Theoretical Systems Neuroscience

Wei Ji Ma (Baylor College of Medicine)

In the last nine lectures of this course, we will study theoretical approaches to neuroscience at the "highest", most integrated level of analysis. The focus will be on quantitative relationships between neural activity and behavior, as well as on mathematical models of behavior. Most of the time, we will use simple perceptual tasks as examples, for several reasons. In such tasks, one can isolate a single stimulus dimension with a clear physical basis (e.g. orientation or spatial location), making it relatively easy to exhaustively present well-controlled stimuli in behavioral experiments. Moreover, much is known about the physiology of basic perceptual processes, which makes it easier to make connections between neural activity and behavior. Finally, we will often be taking a normative point of view, according to which the brain is regarded as performing optimal inference on the state of the world. It seems plausible that if any computations are optimized, they are the ones that have had the longest evolutionary history. However, the theory will often be more general than the specific tasks we study.

Table of Contents

1 Population encoding and decoding 2
1.1 Encoding 2
1.2 Decoding 4
1.2.1 Winner-take-all decoder 4
1.2.2 Center-of-mass or population vector decoder 5
1.2.3 Template-matching decoder 5
1.2.4 Maximum-likelihood decoder 5
1.2.5 Bayesian decoders 6
1.2.6 Sampling from the posterior 7
1.3 How good are different decoders? 7
1.3.1 Cramér-Rao bound 8
1.3.2 Fisher information 10
1.3.3 Goodness of decoders 12
1.3.4 Neural implementation 12
1.4 Decoding probability distributions 13
1.4.1 Other forms of probabilistic coding 13
1.4.2 Discrete variables 14

1 Population encoding and decoding

The main function of the brain is to make use of perceptual input to generate relevant behavioral output. In order to do this, it needs to create and manipulate informative representations of the world. To start with the basics, we only focus on a tiny aspect of the world, namely a single feature of a single object. This could for example be the orientation of a line segment, the spatial location of a dot, the direction of motion of a tennis ball, or the color of a surface. We will call this a stimulus. For many such stimuli, there are many neurons in the brain that respond differentially to different values of it. Moreover, different neurons tend to respond in different ways to the same stimulus. It is even the case that the same neuron, when presented with the same stimulus many times, will exhibit a whole range of responses. In this lecture, we will quantify these notions.

A *population code* is a way of representing information about a stimulus 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 animal's certainty about a feature, as well as for encoding multiple features at once. Encoding is how a stimulus gives rise to patterns of activity (in a stochastic manner), decoding is the reverse process, by which a neural population is "read out", either by an experimenter or by downstream neurons, to produce an estimate of the stimulus.

Population codes are believed to be widespread in the nervous system. 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. The cercal system of the cricket has a population code for wind direction. Secondary somatosensory area (S2) in the macaque has population codes for surface roughness, speed, and force. The post-subiculum 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 appears to be encoded by population codes in the prefrontal cortex.

1.1 Encoding

Let s be the stimulus (or a specific value of the stimulus). We will denote the response of a neuron to the stimulus by r. For a brief stimulus, this is the total number of spikes elicited. For a sustained stimulus, it can be the total number of spikes in a certain time interval. When s is presented many times, different values of r will be recorded. We will denote the mean response by $f(s)$. Unlike r, $f(s)$ is not necessarily an integer. As a function of s, $f(s)$ is called the tuning curve of the neuron. It typically is bell-shaped (for stimuli like orientation, see Slide 9) or monotonic (Slide 10). When it is bell-shaped, then the mode of the function is called the preferred stimulus of the neuron.

The variability of r around its mean in response to s can often reasonably be described as a Poisson process with mean $f(s)$. That means that it is drawn from the following distribution:

$$p(r \mid s) = \frac{e^{-f(s)} f(s)^r}{r!}. \tag{1.1}$$

Note that this is a conditional probability distribution: we are not interested in the distribution of responses in general, but only in response to a specific stimulus. The variance of a Poisson-distributed variable is equal to its mean, which is not completely consistent with measurements. In most cortical neurons, the Fano factor, which is the ratio between variance and mean, is found to be more or less constant over a range of mean activities, but with a value anywhere between 0.3 and 1.8.

