LETTER **Communicated by Mark Goldman**

Change-Based Inference in Attractor Nets: Linear Analysis

Reza Moazzezi
rezamoazzezi@berkeley.edu
Peter Dayan
dayan@gatsby.ucl.ac.uk
Gatsby Computational Neuroscience Unit, UCL, London, WC1N 3AR, U.K.

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. We have recently suggested an alternative interpretation according to which computations are realized by systematic changes in the states of such networks over time. This way of performing computations is fast, accurate, readily learnable, and robust to various forms of noise. Here we analyze the computation of stimulus discrimination in this change-based setting, relating it directly to the computation of stimulus estimation in the conventional attractor-based view. We use a common linear approximation to compare the two methods and show that perfect performance at estimation implies chance performance at discrimination.

1 Introduction

Dynamical interactions among recurrently connected networks of nonlinear cortical neurons are widely believed to play a central role in information processing (Douglas, Martin, & Whitteridge, 1989). One prominent theoretical account of these networks considers them in terms of attractors, with computation consisting of mapping inputs by recurrent dynamics into particular attractors or particular locations on a continuous attractor. A readout mechanism, which is often some form of feedforward network, then reports a characteristic of the final, attracted state. We describe these as forms of conventional attractor-based computation, and the value that emerges from the readout as attractor-based readout.

R. Moazzezi is now at the Redwood Center for Theoretical Neuroscience, Helen Wills Neuroscience Institute, University of California, Berkeley.

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; Pouget, Zhang, Deneve, & Latham, 1998; Deneve, Latham, & Pouget, 1999, 2001; Wang, 2001; Wu & Amari, 2005; Wu, Nakahara, & Amari, 2001; Renart, Song, & Wang, 2003; Camperi & Wang, 1997), involves (null-stable) attractive manifolds that 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).

Unfortunately, there is little evidence that such attractor states exist in cortical networks. Neurons in sensory cortical areas rarely exhibit persistent activity in vivo (Reinagel, 2001; Vinje & Gallant, 2000), which strongly 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 (Brody, Hernandez, Zainos, & Romo, 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.

These facts motivated our recent investigation of the information processing that recurrent network dynamics can perform during the transient evolution of the network's state toward, but without necessarily reaching, a surface attractor (Moazzezi & Dayan, 2008). We showed that significant computations can be performed by the mapping from inputs to changes over time in a statistic of the network's activity. We call this *change-based computation* and the value that emerges from the change in the activity statistic *change-based readout*.

Figure 1A explains the method in the discrimination context that we investigated previously. In this case, input patterns came from one of two classes, and the recurrent network had to determine which. The figure shows an abstract rendition of the (N-dimensional) state space of the network; its weights were set to realize two different continuous attractor manifolds, M1 and M2. The trajectory labeled X, Y, Z, and T shows the evolution of the state of the network from the initial input pattern (X) to the converged state (T) on the attractor.

Conventional attractor-based readout decides about the class of X based on the attractor to which it converges (M1 represents class 1, and M2 represents class 2; for example, here, T is on M1, and so class 1 would be reported). In contrast, the change-based method decides on the input's class based on comparing a statistic of the neural activity at two intermediate points (e.g., Y and Z) as the network's state evolves from X to T. We considered a statistic $\mu : \Re^N \rightarrow \Re$ that maps every state to a scalar. The choice of

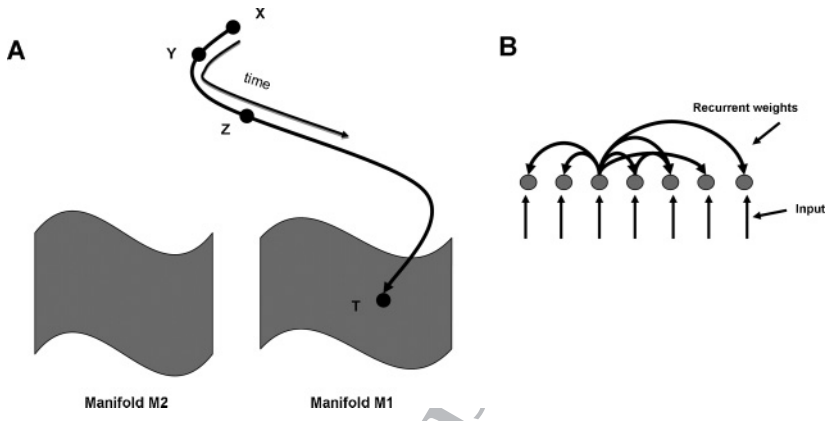

Figure 1: Change-based readout. (A) Point X represents the state of the network associated with the input. As time evolves, the network's trajectory takes it through new states (Y, Z, . . .), finally converging to a point on the low-dimensional manifold M1. Conventional attractor-based readout is based on the final state of the attractor—in this case of discrimination, whether point T is on manifold M1 or M2. By contrast, change-based readout decides based on how a statistic of the neural activity (e.g., the center of mass of the neural activity) changes over time between two points, say X and Y, long before convergence. (B) The model consists of units arranged on a line according to their preferred tuning values. They are recurrently connected. The input is presented only in the first iteration, after which the network evolves toward an attractor.

class is determined by whether this statistic increases or decreases (whether $\mu(Y) > \mu(Z)$ or $\mu(Y) < \mu(Z)$). The statistic we considered is the center of mass (CoM) of the neural activity.

In our previous paper (Moazzezi & Dayan, 2008), we showed that the performance of the change-based method is statistically near optimal, extracting almost all the available information in this and a variety of other tasks. Performance is also robust to very high levels of dynamical noise. Change-based readout permits fast decoding; for instance, in a network with time constants of about 20 ms, near-optimal performance was achieved within 100 ms of the onset of activity. Inferential quality is insensitive to the timing of points Y and Z as long as these points are far enough from each other and from the attractor manifolds. We have also shown that the change-based method allows easy learning using the biologically implausible backpropagation through time (BPTT) algorithm. Finally, change-based readout can offer a perfect solution to the problem of invariance that motivated the original study by Zhaoping, Herzog, and Dayan (2003) that inspired our previous paper. That is, the class of the input X has to be determined in a way that is invariant to particular of its features (in that

case, invariant to the location on a one-dimensional array at which it is presented). If the network's weights are invariant to these features (in that case, being translation invariant to that dimension), then change-based discrimination will also be invariant. In Figure 1A, the attractor manifolds M1 and M2 are defined by these dimensions of invariance. Invariance is critical for many neural computations.

