Capstone Paper

Spencer Boyum

July 2019

1 Introduction

In recent years, the Paradiso lab has collected a large amount of neural data for the purpose of understanding the dynamics of eye movements and fixations. The neural data consists of Local Field Potentials (LFPs) from electrodes placed in the macaque – a group of species of Old World monkeys – primary visual cortex (V1). There are lots of useful predictions that can be made from this data; predicting the onset of new eye fixations, predicting future LFPs, predicting directions of eye movements, predicting the timing of neural spikes, and decoding types of eye fixations are among them. Members of the Paradiso lab have performed rigorous analysis of this data using Support Vector Machines (SVM) and autoregressive (AR) methods, but have not employed neural networks to make predictions [3].

I therefore used recurrent neural networks (RNNs) to make some of these predictions – models were built to predict future LFPs, decode types of eye fixation experiments, and predict neural spikes. I conclude – like previous Paradiso lab research – that LFPs in area V1 have valuable predictive power.

1.1 Related and Prior Work

As discussed in the section above, Paradiso et. al. have used LFPs to classify eye fixations and predict the directions of eye movements [3]. Kim et. al. have trained LSTMs on LFPs to predict future LFPs at various lags [2]. Wang et. al. and Ahmadi et. al. have trained LSTMs on LFPs to decode hand kinematics [4][1].

Additionally, Seth Akers-Campbell, of the Paradiso lab, has built an an autoregressive model that uses a markov process to predict the future LFP.

Lastly, Paradiso lab members have recorded, cleaned, and filtered all of the necessary neural data. In particular, the electrode data has been filtered to extract LFPs, which are then aligned into trials centered around the start of new eye fixations. Therefore, very little data cleaning was required on my part.

2 Data

2.1 Description of Data

2.1.1 Experimental Design

The electrode data is recorded from experiments on primates. In each trial, an animal sits in a dimly-lit room facing a visual display with a dark, grey background. Eye fixations are induced by illuminating a red fixation point on the display; this red dot is the only bright area of the visual display, and thus the primate tends to focus its gaze towards the red point once it is visible [3].

Additionally, an infrared eye tracker is used during each trial to determine eye position and velocity.

2.1.2 Datasets

Analysis was done on two datasets, which I will refer to as **D1** and **D2**.

The data from **D1** consists of LFPs from 16 simultaneous electrode recordings of a primate. The electrode data in **D1** is recorded from experiments intended to induce eye movements. At the onset of each trial, the red fixation point is displayed at one of 12 locations. The positions of the illuminated red dots are spaced around a circle with a visual angle of 7 degrees, in 30-degree increments. A trial continues if the initial fixation lasts longer than 400 milliseconds – as determined by examining the eye tracker data – in which case a red point at the center of the circle is illuminated. The trial is considered successful if the center fixation lasts longer than 200 milliseconds.

The electrode data is aligned into 600 ms trials that are centered around the time of eye fixation onset. In other words, the electrode data stored from each trial is from time $t = -299$ to $t = 300$ – in which the units are in milliseconds – and $t = 0$ is the time at which the animal fixates its gaze on the red point at the center of the display window.

Raw electrode data is collected at 30 khz; LFPs are obtained by low-pass filtering the raw electrode data, at 300 hz and downsampling at a rate of 1 khz. Therefore, in its final form, each trial contains 600 16-dimensional data points. There are 743 trials in total, with 61-62 trials in each saccade direction.

The data from **D2** contains, among other fields, LFPs from 15 simultaneous electrode recordings of a primate. We'll refer to some of the experiments as *saccade* trials, and some as *mirror* trials. The general experiment format for the trials in **D2** differs slightly from that of **D1**. The primate, instead of directly facing the visual display as done in trials in **D1**, faces a mirror that has the display in its reflection. The saccade trials are very similar to the trials in **D1** – a red fixation point is shown at one location, and once the initial fixation begins, a red point at a new location is illuminated, inducing a saccadic eye movement. However, unlike the **D1** trials, eye movements are only made in a single direction. Mirror trials begin in the same manner as saccade trials. A red fixation point is shown at a location on the display, and a fixation point at a new location is illuminated after the initial fixation begins. It is indicated to the primate, at the start of each trial, whether the trial is a saccade or mirror experiment; the primates are trained not to make eye movements on mirror trials. Therefore, in a mirror trial, no eye movement is made when the new fixation point is illuminated. Additionally, immediately after the new dot is displayed, the mirror is rotated so that the new dot moves to the same location as the initial fixation point. Moreover, the mirror rotation is calibrated to simulate the primate's expected eye velocity when a saccadic eye movement is made. In other words, the new fixation point moves to the center of the primate's field of view in the same expected time that it takes a primate to make a saccadic eye movement and fixate on the newly illuminated dot. Mirror and saccade trials are geometrically the same with respect to the primate's eye position – the difference is that the primate makes an eye movement in a saccade trial but not in a mirror trial. Thus, the purpose of comparing LFPs from mirror and saccade trials is to examine whether the brain processes the act of *making* an eye movement to track an object in a different manner than the act of simply fixating on an object that moves into one's center of view.

The electrode data is aligned into 2000 ms trials, which, like the trials in **D1**, are centered around the time of new eye fixation onset. The electrode data from each trial is from time $t = -999$ to $t = 1000$, in which the time units are milliseconds and $t = 0$ is the time of eye fixation onset.

LFPs are created from the raw electrode data by applying a low-pass filter at 300 hz; unlike the data in **D1**, the LFPs in **D2** were not downsampled, and thus each trial contains LFPs sampled at 30 khz. Additionally, in each trial, the times of neuronal spikes – when a neuron fires an action potential – are recorded for 52 neurons from $t = -499$ to $t = 500$.

For both **D1** and **D2**, all of the work of collecting and curating electrode data was performed by other Paradiso lab members prior to the start of my work – both datasets were received in the form described above.

Both of these datasets were stored as Matlab (.mat) objects.

2.2 Data Cleaning and Preparation

The data from **D1** has already been low-pass filtered and downsampled – therefore, no cleaning was required.

Our analysis was performed in Python, and thus it was necessary to transform the .mat object into a .csv file. The original .mat file consists of a 12-dimensional cell array; each cell contains the trials for a particular saccade direction. The trials for each saccade direction are represented by a (num_trials$\times$timesteps_per_trial$\times$ dimensions) array. In this case num_trials = 61 or 62, timesteps_per_trial = 600, and dimensions = 16. To convert to .csv format, all of the trial data was concatenated to form a 445800×16 array such that each row in the array represents a set of LFP voltage readings at a single timestep. Two columns denoting the saccade direction and trial number were added to this array, which was then converted into a .csv file. I will refer to this .csv file as **C1**.

D2 had been low-pass filtered but not downsampled – the LFPs, in their initial form, were sampled at a rate of 30 khz. Matlab's built-in `downsample` function was used to downsample the LFPs to 1 khz.

D2 was transformed from a .mat object to a .csv file. The original .mat file consists of an 834-dimensional struct; each entry contains information for a single trial that includes, among other fields, LFP voltage, spike times, and a binary indication of whether the trial is a saccade or mirror experiment. The LFP voltage for each trial is a (timesteps_per_trial $\times$ dimensions) matrix – in this case, timesteps_per_trial = 2000 and dimensions = 15. Two .csv files, which we'll refer to as **C2** and **C3** were created.

C2 contains LFP voltage and a mirror/saccade trial indicator. To produce **C2**, the LFPs were concatenated to form a $1,668,000 \times 15$ matrix. Two columns – one denoting the trial number for each voltage reading, and one indicating, in binary, whether the trial is a saccade or mirror experiment – were added to this array, which was then converted into a .csv file.

C3 contains LFP voltage, spike time information, and a mirror/saccade trial indicator. Spike times are recorded in list format for each of the 52 neurons. Although each trial consists of 2000 ms of LFP voltage readings, spike times are only recorded for 1000 ms of the trial, centered around eye fixation onset at $t = 0$. Thus spike times are not recorded for the first 500 ms and last 500 ms of a trial. LFP voltage was discarded for the first 500 ms and last 500 ms of a trial – thus, the LFP voltage for each trial was represented by a 1000×15 matrix, centered around $t = 0$. The LFPs were concatenated to form a $834,000 \times 15$ matrix. The spike time arrays for each trial were converted to a 1000×52 matrix, in which each entry is a binary variable. If entry (i, j) is equal to 1, this indicates that neuron j spiked between timesteps i and $i+1$ in the trial. In other words, we use 1 ms bins for spike counts. It's worth noting that there were no cases in which a neuron spiked more than once in a 1 ms period; therefore, using binary variables to represent spike counts is fully descriptive. The spike matrices were concatenated to form a $834,000 \times 52$ matrix. The spike matrix was then merged with the LFP matrix, columns were added to indicate the trial number and whether the trial is a saccade or mirror experiment, and the matrix was converted to a .csv file.

