TN guide

ted moskovitz

May 2020

Contents

Overview

I wrote this as a way of preparing for the final exam for Gatsby's Theoretical Neuroscience course. It's in no way an original work, just a long study guide. I used a few sources throughout: my own lecture notes, Maneesh Sahani's and Peter Latham's notes[1][2], Jorge Menendez's course notes from a few years ago [3], Larry Abbott and Peter Dayan's theoretical neuroscience book [4], and the online Gerstner et al. book on neural dynamics [5]. In some places, I directly quote (sometimes without acknowledgement) from one of these sources, simply because I couldn't think of a clearer way to explain things. To my knowledge, every source I used, however, is listed in the references section. Most figures not obviously plotted in matplotlib can be presumed to be from one of these sources. There are plenty of topics that should be explored in greater depth or precision, and I certainly plan on updating this guide from time to time. One major topic I didn't cover was RNNs, so at some point I may add it. If you spot any inaccuracies, please let me know. Hopefully, this proves at least moderately interesting or useful for you.

1 Biophysics

1.1 Overview and Basics

Setting the Scene Most of the time, there is an excess of negative charge in the interior of a neuron, which, because negative ions repel each other, builds up on the inside surface of the membrane. This in turn causes positive ions to accumulate on the outside surface of the membrane, which acts like a capacitor. The lipid-bilayer membrane generally has pretty high resistance, and would be essentially impermeable, except for the fact that it contains passive and active channels to facilitate movement of ions across it. The effective resistance of the membrane depends on the type and density of these ion channels, most of which are highly selective, only permitting a single type of ion to pass through them.

By convention, we define the extracellular fluid around the neuron to have a potential of 0. Under normal conditions, the internal membrane potential can vary from -90 to $+50$ mV, depending on the opening and closing of ion channels.

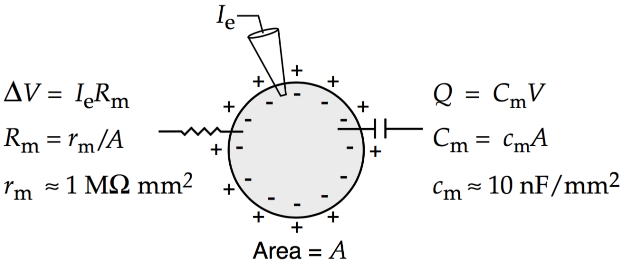

I_e
$\Delta V = I_e R_m$
$R_m = r_m/A$
$r_m \approx 1\ \text{M}\Omega\ \text{mm}^2$
$Q = C_m V$
$C_m = c_m A$
$c_m \approx 10\ \text{nF/mm}^2$
Area = A

Figure 1.1: Basic set-up of a single-compartment neuron model.

Membrane Capacitance and Resistance Intracellular resistance to current flow can cause significant differences in membrane potential in a neuron (especially those with long dendrites and/or axons), but for more compact neurons, we can approximate and say that the whole thing has a single membrane potential. This is called *electrotonic compactness.*

As mentioned above, an excess of negative charge Q typically builds up on the interior surface of the membrane, which can be computed via

$$Q = C_m V, \tag{1.1}$$

where C_m is the membrane capacitance and V is the voltage across the membrane. The membrane capacitance is proportional to the total area of the membrane A, so we can denote the *specific capacitance* by c_m, with

$$C_m = c_m A. \tag{1.2}$$

Similarly, the total membrane resistance R_m is inversely proportional to the area, so we have the *specific resistance* as

$$R_m = \frac{r_m}{A}. \tag{1.3}$$

Differentiating eq. 1.1 with respect to time gives the current required to change the membrane potential at a given rate:

$$C_m \frac{dV}{dt} = \frac{dQ}{dt} = I. \tag{1.4}$$

In other words, the rate of change of the membrane potential is proportional to the rate at which charge builds up inside the cell. Holding the membrane potential steady at a different voltage from its resting value also requires current, the amount of which is determined by Ohm's law:

$$\Delta V = I_e R_m, \tag{1.5}$$

where R_m is assumed to be constant over a range of ΔV. These relationships, along with example numbers, are summarized in Figure 1.1. The rate of change of the membrane potential is also governed by the *membrane time constant* τ_m, which is invariant to the surface area of the neuron:

$$\tau_m = R_m C_m = \left(\frac{r_m}{A}\right)(c_m A) = r_m c_m. \tag{1.6}$$

The value of the membrane time constant typically falls in the range of 10 and 100 ms.

Equilibrium and Reversal Potentials The voltage difference between the interior and exterior of the cell results in electrical forces that facilitate a diffusion of ions across the membrane. Any model that describes the membrane potential of a neuron by a single quantity V is called a *single-compartment model*. When the membrane potential is negative, this drives positive ions into the cell and drives out negative ions. Specific ions also diffuse through designated channels based on concentration gradients. The concentrations of Cl^-, Na^+, and Ca^{2+} are higher on the outside of the cell, so diffusion drives them into the neuron. Conversely, the concentration of K^+ is naturally higher inside the cell, so diffusion drives it out. We define the *equilibrium potential* as the membrane potential at which flow of ions due to electrical forces is exactly canceled by the diffusion of ions due to concentration gradients. For channels that only admit one type of ions, this value is determined by the Nernst equation (see Dayan & Abbott p. 159). The equilibrium potential for a K^+ channel is typically between -90 and -70 mV, for Na^+ it's around $+50$ mV, Ca^{2+} is around $+150$ mV, and for $\text{Cl}^{\text{-}}$ it's usually about -65 to -60 mV.

When a channel admits more than one type of ion, the equilibrium potential is usually a weighted averaged of the selected ions and is known as the *reversal potential*, denoted by $\mathcal{E}$. It's called the reversal potential because the flow of current through the channel switches direction when the membrane potential passes through $\mathcal{E}$. When $V > \mathcal{E}$, positive current flows out, bringing V back to $\mathcal{E}$, and when $V < \mathcal{E}$, there is an inflow of positive current. Therefore, because Na^+ and Ca^{2+} channels have positive reversal potentials, they tend to *depolarize* a neuron – make the membrane potential more positive (as the potential is drawn toward $\mathcal{E}$). Similarly, K^+ channels tend to *hyperpolarize* a neuron – push the membrane potential to be more negative – due to their negative reversal potentials. The reversal potential of $\text{Cl}^{\text{-}}$ channels is around equilibrium for many neurons, so they doesn't really affect current flow, they just change the effective resistance of the cell – this is called *shunting*. Synapses with reversal potentials below the threshold needed for action potential generation are typically called *inhibitory*, while those with reversal potentials above the action potential threshold are typically known as *excitatory*.

It's also useful to consider what happens when, for example, the concentration of ions shifts either intracellularly or extracellularly. For instance, if the extracellular concentration of a negative ion such as $\text{Cl}^{\text{-}}$ increases, the electric force driving it out must increase proportionally to compensate to cancel the increased inward diffusion. Thefore, the reversal/equilibrium potential must *decrease*—becoming more negative will repel the negative ions more strongly. Analogous reasoning can be used in similar cases.

1.2 Membrane Current and Passive Channels

The membrane current is the total current flowing through the ion channels of the membrane. By convention, it's defined to be positive when positive ions are leaving the cell, and negative when positive ions enter the cell. The total membrane current I_m is given by the product of the surface area A and the membrane current per unit area i_m:

$$i_m = \frac{I_m}{A}. \tag{1.7}$$

For many types of channels, the membrane current is approximately proportional to the difference between the current voltage V and the membrane potential. Ohm's law gives us

$$i_m = \sum_x \frac{1}{r_x} \Delta V_x = \sum_k g_x (V - \mathcal{E}_x), \tag{1.8}$$

where $g_x = 1/r_x$ is the channel conductance, and x is an index over channel types. In this section, we assume that the conductances g_x are constant, and thus the current flow is limited to *leakage* current, which includes the currents carried by ion pumps that are involved in maintaining concentration gradients at equilibrium. The ions that we'll consider to be most involved in this process are Na^+, K^+, and $\text{Cl}^{\text{-}}$. We can expand out eq. 1.8 as

$$\begin{aligned} i_m &= \sum_x \frac{1}{r_x} \Delta V_x = \sum_k g_x (V - \mathcal{E}_x) & (1.9) \\ &= g_{\text{Na}^+}(V - \mathcal{E}_{\text{Na}^+}) + g_{\text{K}^+}(V - \mathcal{E}_{\text{K}^+}) + g_{\text{Cl}^{\text{-}}}(V - \mathcal{E}_{\text{Cl}^{\text{-}}}) & (1.10) \\ &= (g_{\text{Na}^+} + g_{\text{K}^+} + g_{\text{Cl}^{\text{-}}}) \left(V - \frac{g_{\text{Na}^+}\mathcal{E}_{\text{Na}^+} + g_{\text{K}^+}\mathcal{E}_{\text{K}^+} + g_{\text{Cl}^{\text{-}}}\mathcal{E}_{\text{Cl}^{\text{-}}}}{g_{\text{Na}^+} + g_{\text{K}^+} + g_{\text{Cl}^{\text{-}}}} \right) & (1.11) \\ &= g_\ell (V - \mathcal{E}_\ell), & (1.12) \end{aligned}$$

where $g_\ell \coloneqq g_{\text{Na}^+} + g_{\text{K}^+} + g_{\text{Cl}^{\text{-}}}$ is the leakage conductance. By convention, external current I_e entering the cell is considered positive, while membrane current leaving the cell is considered negative. From eq. 1.4, we can then

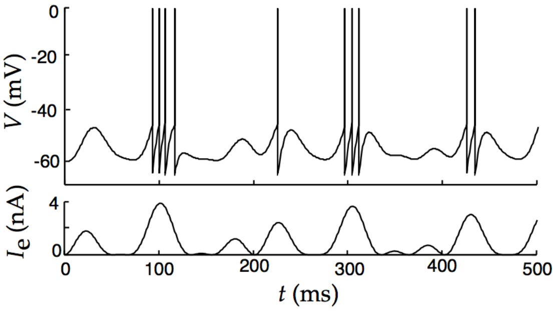


Figure 1.2: Leaky integrate-and-fire model with a time-varying input current.

write the dynamics of a passive channel as

$$\begin{aligned} c_m \frac{dV}{dt} &= -i_m + \frac{I_e}{A} \\ &= -g_\ell (V - \mathcal{E}_\ell) + \frac{I_e}{A}, \end{aligned} \tag{1.13}$$

where I_e is divided by A because we are considering the current flow per unit area.

1.3 Passive Integrate-and-Fire Models

Integrate-and-fire models stipulate that a neuron will usually fire an action potential when its membrane potential reaches a threshold value V_{th} of around -55 to -50 mV. It then rapidly depolarizes before return to a reset value V_{reset}. Ignoring the role of active conductances and relying solely on the leakage in action potential analyses results in the *passive integrate-and-fire model.* The model behaves like an electric circuit with a resistor and capacitor in parallel, the behavior of which is described by eq. 1.13. If we multiply both sides by $r_m = 1/g_\ell$, we get

$$\begin{aligned} r_m c_m \frac{dV}{dt} &= -(V - \mathcal{E}_\ell) + r_m \frac{I_e}{A} \\ \Rightarrow \tau_m \frac{dV}{dt} &= -(V - \mathcal{E}_\ell) + R_m I_e. \end{aligned} \tag{1.14}$$

We can see that when $I_e = 0$, the neuron will relax exponentially with time constant τ_m to $\mathcal{E}_\ell$, its resting potential. In other words, $\mathcal{E}_\ell = V_{reset}$. We can solve for the subthreshold potential $V(t)$:

$$\begin{aligned} \tau_m \frac{dV}{dt} &= -(V(t) - \mathcal{E}_\ell) + R_m I_e \\ \Rightarrow \tau_m \frac{dU}{dt} &= -U(t) \quad (\text{change of vars: } U = V - \mathcal{E}_\ell - R_m I_e) \\ \Rightarrow \int_0^t \frac{dU}{U} &= \int_0^t -\frac{1}{\tau_m} dt' \\ \Rightarrow \log\left(\frac{U(t)}{U(0)}\right) &= -\frac{t}{\tau_m} \\ \Rightarrow U(t) &= U(0) e^{-t/\tau_m} \\ \Rightarrow V(t) &= \mathcal{E}_\ell + R_m I_e + (V(0) - \mathcal{E}_\ell - R_m I_e) e^{-t/\tau_m}. \end{aligned} \tag{1.15}$$

Note that this expression holds only for $V(t) < V_{th}$. Suppose that $V(0) = V_{reset}$. Then the time until the neuron spikes (the interspike interval) t_{isi} is the time at which the voltage reaches the threshold potential:

$$V(t_{isi}) = V_{th} = \mathcal{E}_\ell + R_m I_e + (V_{reset} - \mathcal{E}_\ell - R_m I_e) e^{-t_{isi}/\tau_m}. \tag{1.16}$$

Solving for the firing rate r_{isi} (the inverse of the inter-spike interval) gives

$$r_{isi} = \frac{1}{t_{isi}} = \left[\tau_m \log\left(\frac{V_{reset} - \mathcal{E}_\ell - R_m I_e}{V_{th} - \mathcal{E}_\ell - R_m I_e}\right)\right]^{-1}. \tag{1.17}$$

Note that this expression is valid when $V_{th} - \mathcal{E}_\ell > R_m I_e$. The firing pattern for a simulated neuron with time-varying input current is shown in Figure 1.2. We can use the approximation $\log(1 + z) \approx z$ for small z

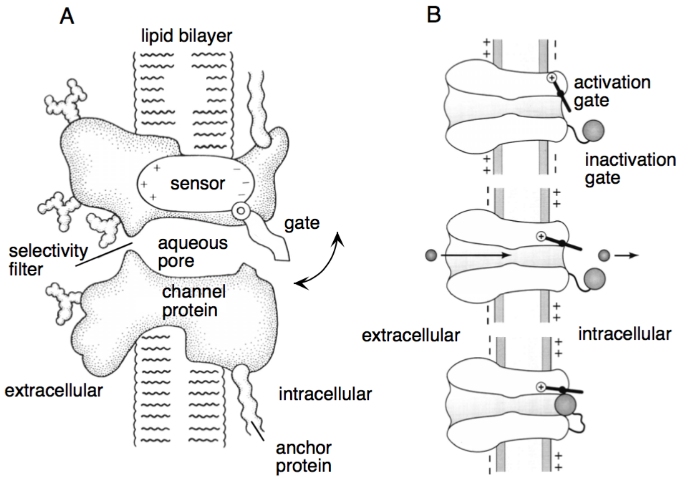


Figure 1.3: Simplified depictions of persistent (A) and transient (B) conductance channels. Descriptions in text.

to show that r_{isi} grows linearly with I_e for large I_e. It's also possible to consider alternative models for the dynamics, such as *quadratic* integrate-and-fire (QIF) and *exponential* integrate-and-fire (EIF) models, which take the following general forms:

$$\tau\frac{dV}{dt} \propto V^2 + \beta V + V_{ext}(t), \tag{1.18}$$

$$\tau\frac{dV}{dt} \propto \exp(V/\gamma V_0) - \alpha V + V_{ext}(t), \tag{1.19}$$

respectively. The neuron is said to fire when $V \to \infty$. The EIF is generally a better model for real neurons because the time it takes to fire can be tuned by its parameters, while for the QIF (and LIF) it's dependent solely on the membrane time constant. The EIF is also a better fit for experimental data.

1.4 Active Channels and Voltage-Dependent Conductance

Many important biophysical properties of neurons arise as a result of changing channel conductances. There are several factors that can lead to varying conductances, such as synaptic conductances that depend on the presence or absence of a neurotransmitter, or channels that depend on internal messenger molecules or the concentration of ions like Ca^{2+}. Here, however, we'll focus on *voltage-dependent* conductances, which depend on the membrane potential of the neuron. Assuming independence among channels, we can define the conductance per unit area of membrane for channel type i as follows:

$$g_i := \rho_i g_i^{open} P_i = \bar{g}_i P_i, \tag{1.20}$$

where ρ_i is the density of channels of type i in the membrane, g_i^{open} is the conductance of an open channel of type i, and P_i is the probability that any given such channel is open at a given time. $\bar{g}_i$ is then the conductance per unit area if all such channels are open; units typically range from $\mu\mathrm{S/mm}^2$ to $\mathrm{mS/mm}^2$. Two important channels are the delayed-rectifier K^+ conductance and the fast Na^+ conductance.

Persistent Conductances The delayed-rectifier K^+ conductance that is responsible for repolarizing a neuron after it fires is an example of a *persistent conductance* channel. Channels with persistent conductance (depicted in figure 1.3A) behave as though they only carry one kind of *gate* that swings open in response to a voltage-dependent sensor. Opening of the gate(s) is termed *activation* and closing of the gate is referred to as *deactivation*. For this type of channel, the probability that it's open, P_{K^+}, increases when the neuron is depolarized and decreases when it is hyperpolarized.

The gating mechanism of the delayed-rectifier K^+ channel consists of four identical subunits, which appear to open independently. In general, if k independent events are required for a channel to open, P_{K^+} can be written as

$$P_{\mathrm{K}^+} = n^k = n^4, \tag{1.21}$$

where n is the probability that any of the gating events has occurred (i.e., that a gate subunit is open; $1-n$ is the probability it is closed). The variable n is called a *gating variable*, and a description of its voltage and time

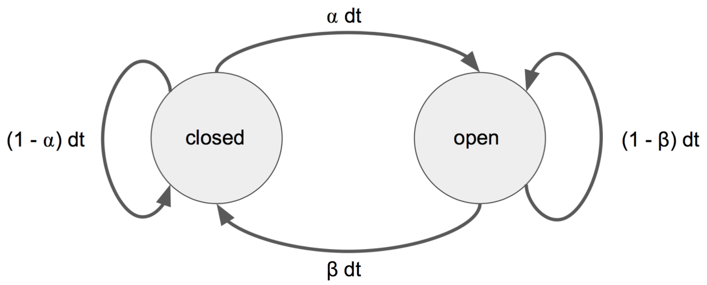


Figure 1.4: Markovian transition dynamics for active channel gates.

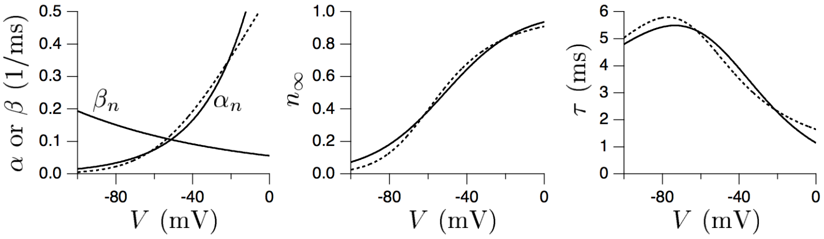


Figure 1.5: Example plots of channel opening and closing rates (left), limiting values for the opening probability (center), and the time constant (right) for the delayed-rectifier K^+ conductance.

dependence is sufficient for a description of the conductance. We model the transition probabilities over a time interval dt follows (summarized in Figure 1.4):

$$\begin{cases} p(\text{closed} \to \text{open}) & = \alpha(V)dt \\ p(\text{open} \to \text{closed}) & = \beta(V)dt. \end{cases} \tag{1.22}$$

To obtain a differential equation governing these gating dynamics, we can write

$$\begin{aligned} n(t+dt) &= p(\text{open at } t+dt) && (1.23)\\ &= p(\text{open}(t))p(\text{open}(t+dt)|\text{open}(t)) + p(\text{closed}(t))p(\text{open}(t+dt)|\text{closed}(t)) && (1.24)\\ &= n(t)(1-\beta dt) + (1-n(t))\alpha dt. && (1.25) \end{aligned}$$

We then use a linear Taylor approximation, $n(t+dt) \approx n(t) + dt\frac{dn}{dt}$. Applying this gets us

$$\begin{aligned} n(t) + dt\frac{dn}{dt} &\approx n(t)(1-\beta dt) + (1-n(t))\alpha dt && (1.26)\\ &= n(t) - n(t)\beta dt + \alpha dt - n(t)\alpha dt && (1.27)\\ \Rightarrow \frac{dn}{dt} &\approx -n(t)\beta + \alpha - n(t)\alpha && (1.28)\\ &= \alpha(1-n(t)) + \beta n(t) && (1.29)\\ &= \alpha - (\alpha+\beta)n(t) && (1.30) \end{aligned}$$

Dividing both sides of eq. 1.30 by $\alpha + \beta$ gives

$$\underbrace{\frac{1}{\alpha(V)+\beta(V)}}_{\tau_n(V)}\frac{dn}{dt} = \underbrace{\frac{\alpha(V)}{\alpha(V)+\beta(V)}}_{n_\infty(V)} - n(t), \tag{1.31}$$

$$\Rightarrow \tau_n(V)\frac{dn}{dt} = n_\infty(V) - n(t). \tag{1.32}$$

This indicates that for a fixed voltage, the opening probability approaches the limiting value $n_\infty(V)$ exponentially with time constant $\tau_n(V)$. Simple thermodynamic arguments (see Dayan & Abbott p. 170) can be made to show that $n_\infty(V)$ is sigmoidal–depolarization causes n to grow towards 1, and hyperpolarization causes it to shrink toward 0. Following this, the opening rate α is an increasing function of V, while β is decreasing. These functions are usually fitted using experimental data obtained from voltage clamping. Example traces of α, β, n_∞, and τ are plotted in Figure 1.5.

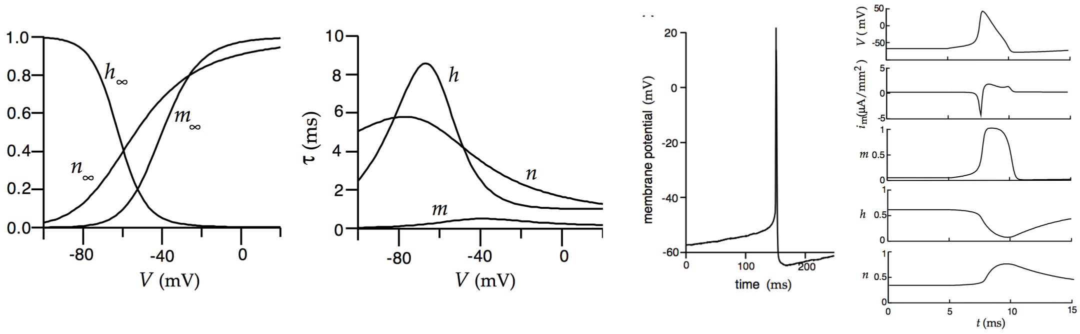


Figure 1.6: Example plots of steady-state values for the opening probabilities of the Na$^+$ and K$^+$ channels (left), along with associated time constants (middle-left), and an example action potential (middle-right) and the traces of each gate during it (right). The behavior of the gates during an action potential is as follows: From a hyperpolarized state, the m gates open quickly (see the time constant), allowing Na$^+$ to flood in. This rapidly depolarizes the neuron, causing the the slower h gates to shut, stopping the influx of Na$^+$ and re-hyperpolarizing the neuron. The persistent n gates then open, causing the slight re-depolarization to the steady-state at the end. This process is summarized by the rightmost panel. Note that if m and h had the same time constants, they would cancel each other's effects and nothing would happen. In general, the time constants determine the width of the action potential.

Transient Conductances Some channels only open transiently when the membrane potential depolarizes because they contain gates with opposite voltage dependences. The fast Na$^+$ conductance is an example of such a channel. Schematically, it can be thought of as having $k = 3$ swinging activation gates m who increase their probability of opening with increasing voltage, and an inactivation gate/ball h ($k = 1$) which closes with depolarization (Figure 1.3B). For the channel to conduct, both sets of gates must be open, which has probability

$$P_{\mathrm{Na}^+} = m^k h = m^3 h. \tag{1.33}$$

The probability variables m and h follow analogous equations to n, with similar forms for α and β. The steady state activation and inactivation functions $m_\infty(V)$ and $h_\infty(V)$, along with the associated time constants, are also similar to those for the K channel (although h_∞ is inverted, as it's an inactivation variable). These functions are visualized in Figure 1.6. To activate such a transient channel, it's required that both the m and h gates are nonzero–to do this maximally, it's best for the neuron to first hyperpolarize (activating h), and then quickly depolarize (activating n). The point of maximum activation is the intersection of the two curves–note that this is approximately the threshold voltage for spiking in a neuron.

1.5 The Hodgkin-Huxley Equations

The Hodgkin-Huxley (HH) equations are simply a condensation of what we've derived so far, modeling the effects passive and active channels on the membrane voltage dynamics. Combining equations 1.14, 1.20, 1.21, and 1.33, and ignoring external current injection, we get

$$C\frac{dV}{dt} = -i_m + \underbrace{\frac{I_{ext}}{A}}_{=0} = -\bar{g}_\ell(V - \mathcal{E}_\ell) - \bar{g}_{\mathrm{Na}^+} m^3 h(V - \mathcal{E}_{\mathrm{Na}^+}) - \bar{g}_{\mathrm{K}^+} n^4 (V - \mathcal{E}_{\mathrm{K}^+}). \tag{1.34}$$

Dividing both sides by $\bar{g}_\ell$ gives

$$\boxed{\tau\frac{dV}{dt} = -(V - \mathcal{E}_\ell) - \rho_{\mathrm{Na}^+} m^3 h(V - \mathcal{E}_{\mathrm{Na}^+}) - \rho_{\mathrm{K}^+} n^4 (V - \mathcal{E}_{\mathrm{K}^+}),} \tag{1.35}$$

where $\rho_{\mathrm{Na}^+} = \bar{g}_{\mathrm{Na}^+}/\bar{g}_\ell \approx 400$ and $\rho_{\mathrm{K}^+} = \bar{g}_{\mathrm{K}^+}/\bar{g}_\ell \approx 120$. We can also generalize eq. 1.32 for the dynamics of the opening probability of each gating variable, giving

$$\boxed{\tau_x(V)\frac{dx}{dt} = x_\infty(V) - x(t), \quad x \in \{m, n, h\}.} \tag{1.36}$$

Equations 1.35 and 1.36 are the **Hodgkin-Huxley** equations. Eq. 1.36 can be equivalently expressed as

$$\tau_x(V)\frac{dx}{dt} = \alpha_x(1 - x(t)) + \beta_x x(t) \quad \text{(see eq. 1.29)}. \tag{1.37}$$

As these are highly nonlinear equations in four variables, they can't be solved directly, and must be approximated. There are two commonly used approximations.

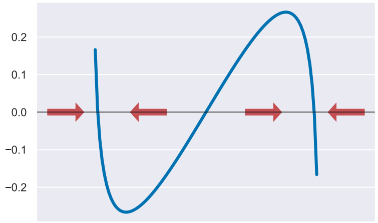


Figure 1.7: HH approximation #1: all gates set to their equilibrium values.

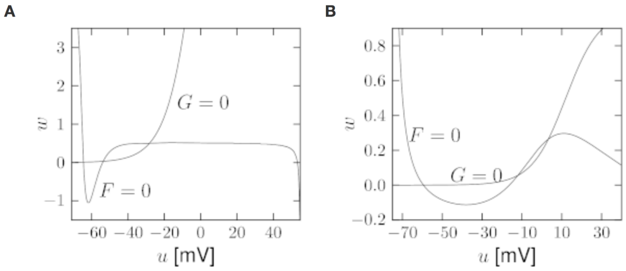


Figure 1.8: HH approximation #2: reduction to a 2D system. The left nullclines (A) are those for the HH model, rigorously reduced to 2D via a linear fitting for $w(t)$ and the right (B) are those for the Morris-Lecar approximation. Notationally, $u = V$, F is the V-nullcline and G is the w-nullcline.

1.5.1 Type I and Type II Neurons

In the first simplifying assumption, we assume that gating variables always hold their steady-state values–that is, $\tau_x = 0 \Rightarrow x = x_\infty(V)\,\forall x$. Then eq. 1.35 becomes

$$\tau\frac{dV}{dt} = -(V - \mathcal{E}_\ell) - \rho_{\mathrm{Na}^+} m_\infty^3 h_\infty (V - \mathcal{E}_{\mathrm{Na}^+}) - \rho_{\mathrm{K}^+} n_\infty^4 (V - \mathcal{E}_{\mathrm{K}^+}) + V_{ext}(t), \tag{1.38}$$

a one-dimensional system. This is equivalent to assuming that the membrane time constant τ is much larger than the gating time constants τ_m, τ_h, and τ_n. This yields a cubic function on the V-$\dot{V}$ phase plane with three roots—the two leftmost roots bound a local minimum (the left root is stable, and the center root is unstable), and the middle and the right root bound a stable local maximum (Figure 1.7). Changing the external current shifts the cubic function up and down. When it is sufficiently high, the left and center roots disappear, leaving only the right (stable) point. On the other hand, setting V_{ext} quite low shifts the function downwards, destroying the center and right roots and leaving only the left (stable) point. Thus, by modulating the external input, the neuron can effectively function as a switch between high (ON) and low (OFF) states. This could be a realistic model, except it results in dynamics that are very energy-intensive—ions pumps need to work incredibly hard to maintain the higher (ON) state. The shape of the resulting dynamics is also inconsistent with experimental evidence.

The second possible approximation (and a more biologically realistic one) is to let $m \to m_\infty(V)$, as the time constant for m is so much lower than for n and h, and to combine the slower n and h conductances—more precisely, we combine n and $1 - h$—into one dynamical variable $w(t)$ with its own reversal potential $\mathcal{E}_w$ and average conductance $\bar{\rho}_w$. This gives the simplified 2D *Morris-Lecar model* of action potential dynamics:

$$\tau\frac{dV}{dt} = -(V - \mathcal{E}_\ell) - \bar{\rho}_w w(t)(V - \mathcal{E}_w) - \bar{\rho}_m m_\infty(V)(V - \mathcal{E}_m) + V_{ext}(t) \tag{1.39}$$

$$\tau_w\frac{dw}{dt} = w_\infty(V) - w(t), \tag{1.40}$$

with $\tau_w(V) \approx \tau_n, h(V)$. Although a simplification, this system retains the qualitative behavior of the HH equations, as visualized in Figure 1.8.

Because the system is 2D, we can easily examine its behavior on the V-w plane. We can see that the nullclines (Figure 1.8B) imply three fixed points, and it turns out the leftmost is always stable, corresponding to the resting membrane potential. The right fixed point is typically unstable, and the center fixed point is a saddle point. Changing the input current via V_{ext} shifts the V-nullcline ($F = 0$) up and down in the plane. We can see that as the external input current increases and the V-nullcline shifts up, the left stable fixed point and the saddle point grow closer together and eventually disappear, leaving only the unstable fixed point at high V. However, since the derivatives around it still point towards the fixed point, the Poicaré-Bendixson theorem tells us that the system must form a limit cycle around it. In other words, if you increase the input current sufficiently—above some threshold I_θ—the neuron starts spiking repeatedly, and the change in the number of fixed points at $I_{ext} = I_\theta$ is called a *bifurcation.* The input current I is then called a *bifurcation parameter.* In neuroscience, I_θ, the threshold current required to induce spiking, is called the *rheobase.*

It's then natural to investigate the frequency of the resulting limit cycle oscillations, as it gives insight into the neuron's firing rate response to a given constant input I—its so-called *gain function.* Consider the behavior of the system when $I < I_\theta$ and the right fixed point is unstable. In this case, trajectories starting to the right of the saddle wrap around the unstable node counter-clockwise, eventually returning to the stable fixed point (Figure 1.9, left). When I grows slightly larger than I_θ and the dynamics bifurcate, this behavior is maintained in the resulting limit cycle, such that the trajectories still pass through the area where the stable fixed point used to be. Moreover, when they pass through, the magnitude of the derivatives decreases, lowering the oscillation frequency and slowing the firing rate. When I grows even larger, this slowdown is alleviated and the spiking frequency increases. Neurons with this type of behavior are called *Type I*, and are characterized by a smooth, monotonic increase in firing rate as the input current increases (Figure 1.10A,B). When two fixed points merge like this, it's called a *saddle node* bifurcation. Intuitively, such dynamics are useful for encoding a continuous quantity, such as the overall strength of pre-synaptic input.

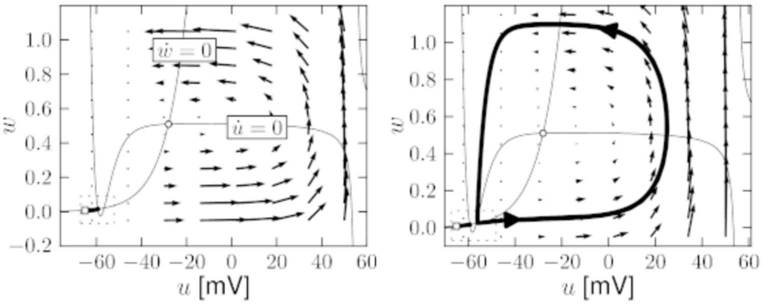


Figure 1.9: The phase plane trajectories for a Type I neuron (left) and a Type II neuron (right).

When the right fixed point is a limit cycle to begin with, however, different behavior occurs. In this case, the oscillatory trajectories pass by just to the right of the saddle, instead of the left stable region (Figure 1.9, right), and the dynamics are stuck at the low fixed point—there is no firing. Then, when I increases above I_θ and the left and center fixed points vanish, trajectories are pushed onto this limit cycle, without entering the region where the stable point used to be and slowing down. This type of transition, from a stable fixed point to a limit cycle, is called a *Hopf bifurcation.* Neurons whose gain function (and firing rate) jumps suddenly to a high value from zero when $I > I_\theta$ are termed *Type II* (Figure 1.10C,D). This type of behavior is useful for encoding a binary variable, acting like a switch with ON/OFF settings.

Additionally, the *Connor-Stevens model* of action potential generation provides an alternative formulation to the HH equations. In the Connor-Stevens model, the fast Na^+ and delayed-rectifier K^+ conductances have slightly different properties—in particular, they have smaller time constants, so action potentials are briefer. Additionally, the Connor-Stevens model incorporates an additional K^+ conductance, called the A-current, that is transient. The membrane current i_m is given by

$$i_m^{CS} := (V - \mathcal{E}_\ell) + \rho_{\mathrm{Na}^+} m^3 h (V - \mathcal{E}_{\mathrm{Na}^+}) + \rho_{\mathrm{K}^+} n^4 (V - \mathcal{E}_{\mathrm{K}^+}) + \bar{g}_A a^3 b (V - \mathcal{E}_A), \tag{1.41}$$

where the gating variables a and b behave similarly to those used in the HH model. The inclusion of the A-current is another way to differentiate Type I and Type II neurons—Type I behavior is obtained when it is turned on, as in the Connor-Stevens model, and Type II behavior is obtained when it is turned off, as in the classic HH formulation.

