Data Science Project

EEG epileptic spike detection

BENCHIKH Mouhamed Amine
BOUCHAMA Salah
JAU Lamine

1 Introduction

An electroencephalogram (EEG) is a recording of brain activity, this brain activity can help your healthcare provider diagnose and monitor brain-related conditions like epilepsy. For epilepsy diagnosis or treatment, the neurologist needs to observe epileptic spikes from eeg data. But the reading of these data by the specialist requires a considerable effort and time because of the duration of the recording of the EEG signals which can take from 20-30 min up to 48-72h in certain cases and more than that, the lack of medical staff in some cases leads to difficulties in guaranteeing diagnoses to patients, that's why signal analysis and automatic anomaly detection present a very good alternative to help in the medical field.

2 Problem formulation

EEG epileptic spike detection is a crucial and challenging problem in both data science and neural science. The purpose is to develop methods that allow us to detect automatically whether an instance (such as an EEG data series) captures an epileptic spike or not. It can be simply transferred to the detection of the presence of inter-ictal spikes in the multichannel EEG recording, That means that high proportion of true events must be detected with a minimum number of false detections. In this context, we are confronted with two main challenges which are : the fact that epileptic spike behaviors are minor compared to normal behaviors and the difference between epileptic spike morphologies in different individuals.

Automated analysis of EEG recordings for assisting in the diagnosis of epilepsy started in the early 1970s, (Gotman, 1999; Tzallas, et al., 2007a, 2007b, 2009; Wilson Emerson, 2002), The spike detection problem is divided into two main steps: feature extraction and classification. and there have been several approaches to this problem in bedding and among them we have : methods based on traditional recognition techniques, known as mimetic techniques which are based on the hypothesis that the process of identifying a transient waveform in EEG recordings as spike could be divided into well-defined steps representing the reasons and expertise of a neurophysiologist (8).

Methods based on morphological analysis : This approach allows decomposition of the raw EEG signal into several physical parts, the background activity and the spike component are separated and the main morphological feature of the spikes is preserved. Morphological analysis is an efficient tool in EEG signal processing.

Methods employed classifier or a clustering technique: supervised and unsupervised are also use to build automatic spike detection models, like K-means algorithm (Exarchos, et al., 2006; Tzallas, Karvelis, et al., 2006), KNN classifier (1) or Support Vector Machine (SVM) (2), and several neural networks and deep learning models are also used in the field of automated spike detection.

3 Proposed solutions

An EEG classifier has two main steps: feature extraction and classification. In our work, we tried two solutions, the first one consists of testing 3 different algorithms using some features and the second one is to try a 1D convolution neural network (CNN) architecture.

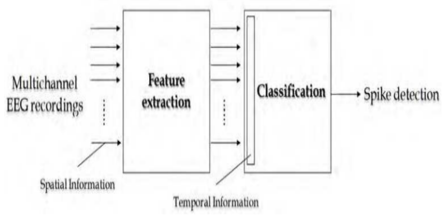

Figure 1: main steps

Our first approach was to extract some features from EEG data and to compare the result between 3 different classifiers (MLP, AdaBoost, XGBoost).

Multilayer perceptron (MLP) : A multilayer perceptron (MLP) is a feedforward artificial neural network that generates a set of outputs from a set of inputs. An MLP is characterized by several layers of input nodes connected as a directed graph between the input and output layers. MLP uses backpropogation for training the network.

AdaBoost : AdaBoost algorithm, short for Adaptive Boosting, is a Boosting technique used as an Ensemble Method in Machine Learning

XGBoost : XGBoost, which stands for Extreme Gradient Boosting, is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library. It provides parallel tree boosting and is the leading machine learning library for regression, classification, and ranking problems.

