Monterrey Institute of Technology and Higher Education
Campus Queretaro
Computing and Mechatronics Department

Use of Deep Learning and EEGs to Diagnose ADHD in Kids

by

Sandra Román Rivera

Integrative Project for the Development of Business Solutions

Computer Systems
Engineering

Adviser: Dr. José Antonio Cantoral Ceballos

Queretaro, Mexico
05/16/2022

"I do not fear computers. I fear the lack of them."

Isaac Assimov

"You can't change who you are and you shouldn't be asked to."

Jonathan Mooney

"I think people need to understand that deep learning is making a lot of things, behind-the-scenes, much better."

Geoffrey Hinton

Acknowledgements

First of all, I'd like to thank Dr. José Antonio Cantoral for all the support and shared knowledge throughout the development of this project. This truly wouldn't have been possible without him.

I'd also like to thank my parents because they've given me everything. If it weren't for them, I wouldn't have been born in the first place. So thank you mom and dad, you're awesome!

Also a huge thanks to all the teachers I encountered throughout my school years for sharing all of their passion and knowledge with me. You rock!

Last, but not least, I'd like to thank my husfriend (husband/boyfriend) and my siblings for always making me feel bad when I procrastinated doing this project. I wouldn't have finished on time if it weren't for you, guys. Thank you!

Abstract

Attention Deficit Hyperactivity Disorder (ADHD) is a neurodevelopmental disorder that affects 5% of the world's population. It has been proven that patients of all ages with this condition can be distinguished from those without it by performing an electroencephalogram (EEG) of the patient and analyzing it. Performing a manual analysis of an EEG is a laborious and slow process. However, analyzing EEGs using Deep Learning techniques has been successful in many fields. There are existing deep learning models that diagnose effectively epileptic seizures and adult ADHD using EEGs. However, no models to diagnose child ADHD were found. A fully connected neural network model was developed to test the dataset to be used was useful to solve the problem. This model obtained a 87.71% accuracy when classifying EEGs of kids with and without ADHD. Then, the models found for adult ADHD and epileptic seizure diagnosis were used as a base to develop a Convolutional Neural Network (CNN) model that differentiates kids with ADHD from kids without it with an accuracy of 99.21%. The dataset was divided into smaller samples for both models due to the small number of samples on it. The results obtained exceeded the expected accuracy of 89.12%. However, further tests and a larger dataset are needed to prove the results are solid.

Contents

Abbreviations

ADHD	**A**ttention **D**eficit **H**iperactivity **D**isorder
AI	**A**rtificial **I**nteligence
CNN	**C**onvolutional **N**eural **N**etwork
DL	**D**eep **L**earning
EEG	**E**lectroencephalogram
ERP	**E**vent-**R**elated **P**otential
LR	**L**earning **R**ate
ML	**M**achine **L**earning
NN	**N**eural **N**etwork
RTV	**R**eaction **T**ime **V**ariability
SMR	**S**ensorimotor **R**ythm
TBR	**T**heta-**B**eta **R**atio

Chapter 1

Introduction

”Normal is overrated.” Lisa Aro

Attention-deficit/hyperactivity disorder (ADHD) is one of the most common mental disorders affecting children. It is more common among boys than girls. An estimated 8.4% of children and 2.5% of adults have ADHD. This disorder is often first identified in school-aged children when it leads to disruption in the classroom or problems with schoolwork. Symptoms of ADHD include inattention, hyperactivity and impulsivity. For this reason, children with ADHD might not be able to keep focus, they might present excess movement that is not fitting to the setting and/or perform hasty acts in the moment without thought. While many of these symptoms are common to young children in general, children with ADHD's hyperactivity and inattention are noticeably greater than expected for their age and cause distress and/or problems functioning at home, at school or with friends [1][2].

Even though ADHD can be truly prejudicial to school age children, there is no lab test to diagnose ADHD. Diagnosis involves gathering information from parents, teachers and others, filling out checklists and having a medical evaluation (including vision and hearing screening) to rule out other medical problems [1][2]. While early ADHD diagnosis is truly beneficial, specially for kids, there's not an objective and efficient way of diagnosing it.

This paper will be about children ADHD and how it can be diagnosed by using a brain activity test called electroencephalogram (EEG) and deep learning (DL) techniques. The structure will be as follows:

Chapter 2 will touch on ADHD and EEG definitions and how they work together to diagnose ADHD. Electroencephalographic differences found between people with and without ADHD will be discussed as well as advaces in electroencephalogram analysis using machine learning. Similar models to the ones developed for this project will be mentioned, specifically one that diagnoses adult ADHD [3] and another one detecting epileptic seizures [4].

Chapter 3 will have a detailed description of the developed CNN model, how it was trained, the dataset used and how it was formatted, the technique used to find the best learning rate (LR) to enter to the model, etc. Relevant code on the development of the CNN and other functions mentioned in this chapter can be found in Appendix A. The evaluation method to be used to determine the success of the model will also be included in a subsection of this chapter. This includes the description of the different experiments to be done.

Chapter 4 is all about the results of the tests described in the previous chapter. The experiments were done first on a fully connected neural network model to test if the selected dataset would work for the needs of this project and later on to the convolutional neural network to get the final results. This will include all the obtained graphs and tables gotten from all experiments. The interpretation of the results gotten is also included here.

Finally, chapter 5 contains the conclusions gotten after finishing the investigation and the work that is thought to be needed in order to take this project to the next level.

1.1 Problem Statement

Having compared the most advanced work done in the state of the art (chapter 2), it's clear that there's a problem that hasn't been tackled yet. Here I explain in detail why said problem hasn't been solved and why it's important that we solve it.

As it was mentioned previously, ADHD affects kids, specially in those in school age, in a negative way. It interferes with their social and learning abilities [5]. Once children are diagnosed with ADHD, certain accommodations can be made for it's negative effects to be diminished and let these kids develop in a smooth way. However, ADHD diagnosis continues to be merely clinical and subjective [3] [6]. Despite studies showing that there are electroencephalographic differences between kids with and without ADHD [5], manual inspection of the brain's electroencephalographic signals is a truly laborious work that takes a lot of time [4].

There's a model that uses convolutional neural networks to distinguish adults with ADHD from adults without it [3]. Even though studies show that EEGs are also useful to make this same classification, but in kids [5], a model that fulfills this task hasn't been found.

1.2 Objectives

The main objective of this project is to develop a deep learning model that analyzes electroencephalographic signals to diagnose ADHD in kids.

This objective can be divided into the following secondary objectives:

1. Develop a fully connected neural network model that classifies a kid's EEG to tell if he/she has ADHD.
2. Develop a convolutional neural network model that performs the same task, but with greater accuracy.
3. Compare the developed models to state of the art models on similar fields and achieve greater than or equal performance.

1.3 Hypothesis

A convolutional neural network model is able to classify a kid into ADHD or control group using electroencephalographic signals achieving an accuracy greater than or equal to models trying to achieve similar tasks.

Chapter 2

State of the Art

This chapter describes Attention Deficit Hyperactivity Disorder (ADHD) and Electroencephalography. These concepts are necessary to understand how electroencephalograms (EEGs) can be used to diagnose ADHD and how Artificial Intelligence helps in the classification of EEGs. Then, an analysis of some of the most important work on the area will be made. With this analysis, we'll be able to see the areas in which there's still work to be done to detect ADHD in kids and what contributions can be made on the subject.

