Processing EEG Data to Predict Seizures using Convolutional Deep Machine Learning Models

Final Report

Elijah Holmberg
Jack Baskin School of Engineering
University of California, Santa Cruz
Santa Cruz, CA
eholmber@ucsc.edu

John Baer
Jack Baskin School of Engineering
University of California, Santa Cruz
Santa Cruz, CA
jjbaer@ucsc.edu

Sruthi Jaganathan
Jack Baskin School of Engineering
University of California, Santa Cruz
Santa Cruz, CA
sjaganat@ucsc.edu

ABSTRACT

Our project uses Electroencephalogram(EEG) Readings and a Convolutional Neural Network to produce a machine learning based algorithm that can detect epileptic seizures. Using data from Children's Hospital Boston we constructed a function such that given a feature vector from 5 seconds EEG Readings we can make a binary determination of whether or not a seizure occurred in that time period.

1 Objective

Epilepsy is one of the most common neurological disorders in which brain activity becomes unusual, and is characterized by epileptic seizures. According to the World Health Organization, about 50 million people are affected by this disorder, and during their lifetime 1 in 26 people will be diagnosed with epilepsy. Detecting epileptic seizures as quickly as possible is essential, as patients that have sudden seizures are vulnerable to suffocation, death, or injury due to fainting and traffic accidents. No cure exists, and this disease is mainly treated with medications and surgery.

EEG (Electroencephalography) plays a main role in detection for epileptic seizures, as it records the electrical activity generated by the brain via electrodes, and provides frequency samplings from multiple locations within the brain. Detection, and labeling with EEG traditionally requires direct examination by physicians. The problem with this approach is that it can easily lead to error as is prone when people are tasked with combing through many hours of data. Furthermore, these traditional methods take a substantial amount of time and effort, while the patient is waiting in a time-critical state. A timely and accurate method of detection for epileptic seizures is essential to alert the patient, and the people around them alike.

In this project, we will explore the applications of Machine Learning in the field of healthcare, by using a convolutional deep machine learning model to detect epileptic seizures from EEG readings. Our main objective was for our model to be able to detect seizures with a greater than 95% accuracy, so it could potentially be used to help aid medical workers in diagnosing people with epilepsy faster, and with greater accuracy, even with a physician not being available.

2 Dataset

The dataset we used was a collection of EEG recordings taken at Children's Hospital Boston. These recordings were taken from 23 cases and collected from 22 subjects during their withdrawal from anti-seizure medication. 5 subjects were males ranging from age 3 - 22 and 17 were females ranging from age 1.5 - 19. Each case contains between 9 and 42 continuous .edf files from a single subject. However, because of hardware limitations, there were gaps between consecutively numbered .edf files, in which the signals are not recorded. While in most cases the gaps are 10 seconds or less, there were several recordings with larger gaps. There are 664 total EEG readings, 129 of which contain 1 or more seizure readings. Each file contains an hour of data which is made from 23 channels with a sampling rate of 256 MHz. The 23 channels were recorded in accordance with the International 10-20 system of EEG electrode positions and nomenclature. The signals were sampled at 256 samples per second with 16-bit resolution.

The initial method for importing these files was through Physiobank, the website hosting the database. They had produced a library called “wfdb” which was very difficult to use and had many bugs. Instead of using the provided deprecated system we chose to use a google cloud storage that had been created to import the files. This resulted in a significantly more precise and fast system which allowed us to load each case file individually.

For our machine learning model’s training data we initially wanted to create a massive set of 5 second epochs selected randomly from all files available per patient. This ended up being more than we were able to accomplish and so we chose to scale down to using specific files for training and testing. Since standard brain waves differ significantly from person to person training and test data had to come completely from the same patient. We chose to use the chb01_03.edf for training data and chb01_04.edf as testing data since these were the first two files from patient 01 that contain a seizure. Unfortunately, our model suffers from overfitting and requires a better training set to produce better results. This would be accomplished by randomizing the selection of epochs from all files available to each patient and then using mini batching to train to model efficiently. At the current stage of the project this would be the most important next step.

