# 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 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’s not.

# 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.

As the main objective is to classify seizures from brain activity 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 just to have an overview to further understand the subject, but this matter is not the main purpose of this paper as it specifies in data processing, architecture, model strategies and classification results.

In this TFG we detail the complete process of epileptic seizure detection, from data processing to seizure recognition. Because this TFG works as a pipeline of different stages, an insight view of each is done, starting from data processing from a well-known database (CHB-MIT) of encephalograms collected from 23 subject with interactable seizures that has been used in previous research. Followed by the strategy of the architecture used to classify the signals to finally understand the classified results into seizure and not seizure.

# Related Work

A lot of research has been made of the brain to further understand it’s capabilities using deep learning algorithms. EEG signals can be used in many ways, below there are different projects with similar objectives of mental workload and seizure classification attempting to interpret and process EEG data, using deep learning techniques to develop frameworks to combine and process these signals.

Mental Workload Detection based on EEG Analysis

A study of mental workload is done in order to work more efficiently, healthier and to avoid accidents since workload compromises both performance and awareness. 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 this 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) 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: Providing a more reliable data by measuring physiological dynamic changes which cannot be controlled consciously, so that is why it’s more reliable. Readings such as electrocardiogram (ECG), electromyograph, electroencephalogram (EEG)... The combination of inputs reports better accuracy than the analysis if each one independently.

The approach is to investigate the ability of 1D-CNN models to recognise two types of mental load from EEG signals and to generalise the model to a population not seen in the training set. They use N-back test (memory demanding games requiring the resolution of simple arithmetic operations adjusting workload) to induce low and medium workload and to classify a simple neural network (NN) it’s trained using only the power spectrum of theta waves. Proposing a personalized model for each individual and a generalist one, preform the models in a dataset of 16 subjects showing outstanding results in a leave-one-out subject test and so generalizing to new unseen subjects.

The method to obtain data all 16 subjects had to first watch a 10-minute relaxing video and afterwards do the N-back-test low, medium and high difficulty. Finally subjects ask a TLX questionnaire for subjective perception of the test difficulty and workload.

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

In conclusion its 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.

EEG Signal Dimensionality Reduction And Classification Using Tensor

Decomposition And Deep Convolutional Neural Networks

This paper has a similar objective of seizure classification and uses the same dataset as this TFG.

Using convolutional neural networks (CNNs), they 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.

Epileptic Seizures Detection Using Deep Learning Techniques: A Review

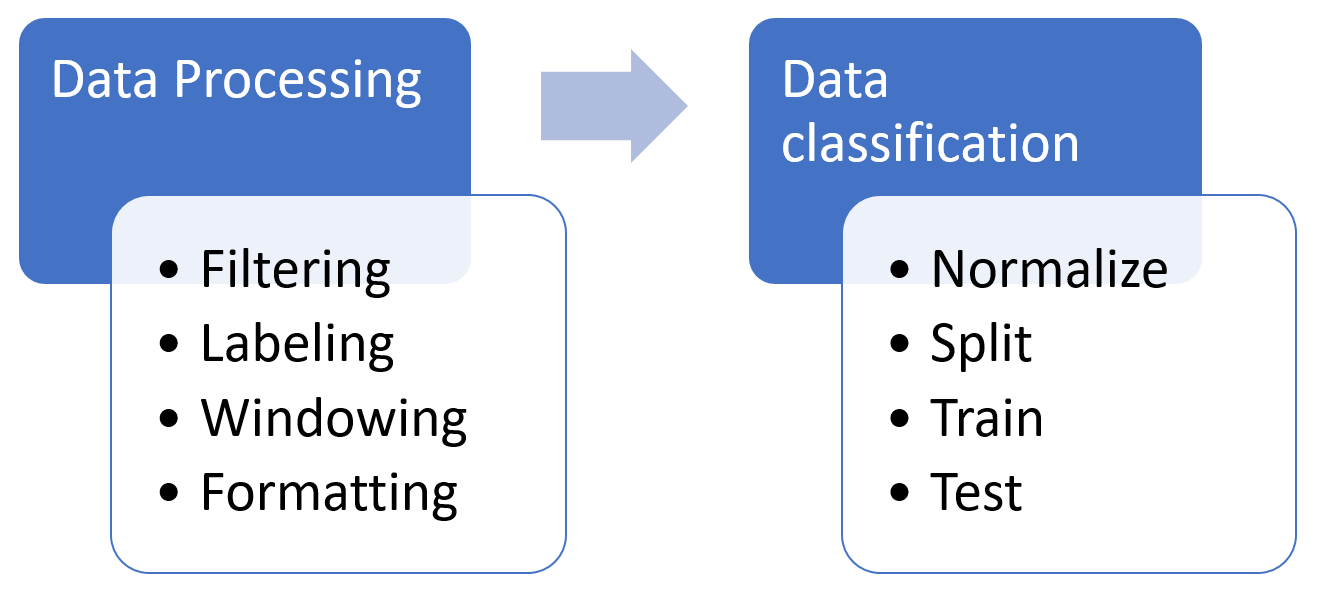
The advent deep learning algorithms in many areas of medicine, such as in the diagnosis of epileptic seizures, has made significant advances. In this study, a comprehensive overview of works focused on automated epileptic seizure detection using DL techniques and neuroimaging modalities is presented. Various methods proposed to diagnose epileptic seizures automatically using EEG and 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.

