



VRIJE  
UNIVERSITEIT  
BRUSSEL



Thesis submitted in partial fulfilment of the requirements for the degree of Master of Science in de Ingenieurswetenschappen: Computerwetenschappen

## EEGPLUS

# Making a Hybrid Brain Computer Interface Using EEG and EMG Signals

Arnau Dillen

2017 - 2018

Supervisor: Prof. Dr. Ann Nowé

Advisor: Felipe Gomez Marulanda

**Sciences & Bioengineering Sciences**





VRIJE  
UNIVERSITEIT  
BRUSSEL



Proef ingediend met het oog op het behalen van de graad van  
Master of Science in de Ingenieurswetenschappen: Computer-  
wetenschappen

## EEGPLUS

Making a Hybrid Brain Computer  
Interface Using EEG and EMG  
Signals

Arnau Dillen

2017 - 2018

Promotor: Prof. Dr. Ann Nowé

Begeleider: Felipe Gomez Marulanda

**Wetenschappen & Bio-ingeniourswetenschappen**



## Abstract

In this thesis we will discuss machine learning (ML) techniques used to interpret noisy biological signals, such as the electroencephalogram (EEG) and electromyogram (EMG). Firstly, An overview on biological signals in the human body is given. Biological background, types of signals and processing techniques will be discussed. Secondly, after giving some general background on ML we discuss existing Brain Computer Interface (BCI) techniques that use biosignals to interact with devices. Several machine learning techniques that have successfully been used for the purpose of classifying biological signals in a BCI setting are discussed. An overview of well known techniques is given and the state-of-the-art is presented. An introduction to hybrid BCI is additionally given together with existing approaches from the literature. Following this overview of literature and background, the proposed hybrid BCI EEGPlus model is detailed. The implemented hybrid model takes EMG inputs in addition to EEG with the purpose of improving exclusive EEG classification. We discuss design alternatives to achieve such a model and motivate the finale model choice. The implementation of this model is then presented together with a conceptual design for an online BCI system that uses this model. Finally, the model is validated by comparing it with the Shallow ConvNet variant from Schirrmeister et al. (2017) on a number of accuracy metrics. No improvement could be achieved in comparison with the state-of-the art. However, we believe that our approach may lead to improvements in the future. This study could be considered as a pilot study to achieving the goals that where established for the EEGPlus model.



## **Acknowledgements**

I would like to thank my thesis supervisor Prof. Dr. Ann Nowé for giving me the opportunity to write this thesis about an interesting subject. Many thanks to my advisor Felipe Gomez Marulanda for helping me a lot with the development and implementation of my experiments, and for advising me on all aspects of writing this thesis. I would additionally like to thank the people at the VUB AI lab for giving about many things, and my student colleagues for providing me with good ideas. Finally, I would like to thank my friends and family for supporting me during the year.



# Contents

<b>List of Figures</b>	<b>iii</b>
<b>List of Tables</b>	<b>iv</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Biological Signals</b>	<b>5</b>
2.1 The Human Nervous System . . . . .	5
2.1.1 Cells in the Brain . . . . .	5
2.1.2 Biological Neural Networks . . . . .	8
2.2 Signals of the Human Body . . . . .	9
2.2.1 Types of Signals . . . . .	10
2.2.1.1 Electroencephalogram . . . . .	10
2.2.1.2 Other Important Signals . . . . .	12
2.2.2 Signal Processing . . . . .	14
<b>3 Machine Learning</b>	<b>19</b>
3.1 Background . . . . .	19
3.1.1 General Overview . . . . .	20
3.1.2 Artificial Neural Networks . . . . .	22
3.1.3 Deep Learning . . . . .	24
3.2 ML for BCI . . . . .	25
3.2.1 The BCI Pipeline . . . . .	25
3.2.2 Classification . . . . .	27
3.3 Hybrid BCI . . . . .	29
<b>4 Design &amp; Architecture</b>	<b>31</b>
4.1 Motivation . . . . .	31
4.2 Design . . . . .	33
4.2.1 Implementation . . . . .	33
4.2.1.1 EEG Feature & Classification Modules . . . . .	35
4.2.1.2 EMG Generator Module . . . . .	37

4.2.1.3	Loss Function Module . . . . .	38
4.2.2	Alternatives . . . . .	40
4.3	Conceptual Online System . . . . .	42
<b>5</b>	<b>Experimental Verification</b>	<b>45</b>
5.1	Experiments . . . . .	45
5.1.1	Data . . . . .	45
5.1.2	Methods . . . . .	48
5.2	Results . . . . .	51
<b>6</b>	<b>Analysis &amp; Conclusion</b>	<b>57</b>
6.1	Hypothesis 1: Data . . . . .	57
6.2	Hypothesis 2: Cost Function (optimization) . . . . .	58
6.3	Hypothesis 3: EMG Generator . . . . .	59
6.4	Hypothesis 4: EEG-EMG Delay . . . . .	59
6.5	Conclusion . . . . .	60
<b>Appendices</b>		<b>77</b>
<b>A</b>	<b>Additional figures and tables</b>	<b>78</b>

# List of Figures

2.1	Multipolar neuron . . . . .	6
2.2	Brain areas . . . . .	8
2.3	EEG electrode 10-20 system . . . . .	10
2.4	Stationary vs Non-stationary time-series . . . . .	17
3.1	Artificial neural network . . . . .	23
3.2	BCI fusion pipeline . . . . .	29
4.1	Overview of the implemented neural network architecture . . . . .	34
4.2	Overview of a conceptual hybrid NN architecture . . . . .	41
5.1	Loss value on the test set during training . . . . .	51
5.2	Misclassification rate during training . . . . .	53
5.3	Cohen's Kappa and $F_1$ -score on the test set during training . . . . .	54
5.4	Confusion matrices of evaluation . . . . .	56
6.1	Confusion matrices of evaluation . . . . .	58
6.2	Back-propagation of gradients in EEGPlus . . . . .	59
6.3	This figure shows the delay constrain between EEG and EMG . . . . .	60
A.1	Loss value on the training set during training . . . . .	78
A.2	Kappa value on the training set during training . . . . .	79
A.3	F-score value on the training set during training . . . . .	79
A.4	Loss value on the validation set during training . . . . .	80
A.5	Misclassification rate on the validation set during training . . . . .	80
A.6	Kappa value on the validation set during training . . . . .	81
A.7	F-score value on the validation set during training . . . . .	81
A.8	Averaged confusion matrix on test set at epoch 10 for EEGPlus . .	82
A.9	Averaged confusion matrix on test set at epoch 10 for ShallowEEG	82
A.10	Averaged confusion matrix on test set at epoch 25 for EEGPlus . .	83
A.11	Averaged confusion matrix on test set at epoch 25 for ShallowEEG	83
A.12	Averaged confusion matrix on test set at last epoch for EEGPlus . .	84
A.13	Averaged confusion matrix on test set at last epoch for ShallowEEG	84

# List of Tables

4.1	Feature module layers . . . . .	35
4.2	Classification module layers . . . . .	35
4.3	Generator module layers . . . . .	38
5.1	Model evaluation performance comparison . . . . .	55

# Chapter 1

## Introduction

In the fast evolving world of technology there is an increasing demand for novel interaction methods, as the current modalities are becoming dated and less appropriate for emerging technologies. For example, Augmented and Virtual Reality devices can not be used effectively with classic mouse and keyboard interactions. New devices and applications require novel ways of interaction. One alternative approach to human computer interaction (HCI) involves using biological signals, also called biosignals, as a means of interfacing with a device. One of the most widely used biosignals for this type of application is named Electroencephalogram (EEG). This signal measures the potential differences related to brain activity by placing electrodes on the scalp of a person. Such a signal falls in the category of electrical biosignals, alternatively named bioelectrical time signals, that measure changes in electrical current caused by electrical potential differences across a certain biological tissue. Other well-known electrical biosignals are Electrocardiogram (ECG) that measures potential differences for the heart, Electromyogram (EMG) for muscle activity and Electrooculogram (EOG) that measures eye activity. There are several other signals of this type, but these are among the best-known and most widely studied.

Alternatively to EEG one can also use the Magnetoencephalogram (MEG) that measures the magnetic field induced by electrical currents. However, non-electrical brain measurements such as magnetic resonance imagining (MRI) and MEG are generally not suitable for wearable robotics due to their non-portability. Current research is attempting to solve this issue. In recent years there have been promising technological advances that should make wearable systems practical and feasible (Boto et al., 2018). These future technologies could then be used alternatively or in conjunction with electrical biosignals to create more robust measurements. Our own research could also benefit from these advances.

When interfacing with a device through sensor measurements of biosignals it is referred to as a Brain Computer Interface (BCI). The idea of BCI has been around

for a long time (Vidal, 1973) and has been extensively studied since then (Subha et al., 2010) in the context of the study of diseases, sleep disorders and several other areas of research. Any discipline where automated detection of a certain physical characteristic of a subject can make use of BCI techniques.

Interaction by means of BCI can not only be used to complement other input modalities, but also allows people with physical disabilities to interact with a computer, even when fully paralyzed. There always is a certain degree of brain activity to be measured as long as the subject is alive, and conscious, thoughts should always be interpretable in theory. BCI also allows for a natural interaction with devices that attempt to reproduce a certain human behaviour, such as a prosthesis that replaces a lost limb for example. Advanced enough prostheses that uses BCI for control can be made to feel as an equivalent, if not better, replacement for the missing limb, assuming that we are able to perfectly interpret a chosen biosignal. However, biological signals are generally measured with substantial noise due to the interference generated by the devices used to measure the signals, or by the interference of other signals that can be found in the body such as signals originating from the heart, lungs and eyes. Although current BCI research has made significant advances, there still are some major challenges left before it becomes a feasible alternative to classic input modalities. Currently one of the main issues to practical use of BCI is that we do not yet entirely understand the workings of the human nervous system. We already know a great deal about the workings of the Central and Peripheral Nervous System (CNS, PNS), such as the exact workings of individual neurons for example. However, how exactly the global high-level interactions between the different parts of the complex human nervous system result in cognition is unknown as of yet. Mostly, not much is known and understood regarding the neuronal networks within the brain.

Another issue with BCI lies in the signal quality of biosignals. There exist two techniques to measure biosignals. These are invasive and non-invasive, with the latter requiring an operation to place electrodes in the brain. Invasive procedures yield better signal quality in comparison with non-invasive methods. This is due to the fact that this technique can target specific parts of the brain directly. The choice of measurement is additionally often accompanied by cost considerations. For example when measuring EEG signals one can choose to use a relatively affordable and non-invasive commercial device that can easily be put on or off without any significant required preparation, such as the OpenBCI headset<sup>1</sup>. However, this comes at the cost of noisy and hard to interpret measurements. In contrast to this, a medical system as used in hospitals, which is less practical, but provides us with a higher signal-to-noise ratio (SNR)<sup>2</sup> and is still non-invasive can be used. Another

---

<sup>1</sup><http://openbci.com/>

<sup>2</sup>Signal-to-noise ratio expresses the quality of a signal, high values are preferred over low ones.

alternative could be to use Electrocorticography (ECoG) for measurements, which is essentially the same as EEG but requires an operation to place electrodes on the surface of the brain. This approach provides a better SNR and measures local brain activity much more precisely. However, this requires an expensive, risky and invasive operation to achieve.

Due to these disadvantages, BCI has not widely been used in practice for HCI. Nevertheless, it is extensively being employed in many different types of research such as diagnosis of Alzheimer's disease (Dauwels et al., 2010) and analysis of sleep (Campbell, 2009). Thanks to recent advancements in such research and the fast evolution of artificial intelligence, usable and inexpensive BCI systems are starting to be used in the real world. Examples are the previously mentioned OpenBCI headset and the solutions provided by Emotiv<sup>3</sup> for EEG. Regarding EMG, the Myo armband<sup>4</sup> and the CTRL-labs armband<sup>5</sup> are among the available commercial devices. Note that OpenBCI can also be used for EMG measurements in addition to EEG.

One of the bigger issues of BCI when using non-invasive EEG signals is that EEG signals have a low SNR. This means that if we learn directly on the EEG signals, important information that is too convoluted with noise will be lost. One solution is to add extra signals that have higher SNR so they can help the model in identifying the missing information that is hidden within the noise. Such a system is called a hybrid BCI system and has been met with a certain degree of success (Novak & Riener, 2015). However, this comes with a drawback. In such a system, the additional signal sources can not be decoupled from the model. As a result, this solution can not be used in certain real world applications where all input signals are not always available. Therefore we decided to investigate a novel manner of creating such a hybrid BCI system.

In this thesis we investigate creating an unconditioned hybrid BCI system that relies solely on EEG as the primary input to make decisions, but can optionally use information from other relevant sensor inputs to increase the model accuracy. We hypothesize that initially joining the signals in the training phase will push the model to find correlations between EEG signals and the secondary signals. This means that the hybrid EEG model may find patterns that could not be discovered using EEG only. Subsequently, we claim that the hybrid EEG model can be decoupled from the other signals to be used in moments where any of the secondary signals are not available. Our experimental setup combines EEG with Electromyography (EMG) to validate this hypothesis.

EMG was chosen as secondary input as it is also frequently used in BCI, es-

---

<sup>3</sup><https://www.emotiv.com/>

<sup>4</sup><https://www.myo.com/>

<sup>5</sup><https://www.ctrl-labs.com/>

pecially for control strategies in prosthetic devices (Alte, 2017). An additional interesting advantage of the performed experiments is that learned lessons from such a model could be used to further or facilitate current research on the matter of motor imagery movements. Motor imagery (MI) is the study of identifying the patterns that occur in measured signals of the brain when a subject thinks of a movement without executing it. More on this will be detailed in subsequent chapters.

The experimental model was implemented in Python and compared with a state-of-the art model in EEG-based BCI model (Schirrmeyer et al., 2017). The main goal of these experiments is to verify the feasibility and usefulness of such a hybrid system in comparison to unimodal systems that learn solely from EEG. We also want to verify whether the assumptions that were made about signal processing, such as the delay between an observed EEG pattern and corresponding EMG activity, are correct. If our hypothesis is confirmed, this approach will solve several issues of BCI for real world applications.

Before the proposed model is presented, a theoretical background will be given to clarify the significance and implications of our methods. In chapter 2 a neurological background related to biological signals together with some relevant theory on signal processing will be discussed. This chapter should provide the necessary information about the complexity of the human systems and serve as an illustration of the importance of certain abstractions and assumptions that had to be made during this research.

In chapter 3 an overview of Machine Learning (ML) related background that is relevant to the subject at hand will be given. Additionally, a full literature overview of ML approaches in BCI will be provided. We will mostly focus on state-of-the-art systems that have been suggested in recent years and classic techniques that have been proven to be effective.

Chapter 4 will elaborate on the proposed hybrid BCI system. Here we look at the design decisions that were made regarding the final system. Furthermore, a potential online application of such a system and how this could be implemented with current technology is proposed. Consecutively, in chapter 5, the experimental verification procedure that was applied for evaluating our proposed system in comparison with the state-of-the-art will be discussed.

Finally, chapter 6 will be a discussion and conclusion regarding our system in light of the experimental results and previously presented implications. A proposal for future work regarding not only our own system but also BCI research in general is also included in this section. The purpose is that this thesis could then serve as a basis for future BCI research in this domain.

# Chapter 2

## Biological Signals

In this chapter we will discuss in more detail how biological signals in our body are generated and how they play an essential role in cognition. Furthermore, we will look at the biological processes that initiate and propagate electrical signals, and at the different types of signals that can be measured. Finally, an overview will be given of how to interpret and (pre-)process biological signals in a scientific manner. This knowledge will serve as the basis for the ML techniques that are discussed discussed in chapter 3.

### 2.1 The Human Nervous System

The nervous system in vertebrates consists of two main components, the Central Nervous System (CNS) and the Peripheral Nervous System (PNS). The CNS is composed of the brain and spine, and fulfills the functions of control center and sensory processing. The PNS is mainly made up of nerves connecting the CNS to the rest of the body and is responsible for transporting signals to and from the CNS. A nerve that transports a signal from the brain is called an *efferent* or *motor* nerve as it serves to send motor commands to the muscles of the body. If a nerve transmits signals to the CNS it is called *afferent* or *sensory*, for it sends sensory input to the CNS for processing.

#### 2.1.1 Cells in the Brain

The brain is composed of two main types of cells, which are neurons and glial cells. In this section we discuss these different types of cells and their participation in cognition.

Neurons are the primary cells of the brain. They are interconnected and send electrical signals to each other, forming a biological neural network, which we will

discuss in the next section. To form connections, neurons have two extensions of the cell body: axons and dendrites. A neuron can have only one axon, but it can have none as well. In contrast, the axons themselves can branch multiple times. A neuron can have multiple dendrites which can form connections with the axons of several other neurons. There are, however, exceptions to this with axons or dendrites forming respective connections. Most neurons send signals from axons and receive them from the dendrites. Figure 2.1 shows a typical multipolar neuron with one axon and multiple dendrites.

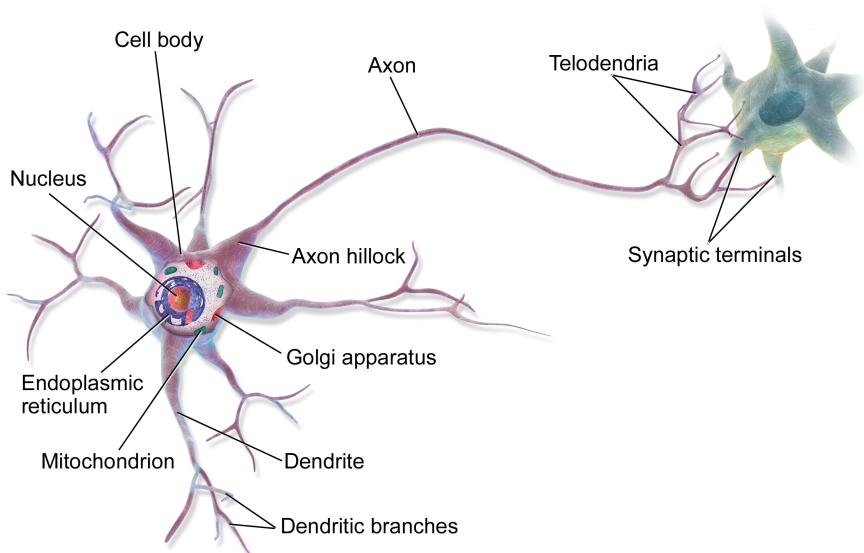


Figure 2.1: A multipolar neuron<sup>1</sup>

The connections between neurons are called synapses. The initiating neuron experiences bio-chemical changes due to an electric signal reaching the synapse, which is referred to as a pre-synaptic stimulus. The chemicals involved in this stimulus are named neurotransmitters. These pre-synaptic changes result in a post-synaptic response which is referred to as depolarization. Due to the opening of certain channels in the membrane, an influx of positively charged ions through voltage-gated ion channels, mostly Sodium and Potassium, causes the potential difference between the inside and outside of the cell to be inverted, i.e., the inside goes from more negative to more positive than the outside. If the strength of this depolarization crosses a certain threshold an Action Potential (AP) is caused and we say that the neuron "fires". Depolarization then travels along the axons of the neuron until it reaches a synapse, where a new pre-synaptic stimulus

---

<sup>1</sup>By BruceBlaus - Own work, CC BY 3.0, <https://commons.wikimedia.org/w/index.php?curid=28761830>

is caused. Consequently, this reaction causes depolarization in other connected neurons, which results in propagation of the AP to connected neurons, initiating a chain reaction to neighbouring neurons. After depolarization, the membrane potential of the neuron returns to the resting state by opening channels that allow positively charged potassium to flow out of the cell. The inside of the cell reverts to being more negatively charged than the outside. This process is called repolarization.

The second type of cells in the brain are glial cells. There are several different types of glial cells, each with their respective behaviour. A single glial cell can consequently fulfill one or more functions. Currently, four main neurological functions have been attributed to glial cells:

1. Surround neurons and hold them in place as scaffolding.
2. Supply nutrients and oxygen to neurons.
3. Insulate neurons from each other (To avoid leaking action potentials resulting in unwanted polarization).
4. Phagocytosis, i.e., disposition of pathogens and biological debris.

Glial cells are also referred to as the glue of the brain and play an important role in the blood-brain barrier separating neurons from the blood stream. One of the most important types of glial cells in the brain are Oligodendrocytes which insulate the axons of neurons by forming a so-called myelin (Hartline, 2008) sheet around them. Next to insulation this has the important role of boosting the electrical signal along the axons. This property is essential for AP's to be propagated over large distances. Loss of myelin is often a result of some neurobiological diseases such as Multiple Sclerosis (MS), which causes loss of certain cognitive functions because AP's can not reach the end of an axon.

The cognitive functions of a person, or an animal, are a result of neurons forming structured networks. The brain consists of multiple interconnected networks which can be assigned certain cognitive functions. Figure 2.2 shows an overview of the different regions of the brain which contain the networks that are responsible for certain functions.

---

<sup>2</sup>Blausen.com staff (2014). "Medical gallery of Blausen Medical 2014". WikiJournal of Medicine 1 (2). DOI:10.15347/wjm/2014.010. ISSN 2002-4436. - Own work, CC BY 3.0, <https://commons.wikimedia.org/w/index.php?curid=60100749>

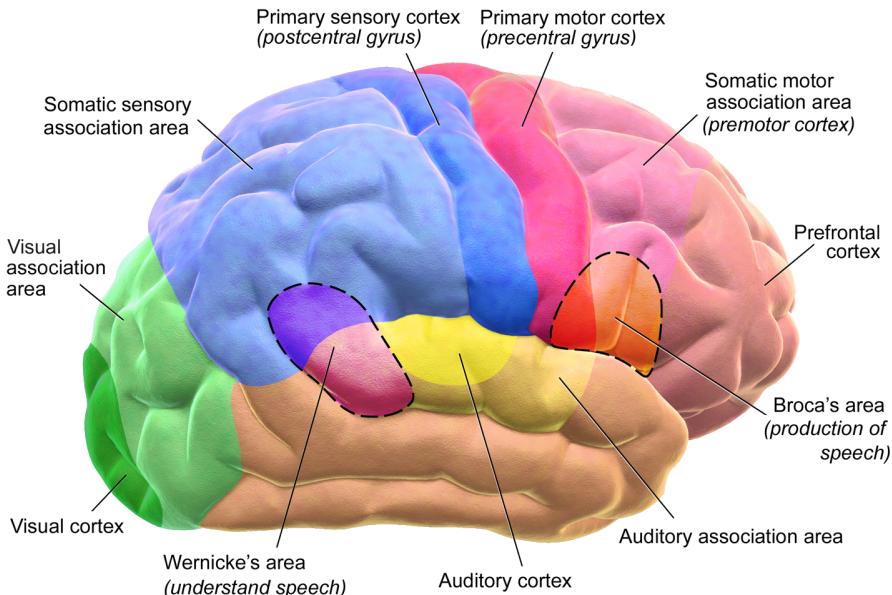


Figure 2.2: Brain areas and their functions<sup>2</sup>

### 2.1.2 Biological Neural Networks

Understanding the structure of the neural networks in our brain can give insights into functions associated with certain areas of the brain and into mechanisms of learning. We will now briefly discuss how these biological processes are related to Machine Learning, which is discussed in the next chapter.

How cognition is achieved exactly through the interaction of connected neurons and propagation of action potentials is not yet understood. Many studies are being performed to identify which functions and behaviours are associated with which regions of the brain. This aspect and the local interactions that were previously described are well-known. Currently most research is focused on identifying how these local behaviours result in high-level mechanisms of cognition through neural networks.

Biological neural networks are mostly studied with simulations that model the well-known local behaviours. Such simulations have become computationally tractable for large-scale networks that model certain regions of the brain in recent years. Example frameworks that simulate realistic neuron interaction are NEURON (Hines & Carnevale, 1997), GENESIS (Bower et al., 2003) and MOOSE (Ray et al., 2008). Gaining an insight into these mechanisms should allow for a better understanding of cognition and the human mind. However, in many cases a simplified model is preferred as it abstracts away from some complex systems that might not be relevant to the research at hand.

