UNIVERSITY
OF OSLO

Master's thesis

Localization and identification of Neural Sources from simulated EEG Signals

Kamilla Ida Julie Sulebakk

Biological and Medical Physics
60 ECTS study points

Department of Physics
Faculty of Mathematics and Natural Sciences

Autumn 2023

Kamilla Ida Julie Sulebakk

Localization and identification of Neural Sources from simulated EEG Signals

Acknowledgements

Massive thank-yous to my supervisor Gaute Einevoll and my co-supervisor Torbjørn Ness.

Contents

Acknowledgements **i**

Introduction **v**
0.1 Motivation . . . 1
0.2 Goal and Objectives . . . 1
0.3 Structure of the Thesis . . . 1

1 Introduction to Neuroscience **3**
1.1 The Neuron . . . 3
1.2 Head Models and Multicompartmental Modeling . . . 5
1.3 Currents and Potentials in the Brain . . . 5
1.4 Electroencephalograpy . . . 5
1.5 EEG Forward modeling . . . 5
1.6 The Inverse Problem . . . 5

2 Electroencephalograpy **7**
2.1 Forward modeling of EEG signals . . . 8
2.2 Multicompartmental modeling . . . 10
2.3 Volume conductors . . . 10
2.4 Current Dipole Approximation . . . 10
2.5 Head Models . . . 13
2.5.1 The New York Head . . . 13
2.6 The Inverse Problem and Source Localization . . . 17

3 Machine Learning and Neural Networks **19**
3.1 Machine Learning and its Fundational Principles . . . 19
3.1.1 Fitting a Machine Learning Model . . . 20
3.1.2 Gradient Descent and Its Variants . . . 20
3.2 Neural Networks . . . 22
3.2.1 Activation functions . . . 24

4 Dipole Source Localization using Neural Networks **31**
4.0.1 Neural Networks . . . 32

4.1 Feed-Forward Neural Network Approach for localizing single dipole sources . . . 32
4.1.1 Validation accuracy . . . 32
4.1.2 Activation functions, Batchsize and Optimization . . . 33
4.1.3 Training, testing and evaluation . . . 33

5 DiLoc - A NN Apporach for Source Localization 35
5.1 Architecture . . . 35
5.2 Simulation of EEG Signals . . . 37
5.2.1 The Effect of dipole location and orientation . . . 37
5.2.2 Noise . . . 39
5.2.3 Final Dataset . . . 40
5.3 Training the DiLoc Network . . . 42
5.4 Diloc as Convolutional Neural Network . . . 43

6 Localizing Single Dipole Sources 45
6.1 Localizing Single Dipole Sources . . . 45
6.2 Convolution Neural Network Approach for localizing single dipole sources . . . 49

7 Extension of the DiLoc Network 53
7.1 Predicting Single Dipole Sources with Amplitudes . . . 53
7.2 Predicting Region of Active Correlated Current Dipoles with Amplitudes . . . 58
7.3 Localizing Multiple Dipole Sources . . . 58

Introduction

Electroencephalography (EEG) is a method for recording electric potentials stemming from neural activity at the surface of the human head, and it has important scientific and clinical applications. An important issue in EEG signal analysis is so-called source localization where the goal is to localize the source generators, that is, the neural populations that are generating specific EEG signal components. An important example is the localization of the seizure onset zone in EEG recordings from patients with epilepsy. A drawback of EEG signals is however that they tend to be difficult to link to the exact neural activity that is generating the signals.

Source localization from EEG signals has been extensively investigated during the last decades, and a large variety of different methods have been developed. Source localization is very technically challenging: because the number of EEG electrodes is far lower than the number of neural populations that can potentially be contributing to the EEG signal, the problem is mathematically under-constrained, and additional constraints on the number of neural populations and their locations must therefore be introduced to obtain a unique solution.

For the purpose of analyzing EEG signals, the neural sources are treated as equivalent current dipoles. This is because the electric potentials stemming from the neural activity of a population of neurons will tend to look like the potential from a current dipole when recorded at a sufficiently large distance, as in EEG recordings. Source localization is therefore typically considered completed when the location of the current dipoles has been obtained. However, an exciting possibility is to try to go one step further and identify the type of neural activity that caused a localized current dipole. For example, the type of synaptic input (excitatory or inhibitory) to a population of neurons, and the location of the synaptic input (apical or basal) will result in different current dipoles (Ness et al., 2022). It has also been speculated that dendritic calcium spikes can be detected from EEG signals, which could lead to exciting new possibilities for studying learning mechanisms in the human brain (Suzuki & Larkum, 2017). Identifying different types of neural activity from EEG signals would however require knowledge of how different types of neural activity are reflected in EEG signals. Tools for calculating EEG signals from biophysically detailed neural simulations

have however recently been developed, and are available through the software LFPy 2.0 (Hagen et al., 2018; Næss et al., 2021). This allows for simulations of different types of neural activity and the resulting EEG signals, opening up for a more thorough investigation of the link between EEG signals and the underlying neural activity.

The past decade has seen a rapid increase in the availability and sophistication of machine learning techniques based on artificial neural networks, like Convolutional Neural Networks (CNNs). These methods have also been applied to EEG source localization with promising results. However, it has not been investigated if CNNs can also identify the neural origin of EEG signals, in addition to localizing neural sources. In this Master's thesis, the aim will be to investigate the possibility of using CNNs to not only localize current dipoles but also identify the neural origin of different types of neural activity, based on simulated data of different types of neural activity and the ensuing EEG signal.

0.1 Motivation

Neurobiology is the study of the nervous system, including the structure, function, and development of neurons and neural circuits. The physics of the neuron is an important component of neurobiology, as it involves understanding the mechanisms by which neurons generate and transmit electrical signals. The basic unit of the nervous system is the neuron, which is capable of producing and transmitting electrical signals, or action potentials, across its membrane. These electrical signals are generated by the flow of charged ions into and out of the neuron, and are essential for communication between neurons and the transmission of information throughout the nervous system.

One technique for studying the electrical activity of the brain is electroencephalography (EEG), which measures the voltage fluctuations resulting from the electrical activity of neurons. EEG is a non-invasive technique that involves placing electrodes on the scalp, and has been used to study a wide range of cognitive and neural processes, including perception, attention, and memory. One of the challenges of interpreting EEG signals is the "inverse problem," which involves determining the location and nature of the underlying sources of electrical activity in the brain.

One approach to solving the inverse problem is source localization, which involves estimating the location and strength of the electrical sources in the brain that are responsible for the measured EEG signals. Source localization is a challenging problem due to the complexity of the brain and the fact that EEG signals are affected by a range of factors, including the conductivity of the scalp and the position and orientation of the electrodes. However, there are a number of techniques and algorithms that have been developed to address these challenges, including dipole modeling, distributed source modeling, and beamforming (Hämäläinen et al., 1993; Grech et al., 2008).

Overall, the physics of the neuron, EEG, and source localization are all important components of neurobiology that have contributed to our understanding of the nervous system and its functioning. By combining knowledge of the physical principles of neural signaling with advanced analytical techniques, researchers are able to gain valuable insights into the underlying neural processes that give rise to behavior and cognition.

0.2 Goal and Objectives

0.3 Structure of the Thesis

Chapter 1

Introduction to Neuroscience

Sections ... and ... are based on the books Neuronal Dynamics by Gerstner, Kistler, Naud and Paninski [**gerstner2014neuronal**] and Principles of computational modelling in neuroscience by Sterratt,Graham, Gillies, and Willshaw [**sterratt2011principles**].

1.1 The Neuron

Neurons are the fundamental units of the central nervous system, forming intricate networks with numerous interconnections. Similar to other cells, neurons have a voltage difference across their cell membrane known as the membrane potential. This membrane potential is essential for signal transmission and processing, allowing neurons to communicate over long distances. At rest, the membrane potential of a neuron typically hovers around -65 mV, indicating that the interior of the cell is negatively charged compared to the external environment [**sterratt2011principles**].

A neuron consists of three distinct parts: the dendrites, the soma, also reffered to as the cell body, and the axon. Dendrites, with their branching structure, play an important role in collecting signals from other neurons. These signals are transmitted to the soma, which acts as the central processing unit, performing essential nonlinear processing. If the total input received by the soma reach a specific threshold, an action potential is generated. This signal generates an electrical current that travels along the axon, leading to the release of neurotransmitters. These neurotransmitters diffuse across the junction, or the synapse, between the sending and receiving neuron. If the receptors on the receiving neuron accept the neurotransmitters, a new electrical signal is generated. This transmission of signals between neurons at specialized junctions is called synaptic inputs.[**gerstner2014neuronal**]

The neuron sending the synaptic input is referred to as the presynaptic cell, while the neuron receiving the synaptic input is called the postsyn-

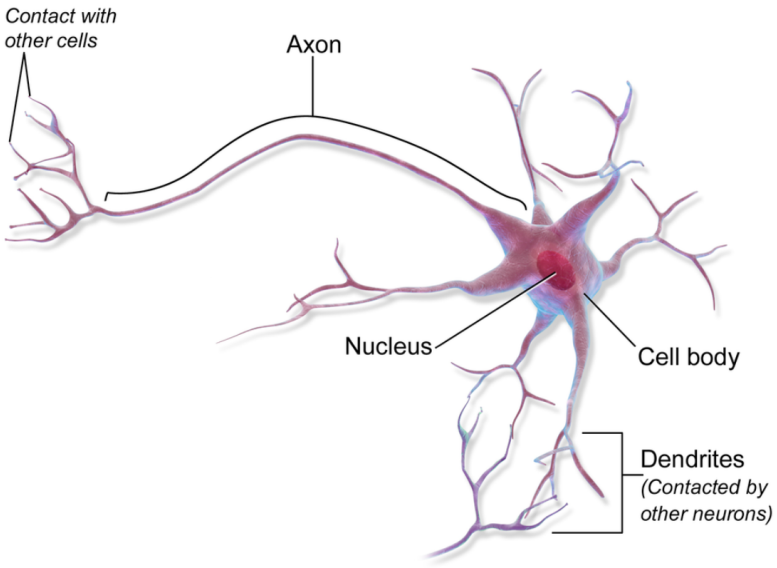

Figure 1.1: An illustation of a single neuron with dendrites, soma (cell body) and axon. The figure is taken from ...

aptic cell. In vertebrates, a presynaptic neuron often connects to more than 10^4 postsynaptic neurons. While many axonal branches terminate near the neuron itself, the axon can also extend over several centimeters to reach neurons in other regions of the brain [**gerstner2014neuronal**].

In figure 1.2 we have provided the basic arcitecture of the neuron.

If the electrical signals that transmit towards the soma reach a certain treshold, usually around -55 mV, the neuron fires, and we say that an action potential is initiated. These action potentials or spikes, typically have an amplitude of about 100 mV and a duration of 1-2 ms [**gerstner2014neuronal**]. As the form of isolated spikesof a given neuron more or less is the same through the propagation along the axon, the information of in these signals does not lay in the shape of the signal. Rather, we can from spike trains - chains of action potentials emitted by single neurons - collect information by looking at the number and timing of spikes. Figure 1.2 depicts multiple membrane potential recordings.

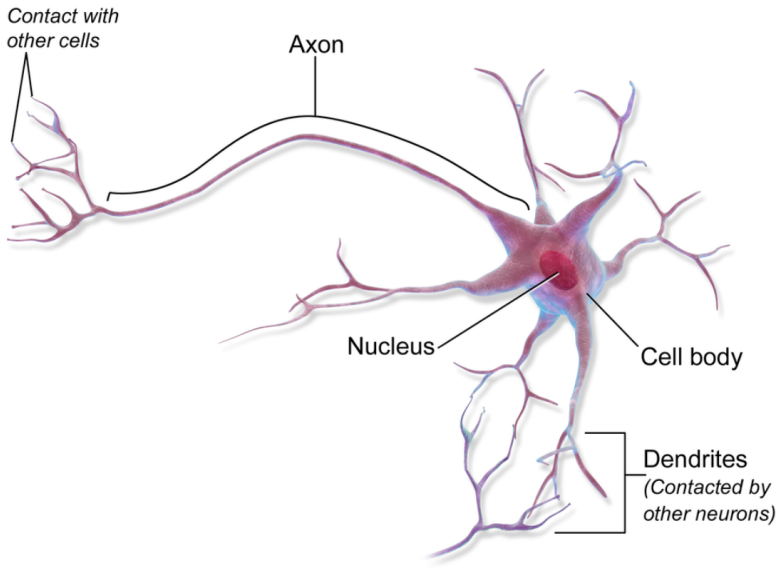

Figure 1.2: An illustation of a single neuron with dendrites, soma (cell body) and axon. The figure is taken from ...

1.2 Head Models and Multicompartmental Modeling

1.3 Currents and Potentials in the Brain

Ohm's law in volume conductors is a more genral statement than its usual form in electrical circuits. It is a linear relationship between vector current density J and the electric field E. The law is then expessed as follows:

$$J = \sigma E, \tag{1.1}$$

where σ is the conductivity of the .... (physical material). (Soruce: Electric Fields of the Brain: The Neurophysics of EEG).

1.4 Electroencephalograpy

1.5 EEG Forward modeling

1.6 The Inverse Problem

Chapter 2

Electroencephalograpy

Neurons communicate with each other through the use of electical currents. When a neuron receives a signal, it generates an electrical current that propegates along the axon and causes the release of neurotransmitters that diffuse across the gap between the sending and the recieving neuron. If the neurotransmitters are accepted by the receptors on the recieving neuron, a new electrical signal, also known as synaptic input will be generated. This transmission of signals between neurons at specialized junctions are called synaptic inputs. The process of electrical communaication between neurons creates electromagnetic fields that can be measured using electroencephalography (EEG).

Electroencephalography (EEG) is a non-invasive technique that has been used for almost a century to study the electrical potentials in the human brain. It has remained one of the most important methods for investigating the brain's activity, with significant applications in both neuroscientific and clinical research [**ilmoniemi2019brain**].

An EEG signal is believed to originate from large numbers of synaptic inputs to populations of geometrically aligned pyramidal neurons [**nunez2006electric**].The signal can be understood as a signature of neural activities that are generated from synaptic inputs to cells in the cortex. Synaptic inputs on its side are electrical or chemical signals that are being transmitted from one neuron to another, causing changes in the membrane potential of the neurons. In other words, neurons are specialized to pass signals, and synapses are the structures that make this transmission possible.

One of the primary uses of EEG is to investigate cognitive processes, diagnose diseases, and estimate functional connectivity. To measure the electrical activity in the brain, small metal disks called electrodes are placed on the scalp, which detect the electrical charges that result from the activity of brain cells. This technique can help identify abnormalities in specific areas of the brain, indicating potential signs of disease. Thus, EEG can be used to evaluate various brain disorders, including lesions, Alzheimer's disease,

epilepsy, and brain tumors. An illustration of the typical EEG measurement setup is depicted in Figure 2.1.

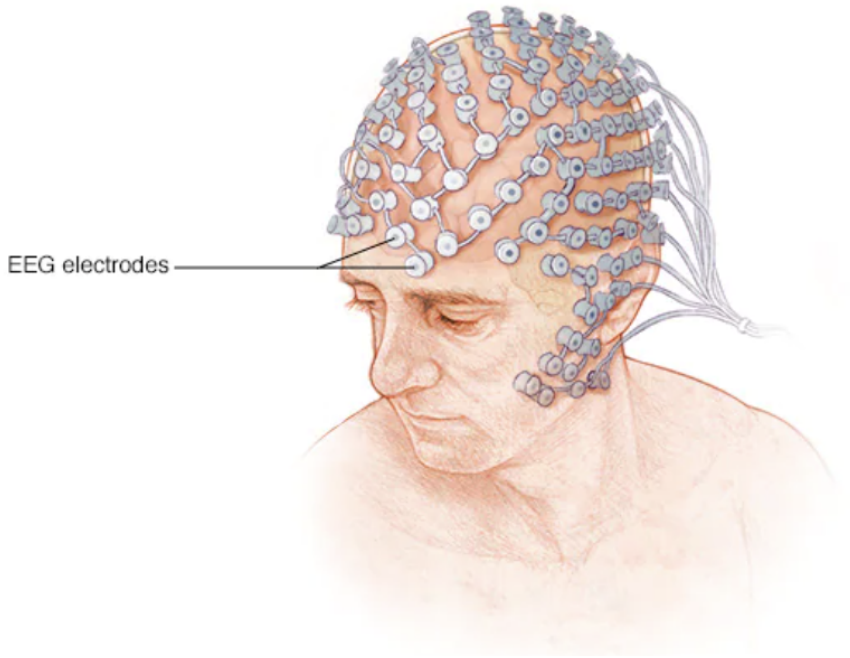

Figure 2.1: Illustration of the EEG method.

