Reading population codes: a neural implementation of ideal observers

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

[1]Brain and Cognitive Science Department, University of Rochester, Rochester, New York 14627, USA

[2]Department of Neurobiology, University of California, Los Angeles, Los Angeles, California 90095-1763, USA

Correspondence should be addressed to A.P. (alex@bcs.rochester.edu)

Many sensory and motor variables are encoded in the nervous system by the activities of large populations of neurons with bell-shaped tuning curves. Extracting information from these population codes is difficult because of the noise inherent in neuronal responses. In most cases of interest, maximum likelihood (ML) is the best read-out method and would be used by an ideal observer. Using simulations and analysis, we show that a close approximation to ML can be implemented in a biologically plausible model of cortical circuitry. Our results apply to a wide range of nonlinear activation functions, suggesting that cortical areas may, in general, function as ideal observers of activity in preceding areas.

Many neurons in the primary visual cortex are tuned to orientation, and their responses as a function of orientation, known as tuning curves, are typically bell-shaped (**Fig. 1a**). From these tuning curves, one can predict how neurons will respond, on average, to a given orientation. However, neurons are noisy, and a neuron whose average response to a grating at a particular orientation is 20 spikes per second might respond at 18 spikes per second on one trial and 23 spikes per second on the next. This variability is evident when we plot the activity of a population of neurons produced by a grating presented at 90 degrees. If the activity of each neuron is plotted on one trial as a function of its preferred orientation, the resulting pattern looks like a noisy hill (**Fig. 1b**). On another trial, the presentation of the same stimulus would lead to a similar hill, but each cell would respond slightly differently. The task faced by the brain is to estimate, on each trial, the orientation of the grating from this noisy hill.

The task of estimating encoded variables is not specific to orientation; many sensory and motor variables are encoded through the activity of large populations of neurons with bell-shaped tuning curves[1,2]. How does the brain perform this estimation, and how well can it do? Several methods, also known as estimators, have been proposed to 'read out' these noisy hills, that is, to extract the encoded variable or variables based on the observed activity[3]. One such method is the population vector estimator[2], which assigns to each neuron a vector whose length is proportional to the neuron's activity and whose direction corresponds to its preferred orientation, sums all the individual vectors to form a population vector, and then estimates the orientation from the angle of the population vector. This is mathematically equivalent to finding the cosine function that best fits through the pattern of activity and using the position of the peak of the cosine as the estimate of direction[4,5] (**Fig. 1c**). This method has received considerable attention recently, primarily because of its mathematical simplicity. Is it, however, the optimal method? A natural way to answer this question is to present the same orientation repeatedly and compute the mean and variance of the estimate. Recall that the hill of activity changes from trial to trial because of the neuronal noise, even if the orientation stays the same. As a result, the estimate also changes from trial to trial. An optimal estimator should be right on average; that is, the mean estimate should equal the presented orientation. An estimator that is right on average is referred to as unbiased, and it is the only type we consider in this paper. An optimal estimator should also have minimum variance; that is, the estimate should be as similar as possible from trial to trial when the orientation is held fixed. It is possible to derive a lower bound on the variance of the estimator if one knows the structure of the neuronal noise, and an estimator is said to be optimal if its variance is equal to this lower bound.

Although the population vector estimator is unbiased, its variance is typically well above the lower bound dictated by the noise; thus, it is not optimal. The problem with the population vector can be seen in Fig. 1c—the cosine function is not the right template for this particular activity pattern. Instead, one should fit a template derived from the tuning curves of the cells, as illustrated in Fig. 1d. Fitting the optimal template is known as maximum likelihood, and this type of estimator reaches the lower bound dictated by the noise (at least for the case considered here, in which the noise exhibited by each of a large number of neurons is independent of the noise exhibited by the others[4–6]). An estimator that reaches the lower bound is often referred to as an ideal observer, because it performs as well as possible given the noise. Because an ideal observer provides an objective yardstick against which one can measure the performance of an animal, it has been used in several recent studies to relate neuronal variability to behavioral variability[7–9].

A natural question is whether biologically plausible networks can implement a maximum likelihood estimator. We show here that the answer is yes, provided that the level of neuronal noise is independent of firing rate. In this case, recurrent networks of nonlinear units with broad tuning curves—the kind of networks found throughout cortex—can achieve maximum likelihood. When the neuronal noise is more Poisson-like, so that the variance increases with mean activity as observed in cortical neurons[10–12], then the type of network considered in this paper is a close approximation to maximum likelihood.

