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

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

Neuroscience is a multidisciplinary field focused on understanding the complexities of the human brain and nervous system. At its core, neuronal communication forms the foundation for brain function, where billions of neurons interact through electrical signals called action potentials. Electroencephalography (EEG) plays a pivotal role in this area by recording and analyzing these electrical potentials in the brain. EEG serves as a non-invasive tool to detect abnormal brain activity and identify neurological disorders such as epilepsy. By exploring electrical brain activity, neuroscience endeavors to advance our comprehension of the human brain and improve diagnostic and therapeutic approaches for various neurological conditions.

In this introductory chapter, we will delve into the foundational aspects of neuronal communication and the underlying principles of EEG recordings. By familiarizing ourselves with these fundamental concepts, we can better appreciate the utility and potential applications of EEG in the dynamic field of neuroscience. Section 1, 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**], delves into the nature of neurons and their intricate communication networks.

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 a result of the selective permeability of the cell membrane to different ions, particularly sodium (Na+), calcium (Ca2+), and chloride (Cl-). At rest, the neuron maintains a relatively higher concentration of sodium ions outside the cell and a higher concentration of potassium ions inside the cell. This difference in ion con-

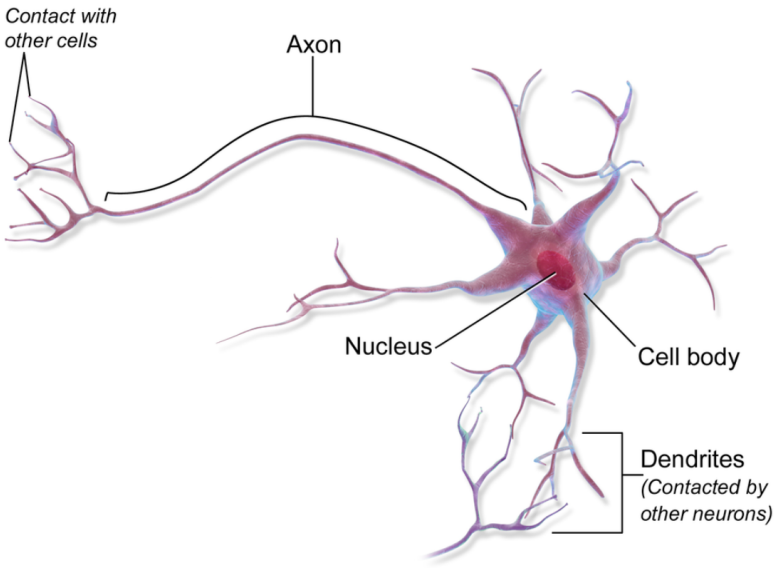

Figure 1.1: An illustation of a single neuron with dendrites, soma (cell body) and axon. The figure is taken from ...

centrations, along with the presence of ion channels that regulate the flow of ions in and out of the cell, contributes to the resting membrane potential. Typically, the membrane potential of a neuron 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 (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 reaches a specific threshold, an action potential is initiated. 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 input [**gerstner2014neuronal**].

In Figure 1.1, we have provided a basic illustration of a single neuron with dendrites, soma (cell body), and axon. [Add proper citation or indicate if it is a self-generated illustration].

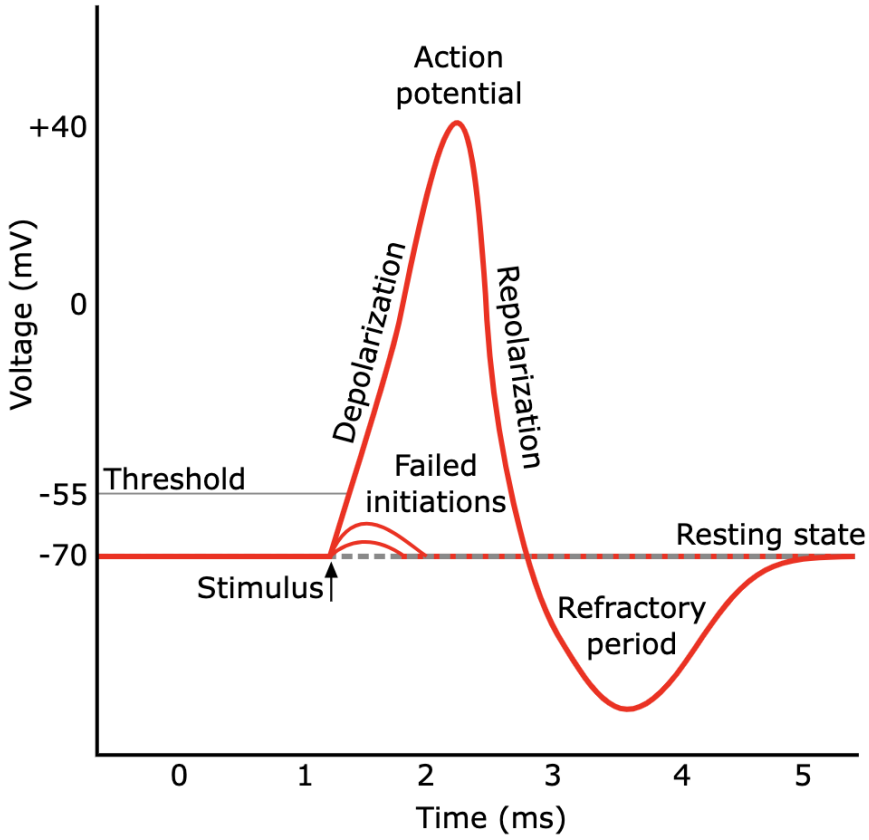

Figure 1.2: An illustation of a single neuron with dendrites, soma (cell body) and axon. The figure is taken from ...

1.1.1 Spike Trains and Action Potentials

When the electrical signals transmitted towards the soma reach the so-called threshold value, usually around -55 mV, the neuron fires. This initiation of an action potential can be seen as a spike in electrical recordings, with an amplitude of about 100 mV and a duration of 1-2 ms [**gerstner2014neuronal**]. Figure 1.2 we have provided an illustration of the typical action potential.

The action potential is characterised by a swift and steep rise in the electrical potential, resulting in a rapid upward, positive spike, followed by a quick decline in the potential vack to its resting state. This form of the action potentials remains relaticely constant throughout the propagation along the axon. When collecting information out of these spikes, it is therefor not the shape of the spikes that is studied. Instead, the information lies in the number and timing of chans of action potentials emmitted by the neuron, also reffered to as spike trais.

Spike trains observed during epileptic seizures exhibit distinguishable characteristics compared to "normal" spike trains in regular neural activity. Epileptic seizures, particularly those associated with spike-and-wave patterns in conditions like absence epilepsy, are characterized by regular, symmetrical, and generalized EEG patterns known as spike-and-wave discharges. These discharges result from bilateral synchronous firing of neurons involving the neocortex and thalamus within the thalamocortical network [Wikipedia]. The spike-and-wave discharges manifest as a repetitive and rhythmic pattern, typically around 2.5 Hz or higher, setting them apart from the more irregular and unpredictable firing of action potentials seen in "normal" spike trains [Neural Dynamics]. During epileptic events, the initiation of these discharges involves complex mechanisms, including the interplay of voltage-gated sodium and calcium channels and the role of inhibitory postsynaptic potentials [Wikipedia]. This stark contrast in the rhythmicity and temporal dynamics of epileptic spike trains highlights the distinct nature of epileptic activity when compared to the more variable and less periodic patterns observed in "normal" neuronal firing [Neural Dynamics].

1.1.2 Anatomy of the Cortex

Neurons in the brain are part of a vast network, interwoven with billions of other neurons and glial cells, creating the complex brain tissue. The brain is divided into various regions, and one essential area is the cortex, a thin but expansive sheet of neurons that folds over other brain structures. Different cortical areas have specific roles; some are specialized in processing sensory information, while others handle working memory or control motor functions [**gerstner2014neuronal**].

The human cerebral cortex consists of up to six layers of neurons. The oldest part of the cortex, known as the archipallium, has a more straightforward structure with three distinct neuronal layers. Within the archipallium, the hippocampus plays a major role in learning and memory functions. It is a crucial cortical structure implicated in the development of some common epilepsy syndromes [**bromfield2006introduction**].

Neurons communicate through synapses, where one neuron sends information (presynaptic cell), and another receives it (postsynaptic cell). In the animal brain, a single presynaptic neuron can connect to over 10,000 postsynaptic neurons. While many axonal branches end close to the neuron itself, some axons extend several centimeters to reach neurons in other brain regions [**gerstner2014neuronal**].

Within the cortex, there are two primary classes of neurons. Pyramidal neurons send information to distant areas of the brain, and they play a crucial role in long-distance communication. On the other hand, basket cells are considered local-circuit neurons, exerting their influence on nearby neurons. Most principal neurons form excitatory synapses, meaning they

stimulate post-synaptic neurons, while most basket neurons form inhibitory synapses, meaning they suppress the activity of principal cells or other inhibitory neurons [**bromfield2006introduction**].

Chapter 2

Electroencephalograpy

Electroencephalography (EEG) is a recording of the electrical activity of the cerebral cortex, representing a vital tool that has significantly contributed to our understanding of neuron interactions and the brain's organizational complexity. As one of the most widely used non-invasive techniques in neuroscience and clinical practice, EEG has played a pivotal role in studying brain activity during various cognitive processes, as well as in diagnosing diseases and estimating functional connectivity.

The roots of EEG trace back to the groundbreaking work of Hans Berger, who recorded the first human brainwave in 1924, marking the beginning of a new era in neuroscience research [**wiki:electroencephalography**]. Since then, EEG has become an indispensable method, providing valuable insights into brain dynamics and functioning. EEG is a valuable tool that can be used to detect abnormalities in specific areas of the brain, aiding in the diagnosis of various brain disorders, including epilepsy, Alzheimer's disease, and brain tumors. By identifying distinct patterns of brain activity associated with these conditions, EEG has become an essential tool for early detection, differential diagnosis, and treatment planning.

In this chapter, our primary objectives are to explore the physiological basis of the EEG technique, shed light on the concept of the inverse problem in EEG, and introduce the use of head models to simulate realistic EEG measurements. Understanding the foundations of EEG and its methodologies will lay the groundwork when we further in this thesis will investigate the possibilities of using simulated EEG measurements to train a neural network for the purpose of localizing the sources generating these signals.

2.1 The Physiological basis of the EEG

Electroencephalography (EEG) is a technique that utilizes small metal disks known as electrodes, placed on the scalp, to detect the electrical charges resulting from the activity of brain cells. The EEG recording electrodes are

typically connected to individual wires, which in turn are linked to channel connectors leading to a differential amplifier bank. An illustration of the typical EEG measurement setup is depicted in Figure 2.1. By measuring the electrical potentials of cortical neuronal dendrites near the brain's surface, EEG provides valuable insights into brain function.

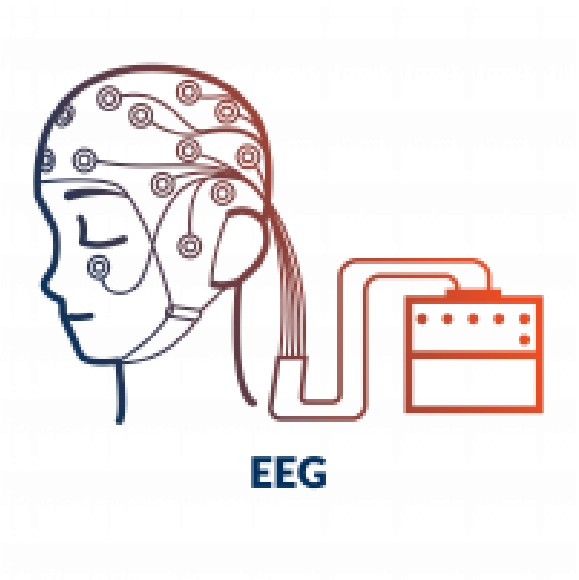

Figure 2.1: Illustration of the EEG method.

When a single pyramidal cell is stimulated and reaches its threshold, it generates an action potential. During this process, the synapse receives an excitatory signal, leading to a post-synaptic potential where positively charged ions enter the cell. As a result, a relatively negative charge is induced in the nearby extracellular space, which refers to the fluid-filled space surrounding the neuron. As the action potential travels down the dendrite, it eventually exits the cell membrane at locations further away from the synapse, and these locations are referred to as the "source." Consequently, an outward flow of positive charge prevails, leading to a relatively positive charge in the extracellular space. This spatial configuration creates an external dipole, with a relatively negative charge at the distant part of the dendrite and a positive charge closer to the cell body [**bromfield2006introduction**].

Since the electrical potential generated by an individual neuron is far too small to be picked up by the recording electrodes, the EEG measurements primarily reflect the summation of synchronous activity from thousands of pyramidal neurons with similar spatial orientation. Neurons with different geometric alignments cannot be measured as their ions do not align in a way that creates detectable waves. Due to the voltage field gradients decreasing with the square of the distance, detecting activity from deep sources

in the brain is more challenging compared to currents closer to the skull [**bromfield2006introduction**].

The EEG is typically described in terms of rhythmic activity and transients, which are divided into frequency bands. Frequency bands are often extracted using spectral methods, and most of the cerebral signals observed in the scalp EEG fall within the range of 1–20 Hz. Abnormal activity can broadly be classified into epileptiform and non-epileptiform activity. Epileptiform activity is characteristic of people with epilepsy and includes spikes, sharp waves, and spike-wave complexes. In this context, spikes refer to hypersynchronized bursts from a sufficient number of neurons, arising from high-frequency bursts of action potentials. Generalized epileptiform discharges often exhibit an anterior maximum, seen synchronously throughout the entire brain, strongly suggestive of a generalized epilepsy [**bromfield2006introduction**].

Detecting and localizing abnormal electrical patterns in EEG represents an important research pursuit. One of the fundamental aspects in this field is the *EEG inverse problem*, which aims to ascertain the spatial distribution of brain activity using potential measures acquired from scalp EEG recordings. In the upcoming section, we will explore the concept of the EEG inverse problem in greater detail and examine its implications for source localization.

2.2 The Inverse Problem and Source Localization

In the field of neuroscience, the inverse problem involves deducing the underlying parameters responsible for a set of measured EEG data. In contrast to the *forward problem*, where known parameters are used to predict the resulting EEG potential, the inverse problem lacks a unique solution. This implies that different configurations of neural sources can produce the same EEG activity distribution on the scalp [**hecker2021convdip**]. Is it then possible to reach a loss equal to 0?

The forward problem entails mathematically modeling the relationship between neural current sources in the brain and the resulting EEG measurements on the scalp. This can be described as:

$$\Phi(t) = L \cdot J(t) \tag{2.1}$$

Here, Φ represents the vector of measured EEG signals at time t, $J(t)$ is the vector of unknown neural current sources at time t, and L is the lead field matrix that connects scalp electrode recordings with neural sources. While the full details of the lead field matrix will be explored later, for now, envision it as a geometric arrangement linking the sensitivity of EEG measurements from diverse scalp locations to potential neural current sources within the brain.

Turning our attention to the inverse problem, its essence lies in estimating neural current sources in the brain using measured EEG data—essentially, the reverse of the forward problem. This relationship can be formulated as:

$$J(t) = L^{-1} \cdot \Phi(t) \tag{2.2}$$

where, L^{-1} denotes the inverse of the lead field matrix.

However, unlike the forward problem, the inverse problem lacks a unique solution due to its ill-posed nature. As a result, localizing the precise neural sources generating EEG signals becomes a demanding and statistically-driven endeavor. To address the complexities arising from numerous unknowns, techniques such as machine learning and neural networks are employed. Neural networks are designed to narrow down potential solutions, facilitating a more robust and meaningful estimation of the neural sources underlying the measured EEG data.

Before embarking on the utilization of neural networks to solve inverse problems, a substantial amount of appropriate EEG data is essential. This data can be obtained through simulated EEG data generated by *forward modeling*. Forward modeling deals with solving the forward problem, 2.1 and describes how the electrical activity in various regions of the human cortex gives rise to EEG signals recorded at the scalp electrodes. This process involves the utialization of *head models* that account for the conductivity of different tissues and the geometry of the head. These head models guide the simulation process, aiding in approximating the real-world EEG recordings.

