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

Supervisors:
Torbjørn Ness and Gaute Einevoll

Acknowledgements

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

Contents

0.1 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 the so-called EEG inverse problem 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. 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 locations of the current dipoles have been obtained. Tools for calculating EEG signals from biophysically detailed neural simulations have however recently been developed, and are available through the software LFPy 2.0 [21]. 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 (ANNs). These methods have also been applied to EEG source localization with promising results. Prior research has demonstrated ANNs achieving localization errors below 5%, exhibiting computational efficiency, and robustness against measurement noise compared to traditional methods [46]. In 2019 Tankelevich introduced a deep feed-forward network capable of discerning the correct source clusters within scalp signals, representing a notable advancement in distributed dipole solutions [45]. Moreover, in a recent study, the authors embarked on an exploration of Convolutional Neural Networks (CNNs) to tackle the EEG inverse problem. This network was designed to identify multiple sources using training data adhering to biologically plausible constraints [22].

In this master's thesis, the primary objective is to investigate the feasibility of employing both a fully-connected neural network and a convolutional neural network for localizing current dipoles. This investigation extends to identifying the strength and radius of dipole clusters, relying on self-simulated EEG data generated with tools available through the LFPy software.

0.2 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. Electroencephalography (EEG) plays a pivotal role in recording and analyzing the electrical communication in the brain. Most importantly 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 aim 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, which will be usefull in chapter 2, where we will explore the principles of EEG recordings. By familiarizing ourselves with the fundamental concepts of neurons, we hope to gain a deeper understanding of the applications of EEG in the dynamic field of neuroscience. The chapter is based on the books "Neuronal Dynamics" by Gerstner, Kistler, Naud, and Paninski [17] and "Principles of Computational Modelling in Neuroscience" by Sterratt, Graham, Gillies, and Willshaw [42].

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 concentrations, along with the presence of ion channels that regulate the flow

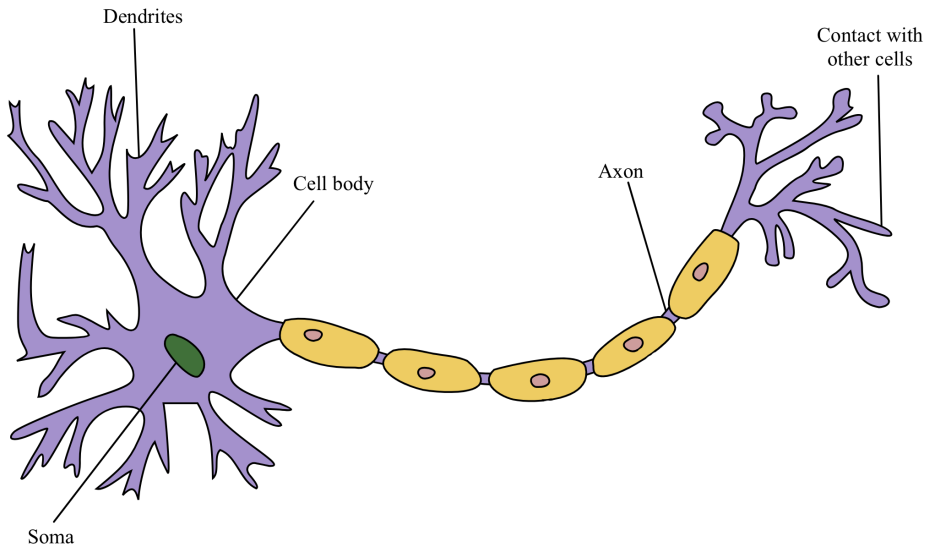

Figure 1.1: Illustation of single neuron with dendrites, soma and axon. The figure has been adapted from Wikimedia Commons with attribution to Quasar Jarosz at English Wikipedia and is licensed under the Creative Commons Attribution-Share Alike 3.0 Unported license [15]. The figure has been edited to remove some of the original details.

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 [42].

A neuron consists of three distinct parts: the *dendrites*, the *soma*, 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 non-linear 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. When the electrical current reaches the *synaptic cleft*, chemical messengers known as *neurotransmitters* are realised. These messengers play the key role in transmitting the signal to the next neuron. If *receptors* on the receiving neuron accept the neurotransmitters, a new electrical signal is generated. This transmission of signals between neurons the synapses is called *synaptic input* [17].

In Figure **??**, we have provided a basic illustration of a single neuron with dendrites, soma, and axon.

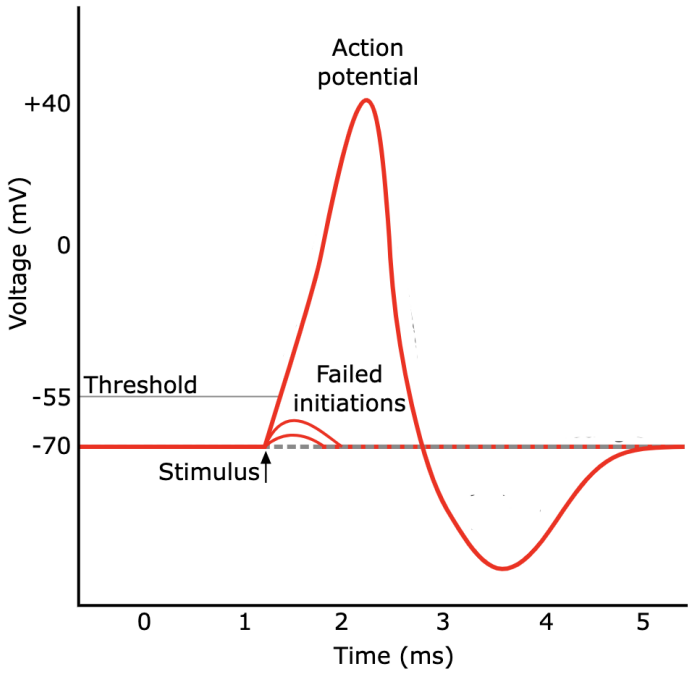

Figure 1.2: Illustation of a characteristic action potential. The figure have been adapted from Wikimedia Commons and are licensed under the Creative Commons Attribution-Share Alike 3.0 Unported license [10]. The figure has been edited to remove some of the original details.

1.2 Spike Trains and Action Potentials

Fix illustation so that it does not contain depolarization, repolarization and refractory period?

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, when observed in intracellular recordings, can be seen as a spike with an amplitude of about 100 mV and a duration of 1-2 ms [17]. In Figure 1.2, we have provided an illustration of the typical intracellular 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 back to the resting state. This form of the action potentials remains relatively 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 chains of action potentials emmitted by the neuron, also reffered to as *spike trains*.

Spike trains observed during epileptic seizures exhibit distinguishable characteristics compared to spike trains in regular neural activity. Epileptic seizures result from bilateral synchronous firing of neurons and manifest as specific EEG patterns known as *spike-and-wave discharges*. These discharges exhibit repetitive and rhythmic patterns, typically occurring at a frequency of around 2.5 Hz, setting them apart from the irregular and asynchronous firing of action potentials seen in typical spike trains[17]. During epileptic events, the initiation of spike-and-wave discharges involves complex mechanisms, including the interplay of voltage-gated sodium and calcium channels and the role of inhibitory postsynaptic potentials [47]. Wikipedia citation. See if I can use Torbjørn book instead.

1.3 Anatomy of the Cortex

Fill in from sources, paper etc. https://www.cell.com/current-biology/pdf/S0960-9822(11)01198-5.pdf, https://www.ncbi.nlm.nih.gov/books/NBK2510/ 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 [17].

The human cerebral cortex is typically divided into six layers. 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 [9].

Neurons communicate through synapses, where *presynaptic neurons* sends, and *postsynaptic cells* receives the electrical signals. 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 [17].

Within the cortex, there are two primary classes of neurons. *Pyramidal neurons* send information to distant areas of the brain, and play a crucial role in long-distance communication. On the other hand, *interneurons* are considered local-circuit neurons, exerting their influence on nearby neurons. Most pyramidal neurons form excitatory synapses, meaning they stimulate post-synaptic neurons, while most interneurons form inhibitory synapses, meaning they suppress the activity of pyramidal cells or other inhibitory neurons [9].

Chapter 2

Electroencephalograpy

Electroencephalography is a prominent recording technique employed to capture the electrical activity of the cerebral cortex. This invaluable tool has significantly enriched our understanding of neuronal interactions and the organizational complexity of the brain. As one of the most widely utilized non-invasive methods in both neuroscience research and clinical practice, EEG has played a pivotal role in studying brain activity during diverse cognitive processes, facilitating disease diagnosis, and assessing functional connectivity.

In this chapter, our primary objectives are to explore the physiological basis of EEG signaling, 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 work will investigate the possibilities of using simulated EEG measurements to train an artificial *neural network* for the purpose of localizing the sources generating these signals. Remeber to mention difference between a neural network and an artificial one.

2.1 The Physiological basis of EEG signaling

EEG signaling? 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 [47]. 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 and treatment planning.

https://www.electrical4u.com/eeg-measurement/, include one more sentence about the amplifier bank. The EEG technique involves the use of small metal disks, known as *electrodes*, strategically placed on the scalp to simultaneously record electrical charges resulting from neuronal activity. Typically, each electrode can capture synchronous signals originating from an area of approximately 6 sq. cm. on the cortical surface [9]. These recording electrodes are then connected to individual wires, which, in turn, link to channel connectors leading to a differential amplifier bank. Figure 2.1 provides an illustration of the typical EEG measurement setup.

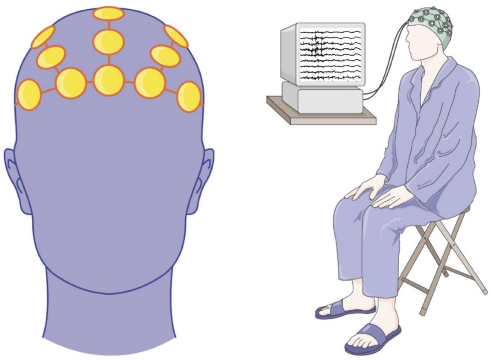

Figure 2.1: Illustration of the EEG method. The figures have been adapted from Wikimedia Commons with attribution to SMART-Servier Medical Art and are licensed under the Creative Commons Attribution-Share Alike 3.0 Unported license [2] [3]. The figures have been edited to remove some of the original details.

Electrical potentials generated by individual neurons are far too small to be picked up by the recording electrodes. Therefore EEG measurements primarily reflect the summation of synchronous activity from thousands of pyramidal neurons with similar spatial orientation. The activity originating from neurons with different geometric alignments cannot be picked up because their individual electrical signals tend to cancel each other out, resulting in undetectable waves at the scalp electrodes [9].

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 typically arises in EEG recordings of patients with epilepsy and includes spikes and sharp waves, that can be seen synchronously throughout the entire brain. In this context, spikes refer to hypersynchronized bursts from a sufficient number of neurons, arising from high-frequency bursts of action potentials [9].

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. Further in this chapter, we will explore the concept of the EEG inverse problem in greater detail and examine its implications for source localization. But first we will dive into the subject of head models for the purpose of simulating neural networks and EEG recordings.

Should propably come after

2.2 The Inverse Problem and Source Localization

In the field of neuroscience, the inverse problem involves deducing the underlying neural activity 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 [22]. Is it then possible to reach a loss equal to 0? Needs to be explained that without restictions this problem lacks uniqe solution. multipole dipoles can sup up etc... but we make restirctions and places where dipoles can be detected.

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

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

Here, Φ represents the vector of measured EEG signals at time t, $p(t)$ is the vector of a single cortical current dipole source 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, we 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:

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

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, methods such as neural networks are employed. Neural network can be designed to refine solutions to the inverse problem, providing robust and meaningful estimations of the neural sources responsible for the measured EEG data.

Before embarking on the utilization of artificial neural networks to solve the inverse problem, 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 use 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 real-world EEG recordings. head models do more than only account for the conductivity of different tissues and the geometry of the head.

2.3 Head Models

To accurately simulate EEG data and facilitate source localization, head models that precisely represent the conductivity distribution within the human head is essential. Head models serve as computational representations of the anatomical structure of the head, encompassing the brain, skull, cerebrospinal fluid, and scalp. They play a major role in the simulation of electrical signals originating from current dipoles, their propagation through various tissue compartments, and their impact on the recorded values at scalp electrodes.

Build up to the current dipole approximation? Next, Lead Fields?

Is this information needed? 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* [31]. 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 Model

The New York Head model (NYHM), developed by the Biomedical Engineering Department at the City University of New York, is a highly detailed computer model tailored for simulating electrical brain 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. Presenting a three-dimensional representation of the head and brain, coupled with precise information about tissue geometry and electrical properties, this head model will be utilized in our work within the context of EEG simulation to generate the data used for dipole source localization.

To precisely calculate EEG signals recorded at various scalp locations, the New York Head Model (NYHM) relies on a fundamental principle known as the *reciprocity theorem*. This theorem states that if you swap the positions of a dipole source and a measurement point, the measured value for the potential will remain the same. We calculate what is called the *lead field* by simulating a current dipole source at an electrode location and computing the resulting potential distribution at different points within the cortex. The lead field values are thus the EEG signals that would be measured at the electrode location when a source is positioned at one of the points in the cortex [31]. This turns out to be more efficient than simulating the EEG signals that would be measured at the electrode positions from the electrical signature of a single dipole within the cortex. The efficiency of this approach becomes evident due to the reduced computational demands when simulating EEG signals measured at recording electrodes. Specifically, simulating the propagation of the electrical field from a single dipole location, propagating through the densely distributed cortex and reaching the recording electrodes, requires tens of thousands of calculations, corresponding to the vast number of dipole locations within the cortex. In contrast, simulating the electrical field from the recording electrodes through the cortex requires calculations equal to the number of recording electrodes, which is significantly smaller, thus greatly simplifying the computational load.

This flipped approach is more efficient? Potential vs. electric field.

To construct the NYHM's lead field matrix (L), the researchers behind the model have performed these calculations for 231 electrode positions on the scalp and 70,000 possible dipole source locations within the cortex. Each location point includes the x-, y-, and z-coordinates, with the x-coordinate ranging from -72 to 72 mm, the y-coordinate spanning -106 to 73 mm, and the z-coordinate extending from -53 to 82 mm. This lead field matrix establishes the mathematical relationship between a current dipole moment

within the brain cortex and the resulting EEG signals. Each row of the matrix corresponds to the Mathematically, the lead field matrix is defined as:

$$L_{ij} = \frac{E_j^{(i)}}{I}, \quad \text{for } j = x, y, z. \tag{2.3}$$

Here, I represents the injected current at the electrode locations, and $\mathbf{E}^{(i)} = (E_x^{(i)}, E_y^{(i)}, E_z^{(i)})$ is the resulting electric field in the brain at . The row index i spans from 1 to $231 \cdot 70,000$. This use of the reciprocity principle greatly simplifies computations because it means that solutions to the forward problem are precomputed for every current dipole location. Consequently, the EEG signal Φ, which encapsulates the values measured at each of the recording electrodes resulting from a current dipole, can be calculated through a straightforward matrix multiplication:

$$\Phi = (\phi_0, \ldots, \phi_{70,000})\Phi = L\mathbf{p}, \tag{2.4}$$

where the current dipole moment $\mathbf{p} = (p_x, p_y, p_z)$ holds the dipole's magnitude in the x, y, z-directions respectively.

For further comprehensive details about the New York Head model, we refer readers to Huang, Parra, and Haufe (2016) [24].

2.4 The Current Dipole Approximation

By accurately simulating EEG data, non-linear optimization algorithms such as neural networks can be utilized for solving the EEG inverse problem. However, the simulation of intricate neuronal dynamics is a computationally expensive and complex task. Not correct. Head models are always based on dipole approximation, and so are lead field matrices. 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.

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 within neural tissue can give rise to the formation of current multipoles, the multipole expansion theorem offers a method to express the extracellular potential, denoted as $V_e(R)$, in terms of various pole contributions. In the context of EEG signals, employing this theorem reveals that $V_e(R)$ can be effectively approximated by a single dipole [18]. Wikipedia has also been used here. Check with book that everything is correct.

Neuronal multipoles depend on the spatial arrangement and symmetry of the charge distribution and result from the interplay of current sources

and sinks [16]. 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 source to the recording point surpasses the extent of the source, the applicability of multipole expansion becomes evident [25] is this correct?. This expansion are 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. Using the multipole expansion theorem, the 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. We note that 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 monopole and 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. Consequently, we are left with an approximation of the extracellular potential, V_e, that relies solely on the dipole contribution:

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

Here we have substituded for C_{dipole} in terms of other properties. Here 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. This condition is inevitably met in EEG studies, as the dipole length cannot exceed the depth of the cortex, while EEG electrodes are typically positioned several centimeters away from the cortical surface on the scalp [31].

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 Ness and Gaute Einevoll. This simulation aligns with the dipole approximation, as it clearly visualizes the distribution of electric charge in the extracellular potential of the neuron's surroundings, revealing distinct dipole patterns when observed from a greater distance.

2.5 The Effect of Dipole Orientation

Figure 2.3 and 2.4 are borrowed from work done by Tornjørn Ness and Gaute Einevoll, and illustrate the impact of dipole orientation on EEG outcomes. Figure 2.3 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 when the dipole is perpendicularly situated within a gyrus, as depicted in Figure 2.3B. 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 2.3C.

Further, Figure 2.4 depicts the EEG signals from identical dipoles positioned in various folding patterns of the cortical surface. These patterns align with the previous observations, as they showcases that the orientation of the current dipole moment significantly influences the EEG outcome. Firstly, Figure 2.4A and 2.4C provide an expanded illustration of the aforementioned scenarios, incorporating additional dipole moments located in a gyrus and a sulcus, respectively. In Figure 2.4B, where a collection of dipoles points randomly upwards and downwards, the EEG signal contribution appears to diminish significantly. Conversely, when the dipoles align in the depth direction of the cortex and are distributed across both gyrus and sulcus, we can expect an EEG contribution in between what we saw from Figure 2.4A and 2.4B, as depicted in Figure 2.4D. Lastly, Figure 2.4E demonstrates the minimal EEG contribution observed when the dipoles are divided between two opposing sulci.

2.6 Solving the EEG inverse problem

In this chapter, we have explored the fundamental concepts that underpin the investigation 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.

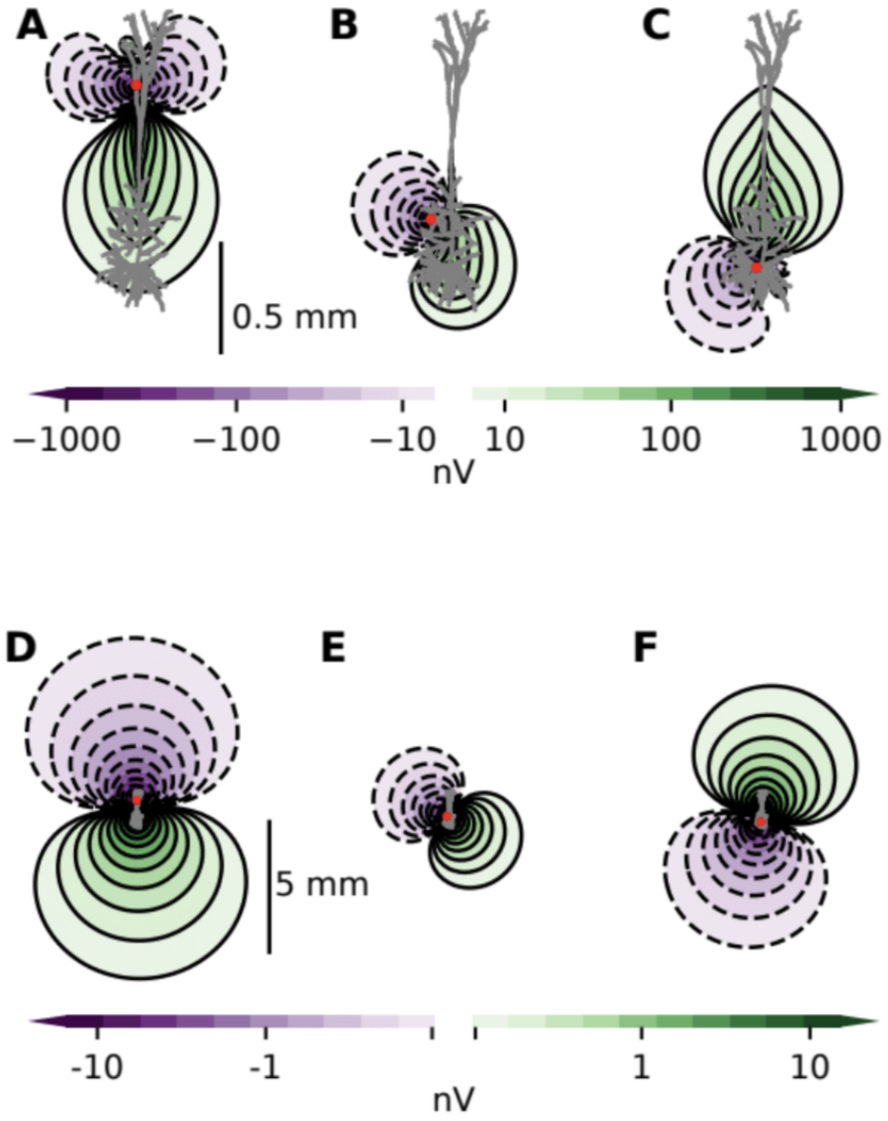

Figure 2.2: Simulation of extracellulat potential showing distinct dipole pattern. The figure has been provided from my supervisiors Torbjørn Ness and Gaute Einevoll.

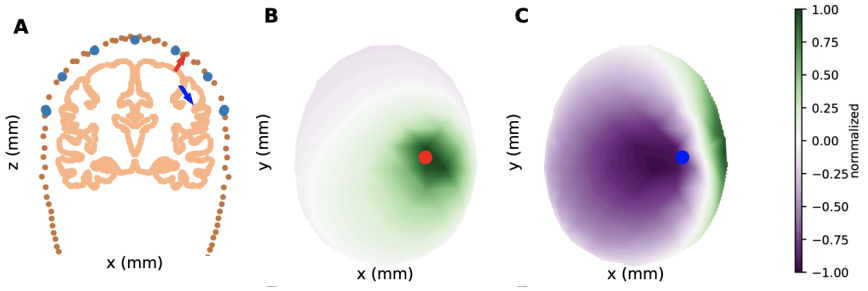

Figure 2.3: A: Two selected dipole locations in the New York head model: one in a gyrus (red) and one in a sulcus (blue). The head model is viewed from the side (x, z-plane). Close to the chosen cross-section plane, EEG electrode locations are marked in light blue. Available dipole locations near the cortical cross-section form an outline of the cortical sheet and are marked in pink. The current dipole moment for all cases was 10^7 nAμm. B: Interpolated color plot of EEG signal from the gyrus dipole, viewed from the top (x, y-plane). The plotted EEG signal is 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 [31].

In the next chapter, we delve into practical implementation by simulating EEG data. Utilizing the New York Head Model 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 EEG source localization challenge.

Chapter 4 explores the methods 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 the use of computational techniques, our aim is to bridge the gap between theoretical understanding and real-world applications.

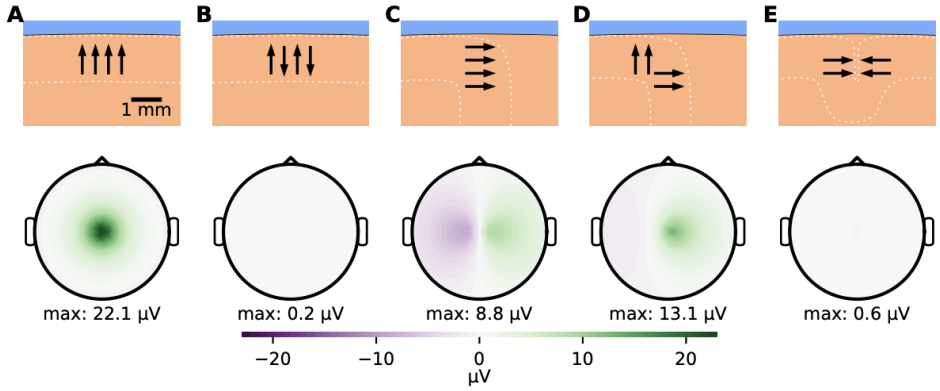

Figure 2.4: Different folding patterns of the cortical surface are represented by white dashed lines. EEG signals are calculated from four identical current dipoles with varying orientations. A: Dipoles aligned in the same direction within a gyrus. B: Dipoles pointing in opposite directions within a gyrus. C: Dipoles aligned in the same direction within a sulcus. D: Dipoles distributed between a gyrus and a sulcus, pointing towards the cortical surface. E: Dipoles divided between opposing sulci, pointing towards the cortical surface. 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 [31].

Chapter 3

Method: Creating EEG Data

In preparation for the application of neural networks to address the inverse problem, the acquisition of an appropriate EEG dataset is essential. This chapter focuses on utilizing the New York Head model in conjunction with the current dipole approximation to construct biophysically realistic EEG data. In Section 3.1, we will provide a detailed exploration of our EEG data simulation methods. Moving forward to Section , we will discuss the introduction of noise and its significance. Section 3.3 summarizes the features of the final dataset, which will be used to train models for solving the EEG inverse problem.

3.1 Simulation of EEG Signals from Single Dipoles

To create EEG data we use the New York Head model and its lead field matrix, which are integrated into the Python module LFPy 2.0 [21]. This software is a userfriendly wrapper around the NYHM, allowing for simulations of neural activity and the resulting EEG signals. Within LFPy, we use the `NYHeadModel` class to calculate EEG signals originating from a desired current dipole moment. For more information about the LFPy module, we refer the reader to Hagen, Næss, Ness and Einevoll (2018) [21].

The implementation of the lead field matrix in the `NYHeadModel` class in LFPy consists of 74,382 discrete points, each representing potential dipole source locations within the cortex. To sample a single data point we position a current dipole moment at one of the possible locations, and calulate its corresponding EEG signal according to the procedure outlined in Chapter 2. To begin with, we maintain uniform magnitudes for the dipole signals. By setting these magnitudes to 10^7 nAμm, the resulting EEG measurements span a range of approximately -1 to 1 μV. The NYHM is solved for 231 electrode possitions. One sample consequently holds the signal measured at each of the 231 electrodes at the scalp.