1.6 Passive Dendrites and Cable Theory

1.6.1 The Cable Equation

One key assumption of the single-compartment neuron model is that the membrane potential is uniform throughout the cell. However, this is a crude approximation in many cases, and membrane potential often varies, especially with respect to long, attenuated extensions such as dendrites and axons, or in the case of rapidly

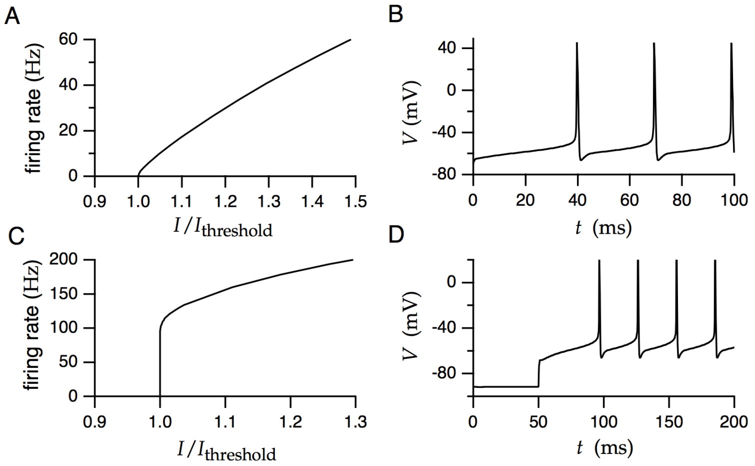


Figure 1.10: Type I (A,B) and Type II (C,D) firing rates and action potentials.

changing potentials. Such differences in potential cause current to flow, and are essential in the propagation of action potentials and other signalling. *Cable theory* is the study of the propagation of such signals. We consider the case of propagation in dendrites first. Here, we assume that the width of the dendrite is small enough that differences in potential do not occur along radial or axial directions, but solely longitudinally. The voltage is then a function of distance along the cable x and time t, $V(x,t)$. To analyze the current flow, we

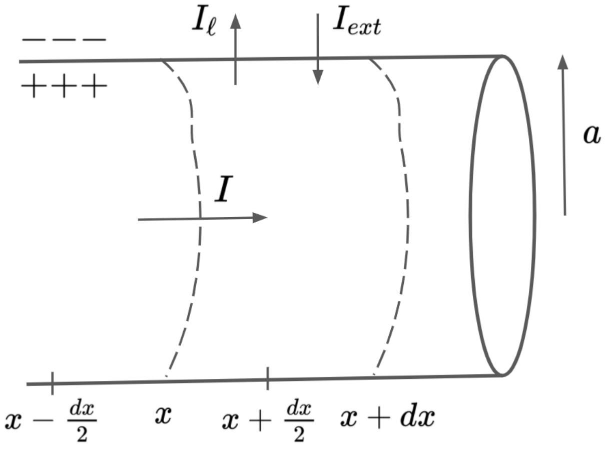


Figure 1.11: Current propagation in a dendrite.

cut up the dendrite (with radius a) into infinitesimal slices of width dx, in which we assume there is no change in current/potential. We assume a leakage current $I_\ell(x,t)$ and an external input current $I_{ext}(x,t)$, and that the current flow $I(x)$ in the dendrite is constant in time. The set-up is summarized in Figure 1.11. Denoting incoming current as positive and outgoing current as negative, the equation for the membrane potential is

$$C\frac{\partial V(x,t)}{\partial t} = I(x - dx/2) - I(x + dx/2) - I_\ell(x,t) + I_{ext}(x,t). \tag{1.42}$$

The equations for the current are simply given by

$$I(x - dx/2) = \frac{V(x-dx) - V(x)}{R}; \qquad I(x + dx/2) = \frac{V(x) - V(x+dx)}{R}, \tag{1.43}$$

where R is the longitudinal resistance. Inserting this into the voltage equation gives

$$C\frac{\partial V(x,t)}{\partial t} = \frac{V(x-dx) - 2V(x) + V(x+dx)}{R} - I_\ell(x,t) + I_{ext}(x,t). \tag{1.44}$$

Performing a second-order Taylor expansion on the voltage results in

$$V(x-dx) - 2V(x) + V(x+dx) \approx V(x) - dx\frac{\partial V}{\partial x} + \frac{dx^2}{2}\frac{\partial^2 V}{\partial x^2} - 2V(x) + V(x) + dx\frac{\partial V}{\partial x} + \frac{dx^2}{2}\frac{\partial^2 V}{\partial x^2} \tag{1.45}$$

$$= dx^2\frac{\partial^2 V}{\partial x^2}. \tag{1.46}$$

Plugging this into eq. 1.44 gives

$$C\frac{\partial V(x,t)}{\partial t} = \frac{dx^2}{R}\frac{\partial^2 V}{\partial x^2} - I_\ell(x,t) + I_{ext}(x,t). \tag{1.47}$$

What we really want, though, is this expression in the limit $dx \to 0$. To get this, we need to know how R and C scale with dx. Resistance is proportional to length and inversely proportional to area: $R = r_L \times \frac{\text{length}}{\text{area}}$, where r_L is the resistivity of the dendrite. For a cylindrical cable with radius a, we then have

$$R = r_L \frac{dx}{\pi a^2}. \tag{1.48}$$

To see how the C scales, recall that, in general, capacitance is proportional to area. Thus, $C = c_m \times \text{area}$, where c_m is the specific capacitance of the membrane. The relevant voltage drop is across the dendritic walls, so we have

$$C = c_m 2\pi a dx. \tag{1.49}$$

Inserting these expressions into eq. 1.47, we get

$$\begin{aligned} c_m 2\pi a dx \frac{\partial V(x,t)}{\partial t} &= \frac{dx^2}{r_L \frac{dx}{\pi a^2}}\frac{\partial^2 V}{\partial x^2} - I_\ell(x,t) + I_{ext}(x,t) \\ \Rightarrow c_m \frac{\partial V(x,t)}{\partial t} &= \frac{a}{2r_L}\frac{\partial^2 V}{\partial x^2} - \frac{I_\ell(x,t)}{2\pi a dx} + \frac{I_{ext}(x,t)}{2\pi a dx}. \end{aligned} \tag{1.50}$$

However, there's still a dependence on dx. To get rid of it (or at least kind of hide it), we can define the current densities

$$i_\ell(x,t) := \frac{I_\ell(x,t)}{2\pi a dx}; \qquad i_e(x,t) := \frac{I_{ext}(x,t)}{2\pi a dx}. \tag{1.51}$$

Inserting these into the above equation almost gives us the passive cable equation. The last thing we need to do is write down an expression for i_ℓ in terms of the voltage. We could use Hodgkin-Huxley type equations, but here we'll stick to passive channels. For that we'll write, as usual,

$$I_\ell = \frac{V - \mathcal{E}_\ell}{R_m}, \tag{1.52}$$

where R_ℓ is the resistance across the membrane. As before, resistance is proportional to distance and inversely proportional to area. However, since the distance across the membrane is narrow compared to the diameter of the dendrite and is essentially constant, we'll ignore it, writing

$$R_m = \frac{r_m}{2\pi a dx}. \tag{1.53}$$

Combining this with the equation for I_ℓ gives

$$\begin{aligned} i_\ell(x,t) = \frac{I_\ell(x,t)}{2\pi a dx} &= \frac{1}{2\pi a dx}\frac{V - \mathcal{E}_\ell}{R_m} = \frac{V - \mathcal{E}_\ell}{2\pi a dx}\frac{2\pi a dx}{r_m} \\ &= \frac{V - \mathcal{E}_\ell}{r_m}. \end{aligned} \tag{1.54}$$

Inserting this into eq. 1.50, we get

$$c_m \frac{\partial V(x,t)}{\partial t} = \frac{a}{2r_L}\frac{\partial^2 V}{\partial x^2} - i_\ell(x,t) + i_e(x,t) \tag{1.55}$$

$$= \frac{a}{2r_L}\frac{\partial^2 V}{\partial x^2} - \frac{V - \mathcal{E}_\ell}{r_m} + i_e(x,t). \tag{1.56}$$

Multiplying both sides by r_m results in the standard cable equation:

$$\boxed{\tau_m \frac{\partial u}{\partial t} = \lambda^2 \frac{\partial^2 u}{\partial x^2} - u + r_m i_e,} \tag{1.57}$$

where $\tau_m = r_m c_m$ is the membrane time constant, $u := V - \mathcal{E}_\ell$, and

$$\lambda := \sqrt{\frac{a r_m}{2 r_L}}, \tag{1.58}$$

where λ is the *electrotonic length*, which defines the length-scale on which voltage varies longitudinally in the dendrite.

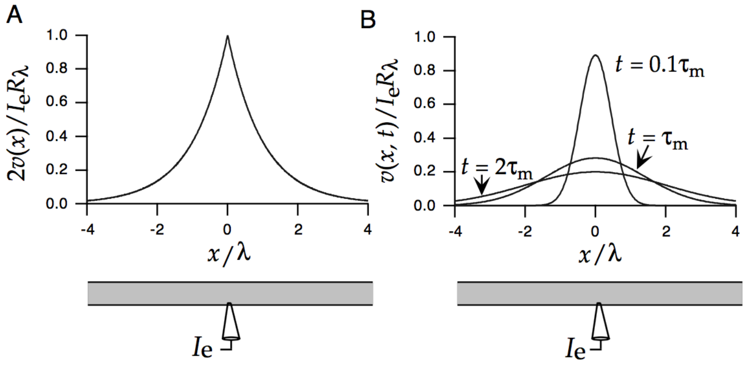


Figure 1.12: Voltage propagation in an infinite cable with injection at $x = 0$. (A) Solution for a constant electrode current. It decays exponentially from the injection point. (B) The solution for a (time-dependent) δ-pulse of current. Its described by a Gaussian centered at the injection point that broadens and shrinks in amplitude over time.

1.6.2 Infinite Cables

To ease analysis, it useful to make the assumption that the cable is effectively infinite; for areas of the dendrite far from either of its ends, this is not a bad approximation.

Constant Current Injection If there is a constant injected current at a spatial location x, we lose the time-dependence and get

$$i_e(x,t) = \frac{I_e}{2\pi a}\delta(x). \tag{1.59}$$

In these conditions, the membrane potential will settle to a steady-state value, resulting in $\frac{\partial u}{\partial t} = 0$. This gives

$$\lambda^2 \frac{\partial^2 u}{\partial x^2} - u + r_m i_e \delta(x) = 0. \tag{1.60}$$

For $x \neq 0$, then, we can solve the following homogeneous second-order ODE (see section A.2):

$$\lambda^2 \frac{\partial^2 u}{\partial x^2} - u = 0 \Rightarrow \frac{\partial^2 u}{\partial x^2} - \frac{1}{\lambda^2}u = 0. \tag{1.61}$$

Then $p = 0$ and $q = -1/\lambda^2$, so by the quadratic formula

$$a, b = \frac{\pm\sqrt{4/\lambda^2}}{2} = \pm\frac{1}{\lambda}, \tag{1.62}$$

and the solution is of the form

$$u(x) = c_1 e^{-x/\lambda} + c_2 e^{x/\lambda}. \tag{1.63}$$

Because $u(x)$ must be bounded when $x \to \pm\infty$, we need $c_1 = 0$ for the region $x < 0$ and $c_2 = 0$ for $x > 0$. Moreover, because the solution must be continuous at $x = 0$, we need $c_1 = c_2 = c$. Thus, we can combine these solutions into a single expression:

$$u(x) = ce^{-|x|/\lambda}. \tag{1.64}$$

It's not too difficult (see Jorge's notes, or Dayan & Abbott p. 209) to show that $c = \frac{I_e R_\lambda}{2}$, where $R_\lambda := \frac{r_m}{2\pi a \lambda} = \frac{r_\ell \lambda}{\pi a^2}$. Then we have

$$u(x) = \frac{I_e R_\lambda}{2} e^{-|x|/\lambda}. \tag{1.65}$$

This solution (with y-axis normalized) is plotted in Figure 1.12A. Thus, λ sets the intrinsic length scale for passive dendrites–they can't be much longer, or any signal propagated along them would vanish.

Instantaneous Current Injection Consider a δ-pulse of current injected at $x = 0$ and time $t = 0$, e.g.,

$$i_e = \frac{\tau_m I_e}{2\pi a}\delta(x)\delta(t), \tag{1.66}$$

so that the current pulse delivers a total charge of $\tau_m I_e$. The derivation for $u(x,t)$ isn't repeated here (see Jorge's notes), but the result is

$$u(x,t) = \frac{I_e R_\lambda}{\sqrt{4\pi t/\tau_m}} \exp\left(-\frac{\tau_m x^2}{4\lambda^2 t}\right) \exp\left(-\frac{t}{\tau_m}\right). \tag{1.67}$$

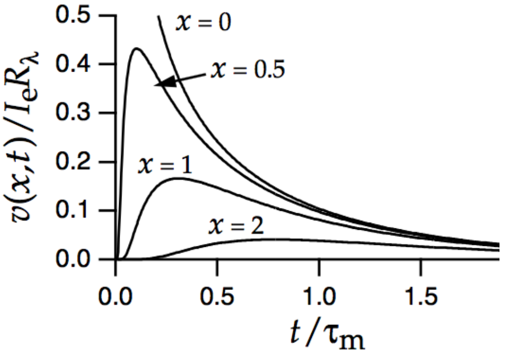


Figure 1.13: Voltage propagation across time for different fixed distances from the point of injection. Greater distances have greater delays in peak time.

We can then see that the spatial dependence is Gaussian, with λ again setting the scale for spatial variation. The width then also increases proportional to the square root of the time since the pulse, creating a widening, flattening curve. This effect is plotted in Figure 1.12B.

The solutions for eq. 1.67 across time at varying fixed distances are plotted in Figure 1.13. We can see that the peak occurs later for distances that are farther from the injection point. Though the voltage does not strictly propagate like a wave, we can measure its "velocity" by the time it takes to reach its maximum at varying distances. This can be done by setting the time derivative of eq. 1.67 to zero, giving

$$t_{max} = \frac{\tau_m}{4}\left(\sqrt{(1+4(x/\lambda)^2} - 1\right). \tag{1.68}$$

For large x, $t_{max} \approx \frac{\tau_m x}{2\lambda}$, corresponding to a velocity of

$$\boxed{v_{dendrite} = 2\lambda/\tau_m} \tag{1.69}$$

in a passive dendrite. For smaller values of x, the voltage propagates more quickly than this expression implies. Therefore, this approximation is more accurate for locations far from the injection site.

1.7 Axons

NOTE: the following section is pretty much taken from Jorge's notes.

Unlike dendrites, axons need to propagate information over longer distances and therefore require higher speeds of propagation. Given that r_L and c_m are intrinsic properties of the cell cytoplasm and phospholipid bilayer, the two parameters we can manipulate to achieve higher speeds are a (axon radius) and r_m (membrane resistance). It turns out the mammalian brain does both. To change r_m, long-range projecting axons are often *myelinated*: they are wrapped with layers of cell membrane (myelin) that effectively increase the membrane resistance. We model this by taking $r_m \to \infty$. Rearranging the passive cable equation to take this limit and then using the same strategy as above to solve for the propagation of a pulse of injected current (Fourier transform in space → solve differential equation in time → inverse Fourier transform of a Gaussian), we get:

$$\begin{aligned} c_m \frac{\partial u}{\partial t} &= \frac{\lambda^2}{r_m}\frac{\partial^2 u}{\partial x^2} - \frac{u}{r_m} + i_e\delta(x)\delta(t) \\ &= \frac{a}{2r_L}\frac{\partial^2 u}{\partial x^2} - \frac{u}{r_m} + i_e\delta(x)\delta(t) \\ \Rightarrow \lim_{r_m\to\infty} \frac{\partial u}{\partial t} &= \frac{a}{2c_m r_L}\frac{\partial^2 u}{\partial x^2} + \frac{i_e}{c_m}\delta(x)\delta(t). \\ \Rightarrow u(x,t) &= \frac{i_e}{\sqrt{\pi D t}}\Theta(t)e^{-\frac{x^2}{Dt}}, \end{aligned} \tag{1.70}$$

where $D = 2a/r_L c_m$. Note the lack of a term decaying exponentially with time, meaning that in this setting the signal propagates as a Gaussian spreading in time, with constant integral (an intuitive result from the fact that myelination effectively eliminates the leak current). This slowing down of the signal decay results in faster "velocity" of the propagating signal in the axon, which we can compute by taking the log of u and then the derivative in time and setting to zero:

$$\begin{aligned}\log u(x,t) &= \log i_e - \frac{1}{2}\log t - \frac{1}{2}\log \pi D - \frac{x^2}{Dt}\\ \Rightarrow \frac{\partial}{\partial t}\log u(x,t) &= -\frac{1}{2t} + \frac{x^2}{Dt^2} = 0\\ \Rightarrow \frac{x^2}{Dt_{max}} &= \frac{1}{2}\\ \Rightarrow t_{max} &= \frac{2x^2}{D} = \frac{r_L c_m x^2}{a}\\ \Rightarrow v_{axon} &= \frac{a}{r_L c_m x}.\end{aligned} \tag{1.71}$$

However, this doesn't seem to work, as $v \propto 1/x$, and so the speed of propagation will decay rapidly. Mammalian nervous systems solve this by having $a \propto x$–axons get thicker as they get longer. This results in

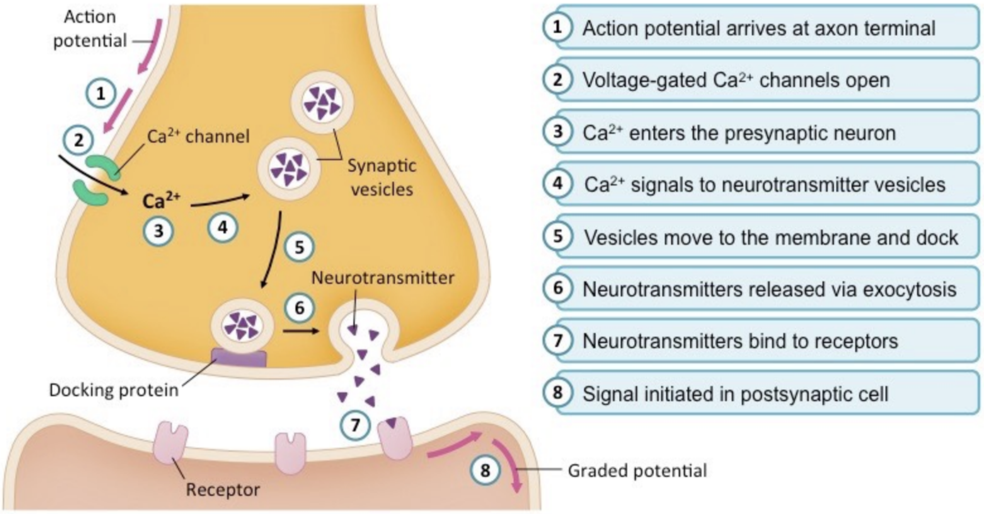
$$v_{axon} = \frac{1}{r_L c_m} = \frac{2\pi a}{r_L C_m}. \tag{1.72}$$

Therefore, we (approximately) have

$$\begin{gathered}\boxed{v_{dendrite} \propto \sqrt{a}}\\ \boxed{v_{axon} \propto a}.\end{gathered} \tag{1.73}$$

However, since the length-scale is still set by λ, the width of the resulting Gaussian is the same as for passive dendrites, and so the signal will still rapidly decay to zero for distances further than $2\sqrt{Dt}$. To solve this, axons separate segments of myelination with so-called *nodes of Ranvier* where there is a high concentration of active Na^+ channels that can initiate an action potential if the membrane potential gets high enough. This is called *saltatory conductance*, since the action potential "jumps" (*salta*, in Spanish) from one node to the next.

1.8 Synaptic Transmission



1 Action potential arrives at axon terminal

2 Voltage-gated Ca2+ channels open

3 Ca2+ enters the presynaptic neuron

4 Ca2+ signals to neurotransmitter vesicles

5 Vesicles move to the membrane and dock

6 Neurotransmitters released via exocytosis

7 Neurotransmitters bind to receptors

8 Signal initiated in postsynaptic cell

Figure 1.14: Visualization of synaptic transmission.

Synaptic transmission at a spike-mediated chemical synapse begins when an action potential invades the presynaptic terminal and activates voltage- dependent Ca^{2+} channels, leading to a rise in the concentration of Ca^{2+} within the terminal. This causes vesicles containing transmitter molecules to fuse with the cell membrane and release their contents into the synaptic cleft between the pre- and postsynaptic sides of the synapse. The transmitter molecules then diffuse across the cleft and bind to receptors on the postsynaptic neuron. Binding of transmitter molecules leads to the opening of ion channels that modify the conductance of the postsynaptic neuron, completing the transmission of the signal from one neuron to the other. Postsynaptic ion channels can be activated directly by binding to the transmitter, or indirectly when the transmitter binds to a distinct receptor that affects ion channels through an intracellular second-messenger signaling pathway (direct quote from Dayan & Abbott). This process is visualized in Figure 1.14.

Neurotransmitter	Receptor	Time constant	Ions	Type
Glutamate	AMPA	fast ($\sim$ 1 ms)	cations	ionotropic
	NMDA	slow	cations, incl. Ca^{2+}	ionotropic
GABA	GABA_A	fast	Cl$^-$ conductance	ionotropic
	GABA_B	flow	K^+ conductance	metabotropic

Table 1: Common neurotransmitters and receptor types.

As with standard channel conductances, synaptic channel conductances can be modeled as the product of an average conductance term and an opening probability: $g_s = \bar{g}_s P$, where in this case

$$P = P_{rel} P_s. \tag{1.74}$$

Here, P_{rel} is the probability that a vesicle successfully releases neurotransmitter from the presynaptic terminal into the synaptic cleft (given the arrival of an action potential), and P_s is the probability that the postsynaptic receptor opens to receive the neurotransmitter. Release probability is governed by two quantities. One is the amount of calcium in the presynaptic terminal, with higher calcium implying higher release probability. The other is release itself: every time a vesicle is released, the probability of subsequent release drops; then it decays exponentially back to baseline (which is calcium dependent). The concentration of calcium in the presynaptic terminal is largely independent of the amount of neurotransmitter.

There are two broad classes of synaptic conductances. In *Ionotropic* receptors, the neurotransmitter binds to the channel directly and activates it, while in *metabotropic* receptors, the neurotransmitter binds to a separate receptor and activates the conductance through an intracellular signaling pathway. Ionotropic conductances activate and deactivate more rapidly than metabotropic receptors, while in addition to opening ion channels, metabotropic receptors can induce long-term changes within the post-synaptic neuron via mechanisms like G-protein-mediated receptors and second-messengers. Serotonin, dopamine, norepinephrine, and acetylcholine all act via metabotropic receptors.

Glutamate and GABA are the major excitatory and inhibitory transmitters in the brain, and both can act ionotropically and metabotropically. The main ionotropic receptor types for glutamate are called AMPA and NMDA. Both AMPA and NMDA receptors use mixtures of cations (positive ions, such as Ca^{2+}) and have reversal potentials around 0 mV. AMPA receptors activate and deactivate rapidly, while NMDA is slower, more permeable to Ca^{2+}, and has an usual voltage dependence. GABA activated two major inhibitory conductances in the brain. GABA_A receptors produce fast ionotropic Cl$^-$ conductance, while GABA_B receptors are metabotropic and slower, producing a longer-lasting K^+ conductance.

In addition to chemical synapses, neurons can communicate via *gap junctions*, which produce a synaptic current proportional to the voltage difference at the two terminals.

1.8.1 Postsynaptic Conductances

The model for postsynaptic membrane current can be described by

$$i_s = \xi_s \sum_j \bar{g}_s P_s^j (V - \mathcal{E}_j), \tag{1.75}$$

$$\xi_s = \begin{cases} 1 & \text{w.p. } P_{rel} \\ 0 & \text{w.p. } 1 - P_{rel}, \end{cases} \tag{1.76}$$

where j indexes receptor types. Usually, at a given synapse, only one type of neurotransmitter will be released by the presynaptic cell. This is *Dale's Law*. Therefore, we'll mostly drop the j superscript in the following analysis. The gating dynamics follow similar rules (i.e., a two-state Markov model) as other active conductances, leading to the dynamics

$$\frac{dP_s}{dt} = \alpha(C)(1 - P_s) - \beta(C) P_s \tag{1.77}$$

$$\Leftrightarrow \tau(C) \frac{dP_s}{dt} = P_\infty^s(C) - P_s \tag{1.78}$$

$$\tau(C) = \frac{1}{\alpha(C) + \beta(C)}, \quad P_\infty^s(C) = \frac{\alpha(C)}{\alpha(C) + \beta(C)}, \tag{1.79}$$

where C is the concentration of the transmitter. The time constant and gating variables are also unique to each receptor/channel type. $\beta(C)$ determines the channel closing rateand is usually assumed to be a small constant

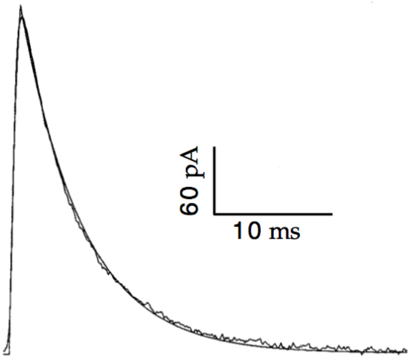


Figure 1.15: A pulse of neurotransmitter following by an exponential decay in channel opening probability.

$(\beta(C) \to \beta)$. The opening rate $\alpha(C)$, however, is dependent on the concentration of transmitter available. If the channel binds to k transmitter molecules, then the opening probability is proportional to the concentration raised to the power k: $\alpha(C) \propto C^k$. Solving eq. 1.78 gives

$$\begin{aligned} P_s(t) &= P^s_\infty(C) + (P_s(0) - P^s_\infty(C))e^{-t/\tau(C)} && (1.80)\\ &\approx 1 + (P_s(0) - 1)e^{-t/\tau} && (1.81)\\ &= 1 + (P_s(0) - 1)e^{-(\alpha+\beta)t} && (1.82)\\ &\approx 1 + (P_s(0) - 1)e^{-\alpha t}, && (1.83) \end{aligned}$$

where the last approximation is due to the fact that $\alpha \gg \beta$ when C is nonzero. We also make the assumption that $P_\infty \approx 1$ if we model the concentration $C(t)$ as a square wave for $t \in [0, T]$:

$$C(t) = \bar{C}\Theta(t)\Theta(T - t), \tag{1.84}$$

where $\bar{C}$ is the average concentration. This is a fairly accurate model, as the concentration decays rapidly after the transmitter is released into the synapse (it diffuses away and is eaten up by enzymes). With $t = 0$ being the moment transmitter is released into the synaptic cleft, the solution can then be written as

$$P_s(t) = \begin{cases} 1 + (P_s(0) - 1)e^{-(\alpha(C)+\beta)t} & t < T \\ P_s(T)e^{-\beta(t-T)} & t \geq T. \end{cases} \tag{1.85}$$

There is then an exponential increase in opening probability for the duration that transmitter is in the synapse, and an exponential decrease with time constant $1/\beta$ once it's no longer being released. If $P_s(0) = 0$, as it is in the case where no transmitter is in the cleft prior to release, then for $t \leq T$

$$P_s(t) = 1 - e^{-(\alpha(C)+\beta)t}, \tag{1.86}$$

which reaches a maximum value

$$P_{max} = P_s(T) = 1 - e^{-(\alpha(C)+\beta)T}, \tag{1.87}$$

at time $t = T$. By plugging this into eq. 1.83, we then have, in general, that

$$\begin{aligned} P(T) &= 1 + (P_s(0) - 1)e^{-(\alpha(C)+\beta)T} \\ &= 1 + P_s(0)e^{-\alpha T} - e^{-(\alpha(C)+\beta)T} \\ &= P_{max} + P_s(0)[1 - P_{max}] \\ &= P_s(0) + P_{max}[1 - P_s(0)]. \end{aligned} \tag{1.88}$$

This is equivalent to the concentration arriving as a delta pulse at $t = 0$. A visualization of this can be found in Figure 1.15. Given eq. 1.87, we can write the following synaptic conductance dynamics for a train of spikes occuring at times $\{t_k\}$:

$$\begin{aligned} \frac{dP_s}{dt} &= \beta P_s + (1 - P_s)P_{max}\sum_k \xi_k \delta(t - t_k), \\ \xi_k &= \begin{cases} 1 & \text{w.p. } P_{rel} \\ 0 & \text{w.p. } 1 - P_{rel}. \end{cases} \end{aligned} \tag{1.89}$$

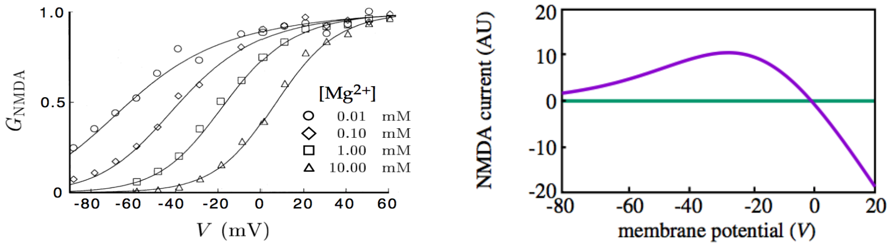


Figure 1.16: NMDA channel conductance (left) and current (right) as a function of voltage.

Another way to view this is having the decay $\tau_s \frac{dP_s}{dt} = -P_s$, with

$$P_s \rightarrow P_s + P_{max}(1 - P_s) \tag{1.90}$$

immediately after each action potential. This produces a sawtooth-like pattern of activation of the channel.

Two other useful formulations for the rise and fall of synaptic conductances are via differences of exponentials and a alpha functions.

1.8.2 NMDA-Mediated Plasticity

The NMDA receptor conductance has an additional and unusual dependence on the post-synaptic potential V. The NMDA current can be written as

$$I_{NMDA} = -G_{NMDA}(V)P_{NMDA}(V - \mathcal{E}_{NMDA}) = -\frac{P_{NMDA}}{1 + \frac{[\mathrm{Mg}^{2+}]}{3.57\ \mathrm{mM}} \exp\left(-V/16.1\ \mathrm{mV}\right)}(V - \mathcal{E}_{NMDA}), \tag{1.91}$$

where

$$G_{NMDA}(V) = \frac{1}{1 + \frac{[\mathrm{Mg}^{2+}]}{3.57\ \mathrm{mM}} \exp\left(-V/16.1\ \mathrm{mV}\right)}. \tag{1.92}$$

P_{NMDA} is the standard channel activation probability. The current and the conductance can be visualized in Figure 1.16. The extra voltage dependence is due to the fact that when the postsynaptic neuron is near its resting potential, NMDA receptors are blocked by Mg^{2+} ions. To activate the conductance, the postsynaptic neuron must be depolarized to knock out the blocking ions. Note that without Mg^{2+}, i.e., $[\mathrm{Mg}^{2+}] = 0$, I_{NMDA} will grow without bound as the membrane potential hyperpolarizes—Mg^{2+} deficiencies can cause seizures. NMDA channels also conduct Ca^{2+} ions, which are key to long-term modification of synaptic strength. They signal the cell to both open more NMDA channels and build more AMPA channels. Because NMDA channel activation requires both pre- and post-synaptic depolarization, NMDA channels can act as coincidence detectors for simultaneous pre-synaptic and post-synaptic activity. This plays an important role in models of plasticity such as the Hebb rule. NMDA-mediated channels are a key factor in long-term plasticity.

1.8.3 Short-Term Plasticity

The history of activity at a synapse can affect both the pre-synaptic release probability and changes conductance at the post-synaptic neuron. *Short-term plasticity* refers to a number of factors that can affect the probability that a pre-synaptic action potential opens post-synaptic channels, and last on the order of $\sim$ 1-10 ms. The effects of *long-term plasticity* can last indefinitely. A simple operational definition of short-term plasticity is as a modification in the release probability P_{rel} at the pre-synaptic neuron. Over the short time-scales that short-term plasticity operates, two phenomena can occur (copied from Jorge's notes):

- *synaptic depression*: post-synaptic potential temporarily decreases with repeated high frequency pre-synaptic spikes, since the stock of readily available neurotransmitter in the presynaptic axon terminal has been depleted, thus lowering the probability of vesicle release on the next spike.

- *synaptic facilitation*: post-synaptic potential temporarily increases with repeated high frequency pre-synaptic spikes, since this leads to a high influx of calcium Ca^{2+} ions into the pre-synaptic axon terminal, thus increasing the probability of vesicle release on the next spike

Example post-synaptic voltage traces for depression and facilitation are plotted in Figure 1.17 Both facilitation and depression can be modeled as pre-synaptic processes that modify the probability of transmitter release.

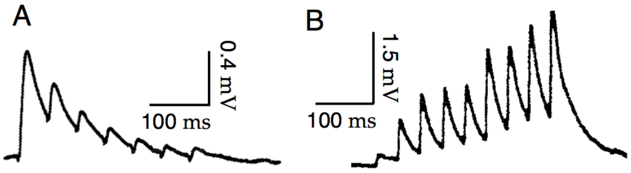


Figure 1.17: Post-synaptic voltage traces for short-term depression (A) and facilitation (B).

After a long period without pre-synaptic action potentials, P_{rel} resets to a baseline P_0 for both facilitation and depression. In periods without activity, the release probability decays exponentially back to its resting value, and at spike arrival times, P_{rel} increases in the case of facilitation and decreases in the case of depression. The dynamics can be summarized as

$$\tau_{rel}\frac{dP_{rel}}{dt} = -(P_{rel} - P_0) - P_{rel}(1 - f_D)\sum_k \delta(t - t_k)\xi_k + (1 - P_{rel})f_F \sum_k \delta(t - t_k). \tag{1.93}$$

In general, it's useful to keep in mind that the average firing rate can be expressed as $\nu = \langle \sum_k \delta(t - t_k)\rangle$. It is simpler, though, to work with this in the form of update rules, where the explicit dynamics are just to decay back to the resting state, and P_{rel} is updated upon the arrival of a pre-synaptic spike:

$$\tau_P\frac{dP_{rel}}{dt} = -(P_{rel} - P_0) \tag{1.94}$$

$$P_{rel} \to \xi_k f_D P_{rel} + (1 - \xi_k)P_{rel} = f_D P_{rel} \qquad \text{(depression)} \tag{1.95}$$

$$P_{rel} \to P_{rel} + f_F(1 - P_{rel}) \qquad \text{(facilitation)}, \tag{1.96}$$

where we set $\xi_k = 1$ to model 100% probability of vesicle release and $f_F, f_D \in [0, 1]$ control the degree of facilitation (higher f_F → stronger facilitation) and depression (smaller f_D → stronger depression). Note that depression depends on ξ_k (aka vesicle release) because it occurs when the pre-synaptic neuron effectively runs out of neurotransmitter—therefore, depression only occurs if neurotransmitter is actually released. In contrast, facilitation occurs because spikes arriving at the pre-synaptic terminal cause repeated influxes of Ca^{2+}, which makes the pre-synaptic neuron more likely to fire again.

Short-term depression can be helpful for normalizing synaptic inputs and detecting changes in firing rate. Consider the average steady-state release probability $\langle P_{rel}\rangle$ (P_{rel} averaged over pre-synaptic spikes drawn from a homogeneous Poisson process with rate r). When we say that $\langle P_{rel}\rangle$ is the average *steady-state* release probability, we define it to mean that the facilitation that occurs after each pre-synaptic action potential is exactly canceled by the average exponential decrease that occurs between spikes. Suppose that the release probability is at its average steady-state value when a spike arrives at time t_k—$P_{rel}(t_k) = \langle P_{rel}\rangle$—and depression occurs:

$$P_{rel} \to f_D\langle P_{rel}\rangle. \tag{1.97}$$

We can solve the resulting ODE for the release probability when the next spike arrives at time t_{k+1}:

$$\begin{aligned} \tau_{rel}\frac{dP}{dt} &= -(P_{rel}(t_k) - P_0)f_D \\ \Rightarrow [P_{rel}(t_{k+1}) - P_0]f_D &= [P_{rel}(t_k) - P_0]f_D e^{-\frac{t_{k+1}-t_k}{\tau_{rel}}} \\ \Rightarrow P_{rel}(t_{k+1}) &= P_{rel}(t_k)e^{-\frac{t_{k+1}-t_k}{\tau_{rel}}} + P_0\left(1 - e^{-\frac{t_{k+1}-t_k}{\tau_{rel}}}\right) \\ \Rightarrow P_{rel}(t_{k+1}) &= P_0 + [P_{rel}(t_k) - P_0]e^{-\frac{t_{k+1}-t_k}{\tau_{rel}}}. \end{aligned} \tag{1.98}$$

Taking the expectation of both sides (and letting $\Delta t := t_{k+1} - t_k$) gives

$$\langle P_{rel}(t_{k+1})\rangle = P_0 + (f_D\langle P_{rel}\rangle - P_0)\left\langle e^{-\Delta t/\tau_{rel}}\right\rangle. \tag{1.99}$$

Because we are averaging over events drawn from a homogeneous Poisson process, the inter-spike interval is

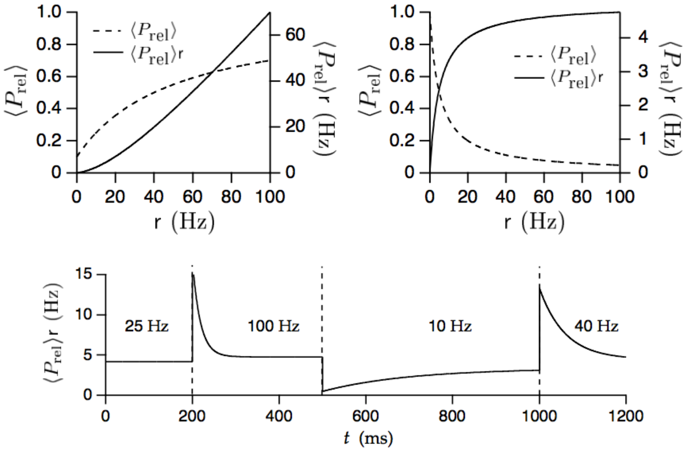


Figure 1.18: (Top) Visualization of the normalization effect of synaptic depression—as the firing rate r increases, $\langle P_{rel}\rangle$ drops proportionally. (Bottom) Depiction of the effect of transient increases in firing rate.

distributed according to an exponential distribution. We can then write

$$\begin{aligned} \left\langle e^{-\Delta t/\tau_{rel}}\right\rangle &= \int_0^\infty P(\Delta t)e^{-\Delta t/\tau_{rel}}\,dt \\ &= \int_0^\infty re^{-r\Delta t}e^{-\Delta t/\tau_{rel}}\,d\Delta t \\ &= r\int_0^\infty \exp\left(-\Delta t\frac{r\tau_{rel}+1}{\tau_{rel}}\right)\,d\Delta t \\ &= \frac{r\tau_{rel}}{r\tau_{rel}+1}. \end{aligned} \tag{1.100}$$

We then have

$$\langle P_{rel}(t_{k+1})\rangle = P_0 + (f_D\langle P_{rel}\rangle - P_0)\frac{r\tau_{rel}}{r\tau_{rel}+1}. \tag{1.101}$$

However, on average, we expect $\langle P_{rel}(t_{k+1})\rangle = \langle P_{rel}\rangle$. Substituting this in, we can solve for the average steady-state $\langle P_{rel}\rangle$:

$$\boxed{\langle P_{rel}\rangle = \frac{P_0\left(1-\frac{r\tau_{rel}}{r\tau_{rel}+1}\right)}{1-f_D\frac{r\tau_{rel}}{r\tau_{rel}+1}} = \frac{P_0}{(1-f_D)r\tau_{rel}+1}.} \tag{1.102}$$

We can then see that at high pre-synaptic firing rates r, the release probability is low: $\boxed{\langle P_{rel}\rangle \propto 1/r}$. Therefore, the rate at which post-synaptic potentials arrive, given by rP_{rel}, stays roughly constant with respect to the pre-synaptic firing rate at steady-state (see the top panels of Figure 1.18). In this way, synaptic depression acts to normalize pre-synaptic inputs across synapses to the same transmission rate (and thus the same time-averaged post-synaptic potential amplitude).

This normalization then also means that such a synapse cannot convey any information about smooth changes (i.e., on a comparable time-scale to τ_{rel}) in the pre-synaptic firing rate–they must be abrupt/transient. Given a transient increase in firing rate $r \to r+\Delta r$, before reaching steady-state (which takes time $\mathcal{O}(\tau_{rel})$), the synaptic transmission rate will briefly rise to

$$(r+\Delta r)\langle P_{rel}\rangle = \frac{(r+)P_0}{(1-f_D)r\tau_{rel}+1} \tag{1.103}$$

before exponentially decaying to steady-state. This is can be seen in the bottom panel of Figure 1.18. As firing rates grow large, this is approximately $\frac{r+\Delta r}{r}$, and therefore the increase in post-synaptic transmission rate is roughly proportional to the *relative*, not absolute, change in pre-synaptic firing rates. Synaptic depression therefore can encode the relative magnitude of transient changes in pre-synaptic firing rate.

2 Models of Synaptic Plasticity

Beyond NMDA-mediated plasticity, long-term changes in synaptic strength are not well understood and are thus modeled at a greater level of physical generality. Rather than consider neurotransmitters and conductances, functional models of synaptic plasticity directly model the change in the strength, or *weight* W_{ij}, of a synapse from pre-synaptic neuron i to post-synaptic neuron j, which can be roughly approximated by

$$W_{ij} = \bar{g}_{ij} P_{rel}^{(ij)}. \tag{2.1}$$

Changes in W_{ij} are generally modeled as a function of the pre- and postynaptic firing rates r_i, r_j:

$$\tau_w \frac{dW_{ij}}{dt} = f_{ij}(r_i, r_j), \tag{2.2}$$

where the time constant τ_w sets the effective learning rate for the synapse (high τ_w → low learning rate, and vice versa).