Different neurons will in general have different tuning curves. Suppose we have a population of n neurons. We label the neurons with an index i, which runs from 1 to n. The tuning curve of the i'th neuron is denoted $f_i(s)$. A set of such tuning curves is shown in Slide 16. In this example, the preferred orientations are distributed uniformly over stimulus space, but they do not need to be, and in general they are not.

Instead of choosing a Poisson distribution to describe neural variability, one can use a normal distribution:

$$p(r \mid s) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(r-f(s))^2}{2\sigma^2}}. \tag{1.2}$$

For large values of the mean, a Poisson distribution is very similar to a normal distribution. A normal distribution can be made into a more realistic description of variability by taking its variance to be proportional or equal to its mean, just as is the case for the Poisson distribution. However, for small values of the mean, any normal distribution runs into problems since it is defined on the entire real line, including negative values, whereas spike counts are always non-negative. One can cut it off at zero, but then the distribution loses the nice properties of the normal distribution.

On a single trial, the response of a single neuron is denoted r_i, and the population response can be written as a vector $\mathbf{r} = (r_1,\ldots,r_N)$. We can characterize the variability of the population response upon repeated presentations of stimulus s by a distribution $p(\mathbf{r}|s)$. This is called the *response distribution* or the *response distribution*, although the word "noise" might be misleading. In more generality, a mathematical description of how the observations are generated probabilistically by a source variable is called a *generative model*.

The simplest assumption we can make to proceed is that variability is independent between neurons. This means that for a given s, the probability distribution from which a spike count in one neuron is drawn is unrelated to the activity of other neurons. In that case, the response distribution of the population is a product distribution:

$$p(\mathbf{r}\,|\,s) = \prod_{i=1}^{N} p(r_i\,|\,s)\,. \tag{1.3}$$

This is called conditional independence, because the probabilities are all for a given *s*. Neurons in real populations are typically not conditionally independent, but exhibit correlations. There is ongoing debate about the importance of the correlations in decoding (see lecture 2).

Exercise 1: When the neurons in a population are Poisson-distributed, $p(\mathbf{r}|s)$ can be written in the form $p(\mathbf{r}\,|\,s) = \frac{\varphi(\mathbf{r})}{\eta(s)} e^{\mathbf{h}(s)\cdot\mathbf{r}}$. Identify the functions $\varphi(\mathbf{r})$, $\eta(s)$, and $\mathbf{h}(s)$.

Recently, a generalization of independent Poisson variability has been proposed that takes the general form $p(\mathbf{r}\,|\,s) = \frac{\varphi(\mathbf{r})}{\eta(s)} e^{\mathbf{h}(s)\cdot\mathbf{r}}$ (exponential family with linear sufficient statistics), but without any specific choices for $\varphi(\mathbf{r})$, $\eta(s)$, and $\mathbf{h}(s)$. The function $\mathbf{h}(s)$ would then be related to the tuning curves and the covariance matrix of the population. This proposal has not yet been fully tested.

1.2 Decoding

Based on a pattern of activity **r**, the brain often has to reconstruct what was the stimulus *s* that gave rise to **r**. This is called decoding, estimating, or reading out *s*. There are many ways of doing this, some of which are better than others. We will discuss some of the most common methods. We keep in mind that decoding is of interest both to downstream neurons (e.g. to eventually generate a motor command) and to experimenters trying to decode their recorded patterns of activity.

1.2.1 Winner-take-all decoder

Suppose that each neuron in the population has a preferred stimulus value. Then, a simple estimator of the stimulus is the preferred stimulus value of the neuron with the highest response:

$$\hat{s} = s_j : \; j = \underset{i}{\operatorname{argmax}}\, r_i\,. \tag{1.4}$$

This decoder disregards the information in all neurons but one (except the fact that their responses are smaller), is sensitive to neuronal variability (the neuron that fires most on average does not necessarily fire most on a single trial), and cannot return values intermediate between the preferred stimuli of the neurons. However, it only requires knowledge of the preferred stimulus of each neuron, which can be measured experimentally with very few trials.