One such model is the artificial neural network (ANN) that is used in the context of machine learning. The precise workings of an ANN will be discussed in section 3.1.2. This model first introduced by Walter Pitts and Warren S. McCulloch (McCulloch & Pitts, 1943) is currently one of the state-of-the-art machine learning methods. Here the biological mechanisms of signal propagation are reduced to mathematical operations. However, due to the fact that the mechanisms on which they are based are currently not well-known, it is currently not fully understood why ANN work as well as they do. While many innovations to ANN were inspired by biological equivalents, other aspects, such as exact computations are not biologically plausible. In contrast, it is possible that identifying biological alternatives to mechanisms that are used in ANN could contribute to better understanding the human brain.

Studying signals of the brain is a different way of identifying mechanism that result in cognition. This approach has been used to identify the different areas responsible for certain functions of the brain. Observed behaviour in measured biological signals could be used to construct simulations with one of the previously mentioned models and discover the mechanisms that results in a certain cognitive function. These findings might in turn be useful to ANN research. The next section gives an overview of the best-known signals of the human body that are used in studies.

## 2.2 Signals of the Human Body

A biosignal refers to any signal generated by living beings, that can be continuously measured and associated with a certain function in the body. The generated continuous signal is usually sampled at a certain frequency and studied in a discretized digital form. We will look at the different types of signals that occur and can be measured in the human body. Biosignals are categorized in non-invasive and invasive types, with invasive types requiring an operation to install measurement apparatus. Non-invasive signals are the main focus of this section. Therefore, all discussed signals are of the non-invasive type unless otherwise specified. Biosignals are most often electrical signals also called bioelectrical signals, as these can most easily be measured and are strongly related to the electrical signals of the nervous system. Therefore, we will mainly discuss the most common ones that are used in most research and BCI applications(Scherer & Rao, 2011).

## 2.2.1 Types of Signals

### 2.2.1.1 Electroencephalogram

There are several types of signals generated by the human body that can be measured and interpreted for BCI applications. One of the most commonly studied biosignals is the Electroencephalogram (EEG) (Schomer & Lopes da Silva, 2018; Blankertz et al., 2006; Millan & Mourino, 2003; Pfurtscheller et al., 2006). This signal is acquired by placing electrodes on the surface of the scalp and measuring the difference in electric potential between a certain electrode and another reference electrode. In essence this measures the AP's that occur in neurons underneath an electrode. By placing electrodes over appropriate regions of the brain, which are shown in figure 2.2, we can measure brain activity related to a specific function. The popularity of EEG compared to other methods is primarily due to its relative low expense, non-invasiveness and portability. This popularity also makes EEG more attractive due to the wide availability of data.

Placement of electrodes usually follows the international 10-20 system (Towle et al., 1993), which is shown in figure 2.3.

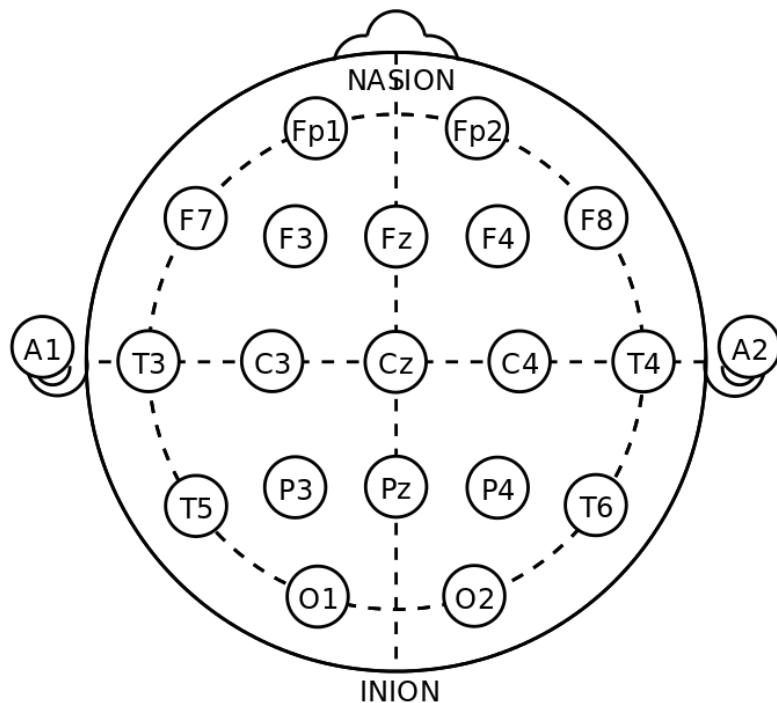


Figure 2.3: International 10-20 system for EEG <sup>3</sup>

---

<sup>3</sup>By トマトン124 (talk) - Own work, Public Domain, <https://commons.wikimedia.org/w/index.php?curid=10489987>

This placement of electrodes allows reproduction and comparison of data that is acquired for EEG studies. The electrode labels correspond to the related lobe of the brain for which electrical activity is measured. This also serves as a guide for where to place electrodes depending on the function of the brain that is of interest. A higher resolution variant of the system (Oostenveld & Praamstra, 2001) exist with more electrode positions and more precise placement. This variant is most often used in modern research.

Two types of electrodes are used for measurement of EEG activity. Wet electrodes are the most common and require the application of electrolytic gel on the surface of the scalp for ideal conductivity. Affixing wet electrodes takes great care and a long time, requiring regular re-application of gel. This makes them only really useful in a lab setting and impractical for BCI systems. Alternatively, dry electrodes exist that can be installed in a headset, such as the previously mentioned OpenBCI headset. This make them into more suitable candidates for use in BCI systems. However, dry electrodes have inferior signal quality in comparison to their wet counterparts. Development of better dry electrodes is currently one of the main concerns of BCI related research.

EEG signal amplitudes are usually within the range of  $\pm 100\mu V$ , while the time resolution is in the range of milliseconds. This means that we need to use a high sample frequency when measuring the continuous signal and discretizing it into a sequence of measurements. The spatial resolution of EEG is around  $5cm^2$ , meaning that we only have a high-level view of the measured brain activity and cannot accurately determine the location of the neurons generating the signal. The poor spatial resolution is mostly due to several layers (meninges, cerebrospinal fluid, skull, scalp) between the source of the signal and the sensor used to measure said signal. The small amplitude of EEG signals also results in a high sensitivity to noise. Interference from other sources of electric potential, such as eye movement or electrical equipment, results in what is called artifacts. These are signals not originating from the brain that are also present in EEG measurements. Therefore, it is required to pre-process a signal for it to be of any use for analysis or classification. Section 2.2.2 will discuss on (pre-)processing of signals.

To achieve better signal quality it is possible to acquire the electrical signals from the brain by implanting electrodes directly in the brain. Electrodes are then placed on the surface of the cortex (electrocorticogram, ECoG) (Miller et al., 2007; Scherer et al., 2009) or within the cortex (intracortical single unit or multi unit recordings) (Maynard et al., 1997; Hochberg et al., 2006). The major drawback of such a system however, is that a risky and costly operation is required to implant the electrodes. Such a system is said to be invasive as opposed to the non-invasive EEG.

The type of activity observed in the brain can be linked to cognitive func-

tion. These types of signals are called event-related potentials (ERPs) and event-related changes in the continuous oscillatory activity of EEG. ERPs are changes in electrical potential that occur within an approximately constant time interval of perceptual, cognitive or motor events. They are therefore used in certain BCI systems. However, a high signal-to-noise ratio (SNR) is required for humans to be able to distinguish them with the naked eye. In studies, this is usually achieved by averaging several trials that are known to contain the pattern that one wants to observe. There are multiple types of ERPs that are used for BCI systems of which an overview is given in Scherer & Rao (2011). One of the most widely used types is the P300 (Donchin et al., 2000) which represents a positive shift in the EEG signal, 300ms after a stimulus is presented.

The second type of activity, changes in oscillatory EEG activity, can be categorized in event-related desynchronization (ERD) and event-related synchronization (ERS) (Pfurtscheller & Lopes da Silva, 1999). ERD corresponds to an attenuation of signal amplitude in certain frequency bands of the signal while ERS refers to an increased amplitude. These changes are induced by the reaction of a subject to an internal or external event and are therefore referred to as induced event-related synchronizations and desynchronizations (ERDS). The most commonly used type of ERDS in BCI is motor imagery (MI), which is due to the kinesthetic imagination of movements. MI corresponds to a subject thinking of producing a movement without executing it. The frequency bands at which MI occurs are "mu" (7-13 Hz) and "central beta" (13-30 Hz).

#### 2.2.1.2 Other Important Signals

One of the alternatives to EEG is the Magnetoencephalogram (MEG) which measures magnetic fields associated with neuronal ionic currents using gradiometers (superconducting quantum interference devices, SQUIDs) in magnetically shielded rooms (Mellinger et al., 2007). This approach can allow for cleaner and more precise measurements. It could also potentially become even more precise and even be used to detect deep cerebral activity occurring in the deepest layers of the brain. However this is not possible yet and there currently are some other major drawbacks compared to EEG. The SQUID approach is not portable and requires a large and cumbersome setup for accurate measurements. Research is currently being performed to make MEG more portable by using spin exchange relaxation-free (SERF) scanners (Sander et al., 2012; Boto et al., 2018).

Neuronal activity can also be measured indirectly by measuring corresponding changes in blood flow and blood oxygenation levels. The blood oxygenation level dependent (BOLD) response is one of the known responses that is measured through functional magnetic resonance imaging (fMRI) (Weiskopf et al., 2004). An alternate method that measures the level of oxyhemoglobin in the blood is near

infrared spectroscopy (NIRS) (Coyle et al., 2007). Both methods have the advantage of being non-invasive, portable and have a high spatial resolution. However, these methods all suffer from a low time resolution, as these changes can only be detected within seconds as opposed to milliseconds for EEG. An important issue with fMRI is that it can interfere with EEG signals if used in conjunction, which is not a problem with NIRS.

Another biosignal associated with neural activity is the galvanic skin response (GSR) also called Electrodermal activity (EDA) (Blain et al., 2008). This signal measures changes in the electrical conductivity of the skin. Mostly used in psychological research, it can be considered as a predecessor to EEG and ECG. Applications to BCI have not been extensively researched yet, but it might prove useful in a hybrid BCI system.

A different bioelectric signal that is often studied is the Electromyogram (EMG) that is related to muscle activity. It is based on the same principles as EEG by placing electrodes on the part of the body where the muscle of interest is located. Due to using the same measuring method as EEG, EMG signals generated by muscles of our face are usually responsible for several of the artifacts in EEG measurements. The properties of the EMG signal can widely differ depending on the muscle that is under observation. The amplitude of a signal can vary from less than  $50\mu V$  up to  $30mV$ , while time and spatial resolution are the same as EEG. Spatial resolution is of lesser importance than with EEG, depending on the size of the measured muscle. An EMG electrode usually measures activity for a single muscle, whereas EEG measures activity of several neurons. This makes EMG more simple to interpret than EEG since the activation of an associated muscle causes a peak in the signal. EEG patterns need to be interpreted and exact neuron activations can not be measured directly. Additionally, the SNR for EMG is higher than EEG since there are less layers of biological tissue between the activated muscle and the measuring electrode. EEG and EMG are occasionally measured together since it can be interesting to study the activity occurring in the brain that is responsible for the activation of certain muscles.

Other well-known bioelectrical signals are the Electrooculogram (EOG) (Frishman, 2013) and Electrocardiogram (ECG) (Cooper, 1986). EOG measures the potential difference between the front and the back of the eye and is mostly used to detect eye movement. ECG is the same as EMG, but for heart muscles. The activity of neurons and muscles can also be measured in different ways. As an alternative to EMG the mechanomyogram (MMG) (Stokes & Blythe, 2001) can also be used. Muscle contractions can be measured from the vibrations caused by the oscillation of muscle fibers. The MMG is also often referred to as phonemyogram, acoustic myogram, sound myogram, vibromyogram or muscle sound.

Finally, it is also possible to monitor mechanical signals, acoustic signals, chem-

ical signals and optical signals. We will not consider the different types of signals in these categories as there currently is limited practical use for these in the context of BCI. However, it could be possible to identify changes in such signals corresponding to neural activity. If these correspondences are known to be caused by certain patterns in neural activity, we could use these for secondary modalities in hybrid BCI.

### 2.2.2 Signal Processing

Usually, typical biological signals that are measured with the previously mentioned techniques have a low SNR. Interference from other sources generates artifacts in the signal. Sources of interference in EEG are often artifacts from eye movement (EOG), facial muscles (EMG) and the power source of measurement equipment. Due to this interference the SNR typically is low, meaning that there is not a lot of information that can be derived from the raw signal. Most of the measurements are too noisy for analysis or use in an ML application. Moreover some information can also be lost due to the discretization of the continuous signal into a digital signal. To address this issue several techniques for processing digital signals exist.

A digital signal can be studied in the following domains: time domain (one-dimensional signals), spatial domain (multidimensional signals), frequency domain, and wavelet domains. The time domain considers the signal as a sequence of discrete measurements over time and is how the signal usually was acquired and stored. Here, we look at changes that occur over time for a single signal. The spatial domain concerns itself with the analysis of several signals acquired from different brain regions. This is the case with EEG for example, where the different electrodes showing activity at the same time indicate possible correlation. Signals are often studied in both time and space, since correlated changes may occur at different places over a period of time. Going from time to spatial domain simply consists of transposing inputs, i.e., switching the x and y axes of the signal graph.

In the frequency domain, the signal is studied with relation to frequencies and what parts of the signal occur within said frequencies. To go from the time domain to the frequency domain of a signal, the Fourier transform is often used. This will decompose the signal into several sine functions, each with their own frequency. This set of sine waves is what is called the frequency spectrum of the signal, which is why frequency-domain analysis is often referred to as spectral analysis. For digital signals this process is more specifically called discrete Fourier transform (DFT) and can be computed with algorithms such as fast Fourier transform (FFT) (Cooley et al., 1969). DFT and FFT are often used interchangeably as DFT usually uses FFT in practice.

The wavelet transform (Rioul & Vetterli, 1991) will represent the time series as a wavelet series. A wavelet is a wave-like oscillation with an amplitude that

begins at zero, increases, and then decreases back to zero. A wavelet series is then the discrete time-series represented as a mathematical function which is generated by the wavelet. This has the advantage of representing both spatial and frequency information of a signal, and being slightly less computationally intensive. The wavelet transform is considered to be quite similar to Fourier transform, as the commonly used Morlet wavelet (Morlet, 1983) is mathematically identical to a short-time Fourier transform using a Gaussian window function (Bruns, 2004).

One of the first and most important aspects of digital signal processing is the sampling frequency at which measurements are sampled and stored. The appropriate sampling rate is usually determined based on the Shannon-Nyquist theorem (Shannon, 1949) that states that if a signal contains a maximal frequency of  $B$  Hertz then  $f_s > 2B$  is the minimal sampling frequency. This frequency is also referred to as the Nyquist frequency. If the sampling rate is below the Nyquist frequency aliasing might occur. Aliasing causes two different signals to be indistinguishable if sampled at the same frequency that is below the Nyquist frequency.

The sequence of all acquired measurements results in a raw time series for the given biosignal with low SNR. Enhancing the SNR of a signal is usually done by applying a brick-wall filter to the raw time series data. Such a filter is an idealized filter that will remove unwanted frequencies from the data that are believed to not belong to the signal at hand. The basis of brick-wall filters are so-called sinc filters, which are idealized filters that perfectly cut off frequency components above a given cutoff frequency. Such a filter will pass frequencies below the cutoff and is therefore referred to as a low-pass filter. Mathematically this means that the impulse response at time  $t$  of such a filter will be given by the normalized sinc function, given in equation 2.1

$$h(t) = \mathcal{F}^{-1}\{H(f)\} = 2B \frac{\sin(2\pi Bt)}{2\pi Bt} = 2Bsinc(2Bt) \quad (2.1)$$

where  $\mathcal{F}^{-1}$  is the inverse Fourier transform and  $B$  is the chosen cutoff frequency. However, an ideal filter is usually not feasible in practice, especially in real-time settings where only a window of samples is available at processing time. This is because the sinc filter has an infinite impulse response (IIR), meaning that the signal will not go back to zero when presented with an impulse. Therefore, an approximation is required that has a finite impulse response (FIR) that goes back to zero after a certain time. In practice this is often achieved by applying a Fourier transform to go to the frequency domain, applying the filter and converting back to the time domain. This approach is usually the fastest with a time complexity of  $O(n \log n)$  in comparison with other methods that have a  $O(n^2)$  complexity, or above.

Other types of brick-wall filters are constructed by using the low-pass filter. The

high-pass filter that cuts off low frequencies is computed by taking the complement of the low-pass filter. The band-pass filter that passes the frequencies within a given frequency range is achieved by subtracting the low-pass response for the maximal frequency from the one for the lower frequency. Other types of filters include the band-stop filter that cuts off frequencies within the given range and the notch-filter that rejects one specific frequency.

Filters can also be classified according to other criteria such as linear or non-linear depending on whether the applied transformation is linear. All previously discussed filters were linear. Another categorization can be causal or non-causal if only past samples are used or future samples are also included respectively. A filter is said to be stable if the produced outputs converge to a constant value with time.

Next to low SNR of a measured signal, a different issue that is also observed in biosignals is the non-stationarity of these signals. A stationary process refers to a stochastic process whose unconditional joint probability distribution does not change over time. Such a process will have the same mean and variance if the distribution is parametric, regardless of the point in time. If a signal is non-stationary it means that the mean, initially being at zero, might increase or decrease for the next sample. The same could be true for the maximal amplitude, or even both. The difference between a stationary time series compared to a non-stationary time series is visualized in figure 2.4.

Due to this property it is not possible to use several statistical techniques that rely on the time series distribution being stationary. The non-stationarity of biosignals also results in measurement distributions being different between recorded subjects and even between separate recording sessions of the same subject. In an online setting this may even be worse as the distribution may change at any time. Methods are necessary to deal with this aspect of biosignals. Subjects could be trained to generate signals that are distributed in a consistent manner, but this would be a cumbersome solution. It can not be expected that people would be prepared to train for this and it might even prove too difficult for humans to consistently produce stationary signals. Methods that do not rely on the recorded subject generating consistently distributed measurements are necessary. Several techniques have been tried to deal with this non-stationarity which will be discussed in section 3.2 of the next chapter. Most techniques are based on clever feature extraction that is robust to non-stationary signals.

A final fact that additionally needs to be taken into account for combined classification is the delay between EEG activity and EMG activations resulting from a movement. The motor commands detected in EEG need to travel to the relevant

---

<sup>4</sup>By Protonk at English Wikipedia, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=41600857>

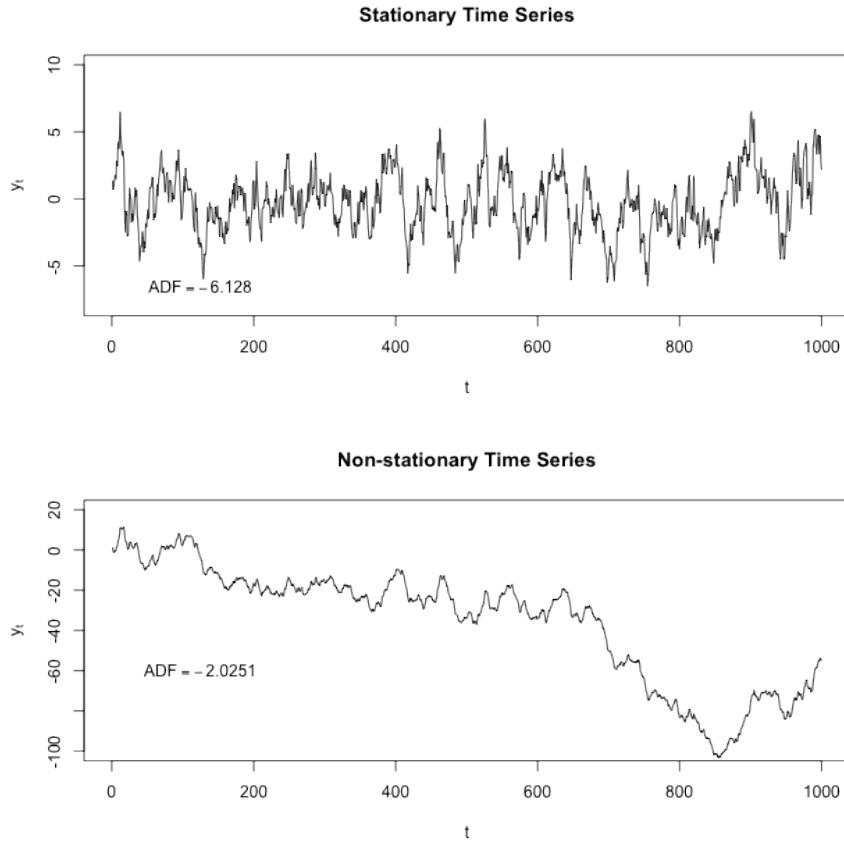


Figure 2.4: Comparison between a stationary (top) and non-stationary (bottom) time series<sup>4</sup>

muscles before activation. Therefore, the measurements related to a movement need to be shifted by an amount of time as large as the interval between EEG and EMG activations. This synchronization is necessary to create trials for each recorded movement that contain the relevant patterns. Currently there is no consensus about this delay, but it is estimated to be somewhere between 10 and 50 ms (Safri & Murayama, 2007; Witham et al., 2011; Schouten & Campfens, 2012; Witham & Baker, 2012). Note that this delay will vary between measured muscles, depending on the distance from the brain. The chosen solution that is used by our system is described in section 5.1.1.



# Chapter 3

## Machine Learning

In this chapter we will outline the basic notions of Machine Learning (ML) that are necessary to understand the technical specifications of our system. Firstly, the background describing what ML is and how it works will be presented in section 3.1. Additionally, we will have a more in-depth look at artificial neural networks and deep learning in subsections 3.1.2 and 3.1.3 respectively, since we chose to use a deep neural network architecture for our model. Secondly, certain well-known ML techniques that have been used in biological signal classification and analysis will be discussed in section 3.2. Section 3.3 provides a more detailed overview of hybrid BCI techniques, as our proposed model also falls within this category.

### 3.1 Background

Machine Learning is the discipline of computer science where the objective is creating algorithms that attempt to solve a problem by learning from data relating to the task at hand. An ML system is a form of artificial intelligence (AI) that can extract knowledge from raw data. This is in contrast with regular algorithms that are explicitly programmed for a specific task and can not generalize beyond what was implemented by the programmer, without modifying the source code. Or alternatively, other AI systems (e.g. logic based) that use a knowledge base of information that was encoded beforehand. ML models solve the problem from experience, i.e. data, rather than knowledge that was implemented by a programmer. The name Machine Learning was first coined by Arthur Samuel in 1959 (Samuel, 1959) when investigating a program that could learn how to play the game of checkers.

### 3.1.1 General Overview

In essence, ML consists of generating the right target output for a given input based on previous experience acquired from training on a set of input-output examples. The purpose of such a model is to also be able to compute correct output for previously unseen input. If the model performs well on unseen inputs, it is said that the model generalizes well beyond the training data. Learning a model thus involves the use of statistical methods that allow for a model to generalize beyond inputs that have been encountered during training.

Gaussian Mixture Models (GMM) is one of the most simple ML models found in literature. It has been used for a variety of applications such as image background removal (Zivkovic, 2004) and voice recognition (Reynolds & Rose, 1995) among others. A GMM model consists of a set of Gaussian distributions that are associated to classes in a classification problem. This parametric model tries to find the appropriate means and standard deviations that describe the Gaussian distributions, which are not known beforehand. Such a model needs to assign appropriate weights to the mixtures for the given input, as to give maximal weight to the one corresponding to the right class. The weights correspond to the probability of belonging to a class and must therefore sum to one. The choice of a class is subsequently based on these weights and is considered as a separate component of the ML system. Since this model operates on a population where class membership is not known beforehand the model is said to be unsupervised. One hyperparameter in this model is the number of mixtures. This model requires the user to provide a number of mixture as the number of classes is unknown. On the contrary, in semi-supervise learning the number of mixtures are known, however, we cannot assume that the data is normally distributed. In this case, other distributions could be used as mixture of models. The choice of a normal distribution can also be considered a hyperparameter. We then call this a general mixture model where all distributions should belong to the same family. Such a model has been used for DNA analysis (Bailey & Elkan, 1994) among other things.

