Report of MA679 Final Project

Handing Zhang, Kosuke Sasaki, Kuangyou Chen

2022/05/11

Introduction

The data set records the result of three various mouse experiments. The experiments of direct interaction and opposite sex use the same six mice, while the experiment that related to zero mazes adds seven more mice. With two additional data frames of mice in the folder, there are 27 data frames total and each of them represents a single mouse in a certain experiment. Each row contains exactly two binary responses and various numbers of normalized independent features corresponding to firing rate of neural signal of cells. In addition, the time spent on behavior observation in each dataset is not constant, ranging around 10 minutes. The time breaks between each row is 10 hz, which is 0.1 second.

The observation of our work by data manipulation and exploratory data analysis shows some missing values only in the D_409 dataset, which is a combination with the behavior record of No.409 mouse and relevant Z-score in direct interaction experiments. We believe that the missing values are not distributed randomly as it shows characteristics with periodic sequence. For all the normalized features in the dataset, no significant outliers are detected.

We decided to focus on the data of Elevated Zero Maze Experiment. What we attempted initially is to fit the Autoregressive(AR) logistic model as a baseline model to the dataset of Z_416(add why we choose Z-416), which shows normalized experiment features of the No.416 mouse. The dataset is split into the training set and test set sequentially, with a ratio of 8 to 2. All the 26 neurons are applied as predictors, with the lag equaling natural numbers from 1 to 20 to predict response.

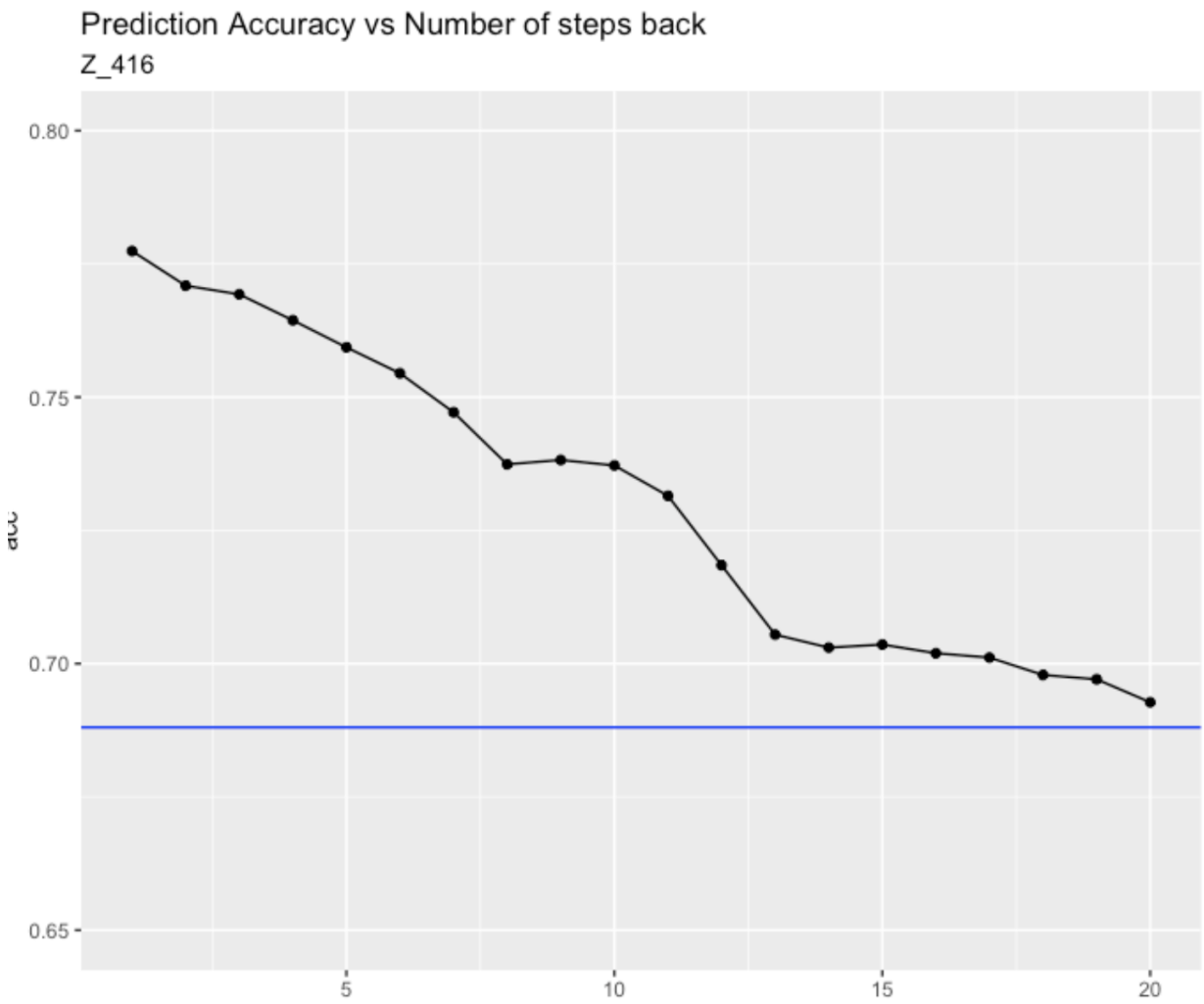
Furthermore, to make a generalized model, we train the Recurrent Neural Network(RNN) with PCA based on Z-416 and fit the model to other mice. We also .

