Machine learning analysis of neural activity

Devi Prasad Gudala

Master of Science
Artificial Intelligence
School of Informatics
University of Edinburgh
2022

Abstract

Learning plays an important role in the cortical regions of the brain. We study this process in the primary visual cortex(V1) of mice while it is performing visual discrimination task. This works acts as an extension for the previous work on one particular aspect. By analyzing the neural activity collected from highly interconnected excitatory(PYR cells) and inhibitory interneurons(PV, SOM, and VIP cells). We find that stimulus onset leads to change in the neural activity and helps us better classify the visual discrimination task. Since we know which aspect of the data is more interesting when we try to find relationships between different neuron classes and stimulus type. Through experimentation we find that when reward is involved the excitatory neurons(PYR cells) contributed more towards making the decision compared to other inhibitory interneurons(pre vs post learning), this behaviour was observed in both xgboost and LSTM models. In the absence of reward the inhibitory neurons contributed more towards classifying it as stimulus type-2 compared to PYR cells(pre vs post learning).

We speculate that after the learning process cortical networks reallign themself and form new networks. In case of visual discrimination task we think that whenever the stimulus is received it forms feedforward current in the excitatory neurons and if the reward is present there was be less activity in the inhibitory network. Whereas absence of reward might trigger a recurrent current in the cortical region and show more activity in inhibitory neurons compared to excitatory neurons.

Acknowledgements

I am thankful to my supervisor, Angus Chadwick for introducing such an interesting inter-disciplinary project and providing constant support throughout the time. I learned a lot during this project and gained valuable experience in the inter-disciplinary field of AI and neuroscience. I enjoyed all the intuitive discussions I had during this time which helped me expand my domain knowledge and boost my confidence to take the next step in the field. I'm also very grateful to my parents and friends, because of their constant support and understanding I was able to relish my time at the University of Edinburgh.

Table of Contents

Chapter 1

Introduction

All the achievements made by the human race are due to our ability to learn from our environment, and yet we do not fully understand how our brains handle this important task. In any organism, the process of learning starts when the gathered data from senses are sent to the brain to derive insights from them for future reference. Especially the visual information plays a huge role during the process of learning. The visual data collected from the retina is sent to visual cortex present in the neo-cortex area of the brain(which is divided into regions V1-V5). This visual information initially passes through the primary visual cortex region(V1) which contains a dense network of excitatory and inhibitory neurons.

It is a well-known fact that neurons communicate using action potentials[14], and the connections between neurons are maintained through synapses(exchange of neurotransmitters). Excitatory neurons are the type of neurons that are more likely to fire an action potential by releasing neurotransmitters such as norepinephrine, glutamate, etc. In the case of inhibitory neurons they are less likely to fire an action potential generally, these neurons use GABA(Gamma Amino Butyric Acid) as neurotransmitter that inhibits the activity. This project uses data collected from layer 2/3 in the V1 region, which contains highly interconnected excitatory neurons PYR cells(Pyramidal cells) and different classes of inhibitory interneurons PV(Paravalbium), SOM(Somatostatin), VIP(Vasoactive Intestinal Peptide) cells. Various studies show that these morphologically unique interneurons connect to various excitatory and inhibitory neurons through synapses [43] and also exhibit a different pattern of firing by expressing peculiar molecular markers [46],[19].

Extensive research shows that during the learning process the sensory information has a considerable impact on the cortical networks by forming unique representations

that help segregate different stimuli [51],[50]. To gain a better insight the authors of [20] made an in-depth analysis of this subject through experiments. In our thesis, we use the data collected from these experiments and try to approach this problem from a machine learning perspective and see if we can gain any valuable insights.

If we look at the bigger picture it is known through diverse studies that the imbalance in the excitatory and inhibitory neurons can cause various mental illnesses such as schizophrenia [23], autism [39], and epilepsy [28]. Even though rodent and humans brain have different cortical structures and complexity, it offers a fundamental understanding regarding mechanisms that underly diversity in different interneuron classes [48].

1.1 Objectives and Contributions

Our main objective is to improve our understanding on how different classes of excitatory and inhibitory neurons behave during the learning process of visual discrimination task. Our hypothesis are as follows:

1. Initally our main objective is to understand how different timesteps contribute towards classification decision and understand in the perspective of neuroscience.

2. After establishing the results from the above objective, we seek to compare and contrast performance of different models as proposed in the chapter 3.

3. By this stage we already have some established facts which help us solve the next objective of understanding how each model expresses the importance of neuron class towards decision making.

4. Lastly we inspect if there are evident patterns found in different test subjects and provide our analysis on it.

5. Some of the sub objectives include investigating how reward influences decision making process.

Some of the main contributions in this project include extending the work done by the original authors of the paper [20] by using new machine learning and deep learning techniques. To add to this work we interpret these results using SHAP framework and explained our finding from a neuroscience perspective. For time series classification problem we have implemented an LSTM network using tensorflow 2.0, and xgboost classifier from an available package XGboost.

1.2 Organisation

The rest of the dissertation is divided into 5 chapters. In chapter 2 we discuss works related to our project such as (neuroscience based studies), time series classification and its applications. Chapter 3 provides deeper understanding of the ML/DL algorithms we plan to use in the further parts to do experiments. Chapter 4 we discuss about different experiments performed using the mentioned methods and their results. In Chapter 5 infer our hypothesis by presenting various discussions with respect to different models. Finally in Chapter 6 we conclude our work with some future prospects to the project. Additional information is presented in the appendix part which is present after the references.

Chapter 2

Background

In this section initially we discuss works that focus on neuroscience aspect of the project i.e we discuss about studies that focus on interactions between different neuron classes for different activities. Following this we discuss about studies focused on time series classification and explainable AI techniques that are relevant to our methods.

2.1 Related Works on different interneurons

Since we know that inhibitory interneurons in the cortical layers are highly interconnected, but to have a clear picture of how interneurons affect other interneurons the authors of the paper [34] conducted some experiments. They found that the PV cells have a strong effect of inhibition on each other but are very less likely to inhibit other interneuron populations. Conversely, SOM cells show less inhibition among themselves but show a strong inhibition towards other populations of interneurons. The VIP cells showed a tendency to inhibit SOM cells more compared to other interneuron cell types. Since we have a basic idea of how these interneurons affect each other it would be interesting to understand how they affect excitatory cells.

From this study, [17] it is known that inhibitory interneurons (GABAergic) control the response of excitatory neurons PYR cells based on the sensory input. In order to better understand how PV and SOM interneurons affect the PYR cells, the authors of [40] conducted experiments on transgenic mice and found that PV cells can act solo and any change in the activity of even one cell caused a change in postsynaptic PYR cells. Whereas SOM cells always acted as a group to influence the PYR cells.

The authors of the paper [9] wanted to understand interactions in cortical networks during locomotion. They discovered that VIP cells show activation during sprinting,

irrespective of visual stimulation. Conversely, they also found that damaging the VIP neurons caused the mice to stop running even when a visual cue is given. The authors theorize that the activation of these VIP cells is due to neurotransmitter based on nicotine from the basal forebrain. Following this study, the author of the paper [2] claimed that the VIP cells that received nicotinic input increased the firing rate of PYR cells in the auditory cortex.

Some studies also suggest that neural activity in these cortical regions is not fully dependent on sensory information but also dependent on recurrent connections. The authors of the paper [15] found that during visual stimulus the PYR cells showed a high correlation among themselves proving that this activity is strongly dependent on the stimulus but not due to recurrent connections. In contrast, PV cells showed very little activation pattern during the stimulus from which we can conclude that most of the activations to PV cells are from the nearby PYR cells with distinct features.

