Communicated by Terrence Sanger

Statistically Efficient Estimation Using Population Coding

Alexandre Pouget
Georgetown Institute for Computational and Cognitive Sciences, Georgetown University, Washington, DC 20007-2197, U.S.A.

Kechen Zhang
Computational Neurobiology Laboratory, Salk Institute, La Jolla, CA 92037, U.S.A.

Sophie Deneve
Georgetown Institute for Computational and Cognitive Sciences, Georgetown University, Washington, DC 20007-2197, U.S.A.

Peter E. Latham
Department of Neurobiology, University of California at Los Angeles, Los Angeles, CA 90095-1763, U.S.A.

Coarse codes are widely used throughout the brain to encode sensory and motor variables. Methods designed to interpret these codes, such as population vector analysis, are either inefficient (the variance of the estimate is much larger than the smallest possible variance) or biologically implausible, like maximum likelihood. Moreover, these methods attempt to compute a scalar or vector estimate of the encoded variable. Neurons are faced with a similar estimation problem. They must read out the responses of the presynaptic neurons, but, by contrast, they typically encode the variable with a further population code rather than as a scalar. We show how a nonlinear recurrent network can be used to perform estimation in a near-optimal way while keeping the estimate in a coarse code format. This work suggests that lateral connections in the cortex may be involved in cleaning up uncorrelated noise among neurons representing similar variables.

1 Introduction

Many sensory and motor variables in the brain are encoded with coarse codes, that is, through the activity of large populations of neurons with broad tuning to the variables. For instance, direction of visual motion is believed to be encoded in the medial temporal (MT) visual area by the responses of a large number of cells with bell-shaped tuning to direction, as illustrated in Figure 1A (Maunsell & Van Essen, 1983).

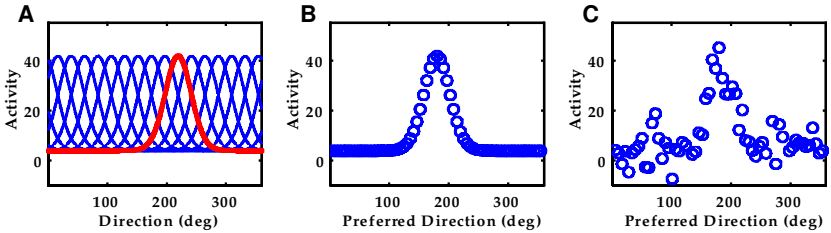

Figure 1: (A) Idealized tuning curves for 16 direction-tuned neurons. (B) Noiseless pattern of activity (o) from 64 simulated neurons with tuning curves like the ones shown in A, when presented with a direction of 180°. (C) Same as in B but in the presence of gaussian noise.

In response to an object moving along a particular direction, the pattern of activity across such a population would look like a noisy hill of activity (see Figure 1C). On the basis of this activity vector, $\mathbf{A}$, the best that can be done is to recover the conditional probability distribution of the direction of motion, θ, given the activity, $p(\theta \mid \mathbf{A})$ (Anderson, 1994; Zemel, Dayan, & Pouget, 1998). A slightly less ambitious goal is to come up with a good guess, or estimate, $\hat{\theta}$, of the direction, θ, given the activity. Because of the stochastic nature of the noise, the estimator is a random variable; that is, for the same image, $\hat{\theta}$ will vary from trial to trial. A good estimator should be unbiased; the conditional mean of the estimator, $E[\hat{\theta} \mid \theta]$, should be equal to the true direction, θ. Furthermore, this unbiased estimator should have the smallest possible conditional variance, $E[(\hat{\theta} - \theta)^2 \mid \theta]$, because the variance determines how well two similar directions can be discriminated using this estimator (Green & Swets, 1966; Paradiso, 1988). This conditional variance is bounded below by the Cramér-Rao bound, which depends on the noise level and the tuning curves of the units (Paradiso, 1988; Papoulis, 1991). Typically, computationally simple estimators, such as the optimum linear estimator (OLE) (Baldi & Heiligenberg, 1988; Pouget, Fisher, & Sejnowski, 1993), are not efficient, in the statistical sense that their variances are several times the bound. By contrast, Bayesian or maximum likelihood (ML) estimators (which are equivalent for the case under consideration in this article) can reach this bound but require more complex calculations (Paradiso, 1988; Seung & Sompolinsky, 1993; Salinas & Abbott, 1994).

These decoding techniques are valuable for a neurophysiologist interested in reading out the population code, but they are not directly relevant for understanding how neural circuits perform estimation. In particular, they all provide the estimate in a format that is incompatible with what we know of sensory representations in the cortex. For example, cells in V4 are estimating orientation from the noisy responses of orientation tuned V1 cells, but, unlike ML or OLE, which provide a scalar estimate, V4 neurons

retain orientation in a coarse code format, as demonstrated by the fact that V4 cells are just as broadly tuned to orientation as V1 neurons (Desimone, Schein, Moron, & Ungerleider, 1985). Such coarse codes have several computational advantages over scalar representations, and it is important to understand how they are maintained throughout the cortex (Hinton, 1992).

Therefore, it seems that a theory of estimation in biological networks should have two critical characteristics: (1) it should preserve the estimate in a coarse code, and (2) it should be efficient, that is, the variance should be close to the Cramér-Rao bound. This article describes a model that satisfies these two requirements. Our model uses lateral connections in a nonlinear recurrent network of direction-tuned neurons to come up with an ML estimate of direction in a coarse code format. We also show how linear recurrent networks are related to the population vector estimator used by Georgopoulos, Kalaska, Caminiti, & Massey (1982), and we provide a performance comparison between various network architectures and classical estimation methods such as OLE and ML.

In this article, we first describe how we generated neuronal patterns of activity used in the simulations. Then we review four estimators that have been previously used in the literature to decode such patterns. Next, we consider linear and nonlinear networks with lateral connections, and we show how they can be used as estimators. We report the results of simulations in which we compared the performance of a nonlinear network to the classical methods. Finally, we show analytically the relation between the nonlinear network and maximum likelihood.

2 Model of Neuronal Responses

The simulations involve estimating the value of the direction of a moving bar based on the activity, $\mathbf{A} = \{a_i\}$, of 64 input units with bell-shaped tuning to direction corrupted by noise. The tuning function of unit i, $f_i(\theta)$, which is the same as the conditional mean response, $E[a_i|\theta]$, was given by:

$$f_i(\theta) = \alpha \exp(\beta(\cos(\theta - \theta_i) - 1)) + \gamma. \tag{2.1}$$

This function is known as the *circular normal distribution*. Its profile is very similar to a gaussian, but it is periodic. α corresponds to the mean peak response, β to the width of the tuning curve, and γ to the mean spontaneous activity of each unit. Cortical neurons commonly show spontaneous activity, although the amplitude of this activity varies from one cortical area to the next. The peaks of the tuning curves, θ_i, were evenly spread over the interval $[0^\circ, 360^\circ]$.

The activity a_i depended on the noise distribution. We used two types of noise, normally distributed with fixed variance, σ_n^2:

$$P(a_i = a|\theta) = \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left(-\frac{(a - f_i(\theta))^2}{2\sigma_n^2}\right),$$

or Poisson distributed:

$$P(a_i = k|\theta) = \frac{f_i(\theta)^k e^{-f_i(\theta)}}{k!}.$$

Figure 1C shows a typical pattern of activity with gaussian noise ($\sigma_n^2 = 7$). Note that the noise is in the activity level, a_i, not in θ. On any given trial, θ is assumed to have a given value; i.e., the probability distribution over θ, $P(\theta)$, is assumed to be a Dirac function.

3 Classical Decoding Methods

We now review four different methods for decoding patterns of neural activity: maximum likelihood (ML), optimum linear estimator (OLE), center of mass (COM), and complex estimator (COMP). We indicate in each case how we computed the variance of these estimators. Simulation results and comparison with recurrent network architecture are provided in the following sections.

3.1 Maximum Likelihood (ML).

The ML estimate is defined as:

$$\hat{\theta}_{\mathrm{ML}} = \arg\max_{\theta} P(\mathbf{A}|\theta).$$

With independent noise between units, finding the ML estimate reduces to curve fitting, or template matching (Paradiso, 1988; Lehky & Sejnowski, 1990; Wilson & McNaughton, 1993). One needs to find the noise-free hill that minimizes distance from the data where the distance metric is determined by the distribution of the noise (if the noise is gaussian, the appropriate distance is the Mahalanobis norm; Duda & Hart, 1973). This step involves a nonlinear regression, which is typically performed by moving the position of the hill until the distance from the data is minimized (see Figure 2B).

The position of the peak of the final hill corresponds to the ML estimate. With a large number of units, this estimate is unbiased, and its variance is equal to the Cramér-Rao bound (Paradiso, 1988; Papoulis, 1991; Seung & Sompolinsky, 1993):

$$E\left[(\hat{\theta}_{\mathrm{ML}} - \theta)^2\right] = \frac{1}{I},$$

where

$$I = E\left[-\frac{\partial^2}{\partial\theta^2}\log P(\mathbf{A}|\theta)\right]. \tag{3.1}$$

If we assume independent noise across units, then:

$$I = \sum_{i=1}^{N} E\left[-\frac{\partial^2}{\partial\theta^2}\log P(a_i|\theta)\right].$$

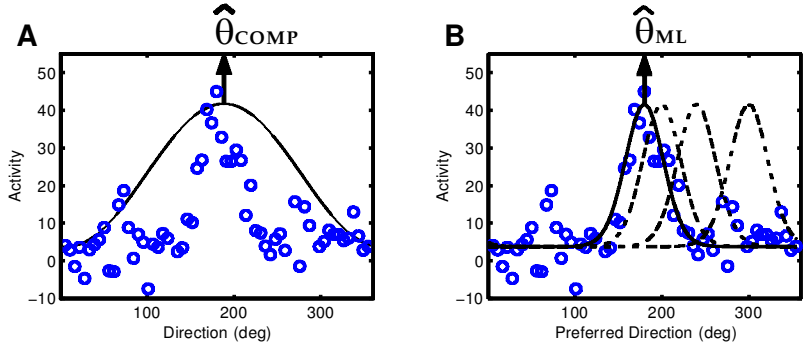

Figure 2: (A) The complex estimator uses the phase of the first Fourier component of the input pattern (solid line) as an estimate of direction. It is equivalent to fitting a cosine function to the input. (B) The ML estimate is found by moving an "expected" hill of activity (dotted line) until the squared distance with the data is minimized (solid line).

