Final Report

Rose Determan, Shuting Li, Ranfei Xu

May 12, 2022

Introduction

Our group focus on the zero maze experiment which record the activation of neurons of each mice in the form of time series and the corresponding location/behavior. Our goal is to apply different model to find the relationship between neuron activity pattern and behavior of mice.

We first focus on the mouse 409 with 25 neurons and extend to all neurons to find out the best predicted model by comparing confusion matrix for each model we tried. And then we apply the optimal model - bidirectional LSTM to the data of all neurons of all mice to detect the stability of the model by comparing the accuracy of different mouse. In order to better display the predicted accuracy of bidirectional LSTM, we also display the Zero Rule Accuracy and LSTM accuracy for all mice.

Take Away Messages:

- **Training/Test Splitting Matter**: to ensure the sequence of the time series nature, we split the data set into chunks instead of randomly selected;
- **Zero Rule Accuracy**: to demonstrate the accuracy of each model, we introduce a baseline estimate which refers to zero rule accuracy. Specifically, we select the most common class in the training data set as our predicted value. Then our goal is to make the model accuracy higher than zero rule accuracy;
- Taking **all neurons** into consideration can improve the predicted accuracy;
- **Try different lags and Shuffle the behavior:** to explore the causal relationship between neuron activation and behaviors.

Simple Model (Logistic Regression)

Starting with Mouse 409

We chose logistic regression as our baseline model, considering its simple structure. Firstly, we applied logistic model into one mouse, No.409, to see the model performance.

Because mouse 409 has 110 neurons' recording, to decrease the dimension of our predictors (neurons), we used PCA to extract the main information of our raw data, and then we chose first 25 principle components as our predictors, to fit the logistic model.

Below is our model prediction result on the 30% testing data, setting 0.5 as the threshold of probability that staying closed arm or open arm. From the confusion matrix, we can see the proportion that predictions match real behaviors is around 71.7%.

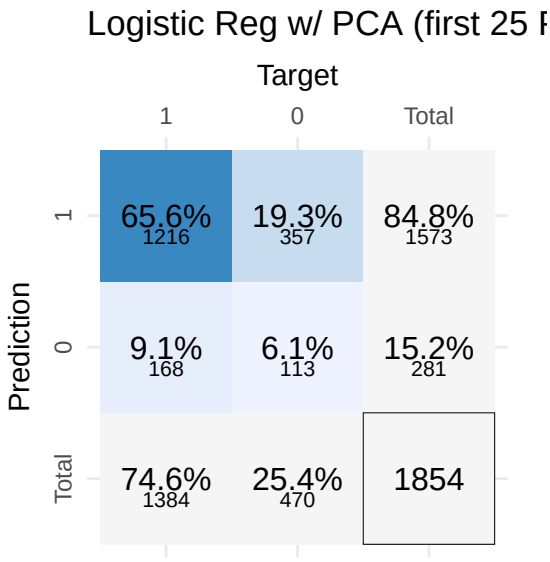


Figure 1: Logistic regression confusion matrix for mouse 409. The accuracy of this model is 0.72 compared to the baseline accuracy of 0.75

All Mice

To explore the performance of logistic model on all mice, we applied logistic model into all mice data separately, with first 25 principle components of each mouse.

After we did prediction for all mice, we combined all results together, and calculated confusion matrix to see the average performance of logistic model. We can see the average accuracy of logistic is around 71.1%.

To identify the model performance difference between mice, we drew ROC curves for all mice, we can see logistic model performance good on mouse 274, but showed worse fitting on mouse 254, mouse 255 and some other mice.

To see more clearly, we also drew the accuracy comparison plot for all mice. Zero Rule accuracy means the accuracy that we always chose the class has major proportion in our raw data, to compare them with model accuracy, we can easily identify logistic model performance on each mouse.

Neural Network Models

Simple Neural Network Models: Mouse 409

Reason we try this model: Does the behavior impact neural activity? We assume the past location has influence on the current state of the neurons.

To improve our prediction accuracy, we tried neural network on mouse 409, this model is more complex and precise than logistic model. Same with logistic, this model use current neuron activities to predict the current mouse location. But the difference is for this model we used all 110 neurons as predictors, because neural network has ability to process big data.