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 it’s 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. Our model 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.

# Objectives

The main goal in this TFG 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:



To fulfil the objectives is crucial to define smaller objectives to make work easier. Within the first objective there are several important parts.

* 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, so all files in the dataset are edf. A script has been programmed to save edf files into parquet format.
* Setting different functions to filter data making sure data fits certain constraints to obtain better results when training the model.
* Get the labelled data, to have a ground truth from the recorded data. This part is essential to understand if the model works as expected.
* Define different functions to define how data enters the model to be trained. There are many ways information can be extracted from data.
* Each model needs to be configured to accept the dimensionality of the data fed to it.
* Work different models to choose what models give better answers form input data.
* 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.

**Methods**

**Dataset**

The dataset is data collected from the Children’s Hospital 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.

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. It’s important to note, some subjects had hardware interruptions while the recording of the EEG, and so when there is an interruption, it’s 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 disposition of the electrodes and the EEG might not work for a sequential approach, for example, if it’s important there is 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 TFG 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 TFG is data from subject 1 to subject 10. Also only few .edf in each subject’s folders have any seizure. If every .edf was used, there would be a lot of data labelled as not seizure than seizure data, so from each subject only the files with seizure are used. Even so, handling only files with seizures the dataset is 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 un subsampling, considering there are few seizures in the hole dataset.
* When training the model, the criteria of cross Entropy should be weighted. To consider no seizure less important than seizure data.

# Data processing

Because in this TFG 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 “03\_ReadEDF.py” script there are different options on how raw data is imported, and also there are options on the way to execute them. During the development of this TFG many tests have been done, there for 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’s 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. Afterword’s it also labels the data and splits it into windows so the model can process data easily.

The script will automatically label all raw data using the summary file in each subject’s folder, so it’s 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, and afterwards only by changing the name of a parameter it can be changed to delta, theta, alpha, beta or gamma’s range frequencies in a dictionary added by default. Everything is modular so it can be changed any time with any range of frequencies. All data is saved in parquet format in a different folder, if plotting is enabled it will plot each subject’s data.

Once the data is filtered and labelled it`s saved two numpys, 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.

# Network

An already done deep learning algorithm is used from the research group IAM from the CVC, 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 this case, having 14 sensors x 5 wavelengths, so 70 raw signals. These signals can be concatenated, or projected.
* 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’s changed to fulfil the objective of clinic seizure detection. In this TFG, different strategies are applied on the input data of the algorithm to further study it’s capabilities as well as using different models to compare results between them.

Once all desired raw data is filtered and saved, the second script to execute is “04\_MExecution.py”, this one 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 different typs of execution for development that can be enabled by boolean variables commented 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 selected model and feeds to it random data using torch with the defined parameters in the function.
* If the model works with no isues with CheckModel, then the training execution can begin. This first phase selects every numpy from n subjects defined at the beginning and trains the model with all the data. Every time the model finishes the epochs of a file, it saves the model in the database.
* In the second phase the previous trained model is loaded to test it and data is defined to set the normalization scalers for testing. 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 using sklearn library and saved to a 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 TFG 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. For the executions done this variable is set to 1 (100%) on training, and 0 on testing phase.

