Exploring Neural Activity in Mice to Predict Test Feedback

6/12

Student Information

Name: Nicholas Fitzpatrick

Student ID: 919939686

Abstract

This project explores the prediction of feedback given from mice using a range of diverse variables. Through exploratory data analysis, patterns were found throughout the data set. It was observed that certain brain areas consistently exhibited similar levels of average spike counts across different trials within the same session. The heterogeneity found in the data was attributed to differences in measured neurons across sessions. The presence of shared brain areas indicated some level of homogeneity between mice. A large data frame with 10 variables was built in order to predict the feedback of mice. A logistic regression model was used and achieved a 71.5% success rate in predicting the feedback type, and demonstrate patterns across sessions. In the future researches should ensure that consistent brain areas are measured across sessions and also take time data into consideration. This study highlights the potential of diverse variables in predicting feedback.

Introduction

The primary objective of this project is to build a predictive model to predict the feedback type of each trial using neural activity data regarding spike trains and the left and right stimuli. To achieve this, a subset of 18 out of 39 recording sessions conducted by Steinmetz et al. (2019) will be used. These sessions involved four different mice: Cori, Frossman, Hence, and Lederberg. In the original study, 13 mice were trained, but due to health complications, experiments were carried out on only 10 of them. During the recording sessions, visual stimuli were presented at the center of screens directy left and right of the mice. The various stimuli consisted of different levels of contrast levels (0, 0.25, 0.5, and 1) in which 0 means that there is no stimulus. The mice were presented with a wheel in front of them in which they were given a water reward when completing a task correctly. The task consisted of spinning the wheel left when: the left contrast was greater than the right contrast, spinning the wheel right when: the right contrast was greater than the lest contrast, and not spinning the wheel when there were no contrasts. If the contrasts were equal the mice were rewarded randomly for left and right turns. The activity of neurons in the mice's visual cortex were made available in the form of spike trains which are collections of different time stamps that correspond to neuron firing. In this project, spike trains of neurons from the onset of the stimuli to 0.4 seconds post-onset are focused on. The original study "identified organizing principles for the distribution and character of the neuronal correlates." By leveraging insights gained from the original study and utilizing the data, a predictive model will be developed that accurately predicts the feedback type for each trial based on a variety of newly found predictors.

Exploratory Data Analysis

Datastructures Described Across Sessions

The data for this project contains a total of eighteen sessions spanning four different mice. The variables within each session include "mouse_name", (The name of the mouse for specific sessions), n_brain_area (the number of unique brain areas), n_neurons (the number of neurons), n_trials (the number of trials in each session), and success_rate (the ratio of successful trials to the total number of trials).

mouse_name	n_brain_area	n_neurons	n_trials	success_rate
Cori	8	734	114	0.6052632
Cori	5	1070	251	0.6334661
Cori	11	619	228	0.6622807
Forssmann	11	1769	249	0.6666667
Forssmann	10	1077	254	0.6614173
Forssmann	5	1169	290	0.7413793
Forssmann	8	584	252	0.6706349
Hench	15	1157	250	0.6440000
Hench	12	788	372	0.6854839
Hench	13	1172	447	0.6196868
Hench	6	857	342	0.7953216
Lederberg	12	698	340	0.7382353
Lederberg	15	983	300	0.7966667
Lederberg	10	756	268	0.6940299
Lederberg	8	743	404	0.7648515
Lederberg	6	474	280	0.7178571
Lederberg	6	565	224	0.8303571
Lederberg	10	1090	216	0.8055556

Table 1.0: Data structure across sessions.

Exploring Neural Activities During Trials

In order to gain a better understanding of the data structure, Session 3 is chosen. The session was arbitrarily chosen and contains 619 neurons located in "DG","VISam","MG","CA1","SPF","root","LP", "MRN", "POST", "NB", and "VISp" parts of the mouse brain. The average number of spikes for each neuron in each brain area is calculated in order to explore neural activity during trials.

Average Spikes Across Neurons of Unique Brain Areas in Session 3: Trial 10

Unique Brain Area	Mean
CA1	2.64
DG	4.65
LP	2.50
MG	2.22
MRN	6.22
NB	2.02
POST	1.00
SPF	3.67
VISam	2.09
VISp	1.39
root	0.92

Table 2.0 Neural Activity in Session 3: Trial 10

Table 2.0 demonstrates data on neural activity in Session 3. Analyzing trial 10 of session 3, it is observed that the MRN (Midbrain Reticular Nucleus) has the highest number of spikes per area compared to other brain regions. Following MRN, the DG (Dentate Gyrus) and SPF(Somatosensory Corex - Primary Forelimn) regions show relatively high spike counts. It is noted that there is significant variability in the mean number of average spikes across different brain areas.

To be able to visualize the results across different sessions and trials, a function has been created. In this instance, two bar plots have been generated from session 3, one utilizing trial 10 and one using trial 15. These plots provide a simple representation of the average spike counts for each brain area that make it easy to compare the two trials. he data table, Table 2.0, demonstrates that in trial 10 of session 3, The highest amount of spikes per area are found in the MRN part of the mouse brain. This is followed by the DG and SPF region.