Machine Learning models are decomposed into parametric and non-parametric models. Parametric models learn the initially unknown parameters (e.g. mean or standard deviation) by minimizing the output error, according to some error measure, with relation to the training examples. A well-known error measure is the mean squared error (MSE) between prediction and target. Alternatively, Non-Parametric models will compute output based directly on the training data. Depending only on the training data may lead to an increase in complexity and computational time if the amount of training data increases. For parametric models, the complexity depends on the number of parameters and the accuracy should increase when more training data is available. However, this come at the cost of often long training times to find appropriate parameters, while non-parametric

models do not need to be explicitly trained.

Depending on the availability of output labels for the training examples, we can distinguish two categories of ML techniques. These are Supervised Learning and Unsupervised Learning. Supervised learning will use the similarity between target and predicted output to train a model. In contrast, unsupervised learning has no available targets and therefore requires other measures that can assess how well the model performs. For example, a low clustering coefficient may be a measure of the performance of a clustering algorithm.

When only partial or restricted feedback is available for training, we can use either Semi-supervised learning, Active learning or Reinforcement Learning. In Semi-supervised learning only part of the target outputs are available, while Active learning can only train on a limited set of examples based on a budget. Here the selection of training examples needs to be optimized too. Reinforcement learning does not use training error as feedback but receives a reward signal based on the chosen actions in a dynamic environment. Examples of such a dynamic environment include driving a vehicle or playing a game against an opponent. Supervised learning tasks can further be divided in regression or classification, depending on whether the target output is a real number or a choice among a discrete set of classes respectively.

Depending on the task at hand and available training data, one needs to choose an appropriate statistical model that can efficiently and effectively learn from training data without over-fitting. Over-fitting occurs when a model does not generalize well and only performs well on the training data, but much worse on unseen inputs. Usually, to investigate if a model is over-fitting on the training data, the entire dataset is split into three distinct sets during training. The test set is a subset of the data that is used solely for evaluation of the trained model. It gives a measure on how well the model performs on unseen data. The validation set is also used solely for evaluation, but is referred to during training as a means to know when training should stop. It is not uncommon to stop training when a certain classification accuracy on the validation set is reached for example. The training set is used to update the model parameters based on the training error.

To achieve good results it is crucial to choose an appropriate approach from the alternative methods detailed above. Another essential design choice for an ML model are the hyperparameters. These are parameters of a model that are fixed beforehand and do not change any further during training. For example, the number of layers in a neural network is a hyperparameter. Certain ML approaches will also attempt to explore the hyperparameter space and find the optimal choice of hyperparameters by means of a heuristic search process.

A non-parametric approach is K-Nearest-Neighbours (KNN), which bases predictions on a number  $k$  of neighbours. These neighbours are defined as training

samples that are closest to the given input based on certain measure of distance. For example Euclidean distance is often used in a vector space where every entry of the vector corresponds to an input variable. Furthermore, another model called decision trees attempts to build a tree that best describes the training data. Branches are followed according to the input values and leaf nodes then correspond to predictions. Support Vector Machines (SVM) is a model that has found success in recent times. This approach will attempt to construct a separating hyperplane, i.e., a multi-dimensional plane in the input parameter space that separates the data in two categories. With two dimensional input this would be a straight line, which is why it is referred to as a binary linear classifier. Since most problems can not be linearly separated, the data is projected into a higher dimension where the separation becomes linear. However, as a result the model becomes untractable for large input dimensions. To avoid computing the projection of input to higher dimensions a kernel is used. This is referred to as the kernel trick (Shawe-Taylor et al., 2004). If there are more than two classes, the multi-class problem is decomposed into several binary classification problems. There are several other possible ML approaches that we will not discuss further. For an overview of well-established ML models we recommend reading Nasrabadi (2007).

### 3.1.2 Artificial Neural Networks

In relation to our research, the most significant ML methods are Artificial Neural Networks (ANN) and deep learning. As mentioned in the previous chapter, ANN are based on the biological neural networks that can be found in our brain. This computational model was first proposed by Walter Pitts and Warren S. McCulloch (McCulloch & Pitts, 1943) and served as the basis for neural networks as we know them today. A simple example of an ANN is shown in figure 3.1.

A neural network consists of several interconnected layers of computational units, also called nodes. All nodes in a network could be connected to each other in a neural network setting. However, researchers have converged to a feed-forward architecture as shown in figure 3.1, where nodes in one layer are connected only to nodes in the next. Additionally, when every node in a layer has an outgoing connection to a node in the next layer, it is said to be fully connected. Other connection variants are discussed in section 3.1.3.

When input is presented to a neural network, the input variables are each assigned to an input node. Therefore, the network should have the same number of input nodes as there are variables in the input. The nodes of the input layer can be connected either directly to the output layer or to the first hidden layer. The output of a node will then become the input of the nodes it is connected to with an outgoing connection. Outputs of all incoming connections are then combined to form this next node's output. To achieve this, every node of each layer will

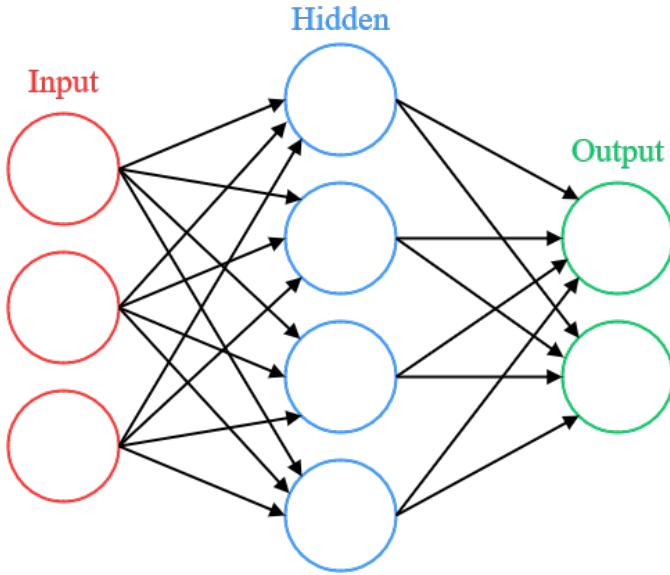


Figure 3.1: A feedforward artificial neural network with a three-node input layer, one four-node hidden layer and a two-node output layer, all fully connected

propagate the input that it receives from the previous layer with formula 3.1. For node  $j$  at layer  $l$

$$o_j^l(x) = \gamma(\sum_i o_i^{l-1}(x)w_{ij}) \quad (3.1)$$

where  $x$  is the original input,  $\gamma$  is called the activation function,  $o_i^{l-1}(x)$  is the output of node  $i$  in layer  $l-1$  and  $w_{ij}$  is the weight of the connection between nodes  $j$  and  $i$ . The activation function is often the sigmoid or tanh function. When the input has been propagated in this manner to the output nodes, these will contain the values that form the model output.

The weights between nodes are the parameters of the model and what needs to be trained in order to compute the correct output. However, as stated by the law of large numbers, the more parameters a model has, the more training data is necessary to successfully train the model. In neural networks the number of parameters can quickly reach thousands or even millions, especially with deep architectures. The very simple architecture from figure 3.1 already has 20 parameters that need to be trained. This is the reason why neural networks usually require more data than other ML models.

A Neural network is trained by the backpropagation algorithm. This algorithm will compute the update to the weights of a layer based on the error by computing the gradient on this error. This error is obtained by using a loss, or cost, function that represents the distance between predicted and target outputs. In this

manner the output error can be used by the layer before the output layer and so on. The error gradients of previous layers, used to update their own weights can be computed from the ones computed in the next layer. They are said to be propagated back to the previous layer. This algorithm had an important role in bringing neural networks from solving toy problems to solving real-world tasks. Due to the complex nature of weight updates in neural networks, an optimizer such as Adam (Kingma & Ba, 2014) is usually used to improve training.

### 3.1.3 Deep Learning

Deep learning refers to ML models that use more than one sequential computational step to extract informative features from the raw data. More generally, deep learning are the tools that allow for these models to work. The purpose is to learn more high level representations of the data until only the information that is useful to the problem at hand remains in the extracted features. This is in contrast to classic ML models that use hand-crafted features that were implemented by the model designer. For example, for image recognition, edges can be used as features. A mapping is then required for the ML system to recognize these edges from the raw pixel values of an image. With representation learning these same features could be learned, often resulting in more robust features. Deep learning is a form of representation learning where a new higher level representation is extracted from the previously extracted features. For image processing the learned features could be edges, corners and contours, and object parts respectively. We refer to the Deep Learning book by Goodfellow et al. (2016) for a more in depth overview of the subject.

In the context of ANN this translates to neural networks with more than one hidden layer. Each layer in the network can be considered as a single sequential step that evaluates all nodes in the layer in parallel. Such a network is therefore called a deep neural network. Due to every additional hidden layer introducing new weights from the nodes of the previous layer and to the ones from the next layer, the complexity of such a model increases by a large amount for every new layer. This can especially be observed when there are many nodes in a single layer, which generally is the case. Since neural networks were initially only able to solve toy problems at significant computational cost, they were dismissed as not very useful in the early days of ML. However, due to recent advances in training techniques and the evolution of hardware training ANN for real-world tasks has become feasible. Deep learning has become possible in recent years and has been able to proliferate, giving rise to even more new training approaches, network architectures and practical solutions.

One of the most crucial aspects and earliest decisions that needs to be made when using neural networks is choosing the appropriate architecture. Depending

on the type of problem we are attempting to solve and the type of data we want to process. Many different types of architectures exist<sup>1</sup>, all serving a different purpose. Models of interest to our research are Convolutional Neural Networks (CNN, or ConvNet)(Lecun et al., 1998) , Recurrent Neural Networks (RNN) (Elman, 1990), Auto Encoders (AE) (Bourlard & Kamp, 1988) and Generational Adversarial Networks (GAN) (Goodfellow et al., 2014). These are relatively early neural network architectures that have found success in recent years thanks to deep architectures and new model variants. They have been used in such diverse tasks as image recognition (Krizhevsky et al., 2012), speech recognition (Sainath et al., 2013; Graves et al., 2013; Graves & Schmidhuber, 2005), initialization of other networks (Vincent et al., 2008) or image super sampling (Ledig et al., 2016). These architectures have also found some success in the field of BCI as we will see in the following section.

## 3.2 ML for BCI

In this section we will review some of the most successful models for BCI and how insights from certain studies that use biosignals can suggest the appropriate approach to take when designing such a model. These studies can also justify many assumptions that are made by some models and by our own. Subsection 3.2.1 reviews the procedure to BCI processing together with examples of models used for these tasks. Subsection 3.2.2 gives an overview of ML models that have been used for classification in a BCI context.

### 3.2.1 The BCI Pipeline

The typical procedure when working with biological signals in the context of ML consists of several steps. These same steps are also present in direct brain-computer interaction with added control and feedback related signals. The initial step in the development of any ML model is data acquisition. In the context of BCI, this consists of taking sensor measurements and either storing them on a machine or sending them to a model for evaluation. In the first case the model works in an offline setting, which is a standard technique for initial validation of most new ML models. The stored data can then be used for training and testing at a later time. The second situation, referred to as online, is most relevant for direct brain-computer interaction, where measurements are collected and used in an online application. Whether working online or offline, the continuously measured data needs to be split up in trials. A trial is a segment of the recorded

---

<sup>1</sup>For an overview of some common neural network architectures: <http://www.asimovinstitute.org/neural-network-zoo/>

signal that is relevant to the problem of interest and usually annotated with a label. For example, a trial could contain the observed pattern when performing a grasping movement with your right hand. It would then be labeled as belonging to the associated class and can be used as a training example for an ML classifier. Online measurements make things more complicated however, as deciding which measurements to send at which moment becomes an additional concern. The remainder of the model should then additionally be able to handle a continuous sequence of trials coming in, often at short time intervals. Often, a novel model is developed offline, but takes possible online use into consideration too.

After the raw data has been collected, a pre-processing step usually takes place. This usually entails performing some of the denoising operations seen in the previous chapter, to enhance the SNR. For research related to EEG and the associated brain functions it is an important step to work with a signal that resembles the true signal. In contrast, ML focused research regularly ignores this step, as using raw data directly can give better results. In an ML context, learning to handle noisy input might be part of what should be learned. Spatial filtering methods are often used for enhancing the SNR of the signal. In the context of EEG, these methods include Laplacian re-referencing (McFarland et al., 1997), common spatial patterns (CSP) (Blankertz et al., 2006), principal component analysis (PCA) (Lugger et al., 1998) and independent component analysis (ICA) (Makeig et al., 1996; Naeem et al., 2006).

It is usually at pre-processing time that the non-stationarity of signals is dealt with. One possible way to deal with non-stationarity is by differencing, i.e., taking the difference between consecutive observations. Applying log transformations can also help stabilize the variance of a time series. After one of these transformations has been applied this data is used instead of the raw measurements. However, these techniques are usually not applicable in real time and are often not compatible with other methods such as filters. The signal is also expected to have high enough SNR in these cases.

The next step in the BCI processing pipeline is feature extraction. This step will attempt to extract relevant and often higher level features from the raw EEG data. The purpose of this step is to get more informative features than what can be learned from regular sensor data. Choosing an appropriate feature extraction technique is an important step in most ML applications. Among some of the well-known methods are band power measures (Scherer et al., 2007), Fourier transform (Müller-Putz et al., 2005), Wavelet packet analysis (Graimann et al., 2004), (adaptive) autoregressive parameters (Schlögl et al., 1997), and phase measures such as the phase-locking value (Brunner et al., 2006) or bi-phase relationships (Darvas et al., 2009). Finding good feature extraction methods is an important aspect of BCI research, as online systems require both fast and accurate methods.

The final step in BCI processing is classification, discussed in the next section together with a more extensive overview of applied models.

### 3.2.2 Classification

When the measured signals have been processed and features have been extracted we can pass these features as inputs to a ML classifier. There are several choices available for this step and numerous models have been investigated. The first classifier that was used for EEG classification was a simple linear classifier based on Bayesian decision rules (Vidal, 1977). Other models that have been investigated are linear and quadratic discriminant analysis (Vidaurre et al., 2006), source analysis (Qin et al., 2004), support vector machines (SVM) (Muller et al., 2003; Shenoy et al., 2008), hidden Markov Models (HMM) (Obermaier et al., 2001), biological evolution inspired Genetic Algorithms (Scherer et al., 2004), clustering methods such as distinction sensitive learning vector quantization (DSLVQ) (Pregenzer & Pfurtscheller, 1999) and greedy search algorithms such as the sequential forward floating selection (Leeb et al., 2007).

Neural networks have also been used as classifiers for EEG data relatively long ago (Peters et al., 1998). However, since recent innovations in deep learning there has been a significant increase in neural network models that attempt to either classify EEG from raw data, or process the data for evaluation by a different classifier. One of the most recent and successful neural network models uses convolutional neural networks to classify raw data (Schirrmeister et al., 2017). A CNN will learn higher level features for each convolutional layer, eliminating the need for potentially costly pre-processing and consequently improving the computation time. This potentially makes such a model a suitable candidate for online applications.

To solve the issue of noisy data and the distribution of the measurements varying between subjects, or even between recording sessions for the same subject several deep-learning techniques have been investigated. One of the main approaches to dealing with non-stationarity has been transfer learning (Jayaram et al., 2016). Transfer learning is a ML method that consists of training a model on a certain source task and then using this trained model for a different, but similar, target task. Several methods for transfer learning exist, mostly being concerned with accelerating training for the target task by using the knowledge gained from the source task. The technique has been widely used in a variety of fields and constitutes an area of research on itself. An extensive review of the concepts of transfer learning is given in Thrun & Pratt (2012), originally published in 1998. In the context of BCI, transfer learning is mostly used to calibrate (i.e. retrain) a model for a new subject (Zheng & Lu, 2016) for which measurements are available. This could potentially allow for a plug-and-play BCI system that only requires a

short calibration session from a new user. Transfer learning was also successfully used in CNN (Sakhavi & Guan, 2017) where the sparsity of data is usually an issue due to the number of weights that need to be trained every time. In a deep learning setting this is of great benefit as it limits the number of training samples required to train a neural network for a new subject.

A different deep learning approach that has been investigated is the use of an autoencoder to extract features from raw EEG data. Autoencoders that remove noise, called denoising autoencoders, have been used to retrieve a corrupted signal (Li et al., 2014; Vincent et al., 2008). This approach allowed to increase the SNR and improve learning for several models. Autoencoders have also been investigated as a way to deal with the non-stationarity of EEG data (Chai et al., 2016) and for transfer learning (Glorot et al., 2011) in the context of EEG based emotion classification.

One interesting approach to improving BCI performance is error-decoding. For this type of task, models need to learn an error-signal (Ferrez & del R. Millan, 2008) from subjects. These are the EEG patterns that occur when a subject perceives an action to be incorrect. This can be used for example to correct mistakes made by a BCI controlled robot in real-time (Salazar-Gomez et al., 2017). In this setting CNN have also been used in conjunction with transfer learning (Völker et al., 2017). Being able to detect error-signals could certainly be useful in practical BCI systems to reinforce correct movement during training and request a re-evaluation of the signal to produce a correct command when an error is detected.

Regarding other biological signals than EEG, the second most widely used signal is probably EMG. It was successfully used for online control of a robot arm (Alte, 2017) as an inexpensive BCI solution. This research investigated several classifiers for EMG, being SVM, K-nearest neighbours and Gaussian classifiers. Other EMG classifiers have used neuro-fuzzy inference (Khezri & Jahed, 2011), wavelet neural networks (Subasi et al., 2006) and CNN (Atzori et al., 2016) in this context. One drawback to using EMG only, however, is that it is slower than EEG, resulting in delayed responses.

Other signals used for BCI include Magnetoencephalography (MEG)(Mellinger et al., 2007), functional magnetic resonance imaging (fMRI) (Weiskopf et al., 2004; Sitaram et al., 2007), near-infrared spectroscopy (NIRS) (Gallegos-Ayala et al., 2014; Coyle et al., 2007) and pupil size oscillation (Mathôt et al., 2016) among others. These approaches were successfully applied as BCI methods, however, they are often not portable or require expensive hardware as previously mentioned.

### 3.3 Hybrid BCI

All models discussed above only made use of a single input modality to make decisions. We believe, like many others (Novak & Riener, 2015), that the accuracy and performance of such a system can be improved by using additional input modalities for classification. We give an overview of some successful models and different methods for fusion of two or more input modalities in the context of BCI.

Fusion of multiple input modalities can happen at several moments on the processing pipeline. When the raw input from the different modalities is combined before any processing happens on the data, the fusion is said to happen at the input level. The combined inputs are then processed together in a common data format. This level of fusion is feasible only if input modalities are very similar, such as two different electrical signals sampled at the same frequency. The combination of measurements from different EEG electrodes can be considered input level fusion. Alternatively, the combination of separate input channels can happen after feature extraction has taken place. This is referred to as feature level fusion and is more often used than input level fusion. In this case the signals would be pre-processed and features extracted, after which the features are combined and passed to the classifier. The highest level of fusion is called decision level fusion. Here the different input modalities are each passed to a separate classifier that makes a decision on the single input modality. In this situation the final decision of the model is based on a combination of the output generated by the individual classifiers. Deciding what fusion level to choose depends on the type of inputs and the problem at hand. This is an important design decision of any multimodal or hybrid ML system. An overview of the pipeline for decision making in BCI with possible places of fusion being indicated is given in figure 3.2.

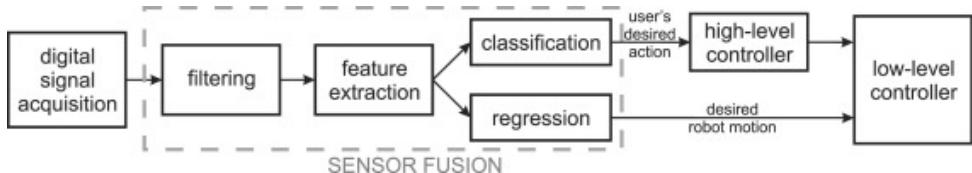


Figure 3.2: Pipeline for processing of sensor input in a BCI system <sup>2</sup>

Next to the level of fusion a choice has to be made related to the input modalities used for fusion. For this aspect, the choices are unimodal fusion and multimodal fusion. With unimodal fusion, different signals from the same input modality are combined. The previously mentioned fusion of different EEG electrode channels can be considered as unimodal fusion. Multimodal fusion will combine

---

<sup>2</sup>Reused from Novak & Riener (2015) under license

modalities of different types to either disambiguate the primary input or make a combined decision.

Additionally with multimodal fusion, there are several manners in which different modalities can be combined. One such way is by using a single fusion algorithm where feature level input is combined and passed to the classifier for evaluation. Alternatively, unimodal switching will use one input modality to choose a sensor processing algorithm that uses a second input modality as its unimodal input. Multimodal switching follows the same principles to choose a sensor algorithm, but uses multiple input modalities to make a decision. Finally, mixing consists of decision level fusion where several, potentially multimodal, classifiers run in parallel and the combined output is 'mixed' at the end to form the final system output. More information on these methods can be found in Novak & Riener (2015).

Among the different modalities that have been used in hybrid BCI, we are most interested in fusion of EEG and EMG signals. Several models have been investigated regarding this method. One such model uses mixing (Leeb et al., 2011) to make a combined decision based on a weighted average. Here, finding appropriate weights is an additional learning task. Combination EEG and EMG was also investigated as a means of robotic control (Chéron et al., 2012). EEG and EMG are also often used in conjunction for analysis of behaviour when performing a task (Kiguchi & Hayashi, 2012). An overview of other models can be found in Dulantha et al. (2013).

Other modalities that were investigated in combination with EEG are EOG (Jiang et al., 2014; He et al., 2017), MEG (Pfurtscheller & Lopes da Silva, 1999), ECG (Pfurtscheller et al., 2008), and EOG (He et al., 2017) among others. Other modalities were also combined in several other studies such as EMG and visual feedback (Markovic et al., 2015). For other investigated methods, see Pfurtscheller et al. (2010) and Novak & Riener (2015).

# Chapter 4

## Design & Architecture

This chapter will discuss the design and implementation of the EEGPlus model that was implemented for validation of our hypothesis. Section 4.1 will motivate the choice of learning model based on insights gained from the information that was presented in previous sections. Following, section 4.2 will discuss the design of the EEGPlus model. Here subsection 4.2.2 proposes some alternatives and motivates the final choice of the implemented model among those alternatives, which is discussed in subsection 4.2.1. Finally, section 4.3 will discuss a conceptual online architecture that could be implemented with the EEGPlus model. Proposed components using existing tools for this conceptual model are also given.

### 4.1 Motivation

Up until now we have established that one of the biggest disadvantages of biosignals is that they have a low signal-to-noise ratio. This means that interference from other sources of electrical fluctuation can hide the patterns that are generated by brain activity. Nevertheless, it has been shown that useful information can be extracted from noisy data that was measured with a commercial device (Marulanda, 2013). As observed in the previous section, it is also possible to combine multiple modalities for an enhanced classification performance. We chose to combine EEG as primary input with EMG as secondary input for the purpose of this thesis. However, the proposed system is designed in such a way to allow for additional or alternative secondary input sources. Since the availability of these additional input modalities can not be guaranteed, the model also allows for removal of a secondary input source after the completion of training. For example, an amputee will be missing the muscles that generate the EMG signal that was used for training. However, as long as the subject is conscious there will be an EEG signal that can be used for BCI. Therefore, a system that can improve EEG

classification by finding correlations with other input modalities seems like a good solution.

There are several possible ways to design a system that allows for one primary modality to be used in conjunction with optional secondary input modalities. If two modalities are fused at the decision level, it is simply a matter of making the final decision, either based solely on the primary modality, or in conjunction with secondary modalities when they are available. However, such an approach would not improve the classification performance of the primary modality when used individually. Classifiers for each modality would need to be trained separately in this case without learning correlations between the different inputs. Finding correlations between modalities, such as patterns in EEG and EMG for the same movement, is an essential aspect of the proposed model.

