Chapter 7

Network Models

7.1 Introduction

Extensive synaptic connectivity is a hallmark of neural circuitry. For example, neurons in the mammalian neocortex each receive thousands of synaptic inputs. Network models allow us to explore the computational potential of such connectivity, using both analysis and simulations. As illustrations, we study in this chapter how networks can perform the following tasks: coordinate transformations needed in visually guided reaching, selective amplification leading to models of simple and complex cells in primary visual cortex, integration as a model of short-term memory, noise reduction, input selection, gain modulation, and associative memory. Networks that undergo oscillations are also analyzed, with application to the olfactory bulb. Finally, we discuss network models based on stochastic rather than deterministic dynamics, using the Boltzmann machine as an example.

Neocortical circuits are a major focus of our discussion. In the neocortex, which forms the convoluted outer surface of the (for example) human brain, neurons lie in six vertical layers highly coupled within cylindrical columns. Such columns have been suggested as basic functional units, and stereotypical patterns of connections both within a column and between columns are repeated across cortex. There are three main classes of interconnections within cortex, and in other areas of the brain as well. Feedforward connections bring input to a given region from another region located at an earlier stage along a particular processing pathway. Recurrent synapses interconnect neurons within a particular region that are considered to be at the same stage along the processing pathway. These may include connections within a cortical column as well as connections between both nearby and distant cortical columns within a region. Top-down connections carry signals back from areas located at later stages. These definitions depend on the how the region being studied is specified and on the

cortical columns

feedforward, recurrent, and top-down connections hierarchical assignment of regions along a pathway. In general, neurons within a given region send top-down projections back to the areas from which they receive feedforward input, and receive top-down input from the areas to which they project feedforward output. The numbers, though not necessarily the strengths, of feedforward and top-down fibers between connected regions are typically comparable, and recurrent synapses typically outnumber feedforward or top-down inputs. We begin this chapter by studying networks with purely feedforward input and then study the effects of recurrent connections. The analysis of top-down connections, for which it is more difficult to establish clear computational roles, is left until chapter 10.

The most direct way to simulate neural networks is to use the methods discussed in chapters 5 and 6 to synaptically connect model spiking neurons. This is a worthwhile and instructive enterprise, but it presents significant computational, calculational, and interpretational challenges. In this chapter, we follow a simpler approach and construct networks of neuron-like units with outputs consisting of firing rates rather than action potentials. Spiking models involve dynamics over time scales ranging from channel openings that can take less than a millisecond, to collective network processes that may be several orders of magnitude slower. Firing-rate models avoid the short time scale dynamics required to simulate action potentials and thus are much easier to simulate on computers. Firing-rate models also allow us to present analytic calculations of some aspects of network dynamics that could not be treated in the case of spiking neurons. Finally, spiking models tend to have more free parameters than firing-rate models, and setting these appropriately can be difficult.

There are two additional arguments in favor of firing-rate models. The first concerns the apparent stochasticity of spiking. The models discussed in chapters 5 and 6 produce spike sequences deterministically in response to injected current or synaptic input. Deterministic models can only predict spike sequences accurately if all their inputs are known. This is unlikely to be the case for the neurons in a complex network, and network models typically include only a subset of the many different inputs to individual neurons. Therefore, the greater apparent precision of spiking models may not actually be realized in practice. If necessary, firing-rate models can be used to generate stochastic spike sequences from a deterministically computed rate, using the methods discussed in chapters 1 and 2.

The second argument comes involves a complication with spiking models that arises when they are used to construct simplified networks. Although cortical neurons receive many inputs, the probability of finding a synaptic connection between a randomly chosen pair of neurons is actually quite low. Capturing this feature, while retaining a high degree of connectivity through polysynaptic pathways, requires including a large number of neurons in a network model. A standard way of dealing with this problem is to use a single model unit to represent the average response of several neurons that have similar selectivities. These ‘averaging’ units can then be interconnected more densely than the individual neurons of the actual network, and so fewer of them are needed to build the model. If neural responses are characterized by firing rates, the output of the model unit is simply the average of the firing rates of the neurons it represents collectively. However, if the response is a spike, it is not clear how the spikes of the represented neurons can be averaged. The way spiking models are typically constructed, an action potential fired by the model unit duplicates the effect of all the neurons it represents firing synchronously. Not surprisingly, such models tend to exhibit large-scale synchronization unlike anything seen in a healthy brain.

Firing-rate models also have their limitations. Most importantly, they cannot account for aspects of spike timing and spike correlations that may be important for understanding nervous system function. Firing-rate models are restricted to cases where the firing of neurons in a network is uncorrelated, with little synchronous firing, and where precise patterns spike timing are unimportant. In such cases, comparisons of spiking network models with models that use firing-rate descriptions have shown that they produce similar results. Nevertheless, the exploration of neural networks undoubtedly requires the use of both firing-rate and spiking models.

7.2 Firing-Rate Models

As discussed in chapter 1, the sequence of spikes generated by a neuron is completely characterized by the neural response function $\rho(t)$, which consists of δ function spikes located at times when the neuron fired action potentials. In firing-rate models, the exact description of a spike sequence provided by the neural response function $\rho(t)$ is replaced by the approximate description provided by the firing rate $\mathrm{r}(t)$. Recall from chapter 1 that $\mathrm{r}(t)$ is defined as the probability density of firing and is obtained from $\rho(t)$ by averaging over trials. The validity of a firing-rate model depends on how well the trial-averaged firing rate of network units approximates the effect of actual spike sequences on the dynamic behavior of the network.

The replacement of the neural response function by the corresponding firing rate is typically justified by the fact that each network neuron has a large number of inputs. Replacing $\rho(t)$, which describes an actual spike train, by the trial-averaged firing rate $\mathrm{r}(t)$ is justified if the quantities of relevance for network dynamics are relatively insensitive to the trial-to-trial fluctuations in the spike sequences represented by $\rho(t)$. In a network model, the relevant quantities that must be modeled accurately are the total inputs to all the neurons in the network. For any single synaptic input, the trial-to-trial variability is likely to be large. However, if we sum the input over many synapses activated by uncorrelated presynaptic spike trains, the mean of the total input typically grows linearly with the number of synapses, while its standard deviation grows only as the square root of the number of synapses. Thus, for uncorrelated presynaptic spike trains, using presynaptic firing rates in place of the actual presynaptic spike trains may not significantly modify the dynamics of the network. Conversely, a firing-rate model will fail to describe a network adequately if the presynaptic inputs to a substantial fraction of its neurons are correlated. This can occur, for example, if the presynaptic neurons fire synchronously.

The synaptic input arising from a presynaptic spike train is effectively filtered by the dynamics of the conductance changes that each presynaptic action potential evokes in the postsynaptic neuron (see chapter 5), and the dynamics of propagation of the current from the synapse to the soma. The temporal averaging provided by slow synaptic or membrane dynamics can reduce the effects of spike train variability and help justify the approximation of using firing rates instead of presynaptic spike trains. Firing-rate models are more accurate if the network being modeled has a significant amount of synaptic transmission that is slow relative to typical presynaptic interspike intervals.

The construction of a firing-rate model proceeds in two steps. First, we determine how the total synaptic input to a neuron depends on the firing rates of its presynaptic afferents. This is where we use firing rates to approximate neural response functions. Second, we model how the firing rate of the postsynaptic neuron depends on its total synaptic input. Firing-rate response curves are typically measured by injecting current into the soma of a neuron. We therefore find it most convenient to define the total synaptic input as the total current delivered to the soma as a result of all the synaptic conductance changes resulting from presynaptic action potentials. We denote this total synaptic current by $I_{\rm s}$. We then determine the postsynaptic firing rate from $I_{\rm s}$. In general, $I_{\rm s}$ depends on the spatially inhomogeneous membrane potential of the neuron, but we assume that, other than during action potentials or transient hyperpolarizations, the membrane potential remains close to, but slightly below, the threshold for action potential generation. An example of this type of behavior is seen in the upper panels of figure 7.2. $I_{\rm s}$ is then approximately equal to the synaptic current that would be measured from the soma in a voltage-clamp experiment, except for a reversal of sign. In the next section, we model how $I_{\rm s}$ depends on presynaptic firing rates.

synaptic current I_s

In the network models we consider, both the output from, and input to, a neuron are characterized by firing rates. To avoid a proliferation of sub- and superscripts on the quantity $r(t)$, we use the letter u to denote a presynaptic firing rate, and v to denote a postsynaptic rate. Note that v is used here to denote a firing rate, not a membrane potential. In addition, we use these two letters to distinguish input and output firing rates in network models, a convention we retain through the remaining chapters. When we consider multiple input or output neurons, we use vectors $\mathbf{u}$ and $\mathbf{v}$ to represent their firing rates collectively, with the components of these vectors representing the firing rates of the individual input and output units.

input rate u
output rate v

input rate vector $\mathbf{u}$
output rate vector $\mathbf{v}$

The Total Synaptic Current

Consider a neuron receiving N_u synaptic inputs labeled by $b = 1, 2, \ldots, N_u$ (figure 7.1). The firing rate of input b is denoted by u_b, and the input rates are represented collectively by the N_u-component vector $\mathbf{u}$. We model how the synaptic current I_s depends on presynaptic firing rates by first considering how it depends on presynaptic spikes. If an action potential arrives at input b at time zero, we write the synaptic current generated in the soma of the postsynaptic neuron at time t as $w_b K_s(t)$ where w_b is the synaptic weight and $K_s(t)$ is called the synaptic kernel. Collectively, the synaptic weights are represented by a synaptic weight vector $\mathbf{w}$, which has N_u components w_b. The amplitude and sign of the synaptic current generated by input b are determined by w_b. For excitatory synapses, $w_b > 0$, and for inhibitory synapses, $w_b < 0$. In this formulation of the effect of presynaptic spikes, the probability of transmitter release from a presynaptic terminal is absorbed into the synaptic weight factor w_b, and we do not include short-term plasticity in the model (although this can be done by making w_b a dynamic variable).

synaptic weights $\mathbf{w}$

The synaptic kernel, $K_s(t) \geq 0$, describes the time course of the synaptic current in response to a presynaptic spike arriving at time $t=0$. This time course depends on the dynamics of the synaptic conductance activated by the presynaptic spike and also on both the passive and active properties of the dendritic cables that carry the synaptic current to the soma. For example, long passive cables broaden the synaptic kernel and slow its rise from zero. Cable calculations or multicompartment simulations, such as those discussed in chapter 6, can be used to compute $K_s(t)$ for a specific dendritic structure. To avoid ambiguity, we normalize $K_s(t)$ by requiring its integral over all positive times to be one. At this point, for simplicity, we use the same function $K_s(t)$ to describe all synapses.

synaptic kernel $K_s(t)$

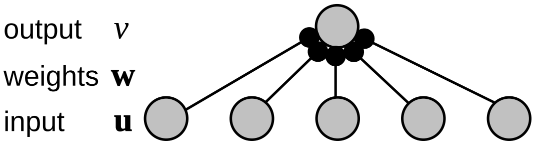

Figure 7.1: Feedforward inputs to a single neuron. Input rates $\mathbf{u}$ drive a neuron at an output rate v through synaptic weights given by the vector $\mathbf{w}$.

Assuming that the spikes at a single synapse act independently, the total synaptic current at time t arising from a sequence of presynaptic spikes occurring at input b at times t_i is given by the sum

$$w_b \sum_{t_i < t} K_s(t - t_i) = w_b \int_{-\infty}^{t} d\tau \, K_s(t - \tau)\rho_b(\tau) \, . \tag{7.1}$$

In the second expression, we have used the neural response function, $\rho_b(\tau) = \sum_i \delta(\tau - t_i)$, to describe the sequence of spikes fired by presynaptic neuron b. The equality follows from integrating over the sum of δ functions in the definition of $\rho_b(\tau)$. If there is no nonlinear interaction between different synaptic currents, the total synaptic current coming from all presynaptic inputs is obtained simply by summing,

$$I_{\text{s}} = \sum_{b=1}^{N_u} w_b \int_{-\infty}^{t} d\tau \, K_{\text{s}}(t-\tau)\rho_b(\tau) \, . \tag{7.2}$$

As discussed previously, the critical step in the construction of a firing-rate model is the replacement of the neural response function $\rho_b(\tau)$ in equation 7.2 by the firing rate of neuron b, namely $u_b(\tau)$, so that we write

$$I_{\text{s}} = \sum_{b=1}^{N_u} w_b \int_{-\infty}^{t} d\tau \, K_{\text{s}}(t-\tau)u_b(\tau) \, . \tag{7.3}$$

The synaptic kernel most frequently used in firing-rate models is an exponential, $K_{\text{s}}(t) = \exp(-t/\tau_r)/\tau_r$. With this kernel, we can describe I_{s} by a differential equation if we take the derivative of equation 7.3 with respect to t,

$$\tau_{\text{s}} \frac{dI_{\text{s}}}{dt} = -I_{\text{s}} + \sum_{b=1}^{N_u} w_b u_b = -I_{\text{s}} + \mathbf{w} \cdot \mathbf{u} \, . \tag{7.4}$$

dot product

In the second equality, we have expressed the sum $\sum w_b u_b$ as the dot product of the weight and input vectors, $\mathbf{w} \cdot \mathbf{u}$. In this and the following chapters, we primarily use the vector versions of equations such as equation 7.4, but when we first introduce an important new equation, we often write it in its subscripted form as well.

Recall that K describes the temporal evolution of the synaptic current due to both synaptic conductance and dendritic cable effects. For an electrotonically compact dendritic structure, τ_{s} will be close to the time constant that describes the decay of the synaptic conductance. For fast synaptic conductances such as those due to AMPA glutamate receptors, this may be as short as a few milliseconds. For a long, passive dendritic cable, τ_{s} may be larger than this, but its measured value is typically quite small.

The Firing-Rate

Equation 7.4 determines the synaptic current entering the soma of a postsynaptic neuron in terms of the firing rates of the presynaptic neurons. To finish formulating a firing-rate model, we must determine the postsynaptic firing rate from our knowledge of I_{s}. For constant synaptic current, the firing rate of the postsynaptic neuron can be expressed as $v = F(I_{\text{s}})$, where F is the steady-state firing rate as a function of somatic input current. F is also called an activation function. F is sometimes taken to be a saturating function such as a sigmoid function. This is useful in cases where

activation function $F(I_{\text{s}})$

the derivative of F is needed in the analysis of network dynamics. It is also bounded from above, which can be important in stabilizing a network against excessively high firing rates. More often, we use a threshold linear function $F(I_s) = [I_s - \gamma]_+$, where γ is the threshold and the notation $[\;]_+$ denotes half-wave rectification as in previous chapters. For convenience, we treat I_s in this expression as if its were measured in units of a firing rate (Hz), i.e. as if I_s is multiplied by a constant that converts its units from nA to Hz. This makes the synaptic weights dimensionless. The threshold γ also has units of Hz.

threshold γ

For time-independent inputs, the relation $v = F(I_s)$ is all we need to know to complete the firing-rate model. The total steady-state synaptic current predicted by equation 7.4 for time-independent $\mathbf{u}$ is $I_s = \mathbf{w} \cdot \mathbf{u}$. This generates a steady-state output firing rate $v = v_\infty$ given by

$$v_\infty = F(\mathbf{w} \cdot \mathbf{u}) \,. \tag{7.5}$$

The steady-state firing rate tells us how a neuron responds to constant current, but not to a current that changes with time. To model time-dependent inputs, we need to know the firing rate in response to a time-dependent synaptic current $I_s(t)$. The simplest assumption is that this is still given by the activation function, so $v = F(I_s(t))$ even when the total synaptic current varies with time. This leads to a firing-rate model in which all the dynamics arise exclusively from equation 7.4,

firing-rate model with current dynamics

$$\tau_s \frac{dI_s}{dt} = -I_s + \mathbf{w} \cdot \mathbf{u} \quad \text{with} \quad v = F(I_s) \,. \tag{7.6}$$

An alternative formulation of a firing-rate model can be constructed by assuming that the firing rate does not follow changes in the total synaptic current instantaneously, as was assumed for the model of equation 7.6. Action potentials are generated by the synaptic current through its effect on the membrane potential of the neuron. Due to the membrane capacitance and resistance, the membrane potential is, roughly speaking, a low-pass filtered version of I_s (see the Mathematical Appendix). For this reason, the time-dependent firing rate is often modeled as a low-pass filtered version of the steady-state firing rate,

$$\tau_r \frac{dv}{dt} = -v + F(I_s(t)) \,. \tag{7.7}$$

The constant τ_r in this equation determines how rapidly the firing rate approaches its steady-state value for constant I_s, and how closely v can follow rapid fluctuations for a time-dependent $I_s(t)$. Equivalently, it measures the time-scale over which v averages $F(I_s(t))$. The low-pass filtering effect of equation 7.7 is described in the Mathematical Appendix in the context of electrical circuit theory. The argument we have used to motivate equation 7.7 would suggest that τ_r should be approximately equal to the membrane time constant of the neuron. However, this argument really applies to the membrane potential not the firing rate, and the dynamics of the two are not the same. Most network models use a value of τ_r that is considerably less than the membrane time constant. We re-examine this issue in the following section.

The second model that we have described involves the pair of equations 7.4 and 7.7. If one of these equations relaxes to its equilibrium point much more rapidly than the other, the pair can be reduced to a single equation. For example, if $\tau_r \ll \tau_s$, we can make the approximation that equation 7.7 rapidly sets $v = F(I_s(t))$, and then the second model reduces to the first model that is defined by equation 7.6. If instead, $\tau_r \gg \tau_s$, we can make the approximation that equation 7.4 comes to equilibrium quickly compared to equation 7.7. Then, we can make the replacement $I_s = \mathbf{w} \cdot \mathbf{u}$ in equation 7.7 and write

firing-rate equation

$$\tau_r \frac{dv}{dt} = -v + F(\mathbf{w} \cdot \mathbf{u}) . \qquad (7.8)$$

For most of this chapter, we analyze network models described by the firing-rate dynamics of equation 7.8, although occasionally we consider networks based on equation 7.6.