2.1 Prior Knowledge

2.1.1 Attention Deficit Hyperactivity Disorder

Attention Deficit Hyperactivity Disorder is a heterogeneous neurodevelopmental disorder [3] characterized by symptoms like lack of attention, hyperactivity and impulsivity [7] [6]. This disorder affects 5% of the world's population [3], both infant and adult. Since this concept was developed, a biological indicator to diagnose it has been searched, but it continues to be merely clinical and subjective [3] [6].

ADHD affects more noticeably kids in school age since it has repercussions in the way they learn, communicate and behave. It even alters their motor skills [5].

2.1.2 Electroencephalograms

An electroencephalogram (EEG) is a non-invasive test used to measure electric activity on the brain. The signals are obtained using electrodes that are placed in specific parts of the patient's scalp. Figure 2.1 shows how each electrode obtains the signal from the zone of the brain that is closest to where it was placed. The test is performed by a technical specialist in encephalographies in a medical office, a hospital or in a laboratory [8].

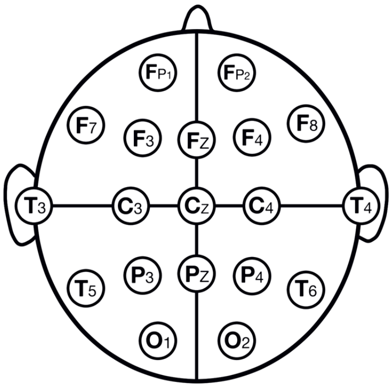

FIGURE 2.1: Example of electrode positions on an EEG [9]

In figure 2.2 we can see an example of the signals obtained when performing an EEG. Each line represents the signal obtained from each electrode through time. Said figure has 16 lines, so that means that the performed EEG it corresponds to obtained signals from 16 different electrodes. It is then said that it is a 16-channel EEG.

On an EEG wee can find different types of neural oscillations, these are measured by their frequency range. The states and frequencies that compose the vast majority of our day-to-day experiences are found between 0,5 and 90 Hz [10]. In figure 2.3 we can observe the different types of waves we can find on an EEG and what each one looks like.

2.1.2.1 Electroencephalography in the Diagnosis of ADHD

Electroencephalography represents a widely established method to evaluate altered functions of the brain from typical ones [11]. This is because it offers beneficial information about the

FIGURE 2.2: Example of signals obtained from an EEG [10]

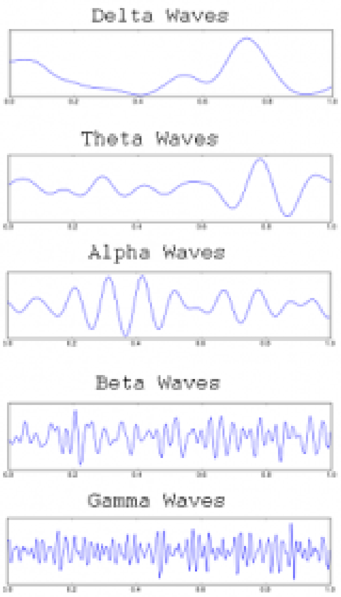

FIGURE 2.3: Example of the different types of waves on an EEG [10]

way the brain works with a higher temporal resolution to detect brain disorders like epileptic seizures, dementia, schizophrenia, sleep apnea and ADHD [12].

An EEG is an appropriate instrument to diagnose ADHD because of the neurobiological and temporal aspects of the symptoms of this disorder [7]. Nowadays, indicators like event-related potential (ERP), reaction time variability (RTV), Theta-Beta ratio (TBR) and sensorimotor rhythm (SMR) can potentially help in the neurobiological characterization of ADHD and help identify pathognomonic and/or endophenotypic indicators based on an EEG that allows its diagnosis, treatment and control [7]. Electroencephalographic signals obtained from opened and closed-eyed patients in resting state are used predominantly to analyze ADHD and its variants in kids and teenagers [12].

2.1.2.2 Electroencephalogram Analysis with Machine Learning

Electroencephalogram analysis is an important tool in neuroscience and neuronal engineering. It even has commercial applications. Many of the analysis tools used in the study of electroencephalograms (EEGs) use machine learning to find relevant information in the neuronal classification and neuroimaging. Current availability of big EEG datasets and advances in machine learning have led to a better analysis of electroencephalographic signals and understanding of information they may contain on brain functions. Automatic classification of this signals is an important step to a more practical use of EEGs through different applications and to making it less dependent on professionals on the area [13].

Artificial intelligence comprehends many areas, one of its branches is deep learning. One of the characteristics of deep learning is that attribute extraction and classification are done on a completely automatized way [14]. For this same reason, this technique benefits the EEG analysis because inspecting an EEG manually is a laborious process that takes a lot of time [4].

2.2 Relevant Work

A considerable amount of investigation work has been done making use of EEGs to detect epileptic seizures with various soft computing techniques, wave transformation methods, and artificial intelligence techniques, However, not much work has been done to diagnose ADHD and its different types [12].

In the area of diagnosing ADHD using electroencephalograms, the following was found:

- **Adults:** Studies show probably unique differences in electroencephalographic activity among the adult population with ADHD. High levels of power in theta waves besides alpha waves activity observation can differentiate adults with ADHD from those without ADHD [6].

- **Teenagers:** It's been proved that those with ADHD present low power and connectivity on the lower alpha band and an elevated power on the upper alpha band compared to those without ADHD. Power on the EEG of the lower alpha band is inversely associated to the severity of ADHD in the patient [15].

- **Kids:** It's been referred that kids with neuropsychological and language disorders can present epileptiform anomalies on an EEG. Studies made by the Ramon y Cajal Hospital in Spain show that EEGs of kids with ADHD show significant anomalies, while on the polysomnographic register, double the patients with ADHD show specific alterations. The need arises to make a night polysomnography to determine ADHD cases [5].

It is obvious that EEGs can be used to distinguish patients with ADHD from those without it, no matter their age.

When it comes to the use of artificial intelligence for the classification of EEGs, it's been proved that convolutional neural networks, recurrent neural networks and deep belief networks show higher accuracy levels than stacked auto-encoder and multi-layer perceptron neural networks [13]. Furthermore, tasks that use deep learning for this, have been classified into 5 general groups: emotion recognition, motor images, mental workload, seizure detection, ERP detection and sleep score [13]. This indicates that the use of convolutional and recurrent neural networks could produce models with greater accuracy because EEG classification is required and ADHD detection comprehends tasks that fall into the mental workload and ERP detection categories.

Harvard Medical School developed a model using convolutional neural networks (CNNs) with a four layer architecture to differentiate adults with ADHD from adults without it. The model was trained using spectrograms of electroencephalographic data from ERPs. An accuracy percentage of 88% $\pm$ 1.12% was achieved when classifying. They evaluated the use of recurrent and shallow neural networks as well, but convolutional neural networks got the best results. On the same way, resting state spectrograms were evaluated, but event-related ones achieved a higher accuracy. While there's still the need to test with more clinical samples, results suggest that deep neural networks are a useful tool for analyzing dynamic EEGs even with small datasets [3].

The main advantage of this model is that it can distinguish patients with ADHD and patients without this disorder. The problem is few clinical samples were used and this samples were taken only from adults. For this same reason, there's no certainty that this model will be capable of doing this same classification, but with kids. Furthermore, the model was trained using spectrograms, which require EEG preprocessing. When it comes to the choice of using CNNs, while this model learns the internal structure of the data and has better performance than techniques designed by hand making use of engineering, the biggest problem is the large amount of parameters that need to be learned and that learning them requires a lots of data. A 1D CNN uses less parameters compared to 2D CNN models and, like so, it achieves a better generalization [4].