An experimental proxy for the value of W_{ij} is the post-synaptic change in membrane potential induced by a pre-synaptic spike. This is called the *post-synaptic potential* (PSP) amplitude. PSPs can be excitatory (E) or inhibitory (I)—note, however, that they are *not* action potentials, but graded potentials. They can sum temporally (repeatedly via the same synapse) or spatially (via multiple synapses) at the post-synaptic soma. Visually, this looks like Figure 1.17B for EPSPs (and inverted across the x-axis for IPSPs).

Experimentally, it's possible to foster *long-term potentiation* (LTP) in a synapse by inducing high-frequency ($\sim$ 100 Hz) bursts of action potentials simultaneously in the pre-synaptic and post-synaptic neurons for a period of hours. More specifically, LTP refers to a long-term increase in synaptic strength, where "long-term" is defined as at least tens of minutes—though changes can persist indefinitely. Similarly, *long-term* depression (LTD) can be induced via a low-frequency ($\sim$ 2 Hz) bursting protocol. In general, LTP occurs when a high pre-synaptic firing rate is accompanied by high post-synaptic firing rates, and LTD occurs when high pre-synaptic firing rates are accompanied by low post-synaptic firing rates. This pattern hints at the famous *Hebb rule*: neurons that fire together, wire together.

Unconstrained, this principle quickly leads to computational challenges. First, a naïve Hebbian learning rule can quickly lead to uncontrolled growth of synaptic strengths. This can easily be addressed by setting a maximum allowed weight value, w_{max}. Weights should also not be allowed to change sign, as synapses cannot change from excitatory to inhibitory. Thus, excitatory weights are limited to the range $[0, w_{max}]$, while inihibitory weights may be limited to the range $[-w_{max}, 0]$, for example. This limiting principle is called *synaptic saturation*. Second, since synapses are modified independently under a naïve Hebb rule, there's nothing to stop them all from reaching the same limiting value w_{max}, causing the neuron to lose selectivity to different stimuli. This can be addressed by introducing *synaptic competition*, such that there is in essence a limited amount of total synaptic strength for which the synapses of a neuron—or a network—must compete.

To start, we'll consider a single post-synaptic neuron with linear dynamics:

$$\tau \frac{dv}{dt} = -v + \mathbf{w}^\mathsf{T}\mathbf{u}, \tag{2.3}$$

where v is the firing rate of the post-synaptic neuron and $\mathbf{u}$ is the vector of pre-synaptic inputs. Because the processes of synaptic plasticity are typically much lower than the dynamics characterized by eq. 2.3, if the stimuli are presented slowly enough, we can replace eq. 2.3 with its steady-state value:

$$v = \mathbf{w}^\mathsf{T}\mathbf{u}. \tag{2.4}$$

Long-term synaptic modification is included in this model by specifying how $\mathbf{w}$ changes as a function of pre- and post-synaptic activity.

2.1 The Hebb Rule

The simplest formulation of the Hebb rule can be written as

$$\tau_w \frac{d\mathbf{w}}{dt} = v\,\mathbf{u}, \tag{2.5}$$

which simply implies that simultaneous pre-synaptic and post-synaptic activity increases synaptic strength. Synaptic weight changes occur slowly, with the total change over a period of time being the sum of the changes induced by each presented input pattern $\mathbf{u}$. If the weights change slowly enough, this total change can simple be computed by calculating the average input pattern during the given time period, and estimating the resulting adaptation to this average. Averaging over the presented inputs gives the averaged Hebb rule:

$$\tau_w \frac{d\,\mathbf{w}}{dt} = \langle v\,\mathbf{u}\rangle. \tag{2.6}$$

In unsupervised learning, v is given by eq. 2.4, and plugging this in gives a correlation-based rule:

$$\tau_w \frac{d\,\mathbf{w}}{dt} = \left\langle \mathbf{w}^\mathsf{T}\,\mathbf{u}\,\mathbf{u}\right\rangle = \left\langle \mathbf{u}\,\mathbf{u}^\mathsf{T}\right\rangle \mathbf{w} = Q\,\mathbf{w}, \tag{2.7}$$

where $Q = \left\langle \mathbf{u}\,\mathbf{u}^\mathsf{T}\right\rangle$ is the input correlation matrix.

Regardless of whether the activity variables are restricted to non-negative values, the basic Hebb rule is unstable. To see this, consider the square of the length of the weight vector $|\,\mathbf{w}\,|^2 = \mathbf{w}^\mathsf{T}\,\mathbf{w}$. We have

$$\begin{aligned}\tau_w \frac{d|\,\mathbf{w}\,|^2}{dt} &= 2\,\mathbf{w}^\mathsf{T}\,\frac{d\,\mathbf{w}}{dt} = 2\,\mathbf{w}^\mathsf{T}\,v\,\mathbf{u} && (2.8)\\ &= 2v\,\mathbf{w}^\mathsf{T}\,\mathbf{u} = 2\tau_w v^2 > 0, && (2.9)\end{aligned}$$

where we plug in eqs. 2.5 and 2.4. We can then see that the length of the weight vector is always increasing, leading to unbounded growth. Therefore, an upper saturation bound must be added (as well as a lower bound, if activities are allowed to be negative) to prevent weight explosion. This still fails to account for synaptic competition, however. In the discrete-time case, we can replace eq. 2.7 with an update rule

$$\mathbf{w} \leftarrow \mathbf{w} + \epsilon Q\,\mathbf{w}, \tag{2.10}$$

where $\epsilon := 1/\tau_w$ is the learning rate.

2.2 The Covariance Rule

If the activity variables $\mathbf{u}$ and v are interpreted as firing rates, they must be non-negative, and thus the basic Hebb rule as described above can only lead to LTP. The relationship between LTP and LTD and pre- and post-synaptic firing rates can be better modeled via the following plasticity rule:

$$\tau_w \frac{d\,\mathbf{w}}{dt} = (v - \theta_v)\,\mathbf{u}, \tag{2.11}$$

where θ_v is a threshold that determines the level of post-synaptic activity above which LTD switches to LTP. Such thresholding can instead be applied to the pre-synaptic activity, via

$$\tau_w \frac{d\,\mathbf{w}}{dt} = v(\mathbf{u} - \boldsymbol{\theta}_u), \tag{2.12}$$

where here $\boldsymbol{\theta}_u$ is a vector of thresholds, above which LTD switches to LTP. These two rules can also be combined by thresholding both pre- and post-synaptic activities, but this results in LTP when both pre- and post-synaptic firing rates are low, which is not found experimentally.

A useful setting for the thresholds is the average of the corresponding variable over the training period–that is, $\theta_v = \langle v\rangle$ or $\boldsymbol{\theta}_u = \langle \mathbf{u}\rangle$. Combining this with $v = \mathbf{w}^\mathsf{T}\,\mathbf{u}$ and averaging, we get

$$\begin{aligned}\tau_w \frac{d\,\mathbf{w}}{dt} &= \langle v(\mathbf{u} - \boldsymbol{\theta}_u)\rangle = \langle v(\mathbf{u} - \langle\mathbf{u}\rangle)\rangle\\ &= \left\langle \mathbf{w}^\mathsf{T}\,\mathbf{u}(\mathbf{u} - \langle\mathbf{u}\rangle)\right\rangle\\ &= \left\langle (\mathbf{u} - \langle\mathbf{u}\rangle)^\mathsf{T}\,\mathbf{u}\right\rangle \mathbf{w}\\ &= C\,\mathbf{w},\end{aligned} \tag{2.13}$$

where

$$C = \left\langle (\mathbf{u} - \langle\mathbf{u}\rangle)^\mathsf{T}\,\mathbf{u}\right\rangle = (\mathbf{u} - \langle\mathbf{u}\rangle)(\mathbf{u} - \langle\mathbf{u}\rangle)^\mathsf{T} = \left\langle \mathbf{u}\,\mathbf{u}^\mathsf{T}\right\rangle - \langle\mathbf{u}\rangle^2 \tag{2.14}$$

is the input covariance matrix. Applying the same process to the post-synaptic thresholding model similarly produces the analogous dynamics with the output covariance matrix.

Although they both average to give eq. 2.13 the rules in eqs. 2.11 and 2.12 result in different effects. Eq. 2.11 modifies synapses only if they have nonzero presynaptic activities. When $v < \theta_v$, this results in what's termed *homosynaptic depression* (occuring only if $u_i > 0$ for some i). In contrast, eq. 2.12 reduces the strengths of inactive synapses if $v > 0$, even if $u_i = 0$ for some input i. This is called *heterosynaptic depression.* Note that setting $\theta_v = \langle v \rangle$ in eq. 2.11 necessitates updating θ_v as the weights are modified. In contrast, the threshold in eq. 2.12 is independent of the weights and therfore does not need to be changed during training to keep $\boldsymbol{\theta}_u = \langle \mathbf{u} \rangle$.

Even though covariance rules include LTD and thus allow weights to decrease, they are still unstable because of the same positive feedback that makes the basic Hebb rule unstable. For either post-synaptic or pre-synaptic thresholding, we get the same result

$$\tau_w \frac{d|\mathbf{w}|^2}{dt} = 2\,\mathbf{w}^\mathsf{T} \frac{d\,\mathbf{w}}{dt} = 2\,\mathbf{w}^\mathsf{T}(v - \langle v \rangle)\,\mathbf{u} \tag{2.15}$$

$$= 2v(v - \langle v \rangle) \tag{2.16}$$

Averaging over time gives the average update

$$\tau_w \left\langle \frac{d|\mathbf{w}|^2}{dt} \right\rangle \propto \langle v(v - \langle v \rangle)\rangle = \mathbb{V}[v] \geq 0, \tag{2.17}$$

where the variance is zero only in the trivial case when the post-synaptic firing rate is constant, and thus the weights still explode.

Because the dynamics are linear, it's possible to easily analyze the effects of applying these learning rules. Because C is symmetric, its eigenvectors $\mathbf{e}_i$ are orthogonal and form a complete basis for the space of $\mathbf{w}$, allowing us to write

$$\mathbf{w}(t) = \sum_i c_i(t)\mathbf{e}_i, \tag{2.18}$$

where the coefficients c_i are simply equal to the scalar projection of $\mathbf{w}$ onto each eigenvector, given by $c_i(t) = \mathbf{w}^\mathsf{T}\,\mathbf{e}_i$ (assuming the eigenvectors are unit length). Solving the covariance rule ODE in eq. 2.13 gives

$$\mathbf{w}(t) = \sum_i c_i(0)e^{-\lambda_i t/\tau_w}\mathbf{e}_i, \tag{2.19}$$

where λ_i is the eigenvalue corresponding to the ith eigenvector. Then as $t \to \infty$, the eigenvector with the largest eigenvalue λ_1 dominates, resulting in

$$\lim_{t\to\infty} \mathbf{w}(t) \propto \mathbf{e}_1, \tag{2.20}$$

as long as $\mathbf{w}(0)$ is not perpendicular to $\mathbf{e}_1$. The post-synaptic activity v then evolves according to the principle eigenvector of the input covariance matrix:

$$v = \mathbf{w}^\mathsf{T}\,\mathbf{u} \propto \mathbf{e}_1^\mathsf{T}\,\mathbf{u}\,. \tag{2.21}$$

Similar analysis holds for the correlation-based Hebb rule without thresholds, with the same result when $Q = C$ (that is, when the inputs have mean zero). Also similar to the case of the basic Hebb rule is the fact that the covariance rules are noncompetitive, but competition can be introduced by allowing the thresholds to slide, as described below.

2.3 The BCM Rule

As described above, eq. 2.11 does not require any post-synaptic activity to produce LTD, and eq. 2.12 can produce LTD without any pre-synaptic activity. In contrast, the BCM rule requires both pre- and post-synaptic activity to change a synaptic weight. It takes the form

$$\tau_w \frac{d\,\mathbf{w}}{dt} = v\,\mathbf{u}(v - \theta_v). \tag{2.22}$$

If the post-synaptic threshold θ_v is held fixed, then the BCM rule, like the previous rules considered, is also unstable. If it's allowed to change, however, than this instability can be avoided. Specifically, θ_v must grow faster than v as v grows large. In one variant of the BCM rule, θ_v obeys the following dynamics:

$$\tau_\theta \frac{d\theta_b}{dt} = -(\theta_v - v^2), \tag{2.23}$$

such that θ_v adapts toward a low-pass filtered version of v, with $\tau_\theta < \tau_w$. Because when $|\,\mathbf{w}\,|$ increases, v increases, the threshold θ_v for LTP will then quickly rise, making it more difficult to increase $|\,\mathbf{w}\,|$. Interestingly, this means that if even one w_i grows large, the threshold will rise, making it harder for the other $w_{j\neq i}$ to grow. This effectively implements a form of competition between weights.

2.4 Synaptic Normalization

The BCM rule effectively implements weight saturation and competition by using the post-synaptic activity v as a proxy for the magnitude of the weights. However, the strength of the weights can also be constrained directly via the L_p norm: $\| \mathbf{w} \|_p := (\sum_i w_i^p)^{1/p}$. Such a constraint is called *synaptic normalization.* Constraining the L_1 norm corresponds to *subtractive normalization*, while using the L_2 norm results in *multiplicative normalization.*

2.4.1 Subtractive Normalization

The learning rule that limits the L_1 norm of $\mathbf{w} \in \mathbb{R}^K$ is given by

$$\tau_w \frac{d\mathbf{w}}{dt} = v(\mathbf{u} - \bar{u}\mathbf{1}), \tag{2.24}$$

where $\mathbf{1}$ is a vector of ones and

$$\bar{u} = \frac{1}{N_u} \sum_{k=1}^{N_u} u_k = \frac{\mathbf{1}^\mathsf{T} \mathbf{u}}{N_u} = \frac{1}{N_u} \|u\|_1 \tag{2.25}$$

is the average L_1 norm (sum) of the input. We can easily see that this rule constrains the sum of the weights:

$$\begin{aligned} \frac{d\| \mathbf{w} \|_1}{dt} &= \frac{d}{dt} \sum_i w_i \\ &= \sum_i \frac{dw_i}{dt} = v \sum_i u_i - vN_u\bar{u} \\ &= vN_u\bar{u} - vN_u\bar{u} = 0. \end{aligned} \tag{2.26}$$

This rule is therefore termed *subtractive* because the same value $vN_u\bar{u}$ is subtracted from the derivatives of each weight.

To better understand the dynamics, we can consider the rule in the expectation over inputs, and plugging in $v = \mathbf{w}^\mathsf{T} \mathbf{u}$ as usual:

$$\begin{aligned} \tau_w \left\langle \frac{d\mathbf{w}}{dt} \right\rangle &= v(\mathbf{u} - \bar{u}\mathbf{1}) = \langle v\, \mathbf{u} \rangle - \langle \bar{u} v \rangle \mathbf{1} \\ &= \langle \mathbf{u}\, \mathbf{u}^\mathsf{T} \rangle\, \mathbf{w} - \frac{1}{N_u} \mathbf{1}^\mathsf{T} \langle \mathbf{u}\, \mathbf{u}^\mathsf{T} \rangle\, \mathbf{w}\, \mathbf{1} \\ &= Q\, \mathbf{w} - \frac{1}{N_u} \mathbf{1}^\mathsf{T} Q\, \mathbf{w}\, \mathbf{1}. \end{aligned} \tag{2.27}$$

Because Q is symmetric, we can performn an eigendecomposition and write $\mathbf{w}(t) = \sum_i c_i(t)\mathbf{e}_i$, where $c_i = \mathbf{w}^\mathsf{T} \mathbf{e}_i$, we get

$$\tau_w \left\langle \frac{d\mathbf{w}}{dt} \right\rangle = \sum_{i=1}^{N_u} \lambda_i c_i(t)\mathbf{e}_i - \frac{1}{N_u} \lambda_i c_i(t) \mathbf{1}^\mathsf{T} \mathbf{e}_i \mathbf{1}. \tag{2.28}$$

Recalling that the orthogonality of the eigenvectors of Q gives $\mathbf{e}_i^\mathsf{T} \mathbf{e}_j = \delta_{ij}$, we can write a differential equation for $c_j(t) = \mathbf{e}_j^\mathsf{T}\, \mathbf{w}(t)$:

$$\begin{aligned} \tau_w \frac{dc_j}{dt} &= \tau_w \mathbf{e}_j^\mathsf{T} \frac{d\mathbf{w}}{dt} \\ &= \mathbf{e}_j^\mathsf{T} \left(\sum_{i=1}^{N_u} \lambda_i c_i(t)\mathbf{e}_i - \frac{1}{N_u} \lambda_i c_i(t) \mathbf{1}^\mathsf{T} \mathbf{e}_i \mathbf{1} \right) \\ &= \lambda_j c_j(t) - \frac{1}{N_u} \sum_{i=1}^{N_u} \lambda_i c_i(t) \mathbf{e}_i (\underbrace{\mathbf{e}_j^\mathsf{T} \mathbf{1}}_{=|\mathbf{e}_j||\mathbf{1}| \cos\theta_j = \sqrt{N_u} \cos\theta_j}) \\ &= \lambda_j c_j(t) - \frac{1}{\sqrt{N_u}} \sum_{i=1}^{N_u} \lambda_i c_i(t) \mathbf{e}_i \cos\theta_j, \end{aligned} \tag{2.29}$$

where θ_j is the angle between $\mathbf{e}_j$ and $\mathbf{1}$. Subtractive normalization then only updates directions of $\mathbf{w}$ close to the identity line $\mathbf{1}$, i.e., directions $\mathbf{e}_j$ in which all the weights grow at around the same rate. If $\mathbf{e}_j$ is perpendicular to $\mathbf{1}$, then $\tau_w \frac{dc_j}{dt} = \lambda_j c_j(t)$, resulting in standard Hebbian dynamics with exponential growth. It can also be easily shown that if the principal eigenvector $\mathbf{e}_1 \propto \mathbf{1}$ (i.e., $\mathbf{e}_1^\mathsf{T}\mathbf{1} = \delta_{1j}\sqrt{N_u}$), then in the limit $t \to \infty$, $\mathbf{w}$ actually converges to the direction of the eigenvector with the *second* highest eigenvalue. This can explain the development of ocular dominance in the eye.

There are some drawbacks of subtractive normalization. First, the use of the global subtractive signal $\bar{u}$ isn't very biologically plausible, as it requires that each synapse knows the inputs to every other synapse. Second, the competition between weights can be *too* strong, as the global subtractive term is relatively larger for weights with smaller derivatives. Without a lower bound on weights, this can drive the weight values arbitrarily negative. With a lower bound at zero, subtractive normalization often produces solutions with one large positive weight and the rest close to zero.

2.4.2 Oja's Rule: Multiplicative Normalization

The synaptic learning rule (aka *Oja's rule*) that constrains the L_2 norm of the weight vector is given by

$$\tau_w \frac{d\mathbf{w}}{dt} = v\,\mathbf{u} - \alpha v^2\,\mathbf{w}, \tag{2.30}$$

where $\alpha > 0$ bounds the L_2 norm of $\mathbf{w}$ (again using $v = \mathbf{w}^\mathsf{T}\,\mathbf{u}$:

$$\begin{aligned} \tau_w \frac{d\|w\|_2}{dt} &= 2\,\mathbf{w}^\mathsf{T}\,\frac{d\mathbf{w}}{dt} = 2\,\mathbf{w}^\mathsf{T}(v\,\mathbf{u} - \alpha v^2\,\mathbf{w}) \\ &= 2v\,\mathbf{w}^\mathsf{T}\,\mathbf{u} - 2\alpha v^2\,\mathbf{w}^\mathsf{T}\,\mathbf{w} = 2v^2 - 2\alpha v^2 \|\,\mathbf{w}\,\|_2^2 \\ &= 2v^2(1 - \alpha\|\,\mathbf{w}\,\|_2^2), \end{aligned} \tag{2.31}$$

which converges to $\|\,\mathbf{w}\,\| = 1/\alpha$. Because the normalization term $\alpha v^2\,\mathbf{w}$ is proportional to the weight, this is called *multiplicative* normalization. We can then analyze the average weight change, with $v = \mathbf{w}^\mathsf{T}\,\mathbf{u}$:

$$\begin{aligned} \tau_w \left\langle \frac{d\mathbf{w}}{dt} \right\rangle &= \left\langle v\,\mathbf{u} - \alpha v^2\,\mathbf{w} \right\rangle \\ &= \left\langle \mathbf{w}^\mathsf{T}\,\mathbf{u}\,\mathbf{u}^\mathsf{T} - \alpha\,\mathbf{w}^\mathsf{T}\,\mathbf{u}\,\mathbf{u}^\mathsf{T}\,\mathbf{w}\,\mathbf{w} \right\rangle \\ &= \mathbf{w}^\mathsf{T}\left\langle \mathbf{u}\,\mathbf{u}^\mathsf{T} \right\rangle - \alpha\,\mathbf{w}^\mathsf{T}\left\langle \mathbf{u}\,\mathbf{u}^\mathsf{T} \right\rangle \mathbf{w}\,\mathbf{w} \\ &= Q\,\mathbf{w} - \alpha\,\mathbf{w}^\mathsf{T}\,Q\,\mathbf{w}\,\mathbf{w}, \end{aligned} \tag{2.32}$$

where, as usual, $Q = \left\langle \mathbf{u}\,\mathbf{u}^\mathsf{T} \right\rangle$ is the correlation matrix. (If the inputs are zero-centered, this is also the covariance matrix.) At convergence, $\left\langle \frac{d\mathbf{w}}{dt} \right\rangle = 0$, so we have

$$\begin{aligned} Q\,\mathbf{w} &= \underbrace{\alpha\,\mathbf{w}^\mathsf{T}\,Q\,\mathbf{w}}_{\lambda}\,\mathbf{w} \\ Q\,\mathbf{w} &= \lambda\,\mathbf{w}, \end{aligned} \tag{2.33}$$

so we can see directly that $\mathbf{w}$ converges to an eigenvector of the input correlation matrix. In particular, it's easy (maybe) to show that it converges to the *principal* eigenvector–that is, the one whose eigenvalue is largest. If there are multiple eigenvectors with the largest eigenvalue, then $\mathbf{w}$ converges to some linear combination of these eigenvectors, with weightings determined by the initial conditions. Because Oja's rule only requires local information to compute its updates, it's more biologically plausible than subtractive normalization.

2.5 Spike-Timing Dependent Plasticity (STDP)

Another experimentally-inspired model for synaptic plasticity is *spike-timing dependent plasticity* (STDP), in which weight updates only occur when pre- and post-synaptic spikes occur within a short time of one another (usually ~ 50 ms), with the effect decaying exponentially with the latency. If the pre-synaptic spike precedes the post-synaptic spike, LTP occurs, and LTD occurs if the post-synaptic spike happens first. This relationship is visualized in Figure 2.1. This model is easily implemented in spiking networks, but in continuous rate-based networks like those we have been considering, it can be approximated via a function $H(\tau)$, where $\tau := t_{post} - t_{pre}$. Thus, $\tau < 0$ corresponds to a post-synaptic spike preceding a pre-synaptic spike, and $\tau > 0$ represents the opposite. More specifically, $H(\tau)$ determines the rate of synaptic modification for an interval τ, with the total rate being found by integrating over all possible values of τ. Assuming that the rate of modification is proportional to the product of the pre- and post-synaptic firing rates (as in Hebbian learning), we can write

$$\tau_w \frac{d\mathbf{w}}{dt} = \int_0^\infty H(\tau)v(t)\,\mathbf{u}(t-\tau) + H(-\tau)v(t-\tau)\,\mathbf{u}(t)\,d\tau, \tag{2.34}$$

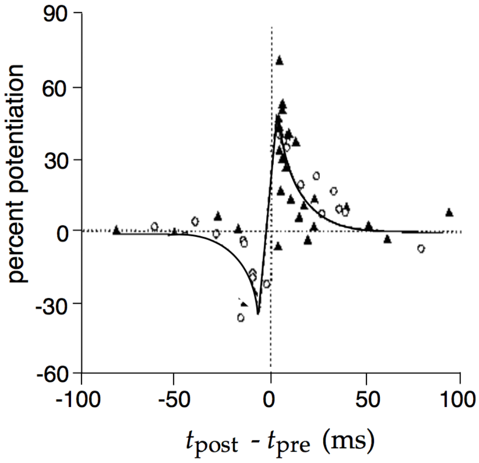


Figure 2.1: Plot of experimentally-observed $H(\tau)$ for timing-dependent potentiation and depression.

where $\text{sign}(H(\tau)) = \text{sign}(\tau)$ such that the first term in the integral corresponds to LTP and the second to LTD.

While the standard STDP rule is unstable (like the basic Hebb rule), it does induce competition among weights—an increase in w_i facilitates the ability of an increase in u_i to lead to an increase in v regardless of the other inputs $u_{j\neq i}$. This then can serve to increase $v(t-\tau)u_{j\neq i}(t)$ and inducing LTD at those synapses. This often results in a highly bimodal weight distribution.

Interestingly, the STDP rule can produce neural responses that are invariant to the spatial position of stimuli. The above differential equation can be approximately solved if one ignores the adaptation of v in response to the adaptation in $\mathbf{w}$:

$$\mathbf{w} = \frac{1}{\tau_w}\int_0^T v(t)\int_{-\infty}^{\infty} H(\tau)u(t-\tau)\,d\tau dt, \tag{2.35}$$

where it is assumed that $\mathbf{w}(0) = 0$ and ignored contributions from the endpoints of the integral. Thus, the final learned weight, $\mathbf{w}(T)$ depends on the temporal correlation between the post-synaptic activity $v(t)$ and the pre-synaptic activity $\mathbf{u}(t)$, temporally filtered by the STDP kernel $H(\tau)$.Consider now the scenario of u(t) arising from an object moving across the visual field. If the filter $H(\tau)$ filters the resulting sequence of inputs over the amount of time the object is present, then it will strengthen the synapses from all pre-synaptic cells responding to the object while it moves, regardless of its position. In the long run, the resulting weights will thus lead to post-synaptic responses independent of the position of the object, producing position-invariant responses to the object such as those seen in inferotemporal cortex (IT).

STDP can also produce predictive coding responses in a recurrent network with fixed feedforward weights. Consider a set of post-synaptic neurons with equally spaced homogeneous tuning curves (e.g. for orientation) and an input stimulus that traverses the stimulus space in the same direction on each presentation (e.g. a clockwise rotating bar). As the stimulus is repeated, the tuning curves will then gradually shift in the opposite direction, since each neuron's recurrent input from neurons selective for the previous stimulus state will be strengthened. On each subsequent presentation, then, a given post-synaptic neuron is more and more likely of firing earlier and earlier. In the long run, this will produce predictive responses anticipating subsequent input according to the input stimulus it was trained on. Such behavior is observed in hippocampal place cells (Dayan & Abbott pages 312-3). (Thank you Jorge.)

3 Networks

Consider the following simplified network model:

$$\tau_m \frac{dV_i}{dt} = f_j(V_i, t) - \sum_{j \neq i} m_{ij} g_j(t)(V_i - \mathcal{E}_j) \tag{3.1}$$

$$\tau_g \frac{dg_j}{dt} = -g_j + \sum_k \delta(t - t_k^{(j)}), \tag{3.2}$$

where $f_i(\cdot, \cdot)$ represents single-neuron dynamics (i.e., leak currents, HH currents) and m_{ij} is meant to encapsulate factors such as synaptic release probabilities and open channel conductances. Note that here (and in the analysis below) i indexes the post-synaptic neurons and j indexes the pre-synaptic neurons. The pre-synaptic conductance g_j is governed by simplified dynamics in which each spike causes an immediate jump in conductance, with the absence of spikes resulting in exponential decay. As a simplification, we define the *synaptic weight* W_{ij} as

$$W_{ij} := -m_{ij}(V_i - \mathcal{E}_j), \tag{3.3}$$

such that we can rewrite eq. 3.1 as

$$\tau_m \frac{dV_i}{dt} = f_i(V_i, t) + \sum_{j \neq i} W_{ij} g_j(t), \tag{3.4}$$

where we define the *synaptic drive* $h_i(t) := \sum_{j \neq i} W_{ij} g_j(t)$. Considering only the effect of one spike, occurring at t_k^j, we can solve eq. 3.2 to get

$$g_j^k(t) = \Theta(t - t_k^{(j)}) \frac{1}{\tau_g} e^{-\frac{t - t_k^{(j)}}{\tau_g}}. \tag{3.5}$$

The factor of $1/\tau_g$ is included here for convenience. We then have

$$\begin{aligned}
\int_0^\infty g_j^k(t)\, dt &= \int_0^\infty \Theta(t - t_k^{(j)}) \frac{1}{\tau_g} e^{-\frac{t - t_k^{(j)}}{\tau_g}}\, dt && (3.6)\\
&= \int_{t_k^{(j)}}^\infty \frac{1}{\tau_g} e^{-\frac{t - t_k^{(j)}}{\tau_g}}\, dt = \left[-e^{-\frac{t - t_k^{(j)}}{\tau_g}} \right]_{t_k^{(j)}}^\infty && (3.7)\\
&= 1. && (3.8)
\end{aligned}$$

The time-average conductance is then given by

$$\begin{aligned}
\langle g_j(t) \rangle_t &= \lim_{T \to \infty} \frac{1}{T} \sum_k \int_{t_k^{(j)}}^\infty g_j^k(t)\, dt && (3.9)\\
&= \lim_{T \to \infty} \frac{1}{T} \sum_k 1 && (3.10)\\
&= \lim_{T \to \infty} \frac{1}{T} n(T) && (3.11)\\
&:= \nu_j, && (3.12)
\end{aligned}$$

where $n(T)$ denotes the total number of spikes up to time T and ν_j is then the average firing rate of the pre-synaptic neurons. In order to analyze networks of spiking neurons, it's easier to perform statistical analysis of firing rates and connectivity than attempt to solve the high-dimensional system of equations suggested by a more precise biophysical model. We'd like to analyze the distribution of firing rates and pattern of connectivity as the number of neurons N grows large.

3.1 Networks that violate Dale's Law

3.1.1 Dense Connectivity

Here, we discuss networks of spiking neurons with fixed, random connectivity, specifically considering *balanced* networks; i.e., those whose dynamics are at an equilibrium. In this section, we don't concern ourselves with the requirement that each neuron can only be excitatory or inhibitory. We define the firing rate ν to be a nonlinear function of the synaptic drive h and any external input I. We consider *dense* networks, in which each neuron is connected to every other neuron. The dynamics are generally of the form

$$\frac{d\nu_i}{dt} = -\nu_i + \phi(h_i), \tag{3.13}$$

$$h_i = \sum_{j=1}^{N} W_{ij}\nu_j, \tag{3.14}$$

where the nonlinearity $\phi(\cdot)$ is generally sigmoidal.At equilibrium, then, we have

$$\nu_i = \phi(h_i + I_i). \tag{3.15}$$

Let $W_{ij} \sim \mathcal{N}(\langle W\rangle, \mathbb{V}[W])$. Then as the number of neurons $N \to \infty$, we say that $h_i \sim p(h)$, where $p(h) = \mathcal{N}(\mu_h, \sigma_h^2)$. To compute the mean μ_h, we have

$$\begin{aligned}
\mu_h &\approx \frac{1}{N}\sum_i h_i = \frac{1}{N}\sum_i\sum_j W_{ij}\nu_j && (3.16)\\
&= \sum_j \nu_j \frac{1}{N}\sum_i W_{ij} \approx \bar{W}\sum_j \nu_j && (3.17)\\
&= \boxed{N\langle\nu\rangle\langle W\rangle}, && (3.18)
\end{aligned}$$

where we note that the approximation in eq. 3.17 is due to the fact that we are only averaging over the rows of W. However, because all entries in W are drawn i.i.d. from the same distribution, the row-wise mean is a good approximation for the full mean. For σ_h^2, we have

$$\begin{aligned}
\sigma_h^2 &= \mathbb{V}_{p(h_i)}[h_i] = \mathbb{V}_i\left[\sum_j W_{ij}\nu_j\right] && (3.19)\\
&= \sum_j \mathbb{V}_i[W_ij\nu_j] = \sum_j \nu_j^2\,\mathbb{V}_i[W_{ij}] && (3.20)\\
&\approx \boxed{N\langle\nu^2\rangle\,\mathbb{V}[W]}. && (3.21)
\end{aligned}$$

The issue, then, is that we need $\langle\nu\rangle = \mathcal{O}(1)$ and $\langle\nu^2\rangle = \mathcal{O}(1)$, in addition to μ_h and σ_h^2, as otherwise the synaptic drive will diverge as $N \to \infty$. Note that the moments of the firing rate ν must be $\mathcal{O}(1)$ from eq. 3.15—the nonlinearity $\phi(\cdot)$ saturates for for high or low inputs, and is therefore bounded. Therefore, we need to introduce scaling to $\langle W\rangle$. If $\langle W\rangle = \mathcal{O}\left(\frac{1}{N}\right)$, we also require $\mathbb{V}[W] = \mathcal{O}\left(\frac{1}{N}\right)$, and so then

$$\mu_h \approx \sqrt{N}\langle\nu\rangle\langle W\rangle = \mathcal{O}(\sqrt{N})\mathcal{O}(1)\mathcal{O}\left(\frac{1}{\sqrt{N}}\right) = \mathcal{O}(1) \tag{3.22}$$

$$\sigma_h^2 \approx N\langle\nu^2\rangle\,\mathbb{V}[W] = \mathcal{O}(N)\mathcal{O}(1)\mathcal{O}\left(\frac{1}{N}\right) = \mathcal{O}(1), \tag{3.23}$$

as desired. Note that if we introduced scaling $W_{ij} \to \frac{1}{\sqrt{N}}W_{ij}$ in eq. 3.14, we would have $\mu_h = \sqrt{N}\langle\nu\rangle\langle W\rangle$. If we then scale $\langle W\rangle$ as $1/\sqrt{N}$, $\mathbb{V}[W] = \mathcal{O}(1/N)$ and we get the same result.

We now have expressions for the mean and variance of the synaptic drive h, but they're in terms of the first and second moments of the firing rates, $\langle\nu\rangle$ and $\langle\nu^2\rangle$, which we don't have. In general, the sample kth moment of the firing rate at equilibrium is given by

$$\begin{aligned}
\langle\nu^k\rangle &\coloneqq \frac{1}{N}\sum_i \nu_i^k && (3.24)\\
&= \frac{1}{N}\sum_i \phi^k(h_i + I_i) \quad \text{(by eq. 3.15)} && (3.25)\\
&= \int p(I)\int p(h)\phi^k(h+I)\,dh\,dI, && (3.26)
\end{aligned}$$

where in the last line we convert the sum to an integral over the distributions of I and h in the large N limit. Observe that we can express $h_i \sim \mathcal{N}(\mu_h, \sigma_h^2)$ as being generated by the process

$$h_i = \mu_h + \sigma_h\xi_i, \tag{3.27}$$

where $\xi_i \sim \mathcal{N}(0,1)$. We can then rewrite eq. 3.26 as

$$\langle \nu^k \rangle = \int p(I) \int p(\xi) \phi^k \left(\underbrace{N\langle\nu\rangle\langle W\rangle}_{\mu_h} + \underbrace{\sqrt{N\langle\nu^2\rangle\,\mathbb{V}[W]}}_{\sigma_h}\,\xi + I \right) d\xi\, dI, \tag{3.28}$$

with $p(\xi) = -\exp(\xi^2/2)/\sqrt{2\pi}$. If we simplify things and let the external input I be constant, this reduces to

$$\boxed{\langle \nu^k \rangle = \int p(\xi)\phi^k(\mu_h + \sigma_h \xi + I) d\xi}. \tag{3.29}$$

[TM: Need to clarify this next part.] Note that if we adopted a different scaling of the weights, such that

$$h_i = \frac{1}{N} \sum_j W_{ij}\nu_j, \tag{3.30}$$

we would have gotten $\mu_h = N\langle\nu\rangle\langle W\rangle$ and $\sigma_h^2 = N\langle nu^2\rangle\,\mathbb{V}[W] = \mathcal{O}(1/\sqrt{N})$. Then as $N \to \infty$, $\sigma_h^2 \to 0$. Applying this to eq. 3.29, we have

$$\begin{aligned} \langle \nu^k \rangle &= \int p(\xi)\phi^k(\mu_h + I) d\xi \\ &= \phi^k(\mu_h + I) \int p(\xi) d\xi \\ &= \phi^k(\mu_h + I). \end{aligned} \tag{3.31}$$

Note that this implies

$$\mathbb{V}[\nu] = \langle \nu^2 \rangle - \langle \nu \rangle^2 = \phi^2(\mu_h + I) - \phi^2(\mu_h + I) = 0, \tag{3.32}$$

so we have

$$p(\nu) \to \delta(\nu - \phi(\mu_h + I)) \tag{3.33}$$

as $N \to \infty$ under this scaling.