Firing-Rate Dynamics

The firing-rate models described by equations 7.6 and 7.8 differ in their assumptions about how firing rates respond to and track changes in the input current to a neuron. In one case (equation 7.6), it is assumed that firing rates follow time varying input currents instantaneously without attenuation or delay. In the other case (equation 7.8), the firing rate is a low-pass filtered version of the input current. To study the relationship between input current and firing rate, it is useful to examine the firing rate of a spiking model neuron in response to a time-varying injected current, $I(t)$. The model used for this purpose in figure 7.2 is an integrate-and-fire neuron receiving balanced excitatory and inhibitory synaptic input along with a current injected into the soma that is the sum of constant and oscillating components. This model was discussed in chapter 5. The balanced synaptic input is used to represent background input not included in the computation of I_s, and it acts as a source of noise. The noise prevents effects such as locking of the spiking to the oscillations of the injected current that would invalidate a firing-rate description.

Figure 7.2 shows the firing rates of the model integrate-and-fire neuron in response to an input current $I(t) = I_0 + I_1 \cos(\omega t)$. The firing rate is plotted at different times during the cycle of the input current oscillations for ω corresponding to frequencies of 1, 50, and 100 Hz. For the panels on the left side, the constant component of the injected current (I_0) was adjusted so the neuron never stopped firing during the cycle. In this case, the relation $v(t) = F(I(t))$ (solid curves) provides an accurate description

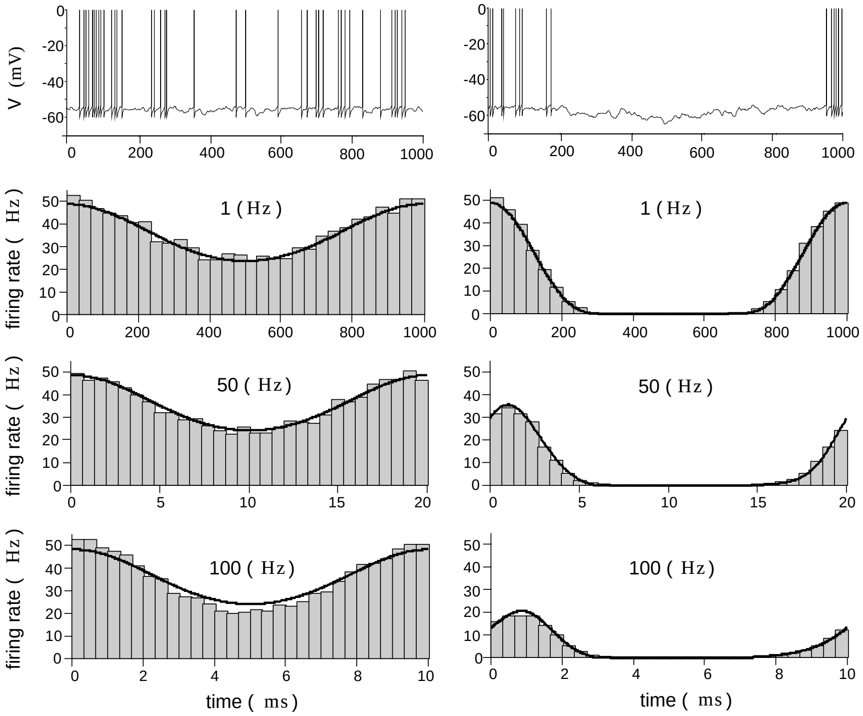

Figure 7.2: Firing rate of an integrate-and-fire neuron receiving balanced excitatory and inhibitory synaptic input and an injected current consisting of a constant and a sinusoidally varying term. For the left panels, the constant component of the injected current was adjusted so the firing never stopped during the oscillation of the varying part of the injected current. For the right panel, the constant component was lowered so the firing stopped during part of the cycle. The upper panels show two representative voltage traces of the model cell. The histograms beneath these traces were obtained by binning spikes generated over multiple cycles. They show the firing rate as a function of the time during each cycle of the injected current oscillations. The different rows show 1, 50, and 100 Hz oscillation frequencies for the injected current. The solid curves show the fit of a firing-rate model that involves both instantaneous and low-pass filtered effects of the injected current. For the left panel, this reduces to the simple prediction $v = F(I(t))$. (Adapted from Chance et al., 2000.)

of the firing rate for all of the oscillation frequencies shown. As long as the neuron keeps firing fairly rapidly, the low-pass filtering properties of the membrane potential are not relevant for the dynamics of the firing rate. Low-pass filtering is irrelevant in this case, because the neuron is continually being shuttled between the threshold and reset values, and so it never has a chance to settle exponentially anywhere near its steady-state value.

The right panels in figure 7.2 show that the situation is different if the input current is below the threshold for firing through a significant part of the oscillation cycle. In this case, the firing is delayed and attenuated at high frequencies as would be predicted by equation 7.7. In this case, the membrane potential stays below threshold for long enough periods of time that its dynamics become relevant for the firing of the neuron.

The essential message from figure 7.2 is that neither equation 7.6 nor 7.8 provides a completely accurate prediction of the dynamics of the firing rate at all frequencies and for all levels of injected current. A more complex model can be constructed that accurately describes the firing rate over the entire range of input currents amplitudes and frequencies. The solid curves in figure 7.2 were generated by a model that expresses the firing rate as a function of both $F(I)$ and of v computed from equation 7.8 (although it reduces to $v = F(I(t))$ in the case of the left panel of figure 7.2). In other words, it is a combination of the two models discussed in the previous section. This compound model provides quite an accurate description of the firing rate of the integrate-and-fire model, but it is more complex than the models used in this chapter.

Feedforward and Recurrent Networks

Figure 7.3 shows examples of network models with feedforward and recurrent connectivity. The feedforward network of figure 7.3A has N_v output units with rates v_a ($a = 1, 2, \ldots, N_v$), denoted collectively by the vector $\mathbf{v}$, driven by N_u input units with rates $\mathbf{u}$. Equations 7.8 and 7.6 can easily be extended to cover this case by replacing the vector of synaptic weights $\mathbf{w}$ by a matrix $\mathbf{W}$, with the matrix component W_{ab} representing the strength of the synapse from input unit b to output unit a. Using the formulation of equation 7.8, the output firing rates are then determined by

feedforward model

$$\tau_r \frac{d\mathbf{v}}{dt} = -\mathbf{v} + \mathbf{F}(\mathbf{W}\cdot\mathbf{u}) \quad \text{or} \quad \tau_r \frac{dv_a}{dt} = -\mathbf{v} + F\left(\sum_{b=1}^{N_u} W_{ab}u_b\right) . \qquad (7.9)$$

We use the notation $\mathbf{W}\cdot\mathbf{u}$ to denote the vector with components $\sum_b W_{ab}u_b$. The use of the dot to represent a sum of a product of two quantities over a shared index is borrowed from the notation for the dot product of two vectors. The expression $\mathbf{F}(\mathbf{W}\cdot\mathbf{u})$ represents the vector with components $F(\sum W_{ab}u_b)$ for $a = 1, 2, \ldots, N_v$.

The recurrent network of figure 7.3B also has two layers of neurons with rates $\mathbf{u}$ and $\mathbf{v}$, but in this case the neurons of the output layer are interconnected with synaptic weights described by a matrix $\mathbf{M}$. Matrix element $M_{aa'}$ describes the strength of the synapse from output unit a' to output unit a. The output rates in this case are determined by

recurrent model

$$\tau_r \frac{d\mathbf{v}}{dt} = -\mathbf{v} + \mathbf{F}(\mathbf{W}\cdot\mathbf{u} + \mathbf{M}\cdot\mathbf{v}) . \qquad (7.10)$$

It is often convenient to define the total feedforward input to each neuron in the network of figure 7.3B as $\mathbf{h} = \mathbf{W}\cdot\mathbf{u}$. Then, the output rates are

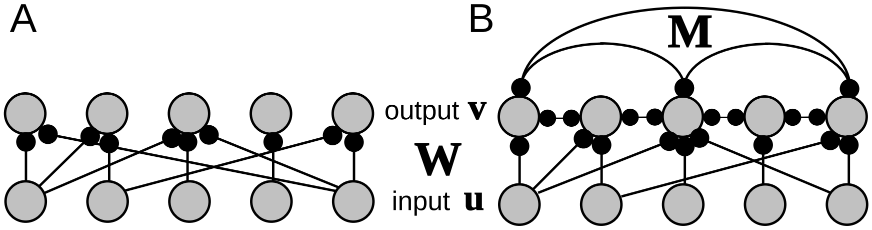

Figure 7.3: Feedforward and recurrent networks. A) A feedforward network with input rates **u**, output rates **v**, and a feedforward synaptic weight matrix **W**. B) A recurrent network with input rates **u**, output rates **v**, a feedforward synaptic weight matrix **W**, and a recurrent synaptic weight matrix **M**. Although we have drawn the connections between the output neurons as bidirectional, this does not necessarily imply connections of equal strength in both directions.

determined by the equation

$$\tau_r \frac{d\mathbf{v}}{dt} = -\mathbf{v} + \mathbf{F}(\mathbf{h} + \mathbf{M}\cdot\mathbf{v})\,. \tag{7.11}$$

Dale's law

Neurons are typically classified as either excitatory or inhibitory, meaning that they have either excitatory or inhibitory effects on all of their postsynaptic targets. This property is formalized in Dale's law, which states that a neuron cannot excite some of its postsynaptic targets and inhibit others. In terms of the elements of **M**, this means that for each presynaptic neuron a', $M_{aa'}$ must have the same sign for all postsynaptic neurons a. To impose this restriction, it is convenient to describe excitatory and inhibitory neurons separately. The firing-rate vectors $\mathbf{v}_\mathrm{E}$ and $\mathbf{v}_\mathrm{I}$ for the excitatory and inhibitory neurons are then described by a coupled set of equations identical in form to equation 7.11,

excitatory-inhibitory network

$$\tau_\mathrm{E} \frac{d\mathbf{v}_\mathrm{E}}{dt} = -\mathbf{v}_\mathrm{E} + \mathbf{F}_\mathrm{E}\left(\mathbf{h}_\mathrm{E} + \mathbf{M}_\mathrm{EE}\cdot\mathbf{v}_\mathrm{E} + \mathbf{M}_\mathrm{EI}\cdot\mathbf{v}_\mathrm{I}\right) \tag{7.12}$$

and

$$\tau_\mathrm{I} \frac{d\mathbf{v}_\mathrm{I}}{dt} = -\mathbf{v}_\mathrm{I} + \mathbf{F}_\mathrm{I}\left(\mathbf{h}_\mathrm{I} + \mathbf{M}_\mathrm{IE}\cdot\mathbf{v}_\mathrm{E} + \mathbf{M}_\mathrm{II}\cdot\mathbf{v}_\mathrm{I}\right)\,. \tag{7.13}$$

There are now four synaptic weight matrices describing the four possible types of neuronal interactions. The elements of $\mathbf{M}_\mathrm{EE}$ and $\mathbf{M}_\mathrm{IE}$ are greater than or equal to zero, and those of $\mathbf{M}_\mathrm{EI}$ and $\mathbf{M}_\mathrm{II}$ are less than or equal to zero. These equations allow the excitatory and inhibitory neurons to have different time constants, activation functions, and feedforward inputs.

symmetric coupling

In this chapter, we consider several recurrent network models described by equation 7.11 with a symmetric weight matrix, $M_{aa'} = M_{a'a}$ for all a and a'. Requiring **M** to be symmetric simplifies the mathematical analysis, but it violates Dale's law. Suppose, for example, that neuron a, which is excitatory, and neuron a', which is inhibitory, are mutually connected. Then,

$M_{aa'}$ should be negative and $M_{a'a}$ positive, so they cannot be equal. Equation 7.11 with symmetric **M** can be interpreted as a special case of equations 7.12 and 7.13 in which the inhibitory dynamics are instantaneous ($\tau_{\text{I}} \to 0$) and the inhibitory rates are given by $\mathbf{v}_{\text{I}} = \mathbf{M}_{\text{IE}}\mathbf{v}_{\text{E}}$. This produces an effective recurrent weight matrix $\mathbf{M} = \mathbf{M}_{\text{EE}} + \mathbf{M}_{\text{EI}} \cdot \mathbf{M}_{\text{IE}}$, which can be made symmetric by the appropriate choice of the dimension and form of the matrices $\mathbf{M}_{\text{EI}}$ and $\mathbf{M}_{\text{IE}}$. The dynamic behavior of equation 7.11 is restricted by requiring the matrix **M** to be symmetric. For example symmetric coupling typically does not allow for network oscillations. In the latter part of this chapter, we consider the richer dynamics of models described by equations 7.12 and 7.13.

Continuously Labeled Networks

It is often convenient to identify each neuron in a network using a parameter that describes some aspect of its selectivity rather than the integer label a or b. For example, neurons in primary visual cortex can be characterized by their preferred orientation angles, preferred spatial phases and frequencies, or other stimulus-related parameters (see chapter 2). In many of the examples in this chapter, we consider stimuli characterized by a single angle Θ, which represents, for example, the orientation of a visual stimulus. Individual neurons are identified by their preferred stimulus angles, which are typically the values of Θ for which they fire at maximum rates. Thus, neuron a is identified by an angle θ_a. The weight of the synapse from neuron b or neuron a' to neuron a is then expressed as a function of the preferred stimulus angles θ_b, $\theta_{a'}$ and θ_a of the pre- and postsynaptic neurons, $W_{ab} = W(\theta_a, \theta_b)$ or $M_{aa'} = M(\theta_a, \theta_{a'})$. We often consider cases in which these synaptic weight functions depend only on the difference between the pre- and postsynaptic angles, so that $W_{ab} = W(\theta_a - \theta_b)$ or $M_{aa'} = M(\theta_a - \theta_{a'})$.

ρ_θ density of coverage

In large networks, the preferred stimulus parameters for different neurons will typically take a wide range of values. In the models we consider, the number of neurons is large and the angles θ_a, for different values of a cover the range from 0 to 2π densely. For simplicity, we assume that this coverage is uniform so that the density of coverage, the number of neurons with preferred angles falling within a unit range, which we denote by ρ_θ, is constant. For mathematical convenience in these cases, we allow the preferred angles to take continuous values rather than restricting them to the actual discrete values θ_a for $a = 1, 2, \ldots, N$. Thus, we label the neurons by a continuous angle θ and express the firing rate as a function of θ, so that $u(\theta)$ and $v(\theta)$ describe the firing rates of neurons with preferred angles θ. Similarly, the synaptic weight matrices **W** and **M** are replaced by functions $W(\theta, \theta')$ and $M(\theta, \theta')$ that characterizes the strength of synapses from a presynaptic neuron with preferred angle θ' to a postsynaptic neuron with preferred angle θ in the feedforward and recurrent case, respectively.

If the number of neurons in a network is large and the density of coverage of preferred stimulus values is high, we can approximate the sums in equation 7.10 by integrals over θ'. The number of postsynaptic neurons with preferred angles within a range $\Delta\theta'$ is $\rho_\theta \Delta\theta'$, so, when we take the limit $\Delta\theta' \to 0$, the integral over θ' appears multiplied by the density factor ρ_θ. Thus, in the case of continuous labeling of neurons, equation 7.10 becomes (for constant ρ_θ)

continuous model

$$\tau_r \frac{dv(\theta)}{dt} = -v(\theta) + F\left(\rho_\theta \int_{-\pi}^{\pi} d\theta'\, W(\theta,\theta')u(\theta') + M(\theta,\theta')v(\theta')\right) . \quad (7.14)$$

As we did previously in equation 7.11, we can write the first term inside the integral of this expression as an input function $h(\theta)$. We make frequent use of continuous labeling for network models, and we often approximate sums over neurons by integrals over their preferred stimulus parameters.

7.3 Feedforward Networks

Substantial computations can be performed by feedforward networks in the absence of recurrent connections. Much of the work done on feedforward networks centers on plasticity and learning, as discussed in the following chapters. Here, we present an example of the computational power of feedforward circuits, the calculation of the coordinate transformations needed in visually guided reaching tasks.

Neural Coordinate Transformations

Reaching for a viewed object requires a number of coordinate transformations that turn information about where the image of the object falls on the retina into movement commands in shoulder-, arm-, or hand-based coordinates. To perform a transformation from retinal to body-based coordinates, information about the retinal location of an image and about the direction of gaze relative to the body must be combined. Figure 7.4A and B illustrate, in a one-dimensional example, how a rotation of the eyes affects the relationship between gaze direction, retinal location, and location relative to the body. Figure 7.4C introduces the notation we use. The angle g describes the orientation of a line extending from the head to the point of visual fixation. The visual stimulus in retinal coordinates is given by the angle s between this line and a line extending out to the target. The angle describing the reach direction, the direction to the target relative to the body, is the sum $s + g$.

