NTNU
Kunnskap for en bedre verden

EEG Controlled Robot Arm

Kenneth Sverre Verlo Jacobsen, Kenneth André Olsen, Christian Ovesen

May 20, 2019

PROJECT / BACHELOR THESIS

Faculty of Infomation Technology and Electrical Engineering

Department of ICT and Natural Sciences

Norwegian University of Science and Technology

Main supervisor: Robin T. Bye

Co-supervisor: Anete Vagale

Preface

This Bachelor thesis is written by three students from NTNU Aalesund as part of the course Automation Engineering. The thesis was carried out during the spring semester of 2019. The thesis will be particularly interesting to engineers, engineering students and those who have a passion for artificial intelligence, cybernetics or robotics.

There was a relaxed relationship towards division of labour between the authors, where division was divided naturally by the authors skills and previous experiences.

The field of artificial intelligence has been growing and it is continuously expanding in its use area. User friendly libraries such as Tensorflow and Keras makes this type of technology easy to implement, and no one knows how this technology will evolve into the future and how "smart" artificial intelligence could get. The authors saw potential in the multitude of ways this type of technology could be implemented. To the authors there seemed to be an inexhaustible amount of implementations. Combining artificial intelligence with other techniques and processes could increase efficiency and performances in many areas still not explored.

The authors were also particularly intrigued about this project because of its level of applicability to real life issues. Specifically the idea of this thesis being able to help people. This may be individuals who have lost an arm or a leg, who have lost mobility either temporarily or permanently or simply people who need extra strength to carry out their job.

Acknowledgement

We would like to thank our mentors Robin T. Bye and Anete Vagale for guidance and support. PhD student Thiago Gabriel Monteiro for comments and assistance with ideas, concepts and the Emotiv hardware and software communication. PhD student Elias Hasle for assistance and recommendations with regards to neural networks. Associate Professor Girts Strazdins for assistance establishing connection to Emotiv's servers. Norwegian University of Science and Technology (NTNU) for support trough hardware. OSC (Offshore Simulator Centre AS) for lending servers to train AIs and OSC service engineer Gard Huse Storebø for help with CUDA.

Summary

This report concerns the development of a system for controlling a robotic arm by the use of electroencephalography equipment, artificial intelligence and digital signal processing methods. The purpose of the project was to investigate in depth these methods, specifically artificial intelligence (AI) and digital signal processing methods in terms of interpretation of electroencephalography (EEG) signals.

Raw data, discrete fourier transform (DFT) and wavelet transform were compared as data processing methods. While fully connected NNs, one and two dimensional CNNs and long short term memory (LSTM) RNNs were compared as different NNs of interest in this regard.

The raw data didn't give sufficient results for controlling the arm. There was some improvement when the data sets size was increased. But the time needed to get a big enough data set to make this a viable option was too long for one person to gather in one session.

With DFT as the processing method the NNs had good predictive qualities, with prediction rates as high as 97%, this was achieved in tandem with a two dimensional CNN. These results were achieved with small data sets and small NNs. The networks were prone to overfitting, so keeping the size down is necessary for these results. The DFT added a lot of time to training, where what would take hours with raw data would take days with DFT.

Wavelet had similar problems to the DFT approach. The time needed to train NNs increased drastically compared with raw data, however the predictive qualities also increased with up to 23%, leaving the total accuracy at 43%.The time spent training was relatively close to DFT and an increase of accuracy only up to a total of 43%, was significantly lower than with DFT.

Control of a simulated robot arm was achieved using the best preforming NN with data of movement. There was also an attempt at controlling the arm trough strictly mental commands. This worked to a very limited degree.

Contents

Terminology

EEG Electroencephalography

AI Artificial Intelligence

ANN Artificial Neural Network

NN Neural Network

API Application Programming Interface

BCI Brain Computer Interface

CNN Convolutional Neural Network

CPU Central Processing Unit

DFT Discrete Fourier Transform

RAM Random Access Memory

RNN Recurrent Neural Network

LSTM Long Short-Term Memory, a variation of a RNN

FFT Fast Fourier Transform

GA Genetic Algorithm

IBM International Business Machines Corporation

GUI Graphical User Interface, makes it possible to interact with a computer

GPU Graphics Processing Unit

IDE Integrated Development Environment

API Application Programming Interface, activates functions from a remote software

NTNU Norwegian university of science and technology

URI	Uniform resource identifier, used to identify a specific resource
TCP	Transmission Control Protocol, connection oriented transmission protocol of information
UDP	User Datagram Protocol, non connection based transmission protocol of information
IP	Internet Protocol is a "best effort" delivery protocol
SDK	Software Development Kit
WSS	WebSocket secure, a URI scheme used for encrypted connections
SSL	Secure Sockets Layer, a security protocol used when communicating over a computer network

Notation

Hz	Unit of frequency
bit	Most basic unit of digital information

Abbreviations

IEEE	Institute of Electronical and Electronic Engineers
I2C	Inter Integrated Circuit
Gnd	Ground in electronical circuits
DOF	Degrees of Freedom, number of configurations for an object
SPS	Samples Per Second

List of Figures

List of Tables

Chapter 1

Introduction

This report will be reviewing the possibility of controlling a robot arm with the use of EEG equipment. And it will be looking at different methods for doing so, including different types of artificial intelligence and signal processing methods. The topic is considered an important topic because of the many possible use areas for such methods.

1.1 Background

There are many use cases for EEG signals controlling a plethora of equipment [27]. Wheelchairs [17], robot arms assisting paralyzed individuals, work loads in many different fields, text producing programs and much more. There are many research groups working on these problems, and EEG signals seems to be the optimal non-invasive way to get commands for these examples.

1.2 Problem Formulation

Would any form of artificial intelligence and signal processing methods be able to interpret different brain states by looking at EEG signals to such a degree that a robot arm could be voluntarily controlled?

Problems to be addressed

- What type of artificial intelligence would work best in combination with different signal processing methods.

1.3 Objectives

The Objectives for this report thesis are:

1. Develop a program for gathering raw EEG data and saving it in data sets for AI training.
2. Develop AI for interpretation of data from the EEG.
3. Make the program code as flexible as possible, allowing for other implementations with minimum change.
4. Implementing the AI for controlling the robot arm, as an example of a use case.
5. Demonstrating the results.

1.4 Limitations

- The initial plans have been limited to a certain degree, due to equipment limitations. The fidelity of the lower grade equipment isn't up to the standard initially planned for, but it was the only option within reasonable cost. The limitations are with regards to noise on the signal. Mental images are weak signals and the lower input impedance on the Emotiv equipment will make discerning the wanted degrees of freedom harder. The ANN created in this project could potentially grant more degrees of freedom, but the limitations on the software provided by Emotiv is four degrees of freedom. There is a possibility to use two of the degrees of freedom to change between the axis of movement on a robot arm, if this ends up being the limitation.
- The equipment and licence was not ordered in time, leading to the equipment arriving one month before the due date. However, the group found creative ways for preparing for

the arrival of the equipment. Emotiv trial licences were acquired, Emotiv headsets were borrowed and EEG data was downloaded from public EEG databases for analyzing and testing.

- The Emotiv Cortex API was newly released so there was no example code for acquiring EEG data from the headsets using Python. Thus, the group had to rely on the Cortex API documentation.
- Tensorflow was not supported by the hardware on all the group members personal computers. This was challenging because there was only one computer at disposal for the project. However, this limitation was minimized by sharing and teamwork.

1.5 Approach

The basic idea for this project was that artificial intelligence in combination with digital signal processing methods would to some degree be able to interpret EEG signals and distinguish individual commands from one another.

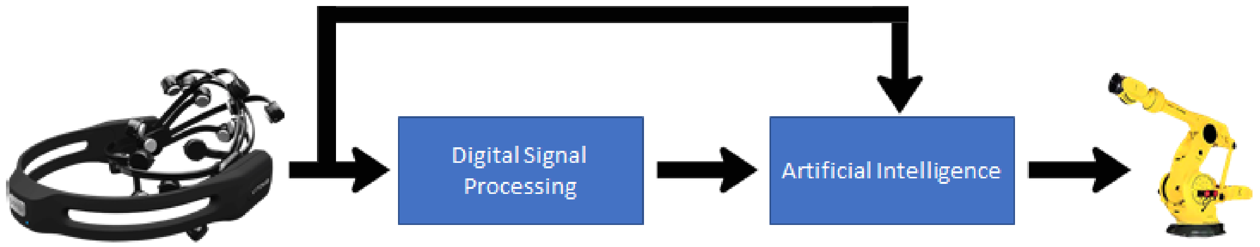


Figure 1.1: Basic System Flowchart

Artificial intelligence has been growing and by the use of Python and libraries such as Tensorflow and Keras, artificial intelligence could be implemented in a task of this manner. There were also powerful signal processing libraries available for use with Python which made it possible to operate both in the time domain and the frequency domain of a signal. Emotiv had developed EEG systems that were within reasonable cost for this type of research. They also had developed software for distinguishing between commands with their EEG systems, thus the group had something to compare against. Software such as RoboDK was useful for controlling robot arms and it was possible to control virtual robot arms directly from Python with RoboDK's libraries.

1.6 Structure of the Report

The rest of the report is structured as follows.

Chapter 2 - Theoretical basis: Chapter two gives an introduction to the theoretical background.

Chapter 3 - Method: Contains a description of the methodology and materials that were considered throughout the project.

Chapter 4 - Result: Contains a description of the finished product.

Chapter 5 - Discussion: A summary of.

Chapter 6 - Conclusions: This chapter presents an overall conclusion.

Chapter 2

Theoretical basis

2.1 GA - Genetic Algorithm

The authors have been using genetic algorithms for optimizing hyperparameters for the implemented AI 3.4.

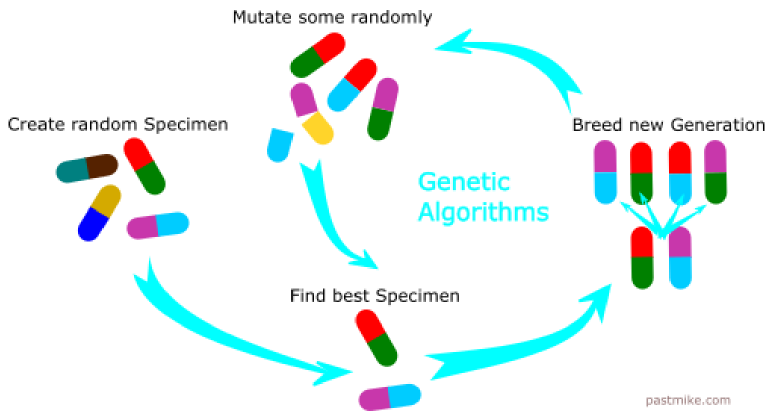


Figure 2.1: Genetic Algorithm

Genetic algorithms are generally used in science and engineering for solving practical problems, with a focus on solving optimization and search problems. Genetic algorithms usually use three types of rules: Selection, crossover and mutation. An initial population is made, with random parameters. This population is then individually trained on the data set, and the best performers are picked out while the rest is cut. These individuals are then used to generate new individuals based on the parameters in two of the better performers, and with a small chance of mutation (random change in one of the parameters). Once the population is back up to the defined population size, the process starts over again. This is repeated over a given number of generations 2.1. Genetic algorithms can by this description be thought of as a metaheuristic inspired by the process of natural selection [21].

2.2 Window function

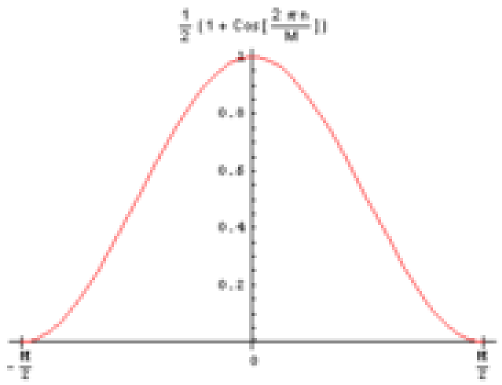


Figure 2.2: Hanning Window Function

For this project the window function's purpose was to minimize spectral leakage on the EEG signal after using the Fourier transform, minimizing unwanted frequency components in the frequency spectrum [15, 29].

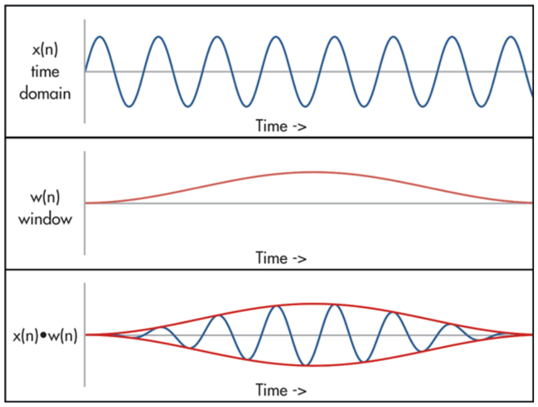


Figure 2.3: Windowed Sinus Signal

The window function is a mathematical function that is multiplied element-wise with the signal. When multiplied with the window function illustrated above, the signal gets tapered down at the ends (apodization function or tapering function[29])[15, 29].

2.3 Fourier Transform

The Fourier transform that is used for this project is the DFT. DFT gives advantages in combination with AI, specifically fully connected NNs and CNNs. When using the Fourier transform one has to compromise between how detailed the resultant frequency spectrum will be, and the delay time the system will have. This is because a time frame needs to be chosen which encapsulates the samples for the Fourier transform. A bigger time frame means more details in the frequency spectrum but it also means a higher delay time for the system, and vice versa.

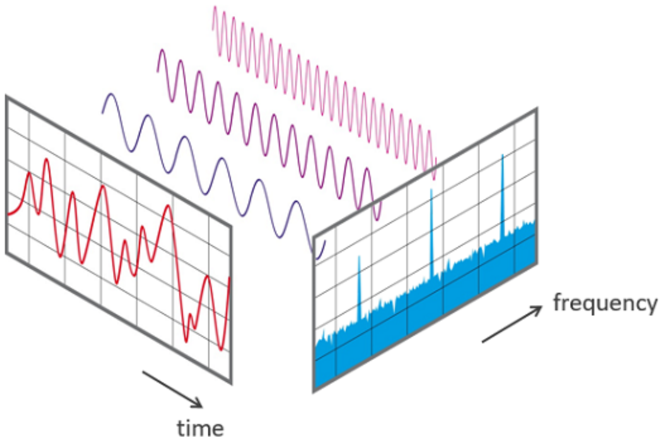


Figure 2.4: Fourier Transform

The Fourier transform transforms a signal from the time domain into the frequency domain. Doing an DFT creates aliasing problems making the upper half of the signal mirrored from the first part. The DFT signal has to be divided by two to remove the mirrored part [15].

$$\frac{\frac{Samples}{sec} * windowsize(sec)}{2} = ElementsInDFTVector \tag{2.1}$$

Calculates the number of elements in the resultant frequency spectrum vector.

$$NumberOfDFT = \frac{((ElementsInSignal) - (ElementsInDFT))}{(indent(SpacingBetweenDFTs))} \tag{2.2}$$

Calculates the number of DFTs that can be sequentially applied over a signal without exceeding the length of the signal. These formulas were useful for the authors while developing the computer algorithms.

2.3.1 Discrete Fourier Transform

The Fourier transform used by the authors is the discrete Fourier transform.

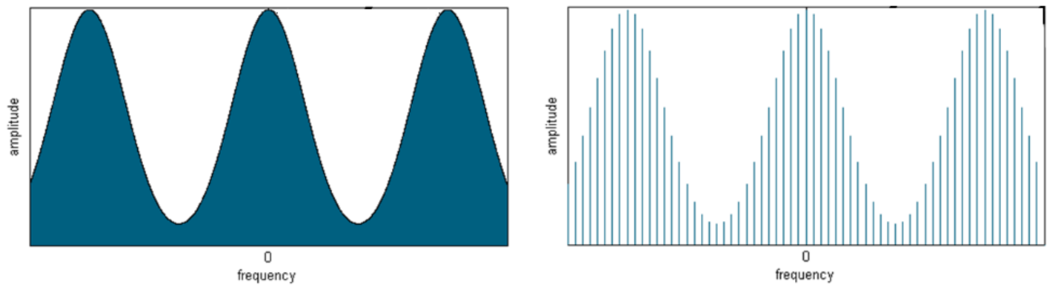


Figure 2.5: Continuous Fourier transform vs discrete Fourier transform

The discrete Fourier transform is the most used Fourier transform within digital signal processing. The discrete Fourier transform can be described as a sampled version of the continuous Fourier transform [15].

2.4 Wavelet transform

The wavelet transform uses variable window sizes depending on the frequency in question. This allows for finer time domain information in the high frequencies and high frequency domain information in the low frequencies [16, 23].