What's more, to identify the question about whether behavior impact neural activity, we explored the relationship between past locations with current neuron activities, the result showed that with 1 and 5 shift forward of location, the model accuracy can be improved a lot, so we assume the past location has influence on the current state of the neurons.

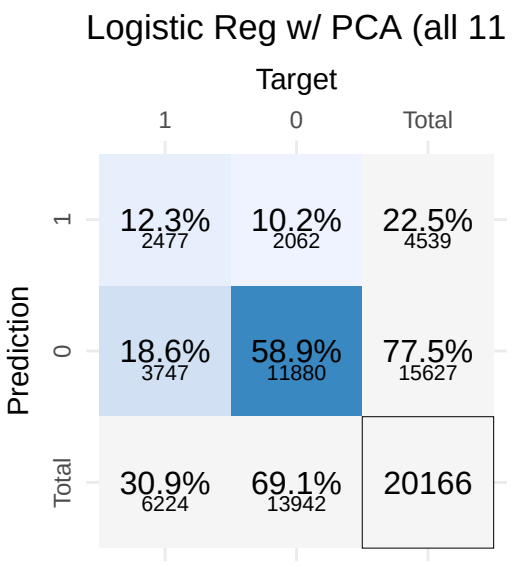


Figure 2: Logistic regression confusion matrix for all mice models. The accuracy of this model is 0.71.

