Optimal computation with attractor networks

Peter E. Latham[1], Sophie Deneve[2], and Alexandre Pouget[3]

[1] Department of Neurobiology
University of California at Los Angeles
Los Angeles, CA 90095-1763

[2] Gatsby Computational Neuroscience Unit
University College London
London WC1 3AR

[3] Department of Brain and Cognitive Sciences
University of Rochester
Rochester, NY 14627

Submitted to *J. Physiol. (Paris)*

Abstract

We investigate the ability of multi-dimensional attractor networks to perform reliable computations with noisy population codes. We show that such networks can perform computations as reliably as possible – meaning they can reach the Cramér-Rao bound – so long as the noise is small enough. "Small enough" depends on the properties of the noise, especially its correlational structure. For many correlational structures, noise in the range of what is observed in the cortex is sufficiently small that biologically plausible networks can compute optimally. We demonstrate that this result applies to computations that involve cues of varying reliability, such as the position of an object on the retina in bright versus dim light.

1. Introduction

Many variables in the brain are encoded in the activity of large populations of neurons with bell-shaped tuning curves (see Fig. 1a). A critical question in neuroscience is: how do networks compute with these codes? How can a network extract, for example, the position of an object in head-centered coordinates from the position of the object on the retina and the position of the eyes in the head, given that all three variables are encoded by population activity? Tasks like this are made especially difficult by the variability in neuronal responses (Fig. 1b): neurons never fire with exactly the same pattern twice, even when an animal is performing identical tasks – say responding to the same stimulus, or producing the same motor response (Tolhurst *et al.* , 1983; Shadlen *et al.* , 1996; Gershon *et al.* , 1998).

The fact that population codes are noisy means that information is lost at every stage of processing, so there is pressure to perform computations reliably. To understand the limits of reliability, we consider a scenario in which a network receives as input information about a set of variables, each encoded in population activity, and the network performs some computation based on that input (such as the one mentioned above). The question we ask is: how reliably can the network do this? In other words, how much of the information in the input can the network extract while it's carrying out the computation?

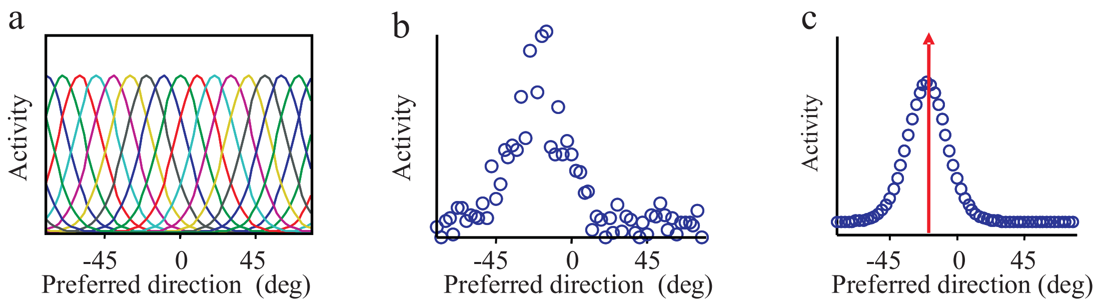

Figure 1: **a**. A set of tuning curves for the direction of motion of an object. Such tuning curves are found throughout the visual cortex, and in particular in area MT. **b**. A noisy pattern of activity for a population of neurons. This pattern arose from an object moving at -30^o. **c**. A smooth hill of activity. The networks we consider here evolve to a smooth hill like this one when initialized with the noisy pattern in panel b. The position of the peak of the smooth hill (marked with a red arrow) provides an estimate of direction of motion. With proper tuning of its parameters, a network can recover the optimal estimate; that is, it can evolve to a smooth hill without losing any of the information coded in the noisy hill.

As a first step toward answering this question, we consider a restricted class of networks for which smooth hills of activity are stable. When a network within this class is initialized with noisy population activity – with noisy hills of activity – the network eventually evolves onto smooth hills, like the one shown in Fig. 1c. Once the smooth hill is obtained, its peak constitutes an estimate of the value of the variable encoded in the noisy hill. The crucial question is whether this estimate can be optimal, that is, whether the estimate can be computed with no loss of information.

For the simple case of a single variable encoded in population activity, as in Fig. 1, we found in our previous work that there is a network that produces an optimal estimate of the encoded variable (Pouget *et al.* , 1998). This result holds so long as the noise is Poisson and is independent among neurons. In a subsequent study, we extended this finding to networks encoding multiple independent variables (Deneve *et al.* , 1999). More recently, we presented simulations suggesting an even more general result: networks encoding multiple variables, related to one another through nonlinear transformations, can be tuned to perform optimal computation even when the reliability of the input variables change from trial to trial (Deneve *et al.* , 2001).

In this paper we prove the above general result. Our proof applies to the case in which the evolution of the network is noise free, which means the only source of noise is the noise corrupting the input hills. Given this assumption, we derive conditions for the existence of a network that can perform optimal computations; that is, for a network that can manipulate population codes without losing any of the information in the input. The conditions are very general and relatively simple; they depend only on the correlational structure of the noise in the input. Interestingly, for small enough noise there is always a network that can perform computations optimally. However, the size of "small enough" depends in detail on the correlational structure.

Letting the network evolve noise-free is a big approximation; we make it because it allows us to derive powerful results telling us when a network can carry out computations reliably and when it can't. The more realistic case of internal noise (synaptic failures, stochastic ion channels, etc.) can be handled by considering the evolution of probability distributions over neuronal activity rather than the neuronal activity itself. This case will be considered in future work; our underlying assumption in this paper is that, for small enough internal noise, the deterministic evolution should provide a reasonable first approximation to the true probabilistic evolution (see Fig. 3).

This paper is arranged as follows. In Sec. 2 we provide an intuitive explanation of how neuronal networks perform efficient estimation. Section 3 contains a formal derivation of our main result, that *any* recurrent network exhibiting an M-dimensional attractor is capable of performing as well as the best possible estimator in the limit of small noise. We provide an estimate of the size of the noise for this result to hold, and show that for uncorrelated noise, or correlated noise that is stimulus independent, it need only be $\mathcal{O}(1)$. However, if the noise is correlated and stimulus-dependent, it must be $\mathcal{O}(1/N)$. In Sec. 4 we extend this result to networks in which the reliability of stimuli is variable. In Sec. 5 we consider an example: correlated, Poisson-like neurons, for which $\mathcal{O}(1/N)$ noise is required to perform optimal computations for the class of networks considered in Secs. 2 and 3. We show that a network does exist that computes optimally for $\mathcal{O}(1)$ noise. However, that network is not so easily implemented in a biological network, and does not readily generalize. Section 6 contains our summary and conclusions.

2. Extracting information from noisy neurons: general considerations

Biological organisms must estimate stimuli from noisy neuronal responses. They must be able, for example, to translate from the noisy hill of activity in Fig. 1a to the value of the variable encoded by that hill, or perform a computation by combining the noisy hills associated with several variables. (We are using "stimulus" in a very general sense;

a stimulus could be an external variable, such as the direction of a moving object, or an internal variable, such as the position of the eyes relative to the head. Stimuli could even consist of some combination of external and internal variables.) The question we ask in this paper is: how well can biologically plausible networks carry out these estimation tasks? In particular, can they do as well as the best possible estimator; that is, can they reach the Cramér-Rao bound (Cover & Thomas, 1991)? Surprisingly, the answer to the latter question is yes, so long as certain conditions are met. In the next section we derive those conditions; in this section we provide an intuitive explanation of why biologically plausible networks might be able to act as optimal estimators.

Formally, the brain performs estimation by implementing a mapping from a set of neuronal responses, denoted $\mathbf{a} \equiv (\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_N)$, to a stimulus or set of stimuli, denoted $\mathbf{s}$. For simplicity, in this section we take both the stimulus and each of the neuronal responses to be one-dimensional; so we let $(\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_N) \to (a_1, a_2, ..., a_N)$ and $\mathbf{s} \to s$, where the a_i and s are scalar variables. For example, the a_i might be firing rates and s the direction of a moving object, as in Fig. 1. We show in the next section, however, that our results apply even when both the stimulus and the response of each neuron is multi-dimensional.

We start by assuming that some estimator exists; i.e., that there is some function of $\mathbf{a}$, denoted $\hat{s}(\mathbf{a})$, that provides an estimate of the stimulus, s. The estimator $\hat{s}(\mathbf{a})$ can be thought of as a many-to-one mapping from $\mathbf{a}$ to $\hat{s}$. To make this explicit, we write

$$\hat{s} = \hat{s}(\mathbf{a})\,. \tag{1}$$

The observation that allows us to construct a network estimator out of the general estimator, $\hat{s}(\mathbf{a})$, is that Eq. (1) can be inverted to provide a one-to-many map from $\hat{s}$ to $\mathbf{a}$. Specifically, if we view activity space as an N-dimensional space whose coordinates are $(a_1, a_2, ..., a_N)$, then, for each value of $\hat{s}$, the set of a_i that satisfies Eq. (1) forms an (N-1)-dimensional subspace, denoted $\mathbf{a}(\hat{s})$. The key feature of this subspace is that every point in it leads to the same estimate, $\hat{s}$, of the stimulus, s; i.e., every point in the subspace $\mathbf{a}(\hat{s})$ produces the same value for $\hat{s}(\mathbf{a})$. If we could construct a network that maps the whole (N-1)-dimensional space to a single point, the location of that point would provide a natural estimate of $\hat{s}$.

