Population Codes: Theoretic Aspects

W J Ma and A Pouget, University of Rochester, Rochester, NY, USA

Introduction

Neurons encode in their spike trains features of the external world, internal states of the organism, or motor commands. Understanding the nature of the neural code is a key step toward understanding how the brain computes. In this article, we focus on one type of neural code, known as population codes, which are used to encode many features throughout the nervous system.

A population code is a way of representing information about a feature through the simultaneous activity of a large set of neurons sensitive to the feature. This set is called a population. A population code is useful for increasing the organism's certainty about a feature as well as for encoding multiple features simultaneously.

In most studies of population codes, the activity of a neuron is quantified either by the number of spikes it emits or by the average rate of spike emission (firing) over a given time interval. The response of all neurons in a population to a stimulus is called the population response or the population pattern of activity. It can encode information about one or more stimulus features. In the case of a visual stimulus such as a moving image, a feature could, for instance, be contrast, orientation, color, shape, speed, identity, or utility. A stimulus feature of interest is itself often also referred to as a stimulus; we follow this terminology.

An important aspect of population codes is the stochastic nature of neuronal responses. This is unlike what is found in a computer, for example, in which words are mapped deterministically onto strings of zeros and ones. There is only one word that corresponds to a given string and vice versa. In contrast, in the brain, the mapping from stimulus to population response is probabilistic. When the same stimulus is presented multiple times, the population response will in general be different each time. As a consequence, different stimuli can elicit the exact same population response, although generally with different probabilities.

The probabilistic mapping from stimulus to population response is called the encoding of the stimulus. The reverse process is decoding – obtaining a stimulus estimate from a given population response.

In this article, we first discuss encoding and give examples of population codes. Modeling the process by which neurons encode information is a prerequisite for the development of decoding techniques because decoding can often be thought of as a mathematical inverse to the encoding process. Next, we describe various decoders along with their performance and their connection to the encoding models. Finally, we address recent developments and open issues. Throughout the article, we consider an example in which the stimulus feature of interest is orientation, which takes values between 0° and 180°, but none of our conclusions is specific to this particular choice of feature.

Encoding

Encoding is the mapping from stimulus to population response. Since the response is variable, this mapping is probabilistic. Describing encoding therefore requires specifying a probability distribution over responses. This distribution will change on varying the stimulus and is therefore a conditional distribution, where the conditioning is on the stimulus. Such a distribution is also called a generative model or a noise model.

Tuning Curve Plus Noise

The response of each neuron in a population to a stimulus is typically characterized as the sum of a mean response and a noise term, both of which may depend on the stimulus. The mean response of a single neuron upon varying the stimulus value is characterized by a tuning curve. Let us consider a population consisting of N neurons, labeled by an index $i = 1, \ldots, N$. We denote the mean response of neuron i to stimulus s by $f_i(s)$. On a single trial, when a stimulus s is presented, the activity of neuron i can be written as:

$$r_i = f_i(s) + \eta_i(s)$$

where $\eta_i(s)$ is zero-mean, possibly stimulus-dependent noise. Tuning curves are usually either bell-shaped or monotonic. If the tuning curve is bell-shaped, the neuron has a preferred stimulus value (i.e., one to which its mean response is highest). Figure 1(a) shows the tuning curve and the noise of a neuron responding to orientation. In this example, the tuning curve is described by a circular Gaussian (von Mises distribution),

$$f_i(s) = Ae^{B\cos(2(s-s_i))} + C$$

where A, B, and C are constants representing amplitude, narrowness, and baseline, respectively, and s_i is the preferred orientation of neuron i. Neurons often have broad tuning curves (in our example, B would be low). This might be surprising if the goal is to encode a stimulus with high precision. However, we will see that this is the strength of population coding: even