For a normally distributed noise with fixed variance, σ_n^2:

$$I = \frac{\sum_{i=1}^{N} f_i'(\theta)^2}{\sigma_n^2}, \tag{3.2}$$

and for a Poisson distributed noise (Seung & Sompolinsky, 1993):

$$I = \sum_{i=1}^{N} \frac{f_i'(\theta)^2}{f_i(\theta)}. \tag{3.3}$$

3.2 Optimum Linear Estimator (OLE). The simplest possible estimator is an estimator that is linear in the activities of the neurons, $\mathbf{A}$ (Pouget et al., 1993):

$$\hat{\theta}_{\mathrm{OLE}} = \mathbf{w}^T \mathbf{A}.$$

A common choice for $\mathbf{w}$ is to take the weight vector minimizing the mean square distance between the estimate, $\hat{\theta}_{\mathrm{OLE}}$, and the true direction, θ:

$$\mathbf{w} = \arg\min_{\mathbf{w}} E\left[(\theta - \hat{\theta}_{\mathrm{OLE}})^2\right].$$

One can think of the linear estimator as being the response of a single output unit with weights $\mathbf{w}$. Note that this estimator is poorly adapted to the estimation of a periodic variable such as direction. In our simulations, we worked around 180 degrees, staying away from the discontinuity at 0 and 360 degrees.

OLE is known to be unbiased for a large number of units, that is, $E[\hat{\theta}_{\mathrm{OLE}} \mid \theta] = \theta$ (Baldi & Heiligenberg, 1988). In this case, its variance given θ is:

$$E\left[\left(\hat{\theta}_{\mathrm{OLE}} - E\theta\right)^2 \mid \theta\right] = \sum_{i=1}^{L} w_i^2 \sigma_i^2, \tag{3.4}$$

where $\sigma_i^2 = \sigma_n^2$ for the normally distributed noise with fixed variance σ_n^2, and $\sigma_i^2 = f_i(\theta)$ for the Poisson distributed noise.

3.3 Center of Mass (COM). This estimator is a one-dimensional version of the population vector used by Georgopoulos et al. (1982) (see also Zohary, 1992; Snippe, 1996). It is defined as:

$$\hat{\theta}_{\mathrm{COM}} = \frac{\sum_{i=1}^{N} \theta_i (a_i - \gamma)}{\sum_{i=1}^{N} (a_i - \gamma)}.$$

The mean spontaneous activity, γ (see equation 2.1), is subtracted from the activities a_i to prevent systematic bias. Like OLE, COM handles poorly the discontinuity between 0 and 360 degrees.

We obtained an approximation of the variance of the COM estimate using computer simulations. These estimates, computed for 201 values of direction, systematically varied from 170 to 190 degrees by increments of 0.1 degree. For each direction, the variance and mean of the estimate were calculated according to:

$$E\left[\hat{\theta}_{\mathrm{COM}} \mid \theta\right] = \frac{1}{L} \sum_{l=1}^{L} \hat{\theta}_{\mathrm{COM}}^{l}$$

$$E\left[\left(\hat{\theta}_{\mathrm{COM}} - E\left[\hat{\theta}_{\mathrm{COM}} \mid \theta\right]\right)^2 \mid \theta\right] = \frac{1}{L-1} \sum_{l=1}^{L} \left(\hat{\theta}_{\mathrm{COM}}^{l} - E\left[\hat{\theta}_{\mathrm{COM}} \mid \theta\right]\right)^2.$$

We used $L = 1000$ trials in all simulations.

3.4 Complex Estimator (COMP). The complex estimator (also known as population vector; Georgopoulos et al., 1982) is defined as the phase of the first Fourier component of the input pattern (Seung & Sompolinsky, 1993):

$$\hat{\theta}_{\mathrm{COMP}} = \mathrm{phase}(z),$$

where

$$z = \sum_{j=1}^{N} a_j e^{i\theta_j}.$$

This estimator is often said to be linear (see Seung & Sompolinsky, 1993; Salinas & Abbott, 1994), but it is important to realize that only z, and not $\hat{\theta}_{\mathrm{COMP}}$, is linear in $\mathbf{A}$. Recovering the phase of a complex number is a nonlinear operation.

This estimator is equivalent to an ML estimator *only* under the assumption that the data were generated according to a cosine tuning function with period 2π corrupted by gaussian noise of fixed variance (see Figure 2A). This estimator is guaranteed to be suboptimal if the noise is nongaussian or if the data are generated by *any* other function and, in particular, the one used in our simulations (see equation 2.1).

We obtained an approximation of the variance of the estimator using computer simulations as described in the previous section.

4 Recurrent Networks

All the methods described so far recover a scalar estimate of direction. We now consider network architectures in which the estimate is kept in a coarse code format. These networks have an input and output layer of 64 units, fully connected from the input to the output layer (feedforward connections) and within the output layer (lateral connections), using periodic boundaries and identical weight matrices for the feedforward and lateral connections (see Figure 3A). We use the notation $\mathbf{A} = \{a_i\}$ for the activity of the input units as specified in equation 2.1 and $\mathbf{O}_t = \{o_{i,t}\}$ for the activities of the output units at time t.

We consider only the case of a transient input; at time zero, we set the activity of the input units to $\{a_i\}$, pass it through the feedforward connections, and then remove the input and let the activities of the output units evolve according to the dynamical equation for this layer.

As we will show, an appropriate choice of the weights and the activation function can ensure that the activity in the output layer, which forms a recurrent network, will evolve toward a stable state, corresponding to a hill-shaped pattern of activity (see Figure 3B, which shows the activity over time for the nonlinear network described below; Zhang, 1996).

We can use the final position of the hill across the neuronal array after relaxation as a coarse code estimate of the direction, θ. In the next two sections, we explore the properties of this estimator for linear and nonlinear activation functions.

4.1 Linear Network. We first consider a network with linear activation functions in the output layer and whose dynamics is governed by the following difference equation:

$$\mathbf{O}_t = ((1-\lambda)\, I + \lambda W)\, \mathbf{O}_{t-1}, \tag{4.1}$$

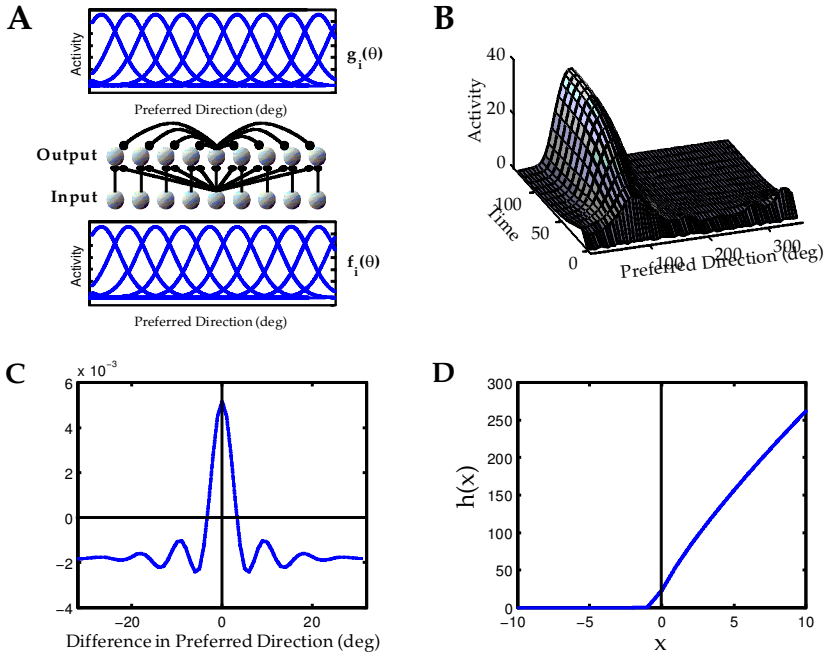

Figure 3: (A) Two-layer network for estimation using coarse code. The first layer generates the noisy activity pattern $\{a_i\}$ according to the tuning function $\{f_i(\theta)\}$. The output layer is a recurrent network that generates a hill of activity corresponding to the tuning function $\{g_i(\theta)\}$. (B) Activity over time in the output layer with a nonlinear activation function in response to an initial small, random pattern of activity. The activity of the units is plotted as a function of their preferred direction of motion. (C) Pattern of weights in the nonlinear recurrent network as a function of the difference in preferred direction between units. (D) Activation function, $h(x)$, of the nonlinear recurrent network.

where λ is a number between 0 and 1, I is the identity matrix, and W is the matrix for the lateral connections. The activity at time 0, $\mathbf{O}_0$, is initialized to $W\mathbf{A}$, where $\mathbf{A}$ is an input pattern (like the one shown in Figure 1C) and W is the feedforward weight matrix, which is set to be equal to the lateral weight matrix (hence, the same notation).

The dynamics of such networks is well understood (Hirsch & Smale, 1974). If each unit receives the same weight vector—if all the rows of W, which we will denote $\mathbf{w}$, are translated versions of one another—a Fourier transform of equation 4.1 (not in time but over the vectors $\mathbf{O}_t$, $\mathbf{O}_{t-1}$, and $\mathbf{w}$)

leads to:

$$\begin{aligned}\mathcal{O}_t &= ((1-\lambda)I + \lambda\mathcal{W})\,\mathcal{O}_{t-1} \\ &= \mathcal{Q}\mathcal{O}_{t-1},\end{aligned}$$

where $\mathcal{O}_t$ and $\mathcal{O}_{t-1}$ are the Fourier transforms of $\mathbf{O}_t$ and $\mathbf{O}_{t-1}$, and $\mathcal{W}$ is a diagonal matrix with the Fourier coefficients of $\mathbf{w}$ along the diagonal. Since $\mathcal{O}_0 = \mathcal{W}\mathcal{A}$, we obtain:

$$\mathcal{O}_t = \mathcal{Q}^t\mathcal{W}\mathcal{A}.$$

Consequently, the network dynamics amplifies or suppresses independently the Fourier component of the initial input pattern, $\mathbf{A}$, by a factor equal to the corresponding component of the Fourier transform of $\mathcal{Q}$. For example, if the first diagonal term of $\mathcal{Q}$ is more than one (resp., less than one), the first Fourier component of the initial pattern of activity will be amplified (resp., suppressed).