1.2.2 Center-of-mass 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:

$$\hat{s} = \frac{\sum_{i=1}^{N} r_i s_i}{\sum_{i=1}^{N} r_i}. \tag{1.5}$$

While 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. It has been applied to experimental data, for instance by Georgopoulos et al. (Georgopoulos, Kalaska et al. 1982) to decode movement direction from population activity in primate motor cortex.

1.2.3 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

$$\hat{s} = \underset{s}{\operatorname{argmin}} \sum_{i=1}^{N} \left(r_i - f_i(s) \right)^2. \tag{1.6}$$

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

Exercise 2: Show that under certain conditions, this is equivalent to

$$\hat{s} = \underset{s}{\operatorname{argmax}}\ \mathbf{r} \cdot \mathbf{f}(s). \tag{1.7}$$

1.2.4 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 selects the stimulus value for which this probability is highest:

$$\hat{s} = \underset{s}{\operatorname{argmax}}\ p(\mathbf{r} \mid s). \tag{1.8}$$

If neural variability is independent Poisson and we assume that $\sum_i f_i(s)$ is approximately independent of *s* (which is the case when the tuning curves are densely and more or less uniformly spaced in stimulus space), then this becomes

$$\hat{s} = \underset{s}{\operatorname{argmax}} \sum_{i=1}^{N} r_i \log f_i(s). \tag{1.9}$$

Exercise 3. If neuronal noise is independent and normally distributed with fixed variance, the maximum-likelihood decoder is equivalent to a decoder we already know. Which one?
Exercise 4. Under a different response distribution, the maximum-likelihood decoder is equivalent to another one of the decoders discussed above. Which response distribution?

From the point of view of an experimenter, a problem with the maximum-likelihood decoder is that $p(\mathbf{r}|s)$ must be known with high precision, that is, one needs to measure the tuning curves and the response distribution of the entire neuronal population. This typically requires a large data set, especially if pairwise correlations are needed.

1.2.5 Bayesian decoders

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

$$p(s|\mathbf{r}) = \frac{p(\mathbf{r}|s)\,p(s)}{p(\mathbf{r})}. \tag{1.10}$$

The prior probability reflects knowledge about the stimulus before the population response is elicited, and 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 (see Slide 29). 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) decoder (also the mode of the posterior):

$$\hat{s} = \underset{s}{\operatorname{argmax}}\, p(s|\mathbf{r}). \tag{1.11}$$

Alternatively, the posterior distribution can be collapsed onto an estimate using a cost function $C(\hat{s}, s)$ (usually symmetrical in *s* and $\hat{s}$), which indicates the cost of reporting an estimate $\hat{s}$ different from the true value *s*. The Bayesian estimate is then the value that minimizes the expected cost. "Expected" means under the beliefs about the stimulus as expressed in the posterior distribution. Therefore, this estimator minimizes the average of $C(\hat{s}, s)$ under $p(s|\mathbf{r})$:

$$\hat{s} = \underset{s}{\operatorname{argmin}} \int C(s, s')\, p(s' \mid \mathbf{r})\, ds'. \tag{1.12}$$

If the loss function is the squared error, $C(\hat{s}, s) = (\hat{s} - s)^2$, the Bayesian estimate is the mean of the posterior distribution.

Exercise 6: Prove this.
Exercise 7: Compute the Bayesian estimate under the absolute error, $C(\hat{s}, s) = \lvert \hat{s} - s \rvert$.
Exercise 8: What is the cost function corresponding to the MAP decoder? (Hint: there is one, but it's weird.)

For Gaussian posterior distributions, both Bayesian estimates are equal to the maximum-a-posteriori estimate. However, posterior distributions are not always Gaussian, as we will see when we study more complex perceptual tasks (Lecture 5).