To detect epileptic seizures, a P-1D-CNN developed by Ullah, Hussain, Qazi, and Aboalsamh obtains an accuracy of 99.1% $\pm$ 0.9%. In the same way, this model uses less memory space and its detection time is really short (less than 0.000481 seconds). Said model promises not only be useful to detect epileptic seizure, but also to develop expert robust systems for similar conditions [4]. Trusting this promise, this same model could be used to efficiently diagnose ADHD.

2.3 Advances in ADHD Diagnosis on Kids Using EEGs and DL

The most interesting work in this case has been made by Harvard Medical School and Ullah, Hussain, Qazi, and Aboalsamh. The models they both developed have outstanding accuracies.

Harvard's CNN is great for diagnosing adults with ADHD [3]. On the other hand, the 1D CNN developed by Ullah and the others is great for detecting epileptic seizure and is promising for diagnosing ADHD as well [4]. However, none of them can certainly diagnose ADHD when we talk about kids.

2.4 Conclusion

What's most important about this chapter is that among all the analyzed projects and models, none of them fully satisfy the need of diagnosing ADHD in kids. In order to do that, a model that classifies kids with and without ADHD needs to be developed. This model would have to achieve accuracies similar to those found on the studied models.

Chapter 3

Solution

To solve the problem of kids not having an efficient and objective way of being diagnosed with ADHD, a one-dimensional convolutional neural network was created. A 1D CNN was chosen because it uses less parameters compared to 2D CNN models and, like so, it achieves a better generalization. Furthermore, this type of neural network was successfully implemented to detect epileptic seizures and the same study suggested it would be useful to treat similar brain conditions like ADHD [4]. The neural network was developed based on Dr. José Antonio Cantoral's video tutorials on convolutional neural networks using PyTorch [16][17]. That, and it being a Facebook framework, was what motivated me to use PyTorch (I'll soon be a Facebook employee so learning the basics of PyTorch will be beneficial for my career). However, other frameworks can be used to achieve similar results. The model has three layers: two one-dimensional convolutional layers with a ReLU activation after each one and then a linear or fully connected layer. The data is passed through a one-dimensional max pool function and is flattened before it is introduced to the linear layer. Both convolutional layers have a kernel size of 3 and a padding of 1 to stick to the output size. The code that initializes the model can be found in Appendix A under the title "Initialize Model".

The code used to train the model can be found in Appendix A under the title "Train Model". The function train receives four parameters: a model, an optimizer, a scheduler and the number of epochs the model will be trained for. The initialization of the used optimizer and scheduler can be found in Appendix A under the title "Initialize Optimizer and Scheduler". Each epoch trains the model using the train loader which is divided into a certain number of mini-batches.

The cost and accuracy of both the train and validation data is calculated and printed for each epoch. The train cost and accuracy are calculated through each epoch adding those of each mini-batch. The validation cost and accuracy are calculated at the end of each epoch using the function accuracy which can be found in Appendix A under the title "Calculate Accuracy and Cost". The cost in both cases is calculated using an average of all cross entropies while accuracy (also in both cases) is calculated getting the percentage of samples that were predicted correctly.

The learning rate introduced in the scheduler made a huge difference when training the model. To find the best learning rate to use, the find_lr function was used. This function can be found in Appendix A under the title "Find Best Learning Rate". This way of finding the best learning rate is based on Dr. José Antonio Cantoral's video tutorials on super-convergence method [18][19]. The basic takeaway is that losses and accuracies for different learning rates ranging from a start value to an end value are calculated. These losses and accuracies are returned by the function and then, the easiest way to know what learning rate should be used is to graph them. Figure 3.1 and 3.2 show an example of this graphs gotten from one of experiments made for this paper. The best learning rate to use is the one with minimum loss and maximum accuracy. The accuracies graph is too noisy in this case, so the losses graph will have higher weight and so the best learning rate given from these graphs would be around 5e-5 (which is marked with a red line).

The dataset used to train the model consists of EEGs from 61 children with ADHD and 60 healthy controls (boys and girls, ages 7-12). The ADHD children were diagnosed by an experienced psychiatrist to DSM-IV criteria, and have taken Ritalin for up to 6 months. None of the children in the control group had a history of psychiatric disorders, epilepsy, or any report of high-risk behaviors. EEG recording was performed based on 10-20 standard by 19 channels (Fz, Cz, Pz, C3, T3, C4, T4, Fp1, Fp2, F3, F4, F7, F8, P3, P4, T5, T6, O1, O2) at 128 Hz sampling frequency. The A1 and A2 electrodes were the references located on earlobes. Electrode positions of the mentioned channels are represented in 3.3 which for convenience is repeated here. Since one of the deficits in ADHD children is visual attention, the EEG recording protocol was based on visual attention tasks. In the task, a set of pictures of cartoon characters was shown to the children and they were asked to count the characters. The number of characters in each image was randomly selected between 5 and 16, and the size of the pictures was large enough to be easily visible and countable by children. To have a continuous stimulus during the signal