In our previous work, we exhibited empirical evidence only for the near optimality of change-based readout, in a highly nonlinear regime (in the transition from X to Z) when the network's state is far from any underlying attractor (see Moazzezi & Dayan, 2008). This nonlinearity makes it quite difficult to analyze the change-based method theoretically and understand how it works. However, linear analysis is possible if the state of the network is near its attractor (or manifold attractor). Therefore, in this letter, we take a first step toward analyzing the method by considering a subtly different discrimination task from the one we considered in the previous work. In the new task, change-based readout operates near optimally in the linear regime near the network's line attractor and therefore allows linear analysis. In particular, 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). The similarity between the discrimination task that we consider in this letter and the conventional estimation task, together with the fact that both readout methods 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 be successful only 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 an 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 nonlinear 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.

2 Methods

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 (noncircular) version of one of the best-studied line attractor networks that 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 & Heeger, 1994; Pouget et al., 1998).

The network consists of N units arranged topographically on a line (see Figure 1B); 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, 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 formulas 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 2A). 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 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, not the absolute location of the high-contrast bar.

2.2 Discrimination Task

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 2A). The carrier is presented at location y, and the signal is presented at location $\varepsilon + y$, and so the task is to decide if $\varepsilon > 0$ or $\varepsilon < 0$ (see Figure 2A). Note that y represents the invariant dimension.

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

$$\bar{a}_{ic} = \frac{e^{-\frac{(y-x_i)^2}{2\sigma^2}}}{H}, \tag{2.1}$$

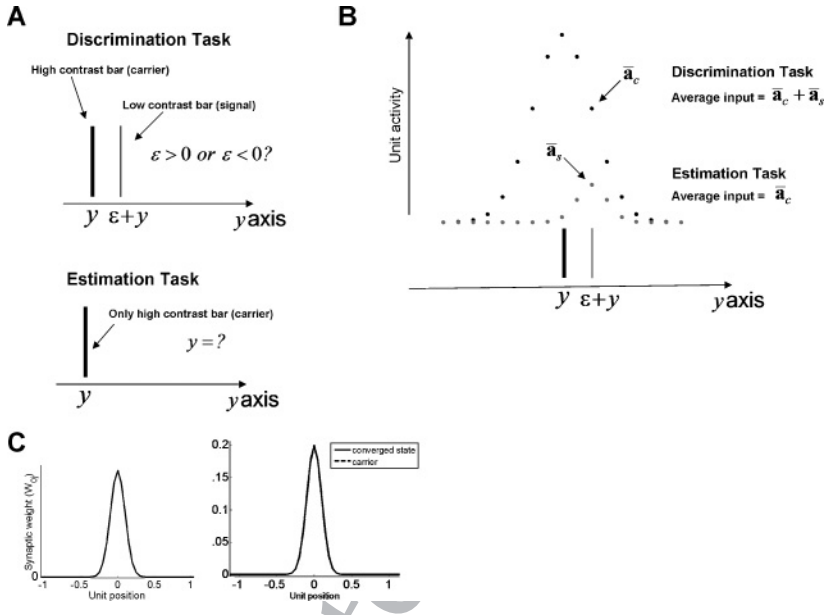

Figure 2: The discrimination and estimation tasks. (A) Top: Discrimination task. Two bars are presented simultaneously to the network, one with high contrast (called the carrier) and one with low contrast (the signal). The task is to decide if the signal is to the left or the right of the carrier. Bottom: 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 $\bar{\mathbf{a}}_c$, and the low-contrast bar (signal) generates the average hill of activity $\bar{\mathbf{a}}_s$. For the discrimination task, on each trial, the actual input is a noisy version of the linear sum of the two hills of activity $\bar{\mathbf{a}}_c + \bar{\mathbf{a}}_s$. For the 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, $\bar{\mathbf{a}}_c$. (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).

where in $\bar{a}_{ic}$, 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

$$\bar{a}_{is} = \frac{\zeta e^{-\frac{((\varepsilon+y)-x_i)^2}{2\eta^2\sigma^2}}}{H}, \tag{2.2}$$

where in $\bar{a}_{is}$, 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

$$\bar{a}_i = \bar{a}_{ic} + \bar{a}_{is}. \tag{2.3}$$

These average activities are cartooned in Figure 2B.

We consider two noise models: scaled Poisson noise and gaussian mean dependent noise. For both noise models, the actual input to the network ($\mathbf{a} = (a_1, a_2, \ldots, a_N)$ is generated in two stages. The first stage controls the strength of the noise, while the second stage is necessary in order to have $\langle \mathbf{a} \rangle = \bar{\mathbf{a}}_c$ (in the absence of the signal) where $\langle \mathbf{a} \rangle$ denotes average over infinite random draws of $\mathbf{a}$ and $\bar{\mathbf{a}}_c = (\bar{a}_{1c}, \bar{a}_{2c}, \ldots, \bar{a}_{Nc})$. 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 nonlinear 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 $\mathbf{a}' = (a'_1, a'_2, \ldots, a'_N)$ according to a Poisson distribution with mean $q\lambda\bar{\mathbf{a}}$ (the reason to decompose the coefficient into q and λ is to simplify the comparison between Poisson and gaussian noise models):

$$P(a'_i | q\lambda\bar{a}_i) = \frac{e^{-q\lambda\bar{a}_i}}{(a'_i)!}(q\lambda\bar{a}_i)^{a'_i}, \tag{2.4}$$

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

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

$$\mathbf{a} = \frac{\mathbf{a}'}{q\lambda}. \tag{2.5}$$

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

$$P\left(a'_i \middle| \frac{\lambda\bar{a}_i}{q}\right) = \frac{1}{\sqrt{2\pi\left(\frac{\lambda\bar{a}_i}{q}\right)}} \exp\left(-\frac{\left(a'_i - \lambda\bar{a}_i\right)^2}{2\left(\frac{\lambda\bar{a}_i}{q}\right)}\right), \tag{2.6}$$

