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

Spring 2023

Kamilla Ida Julie Sulebakk

Localization and identification of Neural Sources from simulated EEG Signals

Acknowledgements

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

Contents

Introduction

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

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

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

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

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

0.1 Motivation

0.2 Goal and Objectives

0.3 Structure of the Thesis

Chapter 1

Background

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

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

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

Overall, the physics of the neuron, EEG, and source localization are all important components of neurobiology that have contributed to our understanding of the nervous system and its functioning. By combining knowledge of the physical principles of neural signaling with advanced analytical tech-

niques, researchers are able to gain valuable insights into the underlying neural processes that give rise to behavior and cognition.

1.1 Introduction to Neuroscience

1.2 Head Models and Multicompartmental Modeling

1.3 Currents and Potentials in the Brain

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

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

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

1.4 Electroencephalograpy

1.5 EEG Forward modeling

1.6 The Inverse Problem

Chapter 2

Electroencephalograpy

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

Electroencephalography (EEG) is a non-invasive technique that has been used for almost a century to study the electrical potentials in the human brain. It has remained one of the most important methods for investigating the brain's activity, with significant applications in both neuroscientific and clinical research [1].

An EEG signal is believed to originate from large numbers of synaptic inputs to populations of geometrically aligned pyramidal neurons [2].The signal can be understood as a signature of neural activities that are generated from synaptic inputs to cells in the cortex. Synaptic inputs on its side are electrical or chemical signals that are being transmitted from one neuron to another, causing changes in the membrane potential of the neurons. In other words, neurons are specialized to pass signals, and synapses are the structures that make this transmission possible.

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

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

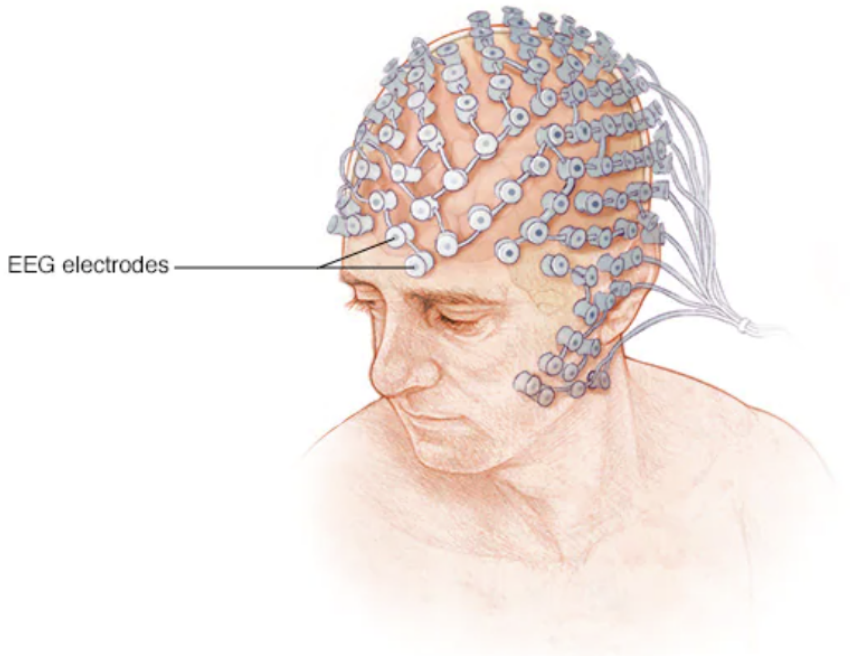

Figure 2.1: Illustration of the EEG method.

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

2.1 Forward modeling of EEG signals

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

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

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

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

where $\mathbf{J}$ is the electric current density in extracellular space, σ is the extracellular conductivity and ϕ is the extracellular electric potential [3]. For simple, symetric head models, the Poisson equation can naturally be solved analytically. However, as for the New York Head which is a complex model that we will be utialize in this thesis, we will be using the numerical method known as Finite Element Method (FEM) when solve the quation.

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

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

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

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

where $\mathbf{r}_n$ is the location of the tranmembrane current I_n, N is the number fof transmembrane currents and σ is the extracellular conductivity [3].

2.2 Multicompartmental modeling

The potential over the membrane, V_m, in long neurons with multi-branched dendrites varies depending on whether the potential is measured in the soma

or at the tip of a sital denrite [3]. Multicompartmental (MC) models are models that account for this spatial variabilty in V_m. In such models, the neural morphology commonly is represented as isopotential cylindrical compartmens, with lengths and diameters derived from reconstructed morphologies, connected with resistors [4]. Single values of V_m can then be computed for each individual compartment, as depiced in Figure 2.2.

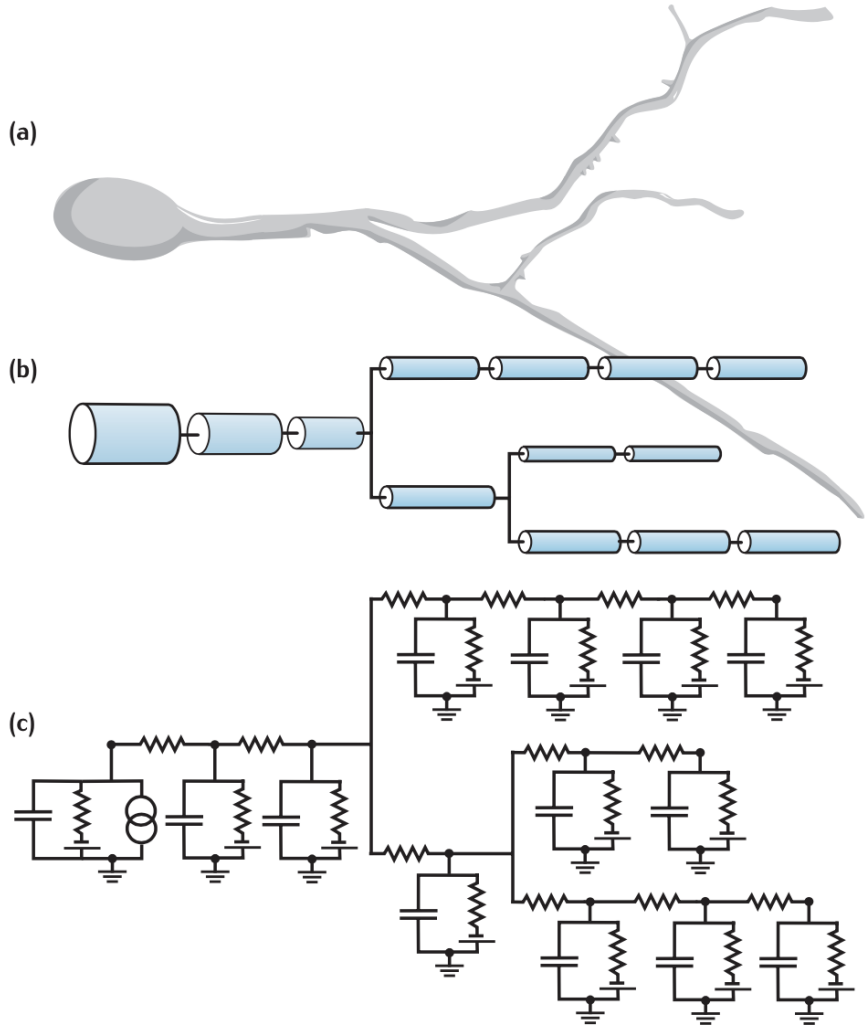

Figure 2.2: A diagram of the development of a multi-compartmental model. (a) The cell morphology is represented by (b) a set of connected cylinders. An electrical circuit consisting of (c) interconnected RC circuits is then built from the geometrical properties of the cylinders, together with the membrane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

The fundamental equation for MC models is given as:

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

where ....

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

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

and

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

2.3 Volume conductors

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

2.4 Current Dipole Approximation

EEG signals are generated from synaptic inputs to cells in the cortex. Synaptic inputs are electrical (or chemical) signals that are being transmitted from one neuron to another, causing changes in the membrane potential of the neurons. In other words, neurons are specialized to pass signals, and synapses are the structures that make this transmission possible.