2.5 ANN - Artificial Neural Network

The authors specifically for this project are trying to train an ANN to recognize different brain states. These brain states are supposed to be voluntarily induced by the subject while wearing an EEG headset by doing different commands/movements, and then recording them in real time.

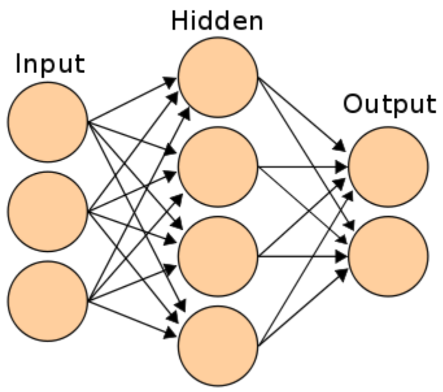


Figure 2.6: Artificial Neural Network

An ANN is loosely inspired by the biological neurons and their connections. There are several types of NNs that perform better at different tasks. These NNs use learning and training data sets to recognize patterns to predict a given array of outputs based on the inputs given [26].

2.6 Fully Connected Neural Network

In a fully connected NN each neuron in each layer is connected to every neuron in the previous layer and each connection has it's own weight [7]. With Python 2.15 a fully connected NN is refered to as "Dense".

2.7 CNN - Convolutional Neural Network

CNNs are specialized at recognizing patterns and were therefore chosen as a suitable choice for predicting what commands were given with the EEG signals provided.

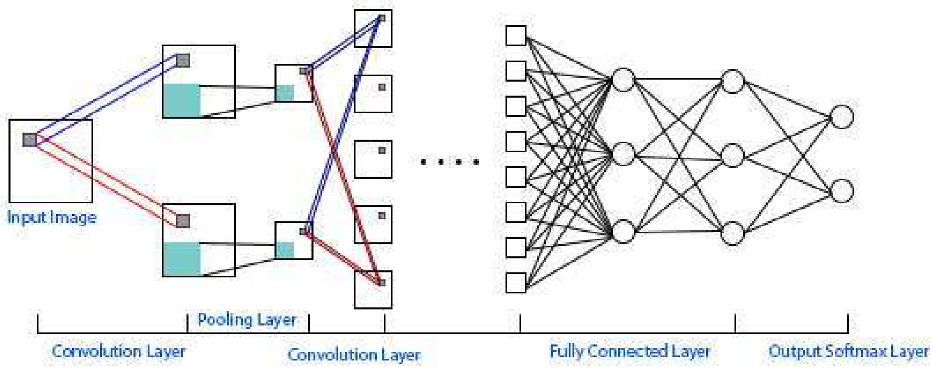


Figure 2.7: CNN

A CNN is a type of NN that is most commonly used for recognizing different visual imagery and patterns. It can resemble the functionality of visual cortex one, two and three. Compared to a regular ANN the CNN has one or more convolutional layers. These layers have filters that use optimization algorithms, allowing for recognition of patterns with some invariance to distortion and shift [20].

2.8 RNN - Recurrent Neural Network

The authors theorize that the RNNs ability to remember previous inputs might increase the accuracy of its predictions.

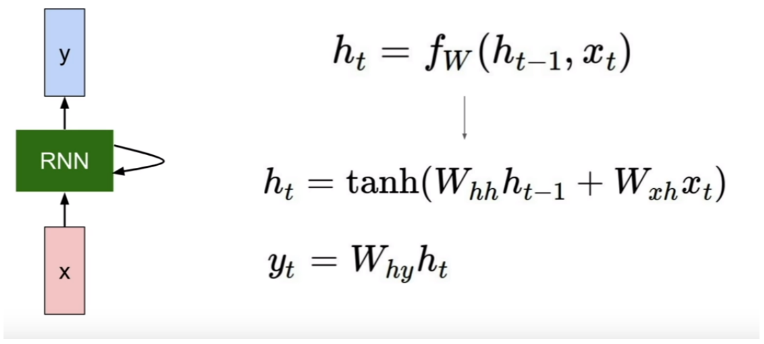

y
RNN
x
$h_t = f_W(h_{t-1}, x_t)$
$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$
$y_t = W_{hy}h_t$

Figure 2.8: Recurrent NN

A RNN can be said to be somewhat cyclic in manner. Previous time step activations are fed as inputs to the network allowing for previous states to impact predictions in the current time step. This allows a RNN to utilize a dynamically changing contextual window over the input sequence history. One problem with RNNs is the vanishing gradient problem, where the adjustments done with "error back propagation" become too small for weights further from the outputs in the NN setup. [24].

2.8.1 LSTM - Long short-term memory

As LSTMs are shown to do better than regular RNNs when it comes to both context free and context sensitive languages. The authors theorize that this will transfer well over to our use case.

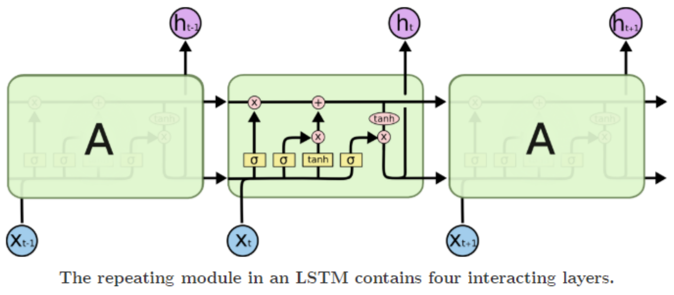

The repeating module in an LSTM contains four interacting layers.

Figure 2.9: Long short-term memory

An LSTM uses memory blocks to overcome some of RNNs weaknesses. The memory blocks store the temporal state of the network and uses gates to control the flow of information, dealing with the problem of long-term dependencies [24, 8].

2.9 Cross Validation

Cross validation is done by splitting a data set into equal pieces and assembling them in all possible ways as train data and test data. This is done to mitigate unknown bias in the data set from train to test, increasing accuracy of the results [19].

2.10 RoboDK

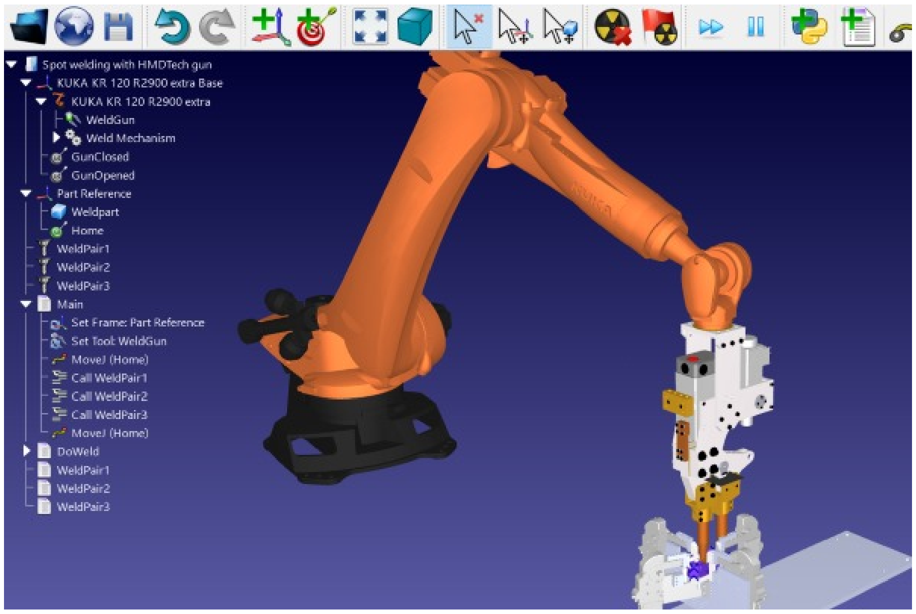


Figure 2.10: RoboDK

[f7]

RoboDK is a offline programming tool well suited for the project as it simulates robot arms. This allows for easy live testing of the prediction, without having to use a physical robot [6].

2.11 EmotivPRO

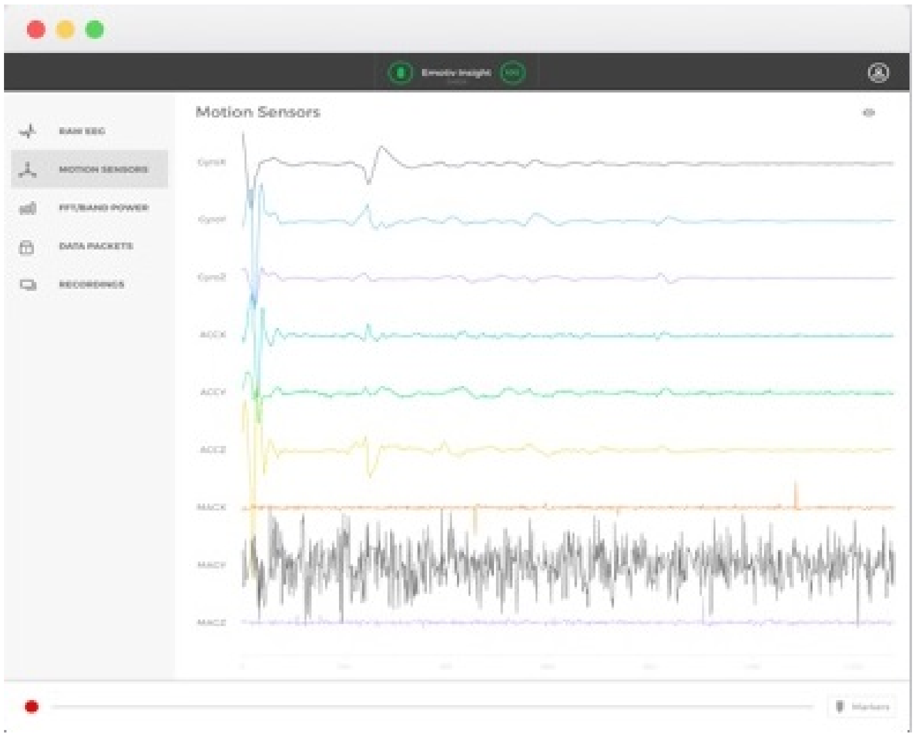


Figure 2.11: EmotivPRO

[f8]

EmotivPro is a software that has features such as recording EEG data, viewing and exporting raw EEG data, performance metrics, motion sensor data and it can display FT/band power data in real time. The authors used this software in the early stages for EEG data acquisition for testing purposes concerning the AI. The authors also used it in the early stages of the project in combination with their own written software because of its useful sensor contact quality GUI.

2.12 EmotivBCI

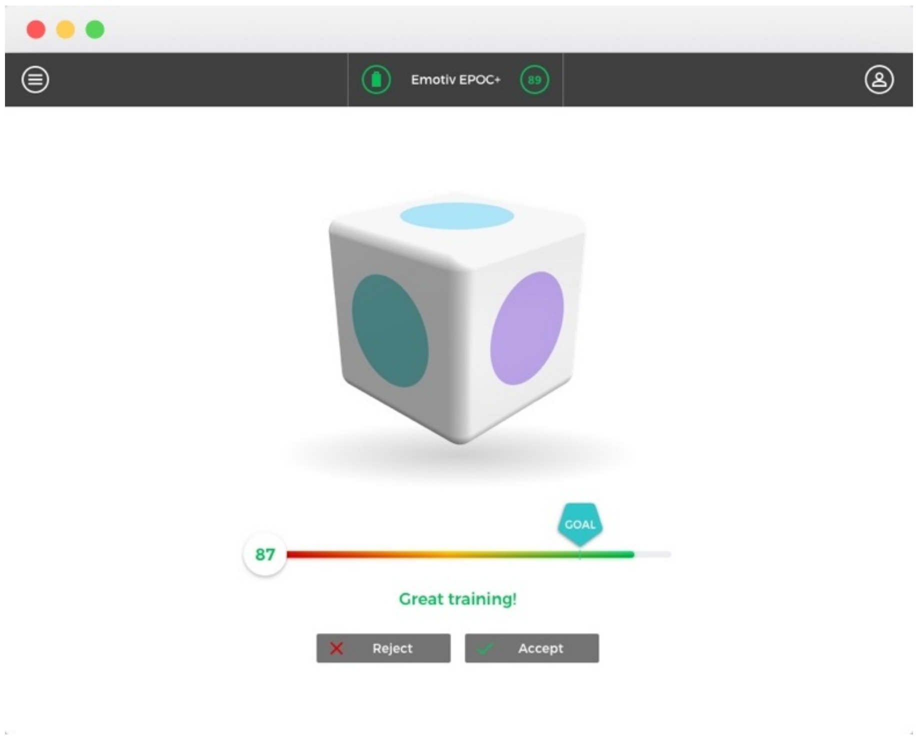
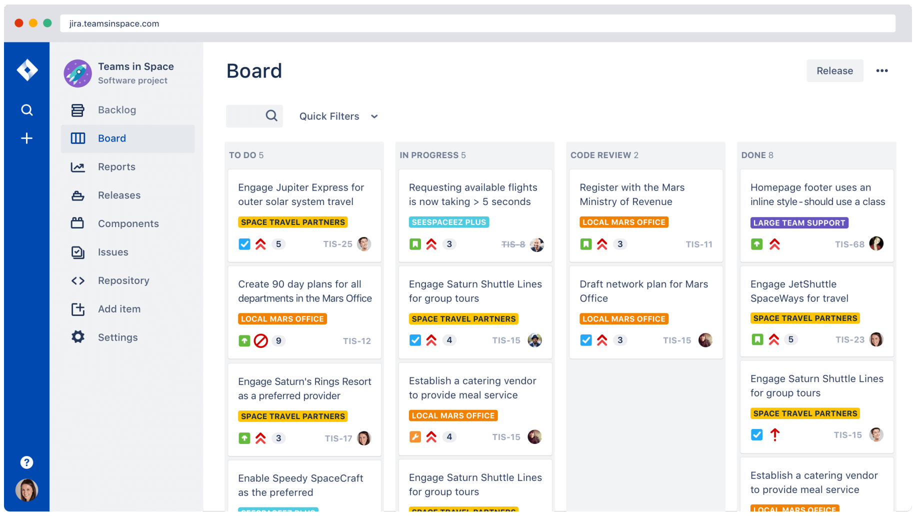


Figure 2.12: EmotivBCI

[f9]

EmotivBCI was used by the authors because it has features for training mental commands. The authors used this feature for guidance and comparisons with their own written software.

2.13 Jira



TO DO 5

- Engage Jupiter Express for outer solar system travel — SPACE TRAVEL PARTNERS — 5 — TIS-25
- Create 90 day plans for all departments in the Mars Office — LOCAL MARS OFFICE — 9 — TIS-12
- Engage Saturn's Rings Resort as a preferred provider — SPACE TRAVEL PARTNERS — 3 — TIS-17
- Enable Speedy SpaceCraft as the preferred

IN PROGRESS 5

- Requesting available flights is now taking > 5 seconds — SEESPACEEZ PLUS — 3 — ~~TIS-8~~
- Engage Saturn Shuttle Lines for group tours — SPACE TRAVEL PARTNERS — 4 — TIS-15
- Establish a catering vendor to provide meal service — LOCAL MARS OFFICE — 4 — TIS-15
- Engage Saturn Shuttle Lines for group tours

CODE REVIEW 2

- Register with the Mars Ministry of Revenue — LOCAL MARS OFFICE — 3 — TIS-11
- Draft network plan for Mars Office — LOCAL MARS OFFICE — 3 — TIS-15

DONE 8

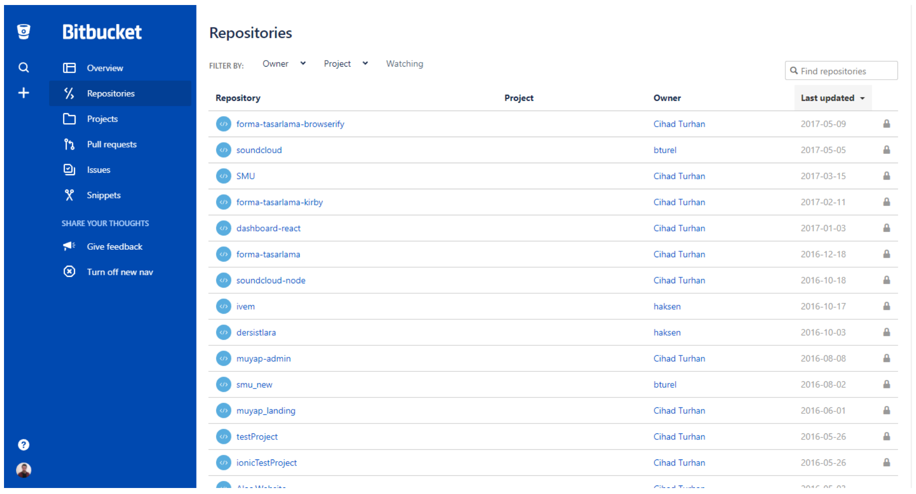
- Homepage footer uses an inline style - should use a class — LARGE TEAM SUPPORT — TIS-68
- Engage JetShuttle SpaceWays for travel — SPACE TRAVEL PARTNERS — 5 — TIS-23
- Engage Saturn Shuttle Lines for group tours — SPACE TRAVEL PARTNERS — TIS-15
- Establish a catering vendor to provide meal service — LOCAL MARS OFFICE

Figure 2.13: Jira

[f10]

Jira developed by Atlassian is used by the authors for organisation and issue tracking purposes. It is particularly useful because it allows the authors to collaborate more effectively and efficiently. Issues can be created with different priorities and assignees, which makes for a more organized atmosphere. The authors' mentors also has the possibility for a more close follow-up concerning the progress for the project.