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

$$\mathbf{a} = \frac{\mathbf{a}'}{\lambda}. \tag{2.7}$$

As we mentioned above, defining the input to the network by equations 2.5 (for Poisson noise) and 2.7 (for gaussian noise) ensures that $\langle\mathbf{a}\rangle = \bar{\mathbf{a}}_c$ (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:

$$\mathbf{u}^{n+1} = \mathbf{f}(\mathbf{W}\mathbf{u}^n), \tag{2.8}$$

where n indicates the iteration number, $\mathbf{u}^{(n)} = (u_1^{(n)}, u_2^{(n)}, \ldots, u_N^{(n)})$ and $\mathbf{u}^{(0)} = \mathbf{a}$ defines the input to the network (equations 2.5 and 2.7 for Poisson and gaussian noise, respectively). The weights $\mathbf{W}$ are translation invariant, and their form (see appendix A) is shown in Figure 2C, left. We define the weights $\mathbf{W}$ and the dynamics of the network $\mathbf{f}(.)$ such that if $\mathbf{u}^{(0)} = \bar{\mathbf{a}}_c$, then the converged state of the network, $\mathbf{u}^{(\infty)}$, and the carrier, $\bar{\mathbf{a}}_c$, are nearly the same (see Figure 2C, left). In general, the weights $\mathbf{W}$and the dynamics of the network $\mathbf{f}(.)$ are defined such that $\mathbf{u}^{(\infty)}$ and $\bar{\mathbf{a}}_c$ have nearly the same form (see Figure 2C, left), but they might not be at the same location because of the discrimination signal and the noise that corrupt $\bar{\mathbf{a}}_c$.

We follow Deneve et al. (2001) in defining the network's nonlinear activation function $\mathbf{f}(.) = (f_1(.), f_2(.), \ldots, f_N(.))$ as the squaring, normalizing nonlinearity,

$$f_i(\mathbf{r}) = \frac{r_i^2}{\sum_j r_j^2}, \tag{2.9}$$

where $\mathbf{r} = (r_1, r_2, \ldots, r_N)$is an arbitrary vector. Normalization has been shown to realize a particularly convenient (Deneve et al., 1999; Wu & Amari, 2005) and neurobiologically relevant (Heeger, Simoncelli, & Movshon, 1996; Carandini, Heeger, & Movshon, 1997) form of attractor network. However, change-based readout does not depend on this. In our earlier paper (Moazzezi & 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 (see Figure 2C right).

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 center of mass (CoM) of the neural activity in the network moves to the left or to the right. The CoM of the neural activity $\mathbf{u}^{(n)}$at iteration n is defined as

$$\mu\left(\mathbf{u}^{(n)}\right) = \frac{\sum_i u_i^{(n)} x_i}{\sum_i u_i^{(n)}}. \tag{2.10}$$

For this discrimination task, the decision about the sign of ε (which is the distance between signal and carrier; see Figure 2A) is based on the sign of $\mu(\mathbf{u}^{(\infty)}) - \mu(\mathbf{u}^{(0)})$ (which is equal to $\mu(\mathbf{u}^{(\infty)}) - \mu(\mathbf{a})$ given $\mathbf{u}^{(0)} = \mathbf{a}$). This is an example of change-based readout. Compared with our previous application of this method, which we described in section 1, 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.

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, $\bar{a}_i = \bar{a}_{ic}$ for all i. The noisy input activity $\mathbf{a}$ is generated from $\bar{\mathbf{a}}$, as for discrimination (equations 2.4 and 2.5 for Poisson noise and 2.6 and 2.7 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

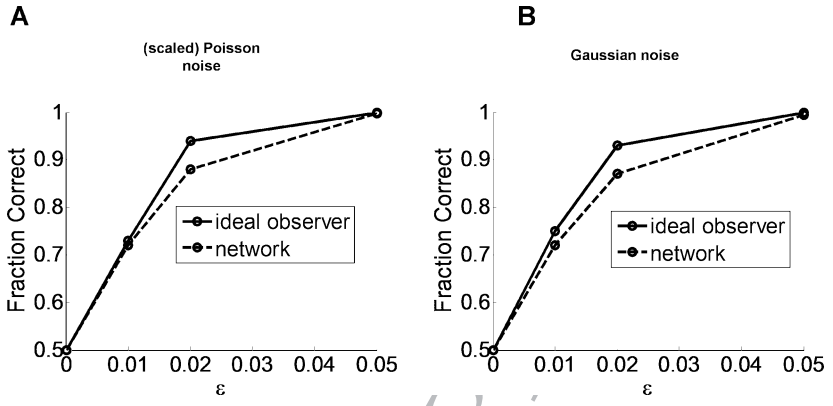

Figure 3: Performance of the network and the ideal observer. (A) Performance level of the ideal observer and the network as a function of ε for Poisson noise. (B) Same as A except that the noise is mean-dependent gaussian. In both cases, the network's performance is near optimal.

location on the attractor of the converged state. We call this attractor-based readout. Since the converged state is smooth, noise free, and unimodal, its location can be accurately characterized by its CoM, that is, $\mu(\mathbf{u}^{(\infty)})$.

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 that 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, nonlinear regime, with large signal and noise ($\zeta = 1, q = 1$). The network is found to have converged by 10 iterations.

3 Results

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 3A and 3B 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.

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 ($\mathbf{a}$). 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.

3.2 Linearization of the Network. Figure 4A shows the carrier, an example input ($\mathbf{a}$) generated from the carrier plus signal (equations 2.4 and 2.5 for Poisson noise and 2.6 and 2.7 for gaussian noise) and the output of the network after convergence. Figure 4B 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 $\mathbf{J}_{cb}$. This filter, which is shown in Figure 4C, predicts the change $\delta\mu_{cb}$ in the CoM of the neural activity as

$$\delta\mu_{cb} = \mu\left(\mathbf{u}^{(\infty)}\right) - \mu(\mathbf{a}) \approx \mathbf{J}_{cb}^{T}.(\mathbf{a} - \bar{\mathbf{a}}_c), \tag{3.1}$$

where $\bar{\mathbf{a}}_c = (\bar{a}_{1c}, \bar{a}_{2c}, \ldots, \bar{a}_{Nc})$ is the carrier, $\mathbf{a} = (a_1, a_2, \ldots, a_N)$ is the input to the network, and $\mathbf{u}^{(\infty)} = (u_1^{(\infty)}, u_2^{(\infty)}, \ldots, u_N^{(\infty)})$ is the converged state of the network. Note that the difference $\mathbf{a} - \bar{\mathbf{a}}_c$ in equation 3.1 includes the signal ($\bar{\mathbf{a}}_s$) for the discrimination task, where $\bar{\mathbf{a}}_s$ is defined as $\bar{\mathbf{a}}_s = (\bar{a}_{1s}, \bar{a}_{2s}, \ldots, \bar{a}_{Ns})$.

