

Exploring Neural Data Final Project

Description of the data

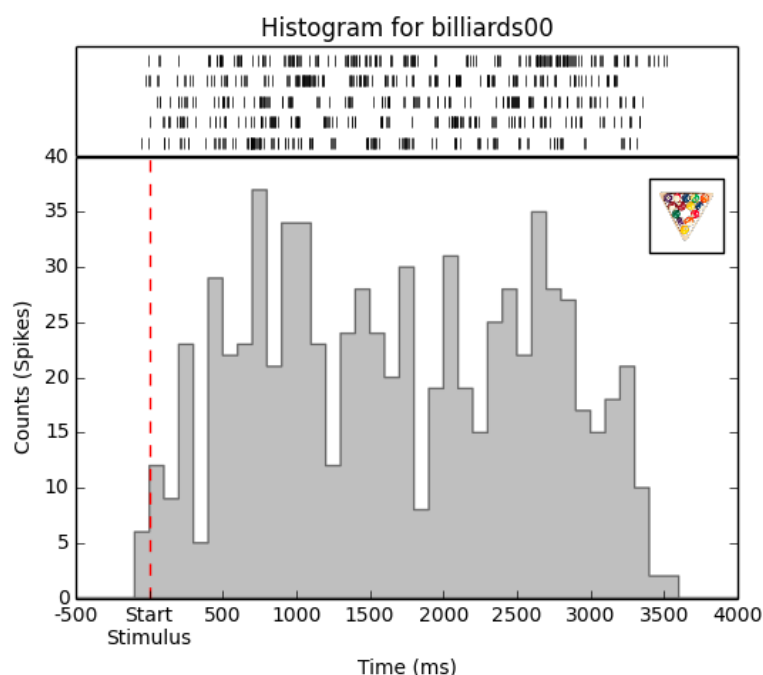
I used the visual processing data set for the final project. The data contained spike times of neural single units recorded in cortices of apes that were exposed to images which were flashed on a screen. The dataset was subdivided into spike times of images that caused a low response of neurons [ineffective targets](#) and spike times of images that caused a high response of neurons [effective targets](#). I filtered the data by sorting the spiketimes of the single units according to their respective picture.

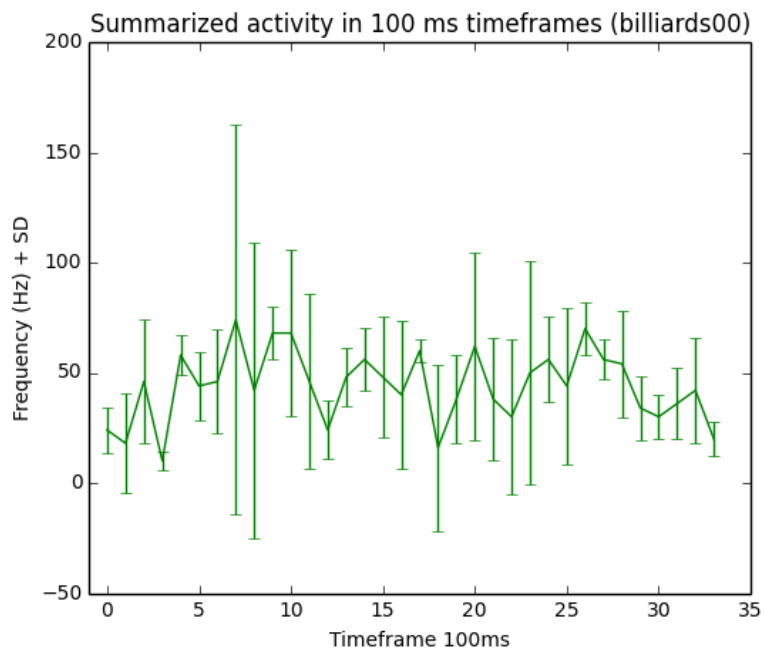
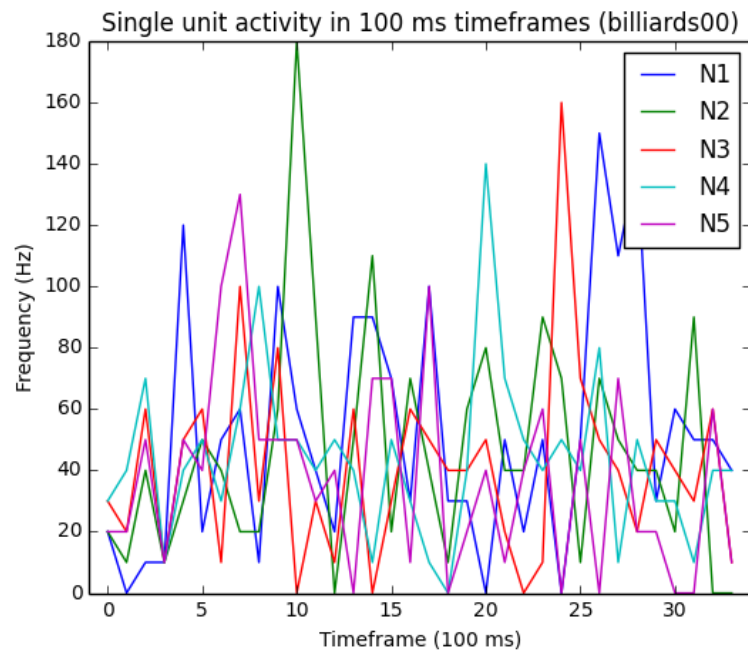
Question addressed in the project