When calculating extracellur potentials, V_e, at large distances from the underlying current sources, one is typically benefitted with using the current-dipole approximation. This approximation is justifided by the fact that a neuron's contribution to V_e becomes increasingly dipolar with increasingly distance and is commonly utialized when simulating EEG signals.

We know that electrical charges can create current multipoles, depending on coordinates and symmetry of the charge distribution [5]. Analogous,

the combination of current sinks and sources set up such charge multipoles. When the distance R from the center of the volume to the recording point is larger that the distance from the volume center to the most peripheral source, multipole expansion can be used [6]. When utializing the multipole expansion theorem equations 2.6 can be expressed as follows:

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

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

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

where we have substituded for C_{dipole} in terms of other properties. $\mathbf{p}$ denotes the current dipoile moment in a medium with conductivity σ. $R = |\mathbf{R}| = |\mathbf{r}-\mathbf{r}_p|$ is the distance between the current dipole moment at $\mathbf{r}_p$ and the electrode lokation $\mathbf{r}$. Finally θ represents the angle between $\mathbf{p}$ and $\mathbf{R}$. This equation is known as the dipole approximation and is a precise approximation for calculating extracellular potential, given that R is much larger than the dipole length $d = |\mathbf{d}|$, like in the case of EEG [3].

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

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

In figure 2.3 we have provided a simulation of the extracellular potential generated by a neuron in response to a single synaptic input, where the

spatial distribution of membrane current was explicitly taken into consideration. Based on the figure, it is apparent that the distribution of electric charge in the extracellular potential of the neurons surroundings exhibits distinct dipole patterns when observed from a greater distance.

2.5 Head Models

The analytical formula presented in Equation 2.8 was dereived based on the assumption of a constant tissue conductivity, denoted as σ. However, because the EEG signals is measured outside the head, it is affected by the different conductivities og the brain, verebrospinal fluid (CSF), skull and scalp. Depending on problem, it may be important to keep in mind that EEG signals, in general, are significantly affected by biophysically details of the head. For instance, the conductivity of the cerebrosipinal fluid exhibits a conductivity of approximately 1.7 S/m, while the conductivity of the scull and scalp is approximately 0.01 S/m and 0.5 S/m, respectively. These conductivity variations highlight the need for more comprehensive and realistic models of the head, known as head models, which take into account such conductivity variations. In addition to acount for the variations in conductivity across different regions of the head, head models also takes into account that the EEG signal from a neuronal population will depend on whether the population is located in a *sulcus* or a *gyrus* [3]. Said with other words, by integrating biophysical details into models, a deeper understanding emerges regarding the impact of various tissues on the distribution of extracellular potential, which in turn, can lead to improvements in the accuracy of EEG signal analysis.

2.5.1 The New York Head

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

Especially important for electrode locations outside of the brain, such as EEG, is the knowledge about how the electrical potentials will be affected by the geometries and conductivites of the various parts of the head [3]. A model that takes these detailes into account is the New York Head Model. The model is based on anatomical and electrical characteristics of 152 adult human brains and is solved for 231 electrode locations.

The New York Head Model is a computer model of the human head used to simulate the electrical activity of the brain. It was created by the Electrical Geodesics Incorporated (EGI) in 2004, and is based on the anatomical and electrical characteristics of the head of a typical adult human. The model consists of a three-dimensional (3D) representation of the head and brain, with detailed information on the geometry and electrical properties of the different tissues and structures within the head. The model

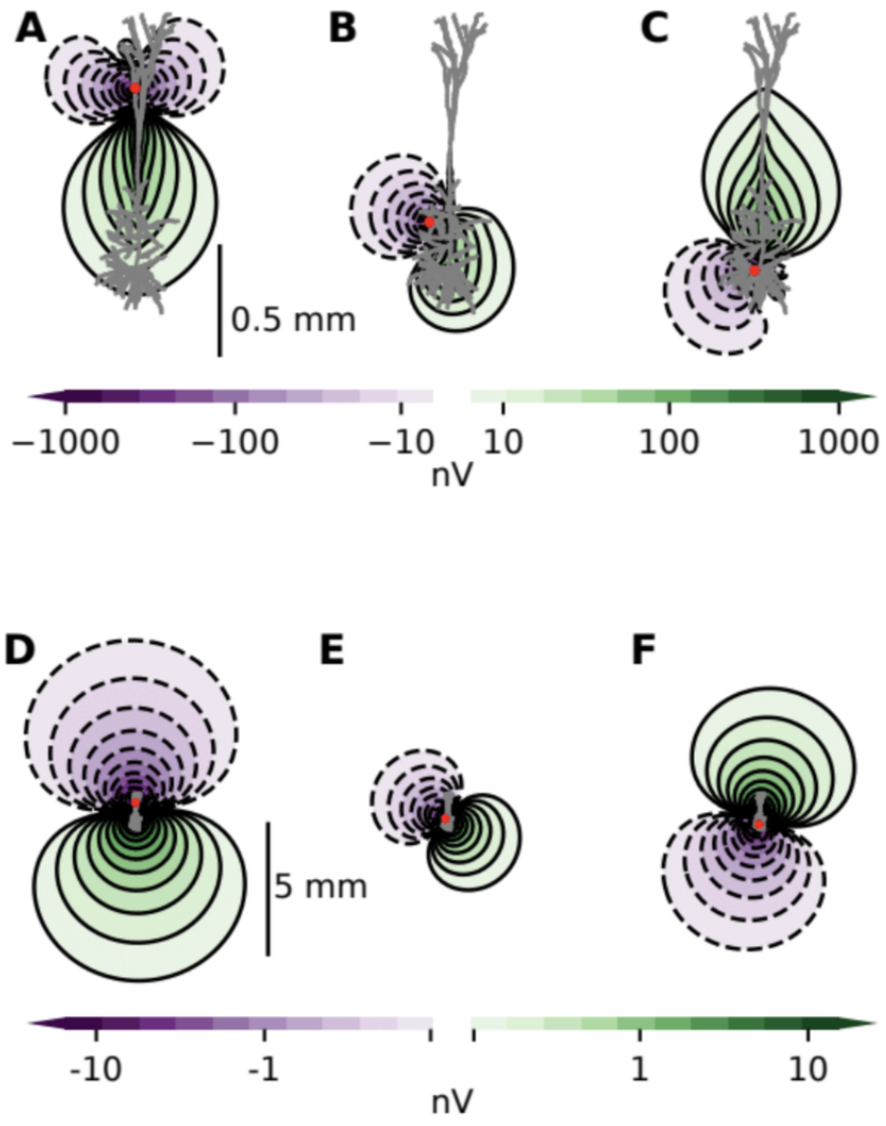

Figure 2.3: Source and explanation.

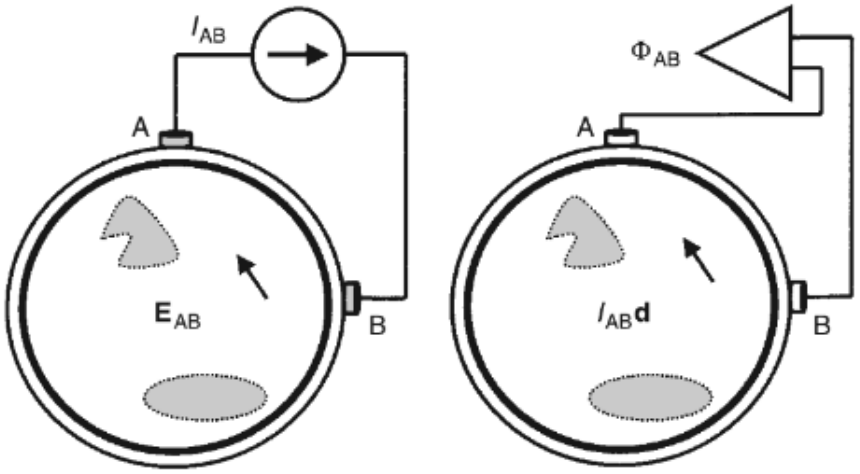

Figure 2.4: A caption here is needed.

includes the scalp, skull, cerebrospinal fluid, gray matter, and white matter. The electrical properties of each of these tissues, such as conductivity and permittivity, are also included in the model.

