Recognition of epileptic seizures from EEG data

Christian Espinosa Reboredo

07/02/2022

Abstract–An electroencephalogram (EEG) is a test that detects electrical activity of the brain. This paper tries to go a step further to interpret seizures from electroencephalograms using deep learning algorithms. The data used in this paper is a public dataset CHB-MIT[1] of recordings of paediatric subjects with intractable seizures. Different methods of data processing are done and documented to make the most of the algorithms used as well as the strategy. The objective is to train an algorithm to classify when the subject is having a seizure and when it is not.

Keywords– electroencephalogram, deep learning, brain activity, classification, EEG analysis

Resumen–Un electroencefalograma (EEG) es una prueba que detecta la actividad eléctrica del cerebro. Este artículo intenta dar un paso más para interpretar los ataques epilépticos a partir de electroencefalogramas utilizando algoritmos de aprendizaje computacional. Los datos utilizados en este documento son de una base de datos pública CHB-MIT[1] de EGG de sujetos pediátricos con convulsiones intratables. Se realizan y documentan diferentes métodos de procesamiento de datos para aprovechar al máximo los algoritmos utilizados, así como la estrategia. El objetivo es entrenar un algoritmo para clasificar cuándo el sujeto está teniendo un ataque epiléptico y cuándo no.

Palabras clave– electroencefalograma, aprendizaje computacional, actividad cerebral, clasificación, análisis EEG

1 INTRODUCTION

AN epileptic seizure is a period of symptoms due to abnormally excessive or synchronous neuronal activity in the brain. This can cause different effects like uncontrolled shaking movements involving much of the body, parts of the body or subtle momentary loss of awareness. In order to understand this issue, it is important to understand how neurons work and interact with each other to conserve what we call consciousness, represented as brain activity and brainwaves.

Neural oscillations are rhythmic or repetitive patterns of neural activity in the central nervous system which can be driven by mechanisms within individual neurons or by interactions. Since 1824 neural oscillations have been observed, fifty years later intrinsic oscillatory behaviour was encountered in vertebrate neurons, but the purpose of these is yet to be fully understood.

The main objective of this paper is to classify seizures from brain activity by building a deep learning architecture. First of all, it will be needed an inside view on how the brain works to have a hint on how and what is done to extract or intercept information from the neurons to process externally in a computer. This information is available annexed in this paper to have an overview to further understand the subject. This matter is not the main purpose of this paper as the work involves data processing, architecture, model strategies and classification results.

In this paper it is detailed the complete process of epileptic seizure detection, from data processing to seizure recognition. This study works as a pipeline of different stages. Therefore, an insight view of each one is done, starting from data processing from a well-known database CHB-MIT of encephalograms collected from 23 subjects with interactable seizures that has been used in previous research. Then followed by the strategy of the architecture used to classify the signals to finally understand the

• E-mail of contact: 1459024@uab.cat
• Degree Specialisation taken: Computació
• Work supervised by: Aura Hernández Sabaté (Ciencies de la Computació)
• Year 2021/22

classified results into seizure and not seizure.

There have been many other studies about seizure recognition, so in this project another approach is conducted to further study this subject. Before starting the study, an overview of different similar projects has been done. As acknowledgement, the scripts written in this paper have been supervised by the team at Computer Vision Centre (CVC) which are working on a project related to this paper.

2 Related Work

A lot of research has been made of the brain, to further understand its capabilities using deep learning algorithms. EEG signals can be used in many ways, such as the ones below. These studies, work on similar objectives of mental workload, seizure classification, attempting to understand and process EEG data, using deep learning techniques to develop frameworks to combine and process these signals.

2.1 Mental Workload Detection based on EEG Analysis

A study of mental workload[2] is done in order to work more efficiently, healthier and to avoid accidents since workload compromises both performance and awareness on the human. The use of EEG signals has a high correlation with specific cognitive and mental states such as workload, proposing a binary neural network to classify EEG features across different mental workloads.

Mental workload is defined as “the cognitive and psychological effort to conclude a task”, observing that depending if workload is too heavy or too light it can affect human performance.

There are two main categories to measure workload:

- **Subjective measures:** Being the most used to asses mental workload, the NASA Task Load Index (TLX)[3] a prominent way to gain insight on perceived workload from the subject based on a weighted average of six sub-variables: mental demand, physical demand, temporal demand, performance, effort and frustration.
- **Physiological measures:** They provide a more reliable data by measuring physiological dynamic changes which cannot be controlled consciously, so that is why it is more reliable. Some examples of these readings are electrocardiogram (ECG), electromyograph, electroencephalogram (EEG)... The combination of inputs reports better accuracy than the analysis if each one independently.

The approach of the mental workload detection based on EEG analysis study is to investigate the ability of one dimensional convolutional neural network (1D-CNN)[4] models to recognise two types of mental load from EEG signals and to generalise the model to a population that has not been take into account in the training set. N-back test (memory demanding games requiring the resolution of simple arithmetic operations adjusting workload) is used to induce low and medium workload and to classify a simple neural network (NN) trained using only the power spectrum of theta waves.

