Final Data Set: Visual and Neural Processing during Search

In this expanded data set, we include additional files that add to the data you worked with in Problem Set 3. In particular, we include the images used for the experiments in addition to initial testing of the neurons using these images shown in isolation (in the center of the display). Recall that in the data you saw in Problem Set 3, each trial came from an experiment in which a particular target image was located in an array of distractor objects and the animal's task was to find the target and press a button that had been associated with that target. In the complete experiment, before running those experimental blocks, a collection of isolated images were flashed on the screen and the neuron's response to those images was collected. This made it possible to find effective and ineffective targets, and to characterize the response of the neuron to both familiar and unfamiliar distractors. The data from this set, therefore, include three sets of trials from three phases of the experiment:

survey_data

These files (`survey_data_set1.npy`, `survey_data_set2.npy`, and `survey_data_set3.npy`) contain rows of trials in which a single stimulus was presented on the screen while spikes were being recorded. To get started, you will want to open the survey_data file and have a look inside:

```
data = np.load('survey_data_set1.npy')[()]


In [9]: data.keys()
Out[9]:

['stim_durations',
 'file',
 'stim_names',
 'description',
 'stimon',
 'em_horiz',
 'fixon',
 'fixoff',
 'month',
 'em_time',
 'spk_channels',
 'stim_scales',
 'spk_times',
 'year',
 'stimoff',
 'stim_extensions',
 'subj',
 'day',
 'em_vert']
```

Of particular interest are the ‘stim_names’, ‘stimon’, and ‘spk_times’ columns. You can create a pandas DataFrame for these using the following shortcut:

```
df = pd.DataFrame({'stimon': data['stimon'],
                   'target': data['stim_names']
                   'spikes': data['spk_times'})
```

One thing you will notice if you examine the DataFrame using, e.g., the `df.head()` function, is that the spikes appear to be the same for multiple trials in a row. This is because multiple single stimuli were flashed in succession between the beginning of a trial and the delivery of juice reward. You will see that stimon increases and then resets with new spike times. This should have little effect on your analysis of these data, but you should be aware that this is how the experiment was actually run.

Of interest is that the target column will contain a name, and you should find a corresponding image (in .png format) for this target in the subfolder called images/. Using the image loading code introduced in the final video from Unit 3, you can see how to load and show these images as part of your analysis. The same images can be used for exploring the other data files as well (`distractor_data` and `search_data`).

The original files have a `'stim_scales'` column, and this will tell you how big, in degrees visual angle, the actual images was when presented in the experiment.

Using the rate analysis code from Problem Set 3, you should be able to compute rates and average neural responses sorted by target stimulus.

Questions for exploration

From the survey data files (set1, set2, and set3 are different recording sessions) you should be able to find out which stimuli were most effective in causing a neural response. You can explore the consistency of these responses and perhaps try to see if there are aspects of the actual images that led to responses that might give you a hint about what neurons in this area of the brain (anterior temporal cortex) might be coding for. You can also look at the timing of the response for different stimuli.

distractor_data

These files (`distractor_data_set1.npy, distractor_data_set2.npy,` and `distractor_data_set3.npy`) contain rows of trials in which a single stimulus was presented, just as above. What is different here, though, is that a subset of the stimuli were actually never seen before and most of these stimuli were used as distractors in the search task that followed. Similar to the survey_data, you will want to open the distrctor_data file and have a look inside:

```
data = np.load('distractor_data_set1.npy')[()]
```

Have a look at the keys in this dataset and you'll see that there is a column called `'description'`. This column will allow you to sort responses by whether a stimulus was a `fam_dist` (familiar distractor), `novel` (a new distractor, not seen by the subject before that session) or `sndmandg` (a code indicating that the image was a known target for the search task).

Set1, set2, and set3 correspond to the similarly named survey files above (and search files below) so there should be some consistency between the neural responses observed in these datasets.

Questions for exploration

Again you can look for patterns in the responses that might be related to either the image content or the familiarity of the images used. You can also keep these data available for helping to analyzing the search data, below.

search_data

These files (`search_data_set1.npy, search_data_set2.npy,` and `search_data_set3.npy`) contain the same search data introduced in Problem Set 3 (these are search_data_set1.npy) in addition to two new files. Of particular interest here is not only the target and target eccentricity, but also information about the distractors. Important columns to explore are:

`array1_pos`:	the position (x,y location in degrees) of all the distractors for a single trial
`array1_img`:	the names of the images at each of those positions
`description`:	df(familiar) or df(novel)

Questions for exploration

Using the target names and array1_pos/array1_img information, you should be able to actually visualize the display shown on a given trial and to compare behavior and neural responses on trials containing novel or familiar distractors. Consider adding eye movements or spike representations for each trial.

Note that the eye position can have a slight offset, which can indicate there is some drift throughout the session. One way to address this is to consider zeroing the eyes to the average position during the end of the fixation on period (around the time of fixoff, another column in the data).