The features of the EEG signal are statistical features, like mean, median, variance, ...ect and there are other which is not a statistical feature like Zero-crossing rate (ZCR). For our solution we have selected 17 features, These features are provided using the wavelet transform on the EEG signals, The wavelet transform translates the time-amplitude representation of a signal to a time-frequency representation that is encapsulated as a set of wavelet coefficients. These wavelet coefficients can be manipulated in a frequency-dependent manner to achieve various digital signal processing effects :
- Amplitude RMS: the square root of the average of the squares of a series of measurements
- Variance : a measure of variability. It is calculated by taking the average of squared deviations from the mean.
- Kurtosis: is a measure of the combined weight of a distribution's tails relative to the center of the distribution.
- Skewness: is a measure of the asymmetry of the probability distribution of a real-valued random variable about its mean.
- Max amplitude.
- Min amplitude.
- Number of zero crossings: A zero-crossing is a point where the sign of a mathematical function changes.
- Fractal dimensionis an approximate value for the box-counting dimension of the graph of a real-valued function or time series.
- Hurst exponent: used as a measure of long-term memory of time series.
- spectral entropy (1.21s): is a measure of its spectral power distribution.
- Total power: the sum of the absolute squares of its time-domain samples divided by the signal length.
- Median frequency : represents the midpoint of the power distribution in the CSA and is the frequency below and above which lies 50 pourcent of the total power in the EEG.
- Peak frequency : is the frequency of maximum power.
- Hjorth mobility or complexity : the ratio of mobility of the first derivative of vibration signal to the mobility of the vibration signal.

The second aproach is based on Deep Convolutional Neural Network (CNN). A Convolutional neural network (or CNN) is a special type of multilayer neural network or deep learning architecture inspired by the visual system of living beings. Each specific neuron receives numerous inputs and then takes a weighted sum over them, where it passes it through an activation function and responds back with an output. CNNs are primarily used to classify images, cluster them by similarities, and then perform object recognition. The CNN is very much suitable for different fields of computer vision and natural language processing, also it provide a very good results in the field of eeg's.

For this solution we decide to use a 1D convolutional networks. The 1D convolutional networks are similar to well known and more established 2D Convolutional Neural Networks, it mainly used on text and 1D signals. In summary, In 1D CNN, kernel moves in 1 direction. Input and output data of 1D CNN is 2 dimensional. The proposed architecture is inspired by a work done for the Automated diagnosis of schizophrenia (5).

The structure in Figure 2 is composed of 4 blocks in which two of them are composed of one Conv-layers and one max-pooling, the two others are composed of One Conv-layers and average-pooling. In the figure 2, (5, 10) represents the kernel size is in shape 10×1 and the number of filters used is 5.

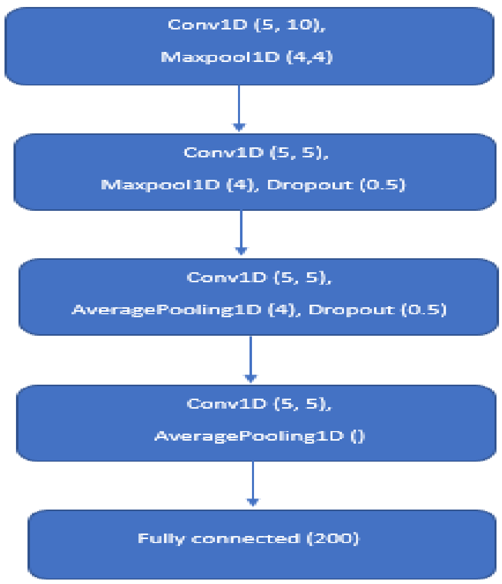

Figure 2: Structure of the CNN network

The LReLU (3) is applied across all layers after Conv-layers, The dimension reduction in the pooling layer is done by a pooling size of 4 with a stride of 4 in max-pooling and a stride of 2 in average-pooling. A number of fully connected layers are used after the convolutional layers and for the last fully connected layer where sigmoid activation is used to output a probability distribution among the two possible classes.