Source localization is a fundamental aspect of EEG signal analysis, with the ultimate aim of accurately identifying the location of neural current sources from EEG data. However, this is a challenging problem due to the inherent ill-posed nature of the inverse problem in electrostatics. This means that there is no unique solution, making it a difficult task to solve. Before digging deeper into the inverse problem, we will first look at how to simulate electrical brain activity and the corresponding EEG signal as it would have been measured by electrodes on the scalp. This method is better known as forward modeling.

2.1 Forward modeling of EEG signals

To better understand the complexities of the inverse problem and the nuances of source localization in EEG, it is helpful to gain a deeper comprehensionin of forward modeling, involving the simulation of electrical activity of the brain as it would be measured by electrodes on the scalp.

As explained in chapter 1, neural activity generates electric currents in the brain, which in turn create electromagnetic fields. In order to calculate extracellular electric potentials, one can envision the head as a 3D volume

conduction, and combine Maxwell's equations with the current conservation law. We then obtain the Poisson equation for computing extracellular potentials:

$$\nabla \cdot \mathbf{J} = \nabla \cdot (\sigma \nabla \phi) \tag{2.1}$$

where $\mathbf{J}$ is the electric current density in extracellular space, σ is the extracellular conductivity and ϕ is the extracellular electric potential [**naess2021biophysically**]. For simple, symetric head models, the Poisson equation can naturally be solved analytically. However, as for the New York Head which is a complex model that we will be utialize in this thesis, we will be using the numerical method known as Finite Element Method (FEM) when solve the quation.

To accurately calculate the extracellular potential(s), ϕ, a well-established two-step forward-modeling approach is used. In the first step, a multicompartmental model is utialized, which takes into account the intricate details of neuron morphologies to determine the transmembrane currents, I_n. In the second step, equation 2.1 is solved, under the assumptions that the extracellular medium acts as a volume conductor with the following properties:

- infinitely large
- linear
- ohmic
- isptropic
- homogeneous
- frequency-independent

The origin of extracellur potentials is spatially distributed membrane currents, entering and escaping the extracellular medium. These currents can be understood as current sources and sinks, and give the extracellular potential, ϕ at the electrode location $\mathbf{r}$:

$$\phi(\mathbf{r}) = \frac{1}{4\pi\sigma} \sum_{n=1}^{N} \frac{I_n}{|\mathbf{r} - \mathbf{r}_n|} \tag{2.2}$$

where $\mathbf{r}_n$ is the location of the tranmembrane current I_n, N is the number fof transmembrane currents and σ is the extracellular conductivity [**naess2021biophysically**].

2.2 Multicompartmental modeling

The potential over the membrane, V_m, in long neurons with multi-branched dendrites varies depending on whether the potential is measured in the soma or at the tip of a sital denrite [**naess2021biophysically**]. Multicompartmental (MC) models are models that account for this spatial variabilty in V_m. In such models, the neural morphology commonly is represented as isopotential cylindrical compartmens, with lengths and diameters derived from reconstructed morphologies, connected with resistors [**sterratt2011principles**]. Single values of V_m can then be computed for each individual compartment, as depiced in Figure 2.2.

The fundamental equation for MC models is given as:

$$c_m \frac{dV_{m,n}}{dt} = -i_{L,n} - \sum_w i_{w,n} - \frac{I_{\text{syn},n}}{\pi dL} + \frac{I_{\text{ext},n}}{\pi dL} \frac{d}{4r_a} \left(\frac{V_{m,n+1} - V_{m,n}}{L^2} - \frac{V_{m,n} - V_{m,n-1}}{L^2} \right) \tag{2.3}$$

where ....

This equation (2.3) assumes that a compartment n has two neighbors - one at $n-1$ and the second at $n-2$. However, when considering the endpoint compartments, this clearly will not be true. A commonly used boundry condition, is the sealed-end condition, where no axial currents leave at the cable endpoints. With other words this means that $I_{0,1} = 0$ in one end of the cable, and $I_{N,N+1} = 0$ in the other end. For the cable endpoints, we are then left with the following expressions:

$$c_m \frac{dV_{m,1}}{dt} = -i_{L,1} - \sum_w i_{w,1} - \frac{I_{\text{syn},1}}{\pi dL} + \frac{I_{\text{ext},1}}{\pi dL} \frac{d}{4r_a} \left(\frac{V_{m,2} - V_{m,1}}{L^2} \right) \tag{2.4}$$

and

$$c_m \frac{dV_{m,N}}{dt} = -i_{L,N} - \sum_w i_{w,N} - \frac{I_{\text{syn},N}}{\pi dL} + \frac{I_{\text{ext},n}}{\pi dL} \frac{d}{4r_a} \left(\frac{V_{m,N} - V_{m,N-1}}{L^2} \right) \tag{2.5}$$

2.3 Volume conductors

As mentioned above, we can use volume conductor (VC) theory to predict the resulting extracellular potential V_e at any given point in space, given that the distribution of neuronal membrane currents is known.

2.4 Current Dipole Approximation

EEG signals are generated from synaptic inputs to cells in the cortex. Synaptic inputs are electrical (or chemical) signals that are being transmitted

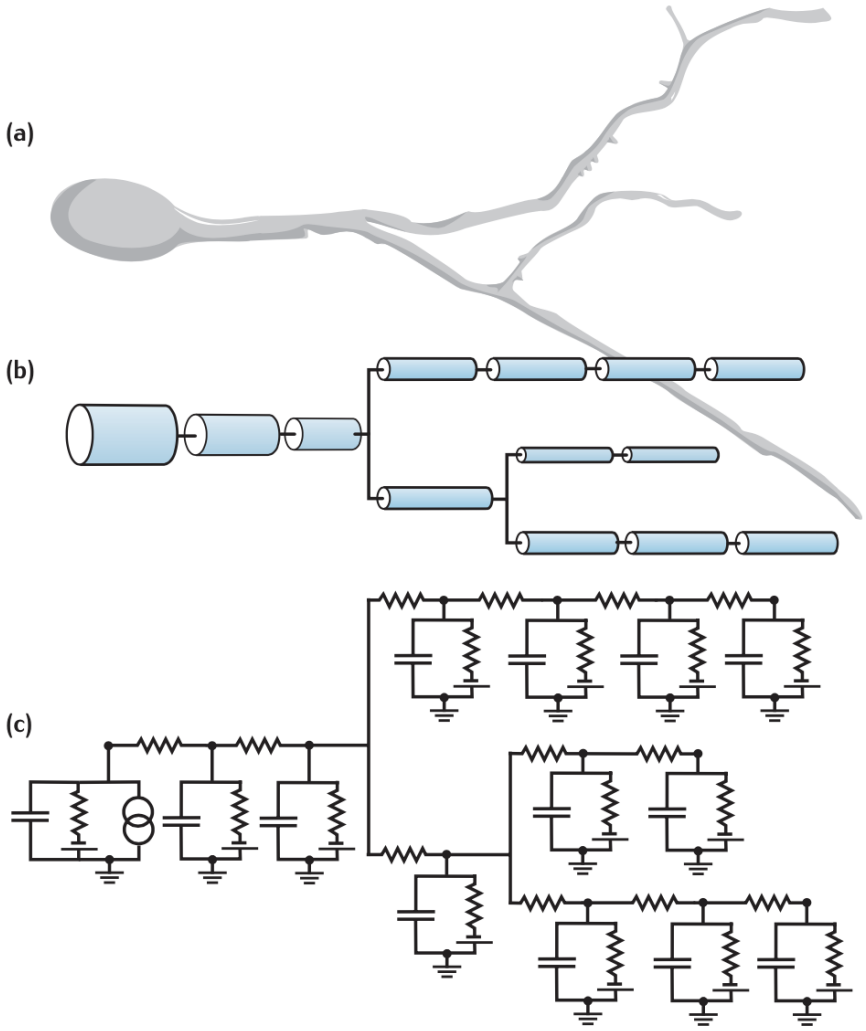

Figure 2.2: A diagram of the development of a multi-compartmental model. (a) The cell morphology is represented by (b) a set of connected cylinders. An electrical circuit consisting of (c) interconnected RC circuits is then built from the geometrical properties of the cylinders, together with the membrane properties of the cell.

from one neuron to another, causing changes in the membrane potential of the neurons. In other words, neurons are specialized to pass signals, and synapses are the structures that make this transmission possible.

When calculating extracellur potentials, V_e, at large distances from the underlying current sources, one is typically benefitted with using the current-dipole approximation. This approximation is justifided by the fact that a

neuron's contribution to V_e becomes increasingly dipolar with increasingly distance and is commonly utialized when simulating EEG signals.

We know that electrical charges can create current multipoles, depending on coordinates and symmetry of the charge distribution [**wiki:multipoles**]. Analogous, the combination of current sinks and sources set up such charge multipoles. When the distance R from the center of the volume to the recording point is larger that the distance from the volume center to the most peripheral source, multipole expansion can be used [**jackson1999classical**]. When utializing the multipole expansion theorem equations 2.6 can be expressed as follows:

$$\phi(R) = \frac{C_{\text{monopole}}}{R} + \frac{C_{\text{dipole}}}{R^2} + \frac{C_{\text{quadrupole}}}{R^3} + \frac{C_{\text{octopole}}}{R^4} + ... \tag{2.6}$$

where the numerators represents the contributions to the extracellular potential. The terms denoted C_{monopole}, C_{dipole} and $C_{\text{quadrupole}}$ represents contributions to the extracellulat potential, V_e, and can in general be extremely complicated as they depend on the relationship between radial coordinates and symmetry of the current source and measurement electrode. However, multiple expansions are often beneficial as usually only the first few terms are needed in order to provide an accurate approximation of the original funtion. This also turns out to be true in our case, as the quadrupole, octople and higher-order contributions to V_e decay more rapidly with distance R than the dipole contibution. Assuming that we are sufficiently far away from the source distribution, all terms above the dipole contribution vanish. As for the monopole contribution, we know that since ..... the net sum of currents over a neuronal membrane is always zero. This means that also the monopole term vanishes, and the expression for the extracellular potential, V_e is approximated by the dipole contribution alone:

$$\phi(\mathbf{r}) \approx \frac{C_{\text{dipole}}}{R^2} = \frac{1}{4\pi\sigma}\frac{|\mathbf{p}|cos\theta}{|\mathbf{r}-\mathbf{r}_p|^2}. \tag{2.7}$$

where we have substituded for C_{dipole} in terms of other properties. $\mathbf{p}$ denotes the current dipoile moment in a medium with conductivity σ. $R = |\mathbf{R}| = |\mathbf{r}-\mathbf{r}_p|$ is the distance between the current dipole moment at $\mathbf{r}_p$ and the electrode lokation $\mathbf{r}$. Finally θ represents the angle between $\mathbf{p}$ and $\mathbf{R}$. This equation is known as the dipole approximation and is a precise approximation for calculating extracellular potential, given that R is much larger than the dipole length $d = |\mathbf{d}|$, like in the case of EEG [**naess2021biophysically**].

Having that a current dipole moment $\mathbf{p}$ can be predicted from an axial current I inside a neuron and the distance vector $\mathbf{d}$ traveled by the axial current we get the following expression: $\mathbf{p} = I\mathbf{d}$. Generalising this equation, we get that the relationship between the current dipole moment, $\mathbf{p}$ and a *set* of neural current sources can be expressed as follows:

$$\mathbf{p} = \Sigma_{n=1}^{N} I_n \mathbf{r}_n \tag{2.8}$$

In figure 2.3 we have provided a simulation of the extracellular potential generated by a neuron in response to a single synaptic input, where the spatial distribution of membrane current was explicitly taken into consideration. Based on the figure, it is apparent that the distribution of electric charge in the extracellular potential of the neurons surroundings exhibits distinct dipole patterns when observed from a greater distance.

2.5 Head Models

The analytical formula presented in Equation 2.8 was dereived based on the assumption of a constant tissue conductivity, denoted as σ. However, because the EEG signals is measured outside the head, it is affected by the different conductivities og the brain, verebrospinal fluid (CSF), skull and scalp. Depending on problem, it may be important to keep in mind that EEG signals, in general, are significantly affected by biophysically details of the head. For instance, the conductivity of the cerebrosipinal fluid exhibits a conductivity of approximately 1.7 S/m, while the conductivity of the scull and scalp is approximately 0.01 S/m and 0.5 S/m, respectively. These conductivity variations highlight the need for more comprehensive and realistic models of the head, known as head models, which take into account such conductivity variations. In addition to acount for the variations in conductivity across different regions of the head, head models also takes into account that the EEG signal from a neuronal population will depend on whether the population is located in a *sulcus* or a *gyrus* [**naess2021biophysically**]. Said with other words, by integrating biophysical details into models, a deeper understanding emerges regarding the impact of various tissues on the distribution of extracellular potential, which in turn, can lead to improvements in the accuracy of EEG signal analysis.

2.5.1 The New York Head

A central problem for EEG is to relate scalp data to brain *current sources* (Electric Fields of the Brain: The Neurophysics of EEG)....

Especially important for electrode locations outside of the brain, such as EEG, is the knowledge about how the electrical potentials will be affected by the geometries and conductivites of the various parts of the head [**naess2021biophysically**]. A model that takes these detailes into account is the New York Head Model. The model is based on anatomical and electrical characteristics of 152 adult human brains and is solved for 231 electrode locations.

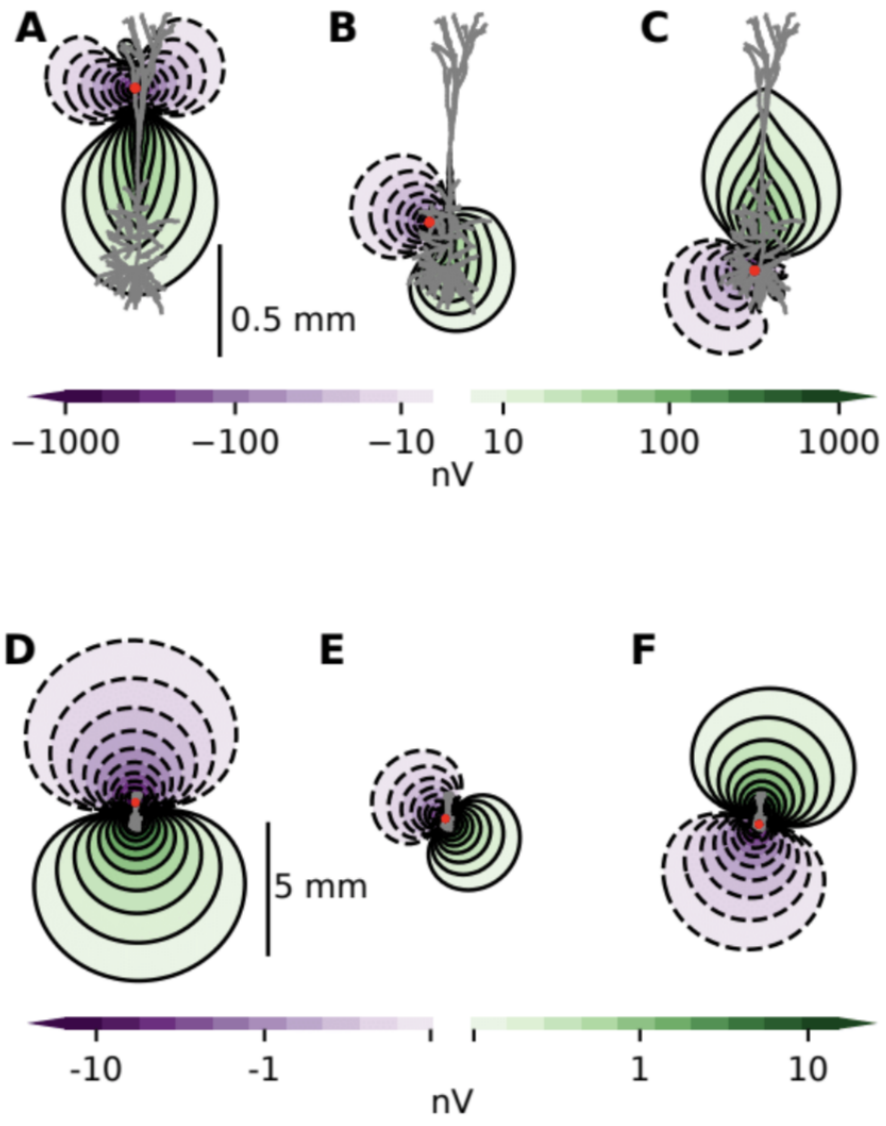

Figure 2.3: Source and explanation.

