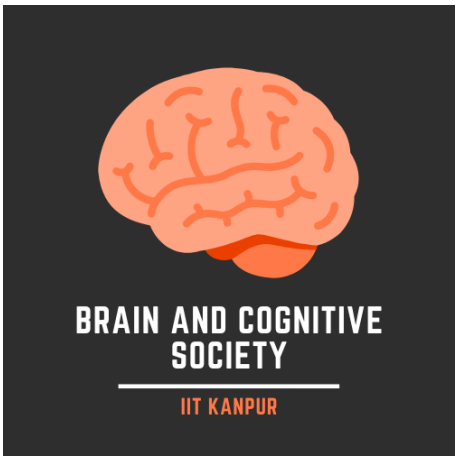


Brain and Cognitive Society

Summer project 2021

Analyzing Steinmetz Dataset

Analysing Steinmetz DataSet

Members:
Bidhan Arya
Praveen Kumar
Ritam Pal
Sagar Agrawal
Sarthak Gupta

Mentor:
Aditya Prakash

July 27, 2021

Introduction

The project aims at understanding and analysing Steinmetz Dataset and learning required tools/libraries. The Dataset provided us with the opportunity to understand/predict the behaviour of the mouse using its neural activity. So, we, finally, formulated our aim to **build a model that can predict the response of mouse based on the neural activity.** The model has a real life implication to predict the behaviour/movement of arms/limbs based on brain activity and thus helping paralysed people to control their limbs using the model. We also analyzed all the neurons and their responses and looked further into the trends of different neurons.

Preparation

Understanding the Brain

Firstly the brain was studied along with functions of neurons. Neurons are nerve cells that control most of the functionalities of the body. It has three major parts. The receiver dendrites, the processor nucleus and the transmitter axon. The neuron receives many electrical signals through the dendrites and then decides whether or not to pass the given signal to the next neuron. Several neurons work together to control a specific activity in brain. Hence it is logical to bundle neurons working on same functionality and study them together. The ones recorded in Steinmetz Dataset are: **Visual Cortex, Hypothalamus, Thalamus, Basal Ganglia, Mid brain, Non-Visual Cortex, Cortical Subplate and some other neurons**

Tools

The language being used in the project is Python. Python was chosen due to its flexibility and abundance of libraries which helped in visualising and analyzing data. The libraries used are as follows:
Numpy: Since python doesn't have a default array data structure, storing in the provided python list, which implements a linked list becomes very slow, and considering the huge volume of the data being studied in these experiments, numpy is used which is based on C language and hence gives a boost in speed.
Matplotlib: In this project we use matplotlib.pyplot which is usually imported as plt. Matplotlib is a data visualization toolkit which helps better understand what the numbers represent.
Seaborn: This library is built on top of matplotlib and is used to enhance and simplify matplotlib commands, giving easier access to plots.
Sklearn: Sklearn is a library built for machine learning. It contains various commonly used algorithms which quickly helped us see the relation between some variables. The picture below shows the prediction of a model as the life, where the true values are represented by the dots. It is evident that the model predicts well.

Steinmetz Dataset:

The Steinmetz Data set is the data of an experiment conducted on mice. In the experiment the mouse was surrounded by three computer screens, and had its fore paws on a wheel. A light was flashed at fixed intervals and the mouse had to turn the wheel according to whether the left or right screen had more contrast. Using Neuropixels, activities in various neurons were recorded pertaining to different locations in the brain. There are many columns in the given data set, some of which are as follows:

- **brain_area**: This contains a list of areas of brain where each neuron belong to. This helps in identifying a cluster of neuron working together.
- **spks**: This contains a 3-D tabulated data for different sessions which represent for each neuron the response in a given time bin. If there is a spike the data recorded is 1, else 0.
- **contrast_left** and **contrast_right**: These columns consist of scaled fraction of light flashed from respective screens.
- **response**: This column consist of data pertaining to the direction that the mouse actually rotated the wheel in.
- **gocue**: This column contains the data of time when the sound was played which marked the beginning of time after which the mouse was allowed to rotate the wheel
- **response times**: This column contains the response was registered, which has to be after the go cue. The mouse can turn the wheel before the go cue (and nearly always does!), but the stimulus on the screen won't move before the go cue.
- **feedback time**: This column contains the data of time when feedback was provided.
- **feedback type**: This column contains the data if the feedback was positive (+1, reward) or negative (-1, white noise burst).
- **wheel**: This column contains the turning speed of the wheel that the mice uses to make a response, sampled at 10ms.
- **pupil**: This column contains the pupil area (noisy, because pupil is very small) + pupil horizontal and vertical position.
- **face**: This column contains the average face motion energy from a video camera.
- **licks**: This column contains the lick detections, 0 or 1.

