Lecture 10: Statistical inference and learning

10.1 Learning

- We'll do learning first, just for fun: Both use past information to affect future performance.
- What role there could be for learning given our approach since we find weights analytically? Possibilities:
  1. learning can help fine-tune models generated this way;
  2. we can compare and contrast learned with analytically found weights to see if they are the same or different, possibly giving us insight into the nature of learning; and
  3. examining the role of learning helps highlight both new challenges for, and the inherent strengths of, this approach.
- Comments on 1.: the weights found using the techniques presented in the part two of this book are typically based on rate model approximations to spiking neurons. How accurate these approximations are depends on the spiking model being used. While there is significant leeway in the framework for making the rate model approximations more or less accurate, there will always be some degree of inaccuracy in the approximation. Thus, the connection weights found using the framework are probably best considered to be good first guesses at a set of weights that perform the desired function. Learning can thus be used to subsequently fine-tune these weights. This would be particularly true in cases like the neural integrator, where there is evidence that an explicit error signal is available to the system for updating subsequent behavior [**?**].
- Let's consider the other possibilities by example.

10.1.1 Learning a communication channel

- Recall that in this network $a_i(x)$ simply drives $b_j(y)$, with no transformation taking place. The purpose of the network is to simply communicate the signal, x, in the original population to the subsequent one; i.e., we want $b_j(y = \hat{x})$ to represent just what $a_i(x)$ does.
- A way to formalize this as learning is to note that maximizing the variance of the responses in the receiving population allows it to carry the most information it can about incoming signals. If it carries exactly the same information as is available from the sending population, we have a communication channel. To maximize the variance, we need to maximize

$$E = \frac{1}{2} \sum_j \left\langle \left(b_j(x) - \bar{b}_j(x) \right)^2 \right\rangle_x . \tag{10.1}$$

- Noting that $b_j(x) = G_j\left[\sum_i \omega_{ji} a_i(x) + J_j^{bias}\right]$ as usual, and taking the derivative of (**??**) with respect to ω_{ji} gives

$$\begin{aligned}\frac{dE}{\delta\omega_{ji}} &= \frac{dG_j\left[\xi\right]}{d\xi}\frac{d\xi}{d\omega_{ij}}\left(b_j(x) - \bar{b}_j(x)\right) \\ \frac{dE}{\delta\omega_{ji}} &= \frac{dG_j\left[\xi\right]}{d\xi}a_i(x)\left(b_j(x) - \bar{b}_j(x)\right)\end{aligned}$$

  where $\xi = \sum_i \omega_{ji} a_i(x) + J_j^{bias}$.

- The G_j term means that the rate of change of the activity of the neuron for input x is important. We can replace it with the simpler term $(b_j(x) > 0)$ as an approximation. This evaluates to 1 when the neuron is active and 0 when it is not. (This does not affect the final results of learning, although it can slightly slow down the learning process.)

- $\bar{b}_j(x)$ is found by keeping a running mean, i.e.,

$$\bar{b}_j(x; t + dt) = (1 - \epsilon)\bar{b}_j(x; t) + \epsilon b_j(x; t)$$

- We use the standard 'delta rule' approach to write the learning rule explicitly:

$$\begin{aligned}\Delta\omega_{ij} &= -\kappa\frac{dE}{\delta\omega_{ji}} &(10.2)\\ &= -\kappa\left(b_j(x) > 0\right)a_i(x)\left(b_j(x) - \bar{b}_j(x)\right), &(10.3)\end{aligned}$$

  where κ is the learning rate.

- This is a typical Hebbian learning rule. That is, it is a *local* learning rule that will construct the expected representation in the b_j population. (see slide)

- Given our previous analyses, it is natural to ask if we can decompose them into their encoding and decoding components. Recall that the general form of the weights we find is

$$\omega_{ji} = \tilde{\phi}_i \phi_j$$

  or, in matrix form,

$$\boldsymbol{\omega} = \tilde{\boldsymbol{\phi}} \odot \boldsymbol{\phi}$$

  where $\odot$ indicates the *outer* product.

- In the communication channel the encoding weights are ± 1. Given the learned weights, $\boldsymbol{\omega}_{learn}$, and the encoding weight vector, $\tilde{\boldsymbol{\phi}}$, we can find the decoders, $\boldsymbol{\phi}$.

- Because the encoding weight vector is not a square matrix, we must take the pseudo-inverse (that is, use SVD) to find the decoders. As a result, the weight matrix we reconstruct using the extracted decoders, $\boldsymbol{\omega}_{recon}$, is different than the original learned matrix (see slide).