Like the maximum-likelihood estimator, this approach can be difficult to use in practice because it requires knowledge of $p(\mathbf{r}|s)$, which is data-intensive. Zhang et al. (Zhang, Ginzburg et al. 1998) applied a Bayesian decoder to recordings from hippocampal place cells and found that it outperformed the population vector decoder.

1.2.6 Sampling from the posterior

A final possible read-out is to draw an estimate of the stimulus from the posterior distribution:

$$p(\hat{s} \mid \mathbf{r}) = p(s = \hat{s} \mid \mathbf{r}). \tag{1.13}$$

In contrast to the previous decoders, this is a stochastic decoder: the same $\mathbf{r}$ can produce different values of $\hat{s}$. This type of decoding is related to Monte-Carlo methods and is particularly relevant when the posterior cannot be computed exactly.

Exercise 9: The sampling decoder will exhibit more variability than the deterministic Bayesian decoders. Why?

1.3 How good are different decoders?

Given that there are so many possible decoders, how can we objectively evaluate how good each of them is? Imagine you have a large number of population patterns of activity all generated by the same value of s. For each pattern, you apply your decoder of interest to obtain an estimate $\hat{s}$. Now look at the distribution of estimates, $p(\hat{s} \mid s)$. There are several criteria for what makes a decoder good. First, you would like the mean estimate to be equal to

the true stimulus value, i.e. $\langle \hat{s} \rangle = s$. Here, the average $\langle \cdot \rangle$ is in principle over $p(\hat{s} \mid s)$ but can also be regarded as one over $p(\mathbf{r} \mid s)$, since $\hat{s}$ is a function of $\mathbf{r}$ for all deterministic decoders (but not for the sampling decoder). The difference between the mean estimate and the true stimulus value is called the *bias* of the estimator, and it may depend on s:

$$b(s) \equiv \langle \hat{s} \rangle - s \,. \tag{1.14}$$

An estimator is called unbiased if $b(s)=0$ for all s. It is not very difficult for a decoder to be approximately unbiased. Most of the decoders we described in the previous section are unbiased in common situations. However, not all unbiased decoders are equally good, since not only the mean matters, but also the variance. The smaller the variance, the better the decoder. Whereas bias can be equal to zero, this is not true for the variance – because of variability in patterns of activity, it is impossible for an unbiased decoder to have zero variance. (It is easy for a *biased* decoder to have zero variance: just take one that ignores the data and always produces the same value. This decoder has no variability, but it is severely biased for all values of s but one.) It turns out that there is a fundamental lower bound on the variance of an unbiased decoder. This is a famous result in estimation theory, known as the Cramér-Rao inequality. We will go through it in some detail because it is an important notion in population coding.

Exercise 10 (bonus): The winner-take-all and center-of-mass decoders explicitly make use of the preferred stimuli of the neurons. This can be done for bell-shaped tuning curves, but when tuning curves are monotonic, as in Slide 10, there is no unambiguous notion of a preferred stimulus. Can these decoders be salvaged somehow? For the sake of this exercise, you can assume a particular form of monotonic tuning curve, for example sigmoids.

1.3.1 Cramér-Rao bound

Based on the response distribution $p(\mathbf{r} \mid s)$, one can define a quantity called the *Fisher information* that the population contains about s:

$$I(s) = -\left\langle \frac{\partial^2}{\partial s^2} \log p(\mathbf{r} \mid s) \right\rangle, \tag{1.15}$$

where ∂ denotes a partial derivative and the average $\langle \cdot \rangle$ is now over $p(\mathbf{r} \mid s)$. Fisher information can alternatively be expressed as

$$I(s) = \left\langle \left(\frac{\partial}{\partial s} \log p(\mathbf{r} \mid s) \right)^2 \right\rangle. \tag{1.16}$$

Exercise 11: Prove this.