We can understand equation 3.1 intuitively (see Figure 4D). $\delta\mu_{cb}$ is the amount the CoM changes as the network evolves from its initial state $\mathbf{u}^{(0)} = \mathbf{a}$ to its final state $\mathbf{u}^{(\infty)}$. The change in CoM of the network activity between the input and the converged state ($\mu(\mathbf{u}^{(\infty)}) - \mu(\mathbf{a})$) is predicted based on the difference of the input pattern and the mean activity pattern associated with the carrier, that is, ($\mathbf{a} - \bar{\mathbf{a}}_c$). $\mathbf{a}$ is a noisy version of the underlying, smooth, mean activity pattern associated with the carrier $\bar{\mathbf{a}}_c + \bar{\mathbf{a}}_s$, and the change in CoM ($\mu(\mathbf{u}^{(\infty)}) - \mu(\mathbf{a})$) 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 $\mathbf{J}_{cb}$ from the Jacobian of the network (see appendix C).

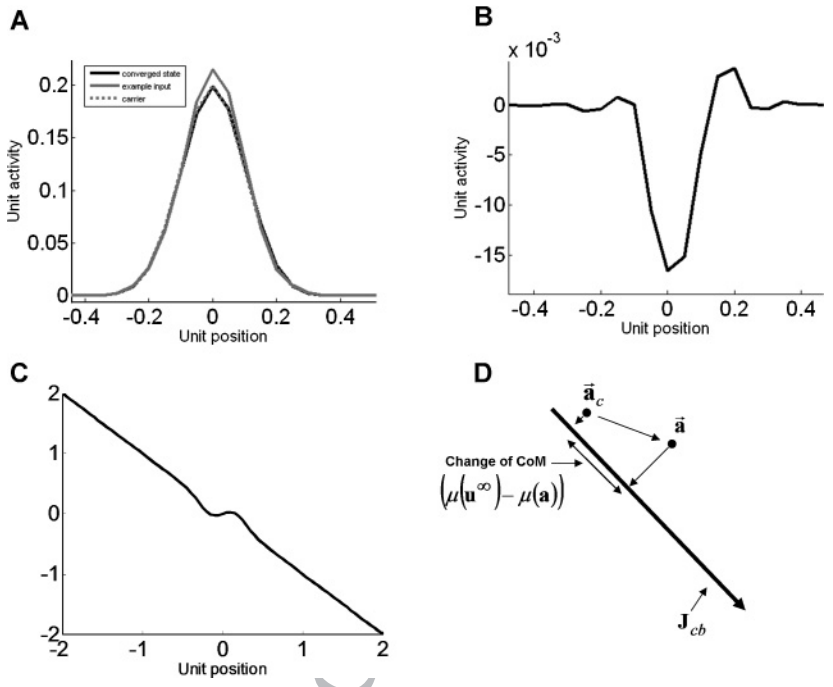

Figure 4: Network attractor and linearization. (A) Carrier (dashed gray line), an example input (solid gray line) and output (solid black line) of the network. (B) The difference between the input and the output of the network, which is very small relative to the size of the input. (C) The equivalent linear filter that predicts the change in the CoM. This was obtained by linearizing about the carrier. (D) Schematic description of equation 3.1. The change in CoM of the network activity between the input and the converged state ($\mu(\mathbf{u}^\infty) - \mu(\mathbf{a})$) is predicted based on the difference of the input pattern and the activity associated with the carrier ($\mathbf{a} - \bar{\mathbf{a}}_c$). The difference of the carrier ($\bar{\mathbf{a}}_c$) and example input ($\mathbf{a}$) is projected onto the change-based (linear) filter ($\mathbf{J}_{cb}$), and its magnitude and sign represent the magnitude and sign of the change in CoM, which is ($\mu(\mathbf{u}^\infty) - \mu(\mathbf{a})$).

We define the signal-to-noise ratio associated with the filter as

$$S/N = \frac{\left|\mathbf{J}_{cb}^T \bar{\mathbf{a}}_s\right|}{\sqrt{\mathbf{J}_{cb}^T \Sigma \mathbf{J}_{cb}}}, \tag{3.2}$$

where Σ is the noise covariance matrix of $\mathbf{a}$. Note that $\bar{\mathbf{a}}_s$ is a function of ε. Figure 5A shows the predicted signal-to-noise ratios for the filter of Figure 4C over the relevant range of ε. Figure 5B shows an example histogram for

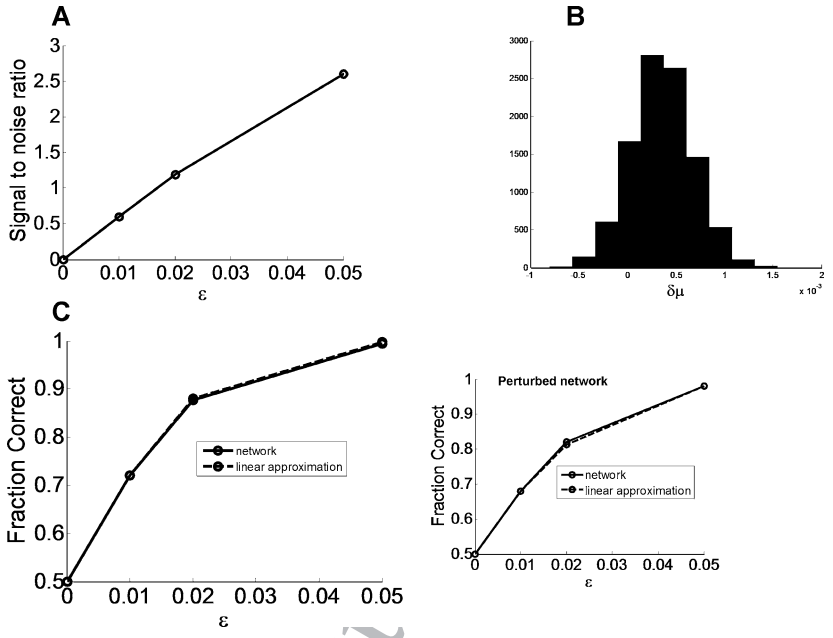

Figure 5: Linear performance. (A) The signal-to-noise ratio (in the change of CoM) predicted by the linear filter that was shown in Figure 4C as a function of ε. (B) An example histogram for the linear prediction of the change in the CoM for $\varepsilon = .02$. This is approximately gaussian because 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 8) 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.