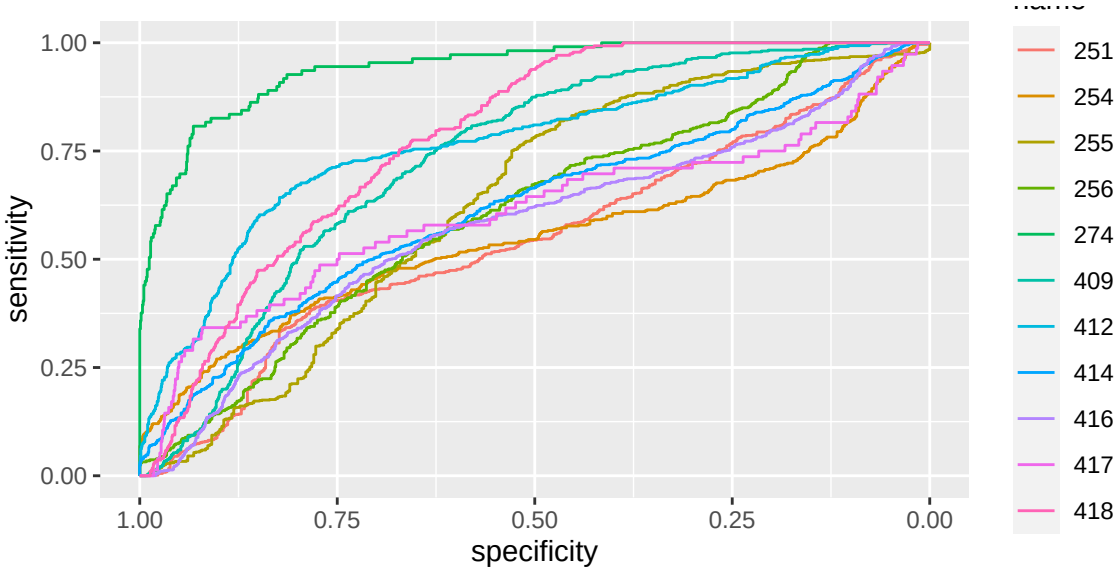


Figure 3: Logistic ROC Curves for all mice

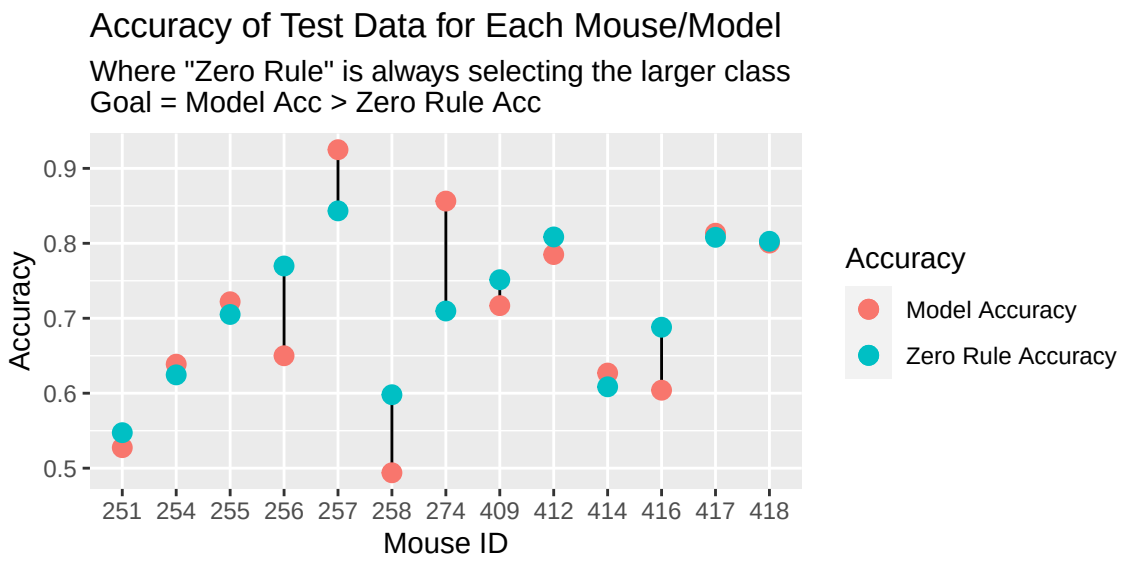


Figure 4: Logistic Accuracy Comparison plot for all mice

```
##                                model_name zero_rule_acc model_acc
## 1       Logistic Reg w/ PCA (first 25 PCs)     0.7464941 0.7168285
## 2 Neural Net with 0 Level Shift of Outcome     0.7464941 0.7696872
## 3 Neural Net with 1 Level Shift of Outcome     0.7464941 0.7853290
## 4 Neural Net with 3 Level Shift of Outcome     0.7464941 0.7384035
## 5 Neural Net with 5 Level Shift of Outcome     0.7467603 0.7991361
```

RNN with time lags : Mouse 409

The reason why we try this model is that simple neural network doesn't take into account about time series data.

Since simple neural network model does not take into account about the order of sequence data, we tried RNN model, which is more helpful in modeling sequence data.

LSTM and Bidirectional LSTM

We were interested in applying the LSTM (long short-term memory) model, since it addresses some of the shortcomings of the RNN structure. Simply, the LSTM model handles long term memory well. Additionally, the LSTM structure includes "gates" that allow the model to remember or forget information.

We also considered a bidirectional LSTM, since it is unknown whether a neuron's activity impacts behavior or the behavior triggers neuron activity. A bidirectional model is one that is fit both forward (in the typical way that a sequential model is fit) and backward). We believe that is the bidirectional model has a higher accuracy than the traditional model, then there is preliminary evidence to suggest that the behavior impacts neuron activity.

https://analyticsindiamag.com/complete-guide-to-bidirectional-lstm-with-python-codes/

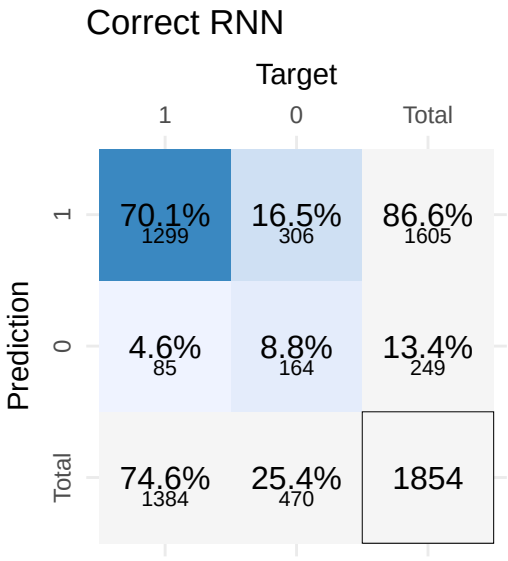


Figure 5: Right RNN confusion matrix for mouse 409. The accuracy of this model is 0.724.