The model was developed to be used for, and improve the accuracy of EEG source localization [7]. To generate predictions of the EEG signals recorded from different scalp locations in response to a given set of source currents, the New York head model uses the lead field matrix (SOURCE), which is a mathematical representation of the relationship between the electrical activity in the brain and the electrical potentials recorded on the scalp.

The lead field matrix is constructed by taking advantage of the reciprocity theorem that states that knowledge of the current density through a volume conducter caused by an injection of current between two stimulating electrodes completely specifies how those same recording electrodes pick up potentials caused by dipole sources in the volume conducter. If one suppose that a pair of stimulating electrodes is placed at locations A and B on the scalp as provided in figure 2.4, an external current source will cause current to flow from electrode B through the brain, and all the way to electrode A. However, due to the geometry, inhomogeneity, and ansisotropy of the head, the current density will vary with location. The amount of current that pass through the brain depends, to a great extent, on the location of the electordes. In general, the brain currents will decrease with decreasing distance between electrodes. Thus, for a fixed pair of electrodes, the lead field vectors can be calculated as a function of position throughout a volume conducter. At each location, the orientation of the lead field vector L_{AB} is the orientation of the dipole source that produces the largest potential difference between the electrodes. The lead field matrix, $\boldsymbol{L}$ is given as:

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

where I is the injected current at the electode locations and E is the resulting electric field in the brain [3]. Moreover, the precise link between a current dipole moment p in the brain and the resulting EEG signals Φ is then related to the lead field matrix as follows:

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

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

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

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

The cortex matrix consists of 74382 points, which refer to the number of possible positions of the dipole moment in the cortex. When generating our data set, we will for each sample randomly pick the position of the dipole moment, such that one sample corresponds to one patient. In our head model we are considering 231 electrodes uniformly distributed across the cortex, meaning that each EEG sample will consist of this many signals for each time step. However, we are not interested in the time evolution of the signals as this does not affect nor say anything about the position of the dipole moment, and we therefor simply pick out the EEG signals for when $t = 0$ (note that the choice of time step could have been randomly picked). Our final design matrix will then consist of 1000 rows, corresponding to each patient, and 231 columns also refereed to as features, representing the signal of each electrode. The final output we are trying to predict is then the one dimensional vector with length 1000, where each element consists of the x-, y- and z- position of the dipole moment. An example of how the input EEG signals may look like is given in appendix A, where we also have marked the dipole moment with a yellow star.

2.6 The Inverse Problem and Source Localization

Chapter 3

Machine Learning and Neural Networks

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

3.1 Machine Learning and its Fundational Principles

"Machine Learning is a subfield of artificial intelligence with the goal of developing algorithms capable of learning from data automatically" [8]. The typical machine learning (ML) problems are addressed using the same three elements. The first element is the dataset $\mathcal{D} = (\mathbf{X}, \mathbf{y})$ where $\mathbf{X}$ commonly is refered to as the design matrix, and consists of independent variables, and $\mathbf{y}$ is a vector consisting of dependent variables. Next, we have the model itself, $f(\mathbf{x}; \boldsymbol{\theta})$. The ML model can be seen as a function used to predict an output from a vector of input variables, i.e. $f : \mathbf{x} \to y$ of the parameters $\boldsymbol{\theta}$. Finally, the third element, allows us to evaluate how well the model performs on the obervations $\mathbf{y}$. This element is known as the cost funtion $\mathcal{C}(\mathbf{y}, f(\mathbf{X}); \boldsymbol{\theta})$.

3.1.1 Fitting a Machine Learning Model

The first step in "fitting" a machine learning model, is to randomly split the dataset $\mathcal{D}$ into train and test sets. This is done in order to make a model compatible with multiple data sets. The size of each set commonly depend on the size of the data set avaible, however a rule of thumb is that the majority of the data are partitioned into the trainng set (e.g., 80%) with the remainder going into the test set [8].

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

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

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

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

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

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

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

3.1.2 Gradient Descent and Its Variants

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

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

for a sufficiently small learning rate η, we are always moving towars a minimum, since $F((w)_n) \geq F((w)_{n+1})$ [9]. After each update, the gradient is recalculated for the updated weight vector $\mathbf{w}$, and the process is repeated

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

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

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

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

Hence, with this iterative process, it is hoped that the sequence $(\mathbf{x}_n)$ converges to the desired local minimum [9].

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 [10].

Stochastic Gradient Descent

The method of Stochastic Gradient Descent (SGD) allows us to compute the gradient by randomly selecting subsets of the data at each iteration, rather than using the entire dataset [10]. The update can be written as:

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

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

Stochastic Gradient Descent with Momentum

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

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

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

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

3.2 Neural Networks

Neural networks are a distinct class of nonlinear machine learning models capable of learning tasks by observing examples, without requiring explicit task-specific rules [11]. The models mimics the way bilogical neurons trasmit signals, with interconnected nodes known as neurons that communicate through mathematical functions across layers. The layers in neural networks contain an arbitraty number of neurons, where each connection is represented by a weight variable.

The network gathers knowledge by detecting relationships and patterns in data using past experiences known as training examples. These patterns are further updated by the usage of appropriate activation functions and finally presented as the output [12]. A neural network consits of many such neurons stacked into layers, with the output of one layer serving as the input for the next. Typically, the neural networks are built up of an input layer, an output layer and layers in between, called hidden layers. In figure 3.1 we have provided the basic architecture of neural networks. Here nodes are depiced as circular shapes, while arrows indicate connections between the nodes.

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

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

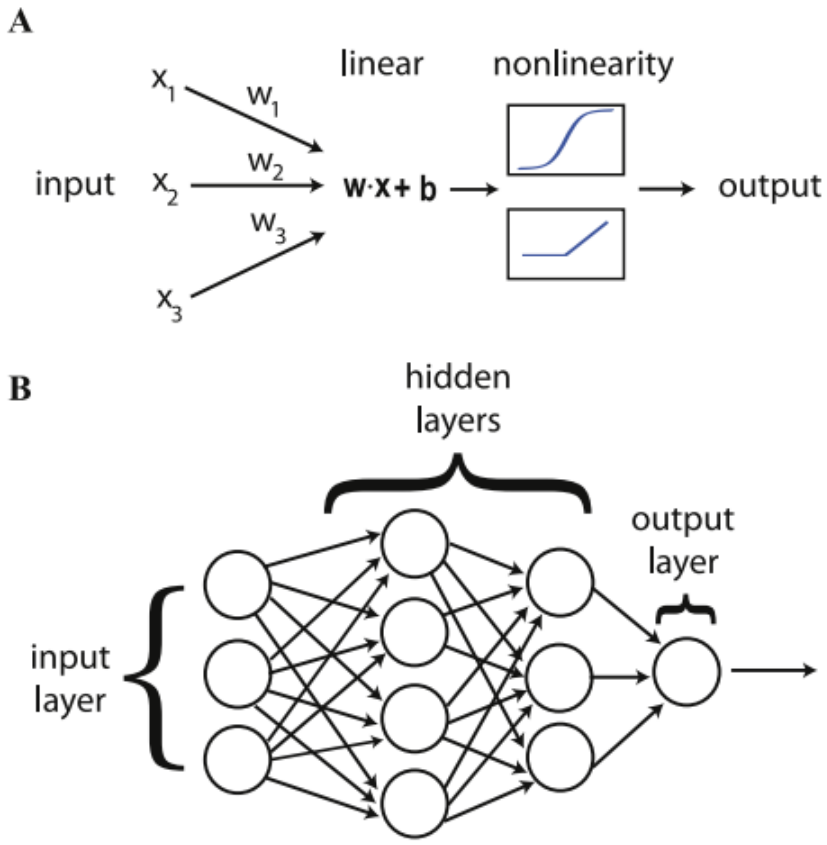

Figure 3.1: **(A)** The fundamental structure of neural networks comprises simplified neuron units that perform a linear operation to assign different weights to inputs, followed by a non-linear activation function.**(B)** These neuron units are organized into layers, where the output of one layer serves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

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

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

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

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

3.2.1 Activation functions

Without activation functions, a neural network would essentially be a linear model, capable only of representing linear relationships between inputs and outputs. While the linear transformations occurs within individual neurons through the weighted sum of inputs, the introduction of non-linear activation functions allows the networks to capture complex relationships and patterns. With other words, activcation functions are important components of neural networks, that help the network learn by making sense of non-linear and complex mappings between input- and corresponding output values. The functions are applied at every node in the hidden layers and the output layer [13].