2.2 Time series classification

During recent decades Time Series Classification(TSC) has gained a lot of popularity due to its applications in various fields such as finance [45], psychology, Biomedical AI, astronomy, and meteorology [37]etc. Due to the rise in time series data collected through sensors there was boom in applications for domain of biomedical AI such as activity recognition [10], phenome classification [12] , health records [35] etc. Fundamentally based on the number of features included time series tasks are divided into two kinds- Univariate time series and multivariate time series. In univariate time series data, there is only one feature that has a temporal dimension. Whereas in multi-variate time series data there are two or more variables that contain a temporal dimension(time dimension).

Following the rise in Time Series(TS) data based on extensive research in the field of ML and DL, there were a lot of algorithms proposed to approach these problems [3]. In the early stages, distance-based metrics were used (like KNN, and euclidean) to solve the time series classification problem. The authors of the paper [32] used warping distance to reduce the multivariate time series data(samples, timesteps, features) into rectangular data (samples, features) and used K nearest neighbors to classify the data. This method was proved to give good results by other research as well [41]. Some research suggests that this multivariate time series classification problem can be approached by feature extraction [49]. But this method did not yield results as good as

distance-based methods and this is due to the fact that sequence data(time series data) do not have existing feature space and it is challenging to obtain these features using engineering techniques since time dimension makes it more complex[52]. To follow up on feature-based methods the authors of the paper [4] suggest using dimensionality reduction to reduce the multivariate time series data to univariate time series data by using tree-based techniques which resulted in Symbolic representation of Multivariate Time series (SMTS) and then we can apply supervised learning algorithms to do the classification task. It was found that tree-based techniques are more efficient in these scenarios.

Since we know that tree based techniques work well, we now try to understand literature that utilizes these techniques to get a much better understanding of the methods. The authors of the paper [8] use Xgboost classifier to do pathway based classification task and analyze gene expression data. It was found that this method outperformed other classifiers and gave convincing results from perspective of biology. Another study uses Xgboost to do cancer classification task [24] due to its high efficiency and accuracy. The authors specify that Xgboost performed better compared to other classifiers in terms of False Positive Rate. This study [36] discusses use of xgboost for image based lukemia classification. The authors find that using xgboost gave better results than using a fully connected layer for classification on backbone networks like Resnet, VGG, etc. In the context of interpretibility tree based methods are proved to more popular. Based on research tree based methods are more suitable for computational biology since they are more interpretible. Based on the material reviewed there are many advantages of using Xgboost method, hence we employ this method to analyze neural activity by converting time series data to tabular data(More details are explained in the further sections).

Now that we have an idea of how to use Xgboost classifier, we move on to the deep learning methods. During the past decade there has been a lot of success in deep learning in various fields[22]. With the sucess of Convolutional Neural Networks in the field of computer vision by authors of paper [21] a lot of hype was generated in deep learning techniques in other fields such as NLP, speech recognition, etc. All these fields have a common aspect, the data they use is sequential in nature which is similar to time series data. The authors of the paper [18] provided the first intensive study on time series classification problem on many datasets which included both univariate and multivariate datasets. Many studies show that convolution neural networks also provide state of the art results for time series tasks[22]. With introduction of recur-

rent neural networks there were lot of new variations created such as LSTMs [13], GRU etc[7]. Since LSTM based networks provided State of the art solutions on many datasets we plan to use this as one of our methods.

Chapter 3

Methodology

In this chapter, we discuss the various methods used for the analysis of neural activity and the motivation to use them. Initially, we define our approach toward checking our hypothesis in section 3.1. Following this, we discuss more details about the proposed methodologies in 3.2.

3.1 Problem setup

To check whether the formulated hypothesis holds we use the data published by authors of [20]. The neural activity was recorded from L2/3 layers present in the V1 region of the mice brain. By using the two-photon calcium imaging technique and GCaMP6f as calcium indicator [6] on different interneuron cell types. Then these areas were re-identified based on the expression of parvalbumin for PV cells, Somatostatin for SOM cells, and Vasonintestinal polypeptide for VIP cells. This data is then pre-processed to time series data by the authors, with 54 timesteps in each trail for both experiments(pre vs post learning). In these experiments(subdivided by trails) the mice learned to distinguish a visual discrimination task in a virtual reality setup done by the authors[20]. There were 3 types of trials done in each experiment(pre, post learning), in each trial the mice started running in the virtual setup. The first kind showed a vertical grating on the screen during which the mice were rewarded for licking the screen. In the second form of trails there was an angled grating(40°proportionate to vertical grating) shown on the screen, during this trail there was no reward provided and there was no punishment given for licking on the screen as well. Lastly, the third form of trial was used in between the first and second forms of trials as a way to reset the premise of the experiment, during this trial (which lasted for a random time) a short stretch of circle

followed by gray was displayed on the screen. These experiments were performed on 8 test subjects(mice) with around 250-500 trials for each test subject(for each experiment pre, post).
For a better understanding, we take neural activity 1 sec before the stimulus onset and 1 sec after the stimulus onset(17 out of 54 timesteps) to get an approximate estimate of neural activity for different stimuli. By using this data we try to do time series classification with labels as different visual stimulus types (type-1 stimulus = vertical grating, type-2 stimulus = angled, type-3 stimulus = gray-corridor). Since we plan to use machine learning and deep learning techniques to measure the classification performance we split the data into train, validation, and test sets($70\%, 10\%, 20\%$). We use train set to train our models , validation set for hyper-parameter tuning and finally test set for measuring performance.

3.2 Methods

In this section, we describe briefly the methods used and the motivation behind them. To start we plan to approach the mentioned problem in two approaches. Initially, we want to see how dependent is the neural activity on the time dimension to make a classification decision. Therefore we convert the dataset into two parts- before the onset of stimulus(0-7 timesteps) and after the onset of stimulus(8-17 timesteps). Then we apply a supervised algorithm to both variations of the dataset and observe how the same kind of model performed the classification task. Following this approach, we then apply neural network based models with a similar protocol. For a fair analysis between ML and neural network based models, we use the same algorithm for extracting insights from the data. The main reason for using ML and deep learning techniques is to find both linear/non-linear correlations in the data and interpret them in the context of neuroscience.

3.2.1 Xgboost

Xgboost also called Extreme Boosting is a tree boosting technique introduced by the authors of the paper [5] which gave state-of-the-art results on various machine learning challenges. This acts as one of the main motivation to use this supervised method for our classification task. Xgboost models try to provide solutions in a way that

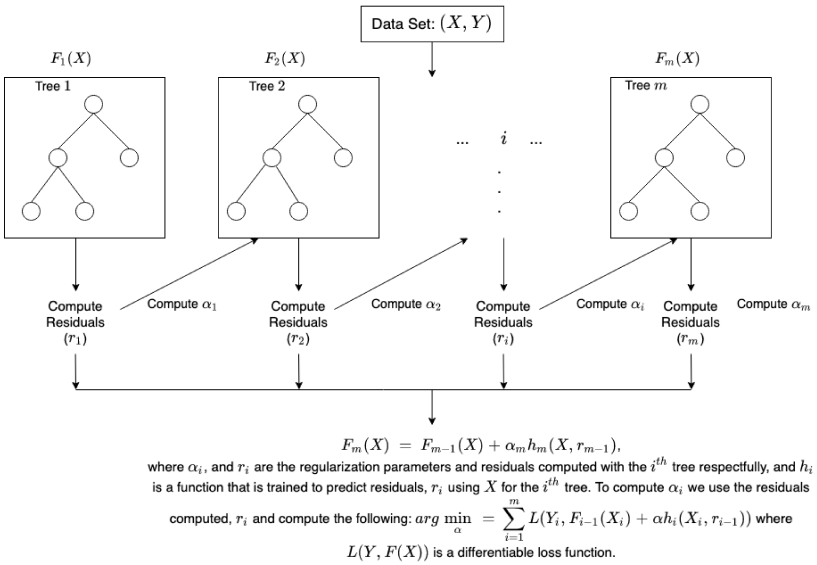

Figure 3.1: Images showing basic working of xgboost method

reduces the False Positives during classification. Xgboost has inbuilt regularization which makes it a good choice in our case since we have limited data.