- We can compare these matrices based on learning with the matrix we originally found analytically, $\boldsymbol{\omega}_{analytic}$. The fact that these three matrices all perform similarly emphasizes that there are many solutions to problems like constructing a communication channel, given the high-dimensionality of weight space.
  - Looking at the commonalities amongst these three solutions also provides us with some clues as to what properties of the weights may be essential for a good communication channel. Here, all of these approaches find a solution that divides the population about equally into 'on' and 'off' neurons. And, the pattern of connectivity is similar in that oppositely tuned neurons tend to have negative weights and the strength of positive weights depends on the similarity of the neurons' tuning curves. So, there is a general structure to these matrices that becomes apparent and understandable once we have a variety of tools for both analyzing and synthesizing weight matrices.
- However, the reconstructed matrix is good at preserving the information in the incoming signal (see slide). So, this kind of decomposition may be useful for analyzing the often mysterious results of employing a learning rule in a given network.
  - What we've done is to take a set of weights generated in any particular way and then derive a set of decoders given encoding assumptions. These decoders can then be used to determine what function those weights compute (given the encoding assumptions) by using the to decode the incoming signal. So, we can use this approach to make a principled guess at the function of a set of learned weights. In this case it's fairly obvious since the function is simple.
- I've used a similar approach to generate a learning rule for associative memory. It's good for heteroassociative and autoassociative memory, and is Hebbian in form. It demonstrates a means of relating 'high-level' learning to neural connection weights as well.
  - Start at the high level by noting that we want to minimize:

$$E = \frac{1}{2}\left[R - T\right]^2$$

where

$$T = \sum_j y_j \phi_j$$

$$y_j = G_j\left[\alpha_j \left\langle \tilde{\phi_j}\left[\hat{C} + \hat{R}\right] + J_j \right\rangle\right]$$

and the representatio of R and C (the two signals being associated) are as usual. Notably, we only want an update rule for one set of weights, so lets find the rule for C. We

have to take the derivative of the high-level error wrt the weights:

$$\begin{aligned} E &= \frac{1}{2}\left[\sum_k \phi_k r_k - \sum_j \phi_j G_j \left[\sum_l \omega_{jl} c_l + \sum_k \omega_{jk} r_k + J_j\right]\right]^2 \\ \frac{dE}{d\omega_{jl}} &= -\left(\sum_k \phi_k r_k - \sum_j \phi_j y_j\right)\left[\frac{d}{d\omega_{jl}} \sum_i \phi_i G_i \left[\sum_l \omega_{il} c_l + \sum_k \omega_{ik} r_k + J_j\right]\right] \end{aligned}$$

we can approximate the G_j nonlinearity as $(y_j > 0)$ giving:

$$\frac{dE}{d\omega_{jl}} = -\left(\sum_k \phi_k r_k - \sum_j \phi_j y_j\right)(y_j > 0)\phi_j c_l$$

and for the standard delta rule:

$$\Delta\omega_{jl} = -\kappa \frac{dE}{d\omega_{jl}}$$

however, the derivative is biologically implausible as it stands, since it isn't local. So, let's multiply by $\alpha_j \tilde{\phi}_j$:

$$\Delta\omega_{jl}\alpha_j\tilde{\phi}_j = \kappa\left(\sum_k \omega_{ik} r_k - \sum_j \omega_{ij} y_j\right)(y_j > 0)\phi_j c_l$$

multiply by $\phi_j$again:

$$\begin{aligned} \Delta\omega_{jl}\alpha_j \left\|\tilde{\phi}_j\right\| &= \kappa\left(\sum_k \omega_{ik} r_k - \sum_j \omega_{ij} y_j\right)(y_j > 0)c_l \\ \Delta\omega_{jl} &= \frac{\kappa}{\alpha_j}\left(\sum_k \omega_{ik} r_k - \sum_j \omega_{ij} y_j\right)(y_j > 0)c_l \end{aligned}$$

because we know that the norm of the encoders is 1 and $\phi_j\tilde{\phi}_j \approx \delta$. Note also that the i index is really an index into the same population so some of these are recurrent weights.