Thus, we can choose W such that the network amplifies selectively the first Fourier component of the data while suppressing the others. The network would be unstable, but if we stop after a large, yet fixed, number of iterations, the activity pattern would look like a cosine function of direction with a phase corresponding to the phase of the first Fourier components of the data. If we now use the position of the peak of the hill, which is the same as the phase of the cosine, as an estimate of direction, we end up with the same value as the one provided by the COMP methods. A network for orientation selectivity proposed by Ben-Yishai, Bar-Or, and Sompolinsky (1995) is closely related to this linear network. Their network is actually nonlinear, but the nonlinearity simply acts as a gain control, which prevents activity from growing to infinity.

Although such networks keep the estimate in a coarse code format, they suffer from two problems: it is unclear how they could be extended to nonperiodic variables, such as disparity, and they are suboptimal since they are equivalent to the COMP estimator.

4.2 Nonlinear Network. We consider next a network with nonlinear activation functions in which the dynamics of the output units is governed by the following difference equations:

$$o_{i,t} = h(u_{i,t}) = a\left(\log(1 + e^{b+cu_{i,t}})\right)^d \tag{4.2}$$

$$u_{i,t} = (1-\lambda)u_{i,t-1} + \lambda\sum_{j=1}^{N} w_{ij}o_{j,t-1}. \tag{4.3}$$

Using vector notations, we rewrite these equations as:

$$\mathbf{O}_t = h(\mathbf{U}_t) = a\left(\log(1 + e^{b+c\mathbf{U}_t})\right)^d \tag{4.4}$$

$$\mathbf{U}_t = (1 - \lambda)\mathbf{U}_{t-1} + \lambda W \mathbf{O}_{t-1}. \tag{4.5}$$

As shown in Zhang (1996), the weights, W, can be set in such a way that a hill of activity of profile, $g(\theta)$, centered at any location on the network is a stable state (see Figure 3B). These kinds of networks are known as line attractor networks, because the set of all hills defines a one-dimensional continuous stable manifold in activity space. The rows of the weight matrix must be a translated version of the same vector, $\mathbf{w}$, which is found by solving:

$$g(\theta) = h(w * g(\theta)), \tag{4.6}$$

where $g(\theta)$ is the desired bell-shaped profile, $*$ is the convolution, and $h(\cdot)$ is the activation function (this equation involves continuous functions but it can be easily discretized to deal with a finite number of units).[1] There is no analytical solution to this equation, but a close approximation can be obtained for a wide variety of bell-shaped profiles of activity and monotonic activation functions (Zhang, 1996).

Thus, the shape of the stable hill is fully specified by the weights and activation function. By contrast, the final position of the hill on the neuronal array depends on only the initial input (Zhang, 1996). Therefore, like ML, the network fits an "expected" function—the stable hill—through the noisy input pattern, $\mathbf{A}$. We will use the notation $g(\theta)$ to refer to this function and $g_i(\theta)$ for the corresponding tuning curves of the output units (see Figure 2A).

For reasons that will become clear, we selected the lateral weights, W, to minimize the $\mathcal{L}^2$ distance between $g(\theta)$—the function corresponding to the stable hill—and the function $f(\theta)$ (see equation 2.1) used to generate the activity patterns, $\mathbf{A}$ (see Zhang, 1996, for details about this procedure based on regularization theory). The resulting weights are locally excitatory with long-range inhibition, a common pattern of connectivity in models of cortical circuitry (see Figure 3C).

The resulting network can be used as an estimator by first initializing the input layer to a vector $\mathbf{A}$, passing the activity through the feedforward connections (which amount to setting $\mathbf{U}_0$ to $W\mathbf{A}$) and iterating equations 4.4 and 4.5 until a stable hill of activity is obtained. The stable hill in the output layer can be treated as a population code for the estimated direction, $\hat{\theta}_{\mathrm{RN}}$ (RN, recurrent network), and a scalar value can be obtained by computing the peak position. We computed the position of the peak using a COMP

[1] Strictly speaking, the weights that solve equation 4.6 in the discrete case lead to a network with N stable fixed points along the one-dimensional manifold, interspersed with N unstable fixed points, where N is the number of units. Therefore, the resulting network is not truly a line attractor network; the eigenvalue, λ, of the Jacobian along the manifold near the attracting fixed point is slightly less than 1. It can be shown, however, that $1 > \lambda > 1 - k/N^2$, where k is a constant independent of N. Therefore, for large N, the dynamics of convergence along the manifold is so slow that it can be ignored for all practical purposes, which is what we do in the rest of the article.

operator applied to the stable pattern of activity, $\mathbf{O}_\infty$, although any unbiased estimator would have worked. Note that this step would not be required in the brain. We have added it only to allow comparison with the other estimators.

Estimates of the bias and variance of the direction estimates were obtained with the same method as that used for the COMP estimator. The activation function, $h(\cdot)$, used in equation 4.2, looks like a linear rectified function (see Figure 3D). It is close to zero for negative x and grows roughly linearly past a threshold. The parameters α, β, and γ in equation 2.1 were set, respectively, to 38, 7, and 3.8 and the parameters a, b, c, and d in equation 4.2 were set to, respectively, 6.3, 5, 10, and 0.8. All of these choices were motivated by the fact that the same parameters and function were used by Zhang (1996) in a previous study. Our results do not depend critically on these particular choice; variations in these parameters do not affect our results.

The standard deviation of the gaussian noise was set to $\sigma_n = 5.8$, which corresponds to a signal-to-noise ratio of 6 for the most active units. By comparison, the signal-to-noise ratio of the most active units when using Poisson noise was 6.5.

5 Simulation Results

Since the preferred directions of two consecutive units in the network are more than 5 degrees apart, we first wondered whether recurrent network (RN) estimates would exhibit a systematic bias—a difference between the mean estimate and the true direction—in particular for directions between the peaks of two consecutive units. Our simulations showed no significant bias for any of the directions tested (see Figure 4). This entails that, with 64 units only, the stable hill can settle in any position, in particular between the peaks of the tuning curves of two successive units.

Next, we compared the standard deviations of the estimates for four estimators—OLE, COM, COMP and ML—to the nonlinear RN. We did not simulate the linear network since it is equivalent to the COMP methods. The standard deviations for the ML and OLE were obtained using equations 3.2, 3.3, and 3.4.

The RN method was found to outperform the OLE, COM, and COMP estimators in both cases and to match the Cramér-Rao bound for gaussian noise (see Figure 5). For noise with Poisson distribution, the standard deviation for RN was only 6.5% above the bound (see Figure 5B). To confirm that ML and RN are similar, we looked at the coefficient of correlation between the two estimates. We obtained a value of 0.98, indicating that the two estimates are almost identical on individual trials.

We also estimated numerically $-\partial\hat{\theta}_{\mathrm{RN}}/\partial a_i|_{\theta=170^\circ}$, the partial derivative of the RN estimate with respect to the initial activity of each of 64 units for a direction of 170 degrees. This derivative in the case of ML matches

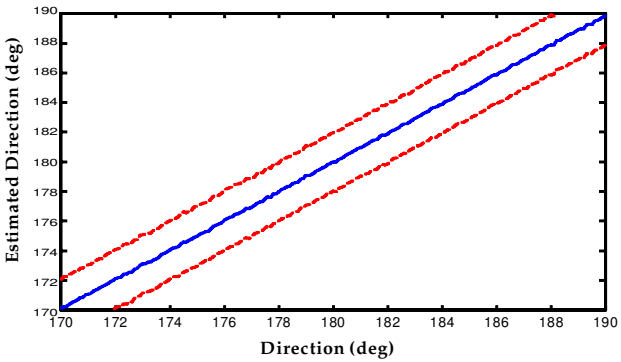

Figure 4: The solid line shows the mean estimated direction as a function of the true direction for normally distributed noise of fixed variance. The estimator is unbiased, that is, the mean estimate is equal to the true direction. The upper and lower dotted lines are one standard deviation away from the mean.

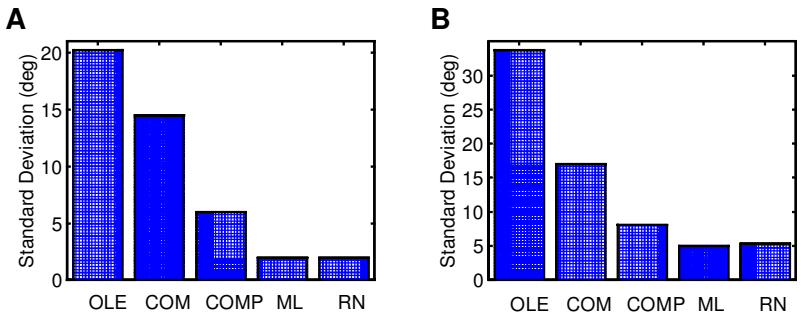

Figure 5: Histogram of the standard deviations of the estimate for all five methods (OLE, optimal linear estimator; COM, center of mass; COMP, complex estimator; ML, maximum likelihood; RN, recurrent network). (A) Noise with normal distribution. (B) Noise with Poisson distribution. In both cases, the value for ML is the Cramér-Rao bound. The RN method reaches this bound for gaussian noise and performs slightly worse for Poisson noise.

closely the derivative of the cell tuning curve, $f_i'(\theta)$. In other words, in ML, units contribute to the estimate according to the amplitude of the derivative of the tuning curve. As shown in Figure 6A, the same is true for RN; $-\partial\hat{\theta}_{\mathrm{RN}}/\partial a_i|_{\theta=170^\circ}$ matches closely the derivative of the units' tuning curves. In contrast, the same derivatives for the COMP estimate (dotted line) or the COM estimate (dash-dotted line) do not match the profile of $f_i'(\theta)$. In particular, units with preferred direction far away from 170 degrees—units whose

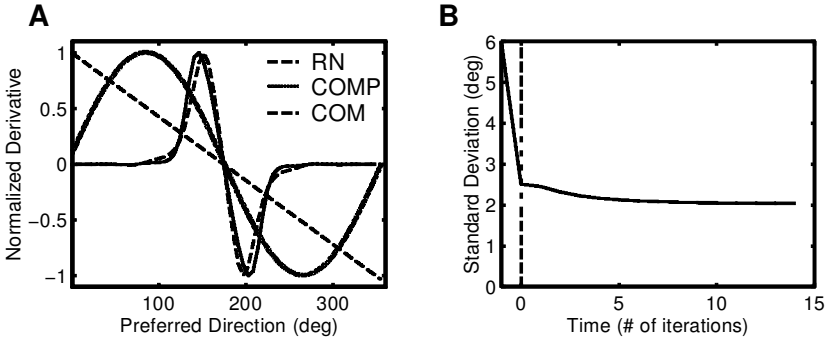

Figure 6: (A) Comparison of $f'(\theta)$ (solid line) and $-\partial\hat{\theta}/\partial a_i|_{\theta=170^\circ}$ for RN, COMP, and COM. All functions have been normalized to one. (B) Standard deviation as a function of time, or number of iterations of the recurrent network. The point at $t = -1$ is the COMP estimator applied to the input activity, $\mathbf{A}$, whereas the point at $t = 0$ corresponds to COMP applied to $W\mathbf{A}$.

activity is just noise—end up contributing to the final estimate, hindering the performance of the estimator.

