CS230

Epileptic Seizure Detection using EEG Data

Hannah Li and **Debashri Mukherjee**
Department of Management Science and Engineering
Stanford University
hannahli@stanford.edu, debm@stanford.edu

Abstract

Epilepsy affects 200,000 people in the US and is the fourth most common neurological disorder. It is typically diagnosed by conducting an electroencephalogram (EEG) to measure the electrical signals in the brain and then examining these recordings for abnormalities. The problem of reading reading EEG data is a costly and time-intensive process that must be conducted by highly trained neurologists, and even so there are often inter-doctor differences in these interpretations. We use deep learning techniques to create a classifier that takes as input EEG data and determines whether the patient was experiencing an epileptic seizure or not. For this binary classification problem, we train a four-layer 1D Convolutional Neural Net on 10350 one second EEG samples. On a test set of 1150 samples, our model attains a test set accuracy of .97, precision of .97, and recall of .99.

1 Introduction

Epilepsy is a common disorder that afflicts more than 200,00 people in the US every year. It is the fourth most common neurological disorder and affects people of all ages. It may occur as a result of a genetic disorder or an acquired brain injury, and is characterized by the occurrence of spontaneous seizures.

The primary method for Epilepsy diagnosis is by conducting an electroencephalogram (EEG), a test that measures electrical signals in the brain. Electrodes are placed in various locations on the scalp and record brain waves in these areas. A neurologist will then look at the waveform data to identify abnormalities. The process of reading EEG data is expensive, time intensive, and fraught with inter-doctor and inter-patient differences. Automating this process can help make an accurate diagnosis of epilepsy in a resource-limited setting.

This project aims to build such a system using Deep Learning methods on electroencephalogram (EEG) data. The EEG signal is time series of waveform data and the input to our algorithm is a one second long recording of an EEG recording, which corresponds to a time series of 178 numerical measurements. Each recording is also labeled with whether or not a seizure is occurring. We then use a Convolutional Neural Network (CNN) to take in this recording and output a diagnosis of seizure or no seizure.

2 Related work

The expensive and time-consuming nature of the process of reading EEG data has made this solution space not particularly novel. There is already a lot of work ongoing in this area with methods being suggested in both the traditional statistical modeling and machine learning space as well as deep learning.

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

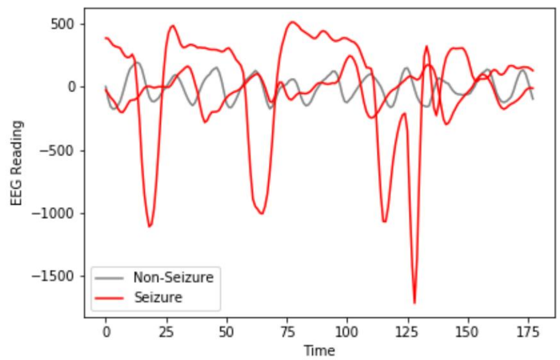

Figure 1: Example of EEG depicting seizure and normal EEG.

2.1 Using traditional machine learning models:

Nicolaou et al. [1] employed support vector machines (SVM) to classify data to detect epileptic seizures and achieved an accuracy of 0.93. They extracted the permutation entropy feature from EEG signals. Similarly, Rasekhi et al. [2] also used Support Vector Machines to classify epileptic seizures. Their algorithm uses the window size of seconds for univariate linear feature extraction. The similarity this study has with ours is the level of detail of the features.

2.2 Using Neural Networks:

A landmark study in this space was done by Swami et al., [3] who built a general regression neural network (GRNN) classifier that achieved perfect accuracy. They used sophisticated hand-crafted features like Shannon Entropy, standard deviation and energy derived from EEG data. This is currently considered state-of-the art for the problem space. Ullah et al. [4] built a pyramidal one-dimensional deep convolutional neural network (P-1D-CNN) in their study to detect epileptic seizures. Their study performed exceedingly well and achieved a mean accuracy of 0.99 in their classification. Their method was very well applied and translated well across the board for different observations. This study was the one that prompted us to try using a 1-D CNN for our solution. Another study was done by Guo et al., [5] using Artificial Neural Networks classifying the line length features that were extracted by using discrete wavelet transform (DWT). They achieved 0.97 accuracy. The study of the related works shows that neural networks tend to perform better than traditional machine learning models for this problem statement. All of these results urged us to use deep learning to find a potential solution to this problem space.