Kommer det tydelig nok frem hvorfor vi gjør dette? To ensure that the

dipole orientations are predominantly aligned with the depth direction of the cortex, as emphasized in Chapter 2, each dipole moment is rotated to be normal to the surface of the cerebral cortex. Note that aligning the dipole orientations perpendicularly to the cortex does not always lead to the dipole's normal vector pointing directly outward toward an EEG electrode. This variability arises due to the complex folding patterns found in the human cortex. In certain cases, when a dipole is located within a sulcus, the electrical signal it generates may follow the contour of the sulcus and enter deeper into the cortex. Consequently, the EEG signal must traverse a longer path before reaching an EEG electrode [31].

In the preceding chapter, we discussed how EEG analysis often revolves around the exploration of specific frequency bands. However, an alternative and widely recognized method for EEG investigation involves the practice of averaging EEG responses to specific stimuli across multiple trials [28]. These temporally aligned segments of EEG signals are commonly referred to as event-related potentials (ERPs). Given that noise and various EEG components are assumed to exhibit random variations across samples, the averaging procedure effectively reduces noise and extracts the event-related activity. In other words, this technique facilitates the identification of the specific time step within the EEG time series where the averaged signal attains its maximum magnitude, resulting in a 'static' EEG signal characterized by reduced noise.

The underlying principle for this approach is the quasi-static approximation, which posits that the potential measured from a dipole propagates instantaneously from the source to the electrode. This simplification is based on the assumption that the EEG signal at any point effectively captures a snapshot of the dipole activity within the brain. This means that if a dipole exhibits oscillations at a particular frequency, the corresponding EEG signal will oscillate at the same frequency, without phase shifts [37].

In our analysis, we sample the EEG data to obtain what we may consider an ERP, effectively representing a static snapshot of the electrode recordings of a current dipole source with magnitude 1 nAm. This alternative to the traditional time series data offers simplification and computational efficiency without significant deviation from standard clinical EEG analysis. This construction is further justified by the fact that our simulated data is noise-free, eliminating the risk of noise interference with the EEG samples. Consequenscly, we are left with time-locked, one-dimensional EEG signals mirroring the electrode recordings of dipoles located at different positioning within the cerebral cortex.

In our data collection, we sample the EEG data to obtain what we may consider an ERP, effectively representing a static snapshot of the electrode recordings of a current dipole source with a magnitude of 1 nAm. In contrast to clinical analyses of ERPs, we simulate one event per sample and do not hesitate to average over multiple events to collect the final signal. This

approach is justified by the fact that our simulated data is noise-free, eliminating the risk of noise interference with the EEG samples. This simulated ERP data can be viewed as an alternative to traditional time series data, offering simplification and computational efficiency without significant deviation from standard clinical EEG sampling. Consequently, we are left with time-locked, one-dimensional EEG signals mirroring the electrode recordings of electrical potentials generated by dipoles located at various positions within the cerebral cortex.

3.2 Noise

Real-world EEG recordings inevitably contain noise, which can disrupt the accurate analysis of brain activity. *Artifacts*, which are signals recorded by EEG but originating from sources other than neuronal communication, pose a particular challenge in real-world data. Some artifacts can mimic genuine epileptiform abnormalities or seizures, underscoring the importance of identifying and distinguishing them from true brain waves [40].

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, are electromagnetic interferences from external factors such as cable and body movements [8].

Filtering techniques are commonly employed to mitigate artifacts in EEG recordings before conducting analyses. It is important to note that while these techniques can significantly reduce noise, they may not completely eliminate all sources of interference. In the context of simulated EEG data, there is no inherent noise present. However, to align the simulated data with real-world EEG samples, controlled noise is intentionally introduced. This process ensures that the simulated EEG data closely resembles real measurements that have undergone appropriate filtering and preprocessing. Consequently, the data maintains a high signal-to-noise ratio (SNR) while incorporating a realistic amount of noise disturbance.

Utilizing realistic data is a critical step in enhancing the robustness of the trained neural network, thereby ensuring its effectiveness in handling real EEG recordings. However, the specific characteristics and quantity of noise have not been the primary focus of our study, and we have therefor adopted a straightforward approach in our methodology. 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. This introduced noise introduces random variations around each data point, all while preserving the overall statistical properties of the dataset.

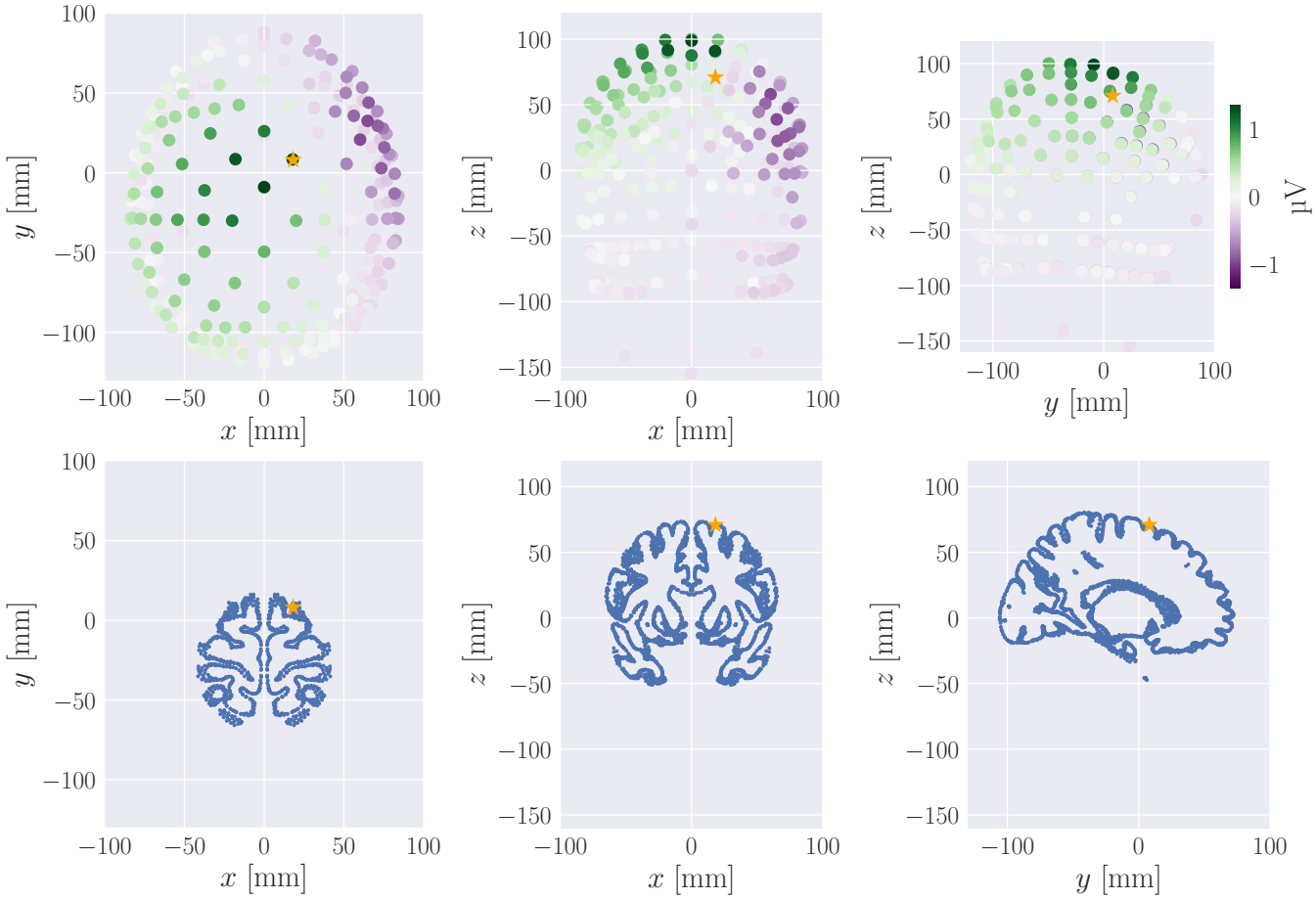

Figure 3.1: EEG measurements for a sample containing one single current dipole source at an arbitrary position within the celebral cortex. Noise is added to the EEG signal. The EEG measure is seen from both sides (xz-plane and yz-plane) and above (the xy-plane). EEG electrode locations are presented as filled circles, where the color of the fill represents the magnitude of the measured signal for the given electrode. The position of the current dipole moment is marked with a yellow star.

3.3 Final Dataset

Remove? The final dataset, $\mathcal{D}$, comprises 70,000 samples, where each sample holds 231 values representing the EEG measurements recorded at each electrode – a configuration directly inherited from the NYHM. Target values for the EEG samples are the x-, y- and z-coordinates of the different dipole sources.

Figure 3.1 presents an example of the input EEG data for a single sample, with added noise. The prominent dipolar pattern in the figure indicates that

the dipole is located within a sulcus. In the figure the EEG measure is visualized from multiple perspectives: the xz-plane, yz-plane, and the xy-plane. The electrode locations are represented by filled circles, with the color of the fill indicating the magnitude of the measured signal at each electrode. The position of the current dipole moment is denoted by a yellow star. As indicated by the colorbar in the figure, the EEG signal for the specific sample ranges from -1 to 1 μV, which is the range that the simulated EEG data for all samples will fall within.

Chapter 4

A Feed-Forward Neural Network Approach to Source Localization

Having obtained a suitable data set we can start building neural networks to address the EEG inverse problem. In this chapter, we provide a comprehensive overview of a feed-forward neural network built to map measured EEG signals to corresponding locations of current dipoles. We will be discussing its architecture and parameters.Introduce what the sections will be about.

4.1 Machine Learning

The field of machine learning is conserned with constructing computer programs that learn from experience. Within this computational discipline we find a spectrum of *tasks* and *experiences* these algorithms aim to understand and accomplish. machine learning tasks involve predicting how a system processes a given sample. The algorithms accomplish these tasks by gaining experience with sets of features, thereby extracting valuable insights into the inherent structure of these datasets.

For the purpose of solving the EEG inverse problem the model is concerned with the *task* of localizing the origins of abnormal electrical brain signals within the human cortex. The machine learning algorithm will gain *experience* through the analysis of corresponding EEG data associated with these abnormal sources.

find cite. Typically, machine learning problems are addressed using the same three elements. The first element is the dataset $\mathcal{D} = (\mathcal{X}, \mathcal{Y})$. $\mathcal{X}$, commonly is referred to as the *design matrix* and $\mathcal{Y}$ is the *target values* we aim the model to predict as function of $\mathcal{X}$. Next, we have the model $f(x; \boldsymbol{\theta})$ itself which is a function $f : \mathcal{X} \to \mathcal{Y}$ used to predict an output $y \in \mathcal{Y}$ from an input $x \in \mathcal{X}$ using the *model parameters* $\boldsymbol{\theta}$. Thirdly we have the cost funtion

$\mathcal{C}(y, f(x;\boldsymbol{\theta}))$ for $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, which allows us to evaluate how well the model performs on the samples in $\mathcal{D}$. Training a machine learning model involves the adjustment of the model parameters $\boldsymbol{\theta}$ to minimize the output of the cost function. The cost function yields high values when the model's output deviates from the desired target, and conversely, it produces smaller values when the predictions align closely with the intended outcomes.

4.2 Neural Networks

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 [23]. The models mimic the way biological neurons transmit signals, with interconnected *nodes* that communicate through mathematical functions. Each connection between nodes is represented by a *weight* variable, which encodes the strength of the communication between the two nodes. During training, the neural network "learns" by adjusting these weights to capture the underlying patterns and relationships in the data.

Change to to Feed Forward Neural Nets? We will consider fully connected feed-forward neural networks whose nodes are grouped into *layers* with the output of one layer serving as the input for the next. Typically, we call the first layer of neural networks the *input layer*, the final layer of the model *output layer* and the layers in between *hidden layers*. What lays in the term "fully connected feed-forward" is that every node within each layer sends their outputs to every node the next layer. Generally speaking, the outputs is only sent forward, through all of the N_l number of layers. Is a the input of the node? The activation of the i-th node in layer l, $a_i^{(l)}$ is the weigted sum

$$z_i^{(l)} = \sum_{j=1}^{N_{l-1}} w_{ij}^{(l)} a_j^{(l-1)} + b_i^{(l)} \tag{4.1}$$

fed through an *activation function*. Activation functions serve as mathematical functions that introduce nonlinearity into the network's computations. They play a critical role in enabling the model to capture non-linear patterns within the data. The bias parameter b, is a non-zero influence applied to the input of each node, which is then added to the product of the inputs and weights. In cases of zero activation weights or inputs, the presence of this bias parameter ensures that the activation of a neuron remains non-zero. This is a fortunate detail in neural networks, that facilitates the activation function's ability to perform non-linear mappings between inputs and outputs between nodes.

4.3 Building a Feed-Forward Neural Network

Did not test other architectures? The feed-forward neural network was one of the first artificial neural network to be adopted and is yet today a widly used algorithm within the domain of machine learning. As a natural starting point, we therefor adopted a fully connected, feed-forward neural network architecture for the purpose of solving the inverse problem. This approach eventually proved to be the most suitable framework among the ones tested. The fully connected feed-forward neural network is known as one of the simpler forms of neural networks, where information is processed in a uni-directional manner, flowing from the input nodes through the hidden layers and ultimately to the output nodes.

4.3.1 Architecture

The optimal number of hidden layers and nodes was meticulously determinated 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 nodes, 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, 64, and 32 nodes, respectively. Finally, the output layer holds an x-, y- and z-coordinate representing the predicted position of the desired dipole source. Figure 4.1 visualizes the construction of the fully connected neural network.

4.3.2 Activation Functions

Activation functions are fundamental components within the architecture of neural networks. Essentially, these functions introduce nonlinearity into the network's computations, thereby enabling the network to capture and model complex, nonlinear relationships within data [41]. This property is essential because many real-world phenomena and data patterns exhibit inherent nonlinearity. Without activation functions, neural networks would essentially be linear regression models, limiting their ability to predict non-linear relationships between inputs and outputs. In neural networks, activation functions is typically applied at every node within the hidden layers and at every output node in the final layer [38].

Drawing inspiration from the behavior of biological neurons, activation functions can be understood as decision-makers within the artificial neural

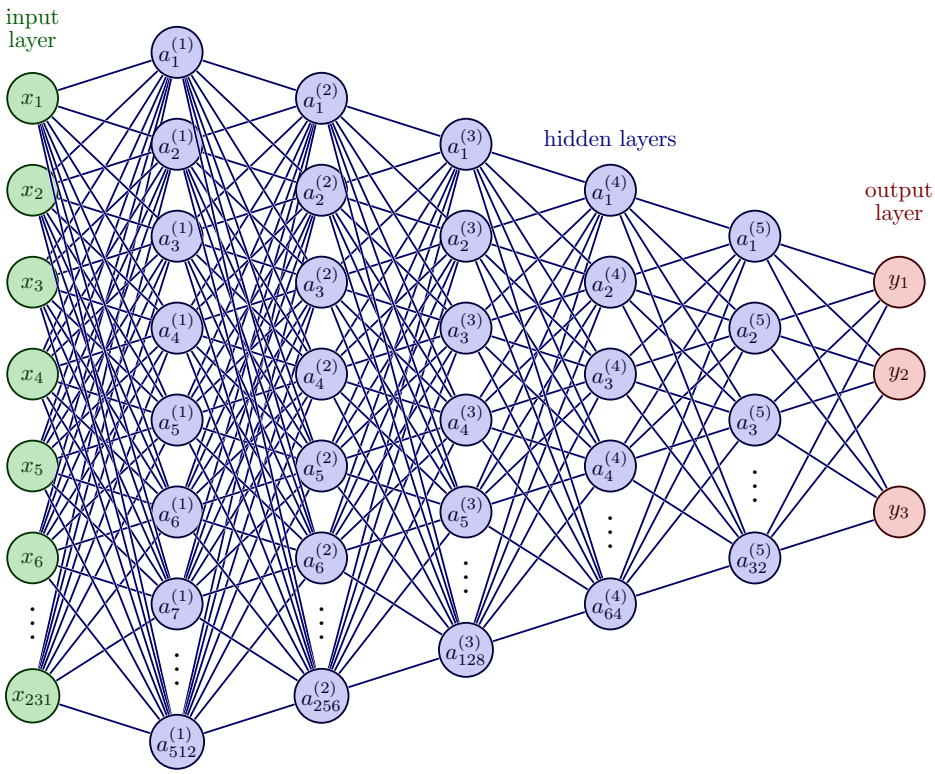

Figure 4.1: **Architecture of the fully connected Feed-Forward Neural Network**. The input layer comprises 231 electrode recording input values, and the network outputs predicted x-, y-, and z-coordinates of the dipole generating the EEG signal. The FCNN consists of five hidden layers.
The vizualization has been made with Latex, with code adapteded from Neural Networks in TikZ with attribution to Izaak Neutelings and is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License [32].

network, determining which information should be relayed to the next node, or [**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. 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"; otherwise, it remains inactive [5].

Rectified Linear Unit

Within the input layer of the FCNN 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:

$$\text{ReLU}(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{4.2}$$

In our network, the input nodes receive raw EEG input data. The nodes proceed to compute the product between their input and initialized weights, add their initialized bias, and then transform the result through the ReLU activation function. This activation function effectively maps the weighted sum onto a positive range, subsequently transmitting this transformed activation to the nodes in the next layer. The fundamental non-linear transformation introduced by the ReLU activation function is visually depicted in Figure 4.2. The widespread adoption of ReLU in neural networks can be attributed to its advantages in terms of computational speed, performance, and generalization capabilities [39]. Explain pros with 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 activation function compress input values into a range between -1 and 1, which helps prevent activations from becoming extremely large. Can i find source for this? By constraining the activations within this range, layers employing this function yield zero-centered outputs, which effectively centers the data and facilitates the learning process for subsequent layers.

Its mathematical representation is as follows:

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}. \tag{4.3}$$

As seen in Figure 4.3, 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.

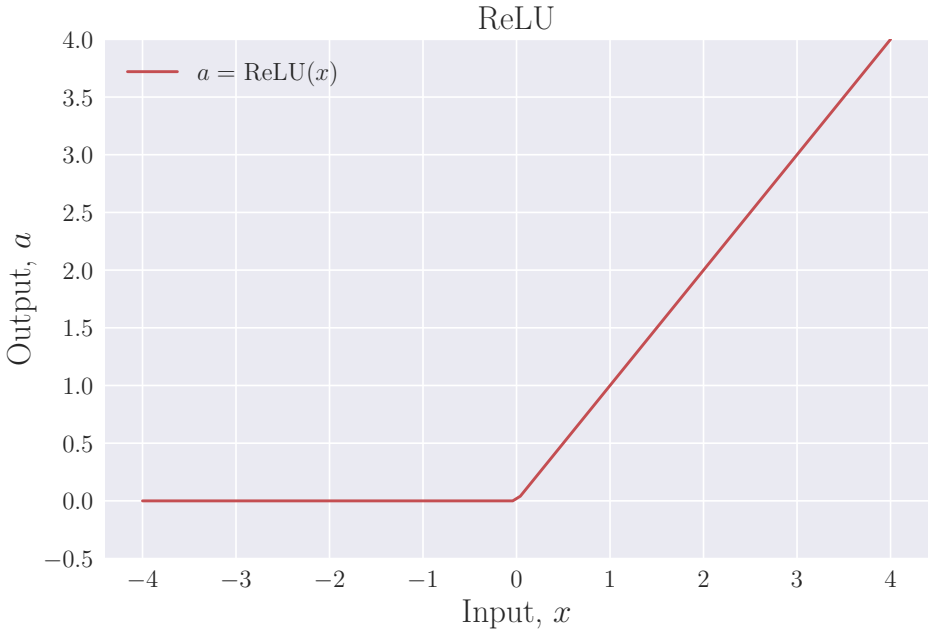

Figure 4.2: **Rectified Linear Unit (ReLU) Activation Function**: preserves positive input values (x) unchanged and enforces a transformation of negative input values to zero.

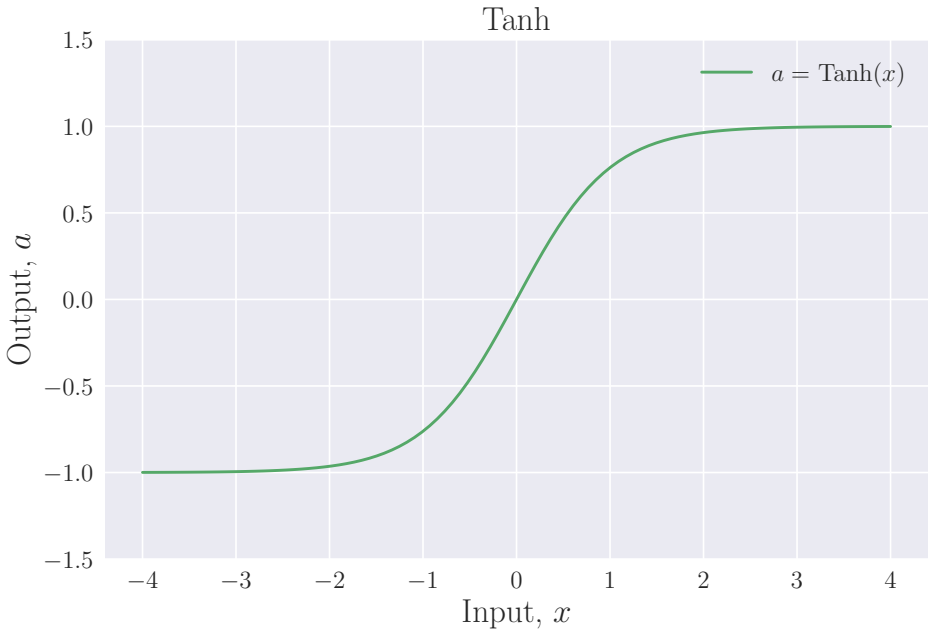

Figure 4.3: **Hyperbolic tangent activation function**: maps the input value x into a continuous range between -1 and 1.

Linear Activation

The FFNN is designed to provide direct and unconstrained predictions of the x-, y-, and z-positions for a desired dipole source. In our specific application, the target positions encompass a range of coordinate values, spanning from -70 to 70 mm for x, -58 to 78 mm for y, and -69 to 59 mm for z.

In the context of regression problems, such as the one we are addressing, it is it is common practice to use linear activations in the output layer. Linear activations in this context, are simply the absence of any activation function at all. Consequently, unlike the ReLU and tanh activations used in the earlier layers, the output nodes in our network do not utilize activation functions. Instead, the output weights and biases for each of the output neurons make the final adjustments to the input of the nodes in the final layer. These adjusted values are then considered as the predicted output values of the network.

Include info about exploding and vanishing gradient ?

4.3.3 Initialization of Parameters

Neural networks train by fine-tuning their parameters θ to capture and adapt to intricate patterns within data. Weights and biases serve as inner parameters of neural networks, crucial for modeling and optimizing these patterns. The total number of parameters is equal to the sum of all connections between layers plus the total number of biases in every layer. In the architecture provided in Figure 4.1, our FFNN comprises just under 300,000 parameters. Dette er så langt jeg har kommet. Må finne bedre kilder på initialization.

Initialization 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 the FCNN, 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:

$$W^{[l]} \sim \mathcal{N}(\mu = 0, \sigma^2 = \frac{1}{n^{[l-1]}}) \tag{4.4}$$

$$b^{[l]} = 0 \tag{4.5}$$

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 the FCNN, where both stability and efficient training are paramount, particularly when dealing with intricate EEG data patterns.

Chapter 5

Training the Fully Connected Feed-Forward Neural Network

This chapter will explore the training process of the fully connected feed-forward neural network (FCNN), outlining the decisions made during its training. Carefully choosing and adjusting elements of the training process ensures FCNN'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.

5.1 Training Methodology Overview

Having constructed the fully connected feed-forward 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 FCNN 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 FCNN, we will explore the following essential elements:

Data Preparation: We begin by preparing the data, which serves as the essential foundation for our machine learning model. This involves tasks such as data segmentation and standardization, ensuring the dataset is well-structured and representative.

Cost Function: Moving forward, we explore 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: The backpropagation algorithm plays an essential role in propagating error gradients through the network, refining the model's weights and biases. We explain the algortihm.

Optimization Algorithm: The choice of optimization algorithm plays an important role in training efficiency and convergence speed. We delve into the tequniqe of gradient decent and its implications in the context of the FCNN.

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: Finally we present the method of adaptive learning rates. We discuss how learning rates significantly influence training dynamics, and how the method for scheduling learning rates during training 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 [30].

In the case of the FCNN'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 the FCNN's performance and its capacity to effectively handle real-world scenarios with previously unseen data.

5.2.2 Data Scaling

Need some modification. Scaling is not done correctly in code. Scaling shoyld be done featurewise. Fix in code and write more detailed here. 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. This iterative process utilizes training data to fine-tune the model's internal representations, improving its predictive accuracy.

this need to be rewritten... Selecting the appropriate cost function is important when addressing machine learning challenges. In our regression task, where the objective is to predict x, y, and z coordinates of single current dipoles, we employ the commonly used Mean Squared Error (MSE) cost function, well known for its continuous measure of model performance during training. However, given that our aim is to minimize the Euclidean distance between predicted dipole localizations (x_i, y_i, z_i) and true values $(\tilde{x}_i, \tilde{y}_i, \tilde{z}_i)$, we aptly refer to MSE as the Mean Euclidean Distance (MED). In essence, when predicting Euclidean coordinates, MSE and MED are conceptually equivalent, as they both aim to minimize the squared differences

between predicted and true values, but MED explicitly emphasizes the Euclidean nature of the coordinates, providing a more accurate reflection of our optimization goal:

$$\mathrm{MED}(\boldsymbol{\theta}) = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}((x_i - \tilde{x}_i)^2 + (y_i - \tilde{y}_i)^2 + (z_i - \tilde{z}_i)^2)} \tag{5.2}$$

Here $\boldsymbol{\theta} = \theta_0, \theta_1, ..., \theta_n$ represents the model parametes, x_i, y_iandz_i denote for the predicted values and $\tilde{x}_i, \tilde{y}_i, \tilde{z}_i$ are the corresponding true values.