The method to obtain data started by having the sixteen subjects to watch a 10-minute relaxing video. Then, perform the N-back-test low, medium and high difficulty tests. Finally, subjects fill in a TLX questionnaire for subjective perception of the test difficulty and workload.

To obtain the data, EEG recordings were done using an EMOTIV EPOC+ headset which has 14 electrodes placed according to the 10/20 system, which provides raw data and power spectrum for the main brain rhythms (theta, alpha, beta low, beta high), at 128 Hz and 8 Hz, respectively.

The study concluded with the following results, reported 95% confidence interval for each class computed for all subjects and recall above 90%. It is suggested to use longer windows to capture EEG non stationary nature.

2.2 EEG Signal Dimensionality Reduction And Classification Using Tensor Decomposition And Deep Convolutional Neural Networks

In this study, EEG Signal Dimensionality Reduction and Classification Using Tensor Decomposition and Deep Convolutional Neural Networks[5] has the same objective of seizure classification. It is the only project I have been able to find using the same dataset CHB-MIT. The study uses convolutional networks (CNN) but with different architectures.

Using convolutional neural networks (CNNs), still suffer from high dimensionality of the training data. The proposed tensor decomposition-based dimensionality reduction algorithm transforms a large set of slices of the input tensor to a concise set of slices which are called super-slices which handles the artifacts and redundancies of the EEG data and also reduces the dimension of the CNNs training inputs. This proposed framework is tested on HCB-MIT data and as results show, the approach outperforms other previous studies.

2.3 Epileptic Seizures Detection Using Deep Learning Techniques: A Review

The development of deep learning algorithms in many areas of medicine, such as in the diagnosis of epileptic seizures, has made significant advances.

In this study Epileptic Seizures Detection Using Deep Learning Techniques[6], a comprehensive overview of works focused on automated epileptic seizure detection using deep learning techniques and neuroimaging modalities is presented. Various methods proposed to diagnose epileptic seizures automatically using EEG and magnetic resonance imaging (MRI) modalities are described. In addition, rehabilitation systems developed for epileptic seizures using deep learning have been analysed, and a summary is provided. The rehabilitation tools include cloud computing techniques and hardware required for implementation of deep learning algorithms. Challenges, advantages and limitations in employing deep learning-based techniques for epileptic seizures diagnosis are presented as well as the most promising models and future works on automated epileptic seizure detection.

2.4 Recognition of mental workload of pilots in the cockpit using EEG signals

The commercial flightdeck is a naturally multi-tasking work environment and so automatic characterization of pilot's workload becomes essential. Electroencephalogram (EEG) signals have shown a high correlation to specific cognitive and mental states like workload. However, there is not enough evidence in the literature to validate how well models generalize in case of new subjects performing tasks of a workload similar to the ones included during model's training.

In this paper Recognition of mental workload of pilots in the cockpit using EEG signals[7] it is proposed a convolutional neural network to classify EEG features across different mental workloads in a continuous performance task test that measures a portion of working memory and working memory capacity. The goal of this paper is to characterize the mental workload of flying pilots in the cockpit from the analysis of EEG signals. The model proposed is valid at a general population level and it is able to transfer task learning to a pilot mental workload recognition in a simulated operational environment.

3 Objectives

The main goal in this study is to detect seizures from the dataset CHB-MIT. To fulfil the objective an architecture has been created working within a pipeline of events. Starting by finding the best data analysis and processing method before feeding it to the deep learning algorithms. There are many sub-objectives to be completed to obtain good results from this architecture and also, it's modular for further expansion and studies. The main strategy of the scripts execute in sequence (Fig. 1)

Within the objective of data classification, another objective to find the most precise architecture is defined to obtain results with the most accuracy possible. In order to fulfil these objectives a set of sub objectives have been defined.

The following points summarize the set of subobjectives:

1. Raw data must be readable, as the data base CHB-MIT is in European Data Format (EDF), a standard file format designed for exchange and storage of medical time series, all files in the dataset are in edf format. A script has been programmed to save edf files into parquet format. Parquet is needed to be able to handle data more easily, as well as improving compatibility with other related work scripts from the CVC.

2. Setting different functions to filter data, making sure data fits certain constraints to obtain better results when training the model. This makes the script modular to obtain data from the dataset CHB-MIT.

3. Label raw data, to have a ground truth from the provided summary files in the dataset. This part is essential to understand if the model works as expected.

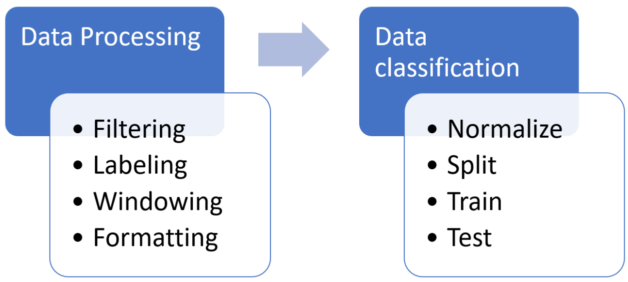
4. Functions create constraints on the dimensionality of the data to fit the input of the model as well as filtering data from the dataset. This functions filter in bandwidth of the data, and excluding files from the dataset where data is not well recorded.