Attractor networks (Hopfield, 1982; Hopfield, 1984; Droulez & Berthoz, 1991; Zhang, 1996; Camperi & Wang, 1998; Laing & Chow, 2001; Wang, 2001) could perform such a mapping. These networks evolve in time, starting from some initial condition, to an attractor – a sub-manifold of their full activity space. An attractor network could, then, take as initial conditions the population activity, $\mathbf{a}$, and evolve in time such that the whole (N-1)-dimensional subspace, $\mathbf{a}(\hat{s})$, goes eventually to the same point. In the full N-dimensional activity space, such a network would admit a line-attractor, so constructing a network estimator out of the general estimator $\hat{s}(\mathbf{a})$ reduces to the problem of finding the appropriate line-attractor network.

Figure 2 shows schematically how a line-attractor network could act as an estimator. The activity, $\mathbf{a}$, in response to a stimulus, s, corresponds to an initial condition for the attractor network. Each initial condition lies on *some* (N-1)-dimensional subspace; i.e., every $\mathbf{a}$ solves Eq. (1) for some $\hat{s}$. Two such subspaces are shown in Fig. 2. Under the action of the line-attractor network, every point in a particular subspace evolves to the same final point, and that point lies on the line labeled $\mathbf{g}(s)$. For example all points lying on the sheet $\mathbf{a}(\hat{s}_1)$ evolve to $\mathbf{g}(\hat{s}_1)$ and all points on the sheet $\mathbf{a}(\hat{s}_2)$ evolve to $\mathbf{g}(\hat{s}_2)$. The final position on the line $\mathbf{g}(s)$ represents the network estimate of the stimulus, s.

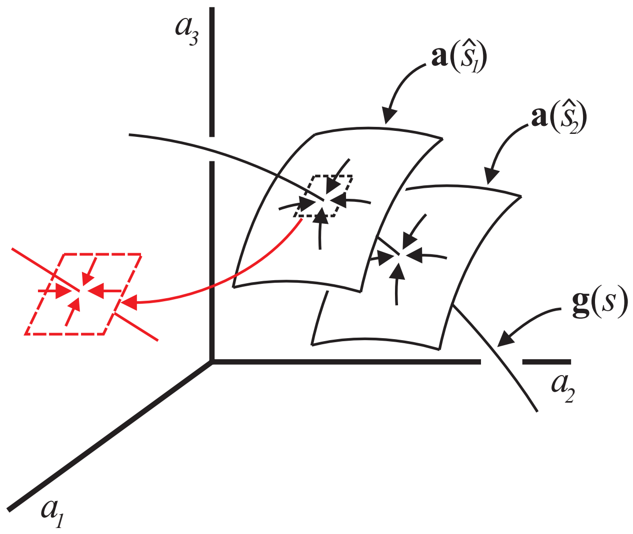

Figure 2: Schematic (3-dimensional cut) of a line-attractor network that mimics $\hat{s}(\mathbf{a})$. The sheets represent (N-1)-dimensional subspaces for two different values of $\hat{s}$; points on the sheets satisfy Eq. (1), with $\hat{s} = \hat{s}_1$ for the upper one and $\hat{s} = \hat{s}_2$ for the lower. Arrows on the sheets indicate trajectories. The line labeled $\mathbf{g}(s)$ is the line-attractor: all initial conditions evolve to some point on this line. The position on the line provides the estimate, $\hat{s}$, of the true stimulus, s. A blowup of the region near the line attractor, shown in red, indicates that the subspaces are locally flat. Consequently, for initial conditions close enough to the line attractor, $\mathbf{g}(s)$, the trajectories of the line-attractor network are well approximated by straight lines. For a network to mimic the estimator $\hat{s}(\mathbf{a})$, at least in the small noise limit, it is necessary that the trajectories be parallel to $\mathbf{a}(\hat{s})$ when $\mathbf{a}$ is near $\mathbf{g}(s)$.

This analysis indicates that every line-attractor corresponds to *some* estimator. The question of interest is: can a line-attractor network do as well as the best possible estimator? This is a hard question to answer in general. However, it's tractable in the limit of small noise. In this limit, the initial condition, $\mathbf{a}$, is near the line attractor, $\mathbf{g}(s)$, which allows us to treat the (N-1)-dimensional subspaces as linear spaces and the trajectories as straight and locally parallel to $\mathbf{a}(\hat{s})$ (red blowup in Fig. 2). Consequently, we can use linear analysis to compute the quality of the network estimator for any line-attractor network – that is, we can compute how well $\hat{s}$ approximates s. This is a key point, because knowing the quality of the estimator for any line-attractor network allows us to find the best possible line-attractor network. Moreover, we can show that the best possible line-attractor network really is good: if the noise is small – $\mathbf{a}(\hat{s})$ is close to the line $\mathbf{g}(s)$, where close is relative to the curvature of $\mathbf{a}(\hat{s})$ – we are guaranteed that the best possible line-attractor network does as well as the optimal estimator, the latter assessed by the Cramér-Rao bound.

The quality of the linear approximation depends, of course, on the smoothness of $\mathbf{a}(\hat{s})$: if $\mathbf{a}(\hat{s})$ exhibits sharp curvature, then our analysis applies only if the noise is very small (see Sec. 5 for an example). In the extreme case in which $\mathbf{a}(\hat{s})$ exhibits one or more singularities, our analysis would break down if the line attractor passed through any of them. Thus, the smoothness of $\mathbf{a}(\hat{s})$ must be checked on a case-by-case basis. For the remainder of this paper, however, we simply assume that $\mathbf{a}(\hat{s})$ is locally smooth.

Although the above discussion focused on a one-dimensional stimulus and one-dimensional responses, the ideas apply to higher-dimensional stimuli and responses as well. In particular they apply to cases where several population codes are combined, as in the three-dimensional case alluded to in the introduction (populations codes for the position of an object on the retina and the position of the eyes in the head are combined to produce a population code for the position of the object relative to the head).

3. Constructing networks that perform optimal estimation

We now show explicitly that attractor networks can act as optimal estimators, in the sense that the network estimate of a set of stimuli from noisy neuronal responses is as good as the best possible estimator. We do this in three steps: we 1) analyze the linearized dynamics of an attractor network, 2) derive an expression for the performance of the network in terms of the distance between the network estimates and the true stimuli, and 3) show how network parameters can be modified to optimize the estimates.

The problem we consider is the following. A set of M stimuli produce a particular pattern of activity. That pattern of activity is fed into a recurrent network that supports an M-dimensional attractor. The network then evolves deterministically in time until it converges onto the attractor (more accurately, until it get exponentially close to the attractor). The point on the attractor it converges to represents the network estimate of the M stimuli. If we denote the stimuli as $\mathbf{s} = (s_1, s_2, ..., s_M)$ and the estimates as $\hat{\mathbf{s}} = (\hat{s}_1, \hat{s}_2, ..., \hat{s}_M)$, we can think of this process as the mapping $\mathbf{s} \to \mathbf{a}(0) \to \hat{\mathbf{s}}$, where the mapping from the stimulus to the initial activity, $\mathbf{a}(0)$, is probabilistic and the mapping from the initial activity to the stimulus estimate is deterministic.

A key observation is that the second half of the mapping, from $\mathbf{a}(0)$ to $\hat{\mathbf{s}}$, may be implemented with an attractor network that evolves deterministically in time according to the equation

$$\tau \frac{d\mathbf{a}(t)}{dt} = \mathbf{H}(\mathbf{a}(t)) - \mathbf{a}(t) \,. \tag{2}$$

where τ is a time constant, $\mathbf{a}(t) = (\mathbf{a}_1(t), \mathbf{a}_2(t), ..., \mathbf{a}_N(t))$ represents the activity of the N neurons in the network, and $\mathbf{H}(\mathbf{a})$ is a function that contains all the details about the network – its connectivity and single neuron and synaptic properties. (The $\mathbf{a}_i(t)$ are vectors because the response of a single neuron may be multi-dimensional – latency to the first spike and spike count, for example.) Our underlying assumption is that $\mathbf{H}(\mathbf{a})$ is such that the network admits an M-dimensional attractor; that is, there is some smooth function, $\mathbf{g}(\mathbf{s})$, satisfying

$$\mathbf{g}(\mathbf{s}) = \mathbf{H}(\mathbf{g}(\mathbf{s})) \,. \tag{3}$$

Since $\mathbf{s}$ is an M-component vector, $\mathbf{g}(\mathbf{s})$ is an M-dimensional manifold.

The network is initialized by transient input at time $t = 0$; this input has both a deterministic and noise component,

$$\mathbf{a}(0) = \mathbf{f}(\mathbf{s}) + \mathbf{N}(\mathbf{s})\,. \tag{4}$$

Here $\mathbf{f}(\mathbf{s})$ is the deterministic tuning curve and $\mathbf{N}(\mathbf{s})$ is the noise. In the limit $t \to \infty$, $\mathbf{a}(t)$ approaches the attractor; i.e., $\lim_{t\to\infty} \mathbf{a}(t) = \mathbf{g}(\hat{\mathbf{s}})$. The point on the attractor, $\hat{\mathbf{s}}$, is the network estimate of the stimuli, $\mathbf{s}$.