3.1.2 Sparse Connectivity

As we've seen, the above dense connectivity structure demands strong restrictions on the weights of the network in order to remain stable. In nature, however, connectivity rates are typically closer to 10%—in other words, real networks are generally *sparse*. We can incorporate this property into our model by introducing a parameter $K \ll N$ that determines the mean number of connections per neuron, such that the probability that any two neurons are connected is equal to $p := K/N$ (the connectivity rate). Under this model, we can derive updated estimates of the mean μ_h and variance σ_h^2 of the synaptic drive, letting

$$w_{ij} = \zeta_{ij}\tilde{w}_{ij}, \tag{3.34}$$

$$\zeta_{ij} = \begin{cases} 1 & \text{w.p. } K/N \\ 0 & \text{w.p. } 1 - K/N \end{cases}, \tag{3.35}$$

such that $\zeta_{ij} \sim \text{Bernoulli}(K/N)$ and $\tilde{w}_{ij} \sim^{iid} \mathcal{N}(\tilde{w}, \sigma_{\tilde{w}}^2)$. Then

$$\boxed{\mu_h = N\langle\nu\rangle\langle w\rangle = N\frac{K}{N}\tilde{w}\langle\nu\rangle = K\tilde{w}\langle\nu\rangle}, \tag{3.36}$$

and

$$\sigma_h^2 = \mathbb{V}[w]\langle\nu^2\rangle \tag{3.37}$$

$$= (\langle\zeta^2\rangle\langle\tilde{w}^2\rangle - \langle\zeta\rangle^2\langle\tilde{2}\rangle^2)\langle\nu^2\rangle \tag{3.38}$$

$$= \left(\frac{K}{N}(\sigma_{\tilde{w}}^2 + \tilde{w}^2) - \frac{K^2}{N^2}\tilde{w}^2\right)\langle\nu^2\rangle \tag{3.39}$$

$$= \boxed{\frac{K}{N}\left(\sigma_{\tilde{w}}^2 + \left(1 - \frac{K}{N}\right)\tilde{w}^2\right)\langle\nu^2\rangle}, \tag{3.40}$$

where the variance factorizes because ζ_{ij} and $\tilde{w}_{ij}$ are independent. We then have that that variance is proportional to the (on average) K non-zero connections plus a correction term for the absent connections (weighted by $1-p$). We can also see that, similar to the dense case, we need $\tilde{w} = \mathcal{O}(1/K)$ and $\sigma_{\tilde{w}}^2 = \mathcal{O}(1/K)$ to maintain viable firing rates. We can then apply eq. 3.29 for constant input I in the large N limit:

$$\boxed{\langle \nu^k \rangle = \int \phi^k \left(K\tilde{w}\langle\nu\rangle + \sqrt{p\,(\sigma_{\tilde{w}}^2 + (1-p)\,\tilde{w}^2)\,\langle \nu^2\rangle} + I \right) p(\xi)\, d\xi} \tag{3.41}$$

The main takeaway is that we can use Gaussian approximations in the large N (or K) limit to derive fairly accurate expressions for the moments of the firing rates of a random network. Note that for the Gaussian approximation to really be accurate, the firing rates would have to be independent, which of course they're not, but due to weak correlations, they're not too bad.

3.2 Wilson-Cowan Equations

We can essentially repeat the same analysis above for networks that have properly designated excitatory (E) and inhibitory (I) populations. There are then four categories of synaptic weights $w_{ij}^{\alpha\beta} > 0$ from α neurons to β neurons, where $\alpha, \beta \in \{E, I\}$. The synaptic drive to an α neuron is then given by

$$h_i^\alpha(t) = \sum_{j\in E} w_{ij}^{\alpha E} g_j^E(t) - \sum_{j\in I} w_{ij}^{\alpha I} g_j^E. \tag{3.42}$$

Repeating the same derivation for the sparse networks above, but with $w_{ij}^{\alpha\beta} = \frac{1}{\sqrt{K}}\zeta_{ij}^{\alpha\beta}\tilde{w}_{ij}^{\alpha\beta}$ and constant input I_α, we get

$$\mu_{\nu_\alpha} = \sqrt{K}(\langle w_{\alpha E}\rangle\langle \nu_E\rangle - \langle w_{\alpha E}\rangle\langle \nu_E\rangle) \tag{3.43}$$

$$\sigma^2 = \sigma_{\alpha E}^2 + \sigma_{\alpha I}^2, \quad \text{where } \sigma_{\alpha\beta}^2 = p\left(\sigma_{w_{\alpha\beta}}^2 + (1-p)\langle w_{\alpha\beta}\rangle^2\right)\langle \nu_\beta^2\rangle, \tag{3.44}$$

$$\langle \nu_\alpha^k \rangle = \int \phi^k \left(\mu_{\nu_\alpha} + \sigma\xi + I_\alpha\right) p(\xi) d\xi. \tag{3.45}$$

If $\tilde{w}^{\alpha E}\langle\nu_E\rangle - \tilde{w}^{\alpha I}\langle \nu_I\rangle \approx 0$ (specifically, if it's $\mathcal{O}(1/\sqrt{K})$), then with small average input, firing rates will remain within a stable firing regime. This is called the *balanced regime*. To understand the dynamics of the full network, we can write write differential equations for the evolution of the mean firing rates of each subpopulation:

$$\begin{aligned} \tau_E \dot{\nu_E} &= \psi_E(\bar{\nu}_E, \bar{\nu}_I) - \nu_E, \\ \tau_I \dot{\nu_I} &= \psi_I(\bar{\nu}_E, \bar{\nu}_I) - \nu_I, \end{aligned} \tag{3.46}$$

where

$$\begin{aligned} \psi_\alpha(\bar{\nu}_E, \bar{\nu}_I) &= \langle \nu_\alpha\rangle(\bar{\nu}_E, \bar{\nu}_I) \\ &= \int \phi^k \left(\sqrt{K}(\bar{w}_{\alpha E}\bar{\nu}_E - \bar{w}_{\alpha I}\bar{\nu}_I) + \sigma\xi + I_\alpha\right) p(\xi) d\xi. \end{aligned} \tag{3.47}$$

(We're being a little lax with notation here: $\bar{\nu} = \langle\nu\rangle$, $\bar{w} = \langle w\rangle$). Note that $\psi(\cdot)$ is sigmoidal—in fact, it's just a Gaussian-smoothed version of $\phi(\cdot)$. These are the *Wilson-Cowan equations*. Analyzing this two-dimensional system can provide insight into the following experimental observations:

1. low average firing rates (~ 0.2 Hz)
2. network oscillations
3. UP and DOWN states of high and low average membrane potential
4. short bumps of membrane potential (~ 10 ms) that are followed by down states of little activity lasting on the order of seconds

The system results in the nullclines plotted in Figure 3.1. To analyze the resulting three fixed points $\{\nu_E^*, \nu_I^*\}$, we can perform standard stability analysis using the Jacobian:

$$J = \begin{pmatrix} \tau_E^{-1}(\partial\,\psi_{EE} - 1) & \tau_E^{-1}\,\partial\,\psi_{EI} \\ \tau_I^{-1}\,\partial\,\psi_{IE} & \tau_I^{-1}(\partial\,\psi_{II} - 1) \end{pmatrix}, \tag{3.48}$$

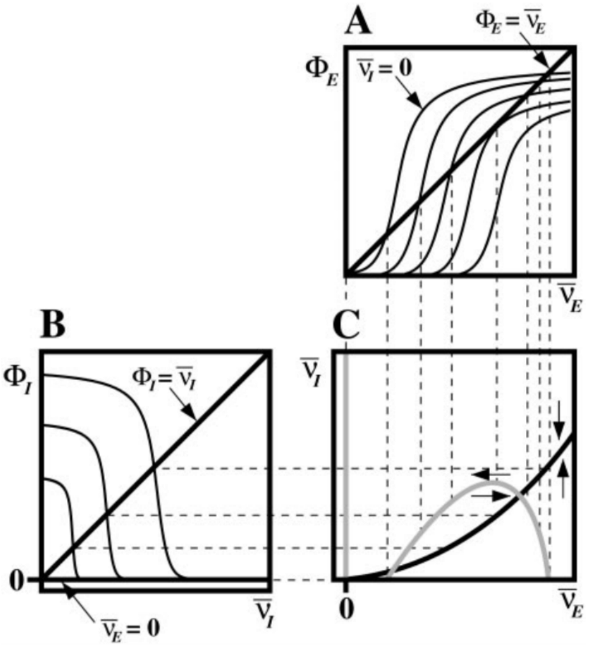


Figure 3.1: Nullcline plots for the Wilson-Cowan equations (taken from Peter's paper). In panel C, the light-colored curve corresponds to the excitatory nullcline, and the dark curve to the inhibitory nullcline. Φ_α is equivalent to ψ_α in the text. Each sigmoid in panels A and B corresponds to a different fixed setting for the other subpopulation firing rate (i.e., each sigmoid in A corresponds to a different fixed $\bar{\nu}_I$). The location of the intersection with the line $\Phi_\alpha = \bar{\nu}_\alpha$ gives the other firing rate, which together produces a point in the phase plane. For example, the value of $\bar{\nu}_E$ at the intersection with the $\bar{\nu}_I = 0$ sigmoid in A gives the coordinate $(\bar{\nu}_E, \bar{\nu}_I) = (\bar{\nu}_E, 0)$ for the E-nullcline in panel C.

where we denote

$$\partial\,\psi_{\alpha\beta} \coloneqq \frac{\partial \psi_\alpha}{\partial \nu_\beta}\Big|_{\nu_E^*,\nu_I^*}. \tag{3.49}$$

In order to be stable, we need $\mathrm{Tr}(J) < 0$ and $|J| > 0$. This gives us the following conditions for stability:

$$\frac{1 - \partial\,\psi_{II}}{\tau_I} > \frac{\partial\,\psi_{EE} - 1}{\tau_E} \tag{3.50}$$

$$(\partial\,\psi_{EE} - 1)(\partial\,\psi_{II} - 1) > \partial\,\psi_{EI}\,\partial\,\psi_{IE}. \tag{3.51}$$

These conditions can also be verified geometrically by perturbation analysis. It turns out that these conditions are equivalent to

$$m_E < m_I \tag{3.52}$$

$$m_E < 0, \tag{3.53}$$

where m_α is the slope of the α-nullcline at the fixed point. In general, eqs. 3.50 and 3.52 indicate that the slope of the inhibitory nullcline at a fixed point must be higher than that of the excitatory nullcline—this is intuitive, as the inhibitory firing rate must be more sensitive to changes in the excititatory firing rate than the excitatory population is itself in order to prevent runaway excitatory activity (similar in spirit to the BCM rule). The second rule(s) has several implications (see Jorge's notes, p. 27), but a key takeaway is that the network requires strong *I*-*I* coupling (a very negative $\partial\,\psi_{II}$) for stable equilibria on unstable branches of the ν_E nullcline.

These conditions then show us that the left and right fixed points in Figure 3.1C must be stable, while the center fixed point is unstable. Unfortunately, this result is inconsistent with the low average firing rates observed in cortex, as the system is only stable for high E and I firing rates and for firing rates of zero, but not for low firing rates. To get a stable fixed point at low firing rates, however, all we need to do is shift the nullclines such that the leftmost fixed point is moved away from the origin. Because ψ_E and ψ_I are monotonically increasing functions of the inputs currents I_E, I_I (which until now have been set to zero), we can do this by simply increasing the input current to each subpopulation. Biologically, this can be interpreted as the presence of *endogenously* active cells in the population, which provide activity even in the absence of input. It can also be thought of as continuous input to the network from other brain areas. This is equivalent to shifting the curves in Figure 3.1A and B to the left and right, respectively, and results in nullclines that look like those in Figure 3.2. However, because $m_E > 0$ at the resulting fixed point, its stability depends on the strengths of the weights. It turns out (see Jorge's notes for more detail), that the dynamics will be stable for low firing rates (experimental observation 1 above) if the average E-E connections are weak, the E-I and I-E connections are strong, and the I-I connections are strong, but not too strong (so as to not completely dampen activity).

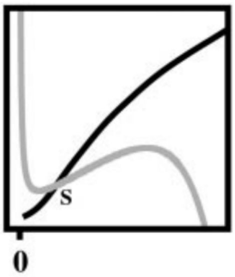


Figure 3.2: Nullcline plots for the Wilson-Cowan equations with injected currents (again taken from Peter's paper). The axes are the same as in Figure 3.1C.

Interestingly, if instead there are strong E-E connections and weak I-I connections, the unstable equilibrium in Figure 3.2 will produce a stable limit cycle around it[1]. This results in oscillatory activity, which is also seen in the cortex (observation 2 above). Such a transition, from a stable limit point to an unstable limit point with oscillatory activity, is a Hopf bifurcation (see), which occurs when changing a parameter (e.g., a time constant τ_α or weight matrix) changes the sign of the real part of an eigenvalue from negative to positive.

Finally, we can add spike-frequency adaptation to the system by expressing the external input I_α as a dynamical variable:

$$I_\alpha = \theta_\alpha - g_\alpha \tag{3.54}$$

$$\tau_{sfa}\frac{dg_\alpha}{dt} = G_\alpha \nu_\alpha - g_\alpha \tag{3.55}$$

where G_α is a constant and θ_α is a constant representing the number of endogenously active cells in the subpopulation. We can see that as the mean firing rate ν_α increases, g_α increases as well, pushing the current negative and shifting the nullclines downward and silencing the endogenously active cells. This results in a series of bifurcations that result in a lone fixed point at high firing rates, then a new fixed point at the origin is created, with $g_\alpha \to 0$, pushing $I_\alpha \to \theta_\alpha$ and resettling the network back to its low firing rate regime. This cycle is shown in Figure 3.3A→B→C, and explains observation 3 above. Finally, excluding spike-frequency adaptation, the positioning of the nullcines in Figure 3.3B results in behavior evocative of observation 4 (see Jorge's notes for more detail).

3.3 Hopfield Networks

NOTE: I haven't seen a more succinct explanation anywhere else, so most of this, save the end, is pretty much a copy of Jorge's Hopfield notes.

A Hopfield network forms an associative memory, such that an input will always result in dynamics converging to the nearest fixed point—or "memory"—in activity space. Consider a fully-connected network of N neurons with binary firing states $s_i(t) \in \{-1, +1\}$. The dynamics are modeled in discrete time, using the update

$$s_i(t+1) := \text{sign}\left(\sum_{j=1}^{N} W_{ij} s_j(t)\right), \tag{3.56}$$

where $\text{sign}(x) = 1$ if $x \geq 0$ and -1 otherwise. The key property of the Hopfield network is its connectivity matrix W, described by

$$\begin{aligned} W_{ij} &= \frac{1}{N}\sum_{m}^{M} \xi_i^m \xi_j^m, \quad W_{ii} = 0 \\ \xi_i^m &= \begin{cases} 1 & \text{w.p. } 1/2 \\ 0 & \text{w.p. } 1/2 \end{cases}, \end{aligned} \tag{3.57}$$

[1] Recall that a limit cycle is a closed orbit in the phase place exhibiting periodic behavior. One arises if (i) the fixed point is unstable and (ii) we can construct a bounding surface around it such that all derivatives on the boundary point toward its interior.

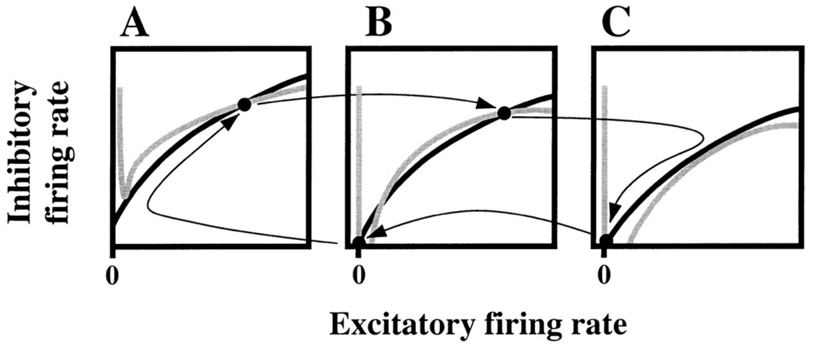


Figure 3.3: Nullcline plots for the Wilson-Cowan equations with spike-frequency adaptation.

where $M < N$. Thus connections are symmetric, with no self-connections (autapses).

Suppose that at time t, we have $s_i(t) = \xi_i^k$, $\forall i$. The update is then

$$\begin{aligned}
s_i(t+1) &= \operatorname{sign}\left(\sum_{j \neq i} W_{ij} s_j(t)\right) \\
&= \operatorname{sign}\left(\sum_{j \neq i} \frac{1}{N} \sum_m \xi_i^m \xi_j^m \xi_j^k\right) \\
&= \operatorname{sign}\left(\frac{1}{N} \sum_{j \neq i} \left[\underbrace{\xi_i^k \xi_j^k \xi_j^k}_{=\xi_i^k} + \sum_{m \neq k} \xi_i^m \xi_j^m \xi_j^k\right]\right) \\
&= \operatorname{sign}\left(\frac{1}{N} \sum_{j \neq i} \left[\xi_i^k + \sum_{m \neq k} \xi_i^m \xi_j^m \xi_j^k\right]\right) \\
&= \operatorname{sign}\left(\xi_i^k + \underbrace{\frac{1}{N} \sum_{j \neq i} \sum_{m \neq k} \xi_i^m \xi_j^m \xi_j^k}_{=\eta_i}\right) \\
&= \operatorname{sign}\left(\xi_i^k + \eta_i\right).
\end{aligned} \tag{3.58}$$

Since the terms inside the sum are independent, in the large $N \to \infty$, the central limit theorem lets approximate η_i with a Gaussian random variable (surprise, surprise):

$$\eta_i \sim \mu + \frac{\sigma}{\sqrt{N}} \zeta_i, \quad \zeta_i \sim \mathcal{N}(0,1) \tag{3.59}$$

$$\mu = \left\langle \sum_{m \neq k} \xi_i^m \xi_j^m \xi_j^k \right\rangle = \sum_{m \neq k} \langle \xi_i^m \rangle \langle \xi_j^m \rangle \langle \xi_j^k \rangle = 0 \tag{3.60}$$

$$\sigma^2 = \frac{1}{N} \sum_{j \neq i} \sum_{m \neq k} \left\langle (\xi_i^m)^2 \right\rangle \left\langle (\xi_j^m)^2 \right\rangle \left\langle (\xi_j^k)^2 \right\rangle = \frac{1}{N} \sum_{j \neq i} \xi_j^k \sum_{m \neq k} \sum_{m \neq k} 1 = \frac{1}{N}(N-1)(M-1) \approx M-1. \tag{3.61}$$

Then we get

$$s_i(t+1) = \operatorname{sign}\left(\xi_i^k + \sqrt{\frac{M-1}{N}} \zeta_i\right), \quad \zeta_i \sim \mathcal{N}(0,1). \tag{3.62}$$

Therefore, if $M \ll N$, $s_i(t+1) \approx \operatorname{sign}(\xi_i^k) = s_i(t)$, and the system is at an equilibrium.

Generalizing the above for all possible $k \in [1, 2, \ldots, M]$, the system then has M such equilibria, each local minimum of the network's *energy* E, given by

$$E = -\frac{1}{2} \sum_{i,j} W_{ij} s_i s_j. \tag{3.63}$$

Therefore, feeding some initializing input will lead the network to converge to the closest equilibrium state, constituting a kind of associative memory. The model can hold M such memories, provided that $M \ll N$ i.e., there are many more neurons than memories). Specifically, this is the case when there are at least 1000 neurons per 138 memories—$M/N \leq 0.138$.

A significant issue, however, is the assumed full-connectivity of the network. As discussed above, real networks are sparse, with low connectivity rates K/N. It turns out that if a neuron is connected to K others on average, then the updated state becomes

$$s_i(t+1) = \text{sign}\left(\xi_i^k + \sqrt{\frac{M-1}{K}}\zeta_i\right), \quad \zeta_i \sim \mathcal{N}(0,1). \tag{3.64}$$

This is an issue if M is not much smaller than K—it means that the equilibria are no longer stable. This can be ameliorated to some degree by adjusting $p(\xi_i^m) = +1$ to be $f < 1/2$ rather than $1/2$—this is called implementing a *sparse memory.* This increases the memory capacity, scaling roughly with K/f. If f is very small, i.e., $\mathcal{O}(1/N)$, then the original $\mathcal{O}(N)$ capacity of the full network is recovered. Unfortunately, in realistic spiking networks, background activity results in a lower bound on f, precluding the recovery of storage.

3.4 Networks Questions: Tips

- If a variable is drawn from a normal distribution, $w \sim \mathcal{N}(\mu, \sigma^2)$, rewrite it as $w = \mu + \delta w$, where $\delta w \sim \mathcal{N}(0, \sigma^2)$; it can also be written as $w \sim \mu + \sigma\xi$, where $\xi \sim \mathcal{N}(0,1)$.
- If there's an indicator variable ξ_i inside a sigmoidal function $\phi(\xi_i)$, in the large N limit, you can write
$$\frac{1}{N}\sum_i \phi(\xi_i) = p(\xi_i = 1)\phi(1) + p(\xi_i = 0)\phi(0). \tag{3.65}$$
- Be on the lookout to draw nullclines like those in Figure 3.1A,B.
- Differential equations like the one that led to eq. 3.5 and the ensuing analysis to get the mean firing rate may show up—in general, if there's an ODE with a sum of delta functions, just consider the equation with respect to one (aka one spike). Before that spike arrives, there's nothing happening (if the only other term is a decay term), so pretend that arrival time is $t = 0$ and just integrate as if the delta wasn't there. The ensuing exponential is just multiplied by the Heaviside function (the integral of the delta function). You can then sum the functions you get from each spike.
- Any time you see $\frac{1}{N}\sum \cdots$ in the large N limit, immediately replace it with $\langle \cdots \rangle$. (And note that $\sum \cdots = N\langle \cdots \rangle$.)
- The strategy used to compute the update mean and variance for the Hopfield network is a useful one.
- A sum of K terms that are all uncorrelated and take the values ±1 has zero mean and K variance.
- Take advantage of the parity of sigmoidal functions. For example, if $\xi = \pm 1$, the oddness of $\tanh(x)$ means that $\xi \tanh(x) = \tanh(\xi x)$.

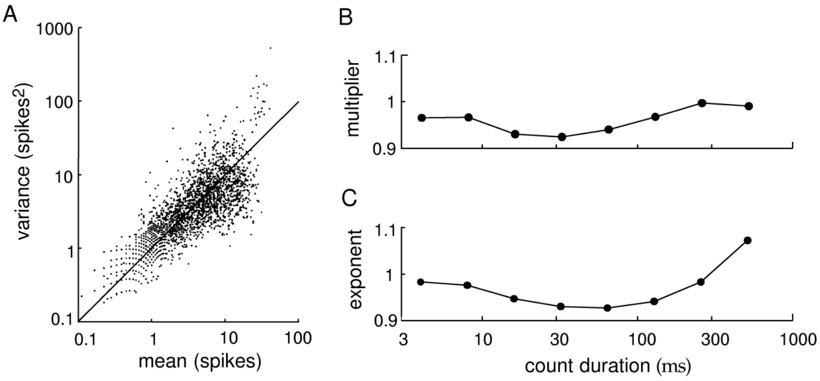


Figure 4.1: Count variability in spike trains.

4 Point Processes

The brain manipulates information through action potentials (aka spikes). It's therefore natural to ask how information is represented in spike trains. While the shapes of action potential time courses may vary slightly, there's no evidence that action potential shape affects vesicle release. Therefore, it seems that the information encoded by a spike is represented entirely by its time of occurrence.

To study the statistics of spike trains, it's necessary to introduce some notation. We formally define a spike train $\mathcal{S}$ as the sequence of times at which a neuron spikes:

$$\mathcal{S} \coloneqq \{t_1, \ldots, t_N\}. \tag{4.1}$$

This can be written as a function in time using Dirac delta functions:

$$s(t) = \sum_{i=1}^{N} \delta(t - t_i), \tag{4.2}$$

which visually has the form of a sequence of zero-width spikes. A spike train can also be represented as cumulative counting function of the number of spikes occurring up to a time t:

$$N(t) = \int_0^{\to t} s(\xi)\, d\xi, \tag{4.3}$$

where the notation $\to t$ indicates that t is not included in the integral. This representation takes the form of a non-decreasing step-function. Finally, we can also represent a spike train as a vector in discrete time with bin width Δt:

$$\mathbf{s} = [s_1, \ldots, s_{T/\Delta t}]^\mathsf{T}; \qquad s_t = \int_{t-\Delta t}^{\to t} s(\xi)\, d\xi. \tag{4.4}$$

Thus, s_t represents the number of spikes landing in each bin. Note that due to neural refractory periods, $\Delta t \approx 1$ ms results in a binary s_t.

Neural responses to repeated stimuli are highly variable, especially in cortex. This variability likely arises in several ways:

- Noise—this may arise either through vesicle release or thermal noise in conductances.
- Internal processes—ongoing processing within the brain is likely to affect sensory responses. This might result in variability that operates on a slower time-scale than noise.

Denote the spike count on the ith repetition ("trial") by $N_i \coloneqq \int_0^T s_i(\xi)\, d\xi$. Experimental evidence shows that the variability in N_i is on the order of the mean (Figure 4.1A). If we fit a model of the relationship via

$$\mathbb{V}[N_i] = A \cdot \mathbb{E}\left[N_i\right]^B, \tag{4.5}$$

the best-fit values turn out to be $A, B \approx$ 1-1.5 (Figure 4.1B,C). Note that when $B = 1$, A defines the *Fano factor*, the ratio of the variance to the mean. Skipping ahead a bit, a Fano factor of 1 is suggestive of a Poisson process, as the mean of a Poisson distribution equals the variance.

To describe spike trains, we define a *point process* as a probabilistic process that produces events of the type

$$\mathcal{S} = \{t_1, \ldots, t_N\} \subset \mathcal{T}, \tag{4.6}$$

where we define $\mathcal{T} := [0, T]$ to be a time interval. Every point process defined on an ordered set is associated with a dual *counting process* which produces events $N(t)$ of the type

$$\begin{aligned} N(t) &\geq 0 \quad \text{(non-negativity)} \\ N(t') &\geq N(t) \text{ if } t' > t \quad \text{(non-decreasing—"staircase" shape)} \\ N(t) - N(s) &:= N[s, t) \in \mathbb{Z} \quad \text{(count within an interval).} \end{aligned} \tag{4.7}$$

$N(t)$ then gives the number of events occurring with $t_i < t$.

4.1 Homogeneous Point Processes

In the simplest form of point process, events are characterized by two features: *independence* and occurrence at a fixed *rate* λ. We define these terms as follows:

1. *Independence*: For all disjoint intervals $[s, t)$ and $[s', t')$, $N_\lambda[s, t) \perp N_\lambda[s', t')$.
2. *Mean event rate*: $\mathbb{E}\left[N_\lambda[s, t)\right] = (t - s)\lambda$.

Independence implies that knowing the number (or times) of one or more events tells us nothing about the other possible events. Note that condition 2 assumes that

$$\lim_{ds \to 0} N_\lambda[s, s + ds) \in \{0, 1\} \tag{4.8}$$

(in other words, that as the bin gets infinitely small, there's either a spike or there isn't). This assumption is called *conditional orderliness*—at most event occurs at one time. Without assuming conditional orderliness, we could instead define the process by giving the whole distribution $N_\lambda[s, t)$. Instead, we will use the more restrictive defining assumption to derive the distribution.

We can use the two conditions above to derive the distribution. First, we divide the interval $[s, t)$ into M bins of length Δ (i.e., $M = (t - s)/\Delta$). If $\Delta \ll 1/\lambda$, conditional orderliness implies that the spike count per bin is binary (note also that the inverse rate $1/\lambda$ is *period* of the process). For a binary random variable (aka an indicator variable), the expectation is the same as the probability of the event, so the mean event rate gives

$$P(N[t, t + \Delta) = 1) = \mathbb{E}\left[N_\lambda[t, t + \Delta)\right] = (t + \Delta - t)\lambda = \lambda\Delta. \tag{4.9}$$

The distribution of $N[s, t)$ is binomial, and the probability of n spikes occurring in the interval is:

$$\begin{aligned} P(N_\lambda[s, t) = n) &= \binom{M}{n} (\lambda\Delta)^n (1 - \lambda\Delta)^{M-n} && (4.10) \\ &= \frac{M!}{n!(M - n)!} \left(\lambda \underbrace{\frac{t - s}{M}}_{=\Delta}\right)^n \left(1 - \lambda\frac{t - s}{M}\right)^{M-n} && (4.11) \end{aligned}$$

and writing $\mu := \lambda(t - s)$, we get

$$= \frac{\mu^n}{n!} \frac{\overbrace{M(M - 1) \cdots (M - n + 1)}^{n \text{ terms}}}{M^n} \left(1 - \frac{\mu}{M}\right)^{-n} \tag{4.12}$$

taking the limit $\Delta \to 0$ or, equivalently, $M \to \infty$

$$= \frac{\mu^n}{n!} 1^n 1^{-n} e^{-\mu} = e^{-\mu} \frac{\mu^n}{n!}. \tag{4.13}$$

Therefore, the spike count is Poisson distributed. As mentioned above, however, we could have dispensed with the conditional orderliness assumption and instead made the equivalent defining property

2. *Count distribution*: $N_\lambda[s, t) \sim \text{Poiss}[(t - s)\lambda]$.

We will now derive a number of properties of the homogeneous Poisson process.

Count Variance We first derive the variance of the count distribution (a key characteristic of the Poisson distribution):

$$\begin{aligned}
\mathbb{V}[N_\lambda[s,t)] &= \left\langle (n-\mu)^2 \right\rangle = \left\langle n^2 \right\rangle - \mu^2 \\
&= \left\langle \underbrace{n(n-1)+n}_{\text{good trick to know}} \right\rangle - \mu^2 \\
&= \underbrace{\sum_{n=0}^{\infty} n(n-1)\frac{e^{-\mu}\mu^n}{n!}}_{=\langle n(n-1)\rangle} + \underbrace{\mu}_{=\langle n\rangle} - \mu^2 \\
&= \mu^2 \underbrace{\sum_{n=0}^{\infty} \frac{e^{-\mu}\mu^{n-2}}{(n-2)!}}_{=0 \text{ for } n=0,1} + \mu - \mu^2 \\
&= 0 + 0 + \mu^2 \underbrace{\sum_{(n-2)=0}^{\infty} \frac{e^{-\mu}\mu^{n-2}}{(n-2)!}}_{=1} + \mu - \mu^2 \\
&= \mu^2 + \mu - \mu^2 = \mu.
\end{aligned} \tag{4.14}$$

Thus, the mean equals the variance and

3. *Fano factor*: $\frac{\mathbb{V}[N_\lambda[s,t)]}{\mathbb{E}[N_\lambda[s,t)]} = 1$.

ISI Distribution We now discuss the statistics governing the *inter-spike interval* (ISI). First, it is fairly straightforward to see that, since the counting processes before and after event t_i are independent, the times to the previous and following spikes are independent as well:

4. *ISI independence*: $\forall i > 1$, $t_i - t_{i-1} \perp t_{i+1} - t_i$.

The full ISI distribution can be derived from the count distribution:

$$\begin{aligned}
P[t_{i+1} - t_i \in [\tau, \tau + d\tau)] &= \overbrace{P[N_\lambda[t_i, t_i + \tau) = 0]}^{=P[\text{no spikes here}]} \times \overbrace{P[N_\lambda[t_i + \tau, t_i + \tau + d\tau) = 1]}^{=P[\text{next spike occurs in infinitesimal interval}]} \\
&= \frac{\mu^0 e^{-\lambda\tau}}{0!} \times \frac{\lambda d\tau e^{-\lambda d\tau}}{1!} \\
&= e^{-\lambda\tau}\lambda d\tau e^{-\lambda d\tau}
\end{aligned}$$

taking $d\tau \to 0$

$$= \lambda e^{-\lambda\tau}\, d\tau, \tag{4.15}$$

and therefore

5. *ISI distribution*: $\forall i \geq 1$, $t_{i+1} - t_i \sim$ iid Exponential$[\lambda^{-1}]$.

From this it follows that

6. *Mean ISI*: $\mathbb{E}\left[t_{i+1} - t_i\right] = \lambda^{-1}$
7. *Variance ISI*: $\mathbb{V}[t_{i+1} - t_i] = \lambda^{-2}$

These two properties imply that the *coefficient of variation* (CV) of the ISIs, defined as the ratio of the standard deviation to the mean, is $CV = \sqrt{\lambda^{-2}}/\lambda^{-1} = 1$.

Joint Density Finally, we consider the joint probability of observing a spike train $\{t_1, \ldots, t_N\}$ in interval $\mathcal{T}$. Spike times are independent and arrive at a uniform rate, giving

$$p(t_1, \ldots, t_N)\, dt_1 \ldots dt_N = P[N \text{ spikes in } \mathcal{T}] \times \overbrace{P[i\text{th } \textit{spike} \in [t_i, t_i + dt_i)]}^{\forall i} \times [\# \text{ of equiv. spike orderings}], \tag{4.16}$$

where the first term is given by the Poisson distribution, the second by the uniform distribution of spike times conditioned on N, and the third is $N!$, giving

$$\begin{aligned} p(t_1, \ldots, t_N)\, dt_1 \ldots dt_N &= \left(\frac{(\lambda T)^N e^{-\lambda T}}{N!}\right) \left(\frac{dt_1}{T} \cdots \frac{dt_N}{T}\right) N! \\ &= \lambda^N e^{-\lambda T}\, dt_1 \ldots dt_N . \end{aligned} \tag{4.17}$$

We will see another way to write down the same expression while considering the inhomogeneous Poisson process below.

4.2 Inhomogeneous Point Processes

The inhomogeneous Poisson process generalizes the constant event-arrival rate λ to a time-dependent rate $\lambda(t)$, while preserving the assumption of independent spike arrival times. It's possible to quickly summarize the properties of the inhomogeneous process by reference to the homogeneous one.

To begin, the two defining properties are

1. *Independence*: For all disjoint intervals $[s, t)$ and $[s', t')$, $N_{\lambda(t)}[s, t) \perp N_{\lambda(t)}[s', t')$.
2. *Count distribution*: $N_{\lambda(t)}[s, t) \sim \text{Poiss}[\int_s^t \lambda(\xi)\, d\xi]$ (i.e., the rate is the expected number of events in the interval).

The variance in the counts is simply a consequence of the Poisson counting distribution, and so the next property follows directly:

3. *Fano factor*: $\frac{\mathbb{V}[N_{\lambda(t)}[s,t)]}{\mathbb{E}[N_{\lambda(t)}[s,t)]} = 1$.

ISI Distribution The independence of counting in disjoint intervals means that ISIs remain independent:

4. *ISI independence*: $\forall i > 1$, $t_i - t_{i-1} \perp t_{i+1} - t_i$.

The full distribution of ISIs is found in a similar manner to that of the homogeneous process distribution:

$$\begin{aligned} P[t_{i+1} - t_i \in [\tau, \tau + d\tau)] &= \overbrace{P[N_{\lambda(t)}[t_i, t_i + \tau)]}^{P(\text{no spike in interval})} \times \overbrace{P[N_{\lambda(t)}[t_i + \tau, t_i + \tau + d\tau)]}^{P(\text{spike at } t_i + \tau)} \\ &= e^{-\int_{t_i}^{t_i+\tau} \lambda(\xi)\, d\xi} \times e^{-\int_{t_i+\tau}^{t_i+\tau+d\tau} \lambda(\xi)\, d\xi} \int_{t_i+\tau}^{t_i+\tau+d\tau} \lambda(\xi)\, d\xi \end{aligned}$$

taking $d\tau \to 0$

$$\begin{aligned} &= e^{-\int_{t_i}^{t_i+\tau} \lambda(\xi)\, d\xi} \times \underbrace{e^{-\lambda(t_i+\tau)\, d\tau}}_{d\tau \to 0} \lambda(t_i + \tau)\, d\tau \\ &= e^{-\int_{t_i}^{t_i+\tau} \lambda(\xi)\, d\xi} \lambda(t_i + \tau)\, d\tau, \end{aligned} \tag{4.18}$$

and thus we have

5. *ISI distribution*: $\forall i \geq 1$, $P(t_{i+1} - t_i) = e^{-\int_{t_i}^{t_i+\tau} \lambda(\xi)\, d\xi} \lambda(t_{i+1})$.

Because the ISI distribution is *not* iid, it is not as useful to consider its mean or variance.

Joint Density The joint probability of the event $\{t_1, \ldots, t_N\}$ can be derived by setting the count in intervals *between* spikes to 0, and the count in an infinitesimal *around* t_i to 1. This gives

$$\begin{aligned} p(t_1, \ldots, t_N)\, dt_1 \ldots dt_N &= \underbrace{P[N[0, t_1) = 0]}_{\text{no spikes before first}} \times P[N[t_1, t_1 + dt_1) = 1] \times \cdots \times \underbrace{P[N(t_N, T) = 0]}_{\text{no spikes after last}} \\ &= e^{\int_0^{t_1} \lambda(\xi)\, d\xi} \times \lambda(t_1)\, dt_1 \times \cdots \times e^{\int_{t_N}^{T} \lambda(\xi)\, d\xi} \\ &= \underbrace{e^{-\int_0^T \lambda(\xi)\, d\xi}}_{\text{all non-spikes}} \underbrace{\prod_{i=1}^{N} \lambda(t_i)\, dt_1 \ldots dt_N}_{\text{all the spikes}} . \end{aligned} \tag{4.19}$$

This form can be generalized to pretty much any point process. Observe that it takes the typical form of a joint distribution—a product of terms times a normalizer. Note also that if we set $\lambda(t) = \lambda$, we recover the result for the homogeneous process.

Time Rescaling Finally, we derive an additional important property of the inhomogeneous process. Let us rewrite the density above by changing variables from t to u according to

$$u(t) := \int_0^t \lambda(\xi)\, d\xi \qquad \text{i.e., } u_i = \int_0^{t_i} \lambda(\xi)\, d\xi \Leftrightarrow \frac{du_i}{dt_i} = \lambda(t_i) \Leftrightarrow du_i = \lambda(t_i)\, dt_i. \tag{4.20}$$

Then

$$\begin{aligned} p(u_1, \ldots, u_n) &= \frac{p(t_1, \ldots, t_n)}{\prod_i \frac{du_i}{dt_i}} \\ &= e^{-u(T)} \frac{\prod_{i=1}^N \lambda(t_i)}{\prod_{i=1}^N \lambda(t_i)} \\ &= e^{-u(T)}. \end{aligned} \tag{4.21}$$

Comparing this to the density for a homogeneous Poisson process shows that the variables u_i are distributed according to a homogeneous Poisson process with mean rate $\lambda = 1$. Thus, $u(t)$ effectively *rescales time* to transform the inhomogeneous process into a homogeneous process. This process is called time rescaling, and is central to the study of point processes in time.