5. Each model needs to be configured to accept the dimensionality of the data fed to it. All data must be in two files of numpy arrays and all input must have the same dimensions. This will make sure all files are read the same way, and the model will train with uniform data.

6. Work different models to choose what models give better answers form input data. The CVC provides different models adapted to related work. These models have different inputs and are initialized in different ways. All models must be initialized and executed the same way, to make it easier to train and test more than one model, to obtain different answers.

7. After all the models results, an overview is done to understand the results and conclude the best way to treat this database, for further investigation.

Fig. 1: Scheme of data processing



4 Methodology

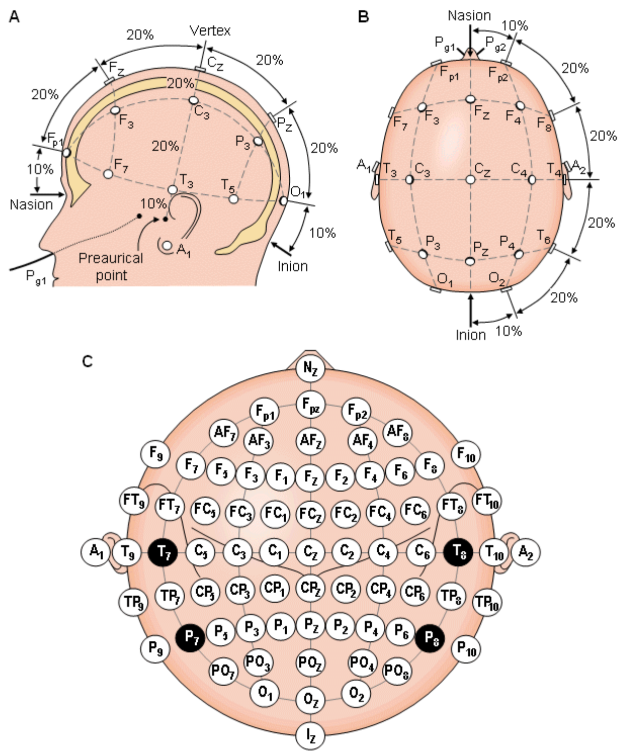
4.1 Dataset

The dataset is data collected from the Children's Hospital of Boston, consisting in EEG recordings of subjects with intractable seizures. The folders classify in 23 cases from 22 subjects (case chb21 and chb1 are the same but 1.5 years apart). The subject's personal information gender and age is in a separate file called SUBJECT-INFO added in this paper as subject_info.csv.

Each case contains between 9 and 42 edf files. There are edf files of EEG signals without seizures and others with recordings of seizures, these defined in RECORDS-WITH-SEIZURES. The files with seizures have the extension edf.seizures which disables the possibility of accessing the file with a normal edf reader library. The seizure data can be read from the summary file.

Most cases have 1 hour of EEG recordings, but some have 1 to 4 hours depending on the case, split between 9 to 42 edf recordings, recorded at 256Hz in 16 bit resolution. The position of the electrodes as well as the nomenclature are in the International 10-20 system.

Fig. 2: The international 10-20 system defining electrodes position seen from different views



It is important to note some subjects had hardware interruptions while the recording of the EEG, and so when there is an interruption, it is noted in the summary.txt file. This kind of interruptions are a problem to get information normally, because the disposition of the electrodes change making it harder to control the position of the different channels. Therefore, the EEG might not work for a sequential approach, for example, if hypothetically an order on how a seizure comes to be, this file would certainly be discarded. To take into account this file the script to process the data in this study should be programmed to do so. For now, the objective is different, this project just classifies if there is or not a seizure, but for further development it should be considered. The data base is very large containing enough uninterrupted data to work with.

The data used to train the model in this project is data from subject 1 to subject 10. Also, only few edf files in each subject's folders have any seizure. If every edf was used, there would be a lot more data labelled as not seizure than seizure data, so from each subject only the files with seizures are used. Even so, handling only files with seizures the dataset is still unbalanced, so for future development two strategies should be considered:

- Files should be cut to have a balance of labelled data of 50% data with seizure and 50% data without. This strategy could end up in subsampling, considering there are few seizures in the hole dataset and seizures only happen for around 90 seconds.
- When training the model, the criteria of cross Entropy should be weighted. To consider no seizure less important than seizure data. This strategy is much more viable to avoid subsampling and consider all data in the database.