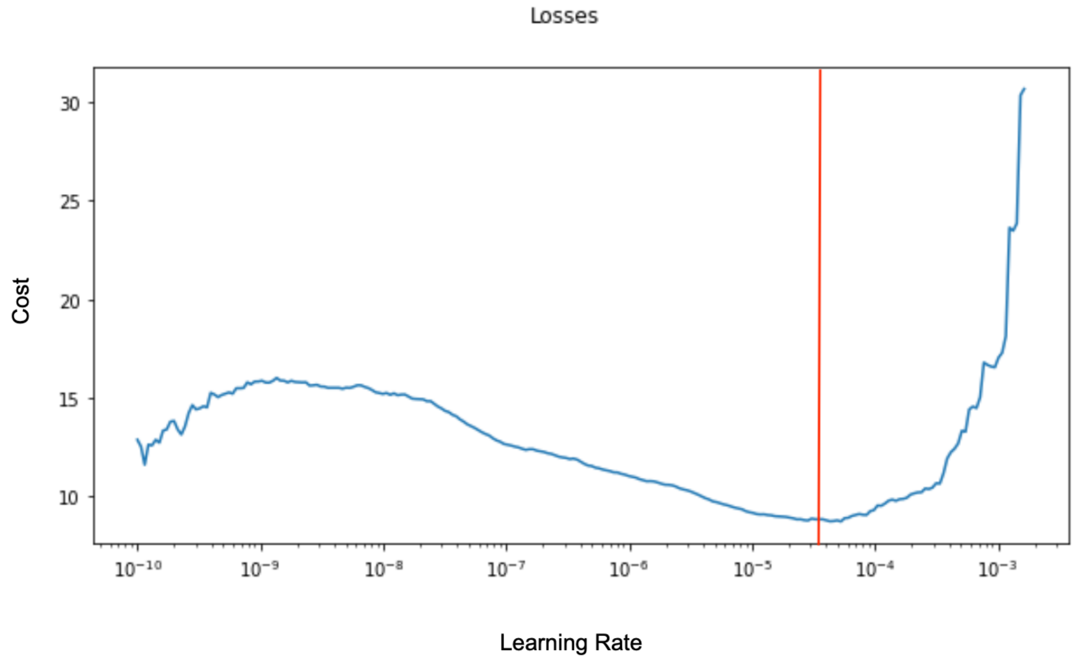

FIGURE 3.1: Example of losses graph to find the best LR

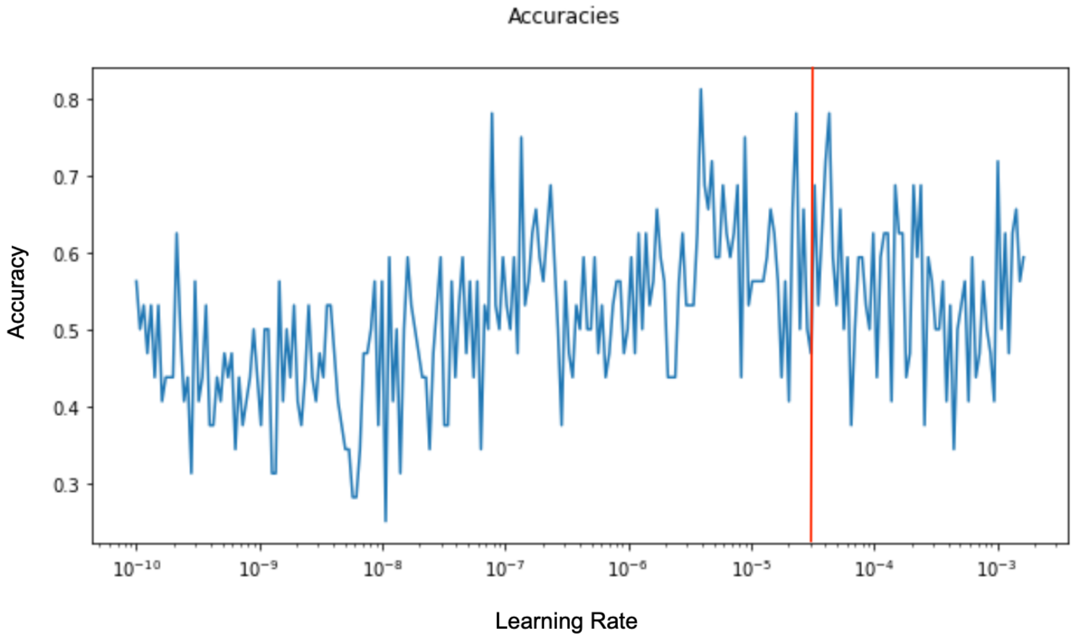

FIGURE 3.2: Example of accuracies graph to find the best LR

recording, each image was displayed immediately and uninterrupted after the child's response. Thus, the duration of EEG recording throughout this cognitive visual task was dependent on the child's performance (i.e. response speed). [20].

Said dataset consists of only 121 EEGs of variable duration. Since 121 is a small number of

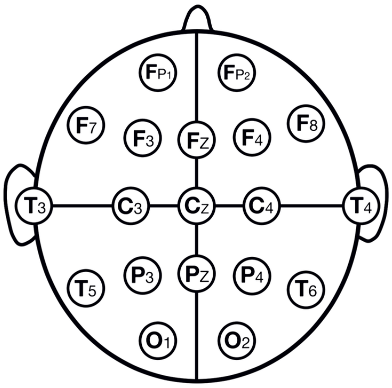

FIGURE 3.3: Example of electrode positions on an EEG [9]

samples, each EEG was splitted into pieces with a duration of one second each. For example, if the EEG of a kid with ADHD was 2 minutes long, the EEG was divided into 120 EEGs, each corresponding to a kid with ADHD (so instead of having a single EEG, we're left with 120 smaller EEGs). That was done for all of the 121 EEGs. The code used to load the dataset and split it both into smaller EEGs and into train, test and validation datasets can be found in Appendix A under the title "Load and Split Dataset". It's important to note that the datasets were also splitted into smaller mini-batches to take advantage of the GPU and train and evaluate the model faster. The mini-batch size used was 64. The training dataset consists of 70% of the data while the other 30% was splitted equally between validation and test datasets.

After doing all the splits, the number of one-second EEGs increased to 16,870. From those shuffled ADHD and Control mini EEGs, 11,809 composed the train dataset, 2,531 the validation dataset and 2,530 the test dataset.

3.1 Evaluation Method

To evaluate if the solution works, different tests will be made. This tests will be measured in the same way the related models [3][4] mentioned on Chapter 2 were evaluated: calculating the model's accuracy. In order to say the model is able to diagnose kids with ADHD correctly, it will need to achieve greater or equal accuracies than those of the related models. Those accuracies are 88% ± 1.12% for the adult ADHD model [3] and 99.1% ± 0.9% for the epileptic seizure model [4]. So, an accuracy greater than or equal to 89.12% is expected.

First, a fully-connected neural network (NN) will be developed just to check if the dataset described in Chapter 3 will do the trick. If the fully-connected NN is able to get an accuracy greater than 70%, we can assume it is learning to differentiate the ADHD group from the Control group. For this, different lengths will be used to split the 121 EEGs of the datasets. The durations to try are 1, 5, 10, 15, 20, 25 and 30 seconds. This different splits will also be tested each with different mini-batch sizes ranging from 8 to 512 samples per mini-batch. The length and mini-batch size that produces the highest test accuracy will then be further explored by experimenting with learning rates that are different, but close to the best learning rate found for the highest accuracy case.

If the fully-connected NN learns as expected and exceeds the expected 70%, a CNN with the characteristics specified in Chapter 3 will be tested. Similar tests than those of the fully-connected NN will be used to test the CNN, but the duration of the split won't change, the one that got the best results for the fully-connected will be used. The mini-batch size will be reduced to two different sizes: the one that got the best results as well and the one that results from multiplying that size by two. The mini-batch size that gets the best result will be further explored also by using different learning rates that are close to the best learning rate gotten from the function explained in Chapter 3. The best result will be chosen and compared to the goal which is a test accuracy equal or greater than 89.12%.

Chapter 4

Results and Interpretation

The experiments gave great results. However, there are a couple of things to note: the split of the samples of the dataset might be affecting results. The split was made due to the limited amount of data that was found. Having a larger dataset would have been really beneficial for the experiments. Also, due to missing calculations, some of the tests have varying number of loops for which the model was trained for and, therefore, the model was trained longer for some experiments which might result in the other experiments looking less accurate than they should. However, the results were great so there was no need to redo said experiments. If someone wants to redo them for peace of mind, that's fine, but it doesn't seem to be necessary at all.

Now here are the so talked about great results. The first experiment was made for the fully connected neural network. In this experiment, the best learning rate was calculated for each test. Table 4.1 shows the results obtained for said experiment. As we can see, the highest test accuracy was 80% using a one second split and a mini-batch size of 256 (test #2). In figure 4.1, it is more clear that the results from tests with split length larger than one are not that suitable for this experiment. We can observe the huge differences between the train and test accuracies. On most tests, both validation and test accuracies were below 70% while the train accuracy reached 100%. The model presented overfitting problems. The split length to be further explored was chosen to be one second because it showed less signs of overfitting and it got the highest test accuracy (80%) which exceeded the expected 70% for this test. The model was able to classify the splitted EEGs in an acceptable way. The next model is expected to work as expected then, but by further exploring this model we'll be able to see if the goal of an accuracy higher than 89.12% can be achieved using this model.