To illustrate these results, we simulated a recurrent network

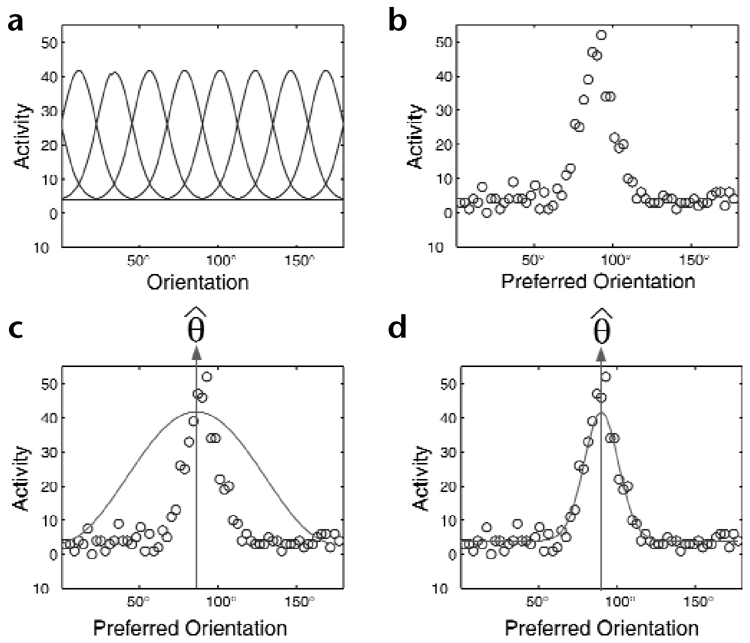

Fig. 1. Methods for reading out population codes. **(a)** Orientation tuning curves for 8 of 64 cells with preferred directions evenly distributed between 0° and 180°. **(b)** A noisy activity pattern in response to a grating presented at 90°. Activities of 64 cells are plotted according to preferred direction. The noise was drawn from a Poisson distribution. **(c)** The population vector reads out this activity curve by fitting it with a cosine function. The peak of the cosine gives an estimate of orientation ($\hat{\theta}$). **(d)** Maximum likelihood fits a template (solid line) derived from the cell tuning curves. When all the tuning curves are identical, this template has the same profile as the tuning curves. Although it is hard to see on this particular trial, the template fit by maximum likelihood **(d)** and the cosine function fit by the population vector **(c)** do not peak at the same location. This is true in general: the estimate obtained from maximum likelihood is very likely to be different from the one obtained with population vector. The maximum likelihood estimate is better—in fact, it is the best of all methods in this context—in the sense that it has minimum variance over trials with fixed orientation.

made up of units whose activation function consisted of a divisive normalization[13–16]. We chose this activation function because recordings in the primary visual cortex show that it provides a good fit to the observed activation function of neurons in V1 (refs. 13–16), and theoretical models reveal that it has several computational advantages[17–19]. We demonstrate numerically that this network can be used to estimate the value of a variable encoded in a population response. We then show that, for neuronal noise independent of firing rate, the variance of the resulting estimator is equal to the minimum achieved by maximum likelihood, whereas for Poisson-like noise, it is very close to the minimum value. We confirmed these results analytically, not only for the special case of networks using divisive normalization, but also for a large class of recurrent networks with other nonlinear activation functions.

Results

Network architecture

We modeled a cortical hypercolumn consisting of a single layer of units with identical spatial receptive fields but differing in preferred orientation and spatial frequency arranged in a two-dimensional array. Two indices described each unit's position, and unit *ij* had preferred orientation θ_i and preferred spatial frequency λ_j. Orientation preferences were mapped along one dimension of the two-dimensional array and spatial frequency preferences along the other, producing a topographic arrangement grouping together units with similar orientation and spatial frequency preferences to resemble arrangements observed in cortex[20].

The network received pooled input ('input activity') from the preceding layer, which represented either another cortical layer or the lateral geniculate nucleus. The resulting activity of the network is referred to as the 'output activity'.

The input to the network depended on the orientation (θ) and spatial frequency (λ) encoded in the previous layer. For a particular value of θ and λ, the total input activity (a_{ij}) was the sum of two terms: the mean input, $f_{ij}(\theta,\lambda)$, and the noise around that mean.

Mean input activity, $f_{ij}(\theta,\lambda)$, followed physiologically realistic tuning curves—bell-shaped profiles with height proportional to contrast (**Fig. 1a**; Methods). Noise was assumed to be independent and to follow a zero-mean Gaussian distribution whose variance was either fixed (σ_{ij}^2 constant) or set to the mean activity [$\sigma_{ij}^2 = f_{ij}(\theta,\lambda)$], which better approximates noise that has been measured in cortex[10–12]. We refer to Gaussian noise with fixed variance as 'flat' and to Gaussian noise with variance equal to the mean as 'proportional' noise.

Units in the network communicated through lateral connections of two types: filtering weights, pooling activity of cells with similar preferred orientations and spatial frequencies (see Methods), as might be accomplished via excitatory lateral connections[21], and divisive normalization weights serving as gain control, possibly mediated by shunting inhibition[16]. Activities in the resulting recurrent network are described by a set of coupled nonlinear evolution equations,

$$u_{ij}(t+1) = \sum_{kl} w_{ij,kl} o_{kl}(t) \tag{1}$$

$$o_{ij}(t+1) = \frac{u_{ij}(t+1)^2}{S + \mu \sum_{kl} u_{kl}(t+1)^2} \tag{2}$$

where $\{w_{ij,kl}\}$ are the filtering weights, $o_{ij}(t)$ is the activity of unit *ij* at time *t*, S is a constant, and μ is the divisive normalization weight.