2.14 Bitbucket



Repository	Project	Owner	Last updated
forma-tasarlama-browserify		Cihad Turhan	2017-05-09
soundcloud		bturel	2017-05-05
SMU		Cihad Turhan	2017-03-15
forma-tasarlama-kirby		Cihad Turhan	2017-02-11
dashboard-react		Cihad Turhan	2017-01-03
forma-tasarlama		Cihad Turhan	2016-12-18
soundcloud-node		Cihad Turhan	2016-10-18
ivem		haksen	2016-10-17
dersistlara		haksen	2016-10-03
muyap-admin		Cihad Turhan	2016-08-08
smu_new		bturel	2016-08-02
muyap_landing		Cihad Turhan	2016-06-01
testProject		Cihad Turhan	2016-05-26
ionicTestProject		Cihad Turhan	2016-05-26

Figure 2.14: Bitbucket

[f11]

Bitbucket owned by Atlassian was used by the authors because it makes for a more collaborative source code development environment. Bitbucket's usefulness came through the most when the authors were at separate locations while trying to collaborate writing source code.

2.15 Programming

2.15.1 Python

Figure 2.15: Python Logo

The authors used Python as the programming language of choice due to Tensorflow being a flexible open source library, well known for its ease of use. Multithreading and multiprocessing is easily implemented through libraries, allowing for parallelizing code.

2.15.2 Tensorflow

Figure 2.16: Tensorflow

[f14]

The Tensorflow library was one of the main reasons the authors chose Python as their programming language of choice for this project. This is because Keras which has powerful machine learning libraries runs on top of Tensorflow, this was a essential part of the systems that was developed for this project.

2.15.3 Keras

Figure 2.17: Keras

[f15]

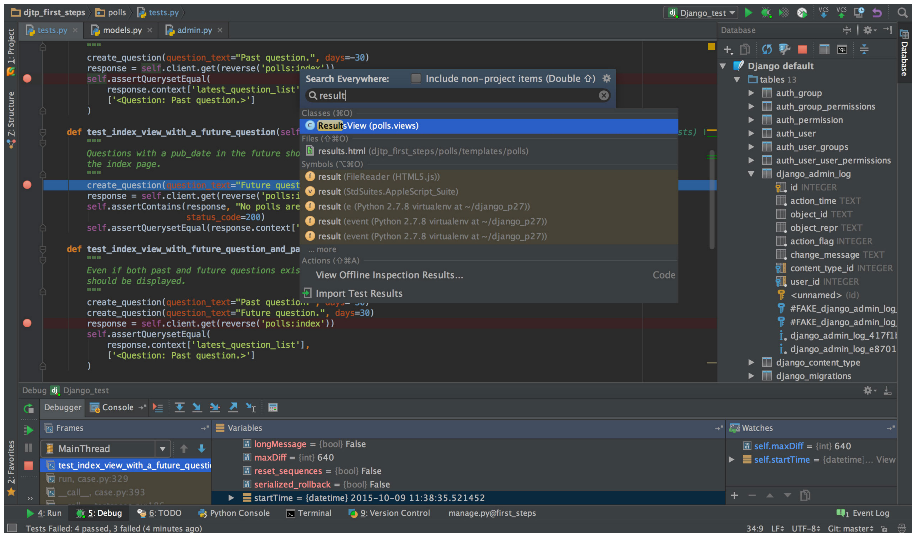
Keras was used because of its NN libraries. For the authors project it runs on top of Tensorflow, and it is designed for NN experimentation and it is also very user friendly.

2.15.4 CUDA

CUDA is collection of libraries that provide accelerated computing. It is used for faster training of ANNs 2.6 in this project [3].

2.15.5 IDE

An integrated development environment (IDE) is a software application that gives most commonly used tools to programmers for produceing code. Two different IDEs were used by the authors for this project. One had experience with PyCharm and another had experience with Eclipse. Both have their advantages and disadvantages, but the choice of IDE mostly comes down to preference.



```
        """
        create_question(question_text="Past question.", days=-30)
        response = self.client.get(reverse('polls:index'))
        self.assertQuerysetEqual(
            response.context['latest_question_list'
            ['<Question: Past question.>']
        )

    def test_index_view_with_a_future_question(self
        """
        Questions with a pub_date in the future sho
        the index page.
        """
        create_question(question_text="Future quest
        response = self.client.get(reverse('polls:i
        self.assertContains(response, "No polls are
                            status_code=200)
        self.assertQuerysetEqual(response.context['

    def test_index_view_with_future_question_and_pa
        """
        Even if both past and future questions exis
        should be displayed.
        """
        create_question(question_text="Past question.", days=-30)
        create_question(question_text="Future question.", days=30)
        response = self.client.get(reverse('polls:index'))
        self.assertQuerysetEqual(
            response.context['latest_question_list'],
            ['<Question: Past question.>']
        )
```

```
longMessage = {bool} False
maxDiff = {int} 640
reset_sequences = {bool} False
serialized_rollback = {bool} False
startTime = {datetime} 2015-10-09 11:38:35.521452
```

```
Tests Failed: 4 passed, 3 failed (4 minutes ago)
```

Figure 2.18: PyCharm

[f6]

PyCharm is developed by the Czech company JetBrains. It is a Python specific IDE, specialized for that task. PyCharm has many usefull features, a code style correction tool being a good example.

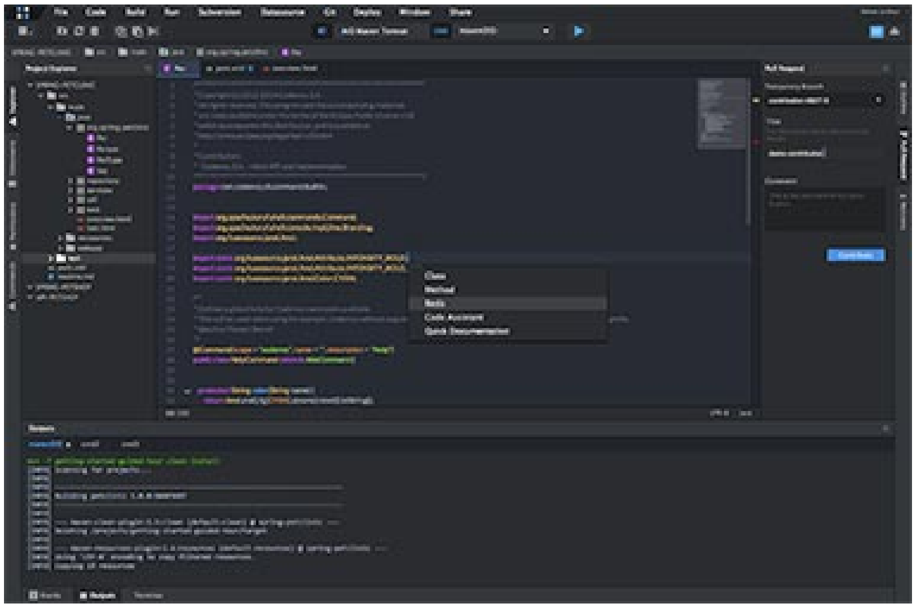


Figure 2.19: Eclipse

[f18]

Eclipse is an open source java IDE. It has a "marketplace" where user made add-ons can be downloaded for personalizing and adding/removing features for customizing to the users preference. One of these add-ons is PyDev, allowing Python programming in Eclipse.

2.15.6 TkInter

Figure 2.20: Tkinter

[f20]

TkInter is a GUI package for Python. It seemed to be the most commonly used GUI package and thus the authors found many useful programming examples that made the implementation relatively problem free.

2.15.7 Cortex

Figure 2.21: Cortex

[f17]

Emotiv's EEG systems store all their data on a web server when not using EmotivPro. Because of this Emotiv have created Emotiv Cortex as a means of communication with this web server. The Emotiv Cortex application was used by the authors for collecting real time data from the EEG systems that had been taken into use. The Cortex application makes it possible for an interface between a program written in Python and Emotiv's EEG systems, which was a essential part for the project.

2.15.8 Pickle

Pickle is a part of the Python Standard Library used to store and load data from files. It was used to store and load sets of prerecorded data.

2.15.9 SQLite

SQLite is an open source library that was utilized. This library builds on the workings of SQL databases while reducing the need for complicated commands. It stores the databases locally and shouldn't be too hard to scale up to servers if that is needed in the future.

2.15.10 Thread

A thread can be thought of as a small sequence of programmed instructions that can be managed independently by a scheduler. In the authors' case, threads are a part of a process [13].

2.15.11 Multithreading

Multithreading means to run several threads in parallel or seemingly so. In cases where there are only a single cored CPU, threads are not executed in parallel but rather share CPU resources by the use of a scheduler. Seemingly they run in parallel, but if the CPU has multiple cores, threads can run simultaneously as is the case for the authors with the hardware that is being used [18].

2.15.12 Thread Safety

In the case where multiple threads share memory resources, a problem can occur. This problem occurs when two or more threads attempt to use the same memory space resource at the same time. For example, one thread could be writing to the same memory space as another thread is reading from, or two threads could be writing to the same memory space at the same time. This leads to the possibility of data becoming corrupted. Thus, the authors had thread safety in mind when writing the program for this project. Because of that, mechanisms such as locks and queues were implemented for keeping a thread safe environment [5].

2.15.13 Lock

The authors used locks for hindering that threads access shared resources simultaneously. When one thread acquires a lock it gives the thread exclusive access to whatever resource it uses. Another thread trying to access the same resource at the same time will be put in a waiting state until the original thread that acquired the lock releases it. This makes it possible for sharing resources in a thread safe way, hindering that multiple threads read or write from the same resource at the same time [1].

2.15.14 Queue

Queues are implemented by the authors because of their usefulness when sharing resources, queues use techniques that makes them for a thread safe option [4].

Chapter 3

Materials and methods

3.1 Project Organisation

The group consists of 3 members participating in the study Automation Engineering. The group members met regularly with mentors as the project unfolded over the semester.

3.2 Data acquisition

The data collection was done by using the Cortex application from Emotiv. This application allowed the authors to stream raw EEG data from Emotiv's headsets. The Epoc+ from Emotiv3.1 was used for this project because it is relatively low priced compared to other EEG systems, and it's usability. It has 14 sensors, is wireless and uses a design that makes it easy and fast to set up compared to other head cap systems. It is also a plus that it fit's all the authors heads for this project, even though the authors have different shaped and sized heads.

Figure 3.1: Epoc+

[f12]

The data acquisition for this project was done with an Emotiv Epoc+ by the authors at the start of the project and at the end. The signals were sampled with the Epoc+ native frequency of 128Hz and it has a bandwidth of 0.16-43Hz, as the EEG headset has digital notch filters at 50Hz and 60Hz. The nodes on the Epoc+ are placed at (AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4) with (P3, P4) as reference nodes [2].

3.2.1 First sample gathering

The sampling was done in different ways as the project was not concluded and the finalized data gathering procedures were not finished due to time constraints. The data collection was done with a program written for the project then held on to the last 30 seconds of live EEG data, storing it. Eighteen commands were initially recorded with borrowed equipment and a 15-day trial version of Emotiv software. Six physical commands, six mental commands and six poses. These were made in a single session, consisting of a specific command with no interruption.

3.2.2 Second sample gathering

The second data gathering was done when the authors got the equipment at the end of the project. Seven commands were recorded in a single session, consisting of 30 seconds recorded as a single command with no interruptions. These commands were a combination of a physical motion and mental commands. They were primarily collected for testing the viability of the code and seeing if there were any problems with the setup of the NNs.

3.2.3 Third sample gathering

Towards the end of the project the authors wanted some larger sets of data for training purposes. Two new data sets where recorded. These new data sets were considerably larger than the ones previously used. Both sets consisted of five commands. The first set's commands were a combination of both physical motion and mental commands, and the second set's commands were mental commands. Each command was recorded in one session with three minutes recorded as training data, and 30 seconds recorded for testing.

3.2.4 Intended method

When discussing data sampling the authors intended to use a system were 30 seconds were recorded for each command entered into a GUI interface and 30 seconds of neutral data as a reference for no commands. These 30 seconds would be randomly cut into a random number of segments with a minimum of 3 seconds for each segment. They would then be displayed in the GUI and the subject would think of the command being displayed at that time. This was intended to make the training samples as close to live data as possible, where the subject might change commands at any time. This was discussed and agreed on with the mentors and consultants as a preferred way of generating the training data.

3.2.5 Conceivable issues

Emotiv headset's has a lower input impedance then other EEG headsets by a margin of 10^3. This could be problematic as the lower impedance increases the power needed for registering a reading from the sensors with the EEG headset. An EEG signal is already weaker than the noise produced gathering the data. A ground and reference node is used to reduce this noise trough signal processing. Increased power draw generates more noise on the signal. This could affect the quality of the signal [14].

Emotiv uses filters that cut of higher frequencies [2]. These frequencies could in theory increase the accuracy of the predictions made by the NNs.

3.3 Feature extraction

This project is on comparing results from different methods of feature extraction and classification. As for the feature extraction the authors concluded that the signal processing methods that would best suit this project was raw data input, DFT of the signal and wavelet transform of the signal. The one common data processing used for all models is data normalization. Data normalization is known for enhanced accuracy performance of ANNs of all types, and is therefore being used by the authors on all data [22, 25].

3.3.1 Raw data

Inputting raw data directly into the ANN model with normalization as the only data processing done, is a good baseline for the comparison with other signal processing methods. The ability of ANNs to recognize patterns is well established [11], and of great interest as to the performance this method would get within this project.

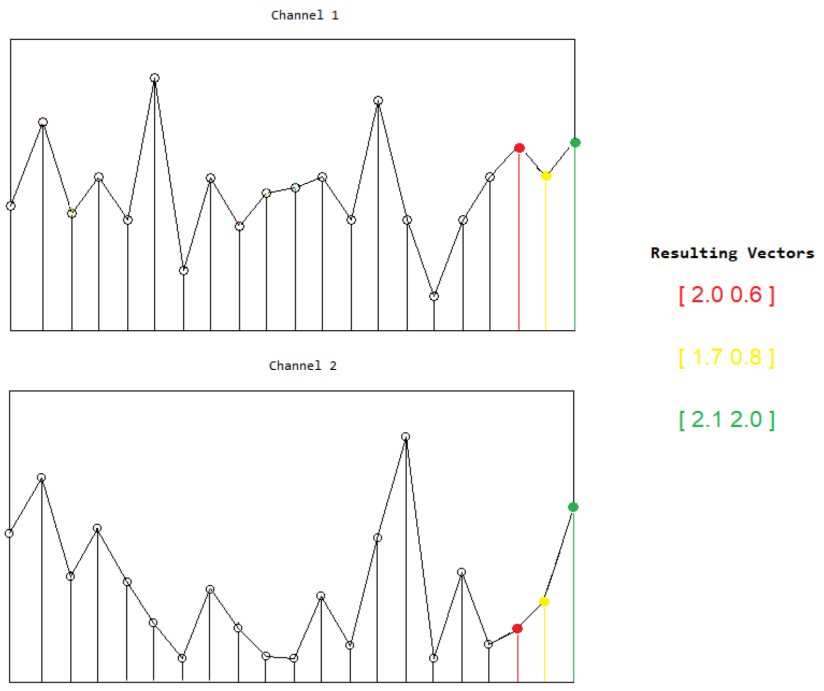


Figure 3.2: Raw Data Illustration

Raw data uses only the latest data from the channels 3.2. The main idea behind using raw data is to rely on RNNs. These networks has memory from past inputs thus the authors theorize there is no need for a time frame as with the DFT method. The figure attempts to show that only one sample from each channel is stored at a time. The resulting vector with the red square brackets is a representation of the resulting vector of the two samples that are marked with red dots in the illustration 3.2.

3.3.2 DFT

Using DFT in correlation with NNs have been shown to both increase accuracy and speed up the training process [28, 10]. These factors makes this a method worth exploring within the scope of this project.