3 Dataset and Features

The dataset was obtained through the UCI Machine Learning Repository and was originally obtained by Andrzejak et. al. (2001).[6]

The dataset is a collection of time series EEG waveform data. Each EEG recording is a 1-dimensional vector of integer valued measurements. The data was taken from 500 patients, with a 23.5 second recording for each patient. The recording for each patient was split further into 23 one second chunks, and each chunk is one training sample in the data set.

In total, the dataset consists of 11500 training samples, and each training sample has 178 measurements. There are 5 different labels for each EEG, corresponding to different states of the patient when the EEG was taken and the location at which the EEG was taken. The five labels are eyes open, eyes closed, tumor identified but EEG taken from healthy brain area, EEG taken from tumor area, and recording of seizure activity.

We focused on binary seizure classification - we combined the remaining four labels as a “negative” label and labeled the seizure examples as “positive”. We had a total of 2300 positive samples and

9200 negative samples. We preprocessed the data by centering the data and subtracting the mean of all $11500 \cdot 178$ recordings across all samples and across all time from each of the samples. We found that this significantly increased the performance of our neural network.

4 Methods

We used a 1D Convolutional Neural Net (CNN) with L convolutional layers (more details to follow), followed by a fully connected layer with a final sigmoid activation function, since we were considering a binary classification problem. We used an Adam Optimizer with a standard learning rate of 0.009, trained on mini-batches of size 64, and used softmax cross entropy with logits as our loss function. The parameters we tuned (discussed in the experiments section) are the number of convolutional layers and the number of filters in each layer.

$$\begin{aligned} J(\theta) &= -\frac{1}{m}\left[\sum_{i=1}^{m}(1-y^{(i)})\log(1-h_\theta(x^{(i)})) + y^{(i)}\log h_\theta(x^{(i)})\right] \\ &= -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=0}^{1} 1\left\{y^{(i)}=j\right\}\log p(y^{(i)}=j|x^{(i)};\theta)\right] \end{aligned}$$

A single forward pass in our model follows the following model:

$$\begin{aligned} INPUT- > \underbrace{CONV1D- > RELU- > MAXPOOL}_{\text{L times}} \\ - > FULLY\ \ CONNECTED- > SIGMOID \end{aligned}$$

with the following specifications:

- Conv1D: stride 1, padding is "SAME"
- Max pool: filter size 2, stride 2, padding is "SAME"

The exact shape of the filters at layer l is a function of the number of filters in the previous layer $l-1$. To see a visualization of our model, see the appendix for the Tensorflow visualization.

5 Experiments/Results/Discussion

To decide our hyperparameters, we ran a random coarse search over the hyperparameters of 'number of layers' and 'number of channels'. We sampled values from a random uniform distribution between 2 and 6 for the 'number of layers' and between 4 and 16 for the 'number of channels'. These numbers were chosen based on the work done in [7] and [8]. After running more than 10 iterations of the random coarse search over the above hyperparameter space, the model achieved its best results with the following values-

Hyperparameter	Optimal Value
Number of layers	4
Number of channels	7-5-9-7

An interesting observation that was noted while performing the random search on the chosen parameters was that the number of filters had almost no visible impact on the cost function. For each of the values of the number of layers, the cost function only steadily decreased with each epoch. There was one anomaly noticed with one iteration that had 5 layers where the cost functions staggered and then did not improve. However, this was out of the ordinary and does not fall in line with the otherwise observed pattern. This leads us to believe that the number of channels in the layer prove to be much more important in determining the cost function and in turn, the performance of the model. For the other hyperparameters of learning rate for gradient descent, mini-batch size, we used the following values-

