Professor:	António Correia	**Course:**	Machine Learning
Theme:	Assignment 2	**Date:**	20th November 2023
Students:	Tomás Dias	**Student Number:**	2020215701
	Ricardo Guegan		2020211358
	Sonia Zavaglia		2023180614

Machine Learning Report

All the code used on this project will be available on the folder `ML_Project.zip` that is annexed to this document.

Introduction

Epilepsy is one of the most prevalent neurologic diseases, characterized by seizures: each one may last for some seconds or some minutes, and affects seriously the motor, perception, language, memory and consciousness, causing a decrease in the social and professional abilities of the patients and their families. The research field aims to predict or detect a seizure based on the information from electrical brain signal (EEG). In this work neural networks have been developed to detect or predict seizures: the networks used are both Shallow Networks (multi-layers without delay and other complex layers) and Deep Networks (CNN and LSTM) whose outputs are 3 classes: *Interictal*, *Preictal* and *Ictal*.

DataSet

The dataset contains data from 2 patients (54802 and 112502), one of them is a seventeen years old male that suffers from SP(25) and SG(6) and the other is a eleven years old female that suffers from CP(4), SP(4) and UC(6).

From this dataset, the P matrix has been built by transposing the FeatVectSel matrix (which contains the 29 features of EEG signal). The Trg_vector has been constructed by changing the values of the Trg column matrix (which identifies the seizures) to use 1 for interictal, 2 for preictal, and 3 for ictal. Finally, the T matrix (the target) has been realized by changing the point 1 with [1 0 0], the point 2 with [0 1 0], and the point 3 with [0 0 1].
In the figure below we can see an example of istants in which there is a preictal phase followed by an ictal phase.

0	0	0	0	0
1	1	0	0	0
0	0	1	1	1

Figure 1: T matrix

Architecture of the Neuralnets

Shallow Network

Our first approach to the problem was a simple *feedforwardnet*, a type of net that only uses fully connected layers. We used a trial and error approach to land on a final model that uses 2 hidden layers of fifty and ten neurons each. However such a model is limited and does not take into account the fact that the brain is a dynamic system. To overcome this limited model, we also built deep networks, which can learn patterns of more complex and dynamic systems such as the human brain.

Convolutional Network

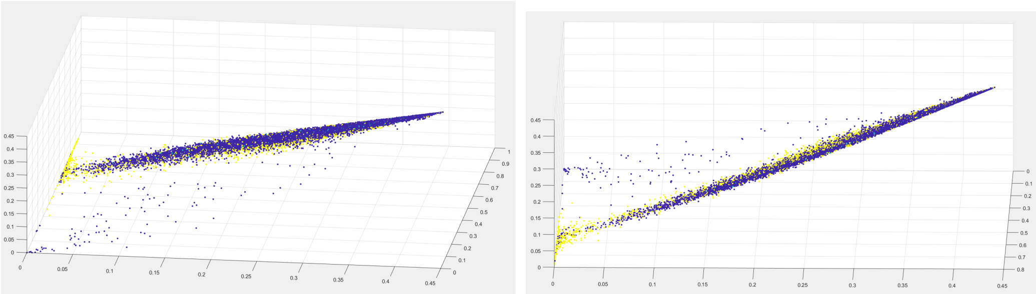
This network made use of a CNN layer, a layer that recieves a 2D image as input, so we firstly converted the features time series into 2D images. The neural network we built is composed by several layers: the input layer (specific for grayscale images), a 2D convolution layer with 9 filters of size 10x10, a custom activation layer using the swish activation function, a max-pooling layer with a pool size of 2x2 and a stride of 2, a fully connected layer with three neurons, a softmax layer (to apply the softmax function to the outputs, converting them into probabilities) and a classification output layer with specified class names and optional class weights. We also considered using a ReluLayer instead of the SwishLayer, but we ended up having better results with the Swish Layer, we believe that it is because we are not working with real images but instead with a time series that we converted into images.

Long Short Time Memory Network

LSTM performs sequence learning and the network we built is made up of an input layer, an LSTM layer (a fully connected layer with one hundred number of hidden units and three output neurons), a softmax layer and classification output layer for the same reassons as on the CNN neural network. We chose 100 hidden units because it had the best performance training time ratio, but we believe that one hundred is the optimal value since on average the ictal sequences have a size that is close to one hundred.

Autoencoders

To reduce the number of features we built the autoencoders. At the beginning we had 29 features and then, after applying the autoencoder, we reduced them to 3, 5, 10, 15 and 20, in order to evaluate the impact of the number of features on the neural networks. We tried both stacking autoencoders and doing the shift to the number of features directly, we obteined better results the second way. We also decided to not use Weigth Regualarization since it leaded to least scattered data and ended up showing worst results.



(a) No regularization **(b)** L2 Regularization

Figure 2: Autoencoders 3 features

Machine Learing Techiques

Class Balancing

Since the number of data points for the ictal class is much lower that for the other classes, we used the class balancing approach to prevent the possibility to detect and classify well just the interictal but not the ictal or preictal points. So we applied the class balancing to both CNN and LSTM in order to equilibrate the number of points of the several classes.
The function balance_CNN creates a training set, taking the first 80% of images from each class, and a test set, taking the remaining 20% of images from each class, ensuring that each class is always represented and that the dataset is balanced.
The function balance_LSTM returns two outputs: P (balanced feature vectors) and target (balanced target labels). The balanced dataset at the end contains an equal number of samples for each class, ensuring a more even distribution during training.