Because the initial conditions are generated probabilistically, the estimate will be different on each trial. We will assume here that the network is unbiased; that is, averaged over trials, $\hat{\mathbf{s}}$ is equal to $\mathbf{s}$. Thus, the quality of the network is determined by how close $\hat{\mathbf{s}}$ is to $\mathbf{s}$ on average. For close, we will use the determinant of the covariance matrix. The covariance matrix, denoted $\langle \delta s_k \delta s_l \rangle$, is given by

$$\langle \delta s_k \delta s_l \rangle = \langle (\hat{s}_k(\mathbf{a}) - s_k)(\hat{s}_l(\mathbf{a}) - s_l) \rangle\,.$$

This expression is, of course, only valid for unbiased estimators. We use the log of the covariance matrix to asses the quality of the estimator because it determines, to a large extent, the mutual information between the noisy neuronal responses, $\mathbf{a}$, and the stimulus, $\mathbf{s}$: the smaller the determinant of the covariance matrix, the larger the mutual information (Brunel & Nadal, 1998). (Strictly speaking, this result applies only when the neurons are uncorrelated. We believe it applies also to correlated neurons, as long as the determinant of the covariance matrix is small. In any case, it is a good starting point.)

To compute the covariance matrix, we take a perturbative approach: we linearize Eq. (2) around a point on the attractor, compute the trajectories analytically, and find the approximate final position on the attractor given the initial condition. A difficulty arises because, unlike point (0-dimensional) attractors, there isn't any unique point on the M-dimensional attractor to linearize around. Because of this non-uniqueness, for now we perturb around an arbitrary point, say $\mathbf{a} = \mathbf{g}(\tilde{\mathbf{s}})$. In principle it doesn't matter what we choose for $\tilde{\mathbf{s}}$, so long as it's close to the starting point, $\mathbf{a}(0)$. However, as we will see below, there is one especially convenient choice for $\tilde{\mathbf{s}}$.

Letting

$$\mathbf{a}(t) = \mathbf{g}(\tilde{\mathbf{s}}) + \delta\mathbf{a}(t)\,, \tag{5}$$

inserting Eq. (5) into Eq. (2) and keeping only linear terms, we find that $\delta\mathbf{a}(t)$ evolves according to

$$\frac{d\delta\mathbf{a}}{dt} = \mathbf{J}(\tilde{\mathbf{s}}) \cdot \delta\mathbf{a} \tag{6}$$

where $\mathbf{J}$ is the Jacobian evaluated on the attractor,

$$J_{ij}(\tilde{\mathbf{s}}) \equiv \frac{\partial H_i(\mathbf{g}(\tilde{\mathbf{s}}))}{\partial g_j(\tilde{\mathbf{s}})} - \delta_{ij}\,, \tag{7}$$

δ_{ij} is the Kronecker delta, and we are using standard dot-product notation: the i^{th} component of $\mathbf{J} \cdot \delta\mathbf{a}$ is $\sum_j J_{ij}\delta a_j$.

Equation (6) has the solution

$$\delta\mathbf{a}(t) = \exp\Big(\mathbf{J}(\tilde{\mathbf{s}})t\Big) \cdot \delta\mathbf{a}(0)\,. \tag{8}$$

To cast Eq. (8) in a more useful form, we re-express $\mathbf{J}$ using of its eigenvector expansion,

$$\mathbf{J}(\tilde{\mathbf{s}}) = \sum_k \lambda_k(\tilde{\mathbf{s}})\mathbf{v}_k(\tilde{\mathbf{s}})\mathbf{v}_k^\dagger(\tilde{\mathbf{s}})$$

where $\mathbf{v}_k(\tilde{\mathbf{s}})$ is the eigenvector of $\mathbf{J}(\tilde{\mathbf{s}})$ with eigenvalue $\lambda_k(\tilde{\mathbf{s}})$ and $\mathbf{v}_k^\dagger(\tilde{\mathbf{s}})$ is the adjoint eigenvector, chosen so that $\mathbf{v}_k(\tilde{\mathbf{s}}) \cdot \mathbf{v}_l^\dagger(\tilde{\mathbf{s}}) = \delta_{kl}$. In terms of these eigenvectors and eigenvalues, Eq. (8) becomes

$$\delta\mathbf{a}(t) = \sum_k \exp\Big(\lambda_k(\tilde{\mathbf{s}})t\Big)\mathbf{v}_k(\tilde{\mathbf{s}})\mathbf{v}_k^\dagger(\tilde{\mathbf{s}}) \cdot \delta\mathbf{a}(0)\,. \tag{9}$$

Since Eq. (2) admits an attractor, M of the eigenvalues are zero – these correspond to perturbations along the attractor – and the rest are negative. For convenience, we rank the eigenvectors in order of decreasing eigenvalue, so $\mathbf{v}_k(\tilde{\mathbf{s}})$ and $\mathbf{v}_k^\dagger(\tilde{\mathbf{s}})$, $k = 1, ..., M$, are the eigenvectors and adjoint eigenvectors whose eigenvalues are zero. (Interestingly, the first M eigenvectors, $\mathbf{v}_k$, can be expressed in terms of $\mathbf{g}$: combining Eqs. (3) and (7), it is not hard to show that $\mathbf{v}_k(\mathbf{s}) = \partial_{s_k}\mathbf{g}(\mathbf{s})$.) In the limit that $t \to \infty$, the only terms in Eq. (9) that survive are the ones with $\lambda_k = 0$; we thus have

$$\lim_{t\to\infty} \delta\mathbf{a}(t) = \sum_{k=1}^{M} \mathbf{v}_k(\tilde{\mathbf{s}})\mathbf{v}_k^\dagger(\tilde{\mathbf{s}}) \cdot \delta\mathbf{a}(0)\,. \tag{10}$$

The value of $\delta\mathbf{a}(\infty)$ given in Eq. (10) tells us the final point on the attractor. Knowing $\delta\mathbf{a}(\infty)$ would allow us to find $\hat{\mathbf{s}}$ in terms of $\tilde{\mathbf{s}}$. However, it is more convenient to choose $\tilde{\mathbf{s}}$ so that $\delta\mathbf{a}(\infty) = 0$, because in that case, $\hat{\mathbf{s}} = \tilde{\mathbf{s}}$. The condition that $\delta\mathbf{a}(\infty) = 0$ is that $\mathbf{v}_k^\dagger(\hat{\mathbf{s}}) \cdot \delta\mathbf{a}(0) = 0$ for $k = 1, ..., M$. Using Eqs. (4) and (5) to express $\delta\mathbf{a}(0)$ in terms of $\mathbf{f}(\mathbf{s})$ and $\mathbf{N}(\mathbf{s})$, and replacing $\tilde{\mathbf{s}}$ with $\hat{\mathbf{s}}$, this condition translates into M equations,

$$\mathbf{v}_k^\dagger(\hat{s}) \cdot [\mathbf{f}(\mathbf{s}) + \mathbf{N}(\mathbf{s}) - \mathbf{g}(\hat{\mathbf{s}})] = 0 \tag{11}$$

for $k = 1, ..., M$.

To find $\hat{\mathbf{s}}$ in terms of $\mathbf{s}$, we let $\hat{\mathbf{s}} = \mathbf{s} + \delta\mathbf{s}$; term by term, this means that $\hat{s}_k = s_k + \delta s_k$. Expanding Eq. (11) to first order in δs, we arrive at the set of equations

$$\mathbf{v}_k^\dagger(\mathbf{s}) \cdot \mathbf{N}(\mathbf{s}) + \mathbf{v}_k^\dagger(\mathbf{s}) \cdot [\mathbf{f}(\mathbf{s}) - \mathbf{g}(\mathbf{s})] + \sum_l \delta s_l \partial_{\hat{s}_l}\Big(\mathbf{v}_k^\dagger(\hat{\mathbf{s}}) \cdot [\mathbf{f}(\mathbf{s}) + \mathbf{N}(\mathbf{s}) - \mathbf{g}(\hat{\mathbf{s}})]\Big)_{\hat{\mathbf{s}}=\mathbf{s}} = 0\,. \tag{12}$$

If the term $\mathbf{v}_k^\dagger(\mathbf{s})\cdot[\mathbf{f}(\mathbf{s})-\mathbf{g}(\mathbf{s})]$ does *not* vanish for all s, then, for some s, δs will be nonzero in the limit that the noise goes to zero, and the network will produce biased estimates. Conversely, if it does vanish, then the network estimator will be unbiased. We assume an unbiased estimator (a condition that must be checked for individual networks), which requires that

$$\mathbf{v}_k^\dagger(\mathbf{s})\cdot[\mathbf{f}(\mathbf{s})-\mathbf{g}(\mathbf{s})]=0 \tag{13}$$

for $k=1,...,M$. If Eq. (13) is satisfied, then Eq. (12) implies that, for small $\mathbf{N}$, $\delta\mathbf{s}\sim\mathbf{N}$. Thus, the term $\delta s_l \partial_{\hat{s}_l}\mathbf{v}_k^\dagger(\mathbf{s})\cdot\mathbf{N}(\mathbf{s})$ that appears in Eq. (12) is $\mathcal{O}(\mathbf{N}^2)$ and can be ignored. With this simplification, we find that $\delta\mathbf{s}$ is given by

$$\delta s_k=\sum_l[\mathbf{v}_k^\dagger(\mathbf{s})\cdot\partial_{s_l}\mathbf{f}(\mathbf{s})]^{-1}\mathbf{v}_l^\dagger(\mathbf{s})\cdot\mathbf{N}(\mathbf{s})\,. \tag{14}$$

In this expression, and in what follows, we are using a shorthand notation for the inverse of a matrix: $[A_{kl}]^{-1}\equiv[A^{-1}]_{kl}$. Thus, $[\mathbf{v}_k^\dagger(\mathbf{s})\cdot\partial_{s_l}\mathbf{f}(\mathbf{s})]^{-1}$ is the kl^{th} component of the inverse of the matrix $\mathbf{v}_k^\dagger(\mathbf{s})\cdot\partial_{s_l}\mathbf{f}(\mathbf{s})$. To derive Eq. (14) we used $(\partial_{s_l}\mathbf{v}_k^\dagger)\cdot[\mathbf{f}-\mathbf{g}]-\mathbf{v}_k^\dagger\cdot\partial_{s_l}\mathbf{g}=-\mathbf{v}_k^\dagger\cdot\partial_{s_l}\mathbf{f}$, which follows from Eq. (13).