2.3 Head Models

To accurately simulate EEG data and facilitate source localization, the utilization of head models that precisely represent the conductivity distribution within the human head is paramount. Head models serve as computational representations of the anatomical structure of the head, encompassing the brain, skull, cerebrospinal fluid, and scalp. They play a pivotal role in the simulation of how electrical signals originating from current dipoles propagate through diverse tissue compartments, influencing the recorded values at EEG electrodes.

EEG signals are significantly influenced by the biophysical intricacies of the head. Notably, the cerebrospinal fluid exhibits a conductivity of approximately 1.7 S/m, while the skull and scalp possess conductivities of about 0.01 S/m and 0.5 S/m, respectively. These conductivity disparities underscore the necessity for comprehensive head models that consider such variations. Beyond conductivity, these models also account for the influence of tissue arrangement on EEG signals, such as whether a neuronal population resides within a *sulcus* or a *gyrus* [**naess2021biophysically**]. By employing such a biophysically detailed head model, one can more accurately simulate

the impact of various tissues on the distribution of extracellular potential. As a result, this model offers a heightened level of precision in representing EEG signals, leading to improved accuracy in EEG source localization solutions.

2.3.1 The New York Head

The New York Head (NYH) model, developed by the Biomedical Engineering Department in New York, exemplifies a highly detailed computer model tailored for simulating brain electrical activity, with an emphasis on EEG source localization. Grounded in high-resolution anatomical MRI data from 152 adult heads, this model allows the segmentation of six distinct tissue types within the head: scalp, skull, cerebrospinal fluid, gray matter, white matter, and air cavities. Its high level of detail and accuracy makes it an excellent tool for simulating and comprehending brain activity in a realistic manner.

By presenting a three-dimensional representation of the head and brain, coupled with precise information about tissue geometry and electrical properties, the NYH model proves instrumental in investigating brain functions, particularly within the context of EEG measurements and source localization.

For EEG simulations, the NYH model is solved for 231 specific positions representing recording electrodes on the scalp. To predict the EEG signals recorded at different scalp locations, the model utilizes a mathematical representation called the *lead field matrix*. This matrix captures the relationship between the electrical activity in the brain and the electrical potentials recorded on the scalp.

The lead field matrix is essential for linking brain current density to the EEG signals recorded on the scalp. It relies on the reciprocity theorem, which connects brain current caused by an injected current between stimulating electrodes to the potentials picked up by recording electrodes. Specifically, for a fixed pair of stimulating electrodes, the lead field vectors are calculated throughout the head to determine the orientation of the dipole source that generates the largest potential difference between the electrodes. Represented by the symbol $\boldsymbol{L}$, the lead field matrix then establishes the relationship between the brain's current dipole moment and the resulting EEG signals. Mathematically, the lead field matrix $\boldsymbol{L}$ is given by:

$$L = \frac{E}{I}, \tag{2.3}$$

where I denotew the injected current at the electrode locations, and E corresponds to the resulting electric field in the brain [**naess2021biophysically**]. This equation provides the precise link between the current dipole moment p in the brain and the recorded EEG signals Φ:

$$\Phi = L \cdot p, \tag{2.4}$$

Is p here the same as J(t) in earlier equation?

In practical terms, when an injected current of 1 mA flows through the brain, it generates an electric potential E measured in V/m. Thus, a current dipole moment $\mathbf{p}$ in the unit of mAm results in EEG signals measured in the unit of V.

For further comprehensive details about the New York Head model, we refer readers to the article: https://www.parralab.org/nyhead/HauHuaPar-embc-2015.pdf.

2.4 The Current Dipole Approximation

By accurately simulating EEG data, non-linear optimization algorithms such as machine learning algortihms and neural networks can be utialized for solving the EEG inverse problem. However, the simulation of intricate neuronal dynamics is a computationally expensive and complex task. An approximation that address this challange and simplifies the simulation phase is the *current dipole approximation*. This approximation is rooted in the observation that the neuron's contribution to the extracellular potential V_e becomes increasingly more dipole-like with an increasing distance.

Find other sources than wikipedia The key insight behind the current dipole approximation lies in the *multipole expansion*, a technique that comes to our aid when the recording point is situated at a significant distance from the source distribution. While electrical charges, in the context of neural tissue, can lead to the creation of current multipoles, the multipole expansion theorem provides a way to express the extracellular potential $V_e(R)$ in terms of various contributions, where it in the case of EEG signaling becomes apparant that $V_e(R)$ can be approximated by a single dipole.

Neuronal multipoles depend on the spatial arrangement and symmetry of the charge distribution and result from the interplay of current sources and sinks [**wiki:multipoles**]. The expression for the extracellular potential associated with different multipole orders may take complicated forms, and be hard to interpreted. However, when the distance R from the center of the volume to the recording point surpasses the distance from the volume center to the outermost source, the applicability of multipole expansion becomes evident [**jackson1999classical**]. Such expansions are namly often beneficial as usually only the first few terms are needed in order to provide an accurate approximation of the original funtion, as we will se now. This representation of the extracellular potential $\phi(R)$ takes the form:

$$V(R) = \frac{C_{\text{monopole}}}{R} + \frac{C_{\text{dipole}}}{R^2} + \frac{C_{\text{quadrupole}}}{R^3} + \frac{C_{\text{octopole}}}{R^4} + ... \tag{2.5}$$

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. Interestingly, the contributions beyond the dipole term decay more rapidly with distance R. This means that in scenarios where we are considerably distant from the source distribution the higher-order terms become negligible, leaving us primarily with the dipole contribution.

This brings us to another interesting observation: The monopole contribution vanishes, stemming from the fact that the net sum of currents over a neuronal membrane is invariably zero. As a consequence, the monopole term dissipates, leaving us with an approximation of the extracellular potential, V_e, that relies solely on the dipole contribution:

$$V_e(\mathbf{r}) \approx \frac{C_{\text{dipole}}}{R^2} = \frac{1}{4\pi\sigma}\frac{|\mathbf{p}|cos\theta}{|\mathbf{r}-\mathbf{r}_p|^2}. \tag{2.6}$$

where we have substituded for C_{dipole} in terms of other properties. Here $\mathbf{p}$ symbolizes the current dipole moment within a medium of conductivity σ. The distance between the current dipole moment at $\mathbf{r}_p$ and the electrode location $\mathbf{r}$ is denoted as $R = |\mathbf{R}| = |\mathbf{r} - \mathbf{r}_p|$. Additionally, θ signifies the angle between $\mathbf{p}$ and $\mathbf{R}$. This equation is recognized as the dipole approximation and stands as a reliable method for calculating the extracellular potential, particularly when the distance R significantly surpasses the dipole length $d = |\mathbf{d}|$. This condition is frequently satisfied in EEG studies [**naess2021biophysically**].

Consequently, we have connected the concept of multipole expansion with the validity of the dipole approximation. Through multipole expansion, we can comprehend the intricate extracellular potential in terms of various contributions, and by carefully considering their behavior, we arrive at the precise dipole approximation.

In Figure 2.2, 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. The Figure has been collected from work done by Torbjørn Næss and Gaute Einevoll. This simulation aligns with the dipole approximation, as it vizualises 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. WTFWTF

2.5 Solving the EEG inverse problem

Can LFPy fit here or should it be in next chapter?

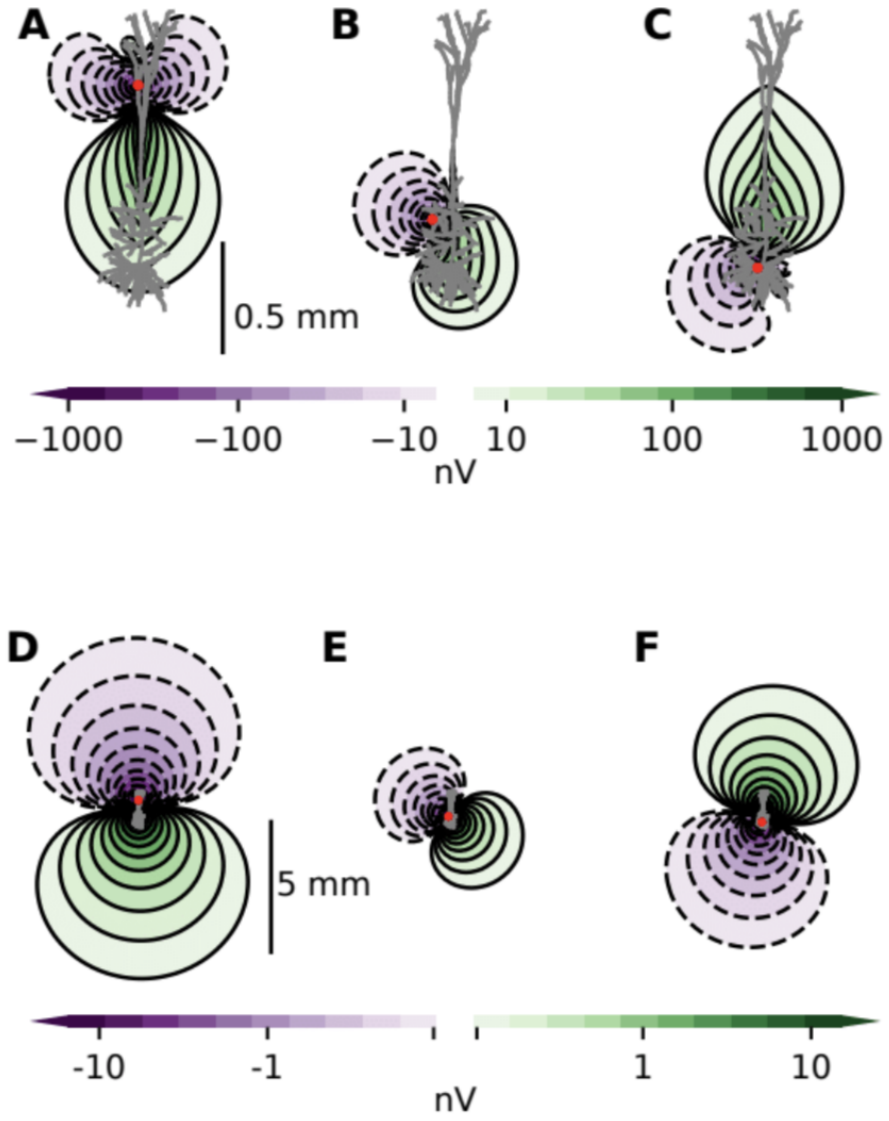

Figure 2.2: Simulation of extracellulat potential showing distinct dipole pattern. The figure has been provided from my supervisiors Torbjørn Ness and Gaute Einevoll.

In this chapter, we have explored the foundational concepts that underpin the resolution of the inverse problem in EEG source localization. With this groundwork in place, we now turn our attention to the subsequent chapters, where we transition from theory to application.

In the next chapter, we delve into practical implementation by simulating EEG data. Utilizing the New York Head and current dipole approximation, we create synthetic EEG data that closely resembles real-world scenarios. This simulated data, generated using the principles discussed in this chapter, serves as a crucial foundation for the subsequent chapters. It brings us a step closer to addressing the intricate EEG source localization challenge.

Chapter 4 propels us into the realm of machine learning and neural networks. Building upon the simulated EEG data, we construct a sophisticated neural network designed to address the inverse problem efficiently. By harnessing computational techniques, our aim is to bridge the gap between theoretical understanding and real-world applications.

Chapter 3

Creating EEG Data

In preparation for the application of neural networks to address inverse problems, the acquisition of a substantial and appropriate EEG dataset is essential. This chapter focuses on utilizing the New York Head model (NYHM) in conjunction with the current dipole approximation to construct biophysically realistic EEG data.

3.1 Simulation of EEG Signals from single dipoles

include range of x, y, z values !! and maybe eeg ? The New York Head model (NYHM) is integrated into the Python module LFPy. Within LFPy, we use the *NYHeadModel* class to calculate EEG signals originating from a current dipole moment $\mathbf{P}$. The current dipole moments of the LFPy kit are expressed in terms of nAμm, while the EEG signals derived from the NYHM are recomputed into units of mV. For more information about the LFPy package, we refer the reader to `https://lfpy.readthedocs.io/en/latest/readme.html#summary`.

The cortical matrix of the NYHM comprises 74,382 discrete points, each corresponding to a possible location for localizing dipole sources. In the context of simulating EEG measurements, the procedure commences with the random selection of positions from these points to serve as the locations for placing dipoles. Each simulated EEG sample entails a solitary dipole positioned at one of the randomly chosen locations. As the primary objective is to address the inverse problem, maintain uniform amplitudes for the dipole signals, as their variation is not of primary concern. By setting these amplitudes to 10^7 nA μm, the resulting EEG measurements span a range of approximately -1 to 1 μV.

To ensure that the dipole orientations are predominantly aligned with the depth direction of the cortex, we assign dipole strengths solely in the z-direction. Moreover, a rotation procedure is employed for each dipole moment, orienting it perpendicular to the cerebral cortex. Occasionally, this

orientation results in a dipole moment pointing outward, toward an EEG electrode. Alternatively, due to the convoluted structure of the cortex, the dipole moment may be directed back into the cortex, eventually aligning with an EEG electrode after traversing a greater distance. This phenomenon arises from the complex folding patterns of the human cortex, where the EEG signal contribution of a dipole moment depends on its position within a sulcus or gyrus [**naess2021biophysically**].

The New York Head model (NYHM) generates EEG signals as time series data, reflecting the structure of real-world measurements. As a result, the inherent format of EEG data deviates from a one-dimensional representation, instead adopting a matrix configuration of 231x1601 dimensions. In this arrangement, 231 values represent measurements from scalp recording electrodes, with 1601 time steps marking the temporal progression. However, as mentioned in the preceding chapter, EEG analysis often centers on specific frequency components within each temporal instance of measurement. This practice effectively reduces the multidimensional EEG data and eliminates less relevant recordings.

In the context of our analysis, which aims to pinpoint sources of neural activity, the transition to one-dimensional data proves to be benefitial. This transition contributes to problem simplification and computational efficiency. Importantly, our approach diverges from the conventional practice of extracting diverse frequency spectra to identify anomalous activity. Instead, we leverage the initial time step of the EEG recording to encapsulate epileptiform behavior of interest. This method works well because there are no confusing signals from brain activity or unclear background noise in the simulated data. This makes the analysis very reliable and solid. Our focus is directed towards the signal at $t = 1$, corresponding to the first time step of the recording. As a result, our methodology yields a one-dimensional EEG signal, effectively encapsulating insights into the spatial distribution of EEG patterns and the interrelation between these patterns and the specific locations of dipole sources within the cerebral cortex.

Should we show a single sample will look like?

3.2 The Effect of dipole location and orientation

According to Naess et al. (2021) [**naess2021biophysically**], EEG signals are not particularly sensitive to minor shifts in the precise location of neural current dipoles. This insensitivity can be explained by the fact that relative to the dimensions of individual neurons and the thickness of the human cortex, EEG electrodes are located far away from cortical neural sources. Given the substantial spatial smearing and the considerable distance between EEG electrodes and cortical sources, millimeter-scale shifts in the positions of neural current dipoles tend to have a limited impact on the EEG signal.

This is further illustrated in Figure 3.1, where we present three EEG signals corresponding to dipoles situated at neighboring points within the NYHM cortical matrix.

The plotted EEG signals of the blue and green dipoles in Figure 3.1 exhibit a considerable overlap. This overlapping pattern is supported by a high correlation coefficient of 0.966, indicating a robust positive relationship between these measurements. In simpler terms, as one variable increases, the other variable increases proportionally - an observation evident from the figure. In contrast, the EEG signals of the blue and red dipoles exhibit less resemblance. The correlation coefficient between these signals is 0.695, denoting a somewhat smaller linear relationship. Importantly, the observed differences can be attributed to the shifts in the normal vector of the red dipole. This shift is also evident for the normal vectors of the blue and green dipole, but is more evident considering the red dipole. When choosing neighboring dipoles in the terms of distance within the NYHM, shifts in the normal vectors also arises due to the rotation procedure used during data generation to ensure dipole orientations are perpendicular to the cerebral cortex. Due to the complex folding of the cortex, shifts in the dipole's location will often lead to corresponding shifts in normal vectors as well. As a result, distinct differences emerge in the corresponding EEG signal.

