Chapter 2: Neural primitives of thought

This chapter introduces some simple mathematical models of neurons. Starting with the dynamics of the membrane potential, we derive more abstract models of neural activity and assess their computational power. These models serve as the primitives from which we will construct neural implementations of cognitive algorithms.

The description of neurophysiology in Chapter 1 was a cartoon—but a *useful* cartoon. What makes it useful? Assembling cognitive algorithms from neural primitives requires us to abstract away from many details. We now consider a simple formalization that captures key aspects of this abstraction. We then develop progressively more abstract models which will be useful for some cognitive applications that we consider later in the book.

1 *A simple model: the leaky integrate-and-fire neuron*

The leaky integrate-and-fire (LIF) model formalizes the postsynaptic membrane as a resistor-capacitor circuit that can be charged up by input current (with some leak) and then discharged when a spike occurs. The membrane potential $\mu(t)$ obeys the following dynamics:

$$C\dot{\mu} = \frac{\mu^0 - \mu(t)}{R} + I(t), \tag{1}$$

where $I(t)$ is the input current at time t, $\mu(t)$ is the membrane potential, $\dot{\mu}$ is its temporal derivative, μ^0 is the resting potential (the membrane potential when input current is 0), R is the membrane resistance (determined by the number of open ion channels), and C is the membrane capacitance (determined by the surface area of the membrane). When $\mu(t)$ crosses a threshold θ, a spike is emitted and the membrane potential is reset to $\mu^{\text{reset}} < \mu^0$, contributing to a brief refractory period during which spiking is suppressed. These dynamics are illustrated in Figure 1.

In the limit $R \to \infty$ (no leak), the LIF neuron becomes a perfect integrator until the threshold is reached. We will examine this case further in the next chapter.

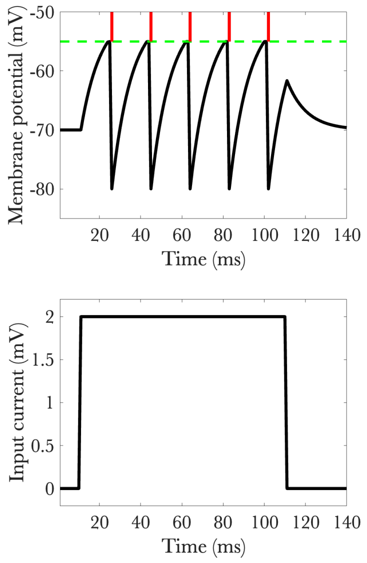


Figure 1: **Leaky integrate-and-fire neuron with step input.** (Top) Membrane potential shown in black, firing threshold in green, spikes in red. (Bottom) Input current. The parameters were chosen to be physiologically plausible: $C = 1, R = 10, \theta = -55, \mu^{\text{reset}} = -80, \mu^0 = -70, I(t) = 2$.

Synaptic integration can be incorporated into the LIF model by making the input current a function of presynaptic spikes. A common assumption is that this function is linear:

$$I(t) = \sum_d w_d z_d(t), \tag{2}$$

where $z_d(t) \in \{0, 1\}$ is the spike train of presynaptic neuron d (i.e., $z_d(t) = 1$ when neuron d spikes at time t), and w_d is its synaptic strength. Linear integration isn't the whole story, but it's a reasonably good approximation, particularly for dendrites near the soma (Branco and Häusser, 2011) and inputs impinging on the small protrusions known as *dendritic spines* (Araya et al., 2006).

For data showing linear integration, see Cash and Yuste (1998, 1999). Studies have also found both sublinear and supralinear integration (reviewed in Grienberger et al., 2015).

We can think about the LIF model as a simple signal processing unit. To make this explicit, we rewrite the membrane potential as a linear filter:

$$\mu(t) = \mu^0 + \int_0^t K_{\text{out}}(t')z(t-t')dt' + \int_0^t K_{\text{in}}(t')I(t-t')dt', \qquad (3)$$

where $z(t)$ denotes the postsynaptic spike train. The input kernel $K_{\text{in}}(t) = \frac{1}{C}\exp\left(-\frac{t}{\tau}\right)$ specifies the dependence of the membrane potential on the history of input currents, where $\tau = RC$ is the membrane time constant. The output kernel $K_{\text{out}}(t) = (\mu^{\text{reset}} - \theta)\exp\left(-\frac{t}{\tau}\right)$ specifies the dependence of the membrane potential on the history of spikes: a downward jump after each spike. In some cases, we will focus on the "subthreshold regime" where the input current is small relative to the threshold. This will license us to neglect the output kernel and just focus on the linear dynamics of the membrane potential.

For a more detailed treatment of the LIF neuron and related models, see Gerstner et al. (2014).

To illustrate, let's consider a constant input current I_{const}. If we ignore the threshold, the membrane potential at time t is given by a convex combination of the resting potential μ^0 and the asymptotic potential $\mu^\infty = \mu^0 + RI_{\text{const}}$ (reached in the limit $t \to \infty$):

$$\mu(t) = \mu^0 \exp\left(-\frac{t}{\tau}\right) + \mu^\infty \left[1 - \exp\left(-\frac{t}{\tau}\right)\right]. \qquad (4)$$

When a constant input current yields an asymptotic membrane potential below the spiking threshold ($\mu^\infty < \theta$), we will designate it a *subthreshold* input. When we talk about the subthreshold regime, we're mainly talking about the case of subthreshold inputs.