Using Eq. (14), it is straightforward to compute the covariance matrix that determines the error in the estimate of the angles, and we find that

$$\langle\delta s_k\delta s_l\rangle=\left[\partial_{s_k}\mathbf{f}(\mathbf{s})\cdot\left(\sum_{ij}\mathbf{v}_i^\dagger(\mathbf{s})\,[\mathbf{v}_i^\dagger(\mathbf{s})\cdot\mathbf{R}(\mathbf{s})\cdot\mathbf{v}_j^\dagger(\mathbf{s})]^{-1}\,\mathbf{v}_j^\dagger(\mathbf{s})\right)\cdot\partial_{s_l}\mathbf{f}(\mathbf{s})\right]^{-1}$$

where $\mathbf{R}(\mathbf{s})$ is the noise covariance matrix,

$$\mathbf{R}(\mathbf{s})\equiv\langle\mathbf{N}(\mathbf{s})\mathbf{N}(\mathbf{s})\rangle\,.$$

Because we now have two covariance matrices, $\mathbf{R}$ and $\langle\delta s\delta s\rangle$, we will consistently refer to $\mathbf{R}$ as the noise covariance matrix and $\langle\delta s\delta s\rangle$ simply as the covariance matrix.

As discussed, above, our measure of the quality of the estimator is the determinant of the covariance matrix. To find the value of $\mathbf{v}_k^\dagger$ that minimizes this determinant, we use the relation $(d/dx)\log\det\mathbf{A}=\mathrm{Tr}\{\mathbf{A}^{-1}\cdot d\mathbf{A}/dx\}$ where Tr denotes trace. After straightforward, but tedious, algebra, we find that

$$\frac{d\log\det\langle\delta s\delta s\rangle}{d\mathbf{v}_k^\dagger}=2\sum_i\left[\mathbf{R}\cdot\mathbf{v}_i^\dagger[\mathbf{v}_i^\dagger\cdot\mathbf{R}\cdot\mathbf{v}_k^\dagger]^{-1}-[\partial_{s_k}\mathbf{f}\cdot\mathbf{v}_i^\dagger]^{-1}\partial_{s_i}\mathbf{f}\right]\,. \tag{15}$$

The determinant of the covariance matrix is minimized when the right hand side of Eq. (15) is zero. This occurs when

$$\mathbf{v}_k^\dagger\propto\mathbf{R}^{-1}\cdot\partial_{s_k}\mathbf{f}\,, \tag{16}$$

at which point the covariance matrix simplifies to

$$\langle \delta s_k \delta s_l \rangle = \left[\partial_{s_k} \mathbf{f} \cdot \mathbf{R}^{-1} \cdot \partial_{s_l} \mathbf{f} \right]^{-1} .$$

Thus, whenever Eqs. (13) and (16) are satisfied, the nonlinear recurrent network given in Eq. (2) leads to a covariance matrix such that

$$\det \langle \delta \mathbf{s} \delta \mathbf{s} \rangle = \frac{1}{\det \left[\partial_{\mathbf{s}} \mathbf{f}(\mathbf{s}) \cdot \mathbf{R}^{-1}(\mathbf{s}) \cdot \partial_{\mathbf{s}} \mathbf{f}(\mathbf{s}) \right]} .$$

The above analysis provided us with the best network estimator within the class of attractor networks, assuming the noise is small. How good is this network, and how small must the noise be? To answer these questions, we use the fact that the lower bound on the determinant of the covariance matrix is given by the inverse of the Fisher Information (Cover & Thomas, 1991),

$$\det \langle \delta \mathbf{s} \delta \mathbf{s} \rangle \geq \frac{1}{\det I} \tag{17}$$

where I is the Fisher information,

$$I_{kl} = \left\langle - \frac{\partial^2}{\partial s_k \partial s_l} \log P(\mathbf{a}(t=0)|\mathbf{s}) \right\rangle . \tag{18}$$

Equation (17) is the multi-dimensional analog of the Cramér-Rao bound.

To compute the Fisher information, Eq. (18), we need to know the distribution of the noise; i.e., we need to know the explicit form of $P(\mathbf{a}(0)|\mathbf{s})$. Let us consider two types of noise: Gaussian with an arbitrary correlation matrix, for which

$$P(\mathbf{a}(0)|\mathbf{s}) = \frac{\exp\left[-(\mathbf{a}(0) - \mathbf{f}(\mathbf{s})) \cdot \mathbf{R}^{-1}(\mathbf{s}) \cdot (\mathbf{a}(0) - \mathbf{f}(\mathbf{s}))/2 \right]}{\left[(2\pi)^N \det \mathbf{R}(\mathbf{s}) \right]^{1/2}} , \tag{19}$$

and Poisson with uncorrelated noise, for which

$$P(\mathbf{a}(0)|\mathbf{s}) = \prod_i \frac{f_i(\mathbf{s})^{a_i(0)} e^{-f_i(\mathbf{s})}}{a_i(0)!} . \tag{20}$$

In Eq. (19), a_i is firing rate, while in Eq. (20), a_i is the number of spikes in an interval. For the Poisson distribution the mean value of $\mathbf{a}(0)$ is $\mathbf{f}$, and the noise covariance matrix is given by

$$\langle (a_i(0) - f_i)(a_j(0) - f_j) \rangle_{\text{Poisson}} = f_i \delta_{ij} \equiv R_{ij}$$

where the subscript "Poisson" indicates an average over the probability distribution given in Eq. (20). Note that we are using the symbol $\mathbf{R}$ for the noise covariance matrix of both the Gaussian and Poisson distributions; which distribution we mean should be clear from the context.

The Fisher information, Eq. (18), for the two cases is given by (Abbott & Dayan, 1999)

$$
\begin{aligned}
I_{kl,\mathrm{Gaussian}} &= \partial_{s_k}\mathbf{f}\cdot\mathbf{R}^{-1}\cdot\partial_{s_l}\mathbf{f}+\frac{1}{2}\mathrm{Tr}\{\mathbf{R}^{-1}\cdot\partial_{s_k}\mathbf{R}\cdot\mathbf{R}^{-1}\cdot\partial_{s_l}\mathbf{R}\} && \text{(21a)}\\
I_{kl,\mathrm{Poisson}} &= \partial_{s_k}\mathbf{f}\cdot\mathbf{R}^{-1}\cdot\partial_{s_l}\mathbf{f}\,. && \text{(21b)}
\end{aligned}
$$

The trace term in Eq. (21a) is a non-negative definite matrix with respect to the indices k and l, as is the first term on the right hand side of Eq. (21a). Thus,

$$
\begin{aligned}
\det[I_{\mathrm{Gaussian}}] &\geq \det[\partial_s\mathbf{f}\cdot\mathbf{R}^{-1}\cdot\partial_s\mathbf{f}]\\
\det[I_{\mathrm{Poisson}}] &= \det[\partial_s\mathbf{f}\cdot\mathbf{R}^{-1}\cdot\partial_s\mathbf{f}]\,.
\end{aligned}
$$

For noise in which the noise covariance matrix, $\mathbf{R}$, depends on the stimulus, s, the network does not appear to reach the Cramér-Rao bound. However, for reasonable noise structures, it turns out that the second term in Eq. (21) vanishes as the noise goes to zero. Consider, for example, a covariance matrix in which the noise is modeled as an overall multiplicative term, which allows us to write $\mathbf{R}=\epsilon\hat{\mathbf{R}}$ where $\hat{\mathbf{R}}$ is independent of ϵ and ϵ vanishes as the noise vanishes. With this change of variable, Eq. (21a) becomes

$$
I_{kl,\mathrm{Gaussian}}=\epsilon^{-1}\partial_{s_k}\mathbf{f}\cdot\hat{\mathbf{R}}^{-1}\cdot\partial_{s_l}\mathbf{f}+\frac{1}{2}\mathrm{Tr}\{\hat{\mathbf{R}}^{-1}\cdot\partial_{s_k}\hat{\mathbf{R}}\cdot\hat{\mathbf{R}}^{-1}\cdot\partial_{s_l}\hat{\mathbf{R}}\}\,.
$$

As the noise, ϵ, goes to zero, the first term dominates and we recover the Cramér-Rao bound. For this to happen, we must have

$$
\epsilon\ll\frac{||\partial_{s_k}\mathbf{f}\cdot\hat{\mathbf{R}}^{-1}\cdot\partial_{s_l}\mathbf{f}||}{||\mathrm{Tr}\{\hat{\mathbf{R}}^{-1}\cdot\partial_{s_k}\hat{\mathbf{R}}\cdot\hat{\mathbf{R}}^{-1}\cdot\partial_{s_l}\hat{\mathbf{R}}\}||}\,.
$$

where $||\cdot||$ denotes a norm over the indices k and l (the details of the norm are not important; we are interested only in scaling with the number of neurons). The term in the denominator, the trace term, typically scales as N, the size of the noise covariance matrix. The scaling of the term in the numerator depends on the correlational structure. For uncorrelated noise, the numerator scales as $\partial_{s_k}\mathbf{f}\cdot\partial_{s_l}\mathbf{f}$, which is $\mathcal{O}(N)$. In this regime, the factor of N in the numerator and denominator cancel, and the network estimate is comparable to the Cramér-Rao bound whenever $\epsilon\ll\mathcal{O}(1)$. For correlated noise, however, the numerator typically asymptotes to a constant for large N (see App. A). Thus, for noise that is correlated and depends on the stimulus, the network does not reach the Cramér-Rao bound unless $\epsilon\ll\mathcal{O}(1/N)$. Such noise is much smaller than is observed in practice, indicating that networks in the class considered here can be sub-optimal.

