CS230

Predicting Parkinson's disease behavioral state from neural and kinematic data

Marissa Lee, Johanna O'Day, Kirsten Seagers
Department of Computer Science
Stanford University
{mrlee1, odayj, kseagers}@stanford.edu

Abstract

We aimed to identify the biomechanical behavioral state of patients with Parkinson's disease during a clinical research task from local field potential voltage recordings from surgically implanted electrodes in the brain. A logistic regression, LSTM and 1D CNN model were explored initially. The 1D CNN proved most promising and thus extensive experiments were performed to tune hyperparameters. The best 1D CNN model performed with an average area under the receiver operating characteristic of 0.70 during holdout cross-validation. This is a promising initial step toward our ultimate goal of predicting freezing behavior in Parkinson's disease. To continue to improve classification performance, objective labels of freezing of gait will be explored.

1 Introduction

Parkinson's disease (PD) is the second most common neurodegenerative disease worldwide. Gait impairment and freezing of gait are common symptoms that lead to falls, reduced mobility, and decreased quality of life. Freezing of gait is an intermittent, involuntary inability to perform alternating stepping. It often occurs when people with PD attempt to initiate walking, turn, or navigate obstacles. Currently, there is no clinical treatment for freezing of gait or gait impairment, and only experimental deep brain stimulation has shown promise for improving freezing behavior. This is in part due to minimal understanding of underlying neuro-biomechanical mechanisms. Our moonshot vision is to predict freezing from a neural signal so that future iterations of deep brain stimulation might alter stimulation to respond to patient states in real time. To date, no one has been able to relate time-varying neural signals to discrete behavioral states. This is a very challenging problem because little is known about neural control of movement, and the nervous system and human motion are incredibly complex.

Freezing behavior can be studied using a turning barrier course where subjects walk in ellipses and figure eights through narrow wall dividers. Generally subjects freeze more when walking in figure eights in the course [1]. With access to a unique dataset of synchronized neural and kinematic (movement) recordings from freely moving people with PD, we aim to identify what part of the turning barrier course a patient is walking through using neural signals, general clinic visit information, and task-specific variables. The inputs to our algorithm are raw neural signals, which were also hand engineered into frequency bands that are deemed important by the field of neurology; arrhythmicity, a coefficient of variation of step time; and a binary input denoting whether the patient is freezing or not. Our algorithm output a prediction of whether the subject was walking in a figure eight or an ellipse of the turning barrier course. We compared the performances of a long short-term memory network (LSTM) and a one-dimensional convolutional neural network (1D CNN) to that of a baseline logistic regression model.

CS230: Deep Learning, Winter 2020, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

2 Related work

As far as we know, no one has attempted to solve this problem using deep learning. In our approach, we leverage similar previous work for architecture ideas and feature engineering. Prior work with local field potential (LFP) neural data and deep networks has only been conducted with data from individuals with epilepsy. Kim et al. implemented an LSTM to predict LFP signals forward in time from previous LFP signals [2]. While this problem is not currently useful for the field of PD, it is encouraging to see that there is enough signal in the LFP waveforms to successfully predict forward in time. Given this approach, we explored an LSTM approach. We also leveraged a blog post describing an implementation of activity classification from inertial measurement unit time series data as an example of a 1D CNN architecture we could implement [3]. One concern we had prior to implementation of either an LSTM or CNN network architecture was whether our neural data changed enough between behavioral states to lend successful classification. LFP signals do not have the same distinctions observed in accelerometer signals between sitting and walking, which were easily identified by the example 1D CNN. To aid our algorithm, we explored how time series data are hand engineered to help networks learn more complex relationships from raw signals. Given the similarity of our voltage signals to audio signals, we performed spectrogram analyses on our LFP signals and used these as hand-engineered feature inputs [4][5][6]. We also encountered applications of neural time series data that similarly performed and supported spectrogram analyses for increased model performance [2][7][8]. When applying spectrogram analysis to our LFP signal, we did not see an improvement in our results. However, when we reduced the number of features given to the network, incorporating power time series for specific frequency bands that the neurology field believes to be of high importance, we did see a slight improvement (spec3D_beta). Comparing our data processing to techniques from the literature was a valuable learning process for our team to hone in on the most appropriate data and model architecture for our problem.

3 Dataset and Features

In collaboration with Dr. Helen Bronte-Stewart's lab in Stanford Neurology, we obtained synchronized bilateral neural LFP and kinematic data from patients with PD walking around the turning and barrier course, designed to elicit impaired walking. This is repeated twice: patients start on the left or right side of the course to capture clockwise and counterclockwise turns around barriers walking in an ellipse or a figure eight [1].