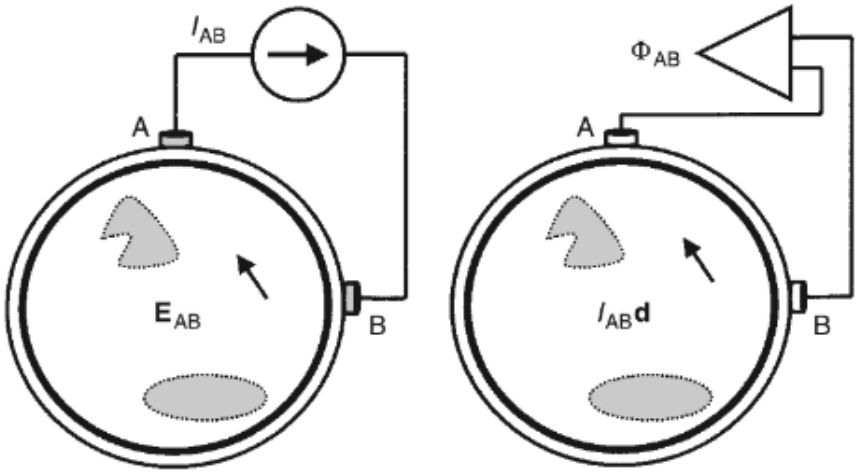

Figure 2.4: A caption here is needed.

The New York Head Model is a computer model of the human head used to simulate the electrical activity of the brain. It was created by the Electrical Geodesics Incorporated (EGI) in 2004, and is based on the anatomical and electrical characteristics of the head of a typical adult human. The model consists of a three-dimensional (3D) representation of the head and brain, with detailed information on the geometry and electrical properties of the different tissues and structures within the head. The model includes the scalp, skull, cerebrospinal fluid, gray matter, and white matter. The electrical properties of each of these tissues, such as conductivity and permittivity, are also included in the model.

The model was developed to be used for, and improve the accuracy of EEG source localization [**huang2016new**]. To generate predictions of the EEG signals recorded from different scalp locations in response to a given set of source currents, the New York head model uses the lead field matrix (SOURCE), which is a mathematical representation of the relationship between the electrical activity in the brain and the electrical potentials recorded on the scalp.

The lead field matrix is constructed by taking advantage of the reciprocity theorem that states that knowledge of the current density through a volume conducter caused by an injection of current between two stimulating electrodes completely specifies how those same recording electrodes pick up potentials caused by dipole sources in the volume conducter. If one suppose that a pair of stimulating electrodes is placed at locations A and B on the scalp as provided in figure 2.4, an external current source will cause current to flow from electrode B through the brain, and all the way to electrode A. However, due to the geometry, inhomogeneity, and ansisotropy of the

head, the current density will vary with location. The amount of current that pass through the brain depends, to a great extent, on the location of the electordes. In general, the brain currents will decrease with decreasing distance between electrodes. Thus, for a fixed pair of electrodes, the lead field vectors can be calculated as a function of position throughout a volume conducter. At each location, the orientation of the lead field vector L_{AB} is the orientation of the dipole source that produces the largest potential difference between the electrodes. The lead field matrix, $\boldsymbol{L}$ is given as:

$$L = \frac{E}{I}, \tag{2.9}$$

where I is the injected current at the electode locations and E is the resulting electric field in the brain [**naess2021biophysically**]. Moreover, the precise link between a current dipole moment p in the brain and the resulting EEG signals Φ is then related to the lead field matrix as follows:

$$\Phi_{AB} = L_{AB} \cdot p, \tag{2.10}$$

Here, an injected current I of 1 mA gives an electric potential E in V/m, meaning that a current dipole moment $\mathbf{p}$ in the unit of mAm gives EEG signals in the unit of V.

The New York Head model has been incorporated in the Python module LFPy, which provides classes for calculation of extracellular potentials from multicomparment neuron models. These tools will be utialized in this thesis. For more information read: `https://lfpy.readthedocs.io/en/latest/readme.html#summary`

The signals received from EEG are known to originate from cortical neural activity, which are often described by using current dipoles. It is therefor reasonable to implement current dipoles in the brain for when generating a biophysical modeling of EEG signals. The brain model used in this thesis is called the New York Head model, and is based on high-resolution anatomical MRI-data from 152 adult heads. The model utilizes the software tool LFPy, which is a Python module for calculation of extracellular potentials from multicompartment neuron models. This model takes into account that electrical potentials are effected by the geometries and conductivities of various parts of the head.

The cortex matrix consists of 74382 points, which refer to the number of possible positions of the dipole moment in the cortex. When generating our data set, we will for each sample randomly pick the position of the dipole moment, such that one sample corresponds to one patient. In our head model we are considering 231 electrodes uniformly distributed across the cortex, meaning that each EEG sample will consist of this many signals for each time step. However, we are not interested in the time evolution of the signals as this does not affect nor say anything about the position of the

dipole moment, and we therefor simply pick out the EEG signals for when $t = 0$ (note that the choice of time step could have been randomly picked). Our final design matrix will then consist of 1000 rows, corresponding to each patient, and 231 columns also refereed to as features, representing the signal of each electrode. The final output we are trying to predict is then the one dimensional vector with length 1000, where each element consists of the x-, y- and z- position of the dipole moment. An example of how the input EEG signals may look like is given in appendix A, where we also have marked the dipole moment with a yellow star.

2.6 The Inverse Problem and Source Localization

Chapter 3

Machine Learning and Neural Networks

Machine learning is a field concerned with constructing computer programs that learn from experience, where the utialization of data improves computer performance across various tasks. Within this broad scope, one notable application lies in the identification of sources generating abnormal electrical brain signals. By employing specific machine learning algorithms, EEG data can be processed and analyzed to accurately localize the sources responsible for the recorded signals. These algorithms learn from the data and uncover patterns that associate the signals with their corresponding sources, effectively solving the EEG inverse problem. In this chapter, we introduce the field of machine learning and provide an overview of relevant tequniqes for solving our specific EEG inverse problem and its wider implications.

3.1 Machine Learning and its Fundational Principles

"Machine Learning is a subfield of artificial intelligence with the goal of developing algorithms capable of learning from data automatically" [**mehta2019high**]. The typical machine learning (ML) problems are addressed using the same three elements. The first element is the dataset $\mathcal{D} = (\mathbf{X}, \mathbf{y})$ where $\mathbf{X}$ commonly is refered to as the design matrix, and consists of independent variables, and $\mathbf{y}$ is a vector consisting of dependent variables. Next, we have the model itself, $f(\mathbf{x}; \boldsymbol{\theta})$. The ML model can be seen as a function used to predict an output from a vector of input variables, i.e. $f : \mathbf{x} \rightarrow y$ of the parameters $\boldsymbol{\theta}$. Finally, the third element, allows us to evaluate how well the model performs on the obervations $\mathbf{y}$. This element is known as the cost funtion $\mathcal{C}(\mathbf{y}, f(\mathbf{X}); \boldsymbol{\theta})$.

3.1.1 Fitting a Machine Learning Model

The first step in "fitting" a machine learning model, is to randomly split the dataset $\mathcal{D}$ into train and test sets. This is done in order to make a model compatible with multiple data sets. The size of each set commonly depend on the size of the data set avaible, however a rule of thumb is that the majority of the data are partitioned into the trainng set (e.g., 80%) with the remainder going into the test set [**mehta2019high**].

When using the expression "fitting a model" one commonly refer to finding the value of $\boldsymbol{\theta}$ that minimizes a chosen cost function, employing data from the training set. One commonly used cost funtion is the squared error, in which can be written as follows:

$$\text{MSE}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=0}^{n-1}(y_i - \tilde{y}_i)^2, \tag{3.1}$$

where $\boldsymbol{\theta} = \theta_0, \theta_1, ..., \theta_n$ denotes the model parametes, $\tilde{y}_i$ represents the predicted value and y_i is the corresponding true value.

A general expression for any type of cost function can be formulated as follows:

$$C(\theta) = \Sigma_{i=0}^{n} c_i(\mathbf{x}_i, \theta) \tag{3.2}$$

In this expression, $c_i(\mathbf{x}_i, \theta)$ represents the cost associated with the i-th data point, where $\mathbf{x}_i$ represents the input data and θ denotes the parameter vector. This notation emphasizes the summation over all data points from 1 to n, where each data point contributes its own cost to the overall cost function.

In order to minimize the cost function and find the optimal values for the model parameters, $\boldsymbol{\theta}$, an optimization alorithm is typically employed. One widely used optimization algorithm is gradient descent, which iteratively updates the parameters based on the negative gradient of the cost function.

3.1.2 Gradient Descent and Its Variants

Gradient Descent (GD) is an iterative optimization algorithm used to locate a local minima of a differentiable function. The core concept of the algorithm is based on the observation that a function $F(\mathbf{x})$ will decrease most rapidly if we repeatedly move in one direction opposite to the negative gradient of the function at a given point $\mathbf{w}$, $-\nabla F(\mathbf{a})$. This means that if

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \eta \nabla F(\mathbf{w}_n) \tag{3.3}$$

for a sufficiently small learning rate η, we are always moving towars a minimum, since $F((w)_n) \geq F((w)_{n+1})$ [**wiki-gradient-descent**]. After each update, the gradient is recalculated for the updated weight vector $\mathbf{w}$,

and the process is repeated [**bishop2006pattern**]. Based on this observation, the iterative process begins with an initial guess x_0 for a local minimum of the function F. It then generates a sequence $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$ such that each element in the sequence is upated according to the rule:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \eta_n \nabla F(\mathbf{x}_n), n \geq 0, \tag{3.4}$$

where $\eta_n \geq 0$. The sequence forms what we call a monotonically decreasing sequence:

$$F(\mathbf{x}_0) \geq F(\mathbf{x}_1) \geq F(\mathbf{x}_2) \geq ... \geq F(\mathbf{x}_n) \tag{3.5}$$

Hence, with this iterative process, it is hoped that the sequence $(\mathbf{x}_n)$ converges to the desired local minimum [**wiki-gradient-descent**].

However, it is important to note that the error function in gradient descent is computed based on the training set, so that each step requires that the entire training set, reffered to as the *batch*, is processed in order to evaluate the new gradient. In that sense, gradient descent is generally considered a suboptimal algorithm. This perception aligns with the algorithms sensitivity to the initial condition, $\mathbf{w}_0$, and the choice of the learning rate η. The sensitivity to initial conditions can be explained by the fact that we to a large extent most often deal with high-dimensional, non-convex cost functions with numerous local minima - where the risk of getting stuck in local minimums if the initial guess is not accurate. Additionally, guessing on a too large learning rate may result in overshooting the global minimum, leading to unpredictable behavior, while a too small learning rate increases the number of iterations required to reach a minimum point, thereby increasing computational time. Stochastic gradient descent, however, is a version of gradient descent that has provided useful in practise for training machine learning algorithms on large data sets [**bishop2006pattern**].

Stochastic Gradient Descent

The method of Stochastic Gradient Descent (SGD) allows us to compute the gradient by randomly selecting subsets of the data at each iteration, rather than using the entire dataset [**bishop2006pattern**]. The update can be written as:

$$\mathbf{w}_{\tau+1} = \mathbf{w}_\tau - \eta \nabla F_n(\mathbf{w}_\tau) \tag{3.6}$$

These smaller subsets taken from the entire dataset are commonly reffered to as mini-batches. In other words, SGD is just like regular GD, except it only looks at one mini-batch for each step. Introducing fluctuation by only taking the gradient on a subset of the data, is beneficial as it enables the algorithm to jump to a new and potentially better local minima, rather that getting stuck in a local minimum point.

Stochastic Gradient Descent with Momentum

Splitting the dataset into mini-batches, as done with SGD, naturally reduces the calculation time. However, adding *momentum*, to the algorithm, not only leads to faster converging, due to stronger acceleration of the gradient vectors in the relevant directions, but also improves the algorithms sensitivity to initial guess of the learning rate η. The momentum can be understood as a memory of the direction of the movement in parameter space, which is done by adding a fraction γ of the weight vector of the past time step to the current weight vector:

$$\mathbf{v}_\tau = \gamma \mathbf{v}_{\tau-1} - \eta \nabla F_n(\mathbf{w}_\tau) \tag{3.7}$$

$$\mathbf{w}_\tau = \mathbf{w}_{\tau-1} + \mathbf{v}_\tau \tag{3.8}$$

Here, $\mathbf{w}\tau$ represents the updated weight vector at iteration τ, $\mathbf{w}_{\tau-1}$ is the previous weight vector, $\mathbf{v}_\tau$ is the updated momentum vector at iteration τ, γ is the momentum coefficient, η is the learning rate, and $\nabla F_n(\mathbf{w}_\tau)$ is the gradient of the cost function F_n computed on the mini-batch.

3.2 Neural Networks

Neural networks are a distinct class of nonlinear machine learning models capable of learning tasks by observing examples, without requiring explicit task-specific rules [**Hjorth-Jensen2022**]. The models mimics the way bilogical neurons trasmit signals, with interconnected nodes known as neurons that communicate through mathematical functions across layers. The layers in neural networks contain an arbitraty number of neurons, where each connection is represented by a weight variable.

The network gathers knowledge by detecting relationships and patterns in data using past experiences known as training examples. These patterns are further updated by the usage of appropriate activation functions and finally presented as the output [**nwankpa2018activation**]. A neural network consits of many such neurons stacked into layers, with the output of one layer serving as the input for the next. Typically, the neural networks are built up of an input layer, an output layer and layers in between, called hidden layers. In figure 3.1 we have provided the basic architecture of neural networks. Here nodes are depiced as circular shapes, while arrows indicate connections between the nodes.

The behaviour of the human brain has inspired the following simple mathematical model for an artificial neuron:

$$a = f\left(\Sigma_{i=1}^{n} w_i x_i\right)) = f(z) \tag{3.9}$$

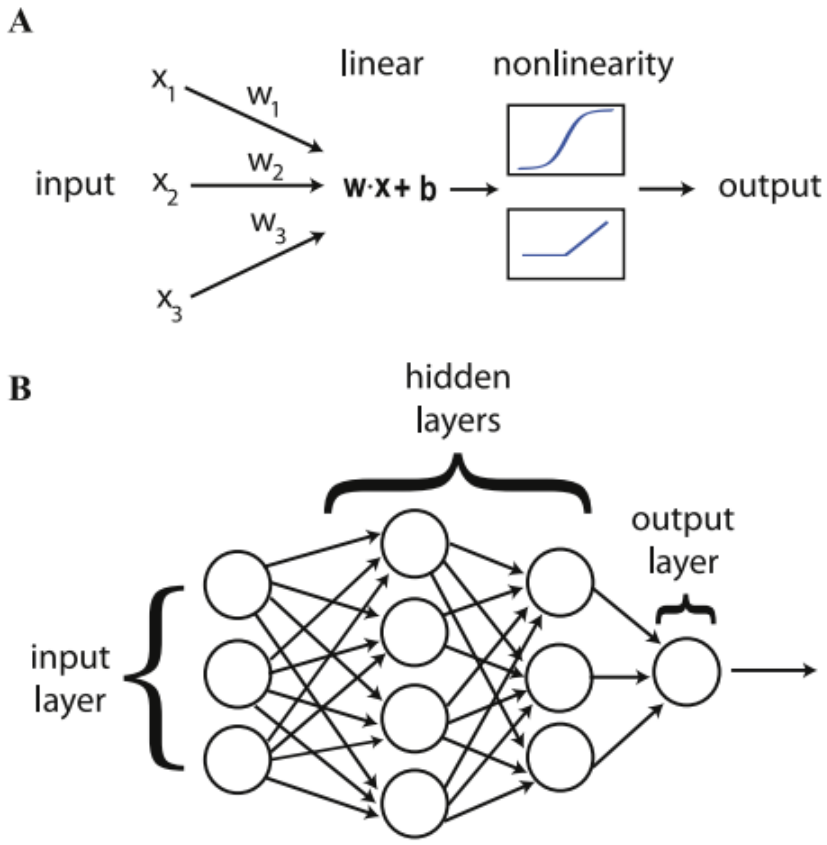

Figure 3.1: **(A)** The fundamental structure of neural networks comprises simplified neuron units that perform a linear operation to assign different weights to inputs, followed by a non-linear activation function.**(B)** These neuron units are organized into layers, where the output of one layer serves as the input to the subsequent layer, forming a hierarchical arrangement.

where a is the output of the neuron, and is the value of the neurons activation function f which has as input a weighted sum of signals $x_i, x_{i+1}, ..., x_n$ recieved by n other neurons, multiplied with the weights $w_i, w_{i+1}, ..., w_n$ and added with bieases $b_i, b_{i+1}, ..., b_n$. The exact function a varies depending on the type of non-linearity that exists in the activation function applied to the input of each neuron. However, in almost all cases a can be decomposed into a linear operation that weights the relative importance of the various inputs, and a non-linear transformation $f(z)$. As seen in equation 3.9, the linear tranformation commonly takes the form of a dot product with a set of neuron-specific weights followed by re-centering with a neuron-specific bias. A more convenient notation for the linear transformation z^i then goes as follows:

$$z^i = \boldsymbol{w}^{(i)} \cdot \boldsymbol{x} + b^{(i)} = \mathbf{x}^T \cdot \mathbf{w}^{(i)}, \tag{3.10}$$

where $\mathbf{x} = (1, \boldsymbol{x})$ and $\mathbf{w}^i = (b^{(i)}), \boldsymbol{w}^{(i)})$. The full input-output function can be expressed by incorporating this into the non-linear activation function f_i, as expressed below.

$$a_i(\mathbf{x}) = f_i(z^{(i)}). \tag{3.11}$$

3.2.1 Activation functions

Without activation functions, a neural network would essentially be a linear model, capable only of representing linear relationships between inputs and outputs. While the linear transformations occurs within individual neurons through the weighted sum of inputs, the introduction of non-linear activation functions allows the networks to capture complex relationships and patterns. With other words, activcation functions are important components of neural networks, that help the network learn by making sense of non-linear and complex mappings between input- and corresponding output values. The functions are applied at every node in the hidden layers and the output layer [**choose_activation_function**].

