Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Querétaro
Departamento de Computación y Mecatrónica

Epileptic Seizure Prediction with Raw Data from Electroencephalograms using Recurrent Neural Networks

by

Juan David Torres Velasco

Business Solution Development Capstone Project

Ingeniería

en

Sistemas Computacionales

Advisor: Dr. José Antonio Cantoral

Santiago de Querétaro, Querétaro, México
19/11/2021

"It was never easy to look into the future, but it is possible and we should not miss our chance."

Andrei Linde

Acknowledgements

This wasn't easy, I want to give a big thanks to Jose Antonio Cantoral, my advisor which was always open to help whenever I needed it and was always supporting me along the way.

I want to thank my mother and father Andrea and Camilo, that they have always pushed me to be the best version of myself and they have taught through actions that anything is possible as long as you put your mind and soul into it. To my sister that has always has have my back and calmed me when I needed it the most.

To Sofia which has always been my support and helped me pass through the stressful time of this investigation with her words of encouragement. To Salvador and Yvonne who have been supporting me all along and have made this investigation much more easier and lighter for me.

Abstract

Currently epileptic seizure prediction with machine learning requires previous processing of the data due to the various information gathered from the electroencephalograms and the noise from various frequencies. Keeping this in mind, the objective of this investigation is to be able to predict epileptic seizures using the raw data collected from EEG's using deep learning techniques without using pre-processing.

Approaches of epileptic seizure prediction were investigated, deep learning, Machine Learning and filters are all great but given the objective and the robustness of the problem the best way was to use RNNs. Recurrent Neural Networks were used to create the deep learning model given their properties and the advantages they bring when used with sequential data or time series data, but the model is nothing without data. Therefore, the MIT database was used in order to obtain the data before each of the seizures and to be able to use this data as the category of pre-ictal which is when there will be a seizure. There were different experiments conducted in order to obtain the best accuracy, the minutes of pre-ictal were changed, the data set cut in order to get more instances and architectures like GRU and LSTMs were used to be able to get the best result.

After various experimentation a 62.56% was obtained for the testing accuracy which is no that much and it is far from the objective. However, the results are promising and with the framework created to extract the data from the MIT data set it is just matter of time and tweaking of the model in order to obtain better results.

Contents

Abbreviations

CNN	**C**onvolutional **N**eural **N**etwork
DL	**D**eep **L**earning
EDF	**E**uropean **D**ata **F**ormat
EEG	**E**lectro**e**ncephalo**g**ram
GRU	**G**ated **R**ecurrent **U**nit
LSTM	**L**ong-**S**hort **T**erm **M**emory
MIT	**M**assachusetts **I**nstitute of **T**echnology
ML	**M**achine **L**earning
RNN	**R**ecurrent **N**eural **N**etwork

Chapter 1

INTRODUCTION

The brain is the main component of the central nervous system, it is the one in charge of sending signals to the whole body so it can function properly, therefore is a vital organ any human being. Epilepsy is a common brain disorder disease, in fact, it is the second most common after migraine, currently, 70 million people approximately have been diagnosed with epilepsy around the world [1].

It affects certain parts of the brain or the whole brain, the symptoms vary from patient to patient, but the most common ones are awareness loss, convulsions, and even loss of consciousness. Epilepsy seizure detection can be ground breaking for people suffering from this disease because most of the time the seizures happen without previous warning, and they can end up in physical injuries or even death [2]. Electroencephalograms (EEGs) are one of the most common ways to diagnose epilepsy, they are recorded through the use of electrodes connected to the part of the body it wants to monitor, in this case the brain and depending on the quantity of electrodes are the channels used, each channel then monitors an specific region of the brain. EEGs are currently the best way to gather information from the brain because they are non invasive and have a low cost compared to other alternatives like surgery.

EEGs are use to detect epilepsy but neurologists are the only ones able to understand the images and signals outputted by them. However, fortunately, machine learning is great at detecting patterns and extracting features, which means that with its use, people can be able to afford a cheaper way to epileptic seizure detection rather than going to a neurologist for a diagnosis given that 75% of people with epilepsy are from low/medium GDP countries [3].

There are three different states when it comes to epileptic seizures, the ictal phase which is when the actual seizure is occurring, the preictal phase is, as it's name states is the moment before the actual seizure, and finally the interictal which is the moment there is no seizure occurring.

The main purpose of this investigation was to find a way to predict epileptic seizure without previous processing of the data with recurrent neural networks using the preictal data as the category number one, and the interictal data as the category number zero. The MIT dataset was used in order to obtain the data required to train the model, and a script was used to separate, balance and export the data as a multiple dimensions array so it could be inputted to the model [2].

The recurrent neural networks are considered one of the best at time series prediction thanks to the architecture they handle, therefore the use of these deep learning technique was the best approach in order to predict effectively the seizure as the data of the EEGs are recorded through time.

Chapter 2

STATE OF THE ART

The main objective of this chapter is to give context of the current state of the art when it comes to epileptic seizure prediction and detection using technology and computers. The definition of different terms is primordial in order to have a better understanding of the current investigation.

2.1 EEG

Electroencephalograms, as it was mentioned before, are the cheapest way in which epilepsy can be detected as it is a none invasive test were it measures the electrical activity of the brain through a series of nodes connected to scalp. The signals are measured through time and they can be measured with different frequencies, but there can be recordings like white noises that may affect the measurements while trying to be interpreted. It is also important to mention that each person is different and one can have a different signal when suffering of an epileptic seizure than other, taking this into account detecting it with the use of EEGs is very difficult and only a specialist could detect the attack.