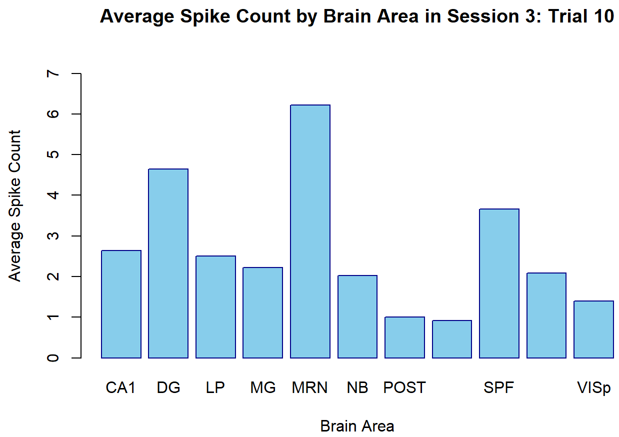

Figure 1.0: Neural Activity in Session 3: Trial 10

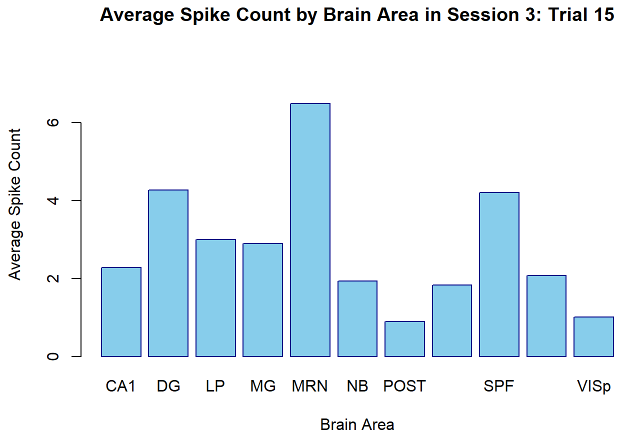

Figure 2.0: Neural Activity in Session 3: Trial 10

Figure 1.0 demonstrates that the highest average spike count across brain areas in session 3: trial 10 is attributed to MRN, DG, and SPF respectively. Figure 2.0 shows that session 3: trial 15 also has MRN, DG, and SPF as the three brain areas with highest average spike count.These results suggest that the unique brain areas consistently display similar levels of average spike counts across different trials within the same session. Potential similarities in spike count patterns among these brain areas will be further explored.

Exploring Changes in Neural Activities Across Trials

To explore the changes in neural activity across trials in Session 3, a data frame containing all trials is generated. This data frame uses relevant variables in order to illustrate the neural activity. The visualizations created from the data will create insights into variations and trends within the neural responses across all trials in session 3.

First 5 Rows of Session 3 Data Frame

CA1	DG	LP	MG	MRN	NB	POST	root	SPF	VISam	VISp	feedback	left contr.
2.738095	4.058824	4.50	2.722628	5.634146	1.674419	1.1111111	1.166667	4.600000	2.061404	1.324561	1	0.50
2.428571	4.441176	5.75	2.715328	6.731707	1.720930	0.9206349	1.583333	4.533333	2.578947	1.070175	1	1.00
2.142857	3.735294	2.75	2.744526	6.024390	3.279070	1.5714286	1.750000	4.600000	2.877193	1.754386	1	0.00
2.095238	3.235294	5.00	2.751825	6.000000	2.046512	1.3650794	1.166667	5.400000	2.684211	1.166667	1	0.00
2.190476	3.176471	3.75	2.635036	5.634146	2.465116	1.3174603	1.333333	4.800000	2.350877	1.254386	1	0.25

Table 3.0: Neural Activity in Brain Regions Across sessions 3

Table 3.0 shows the first five rows of the session 3 data frame that contains the average spike counts for each area, feedback type, the two contrasts, and the trial id. This data frame contains all information necessary for analyzing the nerural activity across all trials within session 3. is utilized in order to create graph the average spike counts against all trials in session 3.

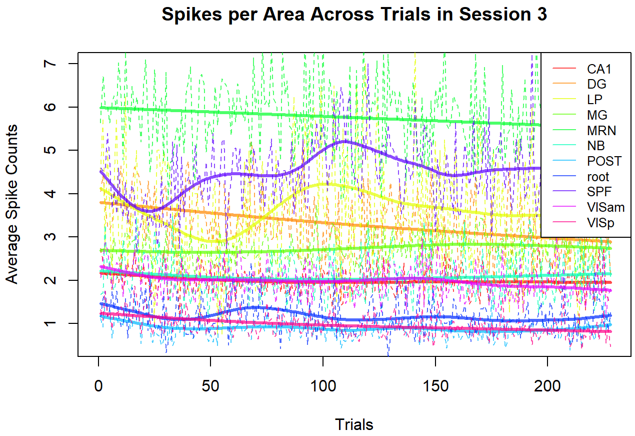

Figure 3.0: Neural Activity Across Trials in Session 3 (Cori)

Figure 3.0 demonstrates the spikes per brain area in session 3 (Cori) across all trials. It is clear to see that the MRN (Midbrain Reticular Nucleus) part of the brain exhibits the most brain activity, followed by SPF (Somatosensory Cortex - Primary Forelimb), and LP (Lateral Posterior Nucleus). The chart supports the prior conclusion made that unique brain areas have similar amounts of average spike counts across trials. The chart further reveals that the order of highest to lowest average spike count areas remains fairly consistent across all trials.

