Bioeng 1586
Homework: Neural and Behavioral Data Analysis

Submit a writeup - a document containing the long-form answers to the questions, and plots generated by your matlab code. Email me your code.

In this assignment you will analyze behavioral and neural data collected during a standard movement neuroscience experiment. The monkey performed a "delayed center-out reaching task". Each trial began with its hand and eyes initially directed to a central spot. The monkey was presented with a target at a peripheral location, but he couldn't touch it yet. After a delay period, a 'go' signal was provided, and the animal initiated the reach. He was free to also look at the target at that time. Eight targets were presented during this experiment, arranged at 45 degree intervals around a circle surrounding the central fixation point. Neural data was recorded while the animal performed the task. The first portion of the assignment analyzes the behavioral data; the neural data are analyzed in the second portion. Although recordings were made on 96 electrodes, only four neurons are included, just to keep things manageable.

You can get the data file at:
https://www.dropbox.com/s/oj6h2hj4zkpsvqn/neuralData.mat?dl=1

Behavioral data analysis:
1) Load the file, reachData.mat. It will instantiate a structure called R.
Each element in R is a single trial of behavioral data for the task. Within each element are fields and substructures. Substructures are indicated with a capitalized first letter. You can access the fields and substructures with the '.' operator. Access them for a particular trial, say trial 3, with R(3).timeGoCue. Access them for all trials by leaving out the index, as in R.timeGoCue. To collect all those values, use the vectorization operator, '[ ]'.

2) Find the unique target locations. There were 8 of them. The fields "target1X" and "target2X" within the substructure TrialParams specify the target location.
Note: be aware that Matlab doesn't do multi-level structure accesses. That is, you can't just collect all of [R.TrialParams.target1X]. Instead, for example, do this:

```
TP = [R.TrialParams];
allTarget1X = [TP.target1X];
```

The initial hand position is at (0,0), and the initial eye position is at (20,20). Keep this in mind for the plots later. The units are millimeters.

3) To begin, you will generate a plot of the animal's hand position for each trial. Trim each hand position trace down to the perimovement period. This keeps the plot from having hundreds of irrelevant overlapping points at zero. The fields timeMoveOnset and timeMoveEnd are the times when the reach began on each individual trial. Once you collect these times, it may be easiest to build another structure that includes just the data you want.

Now, graph the hand position data. The axes are horizontal and vertical position. Plot a small circle at the location of each target. For added flair, color the trajectories according to which target the animal is reaching toward.

5) You can quantify behavior. The reaction time is the time from the presentation of the "go" cue till the time that the movement actually begins. What is the mean and standard deviation of the reach reaction time? (Units are milliseconds.)

6) Does the reaction time depend on the direction of the reach? Is the effect statistically significant?

7) The computer monitor used to present the visual display has a certain refresh rate. Thus, when the computer tells the monitor to project an image, the image may not actually appear for a brief time, since it must hold the new image in its buffer till it completes the current painting cycle. We record the time that the monitor projects the image with a photodetector. That time is stored as timeGoCuePHOTO. What are the mean and standard deviation of the monitor's latency? What would you estimate is the refresh rate of the monitor (in Hz)?

8) There's an art to deciding how much information to include in a plot. Sometimes, seeing how you can add in new information without crowding out the main point can be a rewarding exercise. For example, here you could plot the eye positions on the hand position plot. To do this, take the average eye position during the movement, and plot it. That is, each trial contributes a single point to the plot. Color the points according to the reach direction. Does the animal tend to look toward the reach target, away from it, or something different?

Okay now for the neural data analysis.
Neural data for four cells is included in the R-structure. Here you'll examine the temporal and spatial tuning of those cells. ("Tuning" is the response of a neuron to externally-observable factors such as target location or hand velocity. Hence when anybody looks at neural activity, they almost always do so in the context of behavior or stimulus presentation.)

We will analyze the data at three levels of resolution:
— First, we will observe an individual trial. In this part you will look at the behavioral data and the spike rasters (that is, action potential timings) for the four cells.
— Second, we will analyze the average behavior of each neuron over all trials of a particular type. The main tool for this is the "peristimulus time histogram" (PSTH). (Building a PSTH is a nice application of convolution).
— Third, we will summarize the response of the neurons for reaches in all eight directions as a "tuning curve".

1) For one trial (say, #1000), plot rasters for the four cells alongside the behavior. Plot each of these features of the data:
- Horizontal and vertical components of hand position
- Horizontal and vertical components of eye position
- Which trace appears noisier? Why do you think that is? Consider smoothing the plots so they look a little more comparable.

- Plot vertical markers to show important trial events:
timeCueOnset, which is the time when the instruction cue appears.
timeGoCuePHOTO, which is when the animal is instructed to reach.
- "Rasters" for each cell. That is, a tick mark at the time each cell fired an action potential. You can get these for the first cell as R(1000).cells(1).spikeTimes.
Using this code, you can look at a variety of trials (other than #1000). It is always a good idea to get a "feel" for the raw data like this. Only submit a figure for one trial, but take a look at some others. It's fun and informative to get to know data in that manner. If you notice anything interesting by doing this, I encourage you to share your observations in your writeup.

2) For all reaches to the left, plot a **PSTH** for the four cells.
- Use the time period from 300 ms before the cue appears (timeCueOnset) till 600 ms after the cue appears. These epochs are the “pre-cue” period and the “delay” period.
- We will analyze the data for the target in the upper right corner, at (86, 50).
- As you saw in question 1, the spike times are stored as a list. This is perfect for building rasters, but for building a PSTH, you’d rather have a binary vector, where each element is a 1 if the cell fired an action potential during that millisecond. (This is a train of digital delta functions.) So, build a matrix where the rows are trials and the columns are time (in milliseconds). You will want one such matrix for each cell, and it may be easier to build a three-dimensional matrix where the first dimension is cell number.
- Get the spike times that occur during the 900 ms analysis window. Align the data so that the cue appears at the time of the 300th column in the matrix. For each time where a spike occurs, put a 1 into the matrix. Note that you will have to round the data, since the spike times are recorded at sub-millisecond precision, but here you are binning them at millisecond precision.
- Now, convolve the spike vectors with a Gaussian kernel.
- Label your axes, and make sure your y axis units are correct. Remember that the data are binned at 1 ms intervals.

What is a rough estimate for the “latencies” of these four neuron? (That is, how long after the light flashes does the cell take to respond?) Why isn’t the latency zero? Describe the aspects of neural signal processing that make the latency non-zero.

(As you write the code, keep in mind a function, squeeze(). It eliminates singleton dimensions. That is, if you have a three dimensional matrix of size (37, 1, 219) and you call squeeze on it, you will get back a two dimensional matrix of size (37, 219). This can help you get around some indexing complications.)

3) For each of the four cells, plot a **tuning curve**. This shows the firing rate as a function of target location. For this plot, you can “unwrap” the target array, so it’s a Cartesian plot.These are often easier to look at than are polar plots.

To compute one point on the tuning curve, add up all the spike times during the epoch from 100 ms after the cue till 600 ms after the cue, then take the average.

Then plot the average firing rate as a function of target location. Make sure your y axis is correctly labelled. “Spikes per second” is the conventional label. Fit the data with a cosine. Using the model fit, find the preferred directions (target locations that maximally excite the cell) for the four cells. What are they?

Submit the figures you generated, and answers to the questions as part of a short report on your observations. Also zip up your code and submit that along with your writeup.