Predicting Epileptic Seizures from Intracranial EEG Recordings

Vickram Gidwani
Electrical Engineering
Stanford University
gidwaniv@stanford.edu

Yunha Hwang
Computer Science
Stanford University
yunha@stanford.edu

Gerardo Rendón
Management Science and Engineering
Stanford University
grendon@stanford.edu

Abstract

Epileptic seizures are unpredictable episodes of turbulent brain activity with bodily manifestations, such as violent jerking and other effects. The purpose of this study is to be able to detect seizures in advance, given that they are a source of major stress and can be potentially damaging, even fatal, for victims who suffer from them. To achieve our objective, we trained a Convolutional Neural Network and a Recurrent Neural Network using brain activity recorded by electrodes in dogs. Training samples were 30 second readings of neural activity across 16 different electrode channels. Our final RNN used regularization techniques like dropout and early stopping, and binary cross-entropy loss for computing loss, achieving modest performance.

1 Introduction

Epileptic seizures are characterized by turbulent brain activity accompanied by bodily manifestations. These physical manifestations, which happen as a result of the increased brain activity, may range from changes in perception to more violent, sudden, and uncontrollable movements of body parts. About 1% of the world's population suffers from these sudden spikes of neural activity; of these victims, many have gone through strong feelings of stress and anxiety, frequent medical interventions, and even death as a result of epileptic complications. Thus, any research that pushes the bounds on preventative and/or potential cures for this condition will always be welcomed in the medical community. A large number of lives are affected both directly and indirectly as a result of this phenomenon [1, 2, 3].

The goal of this project is to predict epileptic seizures with enough intervention time to potentially prevent said seizures from fully occurring. In order to achieve this task, we have architected and trained a neural network. Given 30-second intervals of neural activity recorded across 16 implanted electrodes in dogs, our network is meant to classify voltage readings as either preictal or interictal segments - the former denoting the presence of an epileptic seizure in the next 10 minutes and the latter denoting its absence. Due to the physiological homology between dog and human brains, the high spatiotemporal resolution from electrode dog readings can still provide valuable information for later predicting epileptic seizures in humans [2].

2 Related work

The motivation of this project has also come from the proliferation of the use of deep learning techniques to build decoders of neural data as well as the recent development of technologies that can suppress active seizures through the use of deep brain stimulation [3, 4, 5, 6]. We envision a neural network to serve as a decoder for oncoming epileptic seizures in order to activate deep brain stimulating devices such that patients with this debilitating disease can have some control over their symptoms.

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

Seizure prediction has been studied with increasing rigor since the 1970s [3]. Most approaches use a moving window technique where a 10 to 40 second window of EEG data is used in the prediction algorithm [2, 4]. Preprocessing of the temporal data can take many forms, including calculation of spectral power across five well-defined frequency bands in EEG, spike detection and calculation of spike rates, as well as simple de-noising of the data [3, 5, 7, 8, 9, 10, 11]. Our choice of a 30 second window was motivated by this previous work. The most common decoding algorithm is an SVM, although criticisms of this approach include the inability of this method to generalize to multiple patients or datasets [3, 5, 12]. Most approaches use hundreds of hours of iEEG, which we were unfortunately unable to obtain [5].Almost every studied used the EPILEPSIAE/Freiburg database [3, 5], which contains large amount of patient recordings. Unfortunately, we were not able to obtain access to this database either as it is extremely costly. Thus, we used the data from a Kaggle competition in which the performance of the best algorithm was an AUROC of 0.84. [1] We use this as a benchmark for our performance since we used the same dataset.

3 Dataset and Features

Our data set was extracted from a Kaggle challenge titled American Epilepsy Society Seizure Prediction Challenge and was initially presented as intracranial EEG readings from both dogs and humans, respectively [4]. Data from multiple trials across five dogs and two human patients was provided. The dog data was captured with a device that allowed for ambulatory monitoring, so brain signals were being recorded 24/7. The chosen dogs all suffered from the frequent onset of epileptic seizures, allowing for multiple seizure recordings per dog. Each data entry recorded was classified as either preictal, if it was taken right before a seizure, or interictal, if taken for long stretches of time between seizures.

In the end, data from only one dog was used for our data set; the trials from humans and the other four dogs were discarded. Human recordings were not used because they used a different sampling frequency than dog recordings. Other dog recordings were also discarded because of the inability to perfectly associate electrodes in different dogs due to differences in implantation, pathophysiology, and brain homology. What this implies is that the orthogonal dimensions of the data for each dog were represented by different combinations of recording channels. By combining data from all of the dogs, we limit the ability of the network to clearly differentiate the dimensions in the data. For these reasons, we only used data from the dog that had the most trials.