Visual neurons have receptive fields fixed to specific locations on the retina. Neurons in motor areas can display visually evoked responses that are not tied to specific retinal locations, but rather depend on the relationship of a visual image to various parts of the body. Figures 7.5A and B show tuning curves of a neuron in the premotor cortex of a monkey that

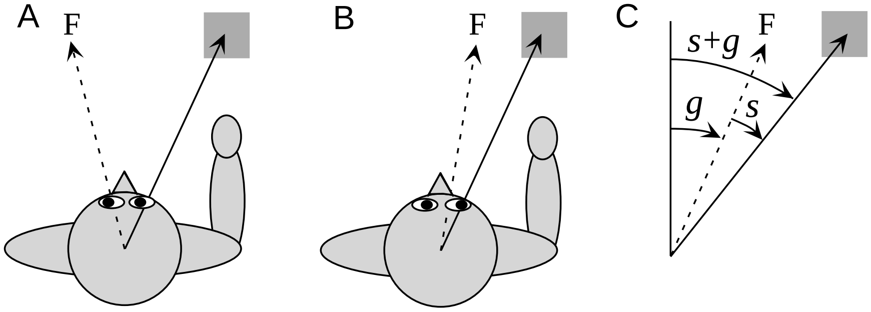

Figure 7.4: Coordinate transformations during a reaching task. A, B) The location of the target (the grey square) relative to the body is the same in A and B, and thus the movements required to reach toward it are identical. However, the image of the object falls on different parts of the retina in A and B due to a shift in the gaze direction produced by an eye rotation that shifts the fixation point F. C) The angles used in the analysis: s is the angle describing the location of the stimulus (the target) in retinal coordinates, that is, relative to a line directed to the fixation point; g is the gaze angle, indicating the direction of gaze relative to an axis straight out from the body. The direction of the target relative to the body-based axis is $s+g$.

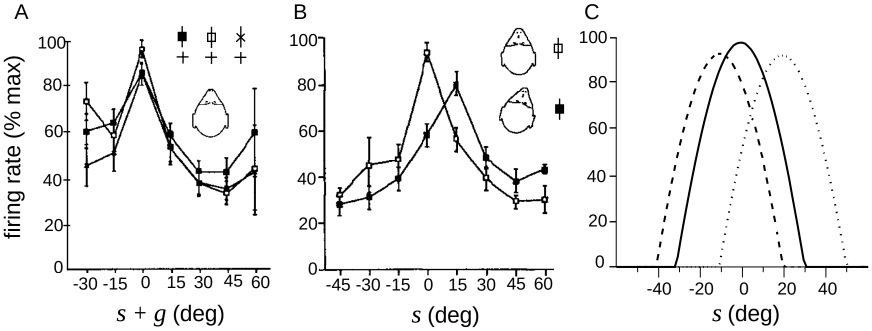

Figure 7.5: Tuning curves of a visually responsive neuron in the premotor cortex of a monkey. Incoming objects approaching at various angles provided the visual stimulation. A) When the monkey fixated on the three points denoted by the cross symbols, the response tuning curve did not shift with the eyes. In this panel, unlike B and C, the horizontal axis refers to the stimulus location in head-based, not retinal, coordinates ($s+g$, not s). B) Turning the monkey's head by 15° produced a 15° shift in the response tuning curve as a function of retinal location, indicating that this neuron encoded the stimulus direction in head-based coordinates. C) Model tuning curves based on equation 7.15 shift their retinal tuning to remain constant in body-based coordinates. The solid, heavy dashed, and light dashed curves refer to $g = 0°$, $10°$, and $-20°$ respectively. The small changes in amplitude arise from the limited range of preferred retinal location and gaze angles in the model. (A,B adapted from Graziano et al., 1997; C adapted from Salinas and Abbott, 1995.)

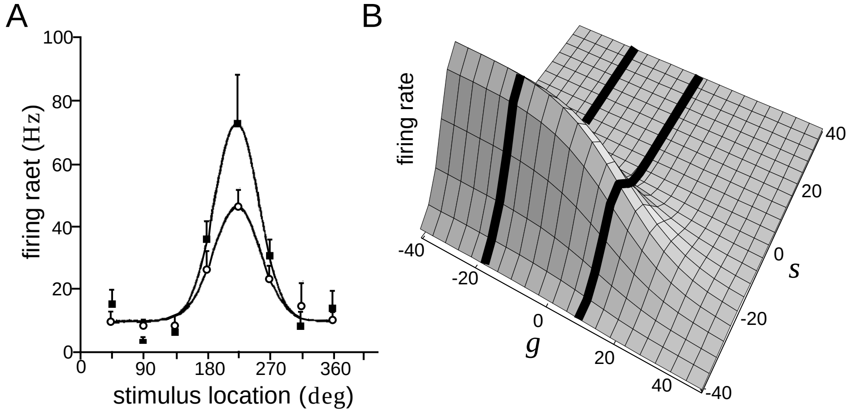

Figure 7.6: Gaze-dependent gain modulation of visual responses of neurons in posterior parietal cortex. A) Average firing-rate tuning curves of an area 7a neuron as a function of the location of the spot of light used to evoke the response. Stimulus location is measure as an angle around a circle of possible locations on the screen and is related to, but not equal to, our stimulus variable s. The two curves correspond to the same visual image but with two different gaze directions. B) A three-dimensional plot of the activity of a model neuron as a function of both retinal position and gaze direction. The striped bands correspond to tuning curves with different gains similar to those shown in A. (A adapted from Brotchie et al., 1995; B adapted from Pouget and Sejnowski, 1995.)

responded to visual images of approaching objects. Surprisingly, when the head of the monkey was held stationary during fixation on three different targets, the tuning curves did not shift as the eyes rotated (figure 7.5A). Although the recorded neurons respond to visual stimuli, the responses do not depend directly on the location of the image on the retina. When the head of the monkey is rotated but the fixation point remains the same, the tuning curves shift by precisely the amount of the head rotation (figure 7.5B). Thus, these neurons encode the location of the image in head- or body-based, not retinal, coordinates.

To account for these data, we need to construct a model neuron that is driven by visual input, but that nonetheless has a tuning curve for image location that is not a function of s, the retinal location of the image, but of $s + g$, the location of the object in body-based coordinates. A possible basis for this construction is provided by a combined representation of s and g by neurons in area 7a in the posterior parietal cortex of the monkey. Recordings made in area 7a reveal neurons that fire at rates that depend on both the location of the stimulating image on the retina and on the direction of gaze (figure 7.6A). The response tuning curves, expressed as functions of the retinal location of the stimulus, do not shift when the direction of gaze is varied. However, shifts of gaze direction affect the magnitude of the visual response. Thus, responses in area 7a exhibit gaze-dependent gain modulation of a retinotopic visual receptive field.

gain modulation

Figure 7.6B shows a mathematical description of a gain-modulated tuning curve. The response tuning curve is expressed as a product of a Gaussian function of $s-\xi$, where ξ is the preferred retinal location ($\xi=-20^\circ$ in figure 7.6B), and a sigmoid function of $g-\gamma$, where γ is the gaze direction producing half of the maximum gain ($\gamma=20^\circ$ in figure 7.6B). Although it does not correspond to the maximum neural response, we refer to γ as the 'preferred' gaze direction.

To model a neuron with a body-centered response tuning curve, we construct a feedforward network with a single output unit representing, for example, the premotor neuron shown in figure 7.5. The input layer of the network consists of a population of area 7a neurons with gain-modulated responses similar to those shown in figure 7.6B. Neurons with gains that both increase and decrease as a function of g are included in the model. The average firing rates of the input layer neurons are described by tuning curves $u = f_u(s-\xi, g-\gamma)$ with the different neurons taking a range of ξ and γ values.

We use continuous labeling of neurons, and replace the sum over presynaptic neurons by an integral over their ξ and γ values, inserting the appropriate density factors ρ_ξ and ρ_γ, which we assume are constant. The steady-state response of the single output neuron is determined by the continuous analog of equation 7.5. The synaptic weight from a presynaptic neuron with preferred stimulus location ξ and preferred gaze direction γ is denoted by $w(\xi,\gamma)$, so the steady-state response of the output neurons is given by

$$v_\infty = F\left(\rho_\xi\rho_\gamma \int d\xi d\gamma\, w(\xi,\gamma) f_u(s-\xi, g-\gamma)\right) . \tag{7.15}$$

For the output neuron to respond to stimulus location in body-based coordinates, its firing rate must be a function of $s+g$. To see if this is possible, we shift the integration variables in 7.15 by $\xi \to \xi-g$ and $\gamma \to \gamma+g$. Ignoring effects from the end points of the integration (which is valid if s and g are not too close to these limits), we find

$$v_\infty = F\left(\rho_\xi\rho_\gamma \int d\xi d\gamma\, w(\xi-g, \gamma+g) f_u(s+g-\xi, -\gamma)\right) . \tag{7.16}$$

This is a function of $s+g$ provided that $w(\xi-g,\gamma+g) = w(\xi,\gamma)$, which holds if $w(\xi,\gamma)$ is a function of the sum $\xi+\gamma$. Thus, the coordinate transformation can be accomplished if the synaptic weight from a given neuron depends only the sum of its preferred retinal and gaze angles. It has been suggested that weights of this form can arise naturally from random hand and gaze movements through correlation-based synaptic modification of the type discussed in chapter 8.

Figure 7.5C shows responses predicted by equation 7.15 when the synaptic weights are given by a function $w(\xi+\gamma)$. The retinal location of the tuning curve shifts as a function of gaze direction, but would remain stationary if it were plotted instead as a function of $s+g$. This can be seen by noting that the peaks of all three curves in figure 7.5C occur at $s+g=0$.

Gain-modulated neurons provide a general basis for combining two different input signals in a nonlinear way. In the network we studied, it is possible to find appropriate synaptic weights $w(\xi, \gamma)$ to construct output neurons with a wide range of response tuning curves expressed as functions of s and g. The mechanism by which sensory and modulatory inputs combine in a multiplicative way in gain-modulated neurons is not known. Later in this chapter, we discuss a recurrent network model for generating gain-modulated responses.

7.4 Recurrent Networks

Recurrent networks have richer dynamics than feedforward networks, but they are more difficult to analyze. To get a feel for recurrent circuitry, we begin by analyzing a linear model, that is, a model for which the relationship between firing rate and synaptic current is linear, $\mathbf{F}(\mathbf{h}+\mathbf{M}\cdot\mathbf{r}) = \mathbf{h}+\mathbf{M}\cdot\mathbf{r}$. The linear approximation is a drastic one that allows, among other things, the components of $\mathbf{v}$ to become negative, which is impossible for real firing rates. Furthermore, some of the features we discuss in connection with linear, as opposed to nonlinear, recurrent networks can also be achieved by a feedforward architecture. Nevertheless, the linear model is extremely useful for exploring properties of recurrent circuits, and this approach will be used both here and in the following chapters. In addition, the analysis of linear networks forms the basis for studying the stability properties of nonlinear networks. We augment the discussion of linear networks with results from simulations of nonlinear networks.

Linear Recurrent Networks

Under the linear approximation, the recurrent model of equation 7.11 takes the form

linear recurrent model

$$\tau_r \frac{d\mathbf{v}}{dt} = -\mathbf{v} + \mathbf{h} + \mathbf{M}\cdot\mathbf{v}\,. \tag{7.17}$$

Because the model is linear, we can solve analytically for the vector of output rates $\mathbf{v}$ in terms of the feedforward inputs $\mathbf{h}$ and the initial values $\mathbf{v}(0)$. The analysis is simplest when the recurrent synaptic weight matrix is symmetric, and we assume this to be the case. Equation 7.17 can be solved by expressing $\mathbf{v}$ in terms of the eigenvectors of $\mathbf{M}$. The eigenvectors $\mathbf{e}_\mu$ for $\mu = 1, 2, \ldots, N_v$ satisfy

eigenvector $\mathbf{e}$

$$\mathbf{M}\cdot\mathbf{e}_\mu = \lambda_\mu \mathbf{e}_\mu \tag{7.18}$$

for some value of the constant λ_μ which is called the eigenvalue. For a

eigenvalue λ

symmetric matrix, the eigenvectors are orthogonal, and they can be normalized to unit length so that $\mathbf{e}_\mu \cdot \mathbf{e}_\nu = \delta_{\mu\nu}$. Such eigenvectors define an orthogonal coordinate system or basis that can be used to represent any N_v-dimensional vector. In particular, we can write

eigenvector expansion

$$\mathbf{v}(t) = \sum_{\mu=1}^{N_v} c_\mu(t)\mathbf{e}_\mu \tag{7.19}$$

where $c_\mu(t)$ for $\mu = 1, 2, \ldots, N_v$ are a set of time-dependent coefficients describing $\mathbf{v}(t)$.

It is easier to solve equation 7.17 for the coefficients c_μ than for $\mathbf{v}$ directly. Substituting the expansion 7.19 into equation 7.17 and using property 7.18, we find that

$$\tau_r \sum_{\mu=1}^{N_v} \frac{dc_\mu}{dt}\mathbf{e}_\mu = -\sum_{\mu=1}^{N_v}(1-\lambda_\mu)c_\mu(t)\mathbf{e}_\mu + \mathbf{h}\,. \tag{7.20}$$

The sum over μ can be eliminated by taking the dot product of each side of this equation with one of the eigenvectors, $\mathbf{e}_\nu$, and using the orthogonality property $\mathbf{e}_\mu \cdot \mathbf{e}_\nu = \delta_{\mu\nu}$ to obtain

$$\tau_r \frac{dc_\nu}{dt} = -(1-\lambda_\nu)c_\nu(t) + \mathbf{e}_\nu \cdot \mathbf{h}\,. \tag{7.21}$$

The critical feature of this equation is that it involves only one of the coefficients, c_ν. For time-independent inputs $\mathbf{h}$, the solution of equation 7.44 is

$$c_\nu(t) = \frac{\mathbf{e}_\nu \cdot \mathbf{h}}{1-\lambda_\nu}\left(1 - \exp\left(-\frac{t(1-\lambda_\nu)}{\tau_r}\right)\right) + c_\nu(0)\exp\left(-\frac{t(1-\lambda_\nu)}{\tau_r}\right) \tag{7.22}$$

where $c_\nu(0)$ is the value of c_ν at time zero, which is given in terms of the initial firing-rate vector $\mathbf{v}(0)$ by $c_\nu(0) = \mathbf{e}_\nu \cdot \mathbf{v}(0)$.

Equation 7.22 has several important characteristics. If $\lambda_\nu > 1$, the exponential functions grow without bound as time increases, reflecting a fundamental instability of the network. If $\lambda_\nu < 1$, c_ν approaches the steady-state value $\mathbf{e}_\nu \cdot \mathbf{h}/(1-\lambda_\nu)$ exponentially with time constant $\tau_r/(1-\lambda_\nu)$. This steady-state value is proportional to $\mathbf{e}_\nu \cdot \mathbf{h}$, which is the projection of the input vector onto the relevant eigenvector. For $0 < \lambda_\nu < 1$, the steady-state value is amplified relative to this projection by the factor $1/(1-\lambda_\nu)$, which is greater than one. The approach to equilibrium is slowed relative to the basic time constant τ_r by an identical factor. The steady-state value of $\mathbf{v}(t)$, which we call $\mathbf{v}_\infty$, can be derived from equation 7.19 as

steady state $\mathbf{v}_\infty$

$$\mathbf{v}_\infty = \sum_{\nu=1}^{N_v} \frac{(\mathbf{e}_\nu \cdot \mathbf{h})}{1-\lambda_\nu}\mathbf{e}_\nu\,. \tag{7.23}$$

This steady-state response can also arise from a purely feedforward scheme if the feedforward weight matrix is chosen appropriately, as we invite the reader to verify as an exercise.

We have considered amplification when $0 < \lambda_1 < 1$. The linear network becomes unstable if $\lambda_1 > 1$. The case $\lambda_\nu = 1$ is special and will be discussed in a later section.

Selective Amplification

Suppose that one of the eigenvalues of a recurrent weight matrix, denoted by λ_1, is very close to one, and all the others are significantly smaller than 1. In this case, the denominator of the $\nu = 1$ term on the right side of equation 7.23 is near zero, and, unless $\mathbf{e}_1 \cdot \mathbf{h}$ is extremely small, this single term will dominate the sum. As a result, we can write

$$\mathbf{v}_\infty \approx \frac{(\mathbf{e}_1 \cdot \mathbf{h})\mathbf{e}_1}{1 - \lambda_1} . \tag{7.24}$$

Such a network performs selective amplification. The response is dominated by the projection of the input vector along the axis defined by $\mathbf{e}_1$, and the amplitude of the response is amplified by the factor $1/(1 - \lambda_1)$, which may be quite large if λ_1 is near one. The steady-state response of such a network, which is proportional to $\mathbf{e}_1$, therefore encodes an amplified projection of the input vector onto $\mathbf{e}_1$.

Further information can be encoded if more eigenvalues are close to one. Suppose, for example, that two eigenvectors, $\mathbf{e}_1$ and $\mathbf{e}_2$ have the same eigenvalue, $\lambda_1 = \lambda_2$, close to but less than one. Then, equation 7.24 is replaced by

$$\mathbf{v}_\infty \approx \frac{(\mathbf{e}_1 \cdot \mathbf{h})\mathbf{e}_1 + (\mathbf{e}_2 \cdot \mathbf{h})\mathbf{e}_2}{1 - \lambda_1} \tag{7.25}$$

which shows that the network now amplifies and encodes the projection of the input vector onto the plane defined by $\mathbf{e}_1$ and $\mathbf{e}_2$. In this case, the activity pattern of the network is not simply scaled when the input changes. Instead, changes in the input shift both the magnitude and pattern of network activity. Eigenvectors that share the same eigenvalue are termed degenerate, and degeneracy is often the result of a symmetry. In the examples considered in this chapter, degeneracy arises from invariance to shifts of the parameter θ by a constant amount. Degeneracy is not limited to just two eigenvectors. A recurrent network with n degenerate eigenvalues near one can amplify and encode a projection of the input vector from the N-dimensional space in which it is defined onto the n-dimensional subspace spanned by the degenerate eigenvectors.