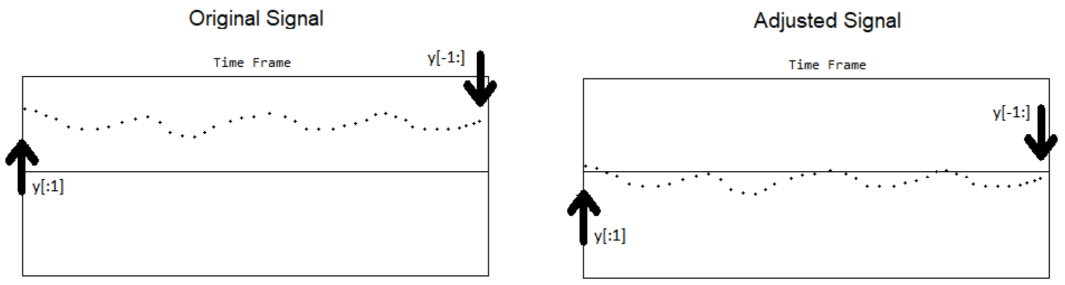


Figure 3.3: Signal Adjustment Illustration

When the EEG signals are received from the Epoc+ they lay with a high baseline of around 4100. This is a problem when windowing and using the Fourier transform, thus the signal needs to be brought down. This is done by the code below were $y[1:]$ is the first sample of the vector signal y and $y[-1:]$ is the last sample in the signal vector y, and this is done for all channel signal vectors.

```
y = y - ((y[:1] + y[-1:]) / 2)
```

Applying this formula to a signal gives the result that is illustrated in figure 3.3 above.

There were tested mainly 3 different signal adjustment methods.

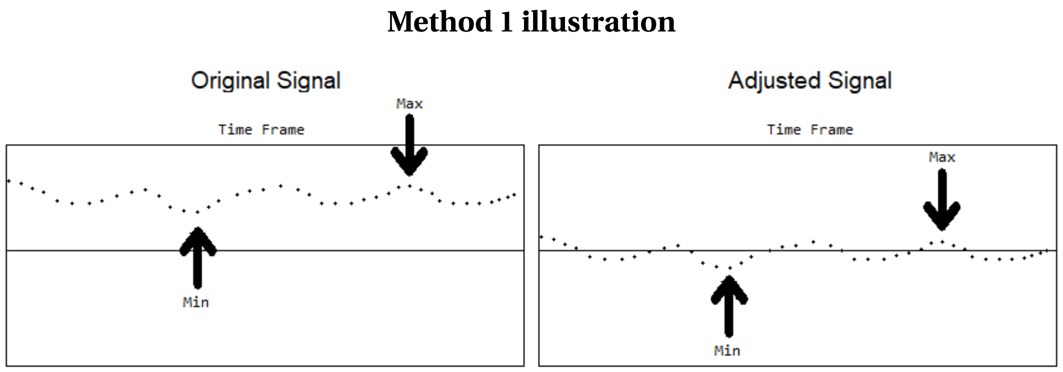


Figure 3.4: Method 1 Signal Normalization Illustration

The first method that the authors used found the minimum within the signal vector and subtracted that from all elements in the signal vector, then following that subtracted the maximum within the signal vector divided by two from the signal vector. This is illustrated in the figure 3.4 above. But by subtracting the mean value of Max and Min from the signal vector the baseline will be wrong in most cases, because the max or the min value will most likely be spikes.

Python code for Method 1:

```
import numpy as np
    y = np.subtract(y, min(y))
    y = np.subtract(y, max(y) / 2)
```

This method was the first signal adjustment method the authors tried, and it was used allot for testing purposes. The second method the authors tried subtracted the mean value from the signal vector. This got the authors a improvement of about 4% in correct command classification percentage with 4 classifiers. The problem with using a mean value is that it is sensitive to spikes, but compared to Method 1 it is an improvement.

Python code for Method 2:

```
import numpy as np
    y = y - np.mean(y)
```

The third method gave the authors an improvement of about 1% compared to the second method. This means about 5% improvement from the first method the authors tested, Method 1. The authors hypothesis is that this was the best method because it gives spikes in the signal least significance. This method could occasionally react to spikes and give wrong baseline but this is unlikely. With a 96 sample signal with one spike the probability of a spike being the first or last sample of the signal vector which would be a problem is 1/48.

Method 3 illustration

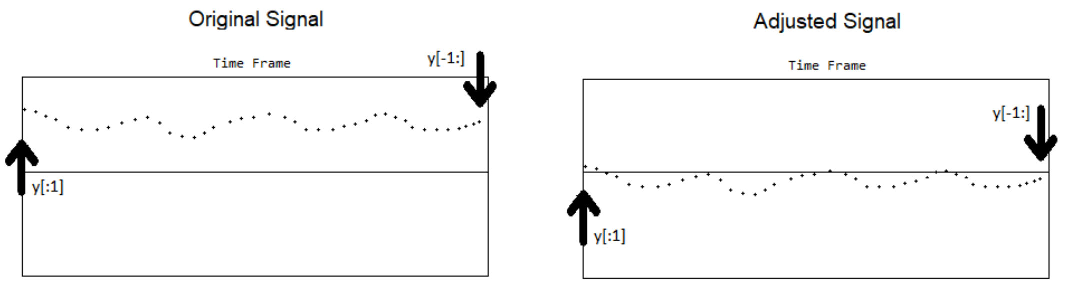


Figure 3.5: Method 3 Signal Normalization Illustration

Python code for Method 3:

```
y = y - ((y[:1] + y[-1:]) / 2)
```

Methods comparison

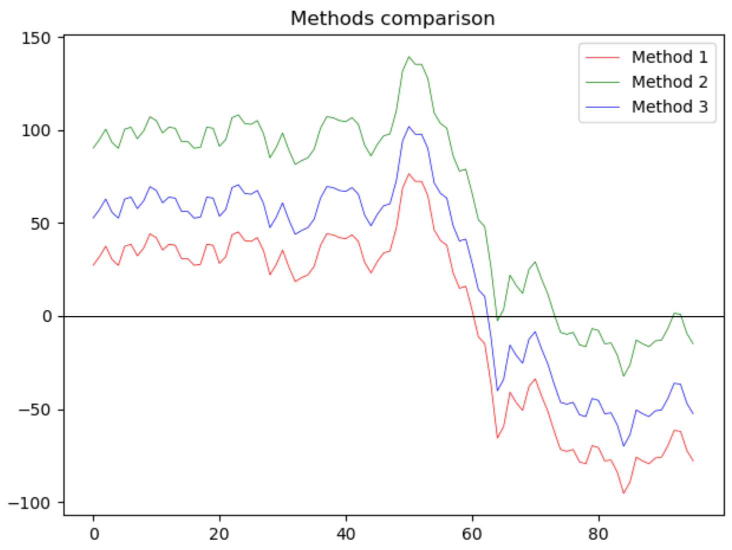


Figure 3.6: PyCharm Figure Methods Comparison

Figure 3.6 shows a comparison of the same 96 sampled signal with the 3 different signal adjustment methods applied.

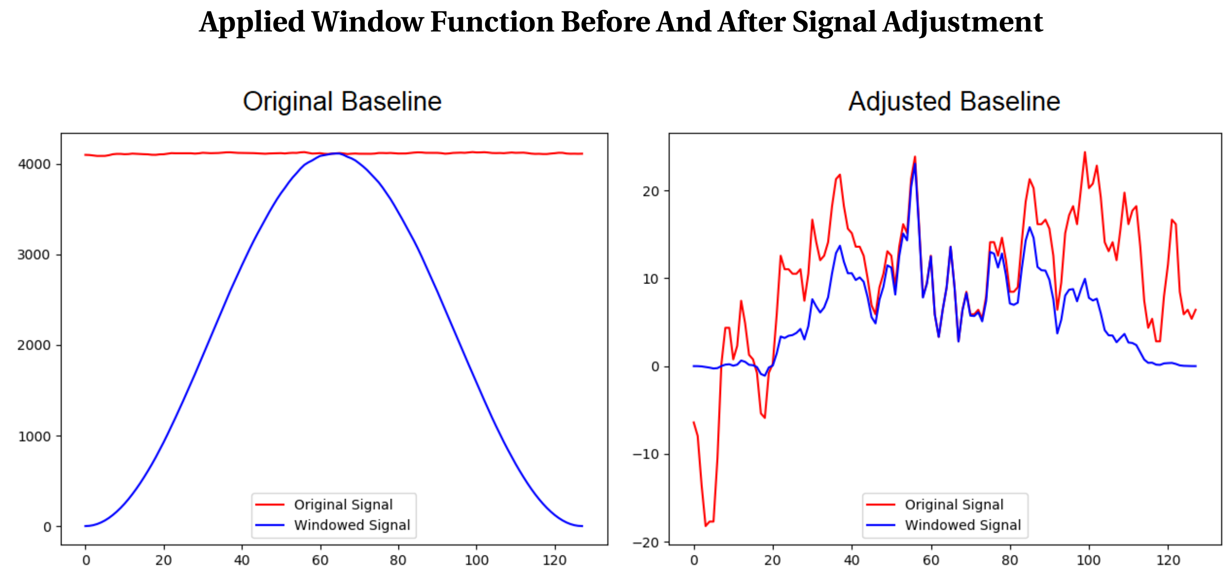


Figure 3.7: Windowed Signal Before And After Adjustment

The figure 3.7 above shows the difference between applying the hanning window function before and after the signal adjustment method. The plot to the left shows that the signal becomes totally distorted to the point that it takes the form of the hanning window function itself. This happens because of the high baseline. The plot to the right looks like it should, the signal gets tapered down at the ends minimizing spectral leakage when applying the Fourier transform.

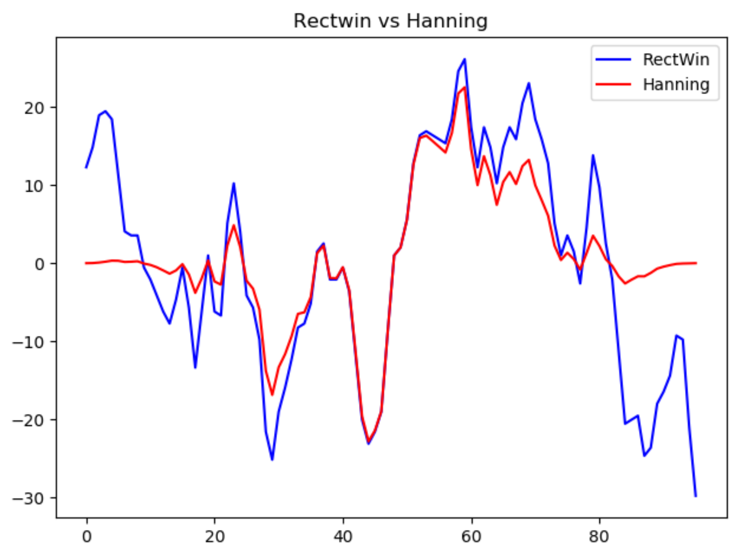


Figure 3.8: PyCharm Figure Windowed Signal

Figure 3.8 above shows a 96 sample section of a EEG signal recorded with the Epoc+, the red graph is the result from the blue graph being element-wise multiplied with the Hanning window function. By doing this before the Fourier transform the error classification percentage went down with about 10% when used in combination with a CNN, compared to when not using a window function before the Fourier transform.

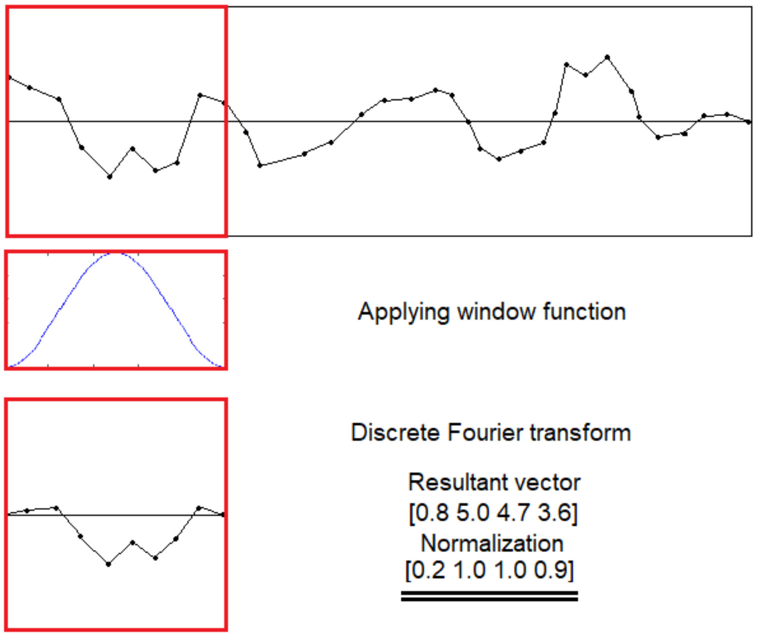


Figure 3.9: Simplified Windowing And Fourier Transform Illustration

When applying the DFT to a signal the type of window function and size of the window used is an important choice for accuracy 2.2. After extensive testing it became evident that the Hanning window function gave the best results. The red frame encloses the samples meant for windowing and then the following discrete Fourier transform is applied to the windowed samples. The resultant vector contains the frequency components amplitudes after applying the discrete Fourier transform for the windowed samples 3.9.

When applying the DFT 2.3.1 on a signal the time frame needs to be taken into account. After some testing a time frame of one and a half seconds was settled on as a good compromise between reaction time of the system and prediction accuracy. Applying a DFT also takes a lot of time, so a balance between compute time and resolution of the signal has to be decided on. Extensive testing showed that by applying a DFT of the last one and a half seconds every five samples gave the best result with accuracy and speed in mind. This makes a lot of overlap in the DFT samples 3.10 but ensures minimal information loss and response time.

After the DFT is applied the resultant vectors needs to be normalized. This was done by using a cap with the value 4. This means that anything with the value above 4 after the Fourier transform would be set to 4. After this step all values were divided by 4. By trial and error this showed to be a robust method with a lot of flexibility and good margins. There were also implemented adjustment techniques on the resultant vector for giving the weaker more high frequency components more amplitude in relation to the lower more stronger frequency components. This was done by raising the frequency components to powers below 1, this because it was discovered by trial and error that by doing so the NNs performed better. The method also allows for data integrity keeping the important information at the same scale, meaning the noisy spikes which do not carry the signal information are cut off by a constant threshold value and the important information remains relatively scaled.

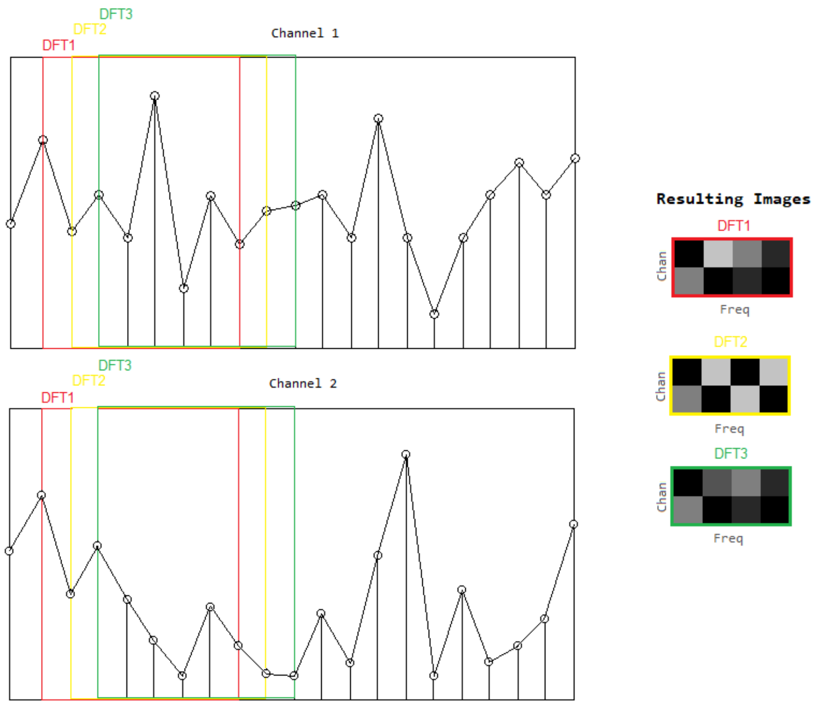


Figure 3.10: Simplified Illustration DFT For a Signal From Two Nodes

The resulting image from a DFT is a two dimensional frequency representation. Consisting of the number of nodes along one axis, and the number of samples in the DFT(Equation 2.1) along the other axis. This is represented with an array where the numbers in the rows and columns represent the strength of each frequency over the 14 nodes. The signal data is at this point ready to be put into the CNN, specifically "Conv2D".

The DFT has to be reshaped to a flat vector for use with other types of ANNs, allowing for the use of DFT with "Conv1D", "Dense" and "LSTM".