However, right now there are many application of Machine Learning in order to detect seizures and data sets that are labeled for it to be a way to help the people that don't have access to someone that can analyze EEGs. Deep Learning has shown better results in terms of seizure prediction and that is why it is important to inform on the two most used algorithms RNNs and CNN in the next pages. But first I will be showing how the actual data of an EEG looks like when there is an ictal

To illustrate better I am going to show a graph of all of the channels signals of one of the files of the data set that is going to be used throughout the investigation. The data shown in the figure 2.1 is from the patient number 1 and the file number 15 which includes a seizure. There the actual change to a seizure can be a appreciated and it can be seen how the blue line changes before and after the seizure. [4]

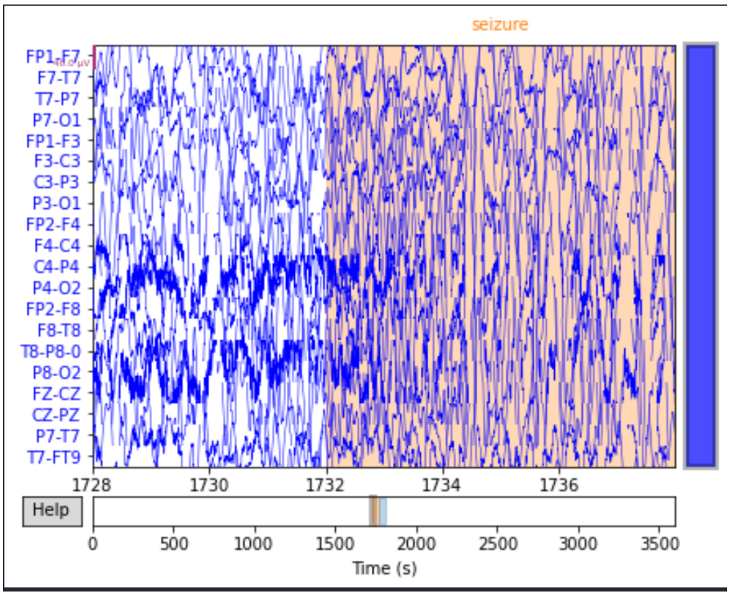

FIGURE 2.1: Patient 1 File 15 EEG data

It is also important to mention the three different seizure stages with illustrations. The three different phase of an epileptic seizure, are the ictal, inter-ictal and pre-ictal. The ictal and the pre-ictal states can be appreciated in the figure 2.2, it can be seen that pre-ictal is the previous segment of ictal just before the ictal starts.

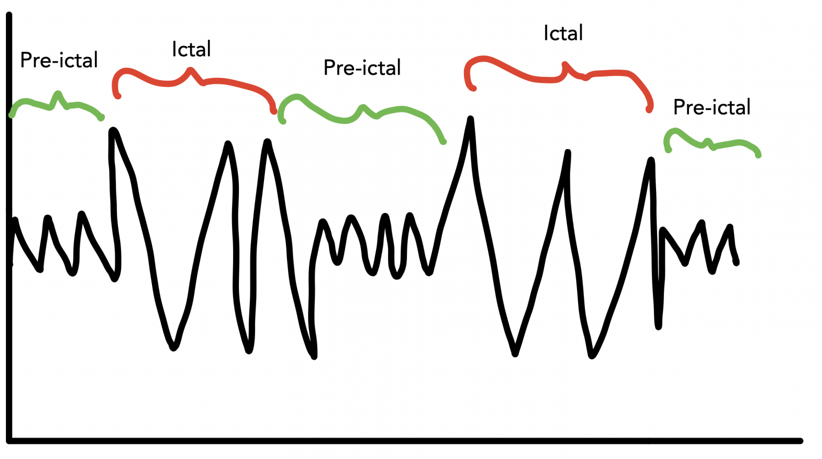

FIGURE 2.2: Pre-ictal and Ictal Illustration

On the other hand, the inter-ictal phase is characterized because it doesn't have any ictal before or after it, this is shown in the figure 2.3 where there are no ictals close.

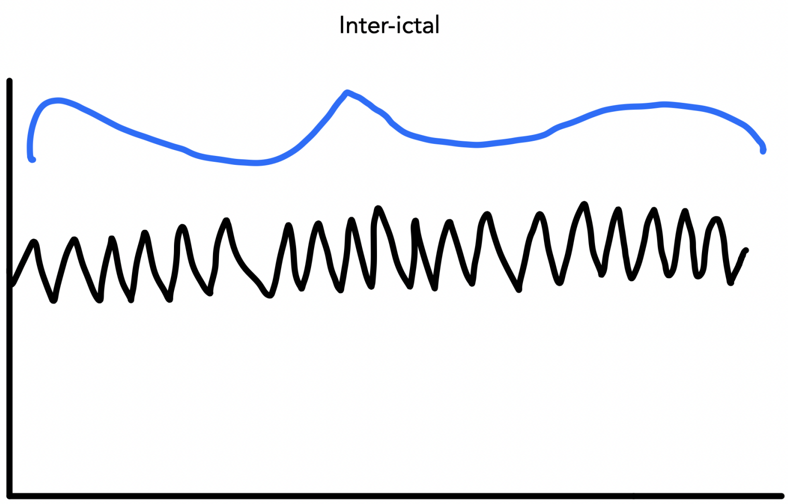

FIGURE 2.3: Inter-Ictal Illustration

2.2 RNN

Recurrent Neural Networks are type of an artificial neural network and these use sequential data or time-series data. Most of the applications of RNN are natural language processing, translation, speech recognition. They are mostly characterized by the memory they have, mainly because they use information from prior inputs to influence the current input and output. This means that the result of the model is determined by the previous data of the sequence. In the figure 2.4 it can be appreciated how the recurrent neural network looks unrolled, as it can be seen the output of one of the layers is then used as an input for the next one and so on, each one of this single layers is also known as a time step a fundamental concept that will be mentioned in the next pages [5].