Early Observations:

The given plot is for mean firing rate of neurons of various brain regions for threes responses. The data is used from **session 11**.
As seen in the plot, the Thalamus seems to be highly active in case of left and right, but in case of no response, there is no such activity during entire duration. This gave us evidence that we can use neural encoding to classify the responses. And foundation of our hypothesis: "to build the model".

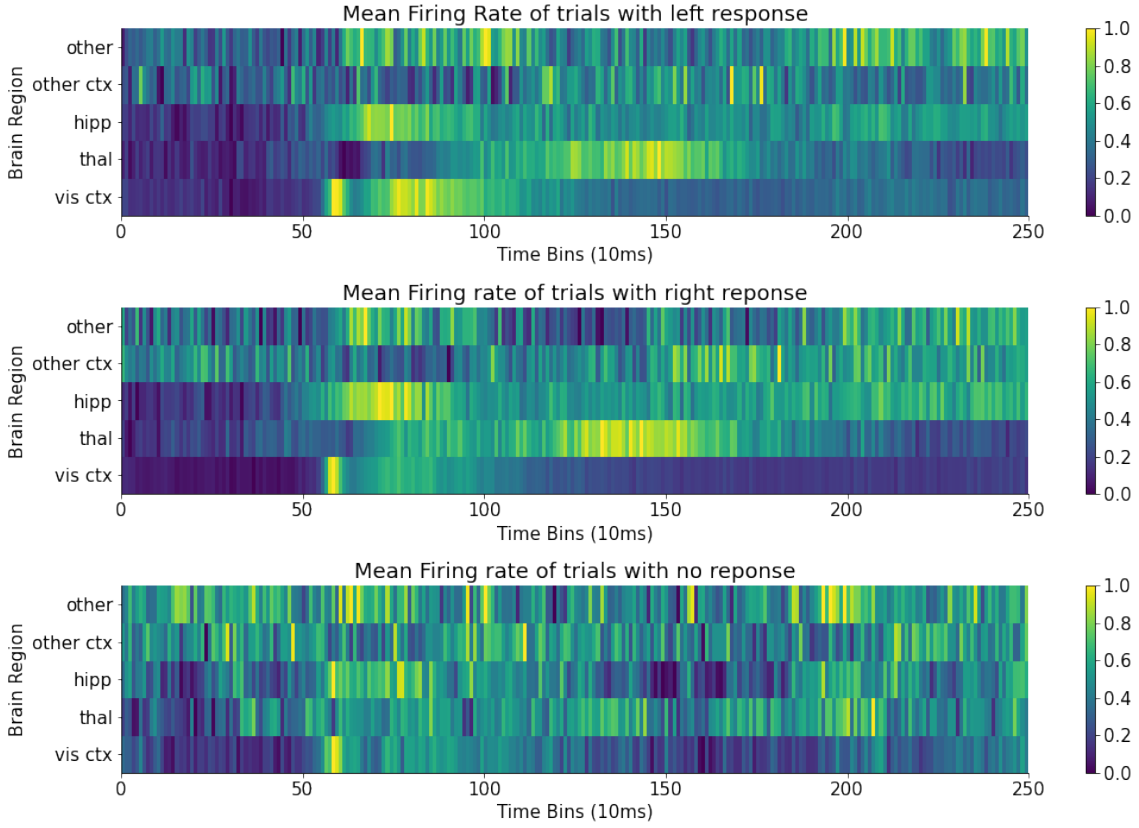


Figure 1: Mean Firing Rate for spikes of Session 11.