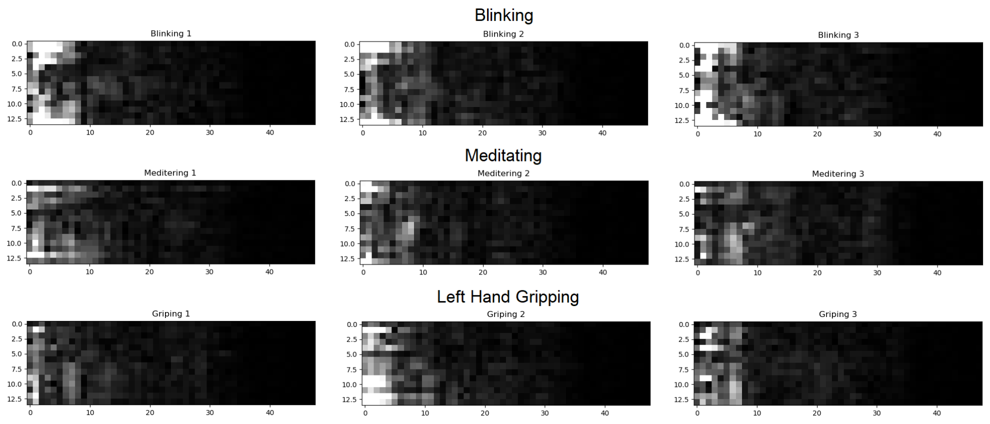


Figure 3.11: PyCharm figure Fourier Transform Comparison

Figure 3.11 above shows a comparison between nine resulting pictures after a 96 sample Fourier transformation of a 14 channel EEG signal recorded with the Epoc+ with one of the authors as the subject. The y-axis represents the channels and the x-axis represents the frequency, and the gray scale pixels ranging from black to white represents the amplitude of the frequency components of the signal in the z-axis, were white correlates with high amplitude and black correlates with low amplitude. In the 3 pictures in the first row the subject is repeatedly blinking, in the 3 pictures in the second row the subject is meditating and in the 3 pictures in the third row the subject is repeatedly gripping with his left hand. All the signals for the individual pictures are from different parts of the signal, so there is no overlap between them. It is quite clear from the pictures that there is a difference between the rows. This indicates that the Fourier transform method is actually picking up on differences between the states. And if a human can visually differentiate between them, AI should also be able to differentiate between them.

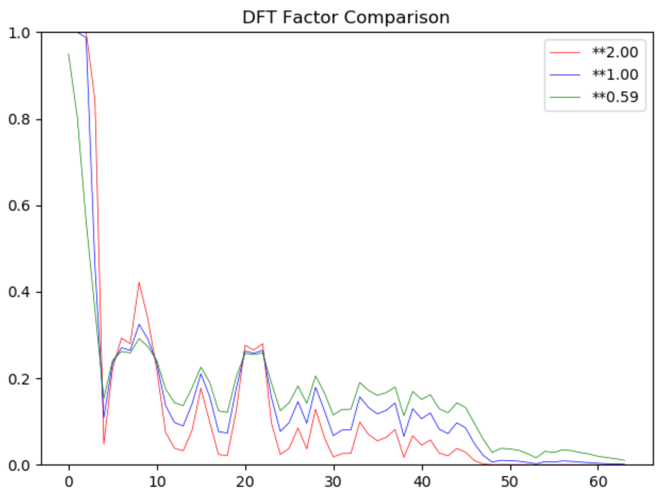


Figure 3.12: PyCharm figure DFT Factor Comparison

The DFT was applied with different powers. Figure 3.12 shows the same 128 sample single channel DFT with 3 different powers, by trial and error the authors settled on a power of 0.59 because this gave the best result. The error classification percentage went down with about 2% compared to when it was applied with the power of 1. At first the authors tried the power of 2 but this resulted in a poorer error classification percentage performance. Python code:

```
import numpy as np
    y = y*np.hanning(128)
    Y = abs(np.fft.fft(y)/n)**0.59
    Y = Y[range(n//2)]
```

The authors hypothesis is that this result comes from increasing the magnitude of the low powered signals relative to the more high powered signals, giving them more importance in the context of AI signal interpretation.

3.3.3 Wavelet transform

Wavelet transform performed on a EEG signal before it is fed into a NN has been shown to increase accuracy in specific cases similar to the one in this project. These factors makes it a fitting method, as it has been shown to help with classification after being applied to an EEG signal 2.4. There are many choices as to what wavelet transform to use. The authors settled on "sym9", as this has been shown to give good results for the type of task done in this project. "coif3" and "db7" were also considered based on the same parameters, but not tested [9].

3.4 Classification

The classification defines which mental command is being given based on the signal from the Emotiv headset. These variables can differ depending on the mental commands assigned to each data set. Movement readings were used while testing the NNs, as movement done by the subject gives a greater response compared to mental commands.

ANNs will be used for the classifications were the accuracy, speed of training and response speed is taken into account as factors for performance. Different types of ANNs will then be compared on that basis, giving a general idea about what one could expect decent results from. The types tested for this project is fully connected NNs, CNNs and LSTMs.

Once the authors had an Epoc+ headset and a trial licence Emotiv's own prediction software was tested. Since Emotiv does not give out validation results and training times the authors had no statistics on Emotiv's software. A statistical comparison between the author's software and Emotiv's software was because of this not possible. However when using the software the author's noticed it was almost impossible to control their software using only mental commands, and that movements had to be used in order to get reliable output from Emotiv's software.

3.4.1 Fully connected NN

In this project the fully connected NN is used as a good base line for comparison. The authors preferred relu as the default activation function and softmax as the activation function for the last layer. This is done for retrieving a percentage read of the confidence the ANN has in the given output being the correct one. Than the node with the highest output reading is chosen as the predicted command.

3.4.2 CNN

A CNN is theorized to be valuable as a predictor considering noise or some physical shift of the headset might change the inputs but not the patterns in the inputs 2.7. Both one dimensional and two dimensional convolutional networks will be tested and compared with both processed and raw data.

3.4.3 LSTM

The LSTMs use of memory blocks allows for predictions taking previous values into account without the vanishing gradient problem prevalent in other RNN solutions 2.8. This could allow for detecting patterns in the time domain, making it an interesting variable for testing purposes in this project [12].

3.4.4 GA

Using GA for optimizing hyperparamaters and testing a wide array of possible networks with different input data is essential in this project, as the possibilities are huge and testing them manually within the time limit for the project would not be possible 2.1.

3.4.5 K-Cross Validation

K-Cross Validation was used to ensure the accuracy of the results. This was done by cutting the data into 5 pieces for validating the results and increasing accuracy of results 2.9.

3.5 Implementation

This project implements the predictions made with a robot arm. Giving motor commands based on the predictions made. At the outset it was discovered that the university had a licence for the program RoboDK. It was then decided to use this for testing as it didn't require the use of a physical robot arm. RoboDK also has en extensive Python 2.15 library, allowing for easy implementation with the code for the project 2.10.

3.6 Approach

The framework for the project was laid out at the outset. This is a project that should be thought of as one in a line of projects. It should be done in a fashion allowing for future projects building on this one. So there has been a lot of focus on making every part as understandable and dynamic as possible with every aspect of the project.

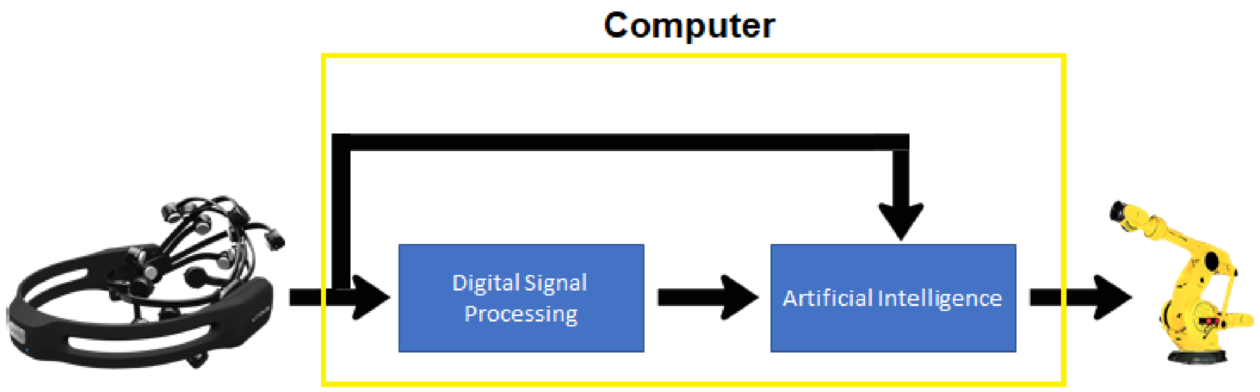


Figure 3.13: Basic System Flowchart

The outlines of the project can be described in four steps. Raw data from the EEG headset, signal processing, done with some of the training, running the data trough ANNs and using the classification to control a robot arm.

3.6.1 Hardware

The selection of EEG landed on the affordable option of Emotiv. Two EEG headsets were purchased, one Epoc+ 3.2 and one Epoc Flex 3.14. The Flex ended up not being used due to time constraints. Using Emotiv equipment to get raw EEG data requires a licence for Emotiv Pro 2.11, this licence gets locked to hardware. A new laptop was then purchased given the restriction with the licence. The laptop chosen was an Alienware m15 as it has a GTX 1060 GPU, allowing for the use of CUDA 2.15.4 to speed up AI training.

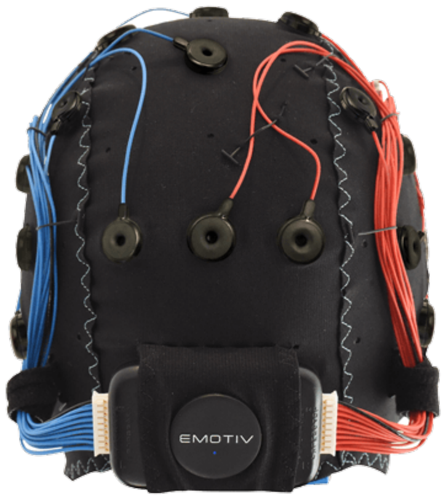


Figure 3.14: Epoc Flex

[f13]

3.6.2 Software

The authors settled on Python 2.15 as the exclusive programming language for this project. Sticking with one programming language makes for better cohesion and organization in the code. Making it easier to pick up by others, and continue working with it.

The initial code structure was intended to have three parts. The first would be Cortex Client handling data collection. The primary function of this part is to get the data from the EEG headset, then processing this data, preparing it for the AI.

Second is the GUI, handling everything related to user interaction, controlling what is going to be run and dealing with database saving and loading. TkInter 2.15.6 was used for the GUI package, as it is the most commonly used one for Python. For saving and loading data the SQLite 2.15.9 library was initially intended to be used and is implemented as its own file. As the project came to its end the parts tieing these pieces together wasn't finished and these parts are now laying idle but ready for implementation by future projects.

The last part being AI and GA. Were the GA optimizes for the hyperparameters regarding the ANNs and the best type of ANN for all the different data sources.

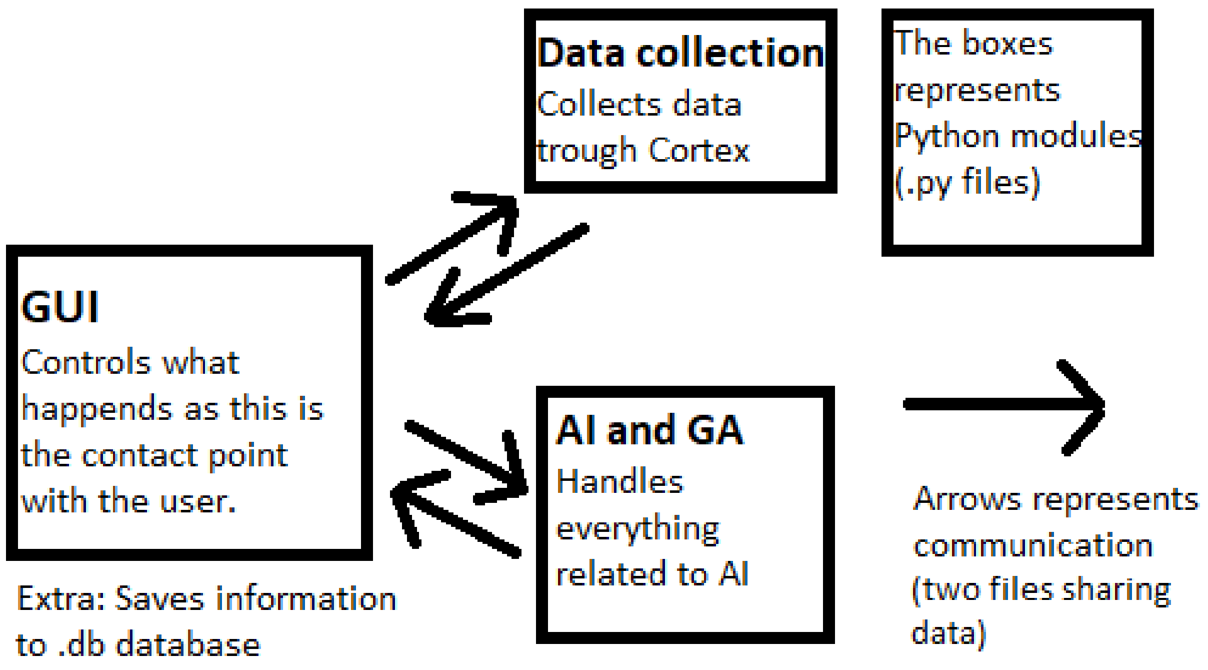

Data collection
Collects data trough Cortex
The boxes represents Python modules (.py files)
GUI
Controls what happends as this is the contact point with the user.
Extra: Saves information to .db database
AI and GA
Handles everything related to AI
Arrows represents communication (two files sharing data)

Figure 3.15: Initial Code Structure

When the different parts were coming together a realization was had regarding the structure of the code. The parts weren't coming together as intended, so it was decided to restructure the code. The GA and ANN was split into two files, and put in a sub folder. Cortex Client was moved to its own folder, allowing for the addition of other files if other types of EEG equipment would be used by later projects.

The original GUI file was split into three pieces: GUI, logic, and dbAPI. The plan for the three files being that the GUI file would handle communication with the user, passing the user commands to the Logic file. The Logic file would start the appropriate modules and execute the commands from the GUI. Logic would also do the data gathering, and passing information that needed to be stored permanently to the dbAPI file. The dbAPI would then save the information in an appropriate location. The same chain of commands would also be used to gather information from the dbAPI and show it to the user, with a command call going from the GUI to Logic to dbAPI, and then information going back from dbAPI to Logic to GUI.

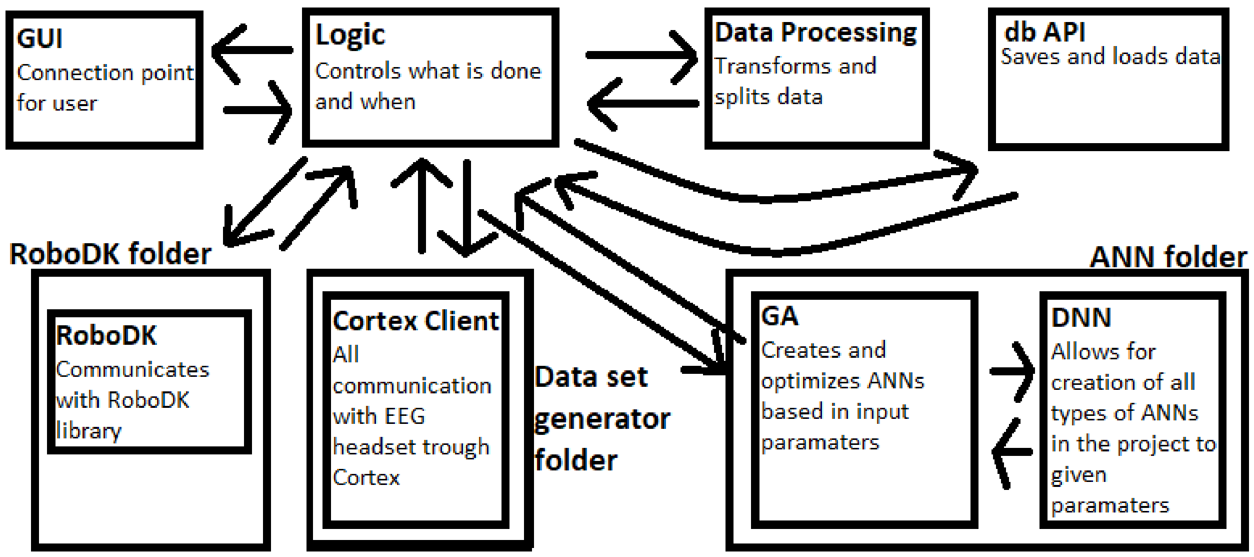

GUI
Connection point for user
Logic
Controls what is done and when
Data Processing
Transforms and splits data
db API
Saves and loads data
RoboDK folder
RoboDK
Communicates with RoboDK library
Cortex Client
All communication with EEG headset trough Cortex
Data set generator folder
ANN folder
GA
Creates and optimizes ANNs based in input paramaters
DNN
Allows for creation of all types of ANNs in the project to given paramaters

Figure 3.16: Current Code Structure

The GUI is made for easy and fast addition that fits the style of the GUI. It is running on its own multiprocess so it won't interfere with the running of the other components of the code. The functions considered for the GUI was profile creation, for saving information on users, allowing for the use of NNs trained in previous sessions. Real time graphing of sensor signals and real time sensor contact quality giving a visual feedback of the situation both with readings and connections. A visual of the AI structure and easy input for choosing hyperparamaters for a single NN, or upper and lower limits to be used with the GA.