This simple LIF model ignores several aspects of real neurons. First, spiking in real neurons drives the membrane potential to around 40 mV before declining to the reset potential; this rise and fall after the threshold is crossed takes about 2 ms. The LIF model assumes that the membrane potential is instantaneously reset after the threshold is crossed. Second, some real neurons exhibit longer-timescale dynamics such as adaptation, where the interval between spikes becomes progressively longer. The LIF model is "memoryless" in the sense that it does not keep around any information about spike history beyond the current state of the membrane potential, and therefore cannot capture adaptation. Third, some real neurons fire sequences of spikes after the threshold is crossed, either periodically ("bursts") or aperiodically ("stutters"). Again, the memoryless property of the LIF model prohibits it from capturing these sequential phenomena. It is possible to generalize Eq. 3 to capture many of these phenomena, though we will not explore such generalizations here (see Gerstner et al., 2014).

2 *Noise*

The LIF model is deterministic. This will produce highly regular spiking activity in response to a constant input. Indeed, a high degree of regularity is observed when cells are recorded in a slice preparation, where the inputs can be precisely controlled (Mainen and Sejnowski, 1995). In contrast, neurons spike in a highly irregular manner when recorded *in vivo* (i.e., by inserting electrodes into the intact brain), following either current injection or the presentation of a stimulus (Figure 2). Holt et al. (1996) argue that irregularity arises from "random" synaptic background activity from other inputs, which is present *in vivo* but absent in the slice preparation. Note that this is random (i.e., unpredictable) only from the point of view of the experimenter who is not able to measure all of the inputs. Regardless of how we characterize the origin of randomness, the end result is that spiking is *effectively* random when we are recording from neurons in the intact brain; we need a model that captures this randomness.

A slice preparation immerses brain slices in artificial cerebrospinal fluid, where they can be recorded or stimulated.

Other sources of randomness are thermal noise and fluctuations in the number of open vs. closed ion channels.

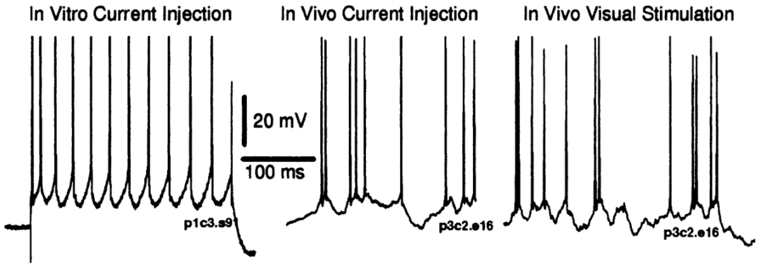


Figure 2: **Voltage traces of neurons recorded in visual cortex**. Reproduced from Holt et al. (1996).

We can transform the LIF model into a stochastic differential equation by adding membrane potential noise $\epsilon(t)$, as illustrated in Figure 3:

$$C\dot{\mu} = \frac{\mu^0 - \mu(t)}{R} + I(t) + \epsilon(t). \tag{5}$$

If the noise reflects the summation of many independent excitatory and inhibitory currents that approximately balance each other out, then the Central Limit Theorem implies that the noise should be approximately Gaussian-distributed: $\epsilon(t) \sim \mathcal{N}(0, \sigma^2)$. The standard deviation σ determines the typical amplitude of the noise. The expected membrane potential $\bar{\mu}(t) = \mathbb{E}[\mu(t)]$ in the subthreshold regime is

Zero-mean, uncorrelated Gaussian noise is known as *white noise*.

identical to the membrane potential of the noiseless LIF. The variance of the membrane potential quickly approaches a steady-state value proportional to σ^2 after a reset.

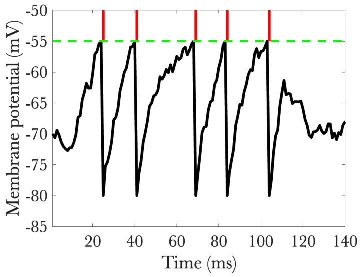


Figure 3: **Leaky integrate-and-fire neuron with step input and membrane potential noise**. Same simulation setup as in Figure 1, with the addition of Gaussian noise to the membrane potential dynamics ($\sigma = 1$).

The assumption of a noisy membrane potential is important for explaining the irregular nature of spiking. When presented with a constant subthreshold input current, neurons tend to spike irregularly. However, if an additional white noise component is superimposed on the constant current, spiking is highly regular when the same noise is presented repeatedly (Mainen and Sejnowski, 1995). To understand this in terms of the stochastic LIF model, we first note that when the membrane time constant is short, small membrane potential fluctuations will have relatively little effect on spiking because they decay quickly—the integration time window is short. Spiking occurs (precisely and reliably) when relatively large fluctuations transiently drive the potential above the threshold. The source of these large fluctuations is the coincidence of synaptic inputs (i.e., multiple presynaptic neurons spiking synchronously). In this sense, the postsynaptic neuron acts a coincidence detector in the subthreshold regime (König et al., 1996).

3 *The linear-nonlinear Poisson model*

The LIF model is a useful mechanistic abstraction, but it can also be challenging to work with analytically in some settings. In this section, we derive a *statistical* abstraction which captures important aspects of real neurons despite sacrificing the mechanistic description of spiking generation. Following Plesser and Gerstner (2000), the first step is to transform the model into a temporal point process, a collection of random variables (spikes in this case) with probabilities specified as a function of time. We will then use the point process to express a static model over an integration window.