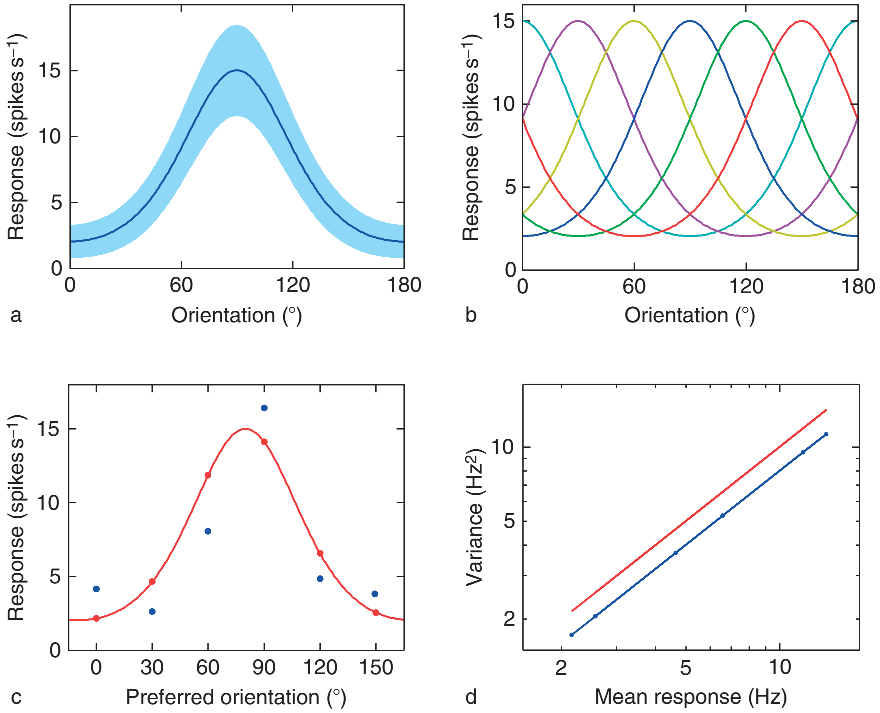

Figure 1 Example of encoding a stimulus in a population code for orientation. (a) Response of a single neuron as a function of stimulus orientation. The line represents the mean activity (tuning curve), which is described by a circular Gaussian. The maximum response was set to 15 spikes/s. This neuron has a preferred orientation of 90°. The shaded area contains responses within one standard deviation from the mean. Variability is taken to be Gaussian with variance proportional to the mean, with negative values set to zero. In this example, the constant of proportionality between mean and variance (Fano factor) is 0.8. (b) Tuning curves for all six neurons in the population. Population code typically involves thousands of neurons, but only six are shown here for clarity. The neurons' preferred orientations were chosen to be equally spaced, with a spacing of 30°. The tuning curves are broad and overlapping. (c) Population response to a contour oriented at 80°. The six neurons are labeled by their preferred orientation. The red circles indicate the mean response of the neurons to this stimulus. The red line shows the mean responses if there were many more neurons. The blue circles show an example of a response on a single trial. (d) Variance as a function of the mean response on a log–log plot. A straight line with slope 1 indicates that variance is proportional to the mean. The red line corresponds to a Poisson process, in which the variance is equal to the mean. The blue circles are the neurons in our example. Since we chose the constant of proportionality to be 0.8, these circles lie below the line of a Poisson process.

when tuning curves are broad and variability is large, stimuli can be encoded with high accuracy because every neuron contributes a piece of information. Figure 1(b) shows tuning curves for all neurons in a population of six.

Single-Neuron Response Distribution

To go beyond the description of mean response plus noise, one can describe the full response distribution of a neuron. The noise distribution is completely unrelated to the shape of the tuning curve. Common assumptions for the response distribution of a neuron are a Poisson distribution (for spike counts) and a normal distribution (for the number of spikes per second). A Poisson distribution is a one-parameter discrete distribution. The probability of a neuron i showing activity r_i in response to stimulus s is given by

$$P(r_i|s) = \frac{\mathrm{e}^{-f_i(s)} f_i(s)^{r_i}}{r_i!}$$

In a Poisson process, the probability at each time step of a neuron firing a spike is independent of the probability at the previous time step. For large means (greater than approximately 10), a Poisson distribution is well approximated by a normal distribution with variance equal to the mean.