Activation functions in neural networks draw inspiration from the behavior of neurons in the brain. Similar to how neurons respond to incoming electrical signals, activation functions determine whether a neuron in a neural network should be activated or not based on the strength of the input. If the input exceeds a certain threshold, the neuron "fires" or becomes activated, otherwise it remains inactive [**analyticsvidhya_activationfunctions**]. By introducing nonlinearity, activation functions enable neural networks to model complex, nonlinear relationships in data.

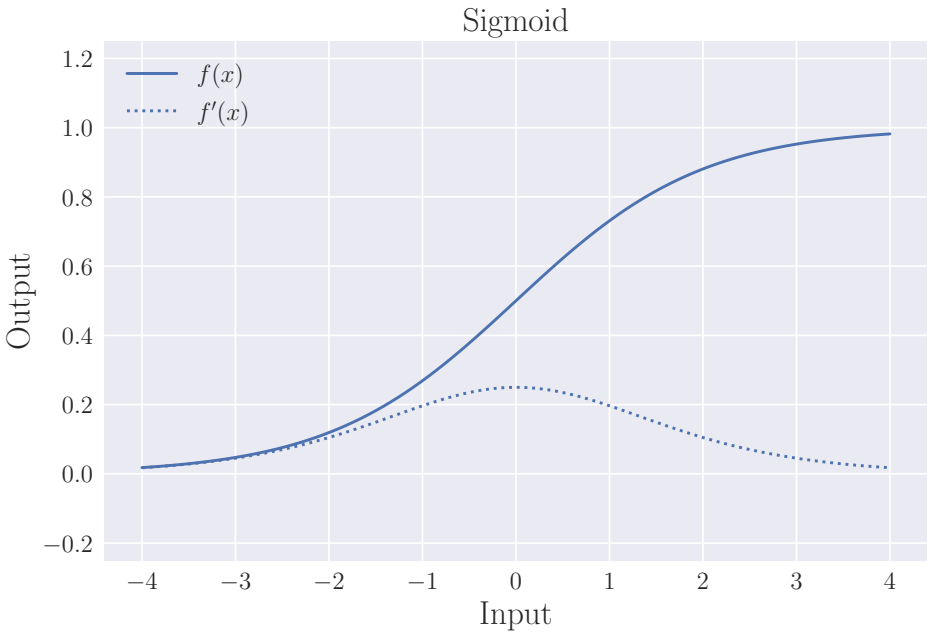

Figure 3.2: Sigmoid activation function.

Sigmoid

The sigmoid activation function is one of the more biologically plausible as the output of inactivated neurons returns zero [**Jensen2022**]. More precised it is a logistic mathematic function meaning that it maps its input to a value between 0 and 1:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{3.12}$$

The function is continuous, ensuring that it is differentiable at every point. This differentiability property plays a crucial role in effective computation of the derivative during the process of backpropagation, as we will explore in more detail in the subsequent sections of this chapter.

The sigmoid activation function maps large negative values towards 0 and large positive values towards 1. Thus, the activation function is commonly utilized in the output layers of neural networks, particularly in classification problems where the desired output can be interpreted as a class label. As we can see from figure 3.2, the function return 0.5 for an input eqal to 0. Due to this, the value 0.5 can be seen as a therhold value which decides wether the input value belongs to what type of two classes.

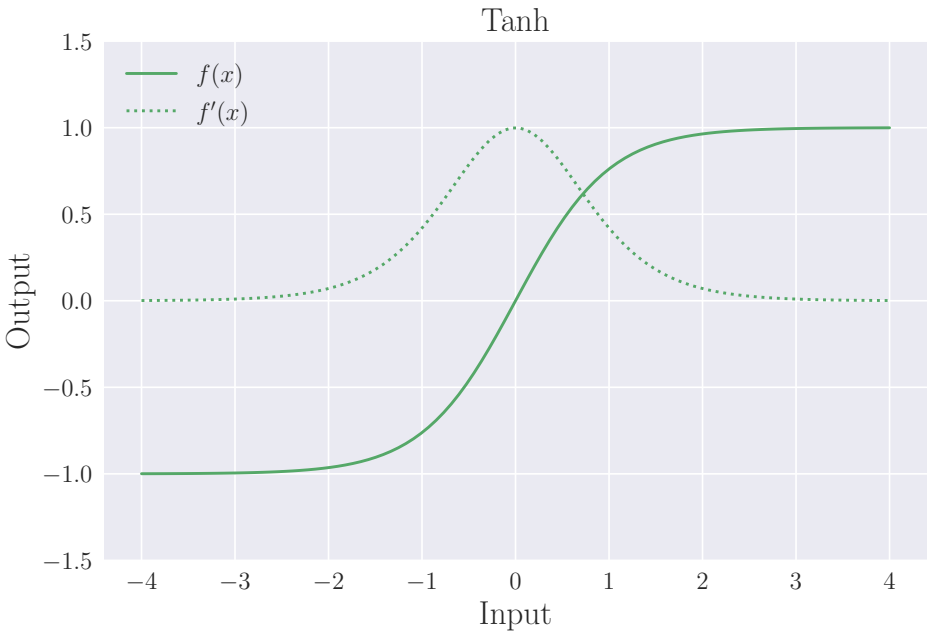

Figure 3.3: Hyperbolig tangent activation function.

Hyperbolic Tengent

The hyperbolic tangent (Tanh) is similar to the Sigmoid function, as it is continuos and differentiable at all points:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3.13}$$

However, compared to the Sigmoid function, the gradient of Tanh is steeper. Moreover this activation function maps its input to a value ranging between -1 and 1 as seen in Figure 3.3

Even though Sigmoid has its advantages, it has been shown that the Hyperbolic tangent performs better than the Sigmoid when approaching complex machine learning problems. The reasons for this will be discussed in later in this chapter.

Rectified Linear Unit

The Rectified Linear Unit (ReLU) activation function is widely recognized for its speed, high performance, and generalization capabilities [**wandb_activation_functions**]. Compared to the Sigmoid and Hyperbolic Tangent functions, ReLU may seem relatively simple, which contributes to its computational efficiency. The function can be mathematically defined as:

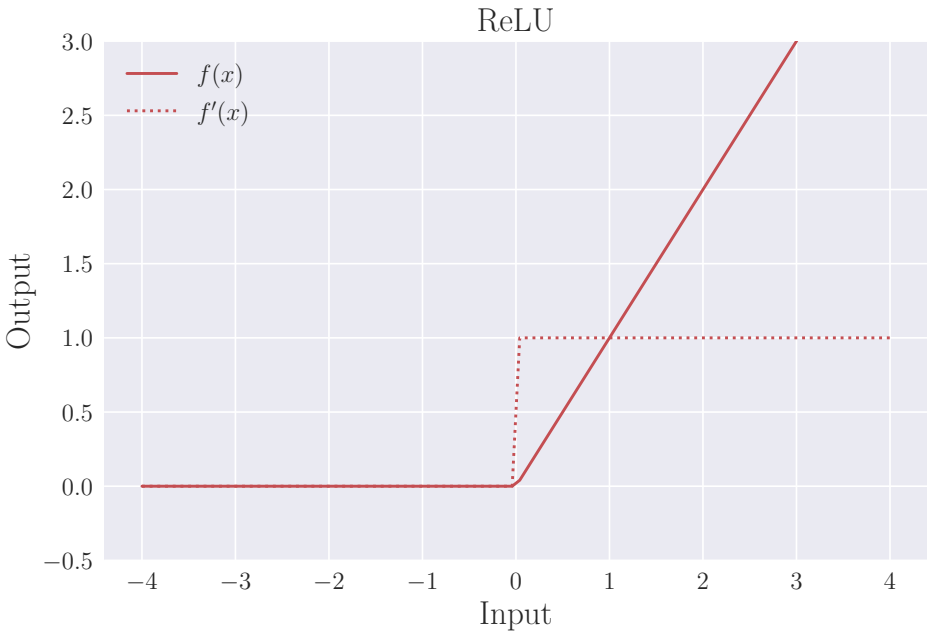

Figure 3.4: ReLU tangent activation function.

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.14}$$

From Figure 3.4, it is evident that ReLU retains the input value when the input is greater than zero, and outputs zero for negative inputs. This sparse nature of the activation function enhances computational efficiency as only a few neurons are activated at any given time.

Back propagation algorithm

The back propagation algorithm is a fundamental tequniqe used in nenural networks in order to adjust the weights for the purpose of minimizing the cost function. To explain the implementation details of this technique, we follow the guidance provided in the book 'A high-bias, low-variance introduction to machine learning for physicists' (Pankaj Mehta, et al., 2019) as it offers a comprehensive treatment of the topic. The back propogation tequniqe leverages the chain rule from calculus to compute gradients for weight adjustments and can be summarized using four equations.

Before Introducing the equations, Mehta et al. establish some useful notation. They start by concidering a total of L layers within the neural network, with each layer identified by an idec l ranging from 1 to L. For each layer, they further assign weights denoted as $\mathbf{w}_{ik}^{l}$, which represent the

connections between the k-th neuron in the previous layer, $l-1$, and the i-th neuron in the current layer, l. Additionally, they assign a bias value b_i^l to each neuron in the current layer.

The first eqation setting up the algorithm is the definition of the error δ_i^l of the i-th neuron in the l-th layer:

$$\delta_i^l = \frac{\partial C}{\partial(z_i^l)}, \tag{3.15}$$

where (z) denotes the weighted input. This equation can be thought of as the change to the cost function by increasing z_i^L infinitesimally. The cost function quantifies the discrepancy between the network's output and the target data. If the error δ_i^L is large, it indicates that the cost function has not yet reached its minimum.

The error δ_i^l can also be interpreted as the partial derivative of the cost function with respect to the bias b_i^l. This gives us the analogously defined error:

$$\delta_i^l = \frac{\partial C}{\partial(z_i^l)} = \frac{\partial C}{\partial(b_i^l)}\frac{\partial C}{\partial(z_i^l)} = \frac{\partial C}{\partial(b_i^l)} \tag{3.16}$$

where it in the last line has been used that the derivative of the activation function with respect to its input evaluates to 1, $\partial b_i^l/\partial z_i^l = 1$, meaning that the rate of change of the activation function does not depend on the specific value of the weighted input z_i^l.

By applying the chain rule, we can express the error δ_i^l in Equation 3.15 in terms of the equations for layer $l+1$. This forms the basis of the third equation used in the backpropagation algorithm:

$$\begin{aligned}\delta_i^l = \frac{\partial C}{\partial z_i^l} &= \sum_j \frac{\partial C}{\partial z_j^{l+1}}\frac{\partial z_j^{l+1}}{\partial z_i^l} \\ &= \sum_j \delta_j^{l+1}\frac{\partial z_j^{l+1}}{\partial z_i^l} \\ &= \sum_j \delta_j^{l+1} w_{ij}^{l+1} f'(z_i^l)\end{aligned} \tag{3.17}$$

Finally the last equation of the four back propagation equations the derivative of the cost function in terms of the weights:

$$\frac{\partial C}{\partial w_{ij}^l} = \delta_i^l a_j^{l-1} \tag{3.18}$$

With these four equations in hand we can now calculate the gradient of the cost function, starting from the output layer, and calculating the error of

each layer backwards. We then have a way of adjusting all the weights and biases to better fit the target data. The back propagation algorithm then goes as follows:

1. **Activation at input layer:** calculate the activations a_i^1 of all the neurons in the input layer.
2. **Feed forward:** starting with the first layer, utilize the feed-forward algorithm through **??** to compute z^l and a^l for each subsequent layer.
3. **Error at top layer:** calculate the error of the top layer using equation 3.15. This requires to know the expression for the derivative of both the cost function $C(\boldsymbol{W}) = C(\boldsymbol{a}^L)$ and the activation function $f(z)$.
4. **"Backpropagate" the error:** use equation 3.17 to propagate the error backwards and calculate δ_j^l for all layers.
5. **Calculate gradient:** use equation 3.16 and 3.18 to calculate $\frac{\partial C}{\partial z_i^l}$ and $\frac{\partial C}{\partial w_{ij}^l} = \delta_i^l a_j^{l-1}$.
6. **Update weights and biases:**
   $w_{jk}^l = w_{jk}^l - \eta \delta_j^l a_k^{l-1}$
   $b_j^l = b_j^l - \eta \delta_j^l$

Initialization of weights and biases

Sigmoid is usually not utilized in the hidden layers of networks due to vanishing or exploding gradient problems. This term is used in scenarios where the gradient becomes very small, making the optimization process slower and less effective. Such a problem hinders the convergence of the network and makes it challenging, if not impossible, for the network to learn meaningful representations from the data. Looking at the derivative of the function shown in Figure 3.2, we see that we encounter such scenarios when the input value is considerably small or large.

An important advantage of using the hyperbolic tangent function over the sigmoid function is that the tanh function is centered around zero. This makes the optimization process much easier as it ensures that the gradients calculated during backpropagation have both positive and negative values, resulting in more balanced weight updates. This, in turn, might lead to faster convergence and more efficient optimization.

The Inverse Problem

Computational neuroscience is a field that aims to understand the principles underlying information processing in the brain using mathematical

and computational tools. The inverse problem in EEG, which involves estimating the location and strength of electrical sources in the brain based on measurements of electrical activity on the scalp, is a key challenge in computational neuroscience. Machine learning techniques, including feedforward neural networks, have been used to address this problem by learning to map the measured EEG signals to estimates of the underlying electrical sources in the brain.

Source localization using machine learning techniques has shown promise for improving the accuracy and efficiency of EEG analysis, and has been applied to a variety of cognitive and clinical applications. For example, machine learning-based source localization has been used to study the neural mechanisms underlying attention, memory, and perception (Wu et al., 2018; Lopes da Silva et al., 2019), as well as to diagnose and monitor neurological disorders such as epilepsy (Safieddine et al., 2019; Shah et al., 2020). These applications demonstrate the potential of machine learning and computational neuroscience to enhance our understanding of the brain and improve clinical outcomes.

Machine learning is a field of computer science that involves using algorithms and statistical models to enable computers to learn from data without being explicitly programmed. One popular type of machine learning algorithm is the feedforward neural network, which is a type of artificial neural network that is often used for tasks such as linear regression. In a feedforward neural network, data is passed through a series of layers of interconnected nodes, or "neurons," which perform mathematical operations to transform the data.

Linear regression is a common machine learning task that involves predicting a continuous quantity, such as the price of a house or the temperature of a city, based on a set of input features. In a feedforward neural network, linear regression can be accomplished by using a single neuron in the output layer of the network that computes a weighted sum of the input features and applies an activation function to produce the predicted output value. The weights on the input features are learned by the network during the training process, which involves adjusting the weights to minimize the difference between the predicted output values and the actual output values in the training data.

Overall, feedforward neural networks are a powerful machine learning tool that can be used to solve a wide range of problems, including linear regression. By adjusting the weights and biases of the neurons in the network during the training process, neural networks can learn to make accurate predictions based on input data, making them a valuable tool for a variety of applications.