In contrast, if modalities would be fused at the input level it would become harder to separate the primary input from the secondary modalities. A classifier trained on a set of inputs can usually not be used effectively with only partial inputs available. This would also introduce the additional issue that different modalities would first have to be processed into a common representation before being sent to the classifier. This is not a significant issue if the input signals are very similar, but excludes more disparate modalities to be used in conjunction. It also does not take into account that appropriate feature extraction techniques should be applied for each distinct modality. We therefore chose to combine inputs at the feature level for our model.

The next matter that needed to be dealt with consisted of identifying a method to learn correlations between different input modalities. A handcrafted solution might work well, but this requires extensive knowledge of both the different input signals and machine learning. This furthermore limits the types of input signals that could be used. One approach that has been successful at learning such correlations is deep learning (Ngiam et al., 2011). It additionally allows for features to be learned, rather than using predefined features, by the same model that learns to classify the data. Deep learning moreover seems like an ideal candidate as it also is used in state-of-the-art classifiers for EEG (Schirrmeister et al., 2017). A deep learning approach would allow for end-to-end learning, meaning that every aspect of the classification pipeline is learned. Such an end-to-end learning approach often performs best as demonstrated by the FasterRNN (Ren et al., 2015) model for object detection.

To achieve the goal of training a neural network that uses EEG as primary input and optionally EMG as secondary input, a novel network architecture was required. A regular neural network that would take both modalities as input could not work when only one of the inputs is present. While using separate networks to classify each modality would not learn any useful correlations between inputs, as previously

mentioned. Several possible architectures where considered for this purpose. Only one of those was implemented and evaluated due to time restrictions. Ideally, all alternative architecture would be investigated at some point to either reject them or validate them. An iterative process of evaluating design alternatives and designing new ones based on the chosen alternative, similar to the procedure that is used in agile software development, seems like an appropriate method. The next section discusses all designs that where considered for such a model and motivates the final choice.

## 4.2 Design

This section introduces the design of the EEGPlus model, named thus, for it accepts EEG inputs in conjunction with optional secondary inputs of a different type. For this implementation they are EMG inputs. The implementation and architecture details of the evaluated architecture are presented in subsection 4.2.1. Design alternatives will be discussed in subsection 4.2.2.

### 4.2.1 Implementation

The chosen neural network model was implemented in the Python programming language with the help of the PyTorch (Paszke et al., 2017) deep learning framework. For processing of EEG data and other biological signals, the MNE (Gramfort et al., 2014) module for Python (Gramfort et al., 2013) was used. Additionally, the Braindecode (Schirrmeyer et al., 2017) toolbox provided us with functionality that allowed for an efficient experimental setup and comparison with existing models provided by the library. Modifications had to be made to the Braindecode code that handles the experiment training loop and monitoring of training performance. These modifications where necessary to allow for multiple predicted outputs and target outputs to be processed and combined. Parts of the Scikit learn (Pedregosa et al., 2011) library were also used for computing result metrics, which will be discussed in the next chapter. The code for this implementation can be found at<sup>1</sup> together with the modified Braindecode code.

The neural network architecture that was implemented attempts to find correlations between EEG and EMG by treating EMG signals as an output rather than an input. This idea is based on the same concepts as the previously discussed DIVA model. The conceptual schema of this architecture is depicted in figure 4.1.

In this model, the neural network takes EEG as input and attempts to predict a class label and a corresponding EMG response in parallel. The loss functions

---

<sup>1</sup><https://bitbucket.org/FelipeG/bci/src/master/>

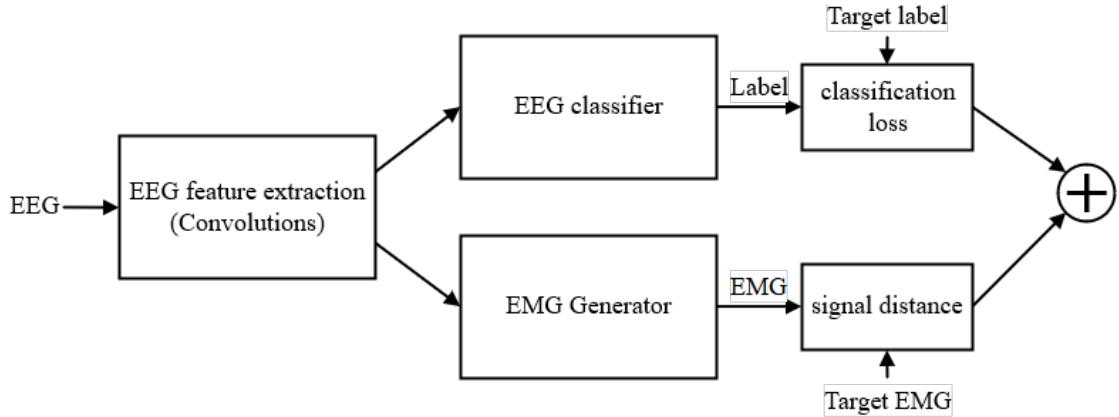


Figure 4.1: Overview of the implemented neural network architecture

that serve to compute the error between predicted and target outputs are then combined, resulting in a single loss value. The belief in this case is that the EEG classifier will be improved because the error between predicted and target EMG outputs will also contribute to the updates of network parameters of the feature extraction and EEG classifier modules thanks to backpropagation. One significant advantage with this architecture, is that removing the EMG input after training is trivial. The network can be used as is and the EMG predictions can be ignored, with no need to modify the network after training.

The input to this network will be a tensor of shape  $(B, C, W, H)$ , where  $B$  is the batch size and  $C$  the number of input channels.  $W$  and  $H$  are the width and height of the input respectively. Some important aspects to note are the following. 2D convolutions are applied here with a width  $W$  equal to the number of samples in a trial, i.e., the window size. The height  $H = 1$  for all layers except for the convolution in time where the channel and height dimensions are switched. This also is the reason why no 1D convolutions are used instead. Another important aspect related to input, is the size of kernels and the stride of the layers in the EMG generator module. Depending on the shape of the output after the last layer of the feature module, appropriate values are essential to generate EMG output of the same shape as the target EMG signal. The window size plays an important role here and will therefore be discussed in more detail in the next chapter.

The ensuing subsections will discuss the implementation details of the different modules of the EEGPlus model. Subsection 4.2.1.1 will detail the implementation of the EEG feature and classification modules. In subsection 4.2.1.2 the EMG generator module is discussed. Here, a motivation for the design will be given and a small discussion is presented regarding possible alternative architectures for this module. Finally, subsection 4.2.1.3 discusses the loss module and possible ways of combining different loss functions.

#### 4.2.1.1 EEG Feature & Classification Modules

The first module of this network will extract features by applying a convolution operation over the space (channels) and time domains. The outputted features then serve as input to two parallel networks that are trained to predict classes and EMG respectively. The feature extraction module and classification module are essentially the same as the shallow network variant investigated in Schirrmeister et al. (2017). The proposed EEGPlus architecture adds a side network after the feature extraction has taken place in this model. The implemented instance of the network is described in table 4.1 for the feature extraction module and table 4.2 for the classification module. The size of the input for this instance is  $(B, 32, 750, 1)$ , i.e., 32 EEG channels with trials of 750 samples. Also note that the input tensor for the feature module is first permuted as mentioned before to  $(B, 1, 750, 32)$ , which is not explicitly shown in the table. The Square and Log modules simply refer to the mathematical operations which are applied to all outputs from the previous layer as activation functions.

module name	module type	kernel size	stride/dilation	output size
conv_time	Conv2d	$25 \times 1$	$1/1$	$b \times 40 \times 726 \times 32$
conv_spat	Conv2d	$1 \times 32$	$1/1$	$b \times 40 \times 726 \times 1$
bnorm	BatchNorm2d			$b \times 40 \times 726 \times 1$
conv_nonlin	Square			$b \times 40 \times 726 \times 1$
pool	AvgPool2d	$75 \times 1$	$15 \times 1$	$b \times 40 \times 44 \times 1$
pool_nonlin	Log			$b \times 40 \times 44 \times 1$
drop	Dropout			$b \times 40 \times 44 \times 1$

Table 4.1: Layers and their configurations of the feature module

module name	module type	kernel size	stride/dilation	output size
conv_classifier	Conv2d	$12 \times 1$	$1/1$	$b \times 3 \times 33 \times 1$
softmax	LogSoftmax			$b \times 3 \times 33 \times 1$

Table 4.2: Layers of the classification module

The types of the different network layers in the tables correspond to those used by the PyTorch modules that implement those layers. Detailed information on these modules and their workings can be found in the PyTorch documentation<sup>2</sup>. The module types should imply their function, but it is recommended to examine the documentation entries on these modules for those unfamiliar with PyTorch or deep learning frameworks in general.

<sup>2</sup><https://pytorch.org/docs/stable/nn.html>

From table 4.1 we can see that the final output of the feature module has shape  $(B, 40, 44, 1)$ . This tensor will contain the features that are passed to both the classifier and generator modules. The ensuing paragraphs detail the propagation of input in the feature and classifier modules.

The feed-forward propagation of input that is described in table 4.1, goes as follows for the feature extraction module. An EEG trial of appropriate length (750 samples here) is presented to the network for each of the selected channels (32 here). The input can therefore be seen as a  $32 \times 750$  matrix that contains the sample values for each trial. Note that we use a shape of  $32 \times 750 \times 1$  in practice, to allow for the next step in the computational graph. This input matrix is permuted to  $1 \times 750 \times 32$  to allow the convolution over the space domain. The original implementation of the ShallowEEG model on which this part of EEGPlus is based allows for a choice between either a convolution in time solely, or convolutions over space and time. The former does not require the permutation of input, however, the latter approach was chosen for the EEGPlus implementation.

After being permuted and going through a convolution over the time and space domains that are described by the first two rows of table 4.1, the input goes through a batch normalization layer. This layer uses statistical techniques to normalize outputs from the previous layer for a whole mini-batch of inputs. The presence of this type of layers is one of the reasons why inputs are presented in batches to the network instead of one by one. Following normalization, a non-linear function is applied element-wise to each output from the last layer. For this implementation a simple square operation was chosen. These outputs then go through a pooling layer that computes the next output by using a pooling function based on a window of outputs from the previous layer. In this implementation average pooling was chosen, however, max pooling can also be used for this purpose. The pooled outputs are also passed through a non-linear function that applies an element-wise operation on the output matrix. A logarithm operation was chosen for this purpose. The classification module finishes with a dropout layer that assigns a value of zero to the output elements of the previous layer with some probability. Such a layer is a commonly used method to avoid over-fitting.

The classification module accepts the EEG features as input and consists of two simple layers. The first layer is another convolution that outputs a  $C \times O \times 1$  matrix with  $C$  the number of classes and  $O$  depending on the chosen kernel size, stride and padding of the final layer. This convolution is then followed by the final SoftMax layer that will output probabilities for each class, by rescaling the outputs of the previous layer between 0 and 1. In this implementation we chose the LogSoftMax layer that additionally applies a logarithm operation after SoftMax. Omitted from table 4.2, the final output is additionally squeezed to remove the empty dimension of 1. Parallel to this, the features outputted by the feature

module are passed to the EMG generator module.

#### 4.2.1.2 EMG Generator Module

Generating EMG signals with a neural network was never attempted before in the literature. Achieving this was not a trivial task, since we want a neural network that will output a window of the same size as the input window. We therefore had to come up with a way to go from EEG features to EMG signals in a neural network setting.

One type of model that attempts to generate an output that is similar to the input is the Generative Adversarial Network (Goodfellow et al., 2014) (GAN). A GAN uses two networks, one to generate images and one to discriminate real and generated images. These two networks are then trained in conjunction. It has been used mostly in the context of image generation and seems to be an appropriate choice for EMG generation too. Therefore, a generator network architecture based on the generator network of the DCGAN model (Radford et al., 2015) was chosen for the EMG generator module of the EEGPlus network. Note that the only association with the GAN model in this context lies in the architecture of the generator module being the same as that of the DCGAN generator network. Training of the module happens in conjunction with the classification module, based on the distance between the generated and target EMG signal rather than the loss that is obtained based on the discriminator output that GAN uses.

A better model could be available, but finding a way to generate EMG from EEG features could be a research subject on its own. An auto-encoder (Bourlard & Kamp, 1988) approach was also considered for this purpose, as this model also attempts to reconstruct a target input from a high level representation. However, this approach was not further investigated, as GAN was chosen for this implementation. Investigating this approach should be an interesting study.

The layers of the final generator model are described in table 4.3. Batch normalization layers (BatchNorm2d) and rectified linear unit layers (ReLU) that follow every transposed convolutional layer but the last are omitted. For more information on convolutions and transposed convolutions, see Dumoulin & Visin (2016).

The extracted EEG features that are outputted by the feature module are the inputs to this module. In a typical GAN model, the input is often a random noise vector. However, here we want the network to learn how to reconstruct EMG signals from EEG features. Upon entering the network, inputs are passed through a transposed convolutional layer. Such a transposed convolution is often referred to as a deconvolution, as it up-samples an image rather than the down-sampling that happens with a convolution. However, it is often said that the term deconvolution is incorrect as it does not perform an actual deconvolution operation. Interested readers are referred to Dumoulin & Visin (2016) again for a

module name	module type	kernel size	stride/dilation	output size
conv_transpose1	ConvTranspose2d	142 × 1	1/1	$b \times 512 \times 71 \times 1$
conv_transpose2	ConvTranspose2d	142 × 1	1/1	$b \times 256 \times 91 \times 1$
conv_transpose3	ConvTranspose2d	142 × 1	1/1	$b \times 128 \times 185 \times 1$
conv_transpose4	ConvTranspose2d	142 × 1	1/1	$b \times 64 \times 373 \times 1$
conv_transpose5	ConvTranspose2d	143 × 1	1/1	$b \times 2 \times 750 \times 1$
emg_out	Tanh			$b \times 2 \times 750 \times 1$

Table 4.3: Layers of the EMG generator module. Batchnorm2d and ReLU after each ConvTranspose2d but last omitted.

more in-depth discussion on convolutions and associated operations. The output of this transposed convolutions layer is an up-sampled representation of the high-level features, as can be observed from the increased third dimension in table 4.3. As is common, the outputs are then passed through a batch normalization layer. The normalized transposed convolution output is consecutively passed through a fully-connected rectified linear unit layer with the purpose of learning a low-level representation from the up-sampled high-level EEG features. This sequence of layers is repeated four times until the final transposed convolutional layer. After this *conv\_transpose5* layer, the output is passed through a fully-connected layer with Tanh activation functions, resulting in the predicted EMG signal. In this module, the size of the kernel for each layer is essential, as the shape of the output depends on it. It needs to be carefully chosen to output an EMG signal of correct length, given the shape of the EEG features.

#### 4.2.1.3 Loss Function Module

The final essential aspect to creating a model that will be improved by the additional EMG output is the loss function. An appropriate loss function is necessary to correctly represent how far the predicted EMG is removed from its target. For this purpose, the cross-correlation between corresponding electrode channels of the predicted and target signal was chosen. This measures the similarity between the vectors of measurements by computing the Pearson product-moment correlation coefficients between them. Negative correlation is considered the same as no correlation and therefore zero. High correlation means similar signals and should therefore correspond to low loss. This is achieved by setting negative correlation coefficients to zero and subtracting the mean correlation between channels from one. Equation 4.1 shows the exact formula for this loss function

$$ccloss = 1 - \max(crosscorrelation(predicted\_EMG, target\_EMG), 0) \quad (4.1)$$

where crosscorrelation refers to the mean value of correlation coefficients between pairs of vectors that represent a single EMG channel of predicted and target signals

respectively. The concrete implementation of this consists of a custom PyTorch module that performs the mentioned operations. This measure was chosen in search of an appropriate distance measure between signals, as alternatives could not readily be found. Finding a way to measure how well an EMG signal can be predicted is an essential aspect to improving the EEGPlus model. Attempting to find a dedicated loss criterion related to EMG, or signals in general, seems like the way to go for this purpose.

The final step that combines the classification and generation loss functions consists of adding the loss functions to each other. For classification, the negative-log-likelihood (NLL) was chosen as loss function in this implementation. However, a different classification loss might be used here as this could be more appropriate for combination. The combined loss can be represented by formula 4.2

$$\text{loss} = \text{nllLoss}(\text{pred\_label}, \text{true\_label}) + \text{ccloss}(\text{pred\_EMG}, \text{target\_EMG}) \quad (4.2)$$

with nllLoss the NLL loss function and ccloss the previously described loss that is based on the cross-correlation. It is also implemented as a PyTorch module that is constructed with a list of losses and combines the different loss criteria it was initialized with on usage. Note that this approach is naive and assumes that there is no significant difference between the two outputted loss values. A weighting scheme or alternative combination method could be investigated as a way to improve this implementation.

Finding an appropriate way to compute a combined loss for multiple types of output is a subject that has not yet been extensively researched. The above described approach might be sufficient, or some advanced type of interpolation might be required to achieve our final goal of learning the correlation between EEG and EMG in parallel with EEG classifications. Alternatively, it might be possible that a whole new types of loss function is more appropriate in this situation. Maybe a single loss function that takes all outputs of the network into account to compute a single error measure is the most appropriate method. Much more research is necessary to identify the best method.

This implementation might not be the best way to achieve better EEG classification from co-training with EMG. Nevertheless, it seems to be a viable solution for now that can still be improved in several ways. Currently, the main issues that could be optimized are the individual module architectures; investigating the ideal way to compare generated and predicted EMG signals and combining different loss criteria beyond summing the individual loss values. The data used for training and the hyperparameters of the model should also be optimized. This will be discussed in the next chapter.

### 4.2.2 Alternatives

A first possible neural network architecture that was considered would be based on the DIVA model (Tourville & Guenther, 2011) that is used to simulate speech production in the human brain. The problem solved by this model is very similar to that of motor function. Both take a certain input signal and attempt to generate motor commands with availability of secondary input modalities. In the case of speech, this is auditory and sensory feedback, while for movement this feedback is mostly visual and sensory. In this situation EMG could also be such a feedback since it is a different input measure that is causally related to the original input. The idea here would be to have a network architecture consisting of different modules that are responsible for each input modality. The primary modalities would then be used to generate motor commands and expected values for each secondary input modality. Based on the error between the true input and the predicted input, corrective motor commands would be performed and feedbacks would backpropagated to the primary input module. However, translating this abstract model to a practical neural network architecture is not evident. This model also seems more appropriate in an online setting, since the corrective motor commands are unnecessary in an offline setting. We therefore chose to not investigate this approach further, keeping it as a conceptual design. However, the final implemented architecture uses certain ideas from this model.

Based on these ideas and with the goal of finding a neural network architecture that can learn a correlation between EEG and EMG signals in parallel with classification of said signals, two architectures were considered. Both seemed to be viable options, but only one was implemented and validated due to limited time. A conceptual overview of the unimplemented architecture is given here, while the implemented one will be discussed in detail in the next section. The motivation for the final choice was relatively arbitrary, mostly taking in consideration which architecture seemed to be the most simple to implement at design time. One possible way that was considered but not implemented is depicted in figure 4.2.

This model would consist of two separate neural networks that each take one of the signals as input. Each of these networks would then extract features, output a predicted movement and send the extracted features to a second network. This second network would then take the combined features of both EEG and EMG as input and also make a prediction for the proposed movement. Training the network in such a manner essentially trains three classifiers in parallel, one for each input signal and a combined one. The purpose would be that information learned by the joint classifier would also be incorporated in the separate classifiers through backpropagation. Switching to a single input modality would then be a matter of removing the connections going from the separate classifier to the joint one. The hypothesis would be that features learned with the joint classifier would

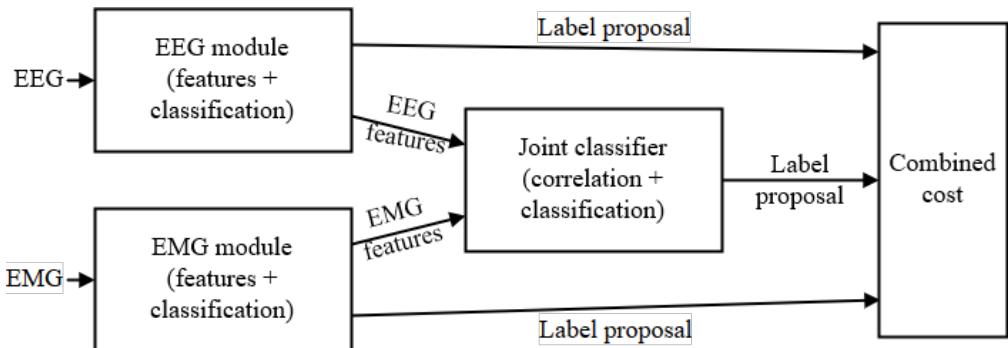


Figure 4.2: Overview of a conceptual hybrid NN architecture

improve classification with EEG input only.

Since the correlations between EEG and EMG are shifted in time, the signals have to be synchronized to learn meaningful information. This synchronization could happen manually by shifting one of the signals by a certain time offset, based on a known delay. However, as previously discussed in section 2.2.2, there is no consensus on the exact delay between EEG and EMG activity. This delay varies between concerned muscles and between subjects as this mostly depends on distance, but also the diameter of the axons of the nerve cells that transport the signal. Due to this issue we also considered that this delay could be learned by a neural network in one way or the other. A model that could learn this delay would have to make one less assumption and save a significant amount of time for research that would be required to identify the delay between EEG and EMG for every experiment otherwise. One possible way this could be solved would be by using a recurrent neural network architecture, which can learn delayed dependencies between consecutive units. A regular recurrent network did not seem practical however, as we do not know beforehand how far correlated patterns are delayed. This is what we want to learn in the first place. One possible architecture could be a variant of the Clockwork RNN (Xie et al., 2016) that is made up of different modules operating at different clock speeds. This approach is not entirely what we want, as clocks are fixed beforehand. However, it seems to be a promising candidate for this purpose. Alternatively, a similar architecture is the hierarchical multiscale recurrent neural network (Chung et al., 2016) that is similar to Clockwork RNN, but does not require clocks of different modules to be given beforehand. These possible architectures were not investigated further since this could be a research subject in itself and falls outside of the scope of the current research question.

A final architecture that was considered, but never investigated further than the conceptual phase, could be based on the Pathnet (Fernando et al., 2017) architecture that is used in transfer learning. The intuition for this would be that

EEG, EMG and hybrid classification can be considered as separate domains that attempt to solve the same problem. The idea would then be to effectively do the inverse of the original Pathnet where it is the input that changes rather than the output, due to a changing target domain. When the time comes to switch to unimodal EEG input, weights on the paths that are related to EMG could then be frozen to zero. Investigating a way to achieve this certainly seems a worthwhile study.

### 4.3 Conceptual Online System

Since the purpose of our system is to serve as a basis for the development of a robust and reliable control system for a robotic prosthesis, we decided to propose a conceptual architecture for such a system in this section. The system presented here was not implemented in practice but should serve as a good starting point for the development of such a system. This design is based on our experience from experimenting with the offline system previously presented and ideas from the literature. In this section a brief overview of how such a system could be designed with concrete hardware and software is provided.

Finding an appropriate model for real-time control through a BCI system is no simple task. Such a system has to be robust to noise, this is especially true for non-stationary biosignals, and should also be accurate and fast. In many such applications, understanding the decision made by the system can also be important. This often excludes deep learning as a candidate since understanding the decisions of neural networks is hard. However, in the case of a continuous control task such as controlling a prosthesis, the performance mostly matters. We therefore chose to develop the previously discussed system with online deployment in mind. This section can be considered as the initial step to developing a real-time system for direct brain-computer interaction.

The different components of a pipeline for sensor processing are depicted in figure 3.2 which was previously shown in section 3.3 from the previous chapter. The first step of acquisition is one of the main issues that was addressed by the EEGPlus model. The purpose was to design a system that could be constructed with a commercial device. Therefore, the OpenBCI<sup>3</sup> headset was chosen as the acquisition device for this conceptual system. This is an open source device that can be printed by anyone with a 3D printer at their disposal. Only the electrodes and control board need to be purchased, but these are also open source designs. This headset uses dry electrodes and can therefore be used with virtually no additional setup once it is assembled. However, this comes at the cost of lower quality

---

<sup>3</sup><http://openbci.com/>

measurements. Tough the classification model was especially designed with this in mind.