Normally distributed neural variability has the form

$$P(r_i|s) = \frac{1}{\sigma_i(s)\sqrt{2\pi}} \exp\left[-\frac{(r_i - f_i(s))^2}{2\sigma_i(s)^2}\right]$$

where $\sigma_i(s)$ is the possibly stimulus-dependent standard deviation. The rationale behind a normal distribution is that neuronal variability is due to the combined effect of a large number of stochastic processes. Both distributions are not fully satisfactory in describing observed neuronal variability: normal distributions assign nonzero probability to negative responses, whereas Poisson distributions are limited to a ratio between variance and mean equal to 1.

This ratio, also called the Fano factor, is experimentally found to take values between 0.3 and 1.8.

The example in Figure 1 uses for each neuron Gaussian variability with variance proportional to the mean, with negative values set to zero. Neurons are assumed to be independent. Figure 1(c) shows a sample population response and the mean population response to a stimulus oriented at 90°. Figure 1(d) shows the relation between variance and mean for the variability chosen here, compared to a Poisson process.

Population Response Distribution

To describe the probabilistic response of an entire population, we need to make assumptions about the joint responses of neurons. The simplest assumption is that all neurons respond independently from each other. Then, the population response distribution is the product of the response distributions of the neurons in the population. That is, if we denote the population response by a vector $\boldsymbol{r} = \{r_i\}_{i=1,\ldots,N}$, its distribution is

$$P(\boldsymbol{r}|s) = \prod_{i=1}^{N} P(r_i|s)$$

This is also called the likelihood function of s. However, responses of neurons in a population are usually not independent of each other but, rather, exhibit noise correlations. A positive pairwise correlation implies that when the activity of one of the neurons in the pair on a given trial is above its mean, the activity of the other neuron is likely to also be above its mean. Population responses with pairwise correlations are typically modeled by a multivariate normal distribution with a nondiagonal covariance matrix. The off-diagonal elements of this matrix measure the pairwise correlations. Experimentally, it is very difficult to measure a covariance matrix for a large number of neurons because of the large amount of data needed. Some data are available for pairwise correlations and they clearly indicate that many neurons have positive correlations. However, the form of the joint probability distribution over a large population of neurons, and particularly whether this distribution can be approximated by a multivariate Gaussian distribution, is unknown.

Higher-order correlations can be described mathematically but are experimentally even more difficult to measure. Consequently, few studies have attempted to measure or investigate higher-order correlations.

Examples of Population Codes

Population codes are believed to be widespread in nervous systems. In electrophysiological experiments, a population response is normally recorded from multiple nearby electrodes in a particular brain area. Population codes have been studied most widely in sensory systems. For instance, in primary visual cortex (V1) and area V4 of the macaque, population codes exist for orientation, color, and spatial frequency. In the hippocampus in rats, a population code exists for the animal's body location. A population code for spatial location in a visual scene or of the body of the organism is also called a topographic map. The cercal system of the cricket has a population code for wind direction. The secondary somatosensory area (S2) in the macaque has population codes for surface roughness, speed, and force. The postsubiculum in rat contains a population code for head direction. Primary motor cortex (M1) in macaque uses populations coding for direction of reach. Even abstract concepts such as number appear to be encoded by population codes in the prefrontal cortex.

Another Encoding Scheme: Encoding Probability Distributions Explicitly

When the encoding schemes described in the previous section were first developed, the central assumption was that a population code encodes a single value at any given time (e.g., the orientation of a contour). Subsequently, several authors noted that population codes might encode probability distributions instead of single values. Encoding probability distributions is a particularly intriguing idea because it would allow the nervous system to perform statistical inference, a very powerful approach when performing computations in the presence of uncertainty. As discussed later, it is in fact possible to recover a probability distribution from a population code from the 'tuning curve plus noise' encoding model described previously.

There also exist other encoding models that naturally lead to decoders that return probability distributions. These are encoding schemes in which the mean response of each neuron is directly a function of the probability that the stimulus is the preferred stimulus of the neuron. This function could, for instance, be a scaled version of the probability:

$$f_i(s) = A \cdot \Pr(s = s_i)$$

where A is a constant. It could also be a linear function of its logarithm,

$$f_i(s) = [A\log\Pr(s = s_i) + B]_+$$

where A and B are constants, and $[x]_+ = x$ if $x \geq 0$ and 0 otherwise. Such codes are sometimes called explicit codes for probability distributions. In explicit codes, neural variability is merely a nuisance and its form has no particular meaning.