To handle different profiles for several users a login screen 3.17 was created

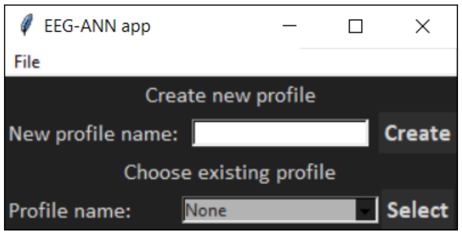


Figure 3.17: Login Screen

A main window is split into four parts 3.20. One part showing headset information, allowing for connection to a headset connected to the PC. A part dedicated to all aspects of the NNs and GAs. One for showing contact quality 3.18 and graph data from the sensors 3.19. A last part for managing all profile related options.

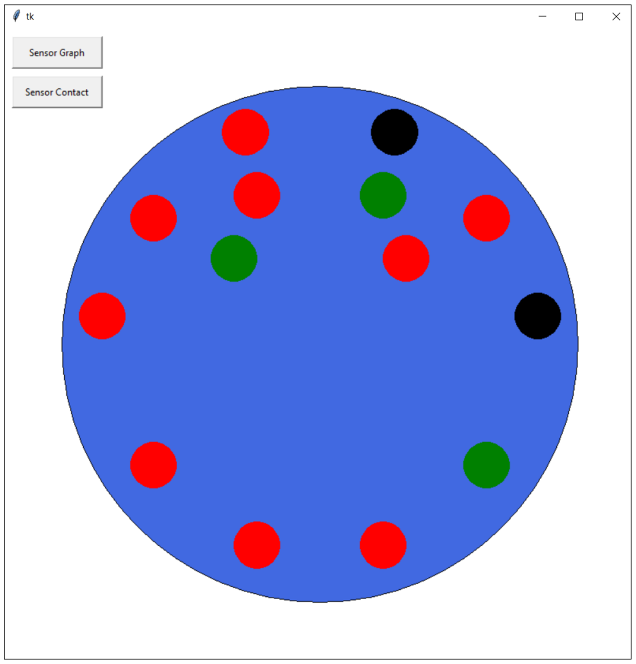


Figure 3.18: Contact Quality. Black is no contact. Red is poor contact. Yellow is average contact. Green is good contact

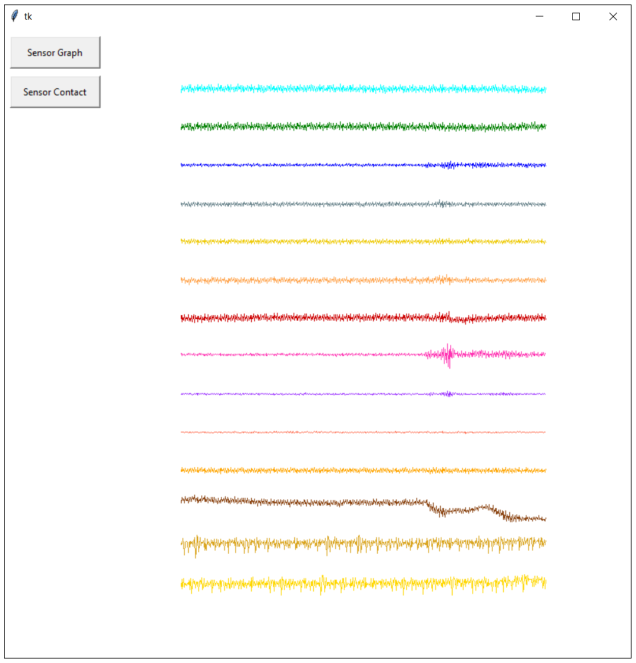


Figure 3.19: Sensor Graphs. One line for each node.

The layout of this window would be easily changeable and reset back to any of three preset options. Personalizing the interface to any users preferences. All parts was intended to be separable. Allowing for spreading the parts over several monitors in separate windows.

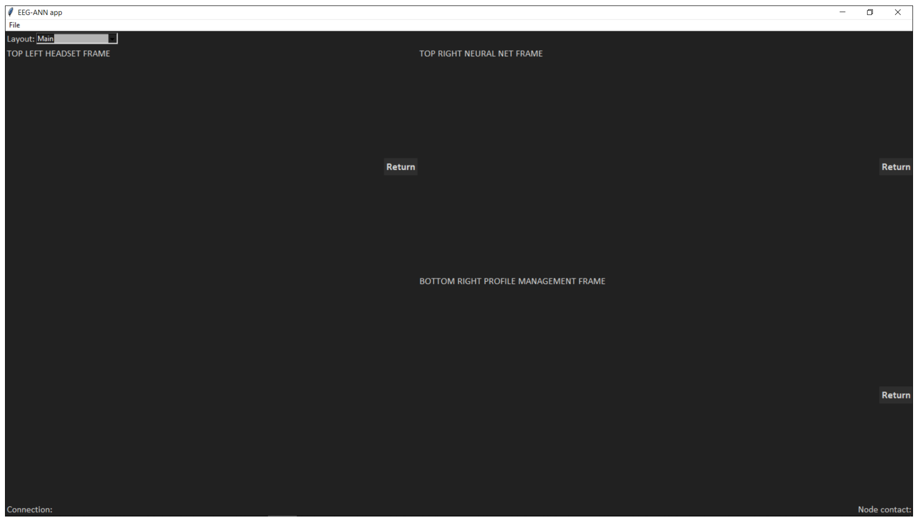


Figure 3.20: Main view

Most of these features were implemented. Some of the features were not updated with the code reconstructions and is now still a few days work away from being implemented back inn. The GUI was abandoned as the equipment got delivered and the authors could start testing to get the crucial parts of the code working properly. Leaving the GUI in an unfinished state.

As seen in the figure 3.16 the Logic file was the center of code. All information and commands ran through or was initiated by the Logic file. The Logic file initiated data gathering and stored or loaded the data sets. Since the GUI was never implemented the commands were initiated directly in the Logic file, and not by the GUI.

Since the dbAPI module did not need to store any local variables, but instead worked as a collection of database commands, it was created as a singleton module handling saving and loading of data. The module had commands for creating, inserting, deleting and selecting both tables and values into or from existing tables. These commands handled the connection and the database communication for the user. The user did not need to have any information about what database system was being used, or how to use it.

The dbAPI was not implemented as Pickle 2.15.8 was used to store the prerecorded data sets and Tensorflow's 2.15.2 own way of storing models was preferred over storing the models manually.

The data processing module, dataProcessing.py, handled all the data processing. There were three main data processing tasks this module handled, FFT 2.4, wavelet transform 2.4 and splitting the data for K-Cross 2.9 validation. Since every command would receive all the information needed through arguments and return a return value directly, this module was created as a singleton module, not storing any local information.

Implementation with Cortex 2.15.7 through CortexClient, was one of the most important parts of the code as it was required to get data from the headset. Emotiv came out with a new software and stopped supporting the old iteration that the mentors had used in previous projects. This switch happened right before the start of the project. As there were no examples for implementation of Cortex with Python, the authors had to write this implementation from scratch. Since Emotiv did not plan to add any examples for Python they did not answer any questions posted regarding Python either. The authors also noticed that Emotiv had a tendency to only reply to quarries one day each month, and not all quarries were answered. At the time of writing the author's CortexClient is one of the closest things to an example for use of Cortex with Python, it has even been posted as an example to a quarry at their github. For any future projects considering to build on this thesis Emotiv's lack of response should be taken into account when choosing hardware.

CortexClient was written to be used as a Python import module. In essence this meant that nothing should be started by the module itself, and all processes and return values should be generated by using functions with arguments. Because of this philosophy the only thing stored as global variables were the user information, the URI, and the WSS connection. This was done to make the use of the module easier, since the user did not have to pass this information as an argument with every command.

The biggest challenge when writing CortexClient turned out to be with the WSS's SSL. Even though Cortex was located on an encrypted WSS address, when communicating with Python Cortex needed SSL to be set to unverified, essentially making the connection unencrypted.

The genetic algorithm file GA was created with customization in mind. Every aspect of the project could be customized and optimized for trough the GA. These aspects could also be opted out of the optimization. All these values were designed to be changeable with simple changes to integer values, making implementation with the GUI easy. K-Cross validation 2.9 was used to ensure the results from the training was reliable and not biased.

The dynamic NN file dNN was created with a GA in mind. It took input parameters that were changed with integer inputs and strings. This allowed for easy randomization of all important parameters for testing. It could create a NN of all types and with all data types used in this project. The size of these NNs, both the number of hidden layers and the neurons in them, and the loss functions were among the changeable parameters. These parameters could be customized with simple input changes. This made implementation with the GUI simpler, as buttons on drop down menus would be able to change all parameters.

Chapter 4

Results

The results show that the data processing done before the training process is vital for the predictive properties of the NNs. Time to train, size of NN and accuracy varies a lot depending on the state the data is in.

4.1 Raw data

Raw data is shown to have a low prediction rate within reasonable data set sizes. The predictions done in this project varied between random guessing and close to random guessing. Were 5 different commands yielded 20% – 30% accuracy, with a trend showing higher prediction accuracy correlating with larger data sets. The data sets gathered were not sufficient to get a usable prediction rate. Neither mental commands and movement related data gave a result barely better then pure guessing.

4.2 DFT

DFT data slowed down the training rate and is extremely prone to overfitting. It dose show way better results with lower loss and higher accuracy, making it more viable for prediction then the other methods used in this project as it reached 97% prediction accuracy. The highest accuracys were achieved with movement related data. Mental commands resulted in noticeably lower accuracy. It was however still usable for controlling the robot arm, though barely.

4.3 Wavelet

Wavelet data resulted in the same drawback as DFT, with slower training rate. Overfitting was not a problem with this method. It gave better loss and accuracy then raw data. It is still not up to values that would make it functional for prediction for use with a robot arm. Neither mental commands and movement related data gave a result usable for the scope of this project.

4.4 Data sheet

The table below shows a small excerpt of results from the GA optimization.

Type	Dataset	Loss test	acc. test	raw	DFT	Wavelet	Net shape	Epochs
Dense	**<120000**	**0.30**	**88%**	**false**	**true**	**false**	**(8, 4)**	**15**
Dense	<120000	1.60	27%	true	false	false	(171, 26, 174, 33)	61
Dense	<120000	1.38	43%	false	false	true	(113, 3)	1
Conv1D	<20000	2.8	0.06%	true	false	false	(-)	-
Conv1D	<120000	1.63	22%	true	false	false	(177,185,150,90,40)	73
Conv1D	<120000	1.89	43%	true	false	false	(140,200,138,95,48)	13
Conv1D	<120000	2.03	43%	true	false	false	(52,184,175,95,23)	13
Conv1D	<12000	0.54	84%	false	true	false	(8, 4)	9
Conv1D	<120000	1.38	43%	false	false	true	(2)	3
Conv1D	<120000	1.38	43%	false	false	true	(78, 99)	3
Conv2D	<120000	0.50	87%	false	true	false	(8, 4)	9
LSTM	<120000	1.60	25%	true	false	false	(228, 243, 38)	13
LSTM	<120000	1.74	29%	false	true	false	(12, 5, 3)	5
LSTM	<120000	1.83	39%	false	true	false	(4)	15
LSTM	<120000	1.46	43%	false	true	false	(2)	8
LSTM	<120000	1.46	43%	false	true	false	(7, 9, 2)	15
LSTM	<120000	1.38	41%	false	false	true	(28, 7, 2, 27, 18)	7
LSTM	<120000	1.38	40%	false	false	true	(10)	12
LSTM	<120000	1.38	41%	false	false	true	(7, 6, 7)	8
LSTM	<120000	1.38	41%	false	false	true	(9, 4)	11
LSTM	<120000	1.38	40%	false	false	true	(3)	13
-	<-	-	-%	-	-	-	(-)	-

Table 4.1: GA Optimization Table

4.5 Fully Connected Neural Network

The fully connected NNs showed limited prediction abilities throughout the project, getting no usable results within the scope of the project. However, this changed at the last minute when one setup outdid all other NNs on the biggest dataset.

4.6 Convolutional Neural Network

CNNs showed varied results depending on the data and type of CNN. The two dimensional version gave the best results, while the one dimensional version gave results close to the other types of NNs. Making the two dimensional version a viable option to control a robot arm.

4.7 Long Short Term Memory(RNN)

LSTMs showed lower loss than fully connected, but nowhere near low enough to give usable results, making it untenable to use for controlling a robot arm.

4.8 Best result

A single NN showed the greatest potential, this was a two dimensional CNN that was tested on its own data set and then tested live. The parameters used for this NN was two lairs with 64 neurons each. 50 epochs were used to train it, and it was fed DFT data from a smaller data set. This setup resulted in a loss rate of 0.04 and a accuracy of 97%.

4.9 Live

The results of the GA optimization shows that most approaches tried in this project is not viable for controlling a robot arm. A methods that was tested early on resulted in usable accuracy for the control as live testing. This was then tested with live and connected to a simulated robot arm. The results from this testing shows that control is clearly possible with data from movement, and barely possible with data from mental commands.

Chapter 5

Discussion

5.1 EEG data interpretation

There has been a continuous discussion among the authors about how to deal with EEG data. Time domain and frequency domain data has been the most discussed subject for this project. They both seem to have their strengths and weaknesses in terms of AI interpretation. The time domain seems to present challenges of complexity and the frequency domain could be seen as not complex enough, resulting in signals that are reduced to the point were it is to limiting.

5.2 Data types

The raw data 3.3.1 show an interesting trend towards better prediction and lower loss from bigger data sets. The authors could not make a data set big enough to find a point were this trend ends, making this a perfect point of inquiry for future projects. The data sets that were tested are somewhat substantial, so a new approach to gathering the data sets should be considered. Combining this with testing to see if there is a commonality of signals from different people could be a way to test this. Gathering data from several people and training a NN on the accumulated data from everyone.

DFT 2.3.1 has shown the most promising results. Being comparable to Emotivs own software. The use case laid out in this project is viable with this approach and further testing with regards to mental commands could be of great interest to future projects. The tendency of this method being easily over fitted should be kept in mind and could be explored.

Wavelet 2.4 gave better results then raw data. These result were still lacking with regards to the goal of this project. This might be a method worth exploring more, as the code for testing wavelet was finished at the tail end of the project resulting in less time to test and refine the parameters for this method. The strange constancy of the results with wavelet regardless of the shape of the NNs is also a point worth looking into. There could be an error in the way the wavelets are constructed that is the root of this phenomenon.

5.3 Neural Networks

The fully connected NNs has performed the worst over course this project, most of the predictions were about the same as random guessing. Larger data sets seem to increase the average accuracy marginally. Since this trend kept going at the same rate up to the largest data set generated for this project, it opens for future exploration of the data set size and it's effect on this type of NN.

The fact that a result came in at the tail end of the project where a specific setup for a fully connected NN outperformed all others on the biggest data set shows that there might still room for further testing when it comes to optimization of the hyperparameters.

The CNNs 2.7 seem to give the best results, both one dimensional and two dimensional CNNs produce the best results in the testing done for this project. They are easily over fitted making the successful networks small, more so when combined with DFT data. The results from a data set only tested on two dimensional CNN did end up with the absolute best result. With a surprisingly large CNN compared with the size needed with the other data sets. This was initially speculated to be an outlay-er seeing how it's not replaceable with other data sets. This setup did however produce a result in live testing that clearly shows it working to a satisfactory degree with motor related data.

The LSTMs 2.8.1 preformed better then the fully connected NNs when compared with the same parameters. This is to be expected as it has the memory gates to utilize data from previous inputs. This does add to the computational load compared to fully connected.

5.4 Mental commands

As the authors got more familiar with Emotiv's EEG systems it became clear that the claims of Emotiv to be able to use mental commands with their software and headset's was not as clear cut as it sounded. Experience showed that Emotiv's own software had problems recognizing mental commands, managing little to nothing with regards to distinguishing individual mental commands. Movement related commands seemed to be within reasonable grasp for this software. This can be down to the softwares ability to predict, or some fault in the way the equipment was used in the testing. The last point seems less likely as the results from the project give similar results as Emotivs own software, making further testing and refining when it comes down to the hyperparameters of interest in future projects. However physical movements seems to grant the most reliable predictions in both softwares. Mental commands might need more nodes over the visual cortex to work properly. This could be an interesting subject for future projects.