To further illustrate this, consider the effect of dipole orientation on EEG outcomes. Figure 3.2, borrowed from work done by Tornjørn Ness and Gaute Einevoll, represent 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, and exhibit distinct EEG patterns. In general, the contribution of an individual current dipole to the EEG signal is maximized the dipole is perpendicularly situated within a gyrus, as depicted in Figure 3.2B. Contrastingly, when a dipole is placed in a sulcus with a perpendicular orientation, a significant EEG contribution may still be observed, however unlike the dipole in the gyrus, it exhibits a more dipolar pattern, as shown in Figure 3.2C.

Further exploration of dipole orientation's impact is presented in Figure 3.3. This figure is borrowed from work done by Torbjørn Ness and Gaute Einevoll. Here, we observe EEG signals from identical dipoles positioned in various folding patterns of the cortical surface. These patterns reveal that the orientation of the current dipole moment significantly influences the EEG outcome. Firstly, Figure 3.3A and 3.3C provide an expanded illustration of the aforementioned scenarios, incorporating additional dipole moments located in a gyrus and a sulcus, respectively. In Figure 3.3B, 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 3.3A and 3.3B, as depicted in Figure 3.3D. Lastly, Figure 3.3E

Figure 3.1: EEG signals plotted against electrode number for three neighbouring dipoles with normal vectors (-0.60, 0.11, 0.79), (-0.75, -0.15, 0.64), (-0.21, 0.32, 0.93) and positions (-12.04, 34.71, 60.07), (-12.37, 33.92, 59.75), (-10.24, 34.61, 60.75). The correlation coefficient between the blue and green dipole is 0.97, while it is 0.70 for the blue and red dipole.

demonstrates the minimal EEG contribution observed when the dipoles are divided between two opposing sulci.

3.3 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

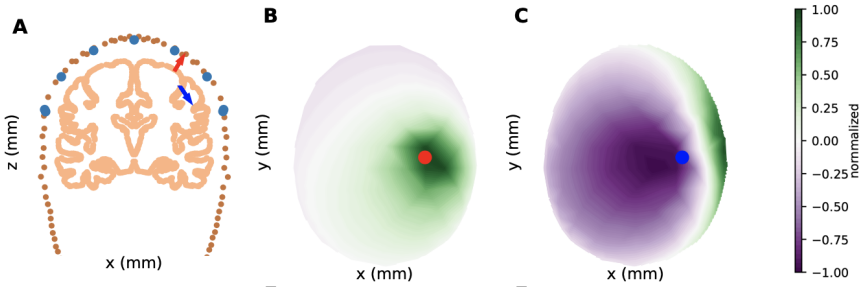

Figure 3.2: A: Two selected dipole locations in the New York head model: one in a gyrus (red) and one in a sulcus (blue). The head model is viewed from the side (x, z-plane). Close to the chosen cross-section plane, EEG electrode locations are marked in light blue. Available dipole locations near the 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 scaled, with a maximum value of 1.1 μV. C: Interpolated color plot of EEG signal from the sulcus dipole. The plotted EEG signal is scaled, with a maximum value of 0.7 μV. This Figure is borrowed from work done by Torbjørn Ness and Gaute Einevoll [**naess2021biophysically**].

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.

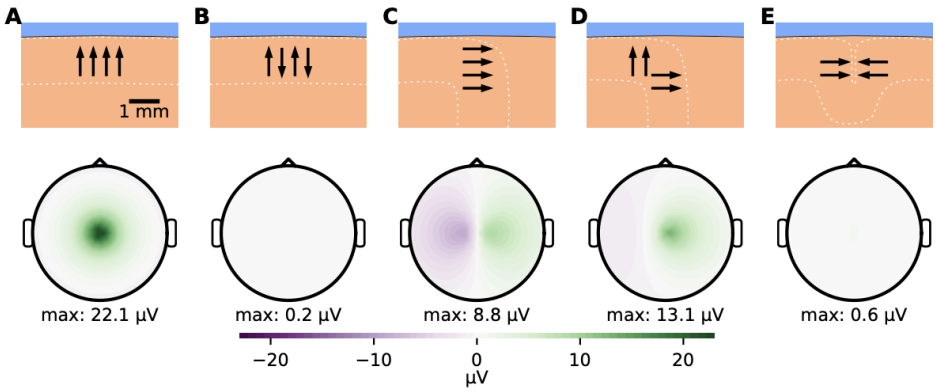

Figure 3.3: Different folding patterns of the cortical surface are represented by white dashed lines. EEG signals are calculated from four identical current dipoles with varying orientations. A: Dipoles aligned in the same direction within a gyrus. B: Dipoles pointing in opposite directions within a 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. Each panel features a dipole moment magnitude of 10 nAm, and the dipoles are positioned at the centers of the arrows in the top row. This Figure is borrowed from work done by Torbjørn Ness and Gaute Einevoll [**naess2021biophysically**].

3.3.1 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 dataset has a size of 70 000 x 231. In practice, the data consists of two separete files holding pairs of EEG data and corresponding target data, where x-, y- and z coordinates of different dipole sources is the answer keys.

Figure 3.4 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 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.

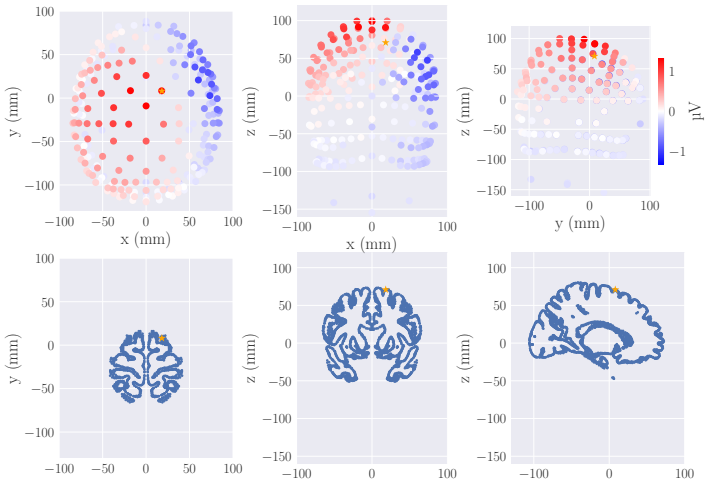

Figure 3.4: EEG for a sample containing one single current dipole source at a random position within the celebral cortex. As for all samples within the data set, 10 percent of normally distributed noise has been added to the original signal. The EEG measure is seen from both sides (x-, z-plane and y-, 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.

Chapter 4

DiLoc - A Neural Network Apporach for Source Localization

When having enough and suitable data for a problem to solve, one can start building tailorsuited neural networks to address the problem. We aim to build a neural network which aims to map measured EEG signals to localized equivalent current dipoles. In this chapter, we will provide a comprehensive overview of the feed forward neural network that we have build, reffering to it as *DiLoc*. We will discuss its architecture, parameters and training process. Moreover we will present an alternative approach using a convolutional neural network for the same purpose of source localization.

scaling

4.1 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 application could lie in the identification of sources generating abnormal electrical brain signals, as we will be performing. 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.

According to Mehta et. al. a definition of machine learning could be "...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. $f : \mathbf{x} \to y$ 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 $\mathcal{C}(\mathbf{y}, f(\mathbf{X});\boldsymbol{\theta})$.

4.1.1 Neural Networks

Include what we mean and how it works with weights and biases In order to solve the inverse problem we will be building a neural network. Neural networks are a distinct class of so-called *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 that communicate through mathematical functions across layers. The layers in neural networks contain an optional number of nodes, where each connection is represented by a weight variable. In the context of neural networks, these weight values can be understood as parameters representing the strenght of communication between nodes. Changing these weight values determines how strongly or weak a nodes's output influences another noded's input. During training, the neural network learns to adjust these weights so to capture the underlying patterns and relationships in the data. The network is able to do so by using past experiences known as training examples. These patterns are further updated by the usage of appropriate non-linear functions, known as *activation function*, and finally presented as the output [**nwankpa2018activation**]. A neural network consits of many such nodes 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 4.1 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, or node:

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

where a is the output of the node, and is the value of the nodes activation function f which has as input a weighted sum of signals $x_i, x_{i+1}, ..., x_n$ recieved by n other nodes, multiplied with the weights $w_i, w_{i+1}, ..., w_n$ and added with bieases $b_i, b_{i+1}, ..., b_n$. The exact expression of a varies depending on the type of non-linearity that exists in the activation function applied to the input of each node. 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 4.1, the

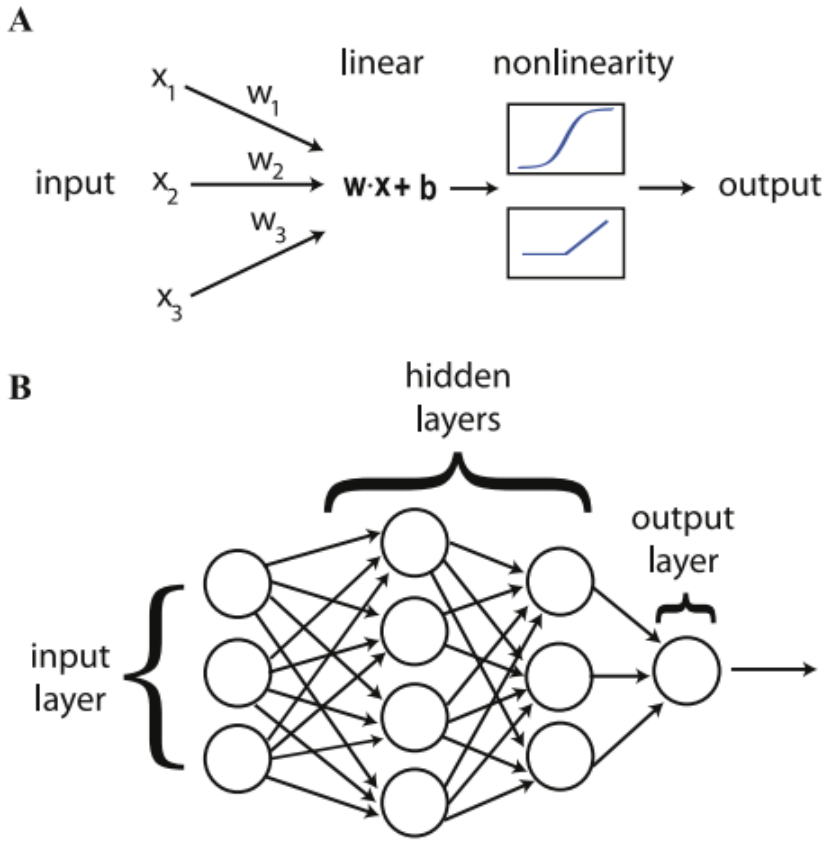

Figure 4.1: **(A)** The fundamental structure of neural networks comprises simplified nodes units that perform a linear operation to assign different weights to inputs, followed by a non-linear activation function. **(B)** These nodes units are organized into layers, where the output of one layer serves as the input to the subsequent layer, forming a hierarchical arrangement.

linear tranformation commonly takes the form of a dot product with a set of node-specific weights followed by re-centering with a node-specific bias. A more convenient notation for the linear transformation z^i then goes as follows:

$$z^i = \boldsymbol{w}^{(i)} \cdot \boldsymbol{x} + b^{(i)} = \mathbf{x}^T \cdot \mathbf{w}^{(i)}, \tag{4.2}$$

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.

$$a_i(\mathbf{x}) = f_i(z^{(i)}). \tag{4.3}$$

4.2 The creation of DiLoc

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 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.

4.2.1 Architecture

The determination of the optimal number of hidden layers and neurons was meticulously executed through an iterative trial-and-error procedure. Various network configurations, including small, medium, and large architectures, were systematically examined. Ultimately, we settled on the medium-sized network configuration. This selection, combined with considerations of additional network attributes, which will be elaborated upon later in this chapter, 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 4.2 visualizes the construction of the fully connected neural network.

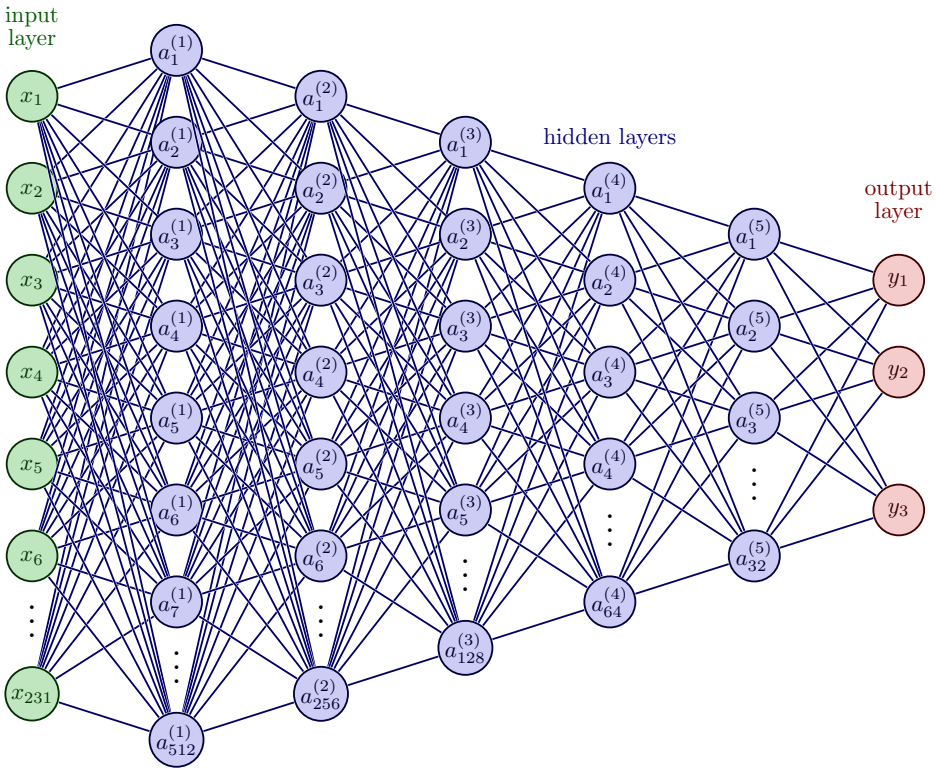

Figure 4.2: Architecture.

4.2.2 Activation Functions

Activation functions are fundamental components within the architecture of neural networks, serving to transform input signals into meaningful outputs. Essentially, these functions introduce nonlinearity into the network's computations, thereby enabling the network to capture and model complex, nonlinear relationships within data [**sharma2017activation**]. This property is essential because many real-world phenomena and data patterns exhibit inherent nonlinearity. In the context of solving the EEG inverse problem, where the EEG data contains intricate, nonlinear patterns, activation functions empower neural networks to effectively model and learn from such data structures.

Without activation functions, neural networks would essentially be linear models, capable only of representing linear relationships between inputs and outputs. While linear transformations occur within individual nodes through the weighted sum of inputs, the introduction of non-linear activation functions allows neural networks to capture complex relationships and patterns. These functions are applied at every artificial neuron in the hidden layers and in the output layer [**choose_activation_function**].

Drawing inspiration from the behavior of biological neurons, activation functions can be understood as decision-makers within the network, determining which information should be relayed to the next artificial neuron. This process is analogous to biophysics, where the axon of one cell takes the output signal from the preceding cell and converts it into a format suitable for input to the next cell [**citation_needed_for_figure**]. Some activation functions can also be directly associated with biological phenomena like action potentials and spikes within neurons. Similar to how real neurons respond to incoming electrical signals, activation functions decide whether a node in a neural network should be activated or not based on the strength of the input it receives. If the input exceeds a certain threshold, the artificial neuron "fires" or becomes activated; otherwise, it remains inactive [**analyticsvidhya_activationfunctions**].

Rectified Linear Unit

Within the input layer of the DiLoc network, nodes utilize the *Rectified Linear Units* (ReLU) activation function. ReLU closely resembles the behavior of biological neurons. Specifically, it echoes the concept of action potential: if a threshold is reached, the neuron fires, otherwise, the neuron remains inactive. For the ReLU function, this means positive input values remain unchanged, while negative input values are suppressed by setting them to zero.

Mathematically, the ReLU function is defined as:

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases} \tag{4.4}$$

The widespread adoption of ReLU in neural networks is attributed to its computational speed, performance, and generalization capabilities [**wandb_activation_functi**
In contrast to the Hyperbolic Tangent activation functions, to which we will return shortly, ReLU offers a more straightforward mathematical representation. As seen in Figure 4.3, the ReLU function maintains the input value when positive and outputs zero for negative inputs. This behavior promotes computational efficiency, as at any instance, only a subset of neurons activate.