10.1.2 More thoughts on learning

- The literature on learning in neural systems is enormous. The previous considerations merely scratch the surface of the subject. Nevertheless, here are some of the lessons supportive of this approach:

  - Because this framework is essentially analytic, it can help us gain new insights regarding standard learning rules. We can apply such rules and then analyze the resulting weight matrices in a way that may give us a new or better functional characterization
  - Given the representational hierarchy we can be confident that such rules and analyses will generalize to systems trafficking in more complex representations.
  - This is true both for analyzing the static transformations in a network, as we have done above, and for learning dynamic transformations. For instance, to learn an attractor, regardless of the complexity of the representation, we can minimize the energy that enforces $\dot{J}_i = 0$, i.e.,

$$E_i = \frac{1}{2} \left\langle [J_i(t) - \langle J_i(t) \rangle_t]^2 \right\rangle_t ,$$

  - Because we have a general means of incorporating input signals and transformations via control theory, accounting for learning in systems with explicit error signals, like the neural integrator, should be straightforward.
  - Most importantly, this approach gives us a means of constructing complex, functional systems, *independent* of learning. This is essential for modeling complex systems because, at the moment, it is not clear how to learn many of the complex behaviors exhibited by neurobiological systems. Such considerations do not make this approach a competitor to a learning-based approach, but rather show ways in which both can contribute to our understanding of complex neural systems.

- Here are some of the challenges highlighted by learning:

  - Can we generate learning rules that give rise to the systems we construct analytically? The superficial answer is 'yes', as the communication channel shows. But, for more complex transformations and more complex control structures, it is not clear what the answer will be. Notice that minimizing the variance between two populations, only constrains one degree of freedom in the network's representations. However, if we want to learn arbitrary transformations, we will need to constrain multiple degrees of freedom.
    * Let's consider this in more detail. Suppose that we have a population, a_i, from which we can encode the lower-order polynomials, x^n. We can thus find decoders, $\phi_i^{(x^n)}$, such that

$$\hat{x}^n = \sum_i a_i(x) \phi_i^{(x^n)}.$$

      Now suppose we want to compute some function, $f(x)$, such that

$$f(x) = \sum_m^M A_m x^n.$$

We know we can compute this function in some b_j population using the connection weights $\omega_{ji} = \tilde{\phi}_j \sum_m A_m \phi_i^{(x^n)}$ as we have in the past. Notice that these weights are now a function of the coefficients A_m. This means that any learning rule must be able to adjust the weights by systemmatically varying these M degrees of freedom *independently*.

- Furthermore, there needs to be a mechanism such that the desired values for the A_m are applied consistently across the population. But, to remain biologically plausible we have to impose these constraints using *local* learning rules. Applying such global constraints does not mean local rules are impossible to derive, just that they need to be highly sophisticated. In particular, they need to be controlled by a set of M-dimensional signals in a neurobiologically plausible way. These important challenges currently remain unresolved.

– Learning also highlights the ambiguity of connection weights. Because this framework analyzes weights as the projection between encoding and decoding vectors, and because projections of different sets of encoding and decoding vectors can result in the same matrix, there is no isolated 'right' decomposition of a weight matrix. This ambiguity can only be overcome in the context of a larger understanding of the system in which the weights are found. As in the case of the communication channel, finding the decoders for a system depends on our assumptions about the encoders. In order for these assumptions to be reasonable ones, and constrained in some way, they need to be made in the context of a larger system. So, just as justifying claims about what is represented by a system depend on the coherence of our overall story about brain function, so does justifying claims about what functions the weights instantiate. This parallel should not be surprising since representations and transformations are characterized in similar ways. The challenge lies in determining methods that are better able to extract all of the structure of a weight matrix consistent with this approach.

– A final challenge lies in trying to find ways to use the framework itself to generate novel learning rules. While this approach seems useful for analyzing the results of typical learning rules (like the Hebbian rule), it would be very useful to be able to have a means of determining plausible learning rules that would give rise to the kinds of high-level structure that the framework can build into models. It is a challenge, in other words, to not only build complex models like those we have discussed but to also explain how they may have come about given the kinds of synaptic changes observed in neurobiological systems.

– All of the preceding challenges can be summarized by noting that we have not provided a systematic way to relate learning to this approach.