The rationale for separating **D1** into two .csv files is that spike counts were not recorded for the first 500 ms and last 500 ms of each trial. The voltage information at the start and end of these trials could be useful for decoding saccade and mirror trials.

2.3 EDA

2.3.1 LFP Voltage

We first examine the LFP voltage data in **D1**. An example of a trial is shown in figure 1. Two intuitive observations can be made about this trial; the LFP voltage of the electrode channels appear to be highly correlated, and LFP voltage tends to rapdily increase after $t = 0$, which is the time of eye fixation onset. Figure 2, which displays the average LFP across trials at each timestep, shows that this intuition is correct – there is a voltage spike soon after $t = 0$, as well as a voltage spike at around $t = 75$. Figure 3 shows that the intuition that electrode channels are highly correlated is also correct – in fact, the *lowest* correlation between any two electrode channels is $r = .43$. This, however, should not be a major problem. The primary purpose

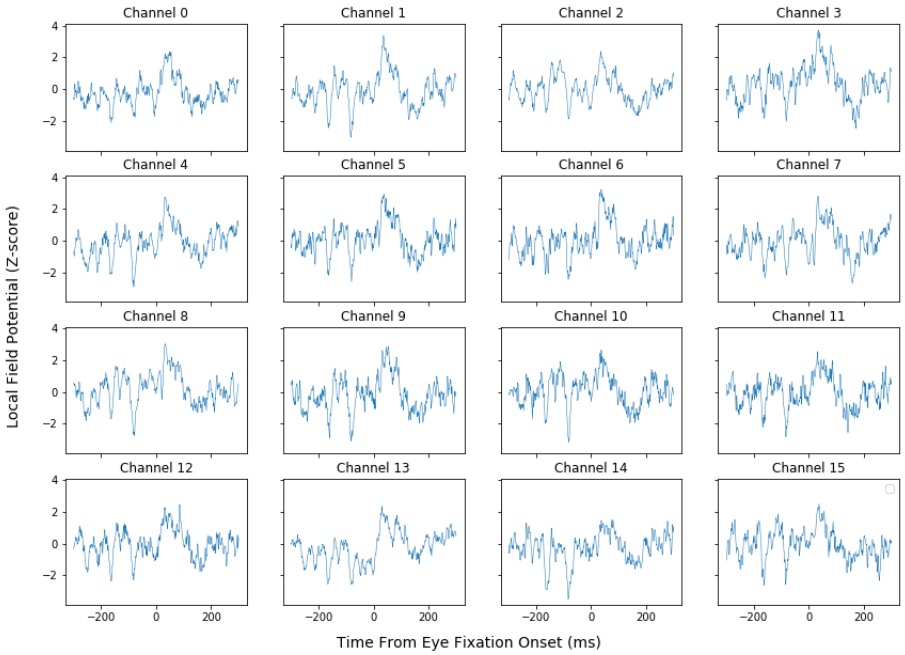

Figure 1: LFP voltage for a single trial

Trial Type	Number of Trials
Mirror	435
Saccade	399

Table 1: Trial counts by type

of the LFP model is prediction rather than explanation, and highly correlated covariates do not necessarily pose a problem from a pure prediction perspective.

I also investigated the LFP dynamics by eye direction. Figure 4 displays the average LFP voltage at each time point in a trial for electrode channel 0 – it is quite apparent that eye direction significantly affects LFP dynamics. Paradiso et. al performed a permutation test to statistically assess this hypothesis; the variations in LFP shape were found to be statistically significant for each pair of saccade directions [3]. This suggests that models will achieve better performance if they are separately trained for each saccade direction or the saccade direction is encoded as input.

2.3.2 Saccade and Mirror Trials

I next examined the data in **D2** in the context of decoding whether a trial is a mirror or saccade experiment. Table 1 shows the trial counts by type of experiment – the two classes are fairly balanced, which is helpful for prediction.

I also examined the differences in LFP shapes between mirror and saccade trials. Figure 5 displays the average LFP shapes of mirror and saccade trials for electrode channel 0. It is apparent that there are significant differences in shapes between the two trial types; the distinctions are especially apparent immediately after $t = 0$, when a new eye fixation begins.