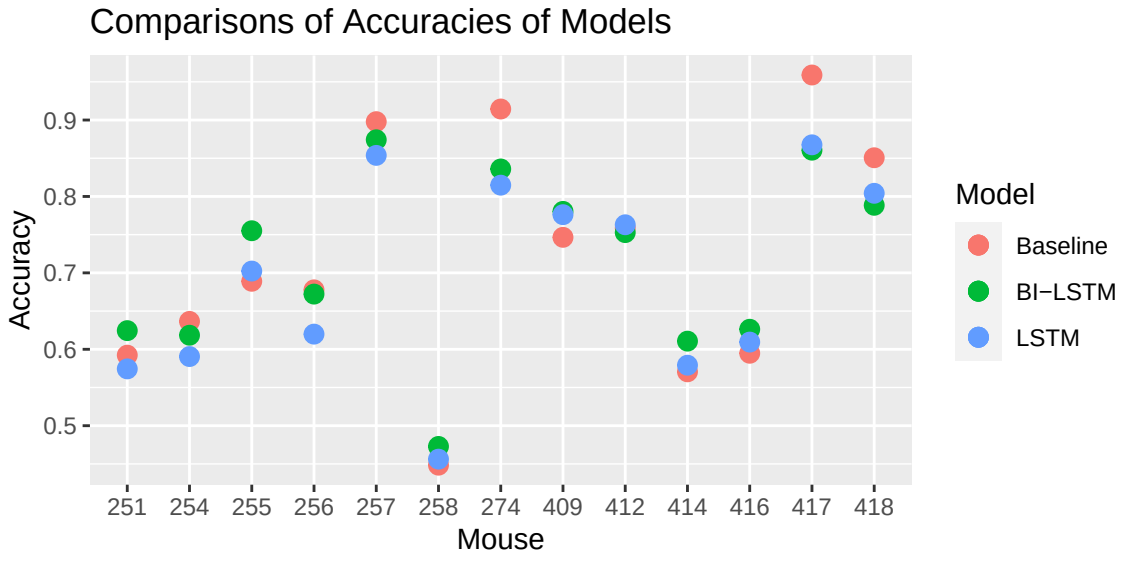


Figure 6: This plot shows the accuracies of each of the three models. The baseline model shows the accuracy when we select the most common class from the training dataset. For 6 of the 13 mice, the baseline model has the highest accuracy, and for 6 mice the bidirectional LSTM model had the highest accuracy. In only one case, the LSTM model had the highest accuracy.

Table 1: Comparisons of accuracies of models

Mouse	Best Model	Baseline	LSTM	Bi-LSTM
409	BILSTM	0.746	0.776	0.780
412	LSTM	0.759	0.763	0.753
414	BILSTM	0.570	0.579	0.611
416	BILSTM	0.595	0.609	0.626
417	Baseline	0.959	0.868	0.861
418	Baseline	0.851	0.804	0.788
251	BILSTM	0.592	0.574	0.624
256	Baseline	0.678	0.620	0.672
257	Baseline	0.898	0.854	0.874
258	BILSTM	0.448	0.456	0.473
274	Baseline	0.914	0.815	0.836
254	Baseline	0.637	0.590	0.618
255	BILSTM	0.689	0.702	0.755

Conclusions

One of our main findings was the difference between randomly splitting the data set and "chunking" the training and testing data. When we randomly split the data into training and testing samples, the models have high testing accuracy, but we lose the sequence of the data. This is also allowing the model to learn from things that have already happened, and this might lead to an misleadingly high accuracy. The appendix shows the results of our incorrect model that led to this discovery.

