# Population Codes

*Alexandre Pouget and Peter E. Latham*

## Introduction

Many sensory and motor variables in the brain are encoded by coarse codes, i.e., by the activity of large populations of neurons with broad tuning curves. For example, the direction of visual motion is believed to be encoded in the medial temporal (MT) visual area by a population of cells with bell-shaped tuning to direction, as illustrated in Figure 1*A*. Other examples of variables encoded by populations include the orientation of a line, the contrast in a visual scene, the frequency of a tone, and the direction of intended movement in motor cortex. These encodings extend to two dimensions—a single set of neurons might contain information about both orientation and contrast—or more.

Population codes are computationally appealing for at least two reasons. First, the overlap among the tuning curves allows precise encoding of values that fall between the peaks of two adjacent tuning curves (Figure 1*A*). Second, bell-shaped tuning curves provide basis functions that can be combined to approximate a wide variety of nonlinear mappings. This means that many cortical functions, such as sensorimotor transformations, can be easily modeled with population codes (see Pouget, Zemel, and Dayan, 2000, for a review).

In this article we focus on decoding, or reading out, population codes. Decoding is the simplest form of computation that one can perform over a population code, and as such, it is an essential step toward understanding more sophisticated computations. It is also important for accurately identifying which variables are encoded in a particular brain area and how they are encoded.

A key element of population codes—and the main reason why decoding them is difficult—is that neuronal responses are noisy, meaning that the same stimulus can produce different responses. Consider, for instance, a population of neurons coding for a one-dimensional parameter: the direction, $\theta$, of a moving object. An object moving in a particular direction produces a *noisy* hill of activity across this neuronal population (Figure 1*C*). On the basis of this noisy activity, one can try to come up with a good guess, or estimate, $\hat{\theta}$, of the direction of motion, $\theta$. In the second and third sections of this article we review the various estimators that have been proposed, and in the fourth section we consider their neuronal implementations.

Additional sources of uncertainty, beside neuronal noise, can come from the variable itself. For example, there is intrinsically more variability in one's estimate of, say, motion on a dark night than motion in broad daylight. In cases such as this, it is not unreasonable to assume that population activity codes for more than just a single value, and in the extreme case the population activity could code for a whole *probability distribution*. The goal of decoding is then to recover an estimate of this probability distribution. We consider an example of this later in the article.

## Models of Neuronal Noise and Tuning Curves

To read a population code, it is essential to have a good understanding of the relation between the patterns of activity and the encoded variables. One common assumption, particularly in sensory and motor cortex, is that patterns of activity encode a single value per variable at any given time. This is a reasonable assumption in many situations (although there are exceptions, as discussed later). For example, an object can move in only one direction at a time, so the neurons encoding its direction of motion have only one value to encode.

Under the assumption of a single value, neuronal responses are generally characterized by tuning curves, noted $f_i(\theta)$, which specify the mean activity of cell $i$ as a function of the encoded variable. These tuning curves are typically bell-shaped, and are often taken

to be Gaussian for nonperiodic variables and circular normal for periodic ones.

Simply measuring the mean activity, however, is not sufficient for performing estimation. A neuron may fire at a rate of 20 spikes/s on one trial but only 15 spikes/s on the next, even though the same stimulus was presented both times. This trial-to-trial variability is captured by the noise distribution, $P(a_i = a|\theta)$, where $a_i$ is the activity of cell $i$. The noise distribution is often assumed to be Gaussian, either with fixed variance or with a variance proportional to the mean (the latter being more consistent with experimental data), and independent. Such a distribution has the form

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

where $\sigma_i^2$ is either fixed or equal to the mean, $f_i(\theta)$. Another popular choice, especially useful if one is counting spikes, is the Poisson distribution:

$$P(a_i = k|\theta) = \frac{f_i(\theta)^k e^{-f_i(\theta)}}{k!} \quad (2)$$

Figure 1*C* shows a typical pattern of activity with Gaussian noise and $\sigma_i^2$ fixed.