Reaching a stable state can take many iterations, which could make the RN method too slow for any practical purpose. Consequently, we looked at the standard deviation of the RN as a function of time—that is, the number of iterations. We found that the convergence to ML is very fast. In fact, initializing $\mathbf{U}_0$ to $W\mathbf{A}$ and $\mathbf{O}_0$ to $h(\mathbf{U}_0)$ is sufficient to obtain a standard deviation very close to the bound, and 5 to 10 iterations leads to the asymptotic values (see Figure 6B). The initialization is mathematically equivalent to one network iteration with the integration constant, λ, set to one (see equation 4.5). We can therefore conclude that there is no need to wait for a perfectly stable pattern of activity to obtain minimum standard deviation and that one network iteration is sufficient to obtain performance close to ML.

So far, the input units (which determine the input patterns, $\mathbf{A}$) and the network units had the same tuning curves: $f(\theta) = g(\theta)$. Next, we explored the effect of varying the amplitude and the width of the input tuning curves while keeping the output tuning curves constant. A comparable situation for ML would be to fit the wrong tuning curve through the data. With ML, an error in the assumed amplitude of the bump would not affect performance (the minimum of the nonlinear regression step is unaffected; see Figure 2B), whereas a mismatch between the actual and assumed width results in suboptimal performance.

Our simulations revealed that both parameters affect the performance of the network estimate (see Figures 7A and 7B). Large differences in amplitude or width lead to a standard deviation much larger than the Cramér-Rao

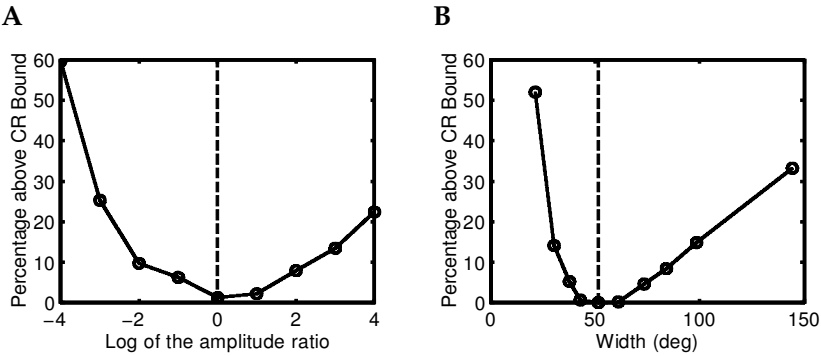

Figure 7: Standard deviation of the RN estimator in terms of percentage above the Cramér-Rao bound, as a function of the amplitude (A) and width (B) of the input bump. (A) The data are plotted as a function of the logarithm in base 2 of the ratio of input-output amplitudes. Changing the gain of the input by a factor of 2 or less affects the performance of the network only moderately. (B) The width of the tuning curve is computed at half the peak value. The network sharpens the input tuning curves for width values above 50 degrees and widens for smaller values. The Cramér-Rao bound is reached only when the widths of the input and output tuning curves differ by less than 10 degrees.

bound. The curves, however, are both quadratic around the minimum, indicating that the network is fairly robust with respect to these kinds of errors. In particular, changing the amplitude by a factor of two has a minimal impact on the standard deviation (see Figure 7A). Nevertheless, unlike ML, performance eventually decreases with larger-amplitude changes.

Finally, Figure 8A shows the covariance matrix of the input unit activities, a_i, when presented repetitively with a direction of 170 degrees. Since the noise was chosen to be independent across units, only the diagonal terms of the covariance matrix—the variances of the individual units—differ from zero. Interestingly, the covariance of the network units after a stable pattern has been reached has a different structure (see Figure 8B). Units with similar direction preferences around 150 degrees are positively correlated while being negatively correlated with units whose direction preference is around 190 degrees, and vice versa. Furthermore, units with preferred directions away from the test direction (outside the interval $170^\circ \pm 30^\circ$) have a variance and covariance close to zero.

Interestingly, these correlations do not reflect the similarity of the tuning curves for units with similar preferred directions. The similarity in tuning curves introduces similarities in the mean responses. By contrast, the covariance matrices plotted in Figure 8B show correlations in the fluctuations about these mean responses. Such correlations are often considered

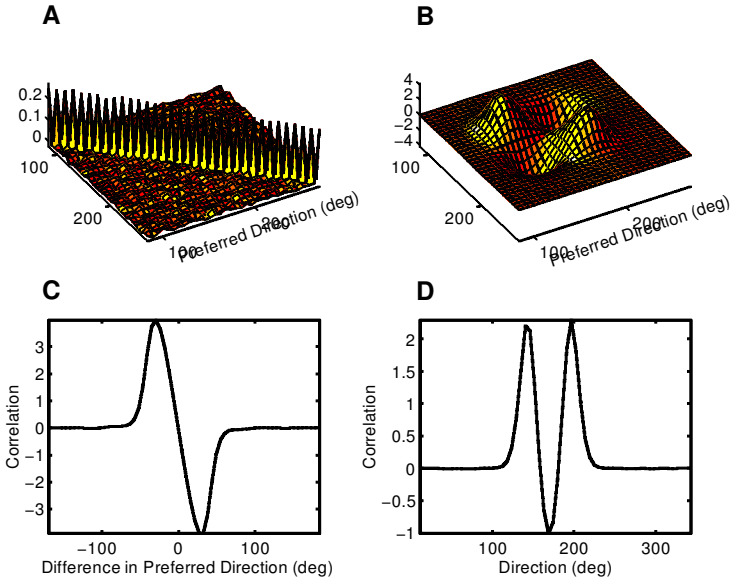

Figure 8: Covariance matrices of the input units (A) and network units (B) for repetitive presentations of a direction of 170 degrees. Only the central part of the covariance matrix is shown (units with preferred directions between 84 and 270 degrees). Whereas the input units are independent, the output units are correlated due to the lateral connections. (C) Correlation of unit with preferred direction 135 degrees with all the other units as a function of the difference in preferred direction. The curve has the same profile as the derivative of the tuning curve. (D) Correlation between two units (preferred directions 158 and 182 degrees) as a function of stimulus direction.

damaging because they reduce the signal-to-noise ratio (Zohary, Shadlen, & Newsome, 1994). We see here, however, that they could be the unavoidable consequence of pooling the unit activities through the lateral connections to clean up the noise in an optimum way.

At first, one might think that this pattern of correlation reflects the weights of the lateral connections; for example, units with similar preferred directions are positively correlated because they are positively interconnected. It turns out, however, that these correlations are the result of fitting a hill to the data. Indeed, the activity of a unit at the end of relaxation, $o_{i,\infty}$, is dependent on the activities of all the other units in a way that is specified by the profile of the stable hill. Consequently, the correlation between pairs of units is determined by the product of their tuning curve derivatives, evaluated at the current direction (170 degrees in Figure 8B). Hence, when plotting the correlation of the units with preferred direction 135 degrees with all the

other units, the resulting curve has the same profile as the derivative of the tuning curve (compare Figure 8C and $f'(\theta)$ shown in Figure 6A). This property is not specific to the RN method but would also apply to any method involving fitting a hill, such as ML or COMP.

The magnitude and sign of these correlations are therefore dependent on the stimulus direction. This is illustrated in Figure 8D, which shows the correlation between units with preferred direction of 158 degrees and 182 degrees as a function of the stimulus direction. Notice that even though the weight of the connection between these two units is negative (-0.08), the correlation can be positive or negative depending on the stimulus direction.

Whether such patterns of correlations exist in the cortex is unknown. Correlations between cells have been reported in area MT (Zohary et al., 1994), but there has been no attempt to relate these correlations to the tuning curve derivatives. It is unlikely, however, that real neurons will exhibit reversal in the correlation sign as large as the one illustrated in Figure 8. Relaxation in our network is a deterministic process, whereas, by contrast, additional noise would be introduced at each iteration if we were to model our units as Poisson process, a more realistic assumption (Shadlen & Newsome, 1994). This extra noise is likely to lead to additional correlations whose form remains to be determined. Nevertheless, we would expect the correlation to change with the stimulus direction in a way consistent with what is illustrated in Figure 8D.

6 Analysis

Our simulations demonstrate that the recurrent network can provide a coarse code estimate of direction that is almost as efficient as the ML estimate. We now prove analytically that the network estimate is indeed close to the ML estimate for small gaussian noise; that is, it is unbiased and efficient. The proof relies on a linearization of the network dynamics around the stable manifold.