Literature Review

Distributed coding of choice, action and engagement across the mouse brain by Nick Steinmetz: The author has fitted the spike data on kernels to determine which neurons are responsible for Visual, Choice, and Action Encoding.

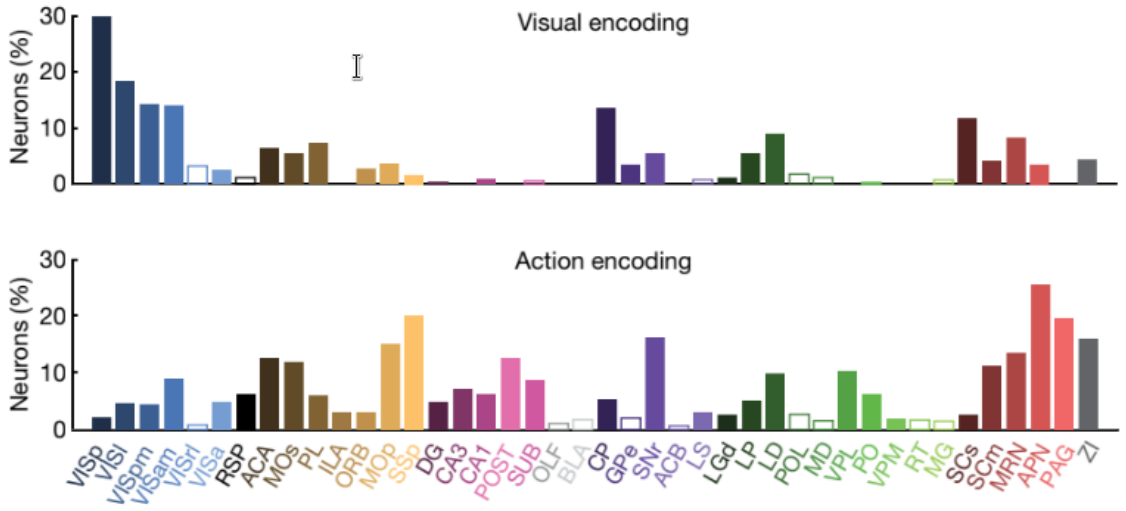


Figure 2: Visual and Action Encoding.

Logistic Regression

Idea

We wanted to predict the response of the mouse based on other features. Since this was a multiclass classification, we first needed to see how balanced the dataset was to be able to use appropriate model. To see what the correct response should have been, we created a list called response which would take the value -1 if the contrast_right was more than contrast_left, 1 if the left contrast was more or 0 if those values were equal. his was done on the base that the mouse was supposed to turn the wheel in the direction of higher contrast flash.
We then added the actual responses that the mice gave by just appending the entries from dat['response'] in a list. These were the results:

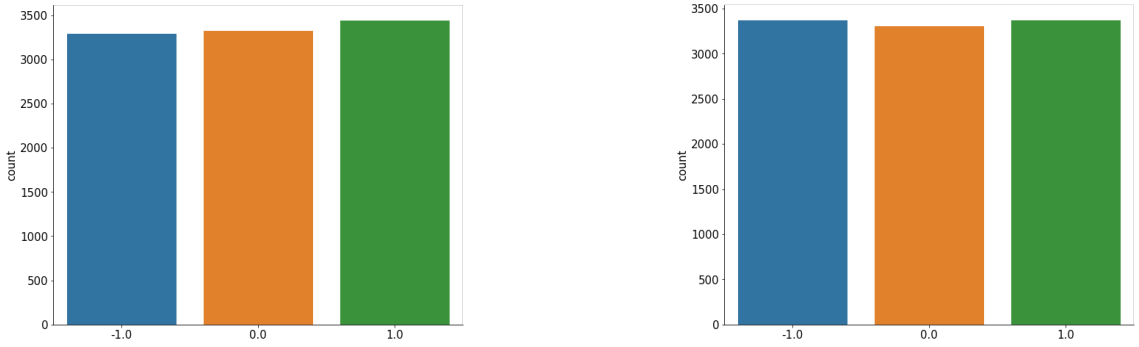


Figure 3: Desired Responses(L) and Actual Responses(R)