To gain a better perspective of Xgboost we start by first understanding boosting algorithm. Similar to random forests there are many decision trees built to make a prediction/classification. In classification tasks, the most commonly occurring class label is taken as the output of the method. But in boosting, trees are not built on random features but on specific features. This process is done by learning from previous trees and understanding which features are important to make a decision. Similar to how various ML algorithms work we try to reduce the loss function during the training process. If we use gradient descent to do this task then it is called gradient boosting. Xgboost is an extension of gradient boosting but it includes a combination of L1 and L2 regularization to prevent overfitting.

If we have a dataset as (X,y) and we have m trees then according to xgboost a general tree F_m can be represented using the formula:

$$F_m(X) = F_{m-1}(X) + \alpha_m * h_m(X, r_{m-1}) \tag{3.1}$$

where α_m, r_m represent the regularization parameter and residuals for the m^th tree respectively. h_m represents the trained function used to predict residual(r_m). Using the

computed resudials we calculate α such that we reduce the loss as mentioned in the eq 3.2. For a much better representation we can refer to this figure 3.1

$$argmin_{\alpha} = \sum_{i=1}^{m} L(Y_i, F_{i-1}(X_i) + \alpha * h_i(X_i, r_{i-1})) \tag{3.2}$$

3.2.2 LSTM networks

Regular feed-forward networks are not compatible to accept sequence-type data as input. To make them compatible we need to send the information from previous timesteps as recurrent connections to the network. Using this ideology Recurrent Neural Networks(RNN) was introduced. But RNNs had a limit to how far they can look back in time. This is due to the fact that as the gradient passed from previous timesteps either vanished or exploded based on the scenarios [29] which impeded learning in the model. This problem was solved by the authors of [13], where they proposed a variation of the RNN network called as LSTM (Long Short Term Memory) network. This was achieved by adding gates to the cell architecture which facilitated the model to learn when to ignore the previous timesteps gradient and when to consider them important as a learnable parameter. Some studies suggest that LSTM/RNN models are considered to be biologically plausible up to an extent [31] and can consider up to 1000 previous timesteps based on the complexity [44]. In this section, we try to describe the basic functionality of LSTM cells.

LSTM cell has 4 main gates- the input gate, forget gate, output gate, and cell state. These 4 gates help modulate the information from previous timesteps and learn features from the data by updating the weight matrices for each gate. If we have an input x with n timesteps then $x(t)$ represents value of input at time t. Then different gates can be defined using the following equations:

$$f_g(t) = \sigma(W_f * x(t) + U_f * h_{t-1} + b_f) \tag{3.3}$$

Eq 3.1 shows how forget gate operates at time step t, here W_f, U_f represent the weight vectors for time step $t, t-1$ respectively. It is to be noted that the weight vectors, bias for all timesteps is same and these vectors get updated during back propogation. Similarly input gate and output gate can be defined as mentioned in Eq 3.2 , 3.3

$$i_g(t) = \sigma(W_i * x(t) + U_i * h_{t-1} + b_i) \tag{3.4}$$

$$o_g(t) = \sigma(W_o * x(t) + U_o * h_{t-1} + b_o) \tag{3.5}$$

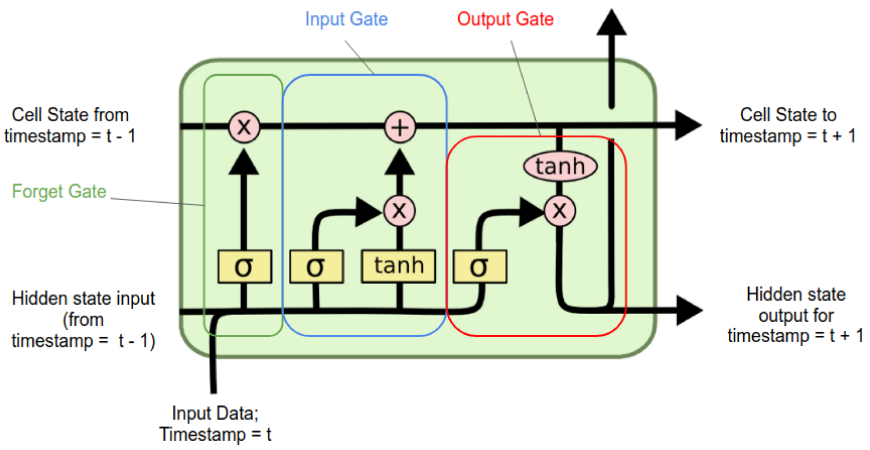

Figure 3.2: LSTM cell [13]

Using these three gates we find the cell state($c_g(t)$) at time t using the following equations:

$$c'_g(t) = tanh(W_c * x(t) + U_c * h_{t-1} + b_c) \tag{3.6}$$

$$c_g(t) = f_t * c_{t-1} + i_t * c'_t \tag{3.7}$$

Eq 3.5 shows how the previous input is taken or disregarded by multiplying previous cell state with forget gate and adding them with the current input multiplied by the current state. Now we calculate the information that needs to be sent to the next cell h_t as follows:

$$h_t = o_g(t) + tanh(c_g(t)) \tag{3.8}$$

Fig: 3.2 shows pictorial representation of the equations used. Input sequence is sent into LSTM network in the form of (batches, timesteps, features). After calculating the loss, the gradient is calculated by using back propagating through time as mentioned in the paper [13], [47].

One of the main reasons for choosing LSTM for our project is because of its state of art results on many sequence-based datasets and its ability to find linear, non-linear correlations present in the data. In case we observe overfitting we can use regularization and dropout to solve this issue. By using LSTM we can also be assured if the result is

model specific or understand variation in results compared to other models(xgboost). The only drawback would be interpreting the results from an LSTM model. Since it is a deep learning model manually accessing information from each layer and trying to analyze it would be a challenging task. Hence we use the explainable AI technique SHAP which has a proven compatibility with deep learning models(more details are presented in the next sub-section).

3.2.3 Explainable AI technique SHAP

We know that in any ML/DL problem we convert the data that we collect into features which are then fed to the model, during the training phase the model tries to understand essential interactions between the features and output some results based on the task. With the rise in State of the art solutions for various problems the complexity of the models also started increasing to the point where human experts cannot understand the underlying factors in the decision-making process of the model. This great success led to additional problems such as a tradeoff between accuracy and interpretability of the models, ethical issues, etc. To overcome these problems there has been recent research in the past decade focused on explainable and interpretable AI [16],[38],[11]. Due to this explosion in various methods to interpret the model predictions, it is hard to understand which technique works best for a given problem. In our project, we focus on one of the most famous and successful technique known as SHAP(SHapely Additive exPlanations) [26]. The main motivation to choose this technique is due to the fact that unlike other methods SHAP is a unified framework proposed to interpret predictions made by any model.