Input Integration

If the recurrent weight matrix has an eigenvalue exactly equal to one, $\lambda_1 = 1$, and all the other eigenvalues satisfy $\lambda_\nu < 1$, a linear recurrent network can act as an integrator of its input. In this case, c_1 satisfies the equation

$$\tau_r \frac{dc_1}{dt} = \mathbf{e}_1 \cdot \mathbf{h} \qquad (7.26)$$

obtained by setting $\lambda_1 = 1$ in equation 7.44. For arbitrary time-dependent inputs, the solution of this equation is

$$c_1(t) = c_1(0) + \frac{1}{\tau_r}\int_0^t dt' \, \mathbf{e}_1 \cdot \mathbf{h}(t') \, . \qquad (7.27)$$

If $\mathbf{h}(t)$ is constant, $c_1(t)$ grows linearly with t. This explains why equation 7.24 diverges as $\lambda_1 \to 1$. Suppose, instead, that $\mathbf{h}(t)$ is nonzero for a while, and then is set to zero for an extended period of time. When $\mathbf{h} = 0$, equation 7.22 shows that $c_\nu \to 0$ for all $\nu \neq 1$, because for these eigenvectors $\lambda_\nu < 1$. Assuming that $c_1(0) = 0$, this means that, after such a period, the firing-rate vector is given, from equation 7.27 and 7.19, by

network integration

$$\mathbf{v}(t) \approx \frac{\mathbf{e}_1}{\tau_r}\int_0^t dt' \, \mathbf{e}_1 \cdot \mathbf{h}(t') \, . \qquad (7.28)$$

This shows that the network activity provides a measure of the running integral of the projection of the input vector onto $\mathbf{e}_1$. One consequence of this is that the activity of the network does not cease if $\mathbf{h} = 0$, provided that the integral up to that point in time is nonzero. The network thus exhibits sustained activity in the absence of input, which provides a memory of the integral of prior input.

Networks in the brain stem of vertebrates responsible for maintaining eye position appear to act as integrators, and networks similar to the one we have been discussing have been suggested as models of this system. As outlined in figure 7.7, eye position changes in response to bursts of activity in ocular motor neurons located in the brain stem. Neurons in the medial vestibular nucleus and prepositus hypoglossi appear to integrate these motor signals to provide a persistent memory of eye position. The sustained firing rates of these neurons are approximately proportional to the angular orientation of the eyes in the horizontal direction, and activity persists at an approximately constant rate when the eyes are held fixed (bottom trace in figure 7.7).

The ability of a linear recurrent network to integrate and display persistent activity relies on one of the eigenvalues of the recurrent weight matrix being exactly one. Any deviation from this value will cause the persistent activity to change over time. Eye position does indeed drift, but matching the performance of the ocular positioning system requires fine tuning of the eigenvalue to a value extremely close to one. Including nonlinear interactions does not alleviate the need for a precisely tuned weight matrix.

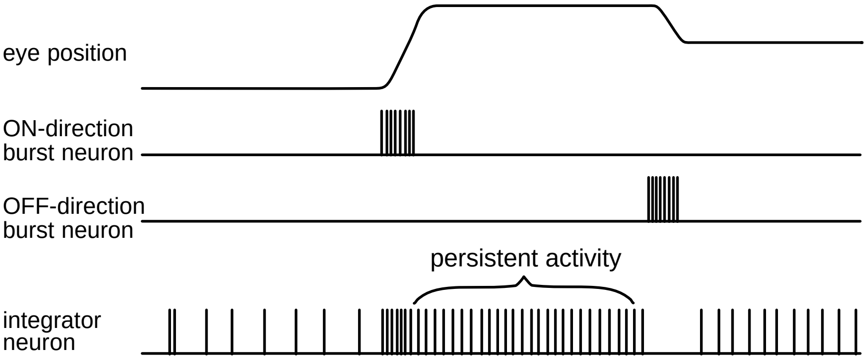

Figure 7.7: Cartoon of burst and integrator neurons involved in horizontal eye positioning. The upper trace represents horizontal eye position during two saccadic eye movements. Motion of the eye is driven by burst neurons that move the eyes in opposite directions (second and third traces from top). The steady-state firing rate (labeled persistent activity) of the integrator neuron is proportional to the time integral of the burst rates, integrated positively for the ON-direction burst neuron and negatively for the OFF-direction burst neuron, and thus provides a memory trace of the maintained eye position. (Adapted from Seung et al., 2000.)

Synaptic modification rules can be used to establish the necessary synaptic weights, but it is not clear how such precise tuning is accomplished in the biological system.

Continuous Linear Recurrent Networks

For a linear recurrent network with continuous labeling, the equation for the firing rate $v(\theta)$ of a neuron with preferred stimulus angle θ is a linear version of equation 7.14,

$$\tau_r \frac{dv(\theta)}{dt} = -v(\theta) + h(\theta) + \rho_\theta \int_{-\pi}^{\pi} d\theta' \, M(\theta - \theta') v(\theta') \tag{7.29}$$

where $h(\theta)$ is the feedforward input to a neuron with preferred stimulus angle θ, and we have assumed a constant density ρ_θ. Because θ is an angle, h, M, and v must all be periodic functions with period 2π. By making M a function of $\theta - \theta'$, we are imposing a symmetry with respect to translations or shifts of the angle variables on the network. In addition, we assume that M is an even function, $M(\theta - \theta') = M(\theta' - \theta)$. This is the analog, in a continuously labeled model, of a symmetric synaptic weight matrix.

Equation 7.29 can be solved by methods similar to those used for discrete networks. We introduce eigenfunctions that satisfy

$$\rho_\theta \int_{-\pi}^{\pi} d\theta' \, M(\theta - \theta') e_\mu(\theta') = \lambda_\mu e_\mu(\theta) \, . \tag{7.30}$$

We leave it as an exercise to show that the eigenfunctions (normalized so that ρ_θ times the integral from $-\pi$ to π of their square is one) are $1/\sqrt{2\pi\rho_\theta}$, corresponding to $\mu = 0$, and $\cos(\mu\theta)/\sqrt{\pi\rho_\theta}$ and $\sin(\mu\theta)/\sqrt{\pi\rho_\theta}$ for $\mu = 1, 2, \ldots$. The eigenvalues are identical for the sine and cosine eigenfunctions and are given (including the case $\mu = 0$) by

$$\lambda_\mu = \rho_\theta \int_{-\pi}^{\pi} d\theta'\, M(\theta') \cos(\mu\theta') \,. \tag{7.31}$$

The identity of the eigenvalues for the cosine and sine eigenfunctions reflects a degeneracy that arises from the invariance of the network to shifts of the angle labels.

The steady-state firing rates for a constant input are given by the continuous analog of equation 7.23,

$$\begin{aligned} v_\infty(\theta) &= \frac{1}{1-\lambda_0} \int_{-\pi}^{\pi} \frac{d\theta'}{2\pi}\, h(\theta') \\ &+ \sum_{\mu=1}^{\infty} \frac{\cos(\mu\theta)}{1-\lambda_\mu} \int_{-\pi}^{\pi} \frac{d\theta'}{\pi}\, h(\theta') \cos(\mu\theta') \\ &+ \sum_{\mu=1}^{\infty} \frac{\sin(\mu\theta)}{1-\lambda_\mu} \int_{-\pi}^{\pi} \frac{d\theta'}{\pi}\, h(\theta') \sin(\mu\theta') \,. \end{aligned} \tag{7.32}$$

Fourier series

The integrals in this expression are the coefficients in a Fourier series for the function h and are know as cosine and sine Fourier integrals (see the Mathematical Appendix).

Figure 7.8 shows an example of selective amplification by a linear recurrent network. The input to the network, shown in panel A of figure 7.8, is a cosine function that peaks at 0° to which random noise has been added. Figure 7.8C shows Fourier amplitudes for this input. The Fourier amplitude is the square root of the sum of the squares of the cosine and sine Fourier integrals. No particular μ value is overwhelmingly dominant. In this and the following examples, the recurrent connections of the network are given by

$$M(\theta - \theta') = \frac{\lambda_1}{\pi\rho_\theta} \cos(\theta - \theta') \tag{7.33}$$

which has all eigenvalues except λ_1 equal to zero. The network model shown in figure 7.8 has $\lambda_1 = 0.9$, so that $1/(1-\lambda_1) = 10$. Input amplification can be quantified by comparing the Fourier amplitude of v_∞, for a given μ value, with the analogous amplitude for the input h. According to equation 7.32, the ratio of these quantities is $1/(1-\lambda_\mu)$, so, in this case, the $\mu = 1$ amplitude should be amplified by a factor of ten while all other amplitudes are unamplified. This factor of ten amplification can be seen by comparing the $\mu = 1$ Fourier amplitudes in figures 7.8C and D (note the different scales for the vertical axes). All the other components are unamplified. As a result, the output of the network is primarily in the form of a cosine function with $\mu = 1$, as seen in figure 7.8B.

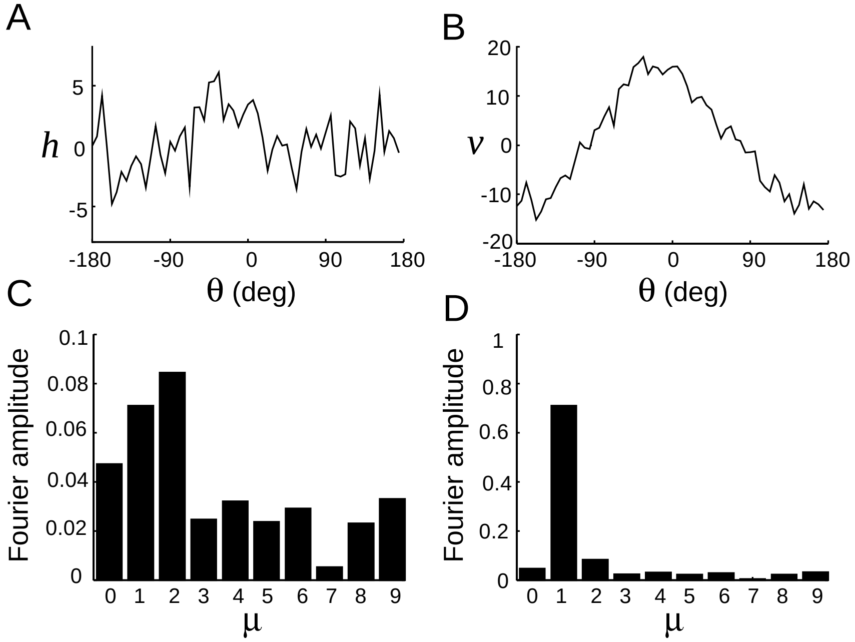

Figure 7.8: Selective amplification in a linear network. A) The input to the neurons of the network as a function of their preferred stimulus angle. B) The activity of the network neurons plotted as a function of their preferred stimulus angle in response to the input of panel A. C) The Fourier transform amplitudes of the input shown in panel A. D) The Fourier transform amplitudes of the output shown in panel B. The recurrent coupling of this network model took the form of equation 7.33 with $\lambda_1 = 0.9$. (This figure, and figures 7.9, 7.12, 7.13, and 7.14, were generated using software from Carandini and Ringach, 1998.)

Nonlinear Recurrent Networks

A linear model does not provide an adequate description of the firing rates of a biological neural network. The most significant problem is that the firing rates in a linear network can take negative values. This problem can be fixed by introducing rectification into equation 7.11 by choosing

rectification

$$\mathbf{F}(\mathbf{h} + \mathbf{M} \cdot \mathbf{r}) = [\mathbf{h} + \mathbf{M} \cdot \mathbf{r} - \boldsymbol{\gamma}]_+ \, . \qquad (7.34)$$

where $\boldsymbol{\gamma}$ is a vector of threshold values that we often take to be $\mathbf{0}$. In this section, we show some examples illustrating the effect of including such a rectifying nonlinearity. Some of the features of linear recurrent networks remain when rectification is included, but several new features also appear.

In the examples given below, we consider a continuous model, similar to that of equation 7.29, with recurrent couplings given by equation 7.33, but

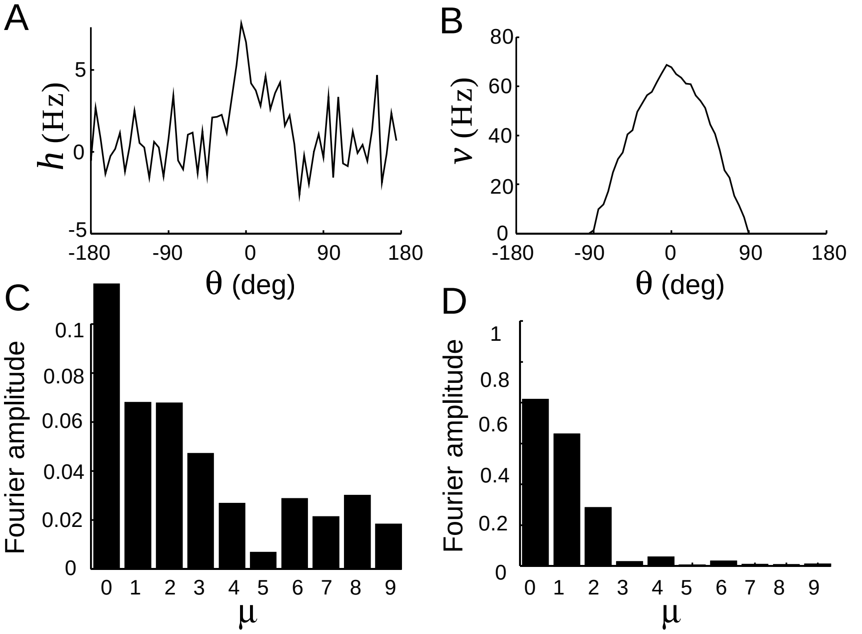

Figure 7.9: Selective amplification in a recurrent network with rectification. A) The input $h(\theta)$ of the network plotted as a function of preferred angle. B) The steady-state output $v(\theta)$ as a function of preferred angle. C) Fourier transform amplitudes of the input $h(\theta)$. D) Fourier transform amplitudes of the output $v(\theta)$. The recurrent coupling took the form 7.33 with $\lambda_1 = 1.9$.

now including a rectification nonlinearity, so that

$$\tau_r \frac{dv(\theta)}{dt} = -v(\theta) + \left[h(\theta) + \frac{\lambda_1}{\pi} \int_{-\pi}^{\pi} d\theta' \, \cos(\theta - \theta') v(\theta') \right]_+ . \tag{7.35}$$

If λ_1 is not too large, this network converges to a steady state for any constant input (we consider conditions for steady-state convergence in a later section), and therefore we often limit the discussion to the steady-state activity of the network.

Nonlinear Amplification

Figure 7.9 shows the nonlinear analog of the selective amplification shown for a linear network in figure 7.8. Once again, a noisy input (figure 7.9A) generates a much smoother output response profile (figure 7.9B). The output response of the rectified network corresponds roughly to the positive part of the sinusoidal response profile of the linear network (figure 7.8B). The negative output has been eliminated by the rectification. Because fewer neurons in the network have nonzero responses than in the linear case, the value of the parameter λ_1 in equation 7.33 has been increased to

1.9. This value, being larger than one, would lead to an unstable network in the linear case. While nonlinear networks can also be unstable, the restriction to eigenvalues less than one is no longer the relevant condition.

In a nonlinear network, the Fourier analysis of the input and output responses is no longer as informative as it is for a linear network. Due to the rectification, the $\nu = 0, 1$, and 2 Fourier components are all amplified (figure 7.9D) compared to their input values (figure 7.9C). Nevertheless, except for rectification, the nonlinear recurrent network amplifies the input signal selectively in a similar manner as the linear network.

A Recurrent Model of Simple Cells in Primary Visual Cortex

In chapter 2, we discussed a feedforward model in which the elongated receptive fields of simple cells in primary visual cortex were formed by summing the inputs from lateral geniculate (LGN) neurons with their receptive fields arranged in alternating rows of ON and OFF cells. While this model quite successfully accounts for a number of features of simple cells, such as orientation tuning, it is difficult to reconcile with the anatomy and circuitry of the cerebral cortex. By far the majority of the synapses onto any cortical neuron arise from other cortical neurons, not from thalamic afferents. Therefore, feedforward models account for the response properties of cortical neurons while ignoring the inputs that are numerically most prominent. The large number of intracortical connections suggests, instead, that recurrent circuitry might play an important role in shaping the responses of neurons in primary visual cortex.

Ben-Yishai, Bar-Or, and Sompolinsky (1995) developed a model at the other extreme, for which recurrent connections are the primary determiners of orientation tuning. The model is similar in structure to the model of equations 7.35 and 7.33, except that it includes a global inhibitory interaction. In addition, because orientation angles are defined over the range from $-\pi/2$ to $\pi/2$, rather than over the full 2π range, the cosine functions in the model have extra factors of 2 in them. The basic equation of the model, as we implement it, is

$$\tau_r \frac{dv(\theta)}{dt} = -v(\theta) + \left[h(\theta) + \int_{-\pi/2}^{\pi/2} \frac{d\theta'}{\pi} \left(-\lambda_0 + \lambda_1 \cos(2(\theta - \theta')) \right) v(\theta') \right]_+ \tag{7.36}$$

where $v(\theta)$ is the firing rate of a neuron with preferred orientation θ.

The input to the model represents the orientation-tuned feedforward input arising from ON-center and OFF-center LGN cells responding to an oriented image. As a function of preferred orientation, the input for an image with orientation angle $\Theta = 0$ is

$$h(\theta) = Ac\,(1 - \epsilon + \epsilon \cos(2\theta)) \tag{7.37}$$

where A sets the overall amplitude and c is equal to the image contrast. The factor ϵ controls how strongly the input is modulated by the orientation angle. For $\epsilon = 0$, all neurons receive the same input, while $\epsilon = 0.5$ produces the maximum modulation consistent with a positive input. We study this model in the case when ϵ is small, which means that the input is only weakly tuned for orientation and any strong orientation selectivity must arise through recurrent interactions.