We will first prove an identity that will come in helpful:

$$\begin{aligned}\left\langle \frac{\partial}{\partial s}\log p(\mathbf{r}\,|\,s)\right\rangle &= \int\left(\frac{\partial}{\partial s}\log p(\mathbf{r}\,|\,s)\right)p(\mathbf{r}\,|\,s)\,d\mathbf{r}\\ &= \int\left(\frac{1}{p(\mathbf{r}\,|\,s)}\frac{\partial}{\partial s}p(\mathbf{r}\,|\,s)\right)p(\mathbf{r}\,|\,s)\,d\mathbf{r}\\ &= \int\frac{\partial}{\partial s}p(\mathbf{r}\,|\,s)\,d\mathbf{r}\\ &= \frac{\partial}{\partial s}\int p(\mathbf{r}\,|\,s)\,d\mathbf{r}\\ &= \frac{\partial}{\partial s}1\\ &= 0.\end{aligned} \tag{1.17}$$

Now we are ready to derive the Cramér-Rao bound. To do this, we invoke the Cauchy-Schwarz inequality, which states that for two random variables *X* and *Y*, the following holds:

$$\mathrm{cov}(X,Y)^2 \le \mathrm{var}(X)\,\mathrm{var}(Y). \tag{1.18}$$

In our case, we use $X = \hat{s}(\mathbf{r})$ and $Y = \frac{\partial}{\partial s}\log p(\mathbf{r}\,|\,s)$. Then we can calculate the covariance in the left-hand side, using the fact that $\langle Y\rangle = 0$ (Eq. (1.17)):

$$\begin{aligned}\mathrm{cov}(X,Y) &= \left\langle \left(X - \langle X\rangle\right)\left(Y - \langle Y\rangle\right)\right\rangle\\ &= \left\langle \left(X - \langle X\rangle\right)Y\right\rangle\\ &= \langle XY\rangle - \langle X\rangle\langle Y\rangle\\ &= \langle XY\rangle\\ &= \left\langle \hat{s}(\mathbf{r})\frac{\partial}{\partial s}\log p(\mathbf{r}\,|\,s)\right\rangle\end{aligned} \tag{1.19}$$

Now we can use the helpful result we derived earlier, Eq. (1.17), and evaluate further:

$$
\begin{aligned}
\operatorname{cov}(X,Y) &= \left\langle \hat{s}(\mathbf{r}) \frac{\partial}{\partial s} \log p(\mathbf{r}\,|\,s) \right\rangle \\
&= \int \hat{s}(\mathbf{r}) \left(\frac{\partial}{\partial s} \log p(\mathbf{r}\,|\,s) \right) p(\mathbf{r}\,|\,s)\, d\mathbf{r} \\
&= \int \hat{s}(\mathbf{r}) \frac{1}{p(\mathbf{r}\,|\,s)} \left(\frac{\partial}{\partial s} p(\mathbf{r}\,|\,s) \right) p(\mathbf{r}\,|\,s)\, d\mathbf{r} \\
&= \int \hat{s}(\mathbf{r}) \left(\frac{\partial}{\partial s} p(\mathbf{r}\,|\,s) \right) d\mathbf{r} \\
&= \frac{\partial}{\partial s} \int \hat{s}(\mathbf{r})\, p(\mathbf{r}\,|\,s)\, d\mathbf{r} \\
&= \frac{\partial}{\partial s} \langle \hat{s} \rangle .
\end{aligned}
\tag{1.20}
$$

We use the fact that $\hat{s}$ is unbiased, $\langle \hat{s} \rangle = s$, to obtain $\operatorname{cov}(X,Y) = 1$. Next, we evaluate the second factor on the right-hand side of Eq. (1.18):

$$
\operatorname{var}(Y) = \langle Y^2 \rangle = \left\langle \left(\frac{\partial}{\partial s} \log p(\mathbf{r}\,|\,s) \right)^2 \right\rangle = I(s). \tag{1.21}
$$

Combining Eqs. (1.20), (1.21), and (1.18), we find

$$
\operatorname{Var} \hat{s}(\mathbf{r}) \geq \frac{1}{I(s)}. \tag{1.22}
$$

This is the Cramér-Rao bound (or inequality). It states that no estimator $\hat{s}$ can achieve a variance that is smaller than the inverse of the Fisher information. An estimator that has the smallest possible variance is called an *efficient estimator*. The Cramér-Rao bound can be generalized to multidimensional (vector) variables.