4.3 Self-Exciting and Renewal Processes

A *self-exciting process* has an intensity function that is conditioned on past events:

$$\lambda(t) \to \lambda(t|N(t), t_1, \ldots, t_{N(t)}). \tag{4.22}$$

We can then define the notation $H(t)$ to represent the event *history* at time t—representing both $N(t)$ and the times of the corresponding events. Then the self-exciting intensity function can be written $\lambda(t|H(t))$. This is actually the most general form of a point process—we can re-express any (conditionally orderly) point process in this form. To see this, consider the point process to be the limit as $\Delta \to 0$ of a binary time series $\{b_1, b_2, \ldots, b_{T/\Delta}\}$ and note that

$$P(b_1, b_2, \ldots, b_{T/\Delta}) = \prod_i P(b_i|b_{i-1}, \ldots, b_1) \propto \prod_i \lambda(b_i|b_1, \ldots, b_{i-1})\Delta, \tag{4.23}$$

and taking the limit $\Delta \to 0$ gives

$$= f(\lambda(t|H(t))), \tag{4.24}$$

for some function $f(\cdot)$.

Renewal Processes If the intensity of a self-exciting process depends only on the time since the last spike, i.e.,

$$\lambda(t|H(t)) = \lambda(t - t_{N(t)}), \tag{4.25}$$

then the process is called a *renewal process.* ISIs from a renewal process are iid and so we can could equivalently have defined the process by its ISI density. This gives an (almost) easy way to write the probability of having observed $\{t_1, \ldots, t_N\}$ in T. Suppose, for simplicity, that there was an event at $t_0 = 0$. Then if the ISIs are distributed according to a probability density $p(\tau)$:

$$p(t_1, \ldots, t_N)\, dt_1 \ldots dt_N = \prod_{i=1}^N \underbrace{p(t_i - t_{i-1})}_{\text{prob. of interval}} \underbrace{\left(1 - \int_0^{T-t_N} p(\tau)\, d\tau\right)}_{\text{prob. of no events from } t_N \text{ to } T}, \tag{4.26}$$

where the last term gives the probability that no more spikes are observed after t_N. Note that if we had not assumed that there was a spike at time $t = 0$, we would have needed a similar term at the front. The conditional intensity—sometimes called the *hazard function*—for the renewal process defined by ISI density $p(\tau)$ is

$$\lambda(t|t_{N(t)})\, dt = \frac{\overbrace{p(t - t_{N(t)})}^{\text{prob. of event at } t \text{ given prev. event at } t_{N(t)}}}{\underbrace{1 - \int_0^{t-t_{N(t)}} p(\tau)\, d\tau}_{\text{prob. of no events in interval of len. } t - t_{N(t)}}}\, dt, \tag{4.27}$$

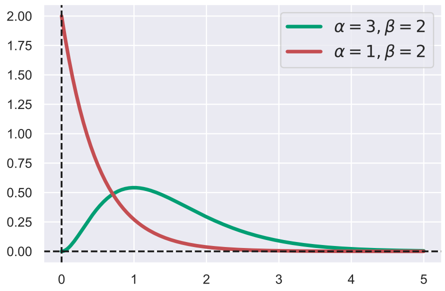


Figure 4.2: Example ISI distributions for gamma-interval processes. The polynomial build-up to the peak for $\alpha > 1$ (given by $\tau^{\alpha-1}$) can be interpreted as a refractory period. Setting $\alpha = 1$ recovers an exponential distribution, the ISI distribution for a homogeneous process.

which is indeed a function only of $t - t_{N(t)}$. Note that the general form for the ISI distribution for a renewal process is given by

$$p(\tau) = \tau e^{-\int_0^\tau \tau'\, d\tau'}, \tag{4.28}$$

an exponential distribution—similar to that of the inhomogeneous Poisson process.

Gamma-Interval Process A renewal process with ISI density given by

$$t_{i+1} - t_i \sim^{\text{iid}} \text{Gamma}[\alpha, \beta], \tag{4.29}$$

where

$$\tau \sim \text{Gamma}[\alpha, \beta] \Rightarrow p(\tau) = \frac{\beta^\alpha}{\Gamma(\alpha)} \tau^{\alpha-1} e^{-\beta\tau} \tag{4.30}$$

is a *gamma-interval process.* This is an important renewal process in theoretical neuroscience, because the ISI distribution has a refractory-like component (see Figure 4.2). A homogeneous Poisson process is a gamma-interval process (and therefore a renewal process) with $\alpha = 1$ (as setting $\alpha = 1$ converts a gamma distribution to an exponential distribution). The parameter α is sometimes called the "order" or the "shape" parameter of the gamma-interval process. Larger values of α shape the polynomial rising part of the gamma density, thus implementing a relative refractory period. The long-time behavior is dominated by the exponential decay with coefficient β.

Interestingly, it's possible to construct a gamma-interval process of integral order α by drawing every αth event from a homogeneous Poisson process (this is trivially true for $\alpha = 1$). Consider the case that $\alpha = 2$ (depicted in Figure 4.3). The ISI probabilities for a process that accepts times t_1 and t_2 and rejects a spike at time t is just the product of the ISI probabilities of the homogeneous process from t_1 to t and from t to t_2, as disjoint intervals are independent. However, we must also integrate (average) over all possible settings of t. Letting $\tau := t_2 - t_1$, we have

$$\begin{aligned}
P(\tau) &= \int_{t_1}^{t_2} \overbrace{P(t - t_1)P(t_2 - t)}^{\text{homog. process ISI probs.}} \, dt && (4.31)\\
&= \int_{t_1}^{t_2} \lambda e^{-\lambda(t - t_1)} \lambda e^{-\lambda(t_2 - t)} \, dt && (4.32)\\
&= \lambda^2 \int_{t_1}^{t_2} e^{-\lambda t} e^{\lambda t_1} e^{-\lambda t_2} e^{\lambda t} \, dt && (4.33)\\
&= \lambda^2 [t_2 - t_1] e^{-\lambda(t_2 - t_1)} && (4.34)\\
&= \lambda^2 \tau e^{-\lambda\tau} = \text{Gamma}[2, \lambda]. && (4.35)
\end{aligned}$$

This can be easily generalized to higher values of α. Thus, taking every αth spike results in a gamma process of order α and rate parameter $\beta = \lambda$.

Inhomogeneous Renewal Processes In an *inhomogeneous* renewal process, the rate depends both on the time since the last spike and on the current time:

$$\lambda(t) \to \lambda(t, t - t_{N(t)}). \tag{4.36}$$

This is also called an "inhomogeneous Markov interval" process. There are two popular ways to construct an inhomogeneous renewal process:

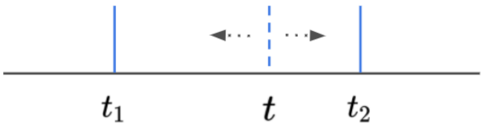


Figure 4.3: Constructing a gamma process by drawing every other spike in a homogeneous Poisson process. The times t_1 and t_2 are spike times that are accepted by the constructed process. The “skipped" spike time time is t.

1. *Time Rescaling*: Given unit-mean ISI density $p(\tau)$ and time-varying intensity $\rho(t)$, define

$$p(t_1, \ldots, t_N)\, dt_1 \ldots dt_N := \prod_{i=1}^{N} p\left(\int_{t_{i-1}}^{t_i} \rho(\xi)\, d\xi\right)\left(1 - \int_0^{\int_{t_N}^{T} \rho(\xi)\, d\xi} p(\tau)\, d\tau\right) \tag{4.37}$$

2. *Spike-Response*:

$$\lambda(t, t - t_{N(t)}) := f(\rho(t), h(t - t_{N(t)})) \tag{4.38}$$

   for a simple f. Often, f just multiplies the two functions (or, equivalently, adds log-intensities). The term "spike-response" comes from the use of such spike-triggered currents to create a potentially more tractable approximation to an integrate-and-fire neuron.

These definitions differ in how ISI density depends on ρ. In *rescaling*, higher rates make time pass faster, so ISI interactions are rescaled. *Spike-response*: a refractory h may not suppress spikes as well at higher rates, but the duration of influence does not change.

4.4 General Spike-Response Processes

This category of processes has come to be used with increasing frequency recently, particularly in a generalized linear form. The product form of spike-response renewal processes can be written as

$$\lambda(t, t - t_{N(t)}) = \exp(\rho(t) + h(t - t_{N(t)})) \tag{4.39}$$

and then generalized to include influence from all (or > 1) past spikes:

$$\lambda(t|H(t)) = \exp\left(\rho(t) + \sum_j h(t - t_{N(t)-j})\right). \tag{4.40}$$

Often, we want to estimate the parameters of a point-process model from spike data. Assuming a generalized linear form makes this easier. To do this, we can write the history influence h in terms of a linear combination of basis functions $h_i(\tau)$:

$$\lambda(t|H(t)) = \exp\left(\rho(t) + \sum_{ij} \alpha_i h_i(t - t_{N(t)-j})\right). \tag{4.41}$$

Note that this essentially defines an exponential family form for the intensity function—learning the weights α_i corresponds to learning the parameters of the sufficient statistics with encoding functions h_i, which define the biophysical properties of the neuron and shouldn't change. If $\rho(t)$ is also written as a linear function of external covariates, then the complete model can be fit by the standard methods used for generalized linear models (GLMs).

The Doubly-Stochastic Poisson (or Cox) Process In the *doubly-stochastic* or *Cox* process, $\lambda(t)$ itself is either a random variable or depends on another random process $x(t)$. One example is the randomly scaled IHPP:

$$\lambda(t) := s \cdot \rho(t), \tag{4.42}$$

with $\rho(t)$ fixed and $s \sim \text{Gamma}(\alpha, \beta)$. These models are useful for modeling a stimulus-dependent response $\rho(t)$ which is modulated by cortical excitability. The counting process for such a DSPP has a relatively high variance, with Fano factor greater than 1. DSPP models also provide a useful way to introduce dependencies between two or more point processes, through correlations in their intensity functions, and are common inputs to log-linear spike response models. One important example of a DSPP is *Gaussian process factor analysis* (GPFA). The intensity functions for GPFA are written as

$$\lambda^i(t) = \exp\left(\sum_k C_{ik}\rho^k(t)\right), \qquad \text{with } C_{ik} \sim \mathcal{GP}(0, K), \tag{4.43}$$

where K is some covariance function.

Joint Models Some examples of joint models are 2D point processes (not considered useful, as they may pair neurons without reason), superimposed processes, and infinitely divisible Poisson processes. It's also useful to consider *multivariate processes*, in which the intensity function for the ith neuron can be written in the general form

$$\lambda^i(t) = f(t, H^i(t), H^j(t), \dots). \tag{4.44}$$

If $f(\cdot,\cdot)$ is linear, the process is called a *Hawkes process*, and can be written as

$$\lambda^i_H(t) = g^i(t) + \sum_{j,n_j} \underbrace{h_{ij}(t - t^j_{n_j})}_{\text{basis fns}}, \tag{4.45}$$

where j indexes neighboring neurons and n_j indexes the spike times of the neighboring neurons. When the dependence is log-linear, it's called a generalized Hawkes process—or a GLM, i.e.,

$$\lambda^i_{\mathrm{GLM}}(t) \coloneqq \exp(\lambda^i_H(t)). \tag{4.46}$$

4.5 Measuring Point Processes

Given data, we can construct generative models for point processes, but it's also important to consider the techniques used to measure and analyze them. Consider a data set in which spike trains from a single neuron are obtained from repeated experiments under constant experimental conditions:

$$s^{(k)}(t) = \sum_{i=1}^{N^{(k)}} \delta(t - t_i^{(k)}) \qquad \text{for trials } k = 1, \dots, K. \tag{4.47}$$

Visually, we can think of this as multiple sequences of delta-spikes occurring at different times, indexed by k. How can we characterize $s^{(k)}(t)$ and its relationship to the stimulus (or task)?

One family of options are parametric point-process models, possibly dependent on a stimulus $a(t)$:

$$s^{(k)}(t) \sim \lambda\left(t, a[0,t), N^{(k)}(t), t_1^{(k)}, \dots, t^{(k)}_{N^{(k)}(t)}, \theta\right). \tag{4.48}$$

In other words, this model attempts to predict the activity from the input stimulus. Such a framework constitutes an *encoding* model. Encoding models are discussed in depth in Section 7. The inverse approach—termed *decoding*—is to construct an algorithm that estimates $a(t)$ from $s^{(k)}(t)$:

$$\hat{a}(t) = F_\theta[s^{(k)}[0,t), \theta]. \tag{4.49}$$

One non-parametric option is to estimate statistics (usually *moments*) of the distribution of $s^{(k)}(t)$.

Another problem is simultaneously modelling responses from multiple cells. If no two processes can generate events at precisely the same time (a form of conditional orderliness), or if simultaneous spiking events are independent, then dependencies between the processes arise through dependence on all previous events in all cells:

$$\lambda^{(c)}(t) \coloneqq \lambda^{(c)}\left(t|N^{(c)}(t), t_1^{(c)}, \dots, t_1^{(c)}, \{N^{(c')}(t), t_1^{(c')}, \dots, t_1^{(c')}\}, \theta\right) \tag{4.50}$$

where c' indexes all other cells. This is analogous to the self-exciting point process intensity function. Dependencies can also be expressed in other forms, for example by DSPPs with the latent random process shared (or correlated) between cells. Such representations may often be more natural or causally accurate.

4.5.1 Mean Intensity and the PSTH

We now return to the case of multiple trials from a single neuron. The simplest non-parametric characterization of a spike process is with the *mean intensity*:

$$\bar{\lambda}(t) \coloneqq \langle s(t) \rangle = \lim_{K\to\infty} \frac{1}{K} \sum_{k=1}^{K} s^{(k)}(t). \tag{4.51}$$

Note that this is *not* the same as the intensity function for the point process marginalized over history (unless it's Poisson, in which case, the intensity is already history-independent):

$$\bar{\lambda}(t, a(\cdot)) := \int \int p(t_1, \ldots, t_{N(t)}) \lambda(t, a(\cdot), N(t), t_1, \ldots, t_{N(t)})\, dt_1 \ldots dt_{N(t)}\, dN(t). \tag{4.52}$$

For finite K, estimating $\bar{\lambda}$ by summing δ-functions yields spikey results. Instead, we can bin using a histogram:

$$\bar{N}[\widehat{t, t+\Delta t}) = \frac{1}{K} \qquad \underbrace{\sum_{k=1}^{K} N^{(k)}[t, t+\Delta t)}_{\text{sum spikes w/in bin across trials}} \qquad . \tag{4.53}$$

This is called the *peri-(post-)stimulus time histogram* (PSTH). If we'd like $\bar{\lambda}(t)$ to be smooth, we can use a kernel $\phi(\tau)$:

$$\widehat{\bar{\lambda}(t)} = \frac{1}{K} \sum_{k=1}^{K} \int \phi(\tau) s^{(k)}(t-\tau)\, d\tau. \tag{4.54}$$

Note the similarity to kernel density estimation (without normalization). The width of ϕ can be chosen adaptively, depending on the local density of spikes. In general, sampling from a smoothed function makes more sense than smoothing a binned histogram. Alternatively, one can also impose a smooth prior prior (e.g., a GP) on a time-varying aspect of the intensity: $\lambda(t)$ for an inhomogeneous Poisson process, or, e.g., $\rho(t)$ for an inhomogeneous gamma-interval of order γ:

$$\begin{aligned} \boldsymbol{\rho} &\sim \mathcal{N}(\mu \mathbf{1}, K_\theta) \\ p(t_1, \ldots, t_N | \boldsymbol{\rho}) &= \prod_{i=1}^{N} \left[\underbrace{\frac{\gamma x_{t_i}}{\Gamma(\gamma)} \left(\gamma \sum_{j=t_{i-1}}^{t_{i-1}} \rho_j \Delta \right)^{\gamma-1}}_{\text{rescaled time under intensity fn}} \exp\left(-\gamma \sum_{j=t_{i-1}}^{t_{i-1}} \rho_j \Delta \right) \right] \end{aligned} \tag{4.55}$$

The posterior on $\rho(t)$ can then be found via approximate inference (e.g., Laplace, EP, etc.).

4.5.2 Autocorrelation and Auotocovariance

The *autocorrelation function* for a process that generates spike trains $s(t)$ is

$$R_{ss}(\tau) = \left\langle \frac{1}{T} \int s(t) s(t-\tau)\, dt \right\rangle, \tag{4.56}$$

where $\langle \cdot \rangle$ denotes an expectation with respect to random draws of $s(t)$ from the process. This is the *time-averaged* local second moment of the joint on $s(t)$ (note that $\bar{\lambda}(t)$ is the *non*-time-averaged first moment). Also note that, since $s(t)$ is a sum of δ functions, $R_{ss}(0) = \int \delta^2\, dt = \infty$ under this definition.

Alternatively, we could define R_{ss} as the time-averaged conditional first moment. In other words, the mean intensity at $t + \tau$, conditioned on an event at time t, averaged over t:

$$R_{ss}^{alt}(\tau) = \frac{1}{T} \int \langle \lambda(t+\tau | \exists i : t_i = t) \rangle\, dt, \tag{4.57}$$

where $\langle \cdot \rangle$ here denotes an expectation with respect to $N(T)$ and $t_{j \neq i}$. In this case, $R_{ss}^{alt}(0) = 0$. For the rest of the discussion, we'll stick to the first (second-moment) definition.

Using the identity $\langle x^2 \rangle = \mathbb{V}[x] + \mu^2 = \langle (x-\mu)^2 \rangle + \mu^2$, we can decompose the autocorrelation function as follows:

$$R_{ss}(\tau) = \bar{\Lambda}^2 + \frac{1}{T} \int (\bar{\lambda}(t) - \bar{\Lambda})(\bar{\lambda}(t-\tau) - \bar{\Lambda})\, dt + \underbrace{\left\langle \frac{1}{T} \int (s(t) - \bar{\lambda}(t))(s(t-\tau) - \bar{\lambda}(t-\tau))\, dt \right\rangle}_{:=Q_{ss}(\tau)}, \tag{4.58}$$

where $\bar{\Lambda}$ is the time-averaged mean rate. $Q_{ss}(\tau)$ is called the *autocovariance* function. This has several notable properties.

- For an (inhomogeneous) Poisson process $Q_{ss}(\tau) = \delta(\tau)$ by independence.

- For a general self-exciting process, $Q_{ss}(\tau)$ gives (to second order) dependence on nearby spike times.
- The autocovariance function is often used to look for oscillatory structure in spike trains (where spikes tend to repeat around fixed intervals, but at random phase with respect to the stimulus) or similar spike-timing relationships.
- But, as any point process is self-exciting, *any* non-Poisson process will have non-δ autocovariance, even if nearby spike-timing relationships are not the most natural (or causal) way to describe the generative process. For example, consider the effects of random—but slow— variations in a non-constant $\lambda(t)$, as in a DSPP.

Estimating Correlation Functions Correlation functions are typically estimated by constructing *correlograms*: histories of time *differences* between (not necessarily adjacent) spikes. The covariance function is estimated by subtracting an estimate of the correlation of the mean intensity:

$$\begin{aligned}\frac{1}{T}\int \hat{\bar{\lambda}}(t)\hat{\bar{\lambda}}(t-\tau)\,dt &= \frac{1}{TK^2}\int \sum_k s^{(k)}(t)\sum_{k'} s^{(k')}(t-\tau) \\ &= \frac{1}{TK^2}\sum_{kk'}\int s^{(k)}(t)s^{(k')}(t-\tau)\,dt.\end{aligned} \tag{4.59}$$

This is called the *shift* or *shuttle* correction. An estimate may also be constructed in the frequency domain, e.g., through a power spectrum, spectrogram, or coherence (for multiple processes). This is usually based on a Fourier transform of binary-binned spike trains.

Cross-Correlations In the case that we are also measuring the relationships among multiple cells, the techniques are analogous to those for single processes. The *cross-correlogram* estimate of the *cross-correlation* function is

$$R_{s^{(c)}s^{(c')}}(\tau) = \left\langle \frac{1}{T}\int s^{(c)}(t)s^{(c')}(t-\tau)\,dt \right\rangle, \tag{4.60}$$

where c indexes the cells. Accordingly, the shit- or shuttle-corrected correlogram estimate of the cross-variance function is then

$$Q_{s^{(c)}s^{(c')}}(\tau) = \left\langle \frac{1}{T}\int\int (s^{(c)}(t) - \bar{\lambda}^{(c)}(t))(s^{(c')}(t-\tau) - \bar{\lambda}^{(c')}(t-\tau))\,dt \right\rangle. \tag{4.61}$$

As for autocovariograms, structure in a cross-covariogram need not imply thar dependencies between individual spike times are the most natural way to think about the interaction between the processes—DSPPs with shared latents may also give significant cross-covariance structure.

4.6 Point Process Tips

- Always keep in mind that, in general, the mean rate $\bar{\lambda}$ corresponds to the inverse of the mean inter-spike interval. Therefore, to find a mean firing rate, derive $p(\tau)$, find $\bar{\tau} = \mathbb{E}_{p(\tau)}[\tau]$, and then $\bar{\lambda} = 1/\bar{\tau}$.

5 Reinforcement Learning

This section assumes familiarity with reinforcement learning. It's mainly meant as reminders/review of biological RL and specific topics which I've seen on past exams. I deliberately don't go into much detail, as it's not something that's emphasized in the course.

5.1 Classical Conditioning

Classical/Pavlovian conditioning encompasses a variety of different training and testing procedures, but its defining feature (in contrast to instrumental conditioning) is that rewards and punishments are delivered to the the animal/agent independent of its actions. In the classic Pavlovian experiment, dogs are repeatedly fed just after a bell is rung. Eventually, the dogs begin to salivate in response to the sound of the bell, anticipating the arrival of food. In this case, the presentation of food is called the *unconditioned stimulus* and the salivation in response to the sight of food is the *unconditioned response*. The sound of the bell is then called the *conditioned stimulus* and the subsequent salivation the *conditioned response*. In the following sections, we construct models of how an animal acquires the expectation of the delivery of reward in response to stimuli.

5.1.1 The Rescorla-Wagner Rule

We describe the presence or absence of stimulus on a particular trial via the binary variable $u \in \{0, 1\}$, and model the value function (the expected reward) as a linear function of the presence of absence of the stimulus via

$$v = wu, \tag{5.1}$$

, for some weight w. The value of w is learned by minimizing the expected squared error $\left\langle (r - v_w(u))^2 \right\rangle$. Differentiating this expression with respect to w yields the *Rescorla-Wagner rule*:

$$\boxed{w \leftarrow w + \epsilon \underbrace{(r - v)}_{\delta} u}, \tag{5.2}$$

where ϵ is the learning rate and can be interpreted biologically as the associability of the stimulus with the reward. The prediction error δ can be interpreted as the firing rates of dopaminergic cells in the ventral tegmental area (VTA)—more on this in Section 5.1.3. Under the Rescorla-Wagner rule, it's easy to see that w will converge to the expected value of the reward, $\langle r \rangle$, at which point the average value of δ is zero. Given a stimulus that is always paired with reward, w will converge exponentially quickly to the average reward value. This phase of learning is called *acquisition*. If the reward is then never presented again in relationship to the stimulus, the w will then decay to zero exponentially quickly. This is called *extinction*. This relationship is visualized in Figure 5.1.

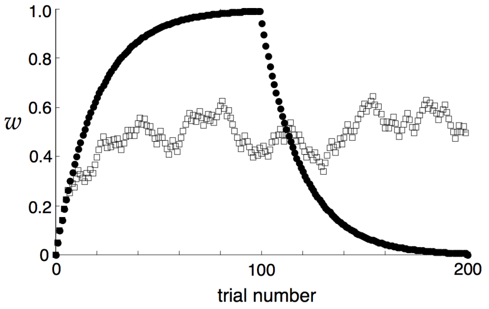


Figure 5.1: Acquisition and extinction in the Rescorla-Wagner rule. The filled circles denote the value of w when the reward is first presented every time the stimulus is, and then when the reward is never given with the stimulus again. The open squares show the path of w when the reward is presented stochastically half the time that the stimulus is presented—here, w fluctuates around $\langle r \rangle = 0.5$.

Blocking *Blocking* is a phenomenon in which two stimuli are presented together before the delivery of reward, but only after the animal has developed an association for one stimulus on its own. Then, when the two stimuli are presented separately, the animal will only display the conditioned response in reaction to the stimulus that was originally associated with reward—it has *blocked* an association from developing with the second stimulus. This effect can be explained with the vector form of the Rescorla-Wagner rule, with $\mathbf{w}, \mathbf{u} \in \mathbb{R}^d$, with $d = 2$ in this case. During the initial training phase, the rule will lead w_1 to converge on $\langle r \rangle$. Then when the second

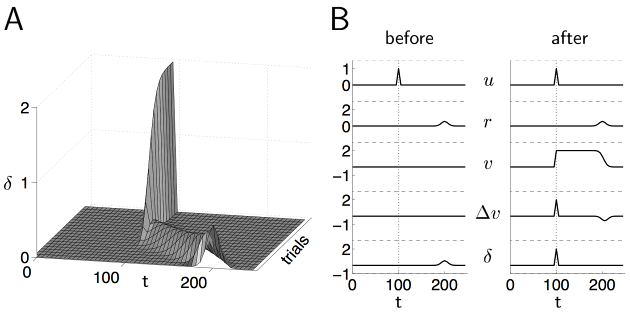


Figure 5.2: (A) The evolution of the TD error across time and trials in an experiment in which the stimulus is presented at time $t = 100$ and a reward is given at time $t = 200$. (B) The relevant values before and after training.

stimulus is presented along with the first, its weight starts at $w_2 = 0$, so the predicted reward is still equal to r: $v = w_1 u_1 + w_2 u_2 = \langle r \rangle$. Then $\delta = 0$, so no further learning takes place.

Overshadowing *Overshadowing* occurs when two stimuli are always presented together in training, but the prediction ends up being shared unequally, in that the weight associated with one stimulus is higher than the other. This can be explained by a generalization of the Rescorla-Wagner rule in which each weight receives its own learning rate. The weight with the higher learning rate reach a higher value than the second weight at convergence.

Secondary Conditioning While quite simple, the Rescorla-Wagner rule can explain a variety of phenomena seen in the learning behavior of animals. One pattern that cannot be reproduced by the Rescorla-Wagner rule, however, is that of *secondary conditioning.* In secondary conditioning, one stimulus is associated with reward, and then an association is learned between a second stimulus and the first. The animal then displays a conditioned response to the second stimulus, even though it has never been paired with the reward. This cannot be modeled with the Rescorla-Wagner rule—in fact, because the reward is never presented with the second stimulus, the delta rule would induce w_2 to become negative (inhibitory conditioning). This is related to the problem of delayed reward, and can be explained by *temporal difference* (TD) learning.

5.1.2 TD-Learning

To measure the reward across a trial, its useful to consider the *return* Z, defined as

$$Z := \sum_{t=0}^{T} \gamma^t r_t, \tag{5.3}$$

where $\gamma \in (0,1)$ is a discount factor. It's useful, then, to redefine the value function as the expected return across a trial, rather than the expected value of the reward at the next time step:

$$v(u_t) := \langle Z \rangle = \left\langle \sum_{t=0}^{T} \gamma^t r_t \right\rangle. \tag{5.4}$$

For a linear value function $v(u_t) := w u_t$, the TD rule is given by

$$\boxed{w_t \leftarrow w_t + \epsilon \delta u_t, \qquad \text{with } \delta = r_t + \gamma v(u_{t+1}) - v(u_t)}. \tag{5.5}$$

We can see that higher values of γ correspond to higher prioritization given to the long-term expectation of future reward, while low values of γ encourage short-sighted behavior. Figure 5.2 shows the evolution of the relevant parameters in an experiment in which the stimulus is presented at time $t = 100$ and a reward is given at time $t = 200$. Over the course of learning, the TD error δ steadily shifts backward in time, ending up at $t = 99$. As it does so, the weight associated with each time step $200, 199, \ldots$ grows, leading to the elevation of the value function from the presentation of the reward on to the delivery of the reward (Figure 5.2B). The value difference $\Delta v_t := v_{t+1} - v_t = \delta_t - r_t$ is negative around $t = 200$ to compensate for the delivery of reward and maintain δ at zero.

Unlike the Rescorla-Wagner rule, TD learning can account for secondary conditioning. In this case, when a second stimulus s_2 is introduced before the first s_1 (which has become associated with the ensuing reward), the positive spike in δ_t at the time that s_1 is presented drives an increase in the value of the weight associated with s_2, thus establishing a positive association between s_2 and the reward through the same process that the association with s_1 was acquired.

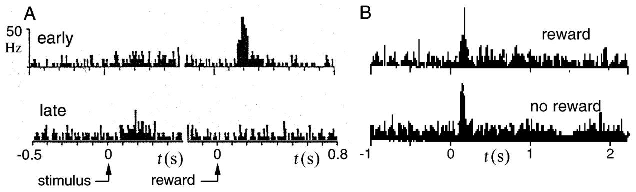


Figure 5.3: Activity of dopaminergic neurons in the VTA obtained from monkeys completing a reaction-time task. (A) The firing rate of a dopaminergic neuron in response to a stimulus followed by a reward early in training (top), and the same neuron late in training (bottom). (B) Dopaminergic responses when a reward is delivered as expected (top) and when it is not (bottom).

5.1.3 Dopamine

The prediction error δ plays a central role in both the Rescorla-Wagner rule and TD learning, and it therefore makes sense to look for a neural representation of this signal. One suggestion for this role are the dopaminergic neurons in the ventral tegmental area (VTA) of the midbrain. There is strong experimental evidence that dopamine is involved in reward learning. Many addictive drugs work by increasing the longevity of the dopamine released onto target structures such as the nucleus accumbens (an area often associated with feelings of pleasure).

In a series of studies performed by Schultz et al. in the 90s, monkeys were trained through instrumental conditioning to response to stimuli such as lights and sounds in order to obtain food and drink rewards. The activities of cells in the VTA were recorded during learning, and examples are plotted in Figure 5.3. In (A), we can see that the increased firing rate moves from the time of the reward to (top) to the time of the stimulus (bottom), just as the prediction error δ does. Similarly, in the top panel of (B), we can see that when a reward is presented as expected, given the conditioned stimulus, there is no change in firing (indicating a prediction error of zero). On the bottom panel of (B), however, there is a decrease in firing rate below the baseline, indicating a negative prediction error. This is strong evidence that the firing rates of dopaminergic neurons in the VTA encode prediction errors.

5.2 Static Action-Choice

In static action-choice tasks, the reward or punishment immediately follows the action taken. *Indirect actor* methods learn a value function and then choose actions based on the expected rewards (ex. DP, Q-learning). *Direct actor* methods seek to optimize the policy by directly maximizing the expected return, usually via gradient ascent (ex. PPO, TRPO).

These ideas can be well-illustrated with the example of a bee foraging for nectar. If the bee follows an indirect actor paradigm, it can learn the expected volume of nectar at each flower it visits via a delta rule, and then base its actions off these estimates.

If the bee follows a direct actor paradigm, the choice of actions is based directly on maximizing the expected average reward, which can be done via gradient ascent. Compared to the indirect actor method, in this case, the direct actor method learns more slowly and is less adaptable if the expected nectar volumes in the environment changes.

5.3 Sequential Action-Choice

In sequential action-choice tasks, unlike static action-choice tasks, rewards may be delayed for several steps after an action is taken. A maze with rewards at the end is a classic example of such an environment. It's in this setting that the usage of dynamic programming and the Markov decision process frameworks that are hallmarks of RL become useful. In methods like policy iteration, we can subdivide the algorithm into the *critic*, which performs policy evaluation (e.g., with TD learning) and the *actor*, which maintains and improves the policy. Neurally, it has been suggested that the dorsal striatum, a part of the basal ganglia, is involved in the selection and sequencing of actions. Terminals of axons projecting from the substantia nigra pars compacta release dopamine onto synapses within the striatum, suggesting they play a gating role. The activity of these dopamine is similar to that of the VTA neurons discussed earlier.

6 Information Theory

6.1 Quantifying Uncertainty

We'd like to quantify the amount of information carried by neural responses about the stimuli that induce them. More formally, we define *information* as the removal of uncertainty about a variable quantity. Consider the sequence of events defined by

$$S \to R \to P(S|R). \tag{6.1}$$

We'd like to know how informative R is about S. For example, S could be identifiable with complete certainty from the value of R, or R could carry no information about S, i.e.,

$$P(S|R) = [0, 0, 1, 0, \dots, 0]^{\mathsf{T}} \qquad \text{(complete information)} \tag{6.2}$$

$$P(S|R) = \left[\frac{1}{M}, \frac{1}{M}, \dots, \frac{1}{M}\right]^{\mathsf{T}} \qquad \text{(no information)}. \tag{6.3}$$

These probabilities also depend on $P(S)$, however. To start our analysis, we'll consider the uncertainty inherent in a probability distribution, termed the *entropy*. Let $S \sim P(S)$. The entropy is the minimum number of bits needed, on average, to specify the value that S takes, assuming $P(S)$ is known. Equivalently, this is the minimum average number of yes/no questions needed to guess S. Note that for the most part we'll be working with discrete distributions here, as formally extending most information theoretic principles to continuous distributions is non-trivial.

6.1.1 Entropy and Conditional Entropy

Suppose there are M equiprobable stimuli: $P(s_m) = p = 1/M$. To specify which stimulus appears on a given trial, we would need to assign each a binary number. The number of bits B_s required is

$$\begin{aligned} 2^{B_s} \geq M \Rightarrow B_s &\leq \log_2 M \\ &= -\log_2 \frac{1}{M}. \end{aligned} \tag{6.4}$$

Now suppose we code N such stimuli, drawn iid, at once. We get

$$\begin{aligned} B_N &\leq \log_2 M^N \\ &\to -N \log_2 \underbrace{\frac{1}{M}}_{=p} = -\sum_s \log_2 p \quad \text{as } N \to \infty \\ \Rightarrow B_s &\to -\log_2 p \text{ bits.} \end{aligned} \tag{6.5}$$

This is called block coding. It is useful for extracting theoretical limits. The nervous system is unlikely to use block codes, may in space.

Now suppose the stimuli are not equiprobable. Write $P(s_m) = p_m$. Then

$$P(S_1, S_2, \dots, S_N) = \prod_m p_m^{n_m}, \tag{6.6}$$

where n_m is the number of $S_i = s_m$. As $N \to \infty$ only "typical" sequences, with $n_m = p_m N$, have non-zero probability of occurring, and they are all equally likely *[TM: not sure why they're all 'equally likely']*. This is called the Asymptotic Equipartition Property (AEP). Thus, eq. 6.5 gives

$$\begin{aligned} B_N &\to -\log_2 \prod_m p_m^{n_m} = -\sum_m n_m \log_2 p_m \\ &= -\sum_m p_m N \log_2 p_m = -N \underbrace{\sum_m p_m \log_2 p_m}_{-\mathbf{H}[s]}. \end{aligned} \tag{6.7}$$

Then $\mathbf{H}[S] = \mathbb{E}\left[-\log_2 P(S)\right]$, also written $\mathbf{H}[P(S)]$ is the *entropy* of the stimulus distribution. Note that we could also derive the expression for entropy via the law of large numbers. From eq. 6.5, we have

$$-\frac{1}{N} \log P(S_1, S_2, \dots, S_N) = -\frac{1}{N} \log_2 \prod_i P(S_i) = -\frac{1}{N} \sum_i \log_2 P(S_i) \to^{N\to\infty} \mathbb{E}\left[-\log_2 P(S_i)\right]. \tag{6.8}$$

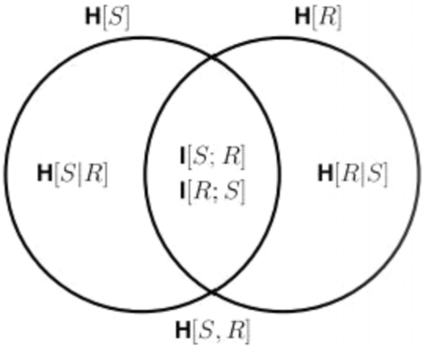


Figure 6.1: Mutual information diagram.

Entropy is a measure of the "available information" in the stimulus ensemble. Now suppose we measure a particular response r which depends on the stimulus according to $P(R|S)$. How uncertain is the stimulus once we know r? Bayes rule gives us

$$P(S|r) = \frac{P(r|S)P(S)}{\sum_s P(r|s)P(s)}, \tag{6.9}$$

so we can write

$$\mathbf{H}[S|r] = -\sum_s P(s|r) \log_2 P(s|r). \tag{6.10}$$

And so the *average* uncertainty in S for $r \sim P(R) = \sum_s P(R|s)P(s)$ is then

$$\begin{aligned} \mathbf{H}[S|R] &= \sum_r P(r)\mathbf{H}[S|r] = \sum_r P(r)\left[-\sum_s P(s|r)\log_2 P(s|r)\right] \\ &= -\sum_{s,r} P(s,r)\log_2 P(s|r). \end{aligned} \tag{6.11}$$

This is called *conditional entropy* of S on R. It is easy to see that:

1. $\mathbf{H}[S|R] \leq \mathbf{H}[S]$ (conditioning cannot increase uncertainty)
2. $\mathbf{H}[S|R] = \mathbf{H}[S,R] - \mathbf{H}[R]$
3. $\mathbf{H}[S|R] = \mathbf{H}[S] \Leftrightarrow S \perp\!\!\!\perp R$.

6.1.2 Mutual Information

A natural definition of the average information gained about S from R is

$$\mathbf{I}[S;R] := \mathbf{H}[S] - \mathbf{H}[S|R]. \tag{6.12}$$

This is the average *mutual information* between S and R. It measures the reduction in uncertainty about S given R. It follows from the definition that

$$\begin{aligned} \mathbf{I}[S;R] &= \sum_s P(s)\log\frac{1}{P(s)} - \sum_{s,r} P(s,r)\log\frac{1}{P(s|r)} \\ &= \sum_{s,r} P(s,r)\log\frac{1}{P(s)} + \sum_{s,r} P(s,r)\log P(s|r) \\ &= \sum_{s,r} P(s,r)\log\frac{P(s|r)}{P(s)} \\ &= \boxed{\sum_{s,r} P(s,r)\log\frac{P(s,r)}{P(s)P(r)}} \\ &= \mathbf{I}[R;S]. \end{aligned} \tag{6.13}$$

This symmetry suggests a Venn-like diagram (Figure 6.1). All of the implied additive and equality relationships hold for two variables, but this representation doesn't hold for more than two. (If you add a third variable, the region of overlap could have negative area—more later).