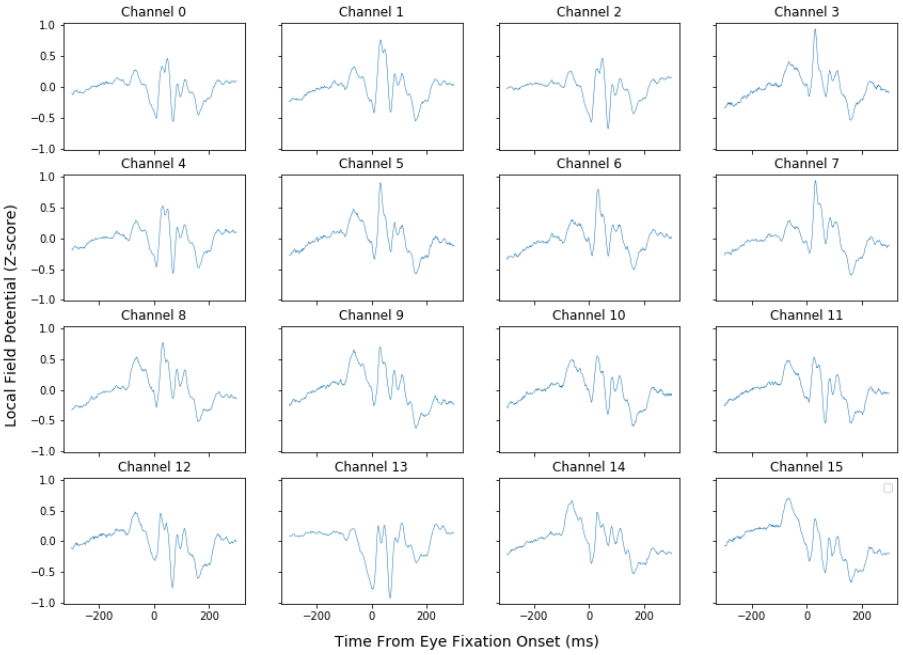

Figure 2: Mean LFP voltage across trials

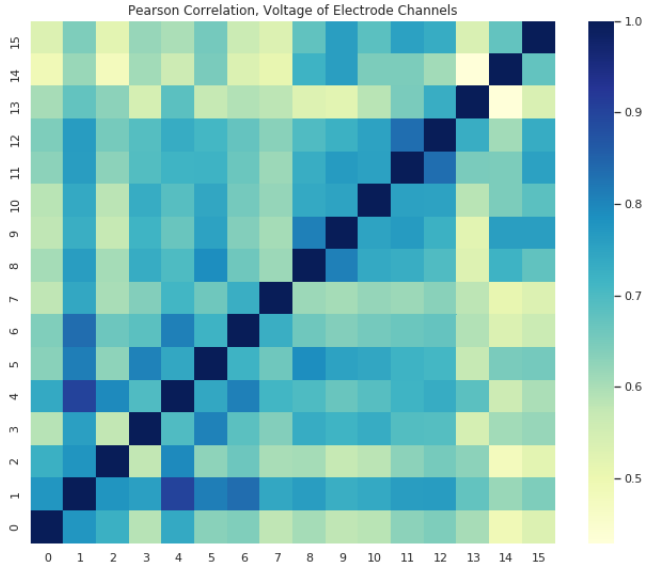

Figure 3: Pearson correlation between electrode channels

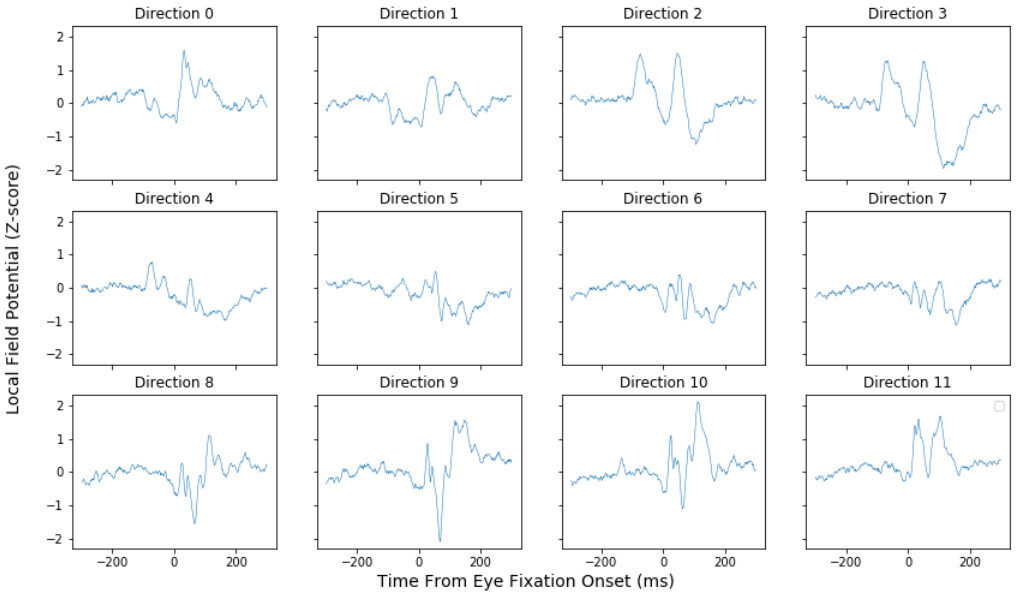

Figure 4: Mean LFP voltage by eye direction for a single electrode channel

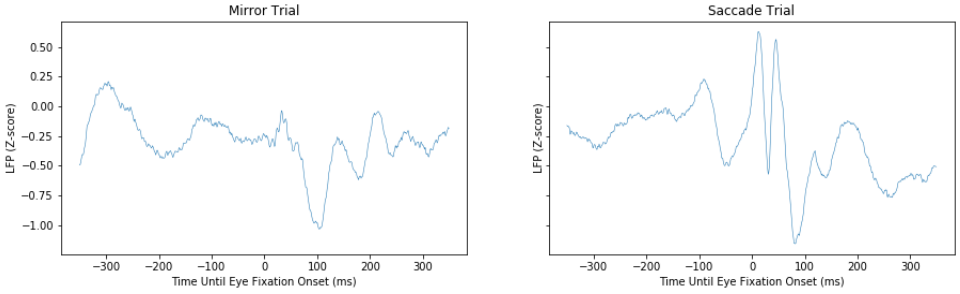

Figure 5: Mean LFP of mirror and saccade trials for a single electrode channel

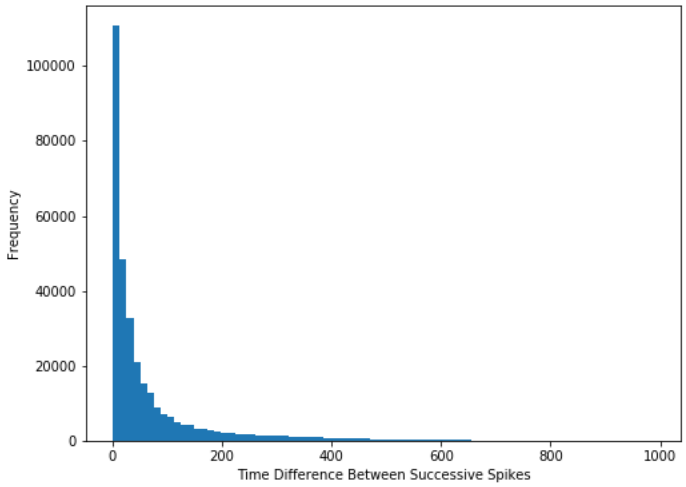

Figure 6: Histogram of time differences between successive spikes

2.3.3 Neuronal Spikes

We lastly examine the neuronal spike data in **D2**. Figure 6 shows the distribution of time differences between successive spikes. The distribution indicates that the at a point in a trial, the probability of a neuron spiking in the near future increases as the time since the last spike grows. This has implications for performance expectations of spike prediction models – a prediction model that, in addition to LFP voltage, takes in past spike times as input, would likely perform better than a model that exclusively uses LFP voltage as input.

Figure 7 shows the probability of a spike – for a 1 ms time window – at each point in a trial. The spike probability increases slightly after the new eye fixation onset, and drastically increases around $t = 400$.

3 Models

3.1 LFP Prediction

Models were built that predict the future LFP, for all channels, based on past LFP voltage readings. LFPs were predicted for many different lags – models were built that predict LFP voltages 1, 5, 10, 25, 50, and 100 milliseconds forward.

3.1.1 Data Preparation

The data from **C1**, which was generated from **D1**, was used to train, test, and evaluate these models.

To avoid overfitting, the trials were split into training, testing, and validation sets using a .64/.2/.16 train/test/validation split. The split was chosen randomly – however, a random seed was set to ensure consistency. It is important to note that trials were never broken up; the dataset was split based on trial indeces rather than single LFP readings. This is especially useful for evaluation; it is critical to have the ability to visualize performance across an entire trial.

The mean and standard deviation of LFP voltages were calculated across the training trials for each LFP channel. Each trial was standardized by subtracting the mean voltage from the LFP and dividing by the

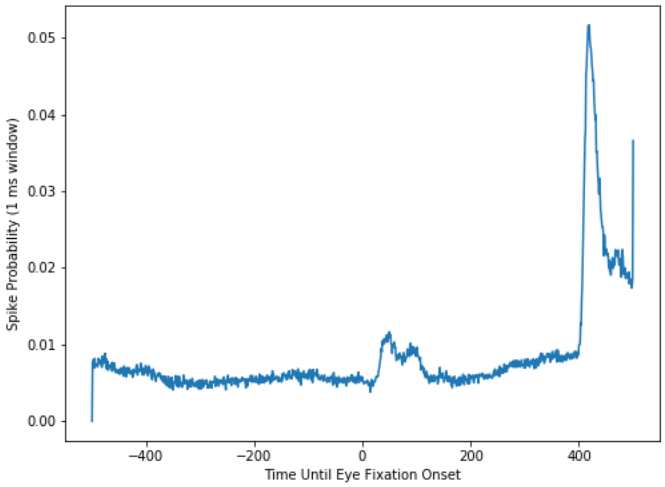

Figure 7: Spike probability by time point in trial

standard deviation for each channel. We found that this significantly improved performance. Moreover, the true voltage readings are not especially useful measures – it's more sensible and interpretable to report metrics such as mean squared error in terms of normalized voltages.

As discussed in 2.3.1, LFP dynamics vary significantly between saccade directions. Thus, for each trial, 12 extra dimensions were added to each LFP voltage reading to one-hot encode the saccade direction.

3.1.2 Model Descriptions

All of the models consist of a single GRU layer. The longer the prediction lag – lag is defined as the time between input and output LFP voltages – the less confident the predictions. Thus, as lag increases, overfitting becomes more of a concern; therefore, fewer units were used for models with longer lags. Table 2 shows the number of units used in the GRU layer for each prediction lag. For each prediction lag, models using 150, 100, 50, 20, 10, and 5 timesteps of past LFP data were trained – a total of 36 models were created. In order to properly compare performance between models that take in different numbers of timesteps, we trained and tested the models such that predictions are not generated for the first 150 milliseconds of any trial. The model that uses the greatest number of timesteps requires 150 ms of past LFP voltage data to make a prediction, and thus the 150 timestep model cannot generate predictions for the first 150 ms of a trial. Although the lower timestep models *can* generate LFP predictions earlier in the trial, doing so would make the model comparisons improper, as LFP predictions may be easier or harder depending on the time point in a trial.

Each GRU model was trained using Adam, with a learning rate of .01, as an optimizer and mean squared error as loss. The models were trained for 100 epochs with a batch size of 512 – the weights were then reverted to that of the epoch with lowest validation loss. This method was employed, as opposed to the more typical approach of early stopping, due to the fact that training error sometimes drastically increases between epochs when an optimizer that incorporates momentum is used. Using an optimizer without momentum, however, significantly increased training time, and thus we opted to use Adam.

Two baseline models were created for the purpose of comparison. One baseline model, which I'll refer to as **average**, simply predicted an LFP voltage of 0 at every time point. This is a sensible baseline model due to the fact that the training set – because of the standardization process – has a mean LFP voltage of 0 for each channel. The other baseline model, which we'll refer to as **previous**, simply predicts the output to be the most recent input timestep.

lag(ms)	**units**
1	200
5	100
10	75
25	50
50	20
100	10

Table 2: GRU units by lag

		lag (ms)					
		1	**5**	**10**	**25**	**50**	**100**
timesteps (ms)	**150**	0.0026	0.2223	0.4815	0.7301	0.8359	0.9021
	100	0.0011	0.2273	0.4877	0.7486	0.8477	0.9056
	50	0.0006	0.2266	0.4716	0.7635	0.8933	0.9438
	20	0.0008	0.2266	0.4781	0.7753	0.9058	0.9442
	10	0.0018	0.2399	0.5088	0.8023	0.9463	0.9940
	5	0.0125	0.2671	0.5310	0.8391	0.9501	0.9926
	average	0.9988	1.0019	1.0055	1.0159	1.0359	1.0416
	previous	0.055871	0.3838	0.7424	1.2357	1.6218	1.9527

Table 3: Test MSEs of normalized LFP voltage by timesteps and lag

3.1.3 Results

Table 3 shows the test MSE of the normalized LFP voltage for each model; figure 8 shows the same information in a plot. There are some promising results; for each lag, the two baseline models are outperformed by every GRU model, and LFP predictions for 1 ms lags in the future are quite accurate. However, the results significantly worsen as the lag increases.