AR (baseline) Model

We first performed some data wrangling, where we created backward lags of different numbers of our data and then combine them. In such way, each row contains the information of not only the current time, but also the information in the past L steps, where L equals the number of lags we create.

We then fit the logistic regression on the binary response of behavior status taking value of 0 or 1, corresponding to the mouse being in the open arm or the closed arm, respectively. We fitted the model for number of lags being from 1 to 20 to see how the length of time backwards taken into consideration can affect the prediction accuracy of our model. For each model we, as described in the introduction, pick the first 80% of the rows as training set and test it on the rest 20%. We did not pick the training and testing set randomly in order to take into consideration the temporal correlation of our time series data.

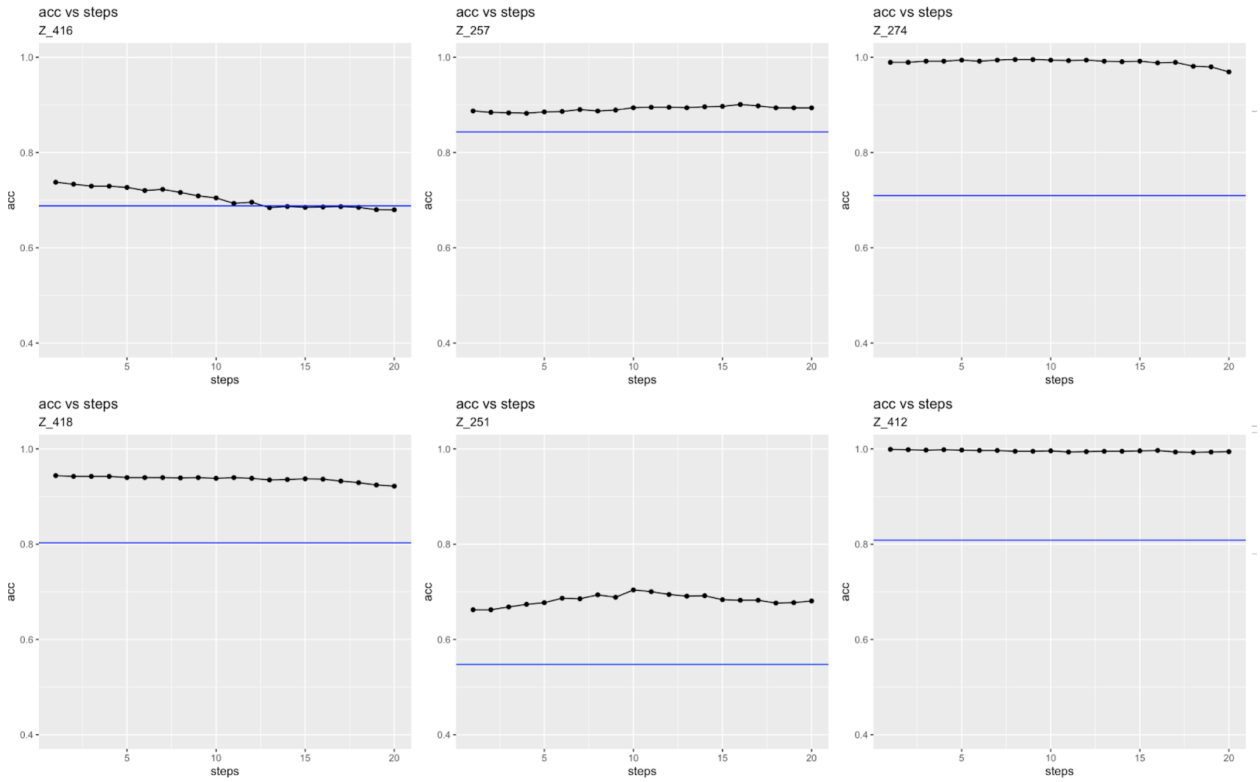
This is how the prediction accuracy changes as we change the lag from 1 to 20 for mouse Z_416