Error Weights

To give more importance to the detection of the Ictal and of the Preictal instants, we needed to use higher weights in the classification layers of the diferent networks, so that when the network is not sure about witch class the point belongs to she opts for a Ictal or Preictal avoiding false negatives but enhancing false positives. Since we consider more important sensetivity that specificity we implemented this technique but we understood that her use is dependent on the problem we are trying to solve.

Measuring the Results

Performance

To test the performance of the classifier we measured the sensitivity (how many true seizures did it preview or detect) and the specificity (how many false seizures did it predict or detect).

Post processing

To evaluate the performance of the classifier in a more elaborated way, we computed sensitivity and specificity in the post_processing function, considering as a hypothesis that a seizure could be detected only if the classifier found 5 consecutive ictal points among the last 10 (using 5 as a tentative threshold).

Results

Table 1: Shallow Neural Network

Patient	ErrorWeigths	Balance	SensitivityDetect	Specificity	SensitivityPredict
	Detect	Yes	73.3301	48.7567	0
112502	Preditct	Yes	72.7938	100	92.5890
	NO	Yes	5.38	98.63	3.72
	Detect	Yes	50.8642	97.2278	0
54802	Predict	Yes	0	98.167	99.8429
	NO	Yes	2.156	100	1.325

Shallow Neural Network

The results that we obtained through the Shallow Neural Network are quite satisfactory, we were able to observe the influence that the error weights can have, and how they dictate the output of the network. As we can observe in the **Table 1** depending on which error weights we chose to use, we can specialize the network on performing well on either prediction or detection. Also to obtain decent results we had to have class balancing always turned on or else the network would always predict *interictal*, the most predominant value in the data Set. The network performed the better in patient 112502 where sensitivity, specificity and *preictal* sensitivity are high, showing that this neural network can recognize very well true negatives and true positives. It is worth noticing that the results presented in the table were the best ones we could obtain and that we had to train and test a lot of times in order to get them. The performance of this network model has a very high variance.

Table 2: Convolution Neural Network

Patient	ErrorWeigths	Balance	SensitivityDetect	Specificity	SensitivityPredict
	Yes	Yes	100	69.56	94.31
112502	NO	Yes	95.69	63.92	97.78
	NO	NO	57.8767	99.9120	7.893
	YES	YES	100	63.217	97.6471
54802	NO	Yes	94.91	70.829	96.60
	NO	NO	57.8767	99.9120	5.6393

Convolution Neural Network

The Convolution Neural Networks gave us very good results, this one is able to perform prediction and detection tasks both at the same time, and being good at both. This one also showed to be more consistent in the sense that almost all the training's managed to achieve sensitivities of 90% and specificity of 65%, unless when we turned off class balancing which lead to an almost complete lost on the ability of detecting Preictals. One of the challenges we had to overcome in order to obtain good results using this architecture was the lack of data for the preictal and the ictal classes since each image used 29 inputs of each class. To get around that problem we used an overlap strategy that would reuse x colons of the previous picture, being x the overlap we chose.

Table 3: Long Short Term Memory Network

Hidden Units	ErrorWeigths	Balance	SEdetect	SP	SEpredict
100	Yes	Yes	100	99.6052	100
100	NO	Yes	99.081	97.8296	100
60	Yes	Yes	99.2331	98.4780	99.8333
40	Yes	Yes	93.0636	90.6933	96.5556
20	Yes	Yes	90.86	91.45	93.12
10	Yes	Yes	86.7052	88.5445	87.2778

Long Short Term Memory Network

The Long Short Term Memory Network as expected was the one that obtained the best results, this is due to her nature of taking into account time and being able to analyse time series allowing her to make correlations between the current input and the inputs previously seen. This Neural Network was also requiring a special type of class balancing: for this network what we did was isolating the seizure and then taking the 500 values before it and the 300 values after. This method allowed us to keep time coherence.

Learnings

1. **Working with Big Quantities of Data**: In this project we had to work with non proportional data, that means that the different classes were not equally represented, to deal with that we had to learn how concepts such as class balancing, this consists of finding an equilibrium between all the classes so that the network can learn to recognize them all. Another technique used was to deal whit the non proportional problem of the data was the implementation of error weights that helped the net learn the least represented classes.

2. **Creating Data**: Since we had few examples of the ictals and that in order to train some models of deep networks, in our case the CNN, we require a lot of data, we had to create new images of the ictals by overlaping images.

3. **Limitations of the Shallow Neural Networks**: We saw that they are not consistent and that they cant correctly identify the diferent all the classes in a complex and dynamic system.

4. **Autoencoders and information Lost**: The use of autoEncoders reduced greatly the training time, but on the other and they lead to some information lost, we found that the balance between information lost and traning time was at 20 features.

5. **The power of deep Networks**: We designed and observed how better and more consistent deep networks can be over shallow ones. We also learned when to use them and possible layers combinations that go well together, such as the CNN and the SwishLayer or the SoftMaxLayer and the ClassificationLayer.

Conclusion

In conclusion, critically analysing the results we obtained using different neural networks, we can state that the *Long Short Term Memory Network* seems to be the best one in preventing and detecting seizures since the sensitivity and the specificity for both patients (112502 and 54802) are quite high. This may be related to the fact that LSTM networks are good at capturing temporal information and dependencies over time. However the realization of a classifier with good performance is influenced by the subjectivity of the patient, as we can see in the previous tables where the results were different between the patient 112502 and the other patient 54802, so it would be interesting to develop a neural network able to adapt to any patient.