6.1 Notation. We start by rewriting the dynamics of the network as follows:

$$\mathbf{O}_t = h(\mathbf{U}_t) = 6.3\left(\log(1 + e^{5+10\mathbf{U}_t})\right)^{0.8} \tag{6.1}$$

$$\mathbf{U}_t = (1-\lambda)\mathbf{U}_{t-1} + \lambda W\mathbf{O}_{t-1} \tag{6.2}$$

$$= (1-\lambda)\mathbf{U}_{t-1} + \lambda W h(\mathbf{U}_{t-1}) \tag{6.3}$$

$$= e(\mathbf{U}_{t-1}). \tag{6.4}$$

As we have done so far, we will use the notation $f_i(\theta)$ to refer to the function corresponding to the tuning curve of the input units with preferred direction θ_i—the mean activity in response to θ—and $g_i(\theta)$ the equivalent function for the output units.

In response to a direction θ_0, the mean activity vector for the input units is given by $\{f_i(\theta_0)\}_{i=1}^N$, and we will use boldface fonts, $\mathbf{f}(\theta_0)$, to refer to this vector. The same convention will be applied all the other functions used in the proof.

The functions $f(\theta_0)$ and $g(\theta_0)$ are defined with respect to the variable $\mathbf{O}_t$. There exist two corresponding functions for the activity variable $\mathbf{U}_t$, which we will denote $f^u(\theta_0)$ and $g^u(\theta_0)$, where $f(\theta_0) = h(f^u(\theta_0))$ and $g(\theta_0) = h(g^u(\theta_0))$, $h(\cdot)$ being the network activation function. $\mathbf{f}(\theta_0)$, $\mathbf{g}(\theta_0)$, $\mathbf{f}^u(\theta_0)$, and $\mathbf{g}^u(\theta_0)$ refer to the corresponding vectors of activity.

In the simulations, we initialized $\mathbf{U}_0$ to $W\mathbf{A}$ and $\mathbf{O}_0$ to $h(\mathbf{U}_0)$ to simulate the propagation of activity through the feedforward connections. To simplify notations in the proof, we will consider instead that $\mathbf{O}_0$ is initialized to $\mathbf{A}$ and $\mathbf{U}_0$ to $h^{-1}(\mathbf{A})$. This modification does not affect the proof because the initialization used in the simulations is equivalent to one iteration of the output network with the integration constant, λ, set to 1, and it turns out that the eigenvectors of the Jacobian for the output network are independent of λ. We will look at the case where $\mathbf{A}$, and therefore $\mathbf{O}_0$, is distributed according to a normal distribution $\mathcal{N}(\langle\mathbf{A}\rangle, \Sigma_0)$ with $\langle\mathbf{A}\rangle = \mathbf{f}(\theta_0)$ and Σ_0 diagonal with all the diagonal terms equal to σ_n^2.

6.2 Linearization. We consider the case in which the functions f and g (and f^u and g^u) are identical.

Since $\mathbf{A}$ is a random variable, we can think of this system as being a random process that generates a temporal sequence of random variables, $\{\mathbf{O}_0, \mathbf{O}_1, \ldots, \mathbf{O}_t, \ldots, \mathbf{O}_\infty\}$, where $\mathbf{O}_0 = \mathbf{A}$, and $\{\mathbf{U}_0, \mathbf{U}_1, \ldots, \mathbf{U}_t, \ldots, \mathbf{U}_\infty\}$.

We first note that our network is globally stable since the dynamics minimizes a Lyapunov function of the form (Cohen & Grossberg, 1983):

$$L = -\frac{1}{2}\sum_{i,j} w_{ij}h(u_i)h(u_j) + \sum_i \int_0^{u_i} zh'(z)\,dz.$$

Since the weights were chosen to solve $\mathbf{g} = h(W\mathbf{g})$, we know that a hill of profile $\mathbf{g}(\theta_0)$, peaking at any location of the neuronal array, is a fixed point. In terms of the variable $\mathbf{U}_t$ the stable activity profile is given by the function $\mathbf{g}^u(\theta_0)$.

Since we consider the case where $\mathbf{f}^u(\theta_0) = \mathbf{g}^u(\theta_0)$, $\mathbf{f}^u(\theta_0)$ is a stable state. Moreover, for small enough noise, most initial patterns, $\mathbf{U}_0$, are less than ϵ away from the stable manifold, that is, the Euclidean distance between $\mathbf{U}_0$ and the nearest point on the manifold is less than ϵ, where ϵ is a small number. Consequently, we can study the behavior of our network by linearizing equation 6.4 around $\langle\mathbf{U}_0\rangle = \mathbf{g}^u(\theta_0)$. Let J^T be the Jacobian of the function $e(\cdot)$ (see equation 6.4) evaluated at $\langle\mathbf{U}_0\rangle$ (we use J^T instead of J to simplify notation later on):

$$\mathbf{U}_t = e(\mathbf{U}_{t-1}) \tag{6.5}$$

$$\approx e(\langle\mathbf{U}_0\rangle) + J^T(\mathbf{U}_{t-1} - \langle\mathbf{U}_0\rangle). \tag{6.6}$$

Combining equation 6.6 and the fact that $e(\langle \mathbf{U}_0 \rangle) = \langle \mathbf{U}_0 \rangle$ (the mean $\langle \mathbf{U}_0 \rangle$ is a stable state), we find that:

$$\tilde{\mathbf{U}}_t \approx J^T \tilde{\mathbf{U}}_{t-1}, \tag{6.7}$$

where $\tilde{\mathbf{U}}_t = \mathbf{U}_t - \langle \mathbf{U}_0 \rangle$. The transpose of the Jacobian J^T is of the form:

$$J^T = (1 - \lambda)I + \lambda W H', \tag{6.8}$$

where H' is a diagonal matrix whose diagonal terms are equal to $h'(g_i^u(\theta_0))$.

We can obtain a similar linear equation for the variable $\tilde{\mathbf{O}}_t$. Indeed, linearizing equation 6.1 yields:

$$\tilde{\mathbf{O}}_t \approx H' \tilde{\mathbf{U}}_t.$$

If we substitute equation 6.8 in equation 6.7 and multiply both sides by H', we obtain:

$$\begin{aligned} H' \tilde{\mathbf{U}}_t &\approx H'((1 - \lambda)I + \lambda W H') \tilde{\mathbf{U}}_{t-1} \\ \tilde{\mathbf{O}}_t &\approx (1 - \lambda) \tilde{\mathbf{O}}_{t-1} + \lambda H' W \tilde{\mathbf{O}}_{t-1}. \end{aligned}$$

Since H' is diagonal and W is symmetric, $H'W = (WH')^T$, which entails:

$$\tilde{\mathbf{O}}_t \approx J \tilde{\mathbf{O}}_{t-1}.$$

Therefore, the Jacobian for $\tilde{\mathbf{O}}_t$ is J. Iterating this equation leads to:

$$\tilde{\mathbf{O}}_t \approx J^t \tilde{\mathbf{O}}_0.$$

$\tilde{\mathbf{O}}_0$ is distributed according to $\mathcal{N}(\mathbf{0}, \Sigma_0)$, where $\mathbf{0}$ is a vector of N zeroes. Since $\tilde{\mathbf{O}}_t$ is related to $\tilde{\mathbf{O}}_0$ by a linear relationship, $\tilde{\mathbf{O}}_t$ is distributed according to $\mathcal{N}(\mathbf{0}, \Sigma_t)$, where:

$$\Sigma_t = J^t \Sigma_0 J^{tT}.$$

Let us define

$$\begin{aligned} J^\infty &= \lim_{t \to \infty} J^t \\ \tilde{\mathbf{O}}_\infty &= J^\infty \tilde{\mathbf{O}}_0. \end{aligned}$$

The existence of a bounded Lyapunov function ensures that all the eigenvalues of J are less than or equal to one, and therefore J^∞ exists. At equilibrium, we have:

$$\Sigma_\infty = J^\infty \Sigma_0 J^{\infty T}. \tag{6.9}$$

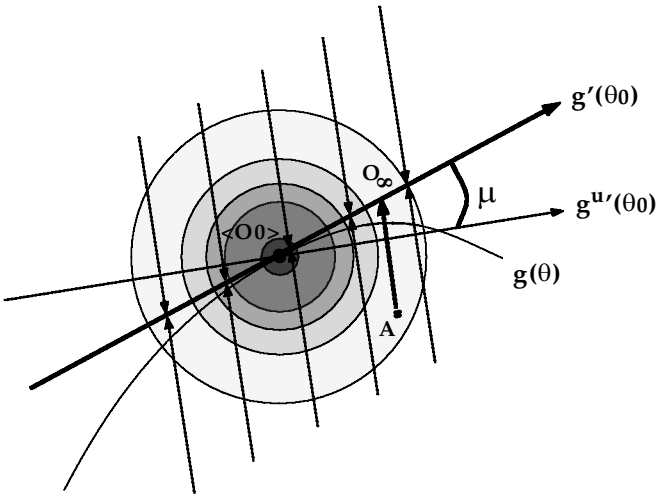

Figure 9: During relaxation, the initial activity $A = \mathbf{O}_0$ is projected onto the tangent, $\mathbf{g}'(\theta_0)$, of the stable manifold defined by the function $\mathbf{g}(\theta)$, along directions orthogonal to $\mathbf{g}^{u'}(\theta_0)$. As a result, the initial distribution of activity (shown as a density plot indicated by the gray circles) is collapsed onto the axis defined by $\mathbf{g}'(\theta_0)$.

6.3 Characterizing the Transformation J^∞. We now show that J^∞ is a projection on a line pointing in the direction of $\mathbf{g}'(\theta_0)$—the derivative of $\mathbf{g}$ with respect to θ evaluated at θ_0— along the directions orthogonal to $\mathbf{g}^{u'}(\theta_0)$ (see Figure 9).

6.3.1 Projection onto $\mathbf{g}'(\theta_0)$. First, we note that:

$$J^\infty \tilde{\mathbf{O}}_\infty = \tilde{\mathbf{O}}_\infty,$$

and furthermore:

$$\begin{aligned} J^\infty J^\infty \tilde{\mathbf{O}}_0 &= J^\infty \tilde{\mathbf{O}}_\infty \\ &= \tilde{\mathbf{O}}_\infty \\ &= J^\infty \tilde{\mathbf{O}}_0. \end{aligned}$$

This is true for arbitrary $\tilde{\mathbf{O}}_0$; thus:

$$J^\infty J^\infty = J^\infty,$$

which is the definition of a projection. Therefore, J^∞ is a projection onto the subspace spanned by $\tilde{\mathbf{O}}_\infty$. Next, we show that this subspace is a line pointing in the direction of $\mathbf{g}'(\theta_0)$.