10.2 Statistical inference

10.2.1 Introduction

- Why statistical inference? Here are some general reasons
  - Rather than consider the *truth and falsity of sentences*, probability theory considers the *likelihood of events or states* of the world. Because we are interested in neurobiological systems *in general*—nearly all of which do not have significant linguistic abilities, but all of which must reason about states of the physical world—it is reasonable to suspect that probability theory is the more appropriate tool for understanding such systems.
  - natural systems most often confront partial, noisy, uncertain information that they must use to make decisions; failure to use this information, no matter how degraded, may mean certain death. Again, probability theory and statistical inference naturally deal with cases in which information is partial.
  - Furthermore, probability theory describes how to incorporate multiple uncertain sources of information to get a more certain result. And, similarly, it describes how to update a current 'take' on the world given novel information; this, of course, is learning.
  - In general, then, probability theory is the best available quantitative tool for describing the kinds of reasoning evident in neurobiological systems.
  - There are already sophisticated probabilistic formalisms for modeling cognitive function (e.g., in computer vision), and for modeling neurobiological function as statistical inference [**?**, **?**]. One of the more general approaches to statistical modeling is *pattern theory*. The main purpose of this approach is to generate general models of complex, real-world patterns—like those encountered by biological systems.
  - Pattern theory incorporates Bayesian inference for describing pattern analysis and recognition. In our terminology, Bayesian inference defines transformations that are useful for working with the complex representations defined in pattern theory.
- Here is a reason specific to neural systems
  - To see why neural systems should be good at performing the relevant calculations, consider the joint distribution of two variables, $\rho(\mathbf{x}, \mathbf{y})$. Probability theory tells us how to write this joint distribution in terms of a relation between the individual distributions, $\rho(\mathbf{x})$ and $\rho(\mathbf{y})$:

$$\begin{aligned} \rho(\mathbf{x}, \mathbf{y}) &= \rho(\mathbf{y}|\mathbf{x})\rho(\mathbf{x}) && (10.4) \\ &= \rho(\mathbf{x}|\mathbf{y})\rho(\mathbf{y}). && (10.5) \end{aligned}$$

  - (Bayes' rule consists of equating the two right hand sides of the following equations and solving for either $\rho(\mathbf{x})$ or $\rho(\mathbf{y})$ as needed.)

– To determine the probability density function (PDF) for either parameter alone, we 'marginalize' (i.e., integrating) the joint:

$$\rho(\mathbf{y}) = \int \rho(\mathbf{x}, \mathbf{y})\, d\mathbf{x} \tag{10.6}$$

$$\rho(\mathbf{x}) = \int \rho(\mathbf{x}, \mathbf{y})\, d\mathbf{y}. \tag{10.7}$$

– Of course, each of these PDFs would be represented in a neural population.

– To proceed, consider the example of vision. We can think of $\rho(\mathbf{x})$ as representing how likely each image is in the environment. More precisely, this estimate is made in the context of some data provided by the environment. Let this data be the image falling on the retina (corrupted by noise). We would thus construct the *conditional* PDF, $\rho(\mathbf{x}|\mathbf{d})$, that corresponds to the probability density function for the true image given the measured image, $\mathbf{d}$.

– Now we want to extract certain properties of the image that are not directly captured by the image falling on the retina (i.e., $\mathbf{d}$), such as objects in the visual field. Let the variable $\mathbf{y}$ correspond to such properties. Generalizing (**??**), the probability density function for the variable $\mathbf{y}$ is a weighted average of the conditional $\rho(\mathbf{y}|\mathbf{x})$:

$$\rho(\mathbf{y}|\mathbf{d}) = \int \rho(\mathbf{y}|\mathbf{x})\rho(\mathbf{x}|\mathbf{d})\, d\mathbf{x}. \tag{10.8}$$

– This equation tells us, for instance, how to determine the probability of the existence of all possible objects in the field of view given the input image, $\mathbf{d}$. (Note: this particular example is oversimplified but serves to demonstrate the potential power of this kind of formulation). Notably, equation (**??**) is simply a linear transform, or projection of $\rho(\mathbf{x}|\mathbf{d})$ into the new space $\rho(\mathbf{y})$. It includes the transformations we have talked about to this point, i.e., $\mathbf{y} = f(\mathbf{x})$, as a subset (e.g, let $\rho(\mathbf{y}|\mathbf{x}) = \delta(\mathbf{y} - f(\mathbf{x}))$, and solving (**??**)).

– Importantly, (**??**) frees us from the assumption of Gaussian statistics. For example, it allows multi-modal distributions in $\rho(\mathbf{x})$ and $\rho(\mathbf{y})$ (see section **??**). As well, the conditional connecting the two spaces, $\rho(\mathbf{y}|\mathbf{x})$, can also be multi-modal, allowing unimodal inputs to support multiple hypotheses. As a result, being able to implement transformations defined by (**??**) results in computationally powerful systems. This is demonstrated by the many successes of artificial neural network (ANN) models, which can implement precisely these transformations. It should come as no surprise that real neurobiological networks are also ideally suited to implementing equation (**??**).