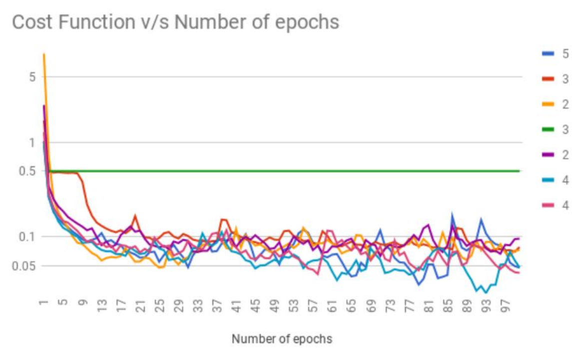

Figure 2: Cost function vs Epochs for varying model architectures

Hyperparameter	Value
Learning rate	0.009
Mini-batch size	64

In our first iteration of the project, we chose to use these values instead of iterating over them with the random coarse search.

We achieved the following confusion matrix with the chosen model on our test set-

n = 1150	Predicted: No	Predicted: Yes
Actual: No	TN = 222	FP = 23
Actual: Yes	FN = 4	TP = 901

To measure these results further quantitatively, we used the values of Accuracy, Precision, Recall and a combined F1 score. In imbalanced datasets, the goal is to improve recall without hurting precision. [9] With our final model parameters described above, we achieved the following scores on our train and test sets-

	Train set	Test Set
Observations	10350	1150
Accuracy	0.98	0.97
Precision	0.99	0.97
Recall	0.99	0.99
F1 Score	**0.99**	**0.98**

Judging by these values, the model doesn't over-fit the train set and performs very well on the test set.

6 Conclusion/Future Work

We can conclude that 1-Dimensional convolutional neural networks perform well in detecting Epileptic seizures using EEG data. They perform significantly better than traditional machine learning models like SVMs that achieve an accuracy of 0.93 on the same task [1] Furthermore, despite using raw, minimally-processed input EEG data, the model performance is close to that of a GRNN that classifies hand-crafted features like Shannon Entropy, Energy and Standard Deviation to achieve 1.0 accuracy [3] If we wanted to extrapolate this work further, this model could be useful for other similar classification problems based on EEG brain signals. A few pitfalls currently in the model is that it correctly classifies many seizure examples that resemble a non-seizure example while misclassifying some samples that appear to correspond to the start of a seizure. That can be noted in the following sample-

To further improve on this, we would perform error analysis specifically to reduce the false negative rate further- particularly focusing on the samples that corresponded to the pattern noted above. If we had more computational resources, we would perform a finer search on the hyperparameter space to

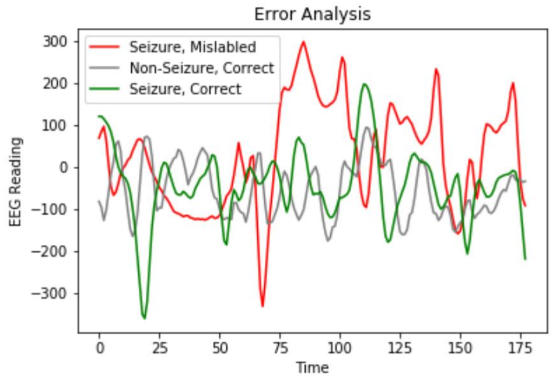

Figure 3: Samples of classified observations

tune the number of channels further. It would also be useful to search through random combinations of varying learning rate and mini-batch size values to find a model configuration that classifies with perfect accuracy. As a final step, we would look into implementing an online forecasting algorithm by training the neural network to recognize signals specific to the start of a seizure. The model currently can read and detect EEG data after it has been completely recorded and processed. However, a system which can detect the start of a seizure can be immensely useful to patients, giving them more time to prepare for the event. Such a system can potentially be made by modifying our model by focusing particularly on the wave forms that correspond to the start of an event.