5.5 Epoc+ vs Epoc Flex

The late arrival of the equipment allowed for too short of a time frame to test the Epoc Flex. It is hard to predict if the Flex would have given better results. The extra nodes might have helped getting better predictions, as larger data sets give better results with some methods. While the best approach seemed to do well without the extra data, seemingly not needing the extra data. The quality of the nodes might also be problematic, since the nodes on the Epoc Flex and Epoc+ are the same, there may not be a large difference in predictability. Better nodes might help predictability more than more nodes would do. This could be of interest in future projects.

Chapter 6

Conclusions

Developing a combined NN and signal processing method for interpreting EEG signals is possible. This has been demonstrated to varying degrees of success in this project. Using movement related data allowed for a satisfactory degree of control over a simulated robot arm. This was not achieved with regards to mental commands. Some vague control could be demonstrated, this was however extremely weak and should not be considered usable.

The combination that managed the best results for this project was a two dimensional CNN with DFT as the signal processing method. This was the method that show most promise in training and the one to be tested live.

More testing should be of great interest as all aspects of these combinations couldn't be fully explored within this project. Data gathering, data processing, further testing of parameters for the NNs and comparing results from different equipment would all be potential future projects as the code is both set up for this and created with future expansion in mind.

Appendices

A Preproject report

Preproject report in it’s entirety:

Preliminary report

Christian Ovesen(476132), Kenneth Sverre Verlo Jacobsen(769854),
Kenneth Andre Olsen(488563)

May 6, 2019

Subject code:
IE303612
Subject:
Bachelor thesis
Document access:
Open
Course:
Automation Engineering
Number of pages/References and Appendix:
14/1
Employer/metors:
NTNU CPS Lab/Robin T. Bye, Anete Vagale
Summery:
In this research project we will look at the feasibility of using a ANN to interpret EEG data. This interpreted data will then be used to control a Sawyer robotic arm. Comparisons in speed and accuracy will be made with other peoples solutions. The research will be expanded on with testing on other utilities or other interpretation methods if time allows for it.

Contents

1 Introduction

As a part of the study Automatiseringsteknikk at NTNU Aalesund we were assigned a bachelor thesis exploring the possibility and suitability of using EEG as a controller. During this feasibility study we created a timetable for the project, decided on rules and regulations regarding the group dynamics and looked into the prerequisites to complete the project. This rapport documents the decisions and findings of the feasibility study.

The assignment was given as a scientific project by NTNU Aalesund, as one of several possible choices. What intrigued us about this particular project was it level of applicability to real life issues. Specifically the idea of being able to help people. This may be individuals who have lost an arm or a leg, who have lost mobility either temporarily or permanently or simply people who need extra strength to carry out their job.

We chose to use EEG to control a Sawyer robot, and our goal will be to be able to freely control the Sawyer in 3D-space.

2 Concepts

- NTNU - Norges teknisk-naturvitenskapelige universitet(Norwegian University of Science and Technology) is an internationally oriented university based in the city Trondheim with campuses in the cities Gjovik and Aalesund in Norway.
- BCI - Braincomputer interface is a communication method between a brain and a computer.
- EEG - Electroencephalography is a non invasive monitoring method to record electrical activity in the brain.
- AI - Artificial Intelligence is intelligence demonstrated by machines, in contrast to the neural intelligence displayed by humans and other animals.
- Algorithm - An algorithm specifies how to solve a class of problems, using calculations, data processing or automated reasoning tasks.
- ANN - Artificial Neural Network is a framework for many different machine learning algorithms to work together and process complex data inputs.
- GA - Genetic Algorithm is a method for optimizing a given task inspired by evolution.
- Python is an interpreted, high-level, general-purpose programming language.
- IDE - Integrated development environment normally consists of a source code editor, build automation tools, and a debugger.
- L167 is a room on the Aalesund campus of NTNU
- 3D-space - Three-dimensional space is a geometric setting in which three values are required to determine the position of an element.
- Sawyer - Sawyer is the revolutionary collaborative robot designed to execute tasks that have been impractical to automate with traditional industrial robots.
- GUI - Graphic User Interface
- Hyperparameter - In machine learning, a hyperparameter is a parameter whose value is set before the learning process begins.
- BitBucket - A web-based git repository for version control, created by Atlassian.

- Confluence - A team collaboration software used to share create documents. Created and published by Atlassian.
- Jira - A team task assignement software by Atlassian.
- Gantt diagram - TODO

3 Project organization

3.1 Project

3.1.1 Assignment for the project - organization

- Write preliminary report.
- Set up and connect the physical components of the system and test it.
- Determine which practical use there can be with this type of technology.
- Choose which equipment that will suit the project the best.
- Determine how we can fetch and use the signals from the EEG equipment with the use of a program written in Python.
- Write a program for the manufacturing of a data set.
- Manufacture data set for training of ANN.
- Training of ANN for interpretation of EEG signals.
- Write a computer program for the interpretation of the signal states from the EEG equipment in real time that can determine different states: Up, Down, Left, Right, Forward, Backwards, Grip and Release.
- Make profiles for the EEG-subjects.
- Determine how to instruct the Sawyer robot arm and how we can send instructions to it.

3.1.2 Project leader

- Responsibilities Responsible to keep the group functioning well, and make sure that the group members are doing their assignments. Also responsible for the biweekly status meeting with the mentors.
- Assignments Call for a biweekly meeting, hold the meeting and present report on the last two weeks.

3.1.3 Secretary

- Responsibilities Responsibilities during biweekly meeting with mentors.
- Assignments Write meeting report.

3.1.4 General members

- Responsibilities The general workload and finishing the thesis report.
- Assignments Completing their assigned tasks within the given time.

3.2 Mentors (Advisor and employer contact)

- Robin T. Bye. (robin.t.bye@ntnu.no)
- Anete Vagale. (anete.vagale@ntnu.no)

4 Agreements

4.1 Agreements with employer

The assignment was given by NTNU without any requirements. In agreement with the employer it was decided to research the us of an ANN to process the data from an EEG to control a Sawyer robotic arm.

4.2 Workplace and resources

4.2.1 Workplace

- L167 at NTNU is available Tuesday till Friday from 0800 - 1600, in the period the 8th of January 2019 till the 3rd of May 2019

4.2.2 Resources

- The EEG equipment will be available for use from the 10th of January till we have finished our thesis.
- The Sawyer robot will be available for use from the 10th of January till we have finished our thesis.

4.2.3 Data security

- Since the focus of the thesis is research no data from the thesis will be withheld from the public.
-

4.2.4 Scheduled reporting

- Every two weeks we will have a meeting and deliver a short report on our progress the last two weeks, and plans for the next two weeks.

4.3 Group dynamics

The group has decided on working together at specific times. We will work from eight to four every weekday, with exceptions on Wednesdays and the days we have the course "Industri 4.0". These work days will be structured with work on the project from eight to two, and writing the bachelor thesis. Coding will be object oriented and work on the classes will be split between the group members, while non coding tasks will be worked on as a group.

The group wants to create a precedence for a structured, focused and steady work flow that can be easily carried over to future projects we will be working on as automation engineers. We want to make the code base flexible, allowing it to be easily adapted to other projects. The fact that the thesis isn't strictly defined or given by a company that wants a specific product allows us the flexibility to expand on the project if time allows for it. This is the reason we want to create a structured and steady work flow with hopes to finish well before the dead line, allowing us time to expand on the project and try to implement the core of the project in more ways or with different methods to give us a bigger scope in the thesis.

5 Project description

5.1 Assignment - goal setting - purpose

The purpose of this project is to control a Sawyer robotic arm with the use of EEG equipment. We will start out small, and expand on what we are able to do based on the time we have at our disposal. Firstly we want to get the core code implemented. That being the code for generating data sets from the EEG readings and the AI for interpreting that data. We will then use that code for controlling a robot arm in 3D space. Comparing our results with results we get with techniques gathered from others doing similar work. Trying to implement the core of the project(interpreting EEG signals) in other ways then a robotic arm is also something we want to try if time allows for it.

5.2 Requirements for solution or project results - specification

The thesis was given by NTNU as a research assignment. This research is aimed at testing the feasibility of using an ANN processing the raw data output from an EEG to control a Sawyer robotic arm. There is no specific requirement for results, economic feasibility or quality.

This research can be considered finished when we reach the time limit and can no longer expand on the work we have done up to that point.

5.3 Planned progression - methods

The approach would be to find a method for interpreting EEG signals starting of by trying a ANN. The training of the ANN will most likely be done with the back-propagation algorithm by using a data-set that we will collect and store. This part entails some experimenting and testing for optimizing both the ANN and how the signals from the EEG should be interpreted before it is fed to the ANN. Here we could optionally use GA for optimizing the hyperparameters for the ANN.

- Pros using ANN for signal interpretation:
  - Can often make sense of difficult and complicated problems where other methods fail.
  - Moderately easy to implement and could work exceptionally well.
- Cons using ANN for signal interpretation:
  - ANN are what is called a black box method, this means we can not describe accurately how it distinguishes in it's interpretation.
  - We can't extract much knowledge about the problem by the use of a ANN.
  - ANN might not work as well as we might expect.
  - The ANN will need to be trained for every user.

- Known weaknesses of ANN:
  - Over-fitting.
  - Could be CPU resource demanding.
  - Needs a relatively big data-set.
- How to overcome/combat weaknesses of the ANN:
  - Over-fitting happens when we use to many neurons and thus the ANN remembers the data-set and looses its ability to generalize. This problem will be avoided by experimentation or optionally by optimizing hyperparameters with GA. This means that we will be using the trial and error method.
  - ANN could be CPU resource demanding. There are several ways to get around this, one is by compromising between the amount of neurons and the classification error percentage we can limit the CPU usage if necessary. We could borrow a high specification computer. Use the GPU instead of CPU for the computations.
  - ANN needs a relatively big data-set. This could be solved by writing a good semi-automated program for fetching the data that we need. There are also other techniques that we could use like duplicating the data and then adding noise to it, this way expanding the data in the data-set.

5.4 Information gathering - plan and execution

There are similar studies looking into EEG signal analyses. Luis Alfredo Moctezuma and Marta Molinas [1] look into the accuracy of biometric recognition when reducing the number of nodes used on the EEG. We want to compare their method of data processing with our solution if time allows for it. This study [1] helps identify nodes of interest and the importance of the number of nodes used. We will have a higher number of variables as output making the accuracy lower and more nodes necessary to get the fidelity we want when controlling a Sawyer robotic arm. Saugat Bhattacharyya, Debabrota Basu, Amit Konar and D.N. Tibarewala [2] used a fuzzy logic algorithm to control a robotic arm from EEG inputs. Their solution would be an interesting comparison for our accuracy.

Locating suitable sources for information of existing studies will be done using scholar.google.com, NTNU's own databases and consulting with our mentors.

5.5 Review - analasys of risk

We know that this project can be a success given enough time and knowledge, since we have found similar projects that have been a success. The manufacturer of the EEG equipment has managed to distinguish between the necessary mental commands needed for controlling the Sawyer robot arm. And we know that the robot arm can be controlled from a computer.

Assessing what will be important and what will be threats for success:

- Important for success:
  - Motivation.
  - A good plan for the project.
  - Time and time management.
  - Good knowledge.
  - Good solutions.

- Good communication between group members and good communication between group and mentors.
- Fast and easy conflict resolutions.
- Reliable equipment.
- Equipment with high enough specifications.

- Threats for success:
  - Equipment malfunction.
  - Starting of in the wrong direction.

Assessing risk elements and safety aspects:

- Risk elements:
  - Equipment malfunction.
- Safety aspects:
  - The robot arm could hurt someone if not careful enough.

5.6 Main activities for further work

Number	Main activities	Responsibility	Expected time	Expected scale
A1	Preperation	All	2 weeks	three people
A2	Thesis writing	All	Whole project	three people
A3	Data set	All	3 weeks	three people
A31	Profile class	General member	2 weeks*	one person*
A32	Data generation class	General member	2 weeks*	one person*
A33	GUI	General member	2 weeks*	one person*
A4	AI	All	3 weeks	three people
A41	Cost function	General member	2 weeks*	one person*
A42	AI training	General member	2 weeks*	one person*
A43	GA optimization	General member	2 weeks*	one person*
A5	Robot arm implementation	All	3 weeks	three people
A6	Accuracy/speed tests	All	2 weeks*	three people
A7	Other implementations	All	Remaining weeks	three people

* subject to change

5.7 Progress plan - management of project

5.7.1 Main plan

- Preperation
  - Feasibility study - Planning and preparing for research.
  - Gathering of information - Finding studies and previous bachelor thesis that are relevant to our research.
  - Equipment testing - Testing of all equipment to be used in research, to make sure it is all functional.

  - Skill development - Developing skills that we are not proficient in yet, that will be needed.
  - All members of the group are responsible for the preparation and will take part in all the sub activities. Expected time consumption is two weeks.
  - Start date: 8. Jan. 2019
    End date: 18. Jan. 2019

- Thesis writing - All members of the group are responsible. Will be written during the entire period.
- Data set - Writing the program that will collect and store the data sets.
  - Profile class - Only one member of the group will be active and responsible for the completion of this task. Expected time consumption is two weeks.
  - Data generation class - Only one member of the group will be active and responsible for the completion of this task. Expected time consumption is two weeks.
  - GUI - Only one member of the group will be active and responsible for the completion of this task. Expected time consumption is two weeks.
  - Activity is expected to be done in three weeks. Divided into two weeks for all three sub activities, and one week assembly and bugtesting.
  - Start date: 21. Jan. 2019
    End date: 8. Feb. 2019
- AI - Writing the AI responsible for interpreting the data received by the EEG.
  - Cost function - Only one member of the group will be active and responsible for the completion of this task. Expected time consumption is two weeks.
  - AI training - Only one member of the group will be active and responsible for the completion of this task. Expected time consumption is two weeks.
  - GA optimization - Only one member of the group will be active and responsible for the completion of this task. Expected time consumption is two weeks.
  - Start date: 11. Feb. 2019
    End date: 1. Mar. 2019
- Robotic arm implementation - All members of the group will be active and are responsible. Expected time consumption is three weeks.
  - Start date: 4. Mar. 2019
    End date: 23. Mar. 2019
- Accuracy/speed tests - Comparing our research and results with other comparable research and methods for data processing in the field. All members of the group will be active and are responsible. Expected time consumption is two weeks.
  - Start date: 25. Mar. 2019
    End date: 5. Apr. 2019
- Other implementations - Expand our research. It is expected to be some weeks left when most of the research is done. These weeks will be used to expand our research. All three members of the group are responsible for this task. Expected time consumption is the remaining weeks.

5.7.2 Management tools

- BitBucket

  - Code merging
- Confluence
  - Documentation
  - ”wiki”
- Jira
  - Gantt diagram
  - Time table
  - Task assignment
  - Activities

5.7.3 Development tools

- EEG Headset.
- Desktop/Laptop
- Robotic arm

5.7.4 Internal control - evaluation

Group members will meet every week to discuss progress on tasks. Targets are reached when the group is satisfied the program works sufficiently to move on.

5.8 Decisions - decision process

Important decisions regarding limitations of the research, goal, progress and group dynamics have been made using a democratic approach. First the group has discussed the decision to be made. After the discussion the decision is taken based on the consensus of the group.

This approach will be used for all important decisions moving forward.

6 Documentation

6.1 Reports and technical documents

- There will be several forms of documentation produced
  - Preliminary report
  - Bachelor thesis
  - Progress reports
  - Agenda document for mentor meetings
- We will work every week day, except Wednesday, that does not overlap the other course for the semester, planed meetings. The work will be 08.00-14.00 for the project and 14.00-16.00 for writing thesis. There are weekly project meetings with a few exceptions, and bi weekly mentor meetings with a few exceptions.

- The preliminary report will be submitted for approval in January. Progress reports and agenda documentation will be submitted weekly and biweekly for approval. TODO - les igjennom - godkjennelse, har jeg tolket punket rett?
- Distrubution of the bachelor thesis will be done by NTNU after it is graded. All code written for the researched will be availeble and distributed through BitBucket.
- All documents are written in overleaf and will be stored there during the project. All group members and mentors will have access to these documents at all times.
- Preliminary report is planned to be finished by fifteenth of January. The thesis will be worked on as specified above. Progress reports and agenda for mentor meetings, will be worked on in the project meetings.