Exploring Homogeneity and Heterogeneity Across Mice and Sessions

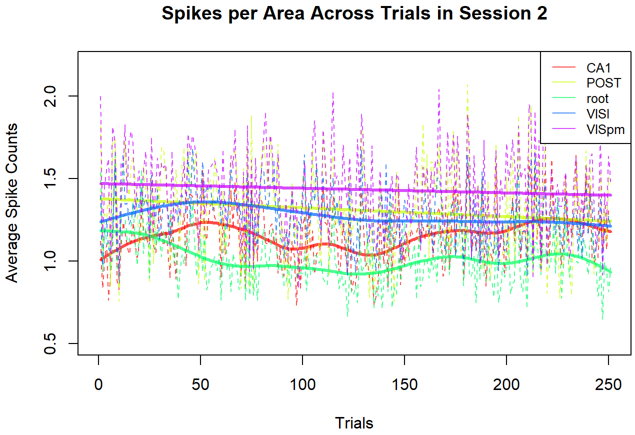

Figure 4: Neural Activity Across Trials in Session 2 (Cori)

Figure 4.0 demonstrates the spikes per area in session 2 (Cori) across all trials and brain areas. The VISpm (primary Motor Cortex - Vibrissal Region) portion of the brain has the greatest average spike count followed by POST (Posterior Association Cortex) and VISl (Primary Visual Cortex - Inferior Region). There are three brain areas shared between session 2 and 3 which are CA1, root, and PPOST. The spike counts very heavily in difference between session 2 and 3, and it can thus be concluded that heterogeneity is likely due to differences in neurons measured in each session. It is possible that each mouse may have unique neural characteristic and responses, resulting in distinct patterns of activity across brain regions. However, the presence of the shared brain areas indicates some level of homogeneity between the mice regarding the regions.

Data Integration

Given that heterogeneity is due to differences in neurons measured in each session, it is essential to find the average number of activated neurons. Calculating this average allows for a better understanding of the overall neural activation patterns in this mice. To acheive this, the total number of activated neurons across all brain areas and trials within each session are summed and then divided by the number of sessions.

First 5 Rows of Average Spike Count Across all Sessions

Session	Trial	AverageSpikeCount
1	1	1.581744

Table 4.0: Neural Activity Across Sessions 3

First 5 Rows of Average Spike Count Across all Sessions

Session	Trial	AverageSpikeCount
1	2	1.311989
1	3	1.844687
1	4	1.381471
1	5	1.425068

Table 4.0: Neural Activity Across Sessions 3

Table 4.0 showcases the first five trials of the average spike counts across neurons. These values will be appended to a larger data frame that will be used to create a predictive model.

Creating A Data Frame

A new data frame will be built in order to incorporate multiple predictors that are relevant for the predictive model. These predictors include the left_contrast, right_contrast, session, total_spikes (representing the total spikes per trial), total_neurons (indicating the total number of neurons per trial), total_areas (representing the number of unique brain areas per trial), and reward (a categorical variable indicating the feedback type). The inclusion of the left/right contrast allows for the contrast level presented to mice to be captured. The AverageSPikeCount represnets the average spike count across neurons, and provides details about the neural activity in each trial. The “mouse” variable assigns a number to different mice (1 for Cori, 2 for Forssmann, 3 for Hench, and 4 for Lederberg), which allows the model to account for variation between mice. This data frame will help understand the relationship between predictors and feedback types and will ultimately lead to the final goal of the project.

Session	Trial	AverageSpikeCount	feedback	left_contr	right_contr	total_spikes	total_neurons	total_areas	mouse	reward
1	1	1.581744	1	0.0	0.5	1161	734	8	1	3
1	2	1.311989	1	0.0	0.0	963	734	8	1	0
1	3	1.844687	-1	0.5	1.0	1354	734	8	1	3
1	4	1.381471	-1	0.0	0.0	1014	734	8	1	0
1	5	1.425068	-1	0.0	0.0	1046	734	8	1	0

Table 5.0: First Five Rows of Data Frame to Build Predictive Model

Predictive Modeling:

The data frame is broken into two sets: a test set comprising of approximately 20 percent of the data and a training set containing the remaining 80 percent. The randomization ensures that the model’s performance is more accurate to unseen data. For this analysis, a logistic regression is utilized. It is chosen due to its capacity of handling both categorical and continuous predictor variables and encapsulate their combined effects. Unlike other models, the logistic regression does not assume linearity or homoscedasticity. Through this approach, a better understanding of how the various predictor variables contribute to predicting the feedback type.

```
##                   Actual Feedback
## Predicted Feedback  -1   1
##                 -1  22  16
##                 1  268 694
```