During the implementation of this script, a lot of problems of dependency on critical libraries has happened. Starting by cuda, for faster results it 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 was the version being worked on at the moment, it had to be switched to an environment with python 3.8 to avoid further issues. The script will be executed with cuda if the libraries are available, if not it will automatically work with the CPU.

When it’s time to save the model, it’s saved in a folder called “trys” in the root directory, like if it was another subject but it only contains pt files, ordered in it by the date of execution.

The models used in the TFG have similar structures. The first one 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 by a function ConVNet which defines 3 different layers using torch, the first one being 16@22x19. The first parameter as number of input features defined by a constant variable defined in the constructor of the model, the second parameter defines the size of the data, 22 being the channels, in this case the electrodes used on the subjects. The second layer is 32@22x9 and the last layer 64@22x4. As seen the dimension is being reduced in the convolutional unit and still one last reduction is done to 64@1x4 before the pooling.

Execution

The execution of the model 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 and a (3,1) kernel size.

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. Because there are few seizures in the hole database it makes it difficult to train a model to understand the existence of a seizure. Also, it was dependent on the position of data in the file, considering there are few or in most cases only one seizure, 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 will never be able to learn what a seizure is, and if its opposite, it might learn what a seizure is but it would never be possible to test if it learned correctly to classify.

In the end the strategy of first training the model with some files and then with others do the testing, was the best way to go considering training each file took around 10 min depending on the computer it was executed with. There was a big difference between executing the model with and without cuda, the first model was executed without cuda because of the hardware architecture and it spent around 30 hours training the model, and with cuda half the time. It’s important to note using a desktop computer to train the model, considering it will get hot finishing the last epochs of every file.

It’s also very resource consuming, considering it loads all the file to the RAM and reads and works with it from there. It could be a problem to many computers, some with 8Gb won’t work, only with 16Gb of memory worked for me. In most cases, the .edf files contain 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’s also excluded.

Results

After training the two models with files from subjects 1 to 10, testing only took 15 to 30min 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 “05\\_ReadParquet.py” script, which reads every parquet from the folder results of every tested subject.

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”. So, the other 5% it’s the Seizure in the data that the model is classifying it as no-seizure. Because data is mainly one-sided to no-seizure the result 95% is because it was only classifying this class, so in the end it’s 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 the two tables should have the same dimensions but here it’s not the case:

The two tables are from the same subject, these represent the results of testing files chb14\_06 and chb14\_11. The two results are from two different files from the same subject 14, files 6 and 11 respectively. The files should have the same size of dimensions in columns, defined by 22 electrodes plus the seizure’s column and the windows column, but 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 (seizure, in read in the table) it’s because when testing the model didn’t 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, and 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. I think the problem is on the equation regarding the overlapping of the windows.

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. 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. Also, there is a constant that sets the overlap for each window. 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 and so the model is not receiving the data as it should.

Future Work

For future work it would be strictly necessary first of all, to change the chunker function defining the windows of all the files. So 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.

With better results the code could give more accurate results so the difference between models could be further proven, to understand the best strategy to obtain 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 to create 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.

Data processing with other ranges of frequencies would be another issue of research as well, because in this paper only theta frequencies are considered and all other excluded, but there’s also delta, alpha, beta, gamma to research with, and it would be interesting to find out if a model could learn better from other bandwidths, and classify which one’s are better for 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 as maybe considering different Brodmann’s areas (explained in the annex), to give more importance to data coming from certain areas in the brain which could cause or give more information about the reason, detonation or existence of seizures.

As this paper the model was trained with different subjects because there’s 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 in other subjects with the same problem. The seizure itself might change between subjects, so maybe it’s 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’s hard to come by with any, but it could help a lot in the learning of models to find out how seizures come to be. Also, other scripts should be created to process data for the model to input the same way, with the same characteristics, as it’s done with this dataset. For sure, this would be much more time consuming.