– To see why they are so suited, notice that $\rho(\mathbf{x}|\mathbf{d})$ is simply a function parameterized by the variables $\mathbf{d}$ (just as $x(\nu; \mathbf{A})$ is parameterized by $\mathbf{A}$). Hence we can create ensembles of neurons to encode $\rho(\mathbf{x}|\mathbf{d})$ and $\rho(\mathbf{y}|\mathbf{d})$

as

$$\begin{aligned} a_i(\mathbf{d}) &= G_i\left[\alpha_i\left\langle\tilde{\phi}_i(\mathbf{x})\rho(\mathbf{x}|\mathbf{d})\right\rangle_{\mathbf{x}} + J_i^{bias}\right] \\ b_j(\mathbf{d}) &= G_j\left[\alpha_j\left\langle\tilde{\phi}_j(\mathbf{y})\rho(\mathbf{y}|\mathbf{d})\right\rangle_{\mathbf{y}} + J_j^{bias}\right], \end{aligned} \tag{10.9}$$

with the corresponding decoding rules

$$\begin{aligned} \hat{\rho}(\mathbf{x}|\mathbf{d}) &= \sum_i \phi_i(\mathbf{x})a_i(\mathbf{d}) \\ \hat{\rho}(\mathbf{y}|\mathbf{d}) &= \sum_j \phi_j(\mathbf{y})b_j(\mathbf{d}). \end{aligned} \tag{10.10}$$

Using equation (**??**) to define the transformation between these two function ensembles, we find that the neuronal variables are related by

$$\begin{aligned} b_j(\mathbf{d}) &= G_j\left[\alpha_j\left\langle\tilde{\phi}_j(\mathbf{y})\rho(\mathbf{y}|\mathbf{x})\rho(\mathbf{x}|\mathbf{d})\right\rangle_{\mathbf{x},\mathbf{y}} + J_j^{bias}\right] \\ &= G_j\left[\alpha_j\left\langle\tilde{\phi}_j(\mathbf{y})\rho(\mathbf{y}|\mathbf{x})\sum_i \phi_i(\mathbf{x})a_i(\mathbf{d})\right\rangle_{\mathbf{x},\mathbf{y}} + J_j^{bias}\right] \\ &= G_j\left[\sum_i \omega_{ji}a_i(\mathbf{d}) + J_j^{bias}\right], \end{aligned} \tag{10.11}$$

where

$$\omega_{ji} = \alpha_j\left\langle\tilde{\phi}_j(\mathbf{y})\rho(\mathbf{y}|\mathbf{x})\phi_i(\mathbf{x})\right\rangle_{\mathbf{x},\mathbf{y}}. \tag{10.12}$$

Within this framework, equations (**??**) and (**??**) tell us how to implement simple feed-forward statistical inference in a neurobiologically plausible network. In particular, the connection weights between neurons in these networks can be understood as the projection of the encoding functions of the output neurons, $\tilde{\phi}_j(\mathbf{y})$, on the conditional, $\rho(\mathbf{y}|\mathbf{x})$, weighted by the decoding function, $\phi_i(\mathbf{x})$, of each input neuron.

– Examining these equations more closely reveals a number of important consequences.

1. Statistical inference in high-dimensional spaces requires estimating high-dimensional integrals. Because of the high degree of convergence on each neuron in typical neural networks, such networks are ideally suited for performing these kinds of transformations.
2. The form of these equations is identical to those for a simple feed-forward ANN. Furthermore, given our past analysis of neurobiological representation, we can see that there is no particular advantage gained by introducing the nonlinearity $G_i[\cdot]$ (especially on the higher-level variables). The nonlinearity is simply due to the nature of the representation found in neurobiologically plausible networks. Perhaps the

nonlinearity is more the result of an implementational constraint on low-power, low-precision physical devices like neurons.

3. This formulation of statistical inference using PDFs represented by a set of basis functions is more general than one that assumes Gaussian statistics.
4. Equation (**??**) can be understood by taking the conditional $\rho(\mathbf{y}|\mathbf{x})$ to be a 'look-up table' (LUT) that specifies the value of $\mathbf{y}$ for every value of $\mathbf{x}$. When viewed in this way, the limitations of (**??**) become clear—every possible consequence given the input $\mathbf{x}$ must be pre-computed (i.e., embedded in ω_{ji}). This is why we referred to our previous example in the visual system as oversimplified; that kind of simple feed-forward formulation of the problem would require astronomically high amounts of resources to represent $\rho(\mathbf{y}|\mathbf{d})$. Also, the conditional, $\rho(\mathbf{y}|\mathbf{x})$, would have to relate every possible image (a high-dimensional space) to every possible object (another very high-dimensional space). As a result, the addition of one new object (i.e., increasing the dimension of the object space by one) requires enough resources to define its relation to every possible image—this problem is commonly called the 'curse of dimensionality.'