4. Stimuli with variable reliability

The analysis in the previous section gave us an optimal network for fixed tuning curves and noise. In the real world, however, stimuli arrive with varying reliability: visual cues, for example, are more reliable in bright light than in dim light. Being able to deal with this situation is a difficult, yet critical, problem, because more than one cue may be available for inferring the value of perceptual variables. For instance, we often locate objects on the basis of their images and sounds, perceive the 3D structure of objects from binocular vision, extract shape from shading, determine structure from motion and perspective, and infer the position of our limbs from their image and proprioceptive feedback. Importantly, we perform these tasks accurately even though the reliability of any one of the cues can vary over a broad range.

Can a single network be optimal when the reliability of the cues is variable? The answer, of course, depends on how variability is encoded, but a reasonable assumption is that it is encoded in firing rate; that is, in the amplitude of the tuning curves, $\mathbf{f}(\mathbf{s})$. The question we address here is: if tuning curves are scaled by a constant factor to reflect the reliability of the cues, can the network still perform optimally?

Let us consider a network in which several stimuli are encoded in hills of activity, and the noise among different hills is independent; networks of this type were shown by Deneve et al. (2001) to be able to perform a broad range of computations optimally. In this type of network, the tuning curve, $\mathbf{f}(\mathbf{s})$, is concatenated into p tuning curves, $\mathbf{f}(\mathbf{s}) = (\mathbf{f}_1(\mathbf{s}), \mathbf{f}_2(\mathbf{s}), ..., \mathbf{f}_p(\mathbf{s}))$, one for each hill of activity (typically, $p = M$, but this is not necessarily the case). To mimic the variable reliability, we allow both the amplitudes of the individual tuning curves and the associated noise to be scaled. Given this scaling, the network is initialized via a slight modification of Eq. (4),

$$\mathbf{a}(0) = (\gamma_1\mathbf{f}_1(\mathbf{s}) + \beta_1\mathbf{N}_1(\mathbf{s}), \gamma_2\mathbf{f}_2(\mathbf{s}) + \beta_2\mathbf{N}_2(\mathbf{s}), ..., \gamma_p\mathbf{f}_p(\mathbf{s}) + \beta_p\mathbf{N}_p(\mathbf{s}))\,.$$

The independence of the noise among different hills implies that $\langle \mathbf{N}_i\mathbf{N}_j\rangle = 0$ if $i \neq j$. Assuming that a network exists that is optimal when $\gamma_i = \beta_i = 1$, we would like to know whether the same network is also optimal when γ_i and β_i are not equal to 1, and if so, how β_i should depend on γ_i to achieve optimality.

The two conditions for optimality are given in Eqs. (13) and (16). Consider first Eq. (13). In terms of the scaled, concatenated tuning curves, this equation becomes

$$\mathbf{v}_k^\dagger(\mathbf{s}) \cdot [\gamma_i\mathbf{f}_i(\mathbf{s}) - \mathbf{g}_i(\mathbf{s})] = 0 \tag{22}$$

for $i = 1, ..., p$. We will assume that Eq. (22) holds for all γ_i; this would be the case, for instance, if $\mathbf{v}_k^\dagger(\mathbf{s})$ were an odd function of its components and $\mathbf{f}_i(\mathbf{s})$ and $\mathbf{g}_i(\mathbf{s})$ were even functions. With this assumption, the network is optimal if Eq. (16) is satisfied. When $\gamma_i = \beta_i = 1$, Eq. (16) can be written

$$\langle \mathbf{NN}\rangle \cdot \mathbf{v}_k^\dagger = c_k\partial_{s_k}\mathbf{f} \tag{23}$$

where the c_k are a set of arbitrary constants and we used $\mathbf{R} = \langle \mathbf{NN}\rangle$. Since the noise associated with the different tuning curves are independent, Eq. (23) breaks up into p equations, one for each set of tuning curves,

$$\langle \mathbf{N}_i\mathbf{N}_i\rangle \cdot \mathbf{v}_k^\dagger = c_k\partial_{s_k}\mathbf{f}_i\,. \tag{24}$$

Scaling $\mathbf{f}_i$ by γ_i and $\mathbf{N}_i$ by β_i, Eq. (24) becomes

$$\beta_i^2 \langle \mathbf{N}_i \mathbf{N}_i \rangle \cdot \mathbf{v}_k^\dagger = \gamma_i c_k \partial_{s_k} \mathbf{f}_i \, . \tag{25}$$

Equation (25) is satisfied, and the network is optimal for tuning curves of arbitrary height, if $\beta_i = \gamma_i^{1/2}$. In other words, if the noise in the input to a network scales as the square root of the firing rate, then that network will be optimal, independent of the amplitude of the input. This is an important result, since the square root scaling is exactly what one finds for neurons that fire with Poisson statistics. Thus, Poisson statistics are in some sense optimal, at least for the kinds of networks we considered here, and nearly Poisson statistics, as are typically observed in cortical neurons (Tolhurst *et al.* , 1983; Shadlen *et al.* , 1996; Gershon *et al.* , 1998), are nearly optimal.

This result confirms what we found in our previous study using computer simulations (Deneve *et al.* , 2001), which is that basis function networks exhibiting attractor dynamics can perform optimal estimation, and they can do so even when cues arrive with varying degrees of reliability. This result applies to any set of variables linked to one another through a nonlinear mapping and coded in the noisy activity of a population of neurons. What we showed here is that a network must exist that computes, from the noisy population codes, the optimal estimate of these variables, and does so regardless of their reliability – so long as the noise is Poisson and the reliability is encoded in firing rate. In such networks, the basis functions enforce the nonlinear mapping between the variables and the attractor dynamics ensures optimal statistical performance.

5. Improved efficiency network

For correlated, stimulus-dependent noise, the class of networks considered in the previous sections reach the Cramér-Rao bound only when the noise is extremely small, on the order of $1/N$. Are there networks that can do better? The analysis of Sec. 2 indicates that there are; all that is required is an optimal estimator, $\hat{\mathbf{s}}(\mathbf{a})$, and an attractor network whose time evolution preserves its inverse, $\mathbf{a}(\hat{\mathbf{s}})$. The surface $\mathbf{a}(\hat{\mathbf{s}})$ may be highly curved, but in principle a network exists whose trajectories remain within the subspace $\mathbf{a}(\hat{\mathbf{s}})$ if they start within that subspace, as in Fig. 2.

To understand the properties of such a network, we consider the simple case of extracting the value of a stimulus that is encoded in the mean firing rate of a population of correlated neurons. For this case, we let the stimulus be one-dimensional – we refer to it simply as s – and we let a_i be the firing rate of the i^{th} neuron. For the conditional distribution at time $t = 0$, $P(\mathbf{a}(0)|s)$, we use the Gaussian distribution given in Eq. (19). (Note that the Gaussian distribution allows negative firing rates. While this is unrealistic, we use it because a more realistic probability distribution would greatly complicate the analysis without changing the underlying result.) In a slight departure from the previous section, we let the tuning curves be linear rather than hills of activity; that is, $f_i(s) = \text{constant} \times s$. For convenience, we set the constant to one, so $f_i(s) = s$. We let the noise covariance have the form

$$R_{ij} = \sigma^2(s)[\delta_{ij} + \rho(1-\delta_{ij})]\,. \tag{26}$$

With this choice for the noise, the stimulus-dependent variance, σ^2, is the same for each neuron, and the pairwise correlation coefficient, ρ, is the same for each pair. For simplicity, we let σ^2 be proportional to s: $\sigma^2(s) = \alpha s$. This would be the case for Poisson-like neurons, in which the error in the estimate of firing rate increases with firing rate; for truly Poisson neurons, α would be one. The correlations in Eq. (26) could come from common input.

The Fisher information, Eq. (21a), is given by (see App. B)

$$I = \frac{1}{\alpha s}\,\frac{N}{N\rho + 1 - \rho} + \frac{N}{2s^2} \tag{27}$$

where, as usual, there are N neurons. The second term in Eq. (27) corresponds to the trace term in Eq. (21a). As discussed in the previous section, in the large N limit this term is negligible compared to the first term only if the noise, α, is extremely small; for this example, it must be much smaller than $2s/N\rho$. Thus, unless $\alpha \ll 2s/N\rho$, the class of networks derived in the previous section will do poorly compared to the Cramér-Rao bound.