In order to prevent the network from over-fitting the training data and achieve better generalization capability, we use the batch-normalization (6) which is layer that allows every layer of the network to do learning more independently and and drop-out (7) which randomly sets input units to 0 with a frequency of rate at each step during training time. so We perform batch normalization after the first two Conv-layer before pooling layer and we put a dropout of 0.5 probability.

```
clear_session()
model = Sequential()
model.add(Conv1D(filters=5,kernel_size=10,strides=1,input_shape=(768,5)))
model.add(BatchNormalization())
model.add(LeakyReLU())
model.add(MaxPool1D(pool_size=4,strides=4))


model.add(Conv1D(filters=5,kernel_size=5,strides=1,))
model.add(LeakyReLU())
model.add(MaxPool1D(pool_size=4,strides=4))#4
model.add(Dropout(0.5))


model.add(Conv1D(filters=5,kernel_size=5,strides=1))#5
model.add(LeakyReLU())
model.add(AveragePooling1D(pool_size=4,strides=2))#6
model.add(Dropout(0.5))

model.add(Conv1D(filters=5,kernel_size=5,strides=1))#7
model.add(LeakyReLU())
model.add(AveragePooling1D(pool_size=4,strides=2))#8

model.add(Conv1D(filters=5,kernel_size=5,strides=1))#9
model.add(LeakyReLU())
model.add(GlobalAveragePooling1D())#10
model.add(Dense(200,activation="relu"))

model.add(Dense(1,activation='sigmoid'))
```

Figure 3: pseudo code of the CNN network

For the learning rate, we adopt a learning rate decay strategy starting with a learning rate of 0.01 decreasing towards 0.0001 by decay steps = 1000.

```
lr_shdler = PolynomialDecay(initial_learning_rate= 0.01,end_learning_rate= 0.0001 ,decay_steps = 10000)
```

We also used the Use K-Fold Cross Optimization, this technique involves randomly dividing the set of observations into k groups, or folds The first fold is treated as a validation set and the others as training set. It used to get better accuracy.

4 experimental results

The solutions above are implemented on python and exactly in google colab which is a cloud service, offered by Google (free), based on Jupyter Notebook and intended for training and research in machine learning. This platform allows to train Machine Learning models directly in the cloud.

The dataset used on this works for this work we have data from two different experiences, 400 epileptic spike data series (positive instances) and 2000 normal data series (negative instances) from experiment 1 and 2. each of this data series also contains 5x768 sample points.

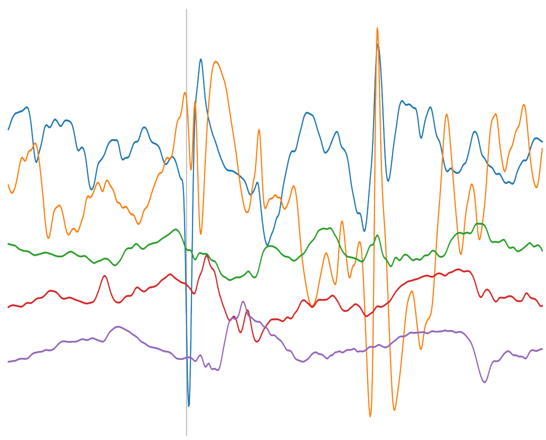

Figure 4: representation of Instance of the data set

Step 1 : in this we have trained our models on only date collected from experiment 1. and the results obtained are the following :

Methods	Kaggle Score
MLP	0.92416
AdaBoost	0.91666
XGBoost	0.88916
CNN-1D	0.93833

as we can see the CNN provides the best result however its cost in time is quite high, we also notice that the accuracy obtained from the training set is quite far from the score obtained on Kaggle and this could be due to an over-filtering.

```
Score per fold
------------------------------------------------------------------------
> Fold 1 - Loss: 0.10211028903722763 - Accuracy: 97.70833253860474%
------------------------------------------------------------------------
> Fold 2 - Loss: 0.046576619148254395 - Accuracy: 98.7500011920929%
------------------------------------------------------------------------
> Fold 3 - Loss: 0.055420972406864166 - Accuracy: 97.50000238418579%
------------------------------------------------------------------------
> Fold 4 - Loss: 0.05503161624073982 - Accuracy: 98.54166507720947%
------------------------------------------------------------------------
> Fold 5 - Loss: 0.04245623201131821 - Accuracy: 98.33333492279053%
------------------------------------------------------------------------
Average scores for all folds:
> Accuracy: 98.16666722297668 (+- 0.48232618061716837)
> Loss: 0.060319145768880846
------------------------------------------------------------------------
```