Activation functions in neural networks draw inspiration from the behavior of neurons in the brain. Similar to how neurons respond to incoming electrical signals, activation functions determine whether a neuron in a neural network should be activated or not based on the strength of the input. If the input exceeds a certain threshold, the neuron "fires" or becomes activated, otherwise it remains inactive [14]. By introducing nonlinearity, activation functions enable neural networks to model complex, nonlinear relationships in data.

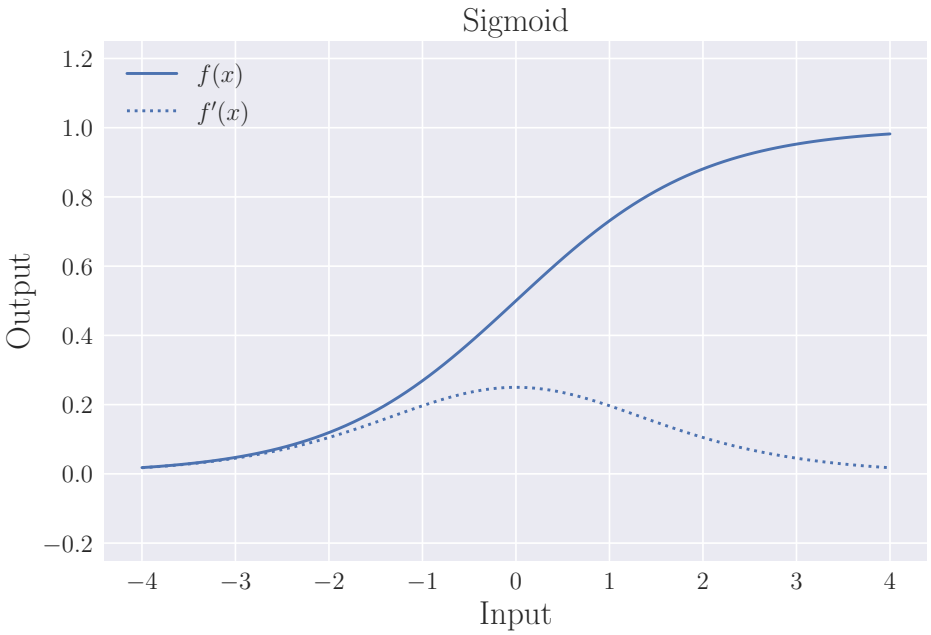

Figure 3.2: Sigmoid activation function.

Sigmoid

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

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

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

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

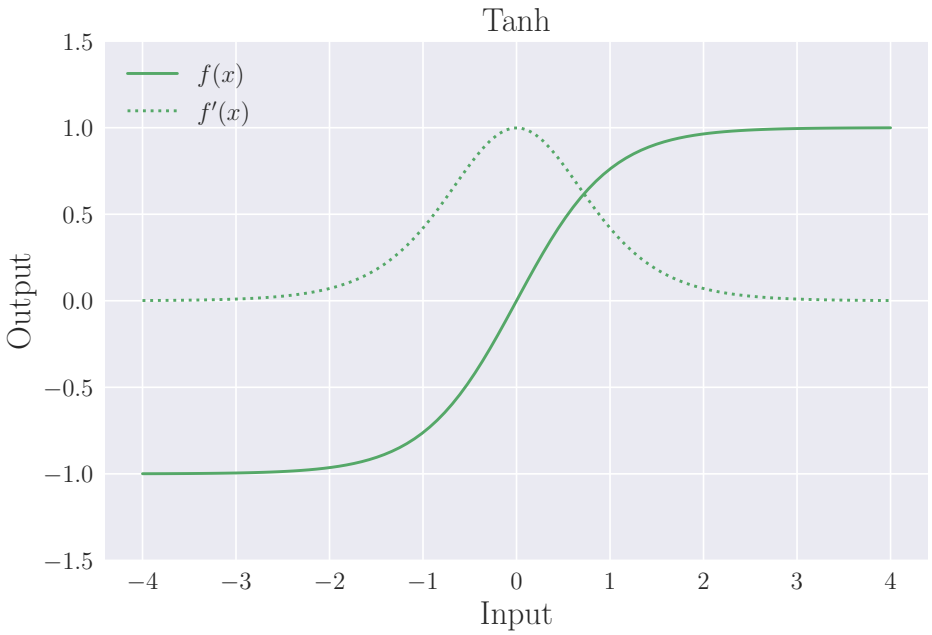

Figure 3.3: Hyperbolig tangent activation function.

Hyperbolic Tengent

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

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

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

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

Rectified Linear Unit

The Rectified Linear Unit (ReLU) activation function is widely recognized for its speed, high performance, and generalization capabilities [15]. Compared to the Sigmoid and Hyperbolic Tangent functions, ReLU may seem relatively simple, which contributes to its computational efficiency. The function can be mathematically defined as:

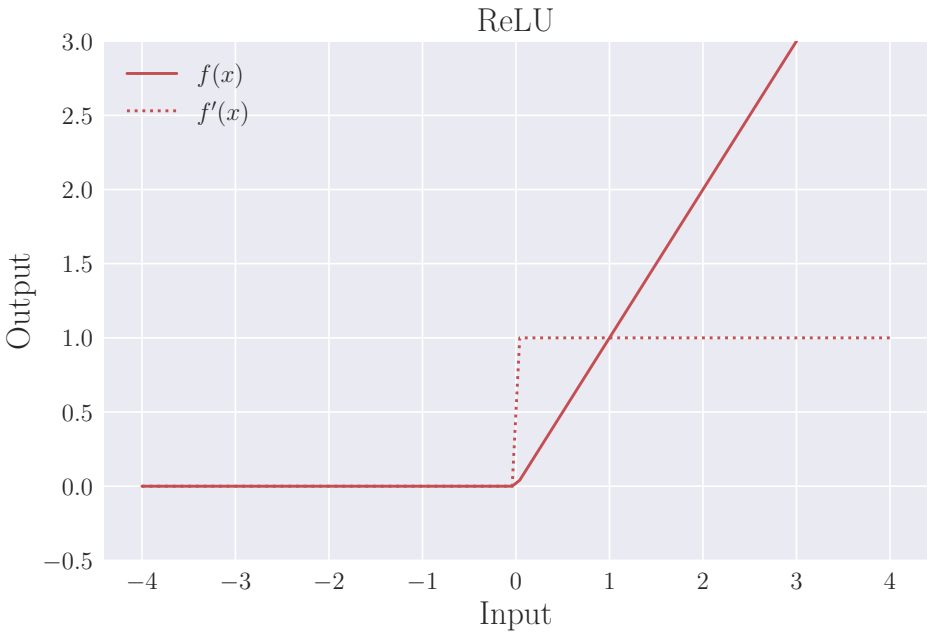

Figure 3.4: ReLU tangent activation function.

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

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

Back propagation algorithm

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

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

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

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

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

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

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

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

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

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

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

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

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

With these four equations in hand we can now calculate the gradient of the cost function, starting from the output layer, and calculating the error of each layer backwards. We then have a way of adjusting all the weights and biases to better fit the target data. The back propagation algorithm then goes as follows:

1. **Activation at input layer:** calculate the activations a_i^1 of all the neurons in the input layer.

2. **Feed forward:** starting with the first layer, utilize the feed-forward algorithm through **??** to compute z^l and a^l for each subsequent layer.

3. **Error at top layer:** calculate the error of the top layer using equation 3.15. This requires to know the expression for the derivative of both the cost function $C(\boldsymbol{W}) = C(\boldsymbol{a}^L)$ and the activation function $f(z)$.

4. **"Backpropagate" the error:** use equation 3.17 to propagate the error backwards and calculate δ_j^l for all layers.

5. **Calculate gradient:** use equation 3.16 and 3.18 to calculate $\frac{\partial C}{\partial z_i^l}$ and $\frac{\partial C}{\partial w_{ij}^l} = \delta_i^l a_j^{l-1}$.