Exercise 12: In this derivation, we have assumed that the estimator is unbiased. Show that for an estimator with given bias $b(s)$, the Cramér-Rao bound takes the following form:

$$
\operatorname{Var} \hat{s}(\mathbf{r}) \geq \frac{\left(1 + b'(s)\right)^2}{I(s)}.
$$

1.3.2 Fisher information

Since Fisher information puts a hard limit on the performance of any possible decoder, it is *a decoder-independent measure of the information content of a population of neurons* (which is why it is called "information" in the first place). Here, the understanding is that the population is characterized by its response distribution. Fisher information reflects the maximum amount of information that can be extracted from a population.

The variance of an estimator 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 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:

$$I(s) = \frac{d'^2}{\delta s^2}, \tag{1.23}$$

where d' is the sensitivity (a measure of performance) and δs is the distance between the two stimuli to be discriminated. Fisher information is subject to the data processing inequality, which states that no operation on the data (in our case population patterns of activity) can increase Fisher information.

Now that we have established Fisher information as a measure of the information content of a population, we can try to compute it in a concrete situation. For independent Poisson noise,

$$p(\mathbf{r} \mid s) = \prod_{i=1}^{N} \frac{e^{-f_i(s)} f_i(s)^{r_i}}{r_i!}. \tag{1.24}$$

Therefore,

$$\begin{aligned}
&\log p(\mathbf{r} \mid s) = \sum_{i=1}^{N} \left(-f_i(s) + r_i \log f_i(s) - \log r_i! \right) \\
&\frac{\partial}{\partial s} \log p(\mathbf{r} \mid s) = \sum_{i=1}^{N} \left(r_i - f_i(s) \right) \frac{\partial \log f_i}{\partial s} \\
&\frac{\partial^2}{\partial s^2} \log p(\mathbf{r} \mid s) = \sum_{i=1}^{N} \left(r_i - f_i(s) \right) \frac{\partial^2 \log f_i}{\partial s^2} - \sum_{i=1}^{N} f_i'(s) \frac{\partial \log f_i}{\partial s},
\end{aligned} \tag{1.25}$$

and Fisher information is

$$\begin{aligned}
I(s) &= -\left\langle \frac{\partial^2}{\partial s^2} \log p(\mathbf{r} \mid s) \right\rangle \\
&= -\sum_{i=1}^{N} \left\langle r_i - f_i(s) \right\rangle \frac{\partial^2 \log f_i}{\partial s^2} + \sum_{i=1}^{N} f_i'(s) \frac{\partial \log f_i}{\partial s} \\
&= \sum_{i=1}^{N} \frac{f_i'(s)^2}{f_i(s)}.
\end{aligned} \tag{1.26}$$

The squares of the derivatives of the tuning curves are very characteristic for Fisher information. The amount of information in a population is strongly determined by the slopes of the tuning curves. One way to see this is by considering the Fisher information as the sum of contributions from individual neurons, $I(s) = \sum_{i=1}^{N} I_i(s)$, and plot the contribution per neuron.

This is shown in Slide 36. The largest contribution to the information about a given stimulus *s* does not come from neurons that respond on average most vigorously to *s*, but from those whose tuning curve is steepest at *s.* The intuition is that the latter neurons are much more sensitive to small changes in *s* (see Slide 37).

1.3.3 Goodness of decoders

Now it is possible to ask which of the decoders described in Section 1.2 is "better", since we can compute the magnitude of their bias as well as their variance. This is typically done numerically, by simulating a large number of population patterns of activity using the response distribution. However, there are some general results that are helpful here. It turns out that in the limit of a large number of observations (in our case, many spikes), the maximum-likelihood estimate is the "best possible" one in the sense that it is both unbiased and efficient. Moreover, its distribution is approximately normal in this limit. Since it is also efficient, it means that the variance of maximum-likelihood estimates for a given *s* satisfies:

$$\sigma_{\mathrm{ML}}^2 = \frac{1}{I(s)}. \tag{1.27}$$

Exercise 13 (bonus): Look up and understand the proof that the maximum-likelihood estimator is asymptotically unbiased and efficient.