Decoding

In order to assess the amount of information contained in a neural population, experimentalists recording neural activity need to be able to obtain a stimulus estimate from the pattern of activity in that population on a single trial. This process is called decoding and can be regarded as the inverse of the encoding process.

In this section, we review several popular decoding techniques. As discussed later, these methods differ in the reliability of the stimulus estimates they produce. An optimal decoder is one that recovers all the information available in the population activity, where information is defined as Fisher information. Thus, using an optimal decoder is a way to determine the information available in a given population code.

The issue of decoding a neuronal population is not only relevant for experimentalists but also important for any computation performed by downstream neurons. Generation of a motor command is an example of obtaining a single estimate based on sensory information encoded in a population. The accuracy of such a computation is limited both by the accuracy with which sensory input is encoded (i.e., the amount of Fisher information available) and by the performance of the brain's decoding method (i.e., the amount of Fisher information extracted). Therefore, the issue of decoding population codes is central not only to analysis of neural data but also to our understanding of how neural computations are being performed.

Fisher Information

The next section reviews various decoding methods for population codes. These methods differ by two criteria: how much data they require and their performance. In general, the more data required, the better the results. However, how can one quantify the performance of a decoder? The performance is determined in part by the noise in the encoding process. Because of this noise, the neural activity is guaranteed to vary from trial to trial even when the stimulus is maintained constant. As a result, any decoder of this neural activity will return estimates that also vary from trial to trial. To evaluate the performance of a particular estimator, it is common to examine two quantities related to the probability distribution over the estimate: the bias and the variance. The bias, which is the difference between the average estimate and the actual stimulus value, indicates any systematic misestimation. The variance of the stimulus estimates quantifies the variability in the estimate from trial to trial and is inversely related to the reliability of the estimate. Ideally, a decoder should have zero bias and minimal variance. A general result, known as the Cramer–Rao bound, specifies the smallest variance that can be achieved by any estimator. It can be calculated from the bias of the estimator and an estimator-independent quantity called the Fisher information for the population code. Fisher information is defined in terms of the conditional probability distribution describing the population response:

$$J(s) = \left\langle \left(\frac{\partial}{\partial s} \log P(r|s) \right)^2 \right\rangle$$

where $\langle \cdot \rangle$ is the expected value taken over $P(\boldsymbol{r} \mid s)$. $P(\boldsymbol{r} \mid s)$ corresponds to the encoding process, as discussed in the previous section. Note that Fisher information is not a function of a particular response since **r** is averaged out. Instead, it measures the quality of an encoding model. Fisher information provides a lower bound on the variance $\sigma_{\hat{s}}^2$ of any estimator. Specifically, the variance of an unbiased estimator $\hat{s}$ of s is guaranteed to satisfy

$$\sigma_{\hat{s}}^2 \geq \frac{1}{J(s)}$$

A generalization of this formula exists for biased estimators. The variance of an estimator also determines the smallest change in the stimulus that can be reliably discriminated. If the variance is small, the estimator can be used to detect tiny changes in s. Accordingly, there is a natural link between Fisher information and discrimination threshold: Fisher information is inversely proportional to the square of the discrimination threshold of an ideal observer of the neural activity, or equivalently, it is proportional to the square of the sensitivity d' of an ideal observer:

$$J(s) = \frac{d'^2}{\delta s^2}$$

where d' is the sensitivity (a measure of performance), and δs is the distance between the two stimuli to be discriminated. For the case of independent Poisson neurons, Fisher is easily calculated to be

$$J(s) = \sum_{i=1}^{N} \frac{f_i'(s)^2}{f_i(s)}$$

where $f_i'(s)$ is the derivative of the tuning curve. This equation implies that the neurons that fire most (for which the stimulus is near the peak of their tuning curves) contribute little to the Fisher information because the slope of the tuning curve is small near the peak. Conversely, neurons that respond with intermediate activity, on the side of the tuning curves where the slope is high, contribute maximally to Fisher information. Intuitively, it is clear that the slope of the tuning curve matters in discrimination tasks: Where the slope is high, a small change in the stimulus will cause a large change in mean activity.

