblique orientation: retinotopy, orientation specificity and monocularity. J Physiol 483 (Pt 3):797-810.

Schoups A, Vogels R, Qian N, Orban G (2001) Practising orientation identification improves orientation coding in V1 neurons. Nature 412:549-553.

Serre T, Oliva A, Poggio T (2007) A feedforward architecture accounts for rapid categorization. Proc Natl Acad Sci U S A 104:6424-6429.

Seung H (1996) How the brain keeps the eyes still. Proc Natl Acad Sci U S A 93:13339-13344.

Seung H, Sompolinsky H (1993) Simple models for reading neuronal population codes. Proc Natl Acad Sci U S A 90:10749-10753.

Shadlen M, Newsome W (1998) The variable discharge of cortical neurons: implications for connectivity, computation, and information coding. J Neurosci 18:3870-3896.

Shafi M, Zhou Y, Quintana J, Chow C, Fuster J, Bodner M (2007) Variability in neuronal activity in primate cortex during working memory tasks. Neuroscience 146:1082-1108.

Snippe H, Koenderink J (1992) Information in channel-coded systems: correlated receivers. Biol Cybern 67:183-190.

Steinman R, Haddad G, Skavenski A, Wyman D (1973) Miniature eye movement. Science 181:810-819.

Swindale N, Cynader M (1986) Vernier acuity of neurones in cat visual cortex. Nature 319:591-593.

Tanaka K (1996) Inferotemporal cortex and object vision. Annu Rev Neurosci 19:109-139.

Tanaka K, Saito H, Fukada Y, Moriya M (1991) Coding visual images of objects in the inferotemporal cortex of the macaque monkey. J Neurophysiol 66:170-189.

Teich A, Qian N (2003) Learning and adaptation in a recurrent model of V1 orientation selectivity. J Neurophysiol 89:2086-2100.

Thorpe S, Fize D, Marlot C (1996) Speed of processing in the human visual system. Nature 381:520-522.

Tsodyks M, Gilbert C (2004) Neural networks and perceptual learning. Nature 431:775-781.

Tsodyks M, Adini Y, Sagi D (2004) Associative learning in early vision. Neural Netw 17:823-832.

Van Essen D, Anderson C, Felleman D (1992) Information processing in the primate visual system: an integrated systems perspective. Science 255:419-423.

Van Rullen R, Thorpe S (2001) Rate coding versus temporal order coding: what the retinal ganglion cells tell the visual cortex. Neural Comput 13:1255-1283.

van Vreeswijk C, Sompolinsky H (1996) Chaos in neuronal networks with balanced excitatory and inhibitory activity. Science 274:1724-1726.

VanRullen R, Thorpe S (2001) Is it a bird? Is it a plane? Ultra-rapid visual categorisation of natural and artifactual objects. Perception 30:655-668.

Vinje W, Gallant J (2000) Sparse coding and decorrelation in primary visual cortex during natural vision. Science 287:1273-1276.

Wang X (2001) Synaptic reverberation underlying mnemonic persistent activity. Trends Neurosci 24:455-463.

Watt R, Campbell F (1985) Vernier acuity: interactions between length effects and gaps when orientation cues are eliminated. Spat Vis 1:31-38.

Weiss Y, Edelman S, Fahle M (1993) Models of perceptual learning in Vernier hyperacuity. Neural Computation 5:695-718.

Weiss Y, Simoncelli E, Adelson E (2002) Motion illusions as optimal percepts. Nat Neurosci 5:598-604.

Westheimer G (1981) Visual hyperacuity. Prog Sensory Physiol 1:1-37.

Westheimer G, McKee S (1977a) Integration regions for visual hyperacuity. Vision Res 17:89-93.

Westheimer G, McKee S (1977b) Spatial configurations for visual hyperacuity. Vision Res 17:941-947.

Wong K, Huk A, Shadlen M, Wang X (2007) Neural circuit dynamics underlying accumulation of time-varying evidence during perceptual decision making. Front Comput Neurosci 1:6.

Wu S, Amari S (2005) Computing with continuous attractors: stability and online aspects. Neural Comput 17:2215-2239.

Wu S, Nakahara H, Amari S (2001) Population coding with correlation and an unfaithful model. Neural Comput 13:775-797.

Xiao L, Zhang J, Wang R, Klein S, Levi D, Yu C (2008) Complete transfer of perceptual learning across retinal locations enabled by double training. Curr Biol 18:1922-1926.

Yang T, Maunsell J (2004) The effect of perceptual learning on neuronal responses in monkey visual area V4. J Neurosci 24:1617-1626.

Yang Y, DeWeese M, Otazu G, Zador A (2008) Millisecond-scale differences in neural activity in auditory cortex can drive decisions. Nat Neurosci 11:1262-1263.

Zemel R, Dayan P, Pouget A (1998) Probabilistic interpretation of population codes. Neural Comput 10:403-430.

Zhang K (1996) Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: a theory. J Neurosci 16:2112-2126.

Zhaoping L, Herzog M, Dayan P (2003) Nonlinear ideal observation and recurrent preprocessing in perceptual learning. Network 14:233-247.