

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

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 general 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 expressed as follows:

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

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

1.4 Electroencephalography

1.5 EEG Forward modeling

1.6 The Inverse Problem

Chapter 2

Electroencephalography

Neurons communicate with each other through the use of electrical currents. When a neuron receives a signal, it generates an electrical current that propagates along the axon and causes the release of neurotransmitters that diffuse across the gap between the sending and the receiving neuron. If the neurotransmitters are accepted by the receptors on the receiving 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 communication 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 [Jilmoniemi2019brain].

An EEG signal is believed to originate from large numbers of synaptic inputs to populations of geometrically aligned pyramidal neurons [Nunez2006electric]. The signal can be understood as a signature of neural activities that are generated from synaptic inputs to cells in the cortex. Synaptic inputs on its side are electrical or chemical signals that are being transmitted from one neuron to another, causing changes in the membrane potential of the neurons. In other words, neurons are specialized to pass signals, and synapses are the structures that make this transmission possible.

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

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

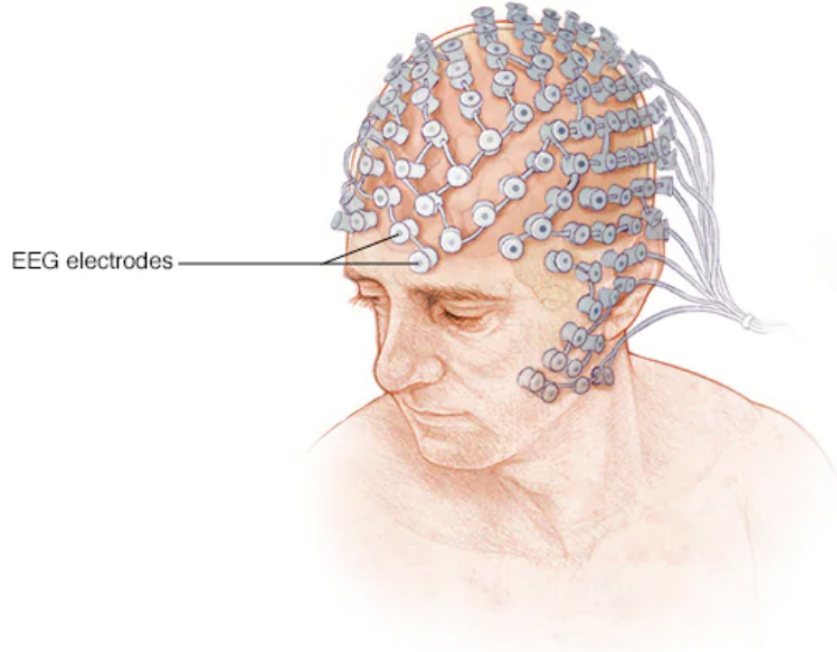


Figure 2.1: Illustration of the EEG method.

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 comprehension 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:

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

where \mathbf{J} is the electric current density in extracellular space, σ is the extracellular conductivity and ϕ is the extracellular electric potential [naess2021biophysically]. For simple, symmetric 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 utilize in this thesis, we will be using the numerical method known as Finite Element Method (FEM) when solve the equation.

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 utilized, which takes into account the intricate details of neuron morphologies to determine the transmembrane currents, I_n . In the second step, equation ?? is solved, under the assumptions that the extracellular medium acts as a volume conductor with the following properties:

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

The origin of extracellular 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} :

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

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

2.2 Multicompartmental modeling

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

The fundamental equation for MC models is given as:

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

where

This equation (??) assumes that a compartment n has two neighbors - one at $n - 1$ and the second at $n + 1$. However, when considering the endpoint compartments, this clearly will not be true. A commonly used boundary 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:

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

and

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

2.3 Volume conductors

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

2.4 Current Dipole Approximation

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

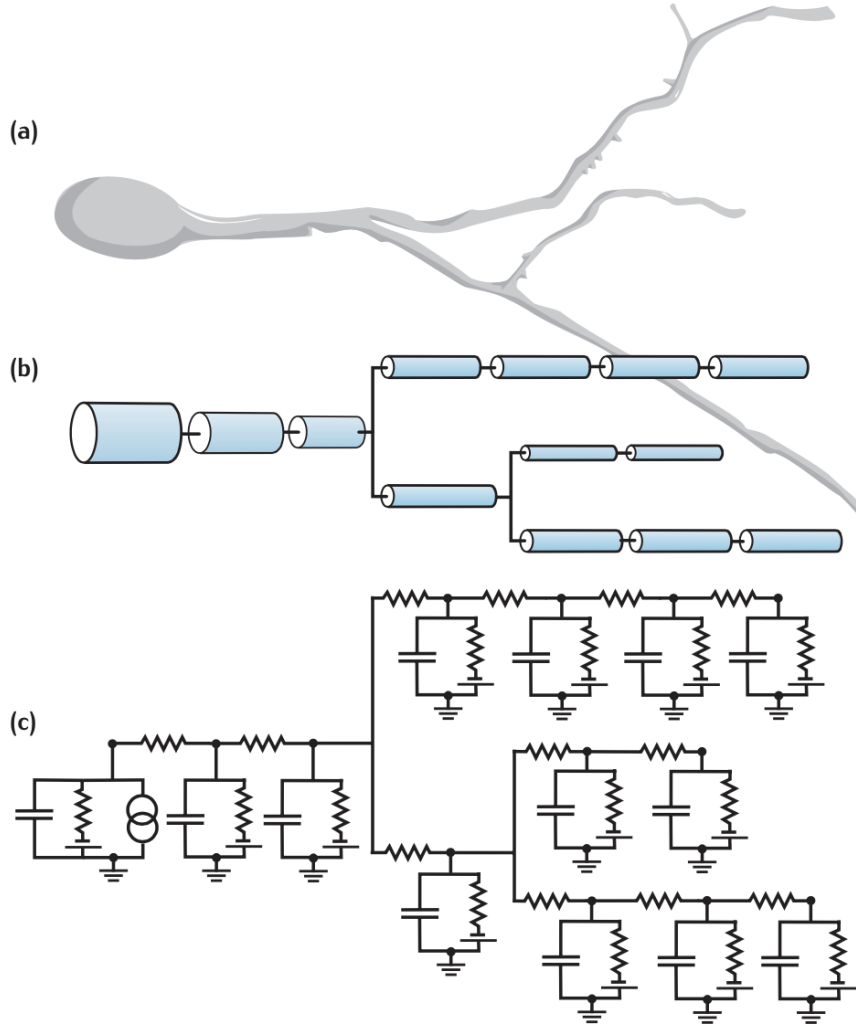


Figure 2.2: A diagram of the development of a multi-compartmental model. (a) The cell morphology is represented by (b) a set of connected cylinders. An electrical circuit consisting of (c) interconnected RC circuits is then built from the geometrical properties of the cylinders, together with the membrane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

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

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

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

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

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

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 extracellular 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 function. This also turns out to be true in our case, as the quadrupole, octopole and higher-order contributions to V_e decay more rapidly with distance R than the dipole contribution. 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:

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

where we have substituted for C_{dipole} in terms of other properties. \mathbf{p} denotes the current dipole 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 location \mathbf{r} . Finally θ represents the angle between \mathbf{p} and \mathbf{R} . This equation is known as the dipole approximation and is a precise approximation for calculating extracellular potential, given that R is much larger than the dipole length $d = |\mathbf{d}|$, like in the case of EEG [naess2021biophysically].

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

$$\mathbf{p} = \sum_{n=1}^N I_n \mathbf{r}_n \quad (2.8)$$

In figure ?? 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 ?? was derived 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 of the brain, cerebrospinal 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 cerebrospinal fluid exhibits a conductivity of approximately 1.7 S/m, while the conductivity of the skull 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 account for the variations in conductivity across different regions of the head, head models also takes into account that the EEG signal from a neuronal population will depend on whether the population is located in a *sulcus* or a *gyrus* [naess2021biophysically]. Said with other words, by integrating biophysical details into models, a deeper understanding emerges regarding the impact of various tissues on the distribution of extracellular potential, which in turn, can lead to improvements in the accuracy of EEG signal analysis.