6. **Update weights and biases:**
   $w_{jk}^l = w_{jk}^l - \eta \delta_j^l a_k^{l-1}$
   $b_j^l = b_j^l - \eta \delta_j^l$

Initialization of weights and biases

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

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

The Inverse Problem

Computational neuroscience is a field that aims to understand the principles underlying information processing in the brain using mathematical and computational tools. The inverse problem in EEG, which involves estimating the location and strength of electrical sources in the brain based on measurements of electrical activity on the scalp, is a key challenge in computational neuroscience. Machine learning techniques, including feedforward neural networks, have been used to address this problem by learning to map

the measured EEG signals to estimates of the underlying electrical sources in the brain.

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

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

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

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

Chapter 4

Results

As mentioned in chapter 1, an important topic in EEG signal analysis is the inverse problem of going from measured EEG signals to localized equivalent current dipoles, so-called source localization. In this chapter we will present the training and performance of the neural networks presented in chapter 4. Section ... and ... deal with training of the simple feed forward neural network and presenting its results, while section ... will discuss how a convolution neural network can be used to obtain the same results. But first, we will take a look at the dataset being feed to the different networks.

4.1 Simulation of EEG Signals

The cortex matrix of the New York Head Model (NYHM) consists of 74382 points, which refer to the number of possible positions for localization of dipole sources. We will train the neural networks using a dataset of self-simulated EEG measurements that correspond to the electromagnetic fields generated by dipole sources. These sources will have randomly selected positions within the cortex matrix. However, to simplify the problem, the stengths of single dipoles (amplitude) are set to 10^7 nA μ m. Moreover, in the cases of single dipole source localization, the direction of the dipole moment is always rotated so that it is normal to the cerebral cortex. In some cases this will result in a dipole moment pointing perpendicular to the skull (directed towards an EEG electorde), while in other cases, due to the structure of the cortex, the dipole moment will point back into the cortex (but eventually towards an EEG electorde). The reason for this is that the human cortex is strongly folded, and the contribution to the EEG signal from a neural population (dipole moment) will depend on whether a dipole is located in a sulcus or a gyrus (source: BookTVN).

4.1.1 Effect of dipole location and orientation on EEG signals

As shown by/ discussed in .... (source: BookTVN) EEG signals are relatively insensitive to small changes in the *location* of neural current dipoles. Even though the intuitive thought might be that neurons in the upper cortical layers dominate the EEG due to the closer distance to the EEG electrode compared to neurons in the lower cortical layers, such differences in location acutually does not make conciderable differences. This finding can be explained by fact that the low conductivity of the skull generates a certain spatial low-pass filtering, that .... .

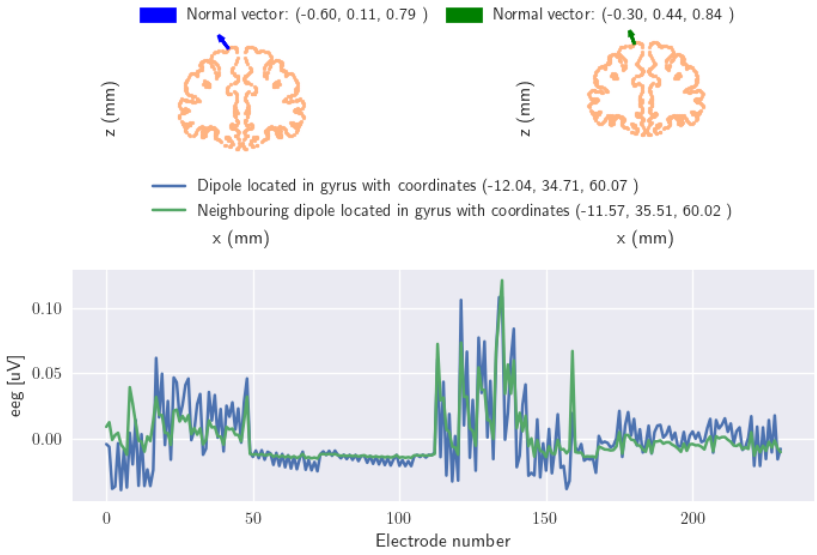

Figure 4.1: EEG signal for neighbouring dipoles.

However, in order to decribe the effect of the *orientation* of the dipoles relative to the EEG electrodes, we have in Figure 4.2 provided the EEG signals from two manually chosen dipole locations in the New York head model. The two dipoles illusrated are located in a gyrus and in a sulcus, both providing a different EEG outcome. In general, the EEG signal contribution from a single current dipole is maximized if the dipole is located in a gyrus, perpendicular oriented. Such a case is depiced in Figure 4.2B. However, if we place a dipole in a sulcus, again with perpendicular orientation, we can observe a substantial EEG contribution, but in contrast to the dipole in the gyrus we notice a more dipolar pattern 4.2C.

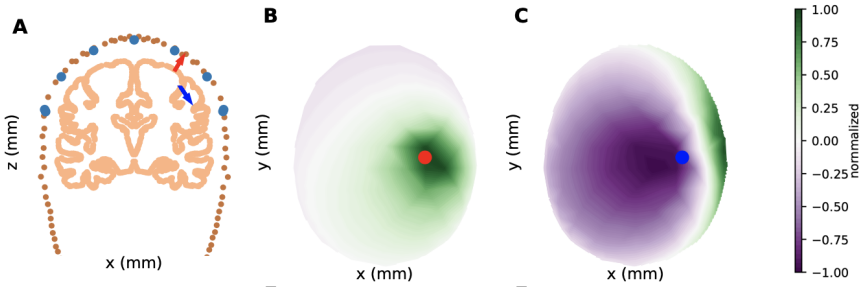

Figure 4.2: A: Two manually chosen dipole locations in the New York head model, located in a gyrus (red) and a sulcus (blue). The head model is seen from the side (x, z-plane). EEG electrode locations close to the chosen cross section plane are marked in light blue. The available dipole locations close to the cortical cross section effectively draw an outline of the cortical sheet, and are marked in pink. The current dipole moment was in all cases 10^7 nAμm. B: Interpolated color plot of EEG signal from the dipole in gyrus, seen from the top (x, y-plane). The plotted EEG signal is normalized but the maximal value was 1.1 μV. C: Interpolated color plot of EEG signal from the dipole in sulcus. The plotted EEG signal is normalized but the maximal value was 0.7 μV. (source: BookTVN)

4.1.2 Noise

As for all experimental data, real EEG recordings contain noise. Artifacts are signals recorded by EEG but with a origin different from those generated by human brain activity. As some artifact may mimic true epileptiform abnormalities or seizures, awereness of artifacts and methods for distinguishing such signals from brain waves is highly important (`https://link.springer.com/chapter/10.1007/978-3-030-03511-2_8`).

There are two different dypes of artifacts, classified according to their origin. Physiological artifacts originate from the patient itself, where the most usual ones are ocular activity, muscle activity, cardiac activity, perspiration and respiration. Technical artifacts, on the other hand, is generated from the environment of the patient, such as cable and body movements or electromagnetic interferences. (`https://www.bitbrain.com/blog/eeg-artifacts`).

Filtering techniques are usually utilized in order to remove artifact from EEG before analyzation of the recordings. But, when it comes to the simulated EEG data of ours, we are in no need to remove such noise, as there simply is none. The simulated EEG data can be understood as already filtered data that has undergone preprocessing steps, to ensure a high signal-to-noise ratio (`https://en.wikipedia.org/wiki/Signal-to-noise-ratio`). Moreover we understand the simulated data as an averaged measure of the typical EEG time series. However, in order to avoid overfiting and for other tecnical detailes, we do need to ass noise to tha data before feeding it to

the neural network. Therefore, to the final dataset of ours we add normally distributed noise of 10 %, with mean .... and standard deviation ... .

4.2 Localizing Single Dipole Sources

4.2.1 The dataset

The data set used to train a simple feed forward neural network consists of 10 000 rows, where each row corresponds to one sample, or let us say - one patient. Within the data set we have 231 columns, also referred to as features, representing the EEG measure at every recording electrode. Thus, we are left with a design matrix of size 10 000 x 231.