						Accuracy		
Test #	Seconds in split	Entries	Mini-batch size	Max LR	Epochs	Train	Val	Test
1	1	2531	512	6,00E-06	200	0,9033	0,7360	0,7317
2	1	2531	256	5,00E-05	100	0,9791	0,7758	0,8000
3	1	2531	128	1,00E-06	50	0,6911	0,6588	0,6653
4	1	2531	64	2,00E-06	25	0,7555	0,6857	0,6815
5	5	499	256	5,00E-06	200	1,0000	0,6405	0,6412
6	5	499	128	5,00E-06	100	1,0000	0,6345	0,6252
7	5	499	64	1,00E-06	50	0,8594	0,5943	0,6432
8	5	499	32	8,00E-07	25	0,7751	0,6726	0,6352
9	10	245	128	3,00E-05	200	0,9728	0,6147	0,6653
10	10	245	64	5,00E-06	100	1,0000	0,6024	0,6653
11	10	245	32	1,00E-06	50	1,0000	0,5860	0,5795
12	10	245	16	4,00E-05	25	1,0000	0,6680	0,7020
13	15	162	128	2,00E-06	200	0,9987	0,5312	0,5925
14	15	162	64	7,00E-06	100	1,0000	0,5500	0,5185
15	15	162	32	1,00E-06	50	1,0000	0,5375	0,5740
16	15	162	16	2,00E-07	25	0,9951	0,5250	0,5987
17	20	118	64	8,00E-07	200	1,0000	0,6324	0,6101
18	20	118	32	4,00E-06	100	1,0000	0,5299	0,5423
19	20	118	16	7,00E-07	50	1,0000	0,5726	0,5762
20	20	118	8	3,00E-07	25	1,0000	0,6325	0,5932
21	25	94	64	2,00E-05	200	1,0000	0,6739	0,5638
22	25	94	32	1,00E-06	100	1,0000	0,5000	0,6063
23	25	94	16	1,00E-06	50	1,0000	0,5869	0,6382
24	25	94	8	1,00E-07	25	0,9677	0,6630	0,6489
25	30	77	64	1,00E-06	200	1,0000	0,5263	0,5714
26	30	77	32	8,00E-07	100	1,0000	0,4605	0,5714
27	30	77	16	8,00E-07	50	1,0000	0,5000	0,5195
28	30	77	8	7,00E-06	25	1,0000	0,6316	0,6753

Table 4.1: Results from fully connected neural network tests

The second experiment was really similar to the previous one, but the split length of one second was fixed and the number of epochs the model was trained for was increased. The results for this second experiment can be found on table 4.2 and the graph on figure 4.2. It's important to note that in this experiment, the number of loops the model was trained for was the same for all of the tests, while on the first experiment, the number of loops wasn't taken into account. This experiment will then show more accurate relations between its different tests. The learning rates were also calculated to be the best. The highest test accuracy in this case, 87.71%, was achieved in test #8 using a 64 mini-batch size. This didn't beat the 89.12% mark, so the one dimensional convolutional neural network will be explored.

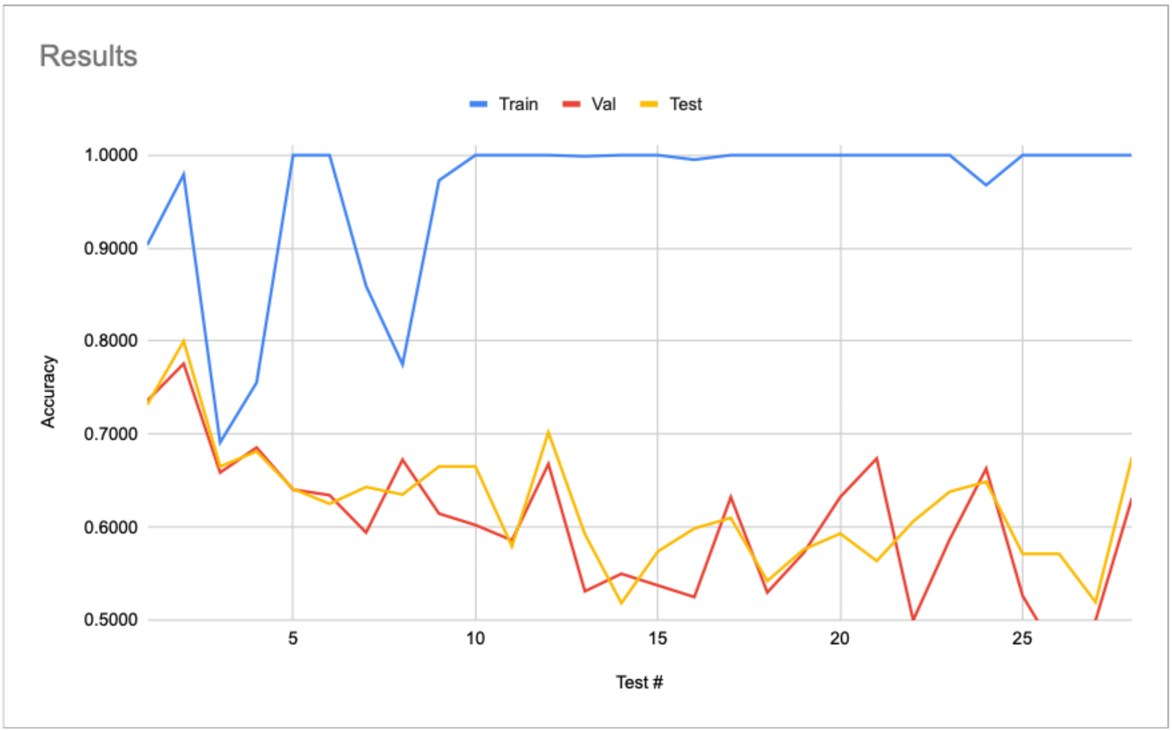

FIGURE 4.1: Graph with results from fully connected neural network tests

							Accuracy		
Test #	Seconds in split	Entries	Mini-batch size	Max LR	Epochs	Loops	Train	Val	Test
1	1	2531	512	5,00E-06	384	1898	0,9986	0,7411	0,7242
2	1	2531	512	5,00E-05	384	1898	1,0000	0,7668	0,7787
3	1	2531	256	1,00E-06	192	1898	0,8888	0,7051	0,6950
4	1	2531	256	1,00E-05	192	1898	1,0000	0,7478	0,7515
5	1	2531	128	5,00E-07	96	1898	0,8153	0,6953	0,6736
6	1	2531	128	5,00E-06	96	1898	0,9981	0,7411	0,7290
7	1	2531	64	9,00E-05	48	1898	1,0000	0,8241	0,8111
8	1	2531	64	9,00E-04	48	1898	0,9990	0,8917	0,8771
9	1	2531	32	5,00E-05	24	1898	1,0000	0,8241	0,8439
10	1	2531	32	5,00E-04	24	1898	0,9236	0,8700	0,8672
11	1	2531	16	2,00E-06	12	1898	0,9551	0,7308	0,7491
12	1	2531	16	2,00E-05	12	1898	0,9999	0,8217	0,8258
13	1	2531	8	6,00E-07	6	1898	0,8150	0,6933	0,6867
14	1	2531	8	6,00E-06	6	1898	0,9792	0,8111	0,7926

TABLE 4.2: Results from fully connected neural network tests with fixed one second split length

The third and last experiment was now using the one dimensional convolutional neural network described in Chapter 3. This experiment also sticked to the one second split length and now it took the 64 mini-batch size and compared it to a 128 mini-batch size. Table 4.3 shows the results of the tests in this experiment. Test #2 showed a higher accuracy than test #1, so more tests with a 64 mini-batch size were made. This tests had more epochs and different learning rates that were close to the best learning rate found. From test #7 on, the number of epochs was decreased because the last few epochs showed the same accuracies over and over again. Tests