Decoders

A decoder takes the population response $\boldsymbol{r}$ as input and produces an estimate $\hat{s}$ of the stimulus s as output.

Decoders can be divided into two categories: those that take the form of neural variability into account and those that do not. The performance of various decoders discussed here is compared in Figure 2 for the example from Figure 1.

Peak response decoder or winner-take-all decoder Suppose that each neuron in the population has a preferred stimulus value. Then, when the population responds to a stimulus, a simple estimator of the stimulus is the preferred stimulus value of the neuron with the highest response:

$$\hat{s}_{\mathrm{WTA}} = s_j : j = \underset{i}{\operatorname{argmax}}\ r_i$$

This very simple decoder disregards the information in all other neurons, is highly sensitive to neuronal variability, and cannot return values intermediate between the preferred stimuli of the neurons. Therefore, it will generally not perform well. However, it only requires knowledge of the preferred stimulus of each neuron, which can be measured experimentally with very few trials.

Weighted-average or population vector decoder A better decoder is obtained by computing a weighted average of the preferred stimulus values of all neurons, with weights proportional to the responses of the respective neurons. Each neuron 'votes' for its preferred stimulus value with a strength proportional to its response:

$$\hat{s}_{\mathrm{WA}} = \frac{\sum_{i=1}^{N} r_i s_i}{\sum_{i=1}^{N} r_i}$$

Although this method performs quite well in many cases, it does not take the form of neuronal variability into account. On a circular stimulus space – for instance, when the stimulus is orientation or motion direction – the equivalent of this method is called the population vector.

Template-matching decoder We can ask the question for which stimulus value s the observed population response is closest to the mean population response generated by s. That is, we match the observed population response with a set of templates (mean population responses for different s). As an error measure, we use the sum-squared difference. This gives the decoder

$$\hat{s}_{\mathrm{TM}} = \underset{s}{\operatorname{argmin}} \sum_{i=1}^{N} (r_i - f_i(s))^2$$

This decoder also does not take the form of neuronal variability into account, but it uses more than only the preferred stimulus values of the neurons.

Maximum-likelihood decoder An important method that does take the form of neuronal variability into account is the maximum-likelihood decoder. This decoder computes the probability that a stimulus value elicited the given population response, and it selects the stimulus value for which this probability is highest:

$$\hat{s}_{\mathrm{ML}} = \underset{s}{\operatorname{argmax}}\ P(\mathbf{r}|s)$$

Figure 2(a) shows a histogram of maximum-likelihood estimates obtained from decoding 1000 trials of the population considered in Figure 1. If the number of neurons in the population is large, the maximum-likelihood estimate is unbiased and under some conditions also achieves the Cramer–Rao lower bound. If the noise is independent and normally distributed with fixed variance, the maximum-likelihood decoder is identical to the template-matching decoder. If the tuning curve is

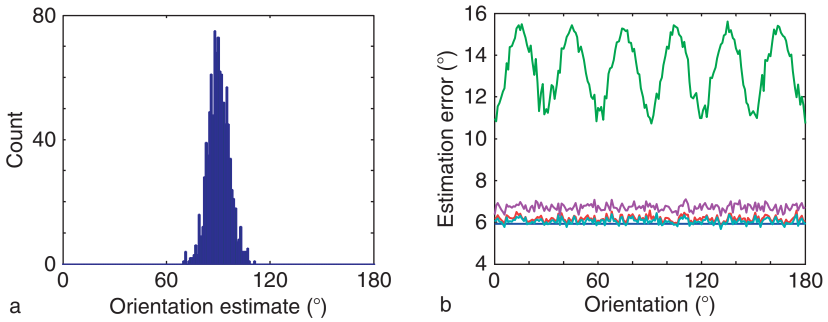

Figure 2 Decoding responses of the population of six neurons of Figure 1. (a) Histogram of maximum-likelihood estimates obtained from 1000 trials of the population responding to an orientation of 90°. The standard deviation of this histogram is the estimation error. (b) The estimation error over 1000 trials as a function of stimulus orientation for several decoders. From top to bottom, the graphs correspond to the peak response decoder (green), the template-matching decoder (purple), the population vector (red), and the maximum-likelihood decoder (cyan). The lowest graph (blue) indicates the Cramer–Rao bound. All decoders are unbiased in this example. Since actual populations usually contain many more than six neurons, estimation errors will be much smaller.