The initial conditions, $o_{ij}(t = 0)$, for evolution equations 1 and 2 were determined by setting $o_{ij}(0)$ to the activity of the input units, a_{ij}. The a_{ij} were obtained by drawing samples from a Gaussian distribution, as described above. The resulting initial condition, $a_{ij} = f_{ij}(\theta, \lambda)$ + noise, resembled a noisy hill (**Fig. 2**, bottom). Once the initial conditions were chosen, iteration of equations 1 and 2 caused network activity to relax to a smooth hill (**Fig. 2**, top) whose coordinates gave estimates of orientation and spatial frequency.

With distance appropriately defined, this stable activity function can be interpreted as a template, and the relaxation process—the evolution of the network—as a template matching procedure. We asked whether the network provided estimates close to maximum likelihood, or whether it provided a nonoptimal estimate like the population vector (**Fig. 1d** and **c**, respectively).

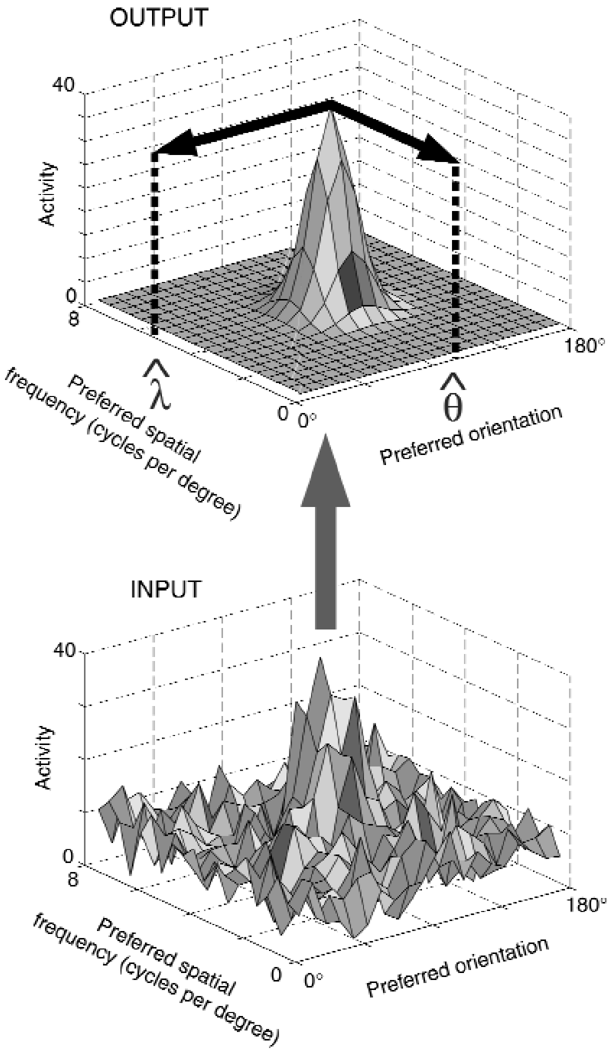

Fig. 2. Activity in the network immediately after initialization and after several iterations. The bottom plot shows a noisy activity function across a two-dimensional array of neurons tuned to orientation and spatial frequency. The top plot corresponds to the stable output curve that appears as a result of the dynamics of the network. The stable hill can be interpreted as a template fit through the noisy hill; the position of its peak can be used to estimate orientation, $\hat{\theta}$, and spatial frequency, $\hat{\lambda}$.

Simulation results

For very low contrast, the activity of all output units decayed to zero, independent of initial conditions. Above some contrast threshold, however, activities converged to a smooth stable peak (**Fig. 2**). The width of this peak was controlled by the width of the filtering weights (the extent of pooling between units of similar orientation and spatial frequency preference) but was independent of input orientation and spatial frequency, θ and λ. On the other hand, the position of the peak depended on θ and λ, and therefore could be used to estimate these quantities, denoted $\hat{\theta}$ and $\hat{\lambda}$, respectively (**Fig. 2**).

As indicated in the introduction, the quality of an estimate can be assessed by looking at its mean and variance over many trials. To compute mean and variance for a particular set of network parameters, we performed 1000 trials with the same input orientation and spatial frequency. On each trial, we generated a different noisy hill of activity, let the network relax to a smooth hill, and then computed the estimates, $\hat{\theta}$ and $\hat{\lambda}$, from its peak.

Estimates of orientation and spatial frequency were unbiased for all network parameters; that is, the mean values of $\hat{\theta}$ and $\hat{\lambda}$ converged to their true values. (This is expected from the symmetry of the network; see Methods.) Network quality was thus determined by the variance of $\hat{\theta}$ and $\hat{\lambda}$ and the covariance between these two quantities, $\langle(\hat{\theta}-\theta)^2\rangle$, $\langle(\hat{\lambda}-\lambda)^2\rangle$ and $\langle(\hat{\theta}-\theta)(\hat{\lambda}-\lambda)\rangle$, respectively, where the angle brackets denote an average over initial conditions, a_{ij}. In our network, we chose input tuning curves, noise and filtering weights that were invariant under interchange of θ and λ (see Methods). Consequently, the variances of $\hat{\theta}$ and $\hat{\lambda}$ were identical and the covariance term, $\langle(\hat{\theta}-\theta)(\hat{\lambda}-\lambda)\rangle$, was zero. Thus, we consider further only the variance of $\hat{\theta}$, $\langle(\hat{\theta}-\theta)^2\rangle$.