7 Contributions

All of the data sourcing, cleaning, modeling and error analysis was jointly conducted by both Hannah Li and Debashri Mukherjee.

References

[1] Nicolaou, N. & Georgiou, J. (2012). Detection of epileptic electroencephalogram based on permutation entropy and support vector machines. *Expert Systems with Applications*, 39(1), 202-209.

[2] J. Rasekhi, M. R. K. Mollaei, M. Bandarabadi, C. A. Teixeira & A. Dourado, Preprocessing effects of 22 linear univariate features on the performance of seizure prediction methods, *Journal of Neuroscience Methods*, vol. 217, no. 1-2, pp. 9–16, 2013.

[3] Swami, P., Gandhi, T. K., Panigrahi, B. K., Tripathi, M. & Anand, S. (2016). A novel robust diagnostic model to detect seizures in electroencephalography. *Expert Systems with Applications* 56, 116-130.

[4] Ihsan Ullah, Muhammad Hussain, Emad-ul-Haq Qazi & Hatim Aboalsamh. (2018). An Automated System for Epilepsy Detection using EEG Brain Signals based on Deep Learning Approach *arXiv:1801.05412*

[5] Guo, L., Rivero, D., Dorado, J., Rabunal, J. R., & Pazos, A. (2010). Automatic epileptic seizure detection in EEGs based on line length feature and artificial neural networks. *Journal of neuroscience methods* 191(1), 101-109.

[6] Andrzejak RG, Lehnertz K, Rieke C, Mormann F, David P, Elger CE. (2001) Indications of nonlinear deterministic and finite dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state, *Phys. Rev.* E, 64, 061907

[7] Felix Achilles, Federico Tombari, Vasileios Belagiannis, Anna Mira Loesch, Soheyl Noachtar & Nassir Navab (2016) Convolutional neural networks for real-time epileptic seizure detection, *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization* DOI: 10.1080/21681163.2016.1141062

[8] U. Rajendra Acharya, Shu Lih Oha, Yuki Hagiwaraa, Jen Hong Tana & Hojjat Adelid (2017) Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals, *Computers in Biology and Medicine*

[9] T. Ryan Hoens & Nitesh V. Chawla (2013) Imbalanced Datasets: From Sampling to Classifiers, *Department of Computer Science and Engineering, The University of Notre Dame, Notre Dame, IN, USA*

[10] 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.

[11] Jones E, Oliphant E, Peterson P, et al. SciPy: Open Source Scientific Tools for Python, 2001-, http://www.scipy.org/ [Online; accessed 2018-03-23]

8 Appendix

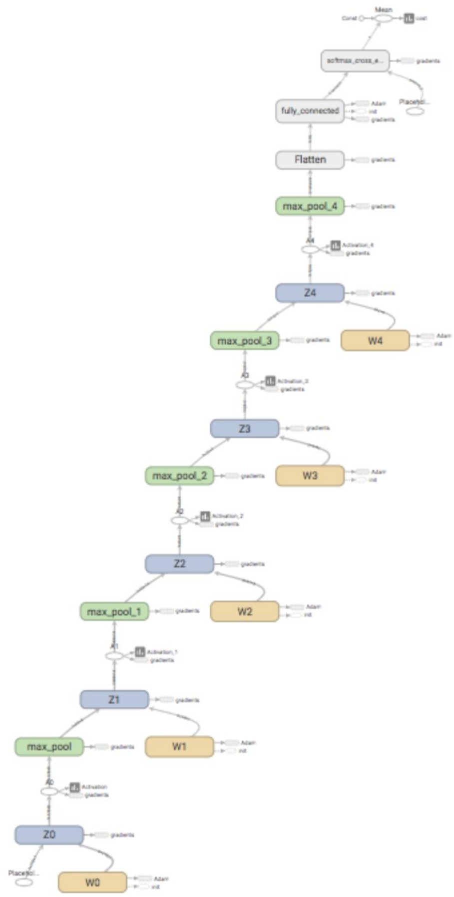

Figure 4: Visualization of the neural network architecture.