I calculated the average frequency of neurons within discrete time intervals of 100 ms in order to find higher order patterns of neural response. I hypothesized that the collective response of the single units may establish a distinct frequency pattern that is specific for ineffective or effective targets, respectively. Therefore, I wrote an extensive python script that visualizes the neural response of the single units in form of histograms and scatter plots.

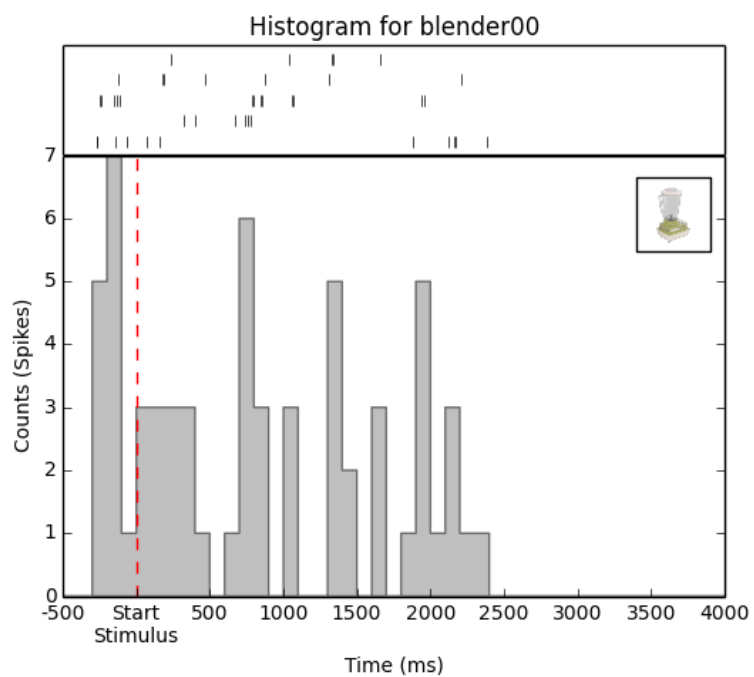
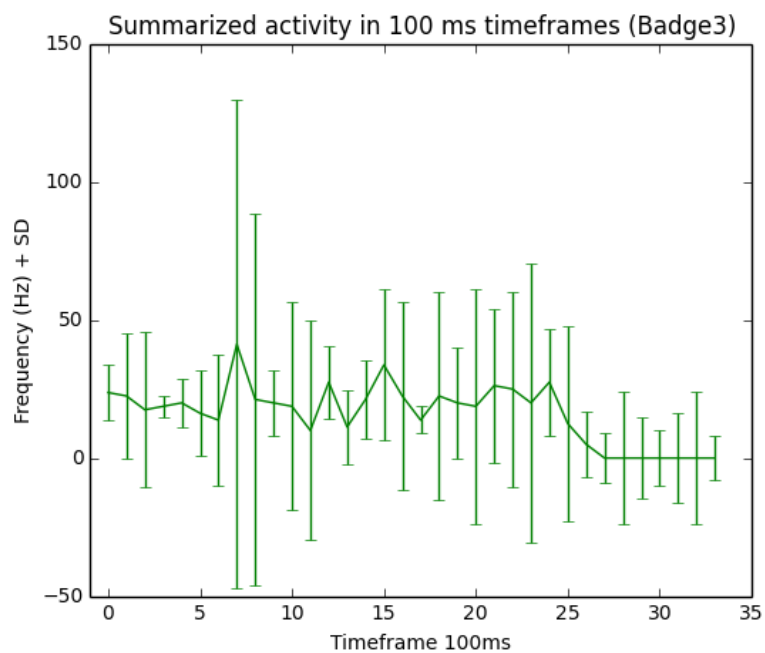
Figures