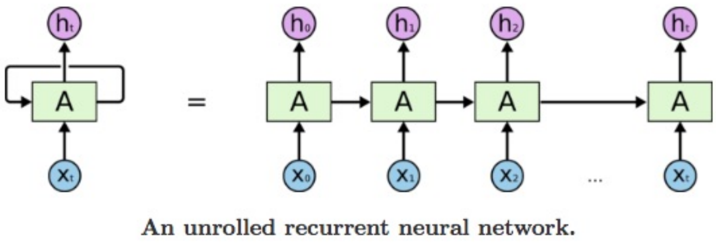

FIGURE 2.4: RNN Unrolled from [6]

I will be describing three types of RNN that are characterized by their unique architecture although they are all based on the same principle. The simple RNN has the characteristic that is a fully connected RNN where the output of the previous is fed back to the input of the next one, basically what was described previously [7].

Secondly, LSTM or Long-Short Term Memory Network is characterized by its longer memory when given to a data set with a very large number of time steps given that the simple RNN had short term memory and they had trouble transporting information from early time steps. The flow of the information is controlled with the use of gates, saving only a sequence of necessary data and trashing the unneeded ones [7].

Thirdly, there's the GRU or Gated Recurrent Unit which is a variant of the LSTM, it combines the input and forgetting gates into a single update gate. This means that the gates are decreased to two which becomes more efficient in the long run [7].

Like any other neural network, after the layer of LSTM or GRU or any other RNN, an activation function is needed to be able to predict and get to the final result. One of the most common activation functions is the Sigmoid function which is great at predicting when the resulting categories are binary, meaning that it takes the value of 1 or 0. This can be illustrated in the figure 2.5 in which the function is mostly or on the zero or on the one and when it is in the middle it is very difficult for it to stay at 0.5. [5]

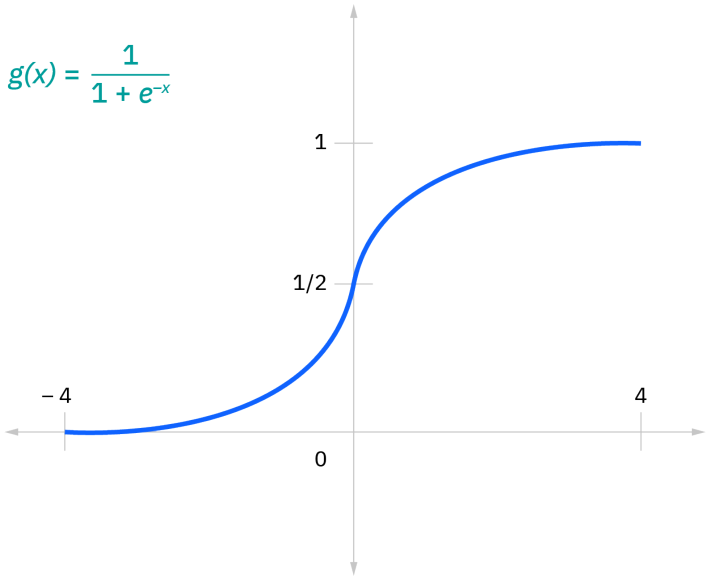

FIGURE 2.5: Sigmoid Function from [8]

2.3 CNN

Convolutional neural networks were introduced firs in the 1980s by Yann LeCun, a computer science researcher, but because of their need for a lot of processing power they couldn't be scaled and therefore weren't used a lot. In 2012 a convnet called AlexNet evidenced in an ImageNet challenge that the processing power was now able to do deep learning and that maybe it was time to use convnets more. Since then, researchers have been using them to perform computer vision tasks. Convnets are mostly used on image and video models because of their ability to detect features and patterns from visual representations.[9]

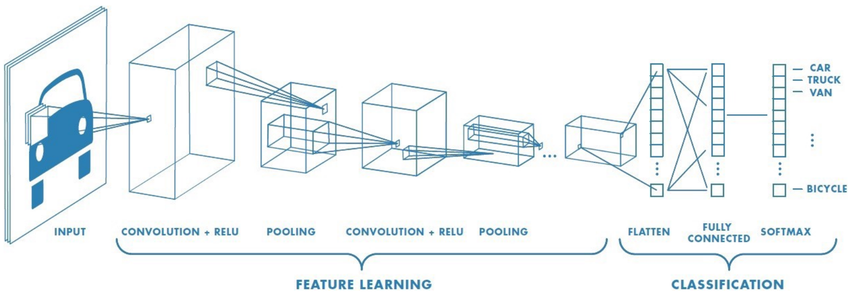

FIGURE 2.6: Convolutional Neural Networks Visual Representation from [10]

As it can be seen in the figure 2.6 The first layer is usually the input and to this layer as well to the next feature learning layers a kernel is applied as a sliding window for each of the pixels in the image as shown in the figure 2.7 where the K multiplies the I and the result is then the dot product of these numbers. The result of this convolution process is then activated by an activation function, usually ReLU known also as the Rectified Linear Unit for image. The resulting layer then will be smaller than the one before and a maxpool layer is applied then to decrease the size of the resulting layer, this process can be done multiple times. After feature learning layers are done, the classification layers are used first to flatten the result layer, then to create a neural network and then using an activation function for the classification. For categorical values the SoftMax function is used and for binary categories the Sigmoid function is used. [2]