Classifying each epoch also presented issues. Our original approach was going to be condensing the entire file into a specific image and then classifying where in the image a seizure was occuring. When this failed to work we then had to classify epochs using text parsing. We settled on 5 second epochs as it seemed space efficient and each seizure lasted on average 20 seconds, providing us four epochs for every recorded seizure.

Every case file contains a “summary.txt” file which lists each edf file in the case file, how many seizures were in each edf file, and the starting and end timings of those seizures. The text parser would go through an available summary file and produce an array with ‘0’s denoting standard activity and ‘1’s denoting seizure activity for a given file. This generates an 720 entry array with each entry being the classification for its respective epoch. Since each file contains 3600 seconds of data and we needed to split it into 5 second we therefore get 720 classified pieces of input per file with the vast majority of them representing standard brain waves.

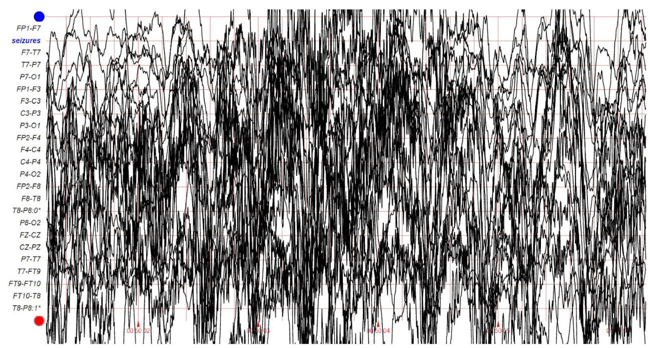

Fig. 1: Seizure Reading

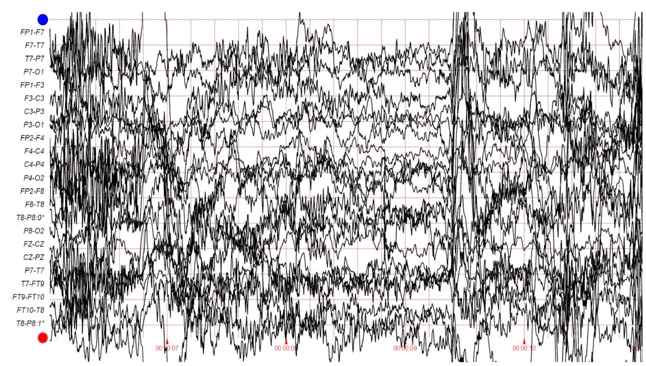

Fig. 2: Standard Reading

3. Data Processing

We tried multiple methods of how to shape the input to our model. The first option we pursued was direct input of images as shown in Figures 1 and 2. The main issue with this is that it would massively expand the size of the data set. Since we already have 50 GB of files converting every file into 720 images would nearly quadruple the space consumed by the data alone. In smaller batches this would be efficient but it was our goal to make the function work for all patients at once.

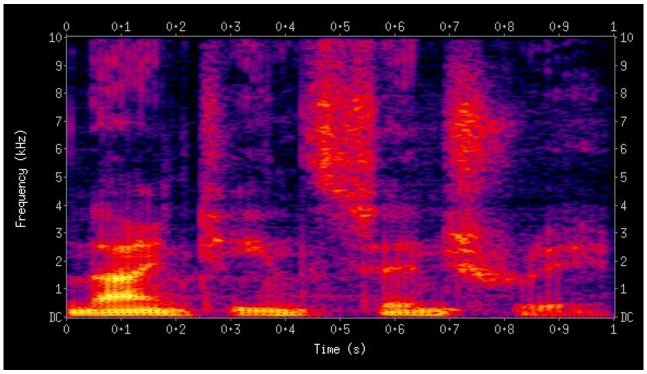

Fig 3: EEG Spectrogram

The second method we pursued for data input was converting the EEGs into Spectrograms. A Spectrogram is a condensed image of frequencies making it the perfect compact way to view long EEG Readings. Unfortunately, the only libraries for conversion of EEG to Spectrogram expect csv typed files as input instead of edf. We attempted to convert edf to csv and then to spectrogram and experienced significant loss of data. While this method would have been ideal for us, we were unable to enact it.