## Estimating a Single Value

We now consider various approaches to reading out a population code under the assumptions that (1) a single value is encoded at any given time, and (2) the only source of uncertainty is the neuronal noise. Most of these methods, known as estimators, seek to recover an estimate, $\hat{\theta}$, of the encoded variable. We first discuss how one assesses the quality of an estimator in general; we then provide descriptions of common estimators used for decoding population activity.

### *Fisher Information*

An estimate, $\hat{\theta}$, is obtained by computing a function of the observed activity $\mathbf{A}$, where $\mathbf{A} = (a_1, a_2, \ldots)$. Because of neuronal noise, $\mathbf{A}$ is a random variable and thus so is $\hat{\theta}$. This means that $\hat{\theta}$ will vary from trial to trial even for identical presentation angles. The best estimators are ones that are unbiased and efficient. An unbiased estimator is right on average: the conditional mean, $E[\hat{\theta}|\theta]$, is equal to the encoded direction, $\theta$, where $E$ denotes an average over trials. An efficient estimator, on the other hand, is consistent from trial to trial: the conditional variance, $E[(\hat{\theta} - \theta)^2|\theta]$, is minimal.

In general, the quality of an estimator depends on a compromise between the bias and the conditional variance. In this chapter, however, we consider unbiased estimators only, for which the conditional variance is the important measure because it fully determines how well one can discriminate small changes in the encoded variable based on observation of the neuronal activity. There exists a theoretical lower bound on the conditional variance, which is known as the Cramér-Rao bound. For an unbiased estimator, this bound is equal to the inverse of the Fisher information (Paradiso, 1988; Seung and Sompolinsky, 1993) which leads to the inequality

$$E[(\hat{\theta} - \theta)^2] \geq \frac{1}{I_{\text{Fisher}}}$$

where

$$I_{\text{Fisher}} \equiv E\left[-\frac{\partial^2}{\partial\theta^2} \log P(\mathbf{A}|\theta)\right]$$

An efficient estimator is one whose conditional variance is equal to the Cramér-Rao bound, $1/I_{\text{Fisher}}$. When $P(\mathbf{A}|\theta)$ is known, it is often straightforward to compute $I_{\text{Fisher}}$. For example, for the Gaussian distribution given in Equation 1,

$$I_{\text{Fisher}} = \sum_{i=1}^{N} \frac{f_i'(\theta)^2}{\sigma_i^2}$$

and for the Poisson distribution given in Equation 2,

$$^{N} \quad f_i'(\theta)^2$$

$$I_{\text{Fisher}} = \sum_{i=1} \frac{}{f_i(\theta)}$$

(Seung and Sompolinsky, 1993).

In both of these expressions, the neurons that contribute most strongly to the Fisher information are those with a large slope (large $f_i'(\theta)$). Therefore, the most active neurons are not the most informative ones. In fact, they are the *least* informative: the most active neurons correspond to the top of the tuning curve, where the slope is zero, so these neurons make no contribution to Fisher information.

### *Voting Methods*

Several estimators rely on the idea of interpreting the activity of a cell, normalized or not, as a vote for the preferred direction of the cell. For instance, the optimal linear estimator is given by

$$\hat{\theta}_{\text{OLE}} = \sum_{i=0}^{N} \theta_i a_i$$

where $\theta_i$ is the preferred direction of cell $i$, that is, the peak of the function $f_i(\theta)$. A variation on this theme is the center of mass estimator, defined as

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

where $\gamma$ is the spontaneous activity of the cells.

