Lecture 3: Temporal representation in spiking neurons

- exercises are due the week after they are assigned unless otherwise noted.
- hand in your code and all plots.
- project topics should be chosen by the beginning of february, so you can get a good start when we miss two lectures in a row (Feb 13 and 20).
- today's lecture ends 15min early.

3.1 The leaky integrate-and-fire (LIF) neuron

3.1.1 Introduction

The LIF neuron is a good model to use as a standard because:

- it is simple
- it produces spikes (see slide)
- it is known to be a limiting case of more complex models
- the parameters in the model map onto known properties of real neurons

A brief overview of some of those know properties (see slides):

- the bilipid cell membrane acts like a capacitor, separating ions inside and outside the membrane
- there is a passive flow of ions through the cell membrane ion channels that results in a 'leak' current
- neurons elicit a stereotypical voltage depolarization (spike) when the soma voltage passes a threshold
- the spike cannot be repeated even with very high input for a short period (2ms) after a spike (absolute refractory period).

I will cover the biophysics of spike generation in more detail next lecture when we examine models that more closely mimick these behaviours. For more information on the molecular biology of neurons, see: http://soma.npa.uiuc.edu/courses/bio303/ch2.html.

3.1.2 Derivation of the LIF model

The LIF neuron is a passive RC circuit coupled to an active spike, $\delta(t)$, generator. The capacitor is the cell membrane. The current across the membrane can be found by differentiating

$$V = Q/C$$

to give

$$J_C = C\frac{dV}{dt}. \tag{3.1}$$

The ionic current, J_R, accounts for this passive leak of charge across the membrane. Ohm's law tells us that the ionic current is

$$J_R = \frac{V}{R}. \tag{3.2}$$

The final current of importance to the LIF model is the membrane current, J_M, at the soma. In effect, this current represents the input to the model. It is current resulting from all of the postsynaptic currents (PSCs) generated at the dendrites. This current is comprised of two distinct components, the bias or background current, J^{bias}, and the drive current, J^d.

Since the flow of charge must be conserved between the inside and outside of the membrane (i.e., Kirchoff's law applies), we know that

$$J_M = J_C + J_R. \tag{3.3}$$

Substituting equations (3.1) and (3.2) into (3.3) and rearranging gives

$$\begin{aligned} J_M &= C\frac{dV}{dt} + \frac{V}{R} \\ \frac{dV}{dt} &= -\frac{1}{\tau^{RC}}\left(V - J_M R\right), \end{aligned} \tag{3.4}$$

where $\tau^{RC} = RC$.

Recall that this ordinary first-order differential equation only describes the passive behavior of the LIF model. Once V crosses the neuron threshold, V_{th}, the gate denoted by τ^{ref} closes and a delta function, $\delta(t_n)$, spike is generated. By short-circuiting the capacitor and resistor, the gate sets the potential across the membrane to zero (i.e., the assumed resting potential) since there is no way for a difference in charge to build up. This gate stays closed for a length of time, τ^{ref}, equal to the absolute refractory period of the neuron.

Some useful insights can be gained into what the LIF model is doing by considering a means of solving this equation.

We begin by assuming that we want a solution in the form of

$$V(t) = F(t)e^{-t/\tau^{RC}}, \tag{3.5}$$

where $F(t)$ is some arbitrary function of time. We assume this because it consists of a derivative of some function, equal to that function times a constant (whose solution is typically Ae^{at}), plus some other arbitrary function of time.

We can now substitute equation (3.5) into equation (3.4) to obtain

$$\frac{d}{dt}\left(F(t)e^{-t/\tau^{RC}}\right) = -\frac{1}{\tau^{RC}}\left(F(t)e^{-t/\tau^{RC}} - J_M(t)R\right).$$

Applying the product rule gives

$$\frac{dF(t)}{dt}e^{-t/\tau^{RC}} + \frac{de^{-t/\tau^{RC}}}{dt}F(t) = -\frac{1}{\tau^{RC}}F(t)e^{-t/\tau^{RC}} + \frac{1}{\tau^{RC}}J_M(t)R.$$

Knowing that $\frac{de^{at}}{dt} = ae^{at}$

$$\begin{aligned}\frac{dF(t)}{dt}e^{-t/\tau^{RC}} - \frac{e^{-t/\tau^{RC}}}{\tau^{RC}}F(t) &= -\frac{1}{\tau^{RC}}F(t)e^{-t/\tau^{RC}} + \frac{1}{\tau^{RC}}J_M(t)R \\ \frac{dF(t)}{dt}e^{-t/\tau^{RC}} &= \frac{J_M(t)R}{\tau^{RC}} \\ \frac{dF(t)}{dt} &= \frac{J_M(t)R}{\tau^{RC}}e^{t/\tau^{RC}}.\end{aligned}$$