SHAP works on the principle that each feature has a contribution toward decision making and some of them contribute positively while the others contribute negatively. Based on various examples it tries to assign an importance score for each feature towards making a particular decision. If we take f as a model we want to interpret at example x, this is done by defining an expected value function e_S, where S is the subset of features defined as the following equation 3.9.

$$e_S = E(f(x)|x_s = x_s^*) \tag{3.9}$$

To get an idea of how important a feature is we try to try to analyze the model performance by adding the feature into our subset S to understand how it effects the

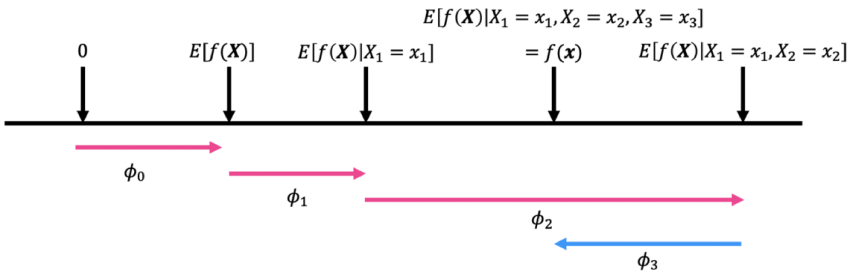

Figure 3.3: Image showing additive property of shap

function e_S. This function is defined as $\phi(i)$ which is calculated as weighted average over all possible subsets according to the following equation 3.10.

$$\phi(i) = \sum_{S \subseteq \{1,\ldots,p\}/\{i\}} \frac{|S|!(p-1-|S|)!}{p!} \left(e_{S\cup\{i\}} - e_S\right) \tag{3.10}$$

In this equation i represents the feature whose importance is being calculated, S denotes the subset of all p features, $e_{S\cup\{i\}}$ denotes the expected value with feature i included in the model, e_S denotes expected value when the feature i is not included for the model prediction. The fig 3.3 presents a much better understanding of the mentioned algorithm.

With slight variations in the aforementioned algorithm, we can adapt it to different types of models. For example authors of the paper, [25] proposed tree shap method for tree-based models such as decision trees, random forest, xgboost, etc and deep SHAP to explain neural network based models. In our analysis, we utilize xgboost and LSTM based networks to understand these models we utilize the variations mentioned by the authors.

Chapter 4

Experiments and Results

The main focus of this chapter is to describe different experiments performed to verify our hypothesis mentioned in section 1.1. Initially, we discuss pre-processing of the data followed by experiments performed using the xgboost classification method, and finally experiments performed using LSTM networks.

4.1 Data Preprocessing

The data that we have needs to undergo some preprocessing steps before feeding them to our models. First, we check for empty trails, and incomplete trails and remove them from the dataset, it is observed that there are very less empty values and by doing this we are not reducing the dataset size significantly. Following this, we try to normalize the data as most ML algorithms and neural networks accept normalized input. Most of the studies[42][1][33], show that normalization of data improves the model performance on various tasks such as classification, regression etc. For this dataset, we apply the min-max normalization technique(eq 4.1) by taking the minimum and maximum value of neuron activity in all trials(Note: This normalization is done separately for each experiment in pre vs post learning setting). The min-max normalization is achieved as follows:

$$X_{norm} = \frac{X_{max} - X}{X_{max} - X_{min}} \tag{4.1}$$

We try normalization for each cell type i.e the pair (X_{max}, X_{min}) is different for different cell types (PYR, PV, SOM, VIP) and observed that we get the same results but with less number of parameters. This is because feature engineering reduces strain on the ML/DL algorithm which in turn reduces the number of parameters of the model.

4.2 Xgboost

Following our reasoning to use Xgboost classifier in section 3.2.1 we try to perform experiments as mentioned in section 3.1. We take the mean value of the neural activity before the onset of stimulus(-1 to 0 sec 8 timesteps) and after the onset of stimulus(0 to 1 sec 9 timesteps). We then fit two different xgboost models m_b,m_a where m_b represents the model which uses data before the onset of stimulus and m_a represents the model that uses data after the onset of stimulus. Our main aim of this experiment is to understand how both models perform the same stimulus classification task before vs after stimulus onset. We apply this transformation for train, val, and test sets so that there is consistency in the experiment. When we import the Xgboost package while creating a model we use the default parameters of the method and find that they already give very good results so when further hyper-parameter tuning was done we did not find much change in the performance. As for the performance metric we use accuracy and we choose this metric because it is important that the model classifies the trail as a type-1 or type-2 stimulus but it should also be able to differentiate trails(type-3) apart from the mentioned types.

As expected m_b shows poor performance compared to m_a (table 4.1). This is because the mice cannot differentiate stimulus type before the onset of stimulus hence the neural activity cannot infer any information regarding this. Whereas after the onset of stimulus it is expected for some cell types to be more dominant and hence the model can better differentiate the stimulus type. This remained true for experiments done post learning as well i.e m_b performance was poor compared to m_a for the same stimulus classification task.

When we look at the results from (table 4.1) we see that before the stimulus onset on average the accuracy was around 60-75%(in both pre vs post learning). To have more insight when we look at different model predictions individually it is observed that the model was able to differentiate stimulus type-3(gray corridor) well compared to others this contributed to the accuracy of the model. Similarly, when we look at the models that used after stimulus onset data we observe that they were able to differentiate stimulus type-1 from stimulus type-2 more efficiently which confirms our hypothesis that changed neural activity helps us better classify the stimulus type.

Pre learning		**Post learning**	
before stimulus(m_b)	**after stimulus(m_a)**	**before stimulus(m_b)**	**after stimulus(m_a)**
66.6	96.29	68.6	96.07
64.2	97	86.6	96.7
66	88	62.5	95.83
77.7	100	78.94	100
64.1	99	73.58	98.11
73.3	93.3	75.51	91.83
82.3	91.11	70.58	100
69.38	100	70.90	98.18

Table 4.1: Performance measure(accuracy) of xgboost classifier(on test set) for 8 test subjects before and after the process of learning

4.2.1 Feature importance with SHAP

Since we know that changed neural activity after the stimulus onset helps us better identify stimulus type, we can now try to understand how different cell types contribute to the classification task. We can achieve this by looking at how the distribution(importance of each neuron) changes pre vs post learning process for each stimulus type.

To achieve the aforementioned task we use SHAP values as mentioned in section 3.2. Since we are dealing with a tree-based classification algorithm(xgboost) we use the built-in functionality Tree explainer [25] to extract the importance of each neuron towards decision making. Since there are more than 100 neurons for each test subject we try to concentrate on different classes of neurons instead of individual ones to have a better understanding of the overall impact. By using SHAP we get an importance matrix of each neuron and since we are focusing on groups of neurons we group them by the class type and then calculate the sum of all the importance values for each group. Here we chose to take the sum of all values because we want to understand the overall impact of each cell type. Now we compare how this changes pre vs post learning process. More detailed discussions are made in the next chapter.

4.3 LSTM networks

The only drawback of using a tree-based classifier was we had to take the mean of the values before and after the stimulus onset for both experiments(Pre vs Post learning). To understand the credibility of these results we try to approach the same hypothesis using a different method(LSTM's) as mentioned in section 3.2. Since LSTMs accept time series data directly, we feed the pre-processed data as mentioned in section 4.1 and use only one model for both before and after the stimulus onset.