An example of how the input EEG data may look like for one sample (before adding noise) is provided in figure 4.3. The figure illustrates the EEG result from a sample containing a single current dipole source at a random position within the celebral cortex. As also can be seen from the characteristic dipolar pattern, the dipole is located in a sulcus. The EEG measure is seen from both sides (x-, z-plane and y-, z-plane) and above (the x-, y-plane). EEG electrode locations are presented as filled circels, where the color of the fill represents the amplitude of the measured signal for the given electrode. The position of the current dipole moment is marked with a yellow star. As can be read off from the figure, the EEG signals, for this given sample, range from between -1 to 1 μV.

4.2.2 Validation accuracy

In Figure 4.6 we have provided the validation accuray, using mean squared error (MSE) and the coefficient of determination (R2-score).

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

$$MSE(\hat{y}, \hat{\tilde{y}}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2 = \frac{1}{3}\sum_{i=1}^{3}((x-\tilde{x})^2 + (y-\tilde{y})^2 + (z-\tilde{z})^2) \quad (4.1)$$

The coefficient of determination is given as follows:

$$R^2(\hat{y}, \tilde{\hat{y}}) = 1 - \frac{\sum_{i=0}^{n-1}(y_i - \tilde{y}_i)^2}{\sum_{i=0}^{n-1}(y_i - \bar{y})^2}, \quad (4.2)$$

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

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

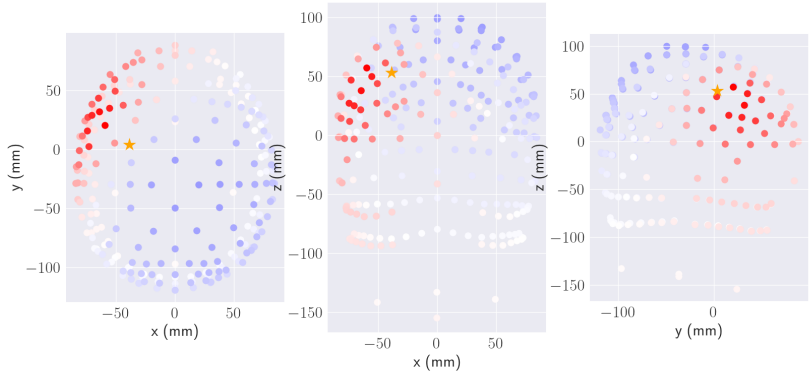

Figure 4.3: EEG for a sample containing one single current dipole source at a random position within the celebral cortex. The EEG measure is seen from both sides (x-, z-plane and y-, z-plane) and above (the x-, y-plane). EEG electrode locations are presented as filled circels, where the color of the fill represents the amplitude of the measured signal for the given electrode. The position of the current dipole moment is marked with a yellow star.

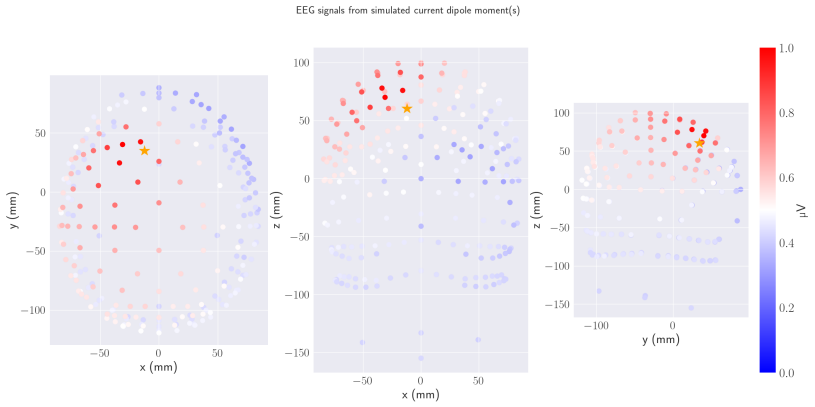

Figure 4.4: The EEG data of a randomly picked sample. As for all samples, 10 percent of normally distributed noise has been added to the original signal.

4.3 Convolution Neural Network Approach for localizing single dipole sources

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

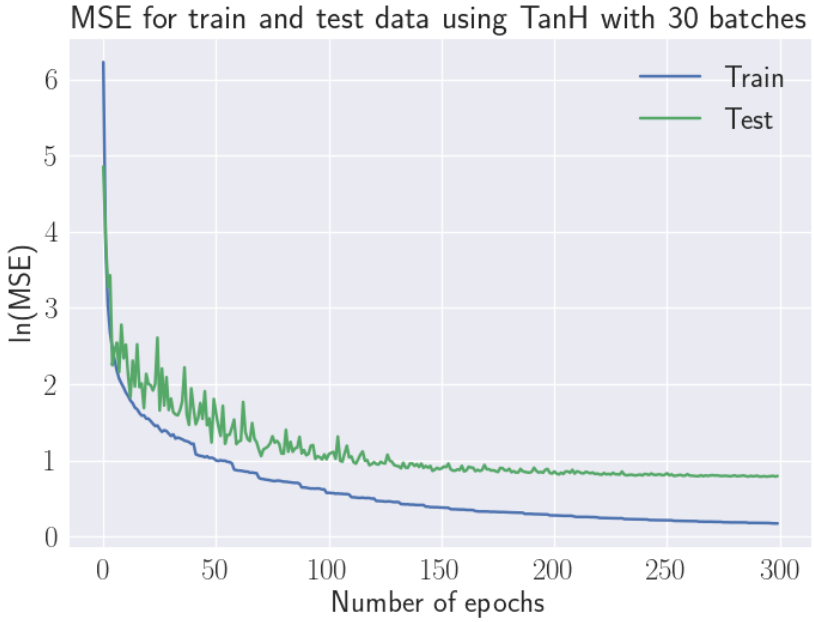

Figure 4.5: The validation accuracy for the simple Feed Forward Neural Network with 50 000 samples and tanh as activation function.

4.4 Region of Active Correlated Current Dipoles

Some results for the prediction of the size and location of current dipole populations.

Printed in terminal:
Epoch 9898/9999 | Train: 0.187 | Test: 4.275
Epoch 9899/9999 | Train: 0.184 | Test: 4.288
Epoch 9900/9999 | Train: 0.201 | Test: 4.279
Target: tensor([-1.0800, -1.9594, 0.4290, 11.0140])
Predicted: tensor([-1.1171, -1.9642, 0.4575, 16.5920])
Target: tensor([-6.7642e-02, 1.5426e+00, -1.0356e-02, 1.5576e+01])
Predicted: tensor([-0.3908, 1.4285, -0.1167, 15.9222])
Target: tensor([-0.6671, -1.0569, 1.8694, 7.1385])
Predicted: tensor([-0.7248, -1.0950, 1.9903, 6.2405])