Kullback-Leibler (KL) Divergence The *KL divergence* is a useful information theoretic quantity that measures the difference between two distributions:

$$\begin{aligned}\mathsf{KL}[P(S)\|Q(S)] &= \sum_s P(s)\log\frac{P(s)}{Q(s)} && (6.14)\\ &= \underbrace{\sum_s P(s)\log\frac{1}{Q(s)}}_{\text{cross entropy}} - \mathbf{H}[P] && (6.15)\end{aligned}$$

The cross entropy can be thought of as being the code length drawn from the wrong distribution. The KL divergence can be thought of as the excess cost (in bits or nats) of building the code according to Q when the true distribution is P. To see that the KL is always non-negative, we can write

$$\begin{aligned}-\mathsf{KL}[P\|Q] &= \sum_s P(s)\log\frac{Q(s)}{P(s)}\\ &\le \log\sum_s P(s)\frac{Q(s)}{P(s)} \qquad \text{(by Jensen)} && (6.16)\\ &= \log\sum_s Q(s) = \log 1 = 0.\end{aligned}$$

Therefore $\mathsf{KL}[P\|Q] \ge 0$, with equality if and only if $P = Q$.

Importantly, the mutual information can be expressed as a KL divergence as follows:

$$\mathbf{I}[S;R] = \sum_{s,r} P(s,r)\log\frac{P(s,r)}{P(s)P(r)} = \mathsf{KL}[P(S,R)\|P(S)P(R)]. \qquad (6.17)$$

Therefore, we can say that (1) the mutual information is always non-negative, $\mathbf{I}[S;R] \ge 0$, and (2) conditioning never increases entropy:

$$\mathbf{I}[S;R] := \mathbf{H}[S] - \mathbf{H}[S|R] \ge 0 \Rightarrow \mathbf{H}[S|R] \le \mathbf{H}[S]. \qquad (6.18)$$

6.2 Properties of Mutual Information and Entropy

6.2.1 Multiple Responses

Two responses to the same stimulus, R_1 and R_2, may provide either more or less information jointly than independently:

$$I_{12} := \mathbf{I}[S;R_1;R_2] = \mathbf{H}[R_1,R_2] - \mathbf{H}[R_1,R_2|S]. \qquad (6.19)$$

We then have

$$\begin{aligned} R_1 \perp\!\!\!\perp R_2 &\Rightarrow \mathbf{H}[R_1,R_2] = \mathbf{H}[R_1] + \mathbf{H}[R_2] && (6.20)\\ R_1 \perp\!\!\!\perp R_2|S &\Rightarrow \mathbf{H}[R_1,R_2|S] = \mathbf{H}[R_1|S] + \mathbf{H}[R_2|S], && (6.21)\end{aligned}$$

from which the following relationships can be deduced: Redundancy occurs when the information provided by R_1 and R_2 about S has an overlap, and so the combined mutual information is less than the sum of the individual mutual informations. Synergy is a case of explaining away in the corresponding graphical model. We also note that $I_{12} > \max\{I_1, I_2\}$, which implies that the second response cannot destroy information.

$R_1 \perp\!\!\!\perp R_2$	$R_1 \perp\!\!\!\perp R_2\|S$		
no	yes	$I_{12} < I_1 + I_2$	redundant
yes	yes	$I_{12} = I_1 + I_2$	independent
yes	no	$I_{12} > I_1 + I_2$	synergistic
no	no	?	any of the above

6.2.2 The Data Processing Inequality

Suppose $S \to R_1 \to R_2$ form a Markov chain; that is, $R_2 \perp\!\!\!\perp S|R_1$. Then

$$\begin{aligned} P(R_2, S|R_1) &= P(R_2|R_1)P(S|R_1) \\ \Rightarrow \frac{P(R_2, S|R_1)}{P(R_2|R_1)} &= P(S|R_1) \\ \Rightarrow P(S|R_1, R_2) &= P(S|R_1). \end{aligned} \tag{6.22}$$

Thus,

$$\begin{aligned} \mathbf{H}[S|R_2] &\geq \mathbf{H}[S|R_1, R_2] = \mathbf{H}[S|R_1] \\ \Rightarrow \mathbf{I}[S; R_2] &\leq \mathbf{I}[S; R_1] \qquad (\mathbf{I}[X;Y] \coloneqq \mathbf{H}[X] - \mathbf{H}[X|Y]) \end{aligned} \tag{6.23}$$

This is the *data processing inequality.* It implies that any computation based on R_1 that does not have separate access to S cannot add information (in the Shannon sense) about the world. Equality holds if and only if $S \to R_2 \to R_1$ as well. In this case, R_2 is called a *sufficient statistic* for S (if that is the causal relationship; otherwise R_1 is the sufficient statistic). This is related to D-separation in graphical modeling.

6.2.3 Entropy Rate

So far we have discussed S and R as single (or iid) random variables. But real stimuli and responses form a time series. Let $\mathcal{S} = \{S_1, S_2, \dots\}$ form a stochastic process. We have

$$\begin{aligned} \mathbf{H}[S_1, S_2, \dots, S_n] &= \mathbf{H}[S_n|S_1, S_2, \dots, S_{n-1}] + \mathbf{H}[S_1, S_2, \dots, S_{n-1}] \\ &= \underbrace{\mathbf{H}[S_n|S_1, S_2, \dots, S_{n-1}] + \mathbf{H}[S_{n-1}|S_1, S_2, \dots, S_{n-2}] + \cdots + \mathbf{H}[S_1]}_{n \text{ terms}}. \end{aligned} \tag{6.24}$$

(Note that products of probabilities translate to sums of entropies, as the entropies exist in log space.) The *entropy rate* of $\mathcal{S}$ is defined as

$$\mathbf{H}[\mathcal{S}] \coloneqq \lim_{n\to\infty} \frac{1}{n} \mathbf{H}[S_1, S_2, \dots, S_n], \tag{6.25}$$

or alternatively as

$$\mathbf{H}[\mathcal{S}] \coloneqq \lim_{n\to\infty} \mathbf{H}[S_n|S_1, S_2, \dots, S_{n-1}]. \tag{6.26}$$

If $S_i \sim^{iid} P(s)$, then $\mathbf{H}[\mathcal{S}] = \mathbf{H}[S]$ (i.e., $\mathbf{H}[S_i|S_j] = \mathbf{H}[S_i]$). If $\mathcal{S}$ is Markov (and stationary) then $\mathbf{H}[\mathcal{S}] = \mathbf{H}[S_n|S_{n-1}]$.

6.2.4 Continuous Random Variables

The discussion so far has involved discrete S and R. Now let $S \in \mathbb{R}$ with density $p(s)$, and suppose we discretize with length Δs. Then the entropy is

$$\begin{aligned} \mathsf{H}_\Delta[S] &= -\sum_i p(s_i)\Delta s \log[p(s_i)\Delta s] \\ &= -\sum_i p(s_i)\Delta s(\log p(s_i) + \log \Delta s) \\ &= -\sum_i p(s_i)\Delta s \log p(s_i) - \log[\Delta s] \underbrace{\sum_i p(s_i)\Delta s}_{\to 1 \text{ as } \Delta s \to 0} \\ &= -\sum_i p(s_i)\Delta s \log p(s_i) - \log \Delta s \\ &\to -\int p(s) \log p(s)\, ds + \infty \qquad \text{as } \Delta s \to 0. \end{aligned} \tag{6.27}$$

The issue, fundamentally, is that with a finite number of bits, you can't exactly encode any $S \in \mathbb{R}$. We define the *differential entropy* by ignoring the lingering infinity:

$$h(S) \coloneqq -\int p(s) \log p(s)\, ds. \tag{6.28}$$

Note that unlike the discrete entropy, $h(S)$ can be negative, as well as $\pm\infty$.

Other information theoretic quantities can be defined similarly in the continuous case. The *conditional* differential entropy is

$$h(S|R) := -\int p(s,r)\log p(s|r)\,ds\,dr, \tag{6.29}$$

and, like the differential entropy itself, may be poorly behaved. Interestingly, however, the mutual information is well-defined, as

$$\begin{aligned}\mathsf{I}_\Delta[S;R] &= \mathsf{H}_\Delta[S] - \mathsf{H}_\Delta[S|R] \\ &= -\sum_i \Delta s p(s_i)\log p(s_i) - \log\Delta s - \int p(r)\left(-\sum_i \Delta s p(s_i|r)\log p(s_i|r) - \log\Delta s\right) dr \\ &\to h(S) - h(S|R) \qquad \text{as } \Delta s \to 0,\end{aligned} \tag{6.30}$$

where the good behavior is due to the cancellation of the $\log\Delta s$s in the second line. To be even more well-behaved in a formal sense, we could instead define the differential mutual information as a KL divergence between the distribution in question and a uniform distribution—all KL divergences are well-behaved in the continuous case.

6.2.5 Maximum Entropy Distributions

We have that $\mathbf{H}[R_1,R_2] \le \mathbf{H}[R_1]+\mathbf{H}[R_2]$, with equality if and only if $R_1 \perp\!\!\!\perp R_2$. If we let $\mathbb{E}_p\left[f(s)\right] = \int p(s)f(s)\,ds$ for some function f, we'd like to know which distribution $p(s)$ has the maximum entropy possible given the constraint on the expectation. To find this out, we can use Lagrange multipliers and variational derivatives:

$$\mathcal{L} = \int p(s)\log p(s)\,ds - \lambda_0\left[\int p(s)\,ds - 1\right] - \lambda_1\left[\int p(s)f(s)\,ds - a\right] \tag{6.31}$$

$$\frac{\delta\mathcal{L}}{\delta p(s)} = 1 + \log p(s) - \lambda_0 - \lambda_1 f(s) = 0 \tag{6.32}$$

$$\Rightarrow \log p(s) = \lambda_0 + \lambda_1 f(s) - 1 \tag{6.33}$$

$$\Rightarrow p(s) = \frac{1}{Z}e^{\lambda_1 f(s)}, \tag{6.34}$$

where $Z = \exp(1-\lambda_0)$. The constants λ_0 and λ_1 can be found by solving the constraint equations. Then,

$$f(s) = s \Rightarrow p(s) = \frac{1}{Z}e^{\lambda_1 s} \qquad \text{Exponential (need } p(s) = 0 \text{ for } s < T) \tag{6.35}$$

$$f(s) = s^2 \Rightarrow p(s) = \frac{1}{Z}e^{\lambda_1 s^2} \qquad \text{Gaussian.} \tag{6.36}$$

In other words, when we're looking for the first moment, the maximum entropy distribution is an exponential distribution, and when we're looking for the second moment (without constraint on the first moment), the maximum entropy distribution is Gaussian. Both results together imply that the maximum entropy point process (for a fixed mean arrival rate) is a homogeneous Poisson—independent, exponentially distributed ISIs.

6.3 Channel Coding

We now consider the conditional $P(R|S)$ which defines the *channel* linking S to R:

$$S \xrightarrow{P(R|S)} R. \tag{6.37}$$

The mutual information

$$\mathsf{I}[S;R] = \mathsf{KL}[P(S,R)\|P(S)P(R)] = \sum_{s,r} \overbrace{P(s,r)}^{P(s)P(r|s)} \log\frac{\overbrace{P(s,r)}^{P(s)P(r|s)}}{P(s)P(r)} = \sum_{s,r} P(s)P(r|s)\log\frac{P(r|s)}{P(r)} \tag{6.38}$$

depends on marginals $P(s)$ and $P(r) = \sum_s P(r|s)P(s)$ as well and thus is unsuitable to characterize the conditional alone. Instead, we characterize the channel by its *capacity* C, defined as

$$\mathsf{C}_{R|S} := \sup_{P(s)} \mathsf{I}[S;R]. \tag{6.39}$$

In other words, the capacity gives the theoretical maximum information that can be transmitted through the channel given all possible source distributions. In other words, the capacity is given by choice for $P(S)$, given the conditional distribution $P(R|S)$, that maximizes the mutual information between R and S. Clearly, this is limited by the properties of the noise.

As an aside, we define spike count or *noise correlation* to be the correlation between fluctuations in responses to a repeated stimulus, and *signal correlation* to be the correlation between two cell's mean responses to different stimuli.

6.3.1 The Joint Source-Channel Coding Theorem (JSCT)

This is a remarkable result of central importance to information theory. Consider the following framework

$$S \xrightarrow{\text{encoder}} \tilde{S} \xrightarrow[\mathsf{C}_{R|\tilde{S}}]{\text{channel}} R \xrightarrow{\text{decoder}} \hat{T}. \tag{6.40}$$

Any source ensemble S with entropy $\boldsymbol{H}[S] < \mathsf{C}_{R|\tilde{S}}$ can be transmitted perfectly (in sufficiently long blocks) with $P_{error} \to 0$. The proof is beyond our scope, but some of the key ideas involved are block coding, error correction, joint typicality, and random codes.

This leads to the *channel coding problem*, which seeks to find the *encoding* $P(\tilde{S}|S)$ (this may be deterministic) that maximizes $\mathbf{I}[S;R]$, given channel $P(R|\tilde{S})$ and source $P(S)$. By the data processing inequality (eq. 6.23) and the definition of capacity (eq. 6.39), we have

$$\mathbf{I}[S;R] \leq \mathbf{I}[\tilde{S};R] \leq \mathsf{C}_{R|\tilde{S}}. \tag{6.41}$$

By the JSCT, then, equality can be achieved (in the limit of increasing block size). Thus, $\mathbf{I}[\tilde{S};R]$ should saturate $\mathsf{C}_{R|\tilde{S}}$. The *Blahut Arimoto algorithm* is a method to find the $P(\tilde{S})$ that saturates $\mathsf{C}_{R|\tilde{S}}$ for a general discrete channel.

6.3.2 The Blahut-Arimoto Algorithm

Given a channel characterized by the conditional distribution $P(R|S)$, we wish to find a source distribution $P(S)$ that maximizes the mutual information $I(R;S)$. First, we show that

$$I(R;S) \geq \sum_{s,r} P(s)P(r|s) \log \frac{Q(s|r)}{P(s)} \tag{6.42}$$

for any conditional distribution $Q(S|R)$.

Proof. Call the expression on the right hand side of the inequality in eq. 6.42 $\tilde{I}(R;S)$. We can subtract $\tilde{I}(R;S)$ from $I(R;S)$:

$$\begin{aligned}
I(R;S) - \tilde{I}(R;S) &= \sum_{s,r} P(s)P(r|s) \log \frac{P(r|s)}{P(r)} - \sum_{s,r} P(s)P(r|s) \log \frac{Q(s|r)}{P(s)} && (6.43)\\
&= \sum_{r,s} P(r) \frac{P(s)P(r|s)}{P(r)} \log \frac{P(s|r)/P(r)}{Q(s|r)/P(s)} && (6.44)\\
&= \sum_{r} P(r) \sum_{s} \frac{P(s)P(r|s)}{P(r)} \log \frac{P(s)P(r|s)/P(r)}{Q(s|r)} && (6.45)\\
&= \sum_{r} P(r) \sum_{s} P(s|r) \log \frac{P(s|r)}{Q(s|r)} && (6.46)\\
&= \sum_{r} P(r) D_{KL}[P(S|r)||Q(S|r)] && (6.47)\\
&\geq 0, && (6.48)
\end{aligned}$$

where the inequality is due to the non-negativity of the KL divergence. Note that equality is achieved when $Q(S|R) = P(S|R)$. □

We can use this result to derive an iterative algorithm to find the optimal $P(S)$. We'll do so using a method reminiscent of EM: alternating a saturation of the bound on the right hand side of eq. 6.42 with respect to Q and to $P(S)$.

Algorithm 1: Blahut-Arimoto

```
Initialize iteration counter $k = 0$, $P^0(s)$ (e.g., uniformly),
while not converged do
    update $Q(S|R)$:
        $Q^{k+1}(s|r) \leftarrow \frac{P(r|s)P^k(s)}{\sum_{s'} P(r|s')P^k(s')}$
    update $P(S)$:
        $P^{k+1}(s) \leftarrow \frac{\phi(s)}{\sum_{s'} \phi(s')}$, where $\phi(s) = \exp\left(\sum_r P(r|s) \log Q^{k+1}(s|r)\right)$
    check $\tilde{I}(Q^{k+1}, P^{k+1})$ for convergence
    $k \leftarrow k + 1$
end
```

we can see that $\tilde{I}$ is maximized with respect to Q when

$$Q(s|r) = P(s|r) = \frac{P(r|s)P(s)}{\sum_{s'} P(r|s')P(s')}. \tag{6.49}$$

To maximize $\tilde{I}$ with respect to $P(s)$, we can set up a Lagrangian:

$$\mathcal{L} = \tilde{I} + \lambda \left(\sum_{s'} P(s') - 1 \right), \tag{6.50}$$

where λ is a Lagrange multiplier preserving the measure of $P(s)$. We can then take the variational derivative to find the optimal $P(s)$:

$$\begin{aligned}
\frac{\delta \mathcal{L}}{\delta P(s)} &= \frac{\delta}{\delta P(s)} \left[\sum_{s',r} P(s')P(r|s') \log \frac{Q(s'|r)}{P(s')} + \lambda \left(\sum_{s'} P(s') - 1 \right) \right] && (6.51)\\
&= \sum_r P(r|s) \log \frac{Q(s|r)}{P(s)} + \lambda && (6.52)\\
&= \sum_r P(r|s) \log Q(s|r) - \sum_r P(r|s) \log P(s) + \lambda && (6.53)\\
&= \sum_r P(r|s) \log Q(s|r) - \log P(s) \underbrace{\sum_r P(r|s)}_{=1} + \lambda && (6.54)\\
&= \sum_r P(r|s) \log Q(s|r) - \log P(s) + \lambda = 0 && (6.55)\\
\Rightarrow \log P(s) &= \sum_r P(r|s) \log Q(s|r) + \lambda && (6.56)\\
\Rightarrow P(s) &= e^{\lambda} e^{\sum_r P(r|s) \log Q(s|r)} && (6.57)\\
\Rightarrow P(s) &= \frac{\exp\left(\sum_r P(r|s) \log Q(s|r)\right)}{\sum_{s'} \exp\left(\sum_r P(r|s') \log Q(s'|r)\right)}. && (6.58)
\end{aligned}$$

This suggests the iterative process summarized in Algorithm 1. From the analysis above, we have that

$$\begin{aligned}
Q^{k+1} &= \operatorname*{argmax}_Q \tilde{I}(Q, P^k) \quad \text{and} && (6.59)\\
P^{k+1} &= \operatorname*{argmax}_P \tilde{I}(Q^{k+1}, P), && (6.60)
\end{aligned}$$

so we know that each iteration will perform coordinate ascent on $\tilde{I}$, analogously to EM:

$$I(Q^k, P^{k-1}) = \tilde{I}(Q^k, P^{k-1}) \leq \tilde{I}(Q^k, P^k) \leq \tilde{I}(Q^{k+1}, P^k) = I(Q^{k+1}, P^k). \tag{6.61}$$

Finally, to show that this process will converge to a *unique* maximum, we need to show that $\tilde{I}$ is concave. Specifically, we need to show that for a fixed $P(S)$, $\tilde{I}$ is concave in $Q(S|R)$, and for a fixed $Q(S|R)$, $\tilde{I}$ is concave with respect to $P(s)$. For the former, we can simply observe from the form of $\tilde{I}$ (eq. 6.42) that $\log Q(S|R)$ is concave, so for fixed P, $\tilde{I}$ is concave with respect to Q. Similarly, as $P \log \frac{1}{P}$ is concave, $\tilde{I}$ is concave with respect to P when Q is fixed.

7 Encoding Models

The central question asked by the study of neural encoding is this: How does the brain encode the salient information in a stimulus into a pattern of neural firing. The relationship between a sensory stimulus $s(t)$ and the neural response $r(t)$ is represented diagrammatically in Figure 7.1. In fact, this diagram hints at two inverse problems: *encoding*, which seeks to model the neural response $\hat{r}(t)$ given a stimulus and *decoding*, which seeks to reconstruct a stimulus $\hat{s}(t)$ given the neural response. In this section, we concern ourselves with the former, and focus primarily on visual stimuli and responses.

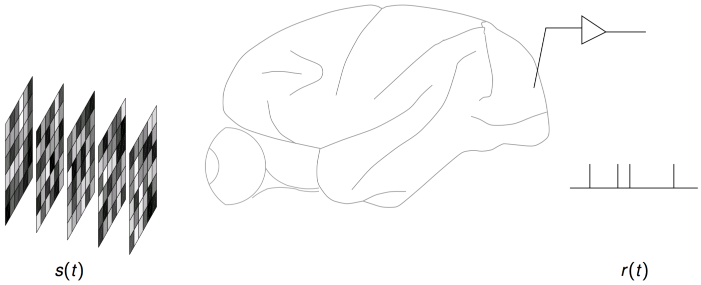


Figure 7.1: The relationship at the center of study in neural encoding and decoding.

The goal then, more formally, is to estimate $p(spike \mid s, H)$ (or $\lambda(t|s[0,t), H(t))$) from data. The naïve approach is to attempt to measure $p(spike, H|s)$ directly for every setting of s. This isn't feasible, however—there is generally too little data and there are too many potential inputs. Instead, we'll estimate some functional[2] $F[p]$ (e.g., mutual information) instead, and fit parameterized models.

Most neurons communicate using action potentials—these are statistically described using a point process, as described earlier:

$$P(spike \in [t, t+dt)) = \lambda(t|H(t),\, s(t),\, \nu(t))\, dt, \tag{7.1}$$

where $\nu(t)$ represents the activity of the network. To fully model the response we need to identify the rate λ. In general, this depends on the spike history $H(t)$ and network activity. We have three possible options:

1. Ignore the history dependence and take the network activity as a source of “noise" (i.e., assume firing is an inhomogeneous Poisson or Cox process, conditioned on the stimulus).
2. Average multiple trials to estimate the mean intensity (PSTH):

$$\bar{\lambda}(t,\, s(t)) = \lim_{N\to\infty} \frac{1}{N} \sum_n \lambda(t|H_n(t), s(t), \nu_n(t)), \tag{7.2}$$

   and try to fit this.
3. We can attempt to capture the history and network effects in simple models.

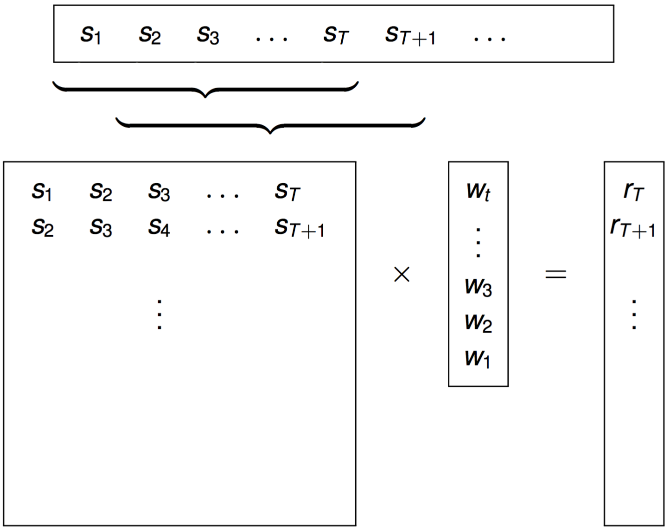
7.1 Linear Models

7.1.1 The Spike-Triggered Average

We begin with the simplest type of model—a linear one. Given a movie composed of a series of stimulus frames presented to an animal and a time-aligned series of spikes in response, we can average the stimulus frames in a certain window of time preceding each spike as a simple estimate of the average stimulus that induces a particular neuron to fire. This is called the *spike-triggered average* (STA)—the process used to calculate it can be visualized in Figure 7.3A. The STA can be interpreted from both a decoding and encoding perspective. From a decoding perspective, it is a way of constructing the mean stimulus given a spike; in other words, the mean of $P(s|r=1)$. From an encoding perspective, the STA can be used as a *predictive filter* to predict the neural response. In other words, we can interpret the firing rate of the neuron at time t as being generated via the convolution

$$r(t) = \int_0^T s(t-\tau)w(\tau)d\tau, \tag{7.3}$$

[2] *Functional* refers to a function/operation whose input is a time series and whose output is a time series.



s_1 s_2 s_3 . . . s_T s_{T+1} . . .
s_1 s_2 s_3 . . . s_T
s_2 s_3 s_4 . . . s_{T+1}
⋮
×
w_t
⋮
w_3
w_2
w_1
=
r_T
r_{T+1}
⋮

Figure 7.2: A visualization of the linear regression used to compute the STA.

where w is the STA filter and T is the window size. We can also rewrite this convolution as a matrix multiplication with a *design matrix* S composed of shifted windows of the stimulus (see Figure 7.2):

$$S\mathbf{w} = \mathbf{r}, \tag{7.4}$$

where $\mathbf{r}$ is a vector containing $r_T, r_{T+1}, \ldots$. Also note that $\mathbf{w}$ is flipped (for the convolution–see Figure 7.2). We can solve this via linear regression, yielding a closed-form expression for the STA:

$$\mathbf{w} = (\underbrace{S^\mathsf{T} S}_{\Sigma_{SS}})^{-1} \underbrace{S^\mathsf{T}\mathbf{r}}_{\text{“unwhitened" STA}}, \tag{7.5}$$

where Σ_{SS} is the stimulus covariance matrix—pre-multiplying $S^\mathsf{T}\mathbf{r}$ with its inverse “whitens"" the STA, accounting for a non-normalized stimulus distribution.

As $r(t)$ is modeled as the result of a convolution, we can also use the Fourier transform and the convolution theorem to construct $\mathbf{w}$:

$$W(\omega) = \frac{S(\omega)R(\omega)}{|S(\omega)|^2}, \tag{7.6}$$

where the Fourier transform of the stimulus $s(\omega)$ usually becomes diagonal in the Fourier basis. This form is usually referred to as a *Wiener filter.*

An important property of the STA is that is an unbiased estimator of the true filter for a spherical (Gaussian) stimulus distribution. This is known as *Bussgang's Theorem.* Stated more formally, Bussgang's Theorem is as follows:

Theorem 1. *If we have samples $\{\mathbf{x}_i, y_i\}$, where y_i is a random variable whose expectation is given by $\mathbb{E}[y_i|\mathbf{x}_i] = f(\mathbf{w}\cdot\mathbf{x}_i)$, then the cross-correlation $\sum_i y_i\mathbf{x}_i$ (i.e., the "spike-triggered average" if y_i is binary) provides an unbiased estimate of $\alpha\mathbf{w}$ (i.e., $\mathbf{w}$ times an unknown constant α if:*

i. *$P(\mathbf{x})$ is spherically symmetric, where we define spherical symmetry to mean that*

$$\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n,\ ||\mathbf{x}_1|| = ||\mathbf{x}_2|| \Rightarrow P(\mathbf{x}_1) = P(\mathbf{x}_2). \tag{7.7}$$

ii. *$\mathbb{E}[y\mathbf{x}] \neq \mathbf{0}$ (i.e., the expected STA is not zero).*

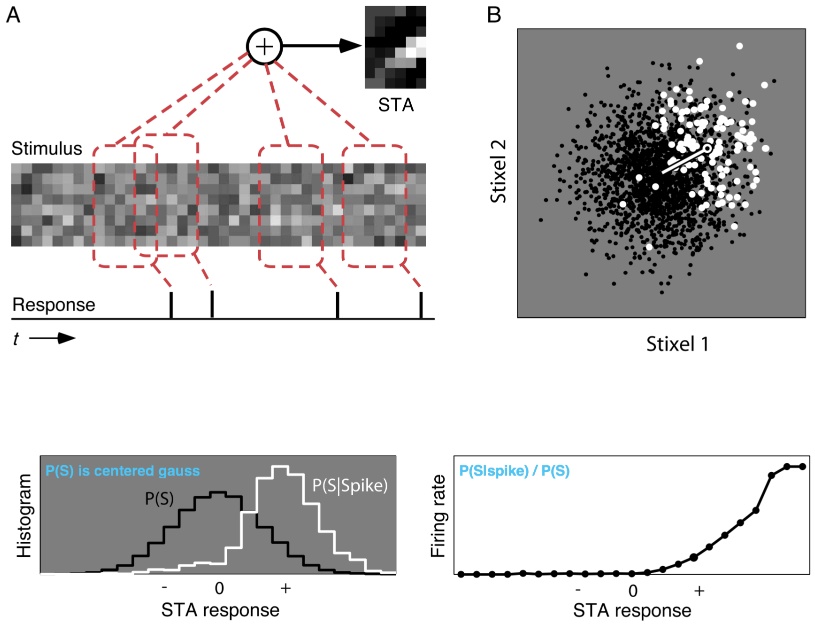


Figure 7.3: Visualization of the STA for a white noise stimulus. (A) The STA is the average of the aggregated stimulus a set window-length of time before each spike. (B) The dots are stimulus values in 2D space—the black dots are those that did not trigger a spike, and the white dots are those that did. The white vector from the center is the STA (note that it maps to the center of the white cloud of points). The probability of spiking is proportional to the projection of a stimulus point onto this vector. By rotational symmetry, the expected value of the spike-inducing (white dot) stimuli lies along this vector, thus making it an unbiased estimate of the maxiumum likelihood value. (Bottom Left) A histogram of all the stimulus values filtered by the STA $P(S)$ (a normal distribution centered at zero) and the spike-triggered stimuli $P(S|\text{spike})$—there is a noticeably higher response for stimuli that induced a spike. (Bottom Right) The resulting firing rate for an LN neuron (more on this below). If $P(\text{spike}) = 1$, then $P(\text{spike}|S) = P(S|\text{spike})/P(S)$. The firing rate curve is thus obtained by dividing the two histograms on the bottom left.

Proof. We want $\mathbb{E}\left[\sum_i y_i \mathbf{x}_i\right] = \alpha \mathbf{w}$. We can write

$$\begin{aligned}
\mathbb{E}\left[\sum_i y_i \mathbf{x}_i\right] &= \sum_i \mathbb{E}\left[y_i \mathbf{x}_i\right] && (7.8)\\
&= \sum_i \left[\sum_{j,k} y_j \mathbf{x}_k P(y_j, \mathbf{x}_k)\right] && (7.9)\\
&= \sum_i \left[\sum_{j,k} y_j \mathbf{x}_k P(y_j|\mathbf{x}_k) P(\mathbf{x}_k)\right] && (7.10)\\
&= \sum_i \left[\sum_k \mathbf{x}_k P(\mathbf{x}_k) \underbrace{\sum_j y_j P(y_j|\mathbf{x}_k)}_{\mathbb{E}[y_j|\mathbf{x}_k]}\right] && (7.11)\\
&= \sum_i \left[\sum_k \mathbf{x}_k P(\mathbf{x}_k) f(\mathbf{w}\cdot\mathbf{x}_k)\right]. && (7.12)
\end{aligned}$$

Then by spherical symmetry, we observe that every $\mathbf{x}_k$ in the sum must have a corresponding $\mathbf{x}_{k'}$ equidistant from the origin (and thus with equal probability) that is symmetric about the STA $\mathbf{w}$. Their sum is then a scaled version of $\mathbf{w}$. That is, $\mathbf{x}_k + \mathbf{x}_{k'} = \beta\mathbf{w}$ for some $\beta > 0$. We have

$$\mathbb{E}\left[\sum_i y_i \mathbf{x}_i\right] = \sum_i \left[\sum_k \mathbf{x}_k P(\mathbf{x}_k) f(\mathbf{w}\cdot\mathbf{x}_k)\right] \tag{7.13}$$

$$= \sum_i \left[\sum_k \mathbf{x}_k P(\mathbf{x}_k) f(\mathbf{w}\cdot\mathbf{x}_k) + \sum_{k'} \mathbf{x}_{k'} P(\mathbf{x}_{k'}) f(\mathbf{w}\cdot\mathbf{x}_{k'})\right] \tag{7.14}$$

$$= \sum_i \left[\sum_{k,k'} (\mathbf{x}_k + \mathbf{x}_{k'}) P(\mathbf{x}_k) f(\mathbf{w}\cdot\mathbf{x}_k)\right] \tag{7.15}$$

$$= \sum_i \left[\sum_k P(\mathbf{x}_k) f(\mathbf{w}\cdot\mathbf{x}_k) \beta\mathbf{w}\right] \tag{7.16}$$

$$= \beta\left[\sum_{i,k} P(\mathbf{x}_k) f(\mathbf{w}\cdot\mathbf{x}_k)\right]\mathbf{w} \tag{7.17}$$

$$= \alpha\mathbf{w}, \tag{7.18}$$

as desired. □

If data is elliptically distributed and not spherical, it can be whitened—the resulting linear regression weights (eq. 7.5) are unbiased. However, linear weights are *not* necessarily maximum likelihood (or otherwise optimal), even for spherical/elliptical stimulus distributions. They may also be biased for general stimuli (binary/uniform or natural).

7.1.2 Limitations

The (whitened) STA gives the minimum squared-error linear model. However, as expected, there are issues. First, as is frequently true in the case of linear regression, there is the potential for overfitting and the need for regularization. The standard methods typically used in regression can be applied here. Second, this framework does nothing to prevent the model from predicting negative firing rates, though this can be ameliorated by re-framing the predictions as relative to some background or baseline firing rate. Third, and perhaps most obvious, real neurons are simply not linear. This doesn't mean that the STA isn't useful, however. It is highly interpretable and can provide an unbiased estimate of cascade filters used in nonlinear models (see later).

In general, though, we'd like to be able to make an absolute measure of model performance. Two things make this difficult. First, measured responses can never be predicted perfectly, even in theory, as the measurements themselves contain inherent noise. Second, even if we discount this, a model may predict poorly because either (i) it is the wrong model, or (ii) the parameters are mis-estimated due to noise. However, there are several possible approaches at our disposal for model evaluation.

First, we can compare the mutual information between the true response and the model prediction $\mathbf{I}[\mathbf{r};\hat{\mathbf{r}}]$ to the mutual information between the true response and the stimulus $\mathbf{I}[\mathbf{r};\mathbf{s}]$, although mutual information estimators are biased. Second, we can compared $\mathbb{E}\left[(\mathbf{r}-\hat{\mathbf{r}})^2\right]$ to $\mathbb{E}\left[(\mathbf{r}-\bar{\boldsymbol{\lambda}})^2\right]$, where $\bar{\boldsymbol{\lambda}}$ is the PSTH gathered over a very large number of trials. However, it may require an impractical amount of data to accurately estimate the PSTH. Third, we can compare the *predictive power* to the *predictable power* (similar to ANOVA methods), where power is meant to mean the percent of variance that could be predicted given the perfect model.

7.2 Nonlinear Models

Despite these corrections and analyses, linear models often simply fail to predict neural responses accurately. Here, we'll discuss several nonlinear approaches to neural encoding: Volterra/Wiener expansions, linear-nonlinear (Wiener) cascades, and nonlinear (Hammerstein) cascades.

7.2.1 Volterra/Wiener Expansions

The Volterra Expansion The *Volterra* expansion is a polynomial-like expansion for functionals (or operators). Let $y = F[x(t)]$, where y is the response and x is the stimulus. Then we approximate the response

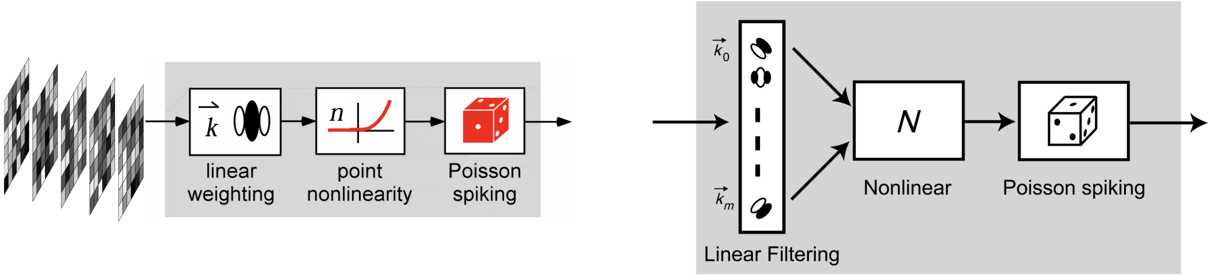


Figure 7.4: Schematics for a single-filter LNP model (left) and a multi-filter LNP model (right).

as

$$\begin{aligned} y(t) \approx k^{(0)} &+ \int k^{(1)}(\tau)x(t-\tau)\,d\tau + \int\int k^{(2)}(\tau_1,\tau_2)x(t-\tau_1)x(t-\tau_2)\,d\tau_1\,d\tau_2 \\ &+ \int\int\int k^{(3)}(\tau_1,\tau_2,\tau_3)x(t-\tau_1)x(t-\tau_2)x(t-\tau_3)\,d\tau_1\,d\tau_2\,d\tau_3 + \cdots , \end{aligned} \tag{7.19}$$

where $k^{(n)}$ are a series of kernels such that the approximation to the functional becomes exact as the number of terms goes to infinity. For finite expansions, however, the relationship of the kernels to the functional is not straightforward—the values of lower-order kernels change as the order of the expansion is increased. A polynomial kernel is a typical choice. For estimation, it's easy to see that the model is linear in the kernels themselves, so it can be estimated just like a linear (first-order) model with an expanded "input."

The Wiener Expansion The *Wiener expansion* gives functionals of different orders that are *orthogonal* for white noise input $x(t)$. We write this as a series of functionals $G_0, G_1, \ldots,$

$$\begin{aligned} G_0[x(t);h_0] &= h_0 \\ G_1[x(t);h_1] &= \int h_1(\tau)x(t-\tau)\,dx\,d\tau \\ G_2[x(t);h_2] &= \int\int h_2(\tau_1,\tau_2)x(t-\tau_1)x(t-\tau_2)\,d\tau_1\,d\tau_2 - P\int h_2(\tau_1,\tau_1)\,dx\,d\tau_1 \\ &\cdots \end{aligned} \tag{7.20}$$

such that it's easy to verify that $\langle G_i[x(t)]G_j[x(t)]\rangle = 0$ for $i \neq j$. This orthogonality allows the kernels to be estimated independently. However, they depend on the stimulus.