Chapter 4

Dipole Source Localization using Neural Networks

In this chapther we will be presenting the neural networks used for the localization of current dipole sources in the human cortex.

Feed Forward Neural Networks

The feedforward neural network (FFNN) was one of the first artificial neural network to be adopted and is yet today an important algorithm used in machine learning. The feed forward neural network is the simplest form of neural network, as information is only processed forward, from the input nodes, through the hidden nodes and to the output nodes [**Hjorth-Jensen2022**].

Convolutional Neural Networks

Convolutional neural networks (CNNs) is an other variant of FFNNs that have drawen inspiration from the functioning of the visual cortex of the brain. In the visual cortex, individual neurons exhibit selective responses to stimuli within small sub-regions of the visual field, known as receptive fields. This property allows the neurons to effectively exploit the spatially local correlations present in natural images. Mathematically, the response of each neuron can be approximated using a convolution operation [**Hjorth-Jensen2022**].

CNNs mimic the behavior of visual cortex neurons by utializing a specific connectivity pattern between nodes in adjacent layers. Unlike fully contected FFNNs, where each node connects to all nodes in the preceding layer, CNNs local connectivity. In other words, each node in a convolutional layer is only connected to a subset of nodes in the previous layer. Typically, CNNs consist of multiple convolutional layers that learn local features from the input data. These layers are followed by a fully connected layer that combines the learned local information to produce the final outputs. CNNs find wide applications in image and video recognition tasks [**Hjorth-Jensen2022**].

4.0.1 Neural Networks

Artificial Neural Networks are computational systems that can learn to perform tasks by considering examples, generally without being programmed with any task-specific rules [**101**].

The biological neural networks of animal brains, wherein neurons interact by sending signals in the form of mathematical functions between layers, has inspired a simple model for an artificial neuron:

$$a = f\left(\Sigma_{i=1}^{n} w_i x_i + b_i\right) = f(z) \tag{4.1}$$

where the output a of the neuron is the value of its activation function f, which as input has the sum of signals $x_i, x_{i+1}, ..., x_n$ received by n other neurons, multiplied with the weights $w_i, w_{i+1}, ..., w_n$ and added with biases.

Most artificial neural networks consists of an input layer, an output layer and layers in between, called hidden layers. The layers consists of an arbitrary number of neurons, also referred to as nodes. The connection between two nodes is associated with a weight variable w, that weights the importance of various inputs. A more convenient notation for the activation function is:

$$a_i(\boldsymbol{x}) = f_i(z^{(i)}) = f_i(\boldsymbol{w^i} \cdot \boldsymbol{x} + b^i) \tag{4.2}$$

where $\boldsymbol{w}^{(i)} = (w_1^{(i)}, w_2^{(i)}, ..., w_n^{(i)})$ and $b^{(i)}$ are the neuron-specific weights and biases respectively. The bias is normally needed in case of zero activation weights or inputs [**101**].

4.1 Feed-Forward Neural Network Approach for localizing single dipole sources

The feedforward neural network (FFNN) was one of the first artificial neural network to be adopted and is yet today an important algorithm used in machine learning. The feed forward neural network is the simplest form of neural network, as information is only processed forward, from the input nodes, through the hidden nodes and to the output nodes.

4.1.1 Validation accuracy

In Figure 6.1 we have provided the validation accuray, using mean squared error (MSE) and the coefficient of determination (R2-score).

The expression for MSE when predicting the x-, y- and z-coordinate, goes as follows:

$$MSE(\hat{y}, \hat{\tilde{y}}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2 = \frac{1}{3}\sum_{i=1}^{3}((x - \tilde{x})^2 + (y - \tilde{y})^2 + (z - \tilde{z})^2) \tag{4.3}$$

The coefficient of determination is given as follows:

$$R^2(\hat{y}, \tilde{\hat{y}}) = 1 - \frac{\sum_{i=0}^{n-1}(y_i - \tilde{y}_i)^2}{\sum_{i=0}^{n-1}(y_i - \bar{y})^2}, \tag{4.4}$$

Where the mean value of y_i is defined by $\bar{y}$:

$$\bar{y} = \frac{1}{n}\sum_{i=0}^{n-1} y_i.$$

4.1.2 Activation functions, Batchsize and Optimization

For the neurons of the input layers we use the linear activation function ReLu, while for the neurons of the hidden and output layers, we chose the much used hyperbolic tangent activation function.

Cost function In order to train the network faster, one commonly split the data set into mini-batches, which is also done here. When splitting the data such a way, the weights of connection between neurons are updated after each propagation, making the network converge considerable faster.

Scaling Every potential distribution presented to the network is first average referenced by subtracting the average of all potential values. Subsequently, the average referenced potentials are normalized by dividing them by the magnitude of the largest. The dipole location parameters are normalized to 1 with respect to the radius of the outer head boundary in the spherical head model (9.2 cm). In the case of a realistically shaped head model, the location parameters are normalized with respect to the radius of the best-fitting sphere for the scalp–air interface.

As was pointed out in the previous section, the optimal dipole orientation (in the leastsquares sense) for a given location can be calculated in a straightforward manner. Therefore, we will use neural networks to estimate only the dipole location parameters.

4.1.3 Training, testing and evaluation

The estimation is found by the network through optimizing the parameters β minimizing the cost function, or said in other words, through finding parameters for the function that produces the smallest outcomes, meaning the smallest errors. The result provided by the network is then compared with the target, for each input vector in the training data. Adjustment of parameters...

When the network is fully trained, we have a final model fit on the training data set. Feeding the network with the test data set, we can assess the performance of the network. The predictions of the fully trained network can now be compared to the holdout data's true values to determine the model's accuracy.

In figure **??** we have provided the bias-variance trade-off for when using Tanh as activation function. We notice that error of the model is approaching 0 and that the variance between the two curves decreases for an increasing number of epochs.

Chapter 5

DiLoc - A NN Apporach for Source Localization

As mentioned in Chapter 1, an important topic in EEG signal analysis is the inverse problem, which aims to map measured EEG signals to localized equivalent current dipoles. This process is commonly referred to as source localization. In this chapter, we will introduce our feedforward neural network, named "DiLoc", created to solve such prosesses. We will provide a comprehensive overview of its parameters, architecture and training process. Moreover we will go through the simulation of the EEG signals for the the final dataset used for training. Additionally, we will present an alternative approach using a convolutional neural network for the same source localization.

5.1 Architecture

The development of DiLoc commenced with a deliberate and cautious approach, focusing on simplicity without compromising on accuracy in tackling diverse versions of the inverse problem. As a natural starting point, we adopted a fully connected, feed-forward neural network architecture, which eventually proved to be the most suitable framework for our purposes.

The input layer of DiLoc incorporates Rectified Linear Units (ReLU) as the activation function. This activation function introduces non-linearity into the model, enabling it to capture complex relationships and patterns within the data effectively. For the hidden layers, we employed the hyperbolic tangent (tanh) activation function. This decision was driven by its ability to squash input values into a range between -1 and 1, ensuring a smooth and differentiable transition during backpropagation. Conversely, in the output layer, we opted for a linear transformation without the use of any activation function. This setup allows the neural network to provide direct and unconstrained predictions for the x-, y-, and z-positions of the desired

dipole source, as required in our application.

The determination of the number of hidden layers and neurons was carried out through a meticulous trial-and-error process. By experimenting with networks of varying complexities, i.e., small, medium, and large, we ultimately settled on the medium-sized network configuration. This choice, in conjunction with the selected activation functions, yielded the most promising results in terms of prediction accuracies.

The input layer is designed with 231 neurons, corresponding to the number of features in our dataset, i.e. the number of recording electrodes for each sample. Subsequently, the network consists of five hidden layers, comprising 512, 256, 128, 62, and 32 neurons, respectively. Finally, the output layer encompasses the three-dimensional coordinates (x, y, and z) representing the predicted position of the desired dipole source. Figure 5.1 visualizes the construction of the fully connected neural network.

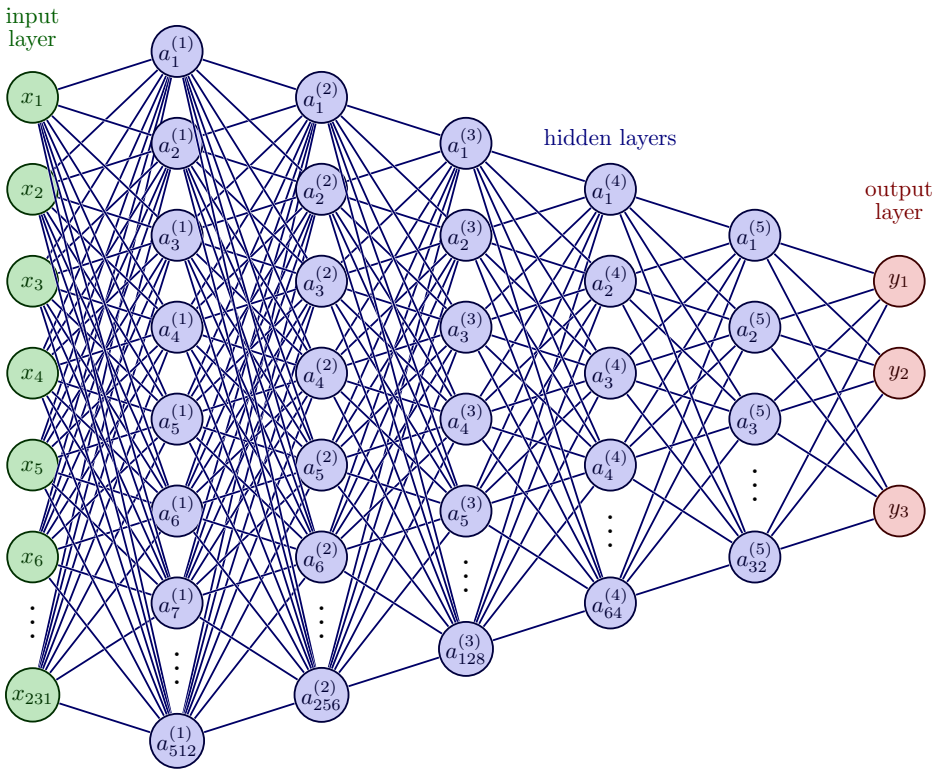

Figure 5.1: Architecture.

5.2 Simulation of EEG Signals

The cortex matrix of the New York Head Model (NYHM) consists of 74,382 points, representing possible positions for localizing dipole sources. For training the neural networks, we utilize a dataset of self-simulated EEG measurements that correspond to the electromagnetic fields generated by these dipole sources. The dipole sources are randomly positioned within the cortex matrix. To simplify our problem, the amplitude of each dipole is fixed at 10^7 nA μm. Additionally, in the case of single dipole source localization, the direction of the dipole moment is always rotated so that it is normal to the cerebral cortex. In some cases this will result in a dipole moment pointing perpendicular to the skull (directed towards an EEG electorde), while in other cases, due to the structure of the cortex, the dipole moment will point back into the cortex (but eventually towards an EEG electorde). The reason for this is that the human cortex is strongly folded, and the contribution to the EEG signal from a neural population (dipole moment) will depend on whether a dipole is located in a sulcus or a gyrus [**naess2021biophysically**]. It is important to note that the EEG recordings capture a time series; however, for our analysis, we focus solely on the signal at t = 1, corresponding to the first time step. This allows us to examine the initial snapshot of the EEG signal's spatial distribution and its relationship to the dipole source locations within the cortex.

5.2.1 The Effect of dipole location and orientation

According to Naess et al. (2021) [**naess2021biophysically**], EEG signals are not particularly sensitive to small variations in the precise location of neural current dipoles. Despite the common belief that neurons in the upper cortical layers would dominate the EEG due to their proximity to the electrode compared to neurons in deeper layers, such location differences do not significantly affect the EEG signals. This phenomenon can be explained by the fact that the low conductivity of the skull introduces a spatial low-pass filtering effect, which mitigates the impact of location discrepancies.

However, when considering the orientation of the dipoles relative to the EEG electrodes, the effect becomes noteworthy. To illustrate this, we present in Figure 5.3 the EEG signals obtained from two manually selected dipole locations within the New York head model. These dipoles are situated in a gyrus and a sulcus, respectively, resulting in distinct EEG outcomes. In general, the contribution of an individual current dipole to the EEG signal is maximized when the dipole is located perpendicularly within a gyrus, as depicted in Figure 5.3B. On the other hand, if a dipole is placed in a sulcus with a perpendicular orientation, a significant EEG contribution can still be observed, but unlike the dipole in the gyrus, it exhibits a more dipolar pattern, as shown in Figure 5.3C.

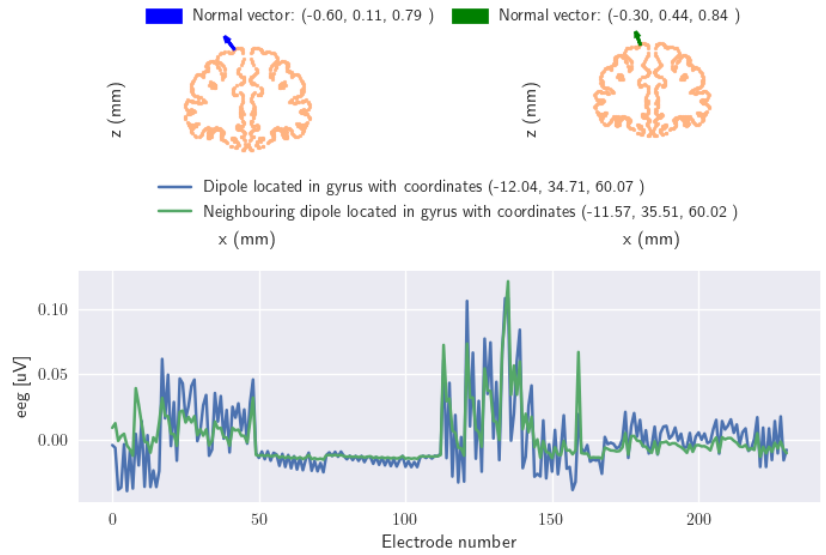

Figure 5.2: EEG signal for neighbouring dipoles.

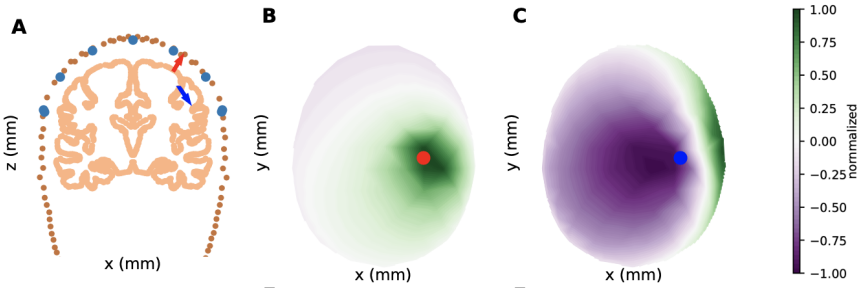

Figure 5.3: A: Two selected dipole locations in the New York head model: one in a gyrus (red) and one in a sulcus (blue). The head model is viewed from the side (x, z-plane). Close to the chosen cross-section plane, EEG electrode locations are marked in light blue. Available dipole locations near the cortical cross-section form an outline of the cortical sheet and are marked in pink. The current dipole moment for all cases was 10^7 nAμm. B: Interpolated color plot of EEG signal from the gyrus dipole, viewed from the top (x, y-plane). The plotted EEG signal is normalized, with a maximum value of 1.1 μV. C: Interpolated color plot of EEG signal from the sulcus dipole. The plotted EEG signal is normalized, with a maximum value of 0.7 μV. [**naess2021biophysically**]

With other words, the EEG outcome is genuinely influenced by the orientation of the current dipole moment generating the signal, as variations in dipole orientation can impact both the direction and distribution of electrical potentials. In Figure 5.4, we present the EEG signals obtained from four identical current dipoles with different orientations, situated in distinct hypothetical folding patterns of the cortical surface. Firstly, Figure 5.4A and 5.4C provide an expanded illustration of the aforementioned scenarios, incorporating additional dipole moments located in a gyrus and a sulcus, respectively. In Figure 5.4B, where a collection of dipoles points randomly upwards and downwards, the EEG signal contribution appears to diminish significantly. Conversely, when the dipoles align in the depth direction of the cortex and are distributed across both gyrus and sulcus, we can expect an EEG contribution in between what we saw from Figure 5.4A and 5.4B, as depicted in Figure 5.4D. Lastly, Figure 5.4E demonstrates the minimal EEG contribution observed when the dipoles are divided between two opposing sulci.