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 5C confirms the empirical rectitude of this analysis.

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 (see Figure 6). 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 B). We will show

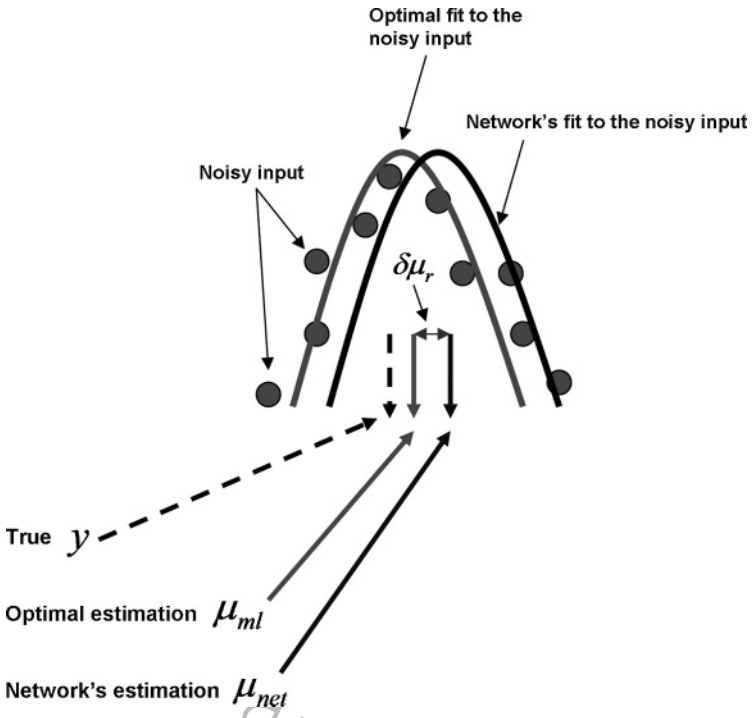

Figure 6: Change-based and attractor-based readout. This schematic shows a case in which noisy activity (circles) came from a true value of $y = 0$. The optimal ML estimate of y is μ_{ml}, which amounts to fitting the shape of the mean activity associated with the carrier to the actual activity (weighting 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 the change-based readout.

that in a formal sense, change-based discrimination exactly exploits the suboptimality of attractor-based estimation.

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

$$\mu_{ml} \approx \mathbf{J}_{ml}^T(\mathbf{a} - \bar{\mathbf{a}}_c) = \mathbf{J}_{ml}^T(\mathbf{n}), \tag{3.3}$$

where μ_{ml} is the optimal estimator of y given the input pattern $\mathbf{a}$, and equation 3.3 predicts this optimal estimation as a linear function of the difference of the input and carrier patterns. The index "ml" stands for "maximum

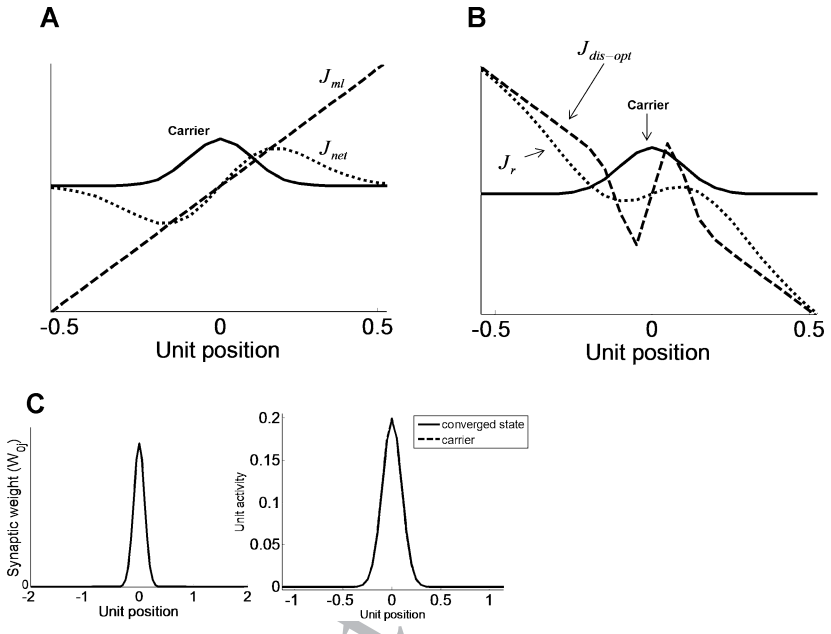

Figure 7: Optimal and network filters. (A) Estimation task. The dashed line shows the optimal filter for estimation. The dotted line shows the network's filter for estimation (J_{net}). The solid line shows the carrier. (B) Discrimination task. The dashed line shows the optimal filter for discrimination. 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. (C) Central portion of the translation-invariant weights (left) and the carrier and converged state of the network (right), the same as Figure 2C (both patterns have very nearly the same form).

likelihood" because the optimal estimator is the maximum likelihood estimator. The linear filter $\mathbf{J}_{ml}$ is derived in appendix B (see the dashed curve in Figure 7A). As we mentioned in section 3.1, for Poisson noise and low levels of gaussian noise, μ_{ml} is the CoM of input pattern $\mathbf{a}$.

If we denote the linear filter associated with attractor-based method by $\mathbf{J}_{net}$ (the dotted curve in Figure 7A), including the readout, then the network's estimate of y is

$$\mu_{net} \approx \mathbf{J}_{net}^T(\mathbf{a} - \bar{\mathbf{a}}_c) = \mathbf{J}_{net}^T(\mathbf{n}). \tag{3.4}$$

Note that the network's estimate of y is the CoM of the converged state of the network. Equation 3.4 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} (see Figure 6) can also be interpreted as the CoM of the converged state of the network. Attractor-based readout would therefore be optimal if $\mathbf{J}_{net} = \mathbf{J}_{ml}$, that is, if the optimal estimate and the network's estimate are the same at every trial ($\mu_{net} = \mu_{ml}$). Given that $\mathbf{J}_{net} \neq \mathbf{J}_{ml}$ (see Figure 7A), there is a difference between the network's estimate and the optimal estimate $\delta\mu_r = \mu_{net} - \mu_{ml}$. We can therefore write $\delta\mu_r \approx \mathbf{J}_r^T \mathbf{n}$, where $\mathbf{J}_r = \mathbf{J}_{net} - \mathbf{J}_{ml}$.