In its initial format - before being preprocessed - the data represented electrode recordings from 16 electrodes sampled at 399.6098 Hz for 10 minutes. This format, however, gave rise to two main problems. First, each 10 minute data matrix was very large: 16 by 239,766 (399.6098 Hz x 600 seconds) entries. Second, we only had 34 such preictal data matrices from our canine subject. Not only were each of our data points too large for our desired architecture, but we didn't have enough of them.

To solve these two problems, we further divided the 10 minute segments into 30 second intervals. Given previous research presented in [3, 5], we felt that all periodic neural activity that would be representative of an oncoming seizure should be captured within a 30 second time frame. Other research has shown that the lowest frequency of relevant data for this task is 0.1 Hz, which should be captured within this time frame as well [3, 7]. Each 30 second segment was represented by a 16 by 11,988 matrix (399.6098 x 30 seconds), giving us more data points and smaller entries.

The objective of the neural network we're training is to classify the 16 by 11,988 matrices of electrode readings as either preictal or interictal segments. In total, We had 680 preictal data points and 640 interictal data points, of which 90% randomly selected samples made up our training set and the other 10% made up our development set.

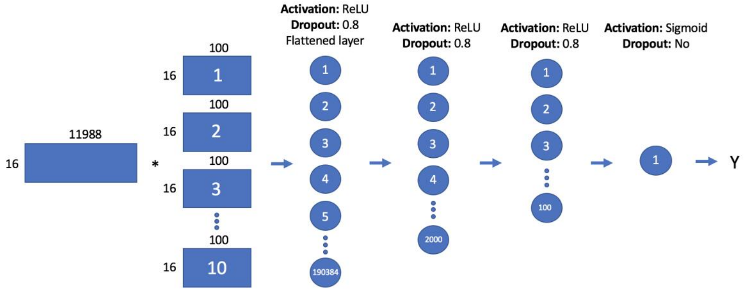
4 Methods

We designed two main architectures for this task. Given previous research showing the use of the Fourier Transform in epilepsy prediction, we used the raw temporal data and the Fourier transformed data on both of these architectures. We reused the basic architectures we had because we felt that they had done a decent job of aggregating all of the information in our large input size and learning relevant features. We made slight adjustments to many parameters, including dropout rate, size, strides, and more, in order to accommodate the different nature of the data.

4.1 Model 1

Our first model was a convolutional neural network built upon the idea that 1D CNNs have been used for sequence data. In brief, a CNN uses a convolutional layer - a weighted layer that traverses the entirety of the input matrix - to extract relevant features and often to reduce the size of the input data. We used 10 filters, a stride of 10, and a filter size of 100 spanning across all 16 input rows. This model didn't yield as good results as the subsequent RNN, which we'll describe in more detail.

Figure 1: Model 1, a Convolutional Neural Network



4.2 Model 2

We built our second model upon the idea that Recurrent Neural Networks are well suited for decoding temporal data. This architecture used Long Short Term Memory (LSTM) cells because of their ability to contain memory only from relevant time points. In brief, an LSTM cell computes weights that allow it to determine whether a new time input is relevant and whether it should "forget" or "keep" information from previous time inputs.

The first layer of this model was a convolution to decrease the length of the input from its 11988 time points. Again, this 1D convolutional layer covered all 16 streams of electrode data (all input rows) with one filter. The number of channels, the filter size and the stride varied depending on whether the input data was raw temporal or Fourier transformed; the former had 32 filters, each with a size of 12 and a stride of 6, while the latter had 16 filters with a size of 10 and a stride of 10. After the convolutional layer, a batch normalization step followed, and then two LSTM layers. The first LSTM layer contained 48 hidden units and the second layer contained 80 hidden units. Also, the first LSTM layer fed a sequence of outputs to the second LSTM layer, whereas the second LSTM layer only fed a vector from its last cell to the output layer. The final output layer was a single fully-connected neuron with a sigmoid activation function, due to its binary output.

We used binary cross-entropy as our loss function and used Adam optimization to update our parameters during backpropagation.

The function that is minimized for binary cross entropy loss, for a single training example, is:

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

The sum of all of these losses makes up the loss for one epoch throughout entire training set. The reason why this function works is because in order to minimize the function, $\hat{y}$ needs to be as large as possible when y = 1 and $\hat{y}$ needs to be as small as possible when y = 0.