As is evident from these figures, the data is very balanced and so we can apply Logistic Regression to predict the classes.
Now to perform Logistic Regression, we somehow needed to convert the 3-D spikes data into a two dimensional table. We took the following approach:

- We took a neuron in one experiment and found all the time bins where it spiked.
- Since many of the neurons spiked without any specific trigger, we set a threshold of 10. If the neuron spiked more than 10 times in a time bin, we considered it active and gave it the value of 1, otherwise we considered the neuron to be dormant and gave it the value of 0.
- For each time bin, we append the data from each experiment to know which neurons were active. We had a total of 10049 trials.
- We then added the columns for various other features like response_time, gocue, etc.
- We saved this file as a csv using *df.to_csv()*. This file can be found *here* .
- This gave us a two dimensional dataframe with 81 features columns, on which we used Logistic Regression and Deep Learning Models.

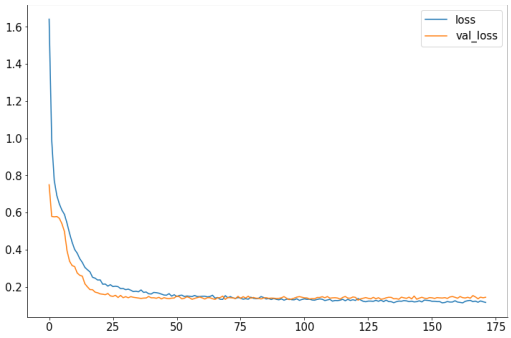
Results

- The results for this model were astonishing. The Logistic Regression model predicted correctly with an accuracy of 89% and F1 score above 90%.

	precision	recall	f1-score	support
-1	0.87	0.84	0.85	862
0	1.00	0.96	0.98	869
1	0.82	0.89	0.85	782
accuracy			0.89	2513
macro avg	0.90	0.89	0.89	2513
weighted avg	0.90	0.89	0.90	2513

Figure 4: Classification report of model trained on Logistic Regression

- We then tried to boost up the performance more by using **Keras** and **Tensorflow**, using *categorical_crossentropy* as the loss function.The validation loss for this model was very good, and the results even surpassed the previous one.



(a)

	precision	recall	f1-score	support
-1	1.00	0.98	0.99	845
0	0.87	0.95	0.91	775
1	0.95	0.89	0.92	893
accuracy			0.94	2513
macro avg	0.94	0.94	0.94	2513
weighted avg	0.94	0.94	0.94	2513

(b)

0.1 Insights

This model performed well beyond our expectations and so we needed to analyze and find the error. So we plotted a correlation matrix to see how the response is corelated with different features, and we got the following results:

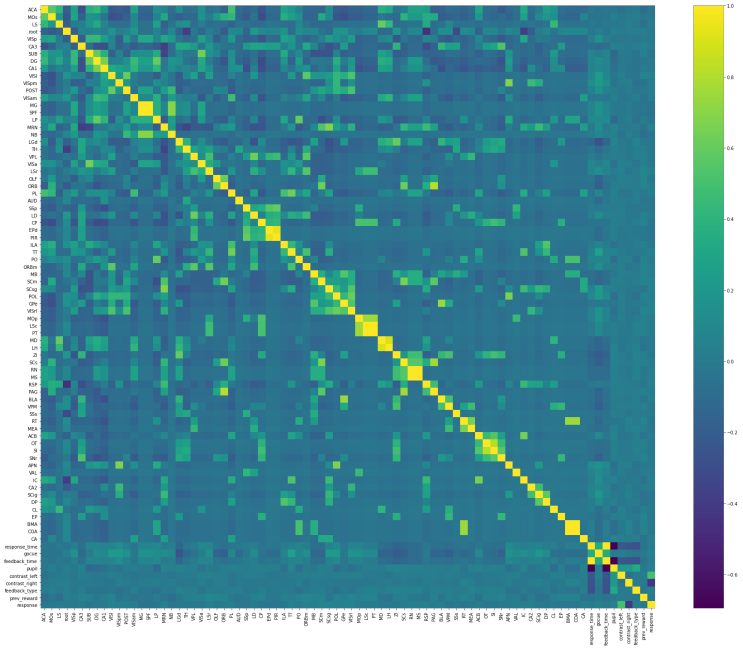


Figure 6: Correlation Matrix

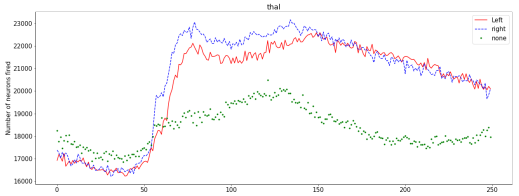
The response by the mice is the last column of the matrix, and it can be seen clearly that it is not very dependent on any columns. Response is somewhat dependent on contrast_left and contrast_right.
This model is not a good one because the responses are more dependent on external features and they seem to be in control.
Hence, we had to look closely on the neurons and their activities.

Analyzing Brain Groups

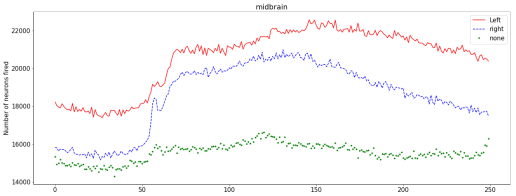
Since the previous model was more focused on external cues, we abandoned those features and went on to do some more feature engineering and visualisation

- We first researched and divided the neurons into one of the eight different groups, *"vis ctx", "thal", "hipp", "other ctx", "midbrain", "basal ganglia", "cortical subplate", "other".*
- We then replaced the name of neurons in the dat['brain_area] with their correspnding brain group.
- We then took a similar approach as before and added all the spikes of neurons in a particular brain group and plotted the result.

Two of the brain groups, the *thal* and *midbrain* which control the movement of body showed following trend:



(a) Thalamus



(b) Midbrain

We can see a spike in the number of neurons fired at 50 (500 ms), this is consistent with the fact the mouse after seeing the flashes at 500ms starts moving the wheel and that thalamus and midbrain is responsible for this. Furthermore, it can be seen that the midbrain gets more activated when the mice rotate the wheel towards left. After this graph we needed to study each neuron in detail so as to know which neuron is more activated during which kind of response.
More about this can be read *here*.

Analyzing Neurons

After the insights about different brain groups, it was evident that not only all neurons participate differently while making a decision, they also favour some responses above others.
This intrigued us to look further into the trends of different neurons during the experiments separately.
Out of the 71 Neurons, some of the neurons favoured the right response

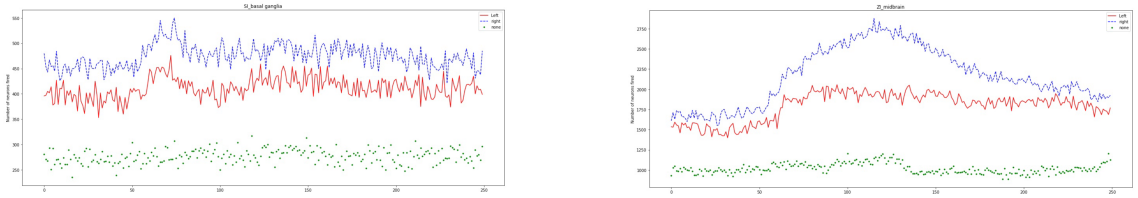


Figure 8: Right response favouring neurons

Some neurons favoured the left response and were more active than when the response was right

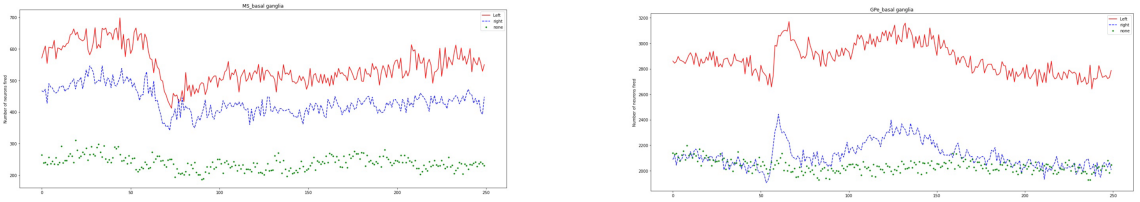


Figure 9: Left response favouring neurons

There were some neurons which had too much noise to extract any useful information from them. They also did not show any specific trend and hence we confirmed that these neurons are not triggered by external stimulus of light