It seems that the more lags wer create, the lower the prediction accuracy becomes.

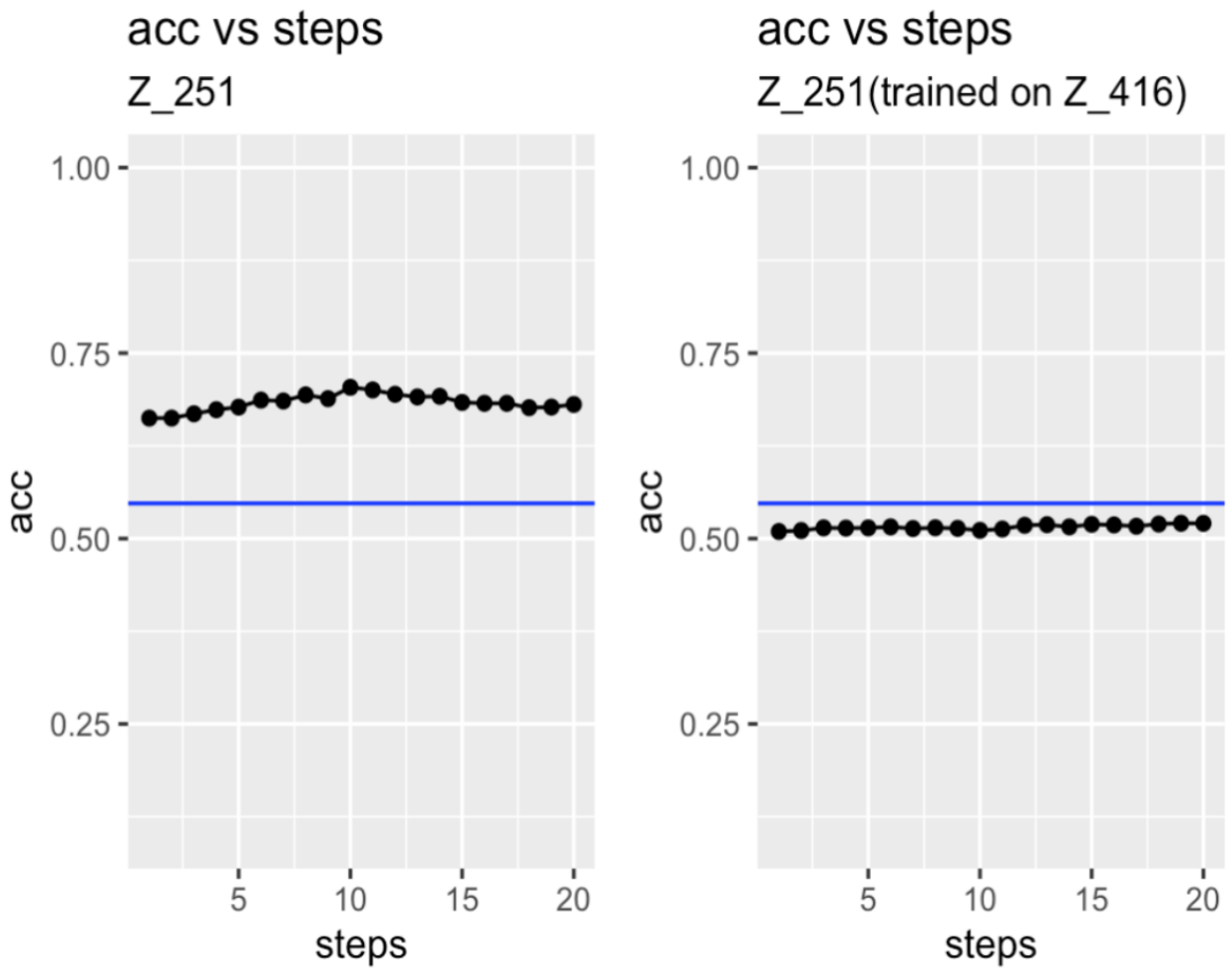
In order to generalize our model and compare the results. We performed PCA to 6 mice in the Elevated Zero Maze Experiment, keeping the 15 largest principal components for each, and fit our AR logistic model, with lags from 1 to 20.