The architecture of the network can be described as follows: 1 LSTM layer(n neurons) followed by a dense layer with m neurons and activation 'a' (taken from a set of [relu, sigmoid, tanh]), finally output layer with 3 neurons(for three different stimulus type). This network is trained using categorical cross entropy as a loss function and Adam optimizer with $lr = 1e-3$. To find the best set of hyper-parameters (m,n,a) we use Keras-tuner [30] so that they maximize the test accuracy(15-30 models were fit to find the best possible parameters for each test subject). We train the network for 50 epochs using the chosen hyper-parameters and observe that the model is pretty stable and gives consistent results. It is important to check if the created models overfit the data and this is achieved by inspecting the loss and accuracy curves of the best model chosen from the parameter search. From Fig(4.1,4.2) it can be verified that for both before and after the process of learning the models do not overfit as there is very less generalization gap(difference between train loss and validation loss). NOTE: All additional training plots are mentioned in the appendix section of the report.

From table 4.2(left) we can observe that LSTM network performs a bit better than xgboost classifier. It is also evident that there is a slight increase in classification accuracy after the learning process compared to before the learning process. It is important to note that we feed the model with time-series data that includes both situations(before/after stimulus onset). Based on this data the network classifies the stimulus type during test time. To understand the effect on the performance of the LSTM network we feed it with only data collected before the onset of stimulus(0-8sec) so that we can verify whether the LSTM network relies upon after stimulus onset data to make predictions.

By following the same protocol as mentioned in this section we get the results as follows (table 4.2(right)). From this, it is evident that for both experiments(before and after the learning process) the performance of the LSTM network decreased compared to the first case where all 17 timesteps were used. Since it is now verified that the

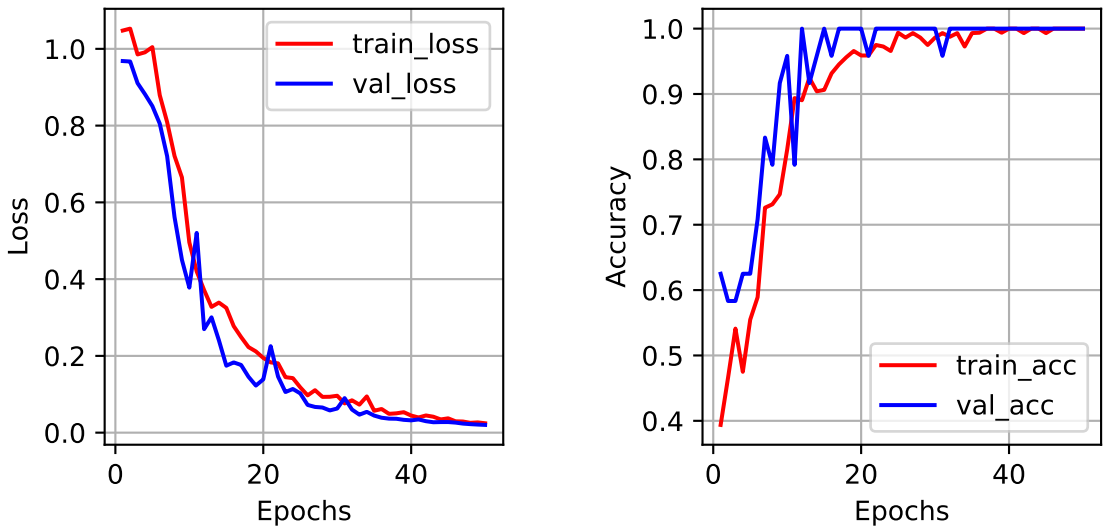

Figure 4.1: Plot showing loss curves, and accuracy curves during lstm training (Pre learning)

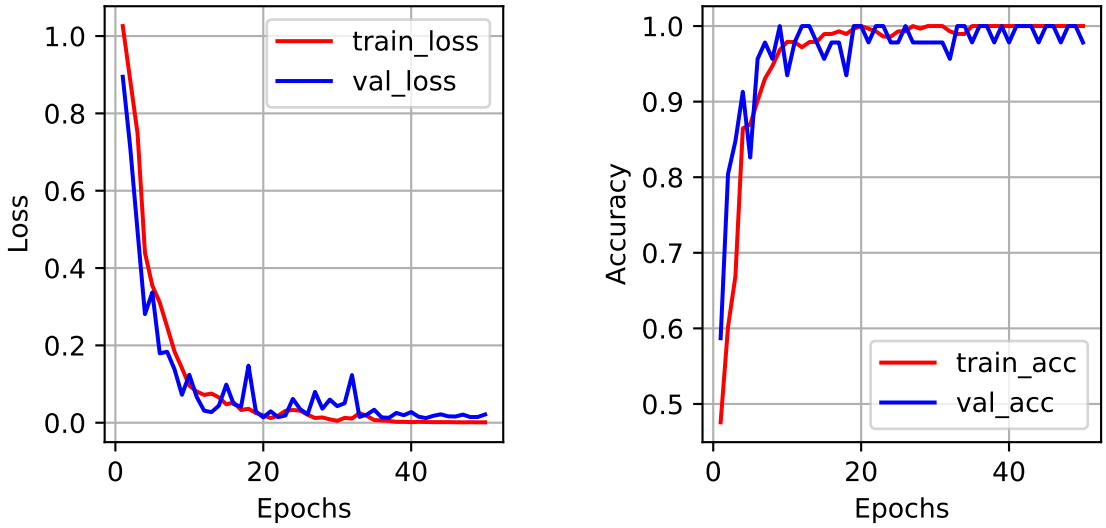

Figure 4.2: Plot showing loss curves, and accuracy curves during lstm training (Post learning)

Timesteps (0-17)		**Timesteps(0-8)**	
before learning	**after learning**	**before learning**	**after learning**
96.29	98.03	72	73.2
92.85	96.6	74.4	72
96	97.91	74	69
98	100	78	71.5
100	100	73.5	75.2
96.29	98.03	68	72
100	100	71.8	69.3
97	99	77	73.4

Table 4.2: Performance measure of LSTM network(on test set) for 8 test subjects before and after the process of learning(left), Performance measure when only first 8 timesteps are used to classify stimulus type(right).

model is looking at the important aspects of the data to make a decision, we try to find the exact feature importance of each cell type for the classification task and compare the changes for pre vs post learning.

4.3.1 Calculating feature importance with SHAP

Similar to the protocol followed by the xgboost classifier we use shap framework to extract the importance of different cell types towards decision making. Since we are dealing with neural networks we use a built in function called as deep SHAP [27] to get importance scores of each neuron at each timestep. Since it is hard to analyze activity for every timestep we take the mean value for before VS after stimulus onset. We then group the neurons by their cell label(PYR,PV,SOM,VIP) and take the sum of SHAP values for each group and compare how the performance changes (pre vs post learning). It is important to note that for the analysis we choose the best model from our previous experiments so that we have a better chance of understanding the insights from the model.(NOTE: More detailed comparison is done in the next chapter)

Chapter 5

Discussions

This chapter we dive deep into the implications of the results in chapter 4 with respect to neuroscience. Discussions and findings are presented regarding the contributions of different cell types (pre vs post learning) in this section for different models for one test subject. Following this we apply the same protocol and discuss findings with respect to other test subjects.

5.1 Importance of different cell types according to LSTM networks

It is essential to know that by importance analysis we try to understand how each cell type contributed towards classifying particular visual stimulus type. Initially we start by discussing with respect to stimulus type-1 (vertical grating) when the test subject is rewarded. According to the model that uses pre-learning data it is observed that the PYR cells had a slight decrease in importance after the onset of stimulus. Whereas the model that utilizes post learning data shows that the importance of PYR cells increased by twice after the onset of stimulus. From previous results (4) we know that after the onset of stimulus there is a change in activity which helps us better classify the stimulus type. So when we compare the importance of PYR cell type after the onset of stimulus (Pre vs Post learning) we find that on an average the importance of PYR cells towards classifying stimulus type-1 increased by 100%(doubled). This might be due to the fact that there is an increase in activity of excitatory neurons when reward is involved.