Due to the squaring step in the 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 [20].

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 [4], 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 [49]. Figure 5.1 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.

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.

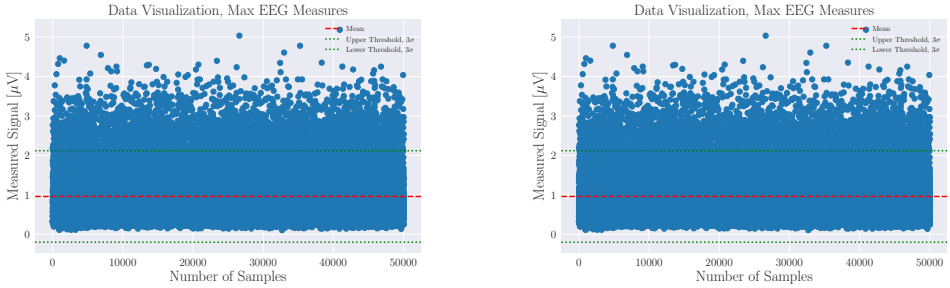

Figure 5.1: **Visualization of data set.** The panels visualize the maximum and minimum EEG measures for each sample within the data set. The plots need modification. Remove threshold etc...

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:**
   $w_{jk}^l = w_{jk}^l - \eta \delta_j^l a_k^{l-1}$
   $b_j^l = b_j^l - \eta \delta_j^l$

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. Check with cite.

5.5 Optimization Algorithm

Doublecheck citing. In order to minimize the cost function and find the optimal values for the model parameters, $\boldsymbol{\theta}$, an optimization alorithm is typically employed. One widely used optimization algorithm is *gradient descent*, which iteratively updates the parameters based on the negative gradient of the cost function [30].

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 $F((w)_n) \geq F((w)_{n+1})$ [48]. After each update, the gradient is recalculated for the updated weight vector $\mathbf{w}$, and the process is repeated [7]. 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 [48].

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 the FCNN [7].

Stochastic Gradient Descent

The Stochastic Gradient Descent (SGD) method, as applied in the context of training the FCNN, 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 [7]. 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 the FCNN, 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.

Add information about the value we are using!!

5.6 Regularization Techniques

Regularization techniques in machine learning help prevent overfitting and improve model interpretability. One common regularization technique is called the L2-norm penalty. In this technique, we add an extra term to our cost function that depends on the size of the model's weights. This added term ensures that the weights do not become excessively large while fitting the training data. By doing this, regularization reduces the risk of the model fitting the training data too closely, which can lead to overfitting. [23].

We can measure the size of the weights using the L2-norm, which results in our cost function taking this form:

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

Here, M represents nodes in a specific layer, K is the number of input features considered for each node and λ is the tuning parameter that controls how much we penalize the flexibility of our FCNN model. Here, flexibility means how much the model can adjust its coefficients to fit the data precisely. When $\lambda = 0$, the L2-norm has no effect, and the estimates produced will be equal to when the cost function is simply MSE. On the other hand, as λ approaches larger values, the impact of the shrinkage pentalty grows, and the coefficients estimates will approach smaller values. With smaller coefficients the model becomes less flexible, and the model will try to find simples pattern in the data, which in turn reduces the chances of overfitting [19].

5.7 Learning Rate Scheduling

The learning rate plays a critical role in the iterative Stochastic Gradient Descent procedure used to update model parameters during training, influencing the size of steps taken to minimize the cost function. To simplify the process of selecting an appropriate learning rate—neither too high nor too small—we employ a *learning rate scheduler*. This scheduler dynamically adjusts the learning rate during gradient descent optimization, enhancing both performance and training efficiency.

Various methods can be used to make the learning rate adaptive, but a common approach involves starting with a higher learning rate at the beginning of training. This allows for larger steps during the initial phases, speeding up progress. As training progresses and the model performance

approaches convergence, smaller steps are taken to fine-tune the parameters near the minimum, preventing overshooting. This strategy balances rapid initial convergence with meticulous fine-tuning later on [44].

For our model we have used the build-in PyTorch scheduler, ReduceLROnPlateau. This learning rate scheduler adjusts the learning rate when the FCNN's loss stops improving after a specified number of epochs. In other words, the learning rate remains unchanged as long as it enhances model performance but is reduced when results starts converging.

The scheduler offers several tunable arguments. We have set the *factor* to 0.2 in the FCNN, indicating the reduction factor for the learning rate when the loss plateaus. After the first update, the new learning rate becomes $\eta_{\text{new}} = 0.001 \cdot 0.2$, and this pattern continues for subsequent updates. The *patience* argument, set to 50, determines the number of epochs with no improvements before reducing the learning rate. The remaining arguments retain their default values and can be explored in greater detail, along with additional information about the scheduler, at https://hasty.ai/docs/mp-wiki/scheduler/reducelronplateau.

Chapter 6

Localizing Single Current Dipole Sources

Figure texts not fixed jet In this chapter, we will delve into the performance of the fully connected feed-forward neural network (FCNN) in solving the inverse problem. Our primary focus centers on investigating whether the accuracy of the neural network can be linked to the precise positioning and orientation of the current dipoles it is tasked with identifying or if the performance remains consistent across all brain regions and structures. Moreover, we shall employ a range of diverse error metrics to rigorously evaluate the performance of the network in this task.

6.1 EEG Sensitivity of Dipole Location and Orientation

According to Ness et al. (2021) [31], EEG signals demonstrate limited sensitivity to minor shifts in the precise location of neural current dipoles. This insensitivity can be attributed to the considerable separation between EEG electrodes and cortical neural sources compared to the dimensions of individual neurons and the thickness of the human cortex. Before evaluating the overall performance of our network, our investigation seeks to ascertain whether neighboring neural dipoles yield nearly indistinguishable EEG signals or exhibit noticeable distinctions. This exploration provides valuable insights into the network's ability to handle the complexity of the problem. If two dipoles located only 1 mm apart yield nearly identical results, it signifies a commendable performance by the network if it ables to distinguish at this scale. We emphasize that in this context, "neighboring" is defined based on the resolution of the NYHM. To assess the similarity in EEG signals between nearby dipoles, we employ the Pearson correlation coefficient.

The Pearson correlation coefficient is a statistical measure that quantifies the relationship between two variables. It is defined as the ratio of the

covariance of the two variables to the product of their standard deviations. The coefficient ranges from -1 to 1, with -1 denoting a perfect negative correlation and 1 indicating a perfect positive correlation. A coefficient of 0 signifies the absence of a linear relationship between the variables [34].

In Figure 6.1, we present EEG signals corresponding to dipoles located in close proximity within the New York Head Model's cortical matrix. The upper panel illustrates the two distinct EEG signals originating from the blue and green dipoles, while the lower panel shows the EEG signal corresponding to the blue and red dipoles. The blue central dipole is positioned between the green and red dipoles, located at the leftmost and rightmost positions in the NYHM cortical matrix, respectively. The Euclidean distance between the blue and green dipoles is 0.914 mm, while the blue and red dipoles are separated by 1.926 mm.

In Figure 6.1, the EEG signals of the blue and green dipoles exhibit significant overlap, with a high correlation coefficient of 0.966, indicating a strong positive relationship. In contrast, the EEG signals of the blue and red dipoles show less resemblance, with a correlation coefficient of 0.695, indicating a weaker linear relationship.

The observed variability in the correlation between the EEG signals of neighboring dipoles can be attributed to differences in the orientation of their respective dipole's normal vectors. The observed smaller value in the correlation matrix between the red and blue dipole, in contrast to the higher correlation between the blue and green dipole, is related to the more pronounced angular disparity in the normal vector alignment between the red dipole and the blue dipole. In contrast, there is a relatively smaller angular difference observed between the green dipole's normal vector and that of the blue dipole.

The distinct differences in the orientation of normal vectors arise from constraints imposed during data generation, which are designed to ensure that dipole orientations remain perpendicular to the complex folding of the cerebral cortex. As a result, neighboring dipoles within the NYHM possess distinct normal vectors, potentially leading to significantly different simulated EEG signals.

Furthermore, it is important to note that the separation between the red dipole and the blue dipole is somewhat greater than the distance between the blue and green dipoles. This variation in separation can be attributed to the specific construction of the NYHM cortex, as established by its developers.

In conclusion, these findings underscore the intricate nature of EEG signals, where the correlation analysis reveals varying degrees of similarity for neighboring dipoles. In some scenarios, it becomes evident that the linear relationship between EEG signals originating from distinct dipoles is relatively small, suggesting the network's capability to separate the EEG signals. In other instances, the EEG signals exhibit significant similarity, which may pose a challenge for the network in distinguishing them.

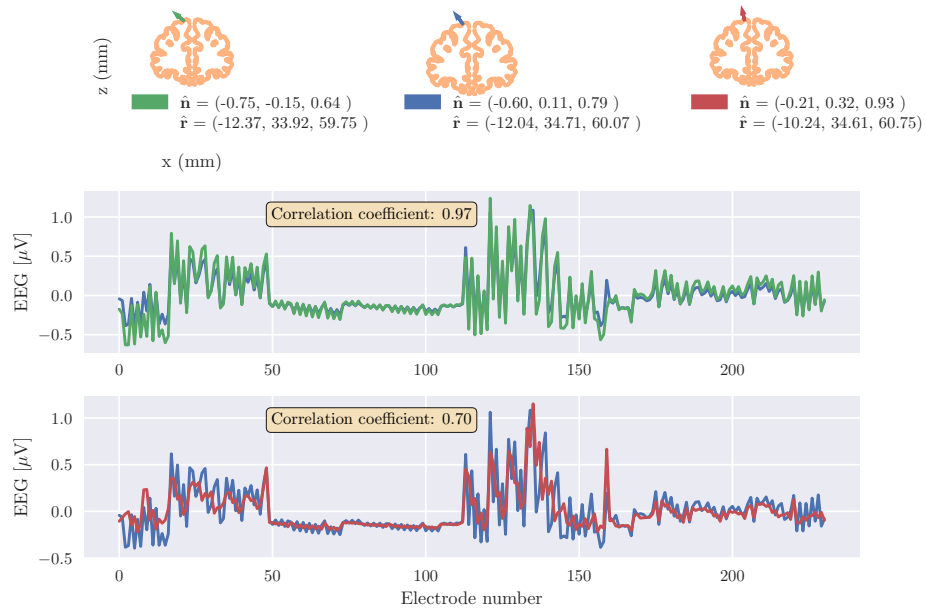

Figure 6.1: EEG signals plotted against electrode number for three neighbouring dipoles with normal vectors (-0.60, 0.11, 0.79), (-0.75, -0.15, 0.64), (-0.21, 0.32, 0.93) and positions (-12.04, 34.71, 60.07), (-12.37, 33.92, 59.75), (-10.24, 34.61, 60.75). The correlation coefficient between the blue and green dipole is 0.97, while it is 0.70 for the blue and red dipole. Which coordinate system? Where is (0,0,0) located?

6.2 Performance Evaluation

Move to training chapter? Based on the findings of Akalin Acar and Makeig 2013 [1] and Biasiucci et al. 2019 [6], source localization using spherical head models typically yields localization errors in the range of 10-30 mm. On the other hand, the utilization of modern subject-specific head models is expected to result in errors of less than 10 mm. The New York Head Model falls within the second category, and we will, therefore, let 10 mm serve as our guideline when evaluating the network's predictions.

Nevertheless, we emphasize that our primary objective is not necessarily the development of a flawless neural network for immediate clinical applications. Our aim is to ascertain whether a simple fully connected feed-forward neural network can effectively discern patterns and approximate dipole positions based on the simulated EEG data, collected using LFPy and the integrated Head Model, as outlined in Chapter 3.

Within this section, we will employ various techniques to assess the network's performance. This evaluation will include tracking the loss during

training, utilizing different sets of error metrics to calculate the average prediction error, and determining the distribution of test samples falling within predefined threshold values.

6.2.1 Results from Training and Validation Process

The FCNN underwent 500 epochs of training, and the convergence of the mean Euclidean distance (MED) is illustrated in Figure 6.2. The entire training process, encompassing data loading and preprocessing, took approximately 1.5 hours, with each epoch having an average training duration of about 11.00 seconds.

The validation loss exhibited a noticeable stabilization trend after approximately 350 epochs. At this point, the training could have been terminated. However, we chose to continue training for the full 500 epochs to confirm that no further improvements in validation data performance were attainable and to emphasize the network's full convergence.

Figure 6.2 visually demonstrates a clear reduction in loss, indicating the network's effective learning of patterns in the data. Validation loss stabilization becomes evident around the 350-epoch mark, while the training loss continues to decrease until it stabilizes between 400 and 500 epochs. This observation suggests that the model did not demonstrate signs of overfitting, as the validation loss remained constant, and the training loss ultimately stabilized. At epoch 500, the MED training loss is equal to 0.317 mm, while the MED validation loss measures 1.781 mm.

Furthermore, Figure 6.3 offers insight into the evolution of validation loss for the individual target coordinates—x, y, and z—as plotted against training epochs. This figure reaffirms that all three target coordinates were equally weighted, resulting in somewhat similar loss values for each. Additionally, it illustrates that the minor fluctuations in loss observed before 350 epochs disappear beyond this threshold, indicating a stable loss for all three target coordinates. This observation aligns with the trend of validation loss stabilization observed at approximately 350 epochs, as shown in Figure 6.2.

6.2.2 Testing Methodology and Accuracy

Move to training chapter? Upon achieving full training convergence, a neural network demonstrates its capacity to discern essential patterns within the training dataset, optimizing its parameters to minimize the cost function. To further assess the performance of the FCNN, our attention turns to an independent test dataset, comprising 20,000 samples. This dataset remains entirely shielded from the network during the training process, serving as an unbiased and objective gauge of the model's predictive accuracy and its ability to generalize effectively to novel, unseen data.

In order to comprehensively evaluate the neural network's performance

Figure 6.2: Training- and validation MED loss for the fully connected feed-forward (FCNN) neural network with 50,000 samples.

on the untouched test dataset and ensure the comparability of our results with other models, we employ a diverse set of error metrics. In addition to the primary metric of mean Euclidean distance used for assessing the accuracy of the network's predictions regarding the positions of current dipoles in the cerebral cortex, utilizing the metrics of Mean Absolute Error (MAE), Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) are utilized in order to evaluating the network's performance with respect to distinct target coordinates. This approach enables us to discern whether the network faces greater challenges in predicting one dimension over others or if it consistently maintains a uniform level of performance across all coordinate dimensions.

The MAE metric provides a direct measure of the magnitude of the difference between predicted coordinates (x, y, or z) and their true values. MAE offers a straightforward and easily interpretable indicator of the model's prediction accuracy for each dimension. It allows us to ascertain whether the network consistently delivers precise predictions for all target values or whether variations in accuracy manifest across different coordinate dimensions.

The MSE on its side introduces a squaring process that significantly penalizes larger errors. MSE measures the variance of residuals in squared units. While this feature emphasizes and addresses substantial error spread within the predictions, it does present values in squared units of the target values.

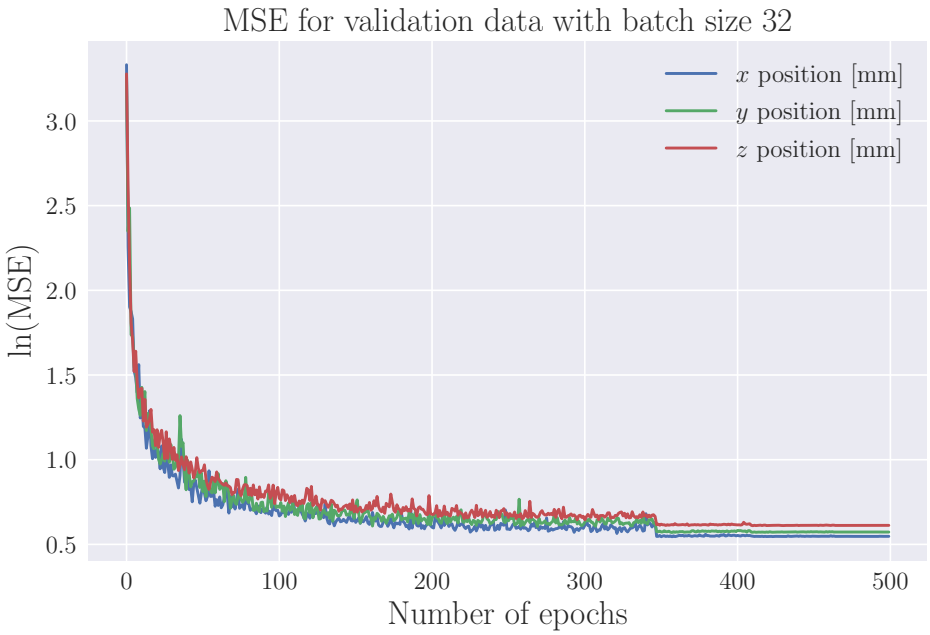

Figure 6.3: Validation MED loss for the separate target values; the x-, y-, z-coordinate.

This can pose a challenge when assessing individual coordinate components.

To mitigate the squared unit issue associated with MSE while still considering error spread, RMSE is included in the evaluation framework. RMSE measures the standard deviation of residuals, balancing the emphasis on significant errors while presenting results in units that align with the target coordinate values.

Both MSE and RMSE exhibit a higher sensitivity to larger errors, as they penalize substantial deviations more strongly when calculating the average squared differences between predicted and actual values. This contrasts with the MAE metric, which treats all errors, regardless of size, on an equal footing. Consequently, a high MSE or RMSE signifies a more extensive error spread, wherein predictions substantially deviate from the true values. Conversely, a lower MSE or RMSE indicates that prediction errors are less dispersed, and most predictions closely align with the true values. Smaller MSE and RMSE values signify a superior model fit and lower prediction error.

Position Spesific Performance

The MED between the network's predictions on the unseen test data and the true target values measures 1.33 mm. It is worth noting that this error is smaller than the one observed in the validation data, mainly due to the

Euclidian Distance for Test Samples		
ED <5 mm	ED <10 mm	ED <15 mm
99.735 %	99.995 %	100%

Table 6.1: **ED between targets and predictions of test samples; Predicting Single Dipole Location with a FFNN**
Performance of the network on the test dataset comprising 20,000 samples, presented as the percentage of samples falling within ED thresholds of 5 mm, 10 mm and 15 mm respectively.

absence of noise in the test data. This high level of precision is further exemplified by the statistics presented in Table 8.3. All predictions for the test samples exhibit an Euclidean distance of less than 15 mm, signifying a consistently high level of accuracy. Furthermore, an impressive 99.995% of the predictions fall within an Euclidean distance of less than 10 mm, and 97.753% of the predictions achieve an Euclidean distance of less than 5 mm. This numbers indicate a notably high degree of precision.

Table 9.2 displays the results for various error metrics corresponding to the network's predictions. Beginning with the MAE values for the x, y, and z-coordinates, we observe results ranging from 0.645 mm to 0.678 mm. These findings indicate that, on average, the network's predictions exhibit errors smaller than 1 mm in each coordinate.

The MSE values range from 0.747 mm^2 to 0.824 mm^2. The smaller MSE values signify high performance, with all values comfortably below a 1 mm^2 threshold. This finding indicate minimal error spread and few significant deviations between predicted and true coordinates.

RMSE values range from 0.864 mm to 0.908 mm. It is important to note that RMSE is slightly higher than the corresponding MSE values due to the square root operation involved. However, all RMSE values remain below the 1 mm threshold, signifying that the predictions maintain a limited error spread, typically staying within 1 millimeter of the actual values.

Notably, among the three coordinates, the z-coordinate exhibits the highest errors accross all metrics. This observation suggests that the network might encounter somewhat greater challenges in accurately predicting the z-coordinate of the dipole source. One plausible explanation is related to the nature of the inverse problem, where the depth of the dipole sources primarily influences the magnitude rather than the pattern of EEG recordings. However, we underscore that this discrepancy in error magnitude is very small and could also be influenced by randomness.

For an assessment of overall positional errors, the table holds the averaged Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) across the three spatial coordinates. The resulting error values reveal minimal deviations, yielding a total MAE of 0.662 mm, a

total MSE of 0.782 mm^2, and a total RMSE of 0.884 mm.

	Error Metrics for Target Values			
	x-coordinate [mm]	y-coordinate [mm]	z-coordinate [mm]	Position Error [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 6.2: **Evaluation of the FFNN performance utilizing different Error Metrics.**
FFNN performance on test dataset consisting of 20,000 samples. The errors are measured using Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE).

To further demonstrate the network's performance, we examine a randomly selected sample from the test dataset. For a dipole located at x = 66.5 mm, y = -26.4 mm, and z = 41.9 mm, the network predicts the coordinates to be $\tilde{x}$ = 66.9 mm, $\tilde{y}$ = -26.1 mm, and $\tilde{z}$ = 41.7 mm. The predicted values closely align with the true values, exhibiting a minimal error of 0.4 mm in the x-coordinate, 0.3 mm in the y-coordinate, and 0.2 mm in the z-coordinate, resulting in a total Euclidean distance of 0.54 mm. In Figure 6.4, we have illustrated the network's predicted position $\mathbf{r}$ of the dipole alongside the target position $\tilde{\mathbf{r}}$.

6.2.3 Performance at Different Brain Structures

Figure 6.5 presents the distribution of Euclidean Distances for various dipole locations within the New York head model cortex matrix. The figure offers valuable insights into the spread of errors across different regions of the cortex, with three cross-sections—front, top, and side—provided for examination. It is important to note that these cross-sections unavoidably 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 Euclidean Distance (ED) values presented in the panels are mostly 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 provide an opportunity to evaluate whether the network exhibits differential performance for dipoles located in a gyrus as opposed to those in a sulcus.

Initially, it might be assumed that EEG signals originating from dipoles in the sulcus present greater challenges for the network's analysis and predic-

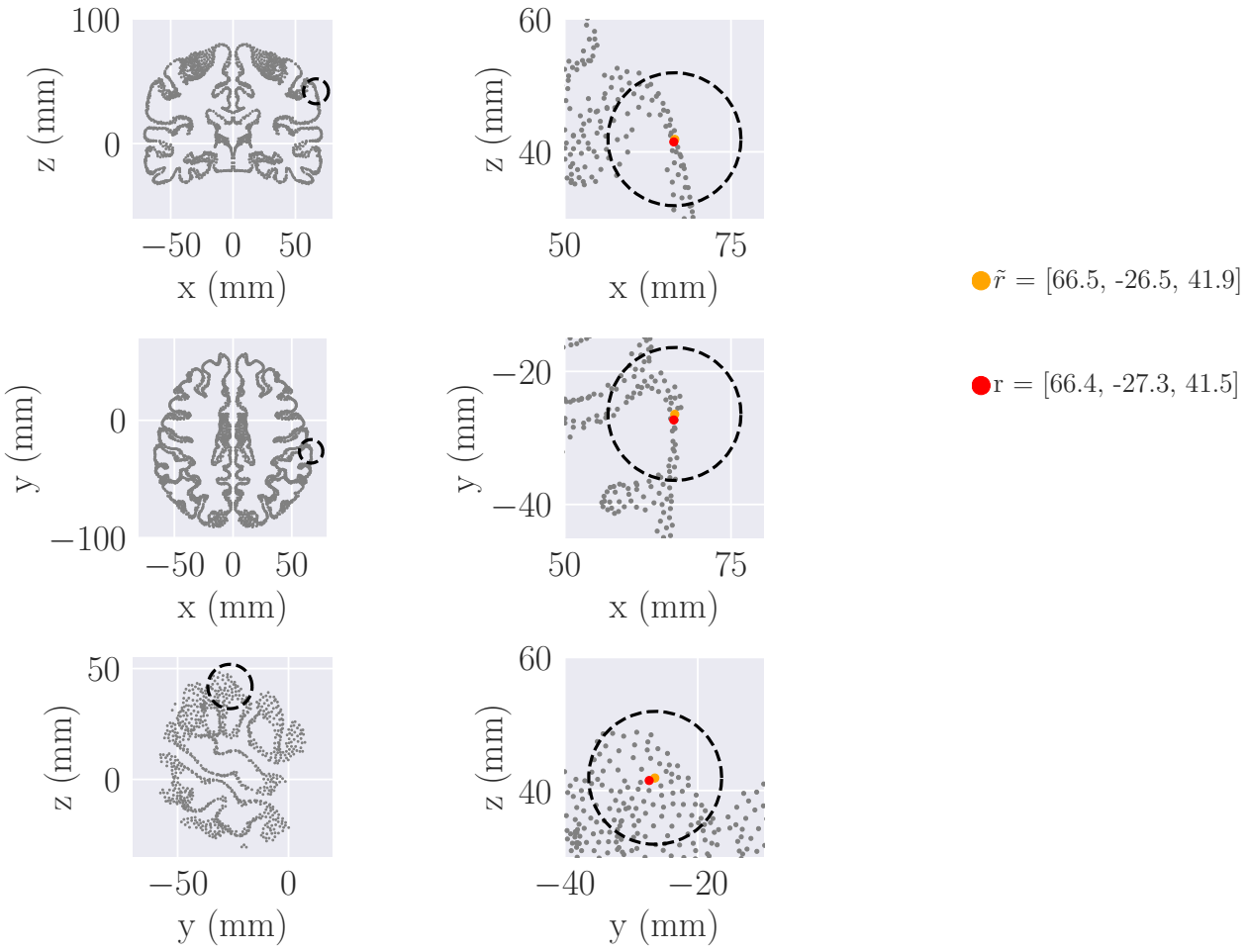

Figure 6.4: The networks predicted position r of a dipole within the test data set and the target position $\tilde{r}$. The target dipole is marked in orange and the predicted position is marked by a red dot.

tion. This assumption is based on the deeper placement of dipoles within a sulcus compared to those in a gyrus, as well as the potential complexities introduced by the dipole's orientation within the cortex. However, upon closer examination of Figure 6.5, it becomes evident that the distribution of ED values does not exhibit a clear correlation with the brain's structural characteristics. The ED values seem to vary across different regions, suggesting that the network's performance is not notably affected by the differentiation between gyri and sulci. Surprisingly, the Mean Euclidean Distance for all data points where dipoles are located in a sulcus is measured to be 1.68 mm, which is even smaller than for dipoles in a gyrus, where the Mean Squared Error measures 1.70 mm. This observation further challenges the initial assumption and indicates that the network demonstrates exceptional accuracy in predicting dipole locations, irrespective of their placement within the cortex.