- Let's consider 4. in more detail. One way to begin to address the curse of dimensionality is to recognize that it is often possible to divide such high-dimensional spaces into statistically independent subspaces. Suppose that $\mathbf{x}$ can be so divided into $\rho(\mathbf{x})$ and $\rho(\mathbf{z})$. Then, the joint density $\rho(\mathbf{x}, \mathbf{z})$ can be written as $\rho(\mathbf{x})\rho(\mathbf{z})$. Now, the transformation for finding the PDF for $\mathbf{y}$ given those for $\mathbf{x}$ and $\mathbf{z}$ becomes

$$\rho(\mathbf{y}|\mathbf{d}_\mathbf{x}, \mathbf{d}_\mathbf{z}) = \int \rho(\mathbf{y}|\mathbf{x}, \mathbf{z})\rho(\mathbf{x}|\mathbf{d}_\mathbf{x})\rho(\mathbf{z}|\mathbf{d}_\mathbf{z})\, d\mathbf{x}\, d\mathbf{z}. \tag{10.13}$$

  An implementation of (**??**) in a neural network has the form

$$b_j(\mathbf{y}) = G_j \left[\sum_{ik} \omega_{jik} a_i(\mathbf{d}_\mathbf{x}) c_k(\mathbf{d}_\mathbf{z}) + J_j^{bias} \right], \tag{10.14}$$

  where

$$\omega_{jik} = \left\langle \tilde{\phi}_j(\mathbf{y})\rho(\mathbf{y}|\mathbf{x}, \mathbf{z})\phi_i(\mathbf{x})\phi_k(\mathbf{z}) \right\rangle_{\mathbf{x},\mathbf{y},\mathbf{z}}. \tag{10.15}$$

- This helps the problem because the spaces $\mathbf{x}$ and $\mathbf{z}$ have a smaller dimensionality that the original $\mathbf{x}$. As a result, the LUT captured by $\rho(\mathbf{y}|\mathbf{x}, \mathbf{z})$ is smaller. To see why, realize that the number of dimensions that must be stored when $\mathbf{x}$ and $\mathbf{z}$ are independent is $D_1 = D_x + D_z$. When they aren't independent, then $D_x \cdot D_z \gg D_1$ dimensions must be stored.

- Also note that equation (**??**) defines a much richer class of inference than that supported by (**??**). In fact, there are a number of possible interpretations of (**??**) in a neurobiological context. Most generally, we can think of the variables $\mathbf{z}$

dynamically changing the connection weights between $\mathbf{y}$ and $\mathbf{x}$ such that the inference between the latter two spaces is carried out in the *context* defined by the $\mathbf{z}$ space. But, notice that (**??**) does not make a distinction between which space is providing the context and which is being modulated. This is what gives rise to different interpretations. Thus $\mathbf{x}$ and $\mathbf{z}$ could represent evidence from two-different modalities such as vision and audition, in which case the circuit is still feed-forward. Alternatively, $\mathbf{z}$ could represent variables in a higher order ('top-down') model and $\mathbf{x}$ could provide the feed-forward ('bottom-up') evidence.

- The powerful set of transformations that results from the generalization of equation (**??**) to (**??**) does not come for free, however. Most noticeably, (**??**) requires multiplicative interactions between the activities of the $\mathbf{z}$ and $\mathbf{x}$ populations. The ubiquitous need for performing this kind of context dependent statistical inference is one more reason that we might expect to find such nonlinearities in dendrites. Others have made similar suggestions.

10.2.2 Interpreting ambiguous input

- The natural environment is often filled with irrelevant or partial information, so there is good reason to expect that phenomena like object recognition do not depend solely on information extracted from sensory signals. Rather, many processes must depend critically on the system's assumptions about the structure of the environment. In other words, 'top-down' information is essential for the successful performance of many tasks.

- It has become clear, both functionally and anatomically, that top-down effects are common. Functionally, top-down psychological effects in vision have been extensively observed (one of the most striking is our inability to see concave faces as anything but convex from about a meter or more away). Anatomically, it is clear that there are massive reciprocal feedback projections from later to earlier visual areas in the primate.

- Consider a 'toy' characterization of a problem faced by animals. Suppose an animal is interested in determining the location of an object in the external world. Since the actual location is a matter of some uncertainty, we can describe the animal's final 'guess' as to where the object is as a probability density function, $\rho(y)$. This guess will be partly based on information provided by the sensory system, $\rho(x|\mathbf{d})$. This PDF can be understood as the sensory system's assignment of the probability that the object is at each location, x, given the noisy measurements, $\mathbf{d}$. Let us suppose for this example that, under certain conditions, this guess leads to a bimodal distribution that equally emphasizes two different locations (see slide).