Cell type	Pre learning		Post learning		% change after stimulus onset
	before onset	after onset	before onset	after onset	
PYR	0.0335	0.031923	0.0149	0.0654790	101.12
PV	0.00058	0.000505	-0.000137	0.000592	17.23
SOM	0.00247	0.000429	0.001228	0.000964	124.71
VIP	0.0036	0.001401	0.000158	0.000028	-98.01

Table 5.1: Importance of different cell types for stimulus type-1(vertical grating) by LSTM model

Cell type	Pre learning		Post learning		% change after stimulus onset
	before onset	after onset	before onset	after onset	
PYR	0.03627	0.03127	0.0149	0.05130	64.08
PV	0.000212	0.000182	-0.000137	0.000367	101.64
SOM	0.001627	0.000504	0.001228	0.001677	232.27
VIP	0.00586	0.000795	0.000158	0.000860	8.18

Table 5.2: Importance of different cell types for stimulus type-2(angled grating) by LSTM model

When we look at inhibitory cell types, pre-learning PV cells importance after stimulus onset towards classifying type 1 stimulus increased by around 13% compared to before onset of stimulus. Post learning task we find that the importance increased by 40%. This amount of variance might be due to the fact that the model is unable to understand the feature importance before the onset of stimlus. If we look at pre-learning vs post learning after the onset of stimulus we find that there is only a slight increase in importance of PV cells(around 17%). In case of SOM cells after the onset of stimulus we find that pre vs post learning there is an increase in feature importance by 124%. For VIP cells it is interesting to note that the feature importance (pre-learning vs post learning) after the onset of stimulus decreases by 98%. This suggests that after the learning process the model learned to give less significance to VIP cells when making a classification decision as stimulus type-1.

Following stimulus type-1 results we now try to understand stimulus type-2 re-

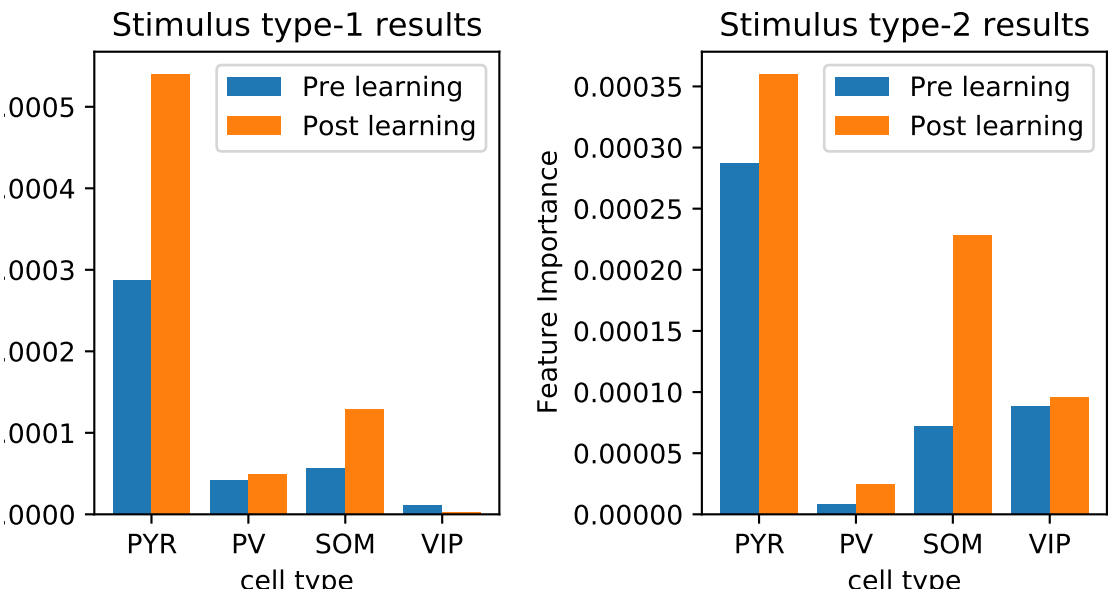

Figure 5.1: Plot showing feature importance of all test subjects together(mean value)

sults(angled grating, where there is no reward). The excitatory neurons (PYR cells) after the onset of stimulus show increase in importance by 64% (pre vs post learning). Here we do not try to compare the importance before the onset of stimulus as during this time we observe that the model performance(LSTM) was around 70% - 80% which is not very accurate and does not provide any reliable information during this time. The activity shown by inhibitory cell types is quite interesting, The PV cells after the onset of stimulus show an increase in importance by around 101% (pre vs post learning). SOM cells shows high surge in importance value (pre vs post learning) by 232%. The VIP cells also follow the same trend and show an increase in importance by 8% (pre vs post learning). From this it is evident that the inhibitory neuron type cells played a big role in classifying given data as stimulus type-2.

These results can be visualized from (table 5.1,5.2). From this we can infer that during stimulus type-1 task as there is an involvement of reward we observed increase in PYR cells importance towards decision making whereas if we look at inhibitory neuron cell types we find that collectively they had a negative impact towards decision making(43.93%) especially VIP cells had a huge negative impact on model performance while classifying given data as stimulus type-1 during test time. In case of stimulus type-2(angled grating) we observe that the inhibitory neurons collectively played an important role as classifying data during test time as stimuluss type-2. This was very less compared to that of the PYR cells importance(64%). We can assume that due to the absence of reward inhibitory neurons showed more activity compared

to the excitatory neurons.

Now we see if the same pattern is observed in other test subjects. To do this it is important to understand that we take mean of shap values instead of sum of shap values for different cell types as the number of neurons of same class differ for all test subjects. Hence we calculate the mean of the shap values for each cell type. Now we take a mean with respect to all the test subjects as well. Fig 5.1 shows the plot for pre vs post learning for all 8 test subjects(mean). We observe an overall similar pattern with variations in values as the number of neurons varied for each test subject. To see if our results are significant or not we do a Wilcoxon signed-rank test which is a non-parametric significance testing method that is suitable to understand if the two distributions are similar or not.

We find that for stimulus type -1 pre-learning we have a test statistic = 0.02 and $p = 0.0361$, post learning with test statistic = 0.029 ,$p = 0.0019$. For stimulus type-2 pre learning we have test statistic = 0.002 and $p = 0.047$, post learning test statistic = 0.0049 and $p = 0.027$. With this we can say that null hypothesis holds true and the same pattern exists in all test subjects as well.

5.2 Imporance of different cell types according to the Xgboost model

From the previous section it is evident that our hypothesis is true but to see if the results are model dependent we analyze the importance of each neuron class type using SHAP on Xgboost models. By following the same protocol we can make sure that the consistency between different experiments remains same. Since we use different xgboost models for before vs after onset of stimulus, and based on the proved hypothesis that model performance is better after the onset of stimulus. We focus on only models trained with data collected after the onset of stimulus.