To see how to construct a more efficient network, we compute the maximum likelihood estimator. This is done by maximizing $\log[P(\mathbf{a}(0)|s)]$, the log likelihood of the conditional distribution. A straightforward calculation (see App. B) yields

$$\frac{d\log P(\mathbf{a}(0)|s)}{ds} = \frac{N}{2s\sigma^2}\left[\sigma^2 + \frac{s^2 - \overline{a}^2}{N\rho + 1 - \rho} - \frac{\overline{\delta a^2}}{1-\rho}\right] \tag{28}$$

where $\overline{a} \equiv N^{-1}\sum_i a_i(0)$ is the initial mean and $\overline{\delta a^2} \equiv N^{-1}\sum_i a_i^2(0) - \overline{a}^2$ is the initial variance.

Setting the right hand side of Eq. (28) to zero and solving for s in terms of $\mathbf{a}$ yields the maximum likelihood estimator, which we denote $\hat{s}_{ML}(\mathbf{a})$. What does the surface $\mathbf{a}(\hat{s}_{ML})$ look like; i.e., what is the shape of the surface in activity space that satisfies $d\log P(\mathbf{a}|s)/ds = 0$? To answer this, it is convenient to make the orthogonal change of variables

$$\mathbf{a}(0) = \sum_{k=0}^{N-1} x_k \mathbf{u}_k \tag{29}$$

where

$$\mathbf{u}_0 = N^{-1/2}(1, 1, ..., 1)\,, \tag{30}$$

i.e., $u_{0i} = N^{-1/2}\,\forall i$, and the $\mathbf{u}_k$ are orthogonal: $\mathbf{u}_k \cdot \mathbf{u}_l = \delta_{kl}$. With this change of variables, $d\log P(\mathbf{a}(0)|s)/ds = 0$ when

$$\frac{x_0^2}{N\rho + 1 - \rho} + \sum_{k=1}^{N-1}\frac{x_k^2}{1-\rho} = N\alpha s + \frac{N}{N\rho + 1 - \rho}s^2\,. \tag{31}$$

The surface associated with Eq. (31) is thus cigar shaped, with the long axis pointing in the $\mathbf{u}_0$ direction and an aspect ratio of $[(N\rho + 1 - \rho)/(1-\rho)]^{1/2} \sim N^{1/2}$. Thus, when N is large, the surface is extremely long and thin. This makes the curvature very tight, so it is not surprising that the linear approximation breaks down and the network derived using linear perturbation does not provide a good estimate unless $\alpha \ll 1/N$.

To understand how to derive a better network estimator, we need an expression for the maximum likelihood estimator. Setting the right hand side of Eq. (28) to zero and using the relation $\sigma^2(s) = \alpha s$, we see that, in the large N limit, this estimator is given by

$$\hat{s}_{ML}(\mathbf{a}) = \frac{\overline{\delta a^2}}{\alpha(1-\rho)} \,. \tag{32}$$

As we show in App. B, in the large N limit, $\hat{s}_{ML}(\mathbf{a})$ is unbiased and its variance is $2s^2/N$, the same as the Cramér-Rao bound. Thus, the estimator derived from maximum likelihood is efficient, in the sense that it reaches the Cramér-Rao bound. That $\hat{s}_{ML}(\mathbf{a})$ is efficient is a peculiarity of high dimensional spaces: for correlated variables, in the large N limit, the mean has a variance that is $\mathcal{O}(1)$ while the variance has a variance that is $\mathcal{O}(1/N)$. The maximum likelihood estimator given in Eq. (32) makes use of this fact, along with the fact that the variance scales with the mean. This result should dispel the myth that averaging large numbers of correlated neurons does not improve the estimate of correlated firing rates (Zohary *et al.* , 1994; Shadlen *et al.* , 1996) – it does improve the estimate; one just has to compute the variance, not the mean.

Is there a neuronal network that can estimate s with a variance equal to the minimum, $2s^2/N$? In principle, yes, but it requires nonlinear synapses. For instance, consider the set of network equations

$$\tau \frac{d\mathbf{a}}{dt} = \mathbf{u}_\perp - [\mathbf{a} - \mathbf{u}_0\mathbf{u}_0 \cdot \mathbf{a}] \, \frac{\mathbf{u}_\perp \cdot \mathbf{a}}{\mathbf{a}\cdot\mathbf{a} - (\mathbf{u}_0 \cdot \mathbf{a})^2} - \mathbf{u}_0(\mathbf{u}_0 \cdot \mathbf{a} - c_0)$$

where $\mathbf{u}_0$ is given in Eq. (30) and $\mathbf{u}_\perp$ is any vector orthogonal to $\mathbf{u}_0$, normalized so that $\mathbf{u}_\perp \cdot \mathbf{u}_\perp = 1$. It is not hard to show that this equation admits a line attractor. In particular, if $\mathbf{a}(0) = \mathbf{f}(s) + \mathbf{N}(s)$, then $\mathbf{a}$ asymptotes to the point $N\overline{\delta a^2}\mathbf{u}_\perp + c_0\mathbf{u}_0$ as $t \to \infty$. Once the network has asymptoted to that point, $\overline{\delta a^2}$ is known, and thus so is the maximum likelihood estimate of s, $\hat{s}_{ML}(\mathbf{a})$, via see Eq. (32).

Unfortunately, it is not clear that such a network is biologically plausible. Nor is it clear that such a network would generalize: we were able to find an analytic expression for $\hat{s}_{ML}(\mathbf{a})$, and thus a network that would compute it, only because we chose a very simple model. For more realistic, and thus more complex models, we do not know how easily an estimator can be found. Nevertheless, it may be possible to train an attractor network so that it's trajectories are confined to the highly curved subspaces that arise when the covariance matrix depends on the stimulus.

6. Discussion

The brain has a hard job: it must store and manipulate vast quantities of information, it must do so quickly and accurately, and it must do so with underlying elements – neurons –

that are not very reliable. In other words, the brain must carry out complex computations using populations of neurons that never fire precisely the same way more than once, even on identical tasks. The question we asked in this paper was: how can biologically plausible networks carry out these tasks with as little information loss as possible?

To address this question, we focused on a particular class of networks: multi-dimensional attractor networks, which are recurrent networks that relax onto a line or higher dimensional manifold in activity space. We chose these networks for several reasons: they are biologically plausible, in the sense that they mimic the highly recurrent connectivity seen in cortex (Braitenberg & Schüz, 1991), there is experimental evidence for the existence of line-attractor networks that code for head direction in rats (Taube *et al.* , 1990a; Taube *et al.* , 1990b), and they can perform a large range of computations (Deneve *et al.* , 2001).

We asked the following question: suppose a multi-dimensional attractor network is initialized with noisy input coding for a set of variables, and after initialization it evolves noise-free. Can the network manipulate the encoded variables – carry out a computation – while extracting all the information contained in the noisy input? What we showed analytically is that the answer is yes, provided only that the noise in the input is small. The size of "small" turns out to depend on the structure of the noise. If the noise among the different input neurons is uncorrelated, or if it is correlated but independent of the encoded variables, then "small" is with respect to $\mathcal{O}(1)$. If the noise is correlated *and* depends on the encoded variables, then "small" is with respect to $\mathcal{O}(1/N)$ where N is the number of neurons. In the latter case, there may be a network that does compute optimally; indeed, the analysis in Secs. 2 and 5 suggests that there is. However, we were not able to prove the existence of such an optimal network in general.

Constructing networks that can perform optimally for fixed input is valuable, but in the real world input often arrives with varying degrees of reliability. For example, if input codes for the position of an object on the retina, that input will be reliable in bright light but unreliable in dim light. Perhaps surprisingly, it turns out that the networks we considered can perform optimally when cues that arrive with varying degrees of reliability, so long as two conditions are met: reliability is coded in the amplitude of the activity (e.g., the firing rate), with more reliable cues exhibiting large amplitudes, and the variance in the noise is proportional to the mean activity. These are both characteristic of cortical neurons, for which the variance in spike count is approximately proportional to the mean (Tolhurst *et al.* , 1983; Shadlen *et al.* , 1996; Gershon *et al.* , 1998). Thus, cortical networks may be able to make use of the natural variability in firing patterns to perform optimal computations.

There are two caveats to this study. The first is that multi-dimensional attractors are structurally unstable, in the sense that small perturbations in network parameters tend to cause systematic drift along the attractor (Zhang, 1996; Wang, 2001). If the drift is too fast, then the network can no longer act as an optimal estimator. However, if the drift is slow relative to the relevant timescale (often only a few hundred ms, but sometimes much longer), then it can be ignored. In addition, Wu and Amari (2002) showed recently that the drift can be stabilized by suitable synaptic facilitation.

The second caveat is that we considered noise-free evolution. In essence, we ignored internal sources of noise that are known to exist in biological networks, such as synaptic failures and stochastic ion channels. Thus, the deterministic network evolution, which we considered here, should be thought of as an approximation to the true probabilistic evolution.

If the internal noise is sufficiently small and/or well behaved, however, networks that are optimal for noise-free evolution should also be near-optimal for noisy evolution, as indicated in Fig. 3.

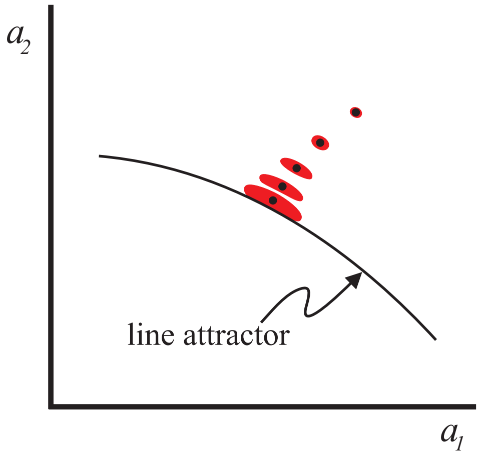

Figure 3: Snapshots of deterministic (black) and probabilistic (red) trajectories in activity space. With no internal noise, the network evolves noise free and follows the black points toward the line attractor. With internal noise, trajectories have a random component, as indicated by the expanding red blobs.