It's apparent from examining table 3 that for lags shorter than 10 ms, there is little benefit to using a model that takes in more than 20 timesteps. However, increasing the number of timesteps becomes useful as the lag grows.

Figure 9 shows MSE by time point in a trial for the 150 timestep model. It is apparent that MSE significantly increases at around +25 ms, which is very soon after the new eye fixation begins. It is unsurprising, given the voltage spikes seen in figure 2, that the LFP becomes more difficult to predict succeeding a new eye fixation.

Figure 10 shows the test MSEs for the 150 timestep model broken up by channel and direction. It is apparent LFPs are easier to predict for some channels and saccade directions than others.

Figure 11 shows the per-channel MSE at each point in a trial for the 150 timestep and 10 ms lag model. LFPs *do* become more difficult to predict at the onset of a new eye fixation for every channel, but the channel number affects the extra factor of .difficulty. Figure 12 shows the MSE, categorized by saccade direction, at each point in a trial for same model. LFPs are clearly *much* easier to predict at the time of eye fixation onset for some saccade directions than others – this graph supports the hypothesis that the MSE spike in Figure 9 is driven by a subset of saccade directions.

Figure 13 displays LFP voltage predictions for a single trial and channel, using the 150 timestep model. The predictions are very good for a lag of 1 ms, and the results are still impressive for 5 and 10 ms – the predictions for the $t = +20$ ms to $t = +50$ ms period are fairly robust. The predictions are much worse, however, for lags that are 25 ms or longer.

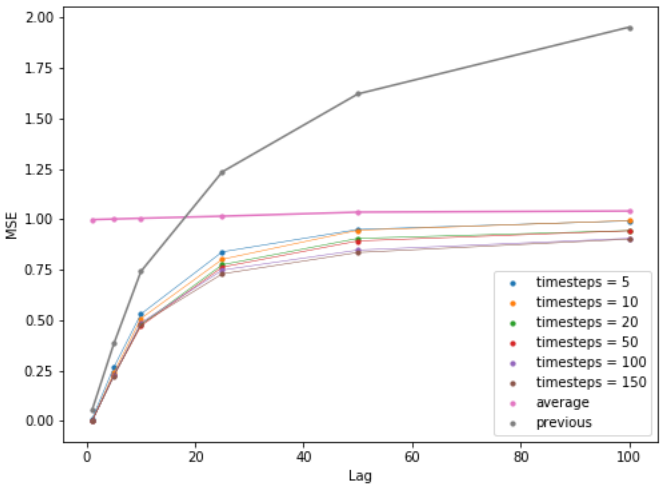

Figure 8: Test MSEs of normalized LFP voltage by timesteps and lag

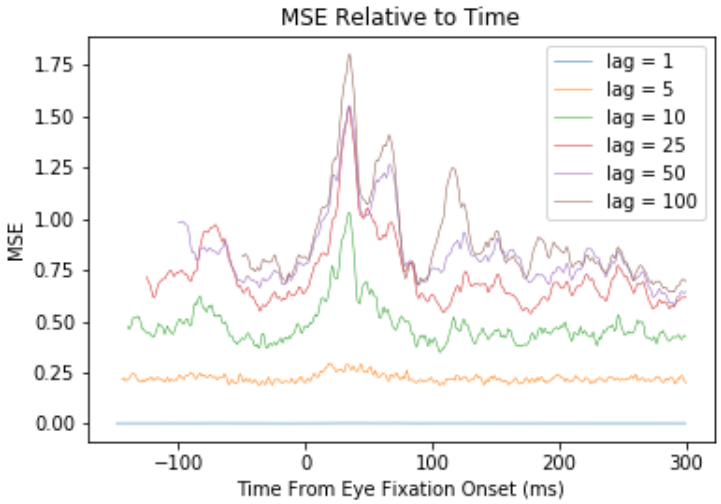

Figure 9: MSE by time point in trial, timesteps = 150

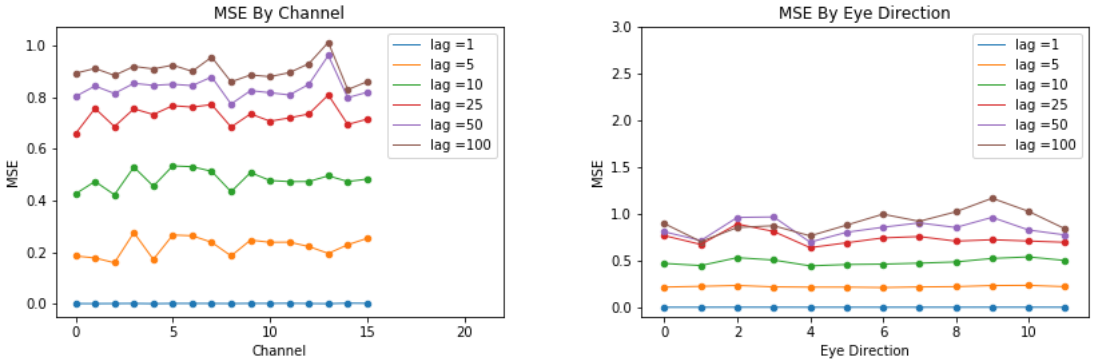

Figure 10: MSE by channel and direction, timesteps = 150

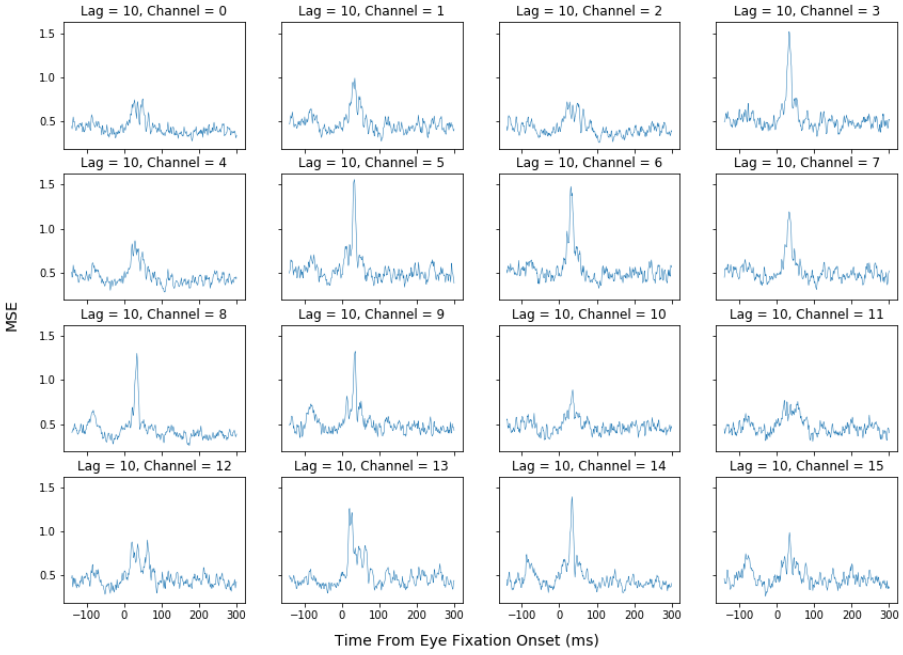

Figure 11: Per-channel MSE at each time point in trial, lag = 10, timesteps = 150

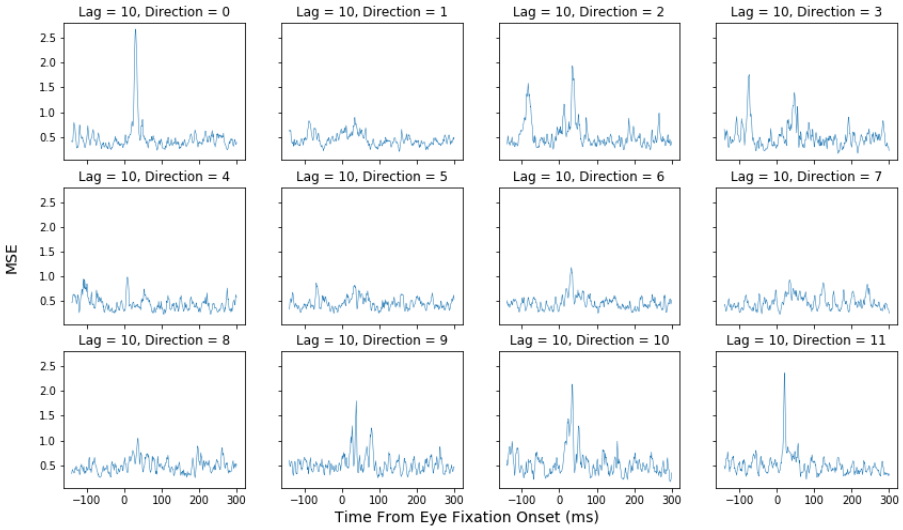

Figure 12: MSE by saccade direction at each time point in trial, lag = 10, timesteps = 150

3.1.4 Other Modeling Approaches

Due to the variation in LFP shapes between saccade directions, I initially attempted to build a separate model for each saccade direction. Two approaches were tried; in one, a separate model was built for each LFP channel, and in another, all LFP channels were predicted simultaneously. The first approach generated very good predictions for 1 ms lags, but had very little predictive ability for longer lags; it's predictions appeared to be quite similar to the **previous** baseline model. The second approach tended to overfit the training data for long lags. This was less of an issue with our final model, likely due to the fact that there is significantly more training data when all saccade directions, rather than a single direction, are included.