The next step in the pipeline consists of processing the continuous signal into discrete windows before it is sent to the classifier. For this purpose the MNE library that was used for the offline implementation can also be used. The library has real-time functionality<sup>4</sup> which can be used for BCI. The API provides server-client facilities that allow for processed data to be sent to a remotely connected device for classification. This approach seems ideal as it is likely that a neural network used for classification will be instantiated on a dedicated device that allows for fast evaluation.

There are two distinct ways to send windows of signal samples to a remote device. The first way employs an additional classifier model that can distinguish whether there is any activity of interest in a given window. Such a model could be deployed on the headset controller itself and allow for only relevant windows to be sent to a dedicated device. This device would contain the software that handles interpretation of a signal into actions to be preformed by the device that is being controlled through BCI. While this approach limits the amount of data that has to be sent to the separate controller, it also requires additional software to be executed on the hardware board of the headset. Alternatively, all sampled data could be sent to the hardware controller for processing, which delegates the task of identifying whether there is any activity at all to the main model. This choice would depend on the reliability and performance of the connection between the headset hardware board and the device controller. If latency of the connection is high, sending all data to a separate device for processing would introduce delayed responses of the system.

Once data has been sent to the dedicated device it has to be processed by a classification module into discrete configuration or actions for the device being controlled. As shown by figure 3.2 it is also possible to achieve this by regression. However, this usually complicates things for the ML model that performs this task. For this step the EEGPlus model discussed in the previous section could be used, or at least a future incarnation. Here the main requirement next to good classification accuracy is speed of the evaluation. The model should be able to evaluate a window of samples within a delay faster than the frequency of window segmentation. For the common sampling frequency of 500 Hz and a window size of 500 samples this should be below a second. This should ideally be even faster to allow for higher sampling rates or smaller window sizes.

Finally, for the controlled device itself, an existing robotic prosthesis was chosen to illustrate this concept. Remaining in the open source domain the prosthesis from

---

<sup>4</sup>[https://www.martinos.org/mne/stable/auto\\_examples/realtime/plot\\_compute\\_rt\\_decoder.html](https://www.martinos.org/mne/stable/auto_examples/realtime/plot_compute_rt_decoder.html)

Open Bionics<sup>5</sup> could be used. Most of the components for these prostheses can be 3D printed by the users themselves. However, any electronic device could be interfaced with by means of a BCI system. Appropriate robotic controller software was not studied yet for this research.

Creating a framework as described above would be the ultimate goal of this research subject. This would allow for anyone to easily get their hands on a plug-and-play BCI system. Users would then have the benefit of trivial setup of a whole BCI system, while developers could tinker with any components of the system if it is open sourced. It could be used for many applications ranging from robot control to video games.

---

<sup>5</sup><https://openbionicslabs.com/>

# Chapter 5

## Experimental Verification

This chapter will discuss the experimental verification procedure that was applied to validate the model design presented in chapter 4. Firstly, the procedure used to acquire and process the training data is discussed in section 5.1.1. Assumptions that were made about the data and suggestions to improve the labeled training data quality will also be presented. Following, section 5.1.2 presents the experiments that were performed to compare the EEGPlus model with the baseline convolutional network (ConvNet) from Schirrmeister et al. (2017), hereby referred to as ShallowEEG. Additional experiments that should give a better insight into the strengths and weaknesses of the model will also be proposed. These proposed experiments were not implemented however. Finally, the results to the implemented experiments are presented and discussed in section 5.2. These experiments were run with a 2.80 GHz Intel(R) Core(TM) i7-7700HQ CPU and 8 GB RAM. Computations were accelerated with a 6 GB NVIDIA GeForce GTX 1060 GPU.

### 5.1 Experiments

#### 5.1.1 Data

To evaluate the EEGPlus model, data that takes the form of labeled trials containing both EEG and EMG channels is required. Finding data from experiments that were performed with both EEG and EMG being recorder was no trivial task however. While labeled data for EEG trials can be found online and is freely available, no public datasets could be found that contain concurrent EEG and EMG recordings. Using a public data set would have been better, as this make it possible to compare results with other models, without the need to implement and test them ourselves. The required data therefore had to be acquired from an

alternative source.

The data used in the performed experiments was provided by our colleagues from the physiology lab (BLITS) at the VUB. No dedicated recordings could be performed in time for our research. This data came from an experiment where the subjects had to walk on a treadmill at different paces, while their EEG and EMG signals were being recorded. Note that identifying the walking phase from an EEG signal is a significantly harder task than identifying hand movements for example, which are used in most BCI research. The experiment was performed on healthy subjects, trans-tibial amputees and trans-femoral amputees. Our experiments only use the data from a single healthy subject due to certain issues discussed further in this section. For EEG, the following 32 electrode channels were recorded: *Fp1*, *Fp2*, *F7*, *F3*, *Fz*, *F4*, *F8*, *FC5*, *FC1*, *FC2*, *FC6*, *T7*, *C3*, *Cz*, *C4*, *T8*, *TP9*, *CP5*, *CP1*, *CP2*, *CP6*, *TP10*, *P7*, *P3*, *Pz*, *P4*, *P8*, *PO9*, *O1*, *Oz*, *O2*, *PO10*. Note that these electrode names come from a higher resolution version of the 10-20 EEG electrode placement system. The muscles that are recorded by EMG are the *Tibialis anterior* and the *Gastronemius*, which can be found at the front and back of the lower leg respectively. Additionally, the signal from a force-sensing resistor (FSR) that was inserted in the heel of the shoe of the subject is available. The intention would be to identify when the foot of the subject is down and when it is in the air based on the EEG and EMG signals. However, only these continuous recordings are available with no labeled trials. We therefore had to create labeled trials ourselves from this data.

Manually segmenting the raw recordings into trials would take a significant amount of time and effort. Performing the required analysis for this falls outside of the scope of this research and is not within our area of expertise. Therefore, it was decided to implement a procedure that would be able to generate these segments from the raw data by analyzing the FSR signal. This signal flatlines, meaning it has zero amplitude, when the FSR is inactive and oscillates with relatively large amplitude when force is applied. These states correspond to the subject's foot being up or down respectively. This is no trivial task however. This signal is noisy and therefore, perfect zero values can not be assumed when flatline occurs. The FSR signal also shows non-stationary behaviour for an unknown reason and can even be completely useless for automated extraction for some recorded subjects. The implemented method is quite naive and should therefore be re-evaluated for future use. Due to this, labeled trials could only be extracted from the recording of a single subject, and this solely for the slow walking segment of the recording.

The procedure to automatically extract labeled trials from the raw recordings starts by segmenting the recording into parts where the subject is walking at slow, medium and fast speeds respectively. Since no reliable segmentation could be achieved for other walking speeds using the presented approach, only the slow

walking segment is further processed. The next step will attempt to place markers when the FSR signal goes into flatline to indicate that the subjects foot goes up at that time point. This is achieved by computing the variance of the signal over a window of samples. After filtering the signal with a low-pass filter, a histogram from all windows of the recording segment is constructed. The upper bound of the histogram bin containing the smallest values is then chosen as a threshold to indicate whether the signal in this window is a flatline. If variance in a window is below the threshold it is considered to be part of a flatline signal. The index from the first sample of the window is then recorded for windows with smallest variance. To avoid placing multiple markers in a single flatline segment, a normal distribution is built for the period between two consecutive foot up events that were identified in this manner. Markers that do not fall within an interval of two standard deviations of the median period, i.e., the 95 percent confidence interval, are rejected. Finally, any missing markers are padded, also based on the median period, by sampling from a normal distribution for the period that was constructed from the previously inserted markers.

Once all markers identifying foot up events have been computed, they are used to extract trials of fixed size from the raw data. Firstly, the data is split into EMG and EEG data and the number of samples for the offset, given the expected delay time between EEG and EMG is computed. Consecutively, the EEG and EMG signals are aligned by removing an amount of samples corresponding to the size of the computed offset from the last EEG samples. The EMG is then shifted back by removing the first offset size samples from the data. This results in aligned signal segments of the same length for both EEG and EMG. The delay was chosen to be 25 ms as this is the mean and most regularly occurring value in literature (Witham et al., 2011; Cisotto et al., 2017). (It was also recommended by researchers in the field)<sup>1</sup>. Since this delay is generally not known beforehand and could also vary between subjects, it should probably not be a fixed value. An alternative way of dealing with this issue might be to learn this delay. This could be achieved by either training a dedicated regression model or by using a neural network architecture that can handle a delay between inputs as discussed in section 4.2.2.

To construct the trials, a sliding window over the segment with no overlap between windows is used. This window will contain a fixed number of samples and is stored as a trial by assigning it a label. Three distinct labels were chosen for the trials with future online use in mind. We believe that this choice of labels could allow an online system to sample windows continuously and classify them

---

<sup>1</sup>[https://www.researchgate.net/post/How\\_to\\_interpret\\_the\\_difference\\_between\\_EEG-EMG\\_time\\_lag\\_cortico-muscular\\_conduction\\_time\\_and\\_LFP-EEG\\_time\\_lag\\_for\\_hand\\_muscles](https://www.researchgate.net/post/How_to_interpret_the_difference_between_EEG-EMG_time_lag_cortico-muscular_conduction_time_and_LFP-EEG_time_lag_for_hand_muscles)

into categories relevant to the online use case. If no marker can be found in the window, or the marker falls outside the interval consisting of 90 percent of the samples around the center of the window, it is considered that no foot up event occurs in this window. When the marker falls within this interval, but outside the interval consisting of 50 percent of the samples around the center of the window it is considered as containing part of the pattern corresponding to a foot up event. If a marker occurs within the 50 percent center samples the full pattern is expected to be contained in the window. The belief is that this approach would allow for anticipation of the next window containing relevant information if a partial pattern is detected. This should mitigate the fact that there is no guarantee that windows sampled at a regular interval will contain the full pattern necessary to identify an event. Note that the values for these intervals were chosen based on intuition, but should ideally be determined experimentally. However, this approach could not be investigated within the given time frame of this research.

In practice, the size of the windows was chosen to be 1.5 seconds long or 750 samples for a sampling rate of 500 Hz. This size is another aspect of the implemented experiments that should most probably be optimized. However, this currently brings with it the issue that the EMG generator module of the EEGPlus network would need to be modified to output predicted trials of the appropriate size. We were not able to figure out how the right kernel size and such for the different layers of this module could be determined from the EEG input size. Therefore, these layer attributes had to be chosen manually to generate output of the appropriate size.

The trials that were extracted in this manner are finally converted from microvolt to millivolt as the unit for the samples. This is a practical consideration related to PyTorch. The training trials can now be used to train a neural network model. However, as previously mentioned the windowing and labeling scheme will probably need to be optimized in the future to achieve optimal results. The dataset is highly imbalanced regarding the no foot up class. This will make training harder as there are barely any examples of a no foot up event occurring. Finding the right method and parameters to generate a balanced and informative dataset should be one of the priorities in future research. Extracting good training data from this dataset would, however, require an extensive study of the dataset that currently falls outside of the scope of this research.

### 5.1.2 Methods

To validate the EEGPlus model, an experiment was run on the data of a single subject from the recordings described in section 5.1.1. Due to the previously described challenges with extracting labeled trials from the recordings, solely the data from the slow walking signal segment was used. This resulted in 90 labeled

trials for training, validation and testing. This dataset was then split up into 60 trials for training and validation, and the remaining trials for evaluation purposes only. The chosen fraction of trials for training is 0.8, therefore resulting in 48 trials for training, 12 trials for validation and 30 for testing.

The model parameters are first initialized with the uniform Xavier initialization method (Glorot & Bengio, 2010) provided by PyTorch. The model is then trained for maximally 50 training epochs, after which the model parameters are reset to the best epoch according to accuracy on the validation set. The model is then further trained until it reaches the same accuracy on the combined set of the training and validation sets. Due to the possibility of early stopping and the second training run after reset, the final number of epochs can vary between training runs. It is not precisely clear what some of the exact stopping conditions are in the experimental loop provided by the Braindecode toolbox. The documentation and code indicate that it should run for the given number of epochs. However, the conditions for early stopping could not be identified. This might be due to a bug and should probably be investigated before performing further experiments in future work.

The batch size for the number of training trials that are evaluated at the same time for a single epoch was chosen to be 16. This seemed as a good trade off between a batch size that is large enough, while still allowing multiple batches given the restricted number of samples. For the optimizer that handles the weight updates of our network, Adam (Kingma & Ba, 2014) was chosen. For the classification loss the negative log likelihood was chosen in both models while the EEGPlus model adds the inverse cross correlation loss to it.

Both the EEGPlus and ShallowEEG models where trained in this manner, while the loss, accuracy,  $F_1$ -score and Cohen's kappa score where measured based on the model predictions at every epoch for each subset of the data. We included these last two scores in addition to the regular accuracy score, as they usually are good alternatives to raw accuracy. These measures often provide better insight into the types of mistakes a model makes. The  $F_1$ -score, or simply F-score, is the harmonic mean of the precision and recall as shown in equation 5.1.

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (5.1)$$

In a binary classification problem, precision refers to the number of examples that were classified correctly, or true positives, over the number of examples that truly belonged to that class. Recall is the number of true positives over the number of examples classified as positive. For multiple categories these measures can be computed by taking the average over between-class scores. The F-score contrasts accuracy that simply counts the number of correctly classified examples over the total number of examples.

Cohen's kappa is a statistic that measures how well the model predictions match

the target outputs. In essence it is a measure of agreement between two raters for the same problem. It takes into account the probability of a prediction matching its target by chance. The formula for the kappa statistic is given in equation 5.2

$$\kappa = \frac{p_o - p_e}{1 - p_e} \quad (5.2)$$

where  $p_o$  is the observed agreement between the predictions and target, i.e. accuracy, and  $p_e$  is the hypothetical probability of chance agreement. The formula for chance agreement is given as  $p_e = \frac{1}{N^2} \sum_k n_{k1}n_{k2}$  where  $N$  is the number of items and  $n_{ki}$  is the number of times category  $k$  occurs in the predictions of rater  $i$ .

A final measure that is computed during training, is the confusion matrix. It can be considered as a table where the rows correspond to the actual class and the columns to the predicted class (or the other way around). The number of times a class is predicted when the target corresponds to a certain class is recorded for every target-prediction pair in this table. The confusion matrix gives an overview of the classification performance of a model and indicates what type of mistakes are mostly made, and which classes are harder to distinguish for a model. Precision and recall among other statistical measures, can be derived directly from this matrix. It also is customary to normalize the counts in the matrix by dividing them by the total number of classified samples, as to have a relative frequency of target-prediction occurrences.

The above measures are computed at every epoch during training for 10 training runs with both models. By aggregating all runs of the same model, an indication of general training performance of the models should be given. The stability of training can also be derived from the variance between separate training runs of the same model. If the variance is low it indicates that the model should mostly perform the same, regardless of the trained instance.

Finally, the models resulting from each run are evaluated on the whole dataset and the metric results are aggregated over all training runs. A statistical paired t-test is performed to compare the evaluation results between models over all training runs. The paired test was chosen since the values originate from both models solving the same problem for repeated training runs. An aggregated confusion matrix is additionally presented for both model to report the types of errors the trained models make, giving an indication on what aspects should be optimized in future work.

An additional experiment that could prove useful would be to evaluate a model that is trained on data from one subject with data from another subject. This is referred to as between-subjects evaluation. It would be especially interesting if a model was trained on EEG and EMG of a healthy subject, and evaluated on an amputee where EMG is not available. Is such an experiment would be succesfull,

the main goal of this research would be met. Another good experiment would be to use data that was measured with a different device to investigate whether it would be possible to use a model trained on data from one device on input originating from another device. Both cases could be investigated in conjunction with transfer learning to create a model that does not need to be re-trained for every subject.

## 5.2 Results

The results on the test set during training are reported in this section. Results for other data sets are only reported in this section if they are deemed relevant to the discussion. Figures and tables reporting results on the training and validation sets that are not reported in this section can be found in the appendices.

The measure that is utilized to evaluate how well the models are performing on the evaluated data is the loss. The loss values for the EEGPlus and ShallowEEG models during training are plotted together in figure 5.1.

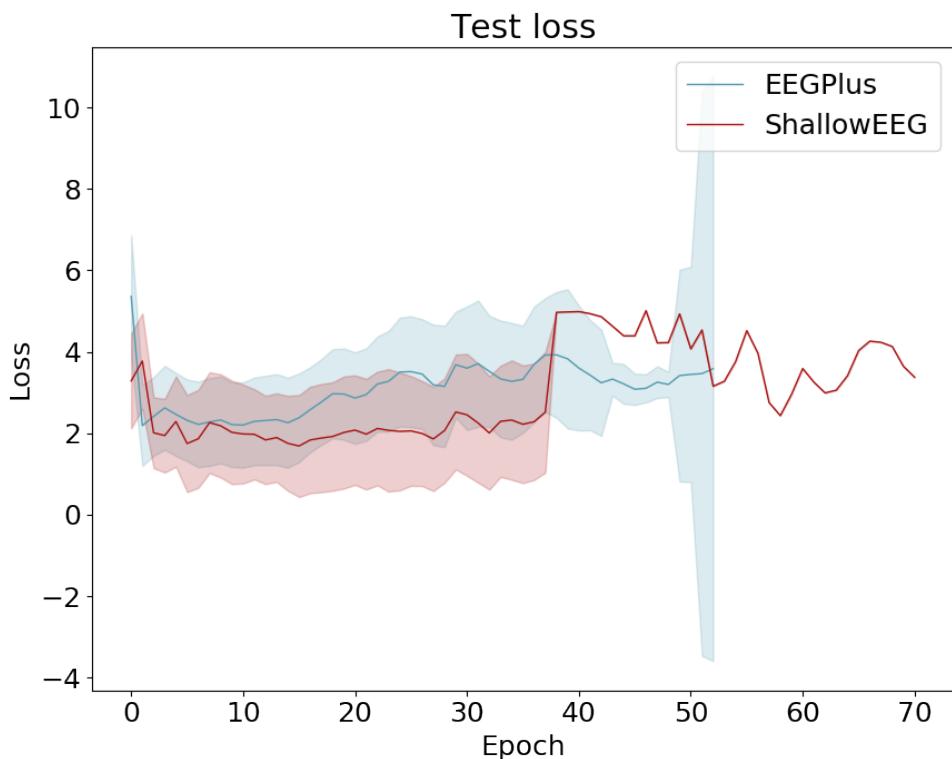


Figure 5.1: Loss value on the test set during training

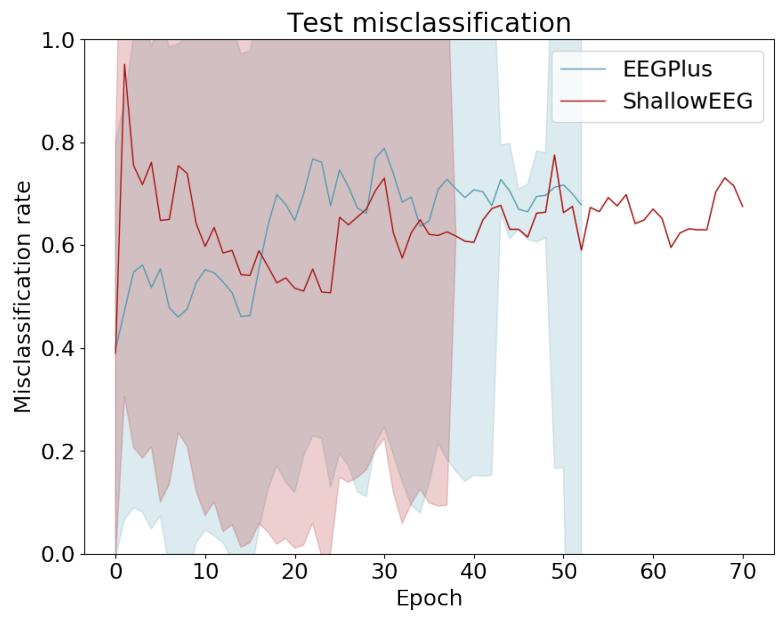
The 95% confidence interval over the different training runs is indicated by the area around the curve filled in with the same color. This interval was computed as an upper bound and lower bound that are respectively two standard deviations higher and lower than the mean value over all training runs. Note that runs can stop earlier or later than the given 50 epochs due to early stopping or the second stop mechanism respectively. The different training runs do not all have the same length due to this. The results for higher epochs will therefore be based on a subset of training runs as can be observed from the significantly increased size of the confidence interval at the end of the curve. For the ShallowEEG model there was only one run that had more than 37 epochs, therefore there was no standard deviation to compute. This is indicated by the confidence interval not being drawn anymore after epoch 37. The line representing the mean value over all runs is then plotted based on the single run that has more than 37 epochs. Additionally, note that the first epoch is 0, when the models are evaluated before training starts. This epoch merely serves as reference performance of a model that has not been trained yet.

Before the analysis of figure 5.1 can be performed, one should be aware that the loss values are not exactly the same for both models. The EEGPlus model will have the added loss value of the EMG generator module added to the chosen negative log likelihood loss. However, it can be observed that both models behave similarly and do not significantly differ when it comes to loss value on the test set. To have a better understanding of the accuracy for both models, the misclassification rate, i.e., the proportion of misclassified examples, on the test set and training set is presented in figure 5.2. It is the inverse of the accuracy, therefore a lower value is preferred.

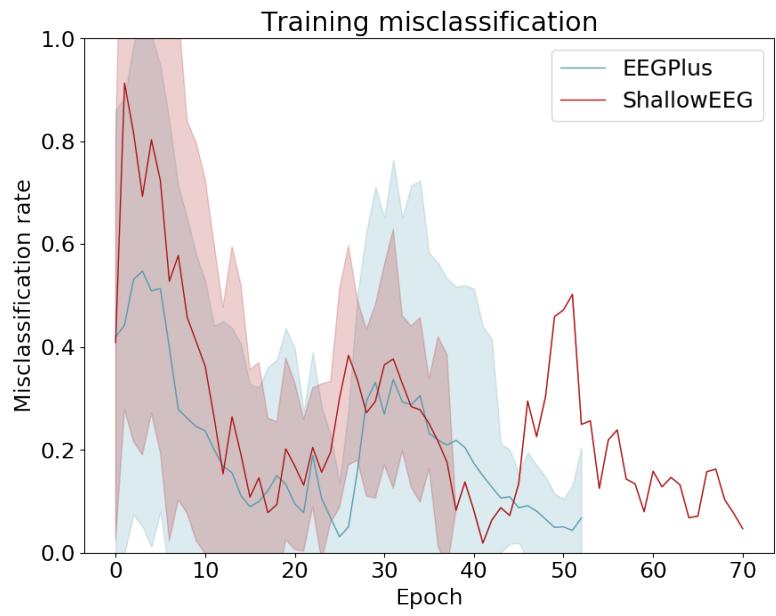
From figure 5.2a we can observe that the confidence interval for this metric is large on the test set, it even goes outside the boundaries of the graph. There does not seem to be a clear agreement between the different training runs of the same model. The test misclassification is also relatively high at the end of training, it even seems to increase from the untrained value during training. When compared to the training misclassification rate in figure 5.2b that reaches around 90% accuracy on average, some serious over fitting seems to be taking place for both models. We can also observe that there does not seem to be a significant difference between models in both cases. This matter will be further discussed and tested at evaluation time.

To have a better insight into the type of mistakes made by the models, the Kappa and F-scores on the test set are plotted over the different training epochs in figure 5.3.