By using ReLU in our first layer we introduces non-linearity into the model and enables it to capture complex relationships and patterns within the EEG data effectively.

Why ReLU in fist layer. Include the problem of dead neurons

Hyperbolic Tangent

For the hidden layers, we employed another actication function, known as the *Hyperbolic Tangent* (tanh). This decision was driven by its ability to

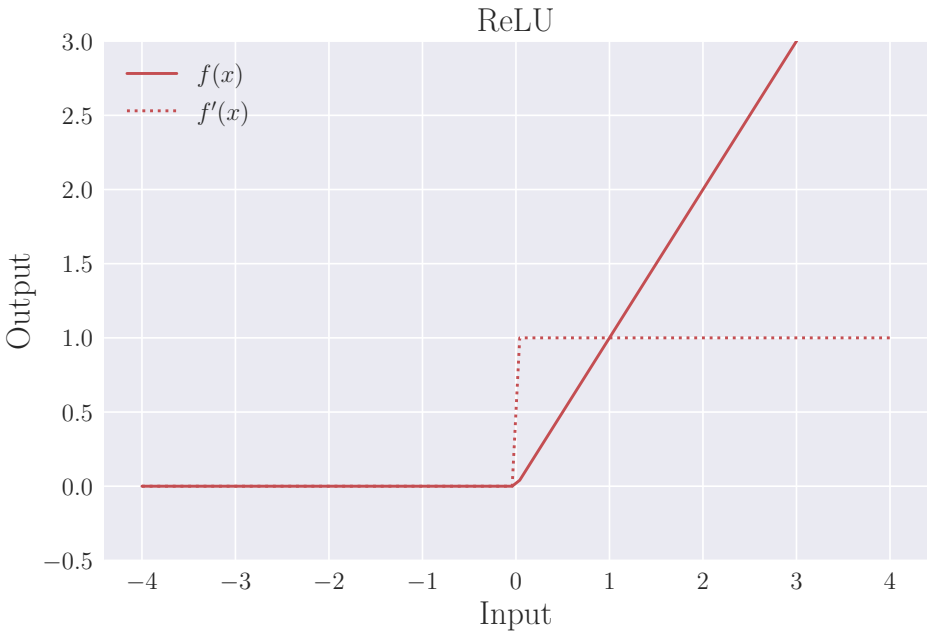

Figure 4.3: ReLU tangent activation function.

compress input values into a range between -1 and 1, which helps prevent activations from becoming extremely large and potentially unstable during training. By constraining the activations within this range, we aim for DiLoc to achieve a stable and effective learning process.

Tanh is continuous and differentiable at all points, that for reasons we will come back to is making it a suitable activation function for neural networks. Its mathematical representation is as follows:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{4.5}$$

As seen in Figure 4.4, the Tanh function smoothly squashes input values into the desired range of -1 to 1, providing a more gentle activation compared to ReLU. While ReLU is known for its computational speed and simplicity, Tanh's bounded output ensures that activations remain within a controlled range.

Linear Activation

Last Layer Linear Avtivation? DiLoc aims to provide direct and unconstrained predictions of the x-, y-, and z-positions for a desired dipole source. In our specific application, these target positions exhibit a range of values, with x ranging from -70 to 70 mm, y from -58 to 78 mm, and z from -69 to 59

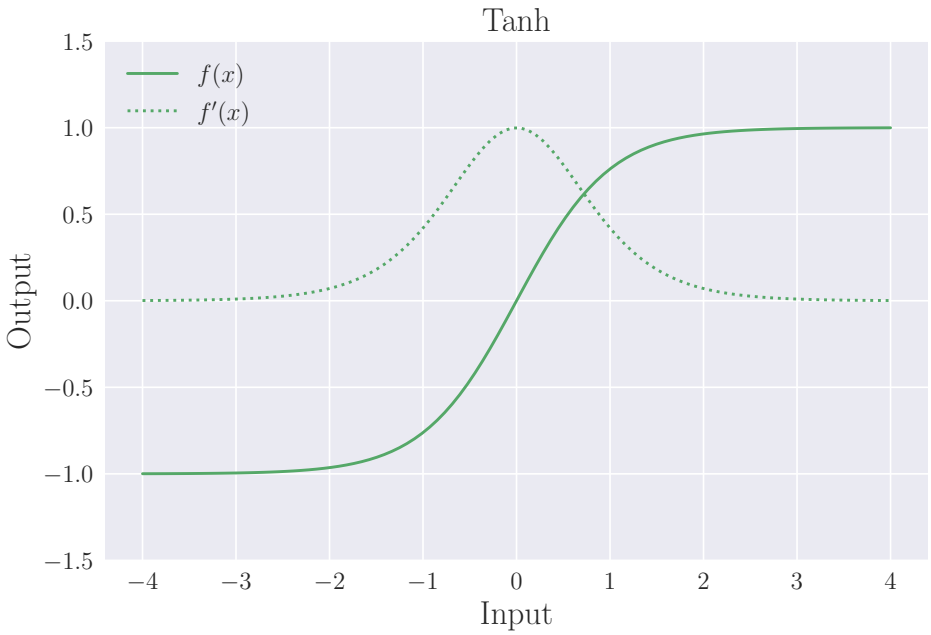

Figure 4.4: Hyperbolig tangent activation function.

mm. This wide range of potential values underscores the need for a regression approach that allows for unbounded and continuous predictions. When dealing with regression problems, linear activations, or rather, no activation function at all, are commonly utialized.

Therefore, in contrast to ReLU and tanh activations used in earlier layers, we adopted linear transformations without the use of an activation function for the nodes within the last layer of DiLoc. his decision aligns with the requirements of our specific task, ensuring that the output of the network is not constrained within a specific range.

Exploding and Vanishing gradient ...

4.2.3 Initialization

All inner parameters within a neural network are adjustable. Initialization of the parameters of weights and biases is an important process used within neural networks. The choice of these initial values can significantly impact how quickly the model converges during training and how well the model generalizes to unseen data.

There are several techniques for initialization in neural networks. In DiLoc, we have utilized the Xavier initialization, as this initialization commonly is paired with the tanh activation function. For every layer l, the weights $W^{[l]}$ and biases $b^{[l]}$ are sampled as follows:

$$W^{[l]} \sim \mathcal{N}(\mu = 0, \sigma^2 = \frac{1}{n^{[l-1]}}) \tag{4.6}$$

$$b^{[l]} = 0 \tag{4.7}$$

In this approach, the weights of layer l are drawn from a normal distribution with a mean (μ) of 0 and a variance (σ^2) of $\frac{1}{n^{[l-1]}}$, where $n^{[l-1]}$ signifies the number of inputs to the neuron. Additionally, biases are initialized to zero. This practice aligns with the idea that biases represent the initial influence of each neuron before any data is processed, making zero initialization a reasonable starting point.

The use of activation functions like tanh can introduce challenges related to vanishing and exploding gradient issues. Xavier initialization addresses these concerns by ensuring that the variance across layers is proportional to $\frac{1}{n^{[l-1]}}$, thereby contributing to the stability of the learning process. Essentially, Xavier initialization strikes a balance in parameter initialization, preserves variance, and mitigates the risk of gradients vanishing during training. This makes it a well-suited choice for DiLoc, where both stability and efficient training are paramount, particularly when dealing with intricate EEG data patterns.

Chapter 5

Training the DiLoc Neural Network

This chapter will explore the training process of the DiLoc neural network, outlining the decisions made during its training. Carefully choosing and adjusting elements of the training process ensures DiLoc's ability to make accurate predictions, establishing it as a valuable tool for solving the inverse EEG problem. Subsequent chapters will delve into the results and outcomes of this training, providing insights into the model's effectiveness. Many choise are based on observation from trial and errors

5.1 Training Methodology Overview

Having constructed the DiLoc neural network architecture, we now embark on the phase of training the model to effectively localize equivalent current dipoles from EEG signals. This chapter delves into the process of training DiLocc and addresses various crucial aspects that influence the performance of the network. Training a neural network requires careful consideration and tuning of several key factors to achieve optimal results. In the process of training DiLoc, we will explore the following essential elements:

Data Preparation:

Cost Function: We begin by discussing the significance of the cost function in guiding the network towards convergence. Understanding how to choose the appropriate cost function tailored to the problem is essential.

Back Propagation: As the backbone of neural network training, the backpropagation algorithm merits our attention for its role in propagating error gradients through the network, refining the model's weights and biases.

Optimization Algorithm: The choice of optimization algorithm plays an important role in training efficiency and convergence speed. We delve into the tequniqe of grasdient decent and its implications in the context of DiLoc. Here it would be natural to mention the batch size we have choosen.

Regularization Techniques: Overfitting is a common challenge in neural network training. We examine regularization techniques like dropout and weight decay to mitigate this issue.

Learning Rate Scheduling: Learning rates significantly influence training dynamics. We discuss methods for scheduling learning rates during training to ensure steady convergence.

5.2 Data Preparation

5.2.1 Data Segmentation

Standardization assumes that your observations fit a Gaussian distribution (bell curve) with a well behaved mean and standard deviation. You can still standardize your data if this expectation is not met, but you may not get reliable results.

The first step in prepering to *fit* a machine learning model, is to perform a data splkit, segregating the data set $\mathcal{D}$ into distinct sets for training, validation, and testing. This partitioning is done in order to make a model robust and compatible with multiple data sets. The size of each set commonly dependent on the size of the data set avaible, however a general guideline is that the majority of the data are allocated into the trainng set with the remainder going into the test set [**mehta2019high**].

In the case of DiLoc's input data, we deal with 70 000 samples of EEG signals. Out of these, 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 will play 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 have fine-tuned the network's parameters to achieve better generalization to unseen data. Upon completion of the training process, the test set comes into play. Comprising 20 000 samples, this set serves as the ultimate benchmark for evaluating the model's capacity to generalize and make accurate predictions on new instances of data. By adhering to this train-validation-test data partitioning,

we ensure a robust evaluation of DiLoc's performance and its capacity to effectively handle real-world scenarios with previously unseen data.

5.2.2 Data Scaling

Ask Vegard. Featurewise or whole data set? Having addressed the data segmentation, we proceed to the task of *data scaling*, which is a highly recommended procedure within machine learning. *Data standarizartion* is one out of two types of data scaling that involves centering the data around the mean μ and scaling it to achieve unit variance σ:

$$\mathcal{D}' = \frac{\mathcal{D} - \mu}{\sigma} \tag{5.1}$$

After standarization, the mean of the data set $\mathcal{D}'$ is 0 and the standard deviation is 1. By standarising the EEG inputs, we ensure that the distribution of the data becomes more uniform in all directions within the feature space, contributiong to more effective EEG signal analysis.

5.3 Cost Function

In the field of machine learning, *cost functions* play a crucial role in evaluating how well models make predictions. These mathematical functions evaluate how well models make predictions by measuring the discrepancy between predicted outcomes and actual target values. This evaluation results in a quantifiable metric known as *loss*. Higher loss values indicate poorer model performance, while lower values reflect more accurate predictions. When using the expression *fitting a model* one commonly refer to the prosess of finding optimal parameters $\boldsymbol{\theta}$ that minimize a chosen cost function. The process utilizes training data and iteratively updates the parameters to fine-tune the model's internal representations, making it more adept at accurate predictions.

Selecting the appropriate cost function is important when addressing machine learning challenges. In regression tasks like ours, the mean squared error (MSE) is a widely used cost function. The MSE is preferred due to its simplicity and continuous measure of model performance during training. It calculates the squared differences between the model's predictions and target values, then takes the mean across the entire dataset:

$$\text{MSE}(\boldsymbol{\theta}) = \frac{1}{N-1}\sum_{i=0}^{N}(y_i - \tilde{y}_i)^2, \tag{5.2}$$

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

Due to the squaring step in the MSE function, which penalizes larger errors more severely, the trained model is less likely to generate outlier predictions with substantial errors. However, in situations where the model produces a single, highly inaccurate prediction, the squaring effect can significantly amplify the error. Nevertheless, in many practical scenarios, the primary focus is not on these individual outliers, but rather on achieving a well-balanced model that delivers satisfactory performance across the majority of predictions [**builtin-ml-loss-functions**].

In our simulation-driven context, devised to accommodate a spectrum of diverse EEG variations, it is noteworthy that the presence of apparent outliers is not rooted in sampling errors, but rather random fluctuations and indicative of distinct scenarios. As delineated in statistical literature [**barnett1994outliers**], an outlier is characterized as an observation that seemingly deviates from the prevailing distribution of a dataset. This departure can be attributed to human or instrumental anomalies in the data acquisition process [**zhang2015outlier**]. Figure 5.2 vividly depicts the distribution of minimum and maximum EEG observations across individual samples within the dataset. While the majority of data points exhibit values within the range of -1.5 to 3 μV, a subset extends beyond these bounds. It is crucial to underscore that these data points do not emerge as stochastic errors stemming from the acquisition process. Instead, they intentionally embody distinctive EEG signals that enrich the dataset's representational fidelity within authentic real-world contexts.

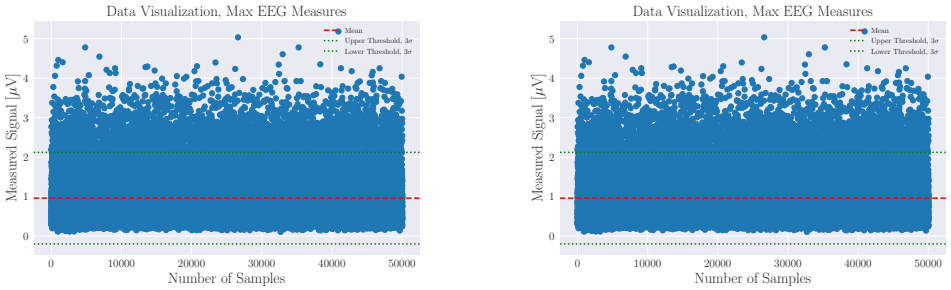

Figure 5.1: **Visualization of data set.** The panels visualize the maximum and minimum EEG measures for each sample within the data set.

5.4 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:

$$\delta_i^l = \frac{\partial C}{\partial(z_i^l)}, \tag{5.3}$$

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:

$$\delta_i^l = \frac{\partial C}{\partial(z_i^l)} = \frac{\partial C}{\partial(b_i^l)}\frac{\partial C}{\partial(z_i^l)} = \frac{\partial C}{\partial(b_i^l)} \tag{5.4}$$

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 5.3 in terms of the equations for layer $l+1$. This forms the basis of the third equation used in the backpropagation algorithm:

$$\begin{aligned}
\delta_i^l = \frac{\partial C}{\partial z_i^l} &= \sum_j \frac{\partial C}{\partial z_j^{l+1}}\frac{\partial z_j^{l+1}}{\partial z_i^l} \\
&= \sum_j \delta_j^{l+1}\frac{\partial z_j^{l+1}}{\partial z_i^l} \\
&= \sum_j \delta_j^{l+1} w_{ij}^{l+1} f'(z_i^l)
\end{aligned} \tag{5.5}$$

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

$$\frac{\partial C}{\partial w_{ij}^l} = \delta_i^l a_j^{l-1} \tag{5.6}$$

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 5.3. 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 5.5 to propagate the error backwards and calculate δ_j^l for all layers.

5. **Calculate gradient:** use equation 5.4 and 5.6 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:**
   $w_{jk}^l = w_{jk}^l - \eta \delta_j^l a_k^{l-1}$
   $b_j^l = b_j^l - \eta \delta_j^l$

Fix ending and cite This description makes clear the incredible utility and computational efficiency of the backpropagation algorithm. We can calculate all the derivatives using a single "forward" and "backward" pass of the neural network. This computational efficiency is crucial since we must calculate the gradient with respect to all parameters of the neural net at each step of *gradient descent*, an optimalization algorithm that we will be delving into in the next section.

5.5 Optimization Algorithm

Go through and cite with **[mehta2019high]** In order to minimize the cost function and find the optimal values for the model parameters, $\boldsymbol{\theta}$, an op-

timization 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.