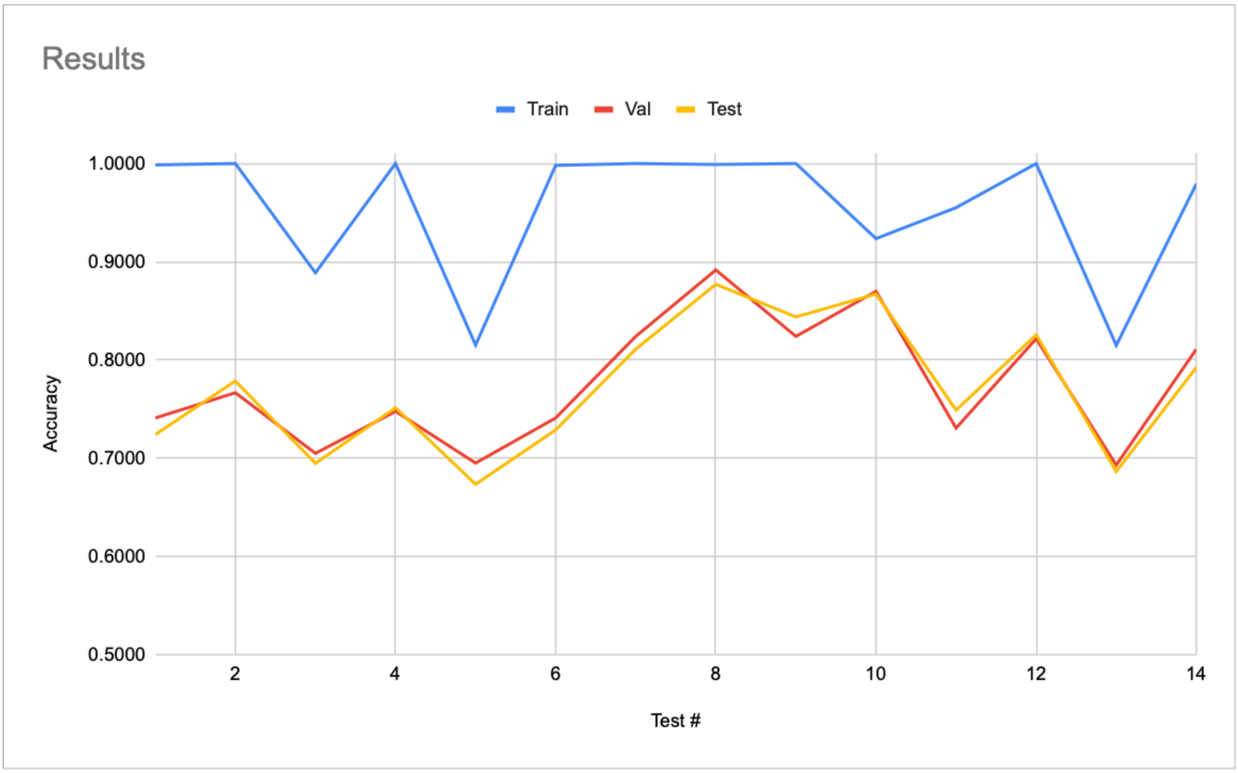

FIGURE 4.2: Graph with results from fully connected neural network tests with fixed one second split length

#9 and #10 were stopped after a few epochs because the printed accuracies showed big jumps, which means the model was diverging. Therefore, from all the tests made for this experiment the one with the highest test accuracy is test #8 with 99.21%. This is also the highest test accuracy from all the performed tests. The model that achieved this was saved on a .pth file that can be found here. Figure 4.3 shows a more visual representation of the mentioned results.

							Accuracy		
Test #	Seconds in split	Entries	Mini-batch size	Max LR	Epochs	Loops	Train	Val	Test
1	1	2531	128	1,00E-04	20	395	0,9880	0,9083	0,9040
2	1	2531	64	1,00E-03	10	395	0,9999	0,9692	0,9633
3	1	2531	64	1,00E-04	30	1186	1,0000	0,9356	0,9328
4	1	2531	64	5,00E-04	30	1186	1,0000	0,9727	0,9700
5	1	2531	64	7,50E-04	30	1186	1,0000	0,9775	0,9802
6	1	2531	64	1,00E-03	30	1186	1,0000	0,9842	0,9862
7	1	2531	64	1,50E-03	25	989	1,0000	0,9885	0,9901
8	1	2531	64	2,00E-03	25	989	1,0000	0,9921	0,9921
9	1	2531	64	2,50E-03	25	989	Crazy Jumps		
10	1	2531	64	5,00E-03	25	989	Crazy Jumps		

TABLE 4.3: Results from 1D convolutional neural network tests

The results exceeded expectations. The expected accuracy was 89.12% and it was beaten by more than 10%. However, the dataset was small, so further investigation is needed. For now,

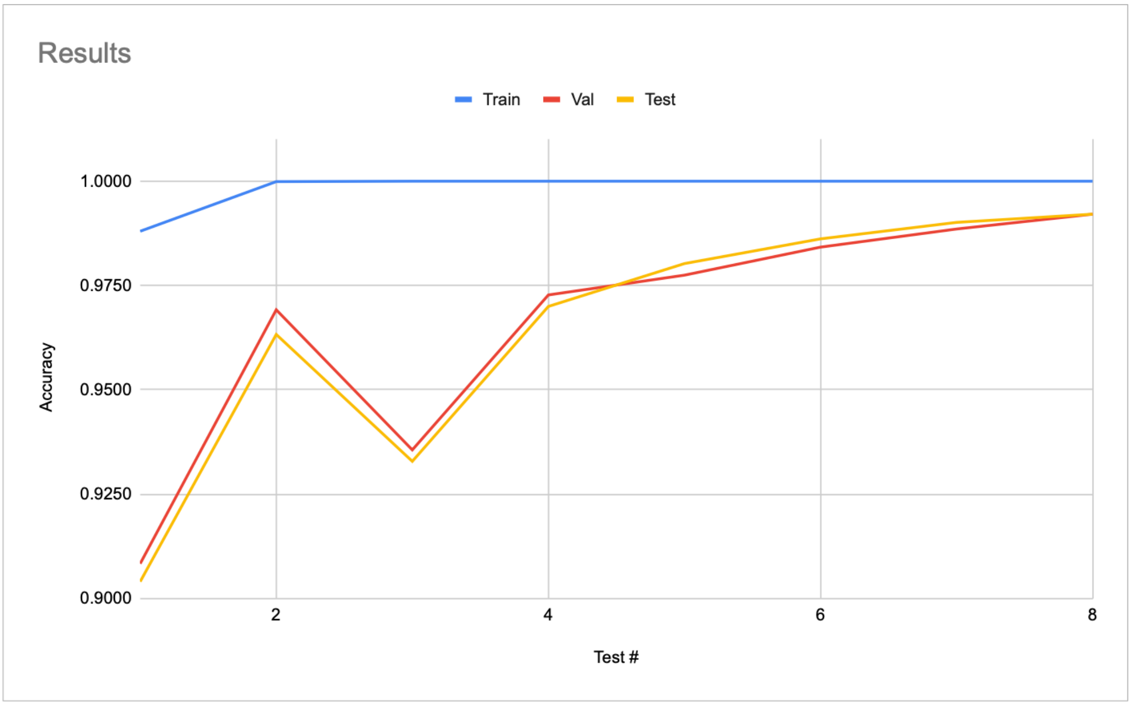

FIGURE 4.3: Graph with results from 1D convolutional neural network tests

the model does what it's supposed to do: classify EEGs differentiating kids with ADHD from those without it even with a higher accuracy than similar models doing this classification for adults [3] and similar accuracy than those predicting epileptic seizures [4]. This proves that one dimensional CNNs are capable of diagnosing ADHD in both adults and kids.