The final method which we used as our final product was the MNE library which converts all kinds of EEG readings into a distinct RAW data type with a plethora of built in functions. These functions allow the visualization of the EEG Readings directly as well as providing important information about the sampling frequency and channels. Using a built in get_data() function and numpy.reshape() we can convert the EEG Readings into a 720x256x23x5 numpy array which feeds directly into our model thanks to numpy's compatibility with Keras.

The massive avantage to this is that during a seizure every single channel experiences a massive increase in the amplitude of their frequencies meaning that we should see larger numbers across every channel making an easy distinction for our algorithm. That being said the hardest part of using this information is generating the training sets.

Initially we attempted to randomly select epochs from these massive arrays to generate batch training sets by splitting and sorting them by classification. Once sorted we could assemble an ideal training set but due to insufficient knowledge of python and the setbacks from needing to select different methods for data processing we ran out of time to make this work.

We did experiment with implementing preprocessing in these arrays in the form of condensing the frequencies temporally. This resulted in a smaller input array and faster training, but a decrease in accuracy and no change in loss. Spatial condensing across channels had similar effects and we therefore did not enact either. Remaining possible improvements could be done by rounding data values or squaring values to highlight sections with drastically increased values.

Since experiments with preprocessing did not bear fruit we then fed these full arrays directly into our model and used specific convolutional layers to achieve the best output we could.

4. Models and Algorithms

After several rounds of trial and error, and at the recommendation of our TA, we chose to use a convolutional neural network for our model. A CNN consists of an input and an output layer, as well as multiple hidden layers which work to apply a convolution operation to the input, before transferring the result to the next layer. Our hidden layers of the CNN consist of convolutional layers, pooling layers, and a flattening layer.

For our convolutional layers, we chose specific temporal and spatial filters as was recommended by the second referenced paper on deep learning. By using filters that are shaped specifically to our input, in this case 23x5, we can decrease the amount of neurons in the following layers and increase how much they are influenced by previous layers.

The pooling layers were a big advantage for CNNs, and were very valuable for this project, as our model required a high number of data that had to be processed. Through the pooling layers, the model progressively reduced the spatial size of the representation and the number of parameters in the network. We chose to use average pooling due to an anomaly that can occur while a human sleeps.

While a human sleeps during specific stages of REM sleep a human's brain can momentarily send signals that look extremely similar to seizures in certain channels. By using average pooling instead of max pooling we can avoid false positives and increase our accuracy. This is also addressed by our input time selection of 5 seconds. These anomalies generally last 2-3 seconds which means that our model will detect an equal amount of normal data and therefore classify correctly. The disadvantage to this is if a seizure ends halfway through an epoch it will not be detected.

In addition, using a CNN fit our project's goal the most as our project was mainly focused on feature vector recognition and classification, which CNNs are very successful at doing [Wang]. From reading the research from Ali Shoeb, who also studied how EEG readings are affected through the brain going through various activities, we mimicked the way he fed his data into the system [Shoeb].

We first extracted the spectrogram features from the EEG data, converting it into a 256 by 23 by 5 feature vector before feeding it into our convolutional network. For the networks, we also used Shoeb's model as inspiration, using 6 layers with kernels based on EEG channels.

The first convolutional layer is a Convolutional 2D Network for processing spectrograms, and takes in the EEG readings of 256 counts of 23 by 5 kernels. From there, our model average-pools the output, compressing the representations, and reducing the computational complexity of the network. Additionally we chose average pooling over max pooling because during a seizure, every single measure should increase in volatility, and we wanted our model to detect and flag a complete increase of EEG channels over the increase of several channels, which a max pooling layer may flag falsely.

Our third layer that the average pooling layer passes the output into is another 2D CNN network, with 32 filters. This is followed by another average pooling layer, reducing the complexity again. These recurrent layers help reduce overfitting. Finally, we filter this output through another 2D CNN with 64 filters, before outputting the result into the final average pooling layer. We further flatten this output data, before passing it into a Rectifier Linear Unit as a non-linear activation layer with a weight size of 256. The ReLU unit feeds into the sigmoid function, which we used for the output unit.

We used binary cross entropy as our loss function on the resulting model when training for the classification of the final destination, of whether or not a five second window is a seizure or not. The final result from running 40 epochs with a batch size of 720, was an average loss per epoch was 0.5, with our accuracy averaging 95%.