A third variation is known as a population vector estimator (Figure 2*A*). This has been extensively used for estimating periodic variables, such as direction, from real data (Georgopoulos et al., 1982). It is equivalent to fitting a cosine function through the pattern of activity and using the phase of the cosine as the estimate of direction:

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

where

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

The first two methods work best for nonperiodic variables; the third one can only be used when the variables are periodic. All three estimators are subject to biases, although careful tuning of the parameters can often correct for them. More important, all three methods are almost always suboptimal (the variance of the estimator exceeds the Cramér-Rao bound). The exceptions occur for a very specific set of tuning curves and noise distributions (Salinas and Abbott, 1994): the center of mass is optimal only with Gaussian tuning curves and Poisson noise, and the population vector is optimal only for cosine tuning curves and Gaussian noise of fixed variance.

### *Maximum Likelihood*

A better choice than the voting methods, at least from the point of view of statistical efficiency, is the maximum likelihood (ML) estimator,

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

When there are a large number of neurons, this estimator is unbiased and its variance is equal to the Cramér-Rao bound for a wide variety of tuning curve profiles and noise distribution (Paradiso, 1988; Seung and Sompolinsky, 1993). The term maximum likelihood comes from the fact that $\hat{\theta}_{\text{ML}}$ is obtained by choosing the value of $\theta$ that maximizes the conditional probability of the activity, $P(\mathbf{A}|\theta)$, also known as the likelihood of $\theta$.

Finding the ML estimate reduces to template matching (Paradiso, 1988), i.e., finding the noise-free hill that is closest to the activity, as illustrated in Figure 2*B*. If the noise is independent and Gaussian, then "closest" is with respect to the Euclidean norm, $\Sigma_i(a_i$

$- f_i(\theta))^2$. For other distributions the norm is more complicated. Template matching involves a nonlinear regression, which is typically performed by moving the position of the hill until the distance from the data is minimized, as shown in Figure 2*B*. The position of the peak of the final hill corresponds to the ML estimate.

The main difference between the population vector and the ML estimator is the shape of the template being matched to the data. Whereas the population vector matches a cosine, the ML estimator uses a template that is directly derived from the tuning curves of the neurons that generated the activity (Figures 2*A* and 2*B*). (When all neurons have identical tuning curves, as for our examples, the template has the same profile as the tuning curves.) It is because the ML estimator uses the correct template that its variance reaches the Cramér-Rao bound. There is, however, a cost: one needs to know the profile of all tuning curves to use ML estimation, whereas only the preferred directions, $\theta_i$, are needed for the population vector estimator.

### *Bayesian Approach*

An alternative to ML estimation is to use the full posterior distribution of the encoded variable, $P(\theta|\mathbf{A})$. This is related to the distribution of the noise, $P(\mathbf{A}|\theta)$, through Bayes's theorem:

$$P(\theta|\mathbf{A}) = \frac{P(\mathbf{A}|\theta)P(\theta)}{P(\mathbf{A})}$$

where $P(\mathbf{A})$ and $P(\theta)$ are the prior distributions over $\mathbf{A}$ and $\theta$. The value that maximizes $P(\theta|\mathbf{A})$ can then be used as an estimate of $\theta$. This is known as a maximum a posteriori estimate, or MAP estimate. The main advantage of the MAP estimate over the ML estimate is that prior knowledge about the encoded variable can be taken into account. This is particularly important when the conditional distribution, $P(\mathbf{A}|\theta)$, is not sharply peaked compared to the prior, $P(\theta)$. This happens, for example, when only a small number of neurons are available, or when one observes only a few spikes per neuron. The MAP estimate is close to the ML estimate if the prior distribution varies slowly compared to the conditional, and the two are exactly equal when the prior is flat. Several authors have explored and/or applied applied this approach to real data (Foldiak, 1993; Sanger, 1996; Zhang et al., 1998).

## Neuronal Implementations

Methods such as the voting schemes or ML estimator are biologically implausible, for one simple reason: they extract a single value, the estimate of the encoded variable. Such explicit decoding is very rare in the brain. Instead, most cortical areas and subcortical structures use population codes to encode variables. This means that, throughout the brain, population codes are mapped into population codes. Hence, V1 neurons, which are broadly tuned to the direction of motion, project to MT neurons, which are also broadly tuned, but in neither area is the direction of motion read out as a single number. The neurons in MT are nevertheless confronted with an estimation problem: they must choose their activity levels on the basis of the noisy activity of V1 neurons.