7 Planed meetings and reports

7.1 Meetings

7.1.1 Meetings with mentors

- Wed 23 Jan 10-11
  - Progress report
  - Time and work flow evaluation
  - Activity discussion
  - Plan revision discussion/reevaluation
- Tue 5 Feb 10-11
  - Progress report
  - Time and work flow evaluation
  - Activity discussion
  - Plan revision discussion/reevaluation
- Tue 19 Feb 10-11
  - Progress report
  - Time and work flow evaluation
  - Activity discussion
  - Plan revision discussion/reevaluation
- Tue 5 Mar 10-11
  - Progress report
  - Time and work flow evaluation
  - Activity discussion
  - Plan revision discussion/reevaluation
- Tue 19 Mar 10-11
  - Progress report
  - Time and work flow evaluation
  - Activity discussion
  - Plan revision discussion/reevaluation
- Tue 2 Apr 10-11
  - Progress report
  - Time and work flow evaluation
  - Activity discussion
  - Plan revision discussion/reevaluation
- Tue 16 Apr 10-11
  - Progress report
  - Time and work flow evaluation
  - Activity discussion
  - Plan revision discussion/reevaluation
- Tue 30 Apr 10-11
  - Progress report
  - Time and work flow evaluation
  - Activity discussion
  - Plan revision discussion/reevaluation
- Tue 14 May 13-14
  - Progress report
  - Time and work flow evaluation
  - Activity discussion
  - Plan revision discussion/reevaluation
- Tue 28 May 10-11

  - Progress report
  - Time and work flow evaluation
  - Activity discussion
  - Plan revision discussion/reevaluation
  - Project finalization and thesis finalization
  - Result discussion

7.1.2 Project meetings

- Thu 17 Jan 09.00-09.45
  - Progress report
  - Work flow reevaluation
  - Time estimation reevaluation
- Thu 24 Jan 09.00-09.45
  - Progress report
  - Work flow reevaluation
  - Time estimation reevaluation
- Thu 31 Jan 09.00-09.45
  - Progress report
  - Work flow reevaluation
  - Time estimation reevaluation
- Thu 7 Feb 09.00-09.45
  - Progress report
  - Work flow reevaluation
  - Time estimation reevaluation
- Thu 14 Feb 09.00-09.45
  - Progress report
  - Work flow reevaluation
  - Time estimation reevaluation
- Thu 21 Feb 09.00-09.45
  - Progress report
  - Work flow reevaluation
  - Time estimation reevaluation
- Thu 28 Feb 09.00-09.45
  - Progress report
  - Work flow reevaluation
  - Time estimation reevaluation
- Thu 7 Mar 09.00-09.45
  - Progress report
  - Work flow reevaluation
  - Time estimation reevaluation
- Thu 14 Mar 09.00-09.45
  - Progress report
  - Work flow reevaluation
  - Time estimation reevaluation
- Thu 21 Mar 09.00-09.45
  - Progress report
  - Work flow reevaluation
  - Time estimation reevaluation
- Thu 28 Mar 09.00-09.45
  - Progress report
  - Work flow reevaluation
  - Time estimation reevaluation
- Thu 4 Apr 09.00-09.45
  - Progress report
  - Work flow reevaluation
  - Time estimation reevaluation
- Thu 25 Apr 09.00-09.45
  - Progress report
  - Work flow reevaluation
  - Time estimation reevaluation
- Thu 2 May 09.00-09.45
  - Progress report
  - Work flow reevaluation
  - Time estimation reevaluation
- Thu 9 May 09.00-09.45
  - Progress report
  - Work flow reevaluation
  - Time estimation reevaluation

- Thu 16 May 09.00-09.45
  - Progress report
  - Work flow reevaluation
  - Time estimation reevaluation
- Thu 23 May 09.00-09.45
  - Progress report
  - Work flow reevaluation
  - Time estimation reevaluation
- Thu 30 May 09.00-09.45
  - Progress evaluation
  - Progress report
  - Hand inn preparation

7.2 Periodic report

7.2.1 Progress reports(including milestone)

Biweekly we will present our mentors with a report of the work done in the last two weeks, and the work planned for the next two weeks.

Report presentation dates:

- Wed 23 Jan 10-11
- Tue 5 Feb 10-11
- Tue 19 Feb 10-11
- Tue 5 Mar 10-11
- Tue 19 Mar 10-11
- Tue 2 Apr 10-11
- Tue 16 Apr 10-11
- Tue 30 Apr 10-11
- Tue 14 May 13-14
- Tue 28 May 10-11

8 Planned deviation management

If things do not go as planned and it can not be handled individually there will first be a discussion between the group members were one or more of the following actions will be determined:

- Problem evaluation.
- Talk to our mentors and/or call for a meeting with our mentors.
- Research about the problem or similar problems.
- Contact manufacturer.
- Find and order alternative equipment.
- Find other solutions/alternatives.

9 Equipment requirements/assumptions for implementation

The main components required for this project are:

- EEG Headset.
  - The EEG headset needs to have good enough specifications for differentiating between the individual mental commands which will be used where the most important specifications are sampling frequency and number of electrodes.
- Robotic Arm.
  - Robotic arm with 6 degrees of freedom that can receive instructions from a desktop/laptop.
- Desktop/Laptop.
  - The computer needs good enough specifications for running a program in real time which identifies and differentiates between individual mental commands and in addition sends commands to a robot arm.

10 References

10.1 References

[1] Moctezuma, Luis Alfredo and Marta Molinas.
Subject identication from low-density EEG-recordings of resting-states: A study of feature extraction and classication,
Future of Information and Communication Conference (FICC 2019) (2019):.

[2] Saugat Bhattacharyya, Debabrota Basu, Amit Konar and D.N. Tibarewala.
"Interval type-2 fuzzy logic based multiclass ANFIS algorithm for real-time EEG based movement control of a robot arm."
Robotics and Autonomous Systems Volume 68, June 2015, Pages 104-115.

B Meeting notes

Kick-off Meeting:

Attendees Christian Ovesen Kenneth Jacobsen Kenneth Andre Olsen Robin T. Bye Anete Vagale

Goals Legg inn agenda dagen før status møte. Statusskjema: 2 ukes perioder: Hva skulle vi? Hva fikk vi gjort? Forskjell? Hva gjør vi neste uke? Innkall til møter, annenhver uke resten av semesteret. Hør med Anders/Ottar om vi kan bruke sawyer hele semesteret, eller om vi må booke tid. Hør med Webjørn hva som skjer med forprosjekt. Og om statusskjema-mal. Bruke wiki (bitbucket, confluence and jira.) Colines i Trondheim. Gjør målbare eksperiment. Mål vår AI opp mot emotivs egen. Andre signal i hjernen: P300, SSVEP

2019-01-23 Meeting notes

Attendees Christian Ovesen Kenneth Jacobsen Kenneth Andre Olsen Robin T. Bye Anete Vagale Thiago Gabriel Monteiro

Goals Use EPOC FLEX till the new one arrives Sawyer does not work with windows(?). Could use kuka instead. Get Emotiv Epoc PRO License RoboDK FFT If he has a way to maximize degrees of freedom with ANN Other

2019-02-27 Meeting notes

Attendees Christian Ovesen Kenneth Jacobsen Kenneth Andre Olsen Robin T. Bye Anete Vagale

Goals Demo Look into progress Plan on future progress

Comments Christian Ovesen Talk to Elias about reinforcement learning.

2019-03-11 Meeting notes

Attendees Christian Ovesen Kenneth Jacobsen Kenneth Andre Olsen Robin T. Bye Anete Vagale

Goals Get permission to use license so we can get started Talk about work restructure Talk about the lack of equipment

Comments Robin T. Bye Dell has apologized. This is what happens when my order has to go through two NTNU people and then Dell and is out of my control. A computer will arrive on 19 March. We cannot lock the license to one of your machines for obvious reasons so I hope you can survive until then and do other tasks until its arrival. Delays should be mentioned in the report and will be taken into account among the censors. See the attachment for specs:

ReplyEditDeleteLikeMar 08, 2019 Robin T. Bye Are you 100% sure that the license is locked to a single computer forever? It would be nice if you could "unlock" it from at computer and lock it to another one later. If so, you could use the license on your own computer already today and transfer it to the new computer when it arrives.

2019-04-02 Meeting notes

Attendees Christian Ovesen Kenneth Jacobsen Kenneth Andre Olsen Robin T. Bye Anete Vagale

Goals New group dynamics Discuss questions related to thesis paper Discuss code reconstruction

2019-04-30 Meeting notes

Attendees Christian Ovesen Kenneth Jacobsen Kenneth Andre Olsen Robin T. Bye Anete Vagale

Goals Individual update on past weeks progress Update on equipment Demo current iteration

Comments Prepare demo. Create commands, with feedback from the audience. Create a short video showing what our program can do Create a longer video showing more detalied version of what is done. Give fully connected network possibility to use a higher number of steps. Possibly give GA lower number of parameters, in order to shorten the computational time. Test some parameters without using GA. Use one GA per type of network.

C Dropbox

https://www.dropbox.com/sh/ohre7r5yj85wkuc/AACFrU9ZUr9OP93vpFqjhtF0a?dl=0

D Source code

https://github.com/KennethJacobsen/EEGANN-NTNU

Bibliography

[1] 7.5.1 Lock Objects. URL https://docs.python.org/2.0/lib/lock-objects.html.

[2] EMOTIV EPOC+ 14 Channel Mobile EEG - Emotiv. URL https://www.emotiv.com/product/emotiv-epoc-14-channel-mobile-eeg/#tab-description.

[3] NVIDIA Developer. URL https://developer.nvidia.com/.

[4] queue — A synchronized queue class — Python 3.7.3 documentation. URL https://docs.python.org/3/library/queue.html.

[5] Reentrancy and Thread-Safety | Qt 5.12. URL https://doc.qt.io/qt-5/threads-reentrancy.html.

[6] Simulator for industrial robots and offline programming - RoboDK. URL https://robodk.com/?gclid=CjwKCAjw_YPnBRBREiwAIP6TJ26_sMDec9MfZlERUmNJ-3vvlikYRsnGImWnev4HLaRT6ag-efkbxhoC8HoQAvD_BwE.

[7] Under The Hood of Neural Networks. Part 1: Fully Connected., . URL https://towardsdatascience.com/under-the-hood-of-neural-networks-part-1-fully-connected-522

[8] Understanding architecture of LSTM cell from scratch with code., . URL https://hackernoon.com/understanding-architecture-of-lstm-cell-from-scratch-with-code-8da40

[9] Noor Kamal Al-Qazzaz, Sawal Hamid Bin Mohd Ali, Siti Anom Ahmad, Mohd Shabiul Islam, and Javier Escudero. Selection of Mother Wavelet Functions for Multi-Channel EEG Signal Analysis during a Working Memory Task. *Sensors (Basel, Switzerland)*, 15(11):29015, 2015. doi: 10.3390/S151129015. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4701319/.

[10] S Ben-Yacoub, B Fasel, and J Ll. Fast Face Detection using MLP and FFT. Technical report. URL https://infoscience.epfl.ch/record/82563/files/avbpa_face.pdf.

[11] Christopher M. Bishop. *Neural networks for pattern recognition.* Clarendon Press, 1995. ISBN 0198538642. URL https://books.google.no/books?hl=en&lr=&id=T0S0BgAAQBAJ&oi=fnd&pg=PP1&dq=neural+network+pattern+recognition&ots=jM7_tN2vql&sig=NWzy8WJoRrLRIR12feCqHpeZQH0&redir_esc=y#v=onepage&q=neural%20network%20pattern%20recognition&f=false.

[12] P.R. Davidson, R.D. Jones, and M.T.R. Peiris. Detecting Behavioral Microsleeps using EEG and LSTM Recurrent Neural Networks. In *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, pages 5754–5757. IEEE, 2005. ISBN 0-7803-8741-4. doi: 10.1109/IEMBS.2005.1615795. URL http://ieeexplore.ieee.org/document/1615795/.

[13] E W Dijkstra, ; L Lamport, ; S Owicki, and D Gries. Verifying properties of parallel programs: an axiomatic approach. Technical Report 9, 1979. URL https://www.microsoft.com/en-us/research/uploads/prod/2016/12/How-to-Make-a-Multiprocessor-Computer-That-Correctly-Executes-Multiprocess-Programs pdf.

[14] Matthieu Duvinage, Thierry Castermans, Mathieu Petieau, Thomas Hoellinger, Guy Cheron, and Thierry Dutoit. Performance of the Emotiv Epoc headset for P300-based applications. *BioMedical Engineering OnLine*, 12(1):56, 2013. ISSN 1475-925X. doi: 10.1186/1475-925X-12-56. URL http://biomedical-engineering-online.biomedcentral.com/articles/10.1186/1475-925X-12-56.

[15] Fredric J Harris. On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform. Technical Report 6, 1978. URL https://www.ak.tu-berlin.de/fileadmin/a0135/Unterrichtsmaterial/EDS_WS0910/Harris_1978_On_the_use_of_windows_for_harmonic_analysis_with_the_discrete_fourier_transform_IEEE.pdf.

[16] Neep Hazarika, Jean Zhu Chen, Ah Chung Tsoi, and Alex Sergejew. Classification of EEG signals using the wavelet transform. *Signal Processing*, 59(1):61–72, 5 1997. ISSN 0165-1684.

doi: 10.1016/S0165-1684(97)00038-8. URL https://www.sciencedirect.com/science/article/pii/S0165168497000388.

[17] Rolf-Magnus Hjørungdal, Filippo Sanfilippo, Ottar L Osen, Adrian Rutle, and Robin T Bye. A Game-based Learning Framework for Controlling Brain-Actuated Wheelchairs. Technical report. URL http://www.robinbye.com/files/publications/ECMS2016Wheelchair.pdf?fbclid=IwAR1w8Iz2LyVJn_L4ZMEMdbFKMDJ12sVx91AY3vuZw7oW_YVqQakFps9o-og.

[18] Intel. Intel Hyper-Threading Technology. URL https://web.archive.org/web/20100821074918/http://cache-www.intel.com/cd/00/00/01/77/17705_htt_user_guide.pdf.

[19] Anders Krogh. Neural Network Ensembles, Cross Validation, and Active Learning. Technical report. URL http://papers.nips.cc/paper/1001-neural-network-ensembles-cross-validation-and-active-learning.pdf.

[20] Ilya Kuzovkin, Raul Vicente, Mathilde Petton, Jean-Philippe Lachaux, Monica Baciu, Philippe Kahane, Sylvain Rheims, Juan R. Vidal, and Jaan Aru. Activations of deep convolutional neural networks are aligned with gamma band activity of human visual cortex. *Communications Biology*, 1(1):107, 12 2018. ISSN 2399-3642. doi: 10.1038/s42003-018-0110-y. URL http://www.nature.com/articles/s42003-018-0110-y.

[21] Melanie (Computer scientist) Mitchell. *An introduction to genetic algorithms*. MIT Press, 1996. ISBN 9780262133166.

[22] S C Nayak, B B Misra, and H S Behera. Impact of Data Normalization on Stock Index Forecasting. Technical report, 2014. URL www.mirlabs.net/ijcisim/index.html.

[23] Hasan Ocak. Automatic detection of epileptic seizures in EEG using discrete wavelet transform and approximate entropy. *Expert Systems with Applications*, 36(2):2027–2036, 3 2009. ISSN 0957-4174. doi: 10.1016/J.ESWA.2007.12.065. URL https://www.sciencedirect.com/science/article/pii/S0957417407006203.

[24] Has¸im Has¸im Sak, Andrew Senior, and Beaufays Google. Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling. Technical report. URL https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43905.pdf.

[25] Bikesh Kumar Singh, N I T Raipur, Kesari Verma, and A S Thoke. Investigations on Impact of Feature Normalization Techniques on Classifier's Performance in Breast Tumor Classification. Technical Report 19, 2015. URL https://pdfs.semanticscholar.org/0d27/0b5a4db360aaf5e862f8b0a734c1b9ff4bd0.pdf.

[26] Marcel van Gerven and Sander Bohte. Editorial: Artificial Neural Networks as Models of Neural Information Processing. *Frontiers in Computational Neuroscience*, 11, 12 2017. ISSN 1662-5188. doi: 10.3389/fncom.2017.00114. URL http://journal.frontiersin.org/article/10.3389/fncom.2017.00114/full.

[27] Tom Verplaetse, Filippo Sanfilippo, Adrian Rutle, Ottar L Osen, and Robin T Bye. On Usage of EEG Brain Control for Rehabilitation of Stroke Patients. Technical report. URL http://www.robinbye.com/files/publications/ECMS2016Stroke.pdf?fbclid=IwAR1ywPfLU2Gk9WYxL3XHxCqLP38xKtAxdYCFAL5ba4MiecCt_nMAqWlvJ-s.

[28] C I Watson, G T Candela, and P J Grother. Comparison of FFT Fingerprint Filtering Methods for Neural Network Classification. Technical report, 1994. URL https://pdfs.semanticscholar.org/a9c4/ba4f7af2b9336a7fb2193d50566ba0f71b2f.pdf.

[29] Eric W. Weisstein. *CRC concise encyclopedia of mathematics.* Chapman & Hall/CRC, 2003. ISBN 1584883472. URL https://books.google.no/books?id=Tm_1ngEACAAJ&dq=isbn:9781584883470&hl=en&sa=X&ved=0ahUKEwiAqIy4toniAhUl2aYKHbekBGMQ6AEIKTAA.