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, $\mathbf{J}_r$ is the linear filter associated with the change in CoM, that is, $\mathbf{J}_r = \mathbf{J}_{cb}$. If the network was a perfect estimator ($\mathbf{J}_{net} = \mathbf{J}_{ml}$), then $\mathbf{J}_{cb} = \mathbf{J}_r = \mathbf{J}_{net} - \mathbf{J}_{ml} = \mathbf{0}$, and therefore by equation 3.1, $\delta\mu_{cb} = 0$, 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 $\mathbf{J}_{net} \neq \mathbf{J}_{ml}$. In appendix B, we derive the form of the optimal filter $\mathbf{J}_{dis-opt}$ for discrimination.

We can also compare the forms of the actual and optimal linear filters for both tasks. Figure 7B presents this comparison for the case of discrimination (also showing the weight matrix, the carrier, and the converged state of the network); Figure 7A presents the comparison for estimation. The filter for estimation is close to the optimal form in its 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 7A shows that the width of the network's linear filter is slightly too narrow for estimation. Figure 8C 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 8A 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 8B shows that the discrimination filter is now much

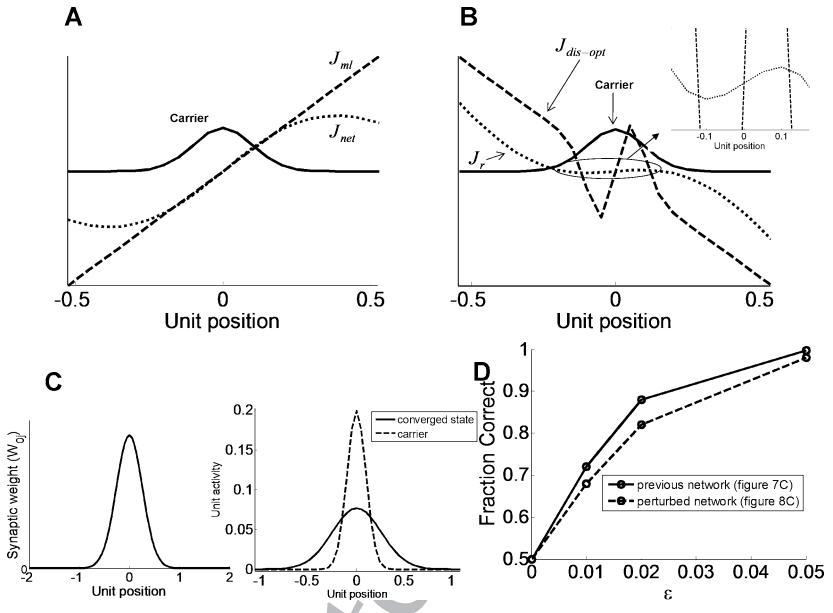

Figure 8: The filters of a perturbed network. The same convention is used for all the plots as in Figure 7. (A) Estimation task. The extra breadth makes the network's filter a better match to the optimal estimation filter. (B) Discrimination task. The broader weights lead to a pattern of activity of the units on the attractor being quite different. This makes the network's change-based filter very shallow. (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 Figures 7C and 8C for discrimination task.

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 8D compares the performance levels of the two sets of weights.

Note that the linear approximation predicts the performance of the perturbed network accurately (see Figure 5C inset), even though the widths of the average input and output patterns are different (Figure 8C, right). As we mentioned earlier, 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 7C, the input is very close to the line attractor. By contrast, for the case of Figure 8C, the input is far from the line attractor. Nevertheless, the linear analysis accurately predicts the performance of the network (see Figure 5C 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

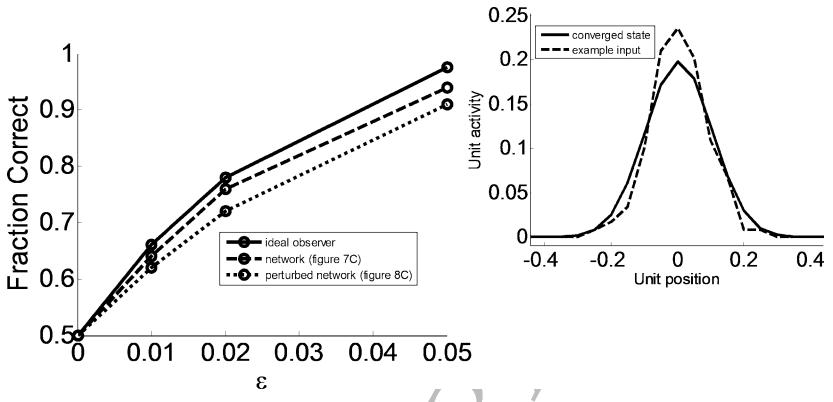

Figure 9: High-noise regime. Performance levels of the ideal observer, the network shown in Figure 7C, and the perturbed network shown in Figure 8C for the discrimination task in the face of high noise. The inset shows an example input and the average input to the network for Poisson noise. They are now substantially different.

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.

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 (see Figure 3 and section 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 (e.g., to overcome noise in the actual readout itself), then this will force a decrease in the performance of attractor-based estimation and there will be a trade-off between the performance of the change-based readout for the discrimination task and the performance of the attractor-based readout for the estimation task.

For the convenience of the analysis, we have so far considered low levels of input noise (and signal). Figure 9 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. 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 8) has the

same effect as in the low-noise case of improving estimation and impairing discrimination (see Figure 9).

4 Discussion

We had previously observed superior performance at invariant pattern discrimination for an inference method based on the change over time in the pattern of activity in a recurrent network (Moazzezi & Dayan, 2008). In this letter, we analyzed the method in more detail. 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 performs well only because 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 multiscale receptive fields, it would be interesting to study the dependence of error rate on this factor.

In previous work on estimation, linear analysis was applied only when the network's state started and remained near its attractor (Pouget et al., 1998). We showed that linear analysis can still offer a reasonable guide 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), and indeed confirmed our previous findings that the quality of inference remains good in this case. The workings of the network in these regimes, and also in the face of the sort of dynamical noise that can significantly disturb inference based on nominal convergence of the network to its attractor (Wu & Amari, 2005; Camperi & Wang, 1997; Compte, Brunel, Goldman-Rakic, & Wang, 2000), remain key targets for future work. 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. Here we assume that this memory process is noiseless.