To study orientation selectivity, we want to examine the tuning curves of individual neurons in response to stimuli with different orientation angles Θ. The plots of network responses that we have been using show the firing rates $v(\theta)$ of all the neurons in the network as a function of their preferred stimulus angles θ when the input stimulus has a fixed value, typically $\Theta = 0$. As a consequence of the translation invariance of the network model, the response for other values of Θ can be obtained simply by shifting this curve so that it plots $v(\theta - \Theta)$. Furthermore, except for the asymmetric effects of noise on the input, $v(\theta - \Theta)$ is a symmetric function. These features follow from the fact that the network we are studying is invariant with respect to translations and sign changes of the angle variables that characterize the stimulus and response selectivities. An important consequence of this result is that the curve $v(\theta)$, showing the response of the entire population, can also be interpreted as the tuning curve of a single neuron. If the response of the population to a stimulus angle Θ is $v(\theta - \Theta)$, the response of a single neuron with preferred angle $\theta = 0$ is $v(-\Theta) = v(\Theta)$ from the symmetry of v. Because $v(\Theta)$ is the tuning curve of a single neuron with $\theta = 0$ to a stimulus angle Θ, the plots we show of $v(\theta)$ can be interpreted in a dual way, as both population responses and individual neuronal tuning curves.

Figure 7.10A shows the feedforward input to the model network for four different levels of contrast. Because the parameter ϵ was chosen to be 0.1, the modulation of the input as a function of orientation angle is small. Due to network amplification, the response of the network is much more strongly tuned to orientation (figure 7.10B). This is the result of the selective amplification of the tuned part of the input by the recurrent network. The modulation and overall height of the input curve in figure 7.10A increase linearly with contrast. The response shown in figure 7.10B, interpreted as a tuning curve, increases in amplitude for higher contrast, but does not broaden. This can be seen by noting that all four curves in figure 7.10B go to zero at the same two points. This effect, which occurs because the shape and width of the response tuning curve are determined primarily by the recurrent interactions within the network, is a feature of orientation curves of real simple cells, as seen in figure 7.10C. The width of the tuning curve can be reduced by including a positive threshold in the response function of equation 7.34, or by changing the amount of inhibition, but it stays roughly constant as a function of stimulus strength.

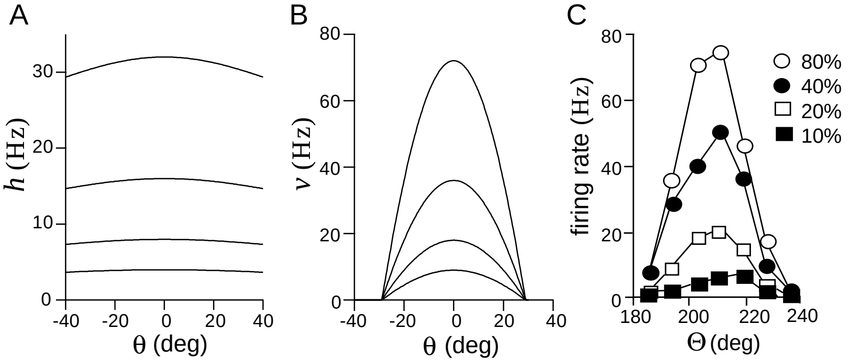

Figure 7.10: The effect of contrast on orientation tuning. A) The feedforward input as a function of preferred orientation. The four curves, from top to bottom, correspond to contrasts of 80%, 40%, 20%, and 10%. B) The output firing rates in response to different levels of contrast as a function of orientation preference. These are also the response tuning curves of a single neuron with preferred orientation zero. As in A, the four curves, from top to bottom, correspond to contrasts of 80%, 40%, 20%, and 10%. The recurrent model had $\lambda_0 = 7.3$, $\lambda_1 = 11$, $A = 40$ Hz, and $\epsilon = 0.1$. C) Tuning curves measure experimentally at four contrast levels as indicated in the legend. (C adapted from Sompolinsky and Shapley, 1997; based on data from Sclar and Freeman, 1982.)

A Recurrent Model of Complex Cells in Primary Visual Cortex

In the model of orientation tuning discussed in the previous section, recurrent amplification enhances selectivity. If the pattern of network connectivity amplifies nonselective rather than selective responses, recurrent interactions can also decrease selectivity. Recall from chapter 2 that neurons in the primary visual cortex are classified as simple or complex depending on their sensitivity to the spatial phase of a grating stimulus. Simple cells respond maximally when the spatial positioning of the light and dark regions of a grating matches the locations of the ON and OFF regions of their receptive fields. Complex cells do not have distinct ON and OFF regions in their receptive fields and respond to gratings of the appropriate orientation and spatial frequency relatively independently of where their light and dark stripes fall. In other words, complex cells are insensitive to spatial phase.

Chance, Nelson, and Abbott (1999) showed that complex cell responses could be generated from simple cell responses by a recurrent network. As in chapter 2, we label spatial phase preferences by the angle ϕ. The feedforward input $h(\phi)$ in the model is set equal to the rectified response of a simple cell with preferred spatial phase ϕ (figure 7.11A). Each neuron in the network is labeled by the spatial phase preference of its feedforward input. The network neurons also receive recurrent input given by the weight function $M(\phi - \phi') = \lambda_1/(2\pi\rho_\phi)$ that is the same for all connected neuron pairs. As a result, their firing rates are determined by

$$\tau_r \frac{dv(\phi)}{dt} = -v(\phi) + \left[h(\phi) + \frac{\lambda_1}{2\pi} \int_{-\pi}^{\pi} d\phi' \, v(\phi') \right]_+ . \qquad (7.38)$$

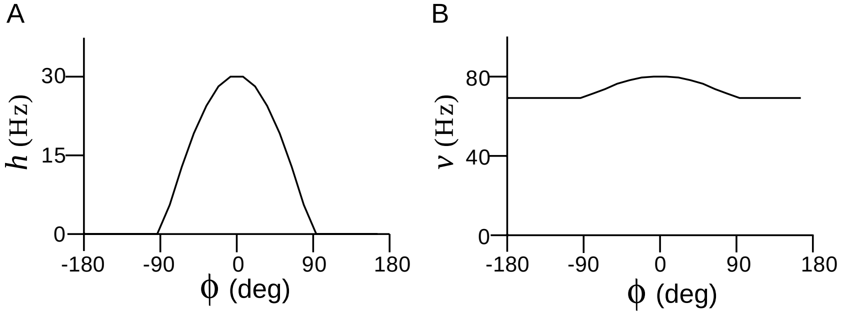

Figure 7.11: A recurrent model of complex cells. A) The input to the network as a function of spatial phase preference. The input $h(\phi)$ is equivalent to that of a simple cell with spatial phase preference ϕ responding to a grating of zero spatial phase. B) Network response, which can also be interpreted as the spatial phase tuning curve of a network neuron. The network was given by equation 7.38 with $\lambda_1 = 0.95$. (Adapted from Chance et al., 1999.)

In the absence of recurrent connections ($\lambda_1 = 0$), the response of a neuron labeled by ϕ is $v(\phi) = h(\phi)$, which is equal to the response of a simple cell with preferred spatial phase ϕ. However, for λ_1 sufficiently close to one, the recurrent model produces responses that resemble those of complex cells. Figure 7.11B shows the population response, or equivalently the single-cell response tuning curve, of the model in response to the tuned input shown in Figure 7.11A. The input, being the response of a simple cell, shows strong tuning for spatial phase. The output tuning curve, however, is almost constant as a function of spatial phase, like that of a complex cell. The spatial-phase insensitivity of the network response is due to the fact that the network amplifies the component of the input that is independent of spatial phase, because the eigenfunction of M with the largest eigenvalue is spatial-phase invariant. This changes simple cell inputs into complex cell outputs.

Winner-Take-All Input Selection

For a linear network, the response to two superimposed inputs is simply the sum of the responses to each input separately. Figure 7.12 shows one way in which a rectifying nonlinearity modifies this superposition property. In this case, the input to the recurrent network consists of activity centered around two preferred stimulus angles, $\pm 90^\circ$. The output of the nonlinear network shown in figure 7.12B is not of this form, but instead

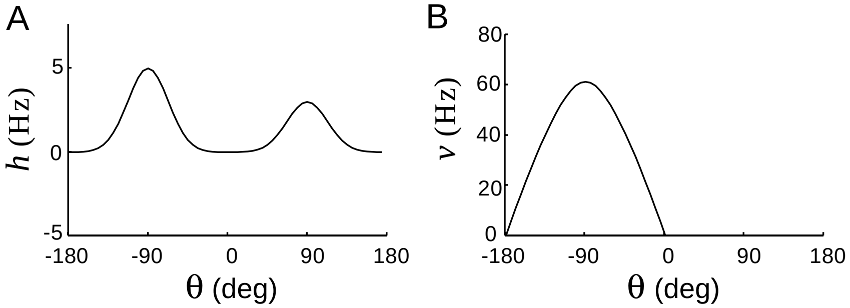

Figure 7.12: Winner-take-all input selection by a nonlinear recurrent network. A) The input to the network consisting of two peaks. B) The output of the network has a single peak at the location of the higher of the two peaks of the input. The model is the same as that used in figure 7.9.

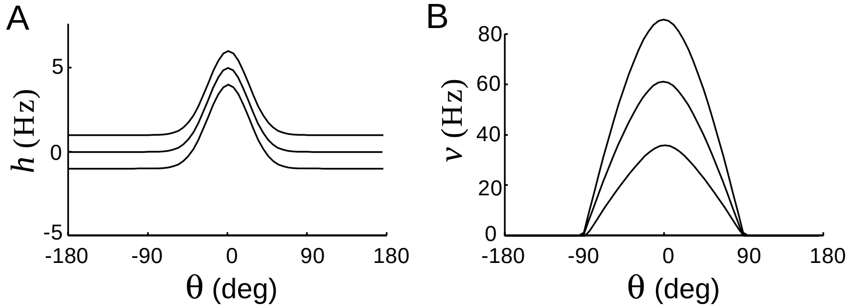

Figure 7.13: Effect of adding a constant to the input of a nonlinear recurrent network. A) The input to the network consists of a single peak to which a constant factor has been added. B) The gain-modulated output of the nonlinear network. The three curves correspond to the three input curves in panel A, in the same order. The model is the same as that used in figures 7.9 and 7.12.

has a single peak at the location of the input bump with the larger amplitude (the one at -90°). This occurs because the nonlinear recurrent network supports the stereotyped unimodal activity pattern seen in figure 7.12B, so a multimodal input tends to generate a unimodal output. The height of the input peak has a large effect in determining where the single peak of the network output is located, but it is not the only feature that determines the response. For example, the network output can favor a broader, lower peak over a narrower, higher one.

Gain Modulation

A nonlinear recurrent network can generate an output that resembles the gain-modulated responses of posterior parietal neurons shown in figure 7.6, as noted by Salinas and Abbott (1996). To obtain this result, we interpret the angle θ as a preferred direction in the visual field in retinal coordinates (the variable we called s earlier in the chapter). The signal corresponding to gaze direction (what we called g before) is represented as a constant input to all neurons irrespective of their preferred stimulus angle. Figure 7.13 shows the effect of adding such a constant term to the input of the nonlinear network. The input shown in figure 7.13A corresponds to a visual target located at a retinal position of 0°. The different lines show different values of the constant input, representing three different gaze directions. The responses shown in figure 7.13B all have localized activity centered around $\theta = 0^\circ$, indicating that the individual neurons have fixed tuning curves expressed in retinal coordinates. The effect of the constant input, representing gaze direction, is to scale up or gain modulate these tuning curves, producing a result similar to that shown in figure 7.6. The additive constant in the input shown in figure 7.13A has a multiplicative effect on the output activity shown in 7.13B. This is primarily due to the fact that the width of the activity profiles is fixed by the recurrent network interaction, so a constant positive input raises (and a negative input lowers) the peak of the response curve without broadening the base of the curve.

Sustained Activity

The effects illustrated in figures 7.12 and 7.13 arise because the nonlinear recurrent network has a stereotyped pattern of activity that is largely determined by interactions with other neurons in the network rather than by the feedforward input. If the recurrent connections are strong enough, the pattern of population activity, once established, can become independent of the structure of the input. For example, the recurrent network we have been studying can support a pattern of activity localized around a given preferred stimulus value, even when the input is uniform. This is seen in figure 7.14. The neurons of the network initially receive inputs that depend on their preferred angles, as seen in figure 7.14A. This produces a localized pattern of network activity (figure 7.14B). When the input is switched to the same constant value for all neurons (figure 7.14C), the network activity does not become uniform. Instead, it stays localized around the value $\theta = 0$ (figure 7.14D). This means that constant input can maintain a state that provides a memory of previous localized input activity. Networks similar to this have been proposed as models of sustained activity in the head-direction system of the rat and in prefrontal cortex during tasks involving working memory.

This memory mechanism is related to the integration seen in the linear model of eye position maintenance discussed previously. The linear network has an eigenvector $\mathbf{e}_1$ with eigenvalue $\lambda_1 = 1$. This allows $\mathbf{v} = c_1\mathbf{e}_1$ to be a static solution of the equations of the network (7.17) in the absence of input for any value of c_1. As a result, the network can preserve any initial value of c_1 as a memory. In the case of figure 7.14, the steady-state activity in the absence of tuned input is a function of $\theta - \Theta$, for any value

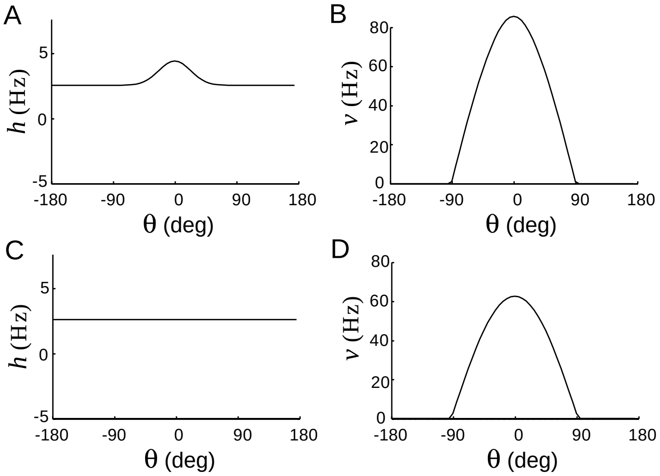

Figure 7.14: Sustained activity in a recurrent network. A) Input to the neurons of the network consisting of localized excitation and a constant background. B) The activity of the network neurons in response to the input of panel A. C) Constant network input. D) Response to the constant input of panel C when it immediately followed the input in A. The model is the same as that used in figures 7.9, 7.12, and 7.13.

of the angle Θ. As a result, the network can preserve any initial value of Θ as a memory ($\Theta = 0^\circ$ in the figure). The activities of the units $v(\theta)$ depend on Θ in an essentially nonlinear manner, but, if we consider linear perturbations around this nonlinear solution, there is an eigenvector with eigenvalue $\lambda_1 = 1$ associated with shifts in the value of Θ. In this case, it can be shown that $\lambda_1 = 1$ because the network was constructed to be translationally invariant.

Maximum Likelihood and Network Recoding

Recurrent networks can generate characteristic patterns of activity even when they receive complex inputs (figure 7.9) and can maintain these patterns while receiving constant input (figure 7.14). Pouget, Zhang, Deneve and Latham (1998) suggested that the location of the characteristic pattern (i.e. the value of Θ associated with the peak of the population activity profile) could be interpreted as a match of a fixed template curve to the input activity profile. This curve fitting operation is at the heart of the maximum likelihood decoding method we described in chapter 3 for estimating a stimulus variable such as Θ. In the maximum likelihood method, the fitting curve is determined by the tuning functions of the neurons, and the curve fitting procedure is defined by the characteristics of the noise perturbing the input activities. If the properties of the recurrent network match these optimal characteristics, the network can approximate maximum likelihood decoding. Once the activity of the population of neurons

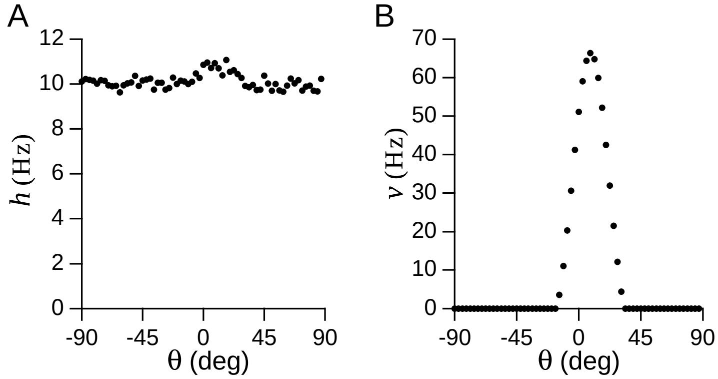

Figure 7.15: Recoding by a network model. A) The noisy initial inputs $h(\theta)$ to 64 network neurons are shown as dots. The standard deviation of the noise is 0.25 Hz. After a short settling time, the input is set to a constant value of $h(\theta) = 10$. B) The smooth activity profile that results from the recurrent interactions. The network model was similar to that used in figure 7.9 except that the recurrent synaptic weights were in the form of a Gabor-like function rather than a cosine, and the recurrent connections had short-range excitation and long-range inhibition. (see Pouget et al., 1998.)

has stabilized to its sterotyped shape, a simple decoding method such as vector decoding can be applied to extract the estimated value of Θ. This allows the accuracy of a vector decoding method to approach that of more complex optimal methods, because the computational work of curve fitting has been performed by the nonlinear recurrent interactions.