The projecting space is spanned by the vectors, $\tilde{\mathbf{O}}$, which are solutions to:

$$J^{\infty}\tilde{\mathbf{O}} = \tilde{\mathbf{O}}. \tag{6.10}$$

The activity patterns that satisfy equation 6.10 correspond to stable states. Therefore $\tilde{\mathbf{O}} = \tilde{\mathbf{O}}_{\infty} = \mathbf{O}_{\infty} - \langle \mathbf{O}_0 \rangle$ where $\mathbf{O}_{\infty}$ and $\langle \mathbf{O}_0 \rangle$ are of the form:

$$\begin{aligned} \mathbf{O}_{\infty} &= \mathbf{g}(\theta_0 + \delta\theta) \\ \langle \mathbf{O}_0 \rangle &= \mathbf{g}(\theta_0) \end{aligned}$$

hence,

$$\begin{aligned} \tilde{\mathbf{O}} &= \mathbf{O}_{\infty} - \langle \mathbf{O}_0 \rangle && (6.11)\\ &= \mathbf{g}(\theta_0 + \delta\theta) - \mathbf{g}(\theta_0) && (6.12)\\ &\approx \delta\theta \mathbf{g}'(\theta_0). && (6.13) \end{aligned}$$

Therefore, J^{∞} is a projection onto $\mathbf{g}'(\theta_0)$. A similar analysis would show that $J^{\infty T}$ is a projection onto $\mathbf{g}^{u\prime}(\theta_0)$.

6.3.2 Projection Along the Directions Orthogonal to $\mathbf{g}^{u\prime}(\theta_0)$. To demonstrate that the projection is along the directions orthogonal to $\mathbf{g}^{u\prime}(\theta_0)$, we need to show that for any vector, $\mathbf{g}^{u\prime}(\theta_0)^{\perp}$, orthogonal to $\mathbf{g}^{u\prime}(\theta_0)$ (i.e., $\mathbf{g}^{u\prime}(\theta_0)^{T}\mathbf{g}^{u\prime}(\theta_0)^{\perp} = 0$), we have, $J^{\infty}\mathbf{g}^{u\prime}(\theta_0)^{\perp} = 0$. We start from the fact that J^{∞} is a projection onto $\mathbf{g}'(\theta_0)$ and therefore:

$$\begin{aligned} J^{\infty}\mathbf{g}^{u\prime}(\theta_0)^{\perp} &= \alpha \mathbf{g}'(\theta_0) \\ \mathbf{g}^{u\prime}(\theta_0)^{T} J^{\infty}\mathbf{g}^{u\prime}(\theta_0)^{\perp} &= \alpha \mathbf{g}^{u\prime}(\theta_0)^{T}\mathbf{g}'(\theta_0) \\ (J^{\infty T}\mathbf{g}^{u\prime}(\theta_0))^{T}\mathbf{g}^{u\prime}(\theta_0)^{\perp} &= \alpha \mathbf{g}^{u\prime}(\theta_0)^{T}\mathbf{g}'(\theta_0) \\ \mathbf{g}^{u\prime}(\theta_0)^{T}\mathbf{g}^{u\prime}(\theta_0)^{\perp} &= \alpha \mathbf{g}^{u\prime}(\theta_0)^{T}\mathbf{g}'(\theta_0) \\ 0 &= \alpha \mathbf{g}^{u\prime}(\theta_0)^{T}\mathbf{g}'(\theta_0). \end{aligned}$$

Since, in general (and in our simulations), $\mathbf{g}^{u\prime}(\theta_0)^{T}$ and $\mathbf{g}'(\theta_0)$ are not orthogonal, we can conclude that:

$$\alpha = 0.$$

In other words, any vector orthogonal to $\mathbf{g}^{u\prime}(\theta_0)$ is an eigenvector of J^{∞} whose eigenvalue is zero. Therefore, J^{∞} is a projection on $\mathbf{g}'(\theta_0)$ along the directions orthogonal to $\mathbf{g}^{u\prime}(\theta_0)$ (see Figure 9A).

Next, we show that the resulting estimator is unbiased and has a variance close to the Cramér-Rao bound.

6.4 Properties of the Network Estimate.

6.4.1 Unbiased Estimator. Since $\tilde{\mathbf{O}}_\infty$ is distributed according to $\mathcal{N}(\mathbf{0}, \Sigma_\infty)$, we have $\langle \mathbf{O}_\infty \rangle = \langle \mathbf{O}_0 \rangle = \mathbf{g}(\theta_0)$, and $\langle \mathbf{O}_\infty \rangle = \mathbf{f}(\theta_0)$ when the functions f and g are identical. This entails that the final activity, $\mathbf{O}_\infty$, is an unbiased estimate of the initial activity $\mathbf{O}_0$.

The network estimate θ_{RN} is obtained by applying a complex estimator to $\mathbf{O}_\infty$. The complex estimator is an unbiased estimate of direction when applied to $\mathbf{O}_0$. Since $\mathbf{O}_\infty$ is an unbiased estimate of $\mathbf{O}_0$, the complex estimator applied to $\mathbf{O}_\infty$, that is, $\hat{\theta}_{RN}$, is unbiased.

6.4.2 Variance of the Network Estimate. Let σ^2_{CR} be the variance corresponding to the Cramér-Rao bound. If the activity of the units, $o_{i,0}$, is independent and normally distributed according to $\mathcal{N}(f_i(\theta_0), \sigma_n^2)$, we have (from equation 3.2):

$$\sigma^2_{CR} = \frac{\sigma_n^2}{\|\mathbf{f}'(\theta_0)\|^2}.$$

We now show that the variance of the network estimate, $\sigma^2_{\hat{\theta}_{RN}}$, is close to σ^2_{CR}.

At the end of relaxation, all the patterns, $\tilde{\mathbf{O}}_\infty$, are confined to the axis defined by $\mathbf{g}'(\theta_0)$. Therefore, the covariance matrix is of the form:

$$\Sigma_\infty = \sigma^2_\infty \frac{\mathbf{g}'(\theta_0)\mathbf{g}'(\theta_0)^T}{\|\mathbf{g}'(\theta_0)\|^2}, \tag{6.14}$$

where σ^2_∞ is the variance of the norm of $\tilde{\mathbf{O}}_\infty$ along the axis $\mathbf{g}'(\theta_0)$. Different patterns correspond to the stable hill placed at different locations. Using equation 6.13, we can now show that σ^2_∞ is related to the variance of the network estimate, $\sigma^2_{\hat{\theta}_{RN}}$, through the following relationship:

$$\begin{aligned}\sigma^2_\infty &= \left\langle \|\mathbf{O}_\infty - \langle \mathbf{O}_\infty \rangle\|^2 \right\rangle \\ &\approx \|\mathbf{g}'(\theta_0)\|^2 \sigma^2_{\hat{\theta}_{RN}}.\end{aligned}$$

Therefore:

$$\sigma^2_{\hat{\theta}_{RN}} \approx \frac{\sigma^2_\infty}{\|\mathbf{g}'(\theta_0)\|^2}. \tag{6.15}$$

Combining equations 6.9, 6.14, and 6.15, we get:

$$\sigma^2_{\hat{\theta}_{RN}} \mathbf{g}'(\theta_0)\mathbf{g}'(\theta_0)^T \approx J^\infty \Sigma_0 J^{\infty T}.$$

We now multiply both sides of this equation by $\mathbf{g}^{u'}(\theta_0)$ on the left and $\mathbf{g}^{u'}(\theta_0)^T$ on the right:

$$\sigma^2_{\hat{\theta}_{RN}} \mathbf{g}^{u'}(\theta_0)^T \mathbf{g}'(\theta_0)\mathbf{g}'(\theta_0)^T \mathbf{g}^{u'}(\theta_0) \approx \mathbf{g}^{u'}(\theta_0)^T J^{\infty} \Sigma_0 J^{\infty T} \mathbf{g}^{u'}(\theta_0).$$

Since $J^{\infty T}\mathbf{g}^{u'}(\theta_0) = \mathbf{g}^{u'}(\theta_0)$ (from the fact that $\mathbf{g}^{u'}(\theta_0)$ is stable):

$$\sigma^2_{\hat{\theta}_{RN}} (\mathbf{g}'(\theta_0)^T \mathbf{g}^{u'}(\theta_0))^2 \approx \mathbf{g}^{u'}(\theta_0)^T \Sigma_0 \mathbf{g}^{u'}(\theta_0)$$

$$\sigma^2_{\hat{\theta}_{RN}} \approx \frac{\mathbf{g}^{u'}(\theta_0)^T \Sigma_0 \mathbf{g}^{u'}(\theta_0)}{(\mathbf{g}'(\theta_0)^T \mathbf{g}^{u'}(\theta_0))^2}.$$

If $f = g$ and $\Sigma_0 = \sigma_n^2 I$:

$$\begin{aligned}
\sigma^2_{\hat{\theta}_{RN}} &\approx \sigma_n^2 \frac{\|\mathbf{f}^{u'}(\theta_0)\|^2}{(\mathbf{f}^{u'}(\theta_0)^T \mathbf{f}'(\theta_0))^2} \\
&\approx \sigma_n^2 \frac{\|\mathbf{f}^{u'}(\theta_0)\|^2}{\|\mathbf{f}^{u'}(\theta_0)\|^2 \|\mathbf{f}'(\theta_0)\|^2 \cos^2 \mu} \\
&\approx \frac{\sigma_n^2}{\|\mathbf{f}'(\theta_0)\|^2 \cos^2 \mu},
\end{aligned}$$

where μ is the angle between the vector $\mathbf{f}'(\theta_0)$ and $\mathbf{f}^{u'}(\theta_0)$.

Therefore, $\sigma^2_{\hat{\theta}_{RN}}$ differs from σ^2_{CR} by a factor inversely proportional to $\cos^2 \mu$ when $f = g$ and $\Sigma_0 = \sigma_n^2 I$. In general, the angle μ will be small if the activation function, h, is close to linear within the network dynamical range. With the tuning curves and activation function we used, the $\cos^2 \mu$ term makes $\sigma^2_{\hat{\theta}_{RN}}$ 2% larger than σ^2_{CR}.