We have shown that biologically plausible recurrent networks can perform optimal computations with noisy population codes, at least for uncorrelated or stimulus-independent noise in the input and for noise-free evolution. This is a first step toward understanding how spiking networks, which do not evolve noise-free, can perform optimal computations, and how they can do so when the noise is correlated and/or stimulus-dependent.

Acknowledgments

P.L. was supported by NIMH Grant #R01 MH62447. A.P. and S.D. were supported by a fellowship from the Sloan Foundation, a young investigator award from ONR (N00014-00-1-0642) and a research grant from the McDonnell-Pew foundation.

Appendix A: Scaling of the Fisher information

The size of the noise for which the perturbatively derived network reaches the Cramér-Rao bound depends on how the first term in the Fisher information (Eq. (21a)) scales with N, the number of neurons. The second term in Eq. (21a) is, if nonzero, proportional to N, so unless the first term also scales as N, the second will dominate. For correlated noise, the first term typically asymptotes to a constant as N becomes large. We will not prove this, as there are counterexamples (Abbott & Dayan, 1999). Instead, we will motivate it using generic arguments, then illustrate those arguments with an example.

For simplicity, we consider the one-dimensional case, so the stimulus, s, is a scalar variable. Then, the first term in Eq. (21a), which we denote I_1, is given by

$$I_1 = \partial_s \mathbf{f} \cdot \mathbf{R}^{-1} \cdot \partial_s \mathbf{f} .$$

Our main tool for studying I_1 is the eigenvalue expansion of $\mathbf{R}^{-1}$,

$$\mathbf{R}^{-1} = \sum_k \frac{\mathbf{u}_k \mathbf{u}_k}{\lambda_k} \tag{A1}$$

where the $\mathbf{u}_k$ are the eigenvectors of $\mathbf{R}$, chosen to be orthogonal ($\mathbf{u}_k \cdot \mathbf{u}_l = \delta_{kl}$), and the λ_k are the corresponding eigenvalues. Using Eq. (A1), I_1 becomes

$$I_1 = \sum_k \frac{(\partial_s \mathbf{f} \cdot \mathbf{u}_k)^2}{\lambda_k} . \tag{A2}$$

To make a crude estimate of scaling with N, we make the following observations: When there are correlations, many of the entries in R_{ij} are nonzero; consequently, the λ_k scale as N. Because of the orthogonality conditions, the individual terms in $\mathbf{u}_k$ scale as $N^{-1/2}$. Since there are N terms in the dot-product, $\partial_s \mathbf{f} \cdot \mathbf{u}_k$, it scales as $N^{1/2}$ and its square scales as N. The factors of N in the numerator and denominator thus cancel, and I_1 scales as

$$I_1 = \sum_k \xi_k$$

where ξ_k is $\mathcal{O}(1)$. Although there are N terms in the sum, only a finite number of them contribute. This is because $\partial_s f_i$ is typically a smooth function of i, while the higher order eigenvectors are rapidly varying. Thus, I_1 is $\mathcal{O}(1)$.

Let us see how this works for the particular example of a translation invariant noise covariance matrix, $R_{ij} = R_{kl}$ if $i - j = k - l$. Typically, R_{ij} depends smoothly on the difference $i - j$, except when $i - j = 0$ (as R_{ii} is the variance). We thus write

$$R_{jl} = r_0 \delta_{jl} + r_{j-l}$$

where r_j is a smooth function of j. The eigenvectors of R_{jl} are exponentials; letting u_{k_j} be the j^{th} component of $\mathbf{u}_k$, we have:

$$u_{k_j} = \frac{e^{2\pi i j k/N}}{N^{1/2}} .$$

Consequently, the eigenvalues are given by

$$\lambda_k = r_0 + N r(k) \tag{A3}$$

where $r(k)$ is the discrete Fourier transform of r_j,

$$r(k) = \frac{1}{N} \sum_j r_j e^{2\pi i j k/N} .$$

Define also $\partial_s f(k)$ as the discrete Fourier transform of $\partial_s f_i(s)$,

$$\partial_s f(k) = \frac{1}{N^{1/2}} \partial_s \mathbf{f} \cdot \mathbf{u}_k = \frac{1}{N} \sum_j \partial_s f_j e^{2\pi i j k/N} \,. \tag{A4}$$

Both $r(k)$ and $\partial_s f(k)$ are $\mathcal{O}(1)$.

Combining Eqs. (A3) and (A4) with (A2), we arrive at

$$I_1 = \sum_k \frac{|\partial_s f(k)|^2}{r_0/N + r(k)} \,. \tag{A5}$$

In the limit of large N, we can replace the sum in Eq. (A5) by an integral, yielding

$$I_1 = \int dk \, \frac{|\partial_s f(k)|^2}{r_0/N + r(k)}$$

which is clearly independent of N in the limit $N \to \infty$.

Although we have not proved that I_1 is independent of N as $N \to \infty$ in general (as, in fact, it isn't), we have shown one common correlational structure for which I_1 does asymptote to a constant. In addition, using the eigenvector expansion for the covariance matrix, we argued that this is a relatively robust feature. It requires only that the eigenvalues of $\mathbf{R}$ scale as N and that the dot product, $\partial_s \mathbf{f} \cdot \mathbf{u}_k$, makes a non-negligible contribution to I_1 only for a finite set of k, even as N goes to ∞.

Appendix B: Correlated neurons with firing rate proportional to the mean

In this Appendix we fill in many of the missing steps in Sec. 5. We 1) compute the Fisher information for a Gaussian distribution with noise covariance matrix given in Eq. (26), 2) compute the derivative of the log likelihood, Eq. (28), and 3) show that the maximum likelihood estimator is unbiased and efficient (i.e., it reaches the Cramér-Rao bound).

All of these results rely on the properties of the noise covariance matrix. To streamline our calculations, we begin by expressing this matrix in terms of its eigenvectors and eigenvalues. We start by rewriting slightly Eq. (A1) to explicitly take into account the overall factor σ^2,

$$\mathbf{R}^{-1} = \sigma^{-2}(s) \sum_k \frac{\mathbf{u}_k \mathbf{u}_k}{\lambda_k} \tag{B1}$$

where $\mathbf{R}$ is given in Eq. (26), the $\mathbf{u}_k$ are the orthogonal eigenvectors of $\mathbf{R}/\sigma^2$, and the λ_k are the corresponding eigenvectors. (This is the same basis chosen in Eq. (29), so $\mathbf{u}_0$ is given by Eq. (30).) It is not hard to show that there are two distinct eigenvalues, $N\rho + 1 - \rho$, which appears once and $1 - \rho$, which appears N-1 times. In a slight abuse of notation, we define

$$\lambda_0 \equiv N\rho + 1 - \rho \tag{B2a}$$
$$\lambda_1 \equiv 1 - \rho \,. \tag{B2b}$$

Since $f_i(s) = s\,\forall i$, $\mathbf{f}(s)$ can be expressed in terms of $\mathbf{u}_0$ as

$$\mathbf{f}(s) = sN^{1/2}\,\mathbf{u}_0\,. \tag{B3}$$

We can now compute the Fisher information, Eq. (21). Recalling that $\sigma^2(s) = \alpha s$, so that $\partial_s\mathbf{R} = s^{-1}\mathbf{R}$, and using Eq. (B3) for $\mathbf{f}(s)$, we have

$$I = N\mathbf{u}_0 \cdot \mathbf{R}^{-1} \cdot \mathbf{u}_0 + \frac{1}{2s^2}\mathrm{Tr}\{\mathbf{R}^{-1} \cdot \mathbf{R} \cdot \mathbf{R}^{-1} \cdot \mathbf{R}\}\,.$$

Using Eq. (B1) for $\mathbf{R}^{-1}$, the orthogonality conditions on the $\mathbf{v}_k$, and the relation $\mathrm{Tr}\{\mathbf{I}\} = N$ where $\mathbf{I}$ is the identity matrix (not to be confused with the Fisher information), it is trivial to show that the Fisher information reduces to the expression in Eq. (27).