4.5 Localizing Multiple Dipole Sources

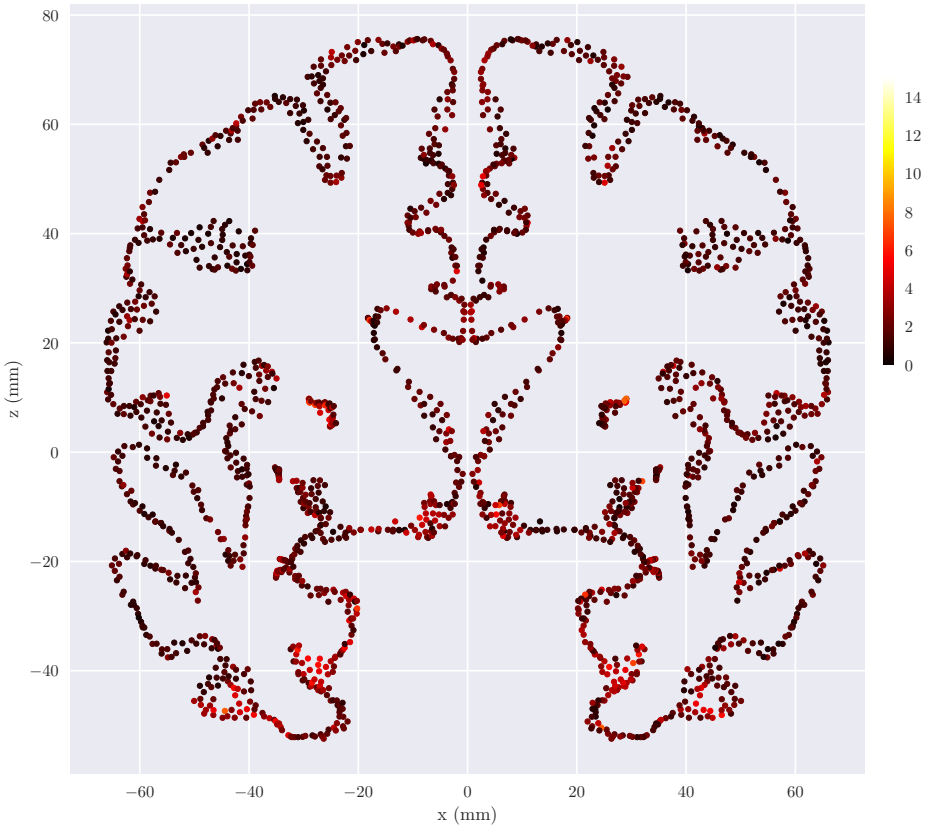

Figure 4.6: Mean absolute error of the network for each dipole location in the cortical y-plane cross section at y = 0.

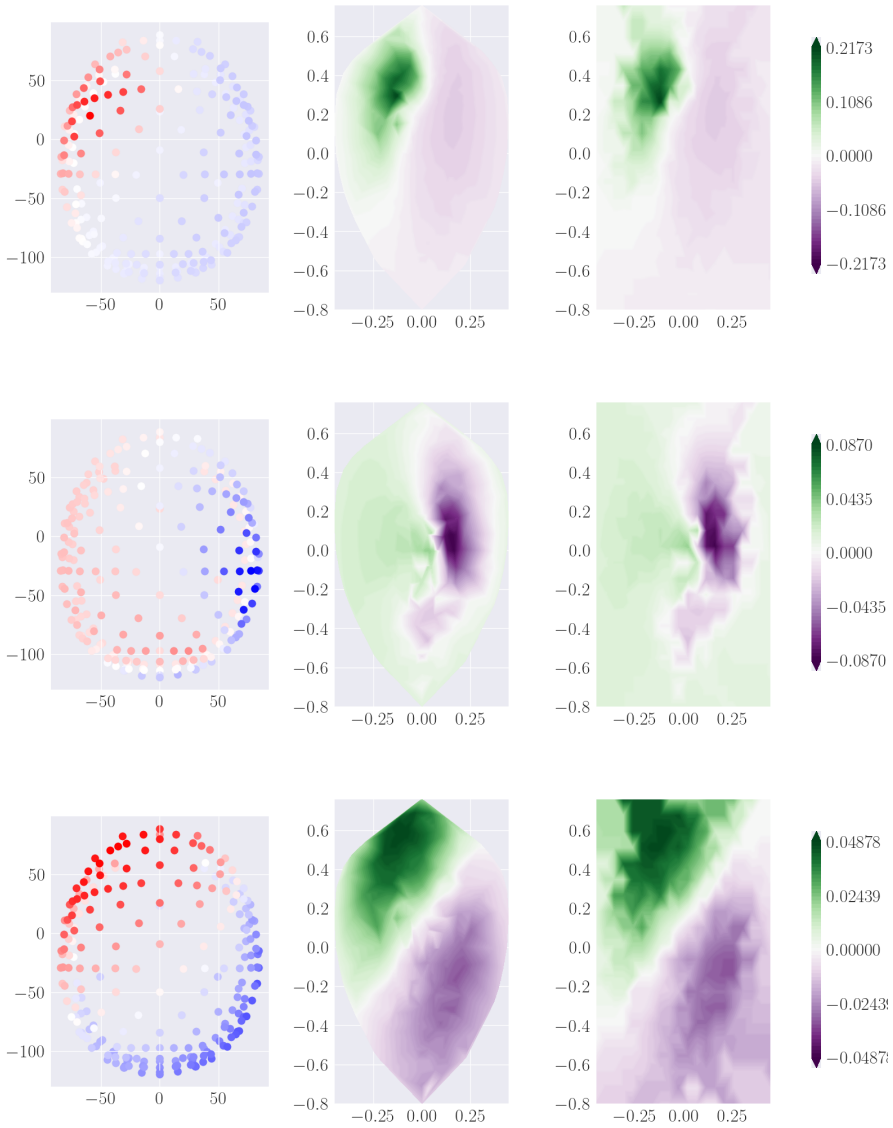

Figure 4.7:
Right: EEG measure for 3 different samples measured in μV.
Middle and Left: Illustration of the interpolation of the EEG data into two-dimensional matrix.

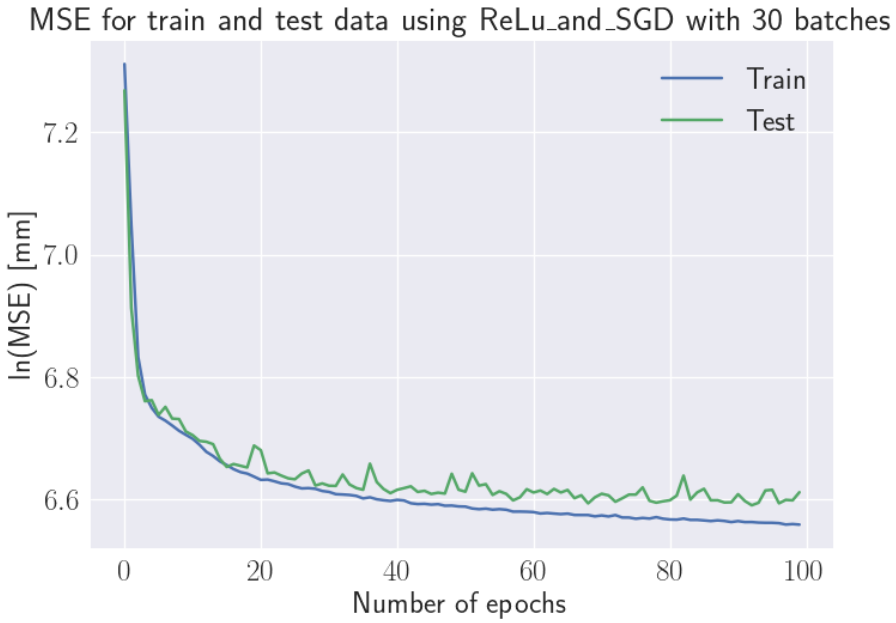

Figure 4.8: The validation accuracy for Convolutional Neural Network with 10 000 samples (20x20 matrix) with ReLU activation function.

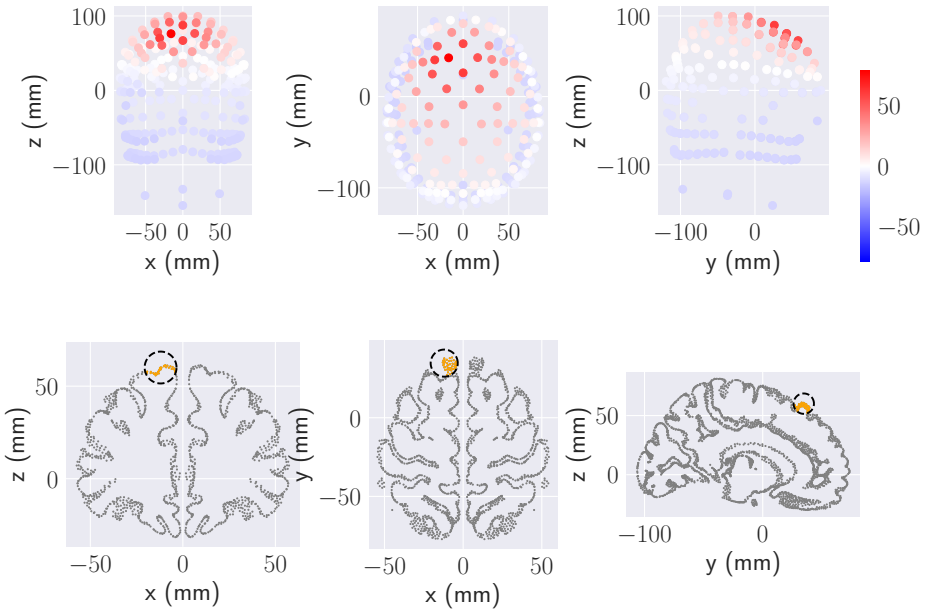

Figure 4.9: EEG for a sample containing a spherical population of current dipole sources with a random center within the celebral cortex. The EEG measure is seen from both sides (x-, z-plane and y-, z-plane) and above (the x-, y-plane). EEG electrode locations are presented as filled circels, where the color of the fill represents the amplitude of the measured signal for the given electrode.