Is this considered "theory" and should be mentioned earlier? FIX MORE
Further, the panels reveal a slightly higher concentration of data points with red marks in the deeper locations within the cortex, indicating higher ED values. This observation is consistent with the slightly higher error values observed for the z-coordinate, as presented in Table 9.2, and aligns with the theory related to the nature of the inverse problem. Deep brain structures often generate weaker EEG signals, making their precise detection challenging. Hence EEG patterns for dipole sources do not cause substantial changes in the pattern of electrical potential recordings; instead, they primarily influence the magnitude of the signals, which may potentially challenge the network's ability to discriminate deep sources from each other. Said with other words, the presence of higher ED values in deeper cortical regions might be attributed to the decreasing signal-to-noise ratio of EEG signals originating from these cortical areas. With that said, we emphasize that what we here reffere to higher, red ED values, are all smaller than 15 mm, and while being somewhat larget than the mean Eucledian distance Error, they can comfortably still be considered as small errors.

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

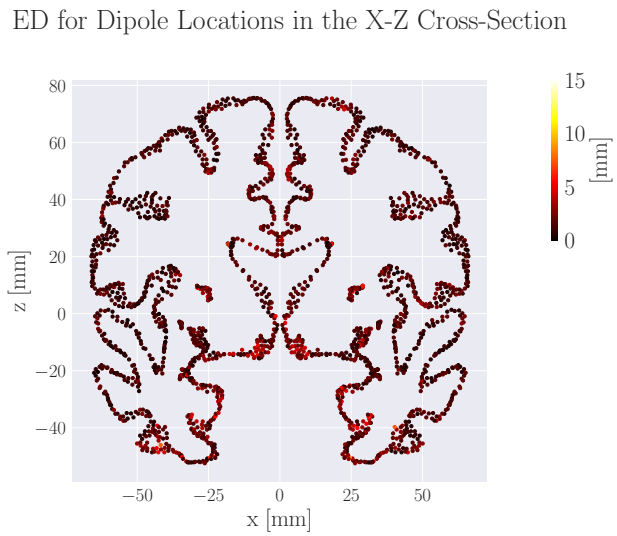

Figure 6.5: Different cross-sections of the cortex from the New York head model, seen from front, top and side. Each scatter point represents a possible position in the cortex matrix. The color of the fill in each circle indicates the Eucledian distance (ED) between the prediction and true value for the position of the specific dipole.

Chapter 7

Convolutional Neural Network Approach for Localizing Single Dipole Sources

In this chapter, 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 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 a neural network's ability to analyze EEG data and yield more accurate predictions for localizing the sources generating the neural signals.

Fix citing.

7.1 Adjusting the Data Set

Convolutional Neural Networks are well-known for their effectiveness in processing image data. Images share an intrinsic structure. Commonly they are stored as three-dimensional arrays; with an width axis, height axis and what is commonly reffered to as a *channel axis*. In image data, the number of channels a pixel has determines how many values are needed to describe its color. A standard RGB image holds three distinct channels used to specify the color of the pixel, while for a grayscale image, there is only one channel used to specify the brighness of that pixel.

In order to utilize CNNs for EEG data analysis, we perform an *interpolation* process on the original dataset presented in Chapter 3. Interpolation is a mathematical technique that estimates values between known data points, effectively filling in the gaps by providing approximations for unobserved data points. By applying interpolation to our EEG data, we create a representation resembling the structure of an image. This transformation converts

the original one-dimensional EEG data into a regular 2D grid format, imparting image-like characteristics while preserving spatial structures.

The resulting data is represented as a 20x20x1 array, mirroring the structure of a grayscale image. Within this array, each element stores the intensity value of the EEG potential recorded at a specific spatial location. Unlike traditional grayscale images where pixel values convey color intensity, in our context, each pixel's value signifies the intensity of the recorded EEG signal at the corresponding position. This representation may empower the CNN to leverage the spatial arrangement of the EEG data, uncover relevant patterns, and discern local relationships, similar to the way CNNs process conventional image data.

By preserving these spatial structures, we hope for the network to effectively utilize local relationships among neighboring recording electrodes. For instance, when one electrode records a high EEG value, the inherent data patterns suggest that nearby electrodes should also register relatively high EEG values due to their close spatial proximity. These spatial relationships may expedite the network's training compared to the simpler fully connected feed-forward neural network (FCNN), as it efficiently captures meaningful representations by harnessing these inherent patterns.

Figure 7.1 illustrates the process of interpolation. The right panel shows scatter plots of samples from the original data, where the EEG data is plotted at corresponding electrode locations at the scalp 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 is organized in an image-like manner.

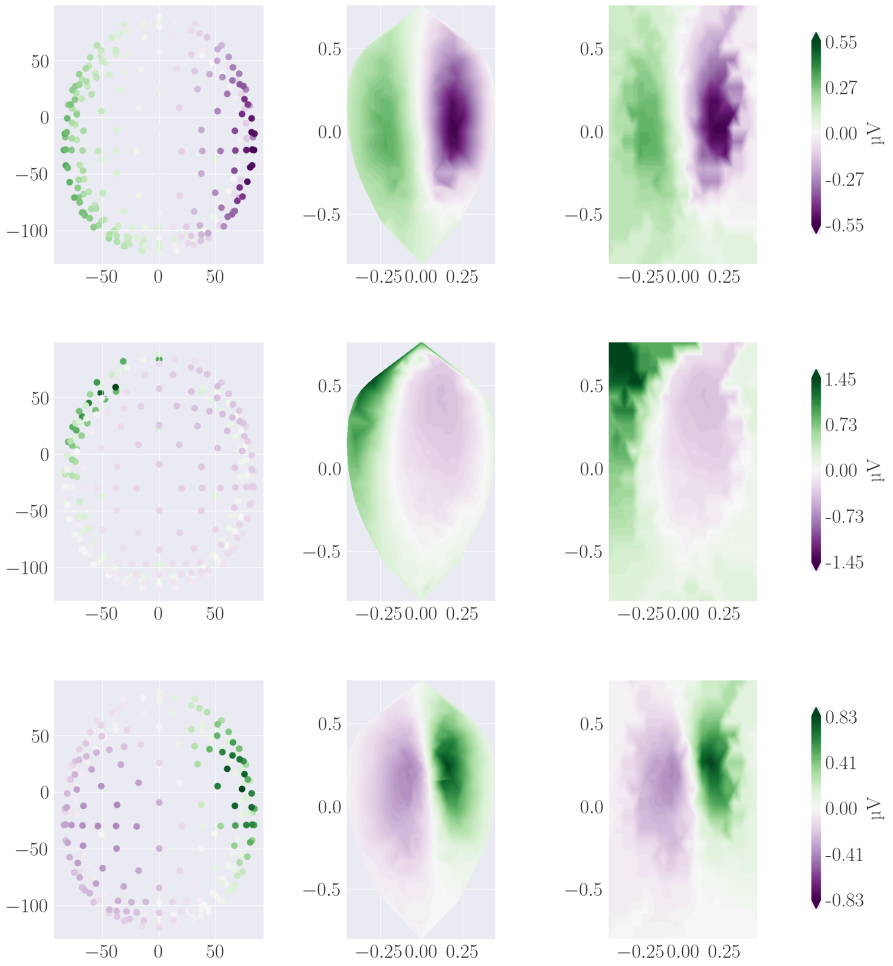

Figure 7.1:
Right Panel: EEG measures for three different samples, expressed in microvolts (μV). Each sample represents an EEG recording at specific electrode positions.
Middle and Left Panels: Illustration of the interpolation of the EEG data into a two-dimensional matrix. The interpolated data represents 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.

7.2 Layers in Convolutional Neural Networks

Convolutional Neural Networks draw inspiration from the information processing mechanisms in the visual cortex of the brain. In the visual cortex,

neurons respond selectively to stimuli within specific regions known as receptive fields. This intricate ability allows neurons to effectively capture spatial correlations found in natural images. In the realm of CNNs, this selective response is approximated mathematically using *convolutional operations*, copupled with *pooling layers* and *fully connected layers* [23].

The selective processing observed in the visual cortex is imitated in CNNs by establishing local connections between nodes in a convolutional layer and a subset of nodes from the preceding layer. This stands in contrast to fully connected feed-forward neural networks, where each node connects to every node in the preceding layer. This selective approach enables CNNs to specialize in learning and capturing localized features from input data, such as edges or textures.

A convolution is performed using a *filter*, often described as a small matrix with learnable parameters. Starting from the top-left corner of the input, a filter is systematically positioned over a specific region, depending on its size. At each location, one computes the element-wise product between the filter and the corresponding region of the input data, subsequently summing these products to yield a single value representing the convolution result at that position. After each computation, the filter shifts to the next position. Once the convolution operation is performed at every possible pixel location, the output elements together constitute the *activation map*. The convolutional processes is repeated for an optional number of filters, generating a new activation map for each filter. Given our EEG input data with dimensions of 20x20x1, using six 5x5 convolutional filters yields an output with dimensions of 16x16 and six channels. In essence, this process reduces the spatial dimensions of the image data while increasing its depth. The convolutional process is visually depicted in Figure 7.2.

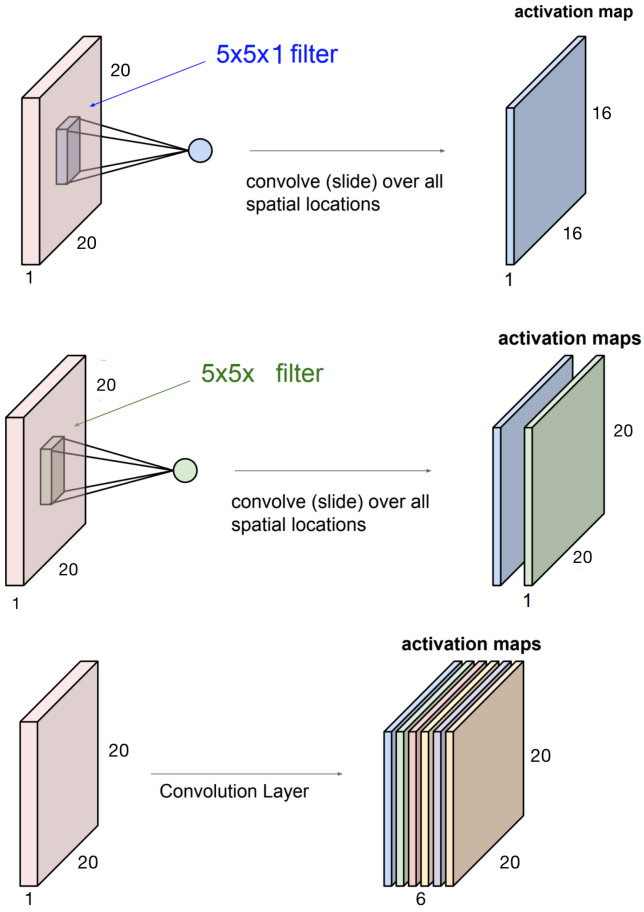

Figure 7.2: Illustration of the convolutional process. The figure has been adapted from Lecture Notes in the course IN5400 from 2021, with attribution to Andreas Kleppe [27].

Following each convolution operation, it is common practice to apply an activation function to each element of the output independently. In our network, we apply the ReLU activation function, as discussed in detail in Chapter 4. The mathematical formulation for each element in the output activation map follows the same structure as in FFNNs:

$$a = f(w^T x + b) \tag{7.1}$$

Here, w represents the learnable coefficients within the convolutional filter, x denotes the input, b stands for the bias, f is the activation function, and a symbolizes the output activation map. Importantly, the nature of w differs from the weights in FFNNs; it represents a local spatial region instead of the entire input. Furthermore, w is reused across all spatial regions,

functioning as a pattern detector applied in a *position-independent* manner. Consequently, convolutional layers entail considerably fewer trainable parameters compared to their fully connected counterparts, which are prevalent in FFNNs [26].

fix cite Following the convolutional layers commonly is the pooling layers. Pooling operations reduce the size of activation maps by the use of a max pooling function that in some sence works very much like a descrete convolution [13]. The max pooling function replaces the linear combination in the convolutional layers with a strided maximum filtering. This method simply chooses the maximum value inside the filter spatial region. By sliding a pooling filter of optional size with a specified *stride*, which determines how much the filter shifts over the input, the highest pattern detector response over the field of view is selected. In this way, spatial information is removed, and the dimension is further decreased.

7.3 Architecture, Hyperparameters and Training

The first layer in our constructed CNN, is a 2D convolutional layer. This layer takes an 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 6, signifying the extraction of 6 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 nodes, followed by 64 nodes. Lastly, we have the output layer with three output nodes, corresponding to the three coordinates of

the source generating the EEG signal. In Figure 7.3, we have provided an illusration of the architecture of the Convolutional neural network.

ReLU activation function is applied after each convolutional and pooling layer, as well as between the fully connected layers. This ReLU activation facilitates the network's ability to capture and propagate complex, non-linear relationships in the learned features. However, in the output layer, a linear transformation is employed without the use of an activation function in order to enable the neural network to provide direct and unconstrained predictions for the x-, y-, and z-coordinates of the desired dipole source.

Throughout the convolutional network, the weights of the fully connected layers are initialized using the Xavier normal distribution, a method that was also employed in the design of the simple FCNN, as described in Chapter 4. This initialization strategy contributes to the convergence and training stability of the network, ensuring it effectively learns and generalizes from the provided EEG data.

The training process for the CNN follows techniques similar to those applied for the simple FCNN, as extensively detailed in Chapter 5. To ensure effective learning and optimal predictions, we employ stochastic gradient descent with momentum as the optimizer, alongside a customized cost function based on the mean Euclidean distance. During training, mini-batches of size 32 are utilized to introduce data variability and prevent the network from converging to local minima. The CNN also maintains a learning rate of 0.001, weight decay of 0.5 and a momentum of 0.35, consistent with the settings used for the FCNN. To facilitate convergence, a learning rate scheduling approach is adopted, gradually reducing the learning rate during training to strike a balance between rapid initial convergence and fine-tuning in later stages. Following the training process, a rigorous evaluation of the CNN's performance is conducted on an independent test dataset, providing an unbiased assessment of its predictive accuracy and its ability to generalize to novel data.

Dele inn i flere subsections slik som i forrige kapittel?

7.4 Performance Evaluation

Figure 6.2 illustrates the convergence of the mean Euclidean distance loss. The neural network underwent 600 training epochs, with each training epoch lasting approximately 25 seconds. This resulted in a total training time of approximately 4 hours. It is important to note that this duration encompasses data loading and preprocessing steps. Notably, the CNN required longer epoch times compared to the previously studied FFNN. This difference can be attributed to the additional data preparation and interpolation required to suit the CNN's architecture. In contrast, the simpler FFNN had quicker epochs since its data was already in the desired format, with preprocessing

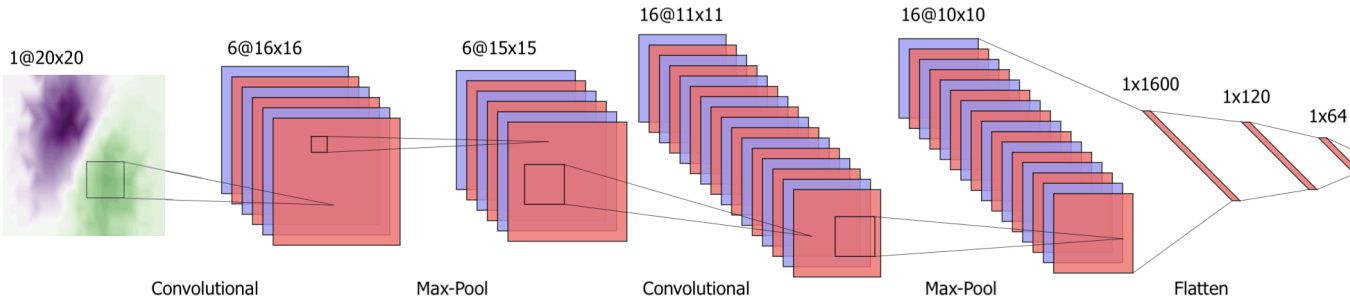

Figure 7.3: Architecture of the Convolutional Neural Network for localizing single current dipoles.
The visualization has been created using the NN-SVG tool, which was adapted from LeNail (2019) and is licensed under the Creative Commons Attribution 4.0 International License [29].

mainly involving data scaling and splitting.

Figure 7.4 reveals that both training and validation losses appeared to stabilize between 400 and 600 epochs. Prior to this phase, there were notficeable fluctuations in validation loss, which gradually subsided as the network approached convergence. The figure suggests that the model did not exhibit signs of overfitting, as both training and validation losses consistently decreased. At the final epoch, the training loss measures 1.460 mm, while the validation loss stabilizes at 2.280 mm.

Additionally, Figure 7.5 provides insights into the evolution of validation loss concerning individual target coordinates across training epochs. This figure offers information similar to what was observed for the FFNN, indicating that all individual target coordinates carried a roughly similar weight, resulting in comparable loss values for each coordinate. Furthermore, it demonstrates that the fluctuations in loss observed before the 400th epoch disappeared beyond this point, signifying stabilization in loss across all three target coordinates. Notably, the x-coordinate exhibited the lowest loss, followed by the y-coordinate, while the z-coordinate stabilized at the highest loss, consistent with observations from the FCNN. We emphasize, however, that these differences are small and may not be statistically significant in the context of our primary interest.

Add histogram instead of table? The mean Euclidean distance of the CNN's predictions on the unseen test data is measured at 1.80 mm. In Table 8.3, we have presented the percentages of samples falling within various Euclidean distance threshold values. Notably, the network exhibits outstanding predictive accuracy, with more than 99.993% of the samples within the test dataset having predictions accurate to within 15 mm. Furthermore, the net-

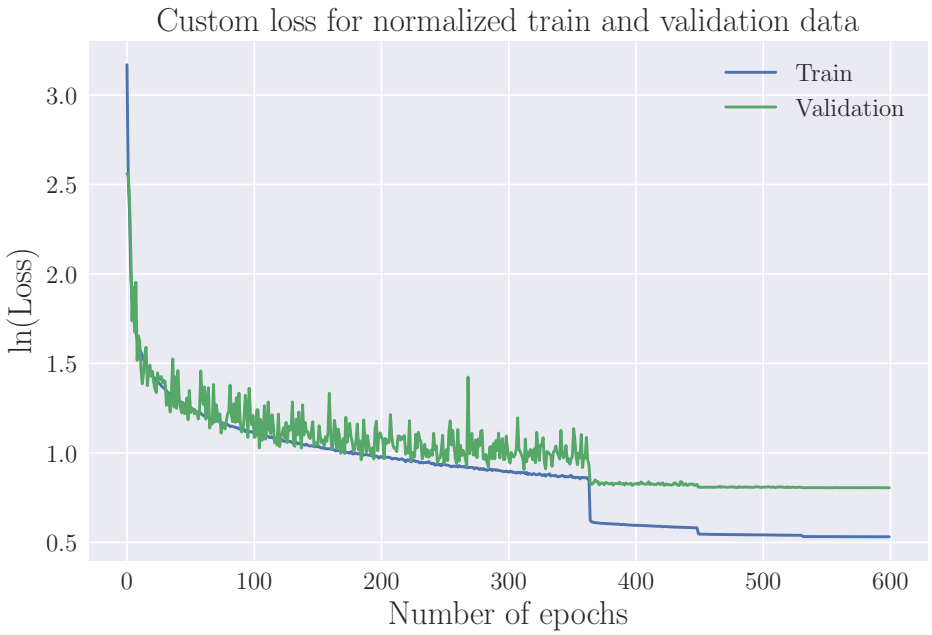

Figure 7.4: Training- and validation MED loss for the convolutional neural network with 50,000 samples and tanh as activation function.

work achieves a performance exceeding 99.815% for the 10 mm threshold Euclidean distance values. Approximately 98.030% of the predictions exhibit an Euclidean distance smaller than 5 mm, a notable achievement that places the CNN's performance just slightly behind the results obtained with the simple FFNN.

Euclidian Distance for Test Samples		
ED <5 mm	ED <10 mm	ED <15 mm
98.030 %	99.815 %	99.993%

Table 7.1: **ED between targets and predictions of test samples; Predicting Single Dipole Locations with a CNN**
Performance of the network on the test dataset comprising 20,000 samples, presented as the percentage of samples falling within ED thresholds of 5 mm, 10 mm and 15 mm respectively.

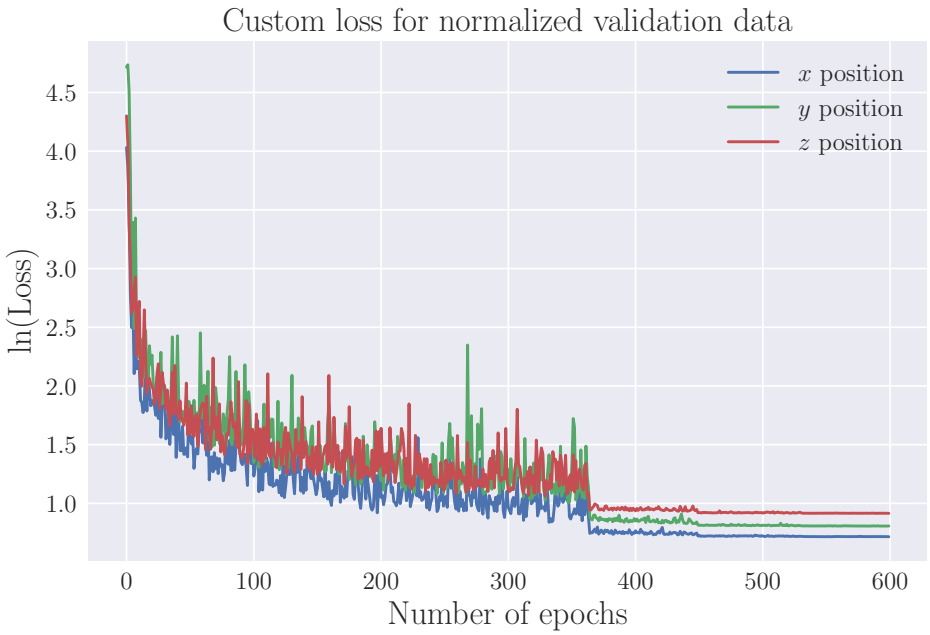

Figure 7.5: Training- and validation MED loss for the convolutional neural network with 50,000 samples and tanh as activation function.

Table 9.2 presents the results for various error metrics associated with the network's predictions. The MAE values for the x, y, and z coordinates fall within a narrow range, from 0.864 mm to 0.919 mm. These findings indicate that, on average, the network's predictions exhibit an error of less than 1 mm in each coordinate, a level of precision consistent with that achieved by the FFNN.

The MSE values, ranging from 1.579 mm^2 to 1.638 mm^2, are slightly higher than those observed with the FFNN. However, these values are still relatively small, signifying high performance. The marginally higher MSE values might indicate a slightly broader spread of errors compared to the previous FFNN. To further assess accuracy, we computed the RMSE values, which range from 1.257 mm to 1.280 mm.

Among the three coordinates, the z-coordinate exhibits the highest MAE. This observation suggests that the network faces greater challenges in accurately predicting the z-coordinate of the dipole source, as indicated earlier in Figure 7.5. However, the y coordinate holds the highest MSE, indicating slightly more significant deviations between predicted and true coordinates compared to the other target coordinates.

For an overall assessment of positional errors, the table provides the average values for MAE, MSE, and RMSE across the three spatial coordinates. These averaged error values remain small, reinforcing also the convolutional

neural network's precision in predicting dipole locations for the inverse problem.

	Error Metrics for Target Values			
	x-coordinate [mm]	y-coordinate [mm]	z-coordinate [mm]	Position Error [mm]
MAE	0.0.864	0.911	0.919	0.898
MSE	1.579	1.710	1.638	1.643
RMSE	1.257	1.308	1.280	1.282

Table 7.2: **Evaluation of the CNN's performance utializing different Error Metrics.**
CNN performance on test dataset consisting of 20,000 samples. The errors are measured using Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE).

To assess the network's performance, we examine a prediction for the same sample within the test set, as we did when evaluating the FFNN. For the dipole located at $\tilde{x} = 66.5$ mm, $\tilde{y} = -26.5$ mm, and $\tilde{z} = 41.9$ mm, the network predicts the coordinates to be $x = 66.4$ mm, $y = -27.3$ mm, and $z = 41.5$ mm. The predicted values closely align with the true values, showing an error of 0.1 mm in the x-coordinate, 0.8 mm in the y-coordinate, and 0.4 mm in the z-coordinate, resulting in a Euclidean distance of 0.99 mm. In Figure 7.6, we have plotted the predicted position $\mathbf{r}$ of the dipole alongside the target position $\tilde{\mathbf{r}}$.