Integrating both sides gives

$$F(t) = \frac{R}{\tau^{RC}}\int_0^t e^{t'/\tau^{RC}}J_M(t')dt',$$

where t' is a 'dummy' variable that ranges over all times between $t = 0$ and now. We can now substitute this expression for $F(t)$ into equation (3.5), which gives

$$V(t) = \frac{R}{\tau^{RC}}\int_0^t e^{-(t-t')/\tau^{RC}}J_M(t')dt'. \tag{3.6}$$

This is a standard convolution integral. In effect, this equation says that the voltage right now (at t) depends on all past current input, $J_M(t')$, where each input is weighted by a function that exponentially decays as that input gets further away (in the past) from the current time.

Evaluating the integral by assuming that the input current, J_M, is constant and substituting variables by letting $u = -(t-t')/\tau^{RC}$, gives

$$V(t) = \frac{R}{\tau^{RC}}\int_0^t e^u J_M \tau^{RC} du.$$

Evaluating the integral gives

$$\begin{aligned}V(t) &= J_M R e^{-(t-t')/\tau^{RC}}\Big|_0^t \\ &= J_M R\left(1 - e^{-t/\tau^{RC}}\right).\end{aligned}$$

Given our assumptions, this solution is physiologically unrealistic but it is a reasonable approximation, especially since we do not know a general form for $J_M(t)$. As well, this mirrors quite well how physiologists find the firing rate curves for real neurons (i.e., by introducing a constant current). Fortunately, it is not difficult to implement (3.6) in computer simulations, so the dynamics are better preserved in practise.

In the last class and exercises, we used the LIF response function; now we can see where it comes from. The time to threshold is equal to:

$$a(t_{th}) = \frac{1}{t_{th} + \tau^{ref}}. \tag{3.7}$$

We can find t_{th}:

$$\begin{aligned} V_{th} &= J_M R \left(1 - e^{-t_{th}/\tau^{RC}}\right) \\ t_{th} &= -\tau^{RC} \ln \left(1 - \frac{V_{th}}{J_M R}\right). \end{aligned} \tag{3.8}$$

Substituting this into (3.7), we find that the firing rate as a function of the input current J_M is

$$a(J_M) = \frac{1}{\tau^{ref} - \tau^{RC} \ln \left(1 - \frac{V_{th}}{J_M R}\right)}. \tag{3.9}$$

$V_{th} = J_{th} R$, so we can cancel the resistance terms. We can now re-write equation (3.9) in the form we first encountered it in section **??**, i.e.,

$$a(x) = \frac{1}{\tau^{ref} - \tau^{RC} \ln \left(1 - \frac{J_{th}}{J_M(x)}\right)}, \tag{3.10}$$

where $J_M(x) = \alpha x + J^{bias}$. The effects of change various parameters are shown in the slides.

3.1.3 Weaknesses of the LIF model

Strengths have already been covered in the introduction. Some weaknesses of this model are:

- they are point neurons (i.e., no spatial extent; no dendrites, no axon, etc)
- complex time courses of different ion conductances are not modeled
- the many other physiologically unrealistic assumptions (R is constant, J_M is static, V_{th} is static, no adaptation, etc.)

One standard complaint with LIF models is that dendrites are linear, but in fact the way $J_M(x)$ is determined is not part of the LIF model itself (though summing dendritic input is common in models that use LIF neurons).

3.2 Temporal codes in neurons

3.2.1 Introduction

Traditionally, there is a debate between two kinds of codes:

1. rate codes: The firing rate of the neuron is what carries information. This was traditionally defined as the rate over a window of about 100ms. Patterns of firing within such windows was irrelevant (i.e., noise).

2. timing codes: The precise pattern of spike generation carries information. So the particular patterns of firing can differentiate stimuli.

So defined, rate codes and timing codes are both contradicted by the evidence, as discussed in the text. Luckily, we don't have to choose either kind of code. The way of examining temporal coding we will explore is general enough to apply to the diverse situations that lead to this artificial distinction to begin with.

3.2.2 Temporal representation

Recall from last class that codes/representations are defined by identiying an encoder, a decoder and their alphabets. The work on the LIF identifies a reasonably realistic neural encoder, generating spikes given some input signal, $x(t)$:

$$\begin{aligned} a(x(t)) &= G[J(x(t))] && (3.11) \\ &= \sum_n \delta(t - t_n), && (3.12) \end{aligned}$$

where

$$J(x(t)) = \alpha\tilde{\phi}x(t) + J^{bias}.$$

In order to complete this characterization, we need to find the decoder. Notably, the encoded alphabets are much easier to identify for peripheral neurons than central ones, but this just means the decoders are defined with respect to a complex set of interactions, not that they differently defined.