4.2 Data processing

Because in this paper the CHB-MIT Scalp EEG Database is being used and all files are in format edf, a first script has been needed to process data, called "03_ReadEDF.py". In the script, there are different options on how raw data is imported. During the development of this project many tests have been done, therefore there are two different ways to execute the script:

- **Single execution:** Where the subject number and the edf file of the subject needs to be provided to execute the script for this single file.
- **Multiple executions:** Where the number of subjects is provided. The script will go through all the first n subjects defined.

It is designed to extract the files from a specific folder hierarchy, where all the encephalograms are classified by subjects. For simplicity this script obtains, filters, plots, and saves in parquets all input data. Afterwards it also labels the data, splits it into windows and saves it in two different files of numpy array, for the model to accept the data.

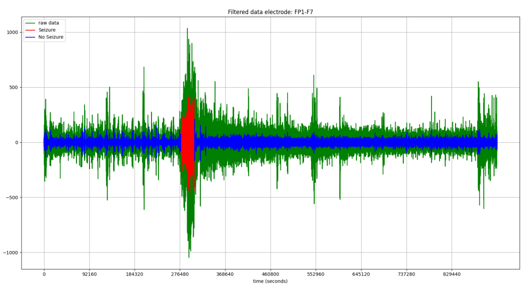
The script will automatically label all raw data using the summary file in each subject's folder, so it is important for it to be present or a label execution error will pop up. The files edf.seizures in every subject's folder were unreadable, even reading the binary was a failure. The script will make sure the file has all the data from the desired electrodes, this is important because there were hardware problems while recording the edfs. Some files have gaps or lack some data, if any edf file has this problem it will automatically be excluded and the user will be notified. Each one has 22 different channels, which are the electrodes of the subject. In order to label the data, a new column is created (the 23rd) as seizure with the information of every row being a seizure or not and also a 24th column to set the observation windows for the model.

Filtering data is done by first setting a maximum range from 0.5hz to 50hz. Afterwards a bandwith can be set only by changing the name of a parameter in a dictionary, to delta, theta, alpha, beta or gamma's range frequencies added by default. All data is saved in parquet format in a different folder named parquets in every subject's folder. All variables are available on top of the script for easy access. If plotting is enabled it will plot each subject's data.

Once the data is filtered and labelled it is saved in two numpy files, one "file_data_x.npy" and "file_data_y.npy". In data_x the file has a numpy array of three dimensions containing number of windows, electrodes and values. In data_y there are two dimensions, number of windows and window_seizure which defines if the window has a seizure in it with a 1 or not with a 0.

With this, all the parquets and arrays are specifically saved and ordered to ensure easy access and fast comprehension of the hierarchy of folders. The database folder has every subject in a separate folder and inside individual folders for edf, parquets, numpys and results.

Fig. 3: Filtered data in Theta range from electrode FP1-F7, subject 1 file 3



4.3 Network

An already done deep learning algorithm from the research group IAM from the CVC is utilised, which is working on a framework to determine the optimal architecture for cognitive state recognition from EEG signals, with the objective to answer different questions:

- How to combine the signals to create the input features for feature extraction? In the case of the IAM group, having 14 sensors x 5 wavelengths, so 70 raw signals. These signals can be concatenated, or projected depending on the strategy desired to execute.
- Which neural network is the best performer?
- Is it better to ensemble the different classifiers before combining the signals?

This model was originally intended to study brain workload, so, with the help of this model it is changed to fulfil the objective of clinic seizure detection. In this study, different strategies are applied on the input data of the algorithm to further study its capabilities as well as using different models to compare results between them.

Once all desired raw data is filtered, labelled and saved, the second script to execute is "04_MExecution.py". The script is in charge of the execution of the model, training and testing to obtain the results classifying the data. All hyperparameters are defined at the beginning of the script as well as the declaration and initialization of the model. There are three different ways to execute the script by enabling or disabling boolean variables in the script.

- **CheckModel** is the first way to execute the model, this one is needed to make sure the model works as intended before using real data to train it. It uses the declared model and feeds to it random data using torch defined by parameters in the function.
- **Phase 1:** If the model works with no issues with CheckModel, then the training execution can begin. This first phase selects every numpy from n subjects defined at the beginning of the script, and trains the model with it. Every time the model finishes the epochs of a file, the model can be saved in the database if the option is enabled.
- **Phase 2:** In the second phase the previous trained model is loaded to test it. A desired subject is defined to obtain the normalization scalers for the testing data. For further development this should be changed to make an average of scalers by all trained data for example, or normalize data before the script's execution. Once the model is tested, a classification report is done by using the metrics library from sklearn and saved to the results folder in the subject's folder.

The script uses numpy arrays as input data stored as previously mentioned in data_x and data_y. These files depending on the strategy of the script can be processed one last time to ensure good results from the classification. In this study data when loaded uses the strategy stated previously as phase 1 and 2, for training and testing. But the script has the option to split the files in train and test by a percentage variable if needed. For the executions done in this study, this variable is set to 1 (100%) on training, and 0 on the testing phase to consider all the data from the file for the purpose of training or testing.