To derive the maximum likelihood estimator, we differentiate $\log \mathbf{P}(\mathbf{a}|s)$ with respect to s, where $P(\mathbf{a}|s)$ is given in Eq. (19). (We use $\mathbf{P}(\mathbf{a}|s)$ rather than $\mathbf{P}(\mathbf{a}(0)|s)$ for clarity.) Denoting differentiation with a prime and again applying the relation $(d/dx)\log\det\mathbf{A} = \mathrm{Tr}\{\mathbf{A}^{-1}d\mathbf{A}/dx\}$, we have

$$\frac{d\log\mathbf{P}(\mathbf{a}|s)}{ds} = \mathbf{f}'(s)\cdot\mathbf{R}^{-1}\cdot(\mathbf{f}(s)-\mathbf{a}) + \frac{1}{2}(\mathbf{f}(s)-\mathbf{a})\cdot\mathbf{R}^{-\mathbf{1}'}\cdot(\mathbf{f}(s)-\mathbf{a}) + \frac{1}{2}\mathrm{Tr}\{\mathbf{R}^{-1}\cdot\mathbf{R}'\}\,. \tag{B4}$$

Using $\mathbf{R}^{-\mathbf{1}'} = -s^{-1}\mathbf{R}^{-1}$, $\mathbf{f}'(s) = s^{-1}\mathbf{f}(s)$ and, as above, $\mathbf{R}' = s^{-1}\mathbf{R}$ and $\mathrm{Tr}\{\mathbf{I}\} = N$, Eq. (B4) becomes

$$\frac{d\log\mathbf{P}(\mathbf{a}|s)}{ds} = \frac{1}{2s}\left[N + (\mathbf{f}(s)+\mathbf{a})\cdot\mathbf{R}^{-\mathbf{1}}\cdot(\mathbf{f}(s)-\mathbf{a})\right]\,. \tag{B5}$$

Using Eq. (B1), it is straightforward to show that $\mathbf{R}$ can be recast in the form

$$\mathbf{R}^{-1} = \sigma^{-2}(s)\left[\mathbf{u}_0\mathbf{u}_0\left(\frac{1}{\lambda_0} - \frac{1}{\lambda_1}\right) + \frac{\mathbf{I}}{\lambda_1}\right]\,. \tag{B6}$$

Inserting Eq. (B6) into (B5) and performing a small amount of algebra, we arrive at

$$\frac{d\log\mathbf{P}(\mathbf{a}|s)}{ds} = \frac{N}{2s\sigma^2}\left[\sigma^2 + \frac{\mathbf{f}(s)\cdot\mathbf{f}(s) - \mathbf{a}\cdot\mathbf{a}}{N\lambda_1} + \left(\frac{1}{\lambda_0} - \frac{1}{\lambda_1}\right)\frac{(\mathbf{f}(s)\cdot\mathbf{u}_0)^2 - (\mathbf{a}\cdot\mathbf{u}_0)^2}{N}\right]\,. \tag{B7}$$

To simplify this expression, as in the main text we define $\overline{a} = N^{-1}\sum_i a_i$ and $\overline{\delta a^2} = N^{-1}\sum_i a_i^2 - \overline{a}^2$. Then, using these definitions and Eq. (30) for $\mathbf{u}_0$, we have $\mathbf{f}\cdot\mathbf{f} = Ns^2$, $\mathbf{f}\cdot\mathbf{u}_0 = N^{1/2}s$, $\mathbf{a}\cdot\mathbf{u}_0 = N^{1/2}\overline{a}$, and $\mathbf{a}\cdot\mathbf{a} = N(\overline{\delta a^2} + \overline{a}^2)$. With these relations, Eq. (B7) becomes

$$\frac{d\log\mathbf{P}(\mathbf{a}|s)}{ds} = \frac{N}{2s\sigma^2}\left[\sigma^2 + \frac{s^2 - \overline{a}^2}{\lambda_0} - \frac{\overline{\delta a^2}}{\lambda_1}\right]\,. \tag{B8}$$

When the definitions of λ_0 and λ_1 (Eq. (B2)) are applied to Eq. (B8), that equation becomes identical to the expression in Eq. (28).

Finally, we show that the maximum likelihood estimate, Eq. (32), is unbiased and efficient when N is large. We start with the mean,

$$\langle \hat{s}_{ML}(\mathbf{a})\rangle = \frac{1}{\alpha(1-\rho)}\left[\frac{1}{N}\sum_i \langle a_i^2\rangle - \frac{1}{N^2}\sum_{ij}\langle a_i a_j\rangle\right]$$

where the angle brackets denote an average with respect to the Gaussian probability distribution given in Eq. (19). Because the distribution is Gaussian, the averages are trivial, and we have

$$\langle \hat{s}_{ML}(\mathbf{a})\rangle = \frac{1}{\alpha(1-\rho)}\left[\frac{1}{N}\sum_i R_{ii} - \frac{1}{N^2}\sum_{ij} R_{ij}\right].$$

Using Eq. (26), the first term inside the brackets is $\sigma^2(s)$ and the second term is $\sigma^2(s)(1-\rho)(1-1/N)$. Thus, in the large N limit, the terms inside the brackets reduce to $\sigma^2(s)(1-\rho)$, and

$$\langle \hat{s}_{ML}(\mathbf{a})\rangle = \frac{\sigma^2(s)}{\alpha}.$$

Finally, using $\sigma^2(s) = \alpha s$, we see that $\langle \hat{s}_{ML}(\mathbf{a})\rangle = s$, so the maximum likelihood estimator is unbiased.

The variance of $\hat{s}_{ML}(\mathbf{a})$, denoted $\langle \delta\hat{s}^2_{ML}(\mathbf{a})\rangle \equiv \langle \hat{s}_{ML}(\mathbf{a})^2\rangle - \langle \hat{s}_{ML}(\mathbf{a})\rangle^2$, can be computed in a similar manner,

$$\langle \delta\hat{s}^2_{ML}(\mathbf{a})\rangle = \frac{1}{\alpha^2(1-\rho)^2}\left[\frac{1}{N^2}\sum_{ij}\left[\langle a_i^2 a_j^2\rangle - R_{ii}R_{jj}\right] - \frac{1}{N^3}\sum_{ijk}\langle a_i^2 a_j a_k\rangle + \frac{1}{N^4}\sum_{ijkl}\langle a_i a_j a_k a_l\rangle\right].$$

Using $\langle a_i a_j a_k a_l\rangle = R_{ij}R_{kl} + R_{ik}R_{jl} + R_{il}R_{jk}$ for any i, j, k, and l, this expression becomes

$$\langle \delta\hat{s}^2_{ML}(\mathbf{a})\rangle = \frac{2}{\alpha^2(1-\rho)^2}\left[\frac{1}{N^2}\sum_{ij} R_{ij}^2 - \frac{2}{N^3}\sum_{ijk} R_{ij}R_{ik} + \frac{1}{N^4}\sum_{ijkl} R_{ij}R_{kl}\right].$$

Using Eq. (26), we find that, to lowest order in $1/N$, the terms inside the brackets add to $\sigma^4(s)(1-\rho)^2/N$. Consequently,

$$\langle \delta\hat{s}^2_{ML}(\mathbf{a})\rangle = \frac{2\sigma^4}{N\alpha^2}$$

Finally, using $\sigma^2 = \alpha s$, we arrive at

$$\langle \delta\hat{s}^2_{ML}(\mathbf{a})\rangle = \frac{2s^2}{N},$$

which is the inverse of the Fisher information, Eq. (27), in the limit of large N.

References

Abbott, L.F., & Dayan, P. The effect of correlated variability on the accuracy of a population code. *Neural Comput.* **11**:91–101, 1999.

Braitenberg, V., & Schüz, A. *Anatomy of the Cortex.* Berlin: Spring-Verlag, 1991.

Brunel, N., & Nadal, J.P. Mutual information, Fisher information and population coding. *Neural Computation* **10**:1731–1757, 1998.

Camperi, M., & Wang, X.J. A model of visuospatial working memory in prefrontal cortex: recurrent network and cellular bistability. *J. Comput. Neurosci.* **5**:383–405, 1998.

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

Deneve, S., Latham, P.E., & Pouget, A. Reading population codes: a neural implementation of ideal observers. *Nat. Neurosi.* **2**:740–745, 1999.

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

Droulez, J., & Berthoz, A. A neural network model of sensoritopic maps with predictive short-term memory properties. *Proc. Natl. Acad. Sci.* **88**:9653–9657, 1991.

Gershon, E.D., Wiener, M.C., P.E., Latham., & Richmond, B.J. Coding strategies in monkey V1 and inferior temporal cortices. *J. Neurophysiol.* **79**:1135–1144, 1998.

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

Hopfield, J. J. Neurons with graded responses have collective computational properties like those of two-state neurons. *Proc. Natl. Acad. Sci.* **81**:3088–3092, 1984.

Laing, C.R., & Chow, C.C. Stationary bumps in networks of spiking neurons. *Neural Comput.* **13**:1473–1494, 2001.

Pouget, A., Zhang, K., Deneve, S., & Latham, P.E. Statistically efficient estimation using population coding. *Neural Comput.* **10**:373–401, 1998.

Shadlen, M.N., Britten, K.H., Newsome, W.T., & Movshon, J.A. A computational analysis of the relationship between neuronal and behavioral responses to visual motion. *J. Neurosci* **16**:1486–1510, 1996.

Taube, J.S., Muller, R.U., & Ranck, Jr., J.B. Head-direction cells recorded from the post-subiculum in freely moving rats. I. Description and quantitative analysis. *J. Neurosci.* **10**:420–435, 1990a.

Taube, J.S., Muller, R.U., & Ranck, Jr., J.B. Head-direction cells recorded from the postsubiculum in freely moving rats. II. Effects of environmental manipulations. *J. Neurosci.* **10**:436–447, 1990b.

Tolhurst, D.J., J.A., Movshon, & A.F., Dean. The statistical reliability of signals in single neurons in cat and monkey visual cortex. *Vis. Research* **23**:775–785, 1983.

Wang, X.J. Synaptic reverberation underlying mnemonic persistent activity. *Trends Neurosci.* **24**:455–463, 2001.

Wu, S., & Amari, S. *Neural Implementation of Bayesian Inference in Population Codes.* Advances in Neural Information Processing Systems **14**: MIT Press, Cambridge MA, 2002.

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

Zohary, E., Shadlen, M.N., & Newsome, W.T. Correlated neuronal discharge rate and its implications for psychophysical performance. *Nature* **370**:140–143, 1994.