As discussed in the text, it is most convenient to think of the simplest population of neurons when discussing temporal decoding, in order to generalize the discussion to large populations later on (see slide).

As in the last lecture, we take the problem to be one of finding the optimal *linear* decoder. So we want to find the linear filter/decoder, $h(t)$, that gives us the best estimate of the input, $x(t)$, given the spike train $\delta(t - t_n)$ from the LIF neuron. That is we want to find $h(t)$ in

$$\hat{x}(t) = \int_0^T h(t - t') \sum_n \delta(t' - t_n) dt'. \tag{3.13}$$

This is a convolution integral, like that seen in the LIF derivation. These integrals are common in linear systems theory, where $h(t)$ is called the impulse response (which completely characterizes any linear system). One way of thinking of our decoding problem is that we're trying to find the impulse response of a linear system that would give the best estimate of $x(t)$ given the spike train. This linearity assumption is only a decoding assumption, not an encoding one.

We can evaluate 3.13 directly because of the delta functions to give

$$\hat{x}(t) = \sum_n h(t - t_n).$$

This says that our estimate of the incoming signal is constructed by summing the filter at each instance a spike occurs (see figure). Our job is to find h(t).

Before we do this, we need to try to limit the space of relevant signals any way we can, to make our decoder a better one. As discussed in the text, natural signals are generally assumed to be stationary (statistics don't vary over time). Because the statistics don't vary, we can write an expression for the ensemble of relevant signals using statistically independent coefficients. This can be done as a Fourier series

$$x(t; \mathbf{A}) = \sum_{n=-(N-1)/2}^{(N-1)/2} A_n e^{i\omega_n t}, \tag{3.14}$$

where $\omega_n = n\Delta_\omega$. These N complex frequency components define real signals and thus $A_n = A^*_{-n}$ (i.e. these coefficients are complex conjugates).

So defining an ensemble of signals can be done by picking a set of amplitude vectors $\mathbf{A}_i$. One way of doing this is by specifying the probability distribution of these vectors as

$$\begin{aligned}\rho(\mathbf{A}) &= \prod_{n=0}^{(N-1)/2} \rho(A_n) \\ &= \prod_{n=0}^{(N-1)/2} \frac{1}{\sqrt{2\pi\mathcal{P}_n}} e^{-A_n^2/2\mathcal{P}_n}\end{aligned}$$

and $A_n = A^*_{-n}$. Here, $\mathcal{P}_n$ is the average power at frequency ω_n over the ensemble.

Now that we have a general means of identifying the relevant set of signals, we can average over them to find the optimal filter, $h(t)$. That is, we want to minimize

$$E = \left\langle [x(t; \mathbf{A}) - h(t) * R(t; \mathbf{A})]^2 \right\rangle_{t,\mathbf{A}}$$

where $*$ is convolution, $R(t; \mathbf{A})$ is the spiking response of the pair of neurons to the input signal, and $x(t; \mathbf{A})$ picks out the particular signal defined by the coefficient vector $\mathbf{A}$ (which, of course, is averaged over). To solve this, it's useful to work in the frequency domain, i.e. to solve:

$$E = \left\langle \frac{1}{2\pi} \left| x(\omega; \mathbf{A}) - h(\omega) R(\omega; \mathbf{A}) \right|^2 \right\rangle_{\mathbf{A},\omega}.$$

To do so, we need to rewrite $x(t;\mathbf{A})$ in that domain as well:

$$\begin{aligned} x(\omega,\mathbf{A}) &= \int x(t,\mathbf{A})e^{-i\omega t}\,dt & (3.15)\\ &= \sum_n A(\omega_n)\int e^{-it(\omega-\omega_n)}\,dt & (3.16)\\ &= 2\pi\sum_n A(\omega_n)\delta(\omega-\omega_n). & (3.17)\end{aligned}$$

So, the error per frequency channel is:

$$E(\omega_n) = \left\langle \left|A(\omega_n) - h(\omega_n)R(\omega_n;\mathbf{A})\right|^2\right\rangle_{\mathbf{A}}. \quad (3.18)$$

There is no analytical solution to this high-dimensional integral, so we can do a Monte Carlo estimate. Letting α denote particular choices of our amplitudes, $\mathbf{A}$, we write the error as

$$\begin{aligned} E &= \frac{1}{N^\alpha}\sum_\alpha \frac{1}{L}\int_0^L \left[x(t;\mathbf{A}^\alpha) - h(t) * R(t;\mathbf{A}^\alpha)\right]^2 dt & (3.19)\\ &= \frac{1}{N^\alpha}\sum_\alpha\sum_n \left|A^\alpha(\omega_n) - h(\omega_n)R(\omega_n;\mathbf{A}^\alpha)\right|^2. & (3.20)\end{aligned}$$

The length of each trial, L, need only be a few times longer than the 'memory' of the neuron, which in the case of the leaky integrate and fire neuron is τ_{RC}. This is because trials separated by L can be assumed to be statistically independent.