2.5.1 The New York Head

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

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

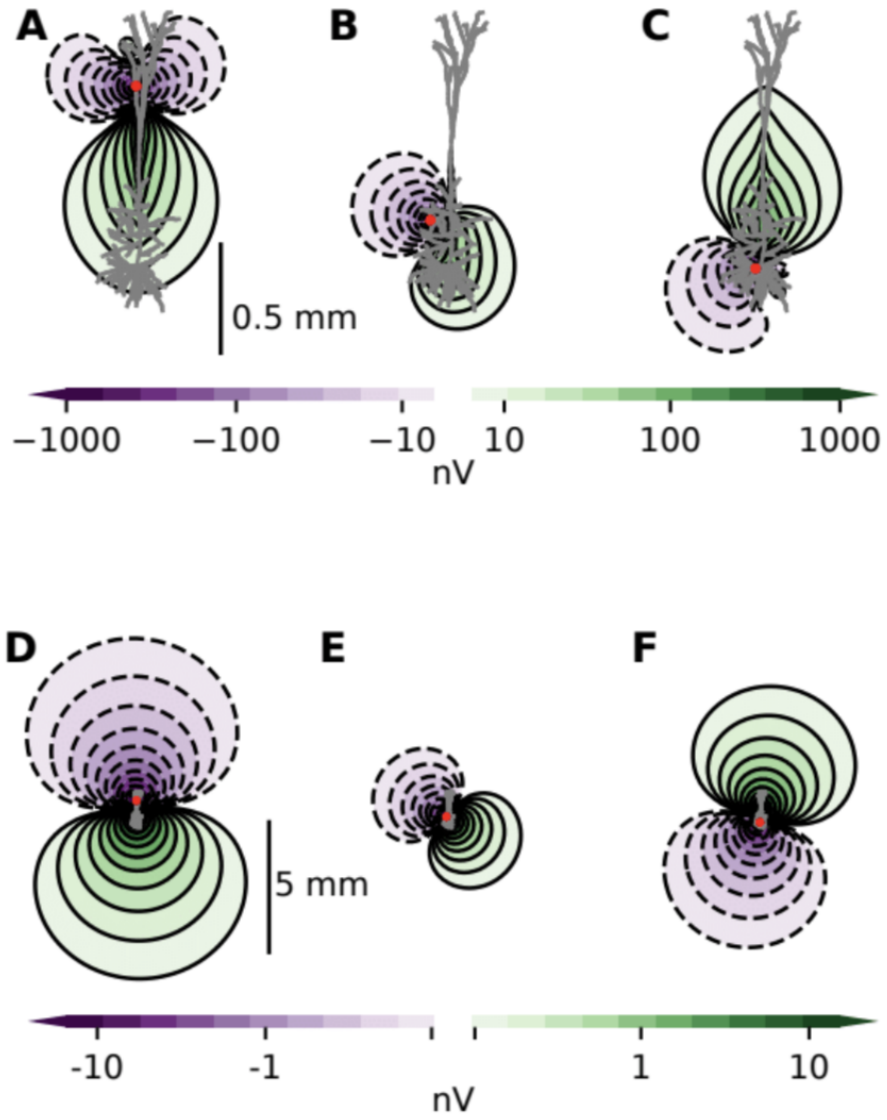


Figure 2.3: Source and explanation.

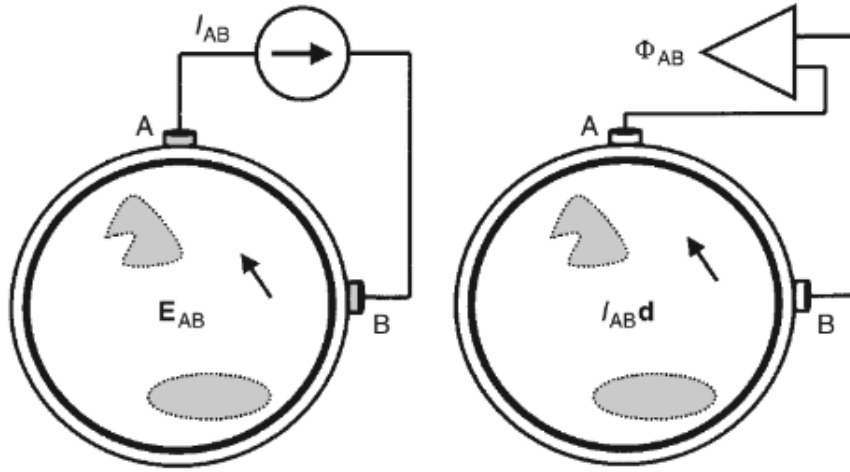


Figure 2.4: A caption here is needed.

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

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

The lead field matrix is constructed by taking advantage of the reciprocity theorem that states that knowledge of the current density through a volume conductor 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 conductor. If one suppose that a pair of stimulating electrodes is placed at locations A and B on the scalp as provided in figure ??, 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 anisotropy 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 electrodes. 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 conductor. 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, \mathbf{L} is given as:

$$L = \frac{E}{I}, \quad (2.9)$$

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

$$\Phi_{AB} = L_{AB} \cdot p, \quad (2.10)$$

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 mA·m 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 multicompartiment neuron models. These tools will be utilized in this thesis. For more information read: <https://lfp.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 therefore 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 multicompartiment 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 utilization 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 techniques for solving our specific EEG inverse problem and its wider implications.