Our dataset consisted of 4.4 minutes (>100k examples) of walking data from a single patient walking in the turning barrier course over two different research clinic visits. Six total walks were recorded, each 40 to 50 seconds in length. A number of walking trial-specific variables were included that would likely affect the neural signal, like which side of the body was more affected by the disease, which direction the patient was turning in the obstacle course (left/right), what part of the course the patient was in (ellipse/figure of eight), and whether the patient was experiencing tremor over time. A neurologist rated binary patient freezing behavior by video.

3.1 Neural Data Processing

Raw local field potentials (two voltage time series) were sampled at 422 Hz from the right and left subthalamic nuclei of the brain, and these were used as inputs to the model. These raw signals were further filtered using a zero-phase 8th order Butterworth bandpass filter, squared and rectified. An amplitude envelope was calculated by linearly interpolating consecutive peaks of the filtered, squared signal to form an envelope of the maximum power. This power time series was created for frequency bins of 4-Hz bandwidths over a range of 2 to 58 Hz [9]. We refer to this input as "spec3D." Several frequency bands have been related to PD gait pathology, so we also summed the power in the alpha (8-12 Hz), low beta (12-20 Hz) and high beta (20-28 Hz) bands [1][9]. We refer to this input as "spec3D_plus." To further simplify inputs, we also used just the beta bands from spec3D, since these bands are known to have elevated power near freezing. We refer to this input as "spec3D_beta". From initial tests, spec3D_beta proved to be the most informative.

3.2 Data Cleaning

All data were normalized within-subject to have mean of zero and standard deviation of one. We conducted holdout cross-validation, where data were split into train (5 walks) and test (1 walk) sets. Each file was divided into windows of 100 samples with 99% overlap. This equated to approximately 90,000 train examples and 17,000 test examples in each iteration of cross-validation.

3.3 Outputs

In response to some of our initial results, we reframed our problem by changing the output of our model. We initially attempted to predict arrhythmicity, a continuous variable that describes gait and moved to a simpler classification problem (predicting binary freezing and then predicting figure eight or ellipse). When we changed outputs, we had to adapt some of our data pre-processsing. With the classification problem, we balanced the dataset to include equal examples of the output. When performing error analysis on our classification to predict freezing, we discovered that the neurologist sometimes included the transition period into freezing as part of the freezing label. As a result, we automated removing 15% of the data before and after a labeled freeze to obtain more accurate labels. After further analysis, we identified other variability in the freezing label. Shuffling movement and freezing were sometimes both classified as freezing. To mitigate the impacts from the objectivity of our output labels, we moved to classifying whether a patient was walking in the ellipse or a figure eight portion of the turning barrier course.

4 Methods

Data manipulation and models were built using Keras [10], Tensorflow [11], Numpy [12], and Scikit-learn [13]. Architectures were designed from scratch.

4.1 1D CNN

We used a 1D convolutional neural network (1D CNN) because it can take parallel time series data as inputs and learn an internal representation of these raw time series data directly. This matches our goal of using neural data (voltage or power time series) as inputs. Our 1D CNN featured a 1D convolutional layer followed by a dropout layer, a MaxPool layer, a flatten layer, and then a fully connected layer using a sigmoid activation (Figure 1). The model takes windows (100 samples long) of time series data consisting of the neural and behavioral features described above as the input and assigns a probability that the window is the output. A probability of 1 corresponds to high confidence in a figure eight label, and a probability of 0 corresponds to a high confidence in an ellipse label. For our cost function, we used binary cross-entropy loss, which is defined by the equation below for prediction $\hat{y}$ and true label y:

$$\mathcal{L}(\hat{y}, y) = -(y\log(\hat{y}) + (1-y)\log(1-\hat{y}))$$

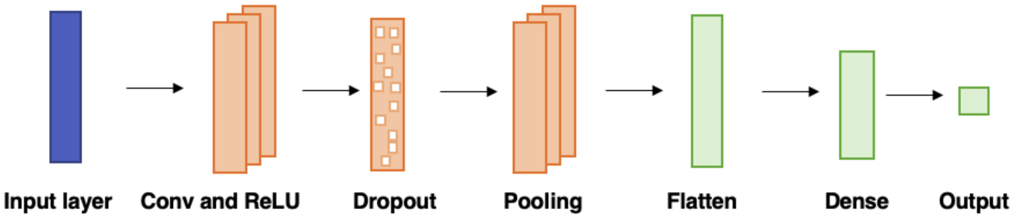

Figure 1: 1D CNN architecture

4.2 LSTM

We also explored using a long short-term memory (LSTM) network, a type of recurrent neural network, due to the time series nature of our input data. LSTM networks are especially equipped for such tasks because there can be lags of unknown duration between these events in a time series, and the LSTM is able to use its feedback connections to make predictions based on previous information in addition to the current information. Our LSTM model consisted of a single layer with four neurons, followed by a fully connected layer with sigmoid activation. We again minimized binary cross-entropy loss.