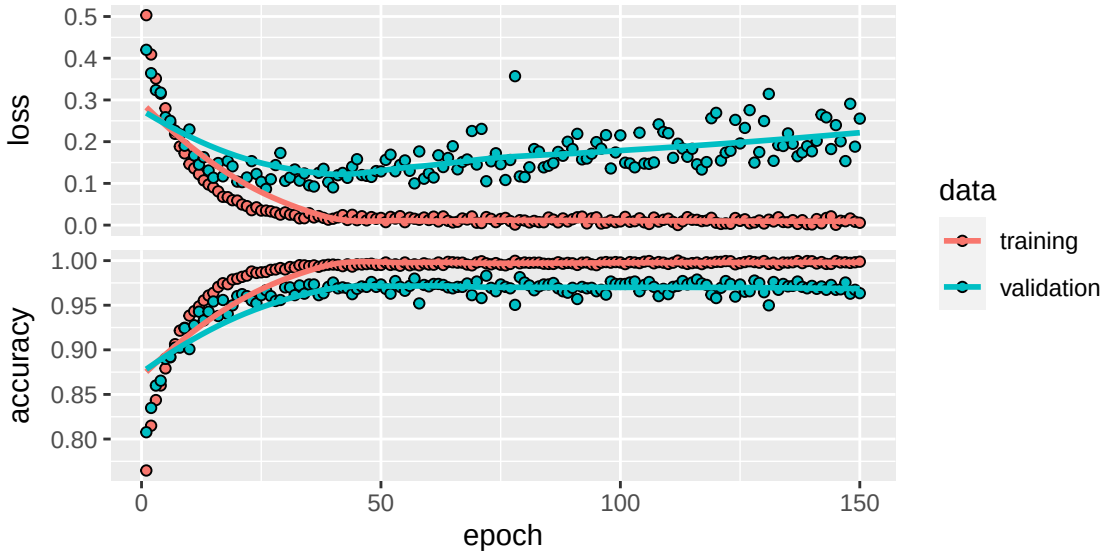
Another finding is the difference between the LSTM model and the Bidirectional LSTM Model. With the LSTM model, we are assuming that the neurons impact behavior, but with the Bidirectional LSTM Model we allow for the possibility that the behavior influences the neuron activity. In our results, we found that the Bidirectional LSTM model leads to higher prediction accuracy, and this may suggest that behavior influences the neuron activity.

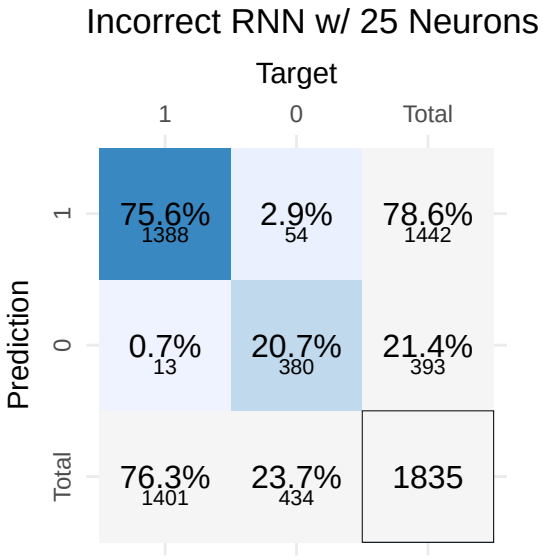
Appendix

Incorrect Version w/ randomly selected train/test : Mouse 409

At first, we incorrectly split the training and testing set randomly. The below shows the *incorrect* model. Interestingly, this incorrectly prepared model performs better than the RNN that is correctly setup.

```
## Model: "sequential_5"
## _________________________________________________________________________________
##  Layer (type)                       Output Shape                    Param #
## =================================================================================
##  simple_rnn_1 (SimpleRNN)           (None, 100)                     12600
##
##  dense_15 (Dense)                   (None, 50)                      5050
##
##  dense_14 (Dense)                   (None, 1)                       51
##
## =================================================================================
## Total params: 17,701
## Trainable params: 17,701
## Non-trainable params: 0
## _________________________________________________________________________________
```




Incorrect RNN w/ 25 Neurons
Target
1
0
Total
Prediction
1
0
Total
75.6%
1388
2.9%
54
78.6%
1442
0.7%
13
20.7%
380
21.4%
393
76.3%
1401
23.7%
434
1835