To establish that our network is an optimal estimator, we needed to show that the variance of $\hat{\theta}$ was as small as possible, given the noise. For unbiased estimators like the ones considered here, the minimum variance (the Cramér-Rao bound[22]; reached by ML for the noise we consider[4,6]) can be computed directly from the tuning curves and the noise distribution[5,23]. We thus computed, for a range of output tuning-curve widths, the variance of the network estimate and compared it to the variance that would be obtained by ML. To vary the widths of the output tuning curves, we adjusted the spatial extent of the filtering weights (see Methods). We found that the best network performance was within 1.6% of maximum likelihood for flat noise and within 5.1% for proportional noise. By comparison, standard deviations of a population vector estimate on the same data were 278% and 98% of the maximum likelihood variance, respectively (**Fig. 3**). For both kinds of noise, performance was optimal for a particular ratio of the tuning-curve width of the input to that of the output units, as demonstrated by the plot of network performance as a function of the width of the output tuning curve for a fixed input tuning curve (**Fig. 3**).

For flat noise, the network performed best (1.6% above maximum likelihood) when output tuning curves were about 30% sharper than input tuning curves (**Fig. 3**). By contrast, for proportional noise, the best estimate (5.1% above maximum likelihood) was obtained when input and output tuning curves were nearly identical. This near-maximum-likelihood property was preserved over a large range of input tuning curve widths in the sense that the widths of the filtering weights, and thus the output tuning curves, could be adjusted to produce network estimates within a few percent of maximum likelihood for any input width. Likewise, large variations in contrast did not affect network estimates, as long as the contrast exceeded the threshold for activation (**Fig. 4b**).

For both types of noise, network performance degraded smoothly as the widths of the output tuning curves deviated from their optimal values. For width ±10° from the optimum, the network still performed within 10% of maximum likelihood. Furthermore, the network outperformed the population vector over almost the entire range of widths tested (**Fig. 3**).

We also found that the network converged to its asymptotic performance in 2–3 iterations (**Fig. 4a**), although it could take several hundred iterations for the network to stabilize. Interestingly, after 2–3 iterations, the tuning of the output units to contrast (**Fig. 4c**) resembled the sigmoidal tuning curves reported for real neurons[19,24] (**Fig. 4d**). On the other hand, after many network iterations, the contrast tuning curve approached a step

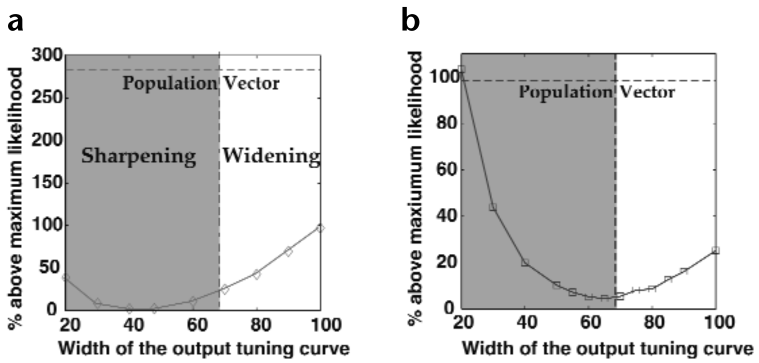

Fig. 3. Network performance compared to maximum likelihood and population vector. Network performance relative to maximum likelihood as a function of the width of the output tuning curves for **(a)** flat noise and **(b)** proportional noise. The width of the input tuning curves was held at 69° (vertical dotted line), and contrast was set to 0.5. Performances of the population vector estimator were applied directly to the input patterns of activity (upper curves). These curves are flat because they only depend on the width of the input tuning curves, which is fixed. For each type of noise, there is an output tuning-curve width for which the network performs very close to ML. The optimal network sharpens the output tuning curves by a factor of 1.4 compared to the input for flat noise, but barely changes the tuning curves for proportional noise.

function, which is not observed experimentally. Therefore, stabilization is not required for optimal performance, and even produces contrast tuning inconsistent with properties of real neurons.

Analysis: general case

Both our network and maximum likelihood fit a template to the initial noisy data and estimate orientation and contrast from its peak. Our numerical results (**Fig. 3**), however, raise three questions. First, why does the network optimize its performance at a specific value of the width of the output tuning curves? Second, why does optimal width depend on the noise distribution? Third, why does the network perform better for flat noise then for proportional noise?