See Gerstner et al. (2014) for more details and extensions.

If we ignore the boundary condition imposed by the spiking threshold and use the steady-state value of the membrane potential variance, the probability of crossing the threshold in the window $[t, t + \Delta]$ is given by:

$$p(\mu(t') = \theta | t' \in [t, t + \Delta]) \propto \Delta \exp\left(-\frac{[\bar{\mu}(t) - \theta]^2}{\sigma^2}\right). \qquad (6)$$

Dividing both sides by Δ yields the approximate intensity function:

This follows from first-passage approximations to the stochastic LIF model.

$$\rho(t) \propto \exp\left(-\frac{[\bar{\mu}(t) - \theta]^2}{\sigma^2}\right), \qquad (7)$$

which specifies the expected firing rate at time t.

The Poisson distribution for count variable x is $p(x) = \frac{\rho^x e^{-\rho}}{x!}$, where $\rho \geq 0$ is the intensity.

The final step is to define a point process parametrized by the intensity function. If we assume that the number of spikes within any interval $[t, t+\Delta]$ is Poisson-distributed with rate $\int_t^{t+\Delta} \rho(t')dt'$, and that the number of spikes is independent across disjoint intervals, then we arrive at the inhomogeneous Poisson process with intensity function $\rho(t)$. Because the intensity function is a composition of linear synaptic integration with a static nonlinearity, the complete model is known as the *linear-nonlinear Poisson* model.

Poisson spiking is widely used in many models that we will discuss throughout this book. It exhibits two distinctive properties that are similar to real neurons:

- The ratio between the variance and the mean of the spike count (also known as the *Fano factor*) is close to 1 (Tolhurst et al., 1983), although this relationship tends to break down for high spike counts (Figure 4, left).
- The interspike interval is approximately exponentially distributed (Figure 4, right). Note, however, that the refractory period following spikes implies that the very short interspike intervals are not possible; because it is peaked at 0, the exponential distribution always overestimates the frequencies of these short intervals.

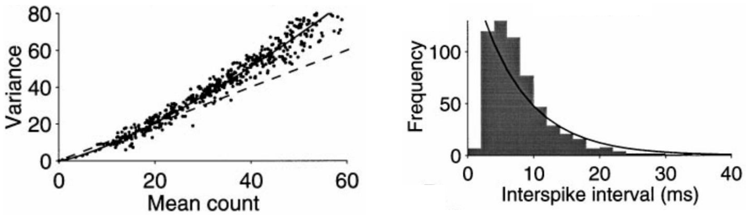


Figure 4: **Spiking statistics in area MT**. Left: Variance of the spike count plotted as a function of the mean spike count. The dashed line shows the relationship expected for spikes generated from a Poisson process. Right: Histogram of interspike intervals. The curve shows the best-fitting exponential distribution. Adapted from Shadlen and Newsome (1998).

It has been argued (controversially) that spike timing is too irregular to propagate reliable information (Shadlen and Newsome, 1998; London et al., 2010). If this is true, neurons should only care about the firing rates of their inputs (not individual spike times), calculated over some appropriately long integration window. This motivates the adoption of a "static" abstraction, where we plug in the steady-state value of the membrane potential, assuming constant input over the integration window:

$$\rho(\infty) \propto \exp\left(-\frac{[\mu^\infty - \theta]^2}{\sigma^2}\right). \tag{8}$$

We will sometimes use the static abstraction to think about neural representations that are relatively stable across short windows of time. Nevertheless, we must keep in mind that these representations are always constructed dynamically.

The results presented so far rely on particular assumptions about membrane potential noise, as well as some approximations. Modeling applications often make other assumptions about the form of the intensity function. For example, we can generalize Eq. 7:

$$\rho(t) = b_1 \exp\left(-\frac{[\bar{\mu}(t) - \theta]^{b_2}}{\sigma^2}\right), \tag{9}$$

where b_1 and b_2 are free parameters. A choice which fits data well is $b_1 = 1/\tau$ and $b_2 = 1$ (Jolivet et al., 2006). This model is "phenomenological" in the sense that it is not derived from underlying mechanisms, but it can still serve as an accurate description of spiking data.

4 *Spikes vs. rates*

See Brette (2015) for a more comprehensive discussion of the issues in this section.

In motivating the static abstraction, we referred to the purported irregularity of spike timing. We now unpack and scrutinize this argument. When an electrophysiologist records from a single neuron over multiple repetitions of a stimulus, a typical observation is that the precise spike times are highly variable across repetitions (Figure 5, top). By temporally binning the spikes and averaging over repetitions, we obtain the *peri-stimulus time histogram* (PSTH). The example shown in Figure 5 (bottom) reveals well-behaved *average* response dynamics despite apparently low reproducibility of spike timing across repetitions. A downstream neuron could easily decode the stimulus orientation from the PSTH even if it had no access to the underlying spikes.