Figure 7.15 shows how this idea works in a network of 64 neurons receiving inputs that have Gaussian (rather than cosine) tuning curves as a function of Θ. Vector decoding applied to the reconstruction of Θ from the activity of the network or its inputs turns out to be almost unbiased. The way to judge decoding accuracy is therefore to compute the standard deviation of the decoded Θ values (chapter 3). The noisy input activity shown in figure 7.15A shows a slight bump around the value $\theta = 10^\circ$. Vector decoding applied to input activities with this level of noise gives a standard deviation in the decoded angle of 4.5°. Figure 7.15B shows the output of the network obtained by starting with initial activities $v(\theta) = 0$ and input $h(\theta)$ as in figure 7.15A, and then setting $h(\theta)$ to a constant (θ-independent) value to maintain sustained activity. This generates a smooth pattern of sustained population activity. Vector decoding applied to the output activities generated in this way gives a standard deviation in the decoded angle of 1.7°. This is not too far from the Cramér-Rao bound that gives the maximum possible accuracy for any unbiased decoding scheme applied to this system (see chapter 3), which is 0.88°.

Network Stability

When a network responds to a constant input by relaxing to a steady state with $d\mathbf{v}/dt = \mathbf{0}$, it is said to exhibit fixed-point behavior. Almost all the network activity we have discussed thus far involves such fixed points. This is by no means the only type of long-term activity that a network model can display. In a later section of this chapter, we discuss networks that oscillate, and chaotic behavior is also possible. But if certain conditions are met, a network will inevitably reach a fixed point in response to constant input. The theory of Lyapunov functions, to which we give an informal introduction, can be used to prove when this occurs.

fixed-point behavior

It is easier to discuss the Lyapunov function for a network if we use the firing-rate dynamics of equation 7.6 rather than equation 7.8. For a network model, this means expressing the vector of network firing rates as $\mathbf{v} = \mathbf{F}(\mathbf{I})$, where $\mathbf{I}$ is the total synaptic current vector, i.e. I_a represents the total synaptic current for unit a. $\mathbf{I}$ obeys the dynamic equation derived from generalizing equation 7.6 to a network situation,

recurrent model with current dynamics

$$\tau_s \frac{d\mathbf{I}}{dt} = -\mathbf{I} + \mathbf{h} + \mathbf{M} \cdot \mathbf{F}(\mathbf{I}) \, . \tag{7.39}$$

Note that we have made the substitution $\mathbf{v} = \mathbf{F}(\mathbf{I})$ in the last term of the right side of this equation. Equation 7.39 is sometimes used instead of equation 7.11 as the dynamical equation governing recurrent firing-rate model networks. For this form of firing-rate model with a symmetric recurrent weight matrix satisfying $M_{aa} = 0$ for all a, Cohen and Grossberg (1983) showed that the function

Lyapunov function L

$$L(\mathbf{I}) = \sum_{a=1}^{N_v} \left(\int_0^{I_a} dz_a \, z_a F'(z_a) - h_a F(I_a) - \frac{1}{2} \sum_{a'=1}^{N_v} F(I_a) M_{aa'} F(I_{a'}) \right) \tag{7.40}$$

has $dL/dt < 0$ whenever $d\mathbf{I}/dt \neq \mathbf{0}$. To see this, take the time derivative of equation 7.40 and use 7.39 to obtain

$$\frac{dL(\mathbf{I})}{dt} = -\frac{1}{\tau_s} \sum_{a=1}^{N_v} F'(I_a) \left(\frac{dI_a}{dt} \right)^2 \, . \tag{7.41}$$

Because $F' > 0$, L decreases unless $d\mathbf{I}/dt = \mathbf{0}$. If L is bounded from below, it cannot decrease indefinitely, so $\mathbf{I} = \mathbf{h} + \mathbf{M} \cdot \mathbf{v}$ must converge to a fixed point. This implies that $\mathbf{v}$ must converge to a fixed point as well.

We have required that $F'(I) > 0$ for all values of its argument I. However, with some technical complications, it can be shown that the Lyapunov function we have presented also applied to the case of the rectifying activation function $F(I) = [I]_+$, even though it is not differentiable at $I = 0$ and $F'(I) = 0$ for $I < 0$. Convergence to a fixed point, or one of a set of fixed points, requires the Lyapunov function to be bounded from below. One way to ensure this is to use a saturating activation function F, so that $F(I)$ is bounded as $I \to \infty$. Another way is to keep the eigenvalues of $\mathbf{M}$ sufficiently small.

Associative Memory

In an associative memory, a partial or approximate representation of a stored item is used to recall the full item. Unlike a standard random access memory, recall in an associative memory is based on content rather than on an address. For this reason, associative memory is also known as content-addressable memory. An example would be recalling every digit of a known phone number given a few of its digits as an initial clue. Associative memory networks have been suggested as models of various parts of the mammalian brain in which there is substantial recurrent feedback. These include area CA3 of the hippocampus and parts of the prefrontal cortex, structures which have long been implicated in various forms of memory. A number of network models exhibit associative memory, the best known being the so-called Hopfield networks (Hopfield, 1982 & 1984).

The models of memory we discussed previously in this chapter store information by means of persistent activity, with a particular item represented by the position of a stereotyped population activity profile. The idea underlying an associative (more strictly, auto-associative) memory is to extend persistent activity to a broader set of different population profiles, which are called memory patterns. Each of these is a fixed point of the dynamics of the network. The memory patterns are determined by and stored within the recurrent synaptic weights of the network, so memory retention does not require persistent activity. Rather, persistent activity is used to signal memory recall and to retain the identity of the most recently retrieved item.

During recall, an associative memory performs the computational operation of pattern matching, finding the memory pattern that most closely matches a distorted or partial activity pattern. This is achieved by initializing the network with an activity profile similar (but not identical) to one of the memory patterns, letting it relax to a fixed point, and treating the network activity at the fixed point as the best matching pattern. This is exactly the analog of the way that the recurrent model of maximum likelihood decoding executes a curve fitting procedure. Each memory pattern has a basin of attraction, defined as the set of initial states for which the network relaxes to that fixed point. The structure of these basins of attraction defines the matching properties of the network. The network dynamics is governed by a Lyapunov function of the form described above, and therefore the network will always relax to a fixed point. Provided that not too many memories are stored, the fixed points will closely resemble the stored memory patterns.

The associative network satisfies the dynamic equation 7.11, with the saturating activation function

$$F(I_s) = 150\text{ Hz}\left[\tanh\left(\frac{I_s - \gamma}{150\text{ Hz}}\right)\right]_+ \tag{7.42}$$

chosen to ensure that the Lyapunov function 7.40 is bounded from below. This is similar to a half-wave rectified activation function with threshold γ, except that it saturates at a firing rate of 150 Hz, which is outside the normal operating range of the units. We use a negative threshold, $\gamma = -20$ Hz, which corresponds to a constant source of excitation rather than a conventional threshold and generates background activity.

When this model is used for memory storage, a number of patterns, denoted by $\mathbf{v}^m$ with $m = 1, 2, \ldots, N_{\mathrm{mem}}$, are stored. Associative recall is achieved by starting the network in an initial state that is almost, but not exactly, proportional to one of the memory patterns, $\mathbf{v}(0) \approx c\mathbf{v}^m$ for some value of m and constant c. In this case, approximately proportional means that a significant number, but not all, of the elements of $\mathbf{v}(0)$ are close to the corresponding elements of $c\mathbf{v}^m$. The network then evolves according to equation 7.11 (with $\mathbf{h} = \mathbf{0}$). If the recall is successful, the dynamics converge to a fixed point proportional to the memory pattern associated with the initial state, that is $\mathbf{v}(t) \to c'\mathbf{v}^m$ for large t, where c' is another constant. Failure of recall occurs if the fixed point reached by the network is not proportional to the memory state $\mathbf{v}^m$.

In the example we consider, the components of the patterns to be stored are set to either 0 or 1. The assignment of these two values to the components of a given $\mathbf{v}^m$ is usually random with the probability of assigning a 1 equal to α and of assigning a 0 equal to $1 - \alpha$. However, in the example we show, two of the patterns have been assigned non-randomly to make them easier to detect in the figures. The parameter α is known as the sparseness of the memory patterns. The sparser the patterns, the more can be stored, but the less information each contains. We are interested in the limit of large N_v, in which case the maximum number of patterns that can be stored, N_{mem}, is proportional to N_v.

memory sparseness α

number of memories N_{mem}

The key to successful recall is in the choice of the matrix $\mathbf{M}$, which is given by

$$\mathbf{M} = \frac{1.25}{(1-\alpha)\alpha N_v} \sum_{m=1}^{N_{\mathrm{mem}}} (\mathbf{v}^m - \alpha\mathbf{n})(\mathbf{v}^m - \alpha\mathbf{n}) - \frac{1}{\alpha N_v}\mathbf{n}\mathbf{n}\,. \tag{7.43}$$

Here $\mathbf{n}$ is defined as a vector that has each of its N_v components equal to one. This form of coupling is called a covariance rule, because the first term on the right side is proportional to the covariance matrix of the collection of patterns. In chapter 8, we study synaptic plasticity rules that lead to this term. The second term introduces inhibition between the units.

vector of ones $\mathbf{n}$

covariance rule

Figure 7.16 shows an example of a network of $N_v = 50$ units exhibiting associative memory. This network stores 4 patterns with $\alpha = 0.25$. Recall of two of these patterns is shown in figure 7.16B and 7.16C. From an initial activity pattern only vaguely resembling one of the stored patterns, the network is able to attain a fixed activity pattern approximately proportional to the best matching memory pattern. Similar results would apply

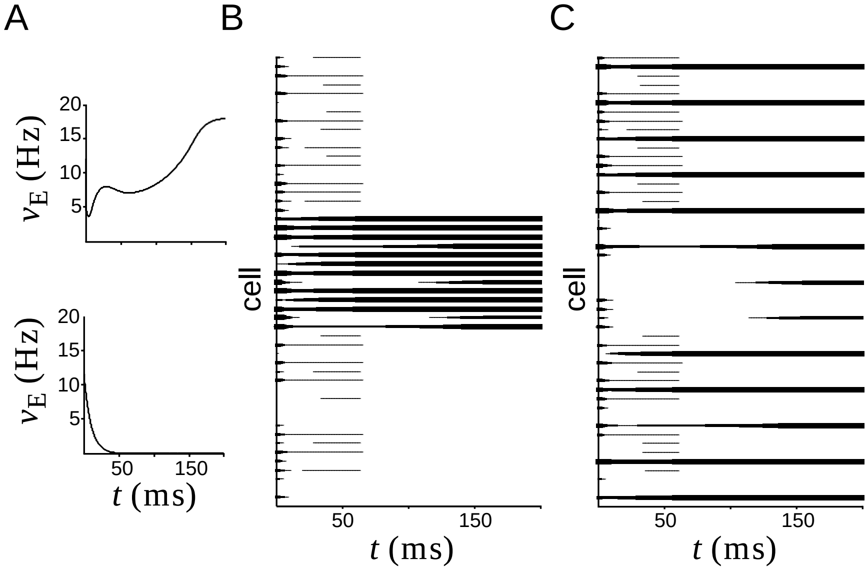

Figure 7.16: Associative recall of memory patterns in a network model. Panel A shows two representative model neurons, while panels B and C show the firing rates of all 50 cells plotted against time. The thickness of the horizontal lines in these plots is proportional to the firing rate of the corresponding neuron. A) Firing rates of representative neurons. The upper panel shows the firing rate of one of the excitatory neurons corresponding to a nonzero component of the recalled memory pattern. The firing rate achieves a nonzero steady-state value. The lower panel shows the firing rate of another excitatory neuron corresponding to a zero component of the recalled memory pattern. This firing rate goes to zero. B) Recall of one of the stored memory patterns. The stored pattern had nonzero values only for cells 18 through 31. The initial state of the network was random but with a bias toward this particular pattern. The final state is similar to the memory pattern. C) Recall of another of the stored memory patterns. The stored pattern had nonzero values only for every fourth cell. The initial state of the network was again random but biased toward this pattern. The final state is similar to the memory pattern.

for the other two memory patterns stored by the network, but it would be more difficult to see these patterns in the figure because they are random.

The rationale behind the weight matrix comes from considering the effect of the recurrent interactions if the activities match one of the memories, $\mathbf{v} = c'\mathbf{v}^1$ for example. A network activity pattern $\mathbf{v} = c'\mathbf{v}^1$ can only be a fixed point if

$$c'\mathbf{v}^1 = \mathbf{F}(c'\mathbf{M}\cdot\mathbf{v}^1)\,, \tag{7.44}$$

which ensures that the right side of equation 7.11 (with $\mathbf{h} = \mathbf{0}$) vanishes. We assume that αN_v components of $\mathbf{v}^1$ are equal to one and the remaining $(1-\alpha)N_v$ are zero. In this case,

$$\mathbf{M}\cdot\mathbf{v}^1 = 1.25\mathbf{v}^1 - (1+1.25\alpha)\mathbf{n} + \boldsymbol{\epsilon} \tag{7.45}$$

where

$$\epsilon = \frac{1.25}{(1-\alpha)\alpha N_v} \sum_{m=2}^{N_{\text{mem}}} (\mathbf{v}^m - \alpha\mathbf{n})(\mathbf{v}^m - \alpha\mathbf{n}) \cdot \mathbf{v}^1 \qquad (7.46)$$

is a term of order of magnitude $\sqrt{N_{\text{mem}}/N_v}$. To begin, suppose that ϵ is small enough to be ignored. Then, equation 7.44 amounts to two conditions, one arising from the nonzero components of $\mathbf{v}^1$ and the other from the zero components,

$$c' = F((0.25 - 1.25\alpha)c') \quad \text{and} \quad -(1 + 1.25\alpha)c' - \gamma < 0\,. \qquad (7.47)$$

The inequality follows from the requirement that the total synaptic current plus the threshold is less than zero so that $F(I_s) = 0$ for these components. On the other hand, the first equation requires that $(0.25 - 1.25\alpha)c' - \gamma > 0$ so that $F > 0$ for the nonzero components of $\mathbf{v}^1$. If ϵ can be ignored and these two conditions are satisfied, $\mathbf{v} = c'\mathbf{v}^1$ will be a fixed point of the network dynamics.

The term ϵ in equation 7.45, which we have been ignoring, is only negligible if $N_{\text{mem}} \ll N_v$. If $N_{\text{mem}} \approx N$, ϵ can become large enough to destabilize the memory states as fixed points. This limits the number of memories that can be stored in the network. Detailed analysis of the maximum value of N_{mem} is complicated by correlations among the terms that contribute to ϵ, but rigorous evaluations can be made of the capacity of the network, both for binary stored patterns (as here), and for real-valued patterns for which the activities of each element are drawn from a probability distribution. Different network architectures can also be considered, including ones with very sparse connectivity between units.

The basic conclusions from studies of associative memory models with threshold linear or saturating units is that large networks can store even larger numbers of patterns, particularly if the patterns are sparse (α is near 0) and if a few errors in recall can be tolerated. Nevertheless, the information stored per synapse is typically quite small. However, the simple covariance prescription for the weights in equation 7.43 is far from optimal. More sophisticated methods (such as the delta rule discussed in chapter 8) can achieve significantly higher storage densities.

7.5 Excitatory-Inhibitory Networks

In this section, we discuss models in which excitatory and inhibitory neurons are described separately by equations 7.12 and 7.13. These models exhibit richer dynamics than the single population models with symmetric coupling matrices we have analyzed up to this point. In models with excitatory and inhibitory sub-populations, the full synaptic weight matrix is not symmetric, and network oscillations can arise. We begin by analyzing a model of homogeneous coupled excitatory and inhibitory populations. We introduce methods for determining whether this model exhibits constant or oscillatory activity. We then present two network models in which oscillations appear. The first is a model of the olfactory bulb, and the second displays selective amplification in an oscillatory mode.

Homogeneous Excitatory and Inhibitory Populations

As an illustration of the dynamics of excitatory-inhibitory network models, we analyze a simple model in which all of the excitatory neurons are described by a single firing rate $v_{\rm E}$, and all of the inhibitory neurons are described by a second rate $v_{\rm I}$. Although we think of this example as a model of interacting neuronal populations, it is constructed as if it consists of just two neurons. Equations 7.12 and 7.13 with threshold linear response functions are used to describe the two firing rates, so that

$$\tau_{\rm E}\frac{dv_{\rm E}}{dt} = -v_{\rm E} + [M_{\rm EE}v_{\rm E} + M_{\rm EI}v_{\rm I} - \gamma_{\rm E}]_+ \qquad (7.48)$$

and

$$\tau_{\rm I}\frac{dv_{\rm I}}{dt} = -v_{\rm I} + [M_{\rm II}v_{\rm I} + M_{\rm IE}v_{\rm E} - \gamma_{\rm I}]_+ \,. \qquad (7.49)$$

The synaptic weights $M_{\rm EE}$, $M_{\rm IE}$, $M_{\rm EI}$, and $M_{\rm II}$ are numbers rather than matrices in this model. In the example we consider, we set $M_{\rm EE} = 1.25$, $M_{\rm IE} = 1$, $M_{\rm II} = 0$, $M_{\rm EI} = -1$, $\gamma_{\rm E} = -10$ Hz, $\gamma_{\rm I} = 10$ Hz, $\tau_{\rm E} = 10$ ms, and we vary the value of $\tau_{\rm I}$. The negative value of $\gamma_{\rm E}$ means that this parameter serves as a source of constant background activity rather than as a threshold.