Both figures 5.3a and 5.3b seem to confirm the observations from figure 5.2a. The kappa score remains around zero for most epochs and seems to end up slightly

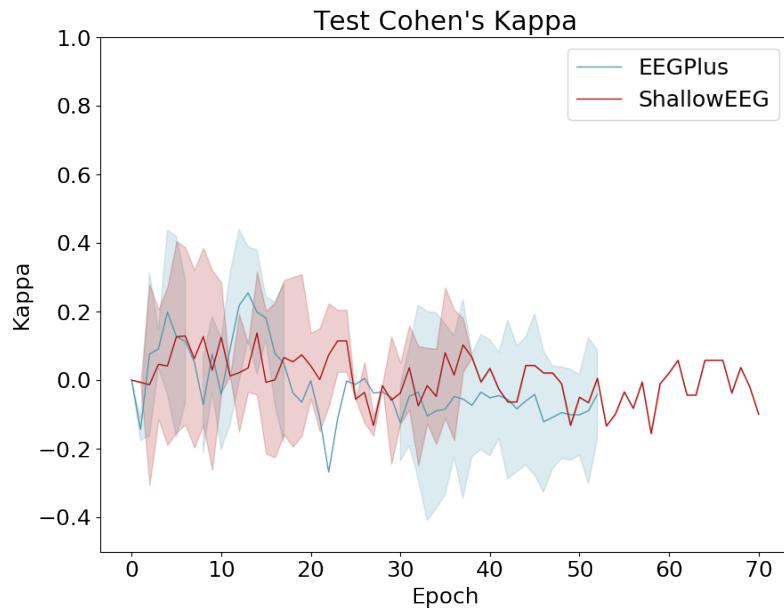


(a) Test set

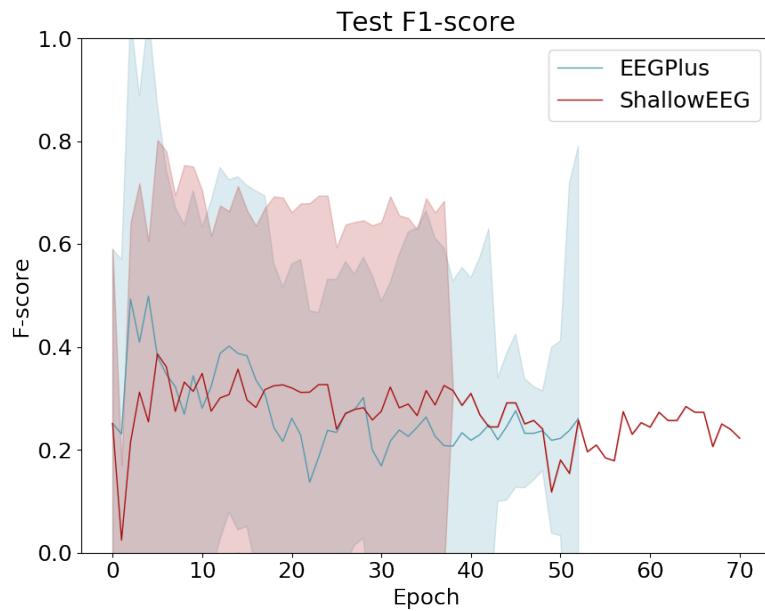


(b) Training set

Figure 5.2: Misclassification rate during training



(a) Cohen's Kappa



(b)  $F_1$ -score

Figure 5.3: Cohen's Kappa and  $F_1$ -score on the test set during training

below zero at the end of training for most training runs. The confidence intervals are relatively small for this measure. Figure 5.3a seems to indicate that agreement between target and predicted outputs is often due to chance. The test set F-score plotted in figure 5.3b shows a similar image as the misclassification rate, with large variance and low values.

The confusion matrices were also computed for every training epoch. However, only the averaged matrices for the final evaluation are presented further in this section. Confusion matrices for epochs 10, 25 and the last epoch can be found in the appendices.

To investigate the final performance of the different models, they were evaluated on the full data set. The previously used metrics were computed for the evaluation of the final model resulting from each training run. This resulted in a total of 10 accuracy, kappa and f-score values for each model. The values for every evaluated model of the same type were then averaged over all training runs to have an aggregate value over all models resulting from one of the training runs. The values for both models are reported in table 5.1 together with the standard deviation over all training runs. The p-values resulting from a statistical paired t-test for every measure are additionally reported in the last row of the table.

<b>Model</b>	<b>Accuracy</b>	<b>Kappa</b>	<b>F-score</b>
EEGPlus	$0.6 \pm 0.034$	$0.285 \pm 0.053$	$0.498 \pm 0.035$
ShallowEEG	$0.601 \pm 0.021$	$0.330 \pm 0.036$	$0.514 \pm 0.027$
P-value	0.941	0.085	0.323

Table 5.1: Comparison of evaluation performance for mean accuracy, Kappa, and F-score over all models resulting from one of the training runs. Standard deviation included to the right of the  $\pm$  sign. The bottom row contains P-values from a paired t-test between samples from both models.

We can observe that all measured accuracy scores seem to be similar for both models. This is statistically confirmed by the high p-values for all metrics. All P-values are above 0.05 and therefore we can not reject the null hypothesis that the means are identical at 5% significance level (or with 95% confidence). We can therefore conclude that there is no significant difference between the evaluation performance of the two models. While the accuracy is not very high, the positive kappa score indicates that proper classifications are not by chance. The score is not very high however, indicating that we can do better.

These results are relatively poor for both models and confirm the supposition that the data should be further processed. The ShallowEEG model would for example get an accuracy of 95% under similar conditions, on the task of identifying

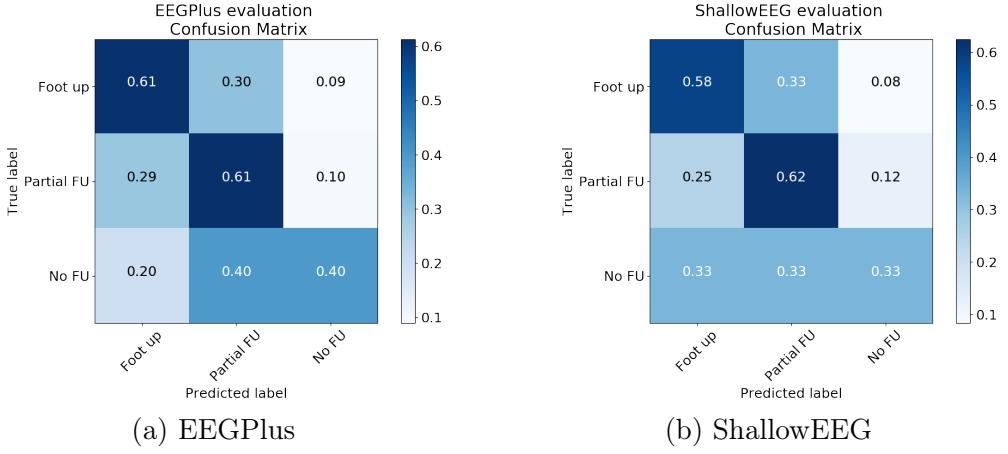


Figure 5.4: Confusion matrices for the outputted predictions during evaluation on the whole dataset.

hand movements, during preliminary experiments. However, these experiments were performed with sample data coming from an EEG only dataset provided by MNE.

To observe which classes were mostly confused during evaluation the normalized confusion matrices for both models are shown in figure 5.4. From these confusion matrices we can see that for the classes Foot Up (FU) and Partial FU, there is an agreement with the overall accuracy value from table 5.1. The diagonal values corresponding to correct classifications are close to the overall accuracy value. For the No FU class there seems to be a uniform distribution over predicted classes for the ShallowEEG model and a tie between Partial FU and No FU predictions for EEGPlus. This is probably due in part to the imbalance of trials with a No FU target label. The observations from figure 5.4a seem to indicate a difficulty in distinguishing a partial foot up event from the absence of one. This further confirms that improved training data quality would probably lead to better performance.

We can conclude that while not improving the classification performance compared to the state of the art, using EMG is certainly not detrimental. Further experimentation is needed to investigate whether there are significant differences between the models on more well-established tasks, such as hand movement classification. We believe that optimizing the different modules and investigating alternate designs of the EEGPlus model should improve performance compared with the ShallowEEG model.

# Chapter 6

## Analysis & Conclusion

To classify EEG signals into corresponding movements for a brain computer interface (BCI) application, a robust and fast Machine Learning (ML) model is required. This research investigated a deep learning model named EEGPlus that is trained on EEG data in conjunction with EMG data. The purpose of this model is to learn a correlation between EEG and EMG inputs that will be useful in improving classification of EEG input only. This is because additional inputs can not be assumed to always be available, while relevant EEG information should always be available as long as a person is conscious. Finding the correlation between EEG and EMG should allow for improved identification of relevant information in noisy EEG data.

The performance of the implemented EEGPlus model was compared with the state-of-the-art ShallowEEG model that furthermore served as the basis for the EEGPlus model. No difference in performance could be confirmed during the performed experiments. Hypothesis as to why theses are the results are discussed in sections 6.1, 6.2, 6.3 and 6.4.

### 6.1 Hypothesis 1: Data

The data provided by the physiology department of the VUB was given as a raw recording. Due to the absence of high-quality labeled training data, we developed a automated method to extract the gait cycle of the subjects. As explained in section 5.1.1, we chose a naive automated method that used the signal from a force sensing resistor (FSR) to extract and label trials. This method might have resulted in some ambiguity in the training data and a dataset that only contained a few samples for one of the classes. In figure 6.1, we can observe how our automated method extracted the foot up events of two subjects. Figure 6.1a shows that when the signal has a high signal-to-noise ratio it is possible to successfully extract markers

for the events. In contrast figure 6.1b shows a large set of false positive labels with this method.

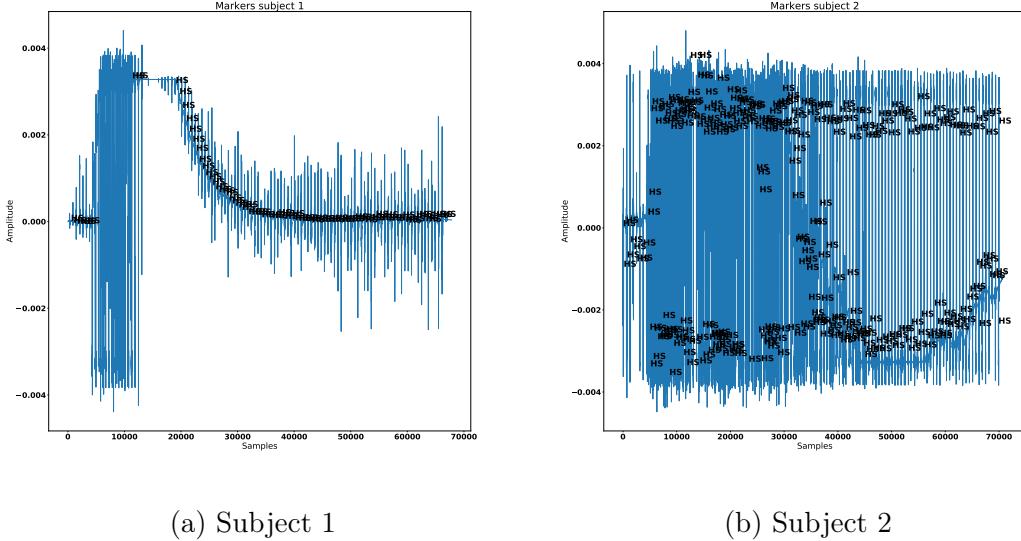


Figure 6.1: Confusion matrices for the outputted predictions during evaluation on the whole dataset.

Based on this analysis, we hypothesize that it becomes more complex for the EEGPlus model to find correlations between EEG and EMG.

## 6.2 Hypothesis 2: Cost Function (optimization)

In section 4.2.1 we described the architecture of the EEGPlus model, where we showed that the EMG and EEG signals are joined by a summation as a single cost function to minimize the overall cost of the network. However, different strategies on how to combine cost functions may prove more appropriate. There is a distinct lack of available literature on how to combine cost functions coming from different signals in neural networks.

In addition, we notice that a further research on how to aggregate the gradients coming from different modules in the network is required. For example, in figure 6.2 we hypothesize that if the gradient directions of the EEG and EMG do not agree, then the network might never converge on the joint objective. Or it may more sensitive to getting stuck in local optima.

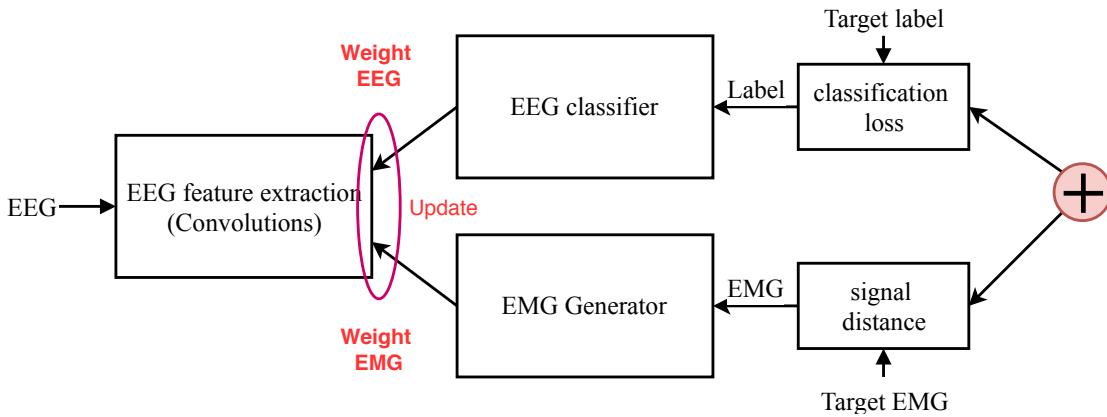


Figure 6.2: This figure shows the limitation of back-propagating the gradients in the EEGPlus network. The highlighted sections (in red) show potential locations where the network may encounter problems.

### 6.3 Hypothesis 3: EMG Generator

In section 4.2.1 we discussed how we use Generative Adversarial Networks (GAN) to generate a target EMG signal from EEG features. This analysis is related to section 6.1, as we may be generating more false positive signals than true positives. As a result, the network may not be able to exploit the existing correlations between the EEG and EMG data. For example, if the EEGPlus model is biased towards the amount of rich information found in the EEG signal, then it will disregard the EMG signal by setting most of the weights of the EMG module to zero. This hypothesis is aligned with the results shown in figure 5.2 where the accuracy of the EEGPlus and ShallowEEG models is shown to be similar. Due to time constraints, it was not possible to perform this analysis. It is important to note that generating EMG signals with a neural network was never attempted before in the literature. Achieving this is not a trivial task.

### 6.4 Hypothesis 4: EEG-EMG Delay

In section 5.1.1 we discussed that our input space is divided into three signal types, EEG, EMG and FSR. The FSR was used to label the data as discussed in 6.1. Then, we use these labels to window the signals in the EMG data. These windows are the markers that indicate the presence of an event in the gait cycle of the subject. To annotate the EEG signals we shifted the EMG window 25 ms in the past, as shown in figure 6.3. This number was taken from the literature on how long it takes for a signal in the brain to travel to the muscles. Based on this setting,

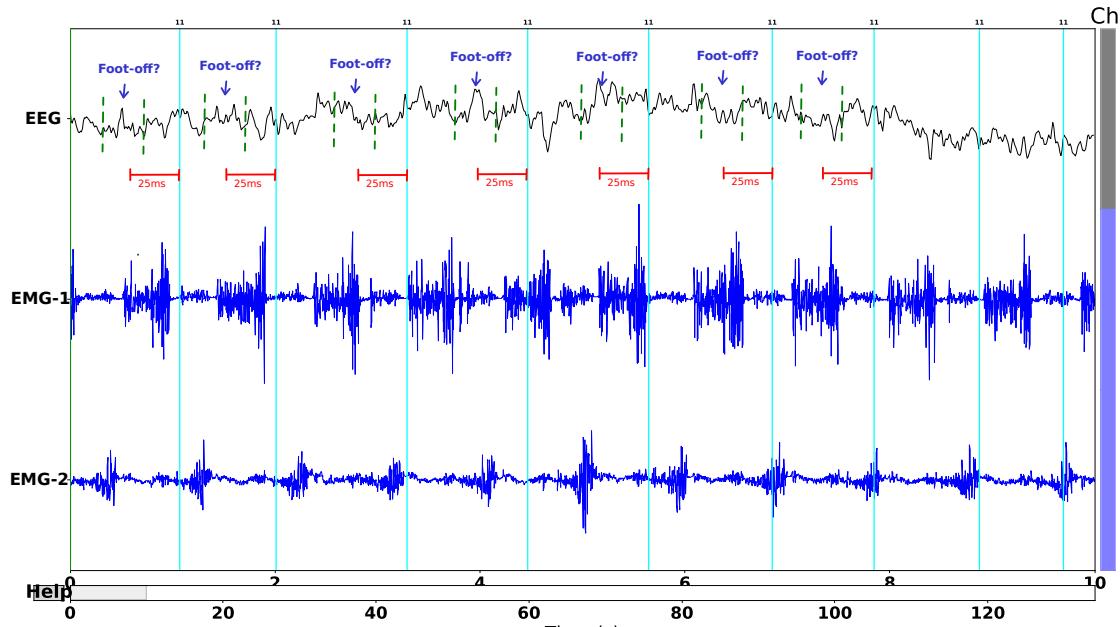


Figure 6.3: This figure shows the delay constrain between EEG and EMG

we are assuming that the delayed window in reference to the EMG, contains the correct information related to the event in the EEG signal.

However, this assumption is not guaranteed, as this delay varies from subject to subject (see section 2.2.2). As a result, we are also including false positives on the input space of the network, increasing the complexity for classification.

## 6.5 Conclusion

We believe that the EEGPlus model could be improved to surpass the ShallowEEG model in performance, given the right design. There are still several alternative designs to be tested and optimizations of the current model that could be made. We believe that improving the performance is a matter of finding the right architecture with the right parameters. Our research resulted in a set of small contributions to different fields of research. For the field of neurobiology, and other fields concerned with biosignal processing, we developed some basic tools that could facilitate data analysis. Though the current version is relatively naive and not yet performant enough for use in practical applications. We believe that with a dedicated development effort, the initial procedures that were created for data processing could be used for automated extraction and analysis, after some improvement. The delay between EEG and EMG activity related to a movement was also instigated. We proposed multiple potential approaches that could learn the delay between EEG

and EMG. While no conclusive evidence could be identified, we believe that future research should benefit from this preliminary study.

Related to the field of artificial intelligence, we performed a preliminary study of approaches to multimodal deep learning. This is an area of research that has not been widely investigated yet. In relation to this, we also explored the idea of using multimodal deep learning in the context of hybrid BCI. An additional contribution related to hybrid BCI, is the fact that our model was designed in such a way to allow for only EEG to be used after training. We laid the basis for further research into the combination of several biosignals through deep learning, with the possibility to remove secondary input signals after deployment of the model.

Suggested future research therefore includes the following proposals. As previously mentioned, one of the priorities should probably be the acquisition of high quality training data that consists of labeled trials. This could be achieved either by having the current data be processed by an expert in the field, improving the automated trial extraction procedure or performing a new dedicated experiment. A new experiment would then be able to take into account that labeled trials are the final data, as opposed to continuous recordings. Trials could be created during the experiment itself in this case. Alternatively, improving the procedure that was used for data processing in our experiments could facilitate analysis of data that was acquired as a continuous recording. This would eliminate the need for manual segmentation and labeling of the data into trials. Regardless of the chosen approach, the assistance of a domain expert would be required to achieve good results.

Other suggested future work entails optimizing the EEGPlus model. However, this should only take place when the previously proposed work has been completed and high quality data is available. This optimization could be achieved either by investigating alternative implementations of the individual modules that make up the model. Or by implementing one of the proposed alternative designs presented in sections 4.2.2 for comparison with the current implementation. In either case, hyperparameter configuration should subsequently be optimized. We would suggest the use of optimization algorithms over the hyperparameter space for this purpose.

As discussed in sections 6.2, investigating alternative cost functions would also be a good idea. Especially a way to combine cost functions from different signals. The delay between EEG and EMG is another point that should be investigated as discussed in section 6.4.

We also recommend that more extensive experimentation be performed on the current model once it is able to surpass the performance of the state-of-the-art. Alternative implementations should also be extensively validated to investigate

potential benefits in certain cases, such as between-subject. Proposed experiments where discussed in section 5.1.2.

Finally, we propose that the conceptual online system described in section 4.3 be implemented. Investigating an online system that has the proposed attributes could benefit the development of an inexpensive, but reliable system. Such a system could then be used to provide prosthesis in developing countries where commercial devices are often unavailable for example. The main advantage of such a system would be that it is constructed from freely available open source components. Further development in an open source manner would most certainly allow for the advancement of BCI research in general.

In conclusion, we can say that while the results of this research were not able to improve upon the state-of-the-art, we laid the foundations for future research into hybrid BCI systems. This study could be considered as a pilot study into investigating the matters that were discussed. Further research should benefit from our findings in achieving the final goal of our research. Creating a hybrid BCI system that allows for additional inputs next to EEG to improve performance, but that can be used with EEG input only.

# Bibliography

- Alte, D. (2017). *Control of a robotic arm using a low-cost bci*. Retrieved from [http://www.ausy.tu-darmstadt.de/uploads/Theses/Abschlussarbeiten/daniel\\_alte\\_BSc\\_thesisc.pdf](http://www.ausy.tu-darmstadt.de/uploads/Theses/Abschlussarbeiten/daniel_alte_BSc_thesisc.pdf)
- Atzori, M., Cognolato, M., & Müller, H. (2016). Deep learning with convolutional neural networks applied to electromyography data: A resource for the classification of movements for prosthetic hands. *Front Neurorobot*, 10, 9. Retrieved from <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC5013051/> (27656140[pmid]) doi: 10.3389/fnbot.2016.00009
- Bailey, T. L., & Elkan, C. (1994). Fitting a mixture model by expectation maximization to discover motifs in bipolymers.
- Blain, S., Mihailidis, A., & Chau, T. (2008). Assessing the potential of electrodermal activity as an alternative access pathway. *Medical Engineering and Physics*, 30(4), 498-505. Retrieved from <http://dx.doi.org/10.1016/j.medengphy.2007.05.015> doi: 10.1016/j.medengphy.2007.05.015
- Blankertz, B., Dornhege, G., Krauledat, M., Muller, K. R., Kunzmann, V., Losch, F., & Curio, G. (2006). The berlin brain-computer interface: Eeg-based communication without subject training. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14(2), 147-152. doi: 10.1109/TNSRE.2006.875557
- Boto, E., Holmes, N., Leggett, J., Roberts, G., Shah, V., Meyer, S. S., ... Brookes, M. J. (2018). Moving magnetoencephalography towards real-world applications with a wearable system. *Nature*, 555, 657 - 661. Retrieved from <http://dx.doi.org/10.1038/nature26147>
- Bourlard, H., & Kamp, Y. (1988). Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics*, 59(4), 291–294. Retrieved from <https://doi.org/10.1007/BF00332918> doi: 10.1007/BF00332918
- Bower, J. M., Beeman, D., & Hucka, M. (2003). The genesis simulation system.