Gradient Descent 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 the direction of the negative gradient of the function at a given point $\mathbf{w}$, $-\nabla F(\mathbf{a})$. This means that if

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \eta \nabla F(\mathbf{w}_n) \tag{5.7}$$

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:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \eta_n \nabla F(\mathbf{x}_n), n \geq 0, \tag{5.8}$$

where $\eta_n \geq 0$. This process the forms a strictly decreasing process:

$$F(\mathbf{x}_0) \geq F(\mathbf{x}_1) \geq F(\mathbf{x}_2) \geq ... \geq F(\mathbf{x}_n). \tag{5.9}$$

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, and is the optimalization algorithm we have choosen when training DiLoc [**bishop2006pattern**].

Stochastic Gradient Descent

The Stochastic Gradient Descent (SGD) method, as applied in the context of training DiLoc [**bishop2006pattern**], is a powerful optimization technique that diverges from traditional gradient descent by randomly selecting subsets of the data during each iteration, as opposed to considering the entire dataset. This approach is particularly beneficial when dealing with extensive datasets. The update step in SGD can be expressed as:

$$\mathbf{w}\tau + 1 = \mathbf{w}\tau - \eta \nabla F_n(\mathbf{w}_\tau) \tag{5.10}$$

Here, $\mathbf{w}\tau$ represents the weight vector at iteration τ, η is the learning rate, and $\nabla F_n(\mathbf{w}\tau)$ is the gradient of the cost function F_n computed on a mini-batch, which refers to a randomly selected subset of the complete dataset.

In essence, SGD mirrors regular gradient descent but restricts its focus to a single mini-batch at each iteration. The introduction of this stochastic element, achieved by sampling from subsets of the data, offers a valuable advantage: it allows the algorithm to explore and potentially escape from local minima, promoting the discovery of superior solutions.

Furthermore, the incorporation of *momentum* into the SGD algorithm enhances convergence speed. Momentum introduces a form of memory into the optimization process by accumulating information about previous movements in parameter space. Specifically, momentum is realized through the following equations:

$$\mathbf{v}\tau = \gamma \mathbf{v}\tau - 1 - \eta \nabla F_n(\mathbf{w}_\tau) \tag{5.11}$$

$$\mathbf{w}\tau = \mathbf{w}\tau - 1 + \mathbf{v}_\tau \tag{5.12}$$

In these equations, $\mathbf{v}\tau$ represents the momentum vector at iteration τ, γ is the momentum coefficient, η is the learning rate, and $\nabla F_n(\mathbf{w}\tau)$ signifies the gradient of the cost function F_n computed on the mini-batch.

During the training of DiLoc, we observed that a learning rate of 0.001 yielded the most favorable results. Moreover, a mini-batch size of 32 was employed. The momentum coefficient was set to 0.35, a value that struck an optimal balance between convergence speed and sensitivity to the initial learning rate parameter.

5.6 Regularization Techniques

Regularization techniques are methods within machine learning that helps in avoiding overfitting while also incresing model interpretability. One common regularization technique is pentaly term known as the *L2-norm*. In this

tequnice one add an extra term to the cost function which is proportional to the size of the weights. By doing so, one simply constrain the size of the weigts, so that they never will grow arbitrarilt large to fit the training data. This way, the regularization technique, reduces the chances of overitting the model [**Hjorth-Jensen2022**].

The size of the weights can be measured using the L2-norm, meaning the cost function will take the form:

$$C(\theta) = \frac{1}{N-1}\sum_{i=0}^{N}(y_i - \tilde{y}_i)^2 + \lambda \sum_{ij} w_{ij}^2. \tag{5.13}$$

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.

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.

5.7 Learning Rate Scheduling

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.

5.8 Metrics of success

In the realm of DiLoc, our neural network tailored for EEG source localization, assessing network performance through standard loss plots becomes less informative due to the normalization of target values. As a result, we turn to a separate, unseen test dataset comprising 20 000 samples to evaluate the network's accuracy. Before comparison, the predictions outputted by DiLoc are denormalized to facilitate a meaningful evaluation against true target values.

To comprehensively gauge the network's predictive abilities on this test dataset, we employ a diverse set of error metrics. While the primary focus is on minimizing the mean Euclidean distance of dipole positions and the absolute error for amplitude and radius, a range of other metrics are also explored for a comprehensive assessment. These metrics include mean absolute error (MAE), normalized mean absolute error considering the value range (NMAE), mean squared error (MSE), and root mean squared error (RMSE).

While metrics such as MAE, NMAE, MSE, and RMSE offer insights into the network's performance and predictions, their clinical interpretation can be intricate in the problems of ours. To address this, we establish threshold values that represent acceptable errors for a majority of predictions. In particular, we are interested in determining the percentage of samples for which the network predicts the Euclidean distance of one or more dipoles within specific thresholds—3 mm, 5 mm, 10 mm, and 15 mm, where 3 mm is considered optimal.

Regarding amplitude and radius predictions, the analysis involves studying the percentage of samples where the network provides predictions with absolute errors equal to 1, 2, and 3 mAμ m, and 1, 3, and 5 mm. Providing a MAE for the amplitude equal to 3 mAμ m corresponds to an error of 30%,and a MAE for the radius equal to 5 mm corresponds to an error of 33%, both of which are intuitively considered large errors. However, if we consider these target values in relation to potential clinical significance, a different perspective emerges. In real-world clinical cases, it could be of significant interest to discern whether a neuron source exhibits the characteristic

of a small amplitude (ranging from 1 to 3 mAμ m), a medium amplitude (ranging from 3 to 6 mAμ m), or a strong amplitude (ranging from 4 to 10 mAμ m). Similarly, the radius values of a small radius (ranging from 1 to 5 mm), a medium radius (ranging from 5 to 10 mm), or a large radius (ranging from 5 to 10 mm) might hold clinical significance when considering the underlying neuronal mechanisms.

This shift in perspective highlights the nuanced interpretation of errors and underlines the importance of clinical context in evaluating the performance of DiLoc. In this light, even what might initially appear as substantial errors can offer valuable insights into the behavior of neuronal sources within real-world scenarios.

In sum, the suite of error metrics, coupled with threshold-based assessments, facilitates an in-depth evaluation of the network's capabilities. This multi-faceted approach bridges the realms of machine learning principles and clinical applicability, encapsulating the overarching goal of achieving accurate, meaningful, and real-world clinical predictions.

5.9 Method that not belong here, but rather for the extended problems

5.9.1 Choosing an Optimal Cost Function

In the field of machine learning, *cost functions* play a crucial role in evaluating how well models make predictions. These mathematical functions compare predicted outcomes to actual values, resulting in a quantifiable metric known as *loss*. Higher loss values indicate poorer model performance, while lower values reflect more accurate predictions.

Selecting the appropriate cost function is pivotal in addressing machine learning challenges. In regression tasks like ours, the mean squared error (MSE) is a widely used cost function, particularly suitable for linear regression. The MSE is preferred due to its simplicity and continuous measure of model performance during training. It calculates the squared differences between the model's predictions and target values, then takes the mean across the entire dataset:

$$\mathrm{MSE}(\boldsymbol{\theta}) = \frac{1}{N-1}\sum_{i=0}^{N}(y_i - \tilde{y}_i)^2, \tag{5.14}$$

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

Due to the squaring step in the MSE function, which penalizes larger errors more severely, the trained model is less likely to generate outlier predictions with substantial errors. However, in situations where the model

produces a single, highly inaccurate prediction, the squaring effect can significantly amplify the error. Nevertheless, in many practical scenarios, the primary focus is not on these individual outliers, but rather on achieving a well-balanced model that delivers satisfactory performance across the majority of predictions [**builtin-ml-loss-functions**].

In our simulation-driven context, devised to accommodate a spectrum of diverse variations, it's noteworthy that the presence of apparent outliers is not rooted in random fluctuations, but rather indicative of distinct scenarios. As delineated in statistical literature [**barnett1994outliers**], an outlier is characterized as an observation that seemingly deviates from the prevailing distribution of a dataset. This departure can be attributed to human or instrumental anomalies in the data acquisition process [**zhang2015outlier**]. Figure 5.2 vividly depicts the distribution of minimum and maximum EEG observations across individual samples within the dataset. While the majority of data points exhibit values within the range of -1.5 to 3 μV, a subset extends beyond these bounds. It's crucial to underscore that these data points don't emerge as stochastic errors stemming from the acquisition process. Instead, they intentionally embody distinctive EEG signals that enrich the dataset's representational fidelity within authentic real-world contexts.

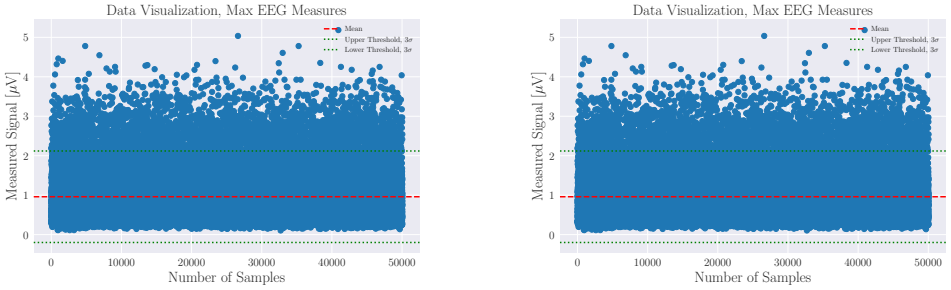

Figure 5.2: **Visualization of data set.** Both panels visualize the maximum EEG measures for each sample within the data set.

When having clear understandings of desired model outcomes, customized cost functions, designed to align precisely with specific objectives may potentially outperform the MSE cost function. Our objective is to create a model that first of all is capable of accurately predicting single dipole positions, and as extensions also multiple dipole positions, amplitudes of dipole signals and radii of extended dipole populations. To achieve this, the ideal cost function should minimize the Euclidean distance between target dipole localizations $\theta_{x,y,z}$ and true values $\tilde{\theta}_{x,y,z}$:

$$\mathrm{MED_{x,y,z}}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=0}^{n-1}\sqrt{(\theta_{x,i}-\tilde{\theta}_{x,i})^2+(\theta_{y,i}-\tilde{\theta}_{y,i})^2+(\theta_{z,i}-\tilde{\theta}_{z,i})^2}, \quad (5.15)$$

Additionally, the cost function should minimize the absolute error between predicted θ_{A} and true amplitude $\tilde{\theta}_{\mathrm{A}}$:

$$\mathrm{MAE}_{\mathrm{A}}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=0}^{n-1} \|\theta_{A,i} - \tilde{\theta}_{A,i}\|, \tag{5.16}$$

Similarly, it should minimize the error between predicted θ_{r} and true radius$\tilde{\theta}_{\mathrm{r}}$:

$$\mathrm{MAE}_{\mathrm{r}}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=0}^{n-1} \|\theta_{r,i} - \tilde{\theta}_{r,i}\|, \tag{5.17}$$

Finally, the cost function should minimize the Euclidean distance among a set of m dipoles:

$$\begin{aligned}\mathrm{MED}_{\mathrm{x}_1,\mathrm{y}_1,\mathrm{z}_1,\ldots,\mathrm{x}_m,\mathrm{y}_m,\mathrm{z}_m}(\boldsymbol{\theta}) = \\ &\left(\frac{1}{n}\sum_{i=0}^{n-1}\sqrt{(\theta_{x_1,i}-\tilde{\theta}_{x_1,i})^2+(\theta_{y_1,i}-\tilde{\theta}_{y_1,i})^2+(\theta_{z_1,i}-\tilde{\theta}_{z_1,i})^2}\right)+\\ &\left(\frac{1}{n}\sum_{i=0}^{n-1}\sqrt{(\theta_{x_2,i}-\tilde{\theta}_{x_2,i})^2+(\theta_{y_2,i}-\tilde{\theta}_{y_2,i})^2+(\theta_{z_2,i}-\tilde{\theta}_{z_2,i})^2}\right)+\ldots+\\ &\left(\frac{1}{n}\sum_{i=0}^{n-1}\sqrt{(\theta_{x_m,i}-\tilde{\theta}_{x_m,i})^2+(\theta_{y_m,i}-\tilde{\theta}_{y_m,i})^2+(\theta_{z_m,i}-\tilde{\theta}_{z_m,i})^2}\right)\end{aligned} \tag{5.18}$$

While the built-in MSE cost function compares predicted and target values linearly, the customized cost function should calculate all possible permutations, ensuring the correct combinations are used for loss calculation.

Whereas the standard built-in MSE cost function calculates the mean squared error between each of the target values, we require the customized cost function to map the predictions of each location for the set of dipoles, effectively minimizing the Euclidean distance. Utilizing Python's built-in MSE cost function, the algorithm simply matches predicted locations with the target locations in the order that the vectors have been arranged. However, this technique lacks exploration of other combinations, potentially misleading the network's weight updates if certain accurate locations are paired with non-preferable targets. By enabling the customized cost function to compute all possible permutations, we ensure that the correct target and predicted location values are paired during loss calculation. To achieve this, the cost function calculates all permutations and selects the one yielding the minimum loss.

For all terms within the cost function, comprehensive unit tests have been developed to confirm its intended functionality. Each problem introduced for the network corresponds to a distinct form of the customized cost function:

$$C(\boldsymbol{\theta}) = \begin{cases} \text{MED}_{\text{x,y,z}}(\boldsymbol{\theta}), & \text{if } \|\boldsymbol{\theta}\| = 3 \\ \text{MED}_{\text{x,y,z}}(\boldsymbol{\theta}) + MAE_{\text{A}}(\boldsymbol{\theta}), & \text{if } \|\boldsymbol{\theta}\| = 4 \\ \text{MED}_{\text{x,y,z}}(\boldsymbol{\theta}) + MAE_{\text{A}}(\boldsymbol{\theta}) + MAE_{\text{r}}(\boldsymbol{\theta}), & \text{if } \|\boldsymbol{\theta}\| = 5 \\ \text{MED}_{\text{x}_1,\text{y}_1,\text{z}_1,\ldots,\text{x}_m,\text{y}_m,\text{z}_m}(\boldsymbol{\theta}), & \text{otherwise} \end{cases} \tag{5.19}$$

Here, $|\boldsymbol{\theta}|$ signifies the length of the target vector. When $|\boldsymbol{\theta}| = 3$, the simplest problem is considered, where the network predicts the coordinates of a single-point current dipole. If $|\boldsymbol{\theta}| = 4$, the network predicts the x, y, and z-coordinates of a single dipole, in addition to the amplitude of the signal strength. When $|\boldsymbol{\theta}| = 5$, the target vector encompasses all previously mentioned values, along with the size of a current dipole population with radius. Finally, for $|\boldsymbol{\theta}|$ greater than 5, the multiple dipole problem is addressed, where the network predicts the locations for two or more point source dipoles situated at distinct positions within the cortex.

In crafting our customized cost function, it is important to acknowledge that, like the built-in Mean Squared Error (MSE) cost function, our formulation inherently treats all target values equally during the optimization process. In other words, the algorithm assigns the same weight to each target value when striving to reduce the overall loss. This approach ensures that errors of equal percentage magnitude in different target values are treated on a level playing field. Consequently, a 1% error for one target value is considered as important as a 1% error for another target value, regardless of the specific range or scale of these values.

It is worth noting that our choice to uniformly weight all target values is an intentional design decision. While alternative approaches, such as assigning different weights to different target values, could have been explored, we prioritize the creation of a balanced model that can accurately predict all facets of our target values. This approach stems from our objective of achieving a comprehensive understanding of EEG signal sources through a holistic and equitable modeling approach.

Moreover, this uniform weighting approach aligns with our broader modeling philosophy, emphasizing the creation of a model that is versatile and adaptable across a spectrum of EEG data variations. Our aim is not only to develop a model capable of accurately predicting target values but also to ensure that its predictive capabilities are unbiased and comprehensive, covering the multifaceted aspects of EEG signal analysis.

In this way, our customized cost function showcases the fusion of machine learning principles with the nuanced requirements of clinical medicine, as we

strive to bridge the gap between technical prowess and real-world medical applications.

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 using DiLoc

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.

