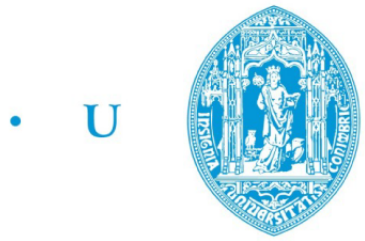
Preventing and Detecting Epileptic Seizures

Francisco Jesus 2020140671
Pedro Louro 2020173394

November 2020



FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Teacher: António Dourado

Index

1 Introduction

With this project, we intend to develop neural network models for prevention and detection of epileptic seizures.

Using the supplied data of recorded brain activity of patients we were tasked with the development of four different neural networks.

2 Data Set

The data set provided consists of recorded brain signals from four patients. Each patient brain signals were extracted from EEG (ElectroEncephaloGram), resulting in 29 features per second of recorded activity corresponding to different frequency bands, from 0.5 to 512Hz. For this work, only two patient's data were considered.

3 Performance

In order to measure the performance of the implemented architectures for the neural networks, the sensitivity (number of true seizures predicted or detected, meaning a greater value means high sensitivity) and specificity (number of false seizures predicted or detected, meaning a low number means high specificity) were calculated for each architecture.

4 Neural Network Architectures and Results

4.1 Pre-Processing - Multilayer

For this section we created a script called SetCreation.m. In this script the data from the patients is processed to be ready to be used on the neural networks for training and testing. To this end there was a need to transpose the features matrix and change the data from Trg to be with 1 for interictal, 2 for pre-ictal and 3 for ictal. After that there was a need to change Trg to have [1 0 0] for interictal, [0 1 0] for pre-ictal and [0 0 1] for ictal. Since there was a great difference between the number of values for interictal in relation to the rest of the classes there was a need to sub-sampling the values of interictal so that the amount of values would be equal to the sum of the values of ictal and preictal and after that it was a matter of making the matrices to have the corrected size by eliminating values from interictal.

4.2 Architecture - Multilayer

In this project, it was created two types of multilayer networks. One without delays and another with delays.

For the one without delays we created a file named FFNN.m where it is create a feedforward neural network. It offers the possibility to add from 1 hidden layer to 3 hidden layers and define the transfer function, which can be purelin, logsig or tansing and define the training function which can be trainlm, trainscg or traingd. The number of neurons for each layer can also be defined. The perform function used is msereg. For the one with delays it was created a file named RecurrentNN.m which creates a layer recurrent neural network which is better for dynamic systems with memory which is the case for this project. The other specifics are the same as the previous network.

Since there is a need to give more importance to situations with seizures and situations immediately before that, then the weights for each of these situations are different, giving a bigger weight for ictal and preictal.

It was used parallelism and GPU for both architectures, to improve their performance.

4.3 Pre-Processing - Deep NN

For both NN used, Long Short-Term Memory(LSTM) and Convolutional(CNN), different methods of pre-processing were used.

Regarding C NN, a script was made in order to split the data set into images of 29x29 and then turned into a 4D array with the format 29x29x1xNumberOfImages, since all images are gray-scale.

Ãs for LSTM NN, another script was made in order to divide the dataset into a cell array, each cell corresponding to 29 features in a second.

Both target vectors were turned into a categorical vector in order to represent the class on that given momement.

4.4 Architecture - Deep NN

In relation to deep neural networks, we created one file name convolution_max.m which creates a convolution neural network. This architecture used a variable number of layers, since the number of convolutional layers varied according to the GUI. For each convolutional layer, a batch normalization layer and a ReLu layer were also added. For the convolutional layer the number of filters and their size is configurable and also the stride of the layers. The pool size for the layers between the convolutional layers it was included a pooling layer in which the poolsize is configurable and its type.

4.5 Architecture - Autoencoder

It was created a script named encodeData.m which created an autoencoder with the number of desired layers for the encoded layer (layer with the minimum number of features). And after that it encodes the data and returns it.

5 PostProcessing

For postprocessing and to have a way to analyse each architecture, we develop two methods to calculate the sensivity and specificity of the results. In the first method we classify point by point for each of the three classes and obtain the number of TP, TN, FP and FN and after that we use the formulas to calculate sensivity and specificity. We use these values to determine the performance of each architecture for prediction and detection of seizures. In the second method we classify seizure by seizure so in this case we remove some points to obtain this.

6 Results

The results are present in the excel file with the potential remarks about those results.