Week 6 Lecture Notes

Points of clarification and fun facts re: video lectures

L1: Modeling Connections between Neurons

- Slide: What do synapses do?

  - Chemical synapses can be excitatory or inhibitory. Excitatory synapses cause the post-synaptic neuron to move towards the action potential threshold (depolarization) and inhibitory synapses cause the post-synaptic neuron to move away from the threshold (hyperpolarization). Whether a synapses is excitatory or inhibitory depends on the neurotransmitter released by the pre-synaptic neuron as well as the nature of the receptors on the post-synaptic neuron.

- Slide: RC Circuit model of the membrane

  - Remember that the circuit diagram for the cell still obeys Kirchoff's laws. Whatever current is flowing into the cell at one point is flowing out of the cell at another point (or several points) (usually through leak channels). However, in most discussions, people just talk about the dynamics happening at a particular point in the membrane.

- Slide: What is this equation really saying?

  - Time constants crop up all the time in neuroscience. Oftentimes they fall directly out of the mathematics (like in solving the RC circuit equation). They are useful because they tell you the scale at which the interesting dynamics are happening (e.g., one time constant might be .1ms whereas another might be 10s - everything in the latter system would happen much more slowly that everything in the former).

- Slide: Modeling synaptic inputs

- The synapse model is very similar to the Hodgkin-Huxley model. We start with our basic RC circuit equation and add in ionic channels. In H-H, these channels were voltage-gated, which made the set of equations rather complicated. In the synapse, though, the open probability of the channels depends on the presynaptic input (which, in this case, is a given). This makes the equations simpler because g_s does not depend on V.

- Slide: Basic Synapse Model

  - If probabilities confuse you (they confuse me!), it can be easier to just think about numbers of channels. If you have N total channels, and n_s are open, then N - n_s are closed. The rate constants are given as the fraction of open channels closing or closed channels opening per unit time, so the number of open channels you can expect to close per unit time is n_s multiplied by the closing rate constant, and the number of closed channels you can expect to open per unit time is N - n_s multiplied by the opening rate constant. If you divide all of your little n's by N, then you get back the differential equation for $P_s(t)$.

- Slide: Linear filter model of a synapse

  - Remember that a linear filter maps one function of time to another: $y(t) = f[x(t)]$. (Square brackets indicate that the argument of f is an entire function $x(t)$, not just the value of $x(t)$ at a particular time.) The filter is linear if $f[ax(t)] = af[x(t)]$ and if $f[x1(t) + x2(t)] = f[x1(t)] + f[x2(t)]$. In the synapse case, this means that the response caused by two spikes at different times will be the summation of the two individual responses.

- Slide: Example: Network of integrate-and-fire neurons

  - This example shows in a very simple way how periodic firing patterns could be generated.

L2: Introduction to Network Models

- Slide: Modeling Networks: Spiking versus Firing Rate

    - Synchrony refers to when the spike trains of many neurons are
      relatively lined up. At any point in time they are either all firing,
      or all silent (in the case of a very synchronous network).

- Slide: Recall: Linear Filter Model of a Synapse

    - For a review on linear filters, see the video tutorials on the
      supplementary page of the course website.

- Slide: From a Single Synapse to Multiple Synapses

    - In these models the structure of the dendritic arbor is usually
      ignored, and we assume that the currents from all input neurons
      sum together with no individual delays. Taking the dendritic
      structure into account can make computations extremely difficult.

- Slide: Simplifying the Input Current Equation

    - The differential equation for I_s is equivalent to saying that $I_s(t)$ is
      equal to the input $(\text{sum}(w_b u_b))$ convolved with an exponential
      filter $K(t)$

- Slide: What if there are multiple output neurons?

    - In these examples, we assume that a nonlinear function F acting
      on a vector simply acts on each element of the vector.

- Slide: Example of Edge Detection in a 2D image

    - An h x w black and white image is really just a list of hw
      numbers. If we line all of these numbers up in a list, rather than
      in a rectangle, we can treat the image as a vector.

L3: The Fascinating World of Recurrent Networks

- Slide: Eigenvectors to the rescue!

- Remember, an eigenvector is a vector **v** such that when you multiply **v** by a matrix A, **v** is only scaled, and the relative proportions of its components do not change. That is, $A\mathbf{v} = l\mathbf{v}$. See this week's supplementary videos for more details.