- Brunner, C., Scherer, R., Graimann, B., Supp, G., & Pfurtscheller, G. (2006). Online control of a brain-computer interface using phase synchronization. *IEEE Transactions on Biomedical Engineering*, 53(12), 2501-2506. doi: 10.1109/TBME.2006.881775
- Bruns, A. (2004). Fourier-, hilbert- and wavelet-based signal analysis: are they really different approaches? *Journal of Neuroscience Methods*, 137(2), 321 - 332. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0165027004001098> doi: <https://doi.org/10.1016/j.jneumeth.2004.03.002>
- Campbell, I. G. (2009). Eeg recording and analysis for sleep research. *Current protocols in neuroscience / editorial board, Jacqueline N. Crawley ... [et al.], CHAPTER*, Unit10.2-Unit10.2. Retrieved from <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2824445/> (19802813[pmid]) doi: 10.1002/0471142301.ns1002s49
- Chai, X., Wang, Q., Zhao, Y., Liu, X., Bai, O., & Li, Y. (2016). Unsupervised domain adaptation techniques based on auto-encoder for non-stationary eeg-based emotion recognition. *Computers in Biology and Medicine*, 79(Supplement C), 205 - 214. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0010482516302797> doi: <https://doi.org/10.1016/j.combiomed.2016.10.019>
- Chéron, G., Duvinage, M., De Saedeleer, C., Castermans, T., Bengoetxea, A., Petieau, M., ... Ivanenko, Y. (2012). From spinal central pattern generators to cortical network: integrated bci for walking rehabilitation. *Neural plasticity*, 2012. doi: 10.1155/2012/375148
- Chung, J., Ahn, S., & Bengio, Y. (2016). Hierarchical multiscale recurrent neural networks. *CoRR, abs/1609.01704*. Retrieved from <http://arxiv.org/abs/1609.01704>
- Cisotto, G., Michieli, U., & Badia, L. (2017). A coherence study on EEG and EMG signals. *ArXiv e-prints*.
- Cooley, J. W., Lewis, P. A. W., & Welch, P. D. (1969). The fast fourier transform and its applications. *IEEE Transactions on Education*, 12(1), 27-34. doi: 10.1109/TE.1969.4320436
- Cooper, J. K. (1986). Electrocardiography 100 years ago. *New England Journal of Medicine*, 315(7), 461-464. Retrieved from <https://doi.org/10.1056/NEJM198608143150722> (PMID: 3526152) doi: 10.1056/NEJM198608143150722

- Coyle, S. M., Ward, T. E., & Markham, C. M. (2007). Brain-computer interface using a simplified functional near-infrared spectroscopy system. *Journal of Neural Engineering*, 4(3), 219. Retrieved from <http://stacks.iop.org/1741-2552/4/i=3/a=007>
- Darvas, F., Miller, K. J., Rao, R. P. N., & Ojemann, J. G. (2009). Nonlinear phase-phase cross-frequency coupling mediates communication between distant sites in human neocortex. *Journal of Neuroscience*, 29(2), 426–435. Retrieved from <http://www.jneurosci.org/content/29/2/426> doi: 10.1523/JNEUROSCI.3688-08.2009
- Dauwels, J., Vialatte, F., & Cichocki, A. (2010). Diagnosis of alzheimers disease from eeg signals: Where are we standing? *Current Alzheimer Research*, 7(6), 487-505. Retrieved from <http://www.eurekaselect.com/node/86614/article> doi: 10.2174/156720510792231720
- Donchin, E., Spencer, K. M., & Wijesinghe, R. (2000). The mental prosthesis: assessing the speed of a p300-based brain-computer interface. *IEEE Transactions on Rehabilitation Engineering*, 8(2), 174-179. doi: 10.1109/86.847808
- Dulantha, L. T., Kenbu, T., Yoshiaki, H., & Kazuo, K. (2013). Towards hybrid eeg-emg-based control approaches to be used in bio-robotics applications: Current status, challenges and future directions. *Paladyn, Journal of Behavioral Robotics*, 4, 147. Retrieved from <https://www.degruyter.com/view/j/pjbr.2013.4.issue-2/pjbr-2013-0009/pjbr-2013-0009.xml> doi: 10.2478/pjbr-2013-0009
- Dumoulin, V., & Visin, F. (2016). A guide to convolution arithmetic for deep learning. *ArXiv e-prints*.
- Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14(2), 179-211. Retrieved from [https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog1402\\_1](https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog1402_1) doi: 10.1207/s15516709cog1402\\_1
- Fernando, C., Banarse, D., Blundell, C., Zwols, Y., Ha, D., Rusu, A. A., ... Wierstra, D. (2017). Pathnet: Evolution channels gradient descent in super neural networks. *CoRR*, *abs/1701.08734*. Retrieved from <http://arxiv.org/abs/1701.08734>
- Ferrez, P. W., & del R. Millan, J. (2008). Error-related eeg potentials generated during simulated brain computer interaction. *IEEE Transactions on Biomedical Engineering*, 55(3), 923-929. doi: 10.1109/TBME.2007.908083

- Frishman, L. J. (2013). Chapter 7 - electrogenesis of the electroretinogram. In S. J. Ryan et al. (Eds.), *Retina (fifth edition)* (Fifth Edition ed., p. 177 - 201). London: W.B. Saunders. Retrieved from <https://www.sciencedirect.com/science/article/pii/B9781455707379000072> doi: <https://doi.org/10.1016/B978-1-4557-0737-9.00007-2>
- Gallegos-Ayala, G., Furdea, A., Takano, K., Ruf, C. A., Flor, H., & Birbaumer, N. (2014). Brain communication in a completely locked-in patient using bedside near-infrared spectroscopy. *Neurology*, 82(21), 1930-1932. Retrieved from <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4049706/> (NEUROLOGY2013546432[PII]) doi: 10.1212/WNL.0000000000000449
- Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 249–256).
- Glorot, X., Bordes, A., & Bengio, Y. (2011). Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th international conference on machine learning (icml-11)* (pp. 513–520).
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. Retrieved from <http://www.deeplearningbook.org>
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2014). Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 27* (pp. 2672–2680). Curran Associates, Inc. Retrieved from <http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf>
- Graimann, B., Huggins, J. E., Levine, S. P., & Pfurtscheller, G. (2004). Toward a direct brain interface based on human subdural recordings and wavelet-packet analysis. *IEEE Transactions on Biomedical Engineering*, 51(6), 954-962. doi: 10.1109/TBME.2004.826671
- Gramfort, A., Luessi, M., Larson, E., Engemann, D., Strohmeier, D., Brodbeck, C., ... Hämäläinen, M. (2013). Meg and eeg data analysis with mne-python. *Frontiers in Neuroscience*, 7, 267. Retrieved from <https://www.frontiersin.org/article/10.3389/fnins.2013.00267> doi: 10.3389/fnins.2013.00267

- Gramfort, A., Luessi, M., Larson, E., Engemann, D. A., Strohmeier, D., Brodbeck, C., ... Hämäläinen, M. S. (2014). Mne software for processing meg and eeg data. *NeuroImage*, 86, 446 - 460. Retrieved from <http://www.sciencedirect.com/science/article/pii/S1053811913010501> doi: <https://doi.org/10.1016/j.neuroimage.2013.10.027>
- Graves, A., r. Mohamed, A., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *2013 ieee international conference on acoustics, speech and signal processing* (p. 6645-6649). doi: 10.1109/ICASSP.2013.6638947
- Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5), 602 - 610. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0893608005001206> (IJCNN 2005) doi: <https://doi.org/10.1016/j.neunet.2005.06.042>
- Hartline, D. K. (2008). What is myelin? *Neuron glia biology*, 4(2), 153–163.
- He, S., Yu, T., Gu, Z., & Li, Y. (2017). A hybrid bci web browser based on eeg and eog signals. In *2017 39th annual international conference of the ieee engineering in medicine and biology society (embc)* (p. 1006-1009). doi: 10.1109/EMBC.2017.8036996
- Hines, M. L., & Carnevale, N. T. (1997). The neuron simulation environment. *Neural computation*, 9(6), 1179–1209.
- Hochberg, L. R., Serruya, M. D., Friehs, G. M., Mukand, J. A., Saleh, M., Caplan, A. H., ... Donoghue, J. P. (2006). Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature*, 442, 164 – 171. Retrieved from <http://dx.doi.org/10.1038/nature04970> (Article) doi: 10.1038/nature04970
- Jayaram, V., Alamgir, M., Altun, Y., Scholkopf, B., & Grosse-Wentrup, M. (2016). Transfer learning in brain-computer interfaces. *IEEE Computational Intelligence Magazine*, 11(1), 20-31. doi: 10.1109/MCI.2015.2501545
- Jiang, J., Zhou, Z., Yin, E., Yu, Y., & Hu, D. (2014). Hybrid brain-computer interface (bci) based on the eeg and eog signals. *Bio-medical materials and engineering*, 24(6), 2919–2925. Retrieved from <https://content.iospress.com/articles/bio-medical-materials-and-engineering/bme1111> doi: 10.3233/BME-141111

- Khezri, M., & Jahed, M. (2011). A neuro-fuzzy inference system for semg-based identification of hand motion commands. *IEEE Transactions on Industrial Electronics*, 58(5), 1952-1960. doi: 10.1109/TIE.2010.2053334
- Kiguchi, K., & Hayashi, Y. (2012). A study of emg and eeg during perception-assist with an upper-limb power-assist robot. In *2012 ieee international conference on robotics and automation* (p. 2711-2716). doi: 10.1109/ICRA.2012.6225027
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR, abs/1412.6980*. Retrieved from <http://arxiv.org/abs/1412.6980>
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 25* (pp. 1097–1105). Curran Associates, Inc. Retrieved from <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>
- Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324. doi: 10.1109/5.726791
- Ledig, C., Theis, L., Huszar, F., Caballero, J., Aitken, A. P., Tejani, A., ... Shi, W. (2016). Photo-realistic single image super-resolution using a generative adversarial network. *CoRR, abs/1609.04802*. Retrieved from <http://arxiv.org/abs/1609.04802>
- Leeb, R., Lee, F., Keinrath, C., Scherer, R., Bischof, H., & Pfurtscheller, G. (2007). Brain computer communication: Motivation, aim, and impact of exploring a virtual apartment. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 15(4), 473-482. doi: 10.1109/TNSRE.2007.906956
- Leeb, R., Sagha, H., Chavarriaga, R., & del R Millán, J. (2011). A hybrid brain-computer interface based on the fusion of electroencephalographic and electromyographic activities. *Journal of neural engineering*, 8(2), 025011.
- Li, J., Struzik, Z. R., Zhang, L., & Cichocki, A. (2014). Feature learning from incomplete EEG with denoising autoencoder. *CoRR, abs/1410.0818*. Retrieved from <http://arxiv.org/abs/1410.0818>
- Lugger, K., Flotzinger, D., Schlögl, A., Pregenzer, M., & Pfurtscheller, G. (1998). Feature extraction for on-line eeg classification using principal components and linear discriminants. *Medical and Biological Engineering and Computing*, 36(3), 309–314. Retrieved from <https://doi.org/10.1007/BF02522476> doi: 10.1007/BF02522476

- Makeig, S., Bell, A. J., Jung, T.-P., & Sejnowski, T. J. (1996). Independent component analysis of electroencephalographic data. In *Advances in neural information processing systems* (pp. 145–151).
- Markovic, M., Dosen, S., Popovic, D., Graimann, B., & Farina, D. (2015). Sensor fusion and computer vision for context-aware control of a multi degree-of-freedom prosthesis. *Journal of Neural Engineering*, 12(6), 066022. Retrieved from <http://stacks.iop.org/1741-2552/12/i=6/a=066022>
- Marulanda, F. G. (2013). *Interpreting eeg signals using artificial intelligence* (Unpublished master's thesis). Vrije Universiteit Brussel.
- Mathôt, S., Melmi, J.-B., van der Linden, L., & Van der Stigchel, S. (2016). The mind-writing pupil: A human-computer interface based on decoding of covert attention through pupillometry. *PLoS One*, 11(2), e0148805. Retrieved from <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4743834/> (PONE-D-15-47494[PII]) doi: 10.1371/journal.pone.0148805
- Maynard, E. M., Nordhausen, C. T., & Normann, R. A. (1997). The utah intracortical electrode array: A recording structure for potential brain-computer interfaces. *Electroencephalography and Clinical Neurophysiology*, 102(3), 228 - 239. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0013469496951760> doi: [https://doi.org/10.1016/S0013-4694\(96\)95176-0](https://doi.org/10.1016/S0013-4694(96)95176-0)
- McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4), 115–133. Retrieved from <https://doi.org/10.1007/BF02478259> doi: 10.1007/BF02478259
- McFarland, D. J., McCane, L. M., David, S. V., & Wolpaw, J. R. (1997). Spatial filter selection for eeg-based communication. *Electroencephalography and Clinical Neurophysiology*, 103(3), 386 - 394. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0013469497000222> doi: [https://doi.org/10.1016/S0013-4694\(97\)00022-2](https://doi.org/10.1016/S0013-4694(97)00022-2)
- Mellinger, J., Schalk, G., Braun, C., Preissl, H., Rosenstiel, W., Birbaumer, N., & Kübler, A. (2007). An meg-based brain–computer interface (bci). *NeuroImage*, 36(3), 581 - 593. Retrieved from <http://www.sciencedirect.com/science/article/pii/S1053811907002261> doi: <https://doi.org/10.1016/j.neuroimage.2007.03.019>

- Millan, J. R., & Mourino, J. (2003). Asynchronous bci and local neural classifiers: an overview of the adaptive brain interface project. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2), 159-161. doi: 10.1109/TNSRE.2003.814435
- Miller, K. J., denNijs, M., Shenoy, P., Miller, J. W., Rao, R. P., & Ojemann, J. G. (2007). Real-time functional brain mapping using electrocorticography. *NeuroImage*, 37(2), 504 - 507. Retrieved from <http://www.sciencedirect.com/science/article/pii/S105381190700417X> doi: <https://doi.org/10.1016/j.neuroimage.2007.05.029>
- Morlet, J. (1983). Sampling theory and wave propagation. In C. H. Chen (Ed.), *Issues in acoustic signal — image processing and recognition* (pp. 233–261). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Muller, K. R., Anderson, C. W., & Birch, G. E. (2003). Linear and nonlinear methods for brain-computer interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2), 165-169. doi: 10.1109/TNSRE.2003.814484
- Müller-Putz, G. R., Scherer, R., Brauneis, C., & Pfurtscheller, G. (2005). Steady-state visual evoked potential (ssvep)-based communication: impact of harmonic frequency components. *Journal of Neural Engineering*, 2(4), 123. Retrieved from <http://stacks.iop.org/1741-2552/2/i=4/a=008>
- Naeem, M., Brunner, C., Leeb, R., Graimann, B., & Pfurtscheller, G. (2006). Separability of four-class motor imagery data using independent components analysis. *Journal of Neural Engineering*, 3(3), 208. Retrieved from <http://stacks.iop.org/1741-2552/3/i=3/a=003>
- Nasrabadi, N. M. (2007). Pattern recognition and machine learning. *Journal of Electronic Imaging*, 16. Retrieved from <https://doi.org/10.1117/1.2819119> doi: 10.1117/1.2819119
- Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., & Ng, A. Y. (2011). Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (icml-11)* (pp. 689–696).
- Novak, D., & Riener, R. (2015). A survey of sensor fusion methods in wearable robotics. *Robotics and Autonomous Systems*, 73, 155 - 170. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0921889014001705> (Wearable Robotics) doi: <https://doi.org/10.1016/j.robot.2014.08.012>

- Obermaier, B., Guger, C., Neuper, C., & Pfurtscheller, G. (2001). Hidden markov models for online classification of single trial eeg data. *Pattern Recognition Letters*, 22(12), 1299 - 1309. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0167865501000757> (Selected Papers from the 11th Portuguese Conference on Pattern Recognition - RECPAD2000) doi: [https://doi.org/10.1016/S0167-8655\(01\)00075-7](https://doi.org/10.1016/S0167-8655(01)00075-7)
- Oostenveld, R., & Praamstra, P. (2001). The five percent electrode system for high-resolution eeg and erp measurements. *Clinical Neurophysiology*, 112(4), 713-719. Retrieved from [http://dx.doi.org/10.1016/S1388-2457\(00\)00527-7](http://dx.doi.org/10.1016/S1388-2457(00)00527-7) doi: 10.1016/S1388-2457(00)00527-7
- Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., ... Lerer, A. (2017). Automatic differentiation in pytorch. In *Nips-w*.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Peters, B. O., Pfurtscheller, G., & Flyvbjerg, H. (1998). Mining multi-channel eeg for its information content: an ann-based method for a brain-computer interface. *Neural Networks*, 11(7), 1429 - 1433. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0893608098000604> doi: [https://doi.org/10.1016/S0893-6080\(98\)00060-4](https://doi.org/10.1016/S0893-6080(98)00060-4)
- Pfurtscheller, G., Allison, B., Bauernfeind, G., Brunner, C., Solis Escalante, T., Scherer, R., ... Birbaumer, N. (2010). The hybrid bci. *Frontiers in Neuroscience*, 4, 3. Retrieved from <https://www.frontiersin.org/article/10.3389/fnpro.2010.00003> doi: 10.3389/fnpro.2010.00003
- Pfurtscheller, G., Leeb, R., Friedman, D., & Slater, M. (2008). Centrally controlled heart rate changes during mental practice in immersive virtual environment: A case study with a tetraplegic. *International Journal of Psychophysiology*, 68(1), 1 - 5. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0167876007002486> doi: <https://doi.org/10.1016/j.ijpsycho.2007.11.003>
- Pfurtscheller, G., & Lopes da Silva, F. H. (1999). Event-related eeg/meg synchronization and desynchronization: basic principles. *Clinical Neurophysiology*, 110(11), 1842-1857. Retrieved from [http://dx.doi.org/10.1016/S1388-2457\(99\)00141-8](http://dx.doi.org/10.1016/S1388-2457(99)00141-8) doi: 10.1016/S1388-2457(99)00141-8

- Pfurtscheller, G., Muller-Putz, G. R., Schlogl, A., Graimann, B., Scherer, R., Leeb, R., ... Neuper, C. (2006). 15 years of bci research at graz university of technology: current projects. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14(2), 205-210. doi: 10.1109/TNSRE.2006.875528
- Pregenzer, M., & Pfurtscheller, G. (1999). Frequency component selection for an eeg-based brain to computer interface. *IEEE Transactions on Rehabilitation Engineering*, 7(4), 413-419. doi: 10.1109/86.808944
- Qin, L., Ding, L., & He, B. (2004). Motor imagery classification by means of source analysis for brain-computer interface applications. *Journal of Neural Engineering*, 1(3), 135. Retrieved from <http://stacks.iop.org/1741-2552/1/i=3/a=002>
- Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR, abs/1511.06434*. Retrieved from <http://arxiv.org/abs/1511.06434>
- Ray, S., Deshpande, R., Dudani, N., & Bhalla, U. S. (2008). A general biological simulator: the multiscale object oriented simulation environment, moose. *BMC Neuroscience*, 9(1), P93.
- Ren, S., He, K., Girshick, R. B., & Sun, J. (2015). Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR, abs/1506.01497*. Retrieved from <http://arxiv.org/abs/1506.01497>
- Reynolds, D. A., & Rose, R. C. (1995). Robust text-independent speaker identification using gaussian mixture speaker models. *IEEE Transactions on Speech and Audio Processing*, 3(1), 72-83. doi: 10.1109/89.365379
- Rioult, O., & Vetterli, M. (1991). Wavelets and signal processing. *IEEE Signal Processing Magazine*, 8(4), 14-38. doi: 10.1109/79.91217
- Safri, N., & Murayama, N. (2007). Comparison of eeg-emg time delays calculated by phase estimates and inverse fft.
- Sainath, T. N., r. Mohamed, A., Kingsbury, B., & Ramabhadran, B. (2013). Deep convolutional neural networks for lvcsr. In *2013 ieee international conference on acoustics, speech and signal processing* (p. 8614-8618). doi: 10.1109/ICASSP.2013.6639347
- Sakhavi, S., & Guan, C. (2017). Convolutional neural network-based transfer learning and knowledge distillation using multi-subject data in motor imagery bci. In *2017 8th international ieee/embs conference on neural engineering (ner)* (p. 588-591). doi: 10.1109/NER.2017.8008420

- Salazar-Gomez, A. F., DelPreto, J., Gil, S., Guenther, F. H., & Rus, D. (2017). Correcting robot mistakes in real time using eeg signals. In *2017 ieee international conference on robotics and automation (icra)* (p. 6570-6577). doi: 10.1109/ICRA.2017.7989777
- Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3), 210-229. doi: 10.1147/rd.33.0210
- Sander, T. H., Preusser, J., Mhaskar, R., Kitching, J., Trahms, L., & Knappe, S. (2012). Magnetoencephalography with a chip-scale atomic magnetometer. *Biomed. Opt. Express*, 3(5), 981–990. Retrieved from <http://www.osapublishing.org/boe/abstract.cfm?URI=boe-3-5-981> doi: 10.1364/BOE.3.000981
- Scherer, R., Muller, G. R., Neuper, C., Graimann, B., & Pfurtscheller, G. (2004). An asynchronously controlled eeg-based virtual keyboard: improvement of the spelling rate. *IEEE Transactions on Biomedical Engineering*, 51(6), 979-984. doi: 10.1109/TBME.2004.827062
- Scherer, R., & Rao, R. P. (2011). Non-manual control devices: Direct brain-computer interaction. In J. Pereira (Ed.), (p. 233-250). Hershey, PA: IGI Global. doi: 10.4018/978-1-60566-206-0.ch015
- Scherer, R., Schloegl, A., Lee, F., Bischof, H., Janša, J., & Pfurtscheller, G. (2007). The self-paced graz brain-computer interface: methods and applications. *Computational intelligence and neuroscience*, 2007, 9. Retrieved from <https://doi.org/10.1155/2007/79826> doi: 10.1155/2007/79826
- Scherer, R., Zanos, S. P., Miller, K. J., Rao, R. P. N., & Ojemann, J. G. (2009). Classification of contralateral and ipsilateral finger movements for electrocorticographic brain-computer interfaces. *Neurosurgical Focus*, 27(1), E12. Retrieved from <https://doi.org/10.3171/2009.4.FOCUS0981> (PMID: 19569887) doi: 10.3171/2009.4.FOCUS0981
- Schirrmeister, R. T., Springenberg, J. T., Fiederer, L. D. J., Glasstetter, M., Eggensperger, K., Tangermann, M., ... Ball, T. (2017). Deep learning with convolutional neural networks for brain mapping and decoding of movement-related information from the human EEG. *CoRR, abs/1703.05051*. Retrieved from <http://arxiv.org/abs/1703.05051>

- Schlögl, A., Flotzinger, D., & Pfurtscheller, G. (1997). Adaptive autoregressive modeling used for single-trial eeg classification-verwendung eines adaptiven autoregressiven modells für die klassifikation von einzeltrial-eeg-daten. *Biomedizinische Technik/Biomedical Engineering*, 42, 162–167. Retrieved from <https://www.degruyter.com/view/j/bmte.1997.42.issue-6/bmte.1997.42.6.162/bmte.1997.42.6.162.xml> doi: 10.1515/bmte.1997.42.6.162
- Schomer, D. L., & Lopes da Silva, F. H. (2018). *Niedermeyer's electroencephalography: Basic principles, clinical applications, and related fields*. Oxford University Press. Retrieved from <https://www.amazon.com/Niedermeyers-Electroencephalography-Principles-Clinical-Applications/dp/0190228482?SubscriptionId=0JYN1NVW651KCA56C102&tag=techkie-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=0190228482>
- Schouten, A. C., & Campfens, S. F. (2012). Directional coherence disentangles causality within the sensorimotor loop, but cannot open the loop. *The Journal of Physiology*, 590(10), 2529-2530. Retrieved from <https://physoc.onlinelibrary.wiley.com/doi/abs/10.1113/jphysiol.2012.228684> doi: 10.1113/jphysiol.2012.228684
- Shannon, C. E. (1949). Communication in the presence of noise. *Proceedings of the IRE*, 37(1), 10-21. doi: 10.1109/JRPROC.1949.232969
- Shawe-Taylor, D., Shawe-Taylor, J., & Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge University Press. Retrieved from <https://books.google.be/books?id=9i0vg12lti4C>
- Shenoy, P., Miller, K. J., Ojemann, J. G., & Rao, R. P. N. (2008). Generalized features for electrocorticographic bcis. *IEEE Transactions on Biomedical Engineering*, 55(1), 273-280. doi: 10.1109/TBME.2007.903528
- Sitaram, R., Caria, A., Veit, R., Gaber, T., Rota, G., Kuebler, A., & Birbaumer, N. (2007). fmri brain-computer interface: A tool for neuroscientific research and treatment. *Intell. Neuroscience*, 2007, 1:1–1:10. Retrieved from <http://dx.doi.org/10.1155/2007/25487> doi: 10.1155/2007/25487
- Stokes, M., & Blythe, G. (2001). *Muscle sounds in physiology, sports science and clinical investigation: Applications and history of mechanomyography*. Medical Intelligence Oxford.