During the implementation of this script, a lot of problems of dependency on critical libraries have happened. For faster results, Cuda is used in all models, but it might not work if the architecture of the graphics card is too old. It is also not compatible with python 3.10 which is the version being worked on at the moment. It has been switched to an environment with python 3.8 to avoid further issues. The script will be executed with Cuda if the libraries are available as well as the drivers, if not it will automatically work with the CPU.

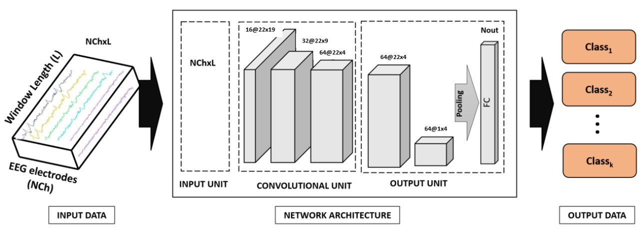
When it is time to save the model, it is saved in a folder called “trys” in the root directory, like if it was another subject but it only contains pt files. The name of the file is automatically created defined by the last subject and file executed, as well as the date considering day, month, year, hour, minute and second when the file got saved. Seconds where necessary to save in the name, to prevent models from overwriting if it had to be saved in the same minute because of low epoch training.

The models used in this study have similar structures. The first one is CNN_ConcatInput and the second one CNN_ProjOut_Conv. As the names suggest these models are Convolutional Neural Networks. Both models have 3 base layers of convolution defined using torch, by a function ConVNet. The first layer is 16@22x19, the first parameter is the number of input features, defined in the constructor of the model, the second parameter defines the size of the data. The channels are 22, in this case the electrodes used to gather information from the subjects. The second layer is 32@22x9 and the last layer 64@22x4. The dimension is being reduced before the pooling in the convolutional unit, to have less neurons to train.

The model CNN_ProjOut_Conv still has one last convolutional layer for data reduction. This layer is 64@1x4 to further reduce the dimension of the data before the two dimensional average pooling. Instead of this last layer, CNN_ConcatInput just changes the sizing of the data to avoid temporal variation.

All hyperparameters of the models are defined at the top of the “04_MExecution.py” script. Such as the size of the kernel (3,1), the number of neurons and the number of output classes (2).

Fig. 4: Scheme of data dimension transformation through model CNN_ProjOut_Conv



5 Execution

The execution of the models has been made with all the previous procedures regarding data processing in this paper. Data from subject 1 to 10 has been used to train the model, and data from 11 to 16 has been used to test the accuracy of the predictions of the model. For every file 50 epochs have been done to ensure a good understanding from the model of the data.

At first the strategy of training and testing the model was done within the files of the subjects. For each file it was split in two by a percentage variable and afterword’s it was trained and tested. This strategy was abandoned because it wasn’t as efficient as if many files where trained, and afterwards tested for results. Because there are few seizures in the hole database it makes it difficult to train a model to understand the existence of a seizure. It was dependent on the position of data in the file, considering there a small amount, in most cases only one seizure per file, if the file was split one half would have a seizure and the other would not have one. This is a big problem if training has no seizure and the testing data has it. It would never be able to learn what a seizure is. Opposite to the previous statement, it might learn what a seizure is, but it would never be possible to test it, if it learned correctly to classify.

Fig. 5: Data from subject 12 file 8 seen scattered through the hole recording

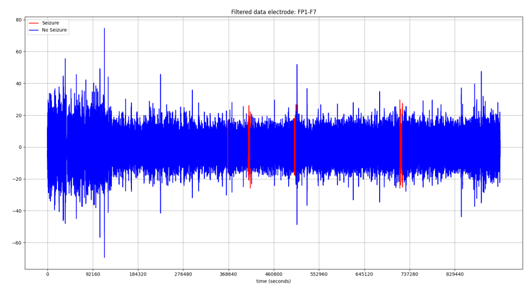
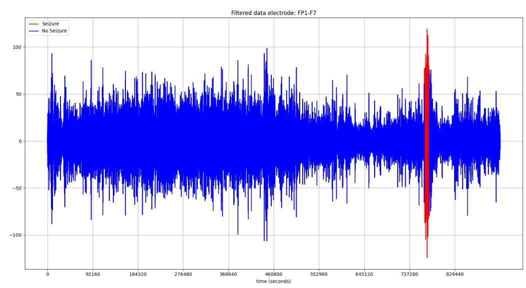


Fig. 6: Data from subject 1 file 3 with only one seizure. This file would not be eligible to split in train and test



In the end the strategy of first training the model with some files and then with others do the testing, was the best strategy, considering training each file took around 10 minutes depending on the computer it was executed with. There was a big difference between executing the model with and without Cuda. The first model CNN_ConcatInput, was executed without Cuda because of the hardware architecture. It spent around 30 hours training the model. The second model CNN_ProjOut_Conv, was executed with Cuda and spent training around 15 hours. It is important to consider using a desktop computer to train the model, considering it will get hot finishing the last epochs of every file.