Phase-Plane Methods and Stability Analysis

The model of interacting excitatory and inhibitory populations given by equations 7.48 and 7.49 provides an opportunity for us to illustrate some of the techniques used to study the dynamics of nonlinear systems. This model exhibits both static (constant $v_{\rm E}$ and $v_{\rm I}$) and oscillatory activity depending on the values of its parameters. Stability analysis can be used to determine the parameter values where transitions between these two types of activity take place.

The firing rates $v_{\rm E}(t)$ and $v_{\rm I}(t)$ arising from equations 7.48 and 7.49 can be displayed by plotting them as functions of time, as in figures 7.18A and 7.19A. Another useful way of depicting these results, illustrated in figures 7.18B and 7.19B, is to plot pairs of points $(v_{\rm E}(t), v_{\rm I}(t))$ for a range of t values. As the firing rates change, these points trace out a curve or trajectory in the $v_{\rm E}$-$v_{\rm I}$ plane, which is called the phase plane of the model.

phase plane

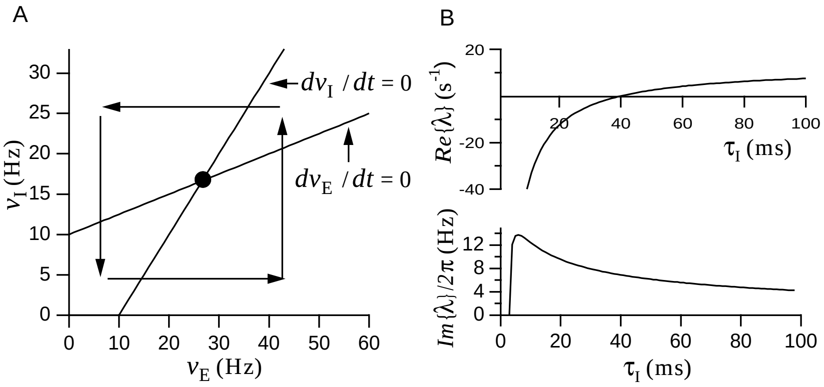

Figure 7.17: A) Nullclines, flow directions, and fixed point for the firing-rate model of interacting excitatory and inhibitory neurons. The two straight lines are the nullclines along which $dv_E/dt = 0$ or $dv_I/dt = 0$. The filled circle is the fixed point of the model. The horizontal and vertical arrows indicate the directions that v_E (horizontal arrows) and v_I (vertical arrows) flow in different regions of the phase plane relative to the nullclines. B) Real (upper panel) and imaginary (lower panel) parts of the eigenvalue determining the stability of the fixed point. To the left of the point where the imaginary part of the eigenvalue goes to zero, both eigenvalues are real. The imaginary part has been divided by 2π to give the frequency of oscillations near the fixed point.

Phase-plane plots can be used to give a geometric picture of the dynamics of a model.

Values of v_E and v_I for which the right sides of either equation 7.48 or equation 7.49 vanish are of particular interest in phase-plane analysis. Sets of such values form two curves in the phase plane known as nullclines. The nullclines for equations 7.48 and 7.49 are the straight lines drawn in figure 7.17A. The nullclines are important because they divide the phase plane into regions with opposite flow patterns. This is because dv_E/dt and dv_I/dt are positive on one side of their nullclines and negative on the other. Above the nullcline along which $dv_E/dt = 0$, $dv_E/dt < 0$, and below it $dv_E/dt > 0$. Similarly, $dv_I/dt > 0$ to the right of the nullcline where $dv_I/dt = 0$, and $dv_I/dt < 0$ to the left of it. This determines the direction of flow in the phase plane, as denoted by the horizontal and vertical arrows in figure 7.17A.

nullcline

At a fixed point of a dynamic system, the dynamic variables remain at constant values. In the model being considered, a fixed point occurs when the firing rates v_E and v_I take values that make $dv_E/dt = dv_I/dt = 0$. Because a fixed point requires both derivatives to vanish, it can only occur at an intersection of nullclines. The model we are considering has a single fixed point (at $v_E = 26.67$, $v_I = 16.67$) denoted by the filled circle in figure 7.17A. A fixed point provides a potential static configuration for the system, but it is critically important whether the fixed point is stable

fixed point

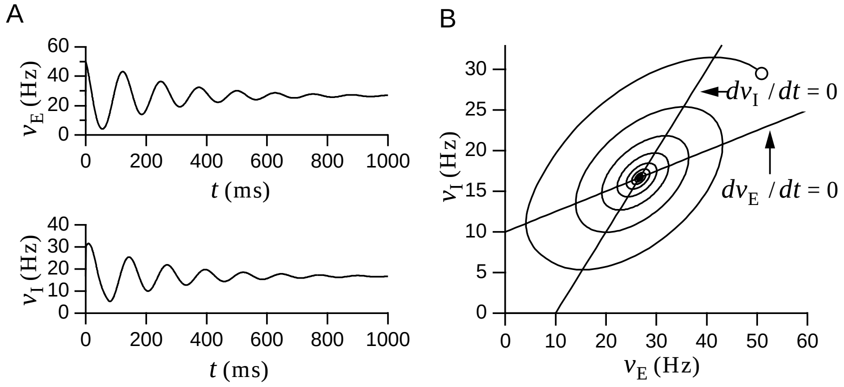

Figure 7.18: Activity of the excitatory-inhibitory firing-rate model when the fixed point is stable. A) The excitatory and inhibitory firing rates settle to the fixed point over time. B) The phase-plane trajectory is a counter-clockwise spiral collapsing to the fixed point. The open circle marks the initial values $v_E(0)$ and $v_I(0)$. For this example, $\tau_I = 30$ ms.

or unstable. If a fixed point is stable, initial values of v_E and v_I near the fixed point will be drawn toward it over time. If the fixed point is unstable, nearby configurations are pushed away from the fixed point, and the system will only remain at the fixed point indefinitely if the rates are set initially to the fixed-point values with infinite precision.

Linear stability analysis can be used to determine whether a fixed point is stable or unstable. This analysis starts by considering the first derivatives of the right sides of equations 7.48 and 7.49 with respect to v_E and v_I evaluated at the values of v_E and v_I that correspond to the fixed point. The four combinations of derivatives computed in this way can be arranged into a matrix

stability matrix

$$\begin{pmatrix} (M_{EE}-1)/\tau_E & M_{EI}/\tau_E \\ M_{IE}/\tau_I & (M_{II}-1)/\tau_I \end{pmatrix} . \tag{7.50}$$

As discussed in the Mathematical Appendix, the stability of the fixed point is determined by the real parts of the eigenvalues of this matrix. The eigenvalues are given by

$$\lambda = \frac{1}{2}\left(\frac{M_{EE}-1}{\tau_E} + \frac{M_{II}-1}{\tau_I} \pm \sqrt{\left(\frac{M_{EE}-1}{\tau_E} - \frac{M_{II}-1}{\tau_I}\right)^2 + \frac{4M_{EI}M_{IE}}{\tau_E\tau_I}}\right) . \tag{7.51}$$

If the real parts of both eigenvalues are less than zero the fixed point is stable, while if either is greater than zero the fixed point is unstable. If the factor inside the square root in equation 7.51 is positive, both eigenvalues are real, and the behavior near the fixed point is exponential. This means that there is exponential movement toward the fixed point if both eigenvalues are negative, or away from the fixed point if either eigenvalue is

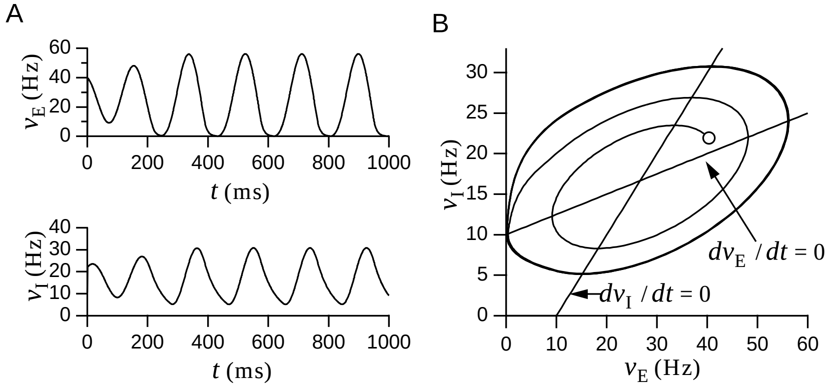

Figure 7.19: Activity of the excitatory-inhibitory firing-rate model when the fixed point is unstable. A) The excitatory and inhibitory firing rates settle into periodic oscillations. B) The phase-plane trajectory is a counter-clockwise spiral that joins the limit cycle, which is the closed orbit. The open circle marks the initial values $v_E(0)$ and $v_I(0)$. For this example, $\tau_I = 50$ ms.

positive. We focus on the case when the factor inside the square root is negative, so that the square root is imaginary and the eigenvalues form a complex conjugate pair. In this case, the behavior near the fixed point is oscillatory and the trajectory either spirals into the fixed point, if the real part of the eigenvalues is negative, or out from the fixed point if the real part of the eigenvalues is positive. The imaginary part of the eigenvalue determines the frequency of oscillations near the fixed point. The real and imaginary parts of one of these eigenvalues are plotted as a function of τ_I in figure 7.17B. This figure indicates that the fixed point is stable if $\tau_I < 40$ ms and unstable for larger values of τ_I.

Figures 7.18 and 7.19 show examples in which the fixed point is stable and unstable, respectively. In figure 7.18A, the oscillations in v_E and v_I are damped, and the firing rates settle down to the stable fixed point. The corresponding phase-plane trajectory is a collapsing spiral (figure 7.18B). In figure 7.19A the oscillations grow, and in figure 7.19B the trajectory is a spiral that expands outward until the system enters a limit cycle. A limit cycle is a closed orbit in the phase plane indicating periodic behavior. The fixed point is unstable in this case, but the limit cycle is stable. Without rectification, the phase-plane trajectory would spiral out from the unstable fixed point indefinitely. The rectification nonlinearity prevents the spiral trajectory from expanding past zero and thereby stabilizes the limit cycle.

limit cycle

There are a number of ways that a nonlinear system can make a transition from a stable fixed point to a limit cycle. Such transitions are called bifurcations. The transition seen between figures 7.18 and 7.19 is a Hopf bifurcation. In this case, a fixed point becomes unstable as a parameter is changed (in this case τ_I) when the real part of a complex eigenvalue changes sign. In a Hopf bifurcation, the limit cycle emerges at a finite fre-

Hopf bifurcation

quency, which is similar to the behavior of a type II neuron when it starts firing action potentials, as discussed in chapter 6. Other types of bifurcations produce type I behavior with oscillations emerging at zero frequency (chapter 6). One example of this is a saddle-node bifurcation, which occurs when parameters are changed such that two fixed points, one stable and one unstable, meet at the same point in the phase plane.

saddle-node bifurcation

The Olfactory Bulb

The olfactory bulb, and analogous olfactory areas in insects, provide examples where sensory processing involves oscillatory activity. The olfactory bulb represents the first stage of processing beyond the olfactory receptors in the vertebrate olfactory system. Olfactory receptor neurons respond to odor molecules and send their axons to the olfactory bulb. These axons terminate in glomeruli where they synapse onto mitral and tufted cells, and also local interneurons. The mitral and tufted cells provide the output of the olfactory bulb by sending projections to the primary olfactory cortex. They also synapse onto the larger population of inhibitory granule cells. The granule cells in turn inhibit the mitral and tufted cells.

mitral cells
tufted cells

granule cells

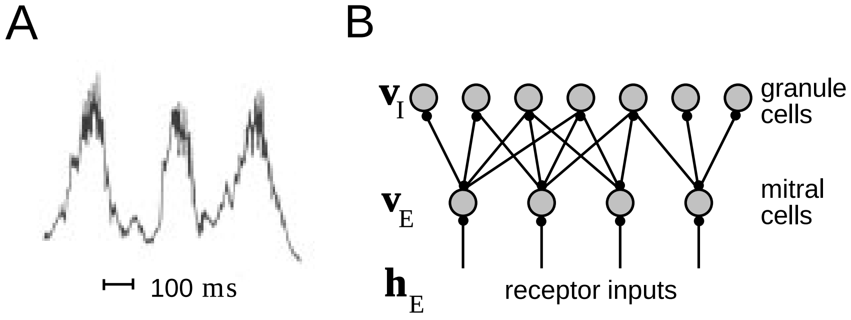

Figure 7.20: A) Extracellular field potential recorded in the olfactory bulb during respiratory waves representing three successive sniffs. B) Schematic diagram of the olfactory bulb model. (A adapted from Freeman and Schneider, 1982; B adapted from Li, 1995.)

The activity in the olfactory bulb of many vertebrates is strongly influenced by a sniff cycle in which a few quick sniffs bring odors past the olfactory receptors. Figure 7.20A shows an extracellular potential recorded during three successive sniffs. The three large oscillations in the figure are due to the sniffs. The oscillations we discuss in this section are the smaller, higher frequency oscillations seen around the peak of each sniff cycle. These arise from oscillatory neural activity. Individual mitral cells have quite low firing rates, and do not fire on each cycle of the oscillations. The oscillations are phase-locked across the bulb, but different odors induce oscillations of different amplitudes and phases.

Li and Hopfield (1989) modeled the mitral and granule cells of the olfactory bulb as a nonlinear input-driven network oscillator. Figure 7.20B shows the architecture of the model, which uses equations 7.12 and 7.13 with $\mathbf{M}_{\text{EE}} = \mathbf{M}_{\text{II}} = 0$. The absence of these couplings in the model is in accord with the anatomy of the bulb. The rates $\mathbf{v}_{\text{E}}$ and $\mathbf{v}_{\text{I}}$ refer to the mitral and granule cells, respectively (figure 7.20B). Figure 7.21A shows the activation functions of the model. The time constants for the two populations of cells are the same, $\tau_{\text{E}} = \tau_{\text{I}} = 6.7$ ms. $\mathbf{h}_{\text{E}}$ is the input from the receptors to the mitral cells, and $\mathbf{h}_{\text{I}}$ is a constant representing top-down input that exists from the olfactory cortex to the granule cells.

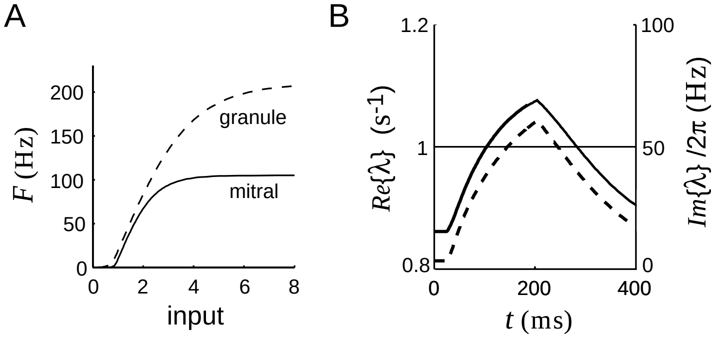

Figure 7.21: Activation functions and eigenvalues for the olfactory bulb model. A) The activation functions F_{E} (solid curve) for the mitral cells, and F_{I} (dashed curve) for the granule cells. B) The real (solid line, left axis) and imaginary (dashed line, right axis) parts of the eigenvalue that determines whether the network model exhibits fixed-point or oscillatory behavior. These are plotted as a function of time during a sniff cycle. When the real part of the eigenvalue becomes greater than one, it determines the growth rate away from the fixed point and the imaginary part divided by 2π determines the initial frequency of the resulting oscillations. (Adapted from Li, 1995.)

The field potential in figure 7.20A shows oscillations during each sniff, but not between sniffs. For the model to match this pattern of activity, the input from the olfactory receptors, $\mathbf{h}_{\text{E}}$, must induce a transition between fixed-point and oscillatory activity. Before a sniff, the network must have a stable fixed point with low activities. As $\mathbf{h}_{\text{E}}$ increases during a sniff, this steady-state configuration must become unstable leading to oscillatory activity. The analysis of the stability of the fixed point and the onset of oscillations is closely related to our previous stability analysis of the model of homogeneous populations of coupled excitatory and inhibitory neurons. It is based on properties of the eigenvalues of the linear stability matrix (see the Mathematical Appendix). In this case, the stability matrix includes contributions from the derivatives of the activation functions evaluated at the fixed point. For the fixed point to become unstable, the real part of at least one of the eigenvalues that arise in this analysis must become larger than 1. To ensure oscillations, at least one of these destabilizing eigenval-

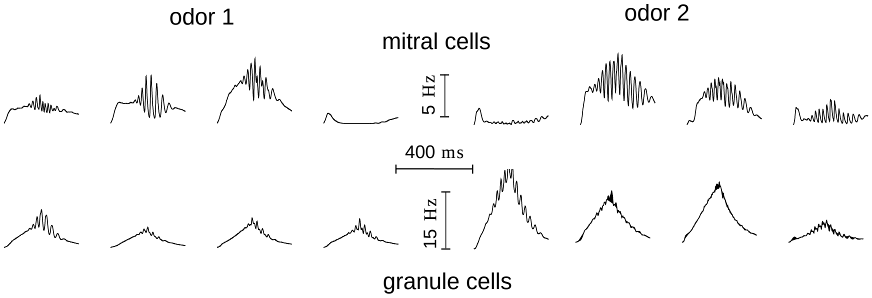

Figure 7.22: Activities of four of ten mitral (upper) and granule (lower) cells during a single sniff cycle for two different odors. (Adapted from Li and Hopfield, 1989.)

ues should have a non-zero imaginary part. These requirements impose constraints on the connections between the mitral and granule cells and on the inputs.