- Subasi, A., Yilmaz, M., & Ozcalik, H. R. (2006). Classification of emg signals using wavelet neural network. *Journal of Neuroscience Methods*, 156(1), 360 - 367. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0165027006001440> doi: <https://doi.org/10.1016/j.jneumeth.2006.03.004>
- Subha, D. P., Joseph, P. K., Acharya U, R., & Lim, C. M. (2010). Eeg signal analysis: A survey. *Journal of Medical Systems*, 34(2), 195–212. Retrieved from <https://doi.org/10.1007/s10916-008-9231-z> doi: 10.1007/s10916-008-9231-z
- Thrun, S., & Pratt, L. (2012). *Learning to learn*. Springer Science & Business Media. Retrieved from [https://books.google.be/books?id=X\\_jpBwAAQBAJ](https://books.google.be/books?id=X_jpBwAAQBAJ)
- Tourville, J. A., & Guenther, F. H. (2011). The diva model: A neural theory of speech acquisition and production. *Lang Cogn Process*, 26(7), 952-981. Retrieved from <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3650855/> (23667281[pmid]) doi: 10.1080/01690960903498424
- Towle, V. L., Bolaños, J., Suarez, D., Tan, K., Grzeszczuk, R., Levin, D. N., ... Spire, J.-P. (1993). The spatial location of eeg electrodes: locating the best-fitting sphere relative to cortical anatomy. *Electroencephalography and Clinical Neurophysiology*, 86(1), 1 - 6. Retrieved from <http://www.sciencedirect.com/science/article/pii/001346949390061Y> doi: [https://doi.org/10.1016/0013-4694\(93\)90061-Y](https://doi.org/10.1016/0013-4694(93)90061-Y)
- Vidal, J. J. (1973). Toward direct brain-computer communication. *Annual Review of Biophysics and Bioengineering*, 2(1), 157-180. Retrieved from <https://doi.org/10.1146/annurev.bb.02.060173.001105> (PMID: 4583653) doi: 10.1146/annurev.bb.02.060173.001105
- Vidal, J. J. (1977). Real-time detection of brain events in eeg. *Proceedings of the IEEE*, 65(5), 633-641. doi: 10.1109/PROC.1977.10542
- Vidaurre, C., Scherer, R., Cabeza, R., Schlögl, A., & Pfurtscheller, G. (2006). Study of discriminant analysis applied to motor imagery bipolar data. *Medical & Biological Engineering & Computing*, 45(1), 61. Retrieved from <https://doi.org/10.1007/s11517-006-0122-5> doi: 10.1007/s11517-006-0122-5
- Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on machine learning* (pp. 1096–1103). New York, NY, USA: ACM. Retrieved from <http://doi.acm.org/10.1145/1390156.1390294> doi: 10.1145/1390156.1390294

- Völker, M., Schirrmeister, R. T., Fiederer, L. D. J., Burgard, W., & Ball, T. (2017). Deep Transfer Learning for Error Decoding from Non-Invasive EEG. *ArXiv e-prints*.
- Weiskopf, N., Mathiak, K., Bock, S. W., Scharnowski, F., Veit, R., Grodd, W., ... Birbaumer, N. (2004). Principles of a brain-computer interface (bci) based on real-time functional magnetic resonance imaging (fmri). *IEEE Transactions on Biomedical Engineering*, 51(6), 966-970. doi: 10.1109/TBME.2004.827063
- Witham, C. L., & Baker, S. N. (2012). Reply from c. l. witham and s. n. baker. *J Physiol*, 590(10), 2531-2533. Retrieved from <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3424770/> doi: 10.1113/jphysiol.2012.231225
- Witham, C. L., Riddle, C. N., Baker, M. R., & Baker, S. N. (2011). Contributions of descending and ascending pathways to corticomuscular coherence in humans. *The Journal of Physiology*, 589(15), 3789-3800. Retrieved from <https://physoc.onlinelibrary.wiley.com/doi/abs/10.1113/jphysiol.2011.211045> doi: 10.1113/jphysiol.2011.211045
- Xie, Y., Zhang, Z., Sapkota, M., & Yang, L. (2016). Spatial clockwork recurrent neural network for muscle perimysium segmentation. In S. Ourselin, L. Joskowicz, M. R. Sabuncu, G. Unal, & W. Wells (Eds.), *Medical image computing and computer-assisted intervention – miccai 2016* (pp. 185–193). Cham: Springer International Publishing.
- Zheng, W.-L., & Lu, B.-L. (2016). Personalizing eeg-based affective models with transfer learning. In *Proceedings of the twenty-fifth international joint conference on artificial intelligence* (pp. 2732–2738). AAAI Press. Retrieved from <http://dl.acm.org/citation.cfm?id=3060832.3061003>
- Zivkovic, Z. (2004). Improved adaptive gaussian mixture model for background subtraction. In *Proceedings of the 17th international conference on pattern recognition, 2004. icpr 2004*. (Vol. 2, p. 28-31 Vol.2). doi: 10.1109/ICPR.2004.1333992

# Appendices

# Appendix A

## Additional figures and tables

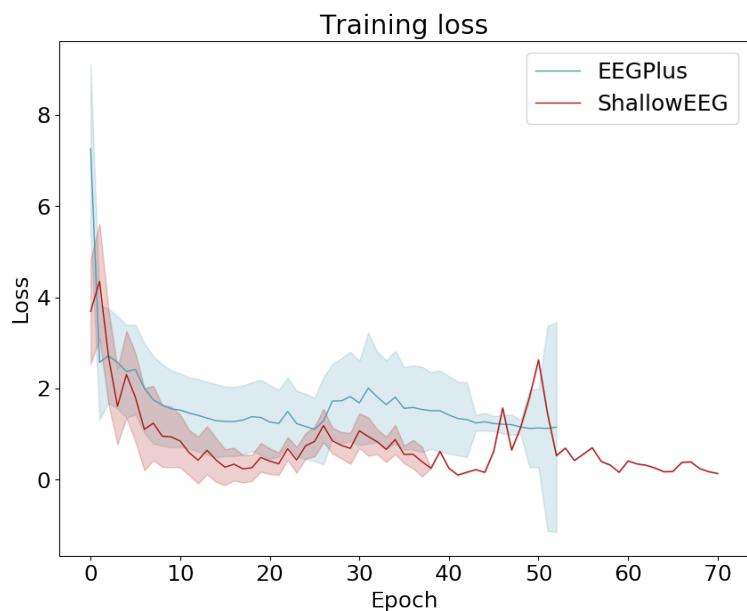


Figure A.1: Loss value on the training set during training

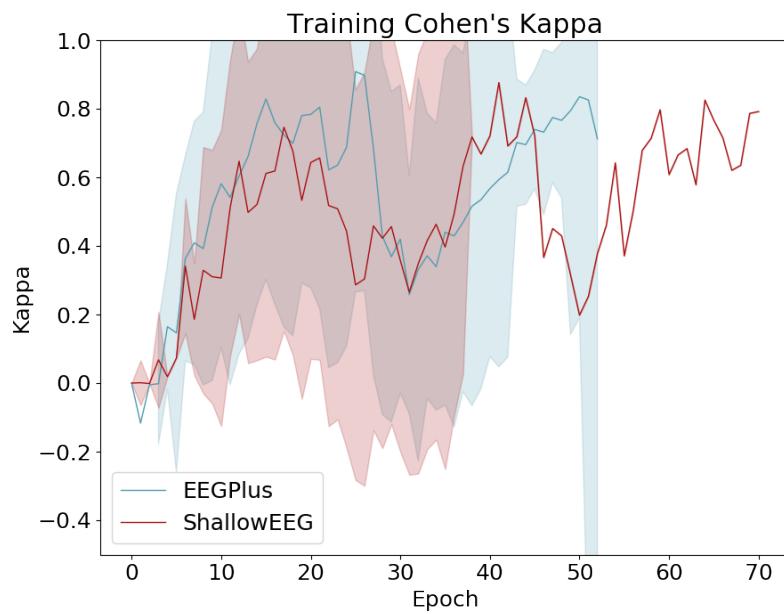


Figure A.2: Kappa value on the training set during training

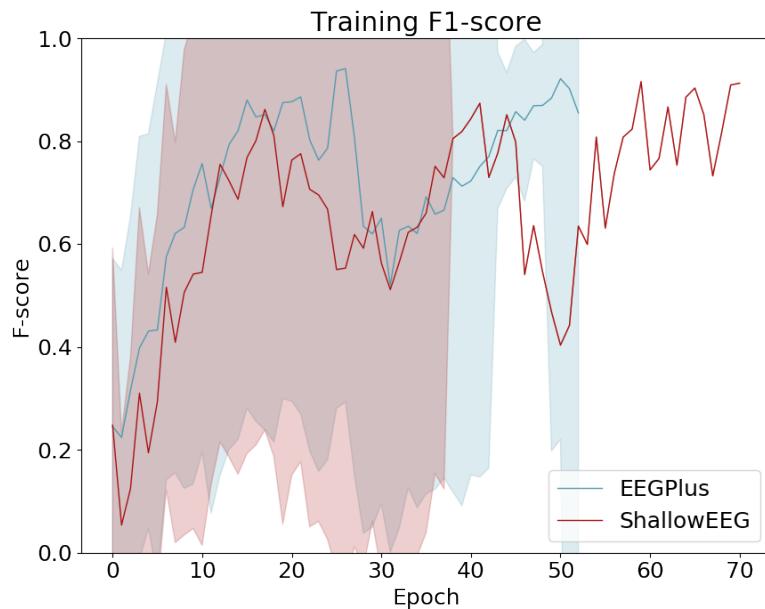


Figure A.3: F-score value on the training set during training

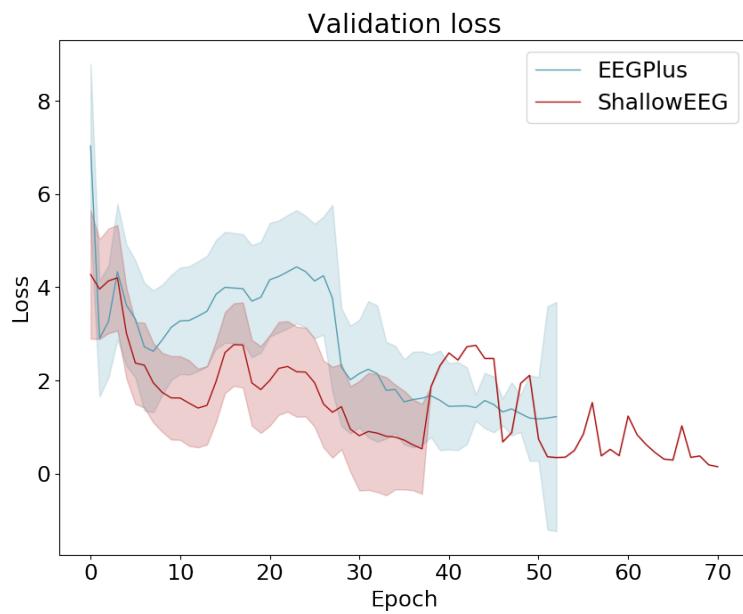


Figure A.4: Loss value on the validation set during training

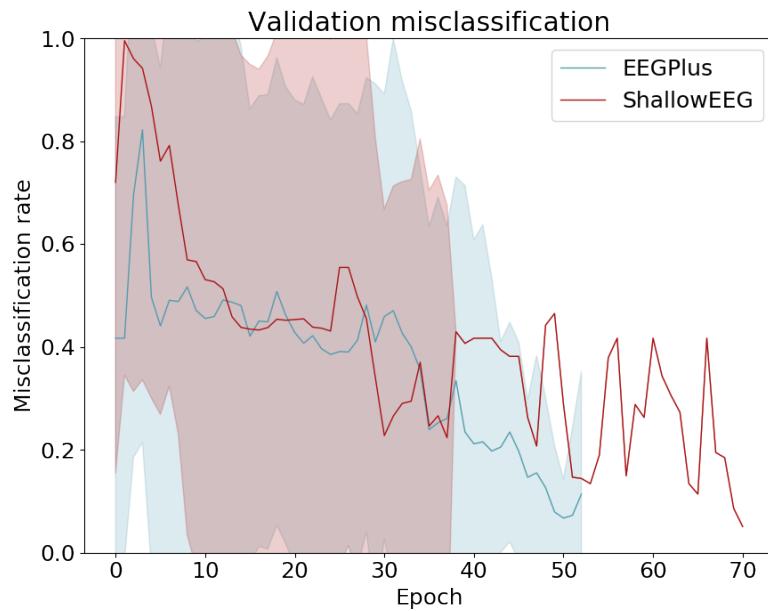


Figure A.5: Misclassification rate on the validation set during training

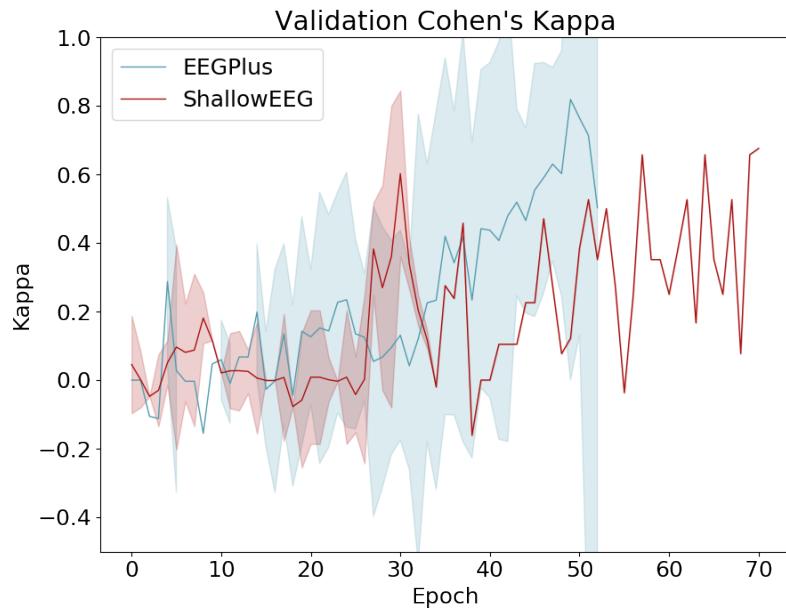


Figure A.6: Kappa value on the validation set during training

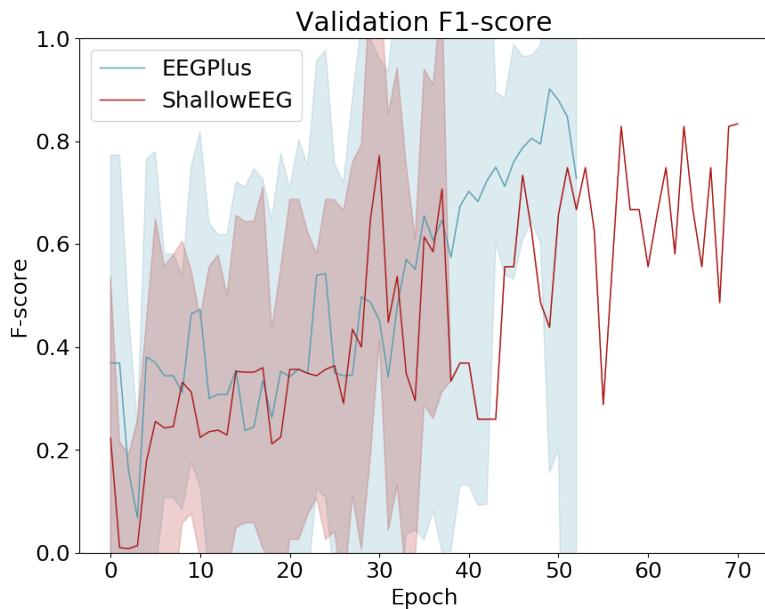


Figure A.7: F-score value on the validation set during training

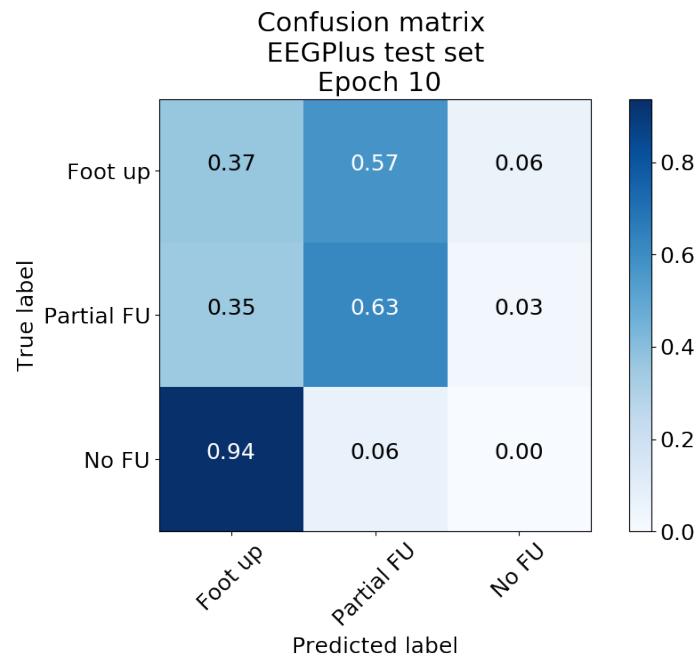


Figure A.8: Averaged confusion matrix on test set at epoch 10 for EEGPlus

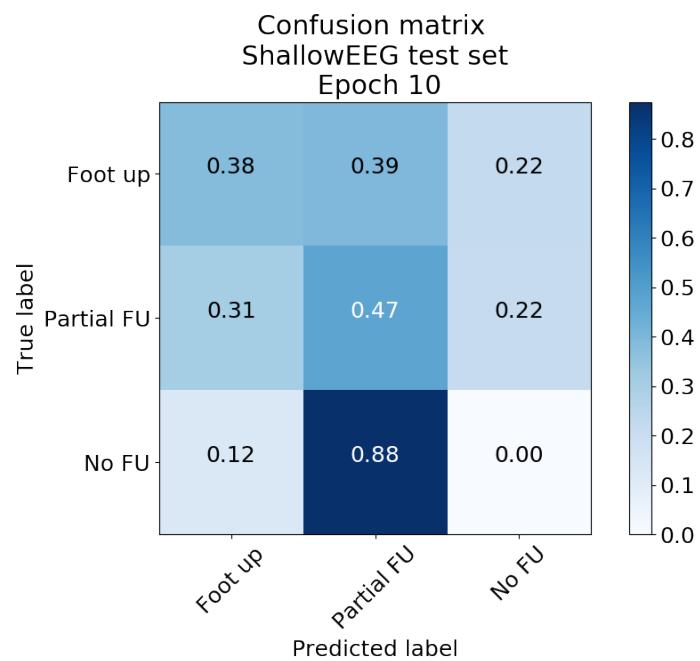


Figure A.9: Averaged confusion matrix on test set at epoch 10 for ShallowEEG

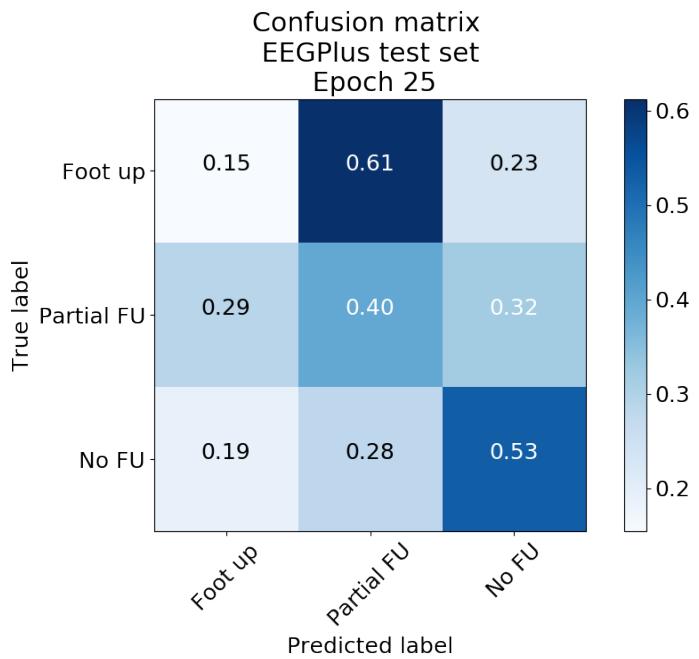


Figure A.10: Averaged confusion matrix on test set at epoch 25 for EEGPlus

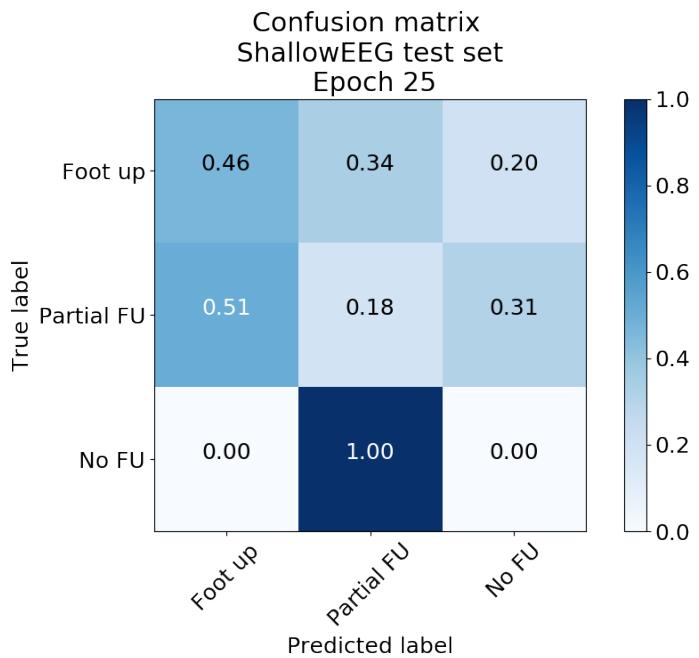


Figure A.11: Averaged confusion matrix on test set at epoch 25 for ShallowEEG

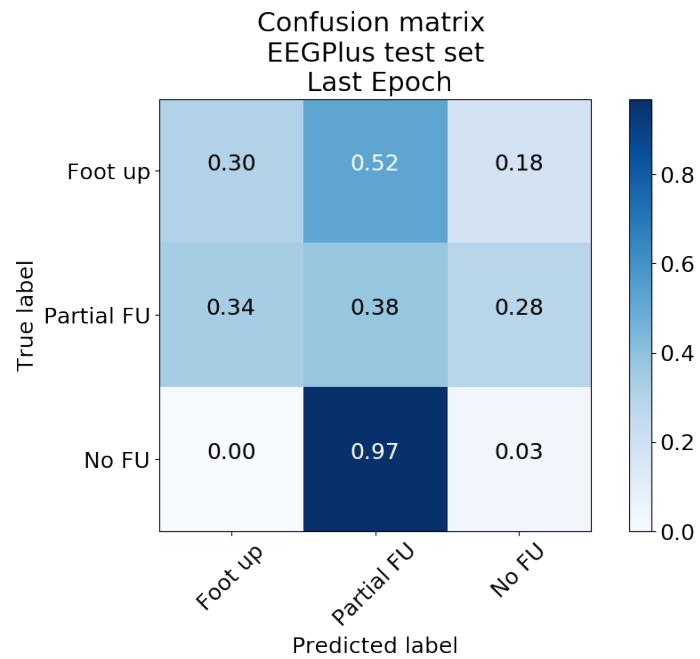


Figure A.12: Averaged confusion matrix on test set at last epoch for EEGPlus

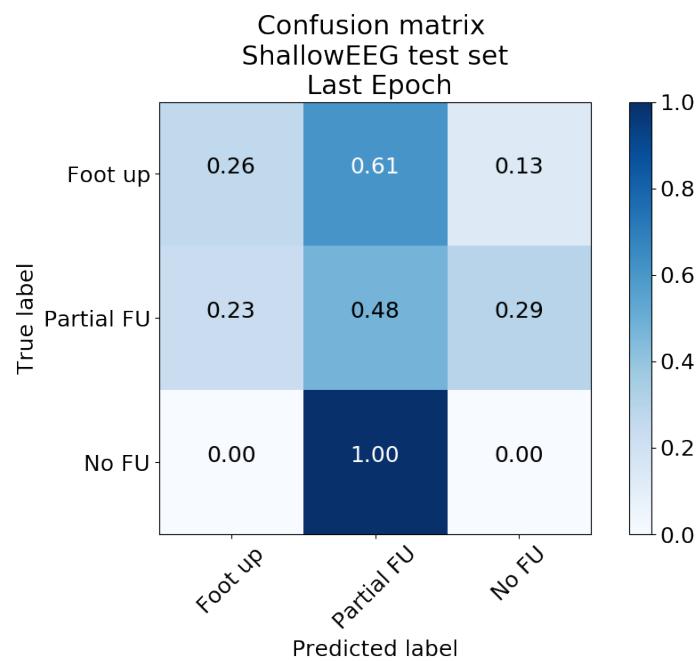


Figure A.13: Averaged confusion matrix on test set at last epoch for ShallowEEG