nature neuroscience

Bayesian inference with probabilistic population codes

Wei Ji Ma[1,3], Jeffrey M Beck[1,3], Peter E Latham[2] & Alexandre Pouget[1]

Recent psychophysical experiments indicate that humans perform near-optimal Bayesian inference in a wide variety of tasks, ranging from cue integration to decision making to motor control. This implies that neurons both represent probability distributions and combine those distributions according to a close approximation to Bayes' rule. At first sight, it would seem that the high variability in the responses of cortical neurons would make it difficult to implement such optimal statistical inference in cortical circuits. We argue that, in fact, this variability implies that populations of neurons automatically represent probability distributions over the stimulus, a type of code we call probabilistic population codes. Moreover, we demonstrate that the Poisson-like variability observed in cortex reduces a broad class of Bayesian inference to simple linear combinations of populations of neural activity. These results hold for arbitrary probability distributions over the stimulus, for tuning curves of arbitrary shape and for realistic neuronal variability.

Virtually all computations performed by the nervous system are subject to uncertainty and taking this into account is critical for making inferences about the outside world. For instance, imagine hiking in a forest and having to jump over a stream. To decide whether or not to jump, you could compute the width of the stream and compare it to your internal estimate of your jumping ability. If, for example, you can jump 2 m and the stream is 1.9 m wide, then you might choose to jump. The problem with this approach, of course, is that you ignored the uncertainty in the sensory and motor estimates. If you can jump 2 ± 0.4 m and the stream is 1.9 ± 0.5 m wide, jumping over it is very risky—and even life-threatening if it is filled with, say, piranhas.

Behavioral studies have confirmed that human observers not only take uncertainty into account in a wide variety of tasks, but do so in a way that is nearly optimal[1–5] (where 'optimal' is used in a Bayesian sense, as defined below). This has two important implications. First, neural circuits must represent probability distributions. For instance, in our example, the width of the stream could be represented in the brain by a Gaussian distribution with mean 1.9 m and s.d. 0.5 m. Second, neural circuits must be able to combine probability distributions nearly optimally, a process known as Bayesian inference.

Although it is clear experimentally that human behavior is nearly Bayes-optimal in a wide variety of tasks, very little is known about the neural basis of this optimality. In particular, we do not know how probability distributions are represented in neuronal responses, nor how neural circuits implement Bayesian inference. At first sight, it would seem that cortical neurons are not well suited to this task, as their responses are highly variable: the spike count of cortical neurons in response to the same sensory variable (such as the direction of motion of a visual stimulus) or motor command varies greatly from trial to trial, typically with Poisson-like statistics[6]. It is critical to realize, however, that variability and uncertainty go hand in hand: if neuronal variability did not exist, that is, if neurons were to fire in exactly the same way every time you saw the same object, then you would always know with certainty what object was presented. Thus, uncertainty about the width of the river in the above example is intimately related to the fact that neurons in the visual cortex do not fire in exactly the same way every time you see a river that is 2 m wide. This variability is partly due to internal noise (like stochastic neurotransmitter release[7]), but the potentially more important component arises from the fact that rivers of the same width can look different, and thus give rise to different neuronal responses, when viewed from different distances or vantage points.

Neural variability, then, is not incompatible with the notion that humans can be Bayes-optimal; on the contrary, as we have just seen, neural variability is expected when subjects experience uncertainty. What it not clear, however, is exactly how optimal inference is achieved given the particular type of noise—Poisson-like variability—observed in the cortex. Here we show that Poisson-like variability makes a broad class of Bayesian inferences particularly easy. Specifically, this variability has a unique property: it allows neurons to represent probability distributions in a format that reduces optimal Bayesian inference to simple linear combinations of neural activities.

RESULTS

Probabilistic population codes (PPC)

Thinking of neurons as encoders of probability distributions is a departure from the more standard view, which is to think of them as encoding the values of variables (like the width of a stream, as in our previous example). However, as several authors have pointed out[8–12], population activity automatically encodes probability distributions. This is because of the variability in neuronal responses, which implies that the population response, $\mathbf{r} \equiv \{r_1, \ldots, r_N\}$, to a stimulus, s, is

[1]Department of Brain and Cognitive Sciences, Meliora Hall, University of Rochester, Rochester, New York 14627, USA. [2]Gatsby Computational Neuroscience Unit, 17 Queen Square, London WC1N 3AR, UK. [3]These authors contributed equally to this work. Correspondence should be addressed to A.P. (alex@bcs.rochester.edu).

Received 16 May; accepted 26 September; published online 22 October 2006; doi:10.1038/nn1790

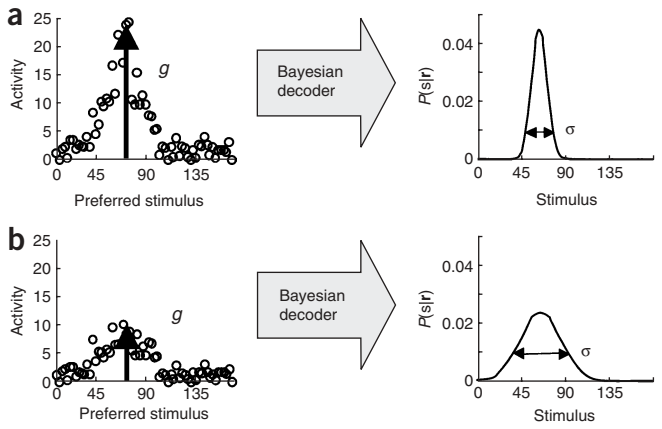

Figure 1 Certainty and gain. (**a**) The population activity, **r**, on the left is the single trial response to a stimulus whose value was 70. All neurons were assumed to have a translated copy of the same generic Gaussian tuning curve to *s*. Neurons are ranked by their preferred stimulus (that is, the stimulus corresponding to the peak of their tuning curve). The plot on the right shows the posterior probability distribution over *s* given **r**, as recovered using Bayes' theorem (equation (1)). When the neural variability follows an independent Poisson distribution (which is the case here), it is easy to show that the gain, *g*, of the population code (its overall amplitude) is inversely proportional to the variance of the posterior distribution, σ^2. (**b**) Decreasing the gain increases the width of the encoded distribution. Note that the population activity in **a** and **b** have the same widths; only their amplitudes are different.

given in terms of a probability distribution, $p(\mathbf{r}|s)$. This response distribution then very naturally encodes the posterior distribution over *s*, $p(s|\mathbf{r})$, through Bayes' theorem[8,9],

$$p(s|\mathbf{r}) \propto p(\mathbf{r}|s)p(s) \qquad (1)$$

To take a specific example, for independent Poisson neural variability, equation (1) becomes,

$$p(s|\mathbf{r}) \propto \prod_i \frac{e^{-f_i(s)} f_i(s)^{r_i}}{r_i!} p(s),$$

where $f_i(s)$ is the tuning curve of neuron *i*. In this case, the posterior distribution, $p(s|\mathbf{r})$, converges to a Gaussian as the number of neurons increases (assuming a flat prior over *s*, an assumption we make now only for convenience, but drop later). The mean of this distribution is close to the stimulus at which the population activity peaks (**Fig. 1**). The variance, σ^2, is also encoded in the population activity—it is inversely proportional to the amplitude of the hill of activity[13–15]. Using *g* (for gain; see **Fig. 1**) to denote the amplitude of the hill of activity, we have $g \propto 1/\sigma^2$. Thus, for independent Poisson neural variability (and, in fact, for many other noise models, as we discuss below), it is possible to encode any Gaussian probability distribution with population activity. This type of parameterization is sometimes known as a product of experts[16].

A simple case study: multisensory integration

Although it is clear that population activity can represent probability distributions, can they carry out any optimal computations—or inference—in ways consistent with human behavior? Before asking how neurons can do this, however, we need to define precisely what we mean by 'optimal'.

In a cue combination task, the goal is to integrate two cues, c_1 and c_2, both of which provide information about the same stimulus, *s*. For instance, *s* could be the spatial location of a stimulus, c_1 could be a visual cue for the location, and c_2 could be an auditory cue. Given observations of c_1 and c_2, and under the assumption that these quantities are independent given *s*, the posterior over *s* is obtained via Bayes' rule, $p(s|c_1, c_2) \propto p(c_1|s)p(c_2|s)p(s)$.

When the prior is flat and the likelihood functions, $p(c_1|s)$ and $p(c_2|s)$, are Gaussian with respect to *s* with means μ_1 and μ_2 and variances σ_1^2 and σ_2^2, respectively, the mean and variance of the posterior, μ_3 and σ_3^2, are given by the following equations (from ref. 17):

$$\mu_3 = \frac{\sigma_2^2}{\sigma_1^2+\sigma_2^2}\mu_1 + \frac{\sigma_1^2}{\sigma_1^2+\sigma_2^2}\mu_2 \qquad (2)$$

$$\frac{1}{\sigma_3^2} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} \qquad (3)$$

Experiments show that humans perform a close approximation to this Bayesian inference—meaning their mean and variance, averaged over many trials, follow equations (2) and (3)—when tested on cue combination[2,3,18,19].

Now that we have a target for optimality—equations (2) and (3)—we can ask how neurons can achieve it. Again we consider two cues, c_1 and c_2, but here we encode them in population activities, $\mathbf{r}_1$ and $\mathbf{r}_2$, respectively, with gains g_1 and g_2 (**Fig. 2**). These probabilistic population codes (PPCs) represent two likelihood functions, $p(\mathbf{r}_1|s)$ and $p(\mathbf{r}_2|s)$. We also assume (for now) that (i) $\mathbf{r}_1$ and $\mathbf{r}_2$ have the same number of neurons, and (ii) two neurons with the same index *i* share the same tuning curve profile; that is, the mean value of both r_{1i} and r_{2i} are proportional to $f_i(s)$. What we now show is that when the prior is flat ($p(s)$ = constant), taking the sum of the two population codes, $\mathbf{r}_1$ and $\mathbf{r}_2$, is equivalent to optimal Bayesian inference. By taking the sum, we mean that we construct a third population, $\mathbf{r}_3 = \mathbf{r}_1 + \mathbf{r}_2$, which is the sum of $\mathbf{r}_1$ and $\mathbf{r}_2$ on a neuron-by-neuron basis: $r_{3i} = r_{1i} + r_{2i}$. If $\mathbf{r}_1$ and $\mathbf{r}_2$ follow Poisson distributions, so will $\mathbf{r}_3$. Therefore, $\mathbf{r}_3$ encodes a likelihood function with variance σ_3^2, where σ_3^2 is inversely proportional to the gain of $\mathbf{r}_3$. Notably, the gain of the third population, denoted g_3, is simply the sum of the gains of the first two: $g_3 = g_1 + g_2$ (**Fig. 2**). Because g_k is proportional to $1/\sigma_k^2$ ($k = 1, 2, 3$), with a constant of proportionality that is independent of *k*, this relationship between the gains implies that $1/\sigma_3^2 = 1/\sigma_1^2 + 1/\sigma_2^2$. This is exactly equation (3). Consequently, the variance of the distribution encoded by $\mathbf{r}_3$ is precisely the variance of the posterior distribution, $p(s|c_1,c_2)$.