Misclassification.Error.Rate	Overall.Accuracy
28.4	71.6

Table 6.0: Results from Self-Created Test Set

Following the results from table 5.0, it is shown that the model achieves an overall accuracy of 71.6% when predicting the feedback type. Although this indicates that the model performs better than random change, there is room for improvement. It is shown that the model tends to favor predicting a feedback type of 1 over -1 in most instances. This suggests a bias towards selecting positive feedback which could effect the overall performance. Although changes can be made in future forms of this model, it can still be concluded that the current model is useful in determining the feedback from the trials. Analyzing the coefficients of the predictor variables the average spike count, mouse, left contrast, and reward, had the greatest significance when determining the likelihood of the feedback. These variables provide great insights beyond the original scope of the study.

Prediction Performance on the Test Sets.

The predictive model will now be applied on the two test sets given from session 1 and session 18. Two more models utilizing only training data from session 1 and 18 will be used to compare against the predictive model that uses all sessions. It is important to note that these session-specific models will have fewer predictors due to some variables being constant within a session.

```
##                   Actual Feedback
## Predicted Feedback  -1   1
##                 -1   0   1
##                 1   55 144
```

Misclassification.Error.Rate	Overall.Accuracy
28	72

Table 7.0: Test Results on Session 1-18

Table 7.0 presents the overall accuracy on data trained from all session to be 72%. The model again exhibits a consistent tendency to predict a feedback type of 1 in almost every instance. Although future iterations of the model can be made to address this issue, it is evident that the current model holds value in determining feedback from trials. To further explore the performance of the model, two separate models will be trained solely on data from session 1 and session 18. These are the sessions in which the test data were taken.

```
## [1] "Session 1 Confusion Matrix"
```

```
##                   Actual Feedback
## Predicted Feedback -1  1
##                 -1 39 76
##                 1  16 69
## [1] "Session 18 Confusion Matrix"
```

```
##                    Actual Feedback
## Predicted Feedback  -1   1
##                  -1   0   2
##                  1   55 143
```

Misclassification.Error.Rate	Overall.Accuracy
46	54

Table 7.1: Test Results on Session 1 Results

Misclassification.Error.Rate	Overall.Accuracy
28.5	71.5

Table 7.2: Test Results on Session 18 Results

Table 7.1 demonstrates a very low accuracy for the model trained on session 1, whereas table 7.2 demonstrates a much higher accuracy of 71.5% for the model trained on session 18. The accuracy of the model that uses all data surpasses those of the models trained solely on the corresponding session’s data. This observation suggests the presence of homogeneity across all sessions, in which combining data from all sessions enhances the accuracy of the feedback prediction.

Discussion and Conclusion:

In order to improve the accuracy of the model, future research could focus on testing the same brain areas across all sessions. By keeping the brain areas consistent, researches can eliminate the confounding effects of different brain regions on the outcome. They could also determine if different regions of the brain tend to serve different functions regarding selection. It is important to consider not only the brain areas themselves but also the number of spikes recorded within those areas. Time data was not utilized in the current model. Studies in the future could further investigate the relationship between the number of spikes, time, and the feedback type which could uncover insights into the neural correlates of the visual choice behavior. By focusing on both specific brain areas and quantitative aspects of neural activity, researchers can gain a better understanding of the underlying mechanism that drive feedback response. In conclusion, future research should not only consider the number of spikes and brain areas but also explore a wider scope of variables such as time-related data.

Reference

Steinmetz, N.A., Zatka-Haas, P., Carandini, M. et al. Distributed coding of choice, action and engagement across the mouse brain. Nature 576, 266–273 (2019). https://doi.org/10.1038/s41586-019-1787-x (https://doi.org/10.1038/s41586-019-1787-x)

chat.openai.com was used in order to help build data frames and models.