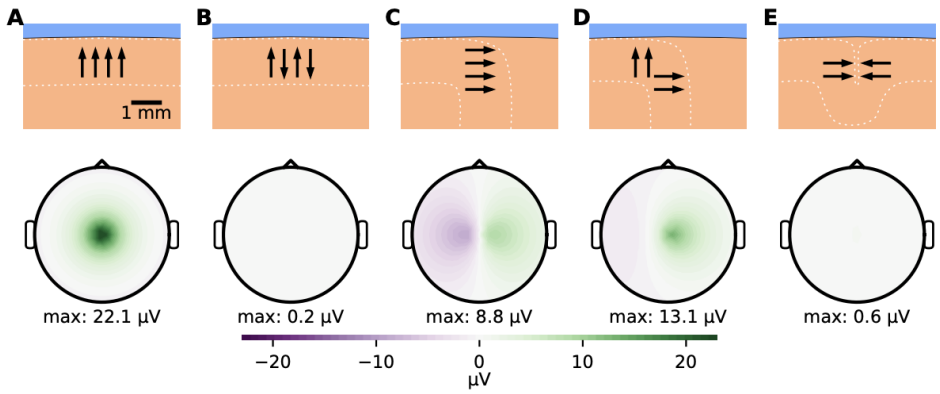

Figure 5.4: Different folding patterns of the cortical surface are represented by white dashed lines. EEG signals are calculated from four identical current dipoles with varying orientations. A: Dipoles aligned in the same direction within a gyrus. B: Dipoles pointing in opposite directions within a gyrus. C: Dipoles aligned in the same direction within a sulcus. D: Dipoles distributed between a gyrus and a sulcus, pointing towards the cortical surface. E: Dipoles divided between opposing sulci, pointing towards the cortical surface. All panels have a dipole moment magnitude of 10 nAm, and the dipoles are positioned at the centers of the arrows in the top row.

5.2.2 Noise

Experimental EEG recordings inevitably contain noise, which can interfere with the accurate analysis of brain activity. Artifacts, which are signals

recorded by EEG but originating from sources other than the human brain, pose a particular challenge. Some artifacts can mimic genuine epileptiform abnormalities or seizures, underscoring the importance of identifying and distinguishing them from true brain waves [**sazgar2019eeg**].

Artifacts can be classified into two categories based on their origin. Physiological artifacts arise from the patient's own physiological processes, including ocular activity, muscle activity, cardiac activity, perspiration, and respiration. Technical artifacts, on the other hand, originate from external factors such as cable and body movements or electromagnetic interferences [**bitbrain**].

Filtering techniques are commonly employed to remove artifacts from EEG recordings prior to analysis. However, in the case of simulated EEG data, the need for artifact removal is eliminated as the data inherently lacks noise. Simulated EEG data can be considered as pre-filtered and preprocessed, ensuring a high signal-to-noise ratio (SNR) [**wiki-snr**]. Nevertheless, to avoid overfitting and account for technical considerations, it is necessary to introduce noise to the data before feeding it into the neural network. This introduction of the noise is vital in order to make the trained neural network more likely to accurately handle real EEG recordings.

In our approach, we recognize that the introduction of noise to the simulated EEG data is an essential step to enhance the robustness of the trained neural network and ensure its ability to handle real EEG recordings effectively. Although the specific characteristics and quantity of noise have not been the primary focus of our study, we have opted for a straightforward approach. Our final dataset incorporates normally distributed noise with a mean of 0 and a standard deviation equal to 10% of the standard deviation observed in the simulated EEG recordings. By introducing this noise, we introduce random variations around each data point while preserving the overall normalization properties of the dataset.

5.2.3 Final Dataset

The final dataset comprises 70 000 rows, where each row corresponds to a single sample or patient. Within the dataset, there are 231 columns representing the features, which denote the EEG measurements recorded at each electrode. Consequently, the design matrix has a size of 70 000 x 231. In practice, the design matrix consists of pairs of input vectors with EEG signals and corresponding output vectors, where the answer key is the x-, y- and z coordinate of different dipole sources.

Figure 5.5 presents an example of the input EEG data for a single sample, with 10% noise added. The illustration showcases the EEG results obtained from a sample containing a solitary current dipole source positioned randomly within the cerebral cortex. The dipolar pattern in the figure indicates that the dipole is located within a sulcus. The EEG measure is visualized

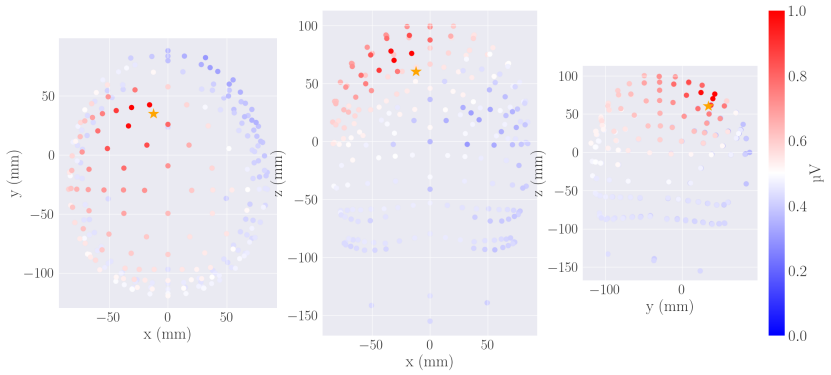

Figure 5.5: EEG for a sample containing one single current dipole source at a random position within the celebral cortex. As for all samples, 10 percent of normally distributed noise has been added to the original signal. The EEG measure is seen from both sides (x-, z-plane and y-, z-plane) and above (the x-, y-plane). EEG electrode locations are presented as filled circels, where the color of the fill represents the amplitude of the measured signal for the given electrode. The position of the current dipole moment is marked with a yellow star.

from multiple perspectives, including the x-z plane, y-z plane, and the x-y plane. The electrode locations are represented by filled circles, with the color of the fill indicating the amplitude of the measured signal at each electrode. The position of the current dipole moment is denoted by a yellow star. As observed from the figure, the EEG signal for this specific sample ranges from -1 to 1 μV.

Prior to being fed into the DiLoc network for training, the dataset was splittied into distinct segments: the train, validation, and test sets. This partitioning is vital for assessing and optimizing the network's performance. Among the 70 000 samples in the final dataset, 50 000 samples are designated for the train and validation data. To ensure a representative and unbiased allocation, 80 percent of these 50 000 samples are randomly assigned to the training set. This training set serves as the core data that the network utilizes during the training process. The remaining 20 percent of the 50 000 samples form the validation set. This set plays the role in preventing overfitting, the phenomenon where the network becomes excessively attuned to the training data and consequently performs poorly on new data. By independently evaluating the model's performance on the validation set throughout training, we can fine-tune the network's parameters to achieve better generalization to unseen data. Once the network completes its training process, the test set comes into play. Comprising 20 000 samples, the test set serves

as the benchmark for assessing the model's ability to generalize and make accurate predictions on new data instances. By adhering to this rigorous train-validation-test data partitioning, we ensure a robust evaluation of the DiLoc model's performance and its capacity to effectively handle real-world scenarios with previously unseen data.

5.3 Training the DiLoc Network

To train the DiLoc network efficiently, several key techniques are employed, including stochastic gradient descent (SGD), mean squared error (MSE) as the cost function, learning rate scheduling, and L1 and L2 regularization. Detailed explanations of these techniques can be found in the Chapter 3.

The objective during training is to find the optimal parameters β that minimize the cost function. The cost function represents the discrepancy between the network's predictions and the actual target values. By iteratively updating the parameters to minimize the cost function, the network fine-tunes its internal representations to make more accurate predictions. MSE is chosen as the cost function for the DiLoc network, as it provides a smooth and continuous measure of the model's performance during training, penalizing larger errors more heavily.

Optimizers play a crucial role in reducing the network's loss and providing accurate results. In this case, SGD with momentum is utilized as the network's optimizer. SGD with momentum enhances the sensitivity of the network to initialized weights and provides fast convergence. The algorithm uses mini-batches of size 32, introducing fluctuation to the data and preventing the network from getting stuck in local minima or saddle points. The momentum hyperparameter, set to 0.35 in this context, helps reduce high variances in the optimization process and accelerates convergence towards the right direction, leading to faster training.

To improve the training process further, learning rate scheduling is employed. This technique adjusts the learning rate over time, allowing the network to take larger steps in the early stages and gradually decrease the learning rate as it approaches convergence. The initial learning rate is set to 0.001, and is further decreased, which provides balance between rapid convergence in the initial phases and fine-tuning towards the end.

Additionally, L1 and L2 regularization techniques are incorporated as optional parameters into the DiLoc network. These regularization methods help prevent overfitting and improve generalization to unseen data. By adding penalty terms to the cost function, L1 and L2 regularization encourage the model to favor simpler and more generalizable solutions.

After the DiLoc network is fully trained on the training dataset, it has learned the optimal parameters to make accurate predictions. The model's performance is evaluated using a separate test dataset, which the network

has not seen during training. This test data provides an unbiased assessment of the model's accuracy and its generalization capabilities to unseen data.

In the upcoming chapters, we will present different approaches to the inverse problem and showcase the performance of the DiLoc network across these approaches. The evaluation results will demonstrate the effectiveness and utility of the trained model in solving the localization task for various scenarios.

5.4 Diloc as Convolutional Neural Network

Chapter 6

Localizing Single Dipole Sources

In this chapter we will present the results from training and performance of the neural networks presented in chapter 4. Section 1 deal with the results and discussion of the simple feed forward neural network, while section 2 will discuss how the alternative convolution neural network performe some of the same results.

6.1 Localizing Single Dipole Sources

We begin by introducing the standard inverse problem for our neural network, DiLoc. In this context, the standard inverse problem refers to the task of predicting the x-, y-, and z-coordinates of dipole current sources responsible for generating measured EEG signals. The goal is to feed the network with EEG data corresponding to the electrical activity from randomly distributed dipoles in the cerebral cortex and have the network accurately output the locations of these current dipoles.

For this specific problem, the network demonstrates remarkable performance even without the use of L1 regularization. However, we include L2 penalty with a value of 0.5 to promote more generalizable solutions. The network is trained for 500 epochs, with each epoch completing in approximately 11.5 seconds. It is worth noting that the validation loss does not decrease significantly after approximately 350 epochs. As a result, fully training the network (for 350 epochs) would not require more than 4025 seconds, or roughly 1 hour and 7 minutes. Despite the validation loss stabilizing, we continued training for the full 500 epochs to ensure that there would be no further improvements in the validation data's performance. By training for a few more epochs beyond the point of loss stabilization, we could confirm that the network had reached its convergence and had effectively learned to generalize well on the given task.

Figure 6.1 illustrates the network's loss as a function of training epochs. A clear trend of decreasing loss can be observed, indicating the network effectively learns the patterns in the data. The validation loss stabilizes around 350 epochs, while the training loss continues to decrease until a point between 400 and 500 epochs. Additionally, Figure 6.2 provides insight into the development of the validation loss for separate target coordinates plotted against training epochs. The figure most of all confirms that all separate target coordinates have been equally weighted, resulting in similar loss values for each of them. Moreover, it showcases that the small fluctuations in the loss, noticeable before 350 epochs, disappear beyond this threshold, indicating a stabilization of the loss for all three target coordinates.This observation aligns with the trend of the validation loss stabilizing at approximately 350 epochs as seen in the previously mentioned figure.

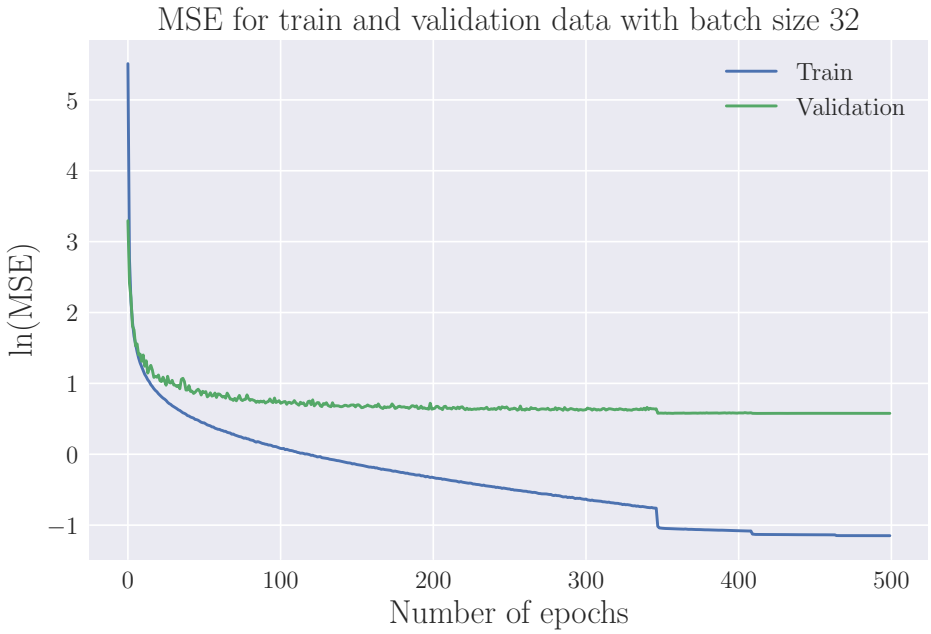

Figure 6.1: Training- and validation loss for DiLoc with 50 000 samples and tanh as activation function.

The DiLoc network's performance is evaluated using a variety of error metrics for the x-, y-, and z-coordinates. The x-coordinate ranges from -72 to 72 mm, the y-coordinate from -106 to 73 mm, and the z-coordinate from -52.66 to 81.15 mm.

Table 7.1 presents the results for the mean absolute error (MAE) in the DiLoc network's predictions. The MAE values for the x-, y-, and z-coordinates range from 0.645 mm to 0.678 mm. These findings indicate that, on average, the network's predictions exhibit an error smaller than

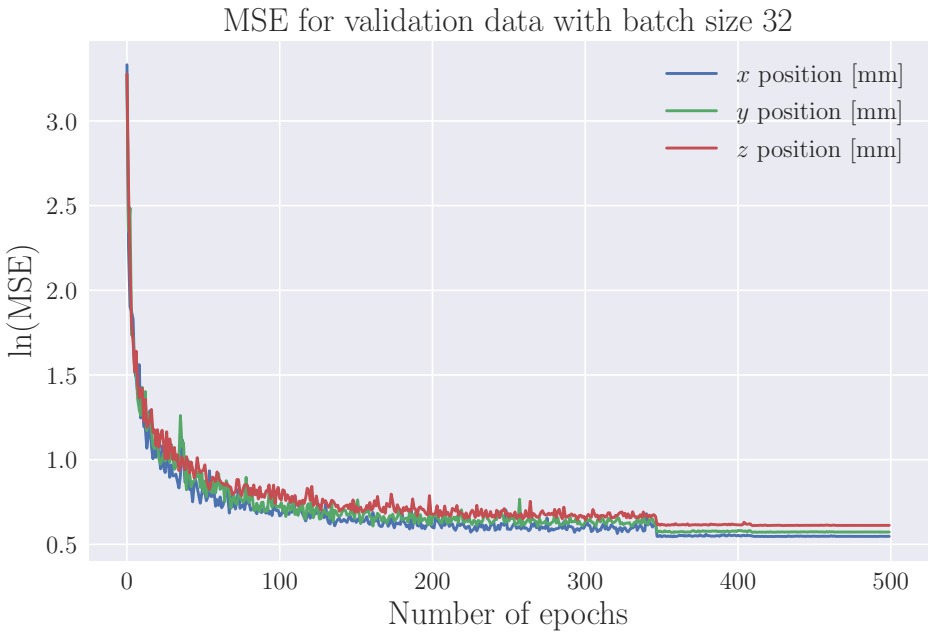

Figure 6.2: Validation loss for the separate target values; the x-, y-, z-coordinate.

1 mm in each coordinate, showcasing a high level of accuracy. The MAE metric is robust and resilient to outliers, making it a suitable measure for assessing the network's general performance.

The mean squared error (MSE) values, ranging from 0.747 mm^2 to 0.824 mm^2, provide a measure of the average squared difference between the predicted and true values. Due to its squared nature, MSE penalizes outliers more significantly compared to MAE. Smaller MSE values signify improved performance, and all MSE values in our evaluation are below 1 mm^2. This observation highlights the network's remarkable precision, particularly when considering the broad range of coordinates involved. The small MSE values indicate the network's ability to provide accurate predictions even for data points that might be considered as "outliers," further demonstrating its robustness.