Chapter 5

Conclusion and Future Work

The most important takeaway from this investigation is that ADHD can objectively be diagnosed by entering EEG data into a convolutional neural network no matter the age of the patient. In this case, an accuracy of 99.21% was obtained using a 1D CNN and 1-second long EEGs when classifying kids with and without ADHD. This classification has also been proven to work for adults [3]. What's also cool about this model is that, besides the splitting, not much preprocessing of the data was done. Other models doing similar tasks [3] use spectrograms or some other kind of costly data preprocessing and get less accuracy.

As good as this sounds, future work is needed. First, a larger dataset should be found to prove the accuracy is correct. The used dataset [20] only consisted of 121 samples which were then converted into 2,531 by splitting each sample into shorter (1-second) samples. It would be great to test if EEGs without being splitted can get the same results, but a larger dataset is needed for that. The split either makes the results less reliable or is a breakthrough that indicates only a one-second EEG is enough to diagnose ADHD.

It would also be interesting to compare different neural network models such as a regular 2D CNN and a recurrent neural network. All this just to compare if better results can be achieved. Even just training the model with the characteristics described in Chapter 3 several more times to get an average of the accuracies would be really beneficial to corroborate the gotten accuracies aren't just a lucky strike.

Appendix A

Relevant Code

Initialize Model

```
import torch.nn as nn

in_channels = 19
channel1 = 76
channel2 = 152

cnnmodel = nn.Sequential(
        nn.Conv1d(in_channels=in_channels, out_channels=channel1, kernel_size=3,
            padding=1),
        nn.ReLU(),
        nn.Conv1d(in_channels=channel1, out_channels=channel2, kernel_size= 3,
            padding=1),
        nn.ReLU(),
        nn.MaxPool1d(2, 2),
        nn.Flatten(),
        nn.Linear(in_features=64*channel2, out_features=2)
    )
```

Train Model

```
import torch
import torch.nn.functional as F
from torch.utils.data import DataLoader

def train(model, optimizer, scheduler = None, epochs = 100):
    model = model.to(device = device)
    for epoch in range(epochs):
        train_correct_num  = 0
        train_total = 0
        train_cost_acum = 0
```

```
        for mb, (x, y) in enumerate(train_loader, start=1):
            model.train()
            x = x.to(device=device, dtype=torch.float)
            y = y.to(device=device, dtype=torch.long)
            scores = model(x)
            cost = F.cross_entropy(input=scores, target=y)
            optimizer.zero_grad()
            cost.backward()
            optimizer.step()
            if scheduler: scheduler.step()

            train_correct_num += (torch.argmax(scores, dim=1) == y).sum()
            train_total += scores.size(0)
            train_cost_acum += cost.item()
            train_acc = float(train_correct_num)/train_total

        val_cost, val_acc = accuracy(model, val_loader)
        train_cost = train_cost_acum/len(train_loader)
        train_acc = float(train_correct_num)/train_total
        print(
            f'Epoch:{epoch}, train cost: {train_cost:.6f}, val cost: {val_cost:.6f},'
            f' train acc: {train_acc:.4f}, val acc: {val_acc:4f}, total: {train_total},'
            f' lr: {optimizer.param_groups[0]["lr"]:.6f}'
        )
```

Initialize Optimizer and Scheduler

```
import torch

epochs = 25
max_lr = 2e-3

optimizer = torch.optim.Adam(cnnmodel.parameters(), 0.0001)

scheduler = torch.optim.lr_scheduler.OneCycleLR(
    optimizer,
    max_lr=max_lr,
    steps_per_epoch=len(train_loader),
    epochs = epochs, pct_start=0.43,
    div_factor=10,
    final_div_factor=1000,
    three_phase=True, verbose=False
)
```

Calculate Accuracy and Cost

```
import torch
import torch.nn.functional as F
```

```
from torch.utils.data import DataLoader

def accuracy(model, loader):
    num_correct = 0
    num_total = 0
    num_cost = 0
    model.eval()
    model = model.to(device=device)
    with torch.no_grad():
        for xi, yi in loader:
            xi = xi.to(device=device, dtype = torch.float)
            yi = yi.to(device=device, dtype = torch.long)
            scores = model(xi)
            num_cost += (F.cross_entropy(scores, yi)).item()
            _, pred = scores.max(dim=1)
            num_correct += (pred == yi).sum()
            num_total += pred.size(0)
        return num_cost/len(loader), float(num_correct)/num_total
```

Find Best Learning Rate

```
import torch
import torch.nn.functional as F
from torch.utils.data import DataLoader

def find_lr(model, optimiser, loader, start_val = 1e-6, end_val = 1, beta = 0.99):
    n = len(loader) - 1
    factor = (end_val / start_val)**(1/n)
    lr = start_val
    optimiser.param_groups[0]['lr'] = lr #this allows you to update the learning rate
    avg_loss, loss, acc = 0., 0., 0.
    lowest_loss = 0.
    batch_num = 0
    losses = []
    log_lrs = []
    accuracies = []
    model = model.to(device=device)
    for i, (x, y) in enumerate(loader, start=1):
        x = x.to(device = device, dtype = torch.float)
        y = y.to(device = device, dtype = torch.long)
        optimiser.zero_grad()
        scores = model(x)
        cost = F.cross_entropy(input=scores, target=y)
        loss = beta*loss + (1-beta)*cost.item()
        avg_loss = loss/(1 - beta**i)

        acc_ = ((torch.argmax(scores, dim=1) == y).sum()/scores.size(0))
        if i > 1 and avg_loss > 4 * lowest_loss:
```

```
            print(f'from here {i, cost.item()}')
            return log_lrs, losses, accuracies
        if avg_loss < lowest_loss or i == 1:
            lowest_loss = avg_loss

        accuracies.append(acc_.item())
        losses.append(avg_loss)
        log_lrs.append(lr)
        cost.backward()
        optimiser.step()
        print(f'cost:{cost.item():.4f}, lr: {lr:.4f}, acc: {acc_.item():.4f}')
        lr *= factor
        optimiser.param_groups[0]['lr'] = lr

    return log_lrs, losses, accuracies
```