What is the optimal strategy for mapping one population code into another? We cannot answer this question in general, but we can address it for the broad class of networks depicted in Figure 3. In these networks, the input layer is a set of neurons with wide tuning curves, generating noisy patterns of activity like the one shown in Figure 1*C*. This activity, which acts transiently, is relayed to an output layer through feedforward connections. In the output layer the neurons are connected through lateral connections.

An update rule (discussed later) causes the activity in the output layer to evolve in time. In the next section we consider networks in which the update rule leads to a smooth hill. The peak of that hill can be interpreted as an estimate of the variable being encoded. As previously, we can assess how well the network did by looking at the mean and variance of this estimate.

We will consider two kinds of networks: those with a linear activation function and those with a nonlinear one.

### *Linear Networks*

We first consider a network with linear activation functions in the output layer, so that the dynamics is governed by the difference equation

$$\mathbf{O}_t = ((1 - \lambda)I + \lambda W)\mathbf{O}_{t-1}, \quad (3)$$

where $\lambda$ is a number between 0 and 1, $I$ is the identity matrix, and $W$ is the matrix for the lateral connections. The activity at time 0, $\mathbf{O}_0$, is initialized to $W\mathbf{A}$, where $\mathbf{A}$ is an input pattern (like the one shown in Figure 1*C*) and $W$ is the feedforward weight matrix (for simplicity, the feedforward and lateral weights are the same, although this is not necessary).

The dynamics of such a network is well understood: each eigenvector of the matrix $(1 - \lambda)I + \lambda W$ evolves independently, with exponential amplification for eigenvalues greater than 1 and exponential suppression for eigenvalues less than 1. When the weights are translation invariant ($W_{ij} = W_{i-j}$), the eigenvectors are sines and cosine. In this case the network amplifies or suppresses independently each Fourier component of the initial input pattern, $\mathbf{A}$, by a factor equal to the corresponding eigenvalue of $(1 - \lambda)I + \lambda W$. For example, if the first eigenvalue of $(1 - \lambda)I + \lambda W$ is more than 1 (respectively less than 1), the first Fourier component of the initial pattern of activity will be amplified (respectively suppressed). Thus, $W$ can be chosen such that the network amplifies selectively the first Fourier component of the data while suppressing the others.

As formulated, the activity in such a network would grow forever. However, if we stop after a large yet fixed number of iterations, the activity pattern will look like a cosine function of direction with a phase corresponding to the phase of the first Fourier component of the data. The peak of the cosine provides the estimate of direction. That estimate turns out to be the same as the one provided by the population vector discussed above.

The unchecked exponential growth of a purely linear network can be alleviated by adding a nonlinear term to act as gain control. This type of network was proposed by Ben-Yishai, Bar-Or, and Sompolinsky (1995) as a model of orientation selectivity.

Although such networks keep the estimate in a coarse code format, they suffer from two problems: it is not immediately clear how to extend them to periodic variables, such as disparity, and they are suboptimal, since they are equivalent to the population estimator.

### *Nonlinear Networks*

To obtain optimal performance, one needs a network that can implement template matching with the correct template—the one used by the ML estimator (see Figure 2*B*). This requires templates that go beyond cosines to include curves that are consistent with the tuning curves of the input units (see Figure 2*B*).

Nonlinear networks that admit line attractors have this property (Deneve, Latham, and Pouget, 1999). In such networks, the line attractors correspond to smooth hills of activity, with profiles determined by the patterns of weights *and* the activation functions. For a given activation function, it is therefore possible to select the weights to optimize the profile of the stable state. Pouget et al. (1998) demonstrated that this extra flexibility allows these networks to act as ML estimators (see Figure 3*B*).