- Given only information from the sensory system, the final guess would be the same as the best guess from the sensory system, since $\rho(x|\mathbf{d})$ incorporates all of the information available about the position of the object. However, if there is also a top-down 'model' of what positions to expect, then using information

from that model may bias $\rho(y)$ towards one of the two modes in $\rho(x|\mathbf{d})$. We can again consider such top-down information as a PDF, $\rho(z|\mathbf{m})$. This PDF captures the biased *a priori* information about how likely the object is to be at any given location in space, z, before any new data, $\mathbf{d}$, is known. This PDF is determined by the parameters, $\mathbf{m}$, which we can think of as summarizing past experience with object positions. The information in $\rho(z|\mathbf{m})$ can thus be used to disambiguate the information available solely from the sensory system.

- In order to determine what final guess should be determined for some particular $\rho(x|\mathbf{d})$ and $\rho(z|\mathbf{m})$, we can look at the relevant joint distribution, $\rho(x, y, z|\mathbf{d}, \mathbf{m})$, which captures *all* of the relations between $\rho(x|\mathbf{d})$, $\rho(y|\mathbf{d}, \mathbf{m})$, and $\rho(z|\mathbf{m})$.

- Given how we have set up the example, we assume that the input data, $\mathbf{d}$, and the internal model parameters, $\mathbf{m}$, are independent, which means that $\rho(x, z|\mathbf{d}, \mathbf{m}) = \rho(x|\mathbf{d})\rho(z|\mathbf{m})$. So we can express $\rho(y|\mathbf{d}, \mathbf{m})$ just as we did in the more general case in (**??**) as

$$\rho(y|\mathbf{d}, \mathbf{m}) = \iint \rho(y|x, z)\rho(x|\mathbf{d})\rho(z|\mathbf{m})\, dx\, dz. \tag{10.16}$$

- From this expression it is clear that we need to know, $\rho(y|x, z)$, the conditional probability that the object is in some location given the top-down and bottom-up information. If there was reason to think that either the top-down or bottom-up information was more reliable, we could use our definition of this PDF to capture that property of the system. In the absence of any such reason, as in this case, we are free to choose this conditional probability. So, we presume that $\rho(y|x, z)$ is

$$\rho(y|x, z) = \frac{1}{\sqrt{2\pi(\alpha(x - z)^2 + \beta)}} e^{-(y - \frac{1}{2}(x+z))^2 / \left(\alpha(x-z)^2 + \beta\right)}. \tag{10.17}$$

- This conditional emphasizes those places in the multi-modal distribution (i.e., $\rho(x|\mathbf{d})$) where the data and model agree, and de-emphasizes those places where they significantly disagree (β ensures non-zero variance, α controls how much the emphasis relies on this difference; this conditional takes the average value of x and z and constructs a distribution around that mean whose variance depends on how similar x and z are at that point.). Notably, this conditional would not work very well with Gaussian statistics, but we have more freedom to choose the conditional when working with general PDFs.

- We have now expressed a high-level formulation of the problem of how we should use previous knowledge to interpret ambiguous sensory data. Recall that this is the same as describing context-sensitive reasoning.

10.2.3 Parameter estimation

- The previous example assumed that there had already been a reasonably good top-down model, parameterized by $\mathbf{m}$, constructed before we attempted to disambiguate the bottom-up information. Ideally, we want a model that is generated

based on the statistics of the environment the animal finds itself in. In order to generate such a model, we can again use statistical inference. Here we consider a toy parameter estimation problem to show how a neural system can extract the relevant parameters governing a probability distribution of some property in the environment.

- Suppose that the system receives a stream of values, x_n, each of which can be thought of as a 'measurement' of the underlying statistical process that is giving rise to these values. In this example we assume that the set of values $x_1, \ldots, x_N$ are generated by a Gaussian process with a mean $\bar{x}$ and variance σ^2 that are fixed for a finite period of time. The purpose of the neural system is to determine estimates of the mean and variance from the measurements x_n signal.

- (If the mean and variance were truly static, there would be no need for any kind of dynamic statistical inference mechanism. So, we can think of the mean and variance as slowly changing or as being 'reset' occasionally.)

- To begin, let us define the conditional probability of obtaining the value of x given $\bar{x}$ and σ as a Gaussian:

$$\rho(x|\bar{x},\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\bar{x})^2/2\sigma^2}. \tag{10.18}$$

  We now assume that the mean, $\bar{x}$, and variance, σ^2, themselves are drawn from a broad distribution, $\rho_0(\bar{x},\sigma)$, whose precise form is not critical. We can express the marginal for x, $\rho(x)$, as

$$\rho(x) = \iint \rho(x|\bar{x},\sigma)\rho_0(\bar{x},\sigma)\, d\bar{x}\, d\sigma, \tag{10.19}$$

  which will also be very broad. We now suppose that a pair of values $\bar{x}$ and σ have been drawn from $\rho_0(\bar{x},\sigma)$ and are used to generate a signal, $x_1, \ldots, x_N$, by a random Gaussian process.