The root mean squared error (RMSE) values, ranging from 0.864 mm to 0.908 mm, represent the average magnitude of errors in the original units (mm). RMSE is the square root of MSE and is slightly higher than the corresponding MSE values, as taking the square root of numbers smaller than 1 results in slightly higher values. Nevertheless, all RMSE values are below 1 mm, further attesting to the network's exceptional accuracy. RMSE provides a measure of the standard deviation of errors around the mean and complements the MSE by assessing the spread of errors in the original units.

The error metrics for the Euclidean distance, derived from the three-dimensional space coordinates, are also calculated and presented in Table 7.1. The values for MAE, MSE, and RMSE are smaller than 1 mm, indicating accurate predictions by the DiLoc network for the inverse problem. The performance metrics for the Euclidean distance corroborate the network's ability to predict the dipole location with a high level of precision and accuracy.

It is worth mentioning that among the three coordinates, the z-coordinate exhibits the highest error values. This observation suggests that the DiLoc network encounters more challenges in accurately predicting the z-coordinate of the dipole source. One plausible explanation for this discrepancy could be attributed to the nature of the inverse problem, where EEG patterns for dipole sources do not produce significant changes in the pattern of electrical potential recording, but rather in magnitude. Additionally, the smaller representation of z-values compared to x- and y-coordinates could contribute to the consistent larger errors in the z-direction. However, despite these challenges, the overall error metrics indicate that the DiLoc network is capable of predicting the dipole location with a reasonable level of accuracy, signifying its practical viability for real-world applications.

	Error for different target values			
	x-coordinate [mm]	y-coordinate [mm]	z-coordinate [mm]	Euclidean Distance [mm]
MAE	0.645	0.665	0.678	0.662
MSE	0.747	0.775	0.824	0.782
RMSE	0.864	0.880	0.908	0.884

Table 6.1: **Evaluation of the DiLoc performance utializing different Error Metrics.**
Network performance on test dataset consisting of 20000 samples. The errors are measured using Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE).

In order to conduct a detailed analysis of the network's performance, Figure 6.3 presents the Mean Squared Error (MSE) for various dipole locations within the New York head model cortex matrix. The figure provides valuable insights into the distribution of errors across different regions of the cortex, with three cross-sections—front, top, and side—depicted for examination. It is important to note that these cross-sections include data points from the training, validation, and test datasets, making these results indicative of the network's overall performance rather than real-world scenarios. However, the analysis aims to examine the distribution of errors and identify potential areas where the network's performance may be weaker, helping to gain valuable insights into its predictive capabilities.

The MSE values presented in the panels are consistently below 1 mm, which indicates a high level of accuracy in the network's predictions. These results are promising and demonstrate the network's ability to estimate dipole locations with a high level of precision. The panels also offer an opportunity to assess whether the network performs differently for dipoles located in the gyrus compared to the sulcus.

Initially, it might be assumed that EEG signals originating from dipoles in the sulcus present greater challenges for the network's analysis and prediction. This assumption is based on the deeper placement of dipoles within the sulcus compared to those in the gyrus, as well as the potential complexities introduced by the dipole's orientation within the cortex. However, upon closer examination of Figure 6.3, it becomes evident that the distribution of MSE values does not exhibit a clear correlation with the brain's structural characteristics. The MSE values appear to vary randomly across different regions, indicating that the network's performance is not significantly influenced by the distinction between the gyrus and sulcus.

Surprisingly, the Mean Squared Error (MSE) for all data points where dipoles are located in sulci is reported to be 1.283 mm, which is even smaller than for dipoles in the gyrus, where the MSE measures 1.349 mm. This observation challenges the initial assumption and indicates that the network demonstrates exceptional accuracy in predicting dipole locations, irrespective of their placement within the cortex. The small Mean Absolute Error (MAE) values further underscore the network's remarkable capacity to effectively capture the intricate features and variations associated with deeper cortical placements. These findings not only attest to the network's robustness but also reinforce its potential for precise dipole localization within the human brain across different cortical structures.

Furthermore, the figures reveal a noticeable concentration of data points with red and yellow marks, indicating higher Mean Squared Error (MSE) values, in the deeper locations within the cortex. This observation partially aligns with the slightly higher error values for the z-coordinate, as presented in Table 7.1, and is consistent with the theory related to the nature of the inverse problem. According to this theory, EEG patterns for dipole sources do not cause substantial changes in the pattern of electrical potential recording; instead, they primarily influence the magnitude of the signals. The presence of higher MSE values in deeper regions might be attributed to the decreasing signal-to-noise ratio of EEG signals originating from these cortical areas.

In conclusion, the detailed analysis of the network's performance through cross-sectional representations provides valuable insights into its predictive capabilities. The consistently low MSE values across different cortical regions demonstrate the network's remarkable accuracy in estimating dipole locations. Moreover, the absence of a clear correlation between MSE values and brain structural characteristics suggests that the network performs robustly across diverse cortical structures. These results have significant im-

plications for the network's potential clinical and research applications, as it showcases its ability to accurately predict dipole locations within the human brain, regardless of their depth and orientation within the cortex.

6.2 Convolution Neural Network Approach for localizing single dipole sources

Some results for the prediction of location for single current dipoles.

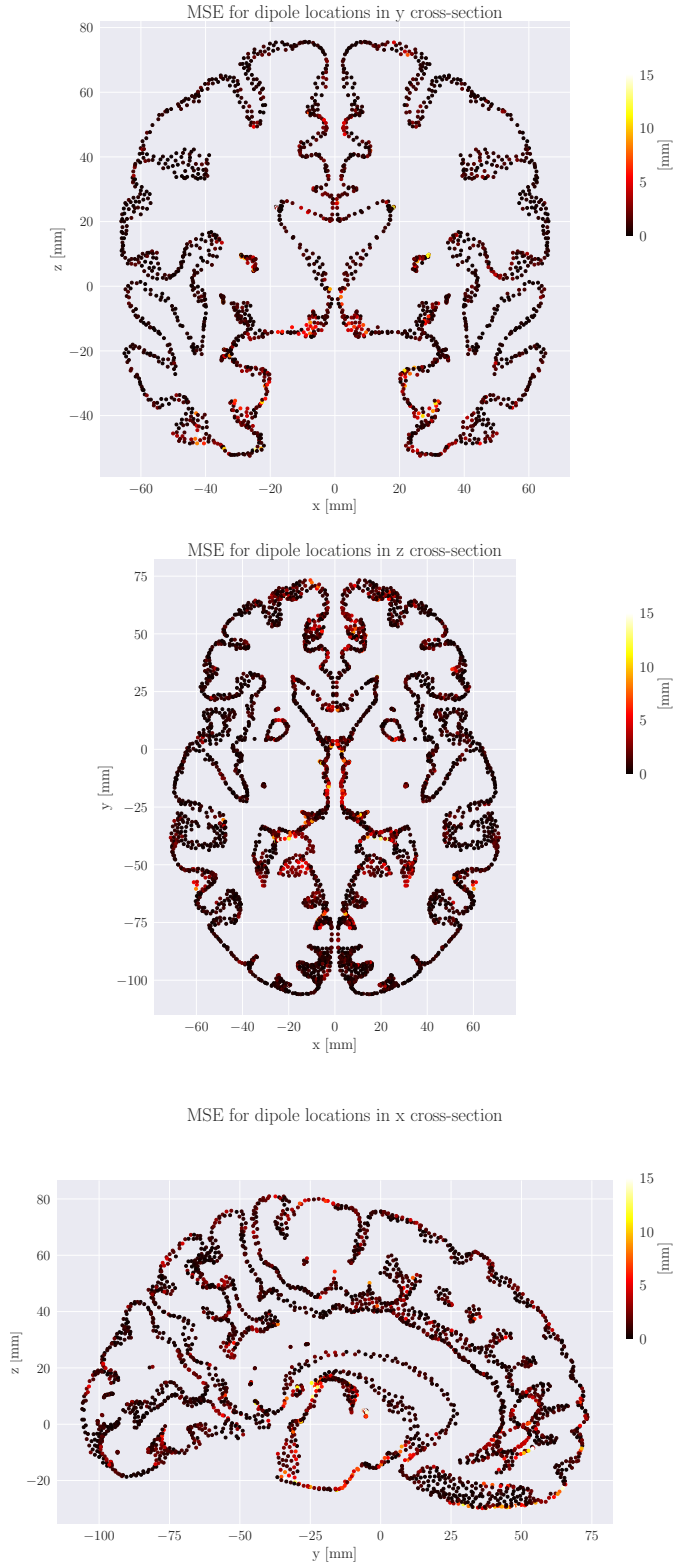

Figure 6.3: Different cross-sections of the cortex from the New York head model, seen from front, top and side. Each point represents a possible position in the cortex matrix. The color of the each point indicates the mean absolute error (MAE) of the neural network when predicting that specific dipole location.

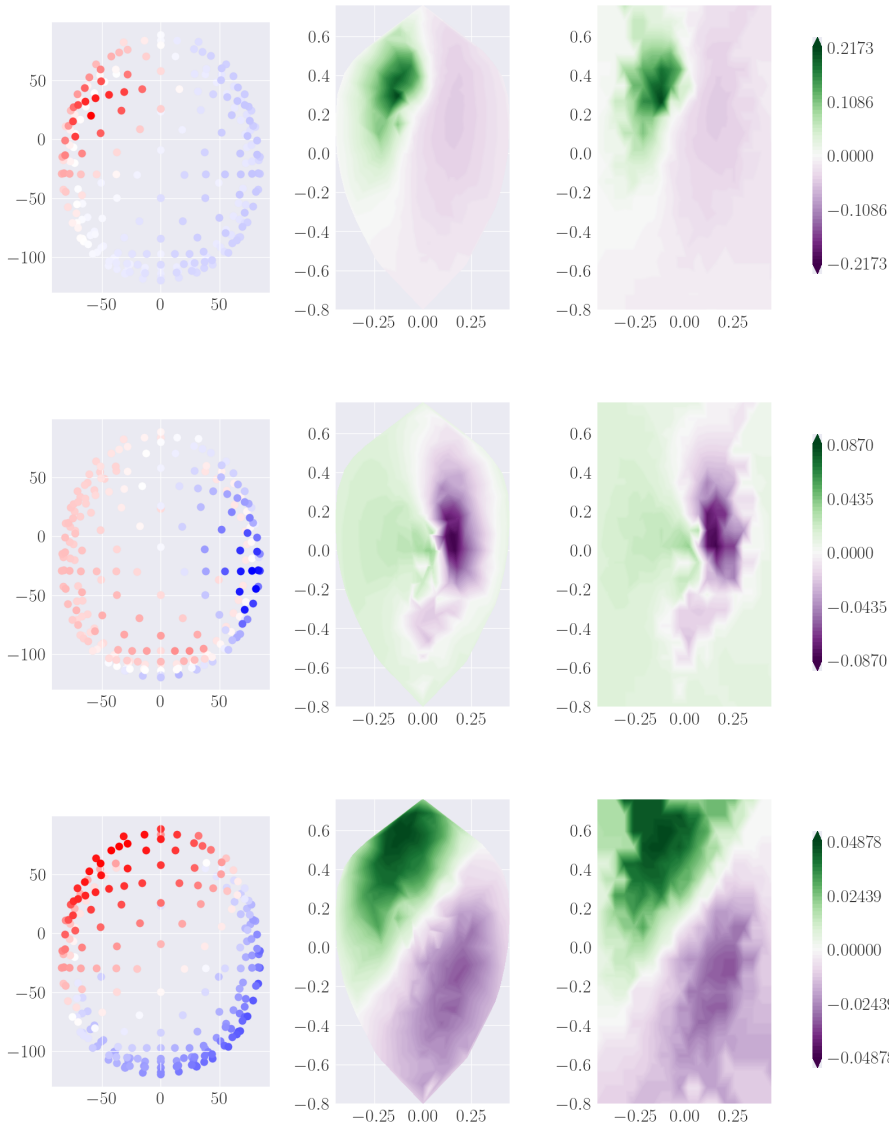

Figure 6.4:
Right: EEG measure for 3 different samples measured in μV.
Middle and Left: Illustration of the interpolation of the EEG data into two-dimensional matrix.

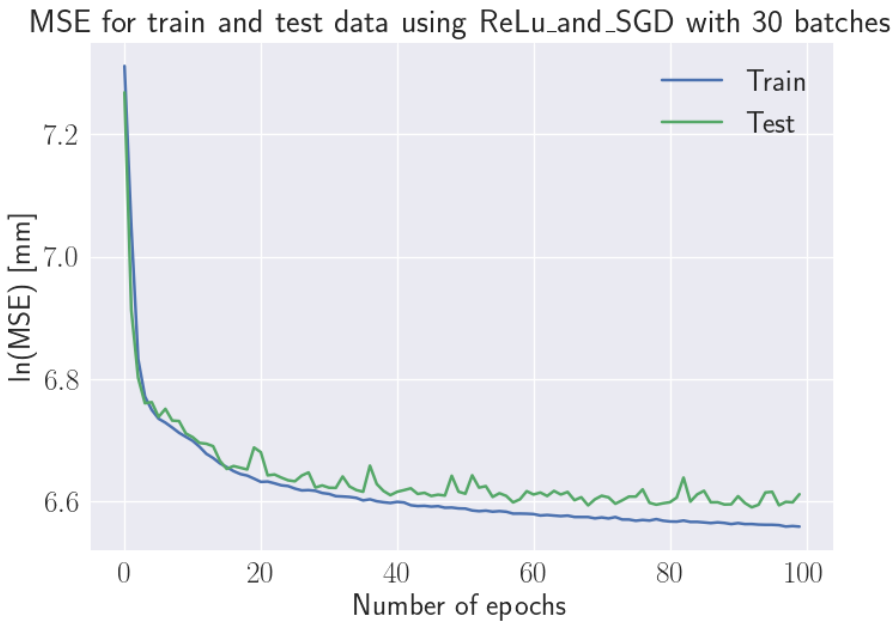

Figure 6.5: The validation accuracy for Convolutional Neural Network with 10 000 samples (20x20 matrix) with ReLU activation function.

Chapter 7

Extension of the DiLoc Network

7.1 Predicting Single Dipole Sources with Amplitudes

In the next problem, we introduce the concept of various amplitudes for single current dipole sources, which adds an additional dimension to our network's output. Besides predicting the coordinates of the dipoles for each sample, the network now also estimates the magnitude of the dipole signals. In real-world scenarios, it might be of interest to not only pinpoint the source of the abnormal activity but also comprehend the extent of abnormality. By incorporating amplitude prediction into our network, we gain valuable insights into the problem at hand and achieve a deeper understanding of the underlying brain activity.

We assign amplitudes to each dipole ranging between 1 and 10 nAμm. Although the dataset still has the same number of features, the number of target values increases by 1. Figure 7.1 provides two examples from the dataset, where the dipole location remains constant while the amplitude of the dipole signal varies. In such cases, the shape of the EEG signal will remain consistent, but the magnitude of the EEG signal will be highest for the dipole with the largest amplitude.

We start off by introducing the extensions of out DiLoc network, that now consists of 7 hidden layers. Whereas the DiLoc neural network introduced for the purpose of localizing single current dipoles had an decreasing architecture with an increasing number of layers, the new DiLoc network now has a slightly more complicated structure. In figure 7.2 we have depiced the construction of the fully connected neural network. We see that DiLoc still takes an input of 231 data points corresponding to the number of recoring electrodes, however, the number of output nodes is increased by 1; x-coordinate, y-coordinate, z-coordinate and amplitude corresponding to

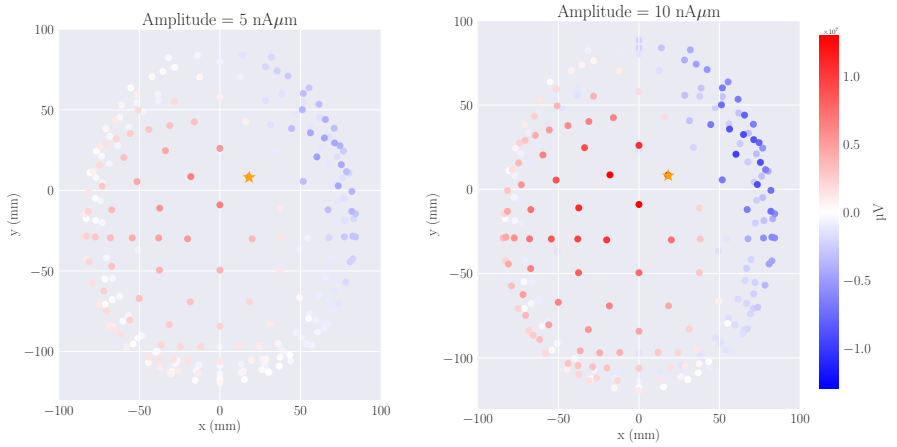

Figure 7.1: EEG data for two samples with current dipole amplitude equal to 5 and 10.

the strenght of the signal for the current dipole moment in the cortex.