More recent work by Deneve et al. (1999) has shown that the ML property is preserved for a wide range of nonlinear activation functions. In particular, this is true for networks using divisive normalization, a nonlinearity believed to exist in cortical microcircuitry. It is therefore possible that all cortical layers are close approximations to ML estimators.

## Estimating a Probability Distribution

So far we have reviewed decoding methods in which only one value is encoded at any given time and the only source of uncertainty comes from the neuronal activity. Situations exist, however, in which either (or both) of these assumptions are violated. For in-

stance, imagine that you are lost in Manhattan on a foggy day, but you can see, faintly, the Empire State building and the Chrysler building in the distance. Because of the poor visibility, the views of these landmarks are not sufficient to specify your exact position, but they are enough to provide a rough idea of where you are (Harlem versus Little Italy). In this situation, it would be desirable to compute the probability distribution of your location given that you are seeing the landmarks; i.e., compute $P(\theta|w)$ where $\theta$ is the position (now a two-dimensional vector) in Manhattan and $w$ represents the views of the buildings. Here, the uncertainty about $\theta$ comes from the fact that you do not have enough information to tell precisely where you are. In such a situation, the neurons could encode the *probability distribution,* $P(\theta|w)$.

Because the encoded entity is a probability distribution rather than a single value, we can no longer use either Equation 1 or Equation 2 as a model for the responses of the neurons; these equations provide only the likelihood of $\theta$, $P(\mathbf{A}|\theta)$. What we need instead is a model that specifies the likelihood of the whole encoded probability distribution, $P[\mathbf{A}|P(\theta|w)]$. Note that $P(\theta|w)$ plays the same role as $\theta$ previously, which is to be expected, now that $P(\theta|w)$ is the encoded entity. It is beyond the scope of this discussion to provide equations for such models, but examples can be found in Zemel, Dayan, and Pouget (1998).

Since **A** is now a code for the probability distribution, the relevant quantity to estimate is $P(\theta|w)$, which we denote $\hat{P}(\theta|w)$. This is still within the realm of estimation theory, so we can use the same tools that we used for the simpler case, such as ML decoding (see Zemel et al., 1998).

To see the difference between encoding a single value and encoding a probability distribution, it is helpful to consider what happens when the neurons are deterministic—that is, when the neuronal noise goes to zero. In this case, the encoded variable can be recovered with infinite precision, since the only source of uncertainty, the neuronal noise, is gone. Thus the ML estimate would be exactly equal to the encoded value, and the posterior distribution, $P(\theta|\mathbf{A})$, would be a Dirac function centered at $\theta$. If the activity encodes a probability distribution, on the other hand, one would recover the *distribution* with infinite precision. However, the uncertainty about $\theta$ may still be quite large (as was the case in our Manhattan example), potentially far from a Dirac function.

It is too early to tell whether neurons encode probability distributions; more empirical as well as theoretical work is needed. But if the cortex has the ability to represent probability distributions, it might be possible to determine how, and whether, the brain performs Bayesian inferences. Bayesian inference is a powerful method for performing computation in the presence of uncertainty. Many engineering applications rely on this framework to perform data analysis or to control robots, and several studies are now suggesting that the brain might be using such inferences for perception and motor control (see, e.g., Knill and Richards, 1996).

## Conclusions

Understanding how to decode patterns of neuronal activity is a critical step toward developing theories of representation and computation in the brain. This article concentrated on the simplest case, a single variable encoded in the firing rates of a population of neurons. There are two main approaches to this problem. In the first, the population encodes a single value, and decoding can be done with Bayesian or maximum likelihood estimators. The underlying assumption in this case is that neuronal noise is the only source of uncertainty. We also saw that within this framework, one can design neural networks that perform decoding optimally. In the second approach, the population encodes a full probability distribution over the variable of interest. Here both the variable and its uncertainty can be extracted from the population activity. This scheme could be used to perform statistical inferences—a powerful way to perform computations over variables whose value is not known with certainty. The challenge for future work will be to determine whether the brain uses this type of code, and, if so, to understand how realistic neural circuits can perform statistical inferences over probability distributions.