General theory and the exponential family of distributions

Does the strategy of adding population codes lead to optimal inference under more general conditions, such as non-Gaussian distributions over the stimulus and non-Poisson neural variability? In general, the sum, $\mathbf{r}_3 = \mathbf{r}_1 + \mathbf{r}_2$, is Bayes-optimal if $p(s|\mathbf{r}_3)$ is equal to $p(s|\mathbf{r}_1)p(s|\mathbf{r}_2)$ or, equivalently, if $p(\mathbf{r}_1 + \mathbf{r}_2|s) \propto p(\mathbf{r}_1|s)p(\mathbf{r}_2|s)$. This is not the case for most probability distributions (such as additive Gaussian noise with fixed variance; see **Supplementary Note** online) but, as shown in **Supplementary Note**, the sum is Bayes-optimal if all distributions are what we call Poisson-like; that is, distributions of the form

$$p(\mathbf{r}_k|s, g_k) = \phi_k(\mathbf{r}_k, g_k)\exp(\mathbf{h}^{\mathrm{T}}(s)\mathbf{r}_k) \qquad (4)$$

where the index *k* can take the value, 1, 2 or 3, and the kernel $\mathbf{h}(s)$ obeys

$$\mathbf{h}'(s) = \sum\nolimits_k^{-1}(s, g_k)\mathbf{f}_k'(s, g_k) \qquad (5)$$

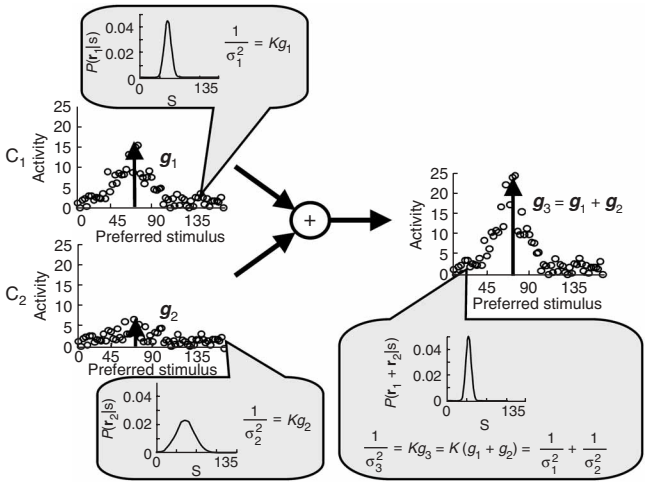

Figure 2 Inference with probabilistic population codes for Gaussian probability distributions and Poisson variability. The left plots correspond to population codes for two cues, c_1 and c_2, related to the same variable s. Each of these encodes a probability distribution with a variance inversely proportional to the gains, g_1 and g_2, of the population codes (K is a constant depending on the width of the tuning curve and the number of neurons). Adding these two population codes leads to the output population activity shown on the right. This output also encodes a probability distribution with a variance inversely proportional to the gain. Because the gain of this code is $g_1 + g_2$, and g_1 and g_2 are inversely proportional to ${\sigma_1}^2$ and ${\sigma_2}^2$, respectively, the inverse variance of the output population code is the sum of the inverse variances associated with c_1 and c_2. This is precisely the variance expected from an optimal Bayesian inference (equation (3)). In other words, taking the sum of two population codes is equivalent to taking the product of their encoded distributions.

Σk is the covariance matrix of $\mathbf{r}_k$, and $\mathbf{f}'_k$ is the derivative of the tuning curves. In the case of independent Poisson noise, identically shaped tuning curves, $\mathbf{f}(s)$, in the two populations, and different gains, it turns out that $\mathbf{h}(s) = \log \mathbf{f}(s)$, and $\phi_k(\mathbf{r}_k, g_k) = \exp(-cg_k)\Pi_i \exp(r_{ki} \log g_k)/r_{ki}!$ with c a constant.

As indicated by equation (5), for addition of population codes to be optimal, the right-hand side of this equation must be independent of both g_k and k. As $\mathbf{f}'$ is clearly proportional to the gain, for the first condition to be satisfied $\Sigma_k(s,g_k)$ must also be proportional to the gain. This is exactly what is observed in cortex, where it is found that the covariance matrix is proportional to the mean spike count[6,20], which in turn is proportional to the gain. This applies in particular to independent Poisson noise, for which the variance is equal to the mean, but is not limited to that distribution. For instance, we do not require that the neurons be independent (that is, that $\Sigma_k(s,g_k)$ be diagonal). Also, although we need the covariance to be proportional to the mean, the constant of proportionality does not have to be 1. This is important because how the diagonal elements of the covariance matrix scale with g determines the Fano factor, and values reported in cortex for this scaling are not always 1 (as would be the case for purely Poisson neurons) but instead range from 0.3 to 1.8 (refs. 6,20).

The second condition, that $\mathbf{h}'(s)$ must be independent of k, requires that $\mathbf{h}(s)$ be identical, up to an additive constant, in all input layers. This occurs, for instance, when the input tuning curves are identical and the noise is independent and Poisson. When the $\mathbf{h}(s)$'s are not the same, so that $\mathbf{h}(s) \rightarrow \mathbf{h}_k(s)$, addition is no longer optimal, but optimality can still be achieved with linear combinations of activity, that is, a dependence of the form $\mathbf{r}_3 = \mathbf{A}_1{}^T\mathbf{r}_1 + \mathbf{A}_2{}^T\mathbf{r}_2$ (provided the functions of s that make up the components of the $\mathbf{h}_k(s)$'s are drawn from a common basis set; details in **Supplementary Note**). Therefore, even if the tuning curves and covariance structures are completely different in the two population codes—for instance, Gaussian tuning curves in one and sigmoidal curves in the other—optimal Bayesian inference can be achieved with linear combinations of population codes.

To illustrate this point, we show a simulation (**Fig. 3**) in which there are three input layers in which the tuning curves are Gaussian, sigmoidal increasing and sigmoidal decreasing, and the parameters of the tuning curves, such as the widths, slopes, amplitude and baseline activity, vary within each layer (that is, the tuning curves are not perfectly translation invariant). As predicted, with an appropriate choice of the matrices $\mathbf{A}_1$, $\mathbf{A}_2$ and $\mathbf{A}_3$ (**Supplementary Note**), a linear combination of the input activities, $\mathbf{r}_3 = \mathbf{A}_1{}^T\mathbf{r}_1 + \mathbf{A}_2{}^T\mathbf{r}_2 + \mathbf{A}_3{}^T\mathbf{r}_3$, is optimal.

Another important property of equation (4) worth emphasizing is that it imposes no constraint on the shape of the probability distribution with respect to s, so long as $\mathbf{h}(s)$ forms a basis set. In other words, our scheme works for a large class of distributions over s, not just Gaussian distributions.

Finally, it is easy to incorporate prior distributions. We encode the desired prior in a population code (using equation (1)) and add that to

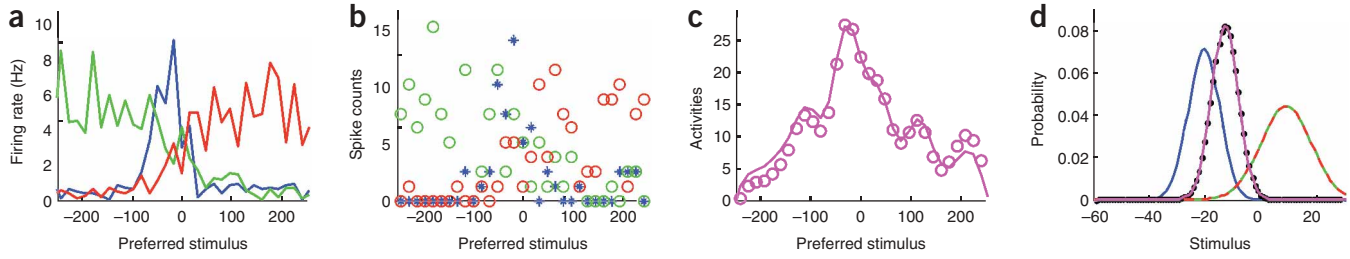

Figure 3 Inference with non–translation invariant Gaussian and sigmoidal tuning curves. (**a**) Mean activity in the three input layers. Blue curves, input layer with Gaussian tuning curves. Red curves, input layers with sigmoidal tuning curves with positive slopes. Green curves, input layers with sigmoidal tuning curves with negative slopes. The noise in the curves is due to variability in the baseline, widths, slopes and amplitudes of the tuning curves and to the fact that the tuning curves are not equally spaced along the stimulus axis. (**b**) Activity in the three input layers on a given trial. These activities were sampled from Poisson distributions with means as in **a.** Color legend as in **a**. (**c**) Solid lines, mean activity in the output layer. Circles, output activity on a given trial, obtained by a linear combination of the input activities shown in **b**. (**d**) Blue curves, probability distribution encoded by the blue stars in **b** (input layer with Gaussian tuning curves). Red-green curve, probability distribution encoded by the red and green circles in **b** (the two input layers with sigmoidal tuning curves). Magenta curve, probability distribution encoded by the activity shown in **c** (magenta circles). Black dots, probability distribution obtained with Bayes rule (that is, the product of the blue and red-green curves appropriately normalized). The fact that the black dots are perfectly lined up with the magenta curve demonstrates that the output activity shown in **c** encodes the probability distribution expected from Bayes rule.

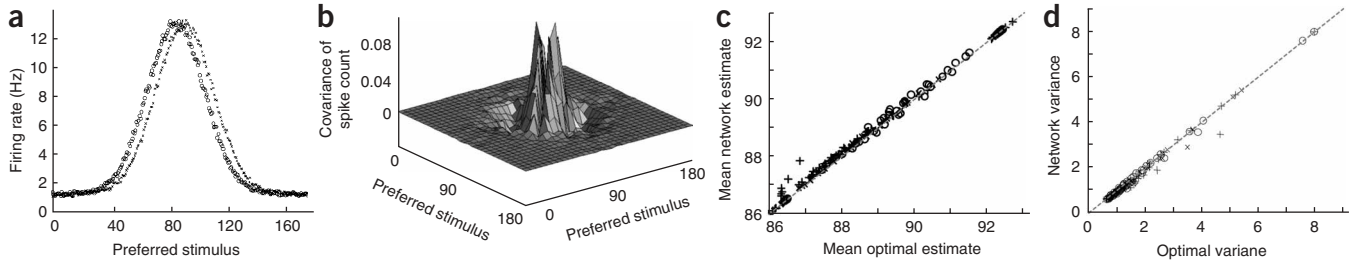

Figure 4 Near-optimal inference with a two-layer network of integrate-and-fire neurons similar in spirit to the network shown in **Figure 2**. The network consisted of two input layers that sent feedforward connections to the output layer. The output layer contained both excitatory and inhibitory neurons and was recurrently connected; the input layers were purely excitatory and had no recurrent connections. (**a**) Average activity in the two input layers for identical gains. The positions of the two hills differ on average by 6 to simulate a cue conflict (the units are arbitrary). (**b**) Covariance matrix of the spike count in the output layer. The diagonal terms (the variances) were set to zero in this plot because they swamp the signal from the covariance (and are uninformative). Because of lateral connections and correlations in the input, output units with similar tuning are correlated. (**c**) Mean of the probability distribution encoded in the output layer when inputs 1 and 2 are presented together (mean network estimate) versus mean predicted by an optimal Bayesian estimator (mean optimal estimate, obtained from equation (2); see Methods). Each point corresponds to the means averaged over 1,008 trials for a particular combination of gains in the input layers. The symbols correspond to different types of input. Circles, same tuning curves and same covariance matrix for both inputs. Plus signs, same tuning curves and different covariance matrices. Crosses, different tuning curves and different covariance matrices (see Methods). (**d**) Same as in **c** but for the variance. The optimal variance is obtained from equation (3). In both **c** and **d**, the data lie near the line with slope = 1 (diagonal dashed line), indicating that the network performs a close approximation to Bayesian inference.

the population code representing the likelihood function. This predicts that in an area encoding a prior, neurons should fire before the start of the trial. Moreover, if the prior at a particular spatial location is increased, all neurons with receptive fields at that location should fire more strongly (their gain should increase). This is indeed what has been reported in area LIP (ref. 21) and in the superior colliculus[22]. One problem with this approach is that the encoded prior will vary from trial to trial due to the Poisson variability. Whether such a variability in the encoded prior is observed in human subjects is not presently known[5].