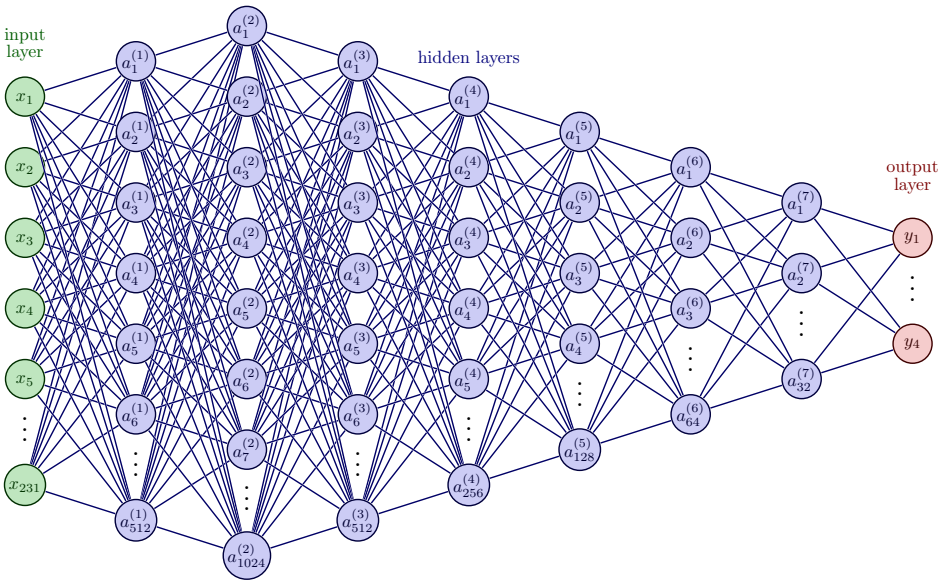

Figure 7.2: Architecture.

In order to make the network able to genralize well and discorver the important patterns, we normalize not only the input data, but also the target data, before training. The scaling provides a set of target values ranging from 0 to 1:

DiLoc for localizing current dipoles with amplitude	
Hyperparameters	**Value**
Hidden layers	6
Optimizer	SGD
Learning rate (initial)	0.001
Momentum	0.35
Weight decay	0.1
Minibatch size	64
Epochs	1000
Dropout	0.5

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{7.1}$$

where z_i is the i^{th} normalzed value in the dataset for one specific target category, x_i is the i^{th} value in the corresponding target data set, and $\min(x)$ and $\max(x)$ is the minimum and maximum value in the specific target data set. This normalization has to be done for each target category separatly. The normalization of the 4 different set of values in the target data set makes the network able to train and pick up on patterens using only one cot function. Without any form of normalization, using only one single cost function for a set of different target values with different units would not be possible.

In the extention of the DiLoc network, we continue using ReLU as activation function in the first layer, and furter the hyperbolic tangent for the hidden layers. However, as we have normalozed the output data to range from 0 to 1, it is now appropriate to utilize the Sigmoid activation function in the output layer. This activation function maps the output values to a value between 0 and 1, which is simply what we want (making the training process faster and more accurate(?)). As for the simple DiLoc model, we still use the tequniqe of adaptive learing rate. In table **??** we have provided the different parameters used in the model.

To assess the network's performance, we analyze the accuracy in relation to training epochs, as depicted in Figure 7.3. It is important to note that the target values have been normalized, resulting in a unitless loss measurement. Therefore, the figure provides a qualitative representation of the network's training progress rather than precise loss values. The plot clearly demonstrates a consistent pattern of decreasing loss as the number of epochs increases, indicating that the network effectively captures the underlying data patterns. Moreover, both the training and validation loss stabilize after approximately 2000 epochs, suggesting that the network may have reached its optimal performance level. In Figure 7.4, we present the loss development for different target values. Once again, we observe that the loss stabilizes

at around 2000 epochs. Notably, for the x-, z-, and y-coordinates, the stabilized loss corresponds to the minimum value reached during the training period. However, for the amplitude target value, this is not the case. From the provided figure, it is apparent that the smallest loss value occurs in a sharp dip just before reaching 500 epochs. This observation implies that if we solely aimed to minimize the loss function with respect to the amplitude value, the most optimal model would have emerged by terminating the training process at that epoch. However, since our objective is to develop a model that accurately predicts both the location and amplitude of the current dipole, we strive to train the model until the total loss is minimized, encompassing both aspects.

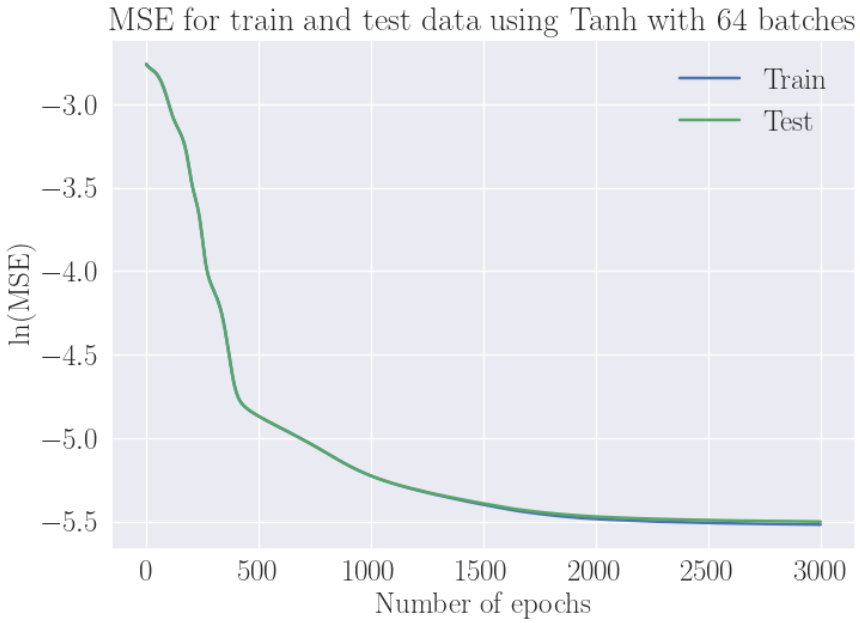

Figure 7.3: The loss for the extended DiLoc network with 50 000 samples and hyperbolic tangent activation function.

In Table 7.1 we have provided the performance of the network by concidering different error metrics. The mean absolute error (MAE) values for the x-, y-, and z- coordinates rnage from 0.8300 mm to 0.8998 mm. This means that, on average, the network's predictions have an error smaller than 1 mm in each coordinate. Considering the range of the coordinates, the MAE values represent a reasonable level of accuracy. The mean squared error penalizes larger errors/outliers more severely than MAE since it involves squaring the differences. In our case the MSE values for the different coordinates range from 1.2134 mm to 1.4110 mm. The hugher MSE values suggest that the predictions of the network may have larger errors in some

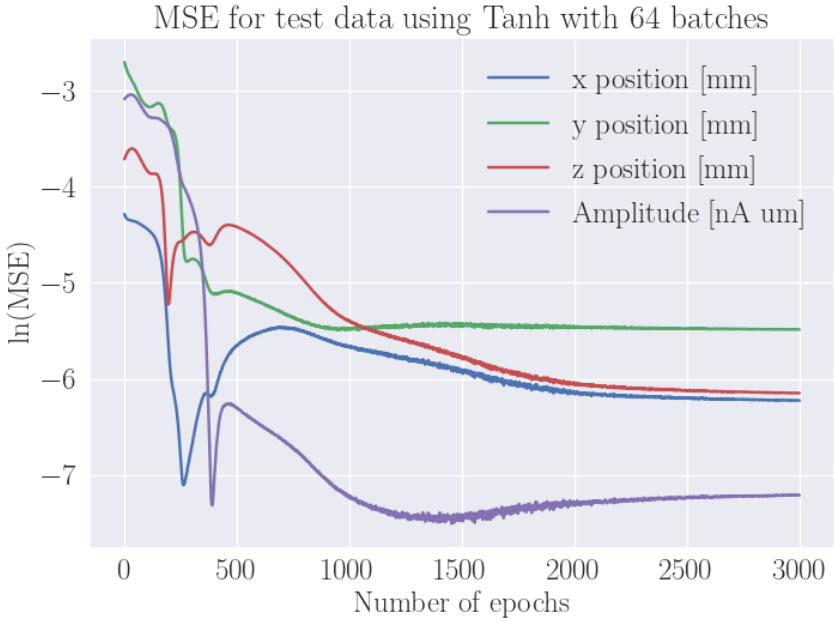

Figure 7.4: The loss developmnent for the different target values as function of epochs.

cases, resulting in a higher average squared difference. However, the magnitude of the mean squared errors is still whithin a reasonable range when analyzing them in the context of the coordinates ranges. Finally the root mean squared error provides a measure of the standard deviation of the errors and helps to understand the spread of errors around the mean. The RMSE values of ours are slighly lower than the corresponding MSE values with a range from 1.1016 mm to 1.1878 mm. The table also presents the error metrics calculated for the euclidean distance. For both MAE, MSE and RMSE the value is higher than the individual coordinate errors, indicating that the errors in the x, y, and z coordinates are not perfectly aligned and contribute to the overall distance. It is worth mentioning that specific points in the cortex matrix may potentially contribute more to the errors. Further investigation could be performed to identify any specific patterns or regions in the cortex that exhibit higher error rates. However, overall the results indicate that the network is able to predict the dipole location with reasonable accuracy. While there are some errors in the predictions, the errors are generally within an acceptable range.

	Error for different target values				
	x-coordinate [mm]	y-coordinate [mm]	z-coordinate [mm]	Euclidean Distance [mm]	Amplitude [nAμm]
MAE	3.627	4.006	3.476	2.949	0.687
MSE	22.595	28.128	22.006	18.410	0.687
RMSE	4.753	5.306	4.691	4.291	0.938

Table 7.1: **Evaluation of the network performance utializing different Error Metrics.**
Performance for the extended DiLoc network on test dataset consisting of 1000 samples. The errors are measured using Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE).

7.2 Predicting Region of Active Correlated Current Dipoles with Amplitudes

In order to further enhance the complexity of our problem, we introduce jet another extension to the DiLoc neural network. By incorporating varying radii and amplitudes for the origins that generate the electrical activity detected by the recording electrodes we transforms the objective of the DiLoc network. This transforms the objective of the DiLoc network from predicting the location of individual current dipole moments to estimating the centers of larger spherical populations. This extension is valuable for the case of real-life scenarios where understanding the extent of brain damage causing abnormal activity of damaged areas may be of interest. Training the DiLoc network on such complex data, we aim to enhance its ability to generalize and serve well to real-world clinical cases.

7.3 Localizing Multiple Dipole Sources

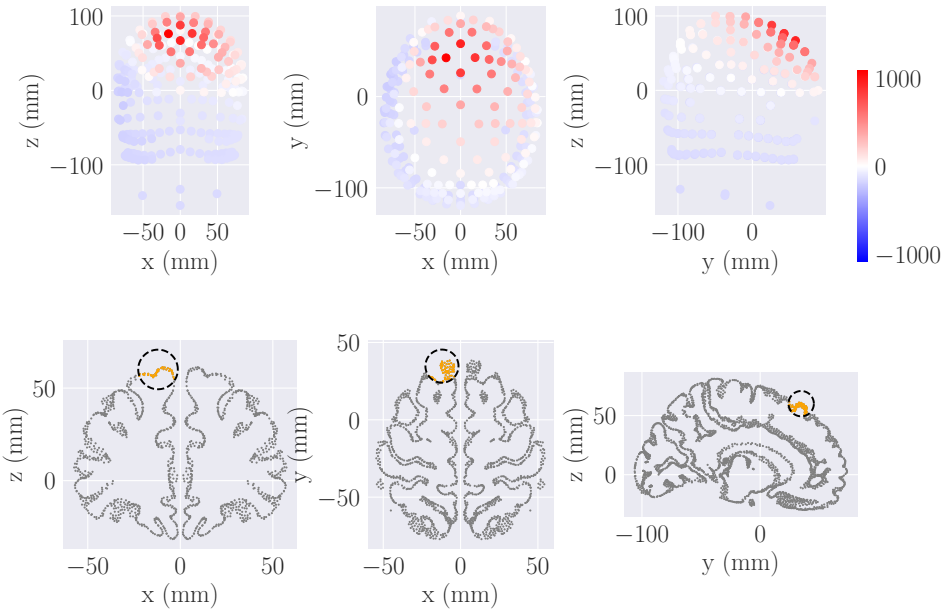

Figure 7.5: EEG for a sample containing a spherical population of current dipole sources with a random center within the celebral cortex. The EEG measure is seen from both sides (x-, z-plane and y-, z-plane) and above (the x-, y-plane). EEG electrode locations are presented as filled circels, where the color of the fill represents the amplitude of the measured signal for the given electrode.

Figure 7.6: The validation accuracy for the simple Feed Forward Neural Network, predicting both center and radius for 50 000 samples, for 5000 epochs, with a learning rate equal to 0.001.

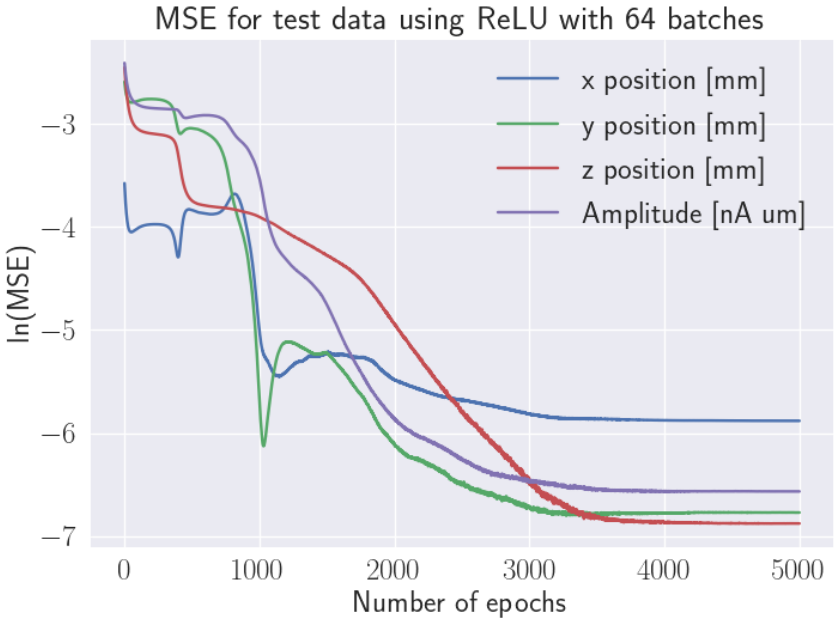

Figure 7.7: The validation accuracy for the simple Feed Forward Neural Network, predicting both center and radius for 10 000 samples, for 10000 epochs, with a learning rate equal to 0.001.

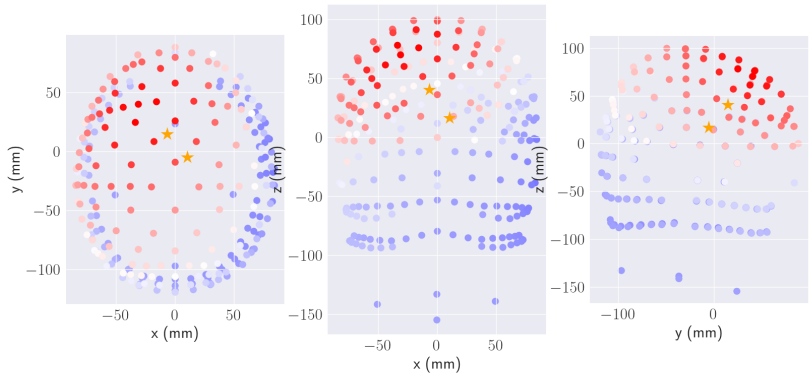

Figure 7.8: EEG for a sample containing two current dipole sources at random positions within the celebral cortex. The EEG measure is seen from both sides (x-, z-plane and y-, z-plane) and above (the x-, y-plane). EEG electrode locations are presented as filled circels, where the color of the fill represents the amplitude of the measured signal for the given electrode. The positions of the current dipole moments are marked with yellow stars.

Figure 7.9: The validation accuracy for the simple Feed Forward Neural Network, predicting two current dipole sources.