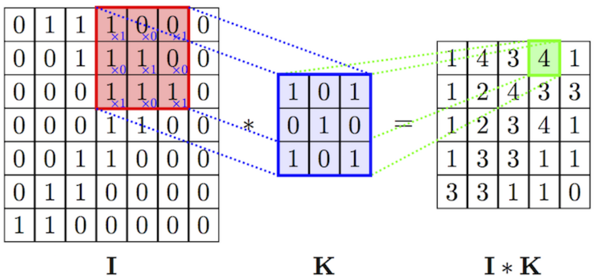

FIGURE 2.7: CNN Process from [11]

2.4 LITERATURE REVIEW

There are many different methods developed to detect seizures with EEGs. The feature extraction of the EEGs is key when working with machine learning and improving its performance, there are a vast majority of feature extraction methods in the literature. These methodologies use features in the time domain, frequency domain, or both at the same time-frequency-time domain [3].

In [2] a conventional machine learning approach was used for the automation of seizures detection, to do so it was necessary to create a machine learning framework and identify the important features necessary to classify a seizure from other activities done by the brain, a 96% of accuracy was reached during the investigation which is significant.

On the other hand, using a similar of machine learning approach in [12] they were able to get to a 99.5% of accuracy, the feature engineering process was done through several components. First, the feature extraction through time, frequency and time-frequency domain using a series of filters like Butterworth filter. Furthermore, feature selection using the statistical methodology t-test and sequential forward floating selection. But before these was done, the Support Vector Machines and K-Nearest Neighbors algorithms were used to pre-process the signals measured so they could be pass by the feature extraction process. The K-Nearest Neighbor algorithm was also used in [13] together with the Fuzzy Rough Nearest Neighbor algorithm, they focused on discriminating and detecting the most relevant features that could help with the classification process applying traditional machine learning algorithms, this model was able to segment EEG signals into ictal and interictal classes which is during a seizure or between seizures respectively.

The conventional machine learning-based models have the problem that they are very sensitive to small changes in the signal patterns because EEGs are non-stationary and can change between different people. Additionally, these are also susceptible to the body's normal activity like muscle movement, eye blinking, and even to white environmental noise. For example, medium noise can decrease by 10% the accuracy of the seizure detection.

However, the use of deep learning could be a great way to address this kind of issues because it is able to discriminate data from EEGs, this is mostly to relate the data samples as a sequence. In [3] a Long Short-Term Memory is used to detect at a high level the EEG patterns for classification. In the end, a SoftMax function was used for training and classification. A similar accuracy was reached with this approach, getting to a higher accuracy than 90%, but with the upside that no pre-processing was needed. Deep learning was used as well in [7] to automate the process of the extraction of features and classification due to its better performance as well as the robustness that makes it better when noise is present, the downside that is mentioned is that more data and time is needed because the models are more robust

In addition to the use of deep learning, there is another machine learning approach that is very well known to be used for image classification and it is the use of convolutional neural networks, which is what in [14] was done. Image representations of the EEGs or spectrogram were used in order to get the patterns of an epileptic seizure without the need for a pre-selection of features, but the downside of this technique is the computational performance that is needed to convert the signals outputted from the EEG to images.

Chapter 3

PROBLEM STATEMENT

The objective of this chapter is to inform about the problem that was tried to be solved in this investigation. As it was said before in this document, 75% of the people with epilepsy come from low/mid countries and if this people have no resources having a way in which they could monitor their health without the need of an expensive specialist would be revolutionary [3].

Currently, the use of machine learning and deep learning to detect epileptic seizures is advanced but it uses previous processing to filter any noise from different frequencies because the ML models are very sensitive and they can highly affect the result of the prediction. Therefore, making the prediction process longer as it has to filter out a series of noises that in the long term will take a long time to if it was intended to be used in a real-time seizure detector. The use of convolutional neural networks neither can solve this problem because the conversion time needed for an EEG to transform into an image can get very high and expensive.

The use of EEGs which is the cheapest solution in terms of monitoring brain activity compared to other methodologies that need surgery. EEGs together with a deep learning model able to detect previously that the seizure is going to occur could be life changing for the people suffering of epilepsy.

Epilepsy can happen suddenly and could even cause death if it isn't know with previous knowledge that a seizure is going to strike, for example if someone is cutting something in the kitchen and suffers from an attack, the result could be devastating, but if it can be known it will happen before hand it could change everything and save many lives.

Chapter 4

PROPOSED SOLUTION

This chapter is mainly to inform the proposed solution on how to solve the issue with epileptic seizures and be able to explain in big picture how it is going to be implemented.

Taking into account the problem, it is clear that the best approach to improve epileptic seizure detection is through deep learning algorithms that thanks to their robustness and their architecture they can detect patterns through different layers which makes it better when there is noise present. Therefore, to continue with the investigation the solution is to use Recurrent Neural Networks which are mostly used for sequential data or time-series data which is just what EEGs are, through the use of this RNNs there will be no need to do previous processing to the signals.

The papers mentioned in the literature review that used deep learning algorithms for the training of their model showed great results that can be achieved even with the noise present in the data this is why the proposed solution will be done through this methodology.

4.1 EXPECTED CONTRIBUTIONS

Even though there have been already many deep learning approaches that address this problem there are very few that have used absolutely no pre-processing which means that there is an area of opportunity to investigate how this could be done with deep learning. The objective of this investigation is to create a deep learning model using Recurrent Neural Networks with an accuracy higher than 70% to detect epileptic seizures without previous processing of the EEG data.