Simulations with integrate-and-fire neurons

So far, our results rely on the assumption that neurons can compute linear combinations of spike counts, which is only an approximation of what actual neurons do. Neurons are nonlinear devices that integrate their inputs and fire spikes. To determine whether it is possible to perform near-optimal Bayesian inference with realistic neurons, we simulated a network like the one shown in **Figure 2** but with conductance-based integrate-and-fire neurons. The network consisted of two input layers, denoted 1 and 2, that sent feedforward connections to the output layer, denoted layer 3. The activity in the input layers formed noisy hills with the peak in layer 1 centered at $s = 86.5$ and the peak in layer 2 at $s = 92.5$ (**Fig. 4a** shows the mean input activities in both layers). We used different values of the positions of the input hills to simulate cue conflict, as is commonly done in psychophysics experiments. The amplitude of each input hill was determined by the reliability of the cue it encoded: the higher the reliability, the higher the hill, as expected for a PPC with Poisson-like variability (**Fig. 1**). The activity in the output layer also formed a hill, which was decoded using a locally optimal linear estimator[23]. Parameters were chosen such that the spike counts of the output neurons exhibit realistic Fano factors (Fano factors ranging from 0.76 to 1.0). As we have seen, Fano factors that are independent of the gain are one of the key properties required for optimality. Additionally, the conductances of the feedforward and lateral connections were adjusted to ensure that the average firing rates of the output neurons were approximately linear functions of the average firing rates of the input neurons. Because of the convergent feedforward connectivity and the cortical connections, output units with similar tuning ended up being correlated (**Fig. 4b**; additional details of the model in Methods and **Supplementary Note**).

The goal of these simulations was to assess whether the mean and variance of the distributions encoded in the output layer are consistent with optimal Bayesian inference (equations (2) and (3)). To simulate psychophysical experiments, we first presented one cue at a time; that is, we activated either layer 1 or layer 2, but not both. We systematically varied the certainty of the cue by changing the value of the gain of the activated input layer. For each gain, we computed the mean and variance of the distribution encoded in the output layer when only one cue was presented. These were denoted μ_1 and σ_1^2, respectively, when only input 1 was active, and μ_2 and σ_2^2 when only input 2 was active. We then presented both cues together, which gave us μ_3 and σ_3^2, the mean and variance of the distribution encoded in the output layer when both cues are presented simultaneously. To test whether the network was Bayes-optimal, we plotted (**Fig. 4c**) μ_3 against

$$\mu_1 \frac{\sigma_2^2}{\sigma_1^2+\sigma_2^2} + \mu_2 \frac{\sigma_1^2}{\sigma_1^2+\sigma_2^2}$$

(equation (2)), and (**Fig. 4d**) σ_3^2 against

$$\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2}$$

(equation (3)) over a wide range of values of certainty for the two cues (corresponding to gains of the two input hills). If the network is performing a close approximation to Bayesian inference, the data should lie close to a line with slope 1 and intercept 0.

It is clear (**Fig. 4c**,**d**) that the network is indeed nearly optimal on average for all combinations of gains tested, as has been found in human data[1–4]. This result holds even when the input layers use different sets of tuning curves and different patterns of correlations (**Fig. 4d**), thus confirming the applicability of our analytical findings. Therefore, linear combinations of probabilistic population codes are Bayes-optimal for Poisson-like noise.

Experimental predictions

These ideas can be tested experimentally in different domains, as Bayesian inference seems to be involved in many sensory, motor and

cognitive tasks. We now consider three specific predictions that can be tested with single- or multiunit recordings:

First, we predict that if an animal exhibits Bayes-optimal behavior in a cue combination task, and the variability of multisensory neurons is Poisson-like (as defined by equation (4)), one should find that the responses of these neurons to multisensory inputs should be the sum of the responses to the unisensory inputs. This prediction seems at odds with the main result that has been emphasized in the literature, namely, superadditivity. Superadditivity refers to a multimodal response that is greater than the value predicted by the sum of the unimodal responses[24]. Recent studies[25,26], however, have shown that the vast majority of multisensory neurons exhibit additive responses in anesthetized animals. What is needed now to test our hypothesis is similar data in awake animals performing optimal multisensory integration.

Our second prediction concerns decision making, more specifically, binary decision making (as in ref. 27). In these experiments, animals are trained to decide between two saccades (in opposite directions) given the direction of motion in a random-dot kinematogram. In a Bayesian framework, the first step in decision making is to compute the posterior distribution over the decision variable, s, given the available evidence. In this particular task, the evidence takes the form of a population pattern of activity from motion-sensitive neurons, probably from area MT. Denoting $\mathbf{r_t}^{\mathrm{MT}}$ to be the population pattern of activity in area MT at time t, the posterior distribution over s since the beginning of the trial can be computed recursively using Bayes' rule,

$$p(s|\mathbf{r}_t^{\mathrm{MT}}, \ldots, \mathbf{r}_1^{\mathrm{MT}}) \propto p(\mathbf{r}_t^{\mathrm{MT}}|s)p(s|\mathbf{r}_{t-1}^{\mathrm{MT}}, \ldots, \mathbf{r}_1^{\mathrm{MT}}) \qquad (6)$$

Note that this inference involves, as with cue combination, multiplying probability distributions. Thus, if we represent the posterior distribution at time $t-1$, $p(s|\mathbf{r_{t-1}}^{\mathrm{MT}}, \ldots, \mathbf{r}_1^{\mathrm{MT}})$, in a probabilistic population code (say in area LIP) then, upon observing a new pattern of activity from MT, we can simply add this pattern to LIP activity. In other words, LIP neurons will automatically implement equation (6) simply by accumulating activity coming from MT. This predicts that LIP neurons behave like neural integrators of MT activity, which is consistent with what a previous study has found[28]. In addition, this predicts that the profile of tuning curves of LIP neurons over time should remain identical; only the gain and the baseline should change. This prediction has yet to be tested.

Third, our theory makes a general prediction regarding population codes in the cortex and their relation to behavioral performance. If a stimulus parameter is varied in such a way that the subject is less certain about the stimulus, the probability distribution over stimuli recovered by equation (1) (as assumed by PPCs) should reflect that uncertainty (in the case of a Gaussian posterior, for example, the distribution should get wider). This prediction has been verified in two cases in which it has been tested experimentally: motion coherence[29,30] and contrast[31,32].

This last prediction may not be valid in all areas of the brain. For instance, it is conceivable that motor neurons encode a single action, not a full distribution over possible actions (as would be the case for any network computing maximum-likelihood estimates; see for instance ref. 33). If that were the case, applying Bayes' rule to the activity of motor neurons would not return a posterior distribution that reflects the subject's certainty about this action being correct.

DISCUSSION

We have argued that the nervous system may use probabilistic population codes (PPCs) to encode probability distributions over variables in the outside world (such as the orientation of a bar or the speed of a moving object). This notion is not entirely new. Several groups[8–10,34] have pointed out that probability distributions can be recovered from neuronal responses through equation (1). However, we go beyond this observation in two ways. First, we show that Bayesian inference—a nontrivial and critically important computation in the brain—is particularly simple when using PPCs with Poisson-like variability. Second, we do not merely propose that population activity encodes distributions—this part is always true, in the sense that equation (1) can always be applied to a population code. The new aspect of our claim is that the probability distributions encoded in some areas of the cortex reflect the uncertainty about the stimulus, whereas in other areas they do not (in particular in motor areas, as discussed at the end of the previous section).

Other types of neural codes beside PPCs have been proposed for encoding probability distributions that reflect the observer's uncertainty[3,11,12,28,35–43]. In most of these, however, the Poisson-like variability is either ignored altogether or treated as a nuisance factor that corrupts the codes. In only one of them was Poisson-like variability taken into account and, in fact, used to compute explicitly the log likelihood of the stimulus[43], presumably because log-likelihood representations have the advantage that they turn products of probability distributions into sums[28,35,41–43]. A crucial point of our work, however, is to show that, when the neural variability belongs to the exponential family with linear sufficient statistics (as is the case in ref. 43), products turn into sums without any need for an explicit computation of the log likelihood. This is important because there are a number of problems associated with the explicit computation of the log likelihood. For instance, the model described in ref. 43 is limited to independent Poisson noise, unimodal probability distributions and winner-take-all readout. This is problematic, as the noise in the cortex is correlated, probability distributions can have multiple peaks (for example, the Necker cube), and winner-take-all is a particularly inefficient read-out technique. More importantly, the log-likelihood approach runs into severe computational limitations when applied to many Bayesian inference problems such as ones involved in Kalman filters[41]. By contrast, the PPC approach works for correlated Poisson-like noise and a wide variety of tuning curves, the latter being crucial for optimal nonlinear computations[34,44]. Our framework can also be readily extended to Kalman filters (J. Beck, W.J. Ma, P.E. Latham & A. Pouget, *Cosyne Abstr.* 47, 2006). Finally, it has the advantage of being recursive: with PPCs, all cortical areas use the same scheme to represent probability distributions (as opposed to log-likelihood schemes, in which some areas use the standard tuning curve plus noise model while others explicitly compute log likelihood). Recursive schemes map very naturally onto the stereotyped nature of cortical microcircuitry[45].

One limitation of our scheme, and of any scheme that reduces Bayesian inference to addition of activities, is that neural activities are likely to saturate when sequential inferences are required. To circumvent this problem, a nonlinearity is needed to keep neurons within their dynamical range. A nonlinearity like divisive normalization[46,47] would be ideal because it is near linear for low firing rates, where uncertainty is large and thus there is much to be gained from performing exact inference, and saturating at high firing rates, where uncertainty is small and there is little to be gained from exact inference (see **Fig. 1**).

In conclusion, our notion of probabilistic population codes offers a new perspective on the role of Poisson-like variability. The presence of such variability throughout the cortex suggests that the entire cortex represents probability distributions, not just estimates, which is precisely what would be expected from a Bayesian perspective (see also ref. 48 for related ideas). We propose that these distributions are

collapsed onto estimates only when decisions are needed, a process that may take place in motor cortex or in subcortical structures. Notably, our previous work shows that attractor dynamics in these decision networks could perform this step optimally by computing maximum *a posteriori* estimates[33].

METHODS

Spiking neuron simulations. A detailed description of the network is given in **Supplementary Note**; here we give a brief overview. The network we simulated is a variation of the model reported in ref. 23. It contains two input layers and one output layer. Each input layer consists of 1,008 excitatory neurons. These neurons exhibit bell-shaped tuning curves with preferred stimuli evenly distributed over the range [0,180] (stimulus units are arbitrary). The input spike trains are near-Poisson with mean rates determined by the tuning curves. The output layer contains 1,260 conductance-based integrate-and-fire neurons, of which 1,008 are excitatory and 252 inhibitory. Each of those neurons receives connections from the input neurons. The conductances associated with the input connections follow a Gaussian profile centered on the preferred stimulus of each input unit.