Executing this type of models is also very resource consuming, considering it loads all the file to RAM to read and work with it from there. It could be a problem to many computers, because the ones with 8Gb will not work. Only with 16Gb of memory worked for me. In most cases, the edf files contains exactly one hour of digitized EEG signals, although those belonging to case chb10 are two hours long, and those belonging to cases chb04, chb06, chb07, chb09, and chb23 are four hours long; occasionally, files in which seizures are recorded are shorter, so the idea of concatenating files of a subject to have one file per subject it is also excluded.

6 Conclusion

6.1 Results

After training the two models with files from subjects 1 to 10, testing only took 15 to 30 minutes to execute with subjects from 11 to 16. The results given by the function classification_report of the library metrics from sklearn were saved in parquets and then loaded with the "05_ReadParquet.py" script. This script reads every parquet from the folder results of every tested subject and displays the results on the terminal.

At first results looked promising viewing a staggering 95% to 100% accuracy in classification from the model. But upon further inspection the results are bad, because I realized the accuracy given by the report was because it predicted all data to be "no seizure". The other 5% is the Seizure in the data that the model is classifying it was no-seizure. Because data is mainly one-sided to no-seizure, the result 95% is because it was only classifying this class. In the end, it is returning the percentage of the amount of data of each class.

Looking up the results and the model, I realized the data in the input of the model is not exactly as expected. There are inconsistencies in the results, for example as shown in the next two tables, these should have the same dimensions but here it is not the case:

TABLE 1: Results from subject chb14 file 6

	0	1	accuracy
precision	0.983	0.0	0.983
recall	1.000	0.0	0.994
f1-score	0.997	0.0	0.994
support	358.00	2.0	0.994

TABLE 2: Results from subject chb14 file 11

	0	accuracy
precision	1.0	1.0
f1-score	1.0	1.0
support	360.0	1.0

The two tables are from the same subject. These represent the results of testing files chb14_06 and chb14_11. Both tables are from are from the same subject 14. These should have the same size of dimensions in columns, defined by 22 electrodes plus the seizure's column and the windows column. The second dimension could change from file to file, because of the interruptions while recording as explained previously in this paper. The reason why the tables are different, one of them lacking from having the class 1 (class seizure, marked in red in the table), it is because when testing the model did not recognize a class as such. In further inspections surprisingly the data was only represented by one class, and as explained earlier, all files fed to the model where files with one or more seizures in them. All reports of testing should have strictly columns 0 and 1.

This problem is obviously an issue regarding data processing. Some files seem to be having problems with the dimensions of the numpy array. It seems to be working fine up until the labelling of the data. Once is read from the parquet and the data is set with the windows, then some problems seem to appear on the dimensions of the windows. My suposition is that the problem is on the equation regarding the overlapping and window sizing.

The size of the windows is set by a variable in a common dictionary with all the data processing characteristics needed to run the script, as well as the overlap. The size of the window is set for the model to be able to work within this window and not with the hole file, as it would be very resource intensive to execute and not as efficient to train. The size of the window is defined by a constant, in this case is set to 10 seconds, because generally, working with seizures the range of the window's size is around 5 seconds to 10 seconds. The equation could be having issues setting the values correctly, because depending on the length of the recording, some files have issues with this matter. In conclusion, the model is not receiving the data as it should.

6.2 Future Work

For future work it would be strictly necessary first of all, to change the chunker function setting the windows of all the files. Assuming the model works as it should, getting the right processed data would be enough to have a consistent result. If the result of the classification is not good enough, I would consider adding a weighted cross entropy loss to avoid so much one-sided data, for the model to learn more uniformly.

When training and testing the model, it would be a good idea to only do these two procedures. A good way to avoid extending training and testing execution time would be to normalize data before any model execution. This way all files, would be normalized considering all files and not just only one, like it is done in the execution script. As previously mentioned, the scalers are obtained by only considering one file. An average of scalers should be considered to keep in touch with the type of signals of every file, especially normalizing data between subjects, because seizure characteristics could change between different subjects, having a different impact on the model.

With better results the difference between models could be further proven, to understand the best strategy to obtain and process data. Not only two models, but all the models offered by the CVC (CNN_ConcatInput, CNN_ProjOut_Conv, CNN_ProjOut_Concat, CNN_ProjOut_AvgW, CNN_ProjChannel, CNN_ProjChannel_v2, Seq_C1D, Seq_C1D_Ensemble).

Regarding biological characteristics, other models that consider data sequences could be also utilised such as Dominant Sequence Transduction models, like states Attention is All you Need paper[8]. Creating a model and maybe consider classifying data in three classes: seizure, no seizure and pre-seizure. This would enable people to predict seizures and further understand the reason of them if these are linked to a sequence or pattern. This type of algorithm is used to understand sequences, such as natural language processing. If the existence of seizures is linked to a history of patterns that could be recorded in encephalograms and processed to predict when it is likely to be another seizure. Models could be trained and future devises could be programmed and created to at least advice or warn the subject a seizure is about to appear. Other problems could be avoided because the subjects could brace themselves, and take precautions such as seating down, closing the mouth, or simply get in a position more secure before the seizure happens.