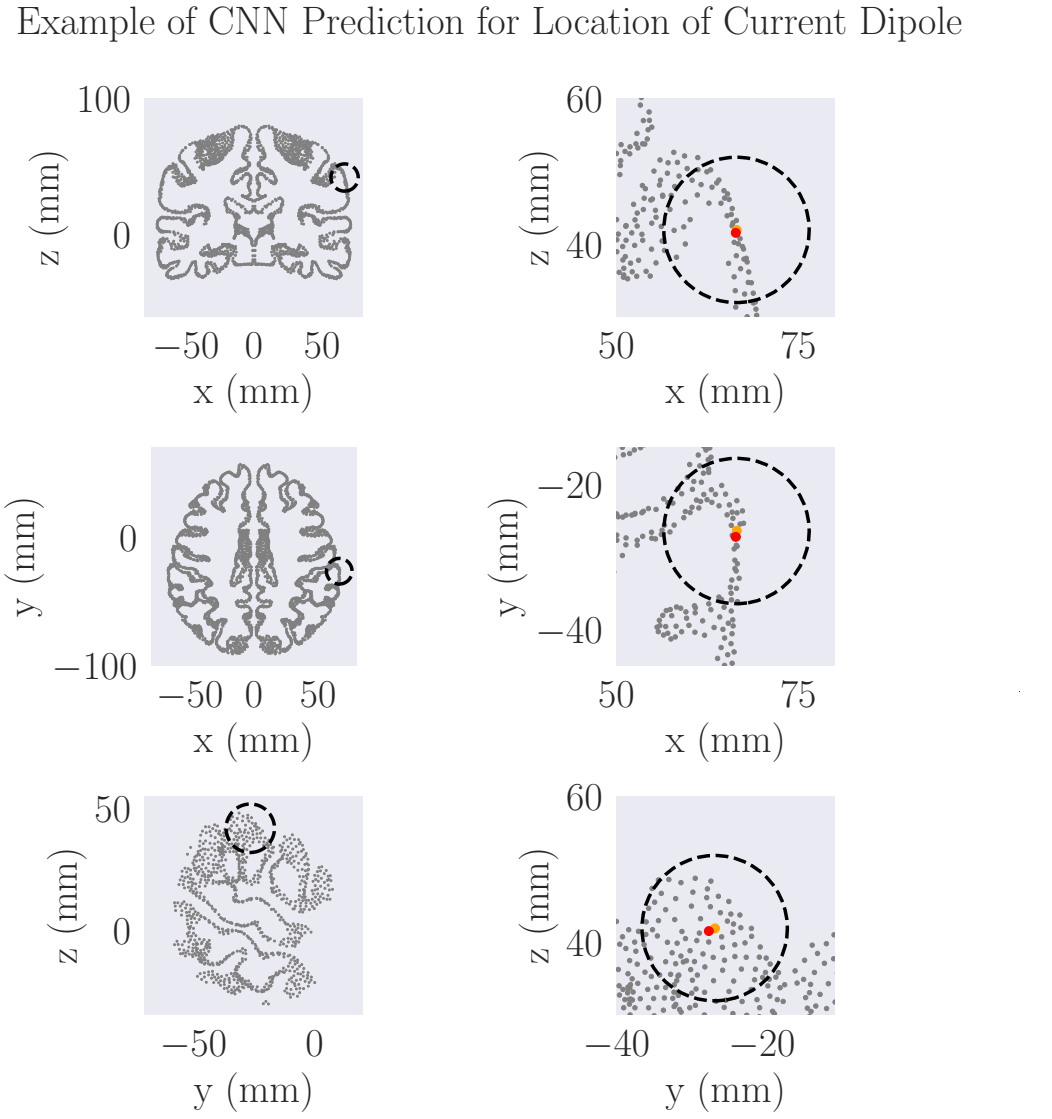

Figure 7.6: The CNN's predicted position r of a dipole within the test data set and the target position $\tilde{r}$ The target dipole is marked in orange and the predicted position is marked by a red dot.

Chapter 8

Localizing Two Current Dipole Sources

In this chapter we want to explore the FFNN's and the CNN's ability to predict the positions of *two* individual dipole sources located at different regions within the cortex, jointly contributing to the EEG signals recorded by the scalp electrodes. These extensions aim to comprehensively evaluate the networks' adaptability and generalization to more intricate problems that might be of interest in real-world applications.

8.1 Adjusting Data Set, Architecture and Hyperparameters

To begin, we generate EEG data to represent the electrical signals originating from two distinct dipoles, with positions $\mathbf{r_1}$ and $\mathbf{r_2}$, localized within the New York Head cortex. The locations of the dipoles are randomly selected from the available positions, allowing for the possibility of neighboring dipole placements within the cortex. Similar to our approach with single dipoles in the previous problem, we maintain equal magnitudes for each dipole. However, we increase the dipole strength to 5 nAm for each, resulting in a combined strength of 10 nAm. This modification leads to EEG measurements within a range of approximately -10 to 10 μV. The collection of EEG data is done for 70,000 samples, where 40,000 will serve as training data, 10,000 as validation data and 20,000 as test data, equal to the splitting procedure as done for the single dipole problem.

In Figure 8.1, we showcase a randomly selected sample from the adjusted dataset, illustrating the simulated EEG data generated by two dipoles. The feed-forward neural network (FFNN) directly employs this adjusted dataset, while the convolutional neural network (CNN) applies interpolation techniques to adapt the data to its architecture.

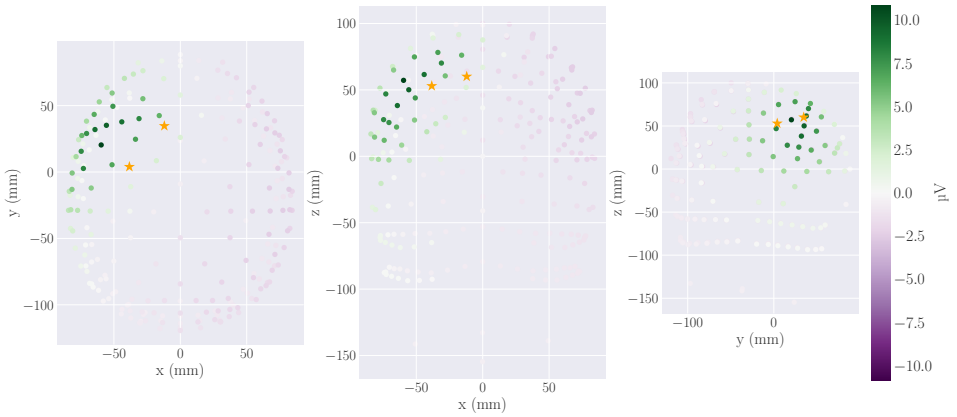

Figure 8.1: EEG data for a randomly selected sample from the dataset containing two dipole sources. These dipoles are positioned at random locations within the cerebral cortex. The EEG measurements are visualized from different perspectives: from above the skull in the xy-plane, from the front in the xz-plane, and from the side in the yz-plane. The EEG electrode locations are represented as filled circles, with the color fill indicating the amplitude of the measured signal at each electrode. The positions of the current dipole moments are marked with yellow stars.

In this more complex problem, the total number of target values has increased to six, covering the $x-$, $y-$, and $z-$coordinates for the location of each dipole. However, the network architecture still maintains its 231 input nodes. The number of hidden layers and nodes, the use of dropout techniques, network weight initialization, choice of activation functions, and other hyperparameters largely mirrors those employed in the single current dipole problem for both networks. However, one notable adjustment has been made to the weight decay value in the FFNN network. This hyperparameter has been fine-tuned to 0.1, as this change resulted in a slightly enhanced performance compared to the previous setting of 0.5 used in the single dipole problem.

8.2 Customized cost function

Our new objective is to build a model capable of accurately predicting the positions of not just *one* but *two* dipoles contributing to an EEG signal. To achieve this, we extend the mean Euclidean distance (MED) cost function used in the previous problem for localizing single dipoles. The extended

MED function is represented as follows:

$$\text{MED}(\mathbf{r_1}, \mathbf{r_2}) = \frac{1}{2n} \sum_{i=1}^{n} \sum_{j=1}^{m} \sqrt{(x_{i,j} - \tilde{x_{i,j}})^2 + (y_{i,j} - \tilde{y_{i,j}})^2 + (z_{i,j} - \tilde{z_{1,i}})^2}. \tag{8.1}$$

In this equation, the variable n denotes the total number of samples, while m represents the number of dipoles under consideration. However, given that our task involves localizing two distinct dipoles, our target vector encompasses two sets of coordinates representing the positions of these dipoles that we aim to predict. Consequently, the neural network also produces an output vector containing two sets of coordinates. Conventional cost functions, including the MED utilized for the single dipole problem, typically assume a one-to-one mapping between the first set of coordinates in the target and the first set of coordinates in the network's output. This rigid mapping assumes the order of these elements without flexibility. Such an approach may overlook the possibility that the other permutation of coordinates may be the correct mapping.

The primary task of our neural network is to discern patterns and learn from the input EEG data. Without explicit instructions, the network cannot inherently determine the correct order in which to arrange the coordinates of the two dipoles in its output vector (i.e., which should come first or second). Neglecting this potential permutation ambiguity could lead to suboptimal mappings, subsequently affecting the weight updates during loss calculation.

To overcome this challenge, we instruct the customized cost function to systematically explore all possible permutations of target and output vectors. This approach ensures that all valid combinations are considered, allowing us to determine which permutation yields the minimum loss. During each epoch, the network calculates the loss (Equation 8.1) for each permutation, ultimately selecting the permutation that results in the smallest loss. This process is repeated for every epoch, facilitating more precise weight updates and adaptation. To validate the intended functionality of the customized cost function, we have developed several unit tests, which can be found on our GitHub repository. This customized cost function is designed to accommodate an arbitrary number of dipoles.

8.3 Performance Evaluation using the FFNN

In Figure 8.2, we display the training and validation mean Euclidean distance loss as a function of training epochs. Our network underwent 800 training epochs, with each lasting approximately 27 seconds. It is noteworthy that the validation loss ceased to improve significantly between 200 and 300 epochs, suggesting that training could have been terminated before reaching 800 epochs. The training loss also plateaued after 400 epochs,

demonstrating no signs of overfitting. Both training and validation losses exhibited considerable fluctuations until reaching epoch 190. At epoch 191, the learning rate was reduced to 0.0002, resulting in a significant decline of the validation loss and eliminating further fluctuations. The validation loss eventually stabilized at a value of 18.71 mm by epoch 800, starting from an initial loss of 105.90 mm. We emphasize that this loss represents the mean Euclidean distance for two dipoles. If we want to calculate the mean Euclidean distance for a single dipole alone, we would need to divide this measure by 2, as we will be doing in the results presented from here.

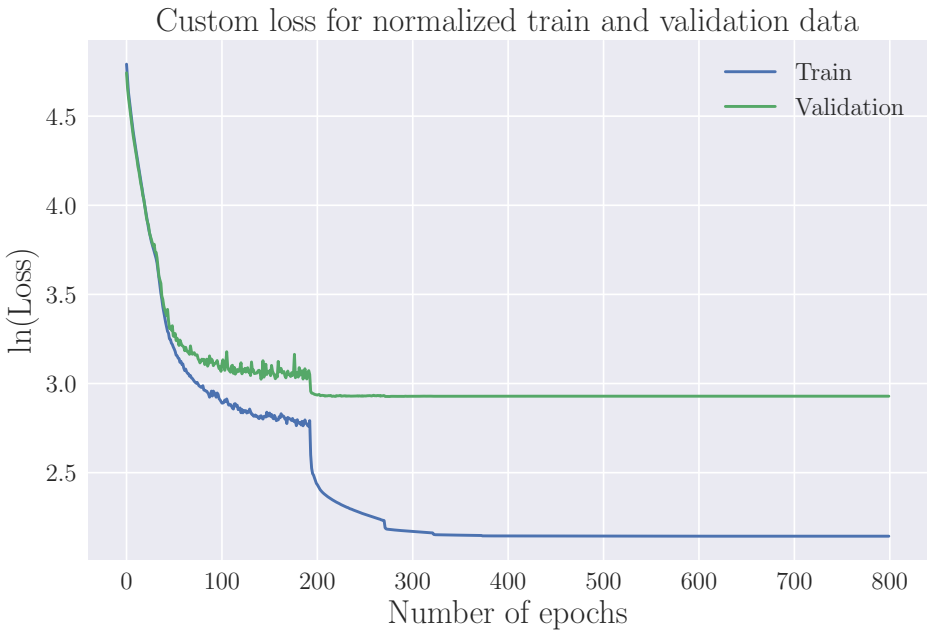

Figure 8.2: Training- and validation loss for the FFNN, predicting two current dipole sources, as function of epochs.

Bør jeg skrive med 3 desimaler eller runde av slik som jeg har gjort her? The mean Euclidean distance between predicted and target dipole locations within the EEG test data originating from two current dipoles is 17.41 mm. This indicates that the average error for a single dipole's position is 8.71 mm. Table 8.3 provides an evaluation of the model's accuracy concerning the Euclidean distance (ED) (averaged over the two dipoles in each sample) between predicted and target dipole positions within the test dataset. The FFNN demonstrates an accuracy of less than 15 mm for 90.85% of the samples in the test dataset. However, as the ED threshold becomes stricter at 10 mm and further at 5 mm, the accuracy decreases to 73.06% and 19.00%, respectively. To visualize the distribution of localization errors across the dataset, Figure 8.6 presents a histogram of the ED between predicted and

Euclidian Distance for Test Samples		
ED <5 mm	ED <10 mm	ED <15 mm
18.995 %	73.055 %	90.850 %

Table 8.1: **ED between targets and predictions of test samples; Predicting Two Dipole Locations with the FFNN**
Performance of the FFNN on the test dataset comprising 20,000 samples, presented as the percentage of samples falling within ED thresholds of 5 mm, 10 mm and 15 mm respectively.

target positions for 20,000 test samples. The data is grouped into bins with a width of 3 mm. Each bin represents the number of samples with prediction errors falling within the specified interval. This histogram reveals that the majority of predictions exhibit errors smaller than 15 mm. Nonetheless, approximately 10% of the predictions fall outside this threshold.

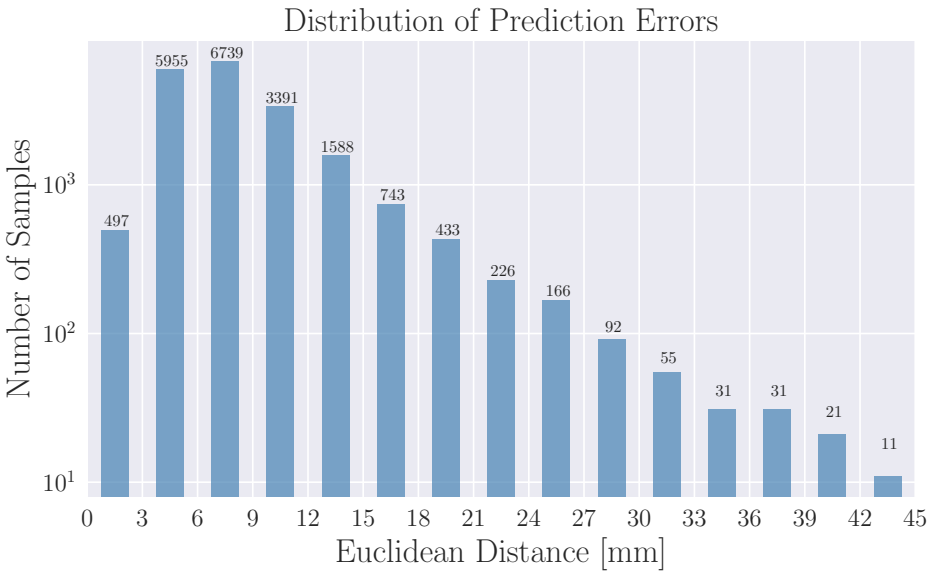

Figure 8.3: Histrogram illustrating the Euclidean Distance for predicted dipole locations, organized into bins of width 3 mm. The ED thresholds are organized into bins of width 3 mm. Each bin holds the number of samples with prediction errors that falls within the given interval.

Table 8.4 offers further evaluation of the model's performance using various error metrics. The MAE for the x-, y-, and z-coordinates of the dipole positions are all below 5 mm, indicating precise predictions in these dimensions. However, the MSE values, ranging from 36.98 mm^2 to 40.83 mm^2, are considerable higher than those observed in the single dipole case, signifying

increased error spread in this more complex scenario. RMSE averages at 6.243 mm, further confirming the model's higher error spred.

Once again, we observe slightly higher errors in all metrics when predicting the depth of the dipole source, suggesting that accurately predicting the z-coordinate remains a somewhat greater challenge for the FFNN in this multi-dipole scenario.

	Error Metrics for Target Values			
	x-coordinate [mm]	y-coordinate [mm]	z-coordinate [mm]	Position Error [mm]
MAE	4.156	4.379	4.485	4.340
MSE	36.977	39.122	40.834	38.978
RMSE	6.081	6.255	6.390	6.243

Table 8.2: **FFNN: Evaluation of the network's performance utializing different Error Metrics.**
FFNN performance on test dataset consisting of 20,000 samples. The errors are measured using Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE).

For illustrative purposes, we examine a specific prediction from our model. Consider a sample containing two dipoles located at $\tilde{\mathbf{r_1}}$ = (52.53, -16.25, 8.60) and $\tilde{\mathbf{r_2}}$ = (-63.48, -45.06, 22.67) in units of mm. Our model predicts the positions $\mathbf{r_1}$ = (55.95, -17.07, 13.43) and $\mathbf{r_2}$ = (-64.09, -48.73, 19.89) in units of mm, respectively. This results in an Euclidean distance of approximately 5.98 mm between $\tilde{\mathbf{r_1}}$ and r_1, and approximately 4.65 mm between $\tilde{\mathbf{r_2}}$ and $\mathbf{r_2}$. Although this prediction exhibits a slightly larger localization error compared to the single-dipole scenario, it still falls below the 1 cm threshold we aim for. It is however important to note that this example represents one of the better-performing predictions, as only 19% of the test dataset's predictions falls below a precision of 5 mm. In Figure 8.4, we visualize the predicted positions ($\mathbf{r_1}$ and $\mathbf{r_2}$) of the dipoles alongside the target positions ($\tilde{\mathbf{r_1}}$ and $\tilde{\mathbf{r_2}}$).

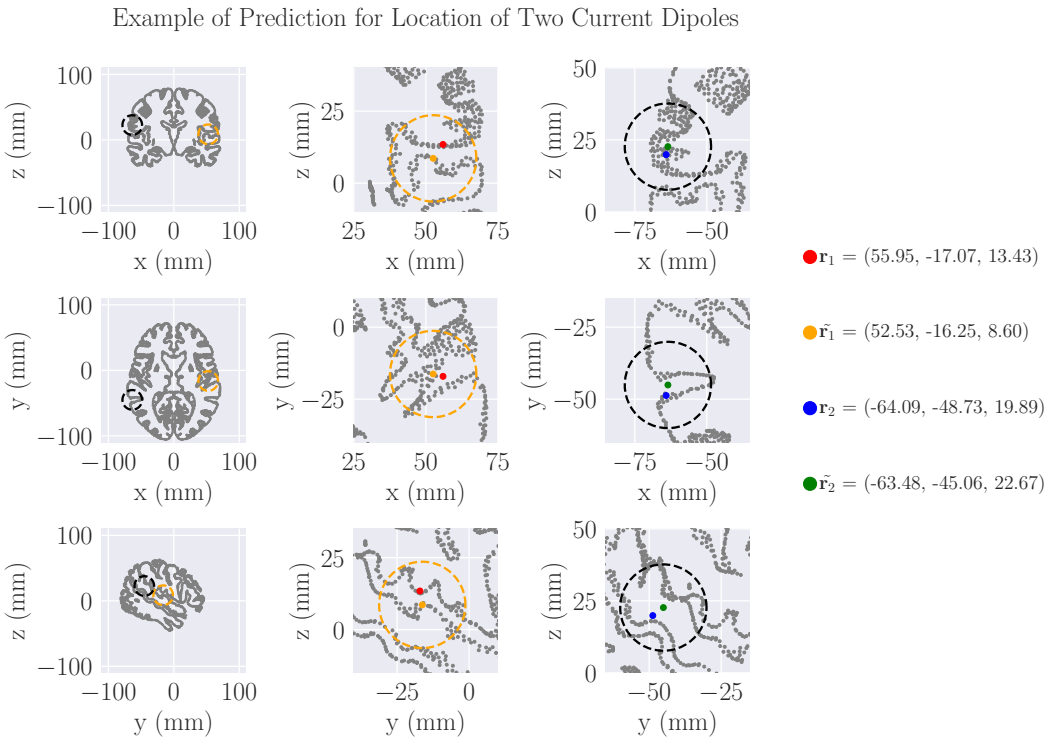

Figure 8.4: Depiction of prediction and true value of the locations of two current dipoles within a sample from the test dataset.

8.4 Performance Evaluation using the CNN

We proceed to assess the performance of our Convolutional Neural Network in the task of predicting the positions of two current dipoles within the NY cortex. In Figure 8.5, we present the training and validation mean Euclidean distance loss as function of training epochs. The CNN underwent a training process, comprising 800 epochs, with each epoch requiring approximately 40 seconds to complete. This training duration represents a slight increase compared to the FFNN. Analogous to our observations with the FFNN model, we noted that the validation loss stabilized around epoch 300, suggesting that an earlier termination of training could have been considered. The training loss, much like the FFNN, reached a plateau after 400 epochs, indicating the absence of overfitting. The learning rate scheduling, mitigate fluctuations in the validation loss, where the learning rate was decreased to 0.0002 at epoch 250, leading to a decline in the validation loss and eventual stabilization at a value of 19.65 mm by epoch 800. Notably, this final validation loss is somewhat higher than that achieved by the FFNN.

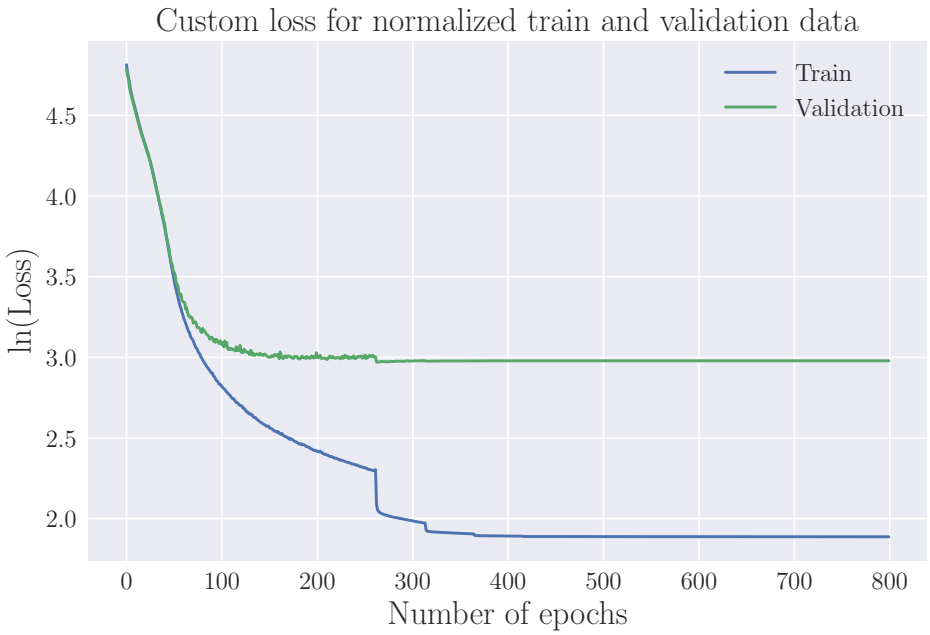

Figure 8.5: Training- and validation loss for the CNN, predicting two current dipole sources, as function of epochs.

Table 8.3 provides an evaluation of the CNN's accuracy, expressed in terms of the Euclidean distance between predicted and target dipole positions for the test dataset. The CNN demonstrates an accuracy smaller than 15 mm for 78.87% of the test samples. However, as the ED threshold is progressively tightened to 10 mm and subsequently to 5 mm, the accuracy diminishes to 50.46% and 7.55%, respectively. To provide a visual representation of the distribution of localization errors, Figure 8.6 showcases a histogram of the ED between predicted and target positions for 20,000 test samples, organized into bins of width 3 mm. As lined out in the previous result section for the FFNN each bin represents the number of samples with prediction errors falling within the specified interval with range 3 mm. However, in contrast to the FFNN model, this histogram reveals that while most predictions exhibit errors smaller than 15 mm, a significant proportion experiences higher errors, with approximately 20% falling beyond this threshold, which is a double from what we saw with the FFNN.

Table 8.4 provides a comprehensive evaluation of the CNN's performance, employing various error metrics. The MAE for the x, y, and z-coordinates of the dipole positions averages at 5.40 mm, indicating accurate localization predictions in these spatial dimensions. However, the MSE values, ranging from 63.53 mm^2 to 79.95 mm^2, are higher than the corresponding values observed in the FFNN model, suggesting a broader dispersion of errors. The

Euclidian Distance for Test Samples		
ED <5 mm	ED <10 mm	ED <15 mm
7.545 %	50.460 %	78.870 %

Table 8.3: **ED between targets and predictions of test samples; Predicting Two Dipole Locations with CNN**
Performance of the network on the test dataset comprising 20,000 samples, presented as the percentage of samples falling within ED thresholds of 5 mm, 10 mm and 15 mm respectively.

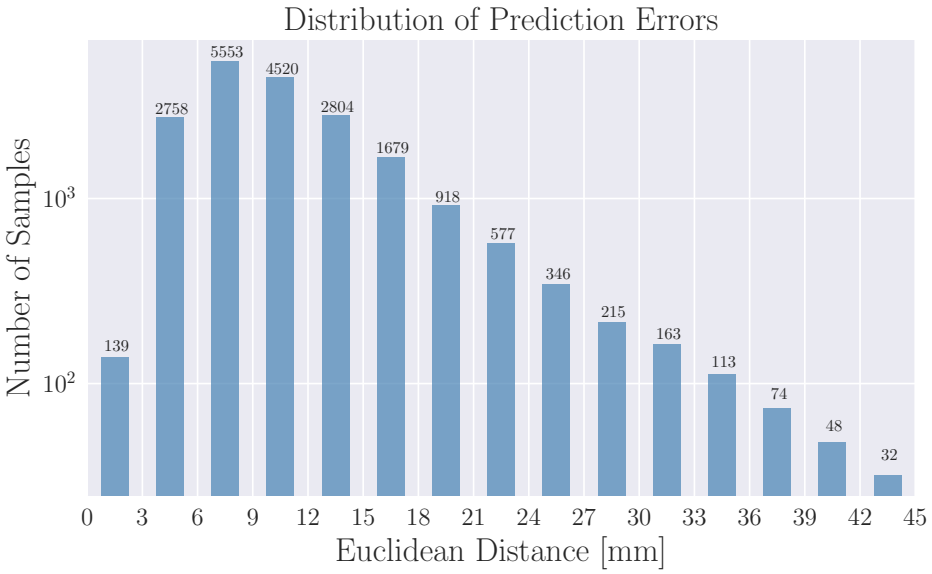

Figure 8.6: Histrogram illustrating the Euclidean Distance for predicted dipole locations, organized into bins of width 3 mm.