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 observed that one could extract nearly all the information from the change in this. For the task we considered in this letter, 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 letter is to design two line attractor networks, for which both get the 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 our previous paper 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.

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, Kreiman, Poggio, & DiCarlo, 2005; Mazor & 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 & Laurent, 2005). Further, recent data suggest that relatively fine-temporal scale aspects of activity, even in early sensory cortical areas, can significantly influence the course of decision making (Yang, DeWeese, Otazu, & Zador, 2008); this is another facet of change-based readout.

Conversely, the sensitivity of change-based readout to transient activity suggests one route toward a psychophysical test of the idea based on noise images (Dosher & Lu, 1999; Mareschal, Dakin, & Bex, 2006). Consider presenting patterns corrupted by explicit spatiotemporal 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 (Nienberg & Cumming, 2009; Wong, Huk, Shadlen, & Wang, 2007). 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 Zhaoping and Dayan (Li & Dayan, 2001) would be more influenced by explicit noise in later epochs).

Along with Zhaoping et al. (2003), our previous work on the bisection task (Moazzezi & Dayan, 2008) required a network whose weights had been partly sculpted by the needs of a particular task. In empirical studies, the pace of perceptual learning in that task is relatively stately (Fahle, 1994; Fahle, Edelman, & Poggio, 1995). We have shown that it is possible (Moazzezi & Dayan, 2008) to learn a set of weights for the network that enables change-based readout to perform much more competently than attractor-based readout using the backpropagation-through-time algorithm (Rumelhart, Hinton, & Williams, 1986). Many different weight patterns were found to work well at the task, suggesting that it might be possible to model the provenance of change-based competence there too.

Appendix A: The Form of the Translation-Invariant Weight Matrix

The weights have the form of a truncated circular gaussian:

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

The central row of this translation-invariant weight matrix is shown in Figures 2C, 7C, and 8C. We used $\gamma = 0.078$ for weights shown in Figures 2C and 7C and $\gamma = .2$ for weight shown in Figure 8C.

Appendix 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 $f_i(y^*)$, and the signal is at ε^*, with activity $s_i(\varepsilon^*, y^*)$. The total mean activity is

$$h_i(\varepsilon^*, y^*) = f_i(y^*) + s_i(\varepsilon^*, y^*), \tag{B.1}$$

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

$$a_i = h_i(\varepsilon^*, y^*) + \delta a_i. \tag{B.2}$$

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

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

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

where ϕ and K are independent of ε, y.

Maximum likelihood inference determines ε and y as the solutions of

$$\frac{\partial \log P(\mathbf{a}, \varepsilon, y)}{\partial \varepsilon} = \frac{\partial \log P(\mathbf{a}, \varepsilon, y)}{\partial y} = 0. \tag{B.3}$$

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 $\sum_i h_i(\varepsilon, y)$ is independent of ε and y, and so

$$\sum_i h_i^{\varepsilon}(\varepsilon, y) = \sum_i h_i^{y}(\varepsilon, y) = \sum_i h_i^{\varepsilon\varepsilon}(\varepsilon, y) = \sum_i h_i^{\varepsilon y}(\varepsilon, y) = 0 \quad \forall \varepsilon, y. \tag{B.4}$$

If we expand the first part of equation B.3, we get

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

which, to first order about $\varepsilon = 0, \delta y = 0$, writing h_i for $h_i(0, y^*)$ 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 B.4, we get

$$\varepsilon \left(\sum_i \frac{\left(h_i^{\varepsilon}\right)^2}{(h_i)}\right) + \delta y \left(\sum_i \frac{\left(h_i^{\varepsilon}\right)\left(h_i^{y}\right)}{(h_i)}\right) = \sum_i \delta a_i \frac{h_i^{\varepsilon}}{h_i}, \tag{B.5}$$

and similarly

$$\varepsilon \left(\sum_i \frac{\left(h_i^{\varepsilon}\right)\left(h_i^{y}\right)}{(h_i)}\right) + \delta y \left(\sum_i \frac{\left(h_i^{y}\right)^2}{(h_i)}\right) = \sum_i \delta a_i \frac{h_i^{y}}{h_i}. \tag{B.6}$$

If we denote $\alpha = \sum_i \frac{(h_i^\varepsilon)^2}{(h_i)}$, $\beta = \sum_i \frac{h_i^\varepsilon (h_i^y)}{(h_i)}$ and $\gamma = \sum_i \frac{(h_i^y)^2}{(h_i)}$, then solving the simultaneous equations B.5 and B.6 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 the ith component

$$J^i_{dis-opt} = \frac{1}{\alpha\gamma - \beta^2} \frac{\gamma h_i^\varepsilon - \beta h_i^y}{h_i}. \tag{B.7}$$

The same procedure can be followed for the case of scaled gaussian noise, and, given the way we defined it, it leads to the same expression as in equation B.7.

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 B.6, 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 ith component

$$J^i_{est-opt} = \frac{1}{\gamma} \frac{h_i^y}{h_i}. \tag{B.8}$$

Appendix 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 $\mathbf{v}^n = \mathbf{u}^n - \bar{\mathbf{a}}_c$ where $\mathbf{u}^0 = \mathbf{a}$. If we denote the Jacobian matrix by $\mathbf{K}$ then we have (Pouget et al., 1998)

$$\mathbf{K}^n \mathbf{v}^0 = \mathbf{v}^n .$$

This equation implicitly assumes that the Jacobian does not change during the evolution of the neural activity. Note that $\mathbf{u}^0 = \bar{\mathbf{a}}_c + \bar{\mathbf{a}}_s + \mathbf{n}$ and that $\bar{\mathbf{a}}_c$ is a function of y ($\bar{\mathbf{a}}_c = \bar{\mathbf{a}}_c(y)$) and $\bar{\mathbf{a}}_s$ is a function of both ε and y ($\bar{\mathbf{a}}_s(\varepsilon, y)$). 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 $\bar{\mathbf{a}}_c(y + \delta y)$:

$$\mathbf{v}^\infty = \mathbf{K}^\infty \mathbf{v}^0 = \bar{\mathbf{a}}_c(y + \delta y) - \bar{\mathbf{a}}_c(y) = \delta y \frac{\partial \bar{\mathbf{a}}_c(y)}{\partial y}. \tag{C.1}$$