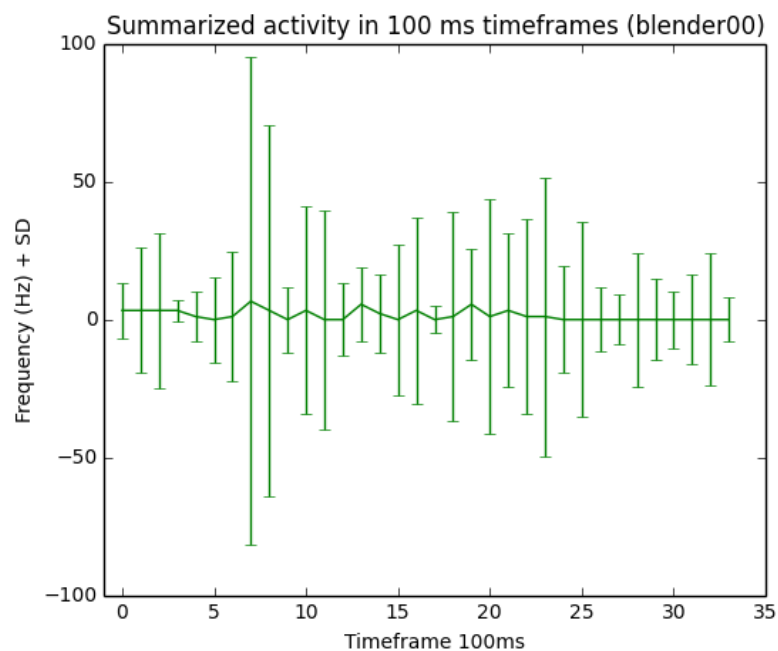
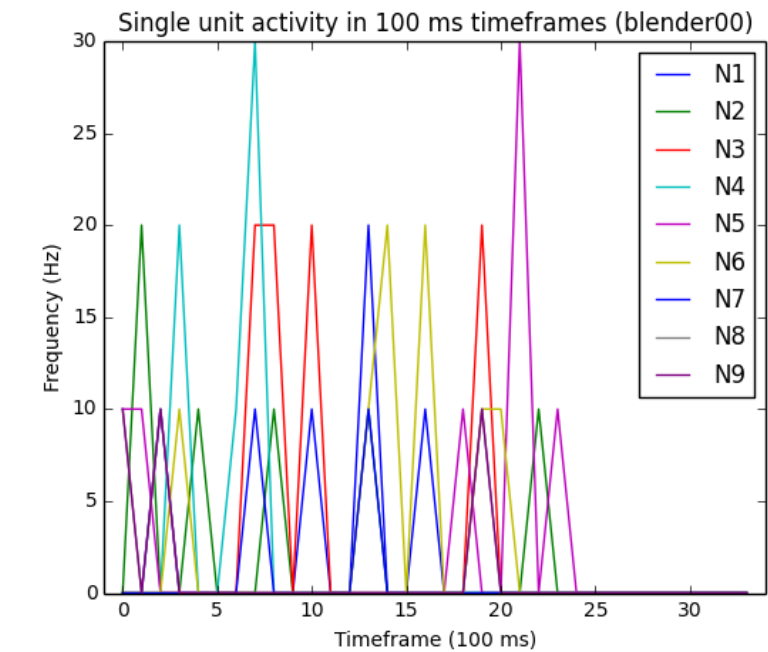
effective targets





ineffective targets





More figures can be found on <https://github.com/sagar87/Exploring-Neural-Data-Final-Project>.

Findings

The frequency pattern of the effective and ineffective targets differed. My data suggested that effective targets are able to cause firing rates up to 200 Hz in multiple neurons over a time span up to 3 s. In contrast, ineffective patterns result in lower firing rates in single neurons and the timespan in which activity could be measured is lower.

The summarized activity of all neurons recorded in the experiment is in ineffective targets more or less constant where as effective targets show oscillating behaviour.

Programming tricks

I used many matrices to get the data processed. The functions which extract the data use extensively use `np.size()` and `np.zeros()` to generate dummy matrices which can then be filled with filtered data.

Problems

The code up so far is inefficient and to some extent buggy. Due to time limitation, unfortunately, I was not able to properly comment the code which makes the code hard to read.

Another problem concerning the analysis is that the time intervalls were selected randomly. The code so far is built to manage other timeframes, however, I was not able to debug the code to assure that functionality.

Follow-up analysis

The code contains parts of a potential follow-up analysis. I want to cluster the images by similarity to find assess the question whether effective or ineffective targets are similar by means of simple euclidian distance. Furthermore, I want to check if a higher time resolution (time frames $< 100\text{ms}$) lead to are more characteristic pattern.

alt text (dis2sublender00.png [600x500])