RMSE averages at 8.397 mm. Consistent with the single-dipole problem and the FFNN's performance in this multi-dipole problem, we observe somewhat higher errors across all metrics when predicting the depth (z-coordinate) of the dipole sources. This suggests that accurate z-coordinate prediction remains the most challenging target. However, it is important to note that the increase in errors is modest and should not be considered significant without further investigation.

	Error Metrics for Target Values			
	x-coordinate [mm]	y-coordinate [mm]	z-coordinate [mm]	Position Error [mm]
MAE	5.395	5.534	6.265	5.731
MSE	63.533	64.433	79.952	69.306
RMSE	7.971	8.027	8.942	8.325

Table 8.4: **CNN: Evaluation of the network's performance utializing different Error Metrics.**
CNN performance on test dataset consisting of 20,000 samples. The errors are measured using Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE).

Chapter 9

Extending the Feed Forward Neural Network

So far we have seen how a simple FCNN and a compatible convolutional neural network performs in the task of localizing both single and multiple current dipoles. In this chapter, we will delve into how minor adjustments in data design and network architecture can enhance the FCNN's ability to identify various attributes of current dipoles. We will introduce two distinct extensions of the initial inverse problem, which will put the predictive capabilities of the FCNN to the test in handling more complex scenarios.

The first case involves extending the data set by assigning individual magnitudes to each dipole source. This extension presents the network with the task of predicting both location of the source and its corresponding magnitude. The second scenario transitions from predicting the location of a single dipole source to estimating the center and radius of a population of dipoles, while also determining the magnitude of the signal strength of the entire dipole population.

In Section 9.1, we will detail the modifications made to the simulated data that is the same for both of the problems, along with the necessary adjustments to the network architecture. Moving on to Sections 9.2 and 9.3, we will present the distinct data simulation choises for each of the problems, alongside the performance of the network in addressing these more complex challenges.

In Section 9.1, we will outline the modifications made to the data and network architecture that have been implemented to address both of the problems. Transitioning to Sections 9.2 and 9.3, we will delve into the specifics of data simulation tailored to each problem, in addition to evaluating the network's performance when dealing with these more complex problems.

9.1 Method: Adjusting Data Set and Architecture

Before delving into the performance of the FCNN regarding the intricate problems of predicting dipole strengt and radius alongside location, we will address important considerations that apply to both extensions. This encompasses instructing the network on how to handle varying units in its output, selecting an optimal cost function tailored to our specific problems, and establishing criteria for assessing the network's performance in the context of its intended tasks.

9.1.1 Scaling of Target Values

Include original ranges one more time? In the extended EEG inverse problems, the neural network is tasked with predicting target values that vary significantly in their range and units. To comprehend why this can be problematic, it is essential to revisit the purpose of the cost function. Neural networks employ cost functions to quantify the error between their predicted outputs and the target values. Depending on the cost function utilized, this error is expressed in units related to the true dimension of the target value. When a neural network has multiple outputs, the overall cost is determined by the sum of distinct errors corresponding to each output. However, adding together error terms with distinct units is a non-intuitive task. Letting the network do so, may lead to an ambiguous interpretation of the cost and hinder meaningful evaluation of the network's performance. Furthermore, the variation in the range of target values poses an additional challenge. This variation can result in certain domains of the data having an imbalanced influence on the overall error calculation, potentially overshadowing targets with smaller ranges. Such an asymmetry in the error calculation can lead to a biased optimization process and hinder the network's ability to effectively learn from the data.

To address these issues, a preprocessing step is required, which involves scaling of the target data. By instructing the neural network to output unitless vectors $\mathbf{y}$ with elements constrained within the range of 0 to 1, these challenges can be mitigated. This instruction can be given to the network by scaling the its target values $\tilde{\mathbf{y}}_i \in \tilde{\mathcal{Y}}$ for $i \in \{0, N-1\}$, where N is the number of samples. The scaling is done by normalizing each component of the target vector $\tilde{\mathbf{y}}_i = [\tilde{y}_{i,0}, \tilde{y}_{i,1}, ..., \tilde{y}_{i,d-1}]$ separately. Here, d denotes the dimension of $\tilde{\mathbf{y}}_i$, which corresponds to the number of target values contained in each sample. The normalized target value $\tilde{y}'_{i,j}$ for the i'th sample and the j'th dimension, is calculated as follows:

$$\tilde{y}'_{i,j} = \frac{\tilde{y}_{i,j} - \min(\{\tilde{\mathbf{y}}\}_j)}{\max(\{\tilde{\mathbf{y}}\}_j) - \min(\{\tilde{\mathbf{y}}\}_j)}. \tag{9.1}$$

In this equation, $\min(\{\tilde{\mathbf{y}}_j\})$ and $\max(\{\tilde{\mathbf{y}}_j\})$ represent the minimum and

maximum values for a specific target dimension across all samples. By normalizing the target values in this manner, the neural network is guided to produce predictions within the range of 0 to 1. Consequently, the network's cost is computed by summing the unitless errors corresponding to each target prediction, for each dimension, ensuring that all target values contribute equally to the overall error calculation. As a result, the neural network learn more effectively, thereby enhancing its performance in various tasks.

9.1.2 Sigmoid as Last Layer Activation Function

In the context of predicting the postion of a single dipole source, the FFNN did not employ any activation in its final layer. However, given that our output data has been normalized, we find it appropriate to use the *Sigmoid* activation function in the output layer. Sigmoid is a logistic mathematical function that maps its input to a bounded range between 0 and 1, as expressed by the equation:

$$f(x) = \frac{1}{1 + e^{-x}}. \tag{9.2}$$

In Figure 9.1 we have provided a visualization of Sigmoid's transformation characteristics. The mapping of input values aligns with our desired output range, with normalized target values for the exteded problems. This choice of activation in the output layer aligns with our desired output range, considering the normalized target values for the extended problem domains. The final layer activation aims to facilitate the training process, constraining the network from generating outputs beyond the intended normalized target range.

9.1.3 Choosing an Optimal Cost Function

The formulation of a customized cost function tailored to the specific modeling objectives is imperative when confronted with machine learning problems in need of multiple network outputs, each characterized by distinct units and interpretations, such as ours. We aim to construct a model that can effectively predict not only the spatial coordinates of a current dipole moment but also accurately estimate the dipole's magnitude and the radii associated with a population of multiple current dipoles. To achieve this objective, our ideal cost function comprises several components.

Firstly, it calculates the Euclidean distance between predicted spatial coordinates $x, y, z \in \mathbf{y}$ and correspnding true values $\tilde{x}, \tilde{y}, \tilde{z} \in \tilde{\mathbf{y}}$ for N number of samples:

$$\text{MED}(\mathbf{y}, \tilde{\mathbf{y}}) = \frac{1}{N} \sum_{i=0}^{N-1} \sqrt{(x_i - \tilde{x}_i)^2 + (y_i - \tilde{y}_i)^2 + (z_i - \tilde{z}_i)^2}. \tag{9.3}$$

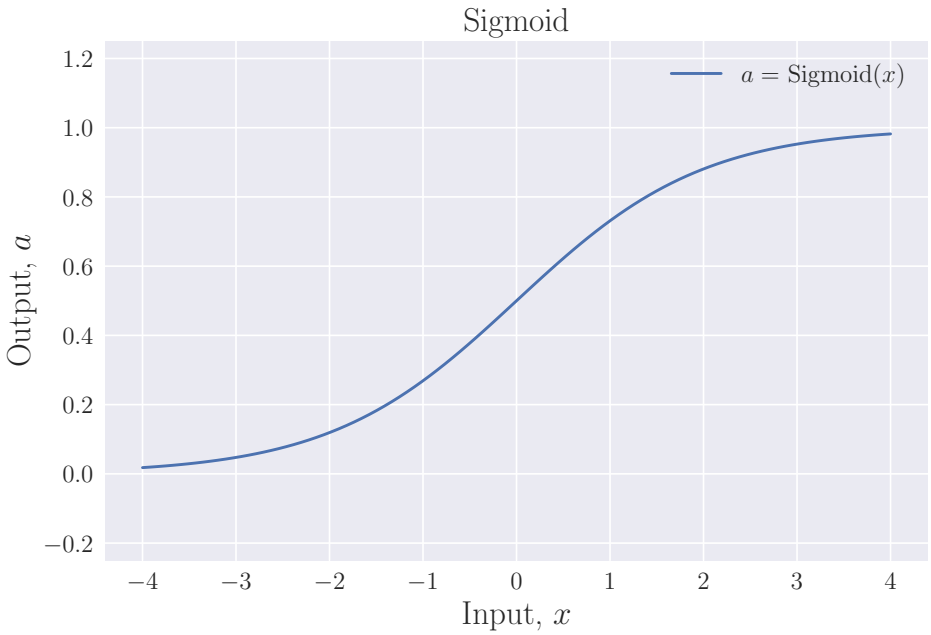

Figure 9.1: **Sigmoid activation function**: transforms the input value x into a smooth S-shaped curve, mapping it to a range between 0 and 1.

Additionally, the cost function calculates the absolute error between the predicted magnitude $A \in \mathbf{y}$ and the true magnitude $\tilde{A} \in \tilde{\mathbf{y}}$:

$$\text{MAE}_A(\mathbf{y}, \tilde{\mathbf{y}}) = \frac{1}{N} \sum_{i=0}^{N-1} |A_i - \tilde{A}_i|. \tag{9.4}$$

Similarly, it aims to quantify the absolute error between the predicted radius r $\in \mathbf{y}$ and the true radius $\tilde{\text{r}} \in \tilde{\mathbf{y}}$:

$$\text{MAE}_r(\mathbf{y}, \tilde{\mathbf{y}}) = \frac{1}{N} \sum_{i=0}^{N-1} |r_i - \tilde{r}_i| \tag{9.5}$$

Remove r=3 (simple dipole) or add r=6 (two dipoles)? With an output $\mathbf{y} \in \mathcal{Y}(d)$, where d represents the dimension of $\mathbf{y}$ and $\mathcal{Y}(d) \subset \mathbb{R}^d$, a collective cost function, tailored for each of the distinct problem scenarios, can be defined as follows:

$$C(\mathcal{Y}, \tilde{\mathcal{Y}}) = \begin{cases} \text{MED}(x, y, z), & \text{if } d = 3 \\ \text{MED}(x, y, z) + \text{MAE}(A) & \text{if } d = 4 \\ \text{MED}(x, y, z) + \text{MAE}(A) + \text{MAE}(r), & \text{if } d = 5 \end{cases} \tag{9.6}$$

In the case where $d = 3$, the simplest problem is considered, where the network predicts the coordinates of a single-point current dipole, as explored in Chapter 6 and Chapter 7. If $d = 4$, the network predicts the x-, y-, and z-coordinates of a single dipole, in addition to the magnitude A of the signal strength. When $d = 5$, the target vector encompasses all previously mentioned values, along with the radius of a current dipole population.

9.1.4 Overview on Arcitecture and Hyperparameters

Figure 9.2 presents a visualization of the architecture of the extended FCNN, which is designed to produce d target values depending on the specific problem to solve. The architecture of the FFNN is equal to that employed for both the single dipole problem discussed in Chapter 6 and the multipole dipole problem examined in Chapter 8, with 231 input nodes, corresponding to the number of electrode recording measurements for one sample, and retains the same number of hidden nodes and layers.

In this extension of the FFNN, we continue to employ ReLU as the activation function in the input layer and the hyperbolic tangent for the hidden layers. This choice of architecture, combined with the selected hyperparameters, has consistently yielded promising results. The key difference in this extended network architecture is the integration of the Sigmoid activation function in the output layer. For an overview of the essential hyperparameters utilized to generate the forthcoming results in this extended model, please refer to Table 9.1, which offers a summary of these overall elements.

This must come somewhere else?

9.1.5 Denormalization for Performance Evalutaion

In this extended version of the fully connected feed-forward neural network (FCNN), assessing network performance through standard train-validation-loss plots becomes less informative due to the normalization of target values and a more complex cost function. Reading off unitless loss values from plots, where the cost function value is plotted against epochs, provides little insight beyond whether the network is capable of decreasing the loss with an increasing number of training iterations.

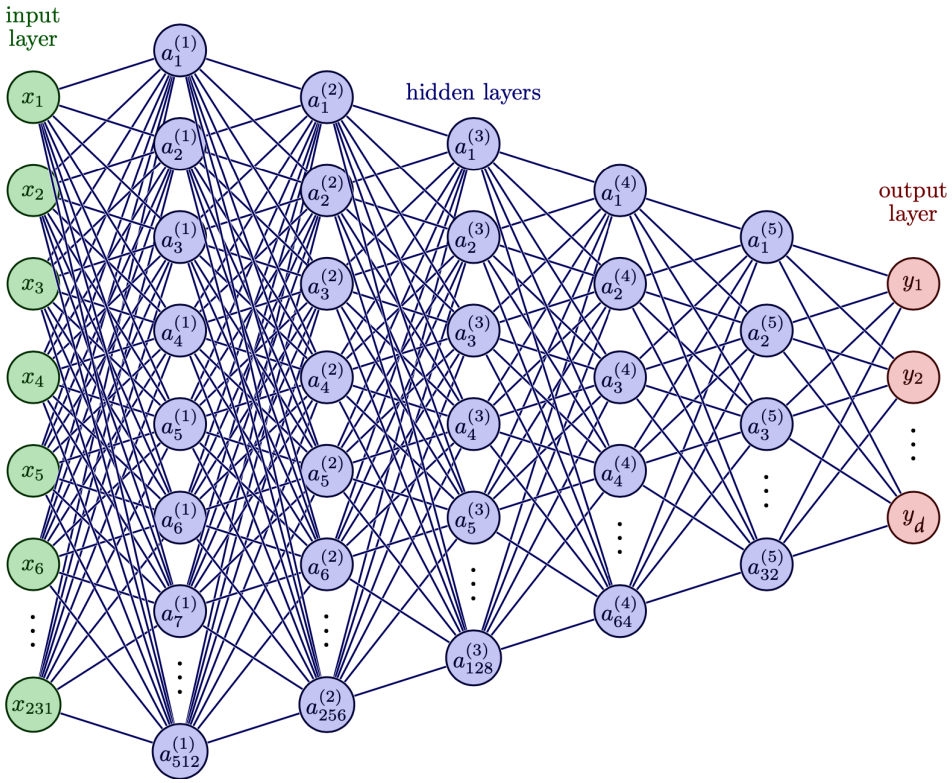

Figure 9.2: **Architecture of the Extended Feed-Forward Neural Network**. The input layer comprises 231 electrode recording input values, and the network outputs predicted d target values depending on the specific problem to solve. The FFNN consists of five hidden layers.
The vizualization has been made with Latex, with code adapteded from Neural Networks in TikZ with attribution to Izaak Neutelings and is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License [32].

When evaluating the network's performance on the test data set, which still comprises 20,000 samples, it is therefore essential that the predictions outputted by the FFNN undergo denormalization. This enables us to facilitate a meaningful evaluation against the true target values. The denormalization process takes a straight forward approach, where we simply do the opposite operations from the once performed during normalization 9.1. The denormalized predicted value $\tilde{y}_{i,j}$ of the network is then given as:

$$\tilde{y}_{i,j} = \left(\tilde{y}'_{i,j} + \min(\{\tilde{\mathbf{y}}\}_j\right)\left(\max(\{\tilde{\mathbf{y}}\}_j) - \min(\{\tilde{\mathbf{y}}\}_j)\right). \tag{9.7}$$

Here, $\tilde{y}'_{i,j} \in \tilde{\mathcal{Y}}'$ is the normalized target value for the i'th sample and the j'th dimension of the target vector , where $i \in \{0, N-1\}$ and $j \in \{0, d-1\}$. In

Extended FFNN	
Hyperparameters	**Value**
Hidden layers	5
Optimizer	SGD
Learning rate (initial)	0.001
Momentum	0.35
Weight decay	0.1
Minibatch size	32
Dropout	0.5
Act.func in first layer	ReLU
Act.func in hidden layers	Tanh
Act.func in last layer	Sigmoid

Table 9.1: Hyperparameters for the Extended FFNN.

the equation $\min(\{\tilde{\mathbf{y}}_j\})$ and $\max(\{\tilde{\mathbf{y}}_j\})$ denotes the minimum and maximum values for the specific target dimension across all samples.

9.2 Predicting Single Current Dipole Sources with Varying Magnitudes

In this section, we introduce the concept of various magnitudes for single current dipole sources, which adds an additional dimension to the output of the FCNN, with $d = 4$. 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 location generating abnormal brain activity but also comprehend the *strength* of abnormality.By incorporating magnitude prediction into the network, we gain valuable insights into the problem at hand, allowing the network to extract more detailed information about the underlying brain activity from the EEG data.

9.2.1 Adjusting Data Set

For the purpose of this extended EEG inverse problem, the data set is modified by introducing varying magnitudes to each dipole, ranging from 1 to 10 nAm. This adjustment does not impact the number of features, which corresponds to the number of electrode recordings for each sample. However, it does increase the number of target values by 1. Figure 9.3 showcases two examples from the data set, where the dipole's location remains constant while the magnitude of the dipole signal varies. We observe that the shape of the EEG signal remains consistent while the strength of the EEG signal

is significantly higher for the dipole with the largest initiated magnitude, as intended.

It is important to note that this data set represents the data before the standardization procedure of the input data. However, scaling the EEG data ensures that each electrode measurement is uniformly affected, maintaining consistent relative differences in magnitude between weaker and stronger dipole signals.

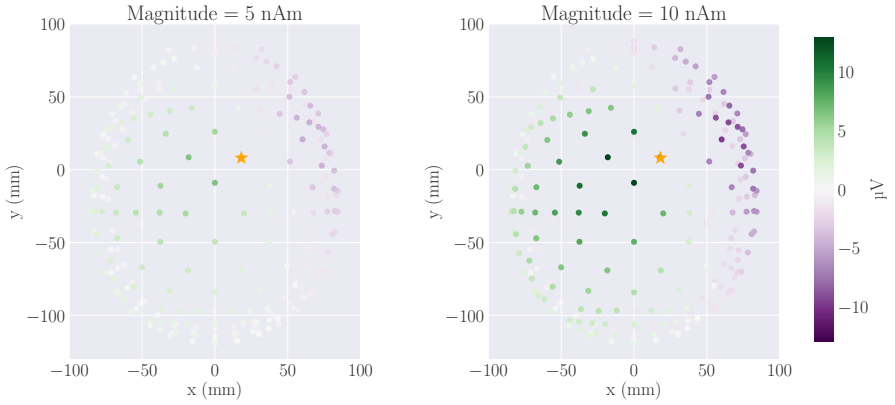

Figure 9.3: EEG data for two samples with current dipole magnitudes equal to 5 and 10 nAm, respectively. The EEG recordings have a range between -10 and 10 μV.

9.2.2 Performance Evaluation

To assess the network's performance, we start by analyzing the accuracy in relation to training epochs, as depicted in Figure 9.4. We emphasize that it is important to note that the target values have been normalized, resulting in unitless loss measurements. 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 is able to capture underlying patterns in the data. Moreover, both the training and validation loss stabilize after approximately 1100 epochs, suggesting that the network has reached its optimal performance level. Each epoch takes about 20 seconds to finish, leaving us with a training time of roughly 8.5 hours in total.

In Figure 9.5, we present the progression of the loss for the target parameters. Notably, after approximately 1100 epochs, the loss stops fluctuating for all target values, suggesting potential full convergence at this stage. It is evident that the loss for the magnitude target converges at a higher value, compared to the target coordinates. The x-, y-, and z-coordinate targets

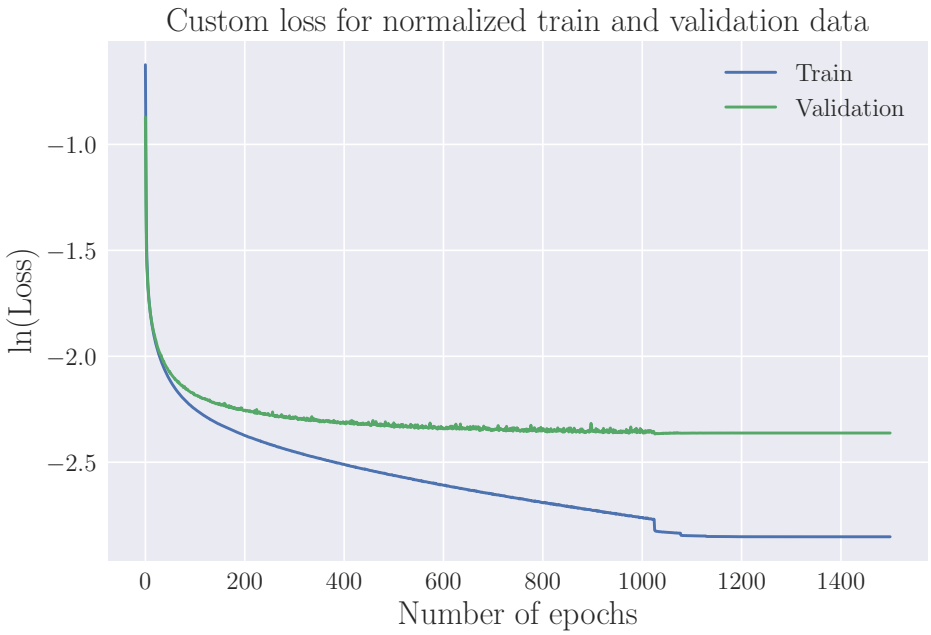

Figure 9.4: Training and validation loss as functions of epochs for the extended FFNN model, which predicts both location and magnitude parameters. The analysis is based on simulated data comprising 50,000 samples and spans a training duration of 1500 epochs.

gradually converge to values that closely align with one another. Among these, the y-coordinate achieves the lowest loss, followed by the x-coordinate, and then the z-coordinate.

In Table 9.2, we present the network's performance across various target categories using different error metrics. Analyzing the outcomes pertaining to the target coordinates reveals a noteworthy consistency in mean absolute error, with an average deviation of less than 1.5 mm from the true values. In contrast to when studying single dipoles with constant strength, the z-coordinate stands for the highest contribtuion to this MAE. The MAE for the magnitude variable equals 0.539 nAm. Having that the magnitude range spans from 1 to 10 nAm, this error translates to a relative error of 6.00%. In other words, the network's predictions for the magnitude variable are, on average, within 6.00% of the true values within this range.

In addition to MAE, we look into the network's performance through the metrics of mean squared error and root mean squared error, which offer complementary insights. For the x-coordinate, the MSE stands at 3.438 mm^2, while for the y- and z-coordinate, it is 3.860 mm^2 and 3.862 mm^2, respectively. The MSE for the magnitude target, measures 0.650 nA^2m^2.

The Root Mean Squared Error, which indicates the standard deviation

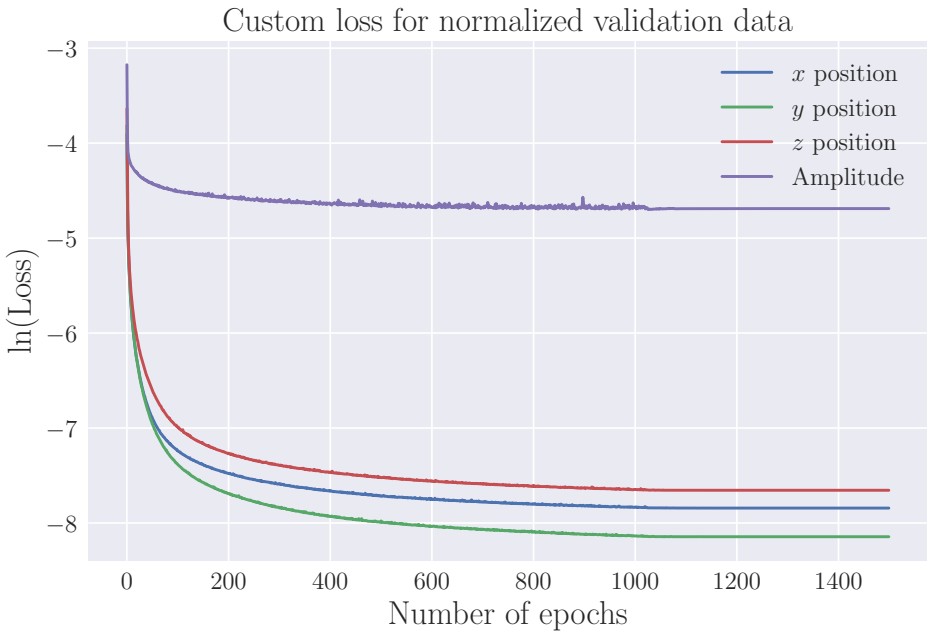

Figure 9.5: Validation loss as a function of epoch for each target value, including the x-, y-, and z-coordinates of the dipole, as well as the magnitude of the dipole signal.

of prediction errors and their distribution around the mean, reports values slightly lower than the corresponding Mean Squared Error values. Specifically, RMSE values measure at 1.854 mm, 1.965 mm, and 1.965 mm for the distinct target coordinates, and 0.806 nAm for the magnitude target. These RMSE values suggest that, on average, the prediction errors align with the overall spread of errors, without significant outliers or extreme deviations from the mean error. This indicates a level of stability and predictability in the error distribution.

In Figure 9.6, we present the AE and SE metrics calculated between predicted and target coordinate values as a function of the magnitude of the dipole strengths. The figures reveal a relatively higher occurance of outliers when the magnitude of the dipole strength is small, with elevated AE and SE values. This trend may initially suggest that the neural network encounters greater challenges in accurately predicting the positions of dipoles with smaller magnitudes. However, we cannot rule out that these disparities might be related to alternative factors such as depth positioning, intricate cortical folding patterns, or other random fluctuations.