Figure 7.21B shows the real and imaginary parts of the relevant eigenvalue, labeled λ, during one sniff cycle. About 100 ms into the cycle the real part of λ gets bigger than 1. Reading off the imaginary part of λ at this point, we find that this sets off roughly 40 Hz oscillations in the network. These oscillations stop about 300 ms into the sniff cycle when the real part of λ drops below 1. The input $\mathbf{h}_{\mathrm{E}}$ from the receptors plays two critical roles in this process. First, it makes the eigenvalue great than 1 by modifying where the fixed point lies on the activation function curves in figure 7.21A. Second, it affects which particular neurons are destabilized and thus, which begin to oscillate. The ultimate pattern of oscillatory activity is determined both by the input $\mathbf{h}_{\mathrm{E}}$ and by the recurrent couplings of the network.

Figure 7.22 shows the behavior of the network during a single sniff cycle in the presence of two different odors, represented by two different values of $\mathbf{h}_{\mathrm{E}}$. The top rows show the activity of four mitral cells, and the bottom rows four granule cells. The amplitudes and phases of the oscillations seen in these traces, along with the identities of the mitral cells taking part in them, provide a signature of the identity of the odor that was presented.

Oscillatory Amplification

As a final example of network oscillations, we return to amplification of input signals by a recurrently connected network. Two factors control the amount of selective amplification that is viable in networks such as that shown in figure 7.9. The most important constraint on the recurrent weights is that the network must be stable, so the activity does not increase without bound. Another possible constraint is suggested by figure 7.14D where the output shows a tuned response even though the input to the network is constant as a function of θ. Tuned output in the absence of tuned input can serve as a memory mechanism, but it would produce persistent perceptions if it occurs in a primary sensory area, for example. Avoiding this in the network limits the recurrent weights and the amount of amplification that can be supported.

Li and Dayan (1999) showed that this restriction can be significantly eased using the richer dynamics of networks of coupled inhibitory and excitatory neurons. Figure 7.23 shows an example with continuous neuron labeling based on a continuous version of equations 7.12 and 7.13. The input is either $h_{\mathrm{E}}(\theta) = 8(1 + \frac{5}{8}\cos(2\theta))$ in the modulated case (figure 7.23B) or $h_{\mathrm{E}}(\theta) = 8$ in the unmodulated case (figure 7.23C). Noise with standard deviation 0.4 corrupts this input. The input to the network is constant in time.

The network oscillates in response to either constant or tuned input. Figure 7.23A shows the time average of the oscillating activities of the neurons in the network as a function of their preferred angles for noisy tuned (solid curve) and untuned (dashed curve) inputs. Neurons respond to the tuned input in a highly tuned and amplified manner. Despite the high degree of amplication, the average response of the neurons to untuned input is almost independent of θ. Figures 7.23B and 7.23C show the activities of individual neurons with $\theta = 0^\circ$ ('o') and $\theta = -37^\circ$) ('x') over time for the tuned and untuned inputs respectively. The network does not produce persistent perception, because the output to an untuned input is itself untuned. In contrast, a non-oscillatory version of this network, with $\tau_{\mathrm{I}} = 0$, exhibits tuned sustained activity in response to an untuned intput for recurrent weights this strong. The oscillatory network can thus operate in a regime of high selective amplification without generating spurious tuned activity.

7.6 Stochastic Networks

Up to this point, we have considered models in which the output of a cell is a deterministic function of its input. In this section, we consider a network model called the Boltzmann machine in which the input-output relationship is stochastic. Boltzmann machines are interesting from the perspective of learning, and also because they offer an alternative interpretation of the dynamics of network models.

Boltzmann machine

In the simplest form of Boltzmann machine, the neurons are treated as binary, so $v_a(t) = 1$ if unit a is active at time t (e.g. it fires a spike between times t and $t + \Delta t$ for some small value of Δt), and $v_a(t) = 0$ if it is inactive. The state of unit a is determined by its total input current,

$$I_a(t) = h_a(t) + \sum_{a'=1}^{N_v} M_{aa'} v_{a'}(t) \, , \tag{7.52}$$

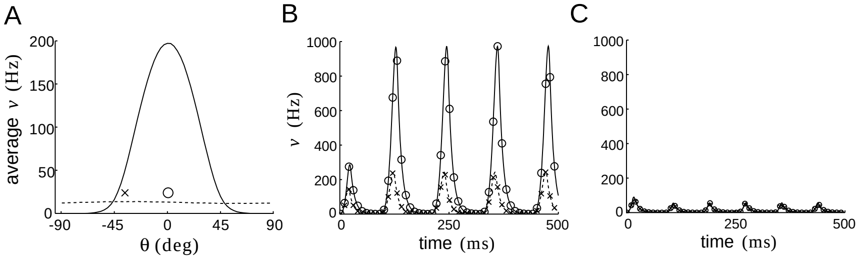

Figure 7.23: Selective amplification in an excitatory-inhibitory network. A) Time-averaged response of the network to a tuned input with $\Theta = 0^\circ$ (solid curve) and to an untuned input (dashed curve). Symbols 'o' and 'x' mark the 0° and -37° points seen in B and C. B) Activities over time of neurons with preferred angles of $\theta = 0^\circ$ (solid curve) and $\theta = -37^\circ$ (dashed curve) in response to a modulated input with $\Theta = 0^\circ$. C) Activities of the same units shown in B to a constant input. The lines lie on top of each other showing that the two units respond identically. The parameters are $\tau_E = \tau_I = 10$ ms, $h_I = 0$, $M_{EI} = -\delta(\theta - \theta')/\rho_\theta$, $M_{EE} = (1/\pi\rho_\theta)[5.9 + 7.8\cos(2(\theta - \theta'))]_+$, $M_{IE} = 13.3/\pi\rho_\theta$, and $M_{II} = 0$. (After Li and Dayan, 1999.)

where $M_{aa'} = M_{a'a}$ and $M_{aa} = 0$ for all a and a' values, and h_a is the total feedforward input into unit a. In the model, units can only change state at integral multiples of Δt. At each time step, a single unit is selected, usually at random, to be updated. This update is based on a probabilistic rather than a deterministic rule. If unit a is selected, its state at the next time step is set stochastically to 1 with probability

$$P[v_a(t + \Delta t) = 1] = F(I_a(t)) \quad \text{with} \quad F(I_a) = \frac{1}{1 + \exp(-I_a)} \, . \tag{7.53}$$

Of course, it follows that $P[v_a(t + \Delta t) = 0] = 1 - F(I_a(t))$. F is a sigmoidal function, which has the property that the larger the value of I_a, the more likely unit a is to take the value one.

Markov chain

Glauber dynamics

Under equation 7.53, the state of activity of the network evolves as a Markov chain. This means that the components of $\mathbf{v}$ at different times are sequences of random variables with the property that $\mathbf{v}(t+1)$ depends only on $\mathbf{v}(t)$, and not on the previous history of the network. The update of equation 7.53 is known as Glauber dynamics.

energy function

An advantage of using Glauber dyanmics to define the evolution of a network model is that general results from statistical mechanics can be used to determine the equilibrium distribution of activities. Under Glauber dynamics, $\mathbf{v}$ does not converge to a fixed point, but can be described by a probability distribution associated with an energy function

$$E(\mathbf{v}) = -\mathbf{h} \cdot \mathbf{v} - \frac{1}{2}\mathbf{v} \cdot \mathbf{M} \cdot \mathbf{v} \, . \tag{7.54}$$

The probability distribution characterizing $\mathbf{v}$, once the network has converged to an equilibrium state, is

$$P[\mathbf{v}] = \frac{\exp(-E(\mathbf{v}))}{Z} \quad \text{where} \quad Z = \sum_{\mathbf{v}} \exp(-E(\mathbf{v})) \, . \tag{7.55}$$

The notion of convergence as $t \to \infty$ can be formalized precisely, but informally, it means that after repeated updating according to equation 7.53, the states of the network are described statistically by equation 7.55. Z is called the partition function and $P[\mathbf{v}]$ the Boltzmann distribution. Under the Boltzmann distribution, states with lower energies are more likely. In this case, Glauber dynamics implements a statistical operation called Gibbs sampling for the distribution given in equation 7.55.

partition function

Boltzmann distribution

Gibbs sampling

The Boltzmann machine is an inherently stochastic device. An approximation to the Boltzmann machine, known as the mean-field approximation, can be constructed on the basis of the deterministic synaptic current dynamics of a firing-rate model. In this case, $\mathbf{I}$ is determined by the dynamic equation 7.39 rather than by equation 7.52, and the model runs in continuous rather than discrete time. The function F in equation 7.39 is taken to be the same sigmoidal function as in equation 7.53. Although the mean-field formulation of the Boltzmann machine is inherently deterministic, $F(I_a)$ can be used to generate a probability distribution over a binary output vector $\mathbf{v}$. This is done by treating the output of each unit a, v_a, as an independent binary variable set to either 1 or 0 with probability $F(I_a)$ or $1 - F(I_a)$ respectively. This replaces the deterministic rule $v_a = F(I_a)$ used in the firing-rate version of the model. Because $v_a = 1$ has probability $F(I_a)$ and $v_a = 0$ probability $1 - F(I_a)$ and the units are independent, the probability distribution for the entire vector $\mathbf{v}$ is

mean-field approximation

$$Q[\mathbf{v}] = \prod_{a=1}^{N_v} F(I_a)^{v_a} (1 - F(I_a))^{1-v_a} \, . \tag{7.56}$$

This is called the mean-field distribution for the Boltzmann machine. Note that this distribution (and indeed $\mathbf{v}$ itself) plays no role in the dynamics of the mean-field formulation of the Boltzmann machine. It is rather a way of interpreting the outputs.

mean field distribution

We have presented two formulations of the Boltzmann machine, Gibbs sampling and the mean-field approach, that lead to the two distributions $P[\mathbf{v}]$ and $Q[\mathbf{v}]$ (equations 7.55 and 7.56). The Lyapunov function of equation 7.40, that decreases steadily under the dynamics of equation 7.39 until a fixed point is reached, provides a key insight into the relationship between these two distributions. In the appendix, we show that this Lyapunov function can be expressed as

$$L(\mathbf{I}) = D_{\mathrm{KL}}(Q, P) + K \tag{7.57}$$

where K is a constant, and D_{KL} is the Kullback-Liebler divergence defined in chapter 4. $D_{\mathrm{KL}}(Q, P)$ is a measure of how different the two distributions Q and P are from each other. The fact that the dynamics of equation 7.39

reduces the Lyapunov function to a minimum value means that it also reduces the difference between Q and P, as measured by the Kullback-Liebler divergence. This offers an interesting interpretation of the mean-field dynamics; it modifies the current value of the vector $\mathbf{I}$ until the distribution of binary output values generated by the mean-field formulation of the Boltzmann machine matches as closely as possible (to at least a local minimum of $D_{\mathrm{KL}}(Q, P)$) the distribution generated by Gibbs sampling. In this way, the mean-field procedure can be viewed as an approximation of Gibbs sampling.

The power of the Boltzmann machine lies in the relationship between the distribution of output values, equation 7.55, and the quadratic energy function of equation 7.54. This makes it is possible to determine how changing the weights $\mathbf{M}$ affects the distribution of output states. In chapter 8, we present a learning rule for the weights of the Boltzmann machine that allows $P[\mathbf{v}]$ to approximate a probability distribution extracted from a set of inputs. In chapter 10, we study other models that construct output distributions in this way.

Note that the mean field distribution $Q[\mathbf{v}]$ is simpler than the full Boltzmann distribution $P[\mathbf{v}]$ because the units are statistically independent. This prevents $Q[\mathbf{v}]$ from providing a good approximation in some cases, particularly if there are negative weights between units, which tend to make their activities mutually exclusive. Correlations such as these in the fluctuations of the states about their mean values can be important for learning. The mean-field analysis of the Boltzmann machine illustrates the limitations of rate-based descriptions in capturing the full extent of the correlations that can exist between spiking neurons.

7.7 Chapter Summary

The models in this chapter mark the start of our discussion of computation, as opposed to coding. Using a description of the firing rates of network neurons, we showed how to construct linear and nonlinear feedforward and recurrent networks that transform information from one coordinate system to another, selectively amplify input signals, integrate inputs over extended periods of time, select between competing inputs, sustain activity in the absence of input, exhibit gain modulation, allow simple decoding with performance near the Cramér-Rao bound, and act as content addressable memories. We used network responses to a continuous stimulus variable as an extended example. This led to models of simple and complex cells in primary visual cortex. We described a model of the olfactory bulb as an example of a system for which computation involves oscillations arising from asymmetric couplings between excitatory and inhibitory neurons. Linear stability analysis was applied to a simplified version of this model. We also considered a stochastic network model called the Boltzmann machine.

Appendix

Lyapunov Function for the Boltzmann Machine

Here, we show that the Lyapunov function of equation 7.40 can be reduced to equation 7.57 when applied to the mean-field version of the Boltzmann machine. Recall, from equation 7.40, that

$$L(\mathbf{I}) = \sum_{a=1}^{N_v} \left(\int_0^{I_a} dz_a \, z_a F'(z_a) - h_a F(I_a) - \frac{1}{2} \sum_{a'=1}^{N_v} F(I_a) M_{aa'} F(I_{a'}) \right) \tag{7.58}$$

When F is given by the sigmoidal function of equation 7.53,

$$\int_0^{I_a} dz_a \, z_a F'(z_a) = F(I_a) \ln F(I_a) + (1 - F(I_a)) \ln(1 - F(I_a)) + k \tag{7.59}$$

where k is a constant, as can be verified by differentiating the right side. The non-constant part of the right side of this equation is just the entropy associated with the binary variable v_a. In fact,

$$\sum_{a=1}^{N_v} \int_0^{I_a} dz_a \, z_a F'(z_a) = \langle \ln Q[\mathbf{v}] \rangle_Q + N_v k \tag{7.60}$$

where the average is over all values of $\mathbf{v}$ with probabilities $Q[\mathbf{v}]$.

To evaluate the remaining terms in equation 7.58, we note that, because the components of $\mathbf{v}$ are binary and independent, relations such as $\langle v_a \rangle_Q = F(I_a)$ and $\langle v_a v_b \rangle_Q = F(I_a) F(I_b)$ are valid. Then, using equation 7.54, we find

$$L(\mathbf{I}) = \sum_{a=1}^{N_v} \left(-h_a F(I_a) - \frac{1}{2} \sum_{a'=1}^{N_v} F(I_a) M_{aa'} F(I_{a'}) \right) = \langle -E(\mathbf{v}) \rangle_Q \, . \tag{7.61}$$

Similarly, from equation 7.55, we can show that

$$\langle \ln P[\mathbf{v}] \rangle_Q = \langle -E(\mathbf{v}) \rangle_Q - \ln Z \, . \tag{7.62}$$

Combining the results of equations 7.60, 7.61, and 7.61, we obtain

$$L(\mathbf{I}) = \langle \ln Q[\mathbf{v}] - \ln P[\mathbf{v}] \rangle_Q + N_v k - \ln Z \, . \tag{7.63}$$

which gives equation 7.57 with $K = N_v k - \log Z$ because $\langle \ln Q[\mathbf{v}] - \ln P[\mathbf{v}] \rangle_Q$ is, by definition, the Kullback-Liebler divergence $D_{\mathrm{KL}}(Q, P)$ (see chapter 4, although there we use base 2 logarithms, while here we use base e logarithms in the definition of D_{KL}, but the difference is only an overall multiplicative constant).

7.8 Annotated Bibliography

Wilson & Cowan (1972, 1973) provide pioneering analyses of firing-rate models. Subsequent analyses related to the discussion in this chapter are presented in **Abbott (1994)**, **Ermentrout (1998)**, Amit & Tsodyks (1991a & b) and Bressloff & Coombes (2000). Rinzel and Ermentrout (1998) discuss phase-plane methods; `XPP` (see `http://www.pitt.edu/~phase`) provides a computer environment for performing phase-plane and other forms of mathematical analysis on neuron and network models.

Our discussion of the feedforward coordinate transformation model followed Pouget & Sejnowski (1995, 1997) and Salinas & Abbott (1995), which built on theoretical work by Zipser & Andersen (1988) to explain parietal gain fields (see Andersen, 1989).

We followed Seung's (1996) discussion of neural integration for eye position, which builds on Robinson (1989).

The notion of a regular repeating unit of cortical computation dates back to the earliest investigations of cortex (see Douglas & Martin 1998). We followed Seung (1996); Zhang (1996) in adopting the theoretical context of continuous line or surface attractors, that has the many applications discussed in the chapter (see also Hahnloser et al., 2000). **Sompolinsky & Shapley 1997** review a recently active debate about the balance of control of orientation selectivity in primary visual cortex between feedforward input and a recurrent line attractor. We presented a model of a hypercolumn; the extension to multiple hypercolumns is used to link psychophysical and physiological data on contour integration and texture segmentation by Li (1998, 1999).

Network associative memories are described and analyzed by Hopfield (1982; 1984) and Cohen & Grossberg (1983), who described a general Lyapunov function. Grossberg (1988); **Amit (1989)**; **Hertz, et al. (1991)** present a host of theory about associative networks, in particular about their capacity to store information. Associative memory in non-binary recurrent networks has been studied in particular by Treves and collaborators (see Rolls & Treves, 1998) and, in the context of line attractor networks, by Samsonovich & McNaughton (1997) and Battaglia & Treves (1998).

We followed Li's (1995) presentation of Li & Hopfield's (1989) oscillatory model of the olfactory bulb.

The Boltzmann machine was invented by Hinton & Sejnowski (1986), and is a stochastic generalization of the Hopfield net (Hopfield, 1982). The mean-field model is due to Hopfield (1984), and we followed the probabilistic discussion in Jordan et al. (1998).