**Roadmap:** Neural Coding
**Related Reading:** Cortical Population Dynamics and Psychophysics; Motor Cortex, Coding and Decoding of Directional Operations

## References


Ben-Yishai, R., Bar-Or, R. L., and Sompolinsky, H., 1995, Theory of orientation tuning in visual cortex, *Proc. Natl. Acad. Sci. USA*, 92:3844–3848.

Deneve, S., Latham, P. E., and Pouget, A., 1999, Reading population codes: A neural implementation of ideal observers, *Nature Neurosci.*, 2:740–745. ◆

Foldiak, P., 1993, The "ideal homunculus": Statistical inference from neural population responses, in *Computation and Neural Systems* (F. H. Eeckman and J. M. Bower, Eds.), Norwell, MA: Kluwer Academic, pp. 55–60. ◆

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

Knill, D. C., and Richards, W., 1996, *Perception as Bayesian Inference*, New York: Cambridge University Press.

Paradiso, M. A., 1988, A theory of the use of visual orientation information which exploits the columnar structure of striate cortex, *Biol. Cybern.* 58:35–49. ◆

Pouget, A., Zhang, K., Deneve, S., and Lathan, P., 1998, Statistically efficient estimation using population codes, *Neural computation*, 10:373–401.

Pouget, A., Zemel, R. S., and Dayan, P., 2000, Information processing with population codes, *Nature Rev. Neurosci.*, 1:125–132.

Salinas, E., and Abbott, L. F., 1994, Vector reconstruction from firing rate, *J. Computat. Neurosc.*, 1:89–108. ◆

Sanger, T. D., 1996, Probability density estimation for the interpretation of neural population codes, *J. Neurophysiol.*, 76:2790–2793.

Seung, H. S., and Sompolinsky, H., 1993, Simple model for reading neuronal population codes, *Proc. Natl. Acad. Sci. USA*, 90:10749–10753.

Zemel, R. S., Dayan, P., and Pouget, A., 1998, Probabilistic interpretation of population code, *Neural Computat.*, 10:403–430. ◆

Zhang, K., Ginzburg, I., McNaughton, B. L., and Sejnowski, T. J., 1998, Interpreting neuronal population activity by reconstruction: Unified framework with application to hippocampal place cells. *J. Neurophysiol.*, 79:1017–1044.

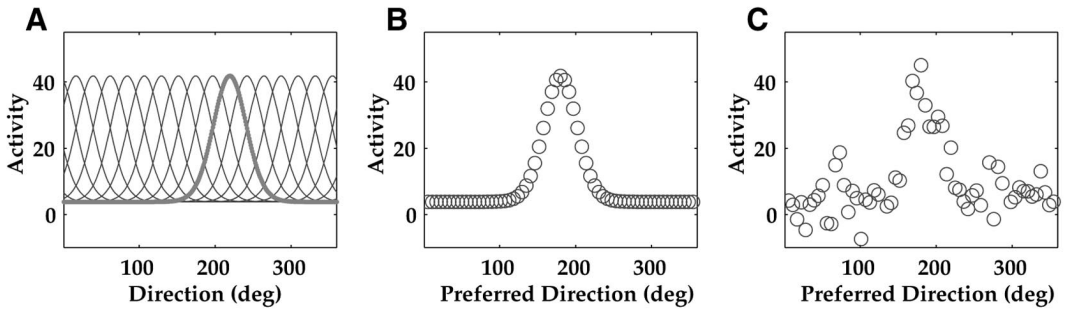


**Figure 1.** *A*, Idealized tuning curves for 16 direction-tuned neurons. *B*, Noiseless pattern of activity (○) from 64 simulated neurons with tuning curves like the ones shown in *A*, when presented with a direction of 180°. The activity of each neuron is plotted at the location of its preferred direction. *C*, Same as *B*, but in the presence of Gaussian noise.