I experimented with a variety of neural architectures, but found that a single GRU layer worked best. As written in the proposal, my initial plan was to use an LSTM layer – I found, however, that LSTMs gave similar, but slightly worse results than GRUs.

3.1.5 Future work

Paradiso et. al. have trained SVMs on this dataset to predict saccade direction based on LFP voltage [3]. It would be useful to build a model that predicts saccade direction using GRU or LSTM layers and compare the results with that of the SVM. It would also be valuable to properly train and tune a model that does not encode the eye direction as input for comparison.

3.2 Decoding Mirror and Saccade Trials

Models were built that identify, based on LFP voltage readings, the probability that a trial is a mirror or saccade experiment.

3.2.1 Data Preparation

The data from **C2**, which was generated from **D2**, was used to train, test, and evaluate this model.

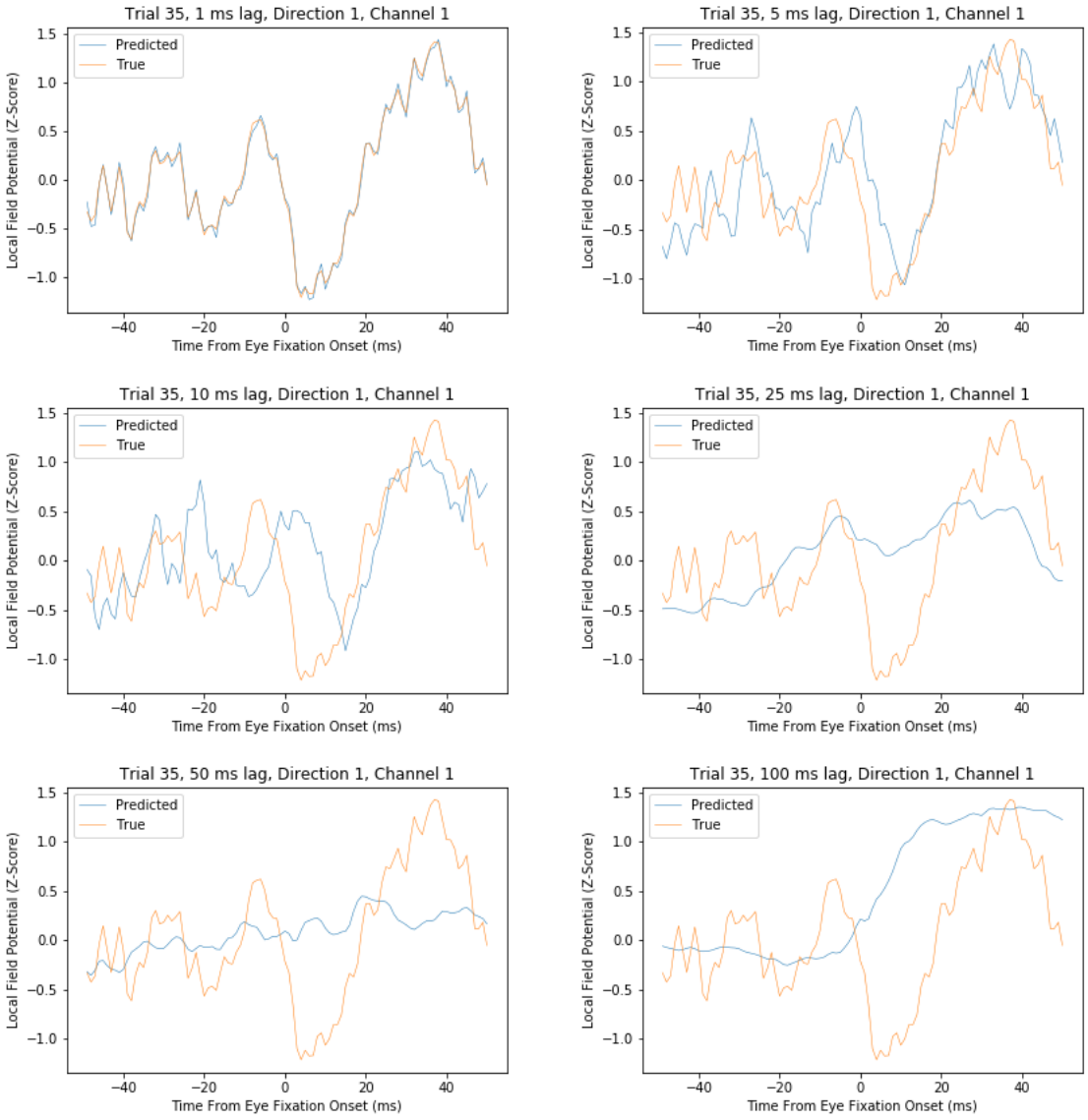

Figure 13: LFP voltage predictions for a single channel and trial, 150 timestep model

I did not use the entire 2000 ms period of LFP readings for each trial, as the start and end of this period held little predictive value. Several different models, each taking in a different number of timesteps of LFP voltages to generate predictions, were trained; in order to properly compare the models, we wanted each model to be able to generate 700 predictions per trial. Therefore, each trial contains LFP data from time $t = (-349 - \text{timesteps}/2)$ to $t = (350 + \text{timesteps}/2)$, where $t = 0$ is the time of eye fixation onset.

To avoid overfitting, trials were split into training, testing, and validation datasets using a .64/.2/.16 train/test/validation split. The split was chosen using a fixed random seed so that results were consistent.

The mean and standard deviation of LFP voltages were calculated across the training trials for each LFP channel. Each trial was standardized by subtracting the mean voltage from the LFP and dividing by the standard deviation for each channel.

Extra dimensions were added to each LFP reading to encode it's time point within a trial. Each trial was divided into 10 ms bins – each trial thus contained (length of trial)/10 bins. The bin number was one-hot encoded for each LFP voltage reading. The rationale behind this decision is that LFP dynamics vary significantly within a trial.

3.2.2 Model Descriptions

Four models were fit – the four models respectively used 100, 50, 20, 10 ms of trial data as input. Every model consisted of a single GRU layer with 5 units, and a final dense layer of one unit with sigmoid activation. **saccade** is the positive class, and **mirror** is the negative class – therefore, a network output represents the probability that a trial is a saccade experiment.

Each network was trained trained using Adagrad, with a learning rate of .01, as an optimizer, and binary cross-entropy loss. The networks were trained for 70 epochs with a batch size of 256. An early stopping callback with a patience of 5, using validation loss as the monitor, was used for each model.

3.2.3 Results

Test accuracy and AUC for each model are shown in table 4. The ROC plots for each model are shown in Figure 14. It's clear from examining this table and plot that using a large number of timesteps, at least up to a point, significantly improves decoding.

Figures 15 and 16 show test accuracy and AUC, respectively, broken up by the centers of LFP windows. To clarify with an example, suppose that AUC is .85 at $t = 100$ for the 100 timestep model; this indicates that AUC for LFPs containing data from $t = 51$ to $t = 150$ is .85. As expected, the model decodes most accurately immediately after the new eye fixation onset, as this is when the difference in LFP dynamics between saccade and mirror trials is most pronounced. An initially unexpected finding is that accuracy and AUC are relatively high during the time period from $t = -300$ to $t = -200$, but dips after $t = -200$. This is somewhat puzzling, as the primate's eye generally does not begin movement until $t = -50$. One hypothesis is that during a saccade trial, a primate makes a decision to move its eye within the $t = -300$ to $t = -200$ time window; this window therefore contains useful decoding information.

Another observation from Figures 15 and 16 is that there is much more variation in accuracy and AUC in nearby LFP window centers when fewer timesteps are used in a model. The AUC and accuracy graphs for the 100 timestep model are fairly smooth; by contrast, the same graphs for the 10 timestep model are very spiky.

Figure 17 shows ROC plots for the subset of LFP windows with centers ranging from $t = 0$ to $t = 99$. The models with more timesteps *especially* outperform the models with fewer timesteps in this trial region.