The connectivity in the output layer is chosen so that the output units exhibit Gaussian tuning curves whose widths are close to the widths of the convolved input (that is, the width after the input tuning curves have been convolved with the feedforward weights). The balance of excitation and inhibition in the output layer was adjusted to produce high Fano factors (0.7–1.0), within the range observed *in vivo*[6,20]. Finally, additional tuning of connection strengths was performed to ensure that the firing rates of the output neurons were approximately linear functions of the firing rates of the input neurons.

We simulated three different networks. In the first (blue dots in **Fig. 4c,d**), for both populations the widths of the input tuning curves were 20 and the widths of the feedforward weights were 15. In the second (red dots in **Fig. 4c,d**), the widths of the input tuning curves were 15 and 25, and the widths of the corresponding feedforward weights were 20 and 10. The effective inputs for the two populations had identical tuning curves (with a width of 35) but, unlike in the first network, different covariance matrices. Finally, in the third network (green dots in **Fig. 4c,d**), the widths of the input tuning curves were 15 and 25, and the width of the feedforward weights was 15. In this case both the tuning curves and the covariance matrices of the effective inputs were different.

Estimating the mean and variance of the encoded distribution. To determine whether this network is Bayes-optimal, we need to estimate the mean and variance of the probability distribution encoded in the output layer. In principle, all we need is $p(\mathbf{r}|s)$, equation (1). The response, however, is 1,008-dimensional. Estimating a distribution in 1,008 dimensions requires an unreasonably large amount of data—more than we could collect in several billion years. We thus used a different approach. The variances can be estimated using a locally optimal linear estimator, as described in ref. 23. For the mean, we fit a Gaussian to the output spike count on every trial and used the position of the Gaussian as an estimate of the mean of the encoded distribution. The best fit was found by minimizing the Euclidean distance between the Gaussian and the spike counts. The points in **Figure 4c,d** are the means and variances averaged over 1,008 trials (details in **Supplementary Note**).

Note: Supplementary information is available on the Nature Neuroscience website.

ACKNOWLEDGMENTS

W.J.M. was supported by a grant from the Schmitt foundation, J.B. by grants from the US National Institutes of Health (NEI 5 T32 MH019942) and the National Institute of Mental Health (T32 MH19942), P.E.L. by the Gatsby Charitable Foundation and National Institute of Mental Health (grant R01 MH62447) and A.P. by the National Science Foundation (grants BCS0346785 and BCS0446730) and by a research grant from the James S. McDonnell Foundation.

COMPETING INTERESTS STATEMENT

The authors declare that they have no competing financial interests.

Published online at http://www.nature.com/natureneuroscience
Reprints and permissions information is available online at http://npg.nature.com/reprintsandpermissions/

1. Knill, D.C. & Richards, W. *Perception as Bayesian Inference* (Cambridge Univ. Press, New York, 1996).
2. van Beers, R.J., Sittig, A.C. & Gon, J.J. Integration of proprioceptive and visual position-information: an experimentally supported model. *J. Neurophysiol.* **81**, 1355–1364 (1999).
3. Ernst, M.O. & Banks, M.S. Humans integrate visual and haptic information in a statistically optimal fashion. *Nature* **415**, 429–433 (2002).
4. Kording, K.P. & Wolpert, D.M. Bayesian integration in sensorimotor learning. *Nature* **427**, 244–247 (2004).
5. Stocker, A.A. & Simoncelli, E.P. Noise characteristics and prior expectations in human visual speed perception. *Nat. Neurosci.* **9**, 578–585 (2006).
6. Tolhurst, D., Movshon, J. & Dean, A. The statistical reliability of signals in single neurons in cat and monkey visual cortex. *Vision Res.* **23**, 775–785 (1982).
7. Stevens, C.F. Neurotransmitter release at central synapses. *Neuron* **40**, 381–388 (2003).
8. Foldiak, P. in *Computation and Neural Systems* (eds. Eeckman, F. & Bower, J.) 55–60 (Kluwer Academic Publishers, Norwell, Massachusetts, 1993).
9. Sanger, T. Probability density estimation for the interpretation of neural population codes. *J. Neurophysiol.* **76**, 2790–2793 (1996).
10. Salinas, E. & Abbot, L. Vector reconstruction from firing rate. *J. Comput. Neurosci.* **1**, 89–107 (1994).
11. Zemel, R., Dayan, P. & Pouget, A. Probabilistic interpretation of population code. *Neural Comput.* **10**, 403–430 (1998).
12. Anderson, C. in *Computational Intelligence Imitating Life* (eds. Zurada, J.M., Marks, R.J., II & Robinson, C.J.) 213–222 (IEEE Press, New York, 1994).
13. Seung, H. & Sompolinsky, H. Simple model for reading neuronal population codes. *Proc. Natl. Acad. Sci. USA* **90**, 10749–10753 (1993).
14. Snippe, H.P. Parameter extraction from population codes: a critical assessment. *Neural Comput.* **8**, 511–529 (1996).
15. Wu, S., Nakahara, H. & Amari, S. Population coding with correlation and an unfaithful model. *Neural Comput.* **13**, 775–797 (2001).
16. Hinton, G.E. in *Proceedings of the Ninth International Conference on Artificial Neural Network* 1–6 (IEEE, London, England, 1999).
17. Clark, J.J. & Yuille, A.L. *Data Fusion for Sensory Information Processing Systems* (Kluwer Academic, Boston, 1990).
18. Knill, D.C. Discrimination of planar surface slant from texture: human and ideal observers compared. *Vision Res.* **38**, 1683–1711 (1998).
19. Gepshtein, S. & Banks, M.S. Viewing geometry determines how vision and haptics combine in size perception. *Curr. Biol.* **13**, 483–488 (2003).
20. Gur, M. & Snodderly, D.M. High response reliability of neurons in primary visual cortex (V1) of alert, trained monkeys. *Cereb Cortex* **16**, 888–895 (2006).
21. Platt, M.L. & Glimcher, P.W. Neural correlates of decision variables in parietal cortex. *Nature* **400**, 233–238 (1999).
22. Basso, M.A. & Wurtz, R.H. Modulation of neuronal activity by target uncertainty. *Nature* **389**, 66–69 (1997).
23. Series, P., Latham, P. & Pouget, A. Tuning curve sharpening for orientation selectivity: coding efficiency and the impact of correlations. *Nat. Neurosci.* **7**, 1129–1135 (2004).
24. Stein, B.E. & Meredith, M.A. *The Merging of the Senses* (MIT Press, Cambridge, Massachusetts, 1993).
25. Stanford, T.R., Quessy, S. & Stein, B.E. Evaluating the operations underlying multisensory integration in the cat superior colliculus. *J. Neurosci.* **25**, 6499–6508 (2005).
26. Perrault, T.J., Jr., Vaughan, J.W., Stein, B.E. & Wallace, M.T. Superior colliculus neurons use distinct operational modes in the integration of multisensory stimuli. *J. Neurophysiol.* **93**, 2575–2586 (2005).
27. Shadlen, M.N. & Newsome, W.T. Neural basis of a perceptual decision in the parietal cortex (area LIP) of the rhesus monkey. *J. Neurophysiol.* **86**, 1916–1936 (2001).
28. Gold, J.I. & Shadlen, M.N. Neural computations that underlie decisions about sensory stimuli. *Trends Cogn. Sci.* **5**, 10–16 (2001).
29. Britten, K.H., Shadlen, M.N., Newsome, W.T. & Movshon, J.A. Responses of neurons in macaque MT to stochastic motion signals. *Vis. Neurosci.* **10**, 1157–1169 (1993).
30. Weiss, Y. & Fleet, D.J. in *Probabilistic Models of the Brain: Perception and Neural Function* (eds. Rao, R., Olshausen, B. & Lewicki, M.S.) 77–96 (MIT Press, Cambridge, Massachusetts, 2002).
31. Anderson, J.S., Lampl, I., Gillespie, D.C. & Ferster, D. The contribution of noise to contrast invariance of orientation tuning in cat visual cortex. *Science* **290**, 1968–1972 (2000).
32. Sclar, G. & Freeman, R. Orientation selectivity in the cat's striate cortex is invariant with stimulus contrast. *Exp. Brain Res.* **46**, 457–461 (1982).
33. Deneve, S., Latham, P. & Pouget, A. Reading population codes: a neural implementation of ideal observers. *Nat. Neurosci.* **2**, 740–745 (1999).
34. Deneve, S., Latham, P. & Pouget, A. Efficient computation and cue integration with noisy population codes. *Nat. Neurosci.* **4**, 826–831 (2001).
35. Barlow, H.B. Pattern recognition and the responses of sensory neurons. *Ann. NY Acad. Sci.* **156**, 872–881 (1969).

36. Simoncelli, E., Adelson, E. & Heeger, D. in *Proceedings 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 310–315 (1991).
37. Koechlin, E., Anton, J.L. & Burnod, Y. Bayesian inference in populations of cortical neurons: a model of motion integration and segmentation in area MT. *Biol. Cybern.* **80**, 25–44 (1999).
38. Anastasio, T.J., Patton, P.E. & Belkacem-Boussaid, K. Using Bayes' rule to model multisensory enhancement in the superior colliculus. *Neural Comput.* **12**, 1165–1187 (2000).
39. Hoyer, P.O. & Hyvarinen, A. in *Neural Information Processing Systems* 277–284 (MIT Press, Cambridge, Massachusetts, 2003).
40. Sahani, M. & Dayan, P. Doubly distributional population codes: simultaneous representation of uncertainty and multiplicity. *Neural Comput.* **15**, 2255–2279 (2003).
41. Rao, R.P. Bayesian computation in recurrent neural circuits. *Neural Comput.* **16**, 1–38 (2004).
42. Deneve, S. in *Neural Information Processing Systems* 353–360 (MIT Press, Cambridge, Massachusetts, 2005).
43. Jazayeri, M. & Movshon, J.A. Optimal representation of sensory information by neural populations. *Nat. Neurosci.* **9**, 690–696 (2006).
44. Poggio, T. A theory of how the brain might work. *Cold Spring Harb. Symp. Quant. Biol.* **55**, 899–910 (1990).
45. Douglas, R.J. & Martin, K.A. A functional microcircuit for cat visual cortex. *J. Physiol. (Lond.)* **440**, 735–769 (1991).
46. Heeger, D.J. Normalization of cell responses in cat striate cortex. *Vis. Neurosci.* **9**, 181–197 (1992).
47. Nelson, J.I., Salin, P.A., Munk, M.H., Arzi, M. & Bullier, J. Spatial and temporal coherence in cortico-cortical connections: a cross-correlation study in areas 17 and 18 in the cat. *Vis. Neurosci.* **9**, 21–37 (1992).
48. Huys, Q., Zemel, R.S., Natarajan, R. & Dayan, P. Fast population coding. *Neural Comput.* (in the press).

Supplementary Materials

This section is organized into three parts. In the first, we show that when the likelihood function, $p(\mathbf{r}|s)$, belongs to the exponential family with linear sufficient statistics, optimal cue combination can be performed by a simple network in which firing rates from two population codes are combined linearly. Moreover, we show that the tuning curves of the two populations don't need to be identical, and that the responses both within and across populations don't need to be uncorrelated. In the second part, we consider the specific case of independent Poisson noise, which provides an example of a distribution belonging to the exponential family with linear sufficient statistics. We also consider a distribution that does not belong to the exponential family with linear sufficient statistics, namely, independent Gaussian noise with fixed variance. We show that, for this case, optimal cue combination requires a nonlinear combination of the population codes. In the third part, we describe in detail the parameters of the network of conductance-based integrate-and-fire neurons.