Data processing with other ranges of frequencies would be another issue of research as well. In this paper only theta frequencies are considered and all other excluded, but there is also delta, alpha, beta and gamma frequency ranges to research with. It would be interesting to find out if a model could learn better from other bandwidths, and classify with the one's which give better results classifying seizures.

The position of the electrodes has been taken care of in the processing script, in a way where if there is an interruption and electrodes are changed, then the file was automatically excluded. But this could be changed to match the data of the previous positions of the electrodes and the positions of the one's after the interruption. This way more data could be used to train the models. As well, considering different Brodmann's areas (explained in the annex), to give more importance with weights to data coming from certain areas in the brain, which could cause or give better results when the model classifies it.

The model was trained with different subjects because there is not enough data to train with. It should be considered if different people have the same "type of seizure" affecting in the learning of the model. If enough data from only one subject was enough to train the model, it should be considered if this model could also predict seizures from other subjects with the same problem. The seizure itself might change between subjects, maybe it is alright to train the model for now with different subjects to consider many possibilities.

Finally other datasets should be researched as well to have a big overview of the difference in data between datasets regarding the seizure issue. It is hard to come by with any, but it could help a lot with the learning of models to find out how seizures come to be. Also, other scripts should be created to process data to input the models the same way as it is done in this study. For sure, this would be much more time consuming but with promising results.

7 Acknowledgments

In the production of this project there has been a lot of people involved helping me to make it possible. When I first started working with Aura Hernandez and Debora Gil, they provided me with the code of the models to change and adapt to the dataset CHB-MIT I have been working with in this study. Because this team is working on the Mental Workload Detection, they provided me with all sorts of scripts of models and data processing. I was able to make an idea of what I wanted, but because of the big difference on the datasets, data processing scripts had to be redone completely as well as modifying the models to fit and work with the new database.

Once I was working mid-way through this project, I started to have problems with my lack of knowledge on how Torch works. I desperately needed an in-depth insight on how the models were created and the reason of the internal structure. With Jose Elias we have been doing all kind of reunions to further understand these issues and also consider other strategies on how to process data more efficiently. Being able to work with someone who already has experience in this field of artificial intelligence and deep learning algorithms has given a boost with the development of this project and my own personal knowledge.

I would like to consider the effectiveness in response from Jordi Pons when there is a problem or simply any doubt. It is someone to surely rely on. He is always around and willing to help, this attitude is well known and appreciated by all.

Last but not least, special thanks to my teacher Aura

Hernandez helping me review, schedule and manage this project as important as it is. Training models, providing information and guidance on any topic, followed by a big interest in the subject and charisma, makes difficult times easier to overcome.

8 Bibliography

Referències

[1] Ary L Goldberger et al. "PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals". A: *circulation* 101.23 (2000), e215 - e220.

[2] M Villaret et al. "Mental Workload Detection Based on EEG Analysis". A: *Artificial Intelligence Research and Development: Proceedings of the 23rd International Conference of the Catalan Association for Artificial Intelligence*. Vol. 339. IOS Press. 2021, pàg. 268.

[3] Sandra G Hart. "NASA-task load index (NASA-TLX); 20 years later". A: *Proceedings of the human factors and ergonomics society annual meeting*. Vol. 50. 9. Sage publications Sage CA: Los Angeles, CA. 2006, pàg. 904 - 908.

[4] Saad Albawi, Tareq Abed Mohammed i Saad Al-Zawi. "Understanding of a convolutional neural network". A: *2017 international conference on engineering and technology (ICET)*. Ieee. 2017, pàg. 1 - 6.

[5] Mohsen Joneidi Mojtaba Taherisadr i Nazanin Rahnavard. "EEG SIGNAL DIMENSIONALITY REDUCTION AND CLASSIFICATION USING TENSOR DECOMPOSITION AND DEEP CONVOLUTIONAL NEURAL NETWORKS". A: (). URL: https://arxiv.org/pdf/1908.10432v1.pdf.

[6] Afshin Shoeibi et al. "Epileptic Seizures Detection Using Deep Learning Techniques: A Review". A: *International Journal of Environmental Research and Public Health* 18.11 (2021), pàg. 5780.

[7] Aura Hernández-Sabaté et al. "Recognition of mental workload of pilots in the cockpit using EEG signals". A: (). (Cons. 01-12-2021).

[8] Ashish Vaswani et al. "Attention is all you need". A: (2017), pàg. 5998 - 6008.

[9] *Synapse*. URL: https://en.wikipedia.org/wiki/Synapse (cons. 20-08-2021).

[10] *Sensory neuron*. URL: https://en.wikipedia.org/wiki/Sensory_neuron (cons. 20-08-2021).

[11] *Motor neuron*. URL: https://en.wikipedia.org/wiki/Motor_neuron (cons. 20-08-2021).