Chapter 5

METHODOLOGY

The main goal of this chapter is to describe the whole methodology that was applied in order to train the Recurrent Neural Network model so it could be able to predict all given an EEG data from a patient whether it would have a seizure.

5.1 DATA SET

To do so firstly as in any Deep Learning model the data is key and the data set used was the MIT CHB-MIT Scalp EEG Database [2]. The database contains 24 recordings in total, each recording has its own duration and quantity of seizures, these contain between 9-42 continuous EDF files and these EEG measurements were sample at 256Hz, this means that for every second 256 records are registered per channel. However, given that almost each recording had different channels it was important to gather the most channels that most had in common and that is why the channels used were the following: *"FP1-F7, F7-T7, T7-P7, P7-O1, FP1-F3, F3-C3, C3-P3, P3-O1, FP2-F4, F4-C4, C4-P4, P4-O2, FP2-F8, F8-T8, T8-P8-0, P8-O2, FZ-CZ, CZ-PZ"* as it is suggested in [15] and this channels are distributed as it is shown in the figure 5.1.

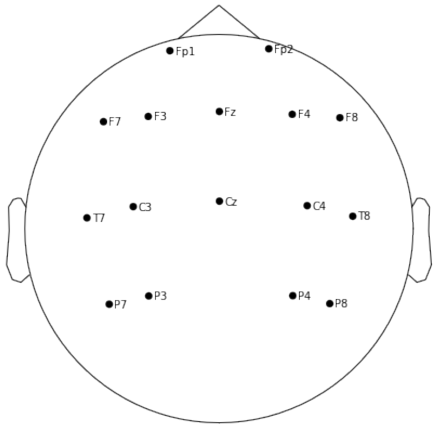

FIGURE 5.1: Scalp plot with channels used

This data set was recorded from 22 subjects, 5 males of ages of 3-22 and 17 females of ages 1.5-19. Additionally, there was another record from a female obtained 1.5 years after the first ones were made and the 24th recording isn't included in the subject info and that is why it wasn't used for the training, test and validation of the model. From the 23 patients it is import to inform that there are 685 files with an average of one hour per file, meaning it is a very challenging data set in which most of the time of the investigation has to be spent reorganizing the data in order for it to be able to be used in the model.

It is worth to say that the data set was measured through a continuous scalp Electroencephalogram, each EDF file mostly containing one hour long. Each patient's record has it's own text file that describes each EDF file with information like file name, file start time, fire end time, the number of seizures in the file and the starting and ending time of the seizure in the file.

5.2 MODEL CREATION

5.2.1 DATA COLLECTION

As it has been mentioned previously, the objective of this model is for it to be able to predict with anticipation if the patient is going to suffer from an epileptic seizure.

Keeping this in mind the data that is needed is the pre-ictal as a category and the inter-ictal as the other category which will conform the data that is going to be used for training, validation and testing.

In order to be able to create the RNN model it was first very important to be able to export this files in a way in which all of the data was separated between pre-ictal and inter-ictal data. This was done through the use of the text files described previously that each of the patients records have. Additionally, EDF file's data was converted into multiple dimension Numpy arrays using a Python library called MNE which works great when reading these file types.

Firstly, it was important to obtain all of the pre-ictal data first, mainly because it was very important to balance the quantity of pre-ictal and inter-ictal data, given that there were more recording and files with inter-ictal data and the model could have been biased or affected. Therefore, it was important to gather all of the files in which there was a seizure, get the exact moment the seizure started and obtain the data needed before the seizure, depending mostly on the defined time needed of pre-ictal, which in this case was the same as the one from inter-ictal. If there was a case in which the file didn't have all the necessary information, then the script was able to get the file before hand and get the missing records.

Once all the pre-ictal data is gathered, then it was important that when the inter-ictal data was collected none of the files used for pre-ictal were used, that is why in the pre-ictal data the list of all of the files used are previously saved so aren't used later. Afterwards the next step was to get the same time used for the pre-ictal data and given that the exact moment of record extraction isn't needed, the first records were obtained depending obviously in the minutes needed.

The way in which the exact amount of records needed to be collected pre-ictal or inter-ictal instance was calculated through the use of the following formula:

$$RecordsPInstance = PreictalTime * SixtySeconds * SamplesPerSecond * NoChannels$$

The structure of the two files that concentrated the records for pre-ictal and inter-ictal is the following *[NoChannels, (RecordsPInstance/NoChannels) *NoInstances]*. Meaning that the whole data could be describe as a table were the columns are composed by each single record and the rows are a single channel, these can be better illustrated in the figure 5.2.