The plots show how the accuracy for the mice changes based on the number of lags. The blue line shows the larger proportion of behavior in this experiment. Each with the first 80% observations as training set.



As we can see from the plots, the model has in general better performance than simply using the proportion of training set as prediction.

We also tried training the model only on Z_416 and test it on other mice. Below is the comparison of the model accuracy between the model trained on the mouse itself and the model trained on Z_416.

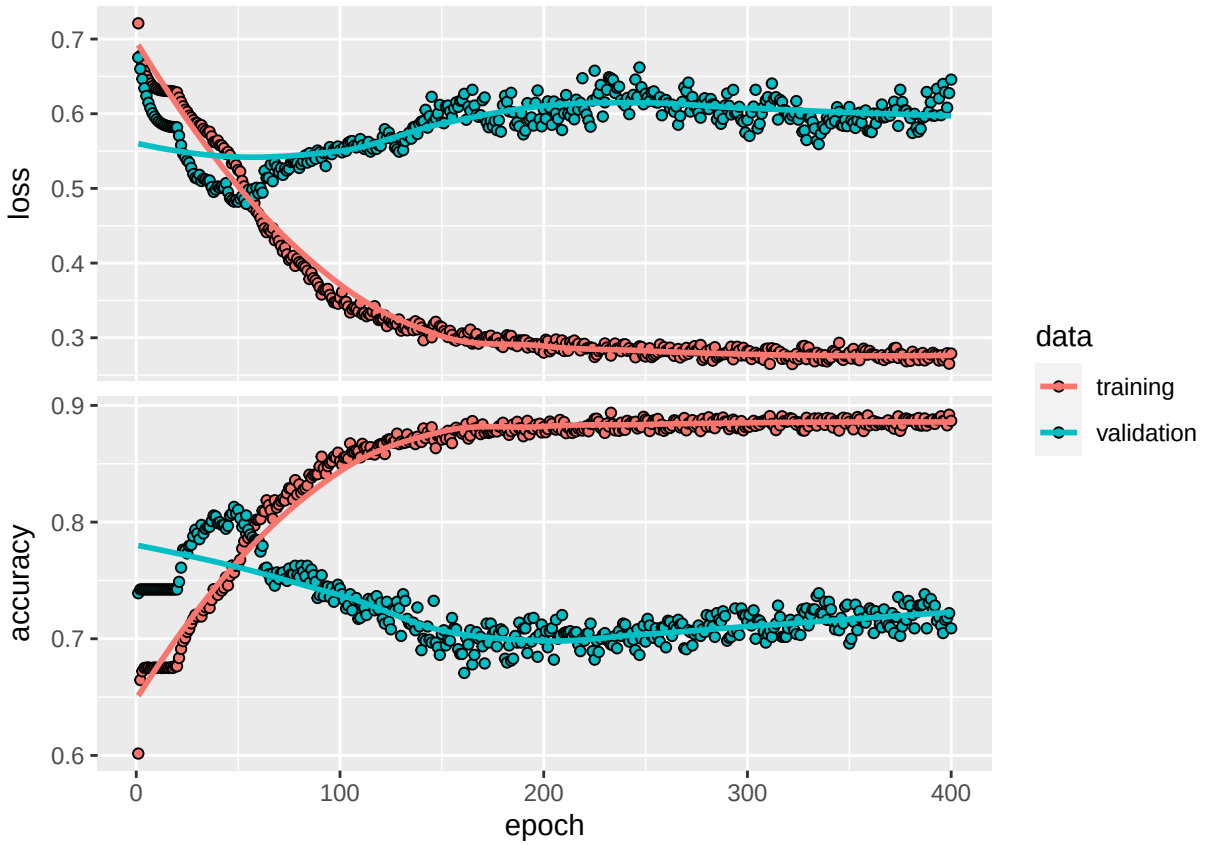


We can see a drastic decrease in accuracy. This might be explained by the differences of cells and consequentially, the principal components between the mice.

RNN Model

With normal PCA

To make generalized model, we first apply Principal Component Analysis (PCA) to all the mice, by which we expect all the mice have similar and same number features.We picked up first 26 principal components for each mouse, which is the possible largest number of