6.1.1 Performance Evaluation

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

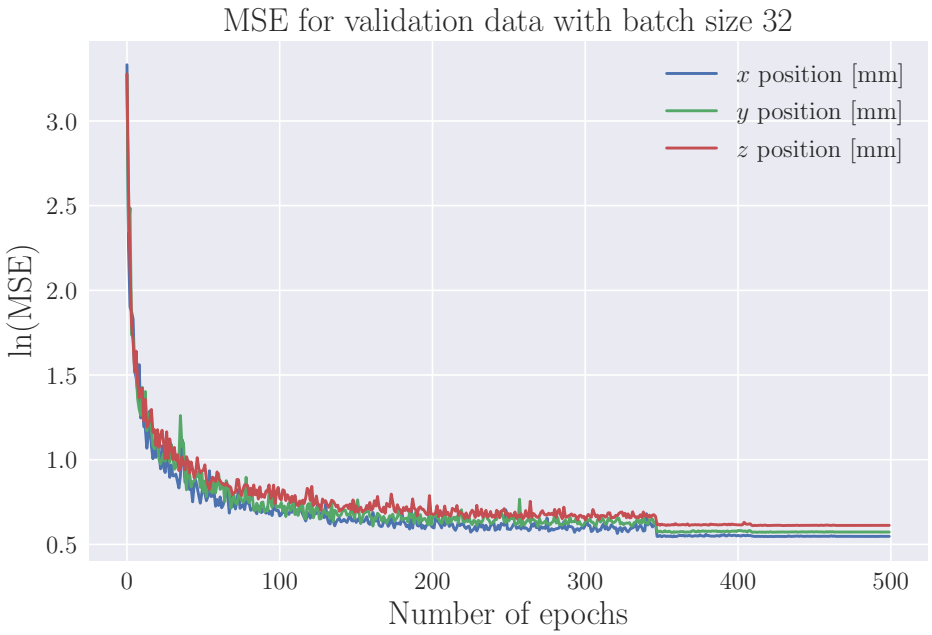

Figure 6.2: Validation loss for the separate target values; the x-, y-, z-coordinate.

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 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. To showcase the networks performance, we can look at one specific prediction of the network. For a dipole located at (x, y, z) coordinates of (66.9 mm, -26.1 mm, 41.7 mm) the network predicted the coordinates to be (66.5 mm, -26.4 mm, 41.9 mm). The predicted values were very close to the true values, with an error of only 0.4 mm in the x-coordinate, 0.3 mm in the y-coordinate, and 0.2 mm in the z-coordinate.

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).

6.1.2 Detailed Analysis of Performance at Different Brain Structures

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 implications 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 Convolutional Neural Network Approach for Localizing Single Dipole Sources

In this section, we explore the utilization of a Convolutional Neural Network (CNN) for the task of localizing simple current dipoles from EEG recordings. The CNN is a sophisticated type of feed-forward neural network that excels at learning spatial features from images. The objective of this investigation is to assess whether leveraging spatial information in EEG recordings as images can enhance Dilocs's ability to analyze the data and yield more accurate predictions for localizing the sources generating the neural signals.

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**].

6.2.1 Data Set

Convolutional Neural Networks (CNNs) are well-known for their effectiveness in processing image data. To harness the potential of CNNs for EEG data analysis, we convert the original dataset presented in Chapter 4 into image-like data through interpolation. Interpolation, a widely-used mathematical technique, estimates values between known data points. This transformation results in a regular 2D grid representation of the original one-dimensional EEG data, effectively creating EEG data with image-like characteristics and preserving spatial structures.

The resulting data is represented as a 20x20 matrix, where each element holds the intensity value of the EEG potential recorded at the corresponding electrode location. We construct this matrix to resemble the shape of a grayscale image, with a single channel added to represent the spatial distri-

bution of the measured EEG signals. However, unlike typical grayscale images where each pixel represents color intensity, in this context, each pixel's value denotes the intensity of the recorded EEG signal at that specific electrode location. This unique representation enables the CNN to leverage the spatial arrangement of the EEG data and learn relevant patterns and local relationships, much like how CNNs process traditional image data.

The preserved spatial structures enable the network to exploit local relationships between neighboring recording electrodes. For instance, when neighboring electrodes record high EEG values, it suggests that the specific electrode should also record a relatively high EEG value due to their close spatial proximity. These spatial relationships may contribute to faster training times for the network, as it can efficiently learn meaningful representations by leveraging these patterns.

Figure **??** illustrates the process of interpolation. The right panel shows the original data, representing the cortex seen from above (x-y-plane). Each measuring electrode is depicted as a circle holding the EEG recording at that specific electrode. The middle and left panels display the contour plots of the original EEG data and the interpolated data, respectively. The contour plot of the interpolated data illustrates how the input data for the convolutional neural network appears in an image-like manner.

Architecture, Hyperparameters and Training

As explained in Chapter 3, Convolutional Neural Networks (CNNs) are structured as a sequence of interconnected layers designed to process and extract meaningful features from the input data. In the case of EEG input data, we adopt a specialized CNN architecture tailored to handle image-like data representations. The data transformation involves constructing a 20x20 matrix, akin to the shape of a grayscale image, with a single channel added to represent the spatial distribution of the measured EEG signals.

The first layer in the network, is a 2D convolutional layer. It takes the input image with one channel and applies six distinct filters, each of size 5x5. These filters are responsible for learning specific spatial patterns and detecting relevant features within the input image. As a result of this convolutional operation, the output tensor's spatial dimensions reduce to 16x16, and the depth becomes six, signifying the extraction of six distinct feature maps. Following the convolution layer, a Max Pooling layer, with kernal size 2x2 and stride 1 is employed. This pooling layer aims to downsample the spatial dimensions of the feature maps while preserving the most salient features. The pooling operation reduces the spatial resolution to 15x15, and the depth remains unchanged at six. Next, a second 2D convolutional layer, takes the six-channel output from the previous pooling layer. This layer employs 16 filters of size 5x5, extracting a more complex hierarchy of features from the input data. The output tensor from this layer has spatial

dimensions of 11x11 and a depth of 16, signifying the presence of 16 distinct feature maps. Following another Max Pooling layer is employed. Similar to the previous pooling operation, this layer further downsamples the spatial dimensions while preserving the depth, resulting in a feature map size of 10x10 with 16 channels. Further, the output from the last pooling layer is flattened into a one-dimensional vector. This process collapses the spatial dimensions of the feature maps, resulting in a 1D tensor of size 1600 (10x10x16). After flattening, the network proceeds with three fully connected, dense, layers. These layers are responsible for incorporating global context and making high-level abstractions from the learned features. The first fully connected layer, consists of 120 neurons, followed by 64 neurons. Lastly,we have the output layer with three neurons, corresponding to the three coordinates of the source generating the eeg signal. In Figure 6.5, we have provided an illusration of the architecture of the Convolutional Neural Network.

The activation function ReLU is applied after each convolutional and pooling layer, introducing non-linearity to the network and enabling it to learn complex relationships within the data. The fully connected layers use the hyperbolic tangent activation function, which introduces non-linearity and scales the output between -1 and 1. Finally, 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.Throughout the network, the weights of the fully connected layers are initialized using the Xavier normal distribution, a widely used technique to set initial weights in deep neural networks, promoting better convergence during training.

The training process for the specialized convolutional neural network (CNN) followed similar techniques to the original DiLoc network, as described in detail in Chapter 5. To ensure effective learning and accurate predictions, stochastic gradient descent (SGD) with momentum was utilized as the optimizer, and mean squared error (MSE) served as the chosen cost function. Additionally, L1 and L2 regularization techniques were incorporated to mitigate overfitting and enhance the network's ability to generalize to new data. During training, mini-batches of size 32 were employed to introduce variability in the data and prevent the network from becoming stuck in local minima. Notably, the CNN employed specific hyperparameters, setting the learning rate to 0.001 and the momentum to 0.009, which were tailored to accommodate the processing requirements of image-like EEG data. To facilitate convergence, a learning rate scheduling approach was adopted, gradually reducing the learning rate during training, striking a balance between rapid initial convergence and fine-tuning towards the later stages. Subsequently, the CNN's performance was rigorously evaluated on an independent test dataset to provide an unbiased assessment of its predictive accuracy and generalization capabilities to novel data.

6.2.2 Performance Evaluation

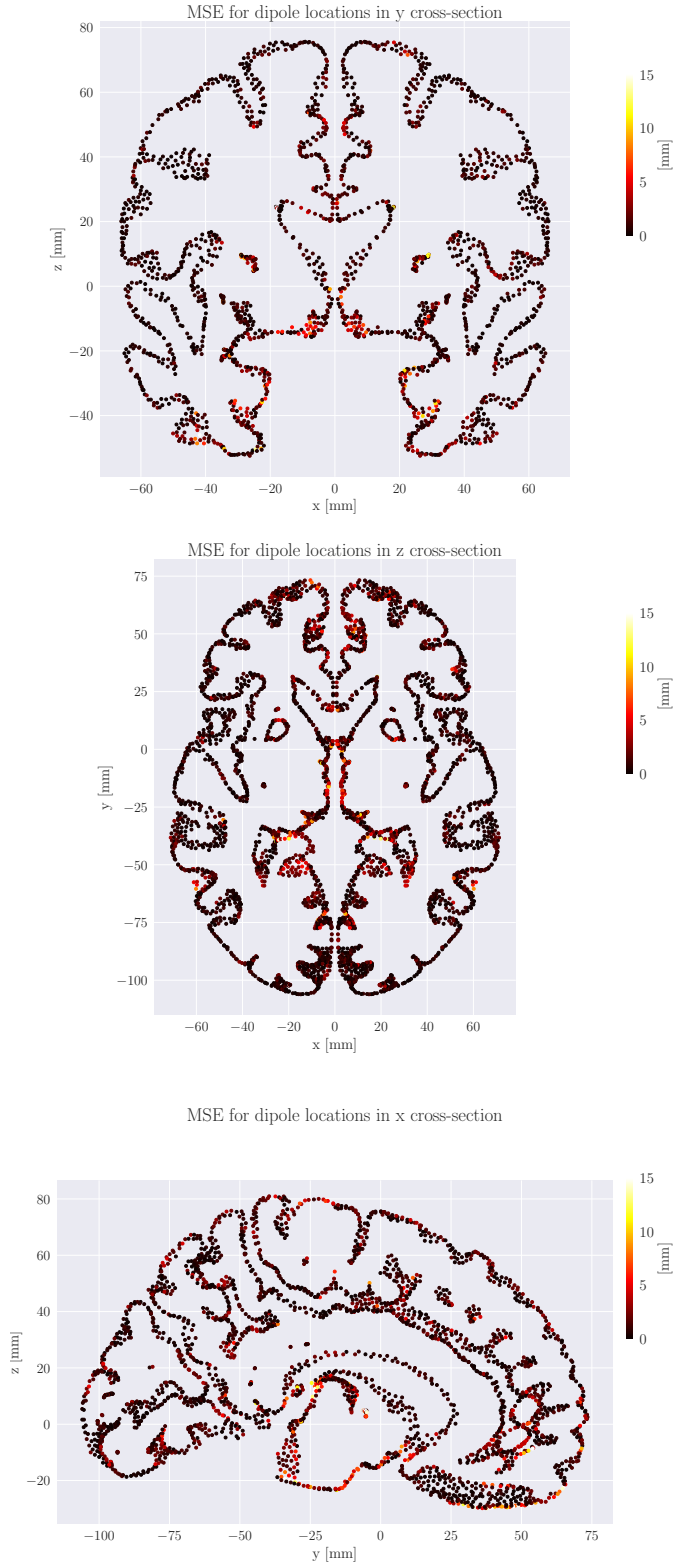

Figure 6.3: Different cross-sections of the cortex from the New York head model, seen from front, top and side. Each point represents a possible position in the cortex matrix. The color of the each point indicates the mean absolute error (MAE) of the neural network when predicting that specific dipole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

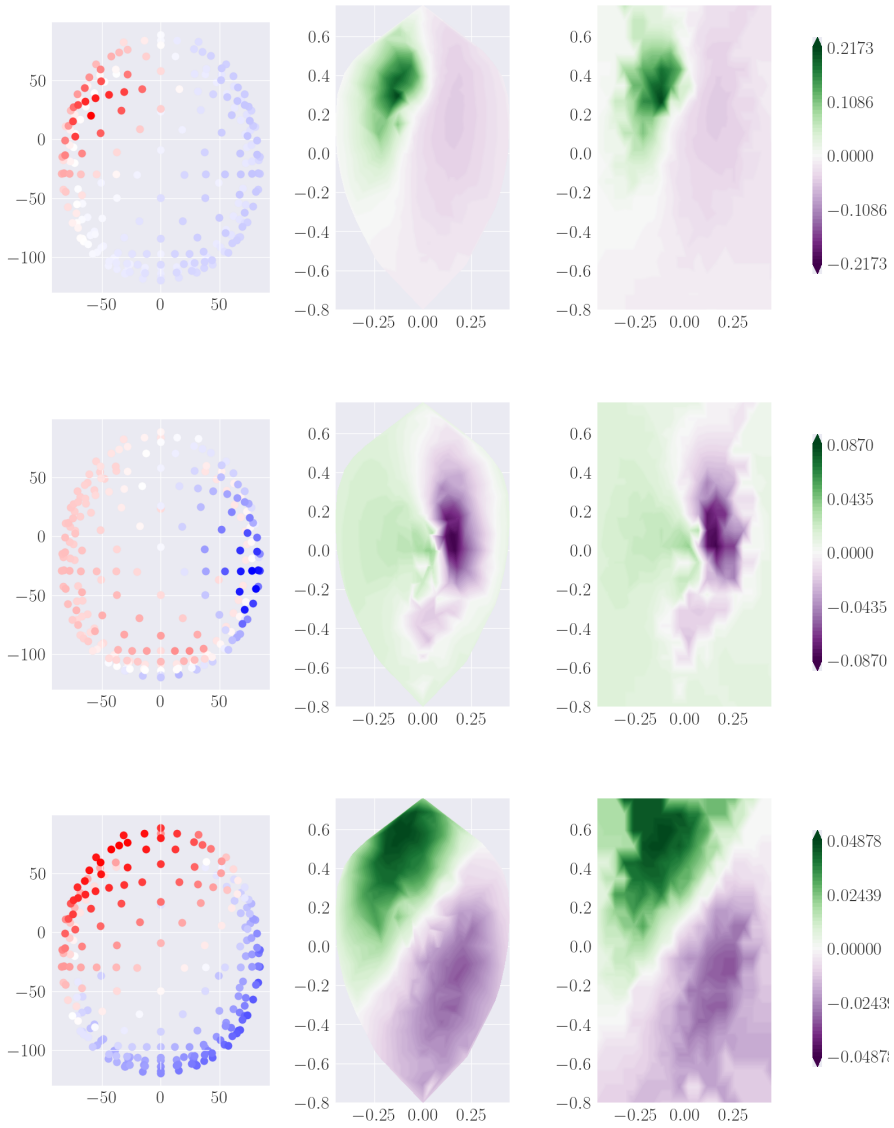

Figure 6.4:
Right Panel: EEG measures for three different samples, expressed in microvolts (μV). Each sample represents an EEG recording at specific electrode positions.
Middle and Left Panels: Illustration of the interpolation of the EEG data into a two-dimensional matrix. The interpolated data represents the transformation of original electrode recordings into a regular 2D grid, effectively converting the one-dimensional EEG data into an image-like format. The contour plots visualize the spatial distribution of EEG potential intensities, with each point in the matrix corresponding to a specific electrode location.

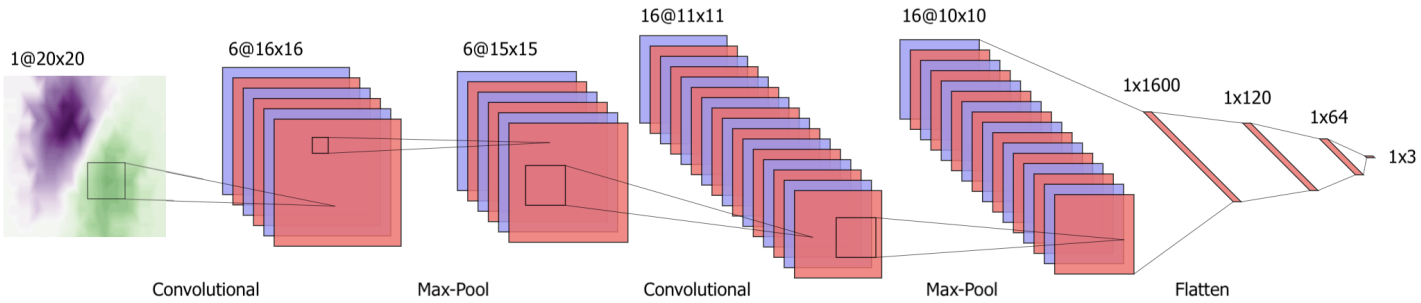

Figure 6.5: The architecture of the Convolutional Neural Network.