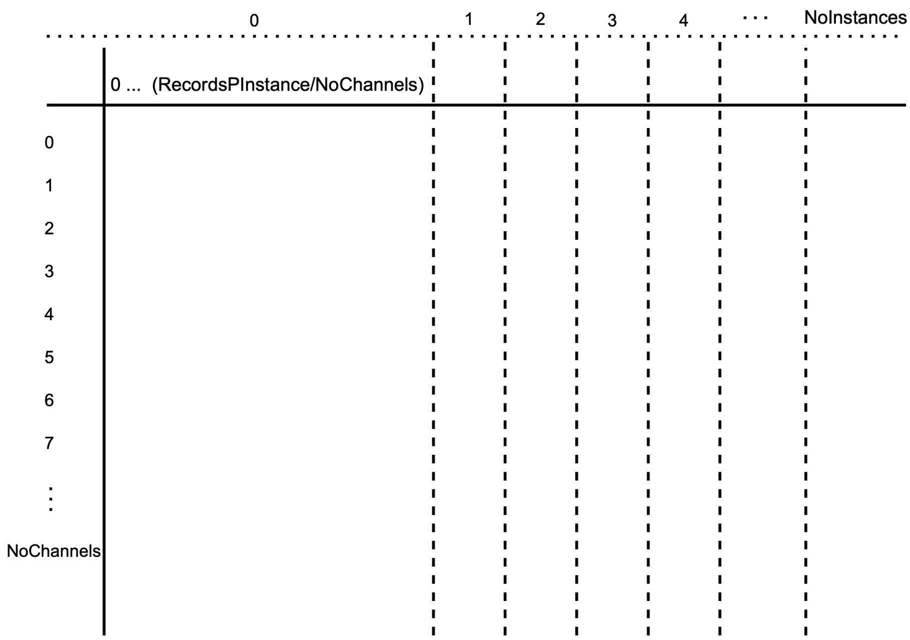

FIGURE 5.2: Data structure after the collection of the pre-ictal and inter-ictal data

5.2.2 DATA PREPARATION

Once the data was structured properly it had to be reshaped because of the RNN time steps, this was done through the use of reshapes and transposing when necessary so that that the data was correctly structured in the way the recurrent neural network needs it. The structure is the following [NoInstance, NoTimeSteps, NoChannels*SecondPTimeStep * 256].

Once it was reshaped, the classes were created, for the pre-ictal data an array of ones of the size of the data was created and the same thing was done for the inter-ictal data but with an array of zeros. Both arrays were put together as well as the pre-ictal and inter-ictal data. Afterwards random numbers from 0 to the size of the data were created without repetition and the classes and data were reorganized so that when they were used for training it the model wouldn't be biased.

Then the data and classes were separated into validation, training and testing. The validation and testing had both 5% of the data and the training was using a 90% of the whole data.

5.2.3 MODEL CREATION

The model will have a Sigmoid activation function given that there are only categories which are if an ictal is going to occur or not. The optimizer that will be used is the Adam optimizer and the loss will be measure through the binary cross-entropy. Once the model was trained, the model will be tested.

As it was mentioned previously Recurrent Neural Networks will be the deep learning algorithm used to train the model in order to get the results proposed, given that they are very good with time-series data given that they have memory which is a way the model can learn from previous input and with this the noise model can be addressed because it will able to detect and learn from input.

I will be trying different RNN layers, SimpleRNN layer which is a fully connected RNN, and LSTM that has longer memory and GRU which works great as well with long memory. The structure I will use is 12 timesteps always as I won't have time to change the number of time steps when testing

Chapter 6

Results

This chapter is intended to showcase the results of the model, its accuracy in testing, validation and training and describe each of the different results given from the variations of models tried and data in order to get to the final result.

6.1 First Approach

The first approach of the the model was using a single 64 unit SimpleRNN layer with its corresponding activation function, optimizer and binary crossentropy. The actual data used firstly had 298 instances this is because in the data set there were 148 seizure from the 23 patients and given that the pre-ictal data has to be taken before an ictal as its name states. This first approach was done with 10 minutes of pre-ictal and the results of this approach were great in training but very poorly in validation and testing, the training accuracy was of 100% but the validation accuracy obtained was of 36.67% with a validation loss of 3.4265 which meant that the model was over-fitting. The data can be seen in the table 6.1.

Model	Intances	Preictal	Timesteps	Train Acc	Val Acc	Val Loss	Test Acc
Simple RNN 64	298	10	12	1	0.3667	3.4265	0.4

Table 6.1: First approach results

6.2 Data Augmentation

Seeing that the model was overfitting, which means that it was going great with the training data but when it was exposed to data out from the one it knows it did badly (like the validation data). One of the best ways to solve over-fitting is through the use of data augmentation and use dropout layers in order to have less of an over-fit.

Having this in mind, and given that 298 data is too little for a problem this complex the best way for the actual training to go better was to increase the instance while maintaining the same layers of the model. The number of instances was increased by using the same amount of minutes in pre-ictal but cutting them into sequences of 2 minutes. By doing this the amount of instances increases by approximately 500% which means, the total instances were now converted into 1341.

The downside of doing this, is that some of the data will have a much more pronounced pre-ictal data if its a pre-ictal data, which means that data will vary much more and it will be more difficult for the model to predict with a high accuracy. But I think that making this will have more positive impact than negative.

6.3 1341 Instances and SimpleRNN

With the number of instances increased and the same layer of SimpleRNN the validation and testing accuracy were both of 50% which still wasn't enough because it is the same as tossing a die, and the validation loss was still high, as it get to a 3.15. This meant that the approach of increasing the instances worked, however they weren't enough yet, which meant that maybe a better approach was to decrease the time cutted to 1 minute which translated into doubling the number of instances.

6.4 2682 instances and SimpleRNN

With a higher number of instances but maintaining the pre-ictal minutes the validation accuracy increased significatively to 57% and the testing accuracy was increased to 0.55 as well. Additionally, the validation loss was decreased to 1.1 which is great for the results. Given that now the the model was doing much better it would be a good idea to try an LSTM layer instead.

6.5 2682 instance and LSTM

After replacing the SimpleRNN to an LSTM layer the validation acurracy decreased to 54.36%, obtaining a 0.9007 of validation loss and a 72.37% accuracy for the training . This showed that given that the timesteps were 12 of 5 seconds each it meant that an LSTM was not necessary for the data. However, I still tried with GRU which is also great for large amounts of data per time-step which in my opinion was not the case.