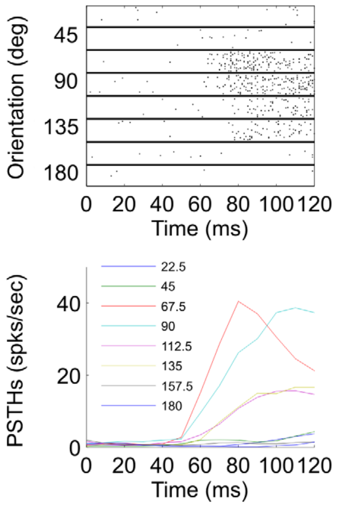


Figure 5: **Neural responses in primary visual cortex**. (Top) Raster plot of spikes for a single neuron in response to repetitions of drifting sinusoidal gratings with different orientations. Each black point is a single spike. (Bottom) The peri-stimulus time histogram obtained by binning and averaging the spikes across repetitions. Each line corresponds to a different stimulus orientation. Reproduced from Shriki et al. (2012).

Of course, neurons never have access to the actual PSTH on a single trial, unless we assume that the tuning of its inputs are homogeneous (i.e., they all have the same tuning functions—a questionable assumption), so that averaging across multiple trials for a single neuron and averaging across multiple neurons for a single trial would be equivalent. When we say that the neural code is rate-based, what we really mean is that neurons are integrating information across time in a such a way that information about precise spike timing is discarded—only the presynaptic firing rates matter for determining the postsynaptic response.

There is evidence that precise spike timing is *not* discarded. First, spike timing can be highly reproducible under certain conditions (e.g., with time-varying input patterns; Mainen and Sejnowski, 1995; de Ruyter van Steveninck et al., 1997)—a necessary precondition for using spike timing to communicate information reliably. Second, discarding spike timing information from retinal output cells reduces the performance of an optimal stimulus decoder to levels well below animal behavioral performance, whereas a model that incorporates spike timing is able to match animal performance (Jacobs et al., 2009).

Similar exercises in somatosensory cortex also demonstrated that spike timing information improves the match to behavioral performance (Mackevicius et al., 2012; Zuo et al., 2015).

Because this is the first time we're talking about decoders, we'll unpack this point further. A stimulus decoder is a distribution $p(s|x)$, where s is the stimulus and x is some measure of neural activity. In this book, we will talk about decoders in two ways. One is as an analysis tool in the hands of neuroscientists. Showing that we can decode stimulus information from neural activity means that the information must be represented by those neurons (though its computational role may be unclear). The second way, dealt with in later chapters, is as a model of neural computation: some downstream neurons can be conceptualized as decoding information from upstream neurons. An optimal decoder uses Bayes' rule to obtain $p(s|x)$.

If a neuroscientist's decoder matches behavioral performance, it means that it is doing approximately as well as the brain's decoder. This provides us with a recipe for reverse engineering what information is being used by the brain's decoder. The examples given above show that using spike timing yields a better match to behavior than firing rate, suggesting that spike timing is used by the brain's decoder.

On the other hand, there is relatively sparse evidence that precise spike timing is important for predicting detailed behavioral events. In contrast, there is abundant evidence that firing rates can predict behavioral events on single trials. For example, the firing rates of single neurons in the motion processing area MT predict a monkey's decisions in a motion discrimination task (Britten et al., 1996). Even response times can be predicted based on firing rates (e.g., Cook and Maunsell, 2002; Roitman and Shadlen, 2002), indicating that they convey enough temporal information to identify behavioral events at the relevant timescale. This doesn't mean that spike timing is never behaviorally relevant, only that we have more evidence from firing rate data. Conspicuously, the best data for the behavioral relevance of spike timing comes from early sensory processing, before timing information has been erased over multiple synaptic transmissions. By the time signals reach higher-level cortex, firing rate may be the only reliable source of information.

See Chapters 4 and 7 for more details about motion discrimination in the brain.

Another place where spike timing seems to matter for behavior is at the motor periphery (Srivastava et al., 2017).

Models like the LIF neuron can operate as integrators (when the membrane time constant is long) and as coincidence detectors (when the membrane time constant is short), as illustrated in Figure 6. The integration mode effectively discards spike timing information, whereas the coincidence detection mode relies on precisely-timed presynaptic spikes in order to produce a postsynaptic spike. A short membrane time constant means that multiple presynaptic spikes need to arrive near-simultaneously in order to push the membrane potential above the firing threshold.

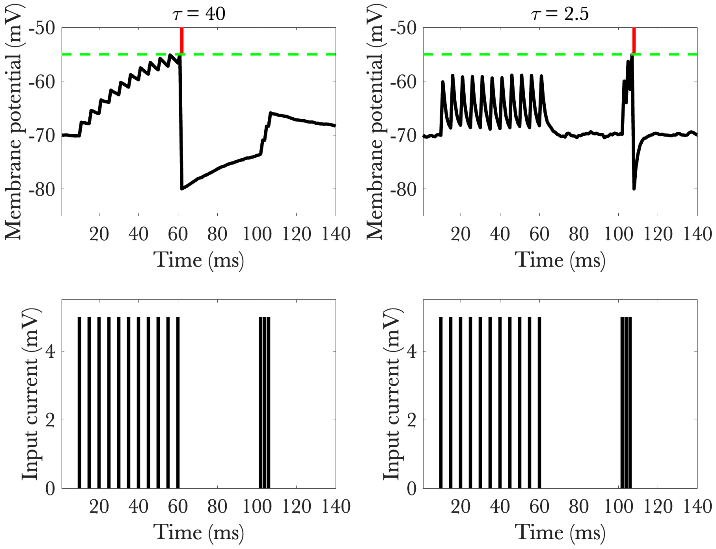


Figure 6: **Leaky integrate-and-fire neuron with pulsed inputs and different membrane time constants**. (Left) Integration mode when the time constant is long ($R = 20, C = 2$). (Right) Coincidence detection mode when the time constant is short ($R = 5, C = 0.5$). Top panels show the membrane potential and spikes; bottom panels show the input current (corresponding to discrete spikes). The first set of spikes is spaced, while the second set is massed. Membrane potential noise is set to be small ($\sigma = 0.1$) in these simulations.