Appendix

```
knitr::opts_chunk$set(warning = FALSE)
suppressWarnings(library(knitr))
suppressMessages(suppressWarnings(library(tidyverse)))
suppressWarnings(library(dplyr))
suppressWarnings(library(gt))

#Loads Given Data
n.session = 18
session=list()
for(i in 1:18){
  session[[i]]=readRDS(paste('C:/Users/nickb/Downloads/STA141Project/session',i,'.rds',sep=''))
}

#creates a tibble using the tidyverse library
meta <- tibble(
  mouse_name = rep('name',n.session),
  n_brain_area = rep(0,n.session),
  n_neurons = rep(0,n.session),
  n_trials = rep(0,n.session),
  success_rate = rep(0,n.session)
)


for(i in 1:n.session){
  tmp = session[[i]];
  meta[i,1]=tmp$mouse_name; #name of mice
  meta[i,2]=length(unique(tmp$brain_area)); #Number unique brain areas
  meta[i,3]=dim(tmp$spks[[1]])[1]; #number of neurons
  meta[i,4]=length(tmp$feedback_type); #number of trials
  meta[i,5]=mean(tmp$feedback_type+1)/2; #calculated success rate

}



# Create the table using gt
table_gt <- gt(meta) %>%
  tab_style(style = list(cell_text(weight = "bold")), locations = cells_column_labels()) %>%
  tab_options(table.width = "auto") %>%
  tab_footnote("Table 1.0:  Data structure across sessions.")

# Print the table
table_gt

i.s=3

i.t=10 # indicator for this trial

spk.trial = session[[i.s]]$spks[[i.t]]
area=session[[i.s]]$brain_area

# We need to first calculate the number of spikes for each neuron during this trial
spk.count=apply(spk.trial,1,sum)

for(i in 1:dim(spk.trial)[1]){
  spk.count[i]=sum(spk.trial[i,])
  }

# Next we take the average of spikes across neurons that live in the same area
spk.average.tapply=tapply(spk.count, area, mean)


# Create data frame
tmp <- data.frame(
  area = area,
  spikes = spk.count
)

# Calculate the average by group using dplyr
spk.average.dplyr <- tmp %>%
  group_by(area) %>%
  summarize(mean = mean(spikes))

# Set the column names
colnames(spk.average.dplyr) <- c("Unique Brain Area", "Mean")

# Set the title
title <- "Average Spikes Across Neurons of Unique Brain Areas in Session 3: Trial 10"

# Create the gt table
table_gt <- gt(spk.average.dplyr) %>%
  tab_header(title = title) %>%
  tab_style(
    style = list(
      cell_text(weight = "bold"),
      cell_fill(color = "lightgray")
    ),
    locations = cells_column_labels(everything())
  ) %>%
  fmt_number(
    columns = c(Mean),
    decimals = 2
  ) %>%
  tab_footnote(
    footnote = "Table 2.0 Neural Activity in Session 3: Trial 10"
    )

# Print the table
table_gt

i.s=3 #session indicator
i.t=10 # trial indicator

average_spike_area<-function(i.t,this_session){
  spk.trial = this_session$spks[[i.t]]
  area= this_session$brain_area
  spk.count=apply(spk.trial,1,sum)
  spk.average.tapply=tapply(spk.count, area, mean)
  return(spk.average.tapply)
  }

# Call the average_spike_area function
spk.average.tapply <- average_spike_area(i.t, session[[i.s]])

# Create a bar chart using barplot
barplot(spk.average.tapply,
        main = "Average Spike Count by Brain Area in Session 3: Trial 10",
        xlab = "Brain Area",
        ylab = "Average Spike Count",
        col = "skyblue",            # Set the color of the bars
        border = "darkblue",        # Set the border color of the bars
        ylim = c(0, max(spk.average.tapply) * 1.2)  # Adjust the y-axis limit for padding
)


i.s=3 #session indicator
i.t=15 # trial indicator

mtext(text = "Figure 1.0: Neural Activity in Session 3: Trial 10", side = 1, line = 4, at = 2)

average_spike_area<-function(i.t,this_session){
  spk.trial = this_session$spks[[i.t]]
  area= this_session$brain_area
  spk.count=apply(spk.trial,1,sum)
  spk.average.tapply=tapply(spk.count, area, mean)
  return(spk.average.tapply)
  }

# Call the average_spike_area function
spk.average.tapply <- average_spike_area(i.t, session[[i.s]])

# Create a bar chart using barplot
barplot(spk.average.tapply,
        main = "Average Spike Count by Brain Area in Session 3: Trial 15",
        xlab = "Brain Area",
        ylab = "Average Spike Count",
        col = "skyblue",            # Set the color of the bars
        border = "darkblue",        # Set the border color of the bars
        ylim = c(0, max(spk.average.tapply) * 1.2)  # Adjust the y-axis limit for padding
)
mtext(text = "Figure 2.0: Neural Activity in Session 3: Trial 10", side = 1, line = 4, at = 2)
i.s=3
n.trial=length(session[[i.s]]$feedback_type)
n.area=length(unique(session[[i.s]]$brain_area ))

# We will create a data frame that contain the average spike counts for each area, feedback type,  the two contrasts, and th
e trial id

trial.summaryThree =matrix(nrow=n.trial,ncol= n.area+1+2+1)
for(i.t in 1:n.trial){
  trial.summaryThree[i.t,]=c(average_spike_area(i.t,this_session = session[[i.s]]),
                          session[[i.s]]$feedback_type[i.t],