Figure 5: Classification result per fold of CNN step 1

Step 2 : In addition of the data of the experiment 1, we added the data from experiment 2 and the results obtained are the following :

Methods	Kaggle Score
MLP	0.93916
AdaBoost	0.92833
XGBoost	0.93583
CNN-1D	0.97416

After the addition of the data of the experiment 2 we can observe a progression in the scores of the models notably for the XGBoost algorithm and the CNN model, we even notice the approaching of the scores obtained at the time of the training and that of kaggle, however with the addition of the data the cost of training on all that of the CNN increased considerably.

```
Score per fold
------------------------------------------------------------------------
> Fold 1 - Loss: 0.05762176215648651 - Accuracy: 97.81249761581421%
------------------------------------------------------------------------
> Fold 2 - Loss: 0.0681648701429367 - Accuracy: 97.60416746139526%
------------------------------------------------------------------------
> Fold 3 - Loss: 0.10345780849456787 - Accuracy: 97.50000238418579%
------------------------------------------------------------------------
> Fold 4 - Loss: 0.07781589776277542 - Accuracy: 97.91666865348816%
------------------------------------------------------------------------
> Fold 5 - Loss: 0.0442764051258564 - Accuracy: 98.33333492279053%
------------------------------------------------------------------------
Average scores for all folds:
> Accuracy: 97.83333420753479 (+- 0.2901747786029932)
> Loss: 0.07026734873652458
------------------------------------------------------------------------
```

Figure 6: Classification result per fold of CNN step 2

5 Conclusion

In our work we were able to test several machine learning methods and we had very good results especially with the CNN model which provides a better result than the other one we tested. In addition to the good results it provides in imaging, the CNN seems to be a good tool for the detection of EEG epileptic spikes, even if in terms of time cost it seems to be the most expensive.

One of the particularities of the CNN is that two trainings can provide 2 different results even if they are close, so it is preferable to use a function that allows to save the best evolution of the model and for that we have the tensorflow callbacks.

finally the automation of EEG epileptic spike detection will be of great help knowing the lack of effective in certain countries which makes the diagnosis of the patient extraimement long or impossible in certain case.

References

[1] Jayanthi et al. J Big Data (2017). Survey on clinical prediction models for diabetes prediction Big Data.

[2] Zhang Z, Parhi KK. Low-complexity seizure prediction from iEEG/sEEG using spectral power and ratios of spectral power. IEEE Trans Biomed Circuits Syst. 2016;10:693.

[3] Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: Proc. icml. vol. 30, p. 3 (2013)

[4] Gotman, 1982; Gotman Gloor, 1976; Guedes de Oliveira, 1983; Kansas, 1983; Ktonas, Luoh, Kejariwal, Reilly, Seward, 1981.

[5] OH, Shu Lih, VICNESH, Jahmunah, CIACCIO, Edward J., et al. Deep convolutional neural network model for automated diagnosis of schizophrenia using EEG signals. Applied Sciences, 2019, vol. 9, no 14, p. 2870.

[6] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15, 1929–1958 (2014), http://jmlr.org/papers/v15/srivastava14a.html

[7] OH, Shu Lih, VICNESH, Jahmunah, CIACCIO, Edward J., et al. Deep convolutional neural network model for automated diagnosis of schizophrenia using EEG signals. Applied Sciences, 2019, vol. 9, no 14, p. 2870.

[8] (Gotman, 1982; Gotman Gloor, 1976; Guedes de Oliveira, 1983; Ktonas, 1983; Ktonas, Luoh, Kejariwal, Reilly, Seward, 1981)