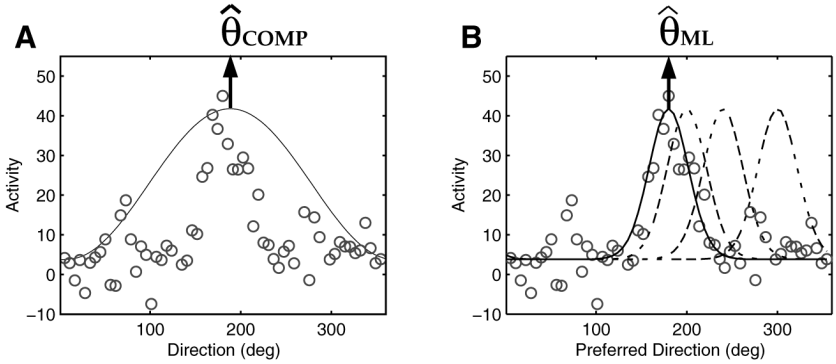


**Figure 2.** *A*, The population vector estimator uses the phase of the first Fourier component of the input pattern (solid line) as an estimate of direction. It is equivalent to fitting a cosine function to the input. *B*, The maximum likelihood estimate is found by moving an "expected" hill of activity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

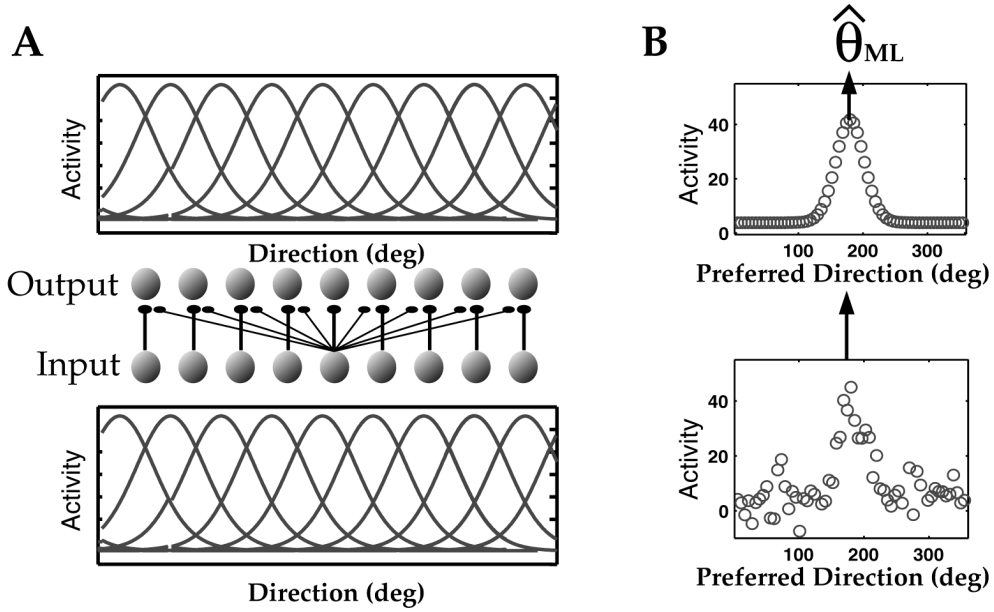


**Figure 3.** *A*, A set of units with broad tuning to a sensory variable (in this case direction) projects to another set of units also broadly tuned to the same variable. This type of mapping between population codes is very common throughout the brain. In this particular network, the output layer is fully interconnected with lateral connections, and receives feedforward connections from the input layer. *B*, Temporal evolution of the activity in the output layer for a nonlinear network. The activity in the output layer is initiated with a noisy hill generated by the input units (bottom). For an appropriate choice of weights and activation function, these activities converge eventually to a smooth hill (top), which peaks close to the location of the maximum likelihood estimate of direction, $\hat{\theta}_{ML}$. This network is performing the template-matching procedure used in maximum likelihood and illustrated in Figure 2*B*.