                        session[[i.s]]$contrast_left[i.t],
                        session[[i.s]]$contrast_right[i.s],
                        i.t)
}

colnames(trial.summaryThree)=c(names(average_spike_area(i.t,this_session = session[[i.s]])), 'feedback', 'left contr.','righ
t contr.','id' )

# Turning it into a data frame
trial.summaryThree <- as_tibble(trial.summaryThree)

first_5_entries <- head(trial.summaryThree, n = 5)

table3_gt <- gt(first_5_entries) %>%
  tab_style(style = list(cell_text(weight = "bold")), locations = cells_column_labels()) %>%
    tab_header(title = "First 5 Rows of Session 3 Data Frame") %>%
  tab_options(table.width = "auto") %>%
  tab_footnote("Table 3.0:  Neural Activity in Brain Regions Across sessions 3")
table3_gt
i.s = 3
  area.col=rainbow(n=n.area,alpha=0.7)
plot(x=1,y=0, col='white',xlim=c(0,n.trial),ylim=c(0.5,7), xlab="Trials",ylab="Average Spike Counts", main=paste("Spikes per
Area Across Trials in Session", i.s))


for(i in 1:n.area){
  lines(y=trial.summaryThree[[i]],x=trial.summaryThree$id,col=area.col[i],lty=2,lwd=1)
  lines(smooth.spline(trial.summaryThree$id, trial.summaryThree[[i]]),col=area.col[i],lwd=3)
  }
legend("topright",
  legend = colnames(trial.summaryThree)[1:n.area],
  col = area.col,
  lty = 1,
  cex = 0.8
)
mtext(text = "Figure 3.0: Neural Activity Across Trials in Session 3 (Cori)", side = 1, line = 4, at = 50)

i.s=2
n.trial=length(session[[i.s]]$feedback_type)
n.area=length(unique(session[[i.s]]$brain_area ))

# We will create a data frame that contain the average spike counts for each area, feedback type,  the two contrasts, and th
e trial id

trial.summaryTwo =matrix(nrow=n.trial,ncol= n.area+1+2+1)
for(i.t in 1:n.trial){
  trial.summaryTwo[i.t,]=c(average_spike_area(i.t,this_session = session[[i.s]]),
                          session[[i.s]]$feedback_type[i.t],
                        session[[i.s]]$contrast_left[i.t],
                        session[[i.s]]$contrast_right[i.s],
                        i.t)
}

colnames(trial.summaryTwo)=c(names(average_spike_area(i.t,this_session = session[[i.s]])), 'feedback', 'left contr.','right
contr.','id' )

# Turning it into a data frame
trial.summaryTwo <- as_tibble(trial.summaryTwo)

i.s = 2
  area.col=rainbow(n=n.area,alpha=0.7)
plot(x=1,y=0, col='white',xlim=c(0,n.trial),ylim=c(0.5,2.2), xlab="Trials",ylab="Average Spike Counts", main=paste("Spikes p
er Area Across Trials in Session", i.s))


for(i in 1:n.area){
  lines(y=trial.summaryTwo[[i]],x=trial.summaryTwo$id,col=area.col[i],lty=2,lwd=1)
  lines(smooth.spline(trial.summaryTwo$id, trial.summaryTwo[[i]]),col=area.col[i],lwd=3)
  }
legend("topright",
  legend = colnames(trial.summaryTwo)[1:n.area],
  col = area.col,
  lty = 1,
  cex = 0.8
)
mtext(text = "Figure 4: Neural Activity Across Trials in Session 2 (Cori)", side = 1, line = 4, at = 50)

result_df <- data.frame(Session = integer(), Trial = integer(), AverageSpikeCount = numeric(), stringsAsFactors = FALSE)

# Iterate over each session
for (i.s in 1:length(session)) {
  # Get the number of trials in the current session
  n_trials <- length(session[[i.s]]$feedback_type)

  # Iterate over each trial
  for (i.t in 1:n_trials) {
    # Extract the spike data for the current trial
    spks_trial <- session[[i.s]]$spks[[i.t]]

    # Calculate the total spike count for the trial
    total_spikes <- apply(spks_trial,1,sum)

    # Calculate the average spike count for the trial
    avg_spikes <- mean(total_spikes)

    # Add the results to the data frame
    result_df <- rbind(result_df, data.frame(Session = i.s, Trial = i.t, AverageSpikeCount = avg_spikes, stringsAsFactors =
FALSE))
  }
}

first_5_entries2 <- head(result_df, n = 5)

table4_gt <- gt(first_5_entries2) %>%
  tab_style(style = list(cell_text(weight = "bold")), locations = cells_column_labels()) %>%
    tab_header(title = "First 5 Rows of Average Spike Count Across all Sessions") %>%
  tab_options(table.width = "auto") %>%
  tab_footnote("Table 4.0:  Neural Activity Across Sessions 3")
table4_gt






# Create an empty list to store the individual trial summaries
trialSummaries <- list()

# Iterate over each variation of i.s
for (i.s in 1:18) {
  n.trial <- length(session[[i.s]]$feedback_type)
  n.area <- length(unique(session[[i.s]]$brain_area))

  # Create a matrix to store the trial summary for the current i.s
  trialSummary <- matrix(nrow = n.trial, ncol = 5)
  for (i.t in 1:n.trial) {
    spks_trial <- session[[i.s]]$spks[[i.t]]
    total_spikes <- sum(spks_trial)
    dim_spks_trial <- dim(spks_trial)
    total_neurons <- dim_spks_trial[1]
    trialSummary[i.t,] <- c(session[[i.s]]$feedback_type[i.t],
                            session[[i.s]]$contrast_left[i.t],
                            session[[i.s]]$contrast_right[i.t],