6.6 2682 instances and GRU

By changing the LSTM layer to a GRU layer the validation accuracy go to the exact same one from the LSTM which was 54.36%, with a 1.0008 of validation loss and a 82.51% of training accuracy. This means that a change in the layer for a better memory is not necessary for this quantity of time-steps and the data after being cut down. This also means that the best approach will be to use a SimpleRNN. Additionally I wanted to try with a higher pre-ictal time and that is why I will be showing the results for a pre-ictal of 16 minutes.

6.7 4292 instances 16 minutes of Pre-ictal

I tried to work with a higher minutes of preictal given that this would increase greatly the amount of instances maybe increasing the accuracy of the model, and indeed, the actual validation accuracy increased to 58% and getting to it with a 0.97 of validation loss, a training accuracy of 85% but with a testing accuracy of 56% which mean that maybe the increase in pre-ictal didn't have as much of an impact as I thought.

6.8 Model 2 2682 instances

Trying to create a new model to get a better accuracy I first added a SimpleRNN with 32 units, a dropout layer of 0.2, a SimpleRNN layer of 64, a Dense layer of 32 units with Relu, the Sigmoid activation function and the final layer with the optimizer and binary crossentropy for loss calculation.

Then with the same pre-ictal of 10 minutes and broken down into small parts of 1 minute each which was the better performing data set I got 53.69% at validation accuracy, validation loss of 0.9736 and a training accuracy of 81.95% which is worst that the single layer of the simple rnn which means the second model wont work quite well. That is why I will be trying a new model.

6.9 Model 3 2682 instances

This new and final model had firstly a 64 unit of SimpleRNN, another layer of SimpleRNN but with a recurrent dropout of 0.2 which means that every certain time 20% of the neurons are dropped. Then I added a Dense Relu layer with 64 units, a Dropout layer of 0.3, a learning rate of 0.0001, the Sigmoid activation function layer and finally the optimizer with the binary cross entropy function to measure loss. When the model is trained there's a callback that is used, which is in charge of reducing the learning rate on plateu of the validation loss in a factor of 0.2 with a patience of 5 and a minimum of learning rate of 0.00001

This model achieved a 61.74% of validation accuracy, with a validation loss of 1.0774 and a training accuracy of 89% with a testing accuracy of 62.56% which means that is the best performing model from all. I will try the pre-ictal of 16 minutes which seems to have performed well with the first model used.

6.10 Model 3 4292 instances

After using the previous model with the data set of the 16 minutes of pre-ictal the results were the following: A validation accuracy of 58% was obtained with a validation loss of 0.78 which is the lowest of all, the training accuracy was of 85.4% and the testing accuracy obtained was of 60% which is great as well for this results. This was the most balanced result of all as it had a low loss and the highest validation and testing accuracy. The loss of the model can be shown in the figure 6.1 and the accuracy can be illustrated in the graph 6.2. In the accuracy

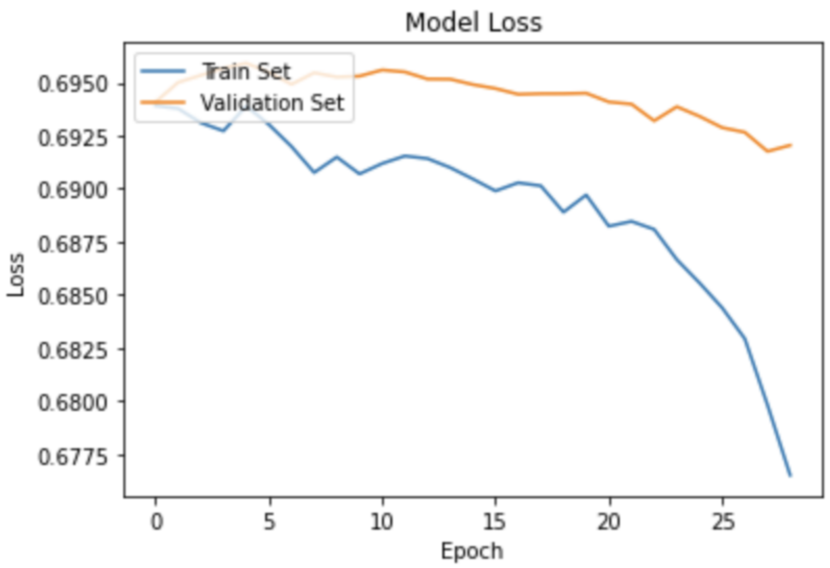

Figure 6.1: Model 3 Loss with 16 pre-ictal and 1 minute cut

It can be seen that in the model loss the training and validation kept decreasing but there is a point in which the actual training set decreases very quickly while the validation set it is still high, this can be explained by the data augmentation that was done through the cutting down of the whole data, as the training set may vary between each of the instances due to the difference between each of them.