The Adam optimization algorithm is a way for quicker weight updating and loss minimizing. Instead of the regular gradient descent weight update, which is

$$w = w - \alpha dw$$

the Adam optimizer updates weights integrating a momentum component and an additional square root division, allowing accelerated movement in the horizontal direction of the loss function but more limited movement in the vertical direction. The equations are the following:

Adam:

$$w = w - \alpha \frac{V_{dw}^{corrected}}{\sqrt{S_{dw}^{corrected}} + \epsilon}$$

Momentum component:

$$V_{dw}^{corrected} = B * V_{dw} + (1 - B) * dw$$

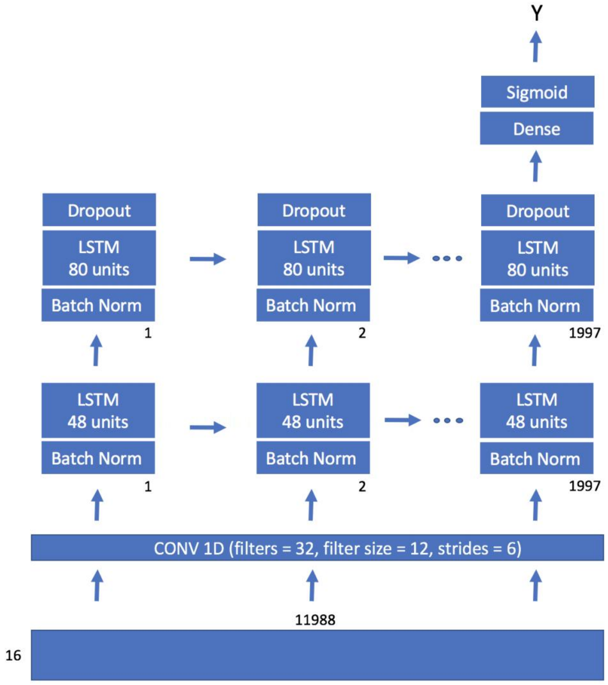
Square root component:

$$S_{dw}^{corrected} = B * S_{dw} + (1 - B) * dw^2$$

In the above equations, both V and S are exponentially weighted averages across time values (up to the point where the weight is updated). They are "corrected" due to potential biases.

We also used dropout, a regularization technique that randomly activates only certain cells on a layer at a pre-defined rate. This was used on the output of the second LSTM layer. Batch normalization was also used before the other two LSTM layers - this method normalizes the samples in each mini-batch. The differences in implementation for the temporal data and Fourier Transformed data were the dropout rates (0.7 vs 0.5), the number of convolutional filters, their size, and their stride.

Figure 2: Model 2, a Recurrent Neural Network with LSTM cells

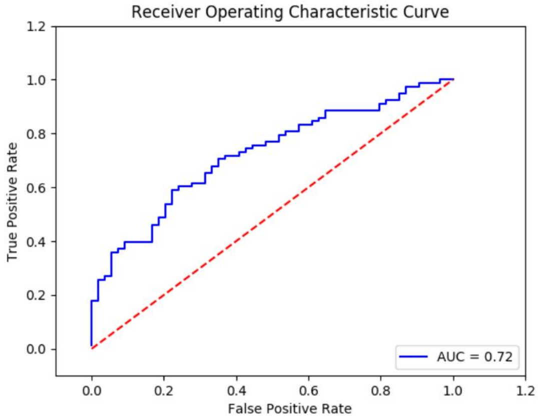


5 Experiments/Results/Discussion

The following table summarizes our results. Model 2, without the initial Fourier transformation, is what achieves the best results. Even though it is still overfitting given the high training accuracy, it was overall the one with the highest test accuracy (69.7%) and AUROC (73.3).

	Training (1188 samples)	Test (132 samples)	AUROC
Model 1	68.5%	62.9%	54.0
Model 1 (FFT)	79.1%	64.4%	54.5
Model 2	99.2%	68.9%	72.0
Model 2 (FFT)	100%	62.9%	63.8

Figure 3: ROC curve for Model 2



As mentioned in the previous section, we experienced many problems with overfitting in Model 2. Initially, we were unable to get our development set accuracy significantly above random chance (50%), while each of our models was able to achieve 100% train set accuracy. As a result, we tried a number of different regularization methods, including dropout, early stopping, and L2 regularization. We found dropout and early stopping to be the most useful techniques and had them implemented in our final model.

Given the results of the Kaggle competition, we know that an AUC of 0.84 is achievable [1]. Unfortunately, our model did not match this level of performance. We believe this is likely due to the combination of the large size of our inputs and the low number of examples. The large input size requires that we have a large model to decode any single input, which requires an even greater data set. Therefore, we believe that our data set was far too small for this task (only 1320 samples including both training and test). While we searched for other labeled intracranial EEG data from preictal and interictal periods recorded at 400 Hz, we were unable to find datasets that met all of these parameters. For example, https://www.epilepsyecosystem.org/ has yet to publish its data, EPILEPSIAE has a paid access policy, and IEEG.org can be hard to navigate to find preictal and interictal labeled data.