Therefore, we have

$$\begin{aligned} &\mathbf{K}^\infty (\bar{\mathbf{a}}_s + \mathbf{n}) = \delta y \frac{\partial \bar{\mathbf{a}}_c(y)}{\partial y} \\ &\Rightarrow \left[\frac{\partial \bar{\mathbf{a}}_c(y)}{\partial y}\right]^T \mathbf{K}^\infty (\bar{\mathbf{a}}_s + \mathbf{n}) = \delta y \left[\frac{\partial \bar{\mathbf{a}}_c(y)}{\partial y}\right]^T \frac{\partial \bar{\mathbf{a}}_c(y)}{\partial y} \\ &\Rightarrow \delta y = \frac{\left[\frac{\partial \bar{\mathbf{a}}_c(y)}{\partial y}\right]^T \mathbf{K}^\infty (\bar{\mathbf{a}}_s + \mathbf{n})}{\left[\frac{\partial \bar{\mathbf{a}}_c(y)}{\partial y}\right]^T \frac{\partial \bar{\mathbf{a}}_c(y)}{\partial y}}. \end{aligned} \tag{C.2}$$

And therefore the network's filter for estimation is

$$\mathbf{J} = \frac{\left[\frac{\partial \bar{\mathbf{a}}_c(y)}{\partial y}\right]^T \mathbf{K}^\infty}{\left[\frac{\partial \bar{\mathbf{a}}_c(y)}{\partial y}\right]^T \frac{\partial \bar{\mathbf{a}}_c(y)}{\partial y}}. \tag{C.3}$$

Acknowledgments

This work was funded by the Gatsby Charitable Foundation. We are grateful to Jeff Beck for helpful discussions and comments.

References

Brody, C. D., Hernandez, A., Zainos, A., & Romo, R. (2003). Timing and neural encoding of somatosensory parametric working memory in macaque prefrontal cortex. *Cerebral Cortex*, *13*, 1196–1207.

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

Carandini, M., & Heeger, D. (1994). Summation and division by neurons in visual cortex. *Science*, *264*, 1333–1336.

Carandini, M., Heeger, D., & Movshon, J. A. (1997). Linearity and normalization in simple cells of the macaque primary visual cortex. *J. Neuroscience*, *17*, 8621–8644.

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

Deneve, S., Latham, P. E., & Pouget, A. (1999). Reading population codes: A neural implementation of ideal observers. *Nature Neuroscience*, *2*, 740–745.

Deneve, S., Latham, P. E., & Pouget, A. (2001). Efficient computation and cue integration with noisy population codes. *Nature Neuroscience*, *4*, 826–831.

Dosher, B., & Lu, Z. L. (1999). Mechanisms of perceptual learning. *Vision Research*, *39*, 3197–3221.

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

Fahle, M. (1994). Human pattern recognition: Parallel processing and perceptual learning. *Perception*, *23*, 411–427.

Fahle, M., Edelman, S., & Poggio, T. (1995). Fast perceptual learning in hyperacuity. *Vision Research*, *35*, 3003–3013.

Ganguli, S., Bisley, J. W., Roitman, J. D., Shadlen, M. N., Goldberg, M. E., & Miller, K. D. (2008). One-dimensional dynamics of attention and decision making in LIP. *Neuron*, *58*, 15–25.

Heeger, D. J., Simoncelli, E. P., & Movshon, J. A. (1996). Computational models of cortical visual processing. *Proc. Natl. Acad. Sci. USA*, *93*, 623–627.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA*, *79*, 2554–2558.

Hung, C., Kreiman, G., Poggio, T., & DiCarlo, J. (2005). Fast read-out of object information in inferior temporal cortex. *Science*, *310*, 863–866.

Li, Z., & Dayan, P. (2001). Position variance, recurrence and perceptual learning. In T. K. Leen, T. G. Dietterich, & V. Tresp (Eds.), *Advances in neural information processing systems*, *13* (pp. 31–37). Cambridge, MA: MIT Press.

Mareschal, I., Dakin, S. C., & Bex, P. J. (2006). Dynamic properties of orientation discrimination assessed by using classification images. *Proc. Natl. Acad. Sci. USA*, *103*, 5131–5136.

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

Moazzezi, R., & Dayan, P. (2008). Change-based inference for invariant discrimination. *Network: Computation in Neural Systems*, *19*, 236–252.

Nienberg, H., & Cumming, B. (2009). Decision-related activity in sensory neurons reflects more than a neuron's causal effect. *Nature*, *459*, 89–92.

Pouget, A., Zhang, K., Deneve, S., & Latham, P. E. (1998). Statistically efficient estimation using population code. *Neural Computation*, *10*, 373–401.

Reinagel, P. (2001). How do visual neurons respond in the real world? *Current Opinion in Neurobiology*, *11*, 437–442.

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

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by backpropagating errors. *Nature*, *323*, 533–536.

Seung, H. S. (1996). How the brain keeps the eyes still. *Proc. Natl. Acad. Sci. USA*, *93*, 13339–13344.

Vinje, W. E., & Gallant, J. L. (2000). Sparse coding and decorrelation in primary visual cortex during natural vision. *Science*, *287*, 1273–1276.

Wang, X-J. (2001). Synaptic reverberation underlying mnemonic persistent activity. *Trends in Neurosciences*, *24*, 455–463.

Wong, K-F., Huk, A. C., Shadlen, M. N., & Wang, X-J. (2007). Neural circuit dynamics underlying accumulation of time-varying evidence during perceptual decision making. *Front. Comput. Neurosci.*, *1*, 1–11.

Wu, S., & Amari, S. (2005). Computing with continuous attractors: Stability and online aspects. *Neural Computation*, *17*, 2215–2239.

Wu, S., Nakahara, H., & Amari, S. (2001). Population coding with correlation and an unfaithful model. *Neural Computation*, *13*, 775–797.

Yang, Y., DeWeese, M. R., Otazu, G. H., & Zador, A. M. (2008). Millisecond-scale differences in neural activity in auditory cortex can drive decisions. *Nature Neuroscience*, *11*, 1262–1263.

Zhang, K. (1996). Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: A theory. *J. Neuroscience*, *16*, 2112–2126.

Zhaoping, L., Herzog, M. H., & Dayan, P. (2003). Quadratic ideal observation and recurrent preprocessing in perceptual learning. *Network: Computation in Neural Systems*, *14*, 233–247.

Received March 30, 2009; accepted May 24, 2010.