- Our goal is thus to design a network that represents a PDF over $\bar{x}$ and σ that starts with the prior, $\rho_0(\bar{x},\sigma)$, and is updated appropriately as the signal arrives. We can define the representation of such a population as

$$\begin{aligned} b_j(x_1,\ldots,x_n) &= G_j\left[\alpha_j \left\langle \tilde{\phi}_j(\bar{x},\sigma)\rho(\bar{x},\sigma|x_1,\ldots,x_n)\right\rangle_{\bar{x},\sigma}\right] \\ \hat{\rho}(\bar{x},\sigma|x_1,\ldots x_n) &= \sum_j b_j(x_1,\ldots,x_n)\phi_j(\bar{x},\sigma). \end{aligned}$$

  To simplify the notation we subsequently use $b_j(x_n)$ and $\rho(\bar{x},\sigma|x_n)$ to indicate the relevant encoding and decoding.

- Because the measurements of the true signal are subject to uncertainty, we take the signal values, x_n, to be the mean of a Gaussian PDF, whose variance, $\sigma^2_{x^n}$, is determined by the amount of noise in the measurement. Since only the mean,

x_n, will change, we write this as the conditional, $\rho(x|x_n)$, that is represented by a neural population, where

$$\begin{aligned} a_i(x_n) &= G_i\left[\alpha_i \left\langle \tilde{\phi}_i(x)\rho(x|x_n)\right\rangle_x + J_i^{bias}\right] \\ \hat{\rho}(x|x_n) &= \sum_i \phi_i(x) a_i(x_n) \end{aligned}$$

define the encoding and decoding.

- These two representations can be related through an update rule that tells us how the density $\rho(\bar{x}, \sigma|x_n)$ should be modified given the next data point x_{n+1}; namely,

$$\begin{aligned} \rho(\bar{x}, \sigma|x_1, \ldots x_{n+1}) &= \int \rho(\bar{x}, \sigma|x)\rho(x|x_{n+1})\, dx \\ &= \int \frac{\rho(x|\bar{x}, \sigma)}{\rho(x)}\rho(x|x_{n+1})\, dx\, \rho(\bar{x}, \sigma|x_n) \\ &\approx \int \rho(x|\bar{x}, \sigma)\rho(x|x_{n+1})\, dx\, \rho(\bar{x}, \sigma|x_n), \end{aligned} \tag{10.20}$$

  where we assume $\rho(x)$ is very broad and hence approximated as a constant whose value can be computed using **(??)** (the first step is a result of applying Bayes' rule). The coupling weights can be found in the usual fashion:

$$\omega_{jil} = \alpha_j \left\langle \tilde{\phi}_j(\bar{x}, \sigma)\rho(x|\bar{x}, \sigma)\phi_i(x)\phi_l(\bar{x}, \sigma)\right\rangle_{x,\bar{x},\sigma}. \tag{10.21}$$

  We can now write the firing rates of the b_j population at time $n+1$ as

$$b_j(x_{n+1}) = G_j\left[\sum_{il} \omega_{jil} a_i(x_{n+1}) b_l(x_n) + J_j^{bias}\right].$$

- Note that this same derivation holds regardless of our explicit restrictions on the shape of $\rho(x|x_n)$ or $\rho(\bar{x}, \sigma|x_n)$. That is, the derivation is general enough such that they need not be Gaussians. This is true despite the Gaussian form for the conditional probability which relates the $\rho(x|x_n)$ representation with the $\rho(\bar{x}, \sigma|x_n)$ representation, i.e., **(??)**.

- The slides show an example time slice of the results of simulating this network using 100 LIF neurons in each population. Though not evident here, as more data is presented to the network, the estimate of the mean and variance, $\rho(\bar{x}, \sigma|x_n)$, becomes narrower. Eventually, the narrowness of the estimate, that is, the certainty of the best guess regarding the true values of $\bar{x}$ and σ of the underlying process, is limited by the goodness of the neural representation in the b_j population. This again demonstrates the importance of accounting for resource constraints when modeling transformations in neurobiological systems.

- This is just one of many ways to predict an unknown process under uncertainty. We have taken an essentially Bayesian approach, but such approaches are closely allied to others, like Kalman filtering)which we consider in more detail in the book) and information optimization.