Learning objectives and outcomes:

This problem set investigates how neurons respond to different stimuli. In problem 1 we will analyze real data recorded from an orientation-selective neuron in the primary visual cortex. We will plot the cell's tuning curves as a function of stimulus orientation, see the dynamics of its response over time, and analyze how reliable its response is to multiple repetitions of the same stimulus. In problem 2, we will build a linear-nonlinear model of the firing rate responses of a synthetic neuron to an auditory stimulus that varies over frequency and time. This model encompasses: a linear filter or kernel known as the spectro-temporal receptive field (STRF), a thresholding nonlinearity, and a spike generator. We will build all the components of this model and generate a synthetic spike train. Then, we will use the generated spike train to recover the neuron's receptive field by applying spike triggered averaging (STA).

These topics and models were covered in lectures 8&9 and recitations 6&7.

The expected learning outcomes for this PSET are:

- Produce raster plots, peri-stimulus time histograms and tuning curves.
- Compute instantaneous firing rates via convolution.
- Write code to simulate a linear nonlinear STRF model and produce visualizations of the different model stages.
- Write code to compute and visualize the STA.
- Be able to present your results as a cohesive and well-structured report that gives sound interpretation of your results and observations.

MATLAB functions you will need:

For accessing a slice of a 3D data matrix and removing any of its singleton dimensions, the function `squeeze` will be useful.

Recall that for changing the orientation of a 2D data matrix you can use the transpose operator (`'`).

For plotting the PSTH you can use a bar graph (`bar`).

Convolution is performed with function `conv`. Check recitation 6 for code examples.

In problem 2 we will use `surf` to generate visualizations of matrices (surface plot). Check the provided code in *generateKernel.m* for syntax examples and plot customizations.

For more information on MATLAB functions and commands, check the provided cheat sheet or the more extensive MATLAB documentation.

Problem 1: Analyzing Spike trains and tuning curves from neurons in the primary visual cortex.

In this exercise, we will analyze spike trains to calculate tuning curves. Tuning curves are useful to understand stimulus preference of a given neuron. The data we will use comes from Mriganka Sur's lab at MIT and consists of a single-neuron recording in the cat primary visual cortex (area V1). The stimulus is a 'full-field' one-dimensional sinusoidal grating that slowly drifts in phase over time. 'Full-field' refers to the fact that the stimulus covers the vast majority of the retinal field of view. You can reproduce one of these gratings by executing the file *stimulus_demo.m* in MATLAB. The neuronal data is saved in the file *dataSur.mat*. Each element of the data matrix is 0 or 1 representing whether or not the neuron spiked, and the data matrix has three dimensions:

a) Dimension 1 represents the 18 different stimulus conditions: 1-16 are different orientations of the grating, from 0 to 337.5 degrees in 22.5 degree increments. (17-18 are a blank stimulus with uniform gray screen of the same average luminance as the grating.)

b) Dimension 2 represents time, sampled at 1 ms. The trial duration is 3500 ms, and the stimulus is turned on at t = 500 ms and off at t = 2500 ms.

c) Dimension 3 represents trials. In this dataset, there are 30 repetitions (trials) of each stimulus condition.

We have made a function *rasterPlot.m* for you that allows you to display a raster plot showing the responses of the neuron, in all trials for a single stimulus condition. This function takes four input arguments. The first argument is a 2D data matrix (each row is a different trial, and each column is a different time bin). The second argument is the width of each time bin in milliseconds. The last two arguments are the stimulus onset and offset times, respectively, in milliseconds. This function will plot the raster and display a blue bar representing when the stimulus occurred.

rasterPlot(squeeze(spikes(6,:,:))',100,500,2500)

1. **Make a raster plot for stimulus #6 (112.5 degrees orientation), and describe what you observe. Does this cell respond to this stimulus orientation?**

initial spiking when experiment started. then major spiking in the interval when stimulus is shown and spiking stops after stimulus is stopped. Yes, cell responds to this stimulus orientation

2. **Plot the peri-stimulus time histogram (PSTH) for stimulus #6. Use 10ms bins, and make your y-axis have units of Hz.**

3. **Compute the instantaneous firing rate for trial 10 (still stimulus #6) by convolving the spike train with a boxcar[1] kernel of width 10ms. Plot this rate in units of Hz as a function of time. Briefly comment on how the instantaneous firing rate of a single trial resembles the PSTH you previously calculated. Discuss possible sources for agreement or disagreement.**

There are significant differences due to inter trial variability in recording and stimulation of animal. I also suppose my conversion to hertz is not accurate.

4. **The tuning curve is a compact way to visualize stimulus preference of a cell. To calculate the tuning curve, we proceed as follows: for each stimulus condition, we compute the average firing rate in the time window from stimulus onset to stimulus offset. Plot the tuning curve (firing rate as a function of stimulus orientation). Make sure that your y-axis is in units of Hz.**

[1] https://en.wikipedia.org/wiki/Boxcar_function. Make sure that the area under the kernel is 1.