Load and Split Dataset

```
import numpy as np
from google.colab import drive
from os import listdir
from os.path import isfile, join
from torch.utils.data import DataLoader, Dataset, random_split
from torch.utils.data import sampler

drive.mount('/gdrive')
%cd /gdrive/MyDrive/[Google Drive path to dataset]
adhd_paths = ['ADHD_part1', 'ADHD_part2']
control_paths = ['Control_part1', 'Control_part2']

def getEEGsFromPaths(paths):
  eegs = []

  for path in paths:
    files = [f for f in listdir(path) if isfile(join(path, f))]

    for file in files:
      eeg_mat = loadmat(path + '/' + file)
      size = len(file)
      file_no_ext = file[:size - 4]
      eeg = eeg_mat[file_no_ext]
      eegs.append(eeg)

  return eegs

adhd_eegs = getEEGsFromPaths(adhd_paths)
control_eegs = getEEGsFromPaths(control_paths)
```

```
CHAN = 19
FREQ = 128
seconds_in_split = 1
SAMPLES = FREQ * seconds_in_split

def split_list(l: list) -> np.array:
  splitedSize = SAMPLES
  wasted = len(l) % splitedSize
  return np.array([l[x:x+splitedSize] for x in range(0, len(l) - wasted, splitedSize)])

adhd_splited_arrays = np.concatenate([split_list(eeg) for eeg in adhd_eegs])
control_splited_arrays = np.concatenate([split_list(eeg) for eeg in control_eegs])

class CustomEEGDataset(Dataset):
    def __init__(self, adhd_data, control_data, transform=None, target_transform=None):
      adhd_y = [1] * len(adhd_data)
      control_y = [0] * len(control_data)

      y = np.concatenate((control_y, adhd_y), axis = 0)
      self.eeg_labels = y

      x = np.concatenate((control_splited_arrays, adhd_splited_arrays), axis = 0)
      x = np.transpose(x, (0, 2, 1))
      self.eegs = x

      self.transform = transform
      self.target_transform = target_transform

    def __len__(self):
        return len(self.eeg_labels)

    def __getitem__(self, idx):
        eeg = self.eegs[idx]
        label = self.eeg_labels[idx]
        if self.transform:
            eeg = self.transform(eeg)
        if self.target_transform:
            label = self.target_transform(label)
        return eeg, label

dataset =  CustomEEGDataset(adhd_splited_arrays, control_splited_arrays)

SIZE = len(dataset)
TRAIN_SIZE = int(SIZE * 0.7)
VAL_SIZE = int(SIZE * 0.15)
TEST_SIZE = SIZE - TRAIN_SIZE - VAL_SIZE
MINIBATCH_SIZE = 64
```

```
train_dataset, val_dataset, test_dataset = random_split(
    dataset,
    [TRAIN_SIZE, VAL_SIZE, TEST_SIZE]
)

train_loader = DataLoader(train_dataset, batch_size=MINIBATCH_SIZE, shuffle = True)
val_loader = DataLoader(val_dataset, batch_size=MINIBATCH_SIZE, shuffle = True)
test_loader = DataLoader(test_dataset, batch_size=MINIBATCH_SIZE, shuffle = True)
```

Bibliography

[1] Psychiatry.org - What is ADHD?

[2] CDC. What is ADHD?, January 2021.

[3] Laura Dubreuil-Vall, Giulio Ruffini, and Joan A. Camprodon. Deep Learning Convolutional Neural Networks Discriminate Adult ADHD From Healthy Individuals on the Basis of Event-Related Spectral EEG. *Frontiers in Neuroscience*, 14, 2020.

[4] Ihsan Ullah, Muhammad Hussain, Emad-ul-Haq Qazi, and Hatim Aboalsamh. An automated system for epilepsy detection using EEG brain signals based on deep learning approach. *Expert Systems with Applications*, 107:61–71, October 2018.

[5] 2022 Viguera Editores S.L.U. Alteraciones electroencefalográficas en niños con trastorno por déficit de atención con hiperactividad : Neurología.com.

[6] Marios Adamou, Tim Fullen, and Sarah L. Jones. EEG for Diagnosis of Adult ADHD: A Systematic Review With Narrative Analysis. *Frontiers in Psychiatry*, 11, 2020.

[7] Priya Miranda, Christopher D. Cox, Michael Alexander, Slav Danev, and Jonathan RT Lakey. In Quest of Pathognomonic/Endophenotypic Markers of Attention Deficit Hyperactivity Disorder (ADHD): Potential of EEG-Based Frequency Analysis and ERPs to Better Detect, Prevent and Manage ADHD. *MDER*, 13:115–137, May 2020. Publisher: Dove Press.

[8] Electroencefalograma: MedlinePlus enciclopedia médica.

[9] Ubicación de los nodos en el EEG.

[10] Ejemplo de EEG.

[11] Anna Kaiser, Pascal-M. Aggensteiner, Martin Holtmann, Andreas Fallgatter, Marcel Romanos, Karina Abenova, Barbara Alm, Katja Becker, Manfred Döpfner, Thomas Ethofer,

Christine M. Freitag, Julia Geissler, Johannes Hebebrand, Michael Huss, Thomas Jans, Lea Teresa Jendreizik, Johanna Ketter, Tanja Legenbauer, Alexandra Philipsen, Luise Poustka, Tobias Renner, Wolfgang Retz, Michael Rösler, Johannes Thome, Henrik Uebel-von Sandersleben, Elena von Wirth, Toivo Zinnow, Sarah Hohmann, Sabina Millenet, Nathalie E. Holz, Tobias Banaschewski, Daniel Brandeis, and on behalf of the ESCAlife-Consortium. EEG Data Quality: Determinants and Impact in a Multicenter Study of Children, Adolescents, and Adults with Attention-Deficit/Hyperactivity Disorder (ADHD). *Brain Sciences*, 11(2):214, February 2021. Number: 2 Publisher: Multidisciplinary Digital Publishing Institute.

[12] R. Catherine Joy, S. Thomas George, A. Albert Rajan, and M.S.P. Subathra. Detection of ADHD From EEG Signals Using Different Entropy Measures and ANN. *Clin EEG Neurosci*, 53(1):12–23, January 2022.

[13] Alexander Craik, Yongtian He, and Jose L. Contreras-Vidal. Deep learning for electroencephalogram (EEG) classification tasks: a review. *J. Neural Eng.*, 16(3):031001, April 2019. Publisher: IOP Publishing.

[14] Afshin Shoeibi, Marjane Khodatars, Navid Ghassemi, Mahboobeh Jafari, Parisa Moridian, Roohallah Alizadehsani, Maryam Panahiazar, Fahime Khozeimeh, Assef Zare, Hossein Hosseini-Nejad, Abbas Khosravi, Amir F. Atiya, Diba Aminshahidi, Sadiq Hussain, Modjtaba Rouhani, Saeid Nahavandi, and Udyavara Rajendra Acharya. Epileptic Seizures Detection Using Deep Learning Techniques: A Review. *International Journal of Environmental Research and Public Health*, 18(11):5780, January 2021. Number: 11 Publisher: Multidisciplinary Digital Publishing Institute.

[15] Ranjan Debnath, Natalie Viola Miller, Santiago Morales, Kaylee R. Seddio, and Nathan A. Fox. Investigating brain electrical activity and functional connectivity in adolescents with clinically elevated levels of ADHD symptoms in alpha frequency band. *Brain Research*, 1750:147142, January 2021.

[16] Pepe Cantoral, Ph.D. Red Neuronal Convolucional en PyTorch / Tutorial de PyTorch para CNN - Parte 1, 2022.

[17] Pepe Cantoral, Ph.D. Red Neuronal Convolucional en PyTorch con Batch Normalization/-Tutorial de PyTorch para CNNs -Parte 2, 2022.

[18] Pepe Cantoral, Ph.D. ¡Entrena redes neuronales mucho más rápido con el método de super-convergencia! Parte 1 - YouTube, 2022.

[19] Pepe Cantoral, Ph.D. ¡Cómo encontrar el Learning Rate Óptimo para el método de super convergencia! Parte 2 - YouTube, 2022.

[20] Ali Motie Nasrabadi. EEG data for ADHD / Control children. June 2020. Publisher: IEEE Type: dataset.