[12] *Projection fiber*. URL: https://en.wikipedia.org/wiki/Projection_fiber (cons. 20-08-2021).

[13] *Interneuron*. URL: https://en.wikipedia.org/wiki/Interneuron (cons. 20-08-2021).

[14] *Cerebrum*. URL: https://en.wikipedia.org/wiki/Cerebrum#Functions (cons. 20-08-2021).

[15] *Brodmann area*. URL: https://en.wikipedia.org/wiki/Brodmann_area (cons. 20-08-2021).

[16] *Seizure*. URL: https://en.wikipedia.org/wiki/Seizure (cons. 20-08-2021).

[17] *Neuron*. URL: https://en.wikipedia.org/wiki/Neuron (cons. 20-08-2021).

[18] *Lateralization of brain function*. URL: https://en.wikipedia.org/wiki/Lateralization_of_brain_function (cons. 20-08-2021).

[19] *Corpus callosum*. URL: https://en.wikipedia.org/wiki/Corpus_callosum (cons. 20-08-2021).

[20] *Cytoarchitecture*. URL: https://en.wikipedia.org/wiki/Cytoarchitecture (cons. 20-08-2021).

[21] H Nahrstaedt. *Lee-Messer C. holgern/pyedflib 2017.*

9 Annex

9.1 Neuron and neural activity

To further understand how brain activity works we first need to study a single neuron and its purpose. A neuron is an electrically excitable cell that has the function to communicate with other cells. It does it by nearly touching other cells called synapsis. It transmits the message through its axon and delivers the message by synapsis[9] to another cell. Neurons are typically classified into types based on their function:

- **Sensory neurons[10]:** Which respond to stimuli of the sensory organs and send the signals to the spinal cord or brain.
- **Motor neurons[11]:** Its axons originate in the brain and spinal cord and innervate the muscles to produce muscle movements.
- **Projection fiber[12]:** These types of neurons are found in the central nervous system and only establish synapses with other neurons, consisting of efferent and afferent fibers uniting the cortex with the lower parts of the brain and with the spinal cord.
- **Interneuron[13]:** Is a neuron of the central nervous system, usually small and with a short axon. It interconnects with other neurons, but never with sensory receptors or muscle fibres, allowing it to perform more complex functions.

Neurons transmit electrical waves originating from a transient change of permeability in the plasma membrane.

Their propagation is due to the existence of a potential difference that arises from different concentrations of ions on either side of the membrane, as described by the Nernst potential, between the inner and outer part of the cell (typically -70 mV). For the transmission of nervous impulses to other neurons, these do it by synapse, being a structure to pass electrical or chemical signals to another neuron or effector cell, there are two types of synapses:[9]:

- **Chemical synapse:** Electrical activity in the presynaptic neuron is converted into the release of a neurotransmitter that binds to the receptors located in the plasma membrane of the postsynaptic cell.

- **Electrical synapse:** Transmission between the first neuron and the second is not by the secretion of a neurotransmitter, but by the passage of ions from one cell to another through gap junctions. Small channels formed by the coupling of protein complexes, based on connexins, in closely adherent cells.

These electrochemical processes when large numbers of neurons show synchronized activity, electric fields that they generate can be large enough to be detected outside the skull, and so using electroencephalography (EEG) or magnetoencephalography (MEG) brain activity can be recorded.

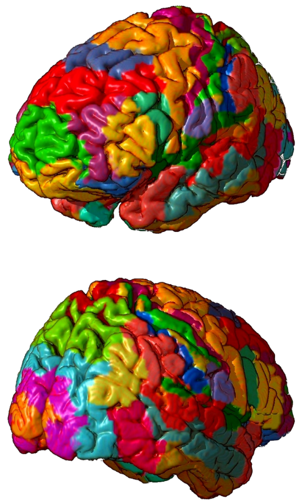
9.2 Structure

Now that we know where brain activity originates from, we can further study how the brain structures. There are many parts in the brain, but for now we are going to focus on the cerebrum because it initiates and coordinates movement, regulates temperatures, speech, judgement, reasoning, problem-solving, emotions, learning…

The cerebrum[14], it is the largest part of the brain, it is divided by the medial longitudinal fissure in two hemispheres. Each of these hemispheres, has an outer layer of grey matter the cerebral cortex, and an inner layer of white matter. The fact that these are separated gives the opportunity for lateralisation of brain functions, which is the tendency of neurological functions to specialise in one hemisphere or the other, but even though the cerebrum is separated, these are connected by the corpus callosum.

The cortex is mapped into fifty different functional areas known as Brodmann's areas[15], defined by its cytoarchitecture (cellular composition), or histological structure and organization of cells. One scheme widely used (from Korbinian Brodmann) splits the cortex into 52 different numbered areas of different cellular structure and different functions.

Fig. 7: Brodmann areas by colors



Having clarified this brain structure, obtaining data with electrodes from brain activity, and the positioning of these is something to keep in touch with depending on what it is being studied.