Now, to do this in practice, it is much easier to apply the technique to a long recording time that is somehow 'artificially' divided in to segments of length L (i.e., windowed). This is best done with a Gaussian window:

$$W(t-t^\alpha) = \frac{1}{L\sqrt{2\pi}}e^{-(t-t^\alpha)^2/2L^2}.$$

We then let t^α take on all values between 0 and T (the length of the long trial).

As shown in the book (skip in class), we can apply this window to the error expression as:

$$E = \frac{1}{T^2}\int_0^T\int_0^T \left[W(t-t^\alpha)\Delta x(t;\mathbf{A})\right]^2 dt dt^\alpha, \quad (3.21)$$

where

$$\Delta x(t;\mathbf{A}) = x(t;\mathbf{A}) - h(t) * R(t;\mathbf{A}). \quad (3.22)$$

We can now write this in the frequency domain to simplify it:

$$E(t^\alpha) = \frac{1}{T}\int_0^T \left|W(t-t^\alpha)\Delta x(t)\right|^2 dt$$

$$\begin{aligned}
&= \int |W(\omega;t^\alpha) * \Delta x(\omega)|^2\, d\omega \\
&= \iiint W(\omega-\omega')e^{it^\alpha(\omega-\omega')}\Delta x(\omega) \\
&\qquad W^*(\omega-\omega'')e^{-it^\alpha(\omega-\omega'')}\Delta x^*(\omega)\, d\omega' d\omega'' d\omega \\
&= \iiint e^{it^\alpha(\omega'-\omega'')}W(\omega-\omega')W^*(\omega-\omega'')|\Delta x(\omega)|^2\, d\omega' d\omega'' d\omega
\end{aligned}$$

Taking the integral over t^α, and knowing

$$\frac{1}{T}\int e^{it^\alpha(\omega'-\omega'')}\, dt^\alpha = \delta(\omega'-\omega''),$$

we get

$$E = \frac{1}{T}\int E(t^\alpha)\, dt^\alpha \tag{3.23}$$

$$= \iint |W(\omega-\omega')|^2|\Delta x(\omega;\mathbf{A})|^2\, d\omega' d\omega. \tag{3.24}$$

This shows that we can replace averaging over trials in the time domain by a convolution wiht the window in the frequency domain. Minimizing this error term lets us find the desired $h(\omega)$:

$$h(\omega) = \frac{\langle A(\omega)R^*(\omega)\rangle_{\mathbf{A}}}{\left\langle |R(\omega;\mathbf{A})|^2\right\rangle_{\mathbf{A}}}. \tag{3.25}$$

This minimization is somewhat involved, but not too complicated. I won't reproduce it here. It is worth noting, however, that the window convolution here is represented by $\langle\cdot\rangle_{\mathbf{A}}$ uses the same window as above *squared*. Squaring the window just increases width by a factor of $\sqrt{2}$. Since the window width is somewhat arbitrary, this isn't extremely important.

We now have a well defined temporal representation:

Encoding

$$R(t;\mathbf{A}) = \sum_{i,k}^{M} \phi_i\delta(t-t_{ik}(\mathbf{A})), \tag{3.26}$$

(in a sense this is partly decoded because of the 'population decoders', ϕ_i) and decoding

$$\hat{x}(t) = h(t) * R(t;\mathbf{A}), \tag{3.27}$$

where we know how to find $h(t)$.

To get an intuitive sense of what is going with temporal decoding, and to compare this to population decoding, we can write $\hat{x}(t)$ as

$$\hat{x}(t;\mathbf{A}) = \sum_{i}^{M} a_i(x(t;\mathbf{A}))\phi_i(t), \tag{3.28}$$

where M is the number of time-steps we have divided the signal into, the a_i are 1, 0, or -1, depending whether or not an 'on' neuron, no neuron, or an 'off' neuron respectively emitted a spike, and the $\phi_i(t)$ are all time-shifted versions of $h(t)$ (i.e., $h(t - t_i)$).

- This makes it clear that the same kind of decoding is going on in both cases.
- It also makes clear that the estimate is just a sum of some filter placed at each spike occurence.