This was implemented with TensorFlow and Keras.

Fig 4: Model Accuracy

5. Results and Analysis

Through our project, we found that a CNN is an excellent model for processing EEG readings and machine learning can be a very good predictor for seizure detection. Even with suboptimal training sets we can generate above 90% accuracy across every file within a case when trained on a file with a seizure in it. This means that while the loss indicates heavy overfitting, the model is capable of accurate classification of seizure data.

After trying various time intervals for the model, we determined that using 5 second intervals lead to the most accurate results. While the previous experiment chose to use 2 seconds we found that we avoid more misclassifications of normal brain function readings and an equal accuracy of classification regarding seizures.

The undesired results we obtained was that our model was overfitting with the test data. As mentioned before using a mini batch approach and sampling data in the method described in section 3 we could most likely improve loss and potentially accuracy as well.

For our first project with machine learning we are very happy with our results and have created an excellent platform for ourselves to continue to work on and expand to something that could be very useful.

6 Future Work

In the future, there is excellent potential to improve the model through increasing the dataset size we are training our model on to include multiple patients' data.

It is entirely possible that our model trained on a single-patient's EEG data could misclassify another patient's epileptic seizure inaccurately, even if EEG readings for patients having seizures do tend to behave very similarly. We would like to experiment training and testing our model on readings from different patients and see how negatively this impacts performance

In addition, our current models are trained on one hospital's methods of taking EEG readings, which could conflict with other hospitals' readings. To improve our model's performance for general-use, we would like to expand our dataset to include other hospitals' publicly available EEG readings as well.

While we continue to learn more about machine learning and optimizing our program we hope to increase the efficiency of the program as well. Ali Shoeb's model was capable of classifying a two second epoch within three seconds, which we currently are incapable of doing.

While classifying and detecting epileptic seizures was our initial goal, throughout the project we realized how valuable it would be if there was a way to detect seizures in real-time through the EEG readings, from a small electronic device or otherwise. Real-time processing of EEG readings would greatly speed up classifications and detections of epileptic seizures, furthermore eliminating a great amount of time and effort that is currently being employed through physicians. But the most exciting prospect this could lead to, is the possibility of forewarning patients when they are about to suffer from a seizure. The biggest cause of harm to patients with epilepsy is the possibility of a seizure at a vulnerable time, leading to suffocation, death, or injury due to fainting and traffic accidents.

While we barely scratched the surface of machine learning, through this project we were able to see the great possibilities of using it to tackle real life problems. In this case we saw how it can be applied to address healthcare problems to solve a repetitive task that can otherwise be performed faster, and more accurately by a machine. We are very excited at how a determined effort can lead to the development of a robust, reliable, and low-cost alternative that could save lives.

7 Contribution

Elijah contributed by importing the data, installing mne, determining what shape data needed to be to feed to the model, generating the first model, researching how to improve the model, and implementing the now unused code for initial data importation and processing.

John Baer contributed towards helping to improve the model, parsing the input data into a usable format, and converting the EEG data into a pdf form to train the model from, which ended up not being used.

Sruthi contributed towards parsing the input data, dividing them into 5 second chunks, checking when seizures occur, and creating an appropriate numpy array based on the data. She helped with ideas, and debugging to make the accuracy better.

Everyone contributed toward the proposal, project report, final video, and final report.

References

Anon. 2020. Epilepsy. (2020). Retrieved March 18, 2020 from https://www.who.int/news-room/fact-sheets/detail/epilepsy

Deshpande, Adit.
"The 9 Deep Learning Papers You Need to Know About"
adeshpande3.github.io.

Dose, Hauke, Moller, Skadkær; Puthusserypady, Sadasivan, Iversen, Helle. "A Deep Learning MI-EEG Classification Model for BCIs" Proceedings of 2018 26th European Signal Processing Conference

Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. CNN-RNN: A Unified Framework for Multi-label Image Classification.

Shoeb. "Application of Machine Learning to Epileptic Seizure Onset Detection and Treatment." PHD Thesis, Massachusetts Institute of Technology, Sept. 2009.