Week 2 Lecture Notes

Points of clarification and fun facts re: video lectures

L1: What is the Neural Code? (19:18)

- Slide: Recording from the brain: fMRI
  - For some fascinating experiments currently being done at UC Berkeley with fMRI recording, see http://gallantlab.org/
- Slide: Raster plots
  - Although continuously varying membrane potentials can encode information, even in spiking cells, action potentials, or spikes, are canonically considered to be the "currency" of the central nervous system. As such, the raster plot portrays only the times of these action potentials. The raster plot is not a suitable visualization device for non-spiking cells (such as rods and cones).
- Slide: Encoding and decoding
  - The notation P(X|Y) denotes the probability of a random variable X *given* that random variable Y has been fixed (everything to the right of the | is assumed given). This is called a conditional probability distribution. If X is independent of Y, then P(X|Y) = P(X)
- Slide: Neural Representation of Information
  - The meaning of *s* depends on the nature of the stimulus.
  - While for simple stimuli such as moving bars of light, a relevant stimulus parameter can be easy to define, complex stimuli, such as faces can be very difficult to "parameterize" in a meaningful way.

L2: Neural Encoding: Simple Models

- Slide: Constructing response models
  - Be careful not to get your probabilities confused. The response of a neuron depends probabilistically on the presented stimulus. This means that you could present the same stimulus twice and get two different responses. Once a response has been "picked out", then the actual spiking activity you measure is also probabilistic. The same response could yield two different patterns of spikes. Thus, there is a double probabilistic nature in this model. If this seems confusing, it is nice to note that the response can also be thought of as the average number of action potentials fired per second, i.e., firing rate.
- Slide: Basic coding model: temporal filtering
  - An intuitive way to think of temporal filtering is to imagine that the system is scanning the stimulus wave-form by sliding a window of a certain width/duration along it. The more the stimulus in the window resembles the filter, the more strongly the system will respond. Thus, the system is "looking for" pieces of the stimulus that resemble the filter.

- Slide: Spatial filtering and receptive fields: difference of Gaussians
    - The Gaussians here are not probability distributions; they just indicate the mathematical form of the filter
- Slide: Next most basic coding model
    - The input/output function is just a way to transform the linearly filtered signal, which could theoretically be very negative or very positive (depending on the stimulus and filter), to a signal whose value lies between 0 and 1.

L3: Neural Encoding: Feature Selection (22:13)

- Slide: Dimensionality reduction
    - The dimensionality of a stimulus is just the number of numbers you need to describe it. If you know nothing at all about the patterns in your stimulus, then you'd need a long list of numbers to describe it, one for each point in time at which the stimulus was sampled. On the other hand, if you knew that your stimulus was always approximately a multiple of some function f(t), then the only number you would need would be the scaling factor, and you'd have a pretty good description of the stimulus. The goal of dimensionality reduction is to try to find a small set of numbers such that knowing the values of those numbers for a given stimulus describes it as best as possible. Sometimes those numbers will correspond to scaling factors (like in the above example), but sometimes they will correspond to things more complex.
    - For the purposes of illustration, we often draw high-dimensional points in a 2D or 3D space
- Slide: determining linear features from white noise
    - The spike-triggered ensemble, or spike-conditioned distribution, is just the set of all the stimuli that trigger a spike. The goal is to find an efficient and faithful representation of the stimuli in this set; for example, you could try to find the average stimulus, as well as a simple way to talk about how all of the stimuli relate to it.
- Slide: Principal component analysis: spike sorting
    - If PCA works well for describing the patterns in your dataset, then the following should be true:
        - Given n principal components f1(t), f2(t), … fn(t), of your dataset, any arbitrary stimulus s(t) in the dataset should be very well represented as a weighted sum of the principal components added to the average stimulus s0(t).
- Slide: Finding interesting features in the retina
    - PCA is a dimensionality reduction technique. Knowing the principal components tells us how to write each stimulus as just a small set of numbers. If there are only two relevant principal components, then each stimulus can be written as two numbers. Each stimulus can therefore be plotted as a point in a 2D space. In

the retina example, doing this clearly shows separation between the two types of stimuli that drove the cell.

L4: Neural encoding: variability

- Slide: Poisson spiking
    - For the mathematically oriented, Poison spiking models represent a spike-train as a stochastic process specified as the sum of Dirac delta functions, each centered around the time of a spike.
- Slide: The General Linearized Model
    - A Poisson process with a time varying rate r(t) is completely characterized by the rate r(t): if you know r(t) then you know all of the statistics of the process. A given spike-train might have a very high probability under a Poisson process with one r(t) and a very low probability under a Poisson process with another r(t). A simple linear-nonlinear model says that r(t) depends only on the stimulus. The GLM generalizes this by saying that r(t) depends on the activity of the neuron as well.
    - Mathematically the goal of these models is to use your data to figure out the r(t) (and thus the associated Poisson process) such that the observed spike-train has the highest probability. A linear-nonlinear model only looks at the stimulus to figure out this r(t), whereas a GLM looks at both the stimulus and the spike train.