1. Probabilistic Population Codes for Optimal Cue Combination

1.1 Bayesian inference through linear combinations for the exponential family

Consider two population codes, $\mathbf{r}_1$ and $\mathbf{r}_2$ (both of which are vectors of firing rates), which code for the same stimulus, s. As described in the main text, this coding is probabilistic, so $\mathbf{r}_1$ and $\mathbf{r}_2$ are related to the stimulus via a likelihood function, $p(\mathbf{r}_1,\mathbf{r}_2|s)$. In a cue integration experiment, we need to construct a third population code, $\mathbf{r}_3$, related to $\mathbf{r}_1$ and $\mathbf{r}_2$ via some function: $\mathbf{r}_3=\mathbf{F}(\mathbf{r}_1,\mathbf{r}_2)$. Given this function, $p(\mathbf{r}_3|s)$ is given by

$$p(\mathbf{r}_3 \mid s) = \int p(\mathbf{r}_1, \mathbf{r}_2 \mid s)\, \delta\left(\mathbf{r}_3 - F(\mathbf{r}_1, \mathbf{r}_2)\right) d\mathbf{r}_1 d\mathbf{r}_2. \qquad \text{(SM1)}$$

When $\mathbf{F}(\mathbf{r}_1,\mathbf{r}_2)$ is not invertible ($\mathbf{r}_3$ does not uniquely identify both $\mathbf{r}_1$ and $\mathbf{r}_2$), such a transformation could easily lose information. Our goal here is to find a transformation that does *not* lose information. Specifically, we want to choose $\mathbf{F}(\mathbf{r}_1,\mathbf{r}_2)$ so that

$$p(\mathbf{r}_3 \mid s) = p\left(\mathbf{F}(\mathbf{r}_1, \mathbf{r}_2) \mid s\right) \propto p(\mathbf{r}_1, \mathbf{r}_2 \mid s) \qquad \text{(SM2)}$$

where all terms are viewed as functions of s and the constant of proportionality is independent of s. If Equation (SM2) is satisfied, then Bayes' rule implies that $p(s|\mathbf{r}_3)$ is identical to $p(s|\mathbf{r}_1,\mathbf{r}_2)$, and one can use $\mathbf{r}_3$ rather than $\mathbf{r}_1$ and $\mathbf{r}_2$ without any loss of information about the stimulus. A function $\mathbf{F}(\mathbf{r}_1,\mathbf{r}_2)$ that satisfies Equation (SM2) is said to be *Bayes optimal*.

Clearly, the optimal function $\mathbf{F}(\mathbf{r}_1,\mathbf{r}_2)$ depends on the likelihood, $p(\mathbf{r}_1,\mathbf{r}_2|s)$. Here we show that if the likelihood lies in a particular family – exponential with linear sufficient statistics – then $\mathbf{F}(\mathbf{r}_1,\mathbf{r}_2)$ is linear in both $\mathbf{r}_1$ and $\mathbf{r}_2$. This makes optimal Bayesian inference particularly simple.

We start by considering the independent case, $p(\mathbf{r}_1,\mathbf{r}_2|s) = p(\mathbf{r}_1|s)p(\mathbf{r}_2|s)$; we generalize to the dependent case later on. As stated above, we consider likelihoods in the exponential family with linear sufficient statistics,

$$p\left(\mathbf{r}_k \mid s\right) = \frac{\phi_k\left(\mathbf{r}_k\right)}{\eta_k\left(s\right)} \exp\left(\mathbf{h}_k^{\mathrm{T}}\left(s\right)\mathbf{r}_k\right) \tag{SM3}$$

where the superscript "T" denotes transpose and k=1, 2. Given this form for $p(\mathbf{r}_k|s)$, we show that if $\mathbf{h}_1(s)$ and $\mathbf{h}_2(s)$ can both be expressed as $\mathbf{h}_k(s)=\mathbf{A}_k\mathbf{b}(s)$ for some stimulus independent matrix $\mathbf{A}_k$ (i=1, 2), then optimal combination is performed by the linear function

$$\mathbf{r}_3 = \mathbf{F}\left(\mathbf{r}_1, \mathbf{r}_2\right) = \mathbf{A}_1^{\mathrm{T}}\mathbf{r}_1 + \mathbf{A}_2^{\mathrm{T}}\mathbf{r}_2 \tag{SM4}$$

In other words, we show that when $\mathbf{r}_3$ is given by Equation (SM4) with $\mathbf{A}_1$ and $\mathbf{A}_2$ chosen correctly, Equation (SM2) is satisfied. Moreover, we show that the likelihood function $p(\mathbf{r}_3|s)$ lies in the same family of distributions as $p(\mathbf{r}_1|s)$ and $p(\mathbf{r}_2|s)$. This is important because it demonstrates that this approach – taking linear combinations of firing rates to perform optimal Bayesian inference – can be either repeated iteratively or cascaded from one population to the next. Finally, in section 1.2 below, we show that the stimulus

dependent kernel functions, $\mathbf{h}_k(s)$, are related to the tuning curves of the populations, $\mathbf{f}_k(s)$, via the relationship

$$\mathbf{f}_k'(s) = \mathbf{\Sigma}_k(s)\mathbf{h}_k'(s) \tag{SM5}$$

where $\mathbf{\Sigma}_k(s)$ is the covariance matrix and $\mathbf{f}_k(s)$ is the tuning curve of the populations i=1,2.

To demonstrate these three properties, we use Equations (SM1) and (SM4), along with $\mathbf{h}_k(s)=\mathbf{A}_k\mathbf{b}(s)$, to compute the left hand side of Equation (SM2),

$$\begin{aligned} p(\mathbf{r}_3 \mid s) &= \int \frac{\phi_1(\mathbf{r}_1)\phi_2(\mathbf{r}_2)}{\eta_1(s)\eta_2(s)} \exp\left(\mathbf{b}^{\mathrm{T}}(s)\mathbf{A}_1^{\mathrm{T}}\mathbf{r}_1 + \mathbf{b}^{\mathrm{T}}(s)\mathbf{A}_2^{\mathrm{T}}\mathbf{r}_2\right)\delta\left(\mathbf{r}_3 - \mathbf{A}_1^{\mathrm{T}}\mathbf{r}_1 - \mathbf{A}_2^{\mathrm{T}}\mathbf{r}_2\right) d\mathbf{r}_1 d\mathbf{r}_2 \\ &= \int \frac{\phi_1(\mathbf{r}_1)\phi_2(\mathbf{r}_2)}{\eta_1(s)\eta_2(s)} \delta\left(\mathbf{r}_3 - \mathbf{A}_1^{\mathrm{T}}\mathbf{r}_1 - \mathbf{A}_2^{\mathrm{T}}\mathbf{r}_2\right) d\mathbf{r}_1 d\mathbf{r}_2 \exp\left(\mathbf{b}^{\mathrm{T}}(s)\mathbf{r}_3\right) \\ &= \frac{\phi_3(\mathbf{r}_3)}{\eta_3(s)} \exp\left(\mathbf{b}^{\mathrm{T}}(s)\mathbf{r}_3\right) \end{aligned} \tag{SM6}$$

where $\phi_3(\mathbf{r}_3) = \int \delta\left(\mathbf{r}_3 - \mathbf{A}_1^{\mathrm{T}}\mathbf{r}_1 - \mathbf{A}_2^{\mathrm{T}}\mathbf{r}_2\right)\phi_1(\mathbf{r}_1)\phi_2(\mathbf{r}_2)\, d\mathbf{r}_1 d\mathbf{r}_2$ and $\eta_3(s) = \eta_1(s)\eta_2(s)$. Meanwhile, the right hand side of Eq. (SM2) is given by

$$\begin{aligned} p(\mathbf{r}_1, \mathbf{r}_2 \mid s) &= \frac{\phi_1(\mathbf{r}_1)\phi_2(\mathbf{r}_2)}{\eta_1(s)\eta_2(s)} \exp\left(\mathbf{b}^{\mathrm{T}}(s)\mathbf{A}_1^{\mathrm{T}}\mathbf{r}_1 + \mathbf{b}^{\mathrm{T}}(s)\mathbf{A}_2^{\mathrm{T}}\mathbf{r}_2\right) \\ &= \frac{\phi_1(\mathbf{r}_1)\phi_2(\mathbf{r}_2)}{\eta_1(s)\eta_2(s)} \exp\left(\mathbf{b}^{\mathrm{T}}(s)\mathbf{r}_3\right). \end{aligned} \tag{SM7}$$

Comparing Equations (SM6) and (SM7), we see that both equations have the same dependence upon s, which implies that Equation (SM2) is satisfied, and thus information in preserved. Therefore, we conclude that optimal cue combination is performed by Equation (SM4), regardless of the choice of measure functions $\phi_1(\mathbf{r}_1)$ and $\phi_2(\mathbf{r}_2)$. While conditional independence of the two populations is assumed in the above derivation, this

assumption is not necessary. Rather (as we show below), it is sufficient to assume that the joint distribution of $\mathbf{r}_1$ and $\mathbf{r}_2$ takes the form

$$p(\mathbf{r}_1, \mathbf{r}_2 \mid s) = \frac{\phi(\mathbf{r}_1, \mathbf{r}_2)}{\eta(s)} \exp\left(\mathbf{b}^{\mathrm{T}}(s)\mathbf{A}_1^{\mathrm{T}}\mathbf{r}_1 + \mathbf{b}^{\mathrm{T}}(s)\mathbf{A}_2^{\mathrm{T}}\mathbf{r}_2\right) \tag{SM8}$$

for any $\phi(\mathbf{r}_1, \mathbf{r}_2)$ (see Equation (SM21)).

So far we have assumed that the likelihood, $p(\mathbf{r}|s)$, is a function *only* of the stimulus, s. In fact, the likelihood often depends on what is commonly called a *nuisance parameter* – something that affects the response distributions of the individual neural populations, but that the brain doesn't care about. For example, it is well known that contrast strongly affects the gain of the population and thus strongly affects the likelihood function. Since contrast represents the quality of the information about the stimulus but is otherwise independent of the actual value of the stimulus, the gain of the population, in this context, represents a nuisance parameter. To model this gain dependence, the likelihood functions for populations 1 and 2 should be written as $p(\mathbf{r}_1|s, g_1)$ and $p(\mathbf{r}_2|s, g_2)$ where g_k denotes gain of population k. Although we could apply our formalism and simply treat g_1 and g_2 as part of the stimulus, if we did that the likelihood for $\mathbf{r}_3$ would contain the term $\exp(\mathbf{b}^{\mathrm{T}}(s, g_1, g_2)\ \mathbf{r}_3)$ (see Equation (SM8)). This is clearly inconvenient, because it means we would have to either know g_1 and g_2, or marginalize over these quantities, to extract the posterior distribution of the stimulus, s.