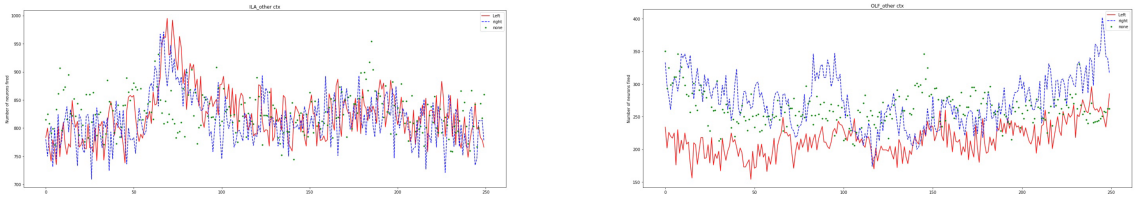


Figure 10: Neurons with noise

The graphs for all the neurons can be found *here.*

Result

We initially wanted to create a model using the neurons as the features and response as the column to be predicted, but there were various problems while creating the models.

- In every experiment not all the neurons are recorded. Out of 71 neurons as feature columns, we only get the data for maximum 6 neurons for an experiment. Due to this reason, the matrix is very sparse.
- Since the three classes to be predicted are not exactly symmetrical, with no response referring to a dormant state, and response referring to an active state, many neurons could only predict the no response and response, as in these types of cases:

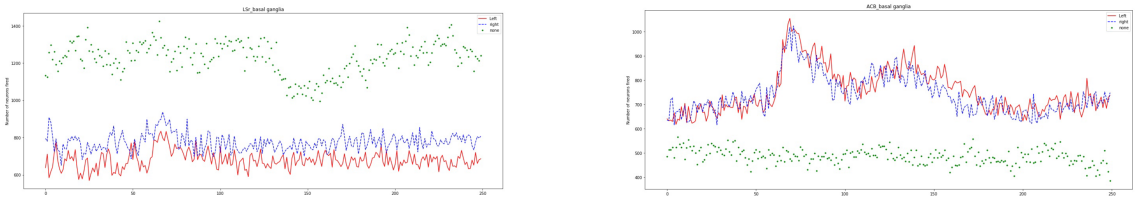


Figure 11: Neurons with responses but no distinction between left and right

Studying all the neurons, we could separate different neurons in three groups, one favouring left response, right response or no clear distinction.

LEFT	RIGHT	NONE
MS_basal_ganglia	SI_basal_ganglia	ABS_basal_ganglia
Gpe_basal_ganglia	SNr_basal_ganglia	LS_basal_ganglia
BLA_cortipal_subplate	CP_basal_ganglia	LSc_basal_ganglia
CA_hipp	EP_cortipal_subplate	LSr_basal_ganglia
CA3_hipp	SCig_midbrain	OT_basal_ganglia
IC_midbrain	MEA_cortipal_subplate	EPd_cortipal_subplate
MB_midbrain	ZI_midbrain	CA1_hipp
APN_midbrain	SUB_hipp	DG_hipp
PAG_midbrain	COA_other ctx	POST_hipp
CA2_hipp	BMA_cortipal_subplate	SCs_midbrain
RN_midbrain	PIR_other ctx	MRN_midbrain
SCm_midbrain		NB_midbrain
MOS_other ctx		ACA_other ctx
ORBm_other ctx		AUD_other ctx
PL_other ctx		DP_other ctx
		ILA_other ctx
		MOp_other ctx
		OLF_other ctx
		ORB_other ctx
		RSP_other ctx
		SSs_other ctx
		SSp_other ctx

Figure 12

Future Works

- We can now create different models for the neurons that we are studying in an experiment.
- We can now work with timeseries models like RNN and LSTM
- We are also looking into probability based models which classifies the response on its probability of belonging to a certain class.