6 Conclusion/Future Work

As mentioned before, the RNN without the initial Fourier transformation was the highest-performing model, most likely because LSTM cells allow RNNs to have memory. What this means is that an RNN can be more valuable in situations where future time events depend on all previous time events, as is the case with the brain activity that we're dealing with.

If we had more time, there are certain things we would've further explored. First, we believe more complex methods of preprocessing may have helped our performance. For example, spike sorting to acquire a spike raster may have served to de-noise the data as would a smoothing filter like a Gaussian kernel convolution. Additionally, we believe a spectrogram might take advantage of both the Fourier-encoding of the data and the temporal nature of the data, as it shows the frequency components over time. Converting the data into a spectrogram before feeding into an LSTM would be another architecture that would be interesting to explore.

As stated earlier, we used data from only one dog due to the incongruence of the data across all study participants and potential misplacement of electrodes. We initially tried using our model on the full dataset, but found the model was taking very long to train and was not reaching appreciable performance in the development set. While this hampered our study, it is important to recognize that in any clinical implant, the decoders should be trained on the data from that specific implant, perhaps optimizing weights in a pre-trained model.

7 Contributions

Vickram: Integrated Fourier transformations. Coded mostly model 1 and some of 2. Tuned hyperparameters. Contributed to writing/editing the report.

Yunha: Coded models 1 and 2. Tuned hyperparameters. Contributed to writing/editing the report.

Gerardo: Did most of the poster, wrote/edited about half of the report, helped with debugging and hyperparameter tuning.

References

[1] “American Epilepsy Society Seizure Prediction Challenge | Kaggle.” Kaggle, 2014, www.kaggle.com/c/seizure-prediction.

[2] Howbert JJ, Patterson EE, Stead SM, Brinkmann B, Vasoli V, Crepeau D, Vite CH, Sturges B, Ruedebusch V, Mavoori J, Leyde K, Sheffield WD, Litt B, Worrell GA (2014) "Forecasting seizures in dogs with naturally occurring epilepsy." PLoS One 9(1):e81920.

[3] Mormann F, Andrzejak RG, Elger CE, Lehnertz K (2007) "Seizure prediction: the long and winding road." Brain 130: 314–333.

[4] Barak, Omri. “Recurrent Neural Networks as Versatile Tools of Neuroscience Research.” Current Opinion in Neurobiology, vol. 46, 1 Oct. 2017, pp. 1–6.

[5] Gadhoumi, Kais, et al. “Seizure Prediction for Therapeutic Devices: A Review.” Journal of Neuroscience Methods, vol. 260, 15 Feb. 2016, pp. 270–282.

[6] Shafer, Patty. “FDA Approval: Medtronic Deep Brain Stimulation for Medically Refractory Epilepsy.” Epilepsy Foundation, 1 May 2018.

[7] Park Y, Luo L, Parhi KK, Netoff T (2011) "Seizure prediction with spectral power of EEG using cost-sensitive support vector machines." Epilepsia 52:1761-1770.

[8] Bandarabadi, Mojtaba, et al. “Epileptic Seizure Prediction Using Relative Spectral Power Features.” Clinical Neurophysiology, vol. 126, no. 2, 1 Feb. 2015, pp. 237–248.

[9] Giannakakis, Giorgos, et al. “Methods for Seizure Detection and Prediction: An Overview.” Modern Electroencephalographic Assessment Techniques Neuromethods, vol. 91, 2 Aug. 2014, pp. 131–157.

[10] Li, Shufang, et al. “Seizure Prediction Using Spike Rate of Intracranial EEG.” IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 21, no. 6, 9 Oct. 2013, pp. 880–886.

[11] Karumuri, Bharat K., et al. “Classification of Pre-Ictal and Interictal Periods Based on EEG Frequency Features in Epilepsy.” 2016 32nd Southern Biomedical Engineering Conference (SBEC), 28 Apr. 2016

[12] Kiral-Kornek, Isabell, et al. “Epileptic Seizure Prediction Using Big Data and Deep Learning: Toward a Mobile System.” EBioMedicine, vol. 27, Jan. 2018, pp. 103–111.

[13] Chollet, François and others. Keras. 2015. Software available from https://keras.io

[14] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., and Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E. Scikit-Learn. Journal of Machine Learning Research, vol. 12. 2011. pp.2825-2830

[15] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.