When examining the correlation coefficients for each of the figures, we find that the correlation between the strength of the dipole and the error in position prediction has a correlation of -0.01 and -0.02 for the AE and SE,

	Error Metrics for Target Values				
	x-coordinate [mm]	y-coordinate [mm]	z-coordinate [mm]	Position Error [mm]	Magnitude [nAm]
MAE	1.348	1.448	1.420	1.405	0.539
MSE	3.438	3.860	3.862	3.720	0.650
RMSE	1.854	1.965	1.965	1.929	0.806

Table 9.2: **Evaluation of the extended FFNN's performance utializing different Error Metrics.**
Performance for the extended FFNN on test data set consisting of 1000 samples. The errors are measured using Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE).

Euclidian Distance for Test Samples		
ED <5 mm	ED <10 mm	ED <15 mm
90.930 %	99.505 %	99.925 %

Table 9.3: **ED between targets and predictions of test samples; Predicting Location and Magnitude with the FFNN**
Performance of the FFNN on the test dataset comprising 20,000 samples, presented as the percentage of samples falling within ED thresholds of 5 mm, 10 mm and 15 mm respectively.

respectively. These correlation coefficients do suggest a negative correlation (decreasing error with increasing magnitudes), but their values are nearly zero, indicating a weak linear relationship. Upon closer inspection of the figure, keeping in mind that it contains 20,000 scatter points, we notice that most of the predictions exhibit an AE smaller than 5 nAm and an SE smaller than 10 nA^2m^2. This suggests that, despite observed variations in predictions for samples with smaller magnitudes, the majority of the FFNN's positional predictions consistently cluster within this range of lower errors.

The Mean Euclidean Distance of the FFNN's predictions on the unseen test data is measured at 2.815 mm. Table 9.3 presents the percentages of predictions of test samples falling within various MED threshold values. Notably, the network achieves an accuracy smaller than 12 mm for 99.8% of the test data. Furthermore, 64.3% of the samples exhibit a MED smaller than 3 mm, an achievement that, while robust, falls slightly short of the FFNN's performance in the previous, less complex problem.

Figure 9.7 displays two panels, depicting the ED between predicted and true target coordinates and AE between predicted and true magnitude of each test sample within bins of width 1 unit. In the left panel, representing the ED histogram, the bins corresponding to ED values of 2 and 3 mm are the most populated, containing the largest proportion of samples. This panel

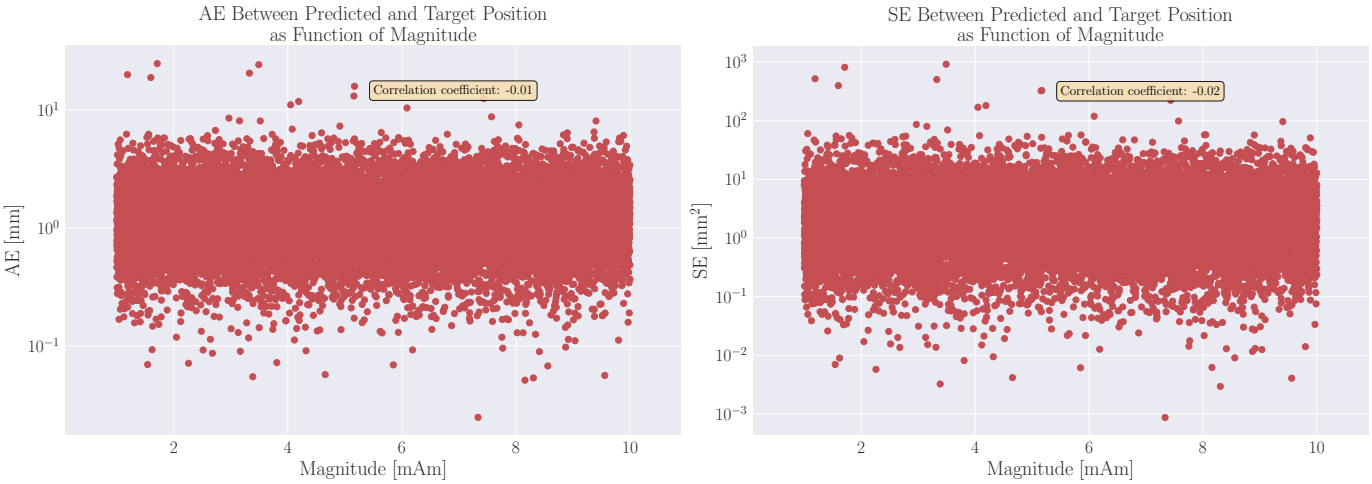

Figure 9.6: Scatter plot of Abolsute Error and Squared Error computed between predicted and target coordinate values, as function of the magnitude of the dipole strengths.

also illustrates that the majority of predicted locations for the test samples exhibit a ED error smaller than 14 mm.

Turning our attention to the right panel, which displays the AE for predicted magnitudes, we observe that the bin holding samples with a magnitude prediction AE less than 1 nAm holds the most samples. Within this bin, 17,014 out of the 20,000 samples can be found, signifying that the network predicts the magnitude with a AE smaller than 1 nAm for approximately 85% of the sample set.

9.3 Predicting Region of Active Correlated Current Dipoles with Magnitudes

Kan jeg omtale populasjonene som sfæriske? In this problem we expand the data set for the FFNN to include varying radii and magnitudes for the origins generating the electrical activity detected by the recording electrodes. In this extension, the FFNN progresses from predicting the location and magnitude of individual current dipole moments to estimating the centers of larger spherical populations. This enhancement aims to enable the network to extract more detailed information from EEG data, which could be valuable in scenarios where understanding the extent of brain abnormalities is crucial.

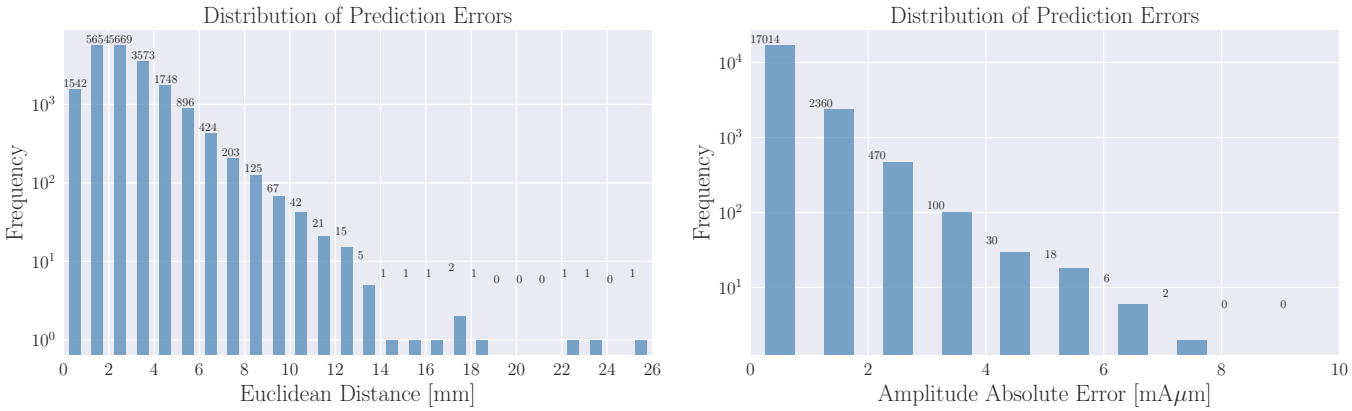

Figure 9.7: Left panel illustrates the distribution of Mean Euclidean Distance for the predicted dipole locations, organized into bins of width 1 mm. Right panel holds the distribution of Mean Absolute Error for the predicted magnitude of dipole electrical signal, presented in a similar format, with bins of width 1 nAm.

9.3.1 Adjusting Data Set

In order to optimize the neural network's capacity for predicting dipole population regions, we made specific adjustments to the data set. Within this framework, each dipole population comprises individual dipoles distributed across all points within the NY head cortex falling within a spherical volume ranging from 1 mm to 15 mm in radius. Notably, guidelines established by Delucchi et al. (1962)[12], Cooper et al. (1965)[11], Ebersole (1997)[14], Nunez and Srinivasan (2006)[36], and Schomer and Lopes da Silva (2018)[33] in the clinical context of spontaneous EEG, suggest that for EEG recordings to detect brain activity, approximately 6-10 cm^2 of contiguous brain tissue must exhibit synchronous activity. A spherical radius of 15 mm translates to an approximate area of 7 cm^2, and thus falls within this criterion [35].

However, in event-related potentials (ERPs) experiments, which is what we think our simulated EEG data represents, a different perspective emerges. In ERPs the brain's responses to specific stimuli is investigated and in order to enhance the reliability of these responses, trial-averaging is a common practice, as outlined in Chapter 3. Consequently, as suggested by these authors, it becomes plausible that EEG activity arising from brain regions smaller than the typical 6-10 cm^2 criterion for spontaneous EEG could be detectable - population sizes that we will be training our network to identify.

However, in the context of event-related potentials (ERPs) experiments, which is how our simulated EEG data can be considered, a different perspective arises. ERPs investigate the brain's responses to specific stimuli, and to enhance the reliability of these responses, trial-averaging is a com-

mon practice, as outlined in Chapter 3. Consequently, as suggested by these authors, it becomes plausible that EEG activity arising from brain regions smaller than the typical 6-10 cm^2 criterion for spontaneous EEG could be detectable — population sizes that we will be training our network to identify.

For simplification reasons, we maintain the maximum amplitude strength of the total populations at 10 nAm. Consequently, we calculate the maximum number of points within a volume sphere with a radius of 15 mm and reduce this number by 10 in order to determine the strength of a distinct dipole within a given area. Having that the maximum number of dipoles that fit within a volume sphere with an ideal center and radius 15 mm was 899 dipoles, we were left with a dipole strength of $10/899$ nAm for each dipole. The strength of a dipole population is thus directly proportional to the radius of the dipole population. While this may not perfectly represent real-world scenarios, it provides a reasonable approximation for our model.

In Figure 9.8, we present an example of a dipole population and the corresponding EEG signal. The upper panels show the EEG signals for the specific sample, seen from different angels. The recording electrode locations are presented as filled circles, where the color of the fill represents the amplitude of the measured EEG signal for the given electrode. The yellow filled circles in the plots in the lower panel represents the diple populations, i.e. positions within the cortex where dipoles have been placed. The plots within the figure are seen from the xz-plane, xy-plane, and yz-plane.

9.3.2 Performance Evaluation

In Figure 9.9, we present the training and validation costumized loss for the FFNN as a function of epochs. We once again emphasize that the network's loss is unitless, meaning that the figure only provides a visual representation of the network's training progress rather than directly interpretable loss values. The network was trained for 1000 epochs. We see a clear trend of decreasing loss with an increasing number of epochs. However, after epoch 800, both training and validation loss appears to reach convergence, and beyond this point, there is no further improvement in loss. We can therefor conclude that overfitting has not taken place during training. At epoch 749, the learning rate is adjusted from 0.001 to 0.00004, resulting in a noticeable reduction and stabilization for both training and validation loss. On average, each epoch takes approximately 23 seconds to complete, resulting in a total training time of approximately 9.5 hours.

Figure 9.10 displays the validation loss for each target value as function of epochs. We observe that the loss curves for the target coordinates reach their minima at 100 epochs and then begin to increase before stabilizing around 800 epochs. Here, the loss curves for the x- and z-coordinates are somewhat similar and stabilize at higher values than the loss for the y-coordinate. Shifting our focus to the magnitude and radius targets, we observe that the

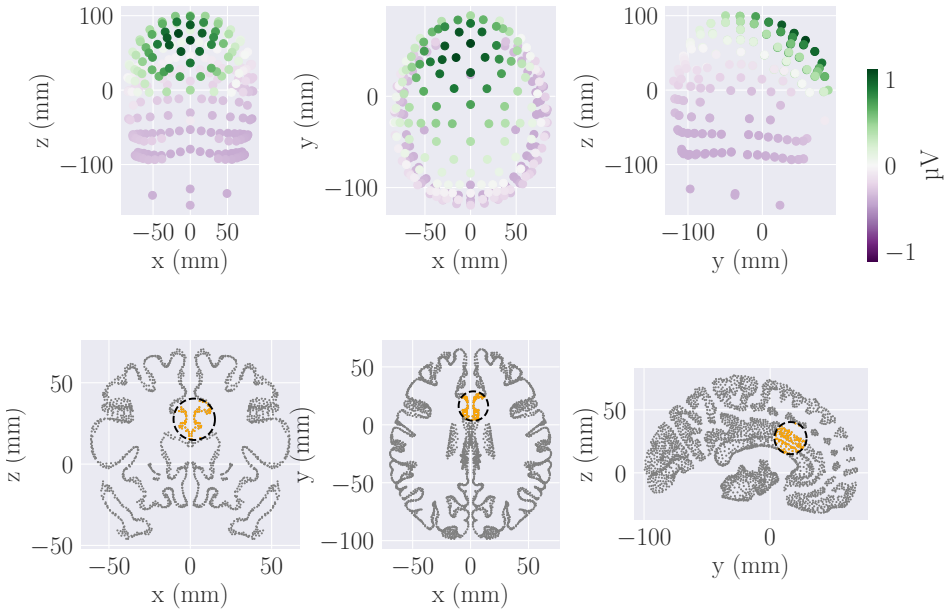

Figure 9.8: EEG for a sample containing a spherical population of current dipole sources with a random center within the cerebral cortex. **Upper panels:** The EEG measure is seen from the front (xz-plane), side (yz-plane), and top (xy-plane) of the cortex. EEG electrode locations are presented as filled circles, where the color of the fill represents the amplitude of the measured signal for the given electrode. **Lower panels:** Gray filled circles represent points within the NY cortex where a dipole could have been placed. Yellow filled circles represent positions within the cortex where dipoles have been placed.

loss evaluations corresponding to these values follow a similar pattern. The loss curves have similar shapes, but the radius loss curve has higher losses than the magnitude curve. Somewhere between 700 and 800 epochs, these loss curves also converge. Although the loss curves for the target coordinates have their lowest values at an earlier stage, the model aims to minimize the *overall* loss, and, therefore, training is not stopped at the point where the coordinate losses have their minima.

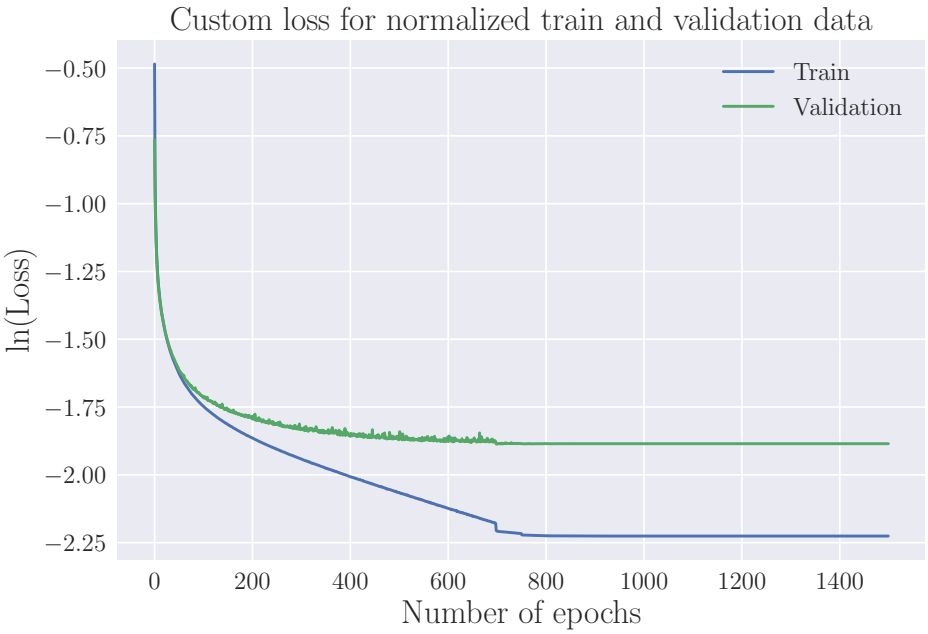

Figure 9.9: Training and validation custom loss trends as functions of epochs for the FFNN model, which predicts the center, magnitude, and radius of dipole populations. The analysis is based on simulated data comprising 50,000 samples and spans a training duration of 1500 epochs.

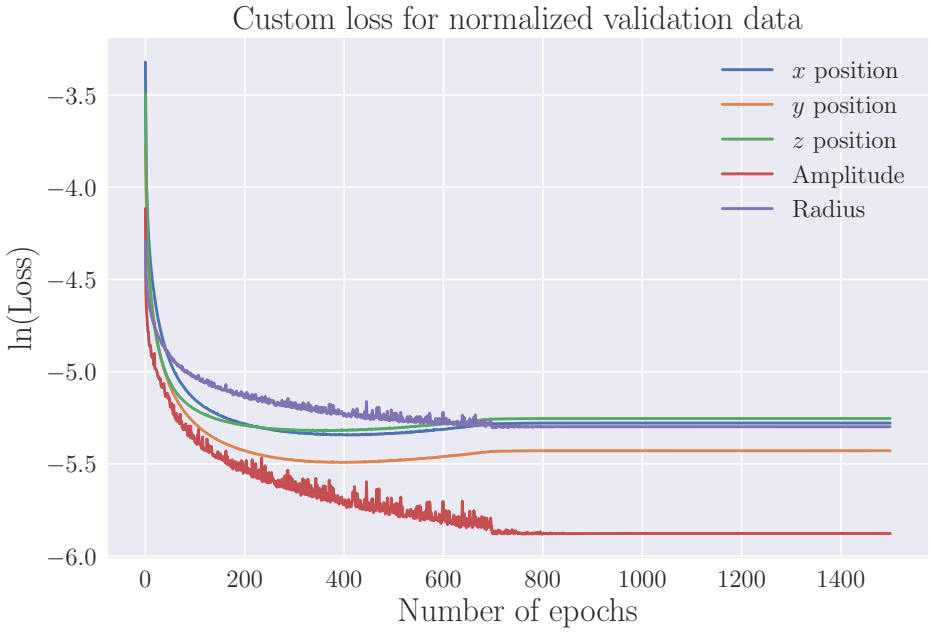

Figure 9.10: Validation MSE loss as a function of epoch for each target value, including the x-, y-, and z-coordinates of the center of the dipole population, as well as the amplitude and radius.

We furter evaluate our model's predictions, by feeding it with unseen test data. By denormalizing the output of the network and comparing it to the target values the MED between the predicted and target coordinates measures to 7.85 mm, MAE between predicted and target magnitude equals 0.33 nAm and MAE between predicted and target radius is 0.76 mm. The MED measures a slightly higher value compared to when predicting only position and magnitude strength of the dipole. However, we note that the MAE corresponding to the magnitude has decreased in this more complex problem, now corresponding to a relative error of only 3.67%. In table **??** we can read off the percentages of the network predictions of test samples that falls within various threshold values. We have that 79.68 % of the predictions has a ED in position smaller than 10 mm, 98.57% an AE in magnitude smaller than 2 nAm and 97.22% an AE in radius smaller than 3 mm.

Table 9.5 further provides an overview of the percentage of samples that fulfill multiple of threshold values at once. These numbers have been found with the initial condition that all predictions fulfill the constraint that the ED between predicted and target position is smaller than 10 mm. A result drawn from the table is that 78.290% of the FFNN's predictions have an AE smaller than 2 nAm in magnitude and 3 mm in radius.

To ensure the model's accuracy is comparable to the work done by others and to further assess its ability to predict the center of dipole populations, in addition to magnitude and radius, we employ the same evaluation metrics as in previous problems: MAE, MSE, and RMSE. The results for these error metrics are presented in Table 9.6.

The MAEs for the target coordinates, i.e., the center of the dipole population, average at 3.96 mm. This is the highest MAE measured among the predictions made by the network across all the problems studied. As observed when predicting both the magnitude and position in the previous problem, the z-coordinate contributes the most to this MAE.

Concerning the MSE, we observe relatively small errors for amplitude and radius, with values of 0.312 and 1.114 mm^2, respectively. The MSE between the predicted and target coordinates ranges from 42.58 mm^2 to 66.816 mm^2, indicating the widest spread of errors compared to the previous problems studied. An increase in RMSE is also evident. This suggests that, on average, the prediction errors do not always align with the overall

Eucledian Distance: Position		
ED <5 mm	ED <10 mm	ED <15 mm
56.045%	79.680%	86.435%

Absolute Error: Magnitude		
AE <1 nAm	AE <2 nAm	AE <3 nAm
92.900%	98.565%	99.685%

Absolute Error: Radius		
AE <1 mm	AE <3 mm	AE <5 mm
74.210%	98.220%	99.770%

Table 9.4

	Predictions within different thresholds (Initial condition for position: ED <10 mm)		
	AE <1 nAm	AE <2 nAm	AE <3 nAm
AE <1 mm	58.075 %	59.940 %	60.055 %
AE <3 mm	74.045 %	78.290 %	78.765 %
AE <5 mm	74.315 %	78.890 %	79.515 %

Table 9.5: **Evaluation of the Extended FFNN utilizing different Error Metrics.** Performance of the extended FFNN on a test data set consisting of 20,000 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.

	Error for different target values					
	x [mm]	y [mm]	z [mm]	Center [mm]	Magnitude [nAm]	Radius [mm]
MAE	3.819	4.342	3.722	3.961	0.325	0.765
MSE	42.581	66.816	41.681	50.359	0.313	1.114
RMSE	6.525	8.174	6.456	7.096	0.559	1.056

Table 9.6: **Dipole Population: Evaluation of the Extended FFNN utilizing different Error Metrics.** Performance of the extended FFNN on a test data set consisting of 20,000 samples. The errors are measured using Mean Absolute Error, Mean Squared Error, and Root Mean Squared Error for various target values.

spread of errors. In other words, the model sometimes provides accurate predictions, while in other cases, it does not. This indicates a lower level of stability and predictability in the error distribution compared to the other problems studied.

In Figure 9.11, we present the MAE and MSE metrics calculated between the predicted and target population center values as functions of radius. The panels within the figure do not indicate any significant correlation between the size of a dipole population and the predictive error regarding its center. This observation aligns with the correlation coefficients, which are approximately 0.01 for AE and 0.00 for SE in the prediction of the center.

The panels also reveal that the majority of the samples are predicted with an absolute error smaller than 15 mm. Moreover, the left panel illustrates that the network predictions on the test samples result in a mean error larger than 1000 mm^2. This suggests a comparatively lower level of stability and predictability in the error distribution, in contrast to the other problems studied.

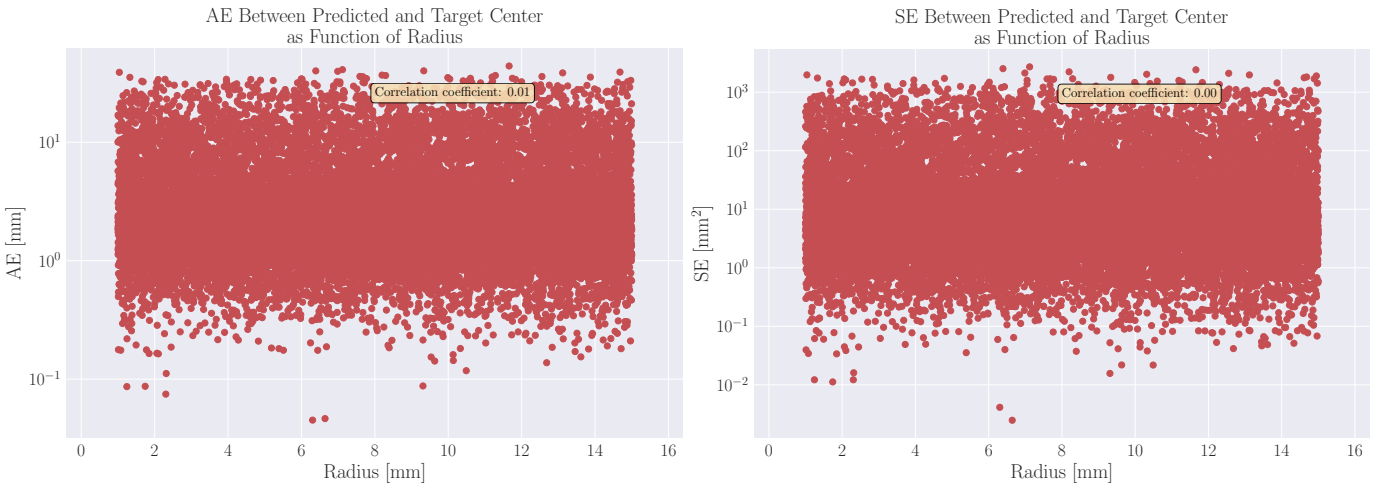

Figure 9.11: **Dipole Population:** Scatter plot of Mean Abolsute Error and Mean Squared Error computed between predicted and target center values, as function of the radius of the dipole populations.

Chapter 10

Discussion

In this chapter, we will delve into the key findings derived from the fully connected neural network (FCNN) and the convolutional neural network (CNN) concerning the localization and identification of current dipole moments. In Section 10.1, we will present an overview of the main results of this thesis, and emphasize the most noteworthy and pivotal findings. Subsequently, sections 10.2 and 10.3 will delve into the architectural aspects of the networks and the dataset employed for training and result acquisition. Section 10.4 is dedicated to contextualizing our results within existing research. Finally, in Section 10.5, we embark on a discussion of future prospects following our approaches.

Can there be a chance that i have in fact overfitted the network? Since the data poins are very close. Also I could have tried to use different headmodel (detailed) and tested the network on these Noise vs no noise say something about time?

10.1 Summary of Main Results

Throughout the various approaches to the EEG inverse problem in this thesis, we have systematically evaluated the neural networks' performance using MED, MSE, and MAE. In Table 10.1, we present a summary of the main results throughout this thesis.

10.1.1 Predicting Dipole Position

As highlighted in Chapter 6, it is possible to achieve a desirable level of accuracy, specifically an error margin smaller than 10 mm, when addressing the inverse EEG problem. This achievement is consistently observed when utilizing head models that accurately represent head geometry and electrode placement, as exemplified by the New York Head Model applied through the LFPy 2.0 software.