                            total_spikes,
                            total_neurons)
  }

  # Set column names for the trial summary matrix
  colnames(trialSummary) <- c('feedback', 'left_contr', 'right_contr', 'total_spikes', 'total_neurons')

  # Convert the trial summary matrix to a data frame
  trialSummary <- as.data.frame(trialSummary)

  # Add the total number of neurons and brain areas as new columns
  trialSummary <- mutate(trialSummary, total_areas = n.area)

  # Add the mouse column based on session ranges
  mouse <- ifelse(i.s %in% 1:3, 1,
                 ifelse(i.s %in% 4:7, 2,
                        ifelse(i.s %in% 8:11, 3,
                               ifelse(i.s %in% 12:18, 4, NA))))
  trialSummary <- mutate(trialSummary, mouse = mouse)

  # Add the reward column based on the conditions
  reward <- ifelse(trialSummary$left_contr == 0 & trialSummary$right_contr == 0, 0,
                   ifelse(trialSummary$left_contr == trialSummary$right_contr & trialSummary$left_contr != 0, 1,
                          ifelse(trialSummary$left_contr != trialSummary$right_contr, 3, NA)))
  trialSummary <- mutate(trialSummary, reward = reward)

  # Add the trial summary to the list
  trialSummaries[[i.s]] <- trialSummary
}

# Combine all trial summaries into a single dataframe
combined_trials <- bind_rows(trialSummaries)

# Merge the final data frames
final_df <- bind_cols(result_df, combined_trials)

# Get the first five rows of final_df
first_five_rows <- head(final_df, 5)

# Create the gt table
table_gt5 <- gt(first_five_rows) %>%
  tab_options(table.width = "auto") %>%
  tab_footnote("Table 5.0: First Five Rows of Data Frame to Build Predictive Model")

# Print the table
table_gt5
# Set the seed for reproducibility
set.seed(124)

# Create training and testing sets
train_indices <- sample(1:nrow(final_df), size = 1000)
testData <- final_df[train_indices, ]
trainData <- final_df[-train_indices, ]

# Train the logistic regression model
logit_model <- glm(as.factor(feedback) ~ right_contr*left_contr + AverageSpikeCount + left_contr + right_contr + Session + t
otal_spikes + total_neurons + total_areas + mouse + reward, data = trainData, family = "binomial")
#print(logit_model)
# Make predictions on the test data
logit_pred <- predict(logit_model, newdata = testData, type = "response")
logit_pred <- ifelse(logit_pred > 0.5, 1, -1)

# Create the confusion matrix
logit_conf <- table(logit_pred, testData$feedback, dnn = c('Predicted Feedback', 'Actual Feedback'))
print(logit_conf)

# Calculate the misclassification rate
misclassification_rate <- (sum(logit_pred != testData$feedback) / length(testData$feedback)) * 100

# Calculate the overall accuracy
accuracy <- 100 * (sum(logit_pred == testData$feedback) / length(testData$feedback))


results <- data.frame(
  "Misclassification Error Rate" = misclassification_rate,
  "Overall Accuracy" = accuracy
)

gt(results) %>%
  tab_footnote("Table 6.0: Results from Self-Created Test Set")
n.test <- 2
test <- list()

for (i in 1:n.test) {
  test[[i]] <- readRDS(paste('C:/Users/nickb/Downloads/STA141Project/test', i, '.rds', sep=''))
}

trialSummaries <- list()
result_df <- data.frame(Session = integer(), Trial = integer(), AverageSpikeCount = numeric(), stringsAsFactors = FALSE)

# Function to calculate average spike count
calculateAverageSpikeCount <- function(spks_trial) {
  total_spikes <- apply(spks_trial, 1, sum)
  avg_spikes <- mean(total_spikes)
  return(avg_spikes)
}

# Iterate over each test dataset
for (i.s in 1:n.test) {
  n.trial <- length(test[[i.s]]$feedback_type)
  n.area <- length(unique(test[[i.s]]$brain_area))

  # Create a matrix to store the trial summary for the current test dataset
  trialSummary <- matrix(nrow = n.trial, ncol = 5)

  for (i.t in 1:n.trial) {
    spks_trial <- test[[i.s]]$spks[[i.t]]
    total_spikes <- sum(spks_trial)
    dim_spks_trial <- dim(spks_trial)
    total_neurons <- dim_spks_trial[1]
    trialSummary[i.t,] <- c(test[[i.s]]$feedback_type[i.t],
                            test[[i.s]]$contrast_left[i.t],
                            test[[i.s]]$contrast_right[i.t],
                            total_spikes,
                            total_neurons)

    # Calculate the average spike count for the trial
    avg_spikes <- calculateAverageSpikeCount(spks_trial)

    # Add the average spike count to the result dataframe
    result_df <- rbind(result_df, data.frame(Session = i.s, Trial = i.t, AverageSpikeCount = avg_spikes, stringsAsFactors =
FALSE))
  }

  # Set column names for the trial summary matrix
  colnames(trialSummary) <- c('feedback', 'left_contr', 'right_contr', 'total_spikes', 'total_neurons')
  # Convert the trial summary matrix to a data frame
  trialSummary <- as.data.frame(trialSummary)