Given the small influence of this term, we will ignore it in the rest of the article. This amounts to treating J^{∞} as an orthogonal projection onto $\mathbf{g}'(\theta_0)$. Projecting the initial activity orthogonally onto $\mathbf{g}'(\theta_0)$ amounts to finding the stable state that minimizes the square distance with $\mathbf{O}_0$. In the presence of independent gaussian noise of equal variance, the ML estimate is also the peak position of the stable state, which minimizes the square distance with the initial activity.

6.5 Nonoptimal Cases.

6.5.1 Nonequal Variance. For arbitrary gaussian noise with covariance matrix Σ_0, the ML estimate is the direction that minimizes the Mahalanobis distance between $\mathbf{O}_0$ and $\mathbf{f}(\theta)$:

$$\theta_{ML} = \arg\min_{\theta} (\mathbf{O}_0 - \mathbf{f}(\theta))^T \Sigma_0^{-1} (\mathbf{O}_0 - \mathbf{f}(\theta)).$$

Since our network minimizes the square distance, it will be suboptimum whenever this Mahalanobis distance differs from the Euclidean distance. This is the case, in particular, when some neurons are noisier than others, that is, when the variance of the noise is not the same for all units.

6.5.2 Correlations. In general, our method is also suboptimal when the activity of the units is correlated. However, it is still optimum for certain types of correlations. For gaussian noise with arbitrary covariance matrices, Σ_0, the variance of the Cramér-Rao bound (obtained from equation 3.1) and the variance of the network estimate (ignoring the difference between $\mathbf{f}'(\theta_0)$ and $\mathbf{f}^{u'}(\theta_0)$, and under the assumption that $f = g$) are given by:

$$\sigma^2_{\hat{\theta}_{RN}} \approx \frac{\mathbf{f}'(\theta_0)^T \Sigma_0 \mathbf{f}'(\theta_0)}{(\mathbf{f}'(\theta_0)^T \mathbf{f}'(\theta_0))^2}$$

$$\sigma^2_{CR} = \frac{1}{\mathbf{f}'(\theta_0)^T \Sigma_0^{-1} \mathbf{f}'(\theta_0)} .$$

These two quantities are equal if and only if $\mathbf{f}'(\theta_0)$ is an eigenvector of Σ_0. This is the case in particular for the covariance matrix of the stable state, Σ_∞. Indeed, all the variance in this case is along the axis defined by $\mathbf{f}'(\theta_0)$. It would be easy to show that this is also the case for any of the covariance matrices Σ_t. In other words, covariance introduced by iterating the network does not affect performance, which is precisely why we reach the Cramér-Rao bound at the end of relaxation.

6.5.3 Large Noise. The size of the domain in which our linear approximation works depends on the amplitude of the second and higher derivatives of the tuning function, h. The activation function we have used is flat for negative inputs and rises almost linearly after a threshold (see Figure 3D). Except for the fast transition from flat to linear rise, the high-order derivatives are all small. This predicts that the network should be able to handle a fairly large amount of noise and still provide optimal performance.

Another factor allows the network to be robust with respect to noise. In our simulations, $\mathbf{U}_0$ is initialized to $W\mathbf{A}$. This first linear averaging step increases the signal-to-noise ratio by a factor proportional to $\sqrt{N}$, where N is the number of input units (since $w_{ij} \sim 1/N$; Zhang, 1996). Therefore, in the simulations, the size of the domain in which our approximation applies is proportional to ϵ for $\mathbf{U}_0$ but $\sqrt{N}\epsilon$ for $\mathbf{A}$.

Our simulations confirm that our network can indeed handle a fairly large amount of noise without a significant decrease in performance. Hence, we have found that signal-to-noise ratio (the ratio of α/σ_n; see equation 2.1) as low as 3 leads to a standard deviation within 5% of the Cramér-Rao bound.

6.5.4 Nongaussian Distributions. Nongaussian noise distributions are a problem only in the first one or two iterations. The central limit theorem states that the average of a large number of random variables converges to a normal distribution. Since $\mathbf{U}_0$ is initialized to $W\mathbf{A}$, $\mathbf{U}_0$ will be normally distributed in the limit of a large number of units. Even if $\mathbf{U}_0$ is not close to a normal distribution, $\mathbf{U}_1$ or $\mathbf{U}_2$ will be, since the averaging process is repeated on each iteration. How much information will be lost in the first iterations cannot be determined in general and depends on the noise distribution.

In the case of a Poisson distribution, the convergence to a normal distribution is likely to be fast since such a distribution is similar to a normal distribution with the variance equal to the mean. Our network is no longer optimum, but our simulations confirm that performance is still close to maximum likelihood.

6.5.5 Different Input and Output Functions. When the input and output functions, f and g, differ, the performance of the network is difficult to predict in the general case. For small differences, however, the linear approximation leads to:

$$\begin{aligned}\tilde{O}_\infty &= J_\infty(O_0 - \langle O_0 \rangle) \\ &= J_\infty(O_0 - \mathbf{f}(\theta_0)) \\ &= J_\infty(O_0 - \mathbf{f}(\theta_0) + \mathbf{g}(\theta_0) - \mathbf{g}(\theta_0)) \\ &= J_\infty(O_0 - \mathbf{g}(\theta_0)) - J_\infty(\mathbf{g}(\theta_0) - \mathbf{f}(\theta_0)).\end{aligned}$$

As long as $\mathbf{g}(\theta_0) - \mathbf{f}(\theta_0)$ is orthogonal to $\mathbf{g}^{u'}(\theta_0)$, $J_\infty(\mathbf{g}(\theta_0) - \mathbf{f}(\theta_0)) = 0$, and the network behaves as if f and g were identical. This is the case, in particular, when f and g differ by their width or amplitude. Indeed, f and g are even functions, whereas $\mathbf{g}^{u'}(\theta_0)$ is an odd function. This explains why performance is minimally affected by such changes, as shown in the simulations (see Figure 7).

6.6 Relation to Linear ML Estimator. Discrimination tasks have been widely used in psychophysics to probe the representation of sensory variables such as orientation or direction. In one variation of the task, subjects are presented with two possible directions, $\theta_0 \pm \delta\theta$, in rapid succession. The task is to determine whether the temporal sequence is $\theta_0 + \delta\theta$ followed by $\theta_0 - \delta\theta$, or the reverse.

Assuming that this task is performed on the basis of the response of direction-tuned neurons such as the ones we have used so far, optimal performance can be obtained by looking at the sign of the difference between the ML estimation of the first and second direction. This reduces to a linear problem when the reference direction, θ_0, is kept constant. Therefore, this task can be performed optimally by a two-layer network (Pouget & Thorpe, 1991; Seung & Sompolinsky, 1993).

Three-layer networks are required when more than one reference direction is used (Pouget & Thorpe, 1991; Mato & Somplinsky, 1996), and mixtures of expert architecture can work for any arbitrary direction, but a large number of hidden units and a gating network are necessary for optimal performances (Mato & Sompolinsky, 1996).

Our network provides an alternative method that does not require a dedicated hidden layer. The iterative process converges onto a linear operator J^{∞}, which is the optimal linear operator for the reference direction θ_0.

7 Discussion

Our results demonstrate that it is possible to perform efficient, unbiased estimation with coarse coding using a neurally plausible architecture. This shows that one of the advantages of coarse codes is to provide a representation that simplifies the problem of cleaning up uncorrelated noise within a neuronal population.

Our model relies on lateral connections to implement a prior expectation on the profile of the activity patterns. As a consequence, units determine their activation according to their own input and the activity of their neighbors. When the noise is small enough, this lateral pooling results in a near-orthogonal projection of the initial activity onto the tangent to the stable manifold; the stable hill corresponds to the one minimizing the square distance with the initial activity. Consequently, the network is very close to ML when the noise is normally distributed with equal variance in each unit.

Cleaning up noise efficiently does not entail that the lateral connections increase the signal-to-noise ratio, or the information content, of the representation. It is a well-known result in information theory that data processing cannot increase information content (Cover & Thomas, 1991; this result holds for Shannon information, but the generalization to Fisher information is straightforward). The fact that we are within 2% of the Cramér-Rao bound when applying a complex estimator to the stable hill of the network demonstrates that our procedure preserves almost completely Fisher information. Our network, however, does not simply preserve Fisher information; it also changes the format of information to make it easily decodable. Whereas ML is the only way to decode the input pattern efficiently, a complex estimator, or even a linear estimator, is sufficient to decode the stable hill while reaching the Cramér-Rao bound (see Figure 10). One can therefore think of the relaxation of activity in the nonlinear recurrent network in two ways: as a clean-up mechanism or as a processing that makes information easier to decode.

If spike trains are the result of a Poisson process (Shadlen & Newsome, 1994), cleaning up noise efficiently is a critical problem for the cortex. As information is transmitted from one area to the next, noise increases, leading to wider and wider activity distribution. Eventually activities are bound to fall outside the neurons' bandwidth, resulting in information loss. Our pro-

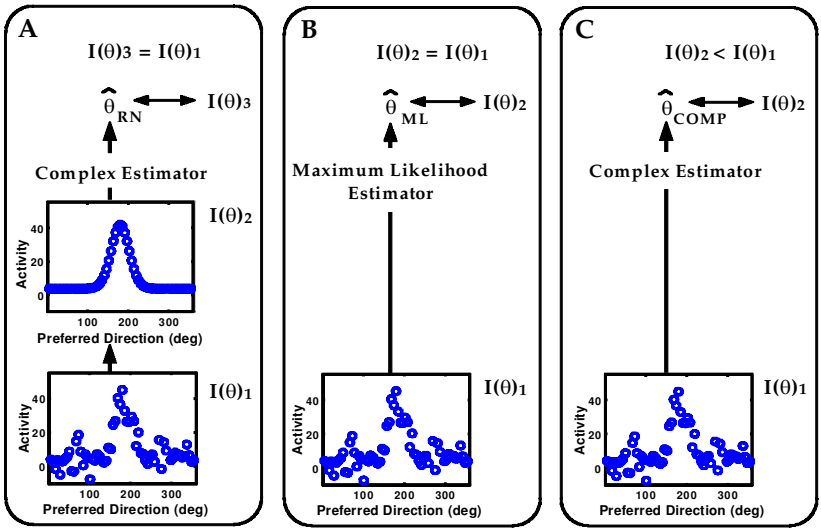

Figure 10: The COMP estimate preserves Fisher information, $I(\theta)_1$, when applied to the stable hill of the recurrent network (A)—as ML does (B)—but not when applied to the initial input (C). Therefore, the network dynamics changes the format of information such that a simple estimator can read out the activity optimally. $I(\theta)_1$, $I(\theta)_2$, and $I(\theta)_3$ refer to the Fisher information about direction at various stages in the estimation process.

cedure can prevent this problem by keeping the activities within a limited bandwidth while preserving the information content.