To address these questions, we used a perturbative analysis—we examined the linearized network dynamics near an equilibrium. Our analysis applied to networks that estimate an arbitrary number of variables; however, we state the results for a single variable, orientation. Our analysis was also valid for nonlinear activation functions other than divisive normalization as long as the network activity relaxes to a smooth curve whose peak gives estimates of the encoded variables (**Fig. 2**). For this class of networks, we found that the smallest variance achievable by such an estimator is given by

$$\left\langle (\hat{\theta}-\theta)^2 \right\rangle_{\text{network min}} = [\partial_\theta \mathbf{f} \bullet \mathbf{R}^{-1} \bullet \partial_\theta \mathbf{f}]^{-1}, \tag{3}$$

where $\mathbf{R}^{-1}$ is the inverse of the covariance matrix of the noise, $\delta_\theta \mathbf{f}$ is a vector whose ith component, $\partial_\theta f_i(\theta)$, is the derivative of the *i*th input tuning curve with respect to θ and '•' denotes the standard dot product. In all simulations, a unit's noise was independent of noise in the other units. This implies that the off-diagonal terms in the covariance matrix of the noise, **R**, are zero. (Diagonal terms correspond to the variance of the noise of each unit.) However, our analysis is not restricted to independent noise, but generalizes to arbitrary covariance matrices.

The minimum variance occurs when network parameters are such that the following equation is satisfied. (The steps necessary to derive this equation can be found at http://neurosci.nature.com/supplementary_info/.)

$$\mathbf{v}^\dagger \propto \mathbf{R}^{-1} \bullet \partial_\theta \mathbf{f}, \tag{4}$$

where $\mathbf{v}^\dagger$ is the adjoint eigenvector of the Jacobian of the lin-

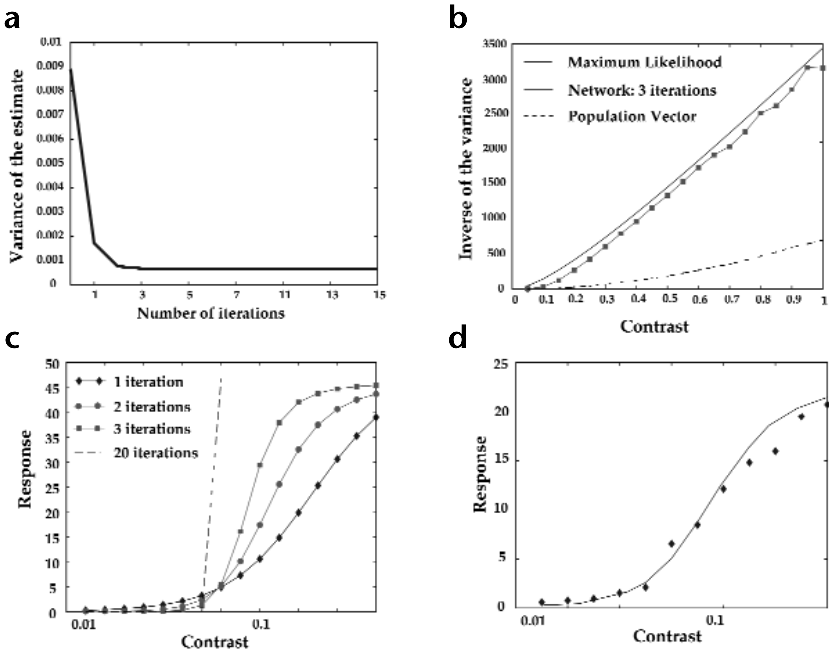

Fig. 4. Temporal evolution of the network estimate and its sensitivity to contrast. **(a)** Variance of the network estimate as a function of the number of iterations. The contrast was 0.5. Two to three iterations are sufficient to reach asymptotic performance. **(b)** Inverse of the variance for the network estimate, $1/\sigma_\theta^2$, as a function of contrast for 3 iterations (squares). We plot $1/\sigma_\theta^2$ instead of σ_θ for visual clarity: σ_θ is so close to zero for contrasts greater than about 0.4 that the different curves are indistinguishable. The solid line indicates the theoretical maximum for $1/\sigma_\theta^2$ (which is achieved by maximum likelihood) and the dotted line corresponds to a population vector estimate. The network stays close to maximum likelihood over a wide range of suprathreshold contrasts. **(c)** Contrast tuning curves as a function of the number of iterations. The output units exhibit realistic contrast tuning curves after 2–3 iterations, but converge to a step function after a large number of iterations. **(d)** Experimental contrast tuning curve (adapted from ref. 24).

earized network with eigenvalue 1. The quantity $\mathbf{v}^\dagger$ depends on network parameters such as the width of the output tuning curves, and, by adjusting those parameters, equation 4 can be satisfied. The resulting network is the best of all networks of the type we have considered, and the width of the output tuning curves for this network is the optimal one. Note also that, as the covariance matrix appears in equation 4, optimal weights and, consequently, the optimal width, depend on the noise distribution. That the minimum variance is achieved only for a particular set of network parameters, and that those parameters depend on the covariance matrix, explain both why performance is optimal for a specific width of the output tuning curves and why the optimal width depends on the noise distribution (**Fig. 3**).

To test our analysis, we computed $\mathbf{v}^\dagger$ using equation 4 and used it to determine the optimal widths for the networks used in the numerical simulations described in the previous section. We found optimal widths of 48° for flat noise and 67° for proportional noise. These were almost exactly the optimal widths for network performance (**Fig. 3**).