4.3 Baseline Logistic Regression

We used logistic regression as a baseline model to compare the performance of more complex architecture deep networks. The logistic regression model consisted of one fully connected layer with a sigmoid activation. This model employed binary cross-entropy loss as well.

5 Experiments/ Results/ Discussion

5.1 Experiments

After short investigations with logistic regression, a LSTM network, and a 1D CNN, we chose to approach our problem with a 1D CNN. A 1D CNN trains relatively quickly while still accommodating time series data. We performed over 100 tests with our 1D CNN, manually tuning our model by adjusting hyperparameters in coarse-to-fine searches. These hyperparameters included learning rate, convolutional layer filter and kernel sizes, number of layers, dropout rate, the type of pooling layer, and regularization parameters.

We ran experiments with several different inputs, ranging from raw neural signal to highly hand-engineered features. We evaluated our various experiments using area under the curve (AUC) of the receiver operating characteristic (ROC) and accuracy based on the optimal threshold identified from the ROC curve. The ROC is a plot of sensitivity (the true positive rate) against nonspecificity (the false positive rate) based on binary classification. The entire curve is generated by setting different thresholds with which to classify predictions based on probabilities. Area under the curve is often used to characterize model performance, with an area of 1 suggesting perfection and an area of 0.5 suggesting random guessing.

From the mean ROC from our holdout cross-validation, we identified the threshold that yielded the point closest to the upper left-hand corner as our optimal threshold. With this threshold, we calculated precision (true positives divided by predicted positives), sensitivity (true positives divided by actual positives), and accuracy (true predictions divided by total predictions).

5.2 Results

The best model we achieved, along with a fraction of our experiments, are shown in Table 1 and Figure 2, which shows the average ROC of our cross-validation, an aggregate confusion matrix normalized by the number of total test examples, and the average loss curves for our train and test sets for our best-performing model.

Model	Mean AUC (SD)	Accuracy at Optimal Threshold
Logistic Regression	0.65 (0.10)	0.60
LSTM	0.65 (0.10)	0.61
Conv32	0.62 (0.06)	0.60
Conv4	0.65 (0.10)	0.62
Conv4 with regularization	0.70 (0.08)	0.69

Table 1: Results of hyperparameter implementation and tuning.

Our best model achieved an aggregate precision of 69%, a sensitivity of 68%, an AUC of 0.70, and an accuracy of 69% at each of its optimal thresholds in cross-validation. We recognize that this is an overestimate of a single model's performance, since these optimal thresholds varied.

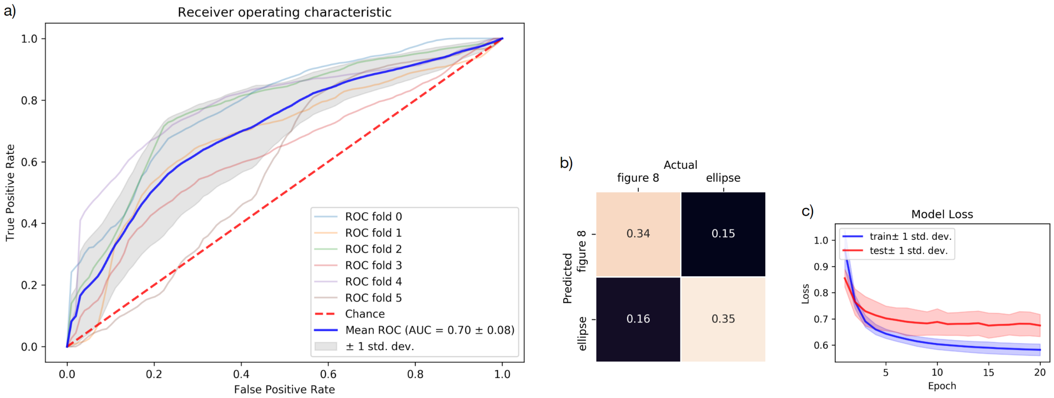

Figure 2: a) Receiver operating characteristic for holdout cross-validation. b) Aggregate confusion matrix of holdout cross-validation folds. c) Average training and test losses over all cross-validation folds.