5. **To which grating orientation is this cell most responsive? What is the significance of the second peak in the tuning curve?**

Most responsive: 112.5 degree
The second peak is for 292.5 degree which is diametrically opposite 112.5 degree so it is essentially the same angle hence a second peak

6. **Compute the average firing rate of the cell for stimulus 17 and 18 (blank stimulus).**

7. **Replot the tuning curve by subtracting the average firing rate for blank stimuli from the curve you previously calculated. Why does the firing rate go negative in this case? What is the interpretation of a negative firing rate?**

Negative as the average no-stimuli firing rate is more than the firing rate at some orientations. So the spikes detected during stimuli are not due to stimuli but mostly due to the no-stimuli spiking reasons. Negative firing rate means that firing rate of the cell will be 0 i.e no spiking

Problem 2: Estimation of spectro-temporal receptive fields (STRF) with spike-triggered averaging (STA)

In this exercise, we will demonstrate how to estimate the spectro-temporal receptive field (STRF) using spike-triggered averaging (STA) for a synthetic example. This will provide us an opportunity to review the linear nonlinear model. In this model, the linear response is given by a filter/kernel (the receptive field) acting on the stimulus. This provides a linear estimate of the firing rate of the neuron. This firing rate is then passed through a thresholding nonlinear function to prevent negative firing rates. Finally, this rate is used to generate spikes. This exercise focuses on building all the components of this model and generate a synthetic spike train. Then, we will use the generated spike train to recover the neuron's receptive field by applying spike triggered averaging.

The steps in this process are:

I.   Generate a synthetic stimulus (file *generateStimulus.m*)
II. Generate a kernel to mimic an STRF (script *generateKernel.m*).
III. Compute the synthetic cell response by filtering the stimulus with the kernel.
IV. Feed the cell response to a thresholding non-linearity and generate spikes. Note that whenever the stimulus is sufficiently correlated with the kernel, the neuron will fire.
V.   Use this spike train to compute the STA (the average stimulus that precedes a spike).
VI. Compare the kernel recovered by STA with the original one.
VII. Comment on the limitations that prevent us from perfectly estimating the STRF.

We begin by constructing a stimulus to present to the neuron. We know the neuron is selective to time-varying sound, but we want to know which sound best excites the neuron. Therefore, we want a stimulus that will try many different sequences of sounds with equal probability. The MATLAB file *generateStimulus.m* generates such a stimulus in the form of a 2D matrix where the rows represent 50 logarithmically spaced tone frequencies and the columns represent time bins of width 1ms. Thus, the entry (i, j) of the matrix is the amplitude of tone i at time step j. The stimulus is constructed so that tones turn on and last for 30ms, with a slight ramping at the onset and offset to make transition sound transitions less abrupt. Each tone has a 10% probability of turning on in a given 30ms window.

To see how this works, run the script *generateStimulus.m* to construct the stimulus matrix. The code will also plot the stimulus matrix. Visualizing an auditory stimulus is hard to interpret. The function *PlayStim.m* allows you to listen the stimulus matrix. Just listen to a few seconds of the stimulus, because the whole stimulus is 1000s long, and may crash your computer if you try to play it. Make sure *PlayStim.m* is in your MATLAB path, and your speakers or headphones are working and execute the following:
To listen to the stimulus run the following lines of code:

```
freq = logspace(2,4,50);% 50 log-spaced freqs 100-10000Hz
dt=0.001;% sampling interval in seconds
PlayStim(Stimulus(:,1:(3/dt)),freq, dt);% play first 3s
```

After listening to the the stimulus, you may want to study the guts of *PlayStim.m* to see how it works – It's cool.

The next step is to construct an STRF. To carry this operation, we have provided you with the script *generateKernel.m*. This script returns an array of 100 time bins by 50 frequency bins, containing the STRF. The STRF is constructed by summing up different bivariate Gaussian distributions.
Run *generateKernel.m* to construct the STRF. The code will plot the stimulus matrix. Notice that the kernel matrix has already been flipped in time. Again, listening is more informative than visual display of auditory stimuli. You can listen to the STRF by using *PlayStim.m*. This is the stimulus that best excites our model neuron.

Now that we have a stimulus matrix and a kernel, we are ready to simulate how the neuron responds to the stimulus. To do this, you will slide the kernel across the stimulus, and at each time step calculate the integral (sum) of the element-wise product between the kernel and the stimulus. This is our estimate of how strongly the neuron is excited by the stimulus at each time. Our simulated neuron should spike whenever this excitatory drive exceeds some threshold.

With this information, complete the following tasks and answer these questions:

1. **Look at the plot produced by *generateKernel.m*. Is this receptive field separable? Briefly explain why or why not.**
2. **Write code to calculate the excitatory drive to the neuron at each time in the stimulus. You can start 100ms into the stimulus (the STRF is 100ms long). Plot a histogram of the excitatory drive.**
3. **Write code to record a spike whenever the excitatory drive to the neuron exceeds a threshold. Choose a threshold such that the neuron spikes approximately 1,500 times over the entire stimulus preentation. To determine the threshold value, it will be useful to look at the previous histogram.**
4. **Make a figure with the following three subplots: 100 seconds of the stimulus (use 'imagesc'); the excitatory drive to the neuron during this part of the**

     stimulus (include a constant dashed line for the threshold); and the spiking response of the neuron to this part of the stimulus.

5. **Now you are ready to compute the STA from the spike train. Write code to calculate the average 150ms of stimulus that precedes each spike. Plot the recovered STA using 'surf' (see *generateKernel.m* for an example of how to plot a matrix using 'surf'). Listen to the STA and the stimulus with *Playstim.m* and briefly comment on how they sound.**

6. **The kernel is 100ms long, and we computed a STA that is 150ms long. What do you notice in the part of the STA corresponding to 150-100ms prior to a spike?**

7. **How well did the STA recover the neuron's receptive field? What parts of the kernel did the STA fail to recover? Why do you think this is? Think of the number of spikes, the type of stimulus needed to perform this analysis, and the baseline firing rate of the neuron.**

9.40 Introduction to Neural Computation
Spring 2018