Some of the most compelling evidence for the coincidence detection mode comes from studies of sound localization (Ashida and Carr, 2011). Many animals localize sound direction by comparing the time difference between auditory signals arriving at each ear. This is implemented neurally by a set of "delay lines" (auditory nerve fibers with a range of conduction times) which function as a map of sound arrival time. The maps for each ear converge on neurons that are sensitive to the coincidence of their inputs, allowing a downstream circuit to identify the sound direction with a level of accuracy that requires microsecond precision in the representation of time delays—a prohibitive feat for systems that rely on neurons with slow time constants.

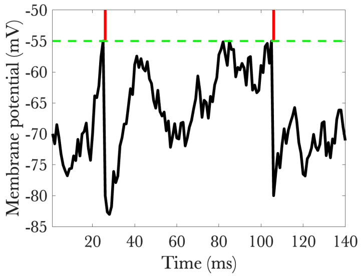


Figure 7: **Leaky integrate-and-fire neuron with weak step input and large membrane potential noise**. Same simulation setup as in Figure 1, with the addition of Gaussian noise to the membrane potential dynamics ($\sigma = 2.5$) and a weak step input ($I(t) = 1$). When excitation and inhibition are exactly balanced, $I(t) = 0$.

When excitation and inhibition of a neuron are approximately balanced, the membrane potential approaches a random walk, moving stochastically up or down until the threshold is reached and the potential is reset (Figure 7). This is known as the *fluctuation-driven regime*, because spiking is driven by random fluctuations in the membrane potential, with highly irregular interspike intervals. A short membrane time constant would be disastrous in this regime, because the spikes would essentially be propagating noise rather than signal. In other words, a coincidence detector would only be detecting spurious coincidences. Whatever signal exists will be weak, necessitating long time constants to average out the noise. As pointed out by

Shadlen and Newsome (1994), this implies that a precise spike timing code is implausible in the fluctuation-driven regime. Cortical circuits do in fact exhibit approximate balance of excitation and inhibition (Wehr and Zador, 2003; Okun and Lampl, 2008). Thus, integration may be a typical operational mode for cortical neurons.

Excitation-inhibition balance emerges naturally in sparsely connected networks of neurons with strong synapses (Van Vreeswijk and Sompolinsky, 1996).

In subsequent chapters we will explore both rate and spike timing codes. It's plausible that both kinds of codes are used by the brain in different circumstances. Our goal will be to understand what those circumstances are and the computational logic underlying them.

5 *Tuning functions*

In many neuroscience experiments, an animal is presented with a stimulus (e.g., an image, sound, etc.) while the firing rates of neurons are measured. This allows the experimenter to plot the average firing rate as a function of some stimulus parameter—a *tuning function* (or *receptive field*). When the stimulus parameter is one-dimensional, this is called a *tuning curve*. For example, some neurons in primary visual cortex are tuned to the local orientation of edges in a specific region of retinotopic space: their firing rates are bell-shaped functions of orientation, with a peak at particular orientations (Fig. 8).

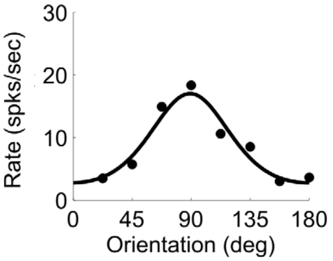


Figure 8: **Orientation-tuned neuron in primary visual cortex**. This is the same neuron shown in Figure 5. Reproduced from Shriki et al. (2012).

In the parlance of Bayesian decision theory, the stimulus parameter is a state variable (i.e., the cause of sensory input). We will therefore use the notation $f_d(s)$ to denote the average firing rate of neuron d in response to state s. The state is not typically available directly to the brain, but must instead be inferred; this is the topic of Chapters 4 and 5.

The tuning curve is a useful abstraction because it tells us something about how the brain encodes state information. We must remember that it is not a mechanistic description of the causal events that go from the (typically hidden) state to the firing rate of a neuron. One of our goals will be to explain how particular tuning functions arise, both mechanistically and in terms of general design principles.

Another thing to remember is that the tuning of single neurons can be highly misleading about the nature of neural computation. Populations of neurons do much of the computational work in the brain, and the relevant information is often distributed in complex ways across many neurons. In other words, tuning functions are generally meaningful only in the context of the roles they play within a population. We will see examples of this throughout the book.

6 Universality

By asserting that these simple neuron models function as *neural primitives of thought*, we are making a promise that they can be used to construct computational systems capable of complex cognition (inference, decision making, learning, memory, attention, etc.). Beyond actually constructing such systems, which is what we'll do in the rest of this book, we can ask a more general question: what is the class of computational systems that we can construct with these neural primitives? In other words, what are the limits of such systems? In the following sub-sections, we define 3 distinct notions of universality, which we connect to the LIF model and related models.