7.2.2 Linear-Nonlinear Cascades: STC and MID

A linear-nonlinear (LN) cascade model is a hierarchical model of encoding in which a linear weighting is applied to an input stimulus, followed by a nonlinear function (usually a non-negative function to prevent negative firing rates). The output of the nonlinearity is then interpreted as the mean rate in a Poisson process. In the simplest form of LN model, there is only a single linear filter—usually the STA (depicted schematically in Figure 7.4, left panel). However, the firing distribution changes along relevant directions in stimulus space (and, usually, along all linear combinations of relevant directions), and therefore more filters are typically needed (Figure 7.4, right panel). The mean of the distribution is captured by the STA, but this only provides one filter. Additional filters can be found via the spike-triggered covariance or the binned (or kernelized) KL divergence, as we'll see below.

Spike-Triggered Covariance The *spike-triggered covariance* (STC) is exactly what it sounds like—the covariance of the stimuli occurring in a fixed time window before each spike in the response. By finding its eigenvectors, we can identify the directions of maximum variance in stimulus space corresponding to a spike in the response. This can be done as follows. First, given the stimulus design matrix X (each row is a shifted window of the stimulus, as defined above), project out the STA:

$$\tilde{X} := X - (X\mathbf{k}_{sta})\mathbf{k}_{sta}^{\mathsf{T}}. \tag{7.21}$$

The prior and spike covariances are then

$$C_{prior} = \frac{1}{N}\tilde{X}^{\mathsf{T}}\tilde{X}; \qquad C_{spike} = \frac{1}{N_{spike}}\tilde{X}^{\mathsf{T}}\mathrm{diag}(Y)\tilde{X}. \tag{7.22}$$

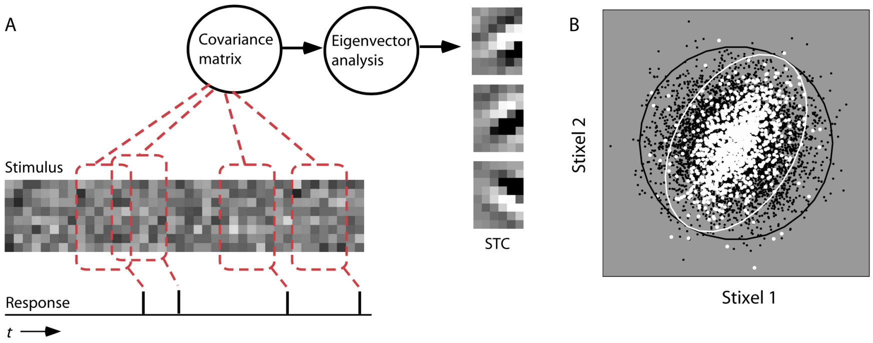


Figure 7.5: The spike-triggered covariance (STC). (A) A visualization of the flow of computation for the STC—its eigenvectors form effective filters for LN models. (B) A visualization of the principle 2D subspace in stimulus space obtained via the eigenvectors of the STC.

The STC Λ is then simply

$$\Lambda = C_{prior} - C_{spike}. \tag{7.23}$$

Then to acquire m filters, we simply find the directions (eigenvectors) with the greatest change in variance:

$$\mathbf{k}_1, \ldots, \mathbf{k}_m = m\text{-}\operatorname*{argmax}_{\|\mathbf{v}\|=1} \mathbf{v}^\mathsf{T} \Lambda \mathbf{v}. \tag{7.24}$$

In other words, we want the eigenvectors of Λ with the largest (absolute) eigenvalues. This process is visualized in Figure 7.5. To derive firing rates and reconstruct the nonlinearity, we can then repeat the same histogram-based process as depicted in Figure 7.3, except in this case, the histograms (and firing rates) are computed in the direction of each filter (eigenvector). This can be visualized in Figure 7.6.

It's important to note that that STC requires that the nonlinearity alter the variance, as otherwise $C_{prior} = C_{spike} \Rightarrow \Lambda = 0$. If this is the case, the derived subspace is unbiased provided that the stimulus distribution is (i) radially (elliptically) symmetric and (ii) independent. In other words, the STC prediction is unbiased if the stimulus distribution is Gaussian and the stimulus dimensions are independent—otherwise, the STC filters will likely fail to identify relevant dimensions. It may be possible to correct for non-Gaussian stimulus by transformation, subsampling, or weighting (with the latter two options coming at the cost of some variance).

Maximally Informative Dimensions It's also possible to consider non-parametric nonlinearities. The method of *maximally informative dimensions* (MID) extends the variance difference idea used to compute the STC to arbitrary differences between marginal and spike-conditioned stimulus distributions:

$$\mathbf{k}_{MID} = \operatorname*{argmax}_{\mathbf{k}} \mathsf{KL}[P(\mathbf{k}^\mathsf{T}\mathbf{x}) \| P(\mathbf{k}^\mathsf{T}\mathbf{x}|spike)]. \tag{7.25}$$

In other words, MID finds the dimensions of the stimulus that are most predictive (informative about) whether or not the cell will spike. In practice, measure the KL divergence requires binning or smoothing—this turns out to be equivalent to fitting a non-parametric nonlinearity (via binning or smoothing). This is difficult to use for high-dimensional LNP models, but the maximum-likelihood viewpoint suggests separable or "cylindrical" basis functions. We can also consider parametric nonlinearities, which leads us to the generalized linear model.

7.2.3 Generalized Linear Models

Generalized linear models (GLMs) are LN models with specified (parametric) nonlinearities and exponential family noise. In general, for a monotonic nonlinear function $f(\cdot)$, we visualize the spiking process as

$$y \sim P[f(\beta \mathbf{x})], \tag{7.26}$$

where β is a weight and $P(\cdot)$ is an exponential family mass function. It turns out that the continuous time point process likelihood with GLM-like dependence of λ on covariates is approximated in the limit of the bin width $\to 0$ by either a Poisson or Bernoulli GLM.

If we choose P to be a Poisson distribution, setting $f(\cdot) = \exp(\cdot)$ is called *canonical* for the distribution—in other words, the natural parameters are $\beta\mathbf{x}$. In general, choosing the canonical link functions (nonlninearities) for a given distribution gives a concave likelihood—in other words, there's a unique maximum. For the Poisson distribution, this property generalized to any f which is convex and log-concave:

$$\log P(y|x) = \log \frac{f(\beta \mathbf{x})^y}{y!} e^{-f(\beta \mathbf{x})} = y \log f(\beta \mathbf{x}) - f(\beta \mathbf{x}) - \log y!. \tag{7.27}$$

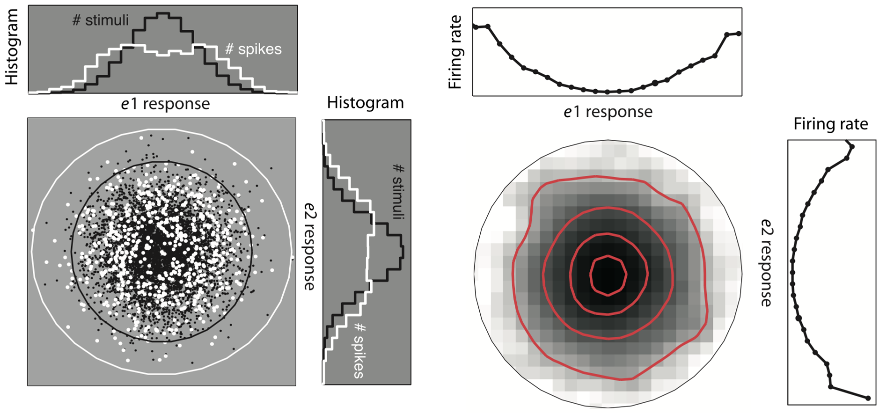


Figure 7.6: Reconstructing the nonlinearity along the directions of the eigenvectors of the STC.

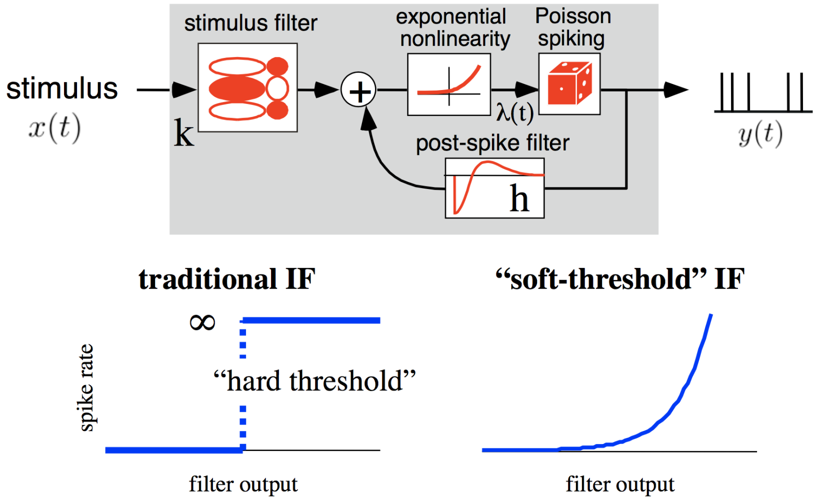


Figure 7.7: The GLM with history dependence. (Top) A model schematic. (Bottom) Spike rate as a function of filter output for a traditional integrate-and-fire (IF) model and the soft-threshold GLM.

This family of link functions includes, for example: thresholded linear functions, threshold-polynomial, and "soft-threshold": $f(z) = \alpha^{-1}\log(1 + e^{\alpha z})$ (e.g., the softplus). To find the maximum likelihood parameters (i.e., β and any parameters in the nonlinearity—which could be a neural network, for example), we can use gradient ascent on the log-likelihood or iteratively reweighted least-squared (IRLS). Regularization by L_2 or L_1 (sparse) penalties—which corresponds to MAP estimation with Gaussian/Laplacian priors—preserves concavity.

The GLM with History-Dependence We can also add recurrent computation to a GLM by including a history-dependent feedback term in the computation, such that conditional intensity/spike rate λ for an exponential link function is given by

$$\lambda(t) = f(\mathbf{k}^\mathsf{T}\mathbf{x}(t) + \mathbf{h}^\mathsf{T}\mathbf{y}(t)) = e^{\mathbf{k}^\mathsf{T}\mathbf{x}(t)} \cdot e^{\mathbf{h}^\mathsf{T}\mathbf{y}(t)}. \tag{7.28}$$

This model is visualized in the top panel of Figure 7.7. In other words, the rate is a product of stimulus- and spike-history-dependent terms, and the output is longer a Poisson process. This is also known as a soft-threshold integrate-and-fire (IF), because unlike traditional IF models in which the firing jumps from zero once the voltage (or filter output) reaches a threshold value, here the spike rate increases smoothly with the filter output (Figure 7.7, bottom). The shape of the post-spike filter (or "waveform") has a strong effect on the spiking behavior of the neuron—note that when it is zero everywhere, the model reduces to the standard GLM, and the spiking pattern is irregular, following a Poisson process. Some example waveforms and the resulting dynamic behaviors for a constant stimulus are plotted in Figure 7.8.

This framework also allows for the coupling of multiple neurons, wherein the activity of one neuron is fed as additional input to the nonlinearity of another (Figure 7.9).

Looking beyond LN models, the idea of responses depending on one or a few linear stimulus projections has been dominant, but cannot capture all nonlinearities. Some experimentally-observed phenomena hint at shortcomings of the LN framework:

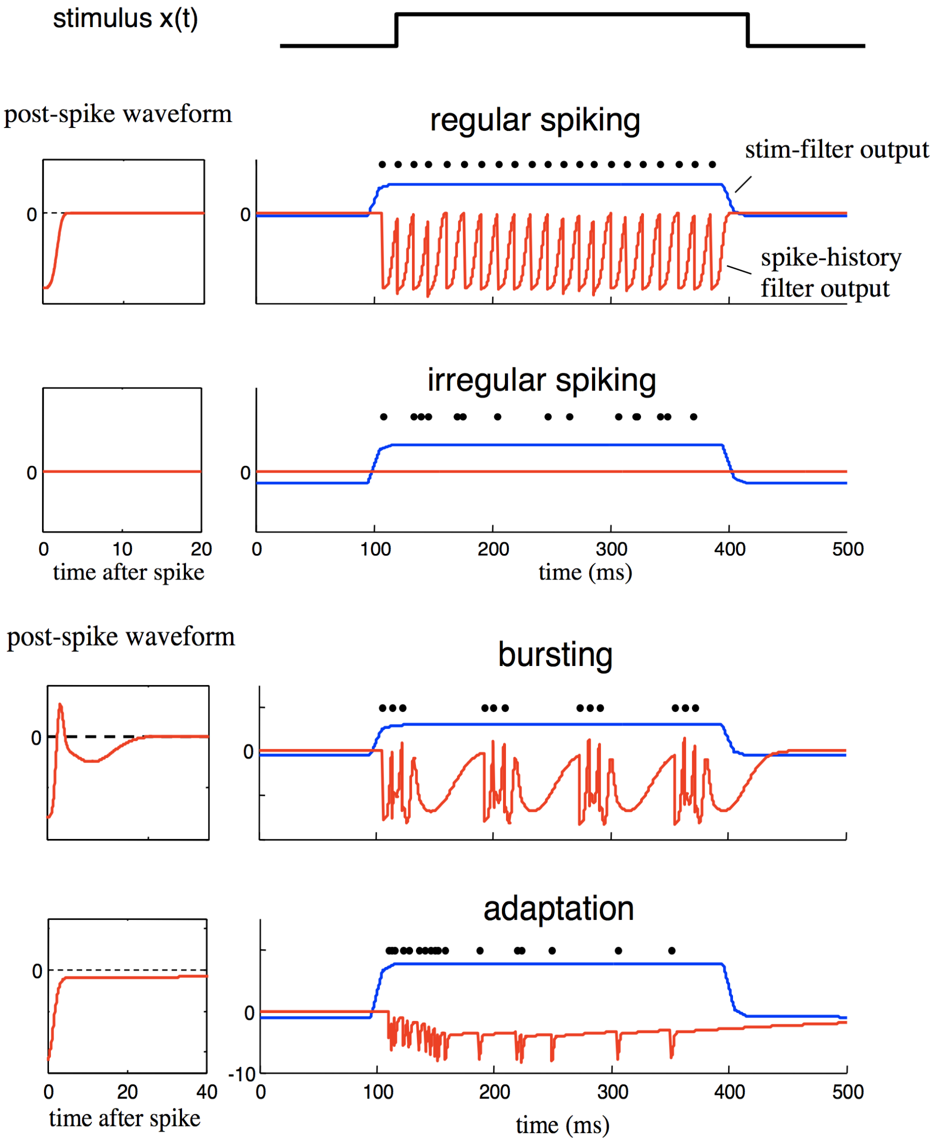


Figure 7.8: The dynamics of a GLM with history dependence in response to constant stimulus depend on the shape of the post-spike filter. Negative values in the post-spike filter inhibit firing, and positive values promote it—for example, in the “bursting" model, there is a brief period of inhibition following by positive feedback (causing the burst) and then negative feedback to stop the burst. In the regular spiking example, the inhibitory period after a spike sets the interval length by inhibiting spiking for a short time.

1. Contrast sensitivity might require normalization by $\|\mathbf{s}\|$.
2. Linear weightings may depend on *units* of stimulus measurement, such as amplitude, energy, and thresholds. This perspective is used in *Hammerstein cascade* models.
3. Neurons, particularly in the auditory system, are known to be sensitive to combinations of inputs, evident in phenomena such as forward suppression, spectral patterns, and time-frequency interactions.
4. Experiments with realistic stimuli have revealed nonlinear sensitivity to only parts of a stimulus.

Many of these questions can be addressed via multilinear (Cartesian tensor) approaches, which I don’t cover here.

7.3 Encoding Tips

- In general, compute the STA via $\mathbf{k} = \langle X^\top r \rangle$, where $\langle \cdot \rangle$ is expectation.

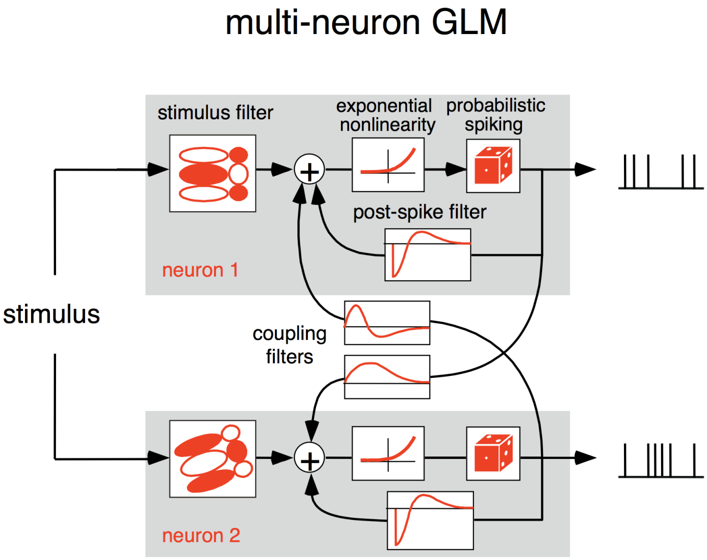


Figure 7.9: A multi-neuron GLM with history dependence.

- Don’t forget that in computing the STC, projecting out the STA in the first step reduces the number of dimensions by one, and that the resulting vector(s) are orthogonal to the STA.

8 Population Coding

Here, we focus on methods for understanding the population-level behavior of neurons, rather than seeking to understand the encoding properties of one or a few neurons at a time. Most neural codes are *distributed*, in that each neuron fires for a range of stimulus values and computations, and often, population activity must be taken together to identify the input stimulus.

There are number of factors that make this complicated, not the least of which is the simple fact that neurons are noisy: there are synaptic release failures, branch-point spike propagation failures, channel noise, and network chaos with which to contend. Neurons often display *heterogeneous dynamics*—there is no simple rule that governs a neuron's response, and the time course of its activity can vary significantly over time. They also display *mixed selectivity*, firing in response to multiple features of a given task. Overall network computation is carried in the coordinated activity of many neurons.

8.1 Optimal Encoding and Decoding

There are two common varieties of question that can be asked about population codes. First, given assumed encoding functions, how well can we (or downstream areas) decode the encoded stimulus value? Second, what encoding schemes would be optimal, in the sense of allowing decoders to estimate stimulus values as well as possible? Before considering these questions, its useful to first formulate some ideas about rate coding in the context of single cells.

8.1.1 Rate Coding and Tuning Curves

In rate coding, we imagine that the firing rate r of a cell represents a single (possibly multidimensional) stimulus value s at any one time:

$$r = f(s). \tag{8.1}$$

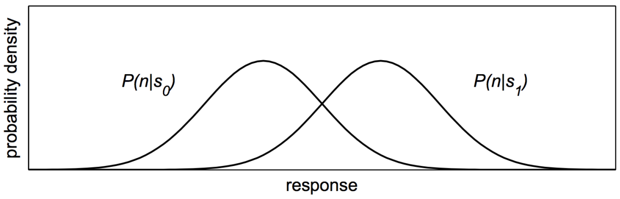
Even if s and r are embedded in time-series, we assume (i) that coding is instantaneous (with a fixed lag) and (ii) that r (and therefore s) is constant over a short time Δ. The actual number of spikes n produced in Δ is then taken to be distributed around $r\Delta$, often according to a Poisson distribution.

The function $f(s)$ is known as a *tuning curve*. Some commonly assumed forms are

- Gaussian: $f(s) = r_0 + r_{max} \exp\left[\frac{1}{2\sigma^2}(s - s_{pref})^2\right]$
- Cosine: $f(\theta) = r_0 + r_{max}\cos(\theta - \theta_{pref})$
- Wrapped Gaussian: $f(\theta) = r_0 + r_{max}\sum_n \exp\left[-\frac{1}{2\sigma^2}(\theta - \theta_{pref} - 2\pi n)^2\right]$
- von Mises ("Circular Gaussian"): $f(\theta) = r_0 + r_{max}\exp[\kappa\cos(\theta - \theta_{pref})]$

8.1.2 Discrete Choices

Suppose we'd like to make a binary choice based on firing rate, e.g., the presence/absence of a signal, up/down, horizontal/vertical, etc. If we call a stimulus corresponding to one option s_0 and the other s_1, we can expect a bimodal distribution over the number of spikes $P(n|s)$: The overlap between these distributions then corresponds to how distinguishable two stimuli are—at high overlaps, decoding the stimulus is essentially at chance, but classification is much easier for lower overlaps. This relationship can be visualized via an ROC curve (Figure 8.1). Given $n_0 \sim P(n|s_0)$ and $n_1 \sim P(n|s_1)$, the area under the ROC curve (AUC) equals $P(n_1 > n_0)$. We can more formally define the *discriminability* d' for two equal variance Gaussians as



$$d' := \frac{\mu_1 - \mu_0}{\sigma}. \tag{8.2}$$

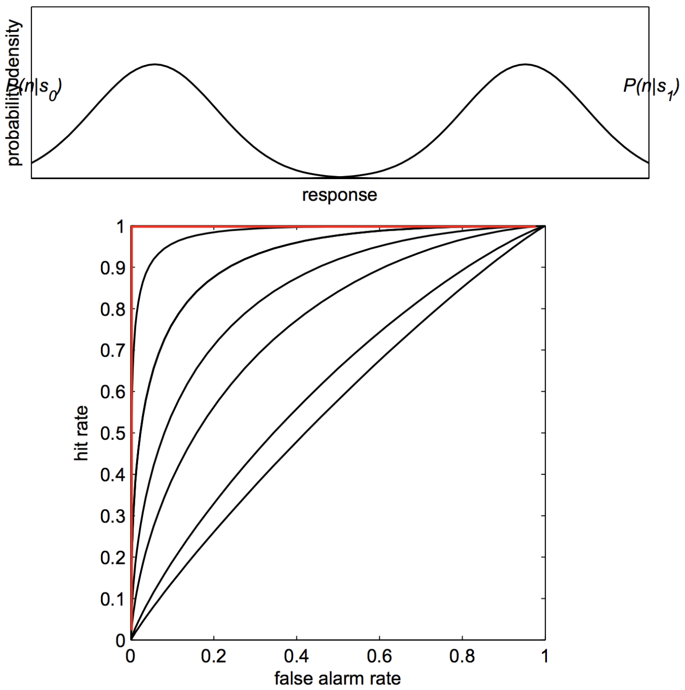


Figure 8.1: Discriminability of stimuli in a binary choice rask. As the overlap between the response distributions decreases (the distributions move further apart), the false positive rate at which the distributions can be distinguished decreases. In other words, when the distributions are completely apart, perfect classification can be achieved with zero false positives (i.e., the top left of the ROC plot), and when they complete overlap, classification happens at chance (the diagonal line).

Then for any threshold

$$d' = \Phi^{-1}(1 - FA) - \Phi^{-1}(1 - HR), \tag{8.3}$$

where Φ is the standard normal CDF, FA is the false alarm (false positive) rate, and HR is the hit (true positive) rate. In other words, the discriminability is the difference in quantile functions. However, this definition is unclear for non-Gaussian distributions.

8.1.3 Continuous Estimation and the Fisher Information

The problem of decoding such a real-valued stimulus from firing rates is called *estimation*. Consider a one-dimensional stimulus that takes on continuous values (e.g., angle, contrast, direction, speed). Suppose a neuron fires n spikes in response to simulus s according to some distribution

$$P(n|f(s)\Delta). \tag{8.4}$$

Given an observation of n, we'd like to estimate s. In order to do so, it's useful to consider the limit given $N \to \infty$ measurements n_i all generated by the same stimulus s^*. In other words, each time s^* is presented, we record the number n_i of resulting spikes. The log-posterior over the stimulus s is then given by

$$\log P(s|\{n_i\}) = \sum_i \log P(n_i|s) + \log P(s) - \log Z(\{n_i\}). \tag{8.5}$$

Taking $N \to \infty$, we have

$$\frac{1}{N} \log P(s|\{n_i\}) \to \langle \log P(n|s \rangle_{n|s^*} + 0 - \log Z(s^*), \tag{8.6}$$

and so

$$\begin{aligned} P(s|\{n_i\}) &\to \frac{1}{Z} e^{N\langle \log P(n|s)\rangle_{n|s^*}} \\ &= \frac{1}{Z'} e^{-N[\langle \log P(n|s^*)\rangle_{n|s^*} - \langle \log P(n|s)\rangle_{n|s^*}]} \\ &= \frac{1}{Z'} e^{-N\mathsf{KL}[P(n|s^*)\|P(n|s)]}. \end{aligned} \tag{8.7}$$

Note that the normalizer Z is changing from line to line, but never depends on s. We can now do a Taylor expansion around the KL divergence in s around s^*:

$$\begin{aligned}
\mathsf{KL}[P(n|s^*)\|P(n|s)] &= -\langle \log P(n|s)\rangle_{n|s^*} + \langle \log P(n|s^*)\rangle_{n|s^*} \\
&= -\langle \log P(n|s^*)\rangle_{n|s^*} - (s-s^*)\Bigg\langle \underbrace{\frac{d}{ds}\log P(n|s)\Big|_{s^*}}_{=0} \Bigg\rangle_{n|s^*} \\
&\quad - \frac{1}{2}(s-s^*)^2 \Big\langle \frac{d^2}{ds^2}\log P(n|s)\Big|_{s^*}\Big\rangle_{n|s^*} + \cdots + \langle \log P(n|s^*)\rangle_{n|s^*} \qquad (8.8)\\
&= -\frac{1}{2}(s-s^*)^2 \Big\langle \frac{d^2}{ds^2}\log P(n|s)\Big|_{s^*}\Big\rangle_{n|s^*} + \cdots \\
&= \frac{1}{2}(s-s^*)^2 J(s^*) + \cdots ,
\end{aligned}$$

where

$$J(s^*) = -\Big\langle \frac{d^2}{ds^2}\log P(n|s)\Big|_{s^*}\Big\rangle_{n|s^*} \tag{8.9}$$

is the *Fisher information* (matrix). Therefore, in aymptopia, the posterior is

$$\begin{aligned}
P(s|\{n_i\}) &\to \frac{1}{Z}e^{-N\mathsf{KL}[P(n|s^*)\|P(n|s)]} && (8.10)\\
&= \frac{1}{Z}e^{-N\frac{1}{2}(s-s^*)^2 J(s^*)} = \frac{1}{Z'}e^{-\frac{1}{2}(s-s^*)^2 J(s^*)} && (8.11)\\
&= \mathcal{N}(s^*, J(s^*)^{-1}). && (8.12)
\end{aligned}$$

Thus, the Fisher information can be seen as measuring the sensitivity of the neural response to changes in the stimulus. Note that this is only a *local* measure of information, as it's contingent on the Taylor expansion around the true stimulus value s^*. Further intuition can be gained by deriving an alternate form of the Fisher information (here extended to a multidimensional stimulus) using the score trick ($\nabla_s \log P = \frac{1}{P}\nabla_s P$):

$$\begin{aligned}
J(s^*) &= -\Big\langle \nabla_s^2 \log P(n|s)\Big|_{s^*}\Big\rangle_{n|s^*} && (8.13)\\
&= -\Big\langle \nabla_s \Big[\frac{1}{P(n|s)}\nabla_s P(n|s)^{\mathsf{T}}\Big]\Big\rangle = -\Big\langle \Big[\nabla_s \frac{1}{P(n|s)}\Big]\nabla_s P(n|s)^{\mathsf{T}}\Big\rangle && (8.14)\\
&= \Big\langle \frac{1}{P(n|s)^2}\left[\nabla P(n|s)\right]\left[\nabla P(n|s)\right]\Big\rangle && (8.15)\\
&= \Big\langle \Big[\frac{1}{P(n|s)}\nabla P(n|s)\Big]\Big[\frac{1}{P(n|s)}\nabla P(n|s)\Big]^{\mathsf{T}}\Big\rangle && (8.16)\\
&= \Big\langle \left(\nabla \log P(n|s)\right)\left(\nabla \log P(n|s)\right)^{\mathsf{T}}\Big\rangle. && (8.17)
\end{aligned}$$

Thus, the Fisher information is the variance of the score function.

The Fisher information is important even outside the large data limit due to the deeper result that is due to Cramér and Rao, which states that for any N, any *unbiased* estimator $\hat{s}(\{n_i\})$ of s will have the property that

$$\mathbb{V}[\hat{s}] = \big\langle (\hat{s}(\{n_i\}) - s^*)^2\big\rangle_{n_i|s^*} \geq \frac{1}{J(s^*)}. \tag{8.18}$$

Thus, the Fisher information gives a lower bound on the variance of any unbiased estimator. This is called the *Cramér-Rao bound.* For estimators with bias $b(s^*) = \langle \hat{s}(\{n_i\}) - s^*\rangle$ the bound is

$$\big\langle (\hat{s}(\{n_i\}) - s^*)^2\big\rangle_{n_i|s^*} \geq \frac{(1+b'(s^*))^2}{J(s^*)} + b^2(s^*). \tag{8.19}$$

The Fisher information is a key tool in the analysis of neural population codes.

The Fisher Information and Tuning Curves We can model each observed spike count n as $n = r\Delta +$ noise, where $r = f(s)$ (in other words, r is the firing rate given by the tuning curve). The Fisher information

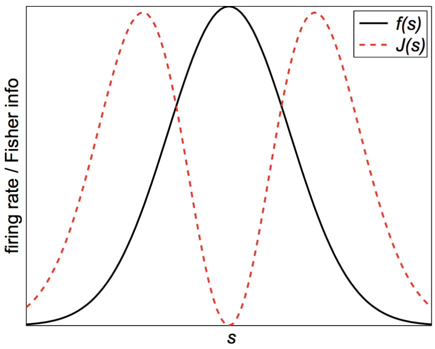


Figure 8.2: Relationship between the Fisher information of a stimulus $J(s)$ and the associated tuning curve $f(s)$.

is then

$$\begin{aligned} J(s^*) &= \left\langle \left(\frac{d}{ds} \log P(n|s) \Big|_{s^*} \right)^2 \right\rangle_{n|s^*} \\ &= \left\langle \left(\frac{d}{dr\Delta} \log P(n|r\Delta) \Delta f'(s^*) \Big|_{f(s^*)} \right)^2 \right\rangle_{n|s^*} \\ &= J_{\text{noise}}(r\Delta)\Delta^2 f'(s^*)^2. \end{aligned} \tag{8.20}$$

In this case, we can see that the Fisher information will be non-negative, it's value changing according to the squared derivative of the tuning curve $f(s)$. The Fisher information then goes to zero wherever the firing rate hits a local maximum or minimum. This relationship is plotted in Figure 8.2.

Poisson Neurons For Poisson neurons, the conditional spike distribution is given by

$$P(n|r\Delta) = \frac{e^{-r\Delta}}{n!}(r\Delta)^n, \tag{8.21}$$

so

$$\begin{aligned} J_{\text{noise}}(r\Delta) &= \left\langle \left(\frac{d}{dr\Delta} \log P(n|r\Delta) \Big|_{r^*\Delta} \right)^2 \right\rangle_{n|s^*} \\ &= \left\langle \left(\frac{d}{dr\Delta} - r\Delta + n\log(r\Delta) - \log n! \Big|_{r^*\Delta} \right)^2 \right\rangle_{n|s^*} \\ &= \left\langle \left(-1 + \frac{n}{r^*\Delta} \right)^2 \right\rangle_{n|s^*} \\ &= \left\langle \frac{\overbrace{(n - r^*\Delta)^2}^{\langle (n-r^*\Delta)^2 \rangle := \mathbb{V}[n|r\Delta] = r^*\Delta}}{(r^*\Delta)^2} \right\rangle_{n|s^*} \\ &= \frac{r^*\Delta}{(r^*\Delta)^2} \\ &= \frac{1}{r^*\Delta}. \end{aligned} \tag{8.22}$$

Note that this isn't too surprising, as the optimal estimator for the mean $r^*\Delta$ is $\widehat{r^*\Delta} = n$ and $\mathbb{V}[n] = r^*\Delta$. The complete stimulus Fisher information, using eq. 8.20 and $r^* := f(s^*)$, is then

$$\begin{aligned} J(s^*) &= J_{\text{noise}}(r\Delta)\Delta^2 f'(s^*)^2 = \frac{1}{r^*\Delta}\Delta^2 f'(s^*)^2 \\ &= \frac{f'(s^*)^2}{f(s^*)}\Delta. \end{aligned} \tag{8.23}$$

8.1.4 Optimal Tuning Curve Widths

Consider a population of neurons $a = 1, \dots, N$ coding for a multidimensional stimulus $s \in \mathbb{R}^D$, with homogeneous tuning curves given by

$$f_a(s) = r_{max}\phi\left(\frac{\sum_{d=1}^{D}(s_d - c_d^a)^2}{\sigma^2}\right) = r_{max}\phi\left(\frac{\xi^a}{\sigma^2}\right), \tag{8.24}$$

where c_d^a are the uniformly distributed tuning curve centers and $\phi(\cdot)$ is a monotonically decreasing function (e.g., $\phi(x) = e^{-x}$ for a Gaussian tuning curve). Note that because the tuning curve width is constant in each dimension, the tuning curves are circularly symmetric. We also assume that the tuning curves are independent, such that $P(r|s) = \prod_a P(r_a|f_a(s))$. Then we can see quickly that the Fisher information for the population is given by the sum of the Fisher information for each neuron:

$$\begin{aligned} J(s) &= \left\langle -\frac{\partial^2}{\partial s^2}\log P(r|s)\right\rangle && (8.25)\\ &= \left\langle -\frac{\partial^2}{\partial s^2}\sum_a \log P(r_a|f_a(s))\right\rangle && (8.26)\\ &= \sum_a\left\langle -\frac{\partial^2}{\partial s^2}\log P(r_a|f_a(s))\right\rangle && (8.27)\\ &= \sum_a J_a(s). && (8.28)\end{aligned}$$

The question we'd like to ask, then, is what tuning curve width σ maximizes the population Fisher information?

To figure this out, we first find the Fisher information for a single neuron a, this time using the covariance form (eq. 8.17):

$$J_{ij}^a(s) = \left\langle\left(\frac{\partial}{\partial s_i}\log P(r_a|f_a(s))\right)\left(\frac{\partial}{\partial s_i}\log P(r_a|f_a(s))\right)^{\mathsf{T}}\right\rangle. \tag{8.29}$$

The derivative is

$$\begin{aligned}\frac{\partial}{\partial s_i}\log P(r_a|f_a(s)) &= \frac{1}{P(r_a|s)}\frac{\partial P(r_a|f_a(s))}{\partial s_i} && (8.30)\\ &= \frac{1}{P(r_a|s)}\frac{\partial P(r_a|f_a(s))}{\partial f_a(s)}\frac{\partial f_a(s)}{\partial\phi(\xi^a/\sigma^2)}\frac{\partial\phi(\xi^a/\sigma^2)}{\partial\xi^a}\frac{\partial\xi^a}{\partial s_i} && (8.31)\\ &= \frac{1}{P(r_a|s)}\frac{\partial P(r_a|f_a(s))}{\partial f_a(s)}r_{max}\frac{1}{\sigma^2}\phi'(\xi^a/\sigma^2)2(s_i - c_i^a), && (8.32)\end{aligned}$$

and so we can write

$$J_{ij}^a(s) = K_a(\xi_a)\frac{(s_i - c_i^a)(s_j - c_j^a)}{\sigma^4}, \tag{8.33}$$

where

$$K_a(\xi_a) = \left\langle 4\left(\frac{1}{P(r_a|s)}\frac{\partial P(r_a|f_a(s))}{\partial f_a(s)}r_{max}\phi'(\xi^a/\sigma^2)\right)^2\right\rangle \tag{8.34}$$

is the only term where the expectation appears since it is the only term dependent on the activity r_a over which the expectation is defined. If we define

$$\xi_i^a := \frac{s_i - c_i}{\sigma}, \tag{8.35}$$

such that $\xi^a = \sigma^2\sum_i(\xi_i^a)^2$, we can see that since $\phi(\xi^a/\sigma^2)$ is a monotonically decreasing function of $(\xi_i^a)^2$, it is symmetric around $\xi_i^a = 0$. Thus, $\phi'(\xi^a/\sigma^2)^2$ and $f_a(s)$ are as well, so $K_a(\xi_a)$ is also symmetric around $\xi_i^a = 0$.

Because we assumed that the tuning curve centers are uniformly distributed about the stimulus space, we can say $P(c_i^a) = p_c$. Taking the limit $N \to \infty$ to approximate sums with integrals, we can then write

$$\begin{aligned} J_{ij}(s) &= \sum_a J_{ij}^a(s) && (8.36)\\ &= \int\cdots\int p_c J_{ij}^a\, dc_1^a\, dc_2^a \dots dc_D^a && (8.37)\\ &= \int\cdots\int p_c K_a(\xi_a)\frac{(s_i - c_i^a)(s_j - c_j^a)}{\sigma^4}\, dc_1^a\, dc_2^a \dots dc_D^a. && (8.38)\end{aligned}$$

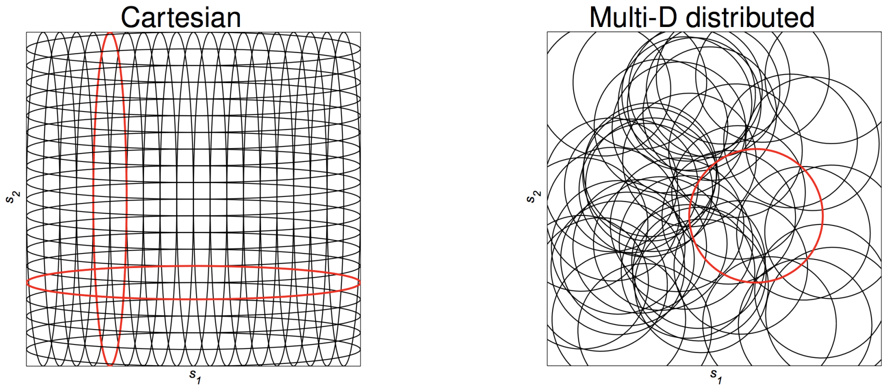


Figure 8.3: Examples of coding frameworks. (Left) A Cartesian code. This is efficient in the number of neurons required, but has difficulty encoding multiple values. (Left) A distributed code. This style requires a high number of neurons, but is more effective at representing multiple stimulus values.