To explain why the network performed better for flat noise then for proportional noise, we compared the network variances to maximum likelihood variance for flat and proportional noise. For Gaussian noise with arbitrary covariance matrix, **R**, the maximum likelihood variance (absolute minimum variance, as specified by the Cramér-Rao bound[23]) is given by

$$\left\langle(\hat{\theta}-\theta)^2\right\rangle_{\mathrm{ML\ min}} = [\partial_\theta \mathbf{f} \bullet \mathbf{R}^{-1} \bullet \partial_\theta \mathbf{f} + \frac{1}{2}\mathrm{tr}\{\mathbf{R}^{-1} \bullet \partial_\theta \mathbf{R} \bullet \mathbf{R}^{-1} \bullet \partial_\theta \mathbf{R}\}]^{-1} \quad (5)$$

where tr{} denotes the trace (sum of diagonal elements) of a matrix. The trace term is zero only when **R** is independent of θ; otherwise, it is greater than zero (it is the sum of the squares of real eigenvalues).

When the covariance matrix is independent of orientation, the trace term in equation 5 vanishes, and the variance obtained by the network, equation 3, is identical to maximum likelihood. Thus, flat noise leads to near-optimal performance (1.6% above maximum likelihood). The reason we did not actually acheive maximum likelihood is that the network is optimal for a precise shape of the output tuning curves, and we adjusted only the width. However, the network's close approximation with adjustment of only a single parameter indicates that the architecture we used is extremely robust.

When the covariance matrix depends on orientation, the trace term in equation 5 is greater than zero, and the maximum likelihood bound drops below the variance achieved by the network. For proportional noise, we found that the trace term predicts network performance 4.6% above maximum likelihood. This is consistent with the 5.1% found in our simulations, considering that we adjusted only the width of the output tuning curves.

In conclusion, the networks we considered achieve maximum likelihood only for noise with a covariance matrix that is independent of the stimulus, although close approximations to maximum likelihood can be obtained for stimulus-dependent noise.

Discussion

Using numerical simulations, we have shown that a recurrent network of units with broad tuning curves and divisive normalization can extract variables encoded by a population of noisy neurons. Moreover, with proper tuning of parameters, the network can implement, or come close to implementing, ML—an ideal observer. As our analysis shows, this result extends beyond divisive normalization networks to any recurrent network whose activity relaxes to a smooth curve peaking at a position that depends on the encoded variables (**Fig. 2**). This curve can be viewed as a template whose position is determined by fitting it to the initial activity of the noisy population (**Fig. 1d**). By adjusting network parameters to modify the shape of the template, the network can be made to estimate optimally. Such adjustments could be made in cortical networks with reinforcement learning[25,26].

Because the network recovers a smooth curve from a noisy one, it also acts as a near-optimal nonlinear noise filter by pooling activities through lateral connections. This pooling, however, tends to widen and distort the population activity because it involves cells with different tuning curves. The nonlinear activation function compensates for these distortions, and the combination of pooled activity and the nonlinearity enforces a particular shape for the smooth hill of activity at equilibrium. After 2–3 iterations, the noise is effectively removed, yielding an accurate assessment of the location of the peak and, thus, of the encoded input variables.

That only 2–3 iterations are needed for asymptotic performance (**Fig. 4a**) makes it possible to implement our scheme in a feed-forward network with 2–3 layers (in addition to the input layer). This is because an iteration in time for a recurrent network is equivalent to propagation from one layer to the next in a feed-forward network[27]. Fast convergence also implies that the smooth activity function need not be perfectly stable for the network to approach maximum likelihood; even if the curve decays to zero, noise would be filtered out in the first 2–3 iterations.

The near-equivalence to maximum likelihood is not restricted to networks that encode two variables and exhibit divisive normalization. It generalizes to networks that encode an arbitrary number of variables and exhibit virtually any nonlinear activation function, as long as the network relaxes to a smooth hill of activity whose peak depends on initial activity. Such networks are equivalent to ML when the noise is Gaussian and the variance is input independent, and are close to ML when the variance depends on input. Thus, neurally plausible networks can essentially behave as ideal observers. We emphasize that this result applies to networks of units with tuning curves that are heterogeneous in width and amplitude.

The class of networks investigated here—recurrent networks of nonlinear neurons with broad tuning curves (**Fig. 2**)—are found throughout the cortex. Therefore, our results raise the possibility that each cortical area can be tuned to perform maximum likelihood estimation of variables encoded in the noisy activity from another area. As a result, each area can behave as an ideal observer, suggesting that the ability to optimally process noisy input may be a general property of cortex.

Methods

The input tuning curves, defined to be the mean response to a stimulus of orientation, θ, spatial frequency, λ, and contrast, *C*, were taken to be circular normal functions with a small amount of spontaneous activity, ν,

$$f_{\mathrm{ij}}(\theta, \lambda) = KC\exp\left(\frac{\cos(\theta-\theta_{\mathrm{i}})-1}{\sigma_\theta^2} + \frac{\cos(\lambda-\lambda_{\mathrm{j}})-1}{\sigma_\lambda^2}\right) + \nu \quad (6)$$

where K, σ_θ and σ_λ are constant, and the units are arranged in a $P_\theta \times P_\lambda$ grid: $\theta_{\mathrm{i}} = 2\pi i/P_\theta$, $i = 1,\ldots,P_\theta$ and $\lambda_{\mathrm{j}} = 2\pi j/P_\lambda$, $j = 1,\ldots, P_\lambda$. Note that spatial frequency is treated as a periodic variable to avoid edge effects; this should have a negligible effect on our results as long as we keep λ far from $2\pi n$, n an integer. We used the circular normal function instead of the Gaussian because this function is periodic.