6.1 Logical universality

The first foray into the question of universality was undertaken by McCulloch and Pitts (1943), who started from the idea that a spike signals the truth value of a proposition represented by the neuron. Their model of a neuron was very similar to the LIF model introduced above, except that it operated in discrete time. At each time step, a neuron receives a binary pattern that represents the truth values for a set of input propositions (represented by the presynaptic neurons). As in the LIF, the inputs are linearly weighted by synaptic strengths, followed by a thresholding operation, $z(t) = \phi(I(t) - \theta)$, where θ is a threshold parameter and $\phi(\cdot)$ is an "activation function" (in this case a step function):

In essence, we can think of the McCulloch-Pitts neuron as the limit of an LIF neuron with $\tau \to 0$, so that it processes its inputs instantaneously.

Later we will consider more general activation functions.

$$\phi(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0. \end{cases} \tag{10}$$

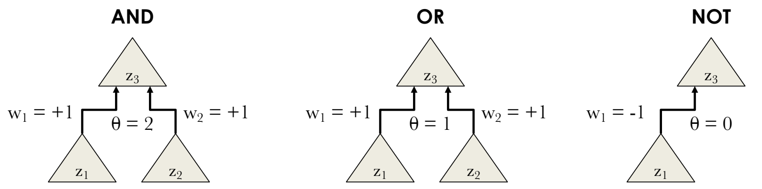
McCulloch and Pitts understood this neuron model as implementing a logical function. A variety of logical functions can be implemented with different choices of thresholds and weights, as illustrated in Figure 9.

More complex functions can be built from these simple elements. For example, a NAND ("not and") function can be built by composing the AND and NOT functions. This construction is significant because the NAND function is a universal element—all other Boolean functions can be constructed out of only NAND functions (Sheffer, 1913).

Logical universality as defined here is also known as *functional completeness*.

6.2 Computational universality

Logical universality says that we can implement any logical function with a set of primitives, but it does not say that we can implement



z_1	z_2	z_3
1	1	1
1	0	0
0	1	0
0	0	0

z_1	z_2	z_3
1	1	1
1	0	1
0	1	1
0	0	0

z_1	z_3
1	0
1	0
0	1
0	1

Figure 9: **Logical functions implemented with McCulloch-Pits neurons**. The truth table for each function is shown below the corresponding circuit.

any computation. To appreciate the difference, consider the following simple problem: determine whether the first and last inputs in a sequence are the same. If the sequence has a fixed length, we can easily construct a circuit out of McCulloch-Pitts neurons that solves the problem. However, what happens when the sequence can be of indeterminate length? Here the McCulloch-Pitts neural circuit runs into trouble. It could run out of neurons if the sequence is long enough—a finite network cannot handle arbitrarily long sequences without additional memory. Even if the sequences can be relatively short, not knowing the sequence length in advance means that you would need a separate circuit for each length (again potentially running out of neurons).

A general solution is to equip the system with a read/write memory (Figure 10, left), which would allow it to write the first input to memory and then compare the last input to the stored memory (ignoring everything else in between). This functionality is beyond the capabilities of simple McCulloch-Pitts circuits, and is in fact the key ingredient to building universal computers. Turing (1936) formalized this idea using what is now known as the *Turing machine*, a device that writes to and reads from a tape of unbounded length, while moving along the tape (or halting) according to rules specified by a transition table. Despite its simplicity, the Turing machine can implement a universal computer in the following sense: there exists a Turing machine that can compute any function which can be computed in a finite number of steps based on a finite set of instructions.

Gallistel and King (2011) argue that the absence of a read/write memory is the fundamental weakness of standard neural network models in neuroscience.

A McCulloch-Pitts circuit with an unbounded read/write memory can achieve computational universality. This is essentially a Turing machine where the transition table is specified by the weights and thresholds of the circuit (as well as input-dependent rules for reading and writing at different memory locations). It remains a fascinating and unresolved question how biology might have implemented such a read/write memory.

The idea of equipping neural circuits with read/write memories has been exploited in modern machine learning to train powerful systems (Graves et al., 2016).

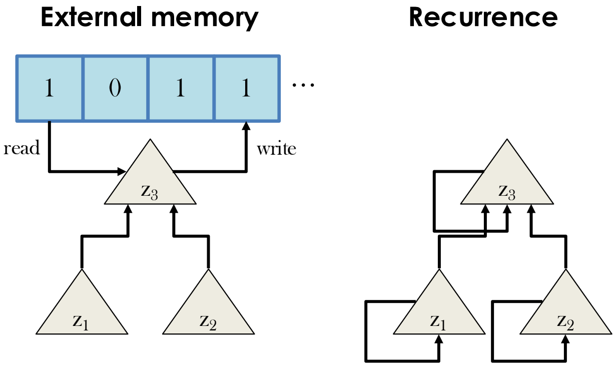


Figure 10: **Motifs for computationally universal circuits**. (Left) Adding an external memory to the McCulloch-Pitts circuit. (Right) Adding recurrent connections.

Another approach, pioneered by Siegelmann and Sontag (1992), is to adopt a neuron model with rational-valued outputs and weights, and to replace the step function non-linearity with a linear function that saturates below 0 and above 1. Recurrent circuits built out of such neurons (where at each time step the previous activation is fed back into the circuit; Figure 10, right) are also computationally universal. We will discuss recurrent networks further in later chapters.

