Problem Set #3: basic models of encoding

For this problem set, you will need the data-file **`LGN_FFdata.mat`**, which is located in the ELMs site under **"Code and Data"**.

---

1. SIMULATED TUNING CURVE OF A V1 NEURON. Consider a V1 neuron with a tuning curve given by a Gaussian curve, with a standard deviation of 30 degrees (sigma). Lets assume this particular neuron has a spontaneous firing rate of 4 Hz (r_0), and is tuned to 90 degrees (theta$_0$), where it responds with a mean firing rate of 50 Hz ($r_{max}+r_0$). As a result, its mean firing rate in response to an oriented bar with angle (theta) is given by the following equation:

$$r(\theta) = r_0 + r_{max}\, e^{-(\theta-\theta_0)^2/(2\sigma^2)}$$

   Also assume that the neuron is Poisson, which means that it fires in response to a particular orientation with a uniform firing rate given by the tuning curve.

   A. Plot the tuning curve. Note that because it is orientation (not direction), you only need to go up to 180 degrees: 180 degrees is the same as 0 degrees. [Also please make sure to label the x- and y- axes appropriately (including the numbers on each axis).]

   B. On a separate graph, plot the standard deviation of the spike count as a function of angle, assuming that the firing rate is estimated from 200 ms of counting. For this, you will want to take advantage of the fact that the variance of the spike count is equal to its mean.

   C. How well does a high firing rate signal the peak of the tuning curve? To address this, first plot the mean number of spikes expected as a function of angle (i.e., proportional to the tuning curve). Then, plot two more curves: the mean with the standard deviation calculated in (B) added and subtracted from it, representing the range of responses expected for each angle. Then, plot a horizontal line at the mean response for 90 degrees. Within the standard deviation pictured, what range of angles might correspond to a firing rate of 50 Hz?

   D. How distinguishable are stimuli represented on the flank of the tuning curve (at 60 degrees)? Would you say that 60 or 90 degrees is better represented by the response? Why?

2. TEMPORAL RECEPTIVE FIELD OF AN LGN NEURON. Download data from an LGN neuron from the elms website, called **`LGN_FFdata.mat`** and load into memory. This is the response of an LGN neuron to multiple repeats of a type of visual stimulation called full-field flicker. You will need three variables loaded into memory: the stimulus **`FFstim`**, the LGN spike train in response **`FFspks`**, and the time resolution of the stimulus **`DTstim`**.

   A. Calculate the average stimulus value for all spike times. [This should be the first element of the spike-triggered average calculated in part B.] Is it what you expect? Why or why not?

   B. Calculate and plot the full spike-triggered average of the neuron, which consists of the average stimulus at spike times (zero latency, calculated in part A), up through 100 ms before a spike. Note: it should have recognizable structure.

3. CALCULATING THE SPIKING NONLINEARITY. If you cannot get the answer to Problem #2 correctly, download the answer (in **`FFrf.mat`** under **`SharedData`**) from the elms site to continue with this problem. This will give you the variable **`Kdan`**. [You may want to verify that you got a reasonable answer anyway, as this problem depends on having the right answer].

   A. Calculate the stimulus convolved with the receptive field, also called the filtered stimulus, or generating function $g(t)$. To do this, use the function provided: **`g_convolve`**. You can type **`help g_convolve`** to see how to use it (note that frac = 1 since you are using the resolution of the stimulus frame without subdividing. The resulting function should have the same length as the stimulus. Normalize the receptive field (divide it by an appropriate number) so that the standard deviation of the filtered stimulus ends up being 1. Plot its value from 800 ms to 1 sec.

      Explanation: Because the output of the receptive field processing is an internal model parameter, it can be scaled however we want. I choose to scale it as described, so that the following steps are consistent and meaningful.

   B. Calculate the histogram for the value of the filtered stimulus $g(t)$ over the experiment. Use **`histc`** to do so, using the range **`-6:0.2:6`**. Plot this, and verify that it looks "Gaussian".

   C. Calculate the histogram for the value of the filtered stimulus for just the spike times. Plot both on the same plot, in different colors. If it is centered around zero, something is wrong.

   D. Plot the spiking nonlinearity, which is given by the answer to part (C) divided by the answer to part (B), for all values of $g(t)$ where the count is greater than zero. [It is equal to zero otherwise.] This gives the probability of a spike for each value of g, i.e., $\Pr\{\text{spk}|\text{g}\}$.

4. [OPTIONAL] MODEL PREDICTIONS AND CROSS-VALIDATION. If you cannot get the answer to problem 1 or 2 correctly, download the answer (called **`FFrf.mat`** under **`SharedData`**) from the ELMs site to continue: this will give you the variables **`Kdan`** and **`NLdan`**.

   A. The file **`LGN_FFdata.mat`** also has multiple repeated trials of the same neuron to a shorter stimulus sequence, **`FFstimR`**. You have previously analyzed the repeats: they are in the variable **`FFspksR`**. Now, you will predict the neurons response. First, using the observed data **`FFspksR`**, calculate the neurons firing rate as a function of time, using a bin size **`DTstim`**. Plot the firing rate between 800 ms and 1 sec. Make sure that the vertical axis is in units of Hz.

   B. Using the normalized receptive field from Problem #2 (or downloaded as **`Kdan`**), calculate the filtered stimulus in response to the stimulus **`FFstimR`**. Make sure it is properly normalized, with a variance close to (but probably not exactly) one. Plot the filtered stimulus between 800 ms and 1 sec.

   C. Using the measured spiking nonlinearity from Problem #3 (or downloaded as **`NLdan`**), calculate the predicted firing rate of your neural model. Note that the nonlinearity will create a probability per spike, which you still need to convert to a firing rate. To do this, you will need to bin the filtered stimulus such that it maps to the bins of the spiking nonlinearity; these bins were defined in Problem #3B. Plot the predicted and actual firing rate between 800 ms-1 sec.

   D. How good are your cross-validated model predictions? Calculate the R-squared value of the model prediction (fraction explainable variance), given by:
      R2 = `1-var(FRdata-FRmod)/var(FRdata)`.

5. [OPTIONAL] HIGH-TIME RESOLUTION FAILING OF THE LN MODEL. Perform all major steps of the LN model (problems 2-4) using a higher time resolution of analysis: using ⅛ the stimulus resolution, i.e., **`DTstim`**`/8`. To do this, you need to up-sample the stimulus, such that each stimulus frame is constant over 8 bins. Your answers should be comparable with each step of the solution at low-time resolution, although you will see the prediction (for some reason…) got much worse. This observation was the basis for a Nature paper I wrote in 2007.