  # Add the total number of neurons and brain areas as new columns
  trialSummary <- mutate(trialSummary, total_areas = n.area)

  # Add the mouse column based on session ranges
  mouse <- ifelse(i.s == 1, "1", "4")
  trialSummary <- mutate(trialSummary, mouse = mouse)

  # Add the reward column based on the conditions
  reward <- ifelse(trialSummary$left_contr == 0 & trialSummary$right_contr == 0, 0,
                   ifelse(trialSummary$left_contr == trialSummary$right_contr & trialSummary$left_contr != 0, 1,
                          ifelse(trialSummary$left_contr != trialSummary$right_contr, 3, NA)))
  trialSummary <- mutate(trialSummary, reward = reward)

  # Add the trial summary to the list
  trialSummaries[[i.s]] <- trialSummary
}

# Combine all trial summaries into a single dataframe
testfinal_df <- bind_rows(trialSummaries)

# Add the AverageSpikeCount column to the testfinal_df dataframe
testfinal_df <- mutate(testfinal_df, AverageSpikeCount = result_df$AverageSpikeCount)


# Set the seed for reproducibility
set.seed(124)

# Create training and testing sets
testData <- testfinal_df
trainData <- final_df

# Train the logistic regression model
logit_model <- glm(as.factor(feedback) ~ right_contr*left_contr + AverageSpikeCount + left_contr + right_contr + reward, dat
a = trainData, family = "binomial")

# Make predictions on the test data
logit_pred <- predict(logit_model, newdata = testData, type = "response")
logit_pred <- ifelse(logit_pred > 0.5, 1, -1)

# Create the confusion matrix
logit_conf <- table(logit_pred, testData$feedback, dnn = c('Predicted Feedback', 'Actual Feedback'))
print(logit_conf)

# Calculate the misclassification rate
misclassification_rate <- (sum(logit_pred != testData$feedback) / length(testData$feedback)) * 100

# Calculate the overall accuracy
accuracy <- 100 * (sum(logit_pred == testData$feedback) / length(testData$feedback))


results <- data.frame(
  "Misclassification Error Rate" = misclassification_rate,
  "Overall Accuracy" = accuracy
)

gt(results) %>%
  tab_footnote("Table 7.0: Test Results on Session 1-18")

# Create a dataframe for session 1
session1_df <- filter(final_df, Session == 1)

# Create a dataframe for session 18
session18_df <- filter(final_df, Session == 18)

# Train the logistic regression model on session 1 data
logit_model_session1 <- glm(as.factor(feedback) ~ right_contr*left_contr + AverageSpikeCount + left_contr + right_contr + re
ward, data = session1_df, family = "binomial")

# Train the logistic regression model on session 18 data
logit_model_session18 <- glm(as.factor(feedback) ~ right_contr*left_contr + AverageSpikeCount + left_contr + right_contr + r
eward, data = session18_df, family = "binomial")

# Make predictions on the test data for session 1
logit_pred_session1 <- predict(logit_model_session1, newdata = testData, type = "response")
logit_pred_session1 <- ifelse(logit_pred_session1 > 0.5, 1, -1)

# Make predictions on the test data for session 18
logit_pred_session18 <- predict(logit_model_session18, newdata = testData, type = "response")
logit_pred_session18 <- ifelse(logit_pred_session18 > 0.5, 1, -1)

# Create the confusion matrices for session 1 and session 18
logit_conf_session1 <- table(logit_pred_session1, testData$feedback, dnn = c('Predicted Feedback', 'Actual Feedback'))
print("Session 1 Confusion Matrix")
print(logit_conf_session1)
logit_conf_session18 <- table(logit_pred_session18, testData$feedback, dnn = c('Predicted Feedback', 'Actual Feedback'))
print("Session 18 Confusion Matrix")
print(logit_conf_session18)
# Calculate the misclassification rate for session 1 and session 18
misclassification_rate_session1 <- (sum(logit_pred_session1 != testData$feedback) / length(testData$feedback)) * 100
misclassification_rate_session18 <- (sum(logit_pred_session18 != testData$feedback) / length(testData$feedback)) * 100

# Calculate the overall accuracy for session 1 and session 18
accuracy_session1 <- sum(logit_pred_session1 == testData$feedback) / length(testData$feedback)
accuracy_session18 <- sum(logit_pred_session18 == testData$feedback) / length(testData$feedback)


results_session1 <- data.frame(
  "Misclassification Error Rate" = misclassification_rate_session1,
  "Overall Accuracy" = 100 * accuracy_session1
)
```

```
results_session18 <- data.frame(
  "Misclassification Error Rate" = misclassification_rate_session18,
  "Overall Accuracy" = 100 * accuracy_session18
)


results_table_session1 <- gt(results_session1) %>%
  tab_footnote("Table 7.1: Test Results on Session 1 Results")

results_table_session18 <- gt(results_session18) %>%
  tab_footnote("Table 7.2: Test Results on Session 18 Results")

#print("Session 1 Confusion Matrix:", logit_conf_session1)
results_table_session1
#print("Session 18 Confusion Matrix:", logit_conf_session18)
results_table_session18
```