Timesteps	100	50	20	10
Accuracy	.7024	.6649	.6225	.6063
AUC	.7896	.7398	.6887	.6636

Table 4: Test accuracy and AUC for each model

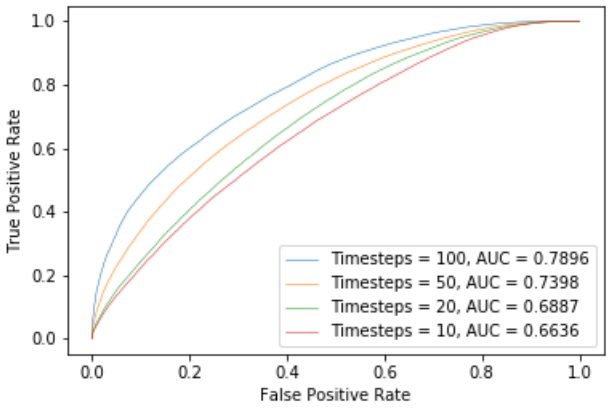

Figure 14: ROC curve for each model

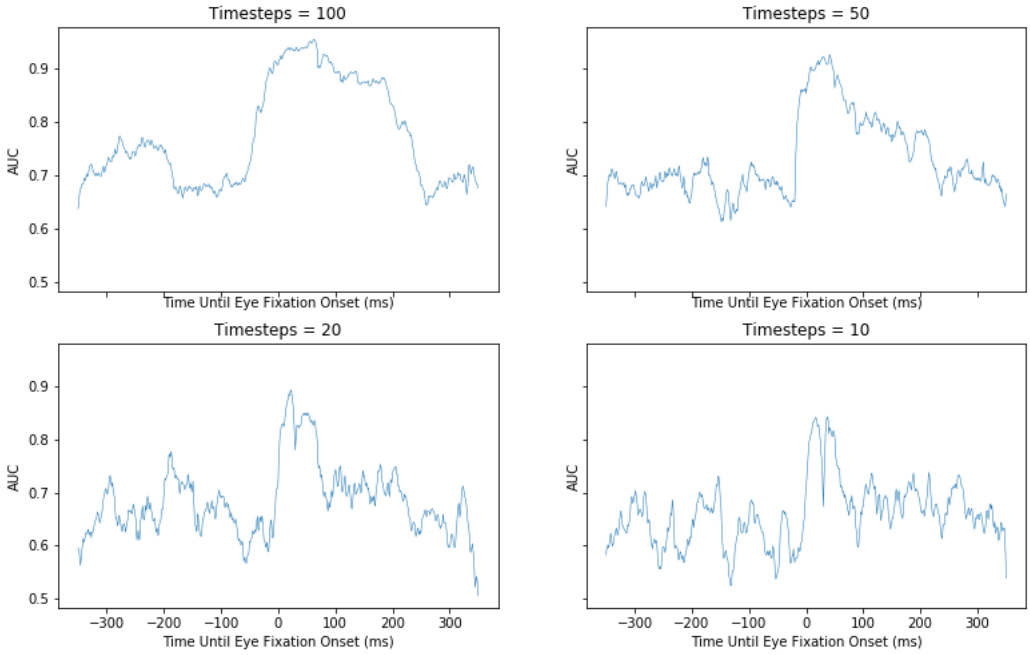

Figure 15: Test AUC by center of LFP window

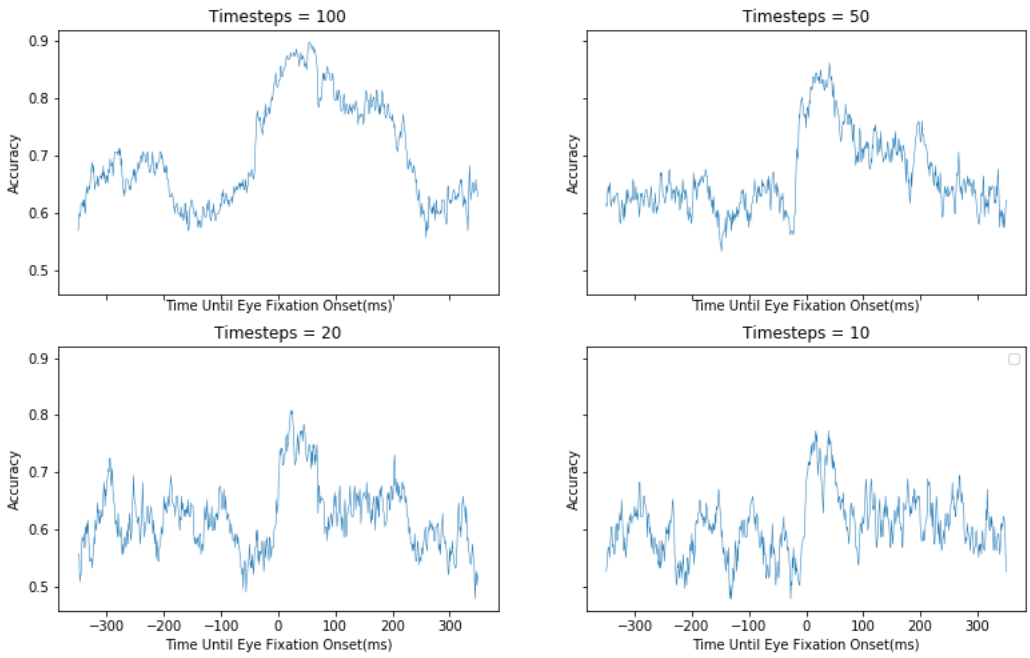

Figure 16: Test accuracy by center of LFP window

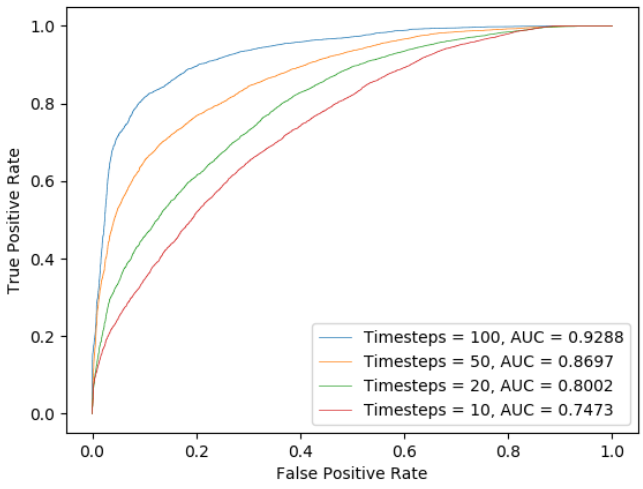

Figure 17: AUC for LFP windows with centers from $t = 0$ to $t = 99$

3.2.4 Other Modeling Approaches

My initial models did not encode within-trial time points of LFP readings. These models had poor performance – there is so much variation in expected LFP shapes between different time points in a trial that fitting a general model is not very effective.

I experimented with several other model architectures, including LSTMs and multilayer perceptrons (MLP), but found that GRUs had better performance.

3.2.5 Future Work

It would be valuable to try to evaluate prediction performance for a model that takes in 200 or 300 timesteps – it's possible that these models would perform noticeably better than the 100 timestep model.

3.3 Neuronal Spike Prediction

Models were fit to predict the probability of at least one spike occurring in a future time window for each of the 52 neurons, using LFP voltages and spike history as inputs. Many models were built, using a variety of timesteps of LFP voltages and spike history, an array of window lengths for spike prediction, and a collection of future time lags, where time lag is defined as the time difference between the most recent LFP and spike input and the start of the spike prediction window. A model with a spike prediction window length of ℓ indicates that the model predicts the probability of at least one spike occurring in an ℓ ms period.

3.3.1 Data Preparation

The data from **C3**, which was generated from **D2**, was used to train, test, and evaluate this model. Each LFP trial in this dataset was 1000 ms, centered around the time of new eye fixation. The entire trial was used for prediction. To avoid overfitting, trials were split into training, testing, and validation datasets using a .64/.2/16 train/test/validation split. The split was chosen using a fixed random seed so that results were consistent.

The mean and standard deviation of LFP voltages was calculated across the training trials for each LFP channel. Each trial was standardized by subtracting the mean voltage from the LFP and dividing by the standard deviation for each channel. Each trial also contains spike information – each 1 ms timestep in the trial has an associated 52-dimensional array that denotes, for each neuron, whether a spike occured in that the relevant time period.

The initial output training data is a 52-dimensional array that indicates whether a spike occurred at a particular timestep in a trial. If the window length for spike prediction is greater than 1, the training data is modified so that a 52-dimensional vector at time $t = k$ indicates whether one or more spikes occurred from time $t = k$ to $t = k + \text{window\_length}$.

3.3.2 Model Descriptions

Models were built for all of the following combinations: 100, 50, and 10 ms of LFP voltage readings and spike information as input, forward lags of 1, 5, 10, and 50 ms, and spike window lengths of 1, 5, 10 and 50 ms. Thus a total of 48 models were created.

Each model consisted of a single LSTM layer with 50 units, and a final dense layer with an output dimension of 52 and sigmoid activation. Thus, each output represents the probability of a neuron spiking within a particular future time window.