Firstly we try to understand importance of each cell type for stimulus type-1(vertical grating), where mice are rewarded. The PYR cells show a slight increase(11%) in importance when compared with pre,post learning models. We can notice that this was the case for LSTM networks as well, in both cases we can argue that due to the presence of reward there is increased neural activity in excitatory cells(PYR). When we look at the different inhibitory neuron classes we observe that the PV cells show a decrease in importance pre vs post learning task whereas SOM cells show a drastic

Cell type	Pre learning	Post learning	% change
PYR	3.698	3.813	11.5
PV	0.047	0.001	-97.87
SOM	0.003	0.0198	560
VIP	0.020	-0.003	-115

Table 5.3: Importance of different cell types for stimulus type-1(vertical grating) by xgboost model

Cell type	Pre learning	Post learning	% change
PYR	3.363	2.356	-29
PV	-0.048	0.0112	123
SOM	0.0018	0.0070	288
VIP	0.0006	0.0035	483.3

Table 5.4: Importance of different cell types for stimulus type-2(angled grating) by xgboost model

increase in importance, Similar to PV cells VIP cells also showed a decrease in importance by around (115%). These results are similar to LSTM network with a variation in scale. This might be due to the fact that XGboost models use tree shap and LSTM models use deep shap, It is know that in tree shap if a feature(neuron in our case) does not contribute towards a decision then a value other than zero is given to it. Whereas in deep shap if the feature does not help in the decision making process it is assigned a value close to zero. One of the other reason might be since we take mean of the timesteps(after t = 0) hence there might be some information loss when data is fed to the model. In case of LSTM models this problem is not present as the model accepts whole time series data.

Following stimulus type-1 results we now try to understand stimulus type-2 results from table 5.4. We can observe that there is a decrease in importance of PYR cells(pre vs post learning). This result is comparable to that of the LSTM results and in both cases we can argue that due to the lack of the reward there can be a decreased neural activity in excitatory neurons(PYR cells). In case of inhibitory neurons we observe that PV,SOM,VIP cells show an increased importance towards classifying the test data as stimulus type-2. These results are comparable with that of the LSTM models, if

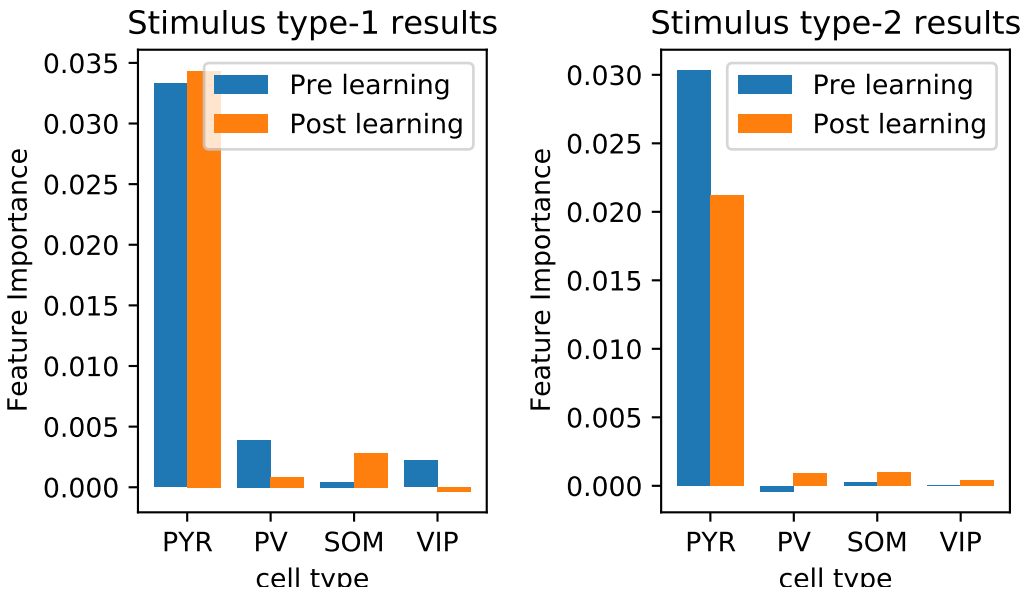

Figure 5.2: Plot showing feature importance of all test subjects together(mean value) using xgboost models

we ignore % change in terms of magnitude(since there is a variability in the model) we can perceive that due to the lack of reward there is an improved neural activity in inhibitory neuron classes.

Following up with these results we try to investigate this in case of other test subjects and find that the results are not what we expect in terms of scale and magnitude. By applying the similar protocol as the LSTM models we take the mean value for all test subjects and fig 5.2 shows the plot for pre vs post learning for 8 mice(mean value). In case of xgboost models when we apply Wilcoxon signed-rank test we find that for stimulus type 1 pre learning $statistic = 0.0049, p = 0.0037$ and post learning $statistic = 0.017, p = 0.024$. For type-2 stimulus pre learning $statistic = 0.00083, p = 0.21$ and post learning $statistic = 0.000671, p = 0.37$. This shows that the null hypothesis is true(as the p values are all below the threshold 0.5) and hence the distributions are similar across different test subjects.

Chapter 6

Conclusions and Future Scope

We conclude our report by summarizing our contribution and findings. We find it interesting that after the onset of stimulus the neural activity is better able to differentiate stimulus type-1 from stimulus type-2. In 4 we discuss performance of both xgboost and LSTM method and did critical analysis on them. Following this in chapter 5 we made discussions in context to neuroscience. We found that the learning process changed network dynamics in the cortical region. According to our results we interpret that the excitatory neurons play a huge role when classifying the stimulus type that involves reward. Whereas the stimulus type in which reward is absent, the inhibitory interneurons were more active compared to the excitatory neurons. It is also observed that some of our conclusions are inline with the works mentioned in the background section.

Since the collected data is very rich there are other aspects of the data that are not explored(due to the scope of the project). It would be interesting to see how velocity of the mouse plays a role with different neuron types during visual discrimination task. To extend our work further we can convert the problem to time series prediction problem, where we try to build a model that predicts the neural activity in the next timestep. By doing so we will have more in-depth analysis of how each individual neuron class effects other neuron classes in the context of reward. This aforementioned task could be achieved using the similar protocol we follow for this project with some fundamental changes. If we look at the other aspects of the project the authors of the original paper also performed experiments during the process of attention switching, this could be a very interesting path to explore.

Bibliography

[1] Peshawa Jamal Muhammad Ali, Rezhna Hassan Faraj, Erbil Koya, Peshawa J Muhammad Ali, and Rezhna H Faraj. Data normalization and standardization: a technical report. *Mach Learn Tech Rep*, 1(1):1–6, 2014.

[2] Caitlin E Askew, Alberto J Lopez, Marcelo A Wood, and Raju Metherate. Nicotine excites vip interneurons to disinhibit pyramidal neurons in auditory cortex. *Synapse*, 73(9):e22116, 2019.

[3] Anthony Bagnall, Jason Lines, Aaron Bostrom, James Large, and Eamonn Keogh. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data mining and knowledge discovery*, 31(3):606–660, 2017.

[4] Mustafa Gokce Baydogan and George Runger. Learning a symbolic representation for multivariate time series classification. *Data Mining and Knowledge Discovery*, 29(2):400–422, 2015.

[5] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[6] Tsai-Wen Chen, Trevor J Wardill, Yi Sun, Stefan R Pulver, Sabine L Renninger, Amy Baohan, Eric R Schreiter, Rex A Kerr, Michael B Orger, Vivek Jayaraman, et al. Ultrasensitive fluorescent proteins for imaging neuronal activity. *Nature*, 499(7458):295–300, 2013.

[7] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[8] Georgios N Dimitrakopoulos, Aristidis G Vrahatis, Vassilis Plagianakos, and Kyriakos Sgarbas. Pathway analysis using xgboost classification in biomedical data. In *Proceedings of the 10th Hellenic Conference on Artificial Intelligence*, pages 1–6, 2018.