Acknowledgments

In the production of this TFG there has been a lot of people involved helping me to make this project as it is right now. When I first started working with it Aura Hernandez and Debora Gil provided me with the code of the models to change and adapt to the dataset I have been working with in this TFG. Because this team is working on the Mental Workload Detection the they provided me with al sorts of scripts in models and data processing. I was able to make an idea of what I wanted, but because of the big difference on de datasets, the data processing scripts had to be redone completely.

Once I was working with this TFG, mid-way I started to have problems in my lack of knowledge on how torch works and I desperately needed an in-depth insight on how the models were created and the reason of the internal structure. With Elias we have been doing all kind of reunions to further understand these issues and also considering other strategies on how to process data more efficiently.

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

Last but not list special thanks to my teacher Aura Hernandez helping me review, schedule and manage this project as important as it is. Much appreciated everyone who has showed interest in this project. I wouldn’t have done it, with any doubt, without the help of people so professional I have been working with, thank you all.

**Annex**

**Neurons**

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, transmits the message through its axon and delivers the message by synapsis to another cell. Neurons are typically classified into types based on their function:

* Sensory neurons: Which respond to stimuli of the sensory organs and send the signals to the spinal cord or brain.
* Motor neurons: Its axons originate in the brain and spinal cord and innervate the muscles to produce muscle movements.
* Projection fiber: Are neurons found in the central nervous system and only establish synapses with other neurons, these consist of efferent and afferent fibers uniting the cortex with the lower parts of the brain and with the spinal cord.
* Interneuron: Is a neuron of the central nervous system, usually small and with a short axon, that 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 due to 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:

* 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: Is one in which 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.

Brain

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, it’s the largest part of the brain, it’s 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, 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.

Having clarified this brain structure, obtaining data with electrodes from brain activity, positioning of these is something to keep in touch with depending on what it’s being studied. The same goes for defining the dataset train and test data, to feed as input to the deep learning algorithm, opening a new field on how to treat and subdivide data to make the most of it.

Apuntes

Diferent tasks to be done:

\begin{figure}[!h]

  \centering

    \includegraphics[width=0.4\textwidth]{img/Tasks.png}

    \caption{General Tasks of the TFG}

    \label{fig-exemple}

  \end{figure}

Note that “Write Work Reaserch” tasks are before every meeting.

\leavevmode\\

\begin{figure\*}[!h]

  \centering

    \includegraphics[width=1\textwidth]{img/Gantt.png}

    \caption{Gantt Diagram of the project}

    \label{fig-exemple}

  \end{figure\*}

Diferent tasks to be done:

Research architecture of models

Understand .edf files

Study different data inputs

Define data classification for input

Adapt Deep Learning algorithm

Obtain any result and Debug

Obtain intelligible results

Create own Neural network

Compare results

Try other architectures

Finalize report, clear other issues

Create poster

Write Work research X

References

<https://en.wikipedia.org/wiki/Neuraloscillation>

<https://en.wikipedia.org/wiki/Seizure>

<http://en.wikibooks.org/wiki/LaTeX>

<http://www.acnweb.org/acta/2002182104.pdf>

<https://arxiv.org/pdf/1908.10432v1.pdf>

<https://paperswithcode.com/dataset/chb-mit>

<https://en.wikipedia.org/wiki/Neuron>

<https://en.wikipedia.org/wiki/Projection_fiber>

<https://en.wikipedia.org/wiki/Interneuron>

<https://en.wikipedia.org/wiki/Motor_neuron>

<https://en.wikipedia.org/wiki/Sensory_neuron>

<https://en.wikipedia.org/wiki/Synapse>

<https://en.wikipedia.org/wiki/Cerebrum#Functions>

<https://en.wikipedia.org/wiki/Lateralization_of_brain_function>

<https://en.wikipedia.org/wiki/Corpus_callosum>

<https://en.wikipedia.org/wiki/Cytoarchitecture>

<https://archive.physionet.org/pn6/chbmit/>?

Michael Petrides (3 December 2013). [*Neuroanatomy of Language Regions of the Human Brain*](https://books.google.com/books?id=DYlqAAAAQBAJ&pg=PA90). Academic Press. p. 90. [ISBN](https://en.wikipedia.org/wiki/ISBN_(identifier)) [978-0-12-405931-3](https://en.wikipedia.org/wiki/Special:BookSources/978-0-12-405931-3).