Every model was trained with binary cross-entropy loss and used Adam as an optimizer. Each model was trained for 20 epochs using early stopping with a patience of 5 and validation loss as a monitor.

3.3.3 Results

When interpreting results, AUC was used rather than accuracy – no analysis of accuracy statistics was performed. This is because spikes are quite rare. There is a mean of 7.01 spikes per neuron per trial, meaning that in each 1 ms time window, a spike occurs at a .7 percent rate. Thus extremely high accuracy can be reached for models with short spike window lengths by simply never predicting any spikes. Accuracy is simply not a useful measure for short spike time windows. Moreover, because the probability of a spike depends on the spike window length, using accuracy as a metric makes comparison of performance across window lengths quite difficult.

Table 5 displays AUC across timesteps, lags, and window lengths. Figure 18 shows AUC across timesteps and lags for a window length of 1 ms. Figure 19 shows ROC curves by lag for a model with 100 timesteps and a 1 ms window size. Some of these results are surprising. Using 100 ms of LFP and spike data only gives a slight advantage over using 10 ms of data. There is also less of a performance difference between models with low and high lags than expected – however, the performance difference between 1 ms and 50 ms lag models is still significant. Window length seems to have little effect on AUC. One strange effect of window length, however, is that across all timesteps, increasing the window length decreases AUC for 1, 5, and 10 ms lags, but increases AUC for 50 ms lags. I have very little intuition as to why this might be the case.

AUC varies significantly by neuron. Table 6 shows AUCs for the neurons with best and worst respective performance across lags and window lengths for a 100 timestep model; Figure 20 shows the distribution of AUCs across neurons for a 100 timestep, 1 ms lag, and 1 ms window length model. It is import to mention, however, that some of this difference is likely random and may not reflect true differences in predictive ability.

Figure 21 shows AUC by time point in trial – where time point is defined by the time of the last LFP voltage reading and spike used for prediction – for a model with 100 timesteps, 1 ms lag, and a 50 ms spike time window. The long spike time window was chosen because there was less variability between nearby time points than in models with shorter spike windows; long spike windows created more interpretable graphs. The dips in AUC after $t = 0$ and $t = 400$ can likely be explained by figure 7; the spike probabilities noticeably increase after these time points; the model likely does not recognize this and assigns lower probabilities to spike occurrences than it should.

Figure 22 displays true and predicted spikes for a single neuron and trial. The model to create this plot used 100 timesteps, 1 ms lag, and a 1 ms window size. The probability threshold for spike prediction in this trial is .0656; we predict the occurrence of a spike if the probability of a spike, as determined by our model is greater than .0656. This was chosen so that the number of true spikes in the trial – 27 – is the same as the number of predicted spikes. The vertical lines denote the occurrence of true or predicted spikes. The green vertical lines indicate that a spike both occurred and the model correctly predicted that a spike would occur at the specified time point. This graph is promising; there are 3 correct predictions out of 27, and several of the incorrect spike predictions, such as the one at around $t = -300$, are very close in time to a true spike.

3.3.4 Limitations and Future Work

Due to time constraints, we did not spend as much time trying out different architectures and tuning parameters as with LFP prediction and mirror/saccade decoding. It is likely that we could construct better models if given more time.

Our models simply predict the probability of one or more spike occurring in a given time window – we would be interested in building a model that instead predicts the spike-count rate for a given window length. Models that use long window sizes might provide more illumination with that approach than our current method.

There are two other models that we would be interested in building; one that predicts spikes purely from LFP data, eschewing any past spike information, and one that predicts spikes purely from past spike information. It would be fascinating to examine the relative importance of LFP voltages and past spike information.

Timesteps = 100 (ms)

lag (ms) \ window length (ms)	1	5	10	50
1	0.8042	0.8069	0.8025	0.7820
5	0.7955	0.7957	0.7913	0.7764
10	0.7814	0.7826	0.7798	0.7729
50	0.7326	0.7408	0.7390	0.7492

Timesteps = 50 (ms)

lag (ms) \ window length (ms)	1	5	10	50
1	0.8048	0.8052	0.8003	0.7794
5	0.7944	0.7948	0.7899	0.7751
10	0.7793	0.7798	0.7781	0.7691
50	0.7310	0.7356	0.7338	0.7440

Timesteps = 10 (ms)

lag (ms) \ window length (ms)	1	5	10	50
1	0.7956	0.7949	0.7895	0.7688
5	0.7848	0.7841	0.7793	0.7648
10	0.7705	0.7698	0.7678	0.7585
50	0.7225	0.7250	0.7281	0.7359

Table 5: Test AUCs by timesteps, lag, and window length

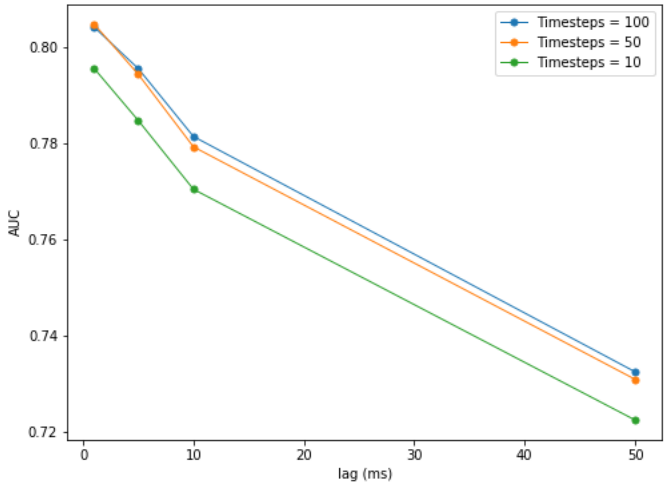

Figure 18: AUC by timesteps and lag, 1 ms spike window

Best Performance

lag (ms) \ window length (ms)	1	5	10	50
1	0.9000	0.8884	0.8736	0.7933
5	0.8702	0.8694	0.8550	0.7822
10	0.8457	0.8462	0.8355	0.7794
50	0.7560	0.7695	0.7596	0.7244

Worst Performance

lag (ms) \ window length (ms)	1	5	10	50
1	0.5776	0.5752	0.5690	0.5470
5	0.5580	0.5666	0.5627	0.5315
10	0.5357	0.5595	0.5569	0.5107
50	0.4621	0.5264	0.5181	0.5106

Table 6: AUC by lag and bin size for neurons with best and worst performance, 100 timestep model

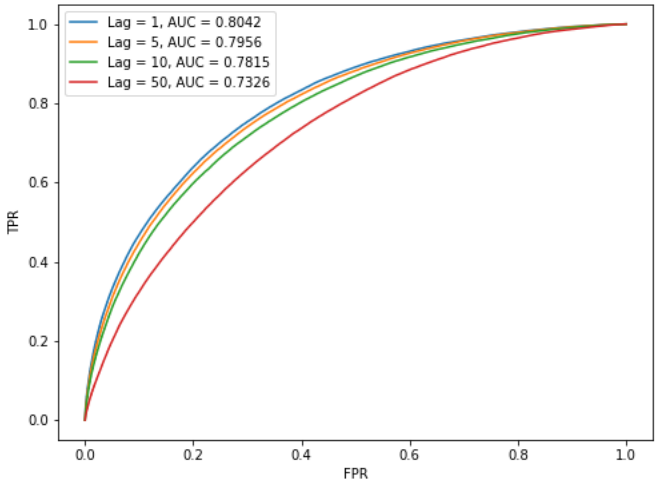

Figure 19: ROC by lag for 100 timestep and 1 ms window length model

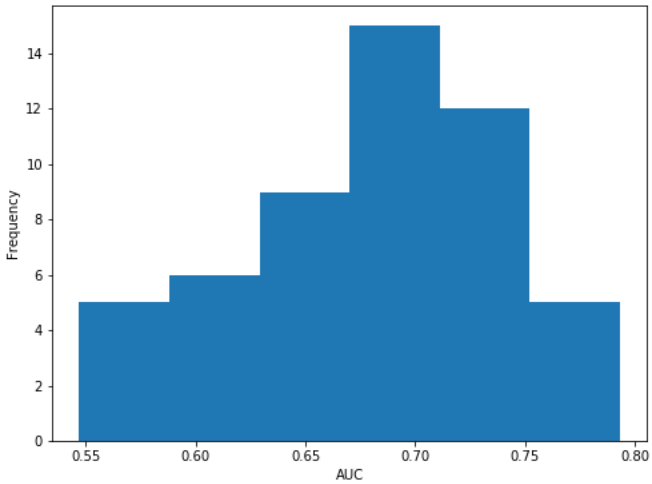

Figure 20: Distribution of AUCs by neuron for 100 timestep, 1 ms lag, and 1 ms window size model