	Summary of Main Results					
	Singe Dipole: FCNN	Single Dipoles: CNN	Two Dipoles: FCNN	Two Dipoles: CNN	Dipole with Magnitude: FCNN	Dipole Population: FCNN
d	3	3	6	6	4	5
Coordinate with highest MAE	z	z	z	z	y	z
MED(x, y, z) [mm]	1.3	1.8	8.71	11.78	2.815	7.85
MSE(x, y, z) [mm^2]	0.782	1.643	38.978	69.306	3.720	50.359
MAE(x, y, z) [mm]	0.662	0.898	4.340	5.731	1.405	3.961
MAE$_A$ [nAμm]	-	-	-	-	0.539	0.33
MAE$_R$ [mm]	-	-	-	-	-	0.76

Table 10.1: This is new table with new updates

In all our fully-connected feed-forward neural network (FCNN) approaches, we have consistently succeeded in localizing current dipole moments with precision that remains under this 10 mm threshold. Whether the task is to predict single dipole locations, two dipole locations at once, or dipole locations alongside dipole characteristics, this level of precision is consistently maintained. Similarly, the convolutional neural network (CNN) effectively localizes single current dipoles, albeit exhibiting predictions just above this threshold when tasked with concurrent localization of two current dipoles.

For single dipole localization, the FCNN achieved an error of less than 10 mm for 99.995% of the test samples, while the CNN reached the same level of accuracy for 99.815% of the samples. Subsequently, when the FCNN was challenged to predict varying magnitudes in addition to dipole location, it maintained an accuracy below 10 mm for 99.505% of the test data. However, when faced with the task of predicting the center of dipole populations of varying sizes, only 79.680% of the test sample predictions fell below the desired threshold. In the scenario where the FCNN and CNN were tasked with predicting the positions of two current dipoles contributing to a collective EEG signal, only 73.055% and 50.460% of the predictions, for the FCNN and CNN respectively, exhibited an accuracy smaller than 10 mm.

The accuracy, quantified by the Mean Euclidean Distance (MED), exhibited a diminishing trend as the output dimension 'd' of the FCNN increased. A similar trend was also observed in the CNN, with the MED notably decreasing when predicting the coordinates of a single dipole compared to predicting the spatial coordinates of two dipoles simultaneously.

CNN in Comparison to FCNN

When comparing the MED between the two networks, we observed that the FCNN yields more satisfactory results compared to the CNN. This preference for the FCNN's performance was the basis for our decision to proceed with the FCNN architecture when extending the EEG inverse problem to encompass the identification of dipole strength, as well as population strength and radius, for an area of active correlated current dipoles.

With this said, we emphasize that the architecture and hyperparameters of the FCNN were explored in greater detail compared to what was done for the CNN. While the FCNN provided the most satisfactory results in our work, we must acknowledge that a CNN might perform equally well or even better with a more optimized architecture and hyperparameters. Consequently, we cannot definitively conclude whether or not the performance of a neural network in predicting spatial coordinates is positively influenced by the preservation of spatial structure in the input data, as is the case for CNNs in general. Given more time, it would have been valuable to further investigate this possibility.

Error Variation

Include rmse in table? remove mse? Following the definition of MAE and RMSE, it can be shown that RMSE is always greater than or equal to MAE. Using these metrices together, offer valuable insights into the variation in error measurements within a dataset: the greater difference between them, the greater the variance in the individual errors in the sample. If the RMSE is equal to the MAE, then all the errors are of the same magnitude, which is considered ideal.

With respect to the MSD between predicted and target spatial coordinates, we observes that the RMSE was consistently somewhat higher than the measured MAE for all postional predictions across all network approaches. This discrepancy suggests the presence of relatively larger errors in the dataset, meaning that the predicitons of the networks and do not consistently align with the overall distribution of errors. This disparity between MAE and RMSE is most pronounced when localizing two dipoles simultaneously and when predicting the centers of dipole populations with varying radii and strengths. While this divergence indicates room for improvement in the models, it is important to acknowledge that some variation in neural network predictions is expected.

Predicting Dipole Depth

Predicting the depth of dipoles, i.e., the z-coordinates, for each approach to the inverse problem yielded varied results. As previously outlined in Chapter add chapter, deep brain structures often generate weak EEG signals

in clinical analyses, making their precise detection challenging. An intriguing aspect to explore within this EEG inverse modeling approach was whether the models, consistent with clinical practices, would encounter difficulties in accurately estimating the depth of dipole positions.

In all neural network approaches, a slightly higher MAE was observed in the z-coordinate. An exception to this trend occurred when predicting dipoles with varying strengths, where the y-coordinate exhibited the highest MAE among the spatial coordinates. One might be inclined to attribute the reason why the prediction of the z-coordinate did not exhibit the highest MAE in that specific scenatio to the fact that, in most of the corresponding data, dipoles in this particular case exhibited significantly larger strengths (with randomly assigned strengths ranging from 1 to 10 nAm) as opposed to when predicting dipoles with a constant strength of 1 nAm. In this context, the correlation between dipole depth and the strength of EEG signals, which is often present in clinical cases, becomes less prominent. However, this explanation is challenged when considering a scenario in which we predicted varying strengths for dipole populations. In this case, the z-coordinate once again demonstrated the highest MAE among the spatial coordinates, despite the increased strength of the dipoles. Furthermore, given that alternating coordinates exhibited the highest MSE, this argument is less substantiated. In terms of MSE, the x-coordinate consistently displayed the lowest value, while predictions for the y- and z-coordinates alternated as having the highest MSE. Thus, it is reasonable to conclude that the somewhat poorer predictions in the z-coordinate, as observed in some of the models, can be attributed to random factors rather than the actual depth of the dipole's location.

Predicting Dipole Strength and Population Radius

Include small, medium, large magnitude? Statistically significant? At least say something about what this error say about the prediction? When predicting dipole strength, we utilized two alternative versions of the FCNN, as presented in Chapter 9. In the first version of the network we focused on single dipoles, where the strength exhibited variation across the dataset. The primary objective of this network was to predict the spatial coordinates for each current dipole, alongside determining the magnitude of the current potential. Within this configuration, our model gave fulfilling spatial predictions, yielding a MED of 2.815 mm. The magnitude predictions exhibited an associated MAE_A of 0.539 nAm. This error corresponded to a relative error of 6.00% within the simulated range of 1-10 nAm.

In contrast, our second, more complex version of the alternative FCNN was tailored to address populations of acitve correlated current dipoles characterized by varying strengths and radii. In this scenario, we achieved a MED of 7.85 mm, and a MAE in magnitude measuring 0.33 nAm. Remark-

ably, the transition from the previous model to this more complex version led to a 5 mm increase in MED, while the MAE_A decreased to 0.33 nAm, which correspods to 3.6% of the target range.

These differences in model performance can be attributed to the inherent correlation between the prediction of radius and strength. The manner in which our simulated data is generated establishes a strong interdependence between these two variables. As a conserquence the model's optimization process will indirectly give more importance to reducing errors in radius and strength, even though we in our customized cost function, gained equal weighting to the target coordinates and the absolute error associated with the predicted radius and strength. In other words, when we reduce the error in radius prediction, it also reduces the error in strength prediction because they are strongly related. In contrast, the position of the dipole, as reflected in the Euclidean distance, is not correlated with strength or radius. Therefore, while reducing errors in strength or radius can indirectly benefit each other due to their correlation, this does not translate into a reduction in the error of the Euclidean distance.

In practice, this results in varying degrees of emphasis on Euclidean distance and absolute errors in strength and radius due to their strong correlation - which for the most complex version of the alternative FCNN, lead to an significantly improvement in MAE_A for the dipole strength. To achieve equal importance for all target values, we could have considered addressing these correlations in the cost function design by assigning weighting factors to determined the relative importance of each of the terms contributing to the total cost.

Predicting Multiple Dipoles

When instructing the neural networks to simultaneously output the positions of two current dipoles, we observed the highest positional errors, measured as the Mean Euclidean Distance, across all scenarios. However, the FCNN provided accurate predictions below the 10 mm threshold, whilst the CNN gave predictions just above this threshold. The increased positional error when predicting an additional dipole position can be attributed to various factors. Firstly, predicting dipoles in close proximity presents a greater challenge for neural networks. The increased challenge arises from the strong electromagnetic field interactions that occur when dipoles are located near each other. These interactions give rise to complex patterns, thereby necessitating the neural network to learn more intricate relationships.

To address these difficulties, one potential approach would be to manipulate the available dipole positioning, ensuring that dipoles are always separated by a minimum or constant distance. However, this approach might limit the network's ability to handle real-world scenarios, and its impact on prediction performance remains uncertain. Alternatively, employing a more

complex neural network architecture with a greater number of adjustable parameters to accommodate the data patterns is an option. Nevertheless, this also increases the risk of overfitting.

Despite the increased positional error compared to scenarios requiring fewer dimensions in the network's outputs, we found the results satisfying, given that the MED for the dipoles in the FCNN predictions remained below the 10 mm threshold.

10.2 Network Architecture

The selection of hyperparameters for the neural networks underwent an iterative process of trial and error, leading to the architecture used in the presented results chapters. A notable feature of the FCNNs is the deliberate use of various activation functions within the hidden layers. Following the initial linear transformation of input data, ReLU activation was employed, followed by the hyperbolic tangent in subsequent hidden layers. For the extended FCNNs predicting outputs with multiple units, we opted for the sigmoid activation function to deal with normalized target values, instead of the standard linear layer often used in regression problems.

The introduction of diverse activation functions across the network layers allows for varying degrees of non-linearity mappings. In the best case scenario, it can enhance the network's ability to generalize effectively to unseen data by capturing distinct patterns, offering a more comprehensive representation of the data. However, it is important to note that this diversity may have made the networks more complex than necessary. While increased flexibility can be advantageous for certain problems, it also raises the risk of overfitting.

Our primary objective in developing various neural network approaches was not to attain perfection, but rather to investigate the network's adaptability in addressing the EEG inverse problem across diverse levels of complexity. This complexity included variations in output strengths and radii of neural activity that generated the input EEG signal. In most of our approaches, we successfully achieved localization errors smaller than 10 mm.

However, in clinical contexts, there exists a considerable distinction between a localization error of 10 mm and 1.0 mm. It is important to underscore that an error of 1.3 mm, as observed in the case of the FCNN predicting dipole position alone, is significantly more accurate than the 8.71 mm error observed when predicting the position of two current dipoles. Both from a modeling perspective and in clinical practice, smaller errors are preferable. Nevertheless, it is essential to recognize that further diminishing the error beyond a certain point may lose clinical significance. While there may be a noticeable percentage improvement in the model's loss, within the context of cortical measurements, such improvements may not be practically

discernible. In other words, striving for additional model optimization can be beneficial up to a certain point, but beyond that, it may not necessarily result in substantial clinical advantages.

10.3 Data Set

In our neural network approach for localizing current dipoles, we employed the New York Head Model integrated into the LFPy Python Module to simulate EEG data. To enhance the realism of the data, we introduced normally distributed noise, characterized by a mean of 0 and a standard deviation amounting to 10% of the standard deviation observed in the simulated EEG recordings, into the final dataset used for both training and validation. In subsequent testing on noise-free data, we found that, in most cases, the neural networks produced predictions with positions falling below 10 mm.

One notable advantage of working with self-simulated data is the high degree of control it affords. This control allowed us to create data for specific scenarios, including the cases with two dipoles, dipoles featuring varying strengths, and populations of dipoles exhibiting varying radii. Having access to the correct answers for training and evaluation enabled us to effectively assess the performance of the neural networks. Furthermore, the New York Head Model, offering 74,382 discrete points for dipole source localization, facilitated the creation of an extensive dataset suited for both training and testing.

Nevertheless, it is essential to acknowledge that our networks have not been tested on real-world EEG recordings. Consequently, it is highly probable that the extent to which our networks faithfully capture the complexity, noise, and variability present in clinical EEG data will not align with the performance demonstrated on simulated validation and test data. This disparity is a well-documented limitation in the field of machine learning, where the performance achieved with both simulated and experimental data may not consistently translate directly to specific real-world scenarios. Consequently, it is expected that our framework's performance will be suboptimal when directly applied to EEG recordings from actual patients.

In future studies, there are valuable insights to be gained by exposing the networks to actual patient recordings, which would provide an opportunity to assess their performance in genuine clinical contexts. Nonetheless, as justed oulined it is highly likely that our network approaches may not perform optimally with real experimental data. In the event that these approaches is applied within an authentic clinical setting, such as for a patients requiring brain surgery, it may be plausible to construct patient-specific head models that can be integrated into our framework, subsequently training the networks with data extracted from the patients head models. Nonetheless, it is crucial to underscore that the effectiveness of such an approach would ne-

cessitate empirical testing within a real-world healthcare environment, and the degree of success in practice remains uncertain.

ReLU speed Could have used an other head model/ simulated data from other source to see how the model would have performed...? Something about LFPy. Easy to use or something??

10.4 Contextualizing: Existing Research on ConvDip

Numerous studies have explored the use of neural networks to address the EEG inverse problem. Among these, ConvDip, a convolutional neural network, stands out as a classification-based solution, that aims to predict the location of multiple dipole clusters along with the extent of these populations.

ConvDip tackled the EEG inverse problem using simulated EEG data, from dipole populations of different extent. The network demonstrated an ability to produce inverse solutions from single time points of EEG data and provided plausible results for real EEG recordings from human participants. The network was trained on samples containing 1 to 5 source clusters and proved effective in handling even more complex cases. It strived to predict the correct source size and localize sources at varying depths, a task achieved by categorizing source extents into five sizes.

To generate realistic training data, the researchers constructed a source model with 5,124 dipoles distributed across the cortex, simulating 31 recording electrodes and incorporating real noise from EEG recordings. Additionally, they employed an alternative head model for test data to prevent overoptimistic results. The training dataset contained 100,000 samples, while the test dataset comprised 1,000 samples.

In ConvDip's case, EEG data was interpolated into a 2D image grid preserving spatial information and without the introduction of new. The network's output vector contained 5,124 elements, corresponding to dipole positions in the source model. Performance was evaluated by comparing the network's active output nodes with the true distribution of dipoles, with node activation indicating the network's prediction of a dipole source at a specific location. The network's ability to estimate source extent was assessed by evaluating the proximity of neighboring outputted dipoles.

ConvDip demonstrated robust localization accuracy across different source depths, similar to our findings. However, it exhibited reduced accuracy in estimating source extent with increased depth. This specific relationship was not explored in our work. For single dipole populations, ConvDip achieved a mean localization error of 11.05 mm, somewhat higher than our FCNN's error of 7.85 mm when prediction the center of our constructed spherical populations. Notably, direct comparisons between these networks are intricate

due to variations in head models and dataset characteristics. Our training data featured more positions for localizing dipoles, allowing our FCNN to capture finer details for the specific head model. Furthermore, our evaluation dataset lacked noise, making ConvDip's performance on more complex, noisy data noteworthy.

ConvDip was tested on real EEG data from human subjects. While the results were promising, it most of all emphasized the importance of individual variability in head anatomy, suggesting individual neural networks to be trained based on specific MRI data for each subject. This approach, though effective, raised concerns regarding computational time. To address this, an individual transfer learning method was suggested, where a network trained on one subject's anatomy could be fine-tuned for subsequent subjects with new training data. This transfer learning method holds potential relevance for our own approach.

ConvDip represents one of the numerous neural network-based approaches for addressing the EEG inverse problem. Although our approach diverges from ConvDip in terms of methodology and architectural design, several overlapping research aspects have been explored.

10.5 Outlook

We have employed simulated EEG data from detailed neural simulations using the software LFPy 2.0 to train neural networks in localizing current dipoles. Building on this foundation, an exciting opportunity presents itself: identifying the specific type of neural activity responsible for a localized current dipole. To the best of our knowledge, neural networks have not yet been harnessed to discern the neural origin of EEG signals while also localizing these neural sources.

LFPy stands out as a versatile tool. Beyond facilitating the simulation of distinct point dipoles, it enables the simulation of various types of neural activity, resulting in specific dipoles. For instance, the type of synaptic input, whether it is excitatory or inhibitory, to a population of neurons, and the synaptic input's location (apical or basal) result in different current dipoles [21]. Furthermore, there is speculation that dendritic calcium spikes can be detected from EEG signals, potentially leading to exciting new possibilities for studying learning mechanisms in the human brain [43].

To decode different types of neural activity from EEG signals, we must first comprehend how diverse neural activities are mirrored in these signals. This understanding can be gained through simulations with LFPy, which provides a platform for simulating varying neural activities and their resulting EEG signals. This, in turn, opens the door to a more comprehensive exploration of the relationship between EEG signals and the underlying neural activity, through neural networks.

Bibliography

[1] Zeynep Akalin Acar and Scott Makeig. 'Effects of forward model errors on EEG source localization'. In: *Brain topography* 26 (2013), pp. 378–396.

[2] SMART-Servier Medical Art. *Equipment - EEG (4669) - Smart-Servier.png*. Licensed under Creative Commons Attribution-Share Alike 3.0 Unported. Accessed on [Insert Access Date]. 2023. URL: https://commons.wikimedia.org/wiki/File:Equipment_-_EEG_(4669)_--_Smart-Servier.png.

[3] SMART-Servier Medical Art. *Equipment - EEG (6199) - Smart-Servier.png*. Licensed under Creative Commons Attribution-Share Alike 3.0 Unported. Accessed on [Insert Access Date]. 2023. URL: https://commons.wikimedia.org/wiki/File:Equipment_-_EEG_(6199)_--_Smart-Servier.png.

[4] Vic Barnett, Toby Lewis et al. *Outliers in statistical data*. Vol. 3. 1. Wiley New York, 1994.

[5] Varun Bhatia. *Activation Functions: How Should the Neurons Trigger?* Medium. [Online]. Available: https://medium.com/analytics-vidhya/activation-functions-how-should-the-neurons-trigger-1349a383ffeb. 2020.

[6] Andrea Biasiucci, Benedetta Franceschiello and Micah M Murray. 'Electroencephalography'. In: *Current Biology* 29.3 (2019), R80–R85.

[7] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006.

[8] Bitbrain. *EEG Artifacts*. Bitbrain Blog. 2020.

[9] Edward B Bromfield, José E Cavazos and Joseph I Sirven. 'An introduction to epilepsy [Internet]'. In: (2006).

[10] Wikimedia Commons. *Action potential.svg*. Licensed under Creative Commons Attribution-Share Alike 3.0 Unported. Accessed on [Insert Access Date]. 2022. URL: https://commons.wikimedia.org/wiki/File:Action_potential.svg.

[11] Ray Cooper et al. 'Comparison of subcortical, cortical and scalp activity using chronically indwelling electrodes in man'. In: *Electroencephalography and clinical neurophysiology* 18.3 (1965), pp. 217–228.

[12] MR Delucchi, Bill Garoutte and Robert B Aird. 'The scalp as an electroencephalographic averager.' In: *Electroencephalography & Clinical Neurophysiology* (1962).

[13] Vincent Dumoulin and Francesco Visin. *A Guide to Convolution Arithmetic for Deep Learning*. FMILA, Université de Montréal and AIRLab, Politecnico di Milano. 2018. arXiv: 1603.07285. URL: https://arxiv.org/pdf/1603.07285.pdf.

[14] John S Ebersole. 'Defining epileptogenic foci: past, present, future'. In: *Journal of clinical neurophysiology* 14.6 (1997), pp. 470–483.

[15] Quasar Jarosz at English Wikipedia. *Neuron Hand-tuned.svg*. Licensed under Creative Commons Attribution-Share Alike 3.0 Unported. Accessed on [Insert Access Date]. 2009. URL: https://commons.wikimedia.org/wiki/File:Neuron_Hand-tuned.svg.

[16] Estimation lemma. *Estimation lemma — Wikipedia, The Free Encyclopedia*. [Online; accessed 22-May-2023]. 2023. URL: https://en.wikipedia.org/wiki/Multipole_expansion.

[17] Wulfram Gerstner et al. *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.

[18] Michele Giugliano, Mario Negrello and Daniele Linaro. *Computational Modelling of the Brain: Modelling Approaches to Cells, Circuits, and Networks*. Springer, 2022.

[19] Prashant Gupta. 'Regularization in Machine Learning'. In: *Towards Data Science* (Nov. 2017). URL: https://towardsdatascience.com/regularization-in-machine-learning-76441ddcf99a.

[20] Sparch Gupta. *The 7 Most Common Machine Learning Loss Functions*. URL: https://builtin.com/machine-learning/common-loss-functions (visited on 16/08/2023).

[21] Espen Hagen et al. 'Multimodal Modeling of Neural Network Activity: Computing LFP, ECoG, EEG, and MEG Signals With LFPy 2.0'. In: *Frontiers in Neuroinformatics* 12 (2018). Original Research Article, p. 92. ISSN: 1662-5196. DOI: 10.3389/fninf.2018.00092. URL: https://doi.org/10.3389/fninf.2018.00092.

[22] Lukas Hecker et al. 'ConvDip: A convolutional neural network for better EEG Source Imaging'. In: *Frontiers in Neuroscience* 15 (2021), p. 569918.

[23] Morten Hjorth-Jensen. 'Neural Networks'. In: *Department of Physics, University of Oslo, Norway* (Oct. 2022). [1] Department of Physics, University of Oslo, Norway
Department of Physics and Astronomy and Facility for Rare Ion Beams, Michigan State University, USA.

[24] Yu Huang, Lucas C Parra and Stefan Haufe. 'The New York Head—A precise standardized volume conductor model for EEG source localization and tES targeting'. In: *NeuroImage* 140 (2016), pp. 150–162.

[25] John David Jackson. *Classical electrodynamics*. 1999.

[26] Andreas Kleppe. 'IN5400 / IN9400 – Machine Learning for Image Analysis'. Lecture 3: Convolutional neural networks, Modification of slides made by Tollef Struksnes Jahren and Sigmund Johannes Ljosvoll Rolfsjord. 2022.

[27] Andreas Kleppe. *Lecture 3: Convolutional neural networks*. IN5400 / IN9400 – Machine Learning for Image Analysis. Feb. 2022.

[28] Juri D Kropotov. *Functional neuromarkers for psychiatry: Applications for diagnosis and treatment*. Academic Press, 2016.

[29] LeNail. 'NN-SVG: Publication-Ready Neural Network Architecture Schematics'. In: *Journal of Open Source Software* 4.33 (2019), p. 747. DOI: 10.21105/joss.00747. URL: https://doi.org/10.21105/joss.00747.

[30] Pankaj Mehta et al. 'A high-bias, low-variance introduction to machine learning for physicists'. In: *Physics reports* 810 (2019), pp. 1–124.

[31] Solveig Næss et al. 'Biophysically detailed forward modeling of the neural origin of EEG and MEG signals'. In: *NeuroImage* 225 (2021), p. 117467.

[32] Izaak Neutelings. *Neural Networks in TikZ*. Licensed under Creative Commons Attribution-ShareAlike 4.0 International License. Adapted by [Your Name or Username] for this work. 2021. URL: https://tikz.net/neural_networks/.

[33] Ernst Niedermeyer and FH Lopes da Silva. *Electroencephalography: basic principles, clinical applications, and related fields*. Lippincott Williams & Wilkins, 2005.

[34] NumPy Developers. *Numpy Corrcoef*. NumPy. 2023. URL: https://numpy.org/doc/stable/reference/generated/numpy.corrcoef.html (visited on 12/09/2023).

[35] Paul L Nunez, Michael D Nunez and Ramesh Srinivasan. 'Multi-scale neural sources of EEG: genuine, equivalent, and representative. A tutorial review'. In: *Brain Topography* 32 (2019), pp. 193–214.

[36] Paul L Nunez and Ramesh Srinivasan. *Electric fields of the brain: the neurophysics of EEG*. Oxford University Press, USA, 2006.

[37] Robert Plonsey and Dennis B Heppner. 'Considerations of quasi-stationarity in electrophysiological systems'. In: *The Bulletin of mathematical biophysics* 29 (1967), pp. 657–664.

[38] Rukshan Pramoditha. 'How to Choose the Right Activation Function for Neural Networks'. In: (Jan. 2022). URL: https://towardsdatascience.com/how-to-choose-the-right-activation-function-for-neural-networks-3941ff0e6f9c (visited on 10/06/2023).

[39] Shweta Saxena. *Activation Functions Compared With Experiments*. 2022. URL: https://wandb.ai/shweta/Activation%20Functions/reports/Activation-Functions-Compared-With-Experiments--VmlldzoxMDQwOTQ (visited on 10/06/2023).

[40] Mona Sazgar et al. 'EEG artifacts'. In: *Absolute Epilepsy and EEG Rotation Review: Essentials for Trainees* (2019), pp. 149–162.

[41] Sagar Sharma, Simone Sharma and Anidhya Athaiya. 'Activation functions in neural networks'. In: *Towards Data Sci* 6.12 (2017), pp. 310–316.

[42] David Sterratt et al. *Principles of computational modelling in neuroscience*. Cambridge University Press, 2011.

[43] Mototaka Suzuki and Matthew E Larkum. 'Dendritic calcium spikes are clearly detectable at the cortical surface'. In: *Nature communications* 8.1 (2017), p. 276.

[44] Adrian Tam. *Using Learning Rate Schedule in PyTorch Training*. Last Updated on April 8, 2023. 2023. URL: https://machinelearningmastery.com/using-learning-rate-schedule-in-pytorch-training/.

[45] Roman Tankelevich. 'Inverse Problem's Solution Using Deep Learning: An EEG-based Study of Brain Activity. Part 1'. In: *Preprint* (2019), pp. 1–15.

[46] Gert Van Hoey et al. 'EEG dipole source localization using artificial neural networks'. In: *Physics in Medicine & Biology* 45.4 (2000), p. 997.

[47] Wikipedia contributors. *Electroencephalography*. Accessed on 29 July 2023. 2023. URL: https://en.wikipedia.org/wiki/Electroencephalography.

[48] Wikipedia contributors. *Gradient descent — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/wiki/Gradient_descent. [Online; accessed 31-May-2023]. 2023.

[49] Kai Zhang and Minxia Luo. 'Outlier-robust extreme learning machine for regression problems'. In: *Neurocomputing* 151 (2015), pp. 1519–1527.