principal components each mouse can have in common. After applying PCA, we trained RNN on Z406 mouse. The RNN model has simple input layer with 10 units and one hidden layer with 2 units of Relu activation function, and output layer with 1 unit of Sigmoid activation function, and was trained in 400 epochs with 64 batch size. The trained RNN results are as follows.



Based on the plots above which show the training process of the RNN, the accuracy of validation dataset converged on 0.7089431, which is almost same as the accuracy of AR model. Next, we fit this RNN model to other mice data with 26 principal components. The accuracy of this model fitted to other mice datasets are as follows.

Mouse	Accuracy
Z_251	0.4280347
Z_257	0.4278932
Z_258	0.5919601
Z_274	0.4031117
Z_409	0.5429681
Z_412	0.4043310
Z_418	0.6856633

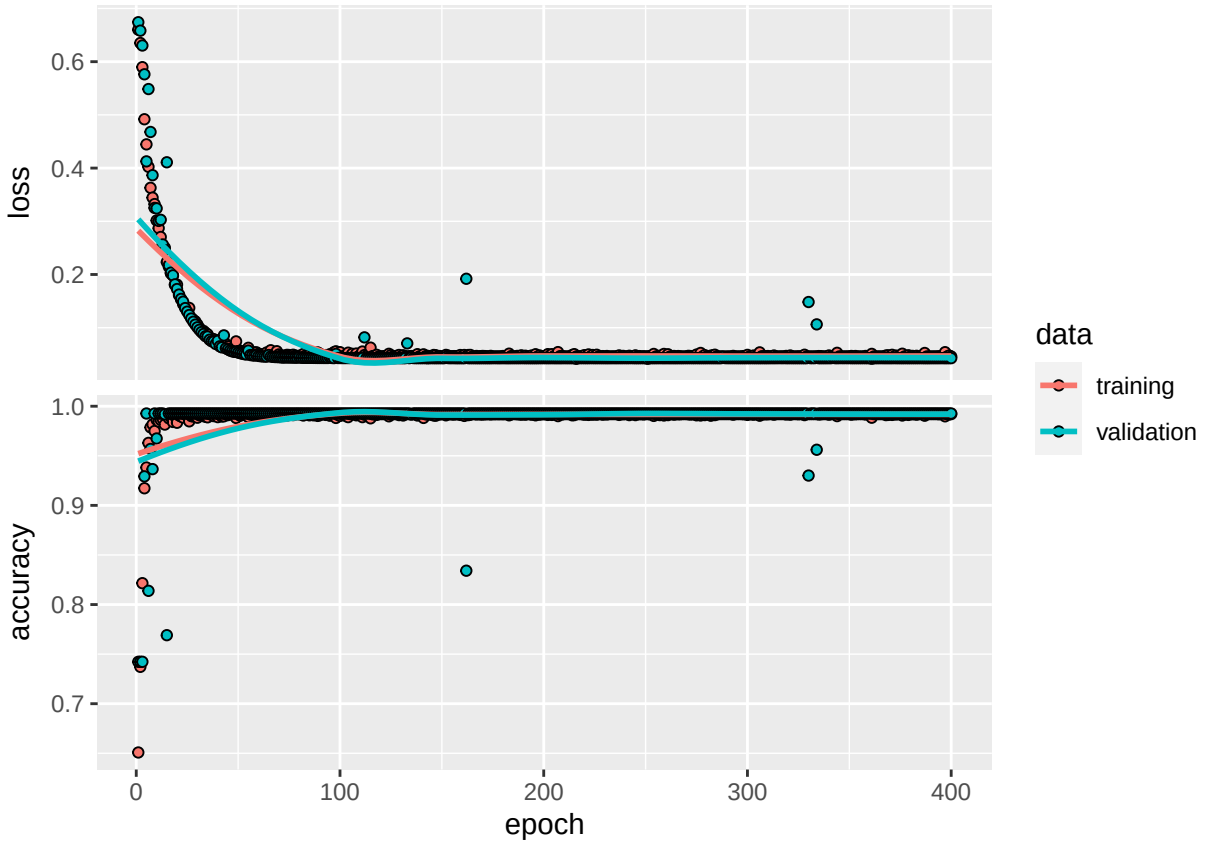
As you can see, accuracy ranges from 0.4031117 to 0.6856633, which seems to be by chance and most of them have accuracy similar to or much worse than just flipping a fair coin.