A similar result can also be proven for neurons with a smooth "sigmoidal-like" (S-shaped) activation function (Kilian and Siegelmann, 1996).

6.3 *Universal function approximation*

Logical and computational universality apply to discrete components (inputs, outputs, and machines). In some settings, we care about approximating functions between continuous inputs and outputs using machines made out of continuous components. Here we ask whether it is possible to build a universal function approximator (i.e., an approximator that can get arbitrarily close to any target function within some class, as detailed below) using simple neural components of the sort that we've already seen, albeit with continuous inputs, outputs, weights, and thresholds. This research program is often associated with circuits of *perceptrons* (Rosenblatt, 1958), which are generalizations of the McCulloch-Pitts neuron model.

Note that if the function approximator is implemented on a digital computer, then inevitably it too must be built out of discrete components (though with possibly arbitrarily high precision).

The typical setup (see Pinkus, 1999, for a review) is to assume that the input to a circuit is a "compact" subset of $\mathbb{R}^N$ (the N-dimensional space of real numbers). Informally, compactness means that there are no holes or excluded boundaries. As before, we'll assume that a neuron's activation is given by a linear combination of its inputs (denoted here by x) followed by a nonlinear activation function $\phi(\cdot)$ with threshold θ. For notational simplicity, we will exclude the time index:

$$z(x) = \phi(\textstyle\sum_n w_n x_n - \theta). \tag{11}$$

In a perceptron, $\phi(x)$ can be a smooth continuous function, such as a sigmoid, $\phi(x) = 1/(1 + e^{-x})$. Now let's consider a circuit constructed out of such perceptrons. We want to know what class of functions this circuit can approximate, in the sense that the maximum absolute difference between the target function output and the circuit output (across all possible inputs) is always less than some small constant ϵ.

A classic result (Funahashi, 1989) is that all continuous functions from a compact subset of $\mathbb{R}^N$ to $\mathbb{R}^M$ can be approximated within ϵ by a linear combination of (possibly infinitely many) perceptrons, provided $\phi(x)$ is nonconstant, bounded, and monotonic. In other words, a 3-layer feedforward circuit consisting of N inputs feeding into a set of perceptrons (the "hidden" layer), which in turn feed into M linear units, can be designed such that it gets arbitrarily close to any continuous function, provided the weights and thresholds are chosen appropriately (how to actually find these parameter values is the major problem of neural network learning, discussed further in Chapter 9). The number of required perceptrons in the hidden layer depends on the function being approximated.

These requirements are satisfied by a sigmoidal activation function, for example.

Many subsequent results have generalized this setup in various ways. For example, instead of arbitrary width (i.e., the number of perceptrons in the hidden layer is unbounded), one can get universal function approximation with fixed width and arbitrary depth (adding more layers; see Gripenberg, 2003). It turns out that depth can be more useful than width in that the same approximation accuracy can in certain cases be achieved with fewer neurons when depth rather than width is increased (Lu et al., 2017).

7 *Conclusion*

This chapter has surveyed a set of "neural primitives" with which we will implement computational models of cognition. These primitives are (in principle) powerful enough to implement any logical function, digital computation, or smooth continuous function. In subsequent chapters, we will show how they can be used to implement the elements of Bayesian decision theory. While the neural primitives are fairly dramatic simplifications of real neurons, we will see that they can nonetheless capture a wide range of data.

Study questions

1. What are the computational advantages and disadvantages of using spike times vs. firing rates?
2. Why are tuning curves informative but potentially misleading when considered in isolation? How does population coding complicate the interpretation of single-neuron tuning?

3. Compare the three notions of universality (logical, computational, and function approximation). How do they differ in scope and implications?

References

Araya, R., Eisenthal, K. B., and Yuste, R. (2006). Dendritic spines linearize the summation of excitatory potentials. *Proceedings of the National Academy of Sciences*, 103:18799–18804.

Ashida, G. and Carr, C. E. (2011). Sound localization: Jeffress and beyond. *Current Opinion in Neurobiology*, 21:745–751.

Branco, T. and Häusser, M. (2011). Synaptic integration gradients in single cortical pyramidal cell dendrites. *Neuron*, 69:885–892.

Brette, R. (2015). Philosophy of the spike: rate-based vs. spike-based theories of the brain. *Frontiers in Systems Neuroscience*, 9:140675.

Britten, K., Newsome, W., Shadlen, M., Celebrini, S., and Movshon, J. (1996). A relationship between behavioral choice and the visual responses of neurons in macaque MT. *Visual Neuroscience*, 13:87–100.

Cash, S. and Yuste, R. (1998). Input summation by cultured pyramidal neurons is linear and position-independent. *Journal of Neuroscience*, 18:10–15.

Cash, S. and Yuste, R. (1999). Linear summation of excitatory inputs by CA1 pyramidal neurons. *Neuron*, 22:383–394.

Cook, E. P. and Maunsell, J. H. (2002). Dynamics of neuronal responses in macaque MT and VIP during motion detection. *Nature Neuroscience*, 5:985–994.

de Ruyter van Steveninck, R. R., Lewen, G. D., Strong, S. P., Koberle, R., and Bialek, W. (1997). Reproducibility and variability in neural spike trains. *Science*, 275:1805–1808.