Unlike OLE, COM, and COMP, the RN estimate is not the result of a process in which units vote from their preferred direction, θ_i. Instead, units contribute according to the derivatives of their tuning curves, $f_i'(\theta)$, as in the case of ML. This feature allows the network to ignore background noise, that is, responses due to other factors beside the variable of interest. This property also predicts that discrimination of directions around the vertical (90 degrees) would be most affected by shutting off the units tuned at 60 and 120 degrees (assuming that the half-width of the tuning curves is around 30 degrees). This prediction is consistent with psychophysical experiments showing that discrimination around the vertical in humans is affected by prior adaptation to orientations displaced from the vertical by ± 30 degrees (Regan & Beverley, 1985).

As we have shown, the cleaning-up process is optimum only if the output and input units have the same tuning curves. It is worth mentioning that learning the weights of the lateral connections with a simple delta rule, a biologically plausible rule, would actually lead to an output pattern matching the input (Zhang, 1996). It is therefore possible that the match occurs

naturally in the cortex as the result of a self-organizing process.

The fact that optimum performance is obtained for matched input and output tuning curves has some interesting implications for orientation selectivity and the role of lateral connections in general in cortical processing. It argues that the pooled input to cortical neurons should have the same mean tuning as the output of the cells, a proposal in line with Hubel and Wiesel's (1962) model of orientation selectivity and recent experimental data by Ferster, Chung, and Wheat (1996). By contrast, several groups have proposed that lateral connections are used to sharpen tuning curves (Sillito, 1975; Heggelund, 1981; Wehmeier, Dong, Koch, & Van Essen, 1989; Wörgotter & Koch, 1991; Somers, Nelson, & Sur, 1995). Our work suggests that this sharpening process can only degrade the representation and that the role of lateral connections may be better described in terms of cleaning up noise, or changing the format of information, rather than sharpening tuning curves (Pouget & Zhang, 1996).

These considerations must be tempered by the fact that our attractor network is a poor model of cortical circuitry in V1. This model is neurally plausible in the same way Hopfield network are: its style of computation and the representation used are similar to the ones used in the cortex. Several aspects of this model, however, are clearly implausible. V1 circuits are not stable in the awake state, that is, V1 neurons do not keep on firing when the stimulus is extinguished, and inputs are typically not transient. We believe, however, that the modifications required will not affect these conclusions, and we intend to explore this issue further.

Our approach can be readily extended to any other periodic sensory or motor variables. For nonperiodic variables such as the disparity of a line in an image, our network needs to be adapted since it currently relies on circularly symmetric weights. Simply unfolding the network will be sufficient to deal with values around the center of the interval under consideration, but more work is needed to deal with boundary values. We can also generalize this work to arbitrary mapping between two coarse codes for variables x and y where y is a function of x. Indeed, a coarse code for x provides a set of radial basis functions of x that can be used subsequently to approximate arbitrary functions. It is even conceivable that a similar approach can be used for one-to-many mappings, a common situation in vision or robotics, by adapting our network such that several hills can coexist simultaneously. We are currently exploring such architectures.

Acknowledgments

This research was supported in part by a training grant from the McDonnell-Pew Center for Cognitive Neuroscience and a Department of Defense grant (DAMD17-93-V-3018) (A. P.). We thank Peter Dayan, Laurenz Wiskott, Terry Sanger, Richard Zemel, and an anonymous reviewer for their valuable comments and insightful suggestions.

References

Anderson, C. H. (1994). Basic elements of biological computational systems. *International Journal of Modern Physics C*, *5*(2), 135–137.

Baldi, P., & Heiligenberg, W. (1988). How sensory maps could enhance resolution through ordered arrangements of broadly tuned receivers. *Biological Cybernetics*, *59*(4–5), 313–318.

Ben-Yishai, R., Bar-Or, R. L., & Sompolinsky, H. (1995). Theory of orientation tuning in visual cortex. *Proceedings of the National Academy of Sciences USA, 92*, 3844–3848.

Cohen, M., & Grossberg, S. (1983). Absolute stability of global pattern formation and parallel memory storage by competitive neural network. *IEEE Transactions SMC*, *13*, 815–826.

Cover, T. M., & Thomas, J. A. (1991). *Elements of information theory*. New York: Wiley.

Desimone, R., Schein, S. J., Moran, J., & Ungerleider, L. G. (1985). Contour, color and shape analysis beyond the striate cortex. *Vision Research, 25*(3), 441–452.

Duda, R. O., & Hart, R. E. (1973). *Pattern classification and scene analysis*. New York: Wiley.

Ferster, D., Chung, S., & Wheat, H. (1996). Orientation selectivity of thalamic input to simple cells of cat visual cortex. *Nature*, *380*, 249–252.

Georgopoulos, A. P., Kalaska, J. F., Caminiti, R., & Massey, J. T. (1982). On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *Journal of Neuroscience, 2*(11), 1527–1537.

Green, D. M., & Swets, J. A. (1966). *Signal detection theory and psychophysics*. New York: Wiley.

Heggelund, P. (1981). Receptive field organization of simple cells in cat striate cortex. *Experimental Brain Research, 42*, 89–98.

Hinton, G. E. (1992). How neural networks learn from experience. *Scientific American*, *267(3)*, 145–151.

Hirsch, M., & Smale, S. (1974). *Differential equations, dynamical systems and linear algebra*. New York: Academic Press.

Hubel, D., & Wiesel, T. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal of Physiology (London)*, *160*, 106–154.

Lehky, S. R., & Sejnowski, T. J. (1990). Neural model of stereoacuity and depth interpolation based on a distributed representation of stereo disparity. *Journal of Neuroscience, 10(7)*, 2281–2299.

Mato, G., & Sompolinsky, H. (1996). Neural network models of perceptual learning of angle discrimination. *Neural Computation, 8*, 270–299.

Maunsell, J. H. R., & Van Essen, D. C. (1983). Functional properties of neurons in middle temporal visual area of the Macaque monkey. I. Selectivity for stimulus direction, speed, and orientation. *Journal of Neurophysiology, 49*(5), 1127–1147.

Papoulis, A. (1991). *Probability, random variables, and stochastic process*. New York: McGraw-Hill.

Paradiso, M. A. (1988). A theory of the use of visual orientation information which exploits the columnar structure of striate cortex. *Biological Cybernetics*,

58, 35–49.
Pouget, A., Fisher, S. A., & Sejnowski, T. J. (1993). Egocentric spatial representation in early vision. *Journal of Cognitive Neuroscience*, *5*, 150–161.
Pouget, A., & Thorpe, S. J. (1991). Connectionist models of orientation identification. *Connection Science*, *3(2)*, 127–142.
Pouget, A., & Zhang, K. (1996). A statistical perspective on orientation selectivity in primary visual cortex. In *Society for Neuroscience Abstracts*, vol. 22.
Regan, D. M., & Beverley, K. I. (1985). Post-adaptation orientation discrimination. *Journal of Optical Society of America*, *2*, 147–155.
Salinas, E., & Abbott, L. F. (1994). Vector reconstruction from firing rate. *Journal of Computational Neuroscience*, *1*, 89–108.
Seung, H. S., & Sompolinsky, H. (1993). Simple model for reading neuronal population codes. *Proceedings of National Academy of Sciences, USA*, *90*, 10749–10753.
Shadlen, M. N., & Newsome, W. T. (1994). Noise, neural codes and cortical organization. *Current Opinion in Neurobiology*, *4*, 569–579.
Sillito, A. M. (1975). The contribution of inhibitory mechanisms to the receptive field properties of neurones in the striate cortex of the cat. *Journal of Physiology (London)*, *250*, 305–329.
Snippe, H. P. (1996). Parameter extraction from population codes: A critical assessment. *Neural Computation*, *8*(3), 511–530.
Somers, D. C., Nelson, S. B., & Sur, M. (1995). An emergent model of orientation selectivity in cat visual cortical simple cells. *Journal of Neuroscience*, *15*(8), 5448–5465.
Wehmeier, U., Dong, D., Koch, C., & Van Essen, D. (1989). Modelling the visual system. In C. Koch & I. Segev (Eds.), *Methods in neural modelling* (pp. 335–359). Cambridge, MA: MIT Press.
Wilson, M. A., & McNaughton, B. L. (1993). Dynamics of the hippocampal ensemble code for space, *Science*, *261*, 1055–1058.
Wörgotter, F., & Koch, C. (1991). A detailed model of the primary visual pathway in the cat: Comparison of afferent excitatory and intracortical inhibitory connection schemes for orientation selectivity. *Journal of Neuroscience*, *11*, 1959–1979.
Zemel, R. S., Dayan, P., & Pouget, A. (1997). Population code representations of probability density functions. In M. C. Mozer, M. I. Jordan, & T. Petsche (Eds.), *Advances in neural information processing systems*, *9*. Cambridge, MA: MIT Press.
Zemel, R. S., Dayan, P., & Pouget, A. (1998). Probabilistic interpolation of population codes. *Neural Computation*, *10*(2), 403–430.
Zhang, K. (1996). Representation of spatial orientation by the intrisinc dynamics of the head-direction cell ensemble: A theory. *Journal of Neuroscience*, *16*(6), 2112–2126.
Zohary, E. (1992). Population coding of visual stimuli by cortical neurons tuned to more than one dimension. *Biological Cybernetics*, *66*, 265–272.
Zohary, E., Shadlen, M. N., & Newsome, W. T. (1994). Correlated neuronal discharge rate and its implication for psychophysical performance. *Nature*, *370*, 140–143.

Received September 23, 1996; accepted June 20, 1997.