On any given trial, a_{ij}, the input to cortical unit ij is sampled from a Gaussian noise distribution with variance σ^2_{ij},

$$P(a_{\mathrm{ij}} - f_{\mathrm{ij}} \mid \theta,\lambda) = \frac{1}{\sqrt{2\pi\sigma_{\mathrm{ij}}^2}}\exp\left(\frac{[a_{\mathrm{ij}} - f_{\mathrm{ij}}(\theta,\lambda)]^2}{2\pi\sigma_{\mathrm{ij}}^2}\right). \quad (7)$$

These inputs supply the initial conditions: $o_{\mathrm{ij}}(t = 0)$, the initial activity of the network, is set to a_{ij}.

The weights implement a two-dimensional Gaussian filter,

$$w_{\mathrm{ij,kl}} = w_{\mathrm{i-k\,,j-l}} = K_{\mathrm{w}} \exp\left(\frac{\cos[2\pi\,(i-k)/P_\theta]-1}{\delta_{\mathrm{w}}^2} + \frac{\cos[2\pi\,(j-l)/P_\lambda]-1}{\delta_{\mathrm{w}}^2}\right) \quad (8)$$

where K_{w} is constant and $\delta_{\mathrm{w}\theta}$ and $\delta_{\mathrm{w}\lambda}$ control the width of the weights.

The maximum likelihood variance in the estimate of $\hat{\theta}$ (equal to the Cramér-Rao bound for the noise used here) can be computed from the probability distribution given in equation 7; the resulting expression is

$$\left\langle(\hat{\theta}-\theta)^2\right\rangle_{\mathrm{ML}} = \left[\sum_{\mathrm{i}=1}^{P_\theta}\sum_{\mathrm{j}=1}^{P_\lambda}\frac{f_{\mathrm{ij}}'^2}{f_{\mathrm{ij}}} + \frac{1}{2}\left[(\log\sigma_{\mathrm{ij}}^2)'\right]^2\right]^{-1} \quad (9)$$

where the angle brackets denote an average over trials and a prime denotes a derivative with respect to θ. An essentially identical expression exists for the minimum variance in $\hat{\lambda}$; the only difference is that the derivative with respect to θ is replaced by a derivative with respect to λ. Note that for fixed variance, $(\log\sigma_{\mathrm{ij}}^2)' = 0$, and the second term on the right hand side of equation 9 (trace term discussed in analysis) is zero, whereas for noise proportional to the mean activity, $(\log\sigma_{\mathrm{ij}}^2)' = f'_{\mathrm{ij}}/f_{\mathrm{ij}}$.

In all simulations, we used a 20 × 20 array of units ($P_\theta = \mathrm{P}_\lambda = 20$), and the parameters were set to the following values: $K = 74$, $\nu = 3.7$, $\sigma_\theta = \sigma_\lambda = 0.38$, $\mu = 0.002$, $K_{\mathrm{w}} = 1$. For Gaussian noise with fixed variance, $\sigma_{\mathrm{n}}{}^2 = 25$. The parameters $\delta_{\mathrm{w}\theta}$ and $\delta_{\mathrm{w}\lambda}$, which affect the extent of spatial pooling of the filtering weights and thus width of the output tuning curves, were kept equal and were systematically varied within the interval [0.14, 0.718].

Because our noise distribution and filtering weights were symmetric with respect to the interchange of θ and λ (equation 6 with $\sigma_\theta = \sigma_\lambda$ and equation 8 with $\delta_{\mathrm{w}\theta} = \delta_{\mathrm{w}\lambda}$,), by symmetry, the variance in $\hat{\theta}$ is equal to the variance in $\hat{\lambda}$, the covariance, $\langle(\hat{\theta}-\theta)(\hat{\lambda}-\lambda)\rangle$, vanishes and the network is unbiased, $\langle\hat{\theta}\rangle = \theta$, and $\langle\hat{\lambda}\rangle = \lambda$. Thus, in Results we computed only $\langle(\hat{\theta}-\theta)^2\rangle$. We used the standard formula, valid for unbiased estimators,

$$\left\langle(\hat{\theta}-\theta)^2\right\rangle_{\mathrm{network}} = \frac{1}{N-1}\sum_{\mathrm{i}=1}^{N}(\hat{\theta}_{\mathrm{i}}-\theta)^2$$

where N is the number of trials, and $\hat{\theta}$ was determined using a complex estimator[4,5], equivalent to a population vector estimator[2],

$$\hat{\theta} = \mathrm{phase}\left(\sum_{\mathrm{kj}} a_{\mathrm{kj}} e^{\mathrm{i}\theta_{\mathrm{j}}}\right). \quad (10)$$

(Note that in this equation, i is used not as an index but as notation for $\sqrt{-1}$.) Each trial consisted of initializing the network with a noisy input function [$o_{\mathrm{ij}}(t=0) = a_{\mathrm{ij}}$, as described above], iterating equations 1 and 2 and computing $\hat{\theta}$ from equation 10.