Funahashi, K.-I. (1989). On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2:183–192.

Gallistel, C. R. and King, A. P. (2011). *Memory and the Computational Brain: Why Cognitive Science Will Transform Neuroscience*. John Wiley & Sons.

Gerstner, W., Kistler, W. M., Naud, R., and Paninski, L. (2014). *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*. Cambridge University Press.

Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwińska, A., Colmenarejo, S. G., Grefenstette, E., Ramalho, T., Agapiou, J., et al. (2016). Hybrid computing using a neural network with dynamic external memory. *Nature*, 538:471–476.

Grienberger, C., Chen, X., and Konnerth, A. (2015). Dendritic function in vivo. *Trends in Neurosciences*, 38:45–54.

Gripenberg, G. (2003). Approximation by neural networks with a bounded number of nodes at each level. *Journal of Approximation Theory*, 122:260–266.

Holt, G. R., Softky, W. R., Koch, C., and Douglas, R. J. (1996). Comparison of discharge variability in vitro and in vivo in cat visual cortex neurons. *Journal of Neurophysiology*, 75:1806–1814.

Jacobs, A. L., Fridman, G., Douglas, R. M., Alam, N. M., Latham, P. E., Prusky, G. T., and Nirenberg, S. (2009). Ruling out and ruling in neural codes. *Proceedings of the National Academy of Sciences*, 106:5936–5941.

Jolivet, R., Rauch, A., Lüscher, H.-R., and Gerstner, W. (2006). Predicting spike timing of neocortical pyramidal neurons by simple threshold models. *Journal of Computational Neuroscience*, pages 35–49.

Kilian, J. and Siegelmann, H. T. (1996). The dynamic universality of sigmoidal neural networks. *Information and Computation*, 128:48–56.

König, P., Engel, A. K., and Singer, W. (1996). Integrator or coincidence detector? the role of the cortical neuron revisited. *Trends in Neurosciences*, 19:130–137.

London, M., Roth, A., Beeren, L., Häusser, M., and Latham, P. E. (2010). Sensitivity to perturbations in vivo implies high noise and suggests rate coding in cortex. *Nature*, 466:123–127.

Lu, Z., Pu, H., Wang, F., Hu, Z., and Wang, L. (2017). The expressive power of neural networks: A view from the width. *Advances in Neural Information Processing Systems*, 30.

Mackevicius, E. L., Best, M. D., Saal, H. P., and Bensmaia, S. J. (2012). Millisecond precision spike timing shapes tactile perception. *Journal of Neuroscience*, 32:15309–15317.

Mainen, Z. F. and Sejnowski, T. J. (1995). Reliability of spike timing in neocortical neurons. *Science*, 268:1503–1506.

McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5:115–133.

Okun, M. and Lampl, I. (2008). Instantaneous correlation of excitation and inhibition during ongoing and sensory-evoked activities. *Nature Neuroscience*, 11:535–537.

Pinkus, A. (1999). Approximation theory of the MLP model in neural networks. *Acta Numerica*, 8:143–195.

Plesser, H. E. and Gerstner, W. (2000). Noise in integrate-and-fire neurons: from stochastic input to escape rates. *Neural Computation*, 12:367–384.

Roitman, J. D. and Shadlen, M. N. (2002). Response of neurons in the lateral intraparietal area during a combined visual discrimination reaction time task. *Journal of Neuroscience*, 22:9475–9489.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408.

Shadlen, M. N. and Newsome, W. T. (1994). Noise, neural codes and cortical organization. *Current Opinion in Neurobiology*, 4:569–579.

Shadlen, M. N. and Newsome, W. T. (1998). The variable discharge of cortical neurons: implications for connectivity, computation, and information coding. *Journal of Neuroscience*, 18:3870–3896.

Sheffer, H. M. (1913). A set of five independent postulates for Boolean algebras, with application to logical constants. *Transactions of the American mathematical society*, 14:481–488.

Shriki, O., Kohn, A., and Shamir, M. (2012). Fast coding of orientation in primary visual cortex. *PLoS Computational Biology*, 8:e1002536.

Siegelmann, H. T. and Sontag, E. D. (1992). On the computational power of neural nets. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 440–449.

Srivastava, K. H., Holmes, C. M., Vellema, M., Pack, A. R., Elemans, C. P., Nemenman, I., and Sober, S. J. (2017). Motor control by precisely timed spike patterns. *Proceedings of the National Academy of Sciences*, 114:1171–1176.

Tolhurst, D. J., Movshon, J. A., and Dean, A. F. (1983). The statistical reliability of signals in single neurons in cat and monkey visual cortex. *Vision Research*, 23:775–785.

Turing, A. M. (1936). On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, s2–42:230–265.

Van Vreeswijk, C. and Sompolinsky, H. (1996). Chaos in neuronal networks with balanced excitatory and inhibitory activity. *Science*, 274:1724–1726.

Wehr, M. and Zador, A. M. (2003). Balanced inhibition underlies tuning and sharpens spike timing in auditory cortex. *Nature*, 426:442–446.

Zuo, Y., Safaai, H., Notaro, G., Mazzoni, A., Panzeri, S., and Diamond, M. E. (2015). Complementary contributions of spike timing and spike rate to perceptual decisions in rat s1 and s2 cortex. *Current Biology*, 25:357–363.