Fortunately, it is easy to show that this problem can be avoided if the nuisance parameter does not appear in the exponent, so that the likelihood is written

$$p(\mathbf{r} \mid s, g) = \phi(\mathbf{r}, g) \exp\left(\mathbf{h}^{\mathrm{T}}(s)\mathbf{r}\right), \tag{SM9}$$

which is Equation (6) in the main text. When this is the case, either specifying g or multiplying by an arbitrary prior on g and marginalizing yields a conditional distribution $p(\mathbf{r}|s)$ which is in the desired family. If $\mathbf{h}(s)$ had been a function of g then this would not necessarily have been the case. Note that the normalization factor, $\eta(s,g)$, from Equation (SM8) is not present in Equation (SM9). This is because, marginalization of $p(\mathbf{r}|s,g)$ with

respect to an arbitrary prior $p(g)$ will only leave the stimulus dependent kernel h(s) unchanged when the partition function, $\eta(s,g)$, factorizes into a term which depends only on s and a term which depends only on g, or equivalently, when

$$\begin{aligned} 0 &= \frac{d}{dg}\frac{d}{ds}\log\eta(s,g) \\ &= \frac{d}{dg}\frac{d}{ds}\log\int\phi(\mathbf{r},g)\exp\left(\mathbf{h}(s)^{\mathrm{T}}\mathbf{r}\right)d\mathbf{r}\,. \\ &= \frac{d}{dg}\mathbf{h}'(s)^{\mathrm{T}}\mathbf{f}(s,g) \end{aligned} \tag{SM10}$$

However, when g is the gain, $\mathbf{f}(s,g) = g\bar{\mathbf{f}}(s)$, where $\bar{\mathbf{f}}(s)$ is independent of g. This implies

$$\begin{aligned} 0 &= \frac{d}{dg}\mathbf{h}'(s)^{\mathrm{T}}\, g\bar{\mathbf{f}}(s) \\ &= \mathbf{h}'(s)^{\mathrm{T}}\,\bar{\mathbf{f}}(s) \end{aligned}\,. \tag{SM11}$$

However, since

$$\frac{d}{ds}\log\eta(s,g) = \mathbf{h}'(s)^{\mathrm{T}}\, g\bar{\mathbf{f}}(s) \tag{SM12}$$

we can conclude (by combining SM11 and SM12) that when g is the gain, η(s,g) only factorizes when it is independent of s. Fortunately, this seemingly strict condition is satisfied in many biologically relevant scenarios. For example, this is the case if the function $\phi(\mathbf{r},g)$ and $\mathbf{h}(s)$ are both shift invariant, a standard assumption in theoretical studies of population codes. Here shift invariance means that

$$\begin{aligned} \phi(\mathbf{Sr},g) &= \phi(\mathbf{r},g) \\ \mathbf{h}(s+k\Delta s) &= \mathbf{S}\mathbf{h}(s) \end{aligned} \tag{SM13}$$

where the N dimensional matrix $\mathbf{S}$ takes the form $s_{ij} = 1$ when mod(i-j-k,N)=0 and is zero otherwise, k is an integer which tells us how much the indices are shifted, and N is the number of neurons. Note that Equation (SM13) guarantees translation-invariant kernels,

$$h_i(s) = h(s - s_i), \tag{SM14}$$

and also translation invariant tuning curves and covariance matrices. Using the definition of the partition function, $\eta(s,g)$, and noting that det($\mathbf{S}$)=1, we see that

$$\begin{aligned}
\eta(s,g) &= \int \phi(\mathbf{r},g)\exp(\mathbf{h}^{\mathbf{T}}(s)\mathbf{r})d\mathbf{r} \\
&= \int \phi(\mathbf{Sz},g)\exp\left(\mathbf{h}(s)^{\mathbf{T}}\mathbf{Sz}\right)d\mathbf{z} \\
&= \int \phi(\mathbf{z},g)\exp\left(\left(\mathbf{S}^{\mathbf{T}}\mathbf{h}(s)\right)^{\mathbf{T}}\mathbf{z}\right)d\mathbf{z} \\
&= \int \phi(\mathbf{z},g)\exp(\mathbf{h}^{\mathbf{T}}(s-k\Delta s)\mathbf{z})d\mathbf{z} \\
&= \eta(s-k\Delta s,g).
\end{aligned} \tag{SM15}$$

Since k was arbitrary, $\eta(s,g)$ must be independent of s, and therefore is constant and any g dependence can be absorbed into $\phi(\mathbf{r},g)$.

Alternatively, we could also have concluded that $\eta(s,g)$ is independent of s by simply assuming that silence is uninformative, i.e. $p(s|\mathbf{r}=\mathbf{0},g)$ is equal to the prior $p(s)$, i.e.

$$\begin{aligned}
p(s) &= p(s \mid \mathbf{r}=\mathbf{0},g) \\
&= \frac{\phi(\mathbf{0},g)\,p(s)}{\eta(s,g)}\left(\int \frac{\phi(\mathbf{0},g)\,p(s')}{\eta(s',g)}ds'\right)^{-1}. \\
&= \frac{p(s)}{\eta(s,g)}\left(\int \frac{p(s')}{\eta(s',g)}ds'\right)^{-1}
\end{aligned} \tag{SM16}$$

Since the second term in the product on the right hand side is only a function of g equality holds only when $\eta(s,g)$ is independent of s. As shown in Fig. 3 in the main text, this condition can hold even when the tuning curves are not perfectly translation invariant.

1.2 Relationship between the tuning curves, the covariance matrix and the stimulus dependent kernel ***h****(s)*

In this section, we show that our approach works for a very wide range of tuning curves and covariance matrices. This follows from the combination of two facts 1) optimal combination via linear operations requires only that the stimulus dependent kernels, $\mathbf{h}_1(s)$ and $\mathbf{h}_2(s)$, be drawn from a common basis, i.e. $\mathbf{h}_k(s)=\mathbf{A}_k\mathbf{b}(s)$ and 2) the tuning curves and covariance matrix are related to the stimulus dependent kernels $\mathbf{h}(s)$ through a simple relationship (Equation (SM18) below). The first of these was shown in the previous section; the second we show here.

For any distribution of the form of Equation (SM9) , a relationship between the tuning curve and the stimulus dependent kernel can be obtained through a consideration of the derivative of the mean, $\mathbf{f}(s,g)$, with respect to the stimulus,

$$\begin{aligned}
\mathbf{f}'(s,g) &= \frac{d}{ds}\frac{\int \mathbf{r}\phi(\mathbf{r},g)\exp\left(\mathbf{h}^{\mathrm{T}}(s)\mathbf{r}\right)d\mathbf{r}}{\int \phi(\mathbf{r},g)\exp\left(\mathbf{h}^{\mathrm{T}}(s)\mathbf{r}\right)d\mathbf{r}} \\
&= \frac{\int \mathbf{r}\mathbf{r}^{\mathrm{T}}\mathbf{h}'(s)\phi(\mathbf{r},g)\exp\left(\mathbf{h}^{\mathrm{T}}(s)\mathbf{r}\right)d\mathbf{r}}{\int \phi(\mathbf{r},g)\exp\left(\mathbf{h}^{\mathrm{T}}(s)\mathbf{r}\right)d\mathbf{r}} \\
&\quad - \frac{\int \mathbf{r}\phi(\mathbf{r},g)\exp\left(\mathbf{h}^{\mathrm{T}}(s)\mathbf{r}\right)d\mathbf{r}}{\int \phi(\mathbf{r},g)\exp\left(\mathbf{h}^{\mathrm{T}}(s)\mathbf{r}\right)d\mathbf{r}}\frac{\int \mathbf{r}^{\mathrm{T}}\mathbf{h}'(s)\phi(\mathbf{r},g)\exp\left(\mathbf{h}^{\mathrm{T}}(s)\mathbf{r}\right)d\mathbf{r}}{\int \phi(\mathbf{r},g)\exp\left(\mathbf{h}^{\mathrm{T}}(s)\mathbf{r}\right)d\mathbf{r}} \qquad \text{(SM17)} \\
&= \left\langle \mathbf{r}\mathbf{r}^{\mathrm{T}}\right\rangle_{s,g}\mathbf{h}'(s) - \mathbf{f}(s,g)\mathbf{f}^{\mathrm{T}}(s,g)\mathbf{h}'(s) \\
&= \mathbf{\Sigma}(s,g)\mathbf{h}'(s).
\end{aligned}$$

Here $\mathbf{\Sigma}(s,g)$ is the covariance matrix and we have expressed the partition function $\eta(s,g)$ in its integral form. Clearly, since the covariance matrix may depend upon the stimulus, there is a great variety of tuning curves which may be optimally combined.

When the gain is present as a nuisance parameter, this relationship may also be used to demonstrate that the covariance matrix must be proportional to the gain. This is because we can rewrite Equation (SM18) as

$$\mathbf{h}'(s) = \mathbf{\Sigma}^{-1}(s,g)\mathbf{f}'(s,g) \tag{SM18}$$

This corresponds to Equation (7) in the main text. As noted above, the kernel **h**(s) must be independent of gain for the optimality of linear combinations. Since $\mathbf{f}'(s,g) = g\bar{\mathbf{f}}'(s)$ where $\bar{\mathbf{f}}(s)$ is independent of gain, this occurs if the covariance matrix is also proportional to the gain. Since the diagonal elements of the covariance matrix correspond to the variance, the constant of proportionality gives the Fano factor. The precise value of the constant of proportionality, and thus of the Fano factor, is not important, so long as it is independent of the gain.

1.3 Constraint on the posterior distribution over s

The basis from which **h**(s) is drawn not only determines whether or not two populations may be optimally combined, but also places some restrictions on the set of posterior distributions that can be represented. These restrictions, however, are quite weak in the sense that, for proper choices of the kernel **h**(s), a very wide range of posterior distributions can be represented.

For instance, consider the case in which the partition function, $\eta(s,g)$, is independent of s, so that the posterior distribution is simply

$$p(s \mid \mathbf{r}) \propto \exp\left(\mathbf{h}^{\mathrm{T}}(s)\mathbf{r}\right) \tag{SM19}$$

Thus, the log of the posterior is a linear combination of the functions that make up the vector **h**(s), and we may conclude that almost any posterior may be well approximated when this set of functions is "sufficiently rich." Of course, it is also possible to restrict the set of posterior distributions by an appropriate choice for **h**(s). For instance, if it is desirable that the posterior distribution be constrained to be Gaussian, we could simply restrict the basis of **h**(s) to the set quadratic functions of s. Equation (SM20) also indicates why gain is a particularly important nuisance parameter for distributions in this family: an increase in the amplitude of the population pattern of activity, **r**, leads to a significant increase in the sharpness of the posterior through the exponentiation.

1.4 Neural variability and the exponential family with linear sufficient statistics

In the above derivation we made no explicit assumptions regarding the covariance structure of the joint distribution of $\mathbf{r}_1$ and $\mathbf{r}_2$. Fortunately, as with the Gaussian distribution, there are members of this family of distributions which are capable of modeling the first-order and second-order statistics of any response distribution, as long at the tuning curves depend on the stimulus. A complete set of restrictions can be obtained through a consideration of the higher s derivatives of either the tuning curve or the partition function. However, as with the Gaussian distribution, these restrictions concern only the third and higher moments.