5.3 Discussion

The biggest challenge we faced at the beginning of this project was with overfitting our training sets. Consequently, our largest improvements resulted from decreasing the number of filters to reduce the number of model parameters; and implementing L1 regularization in our convolutional layer.

Much of our challenge going forward has to do with input features. When we run the model without neural data as an input (using only movement characteristics), the model performs comparably to our best-performing model (Table 2). When we run the model with neural data as the only features, the model performs close to chance. This contradicts our hypothesis and knowledge of biology, which would suggest that neural data is predictive. Our more complex model could potentially learn the relationships between our input features. To use a more complex model without overfitting, we will need to collect more data.

Even hand-engineering neural signals remain noisy and have high variance within patients. Human ability to obtain information about movement from neural signals like these is little to none, making this a difficult problem. Research to understand the biological relationships that exist between neural signals and movement will be critical to developing useful models.

Features	**Mean AUC (SD)**	**Accuracy at Optimal Threshold**
All	0.70 (0.08)	0.69
Arrhythmicity + freeze	0.70 (0.12)	0.66
Neural + arrhythmicity	0.66 (0.06)	0.62
Neural + freeze	0.60 (0.09)	0.58
Neural only	0.49 (0.04)	0.50

Table 2: Results of varying input features.

6 Conclusion/ Future Work

Our 1D CNN with all features or with only arrhythmicity and freeze features performed with the highest mean AUCs of 0.70, and our neural-only model performed poorly. This poor performance is likely due to high variance in the neural data and the small dataset. To work toward our future goal of predicting freezing events in PD, we will employ an objective measure of freezing based on an inertial sensor to decrease variance in the true labels, and obtain longer recordings (hours) from patients walking.

7 Contributions

All team members contributed equally to the direction of the project through weekly meetings and frequent check-ins. ML conducted data manipulation, implemented baseline models, and tuned hyperparameters in the final model. JO gathered data, conducted initial formatting and manipulation, and served as our AWS master. KS conducted further data manipulation, set up the model, and ran experiments. Our thanks to Lukasz Kidzinski, Pranav Rajkapur and the Helen Bronte-Stewart lab for their support and guidance.

8 Code

Google Drive link to Colab notebooks:

`https://drive.google.com/drive/folders/1fxtvgr7iul3WUJftkycUI9RvSzMR6_xh?usp=sharing`

References

[1] Syrkin-Nikolau J et al. Subthalamic neural entropy is a feature of freezing of gait in freely moving people with parkinson's disease. *Neurobiol. Dis.*, 108:288–297, Sep. 2017. doi: 10.1016/j.nbd.2017.09.002.

[2] Kim L et al. Predicting local field potentials with recurrent neural networks. In *International Conference of the IEEE Engineering in Medicine and Biology Society, Orlando*, 2016. doi: 10.1109/EMBC.2016.7590824.

[3] Brownlee J. `https://machinelearningmastery.com/cnn-models-for-human-activity-recognition-time-series-classification/`, 2018.

[4] Lee H, Largman Y, Pham P, and Ng A. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in Neural Information Processing Systems 22*, 2009.

[5] Marolt M, Kavcic A, and Privosnik M. Neural networks for note onset detection in piano music. In *International Computer Music Conference, Gothenberg*, 2002.

[6] Wyse L. Audio spectrogram representations for processing with convolutional neural networks. In *International Conference on Deep Learning and Music, Anchorage*, 2017.

[7] Livezey JA, Bouchard KE, and Chang EF. Deep learning as a tool for neural data analysis: Speech classification and cross-frequency coupling in human sensorimotor cortex. *PLOS Computational Biology*, 15(9):e1007091, 2019. doi: https://doi.org/10.1371/journal.pcbi.1007091.

[8] Ruffini G et al. Deep learning with eeg spectrograms in rapid eye movement behavior disorder. *Front. Neurol.*, 10:806, 2019. doi: 10.3389/fneur.2019.00806.

[9] Anidi C et al. Neuromodulation targets pathological not physiological beta bursts during gait in parkinson's disease. *Neurobiol. Dis.*, 120:107–117, Dec. 2018. doi: 10.1016/j.nbd.2018.09.004.

[10] Chollet F et al. Keras. `https://keras.io`, 2015.

[11] Abadi M et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[12] Oliphant T. NumPy: A guide to NumPy. USA: Trelgol Publishing, 2006–. [Online; accessed <today>].

[13] Pedregosa F et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.