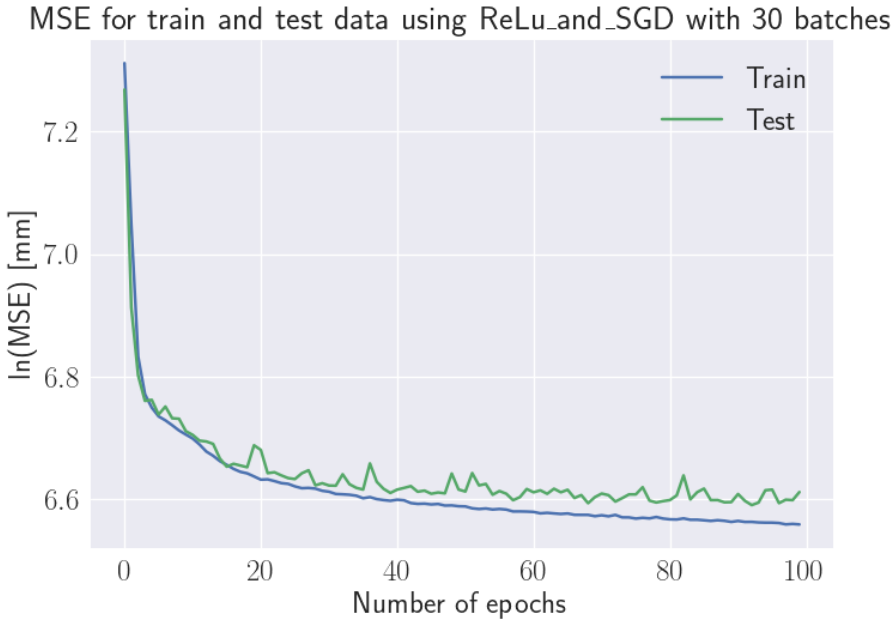

Figure 6.6: The validation accuracy for Convolutional Neural Network with 10 000 samples (20x20 matrix) with ReLU activation function.

Chapter 7

Extending the DiLoc Network

In this chapter, we explore how various extensions and small modifications of the DiLoc network enhance its capability to address intricate and demanding inverse problems. We delve into three challenging scenarios, each designed to push the boundaries of DiLoc's predictive performance. The first extension involves assigning individual amplitudes to each dipole source, presenting the network with the task of predicting both the source locations and their corresponding amplitudes. Subsequently, the second scenario transitions from predicting the location of a single dipole source to estimating the center and radius of a population of dipoles, alongside the amplitude of the electrical signals generated. This extension introduces additional complexities to the localization process. Lastly, we investigate DiLoc's ability to predict the locations and amplitudes of two individual dipole sources that jointly contribute to the EEG signals recorded by the electrodes. These extensions aim to comprehensively evaluate the network's adaptability and generalization to increasingly intricate real-world situations. Throughout the chapter, we systematically evaluate the network's performance, providing insights into its strengths and limitations when confronted with these novel challenges.

7.1 Predicting Single Dipole Sources with Amplitudes

In this section, we introduce the concept of various amplitudes for single current dipole sources, which adds an additional dimension to the output of DiLoc. 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.

7.1.1 Adjustments in Data Set and Architecture

We assign amplitudes to each dipole ranging between 1 and 10 nAμm. By now the dataset still has the same number of features, however 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. We observe that the shape of the EEG signal remain consistent, while the magnitude of the EEG signal is highest for the dipole with the largest amplitude. It should therefor not be a problem for the network to separate such cases and the network shold be able to provide accurate predictions for the amplitude in both cases. From the Figure it is also apperent that the EEG recordings ranges between -10 and 10 μV. Can I find litterature that support the range?

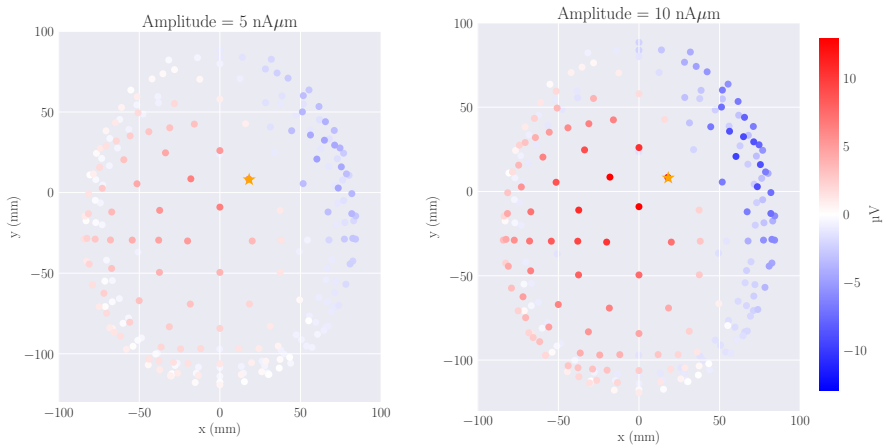

Figure 7.1: EEG data for two samples with current dipole amplitude equal to 5 and 10 nAμm. The EEG recordings have a range between -10 and 10 μV.

In figure 7.2 we have depiced the construction of the extended DiLoc network, that now also provides for the amplitude. 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 the strenght of the signal for the current dipole moment in the cortex.

As was done for the previous problem, the input data is scaled by subtracting the mean and dividing on the variance. However, as DiLoc deals with multipole output units, specifically millimeters (mm) and nanoampere-micrometers (nAμm), a scaling process is implemented also on the output targets to effectively minimize the cost function. The cost function calculates the differences between the predicted output and the actual target values.

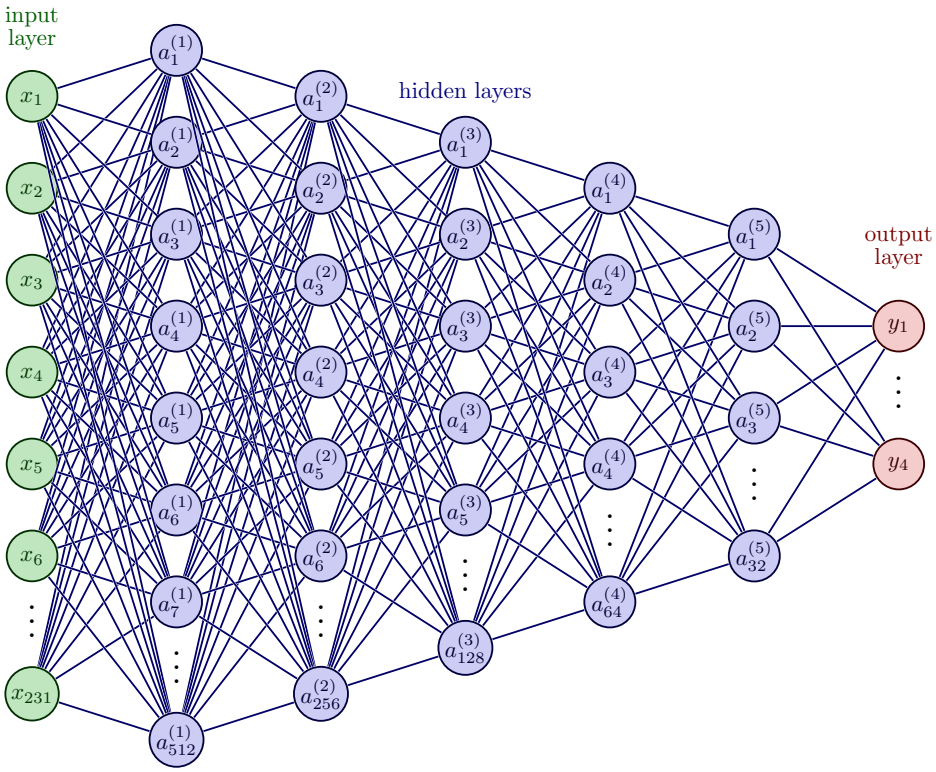

Figure 7.2: Architecture dipole with amplitude.

When output values have different ranges and units, there is a risk that certain dimensions of the target values may dominate the overall error calculation, while others with smaller ranges might be neglected. Consequently, the neural network might overly prioritize reducing errors in the larger range values, hindering its ability to accurately learn patterns and generalize well for the smaller range values.

To address this issue, the output values are normalized to a common range of 0 to 1 using the following normalization formula:

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

Here, z_i represents the i^{th} normalized value in the dataset for a specific target category, x_i is the i^{th} value in the corresponding target dataset, and $\min(x)$ and $\max(x)$ are the minimum and maximum values in that specific target dataset.

It is important to perform this normalization separately for each target category. Doing so allows the neural network to effectively train and discern patterns using only one cost function. Without this normalization, employing a single cost function for a set of different target values with distinct units

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	32
Epochs	5000
Dropout	0.5
Act.func in first layer	ReLU
Act.func in hidden layers	Tanh
Act.func in last layer	Sigmoid

would not be feasible. By applying this normalization, we aim to provide a more balanced cost function where all target values contribute equally to the overall error calculation. Consequently, the network can learn effectively from the data and achieve better results in its tasks.

In the extension of the DiLoc network, we maintain the use of ReLU as the activation function in the first layer, and hyperbolic tangent for the hidden layers, as this architecture, combined with the choise of parameter valued gave the best results. However, considering that the output data has been normalized to a range from 0 to 1, we deem it appropriate to employ the Sigmoid activation function in the output layer. The Sigmoid function maps the output values to a range between 0 and 1, which aligns with our desired output range. This choice may potentially facilitate the training process, as it enables the network to converge more effectively towards the desired outputs.

As for the simple DiLoc model, we continue to utilize the technique of adaptive learning rate, which can be advantageous for optimizing the network's parameters more efficiently. For an overview of the overall parameters employed in the model, please refer to Table **??**, which provides a summary of these essential elements.

7.1.2 Performance Evaluation

To assess the network's performance, we start by analyzing 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

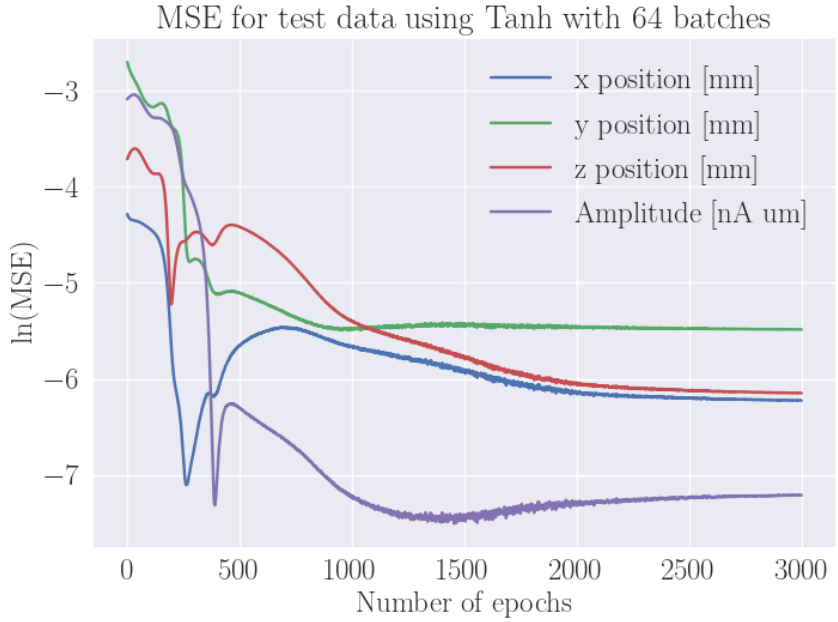

Figure 7.4: The loss developmnent for the different target values as function of epochs.

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 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 extend the DiLoc neural network to incorporate varying radii and amplitudes for the origins generating the electrical activity detected by the recording electrodes. This transformation alters 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 real-life scenarios where understanding the extent of brain damage causing abnormal activity in damaged areas may be of interest. By training the DiLoc network on such complex data, we aim to enhance its ability to generalize and perform effectively in real-world clinical cases.

7.2.1 Adjustments in Data Set and Architecture

For the purpose of enabling the network to predict the areas of dipole populations, we make adjustments to the dataset. The dipole populations are represented as spherical volumes in the NY head cortex, with the radius for each population ranging from 1 mm to 15 mm. To ensure realism, we maintain the maximum amplitude strength of the total populations at 10 mAμm. Consequently, we calculate the maximum number of points within a volume sphere with a radius of 15 mm and reduce this number by 10 to determine the strength of each dipole within the given area. This leaves us with a strength of 10/899 for each dipole. The strength of a dipole population is thus directly proportional to the size of the dipole population. While this may not perfectly represent real-world scenarios, it provides a reasonable approximation for our model.

In Figure 7.5, we present an example of a dipole population and the corresponding EEG signal. The yellow filled circles in the plots in the up-

per panel represents the diple populations, i.e. positions within the cortex where dipoles have been placed. The lower panel shows the EEG signals for the specific sample, with EEG electrode locations presented as filled circles, where the color of the fill represents the amplitude of the measured signal for the given electrode. The plots within the figure are seen from both the x-z plane, x-y plane, and the y-z plane.

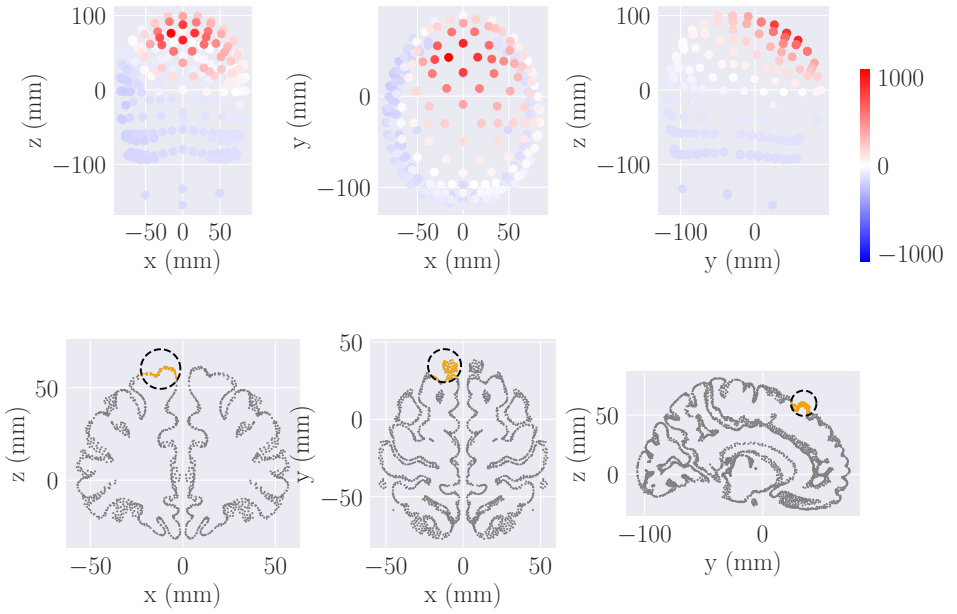

Figure 7.5: EEG for a sample containing a spherical population of current dipole sources with a random center within the cerebral cortex. The EEG measure is seen from both sides (x-z plane and y-z plane) and above (the x-y plane). EEG electrode locations are presented as filled circles, where the color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

As for the dataset, the number of target values is now 5: x, y, z-coordinates of the center of the dipole population, amplitude, and radius. The number of features is not modified and still holds the number of 231, representing the recording electrodes. The new architecture of the DiLoc network is presented in Figure **??**.