To evaluate this integral, we can exploit the fact that $K_a(\xi_a)$ is symmetric around $\xi_i^a = 0$ by performing a change of variables $c_i^a \to \xi_i^a$, such that $dc_i^a = -\sigma\, d\xi_i^a$:

$$J_{ij}(s) \approx \frac{1}{\sigma^2} \int \cdots \int p_c K_a(\xi_a) \xi_i^a \xi_j^a \, \sigma d\xi_1^a \, \sigma d\xi_2^a \ldots \sigma d\xi_D^a \tag{8.39}$$

$$= \frac{\sigma^D}{\sigma^2} \int \cdots \int p_c K_a(\xi_a) \xi_i^a \xi_j^a \, d\xi_1^a \, d\xi_2^a \ldots d\xi_D^a. \tag{8.40}$$

Because $K_a(\xi^a)$ is symmetric around $\xi_i^a = 0$, we have $\int_{-\infty}^{\infty} p_c K_a(\xi^a) \xi_i^a = 0$, so when ß $\neq j$,

$$J_{ij}(s) \approx \frac{\sigma^D}{\sigma^2} \int \cdots \xi_j^a \int p_c K_a(\xi_a) \xi_i^a \, d\xi_i^a \, d\xi_{i+1}^a \ldots d\xi_D^a = 0. \tag{8.41}$$

However, when $i = j$, we get

$$J_{ii}(s) \approx \frac{\sigma^D}{\sigma^2} \int \cdots \int p_c K_a(\xi_a) (\xi_i^a)^2 \, d\xi_1^a \, d\xi_2^a \ldots d\xi_D^a = \sigma^{D-2} A, \tag{8.42}$$

where A is independent of σ.

We can therefore see that the total Fisher information in the population is proportional to σ^{D-2}, where D is the dimensionality of the stimulus. This yields some interesting insights about the optimal tuning curve width σ:

- If $D = 1$, the tuning curves should want to be as narrow as possible, down to the smallest resolution between neighboring c_i^a.
- If $D = 2$, the Fisher information is independent of tuning curve widths.
- If $D > 2$, the wider the tuning curve the better—optimality is achieved when the tuning curve spans the full stimulus space.

This analysis was derived by [6], but the write-up is pretty much verbatim from [3]. It also turns out that if the tuning curve widths are allowed to vary between stimulus dimensions (i.e., they are no longer circularly symmetric), then maximizing the Fisher information gives a Cartesian code in which the optimal tuning curve width is narrow in some dimensions and wide in others (Figure 8.3, left).

8.2 Doubly Distributional Population Codes (Dayan & Sahani, 2003)

The authors present a new model for population coding that, unlike previous approaches, is able to account for the presence of multiple stimuli and representational uncertainty [7]. The authors define the standard model of population coding as follows: the firing rate of the ith neuron r_i is distributed around the mean rate $f_i(s)$ for a given value of a stimulus s, where the plot of f_i for different values of s defines the neuron's tuning curve. This is written as

$$r_i \sim f_i(s). \tag{8.43}$$

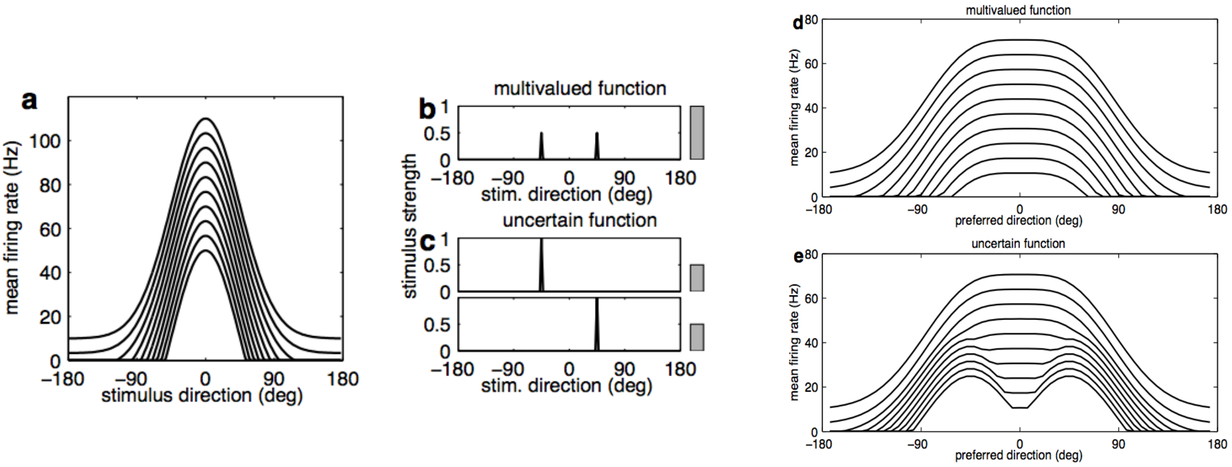


Figure 8.4: Encoding multiplicity and uncertainty with a DDPC. Panel (a) shows the tuning curves for 10 neurons with a preferred stimulus orientation of 0°. Panels (b) and (c) show two types of input stimuli: (b) is a multi-valued function, and (c) contains a function whose value is uncertain. However, it is important to note that both functions have an expected value of 0°. Panels (d) and (e) show the resulting DDPC representations of the multi-valued and uncertain functions, respectively. We can see that despite having identical expectations, the DDPC is able to differentiate between the two. Figure from [7].

The population firing rate $\mathbf{r} = \{r_i\}$ defines the representation of the stimulus, and the populations that respond to different types of stimuli can be broadly overlapping, thus adding robustness to noise and cell death. They show that standard population coding models of neural activity fail to account for two common scenarios that occur in stimulus presentation: *multiplicity* and *uncertainty.* Multiplicity is when multiple values of s need to be represented simultaneously (e.g., different sounds are produced by spatially separate sources). Uncertainty arises through two possible sources: noise and ill-posedness in perceptual inference, an issue arising from the fact that cortical representations must be themselves be computed from the uncertain outputs of sensory neurons, not directly from the value of s (which is essentially a latent variable). This results in the possibility that two different values of s could produce the same intermediate representation via sensory neurons–in other words, two different stimuli could result in the same input to cortical neurons.

A generalization of the standard model is the *distributional population code* (DPC), which views the population activity as being a *function* $m(s)$ over the stimulus, rather than encoding only a single value. This is an exact generalization because if $m(s)$ is a delta function on s, then it recovers the old model. Using the same notation as above, the ith firing rate is given by

$$r_i \sim \sigma_i \left(\int f_i(s) m(s)\, ds \right), \tag{8.44}$$

where $\sigma_i(\cdot)$ is a fixed nonlinearity. When $m(s)$ is a delta function, $r_i \sim \sigma_i(f_i(s))$, equivalent to the original model. When that's not the case, the population of firing rates $\mathbf{r}$ is sufficient to decode an approximation to $m(\cdot)$. However, a crucial shortcoming is that DPCs can encode either multiplicity *or* uncertainty, but it cannot distinguish between the two, as they are both represented the same way, nor can it represent both at the same time.

To meet this need, the authors introduce the *doubly distributional population code* (DDPC), which encodes uncertainty about the functions $m(s)$. In a DDPC, the population activity encodes a probability distribution $p[m]$ over functions $m(s)$, with $m(s)$ capturing potential multiplicity and $p[m]$ capturing uncertainty. As an example, if multiple moving stimuli are presented, $m(s)$ would represent the activation strengths of different sensory receptors. Then a perfectly certain $m(\cdot)$ would be encoded by neuron i with linear filter function $f_i(s)$ and nonlinearity $\sigma_i(\cdot)$, producing mean

$$\phi_i[m] = \sigma_i \left(\int f_i(s) m(s)\, ds \right). \tag{8.45}$$

However, issues such as ill-posedness also require that uncertainty about $m(s)$ be represented via a distribution $p[m]$, which can be learned through experience. The mean activity of the ith neuron is then the average over $p[m]$ of its DPC activity:

$$r_i(p[m]) = \langle \phi_i[m] \rangle_{p[m]} = \left\langle \sigma_i \left(\int f_i(s) m(s)\, ds \right) \right\rangle_{p[m]}. \tag{8.46}$$

This representation of $p[m]$ in firing rates is the *doubly distributional population code* (DDPC). The authors then demonstrate empirically that the DDPC is indeed able to differentiate between cases of multiplicity and uncertainty (Figure 8.4), and then through a MAP framework, that this difference is sufficient to *decode* the activity to an accurate representation of the original stimulus.

8.3 Overview of Latent Variable Approaches

To make sense of population dynamics, we can also turn to latent variable methods, assuming that while the activity space is high-dimensional, only a handful of underlying states are truly meaningful. In this framework, coordinates on the latent manifold of possible activity combinations effectively become latent variables—if the meaningful content is truly low-dimensional, so are the dynamics (e.g., they exist as a path on the manifold; Figure 8.5).

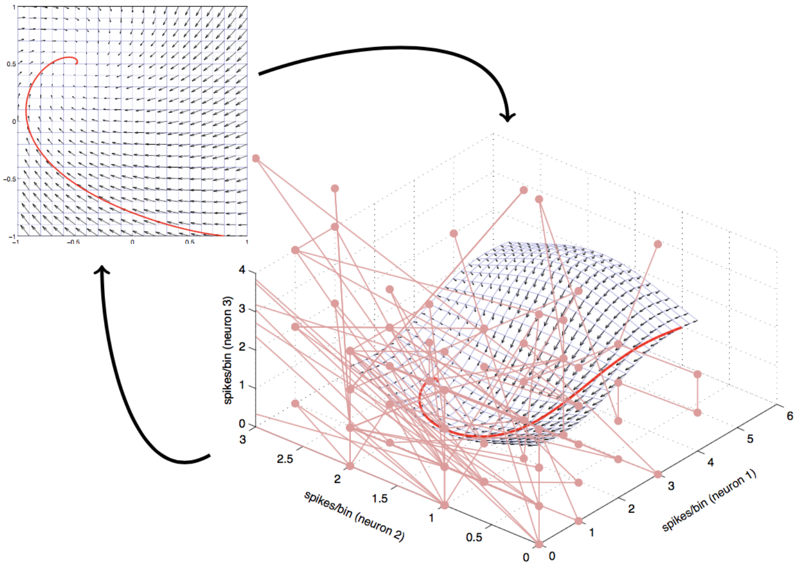


Figure 8.5: Example of a 3D (aka, three neuron) neural activity space with low(er) dynamics.

Under this assumption, there are three families of approaches we can consider: (i) static dimensionality reduction, which requires that the dominant force behind neural variability is not noise but rather computational variability within the manifold, (ii) low-dimensional latent dynamics, which assumes that noise may lift data off the manifold, but only its projection onto the manifold can influence the future evolution of the system, and (iii) supervised modeling.

8.3.1 Static Dimensionality Reduction

Under this framework, there are a number of methods we can use. These include linear Gaussian methods, such as (P)PCA and factor analysis (FA). It's important to remember that PPCA and PCA are invariant to rotations of the input, while FA is not, and FA is invariant to measurement scale, while PCA and PPCA are not. Therefore, (P)PCA are generally preferred when we expected rotational symmetry, i.e., when there's nothing special about one axis of the input space compared to another. However, this doesn't turn out to be the case in neural population analysis—rather, invariance to scale results in a more accurate model, and FA is generally preferred (Figure 8.6).

The assumptions of Gaussian noise and mean-independent, stationary variance are unrealistic for real spike counts, particularly in small bins. Square-rooting improves things, but is inaccurate for small counts and transforms the shape of the manifold. Instead, one can use a conditionally Poisson count distribution via (i) Poisson factor analysis (PFA), (ii) exponential family PCA, or (iii) a covariance transformation. Options (i) and (ii) follow the frameworks of FA and PCA, respectively, but with non-Gaussian noise assumptions—typically Poisson. Exponential family PCA also typically includes a regularization factor to limit the rank of solutions. We can also use methods like canonical correlations analysis (CCA).

Dynamics There are a number of dynamics-based approaches to modeling population coding. Examples include slow feature analysis (SFA), Guassian processes (GPs) and Gaussian process factor analysis (GPFA), linear-Gaussian state-space models (LGSSMs, such as Kalman filters), linear dynamical systems (LDS; related

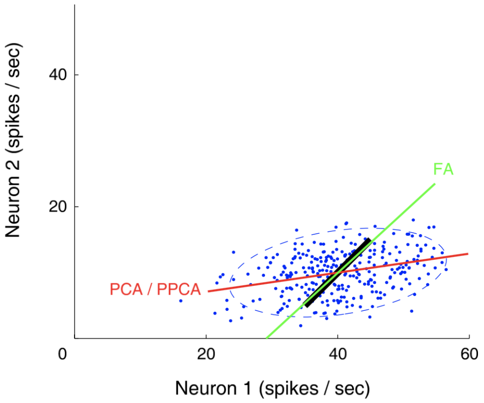


Figure 8.6: The scale-invariance of FA typically better captures the latent variables in neural data compared to PCA. Here, the black line represents the underlying mean. Note that the PCA estimate is pulled along the axis of greatest variation—it is not scale-invariant.

to the Kalman filter), Poisson noise LDS (here, EM is intractable—requires approximation), and recurrent linear models.

Supervised Approaches In this family of approaches, models try to predict known experimental factors or covariates (e.g., movement speed or stimulus identity) from multivariate data. The methods considered are also used to study structure in in the condition averages—this is equivalent to having one trial per condition. Averaging might make the noise more Gaussian, but still without equal variance.

To predict categorical factors, such as stimulus identities or behavioral instructions, methods like multifactor decomposition of variance are essential. More generally, we can also use, for instance, linear discriminant analysis (LDA), demixed PCA (DPCA), or Wiener filtering.

References

[1] Maneesh Sahani and Peter Latham. URL http://www.gatsby.ucl.ac.uk/teaching/courses/sntn/sntn-2018/lectures.html.

[2] P. E. Latham, B. J. Richmond, P. G. Nelson, and S. Nirenberg. Intrinsic dynamics in neuronal networks. i. theory. *Journal of Neurophysiology*, 83(2):808–827, Jan 2000. doi: 10.1152/jn.2000.83.2.808.

[3] Jorge A Menendez. Gatsby theoretical neuroscience notes. 2018. URL http://www.gatsby.ucl.ac.uk/teaching/courses/sntn/sntn-2018/resources/jorge/JorgeTNnotes.pdf.

[4] Peter Dayan and L. F. Abbott. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. The MIT Press, 2005. ISBN 0262541858.

[5] Wulfram Gerstner, Werner M. Kistler, Richard Naud, and Liam Paninski. *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*. Cambridge University Press, USA, 2014. ISBN 1107635195.

[6] Kechen Zhang and Terrence J. Sejnowski. Neuronal tuning: To sharpen or broaden? *Neural Computation*, 11(1):75–84, 1999. doi: 10.1162/089976699300016809.

[7] Maneesh Sahani and Peter Dayan. Doubly distributional population codes: Simultaneous representation of uncertainty and multiplicity. *Neural Computation*, 15(10):2255–2279, 2003.

NOTE: a lot of the math is taken from Jorge's TN notes. (Thank you Jorge!)

A Differential Equations

A.1 First-Order Linear ODEs: Integrating Factors

Say we have an ODE of the form

$$\frac{dy}{dx} + p(x)y(x) = g(x). \tag{A.1}$$

This isn't separable, so we can't simply integrate. Instead, we can take advantage of the product rule and define the *integrating factor* $v(x)$:

$$v(x) := \int p(x)\, dx \Rightarrow \frac{dv}{dx} = p(x). \tag{A.2}$$

Then

$$\begin{aligned}
\frac{d}{dx} y(x)e^{v(x)} &= \frac{dy}{dx} e^{v(x)} + y(x)\frac{dv}{dx} e^{v(x)} && \text{(A.3)}\\
&= \frac{dy}{dx} e^{v(x)} + y(x)p(x)e^{v(x)} && \text{(A.4)}\\
&= \left(\frac{dy}{dx} + y(x)p(x)\right) e^{v(x)} && \text{(A.5)}\\
&= g(x)e^{v(x)}. && \text{(A.6)}
\end{aligned}$$

Then we can simply obtain the solution via integration:

$$\begin{aligned}
\int \frac{d}{dx} y(x)e^{v(x)}\, dx &= \int g(x)e^{v(x)}\, dx && \text{(A.7)}\\
\Rightarrow y(x)e^{v(x)} &= \int g(x)e^{v(x)}\, dx && \text{(A.8)}\\
\Rightarrow y(x) &= e^{-v(x)} \int g(x)e^{v(x)}\, dx. && \text{(A.9)}
\end{aligned}$$

A.2 Homogeneous Second-Order ODEs

Consider a differential equation of the form

$$\frac{d^2y}{dx^2} + p\frac{dy}{dx} + qy(x) = 0, \tag{A.10}$$

with constant coefficients $p, q \in \mathbb{R}$. If we can find a pair $a, b \in \mathbb{R}$ such that $p = -(a + b)$ and $q = ab$, we can convert this second-order ODE into a first-order ODE:

$$\begin{aligned}
y'' + py' + qy &= y'' - (a + b)y' + aby && \text{(A.11)}\\
&= (y' - ay)' + b(ay - y') && \text{(A.12)}\\
&= 0 && \text{(A.13)}\\
\Rightarrow (y' - ay)' &= b(y' - ay). && \text{(A.14)}
\end{aligned}$$

Letting $u \coloneqq y' - ay$, we can then solve the resulting first-order ODE:

$$\begin{aligned} u' &= bu \\ \Rightarrow u(x) &= Ce^{bx} \\ \Rightarrow y' - ay &= Ce^{bx} \\ \Rightarrow y'e^{-ax} - aye^{-ax} &= Ce^{(b-a)x} \\ \Rightarrow \frac{d}{dx}(y(x)e^{-ax}) &= Ce^{(b-a)x} \\ \Rightarrow \int \frac{d}{dx}(y(x)e^{-ax})\,dx &= \int Ce^{(b-a)x}\,dx \\ \Rightarrow y(x)e^{-ax} &= \underbrace{\frac{C}{b-a}}_{c_1} e^{(b-a)x} + c_2 \\ \Rightarrow y(x) &= c_1e^{bx} + c_2e^{ax}. \end{aligned} \tag{A.15}$$

If $a = b$, we get $\frac{d}{dx}ye^{-ax} = C \Rightarrow y(x) = e^{ax}(c_1x + c_2)$. Therefore, all that's needed to solve such an ODE is to find a, b such that $p = -(a + b)$ and $q = ab$. This is done easily by noting that a, b are the solutions to the quadratic equation

$$r^2 + pr + q = r^2 - (a + b)r + ab = (r - a)(r - b) = 0. \tag{A.16}$$

This equation is called the *characteristic equation* of the ODE. The quadratic formula gives the solution:

$$a, b = \frac{-p \pm \sqrt{p^2 - 4q}}{2}. \tag{A.17}$$

A.3 Ricatti Equations

NOTE: Thank you Wikipedia.

A *Ricatti equation* is any first-order ODE that is quadratic in the unknown function. In other words, it is of the form

$$\ddot{y}(t) = q_0(t) + q_1(t)y(t) + q_2(t)y^2(t), \tag{A.18}$$

where $q_0, q_2 \neq 0$. To solve such an equation, we can reduce it to a linear second-order ODE, solvable using the method described above.

Given eq. A.18, we can say that wherever q_2 is non-zero and differentiable, $v :== yq_2$ satisfies a Ricatti equation of the form

$$\dot{v} = v^2 + R(t)v + S(t), \tag{A.19}$$

where $S = q_2q_0$ and $R = q_1 + \frac{\dot{q}_2}{q_2}$, because

$$\begin{aligned} \dot{v} &= \frac{d}{dt}(yq_2) = \dot{y}q_2 + y\dot{q}_2 \\ &= (q_0 + q_1y + q_2y^2)q_2 + v\frac{\dot{q}_2}{q_2} = q_0q_2 + \left(q_1 + \frac{\dot{q}_2}{q_2}\right)v + v^2. \end{aligned} \tag{A.20}$$

Substituting $v = -u'/u$, it follows that u satisfies the linear second order ODE given by

$$\ddot{u} - R(t)\dot{u} + S(t)u = 0, \tag{A.21}$$

since

$$\dot{v} = -\frac{d}{dt}(\dot{u}/u) = -\ddot{u}/u + (\dot{u}/u)^2 = -\frac{\ddot{u}}{u} + v^2, \tag{A.22}$$

so that

$$\frac{\ddot{u}}{u} = v^2 - \dot{v} = -S + R\frac{\dot{u}}{u} \tag{A.23}$$

and hence

$$\ddot{u} - R(t)\dot{u} + S(t)u = 0. \tag{A.24}$$

Therefore, to solve an equation of this form, all we need to do is find S and R using the expressions above, then solve eq. A.21 for u. The final result is given by

$$y = -\frac{\dot{u}}{q_2u}. \tag{A.25}$$

B Dynamical Systems Analysis

B.1 1D Systems

For 1D systems, we can simply plot the derivative on the phase plane with $\dot{x} = f(x)$ on the y-axis and x on the x-axis. Fixed points then occur wherever $f(x) = 0$. Given a fixed point x^*, we can say that it's stable if $f(x^* + \delta x) < 0$ and unstable if $f(x^* + \delta x) > 0$, for some zero-centered small Gaussian perturbation δx. This is intuitive, as a negative slope at the fixed point implies that the derivative is positive to the left of the point (x is increasing) and it is negative to the right of the point (x is decreasing)—thus the dynamics are converging to the fixed point. The converse is clearly true for a positive slope.

The sign of the slope can be checked by linearizing the derivative around the fixed point via a first order Taylor expansion:

$$f(x^* + \delta x) \approx \underbrace{f(x^*)}_{0} + \delta x \frac{df}{dx}\Big|_{x=x^*}. \tag{B.1}$$

(More details on linearization below.) Another useful fact to keep in mind is that 1D ODEs cannot express periodic behavior. You can't draw a phase portrait on the real line that oscillates—it will always either tend to $\pm\infty$ or a fixed point.

B.2 2D Systems

Consider a dynamical system of the form:

$$\begin{aligned} \frac{dx}{dt} &= f(x, y) \\ \frac{dy}{dt} &= g(x, y). \end{aligned} \tag{B.2}$$

Such a system is called *autonomous* because the evolution of the variables only depends on x, y and nothing else. To understand such a system, we'd like to know the behavior of $x(t)$ and $y(t)$ over time. For very simple systems, we can compute trajectories directly by picking an initial condition and solving

$$\frac{dy}{dx} = \frac{g(x, y)}{f(x, y)}, \tag{B.3}$$

but this is usually not feasible. Instead, a qualitative understanding can be obtained by studying the *nullclines* of the system in the x-y plane:

$$f(x, y) = 0 \quad \text{(x-nullcline)} \tag{B.4}$$

$$g(x, y) = 0 \quad \text{(y-nullcline).} \tag{B.5}$$

Nullclines can guide understanding via the following rules

1. trajectories can only cross the x-nullcline vertically (as $dx/dt = 0$)
2. trajectories can only cross the y-nullcline horizontally (as $dy/dt = 0$)
3. the nullclines partition the phase plane such that dy/dx only changes sign when it crosses a nullcline
4. fixed points in the dynamics occur where nullclines intersect

The last point is of central interest, as what we'd really like to know is the long-term behavior of the system, especially around the fixed points. We can do so via *linear stability analysis.* Consider a fixed point (x^*, y^*) of the above system, found via

$$f(x^*, y^*) = g(x^*, y^*) = 0. \tag{B.6}$$

To analyze the behavior of the system near this point, we introduce a small perturbation

$$(\tilde{x}(t), \tilde{y}(t)) := (x^* + \delta x(t), y^* + \delta y(t)) \tag{B.7}$$

and study the dynamics. If, as $t \to \infty$, $(\tilde{x}(t), \tilde{y}(t)) \to (x^*, y^*)$, or, equivalently, $(\delta x(t), \delta y(t)) \to (0, 0)$, then the fixed point is said to be *stable*. We have

$$\frac{d\delta x}{dt} = \frac{d\tilde{x}}{dt} = f(x^* + \delta x, y^* + \delta y) \approx f(x^*, y^*) + f_x(x^*, y^*)\delta x + f_y(x^*, y^*)\delta y \tag{B.8}$$

$$= f_x(x^*, y^*)\delta x + f_y(x^*, y^*)\delta y \tag{B.9}$$

$$\frac{d\delta y}{dt} = \frac{d\tilde{y}}{dt} = g(x^* + \delta x, y^* + \delta y) \approx g(x^*, y^*) + g_x(x^*, y^*)\delta x + g_y(x^*, y^*)\delta y \tag{B.10}$$

$$= g_x(x^*, y^*)\delta x + g_y(x^*, y^*)\delta y \tag{B.11}$$

where $f_x(a, b) := \frac{\partial f}{\partial x}\Big|_{x=a,\, y=b}$. We can condense this system in matrix form as

$$\frac{d\mathbf{x}}{dt} = J\mathbf{x}, \tag{B.12}$$

where

$$\mathbf{x} := \begin{pmatrix} \delta x \\ \delta y \end{pmatrix}, \qquad J := \begin{pmatrix} f_x(x^*, y^*) & f_y(x^*, y^*) \\ g_x(x^*, y^*) & g_y(x^*, y^*) \end{pmatrix}. \tag{B.13}$$

Note that J is the *Jacobian* of the vector-valued function $\mathbf{f}(x, y) = (f(x, y)\, g(x, y))^\mathsf{T}$ evaluated at (x^*, y^*). This *linearization* of the dynamics around the fixed point produces a solvable system. The general solution is given by

$$\mathbf{x}(t) = c_1 e^{\lambda_1 t}\mathbf{v}_1 + c_2 e^{\lambda_2 t}\mathbf{v}_2, \tag{B.14}$$

where $\lambda_1, \lambda_2 \in \mathbb{C}$ and $\mathbf{v}_1, \mathbf{v}_2$ are the eigenvalues and eigenvectors, respectively, of J. We can then see clearly that if $\mathrm{Re}(\lambda_1) < 0$ and $\mathrm{Re}(\lambda_2) < 0$, then $\mathbf{x} \to 0$ as $t \to \infty$, and the fixed point is stable. Otherwise, (x^*, y^*) could be unstable, a saddle node, or a limit cycle. In general, we only need to calculate the eigenvalues of J to determine the system's qualitative behavior around a fixed point:

$$J\mathbf{v} = \boldsymbol{\lambda}\mathbf{v} \tag{B.15}$$

$$\Rightarrow (J - \boldsymbol{\lambda} I)\mathbf{v} = 0 \tag{B.16}$$

$$\Rightarrow |J - \boldsymbol{\lambda} I| = 0 \tag{B.17}$$

$$\Rightarrow (J_{11} - \lambda)(J_{22} - \lambda) - J_{12}J_{21} = 0 \tag{B.18}$$

$$\Rightarrow \lambda^2 - J_{11}\lambda - J_{22}\lambda + J_{11}J_{22} - J_{12}J_{21} = 0 \tag{B.19}$$

$$\Rightarrow \lambda^2 - (J_{11} + J_{22})\lambda + J_{11}J_{22} - J_{12}J_{21} = 0 \tag{B.20}$$

$$\Rightarrow \lambda^2 - \underbrace{\mathrm{Tr}(J)}_{T}\lambda + \underbrace{|J|}_{D} = 0 \tag{B.21}$$

$$\Rightarrow \lambda = \frac{T \pm \sqrt{T^2 - 4D}}{2}, \tag{B.22}$$

where eq. B.17 follows from the fact that if $\mathbf{v} \neq \mathbf{0}$, then $J - \boldsymbol{\lambda} I$ has a nonzero nullspace (is not full rank) and thus has a determinant of zero. Thus, for the fixed point to be stable and $\mathrm{Re}(\lambda_1), \mathrm{Re}(\lambda_2) < 0$, we need

$$T < 0, \quad D > 0. \tag{B.23}$$

A full description of qualitative behavior around fixed points is given in Figure B.1.

Fixed point	Tr[$\mathbf{J}$]	Det[$\mathbf{J}$]	Real part	Imaginary part
stable node	$T < 0$	$T^2 > 4D > 0$	$\mathrm{Re}(\lambda_\pm) < 0$	$\mathrm{Im}(\lambda_\pm) = 0$
stable spiral	$T < 0$	$4D > T^2 > 0$	$\mathrm{Re}(\lambda_\pm) < 0$	$\mathrm{Im}(\lambda_\pm) \neq 0$
unstable node	$T > 0$	$T^2 > 4D > 0$	$\mathrm{Re}(\lambda_\pm) > 0$	$\mathrm{Im}(\lambda_\pm) = 0$
unstable spiral	$T > 0$	$4D > T^2 > 0$	$\mathrm{Re}(\lambda_\pm) > 0$	$\mathrm{Im}(\lambda_\pm) \neq 0$
center (limit cycle??)	$T = 0$	$D > 0$	$\mathrm{Re}(\lambda_\pm) = 0$	$\mathrm{Im}(\lambda_\pm) \neq 0$
saddle	-	$D < 0$	$\mathrm{Re}(\lambda_+) > 0 > \mathrm{Re}(\lambda_-)$	$\mathrm{Im}(\lambda_\pm) = 0$
star/degenerate node	$T^2 = 4D$	$D \geq 0$	$\mathrm{Re}(\lambda_+) = \mathrm{Re}(\lambda_-)$	$\mathrm{Im}(\lambda_\pm) = 0$

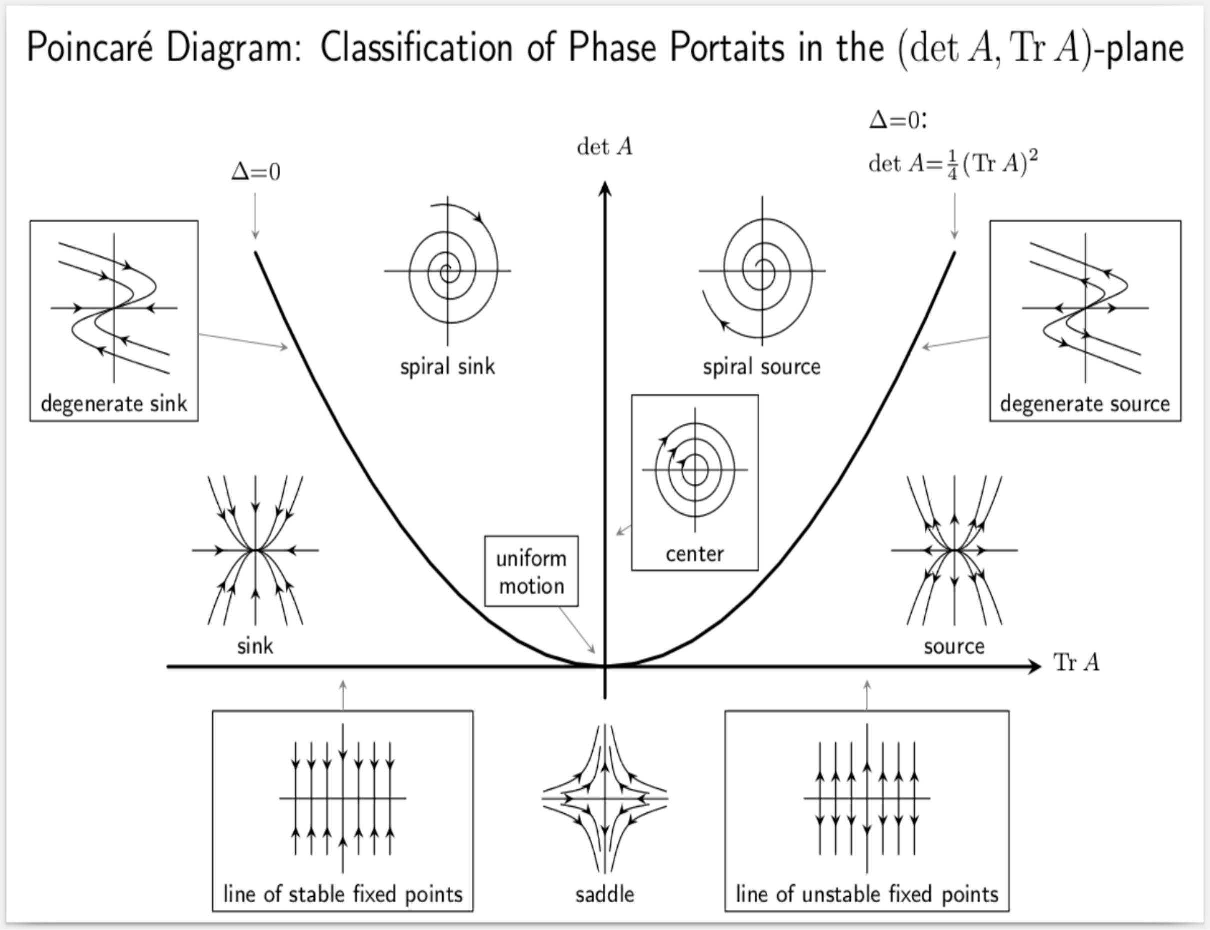

Poincaré Diagram: Classification of Phase Portaits in the (det A, Tr A)-plane
Δ=0
det A
Δ=0:
det $A=\frac{1}{4}$(Tr A)^2
degenerate sink
spiral sink
spiral source
degenerate source
center
uniform motion
sink
source
Tr A
line of stable fixed points
saddle
line of unstable fixed points

Figure B.1: Fixed point stability analysis (thank you Jorge).

C Fourier Transforms

Given a function $f(x)$ in space or time (e.g., x in units of m or s), one can express it in the frequency domain via its Fourier transform $F(\omega)$:

$$F(\omega) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i \omega x}\, dx \quad \text{(FT)} \tag{C.1}$$

$$f(x) = \int_{-\infty}^{\infty} F(\omega) e^{2\pi i \omega x}\, d\omega \quad \text{(IFT)}, \tag{C.2}$$

where ω is in the inverse units of x (e.g., m^{-1} or s^{-1}). When the units of ω don't matter to the given derivation, the 2π can be dropped. One must then simply rescale $\omega \to \omega/2\pi$ to interpret it as a frequency in inverse units of x (e.g. Hz for x in seconds). Some common/useful Fourier transforms to remember are given in Table 2.

A useful property of the Fourier transform is the *convolution theorem*, which states that the Fourier transform of the convolution of two functions f and g is equal to the product of their Fourier transforms:

$$\mathcal{F}\{f * g\} = \mathcal{F}\{f\}\mathcal{F}\{g\}, \tag{C.3}$$

$f(x)$	$F(\omega)$
1	$\delta(\omega)$
$\delta(x)$	1
e^{iax}	$\delta\left(\omega - \frac{a}{2\pi}\right)$
$\sin ax$	$\frac{1}{2i}(\delta(\omega + \frac{a}{2\pi}) + \delta(\omega - \frac{a}{2\pi}))$
$\frac{d^n}{dx^n} f(x)$	$(2\pi i\omega)^n F(\omega)$
e^{-ax^2}, $a > 0$	$\sqrt{\frac{\pi}{a}} e^{-\frac{\pi^2\omega^2}{a}}$
$\Theta(x)e^{-ax}$, $a > 0$	$\frac{1}{2\pi i\omega + a}$

Table 2: Useful Fourier transforms.

where $\mathcal{F}\{\cdot\}$ denotes the Fourier transform and $*$ denotes convolution:

$$(f * g)(x) \coloneqq \int f(x')g(x - x')\, dx'. \tag{C.4}$$

We can prove this as follows.

Proof. Let $h(x) \coloneqq f * g$. Then its Fourier transform $H(\omega)$ is

$$\begin{aligned}
H(\omega) &= \int h(x) e^{-2\pi i\omega x}\, dx && \text{(C.5)}\\
&= \int \int f(x')g(x - x')\, dx'\, e^{-2\pi i\omega x}\, dx && \text{(C.6)}\\
&= \int f(x') \int g(x - x') e^{-2\pi i\omega x}\, dx\, dx' && \text{(C.7)}\\
\text{let } u \coloneqq x - x' \;\Rightarrow &= \int f(x') \int g(u) e^{-2\pi i\omega (u+x')}\, du\, dx' && \text{(C.8)}\\
&= \int f(x') e^{-2\pi i\omega x'}\, dx' \int g(u) e^{-2\pi i\omega u}\, du && \text{(C.9)}\\
&= F(\omega)G(\omega). && \text{(C.10)}
\end{aligned}$$

□

D Assorted Useful Definitions and Identities

- $\frac{d}{dx}\tanh(x) = 1 - \tanh^2(x)$
- $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
- $\log(1 + x) \approx x$ for small x
- $(1 + x/n)^n \to e^x$, for $n \to \infty$
- $\frac{d}{dx}\text{arctanh}(x) = \frac{1}{1-x^2}$
- $\int_{-\infty}^{\infty} \delta(t - T) f(t)\, dt = f(T)$
- $\int_{-\infty}^{\infty} \delta(t - T)\, dt = \Theta(t - T) \Leftrightarrow \frac{d}{dt}\Theta(t) = \delta(t)$
- The entropy of a Gaussian is $\frac{1}{2}\log(2\pi e\sigma^2)$
- The Gaussian CDF is $\Phi(z) \coloneqq \int_{-\infty}^{z} e^{-t^2/2}/\sqrt{2\pi}\, dt$. Some useful properties to keep in mind are
  - Probability: $1 - \Phi(r) = P[z > r]$
  - Symmetry: $1 - \Phi(r) = \Phi(-r)$
  - Sigmoidal, with $\Phi(0) = 1/2$
  - In general, draw out the Gaussian and bounds to help convert between probabilities and cdfs
- The parity of a function: f is *even* if $f(x) = f(-x)$, and *odd* if $f(x) = -f(-x) \Leftrightarrow -f(x) = f(-x)$.

- Differentation/integration switch the parity of a function.

- The integral of an even function from $-\infty$ to ∞ is twice the integral from 0 to ∞, the integral of an odd function from $-\infty$ to ∞ is zero.

- The $\text{sign}(\cdot)$ function is an odd function.

- $\text{sign}^2(x) = 1$ always.