Together with Equation (SM18), these arguments indicate that a broad class of correlation structures between populations can also be incorporated into this encoding scheme. Specifically, in Equation (SM18) we did not specify whether or not the responses referred to one or two populations. Thus, the vector mean and covariance matrix of Equation (SM18), could have referred to a pair of correlated populations, i.e.,

$$\mathbf{f}(s,g)=\begin{bmatrix}\mathbf{f}_1(s,g)\\ \mathbf{f}_2(s,g)\end{bmatrix},\quad \mathbf{\Sigma}(s,g)=\begin{bmatrix}\mathbf{\Sigma}_{11}(s,g) & \mathbf{\Sigma}_{12}(s,g)\\ \mathbf{\Sigma}_{21}(s,g) & \mathbf{\Sigma}_{22}(s,g)\end{bmatrix},\quad \text{and}\quad \mathbf{h}(s)=\begin{bmatrix}\mathbf{h}_1(s)\\ \mathbf{h}_2(s)\end{bmatrix}. \tag{SM20}$$

When this is the case, the two populations may be optimally combined, provided $\mathbf{h}_1(s)$ and $\mathbf{h}_2(s)$, as obtained from Equations (SM18) and (SM21), are independent of g and linearly related, or more generally, drawn from a common basis.

2. An example showing explicitly that a linear combination is optimal (Poisson neurons), and a second example showing that a linear combination is not optimal (Gaussian neurons).

2.1 Independent Poisson neurons

We now consider an example of a distribution that belongs to the exponential family with linear sufficient statistics, namely the independent Poisson distribution. We also assume

that the neurons have Gaussian tuning curves which are dense and translation invariant, i.e., $\sum_i f_i(s) = c$, where c is some constant. For this case, we have

$$\begin{aligned} p(\mathbf{r} \mid s, g) &= \prod_i \frac{\left(g f_i(s)\right)^{r_i}}{r_i!} \exp\left(-g f_i(s)\right) \\ &= \exp\left(-g \sum_i f_i(s)\right) \prod_i \frac{(g)^{r_i}}{r_i!} \exp\left(r_i \log\left(f_i(s)\right)\right) \\ &= \exp(-gc) \left(\prod_i \frac{(g)^{r_i}}{r_i!}\right) \exp\left(\sum_i r_i \log\left(f_i(s)\right)\right) \\ &= \phi(\mathbf{r}, g) \exp\left(\mathbf{h}^{\mathrm{T}}(s)\mathbf{r}\right). \end{aligned} \tag{SM21}$$

Here $h_i(s) = \log(f_i(s))$ and g represents, as usual, the gain. Clearly, this likelihood function satisfies Equation (SM9). The stimulus dependent kernel $\mathbf{h}(s)$ in this case is simply the log of the tuning curves. Moreover, it is easy to show that if we marginalize out the gain we obtain a likelihood function, $p(\mathbf{r}|s)$, that satisfies Equation (SM3) regardless of the prior on g. In other words, for independent Poisson noise, optimal cue combination only involves linear combination of population pattern of activity. Moreover, for Gaussian tuning curves, the log of each $f_i(s)$ is quadratic in s, implying that the resulting posterior distribution is also a Gaussian with a variance, $\sigma^2(\mathbf{r})$, that is inversely proportional to the amplitude, i.e.,

$$\frac{1}{\sigma^2(\mathbf{r})} = \sum_i \frac{r_i}{\sigma_i^2}. \tag{SM22}$$

Here, σ_i is the width of the i^{th} tuning curve.

2.2 Gaussian distributed neurons

In the above example, the assumption that the tuning curves are dense insures the parameter g can be marginalized without affecting the stimulus dependence of the likelihood function. This is not, however, always the case. For example, consider a

population pattern of activity that has some stimulus-dependent mean $g\mathbf{f}(s)$ that is corrupted by independent Gaussian noise with a fixed variance σ^2, i.e.,

$$\begin{aligned} p(\mathbf{r} \mid s, g) &= \left|2\pi\sigma^2\right|^{-N/2} \exp\left(-\frac{1}{2\sigma^2}\left(\mathbf{r} - g\mathbf{f}(s)\right)^{\mathrm{T}}\left(\mathbf{r} - g\mathbf{f}(s)\right)\right) \\ &= \left|2\pi\sigma^2\right|^{-N/2} \exp\left(-\frac{\mathbf{r}^{\mathrm{T}}\mathbf{r}}{2\sigma^2}\right)\exp\left(-\frac{g^2\mathbf{f}^{\mathrm{T}}(s)\mathbf{f}(s)}{2\sigma^2}\right)\exp\left(\frac{g\mathbf{f}^{\mathrm{T}}(s)\mathbf{r}}{2\sigma^2}\right) \\ &= \left|2\pi\sigma^2\right|^{-N/2} \exp\left(-\frac{\mathbf{r}^{\mathrm{T}}\mathbf{r}}{2\sigma^2}\right)\exp\left(-\frac{g^2 c}{2\sigma^2}\right)\exp\left(\frac{g\mathbf{f}^{\mathrm{T}}(s)\mathbf{r}}{2\sigma^2}\right) \\ &= \phi(\mathbf{r}, g)\exp\left(g\mathbf{h}^{\mathrm{T}}(s)\mathbf{r}\right). \end{aligned} \quad \text{(SM23)}$$

Here, $\mathbf{h}(s) = \mathbf{f}(s)/\sigma^2$ and the density of the tuning curves implies that $\mathbf{f}(s)^{\mathrm{T}}\mathbf{f}(s)$ is constant, independent of s. Unlike the independent Poisson case, it is now impossible to marginalize an arbitrary prior on the gain without affecting the stimulus dependence of the likelihood function. Of course, if the gains of two such populations are known, optimal Bayesian inference is performed by the linear operation,

$$\mathbf{r}_3 = g_1\mathbf{r}_1 + g_2\mathbf{r}_2. \quad \text{(SM24)}$$

However, if the gains of both populations are not constant across trials, then the use of Equation (SM25) requires that the weights of the linear operation be changed on a trial by trial basis. That is, the gain of each population must be approximated, presumably from the activities of the populations themselves, such that

$$\mathbf{r}_3 = g_1(\mathbf{r}_1)\mathbf{r}_1 + g_2(\mathbf{r}_2)\mathbf{r}_2. \quad \text{(SM25)}$$

Thus, for additive Gaussian noise, optimal cue combination cannot be performed by a linear operation.

3. Simulations with simplified neurons

This simulation (summarized in Fig.3 of the main text) illustrates the optimality of our approach for a network with different types of tuning curves in the input layers. Here we provide the details of those simulations.

The input contains three layers, with N neurons in each. One of the layers has Gaussian tuning curves; the other two have sigmoidal tuning curves; one monotonically increasing and the other monotonically decreasing. In all cases the noise is independent and Poisson. We generated the tuning curves using a two step process. First we generated the kernels, $\mathbf{h}_k(s)$, for each input layer (k=1, 2 or 3) by combining linearly a set of basis functions, denoted $\mathbf{b}(s)$, using three distinct matrices, $\mathbf{A}_1$, $\mathbf{A}_2$ and $\mathbf{A}_3$. We then used the exponential of these kernels as input tuning curves. This is the correct choice of tuning curves when the noise is independent and Poisson.

The activity in the output layer was obtained by summing the input activity multiplied by the transpose of $\mathbf{A}_1$, $\mathbf{A}_2$ and $\mathbf{A}_3$ (as specified by Equations (SM4) and (SM29); see below). This procedure ensures that the kernel of the output layer is simply the basis set, $\mathbf{b}(s)$, used to generate the input kernels.

Generating the input kernel $\mathbf{h}_k(s)$ *and input tuning curves* $\mathbf{f}_k(s)$

We first generated a set of N basis functions defined as

$$b_i(s) = \log\left[M\left(\exp\left(-\frac{(s-s_i)^2}{2\sigma_i^2} \right) + c_i \right) \right]$$

with N = 51, M = 1, σ_i^2 = 32, c_i = 0.1 and s_i = -400+16**i*. These basis functions were combined linearly to obtained the kernels in each of the input layers

$$\mathbf{h}_k(s) = \mathbf{A}_k \mathbf{b}(s) \qquad \text{(SM26)}$$

where again k=1,2 and 3 (corresponding to the three input layers), and $\mathbf{A}_k$ is a matrix of coefficients specific to each input layer. The matrices $\mathbf{A}_k$ were obtained using linear

regression with a regularizer (to keep the weights smooth and small). Specifically, we used

$$\mathbf{A}_k^T = \left[\mathbf{C_b} + d\mathbf{I}\right]^{-1} \mathbf{C}^k_{\mathbf{bh}^*} \qquad \text{(SM27)}$$

where $\mathbf{C_b}$ is the covariance matrix of the basis set **b** (across all values of *s*, assuming a uniform distribution over the range [-400, 400]), $\mathbf{C}^k_{\mathbf{bh}^*}$ is the covariance between **b** and the target kernel $\mathbf{h}^*$ for input layer *k*, and **I** is the identity matrix. The parameter *d* (the regularizer parameter) was set to 1.

The i^{th} target kernel in the Gaussian input layer was given by

$$h_i^*(s) = \log\left[M\left(\exp\left(-\frac{(s - s_i)^2}{2\sigma_i^2} \right) + d_i \right) \right]$$

with $N = 51$, $M = 1\pm0.5$, $\sigma_i^2 = 32\pm16$, $d_i = 0.1\pm0.1$ and $s_i = -400+16*i\pm4$. In all cases, the notation ± means that the parameters were drawn uniformly from the corresponding range of values (e.g. 32±16 = [16, 48]). The random components were added to introduce variability in the width, position, baseline and amplitude of the input tuning curves.

For the monotonic increasing sigmoidal input layer, the i^{th} target kernel was given by,

$$h_i^*(s) = \log\left[M\left(\frac{1}{1 + \exp\left(-(s - s_i)/t\right)} + d_i \right) \right]$$

with $N = 51$, $M = 1\pm0.5$, $t = 32\pm16$, $d_i = 0.1\pm0.1$ and $s_i = -400+16*i\pm-4$. The same equation and parameters was used in the monotonic decreasing sigmoidal input layer, with a reversed sign in the exponential. The input tuning curves, $\mathbf{f}_k(s)$, were then obtained by taking the log of the input kernels, $\mathbf{h}_k(s)$.

Note that because of the approximation introduced by the linear regression step (Equations (SM27) and (SM28)), the input tuning curves are not exactly equal to the log of the target kernels. Nonetheless, they are quite close and, as a result, the tuning curves in the first input layer were indeed Gaussian, while the tuning curves is the other two layers were sigmoidal (see Fig. 3a in the main text).

Generating one trial

The activity in the input layers on each trial (see Fig. 3b in the main text) were obtained by drawing spike counts from a multivariate independent Poisson distribution with means $\mathbf{f}_k(s)$. The resulting activities, $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{r}_3$, were then combined to obtain the activity in the output layer according to (see Equation (SM4)):

$$\mathbf{r}_o = \left[\mathbf{A}_1^{\mathrm{T}} \mathbf{r}_1 + \mathbf{A}_2^{\mathrm{T}} \mathbf{r}_2 + \mathbf{A}_3^{\mathrm{T}} \mathbf{r}_3 \right]^+ \qquad \text{(SM28)}$$

where the rectification $[]^+$ is defined as $[x]^+ = \max(0,x)$. This rectification is used to ensure that all component of $\mathbf{r}_o$ are non-negative (as is the case for neural activity). This introduces a slight approximation in our scheme but, as can be seen from Fig. 3 in the main text, this has virtually no impact on our results.

Decoding the probability distributions

For a given pattern of activity, $\mathbf{r}_k$, in a layer k, the corresponding probability distributions is obtained through

$$p\left(s \mid \mathbf{r}_k\right) = \frac{1}{Z} \exp\left(\mathbf{h}_k^{\mathrm{T}}\left(s\right) \mathbf{r}_k\right)$$

Z is chosen to ensure that the integral of $p(s|\mathbf{r}_k)$ with respect to s is equal to 1. Note that in the output layer, the i^{th} kernel in the output layer is given by $b_i(s)$.