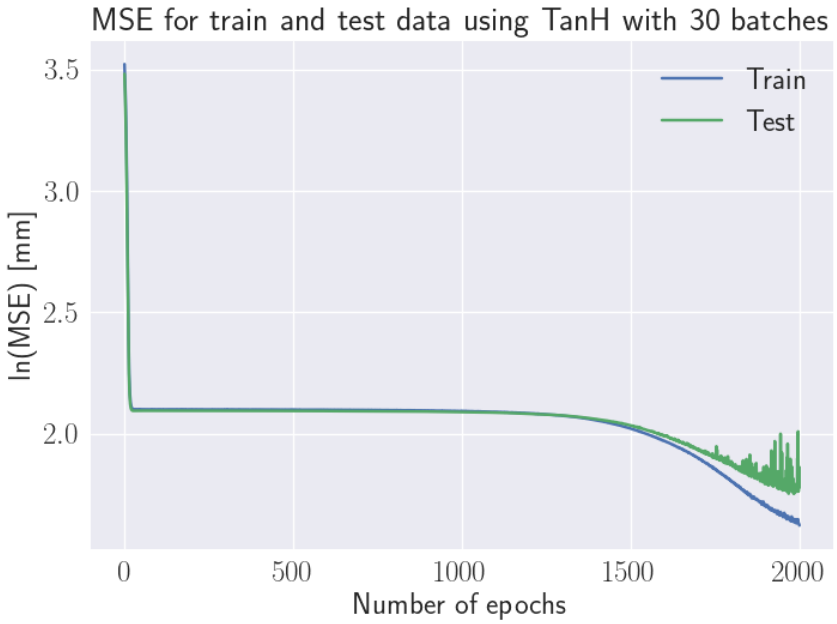

Figure 4.10: The validation accuracy for the simple Feed Forward Neural Network, predicting both center and radius for 10 000 samples, for 2000 epochs, with a learning rate equal to 0.0001.

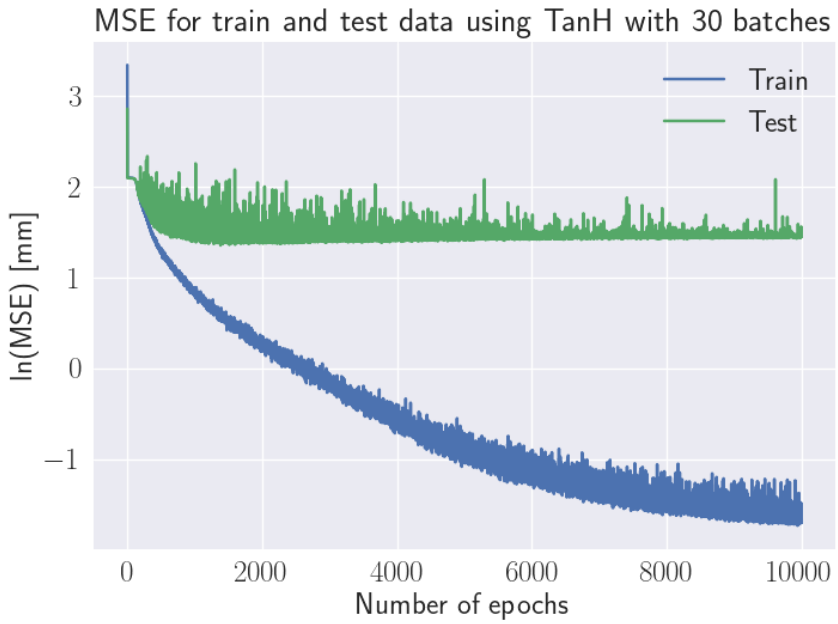

Figure 4.11: The validation accuracy for the simple Feed Forward Neural Network, predicting both center and radius for 10 000 samples, for 10000 epochs, with a learning rate equal to 0.001.

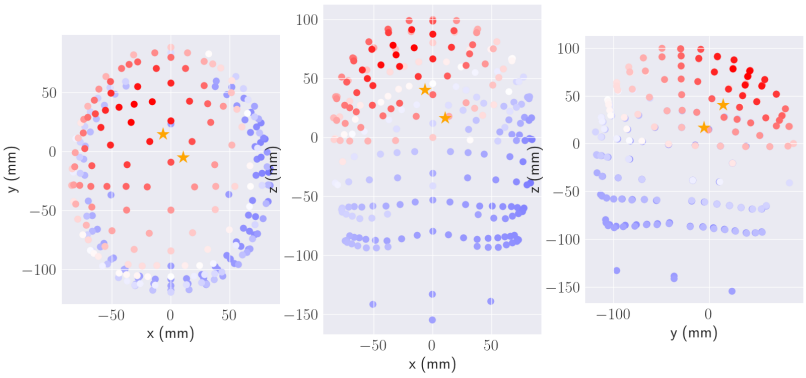

Figure 4.12: EEG for a sample containing two current dipole sources at random positions within the celebral cortex. The EEG measure is seen from both sides (x-, z-plane and y-, z-plane) and above (the x-, y-plane). EEG electrode locations are presented as filled circels, where the color of the fill represents the amplitude of the measured signal for the given electrode. The positions of the current dipole moments are marked with yellow stars.

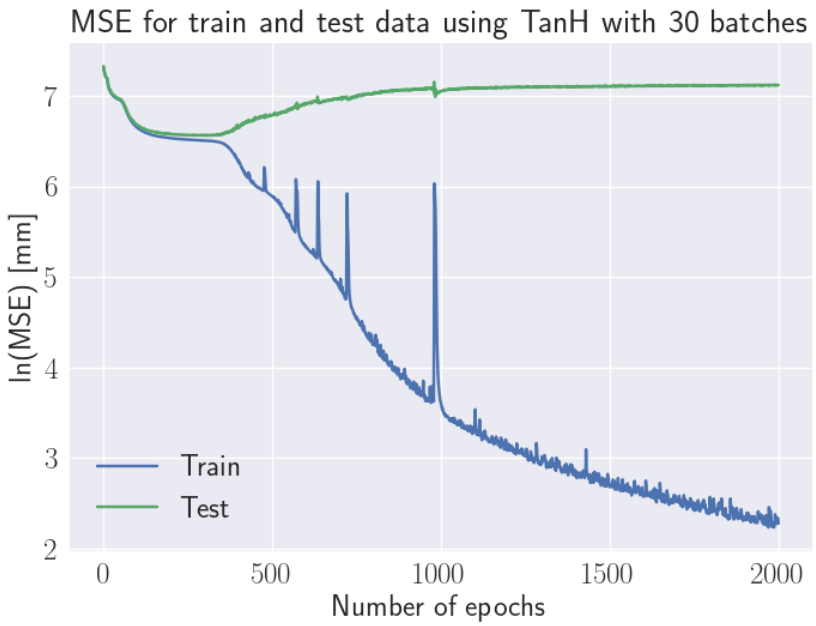

Figure 4.13: The validation accuracy for the simple Feed Forward Neural Network, predicting two current dipole sources.

Bibliography

[1] Risto J Ilmoniemi and Jukka Sarvas. *Brain signals: physics and mathematics of MEG and EEG*. Mit Press, 2019.

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

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

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

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

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

[7] 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.

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

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

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

[11] 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.

[12] Chigozie Nwankpa et al. "Activation functions: Comparison of trends in practice and research for deep learning". In: *arXiv preprint arXiv:1811.03378* (2018).

[13] 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).

[14] 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.

[15] 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).