Similar to the previous problem, we normalize the target values to ensure they all range from 0 to 1. Moreover, in this extension of the DiLoc network, we use the same activation functions as in the previous problem with ReLU as the activation function in the first layer, hyperbolic tangent for the hidden layers, and the Sigmoid activation function in the output layer. As with the previous problems, we have explored various network architectures and activation functions, but the current configuration has shown the best per-

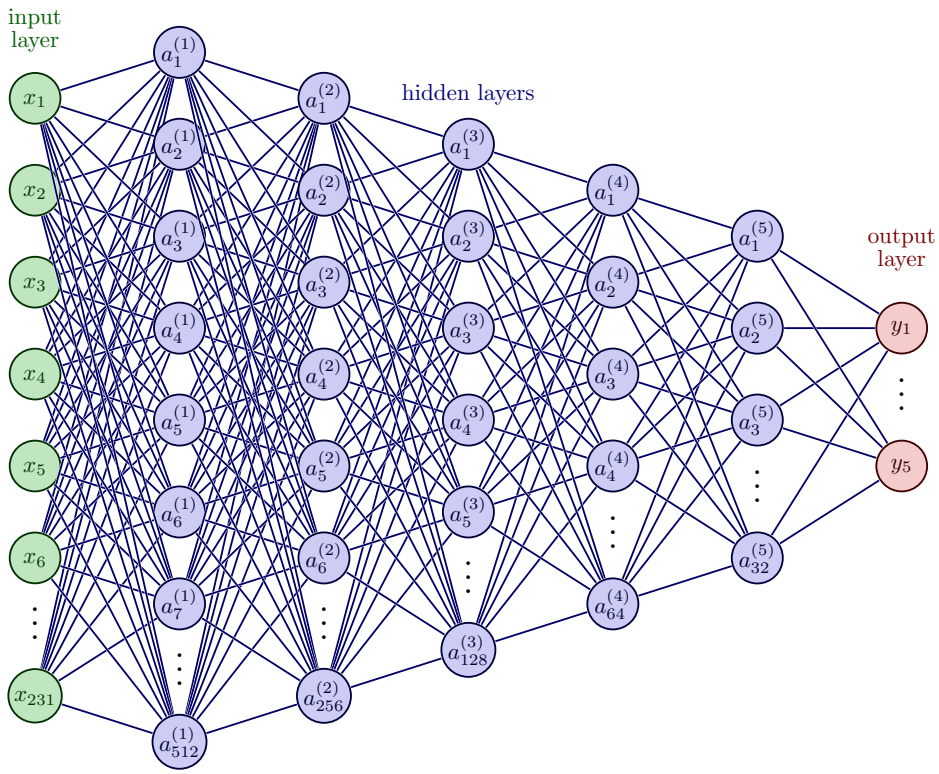

Figure 7.6: Architecture of the dipole area prediction network.

formance in terms of accurate predictions for this problem. It is important to emphasize that our primary goal is to find a network that can effectively solve the problem and provide accurate predictions, rather than necessarily seeking the best possible configuration.

7.2.2 Performance Evaluation

In Figure **??**, we present the training and validation Mean Squared Error (MSE) loss for the ConvDip network as a function of epochs. The network was trained for 5000 epochs, but the validation loss appears to converge at around 3000 epochs, while the training loss stabilizes at approximately 4000 epochs. Notably, there are no signs of overfitting, which is a positive outcome. Each epoch took approximately 7 seconds, resulting in a total training period of approximately 9 hours.

Figure 7.8 displays the validation loss for each target value as a function of epochs. The losses for all coordinate target values (x, y, and z) are minimized almost identically, with the y-coordinate loss slightly smaller. The amplitude target value is minimized most effectively by ConvDip, while the radius target value has the highest loss. We keep in mind that the amplitude and

radius taget value correlates to some extent, as the amplitude is proportional to the radius. Although the exact MSE values for the target values cannot be directly read from the figure due to normalization, the overall pattern indicates that ConvDip successfully captures data patterns and minimizes the cost function.

Figure 7.7: 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.

To assess the extent to which the network can predict the center of the dipole populations, in addition to amplitude and radius, we utilize the same evaluation metrics as described in chapter 6. Table 7.2 presents the Mean Absolute Error (MAE), Normalized MEan Absolute Error (NMAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) for the different sets of target parameters.

The MAEs for all coordinates and the Euclidean distance lie between 4 and 5 millimeters. Looking at the NMAE, we observe that the loss for the z-coordinate is somewhat larger than for the other coordinates, with 3 %, similar to our observations when testing DiLoc's ability to predict amplitude in addition to location. However, this slightly larger error is not significant and may as well be attributed to randomness. What is worth mentioning is that all cordinates

As for the amplitude and radius targets, the MAEs are remarkably small. For amplitude, ranging from 1 to 10 mAμm, the absolute error is approx-

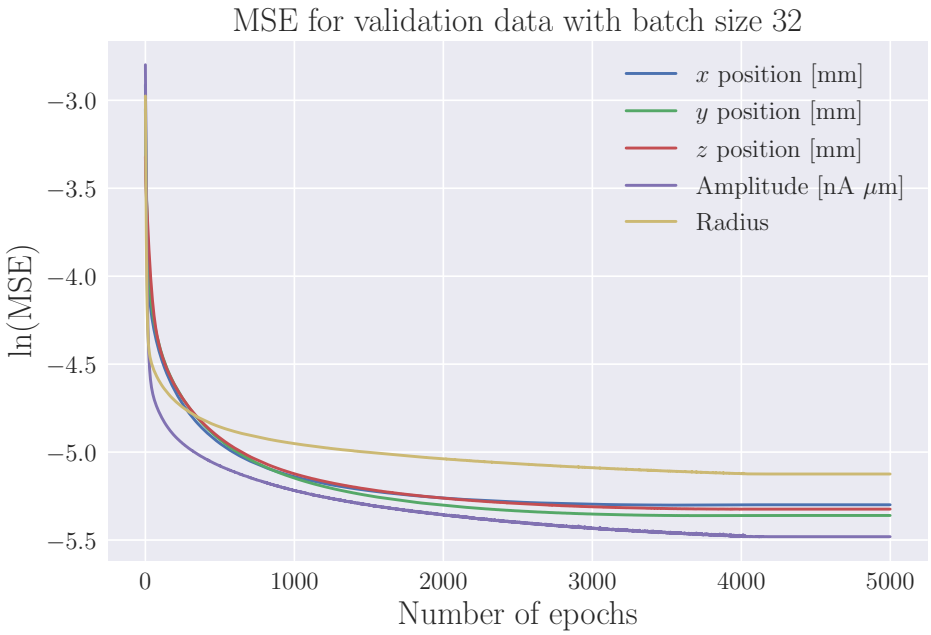

Figure 7.8: The validation accuracy as function of epoch for each target value: x, y, z-coordinates of the center of the dipole population, amplitude, and radius.

imately 4.33% of the range of the actual amplitude values, indicating that the model's amplitude predictions are reasonably close to the true amplitude values. Similarly, the MAE for the radius, with a range from 1 to 15 mm, is approximately 6.07%, suggesting that the model's predictions are relatively accurate for radius.

Regarding the MSE, we observe relatively small errors for the amplitude and radius, with values of 0.364 and 1.291 (mm^2) respectively. However, as for the coordinate target values, we encounter relatively larger MSE values. This difference in scale between the MAE and MSE suggests the presence of outliers.

7.3 Localizing Multiple Dipole Sources

In this final extension of the DiLoc neural network we want to train the model in predictinf the positions of not just one but two individual dipole sources, which collabraticely generate the recored EEG signal. This novel extension pushes the boundaries of the network's capabilities, requiring it to grapple with the complex task of identifying and localizing multiple distinct dipole sources within the brain.

	Error for different target values					
	x [mm]	y [mm]	z [mm]	Center [mm]	Amplitude [nAμm]	Radius [mm]
MAE	4.257	4.868	4.126	4.417	0.390	0.850
NMAE	2.955	2.711	3.083	2.916	4.333	6.071
MSE	47.141	68.776	46.192	54.036	0.364	1.291
RMSE	6.866	8.293	6.796	7.351	0.604	1.136

Table 7.2: **Evaluation of DiLoc utilizing different Error Metrics.** Performance of the extended DiLoc network on a test dataset consisting of 20000 samples. The errors are measured using Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) for various target values.

7.4 Previous work

We acknowledge that similar research has been conducted by other groups, including the developers of the ConvDip convolutional neural network. The ConvDip network was designed to produce inverse solutions for EEG data, specifically focusing on predicting the positions of varying numbers of sources from a single time point of EEG data.

The researchers behind ConvDip explored the feasibility of utilizing CNNs to solve the EEG inverse problem for multiple sources using training data that adheres to biologically plausible constraints. Similar to DiLoc, ConvDip was trained to operate on single time instances of EEG data and predict the positions of sources based on potentials measured with scalp electrodes. However, it is worth noting that unlike our approach, the ConvDip group considered dipole clusters rather than single dipoles. This approach aligns more closely with the previous problem in which we focused on dipole populations.

For generating the simulated data, the researchers created a source model consisting of 5124 dipoles distributed along the cortical surface (also referred to as the cortex). They selected 31 recording electrodes and computed the leadfield matrix using a head model with dipole orientations fixed orthogonally to the cortical surface, similar to our methodology. To enhance the realism of the training data, real noise from pre-existing EEG recordings conducted with the same set of electrodes was added. Additionally, the group created separate test data using an alternative head model to avoid potential overoptimistic results, a phenomenon they referred to as the "inverse crime." The training dataset consisted of 100,000 samples, while the test dataset comprised 1000 samples.

In order to prepare the EEG input data for spatial convolutions, it was interpolated onto a 2D image of size 7 x 11. As expected with interpolation,

this procedure does not introduce new information to the EEG data. The output of ConvDip is a vector of size 5,124, corresponding to the dipoles in the source model. For a comprehensive description of the ConvDip network, we refer readers to the paper: paper: https://www.frontiersin.org/articles/10.3389/fnins.2021.569918/full.

Although the complexity of our original DiLoc network (FFNN) is significantly smaller compared to ConvDip, we still desired to investigate its performance in this more challenging task.

INCLUDE THIS We will now evaluate the ability of ConvDip to estimate the correct size of sources and to correctly localize sources with varying depth.

7.4.1 Adjustments in Data Set and Architecture

To begin with, we simulate EEG data corresponding to the electrical signals originating from multiple individual dipoles located in the brain. Initially, we allow unrestricted distances between the dipole sources. However, to avoid overcomplicating the problem, we assign each dipole within a sample with the same magnitude of amplitude. Consequently, for the dipole population problem, the total amplitude for a set of dipoles is fixed at 10 mAμm. Figure 7.9 displays two plots of randomly selected samples, illustrating the simulated EEG data when multiple dipoles generate the signal. In the first sample, two dipoles generate the EEG signal, each having an amplitude of 2.4 mAμm. In the second sample, three dipoles generate the EEG signal, and each dipole has an amplitude of 1.13 mAμm.

The total number of target values for this problem has increased to 8, encompassing the x, y, and z-coordinates for the location, as well as the amplitude, of each dipole. Since we constrained each dipole within a sample to have the same amplitude, it is not necessary to have separate output values for the amplitudes of each dipole. Nevertheless, we modified the architecture of the network, considering the possibility of outputting amplitude target values with varying values for each dipole. Apart from this adjustment, the network still comprises 231 input nodes, and the target values have been normalized to range from 0 to 1. The logic and choice of activation functions, as well as hyperparameters, remain consistent with those used in previous problems. Figure 7.10 illustrates the updated network architecture.

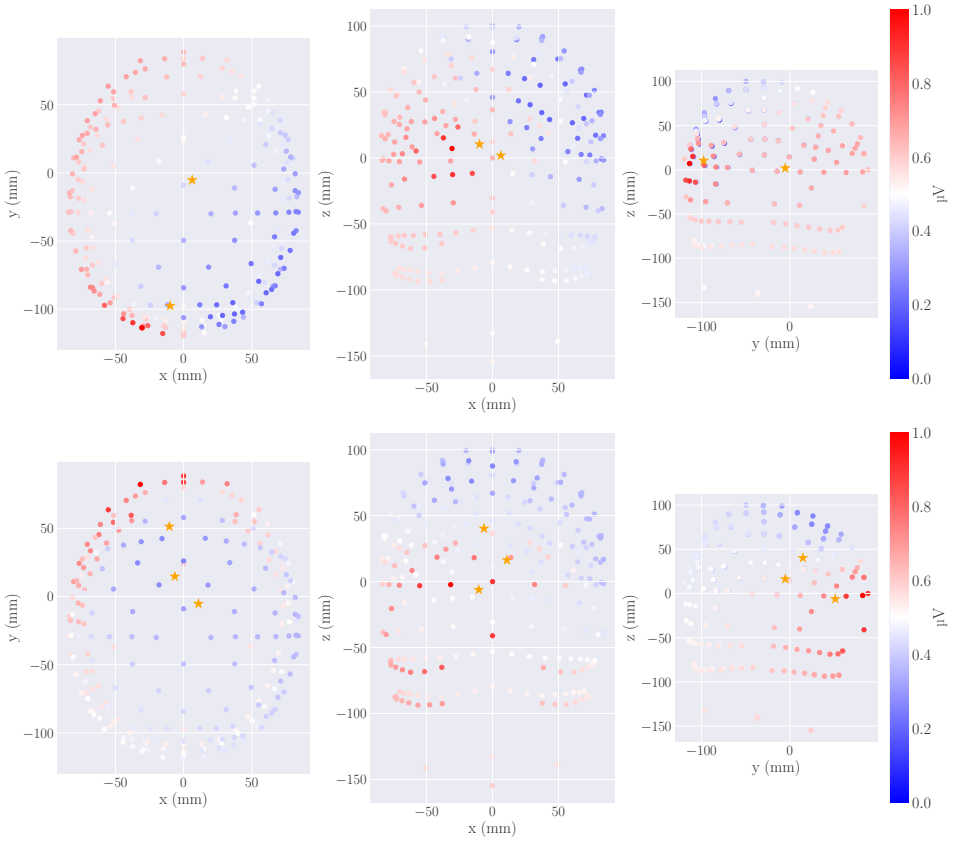

Figure 7.9: EEG for two samples containing two and three current dipole sources, respectively, at random positions within the cerebral cortex. The EEG measures are seen from both sides (x-z plane and y-z plane) and from above the skull (x-y plane). EEG electrode locations are presented as filled circles, 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.

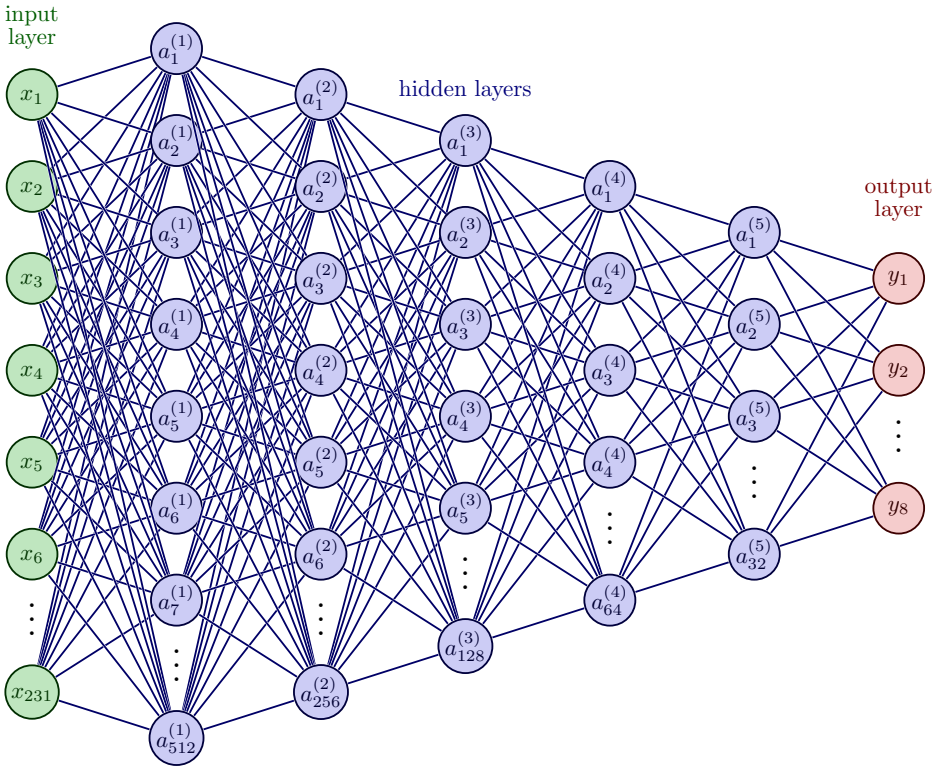

Figure 7.10: Architecture of the multiple dipoles network.

Figure 7.11: The validation accuracy for the simple Feed Forward Neural Network, predicting two current dipole sources.

Chapter 8

Discussion

This will be a chapter of discussion.

ReLU in first layer Vanishing/exploding gradients