We expected that the Principal Components of each mouse after applying PCA would have similar properties and contribute to mouse behavior to some extent in the same way, in which case we could predict mouse behavior by fitting a single RNN model to all the mice. However, it seems not be the case in our experiments, and Principal Components of each mouse may have each properties.

The RNN with different PCA for Y=1 and Y=0

In the previous section, a single PCA was applied to each mouse, but the Principal Components obtained from the PCA might not have similar distributions across mice. Therefore, we decided to fit a different PCA

to each mouse for Y = 1 and 0 behaviors in the hope that this would increase the information Principal Components can explain and each mouse has similar distribtuions of Principal Components each other. The results are as below.



When we trained the same RNN model on Z416 mouse with Principal Components obtained from the PCA applied separately to the observations with Y=1 and Y=0, we saw a dramatic improvement of the accuracy when fitting the model to validation dataset of Z416. The accuracy was almost 1. This is presumably because each of the Principal Components at Y = 1 and Y = 0 shows a specific distribution, and therefore the estimation by RNN is more accurate.
Then we fit this model to all the other mice, and the results are as follows.

Mouse	Accuracy	Proportion_of_Y1
Z_251	0.9572670	0.4526266
Z_257	0.9240356	0.8432806
Z_258	0.5987914	0.5980058
Z_274	0.2661480	0.7096774
Z_409	0.5219291	0.7514554
Z_412	0.8099548	0.8084935
Z_418	0.2802403	0.8027255

This time, we have some accuracy which seems not to be by chance compared to the proportion of Y=1 for each mouse, which are the ones of Z251 and Z418. The accuracy of the two are extremely high and low. Based on this results we can expect that by separating Y = 1 and Y = 0 when applying PCA, the distribution that Principal Components had for each mouse became very similar to or the opposite of the distribution of Principal Components for Z416, and it lead to the extremely high accuracy and low accuracy when the RNN fitted to Z251 and Z418. The other mice also show relatively high or low accuracy compared to the results when applying PCA without separating Y=1 and Y=0. These results suggest that we could

compare the Principal Components obtained by applying PCA to the observed values of Y=1 and Y=0, and if the distribution is similar between mice, we can fit a generalized RNN model to the mice.

Conclusion

For the AR model, we see around 75% of prediction accuracy on Z416 when taking lags of 5. By performing PCA, we fit the model on other mice and the prediction accuracy varies, probably due to the inbalance of our outcome. When trying the model on Z416 and testing on other mice, we see drastic decrease in accuracy, indicating that PCA is not having a good performance in generalizing the model.

To create generalized model, we tried to train the RNN on Z416 Principal Components and fit it to all the mice with the same number of Principal Components, which have low accuracy across mice. However, when applying PCA by separating Y=1 and Y=0 observations in each mouse, the RNN model trained on Z416 has extremely high accuracy of the validation dataset when fitting it to M416 itself. Furthermore, the generalized RNN sometimes shows very high accuracy which suggests that we could make generalized RNN for the mice with similar Principal Components distributions.