Other decoders will have a larger bias and/or a larger variance than the maximum-likelihood decoder, although in simple situations, some of them may come very close. Typically, the winner-take-all decoder and the sampling decoder are rather poor, but one might choose them for their computational advantages.

Exercise 14: Simulate a small population of independent Poisson neurons with Gaussian tuning curves. Evaluate the winner-take-all, template-matching, weighted-averaging, sampling, and maximum-likelihood decoders in terms of bias and variance. Plot in a single figure estimator variance as a function of the true value of *s*, along with the Cramér-Rao bound.

1.3.4 Neural implementation

So far, we have discussed many decoders from an abstract perspective. However, the brain itself also has to do decoding, for example in generating a response to a stimulus. Therefore, if we want to know whether a particular decoder is used in performing a perceptual task, the question needs to be answered how it can be implemented in neural networks. Fortunately, this problem has been solved for the maximum-likelihood decoder. Under certain assumptions on the form of neural variability, a line attractor network can turn a noisy population pattern of

activity into a smooth pattern that peaks at the maximum-likelihood estimate (Deneve, Latham et al. 1999).

The winner-take-all decoder can easily be implemented using a nonlinearity (to enhance the maximum activity) and global inhibition (to suppress the activity of other neurons). Neural implementations of the other decoders we discussed are not known.

Exercise 15 (bonus): Implement a winner-take-all decoder. Use an output population of firing-rate neurons that receives a noisy population pattern of activity as input. The output neuron receiving the highest input should remain active, while the others go silent.
Exercise 16: Based on the fact that the winner-take-all decoder can be implemented, what can you say about a possible implementation of the template-matching decoder? (Hint: use Eq. (1.7).

1.4 Decoding probability distributions

So far we have considered the process of obtaining a single estimate of the stimulus, $\hat{s}$, from a population pattern of activity **r**. The underlying assumption is that a neural population encodes only a single value at a given time. This has been the dominant notion of neural coding in systems neuroscience over the past decades. However, starting with the work by Foldiak (Foldiak 1993) and Sanger (Sanger 1996), the idea has developed that population codes may encode full probability distributions over the stimulus, instead of single values $\hat{s}$. The Bayesian decoders described in Section 1.2.5 fall in this category. Given an observed pattern of activity, they return a full distribution over the encoded stimulus, denoted $p(s|\mathbf{r})$. So far, we have always turned this distribution into a single estimate by taking the mode, mean, or median, or by sampling. However, by doing this, we lost valuable information. Namely, a probability distribution over the stimulus also reflects the *uncertainty* about the stimulus, or in other words, one's confidence about the judgment of the stimulus. Broader probability distributions mean more uncertainty, even though the stimulus estimate $\hat{s}$ may be the same. It is particularly important to retain information about uncertainty if the task does not require to make an immediate judgment about the stimulus, but instead requires to process it further, as is the case in many perceptual tasks. Encoding probability distributions would allow the nervous system to perform probabilistic (Bayesian) inference, a very powerful approach to performing computations in the presence of uncertainty.

1.4.1 Other forms of probabilistic coding

Bayesian decoding 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. In the encoding section, 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 comes 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.

There also exist other *encoding* models that lead to decoders that return probability distributions (Pouget, Dayan et al. 2003). These are ones 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:

$$f_i(s) = A \cdot \Pr(s = s_i) \tag{1.28}$$

with *A* a constant (see Slide 41). It could also be a linear function of its logarithm,

$$f_i(s) = \left[A \log \Pr(s = s_i) + B \right]_+, \tag{1.29}$$

with *A* and *B* 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.