[9] Yu Fu, Jason M Tucciarone, J Sebastian Espinosa, Nengyin Sheng, Daniel P Darcy, Roger A Nicoll, Z Josh Huang, and Michael P Stryker. A cortical circuit for gain control by behavioral state. *Cell*, 156(6):1139–1152, 2014.

[10] Yun Fu. *Human activity recognition and prediction*. Springer, 2016.

[11] Krishna Gade, Sahin Cem Geyik, Krishnaram Kenthapadi, Varun Mithal, and Ankur Taly. Explainable ai in industry. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3203–3204, 2019.

[12] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18(5-6):602–610, 2005.

[13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[14] Alan L Hodgkin and Andrew F Huxley. Action potentials recorded from inside a nerve fibre. *Nature*, 144(3651):710–711, 1939.

[15] Sonja B Hofer, Ho Ko, Bruno Pichler, Joshua Vogelstein, Hana Ros, Hongkui Zeng, Ed Lein, Nicholas A Lesica, and Thomas D Mrsic-Flogel. Differential connectivity and response dynamics of excitatory and inhibitory neurons in visual cortex. *Nature neuroscience*, 14(8):1045–1052, 2011.

[16] Andreas Holzinger, Anna Saranti, Christoph Molnar, Przemyslaw Biecek, and Wojciech Samek. Explainable ai methods-a brief overview. In *International Workshop on Extending Explainable AI Beyond Deep Models and Classifiers*, pages 13–38. Springer, 2022.

[17] Jeffry S Isaacson and Massimo Scanziani. How inhibition shapes cortical activity. *Neuron*, 72(2):231–243, 2011.

[18] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data mining and knowledge discovery*, 33(4):917–963, 2019.

[19] Yasuo Kawaguchi and Yoshiyuki Kubota. Physiological and morphological identification of somatostatin-or vasoactive intestinal polypeptide-containing cells among gabaergic cell subtypes in rat frontal cortex. *Journal of Neuroscience*, 16(8):2701–2715, 1996.

[20] Adil G Khan, Jasper Poort, Angus Chadwick, Antonin Blot, Maneesh Sahani, Thomas D Mrsic-Flogel, and Sonja B Hofer. Distinct learning-induced changes in stimulus selectivity and interactions of gabaergic interneuron classes in visual cortex. *Nature neuroscience*, 21(6):851–859, 2018.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[22] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[23] David A Lewis. Gabaergic local circuit neurons and prefrontal cortical dysfunction in schizophrenia. *Brain Research Reviews*, 31(2):270–276, 2000.

[24] Yuanyuan Li, Kai Kang, Juno M Krahn, Nicole Croutwater, Kevin Lee, David M Umbach, and Leping Li. A comprehensive genomic pan-cancer classification using the cancer genome atlas gene expression data. *BMC genomics*, 18(1):1–13, 2017.

[25] Scott M. Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature Machine Intelligence*, 2(1):2522–5839, 2020.

[26] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[27] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vish-

wanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.

[28] Chris J McBain and André Fisahn. Interneurons unbound. *Nature Reviews Neuroscience*, 2(1):11–23, 2001.

[29] Michael C Mozer. Induction of multiscale temporal structure. *Advances in neural information processing systems*, 4, 1991.

[30] Tom O'Malley, Elie Bursztein, James Long, François Chollet, Haifeng Jin, Luca Invernizzi, et al. Keras Tuner. https://github.com/keras-team/keras-tuner, 2019.

[31] Randall C O'Reilly and Michael J Frank. Making working memory work: a computational model of learning in the prefrontal cortex and basal ganglia. *Neural computation*, 18(2):283–328, 2006.

[32] Carlotta Orsenigo and Carlo Vercellis. Combining discrete svm and fixed cardinality warping distances for multivariate time series classification. *Pattern Recognition*, 43(11):3787–3794, 2010.

[33] S Patro and Kishore Kumar Sahu. Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462*, 2015.

[34] Carsten K Pfeffer, Mingshan Xue, Miao He, Z Josh Huang, and Massimo Scanziani. Inhibition of inhibition in visual cortex: the logic of connections between molecularly distinct interneurons. *Nature neuroscience*, 16(8):1068–1076, 2013.

[35] Alvin Rajkomar, Eyal Oren, Kai Chen, Andrew M Dai, Nissan Hajaj, Michaela Hardt, Peter J Liu, Xiaobing Liu, Jake Marcus, Mimi Sun, et al. Scalable and accurate deep learning with electronic health records. *NPJ digital medicine*, 1(1):1–10, 2018.

[36] S Ramaneswaran, Kathiravan Srinivasan, PM Vincent, and Chuan-Yu Chang. Hybrid inception v3 xgboost model for acute lymphoblastic leukemia classification. *Computational and Mathematical Methods in Medicine*, 2021, 2021.

[37] T Subba Rao, MB Priestly, and O Lessi. Applications of time series analysis in astronomy and meteorology, 1998.

[38] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[39] JLR Rubenstein and Michael M Merzenich. Model of autism: increased ratio of excitation/inhibition in key neural systems. *Genes, Brain and Behavior*, 2(5):255–267, 2003.

[40] Mir-Shahram Safari, Javad Mirnajafi-Zadeh, Hiroyuki Hioki, and Tadaharu Tsumoto. Parvalbumin-expressing interneurons can act solo while somatostatin-expressing interneurons act in chorus in most cases on cortical pyramidal cells. *Scientific Reports*, 7(1):1–14, 2017.

[41] Skyler Seto, Wenyu Zhang, and Yichen Zhou. Multivariate time series classification using dynamic time warping template selection for human activity recognition. In *2015 IEEE symposium series on computational intelligence*, pages 1399–1406. IEEE, 2015.

[42] Dalwinder Singh and Birmohan Singh. Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, 97:105524, 2020.

[43] Peter Somogyi and Thomas Klausberger. Defined types of cortical interneurone structure space and spike timing in the hippocampus. *The Journal of physiology*, 562(1):9–26, 2005.

[44] Ralf C Staudemeyer and Eric Rothstein Morris. Understanding lstm–a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv:1909.09586*, 2019.

[45] Ruey S Tsay. *Multivariate time series analysis: with R and financial applications*. John Wiley & Sons, 2013.

[46] Yun Wang, Anirudh Gupta, Maria Toledo-Rodriguez, Cai Zhi Wu, and Henry Markram. Anatomical, physiological, molecular and circuit properties of nest basket cells in the developing somatosensory cortex. *Cerebral cortex*, 12(4):395–410, 2002.

[47] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[48] Carl P Wonders and Stewart A Anderson. The origin and specification of cortical interneurons. *Nature Reviews Neuroscience*, 7(9):687–696, 2006.

[49] Zhengzheng Xing, Jian Pei, and Eamonn Keogh. A brief survey on sequence classification. *ACM Sigkdd Explorations Newsletter*, 12(1):40–48, 2010.

[50] Yin Yan, Malte J Rasch, Minggui Chen, Xiaoping Xiang, Min Huang, Si Wu, and Wu Li. Perceptual training continuously refines neuronal population codes in primary visual cortex. *Nature neuroscience*, 17(10):1380–1387, 2014.

[51] Tianming Yang and John HR Maunsell. The effect of perceptual learning on neuronal responses in monkey visual area v4. *Journal of Neuroscience*, 24(7):1617–1626, 2004.

[52] Yi Zheng, Qi Liu, Enhong Chen, Yong Ge, and J Leon Zhao. Time series classification using multi-channels deep convolutional neural networks. In *International conference on web-age information management*, pages 298–310. Springer, 2014.