Note: a mathematical appendix is available at http://neurosci.nature.com/ supplementary_info/ .

Acknowledgements

We thank Laurent Itti for discussions on divisive normalization.

Received 13 May; Accepted 2 June 1999

1. Maunsell, J. H. & Van Essen, D. C. Functional properties of neurons in middle temporal visual area of the macaque monkey. I. selectivity for stimulus direction, speed, and orientation. *J. Neurophysiol.* **49**, 1127–1147 (1985).
2. Georgopoulos, A. P., Kalaska, J. F., Caminiti, R. & Massey, J. T. On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *J. Neurosci.* **2**, 1527–1537 (1982).
3. Salinas, E. & Abbott, L. Vector reconstruction from firing rate. *J. Comput. Neurosci.* **1**, 89–108 (1994).
4. Seung, H. S. & Sompolinsky, H. Simple model for reading neuronal population codes. *Proc. Natl. Acad. Sci. USA* **90**, 10749–10753 (1993).
5. Pouget, A., Zhang, K., Deneve, S. & Latham, P. E. Statistically efficient estimation using population coding. *Neural Comput.* **10**, 373–401 (1998).
6. Paradiso, M. A. A theory of the use of visual orientation information which exploits the columnar structure of striate cortex. *Biol. Cybern.* **58**, 35–49 (1988).
7. Hawken, M. J. & Parker, A. J. in *Vision: Coding and Efficiency* (ed. Blakemore, C.) 103–116 (Cambridge Univ. Press, Cambridge, UK, 1990).
8. Britten, K. H., Shadlen, M. N., Newsome, W. T. & Movshon, J. A. The analysis of visual motion: A comparison of neuronal and psychophysical performance. *J. Neurosci.* **12**, 4745–4765 (1992).
9. Zohary, E., Shadlen, M. N. & Newsome, W. T. Correlated neuronal discharge rate and its implication for psychophysical performance. *Nature* **370**, 140–143 (1994).
10. Tolhurst, D. J., Movshon, J. A. & Dean, A. D. The statistical reliability of signals in single neurons in cat and monkey visual cortex. *Vision Res.* **23**, 775–785 (1982).
11. Shadlen, M. N. & Newsome, W. T. Noise, neural codes and cortical organization. *Curr. Opin. Neurobiol.* **4**, 569–579 (1994).
12. Gershon, E. D., Weiner, M. C., Latham, P. E. & Richmond, B. J. Coding strategies in monkey V1 and inferior temporal cortices. *J. Neurophysiol.* **79**, 1135–1144 (1998).
13. Nelson, M. E. A mechanism for neuronal gain control by descending pathways. *Neural Comput.* **6**, 242–254 (1994).
14. Heeger, D. J. Normalization of cell responses in cat striate cortex. *Vis. Neurosci.* **9**, 181–197 (1992).
15. Carandini, M. & Heeger, D. J. Summation and division by neurons in primate visual cortex. *Science* **264**, 1333–1336 (1994).
16. Carandini, M., Heeger, D. J. & Movshon, J. A. Linearity and normalization in simple cells of the macaque primary visual cortex. *J. Neurosci.* **17** 3061–3071 (1997).
17. Simoncelli, E. P. & Heeger, D. J. A model of neuronal responses in visual area MT. *Vision Res.* **38**, 743–761 (1998).
18. Li, Z. A neural model of contour integration in the primary visual cortex. *Neural Comput.* **10**, 903–940 (1998).
19. Lee, D. K., Itti, L., Koch, C. & Braun, J. Attention activates winner-take-all competition among visual filters. *Nat. Neurosci.* **2**, 375–381 (1999).
20. Hubel, D. & Wiesel, T. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiol. (Lond.)* **160**, 106–154 (1962).
21. Shadlen, M. N., Britten, K. H., Newsome, W. T. & Movshon, T. A. A computational analysis of the relationship between neuronal and behavioral responses to visual motion. *J. Neurosci.* **16**, 1486–1510 (1996).
22. Papoulis, A. *Probability, Random Variables, and Stochastic Processes* (McGraw-Hill, New York, 1991).
23. Abbott, L. & Dayan, P. The effect of correlated activity on the accuracy of a population code. *Neural Comput.* **11**, 91–101 (1999).
24. Tolhurst, D. J., Movshon, J. A. & Thompson, I. D. The dependence of response amplitude and variance of cat visual cortical neurons on stimulus contrast. *Exp. Brain Res.* **41**, 414–419 (1981).
25. Barto, A. G. in *The Computing Neuron* (eds. Durbin, R., Miall, C. & Mitchison, G.) 73–98 (Addison-Wesley, Wokingham, 1989).
26. Mazzoni, P., Andersen, R. A. & Jordan, M. I. A more biologically plausible learning rule for neural networks. *Proc. Natl. Acad. Sci. USA*, **88**, 4433–4437 (1991)
27. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. in *Parallel Distributed Processing* (eds. Rumelhart, D. E., McClelland, J. L. & PDP Research Group) 318–362 (MIT Press, Cambridge, Massachusetts, 1986).