1.4.2 Discrete variables

So far, we have discussed encoding and decoding of continuous variables. However, some variables only take on discrete values. This could be because they are discrete in nature (binary variables such as "was the target present?", or numerosity), or because the experimental set-up only allows for a discrete number of responses (e.g. "in which of *N* directions was the motion"?). Population codes for discrete stimuli are treated in much the same way as for continuous stimuli. Posterior distributions are now discrete, i.e. $p(s=s_i|\mathbf{r})$, where *i* labels the possible values of the stimulus. In particular, if the stimulus is binary (let's say $s=0,1$), the posterior distribution on a single trial is determined by a single number, namely $p(s=0|\mathbf{r})$ or $p(s=1|\mathbf{r})$, since they sum to 1. Applying Bayes' rule to the former probability gives:

$$\begin{aligned} p(s=0|\mathbf{r}) &= \frac{p(\mathbf{r}|s=0)\,p(s=0)}{p(\mathbf{r})} \\ &= \frac{p(\mathbf{r}|s=0)\,p(s=0)}{p(\mathbf{r}|s=0)\,p(s=0) + p(\mathbf{r}|s=1)\,p(s=1)}, \end{aligned} \tag{1.30}$$

and similarly for $p(s=1|\mathbf{r})$. It is often convenient to express the posterior by taking the logarithm of the ratio of the posterior probabilities of the alternatives:

$$L = \log \frac{p(s=1|\mathbf{r})}{p(s=0|\mathbf{r})} = \log \frac{p(\mathbf{r}|s=1)}{p(\mathbf{r}|s=0)} + \log \frac{p(s=1)}{p(s=0)} \quad (1.31)$$

Here, we have used the fact that the normalization, $p(\mathbf{r})$, is common to both posterior probabilities and therefore drops out. L is called the log posterior ratio or log odds. The first term in the last line is called the log likelihood ratio, and the second term is called the log prior ratio. All three expressions take values on the entire real line. Given L, one can reconstruct $p(s=0|\mathbf{r})$ and $p(s=1|\mathbf{r})$. The absolute value of L is a measure of the amount of certainty that the stimulus is either alternative. When there is equal evidence for each alternative, $L=0$.

The deterministic Bayesian decision rules we discussed above (mode, mean, median) all amount to the same "probability maximizing" rule when the stimulus is discrete, namely to pick $s=1$ when $L>0$ and $s=0$ when $L<0$. The rule is not defined when $L=0$, but a possible strategy is to arbitrarily pick $s=0$ or $s=1$. The absolute value of L is also a measure of confidence about the decision.

Sampling from a posterior over a discrete variable s means to flip a biased coin on each trial, which produces the response "$s=1$" with probability $p(s=1|\mathbf{r})$. For example, when $p(s=1|\mathbf{r})=0.55$, the deterministic rules would respond "$s=1$" for sure, whereas sampling would lead to a probability of 55% to respond "$s=1$". Sampling from the posterior over a discrete variable is also called *probability matching*. Though suboptimal, it is of interest because humans and animals seem to use this strategy in many decision tasks. Whether observers "match" or "maximize", and under what conditions, has been the subject of debate.

For discrete variables, bias and variance of a decoder are not well-defined. Instead, performance of a decoder is measured by two numbers, percentage correct in each of both stimulus conditions. The best possible decoder is the Bayesian decoder (with maximizing), provided that the prior distribution is chosen in accordance with the experimental frequencies of the stimuli.

Deneve, S., P. Latham, et al. (1999). "Reading population codes: a neural implementation of ideal observers." Nature Neuroscience **2**(8): 740-745.

Foldiak, P. (1993). The 'ideal homunculus': statistical inference from neural population responses. Computation and Neural Systems. F. Eeckman and J. Bower. Norwell, MA, Kluwer Academic Publishers**:** 55-60.

Georgopoulos, A., J. Kalaska, et al. (1982). "On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex." Journal of Neuroscience **2**(11): 1527-1537.

Pouget, A., P. Dayan, et al. (2003). "Inference and Computation with Population Codes." Annual Review of Neuroscience.

Sanger, T. (1996). "Probability density estimation for the interpretation of neural population codes." Journal of Neurophysiology **76**(4): 2790-3.

Zhang, K., I. Ginzburg, et al. (1998). "Interpreting neuronal population activity by reconstruction: unified framework with application to hippocampal place cells." Journal of Neurophysiology **79**(2): 1017-44.