4. Simulations with conductance-based integrate-and-fire neurons

The objective of the simulations was to demonstrate that networks of conductance-based integrate-and-fire neurons can perform near-optimal Bayesian inference. As a case study, we used a cue combination task in which the network combines two cues whose reliabilities are systematically varied from trial to trial.

Network architecture

The network consists of two unconnected input layers and one output layer. Each input layer contains 252 independent excitatory neurons firing with near-Poisson statistics. The output layer contains 1008 excitatory and 252 inhibitory neurons. The preferred stimuli of the neurons in each layer are equally spaced and uniformly distributed. An excitatory neuron in the output layer receives 24 connections from neurons in each input layer, an inhibitory one receives 16. Connections are drawn randomly without replacement from a Gaussian probability distribution over the stimulus, centered at the preferred stimulus of the output neuron and with a width of σ_{kernel}. Specifically, the probability of making a connection from neuron j to neuron i, denoted p_{ij}, is given by

$$p_{ij} = \frac{\exp\left(-\frac{\left(s_i - s_j\right)^2}{2\sigma_{\text{kernel}}^2}\right)}{\sum_j \exp\left(-\frac{\left(s_i - s_j\right)^2}{2\sigma_{\text{kernel}}^2}\right)} \qquad \text{(SM29)}$$

All connection strengths are equal and constant with value w for a given input layer.

Within the output layer there are two types of lateral connections: inhibitory to excitatory and excitatory to inhibitory. Each excitatory neuron receives 30 connections from inhibitory neurons, and each inhibitory neuron receives 40 connections from excitatory neurons. These are randomly drawn without replacement from a uniform distribution, and the connection strengths are all 1.

Input layers

Neurons in the input layers fire at a constant rate, except that there is a refractory period of 3 ms. More specifically, the probability of firing in any small interval *dt* is a constant times *dt*, except within 3 ms of a previous spike, in which case the probability of firing is 0. As a result, the variance of the spike counts of an input neuron across trials is approximately equal to their mean. The rates are obtained from a Gaussian distribution centered at a given stimulus, with width σ_{input}, plus a baseline set to a fraction of the amplitude (peak rate minus baseline rate),

$$\left\langle r_i \right\rangle_s = g\left(\exp\left(-\frac{\left(s - s_i\right)^2}{2\sigma_{\text{input}}^2} \right) + c \right). \tag{SM30}$$

We used σ_{input} and $c = 0.1$. The gain, *g*, is fixed on any given trial. In the case of the visual system, the amplitude would be related to the contrast of a presented image. The higher the contrast, the higher the input gain, the higher the output gain, and the less variable the estimate of stimulus.

Output layer

The output layer consists of conductance-based integrate-and-fire neurons. The membrane potential, $V_i(t)$, of output neuron *i* as a function of time *t* is described by

$$C\frac{dV_i}{dt} = -g_L\left(V_i - E_L\right) - g_{iE}\left(t\right)\left(V_i - E_E\right) - g_{iI}\left(t\right)\left(V_i - E_I\right) - g_{iA}\left(t\right)\left(V_i - E_A\right) \tag{SM31}$$

where *C* is the capacitance of the membrane and E_E, E_I, E_A, and E_L are reversal potentials. The conductance $g_{iE}(t)$ contains the contributions of the spikes from all excitatory presynaptic neurons. If neuron *i* is of type *a* (which can be *E* or *I*), then this conductance is given by

$$g_{ia}\left(t\right) = \sum_{jk} w_{ij}\bar{g}_{aj}\alpha_{\tau_a}\left(t - t_j^k - d_{ij}\right), \tag{SM32}$$

where $\bar{g}_{aj}$ is the peak conductance following a single incoming spike from the j^{th} excitatory presynaptic neuron, w_{ij} is the conductance weight defined above (1 for $E \rightarrow I$ and $I \rightarrow E$ connections, 0 for $E \rightarrow E$ and $I \rightarrow I$ connections, and w for connections from the input to output layer), t_j^k is the time of the k^{th} spike from neuron j, and d_{ij} is the synaptic delay between neurons i and j. The effect of a spike on the conductance is given by an alpha-function,

$$\alpha_\tau(t) = \frac{t}{\tau}\exp\left(1-\frac{t}{\tau}\right), \quad \text{for } t>0, \text{ and } 0 \text{ otherwise.} \tag{SM33}$$

The synaptic conductance at an excitatory neuron caused by spikes from inhibitory presynaptic neurons follows an expression analogous to Equation (SM30).

The after hyperpolarizing conductance, $g_{iA}(t)$, is induced by the cell's own spikes: $g_{iA}(t) = \bar{g}_A \sum_k \alpha_{\tau_A}\left(t - t_i^k - d_A\right)$, with d_A a delay, t_i^k the time of the cell's own k^{th} spike. The leak conductance, g_L, is constant. When the membrane potential exceeds the spike threshold (-55 mV), a spike is emitted, the potential is reset to -60 mV, where it is held for an absolute refractory period. This refractory period is 3 ms for excitatory neurons and 1.5 ms for inhibitory neurons. Moreover, the spike threshold is elevated by 10 mV and exponentially decays back to -55 mV with a time constant of 10 ms; this mimics a relative refractory period.

Parameters

The reversal potentials are $E_E = 0$ mV, $E_I = -70$ mV, $E_A = -90$ mV, and $E_L = -70$ mV. The time constants for the conductances are $\tau_E = 1$ ms and $\tau_I = \tau_A = 2$ ms. Excitatory neurons have $C = 0.5$ mF, $g_L = 25$ nS, and $\bar{g}_A = 40$ nS. Inhibitory neurons have $C = 0.2$ mF, $g_L = 20$ nS, and $\bar{g}_A = 20$ nS. The synaptic delays, d_{ij}, between inhibitory and excitatory neurons in the output layer are randomly drawn

from a zero-bounded normal distribution with mean 3 ms and standard deviation 1 ms, with no delay exceeding 6 ms; the delay d_A is 1 ms. The peak conductances are given as follows: $\bar{g}_{aj} = 12$ nS if $a = E$ and j refers to a neuron in the input layer; $\bar{g}_{aj} = 10$ nS if $a = I$ and j refers to a neuron in the input layer or if $a = E$ and j refers to an inhibitory neuron in the output layer; $\bar{g}_{aj} = 3$ nS if $a = I$ and j refers to an excitatory neuron in the output layer.

For each combination of gains in the input layers we ran 1008 trials. Each trial lasted 500 ms. The equations were integrated using the Euler method with a time step of 0.5 ms.

Three networks were tested:

- σ_{input} =20, σ_{kernel} =15, and w=1 for both input layers;
- σ_{input} =15, σ_{kernel} =20, and w=1.78 for input layer 1, while σ_{input} =25, σ_{kernel} =10, and w=0.77 for input layer 2;
- σ_{input} =15, σ_{kernel} =15, and w=1.78 for input layer 1, while σ_{input} =25, σ_{kernel} =15, and w=0.45 for input layer 2.

Estimating the mean and variance of the posterior distribution

Ideally, one would like to use Bayes' theorem to estimate, on every trial, the mean and variance of the posterior of the distribution encoded by the excitatory neurons in the output layer. Unfortunately, this requires that we first measure the likelihood function, $p(\mathbf{r} \mid s)$. Estimating a probability distribution over 1008 neurons is in practice impossible unless the neurons are independent, which is not the case in these simulations.

Instead, we used an approach which is very similar to the one used in human experiments [1,2]. On every trial, we estimated the mean of the distribution by decoding the output pattern of activity. We then used the mean and variance of this estimate over 1008 trials as estimates of the mean and variance of the posterior distribution. This method will converge to the right values when all distributions are Gaussian and the decoder is optimal. Unfortunately, the optimal decoder also requires knowledge of $p(\mathbf{r} \mid s)$. Therefore, we used a (potentially) suboptimal decoder instead. Specifically, for the mean, we estimated the value of s by applying a least-squares fit of a Gaussian to the population

pattern of activity on a single trial, with the amplitude, width, and peak location as parameters. (We also fit Gaussians with fixed width and amplitude and used only the peak location as a parameter; the results were the same.) The value of the peak location was used as an estimate of s. We repeated these steps over 1008 trials, and reported the estimate averaged over all trials, as is common in psychophysics. Because our decoder is not optimal, our estimates of the mean are not as good as they could be. However, we use the same estimator when only one input is active and when both are active, so a difference in optimality is expected to cancel. To estimate the variance, we used a locally optimal linear estimator [3]. We also computed the variance of the estimates themselves; these were nearly identical.

Comparing network performance to the predictions of optimal Bayesian inference

We first simulated our network with only input layer 1 active. Spike trains were generated in the input layer as described above, with a Gaussian profile centered at s_2=89.5 with gain g_1 (Fig. 3a). The gain could take any integer value between 3 and 18 spikes per second, in increments of 3 spikes/s. For a given gain we performed 1008 trials, and for each trial we measured the spike counts over 500 ms for every neuron and estimated the mean of the posterior distribution (denoted μ_1) as described above. We repeated these steps with only input layer 2 active. Spikes in the input layer followed a Gaussian profile centered at s_2=95.5 with gain g_2 (Fig. 3a). Note that we introduced a cue conflict of 6º, which is fairly large. Again we computed the mean (μ_2) of the posterior distribution encoded in the output layer. Finally, we performed simulations with both input layers active, using all combinations of gains, for a total of 36 (6×6) conditions. We used the same input spike trains as the ones generated when only one input layer was active. The output spike counts were used to compute estimates of the mean of the encoded distribution (μ_3).

After collecting all data, we computed a locally optimal linear estimator from 25,000 trials randomly chosen from all combinations of gains. The weight vector obtained in this manner was subsequently used to estimate the variances in every single condition. For each combination of gains we thus obtained estimates of σ_1^2 (only input

layer 1 active), σ_2^2 (only input layer 2 active), and σ_3^2 (both input layers active). Importantly, we did not train a different estimator for every single combination of gains, but only an overall one. The intuition behind this is that the nervous system does not have the luxury of using specialized decoding circuitry for every possible contrast level of an incoming stimulus.

We then plotted μ_3 against $\mu_1 \frac{\sigma_2^2}{\sigma_1^2+\sigma_2^2} + \mu_2 \frac{\sigma_1^2}{\sigma_1^2+\sigma_2^2}$ (Equation (4), main text), and σ_3^2 against $\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2}$ (Equation (5), main text) for each combination of gains. If the network is performing a close approximation to Bayesian inference, the data should lie close to a line with slope 1 and intercept 0. This procedure was followed separately for each of three networks described above. As can be seen in Fig. 3c,d, it is clear that the network is indeed nearly optimal for all combinations of gains tested, in all three conditions.

References

1. Knill, D. C. & Richards, W. *Perception as Bayesian Inference* (Cambridge University Press, New York, 1996).
2. Ernst, M. O. & Banks, M. S. Humans integrate visual and haptic information in a statistically optimal fashion. *Nature* **415**, 429-33 (2002).
3. Series, P., Latham, P. & Pouget, A. Tuning curve sharpening for orientation selectivity: coding efficiency and the impact of correlations. *Nature Neuroscience* **10**, 1129-1135 (2004).