Gaussian and noise is independent Poisson, the maximum-likelihood estimate equals the weighted average.

The problem with this method is that $P(\boldsymbol{r} \mid s)$ must be known with high precision; that is, one needs to measure the tuning curves and the noise distribution of the entire neuronal population. This typically requires a large data set, especially if pairwise correlations are needed.

Bayesian decoders Bayesian decoders use Bayes' rule to express the probability of a stimulus given a response, $P(s \mid \boldsymbol{r})$, as the normalized product of the probability of this response given a stimulus, $P(\boldsymbol{r} \mid s)$, and the prior probability of the stimulus, $P(s)$:

$$P(s|\boldsymbol{r}) = \frac{P(\boldsymbol{r}|s)P(s)}{P(\boldsymbol{r})}$$

The prior probability reflects knowledge about the stimulus before the population response is elicited, and it can have been generated on the basis of previous experience. The probability distribution obtained in this way is referred to as the posterior probability distribution over the stimulus. When the number of neurons is large, this distribution is usually a narrow normal distribution. It can be collapsed onto an estimate by taking the value that has the highest posterior probability; this is called the maximum *a posteriori* (MAP) estimator:

$$\hat{s}_{\mathrm{MAP}} = \underset{s}{\operatorname{argmax}}\ P(s|\boldsymbol{r})$$

Alternatively, the posterior distribution can be collapsed onto an estimate using a loss function $L(s, s')$, which indicates the cost of reporting an estimate s' different from the true value s. The Bayesian estimate is then the value that minimizes the cost averaged over the posterior distribution:

$$\hat{s}_{\mathrm{Bayes}} = \underset{s}{\operatorname{argmin}} \int L(s,s')P(s'|\boldsymbol{r})\mathrm{d}s'$$

If the loss function is the squared error, the Bayesian estimate is the mean of the posterior distribution. For Gaussian posteriors, it is therefore equal to the MAP estimate. If the loss function is the absolute error, the Bayesian estimate is the median.

Like the maximum-likelihood estimator, this approach can be difficult to use in practice because it requires knowledge of $P(\boldsymbol{r} \mid s)$, which is data-intensive.

Neural implementation Decoding is not only of interest to scientists attempting to determine the function of a neural population but also important to the brain. The brain faces the task of generating motor output on the basis of information encoded in a population pattern of activity. The optimal strategy for many cost functions is to pick the motor command that corresponds to the MAP estimate of the stimulus. Recurrent neural networks with population codes have been shown to be able to compute these estimates in some simple cases, but there is still no general theory of Bayes-optimal computations and learning in neural circuits.

Decoding Probability Distributions

As mentioned previously, several groups have proposed that population codes may encode full probability distributions over the stimulus instead of single values $\hat{s}$. The Bayesian decoder described in the previous section falls in this category. Given an observed pattern of activity, it returns a full distribution over the encoded stimulus, denoted $P(s \mid \boldsymbol{r})$. However, this is not the only way to decode a probability distribution from a population code. Other approaches are possible depending on the nature of the encoding steps. Previously, we described several encoding models in which the mean activity of a neuron is directly related to the encoded probability of its preferred stimulus. If this is the case, the decoder should be designed to invert this encoding process. Such decoders are significantly more complicated because they ultimately involve computing a probability distribution, or an estimate, not just over s but over all possible probability distributions over s. The difficulty derives from the fact that even when s is a scalar and therefore lives in a one-dimensional space, the space of probability distribution over s has infinitely many dimensions. Several approximations have been designed to deal with this problem.

Discussion and Open Issues

We have discussed two general ways of encoding information in population codes and evidence for population codes in the brain. We described several methods for obtaining a stimulus estimate from a population response, as well as a new view in which the entire posterior distribution is computed.