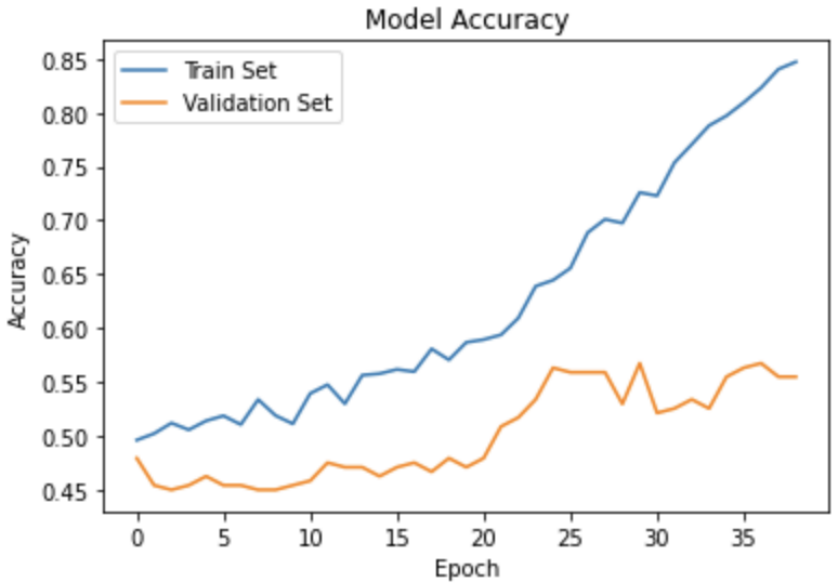

Figure 6.2: Model 3 Accuracy with 16 pre-ictal and 1 minute cut

In the model accuracy it can be illustrated that although in both cases the validation and training set are increasing gradually, there is a point where the validation accuracy stays flat while the training accuracy keeps increasing quite rapidly. This can be explained as well because each of the instances may have a variation due to the cut down.

However, in my opinion a better accuracy can be achieved if the model if more changes on the model are done, as well as the changes in the time-step of the data and the pre-ictal data is changed.

Chapter 7

CONCLUSION

In conclusion the Recurrent Neural Network model was greatly improved from the first approach with the cutting down of the pre-ictal time into smaller pieces, it went from 36.67% to a 57% with any changes in the model. Additionally, with model changes the actual accuracy went up to 62.56% which is a great improvement from the single layer of SimpleRNN. However as I mentioned before, the cutting of the data must have affected significantly to the actual accuracy and loss of the model, mainly because each instance has a different signal if it is from a pre-ictal model given that the signal 1 minute before an ictal is significantly different than from a signal 10 minutes before an ictal. An not only this but the actual validation data has the same problem, which means that at the end when trying to predict, you can't recognize from which part of the pre-ictal the data is coming from.

Having the objective of the investigation in mind, although the accuracy achieved of the investigation was of 62.56% and the objective was to get it higher than 70% the most difficult part was done, which was the manipulation of the data and prepare them for the actual model.

So on one side, the whole script or framework created to gather the data from the MIT data set was a great accomplishment and a huge contribution given that with this code the only thing needed in order to improve is to obtain the data in a different way and concentrate in improving the actual model using new techniques or changing the approach like using the transformer architecture. The actual investigation has a bright future and the application in my opinion could be life changing

The applications of the actual investigation, concentrating mainly on the non pre-processing of the data is pretty revolutionary because only having a device that can tell you with 1 minute of anticipation that you will be having an epileptic seizure can be groundbreaking for the people suffering from this disease. For example, let's imagine a person that suffers from epilepsy is driving and it gets a notification that he or she is going to suffer from a seizure, they can act very quickly and go to the other side of the road in order for them to get prepared and with this preventing a tragic accident.

Additionally, as it was mentioned in the beginning just having this in a hospital were there is no access to an specialist could be great because with the deep learning the hospital with the EEG could be able to know whether the person will be having a seizure or even just to diagnose if someone has epilepsy.

I hope in a future an investigation can use the framework I created and hopefully be able to improve upon the model which is as for right now the most important aspect to be able to improve the accuracy and in a future have a real time analyzer that can detect for people when a seizure is coming before something tragic happens.

Bibliography

[1] Greg Rogers. Epilepsy: The Facts.

[2] Ali Shoeb and John Guttag. Application of Machine Learning To Epileptic Seizure Detection, July 2009.

[3] Ramy Hussein, Hamid Palangi, Rabab Ward, and Jane Wang. Epileptic Seizure Detection: A Deep Learning Approach, March 2018.

[4] U. Rajendra Acharya, S. Vinitha Sree, G. Swapna, Roshan Joy Martis, and Jasjit S. Suri. Automated EEG analysis of epilepsy: A review. *Knowledge-Based Systems*, 45:147–165, 2013.

[5] IBMCloud. Recurrent Neural Networks, September 2020.

[6] Christopher Olah. Understanding LSTM Networks, August 2015.

[7] Afshin Shoeibi, Marjane Khodatars, Navid Ghassemi, and Mahboobeh Jafari. Epileptic Seizures Detection Using Deep Learning Techniques: A Review.

[8] Tali Laibovich-Raveh and Daniel Lewis. *A new method for calculating individual subitizing ranges ER*. June 2018.

[9] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6, 2017.

[10] Sumit Saha. A Comprehensive Guide to Convolutional Neural Networks, December 2018.

[11] Marc Chaumont. Deep learning in steganography and steganalysis. pages 321–349. January 2020.

[12] Marzieh Savadkoohi, Timothy Oladunni, and Lara Thompson. A machine learning approach to epileptic seizure prediction using Electroencephalogram (EEG) Signal.

[13] Aayesha1, Muhammad Bilal, Muhammad Afzaal, Muhammad Shuaib, and Muhammad Fayaz. Machine learning-based EEG signals classification model for epileptic seizure detection, January 2021.

[14] Catalina Gomez, Pablo Arbelaez, Miguel Navarrete, Catalina Alvarado, Mario Valderrama, and Michel Le Van Quyen. Automatic seizure detection based on imaged-EEG signals through fully convolutional networks, 2020.

[15] Luciano Bongiorni and Alexandre Balbinot. Evaluation of recurrent neural networks as epileptic seizure predictor. *Array*, 8:100038, 2020.