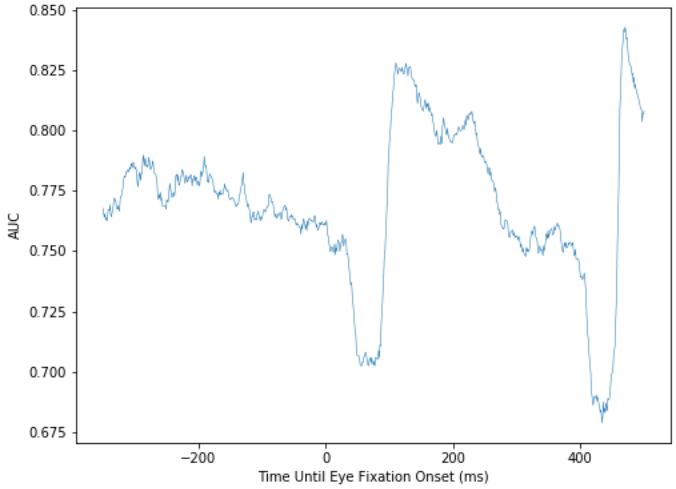

Figure 21: AUC by time point in trial, timesteps = 100, lag = 1, window length = 50

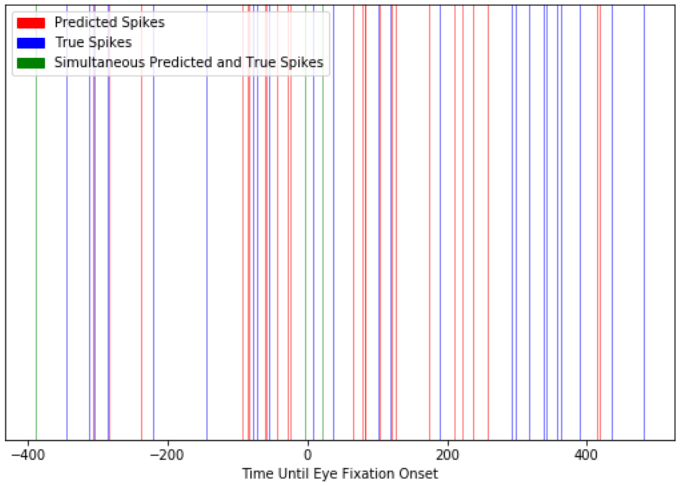

Figure 22: Comparison of true and predicted spikes for a single trial and neuron for model with 100 timesteps, 1 ms lag, and 1 ms window size

4 Tool-chains

As described in section 2.2, the data was converted from Matlab to .csv format – we wrote several Matlab scripts to perform this task. Matlab was also used for some data preparation tasks. Matlab was used to downsample LFPs.

All analysis was performed in Python; Keras was employed to train all of the models; Pandas was used to read and load the .csv files; NumPy was used to store and prepare the datasets; Matplotlib was used to create all of the necessary plots. Models were trained using CCV's GPU resources.

5 Intersection with DSI competency areas

1. **Competency in Probability, Statistics and Machine Learning, Data and Computational Science**

   This capstone required competency in all of these areas. The project involved building machine learning models, two of which – the mirror/saccade decoding and spike prediction models – made inherently probabilistic predictions in that their outputs estimated the probability of a particular outcome. It also required extensive data wrangling; among other tasks, it was necessary to convert Matlab files to .csv files, create sequences for LSTM and GRU layers, and encode variables.

2. **The ability to formulate appropriate questions, perform analyses, and draw appropriate conclusions from big data sets**

   The project certainly required these abilities. I worked with a large amount of data; all of the datasets consist of over 400,000 LFP voltage readings. I conducted rigorous analysis of the results of my models and drew important conclusions from them; I concluded, for example, that LFPs are more difficult to predict immediately after eye fixation onset, and that using a large number of timesteps does little to improve spike prediction.

   Specific questions that guided our analysis were formulated about the datasets. Among the important questions were: what is the longest lag at which it is still possible to make useful LFP predictions? What is the time point of LFP voltage in a trial that's most useful for mirror/saccade decoding?

3. **An appreciation for the limitations of data analysis**

   The project showed the limitations of data analysis at many turns. For example, overfitting was a massive problem for many of the models that we tried; extensive tinkering was necessary to build a well-performing model. And the project provided a sense of what can and cannot be predicted; the 100 ms lag LFP prediction model, for example, had dissapointing results, as it performed only slightly better than an extremely simple baseline model. But this was valuable information to know, and led to an important question: is there simply too much certainty 100 ms in the future to accurately predict LFP voltage?

4. **An understanding of the societal impacts of data and its analysis developed through case studies of relevant topics**

   The capstone had limited overlap with this competency area. However, it is worth noting that neural decoding, as a more general concept, does have significant societal implications. Neural decoding, for example, could lead to innovations such as robotic limbs. By mapping neural activity to muscle movements, a robotic limb could then be designed that instantly responds to a user's neural activity. It also carries some ethical concerns – it is possible that mind-reading and mind-control devices could be built if predictions about one's future neural activity noticeably improve. These questions, however, were outside the scope of this projects.

6 Updates to Capstone Proposal

Some, but not all, of my stated goals in the capstone proposal were accomplished. Two milestones mentioned in the proposal – building a model to predict the future LFP, and constructing a model to predict spikes – were completed. However, I mentioned that I wished to investigate the hidden dynamics of the models – due to time constraints this task was not performed. The proposal also specified that I would compare the results of an RNN LFP prediction model with Seth's autoregressive model that uses a Markov process to predict the LFP. This was never accomplished – this was due to the fact that Seth's model only predicts the LFP at very short future lags, so most of the results are not directly comparable to the GRU model described in this report. Therefore, I felt that the comparison wouldn't be particularly illuminating, and it would be better to focus on other aspects of the project.

In the proposal, one of my stated milestones was to "build an LSTM to predict eye movements." This is somewhat vague; predicting eye movements can mean a variety of things. The mirror/saccade decoding model *does*, in some sense, predict eye movements; an eye movement is made in a saccade trial, and no movement is made in a mirror trial; our model tries to estimate the probability that a movement was made at various points in the trial. However, that was not what I had in mind when the proposal was written; I believed that my goal – which would still be an interesting, and, I think, worthwhile problem to solve – was to predict the probability that the eye is moving at a given time point. This was not accomplished; my capstone supervisors indicated that it would be most helpful to build a model that decodes mirror and saccade trials or decodes saccade direction from LFP voltage readings.

In the proposal, I specifically mentioned that I planned on using an LSTM for every model. LSTMs were only used for spike prediction models; every other model used GRUs, as they were found to have superior performance. GRUs are, however, essentially a slight variation of LSTMs; this is not an especially significant change.

7 Conclusions

I conclude that LFPs from electrodes in V1 contain lots of useful information regarding brain activity that can be used to make predictions; LFPs can be used to predict spikes, decode types of eye fixation trials, and predict future LFPs with non-trivial accuracy. Moreover, it appears that if given enough data, RNNs – specifically GRUs and LSTMs – are effective models for the purpose of making various predictions from LFPs.

8 Next Steps

Potential future work is discussed in sections 3.1.5, 3.2.5, and 3.3.4.

9 Acknowledgements

Seth Akers-Campbell and Michael Paradiso have been essential aides. Seth provided and explained the format of the relevant datasets, was readily available for general clarification questions, and was immensely helpful in writing the capstone proposal; he provided me with useful, accessible information about the project that made the process of writing the proposal much easier. Dr. Paradiso provided a very helpful overview of the neuroscientific motivations for conducting this style of research, and gave extremely clear explanations of the often confusing experimental designs used to collect the relevant data.

I'd also like to thank Thomas Serre for taking the time to meet with me, suggesting relevant papers, and offering computing resources.

10 Time Spent

I spent about 10 hours writing the capstone proposal, 7 hours preparing the capstone pitch, 15 hours reading relevant papers, 100 hours converting data, building models, and analyzing results for LFP prediction, 30 hours converting data, building models, and analyzing results for mirror and saccade decoding, 50 hours converting data, building models, and analyzing results for spike prediction, 30 hours preparing the capstone report, 8 hours preparing the capstone pitch, and 8 hours preparing the Capstone presentation.

References

[1] Nur Ahmadi et al. "Decoding Hand Kinematics from Local Field Potentials Using Long Short-Term Memory (LSTM) Network". *Arxiv Preprint*, 2019. arxiv: 1901.00708.

[2] Louis Kim et al. "Predicting local field potentials with recurrent neural networks". *38th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc*, pages 808–811, 2016.

[3] Michael Paradiso et al. "Transsacadic Information and Corollary Discharge in Local Field Potentials of Macaque V1". *Frontiers in Integrative Neuroscience*, 12(63), 2019. doi: 0.3389/fnint.2018.00063.

[4] Y. Wang et al. "Decoding hindlimb kinematics from primate motor cortex using long short-term memory recurrent neural networks" . *IEEE Engineering in Medicine and Biology Society. Conference*, 2018. doi : 10.1109/EMBC.2018.8512609.