An issue we have not addressed is how the performance of a population code can be improved. Population codes and their neuronal decoders can change in ways that are directly coupled to changes in behavior. For instance, attention can serve to improve the performance of a population code by increasing the amount of information that is retained between consecutive stages of processing. The phenomenon of perceptual learning, in which observers reduce the variance of their performance (i.e., their stimulus estimates) through practice, is thought to be due in part to a similar improvement and in part to an improvement of the decoder used by the brain for the task-relevant population. The mechanisms through which learning can improve population codes are the subject of ongoing research, both empirical and theoretical.

An open issue in population coding is the impact of correlated variability in a population code on the information contained in it. Part of the debate concerns identifying which quantity is suitable to measure this information. Measures in the literature are all based on either Fisher information or Shannon information. The former has the advantage that it is computed for a particular stimulus and has an interpretation in terms of the performance of an ideal observer in a discrimination task. Another part of the debate centers around the assertion that what matters for behavior is not the information that is present in correlations but, rather, the information that would be lost if correlations are neglected in decoding. Evidence shows that neglecting correlations between pairs of neurons in decoding causes only a small loss in Fisher information. An important caveat is that even small correlations can have a significant effect in large populations. In cases in which correlations are important for decoding, a further question is how they can be modeled with a tractable number of parameters – and from an organism's standpoint, how they can be learned. Analyzing empirical population data with a range of decoders will help to elucidate the role of correlations.

Most studies of population codes ignore any information that might be present in spike timing. However, spike timing is known to be important in several perceptual tasks, such as sound localization, and simultaneous firing of large numbers of neurons – synchrony – has even been proposed as an operational principle of the brain. Studies have begun to investigate population coding for dynamic stimuli, which involves decoding population spike patterns, but much work remains to be done on this topic.

Although this article only discussed issues related to coding and decoding, it is important to keep in mind that population codes in the brain are used for computation, which eventually controls behavior. Interestingly, population codes are particularly well suited to optimal nonlinear computations because they provide basis functions. These basis functions can be recombined linearly to implement a very large class of nonlinear computations. Moreover, when embedded in recurrent networks, these computations can be nearly optimal in a Bayesian sense. The latest studies on this topic focus on whether these codes can be used to perform any kind of Bayesian inference, a type of inference that provides a robust approach to computation in the presence of uncertainty.

Despite these open questions, there is little doubt that population codes are a powerful and general way of carrying information in neuronal activity. Closely integrated experimental and theoretical work is needed to fully understand how population codes perform computations and how their decoders are implemented in neural circuitry.

See also: Bayesian Cortical Models; Bayesian Models of Motor Control; Information Coding; Natural Images: Coding Efficiency; Neural Coding of Spatial Representations; Neural Coding in Primary Motor Cortex.

Further Reading

Anderson CH and van Essen DC (1994) Neurobiological computational systems. In: Zurada JM, Marks RJ, and Robinson CJ (eds.) *Computational Intelligence: Imitating Life*, pp. 213–222. Piscataway, NJ: IEEE Press.

Averbeck BB, Latham PE, and Pouget A (2006) Neural correlations, population coding and computation. *Nature Reviews Neuroscience* 7: 358–366.

Dayan P and Abbott LF (2001) *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems.* Cambridge, MA: MIT Press.

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

Ma WJ, Beck JM, Latham PE, and Pouget A (2006) Bayesian inference with probabilistic population codes. *Nature Neuroscience* 9: 1432–1438.

Pouget A, Dayan P, and Zemel RS (2003) Inference and computation with population codes. *Annual Reviews in Neuroscience* 26: 381–410.

Rao RP (2004) Bayesian computation in recurrent neural circuits. *Neural Computation* 16: 1–38.

Seung S and Sompolinsky H (1993) Simple models for reading neuronal population codes. *Proceedings of the National Academy of Sciences of the United States of America* 90: 10749–10753.

Zemel R, Dayan P, and Pouget A (1998) Probabilistic interpretation of population code. *Neural Computation* 10: 403–430.

Zhang K and Sejnowski TJ (1999) Neuronal tuning: To sharpen or to broaden? *Neural Computation* 11: 75–84.