- While making the matrix M symmetric makes the mathematics easier, it is not always true that recurrent connections have such a reciprocal nature. However, there are several cases where this is a good approximation.

- Slide: Use Eigenvectors to Solve for Network Output **v**(t)

  - Remember linear combinations? Here they are again! **v**(t) is a weighted sum of basis vectors $\mathbf{e}_i$. Here, however, the weights $c_i(t)$ are time dependent, so **v**(t) is a different weighted sum of the same basis vectors at different times. (The basis vectors, however, are not time-dependent.)

  - Here's the intuition for why this works:

    - Our original differential equation talks about how the time-derivative of the vector **v** (i.e., the vector of the time-derivatives of each of its components) depends on various other quantities, including **v** itself. Thus, if **v** is an N-dimensional vector, this is equivalent to a set of N *coupled* differential equations. Coupled differential equations are a pain the neck, so we take the eigenvector approach. The eigenvector approach finds a new representation of **v** (i.e., instead of listing its components, we represent it as the set of coefficients in a weighted sum of basis functions) such that this set of differential equations is no longer coupled. This allows us to solve each one individually, instead of solving them all at once.

- Slide: Eigenvectors Determine Network Stability!

- Steady state solutions are found by setting all time-derivatives to zero.

- Slide: Example of a Linear Recurrent Network

    - Labeling neurons with angles isn't actually all that crazy — for example, in visual cortex there are neurons that respond to bars of light at certain angles, and in the cricket (remember the cricket??) there are neurons that respond to movement along certain directions/angles. Thus, we commonly label neurons by the angle of the stimulus that best excites the neuron.

- Slide: Amplification in the Linear Recurrent Network

    - Remember that here, the input to **v** is **h**, where the i-th element of **h** is the activity of the input neuron corresponding to the i-th angle. Ideally, **v** will tell us something interesting about **h** that might not be obvious just by looking at **h**. One interesting thing it might tell us is how much of **h** is lined up with one of the eigenvectors of M (i.e., the projection of **h** onto a particular feature tells us how much **h** is lined up with that feature). In this example network, **v** tells us this by simply being that eigenvector (which is a vector) multiplied by a scalar component, this scalar component being an amplified version of the projection of **h** onto the eigenvector.

- Slide: Nonlinear Recurrent Networks:

    - Basically, rectifying a vector/function just sets all negative values to zero.

- Slide: Gain Modulation in the Nonlinear Network

    - Gain modulation can be useful when trying to separate several signals that are all close together.

- Slide:What about Non-Symmetric Recurrent Networks?

- That excitatory neurons create only excitatory connections and inhibitory neurons create only inhibitory connections is a manifestation of Dale's Law: a single neuron releases the same set of neurotransmitters from all of its synapses. Further, the neurotransmitter released by a neuron is usually the key factor in determining whether that neuron's synapses onto other neurons are excitatory or inhibitory. However, the strengths of a neuron's synapses can vary.

- Slide: Linear Stability Analysis

  - Remember that a dynamical system will move toward stable fixed points and away from unstable fixed points. Thus, to start to understand the system's behavior, the first thing to do is to find the fixed points and then ask if they are stable or unstable.

  - Linear stability analysis is very related to approximating the instantaneous slope of a curve by a straight line tangent to the curve. In this case, however, we zoom in on a *fixed point* of our system (see the Week 5 video tutorials about dynamical systems) and represent the whole dynamical system as being linear around that fixed point — just like fitting a straight line to a curve, as long as we don't move too far away from the fixed point, this will be a pretty good approximation. Then, linear dynamical systems (in any number of dimensions) have the property that it is easy to classify their fixed points, so we can determine the stability of the system around that fixed point.

      - If at least one of the eigenvalues of the Jacobian around a fixed point has positive real part, then the system will move away from the fixed point (it is unstable). If all of the eigenvalues have negative real part, then the system will move toward the fixed point (it is stable). If the eigenvalues have imaginary components, then there will be an added oscillation to the system's behavior (this comes from the fact that around a fixed points the dynamics are exponential

in time, and $\exp(it) = \cos(t) + i\sin(t)$, which is how you get oscillations from an exponential equation).