3.1 Machine Learning and its Foundational Principles

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

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 available, however a rule of thumb is that the majority of the data are partitioned into the training set (e.g., 80%) with the remainder going into the test set [mehta2019high].

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

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

where $\theta = \theta_0, \theta_1, \dots, \theta_n$ denotes the model parameters, \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:

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

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, θ , an optimization algorithm 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

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

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

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

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

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

$$F(\mathbf{x}_0) \geq F(\mathbf{x}_1) \geq F(\mathbf{x}_2) \geq \dots \geq F(\mathbf{x}_n) \quad (3.5)$$

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

However, it is important to note that the error function in gradient descent is computed based on the training set, so that each step requires that the entire training set, referred 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 algorithm's 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 proved useful in practice for training machine learning algorithms on large data sets [bishop2006pattern].

Stochastic Gradient Descent

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

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

These smaller subsets taken from the entire dataset are commonly referred 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 than 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:

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

$$\mathbf{w}_\tau = \mathbf{w}_{\tau-1} + \mathbf{v}_\tau \quad (3.8)$$

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

3.2 Neural Networks

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

The network gathers knowledge by detecting relationships and patterns in data using past experiences known as training examples. These patterns are further updated by the usage of appropriate activation functions and finally presented as the output [nwankpa2018activation]. A neural network consists 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 ?? we have provided the basic architecture of neural networks. Here nodes are depicted 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:

$$a = f(\sum_{i=1}^n w_i x_i) = f(z) \quad (3.9)$$

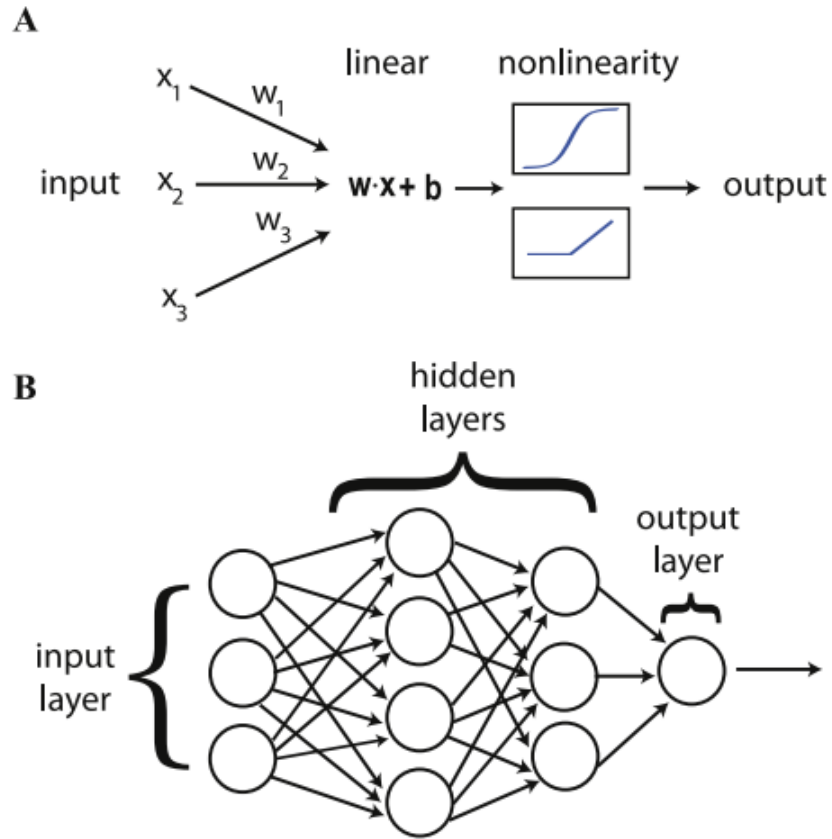


Figure 3.1: **(A)** The fundamental structure of neural networks comprises simplified neuron units that perform a linear operation to assign different weights to inputs, followed by a non-linear activation function. **(B)** These neuron units are organized into layers, where the output of one layer serves 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}, \dots, x_n recieved by n other neurons, multiplied with the weights w_i, w_{i+1}, \dots, w_n and added with bieases b_i, b_{i+1}, \dots, 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 ??, 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:

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

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

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

3.2.1 Activation functions

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

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

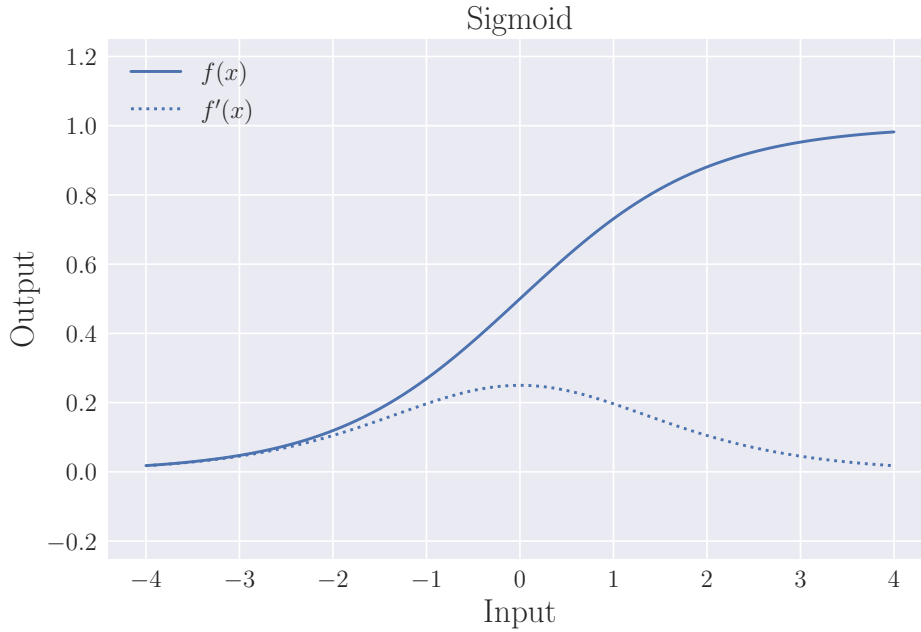


Figure 3.2: Sigmoid activation function.

Sigmoid

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

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

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 ??, the function returns 0.5 for an input equal to 0. Due to this, the value 0.5 can be seen as a threshold value which decides whether the input value belongs to what type of two classes.

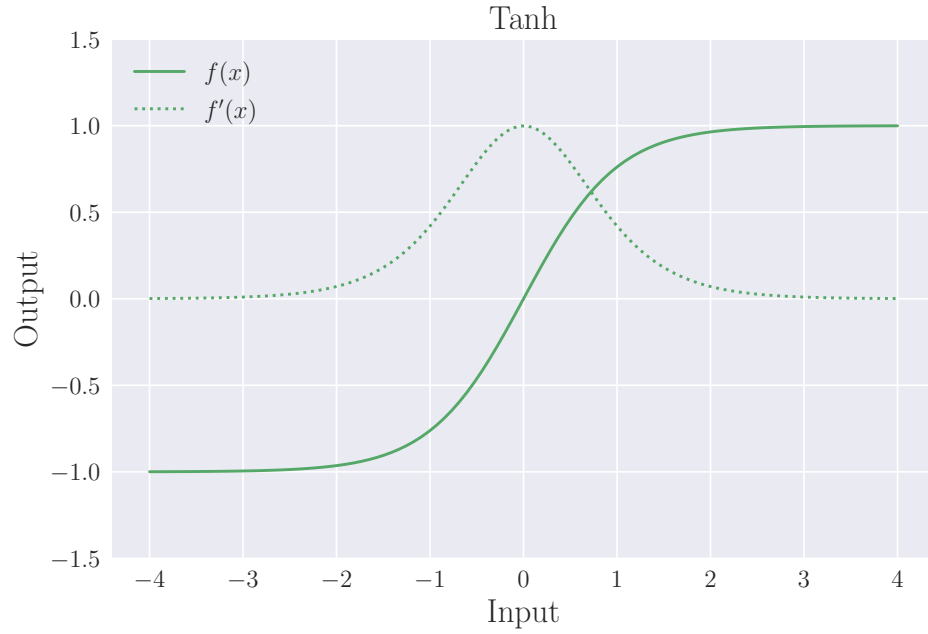


Figure 3.3: Hyperbolic tangent activation function.

Hyperbolic Tangent

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

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

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 ??

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

Rectified Linear Unit

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

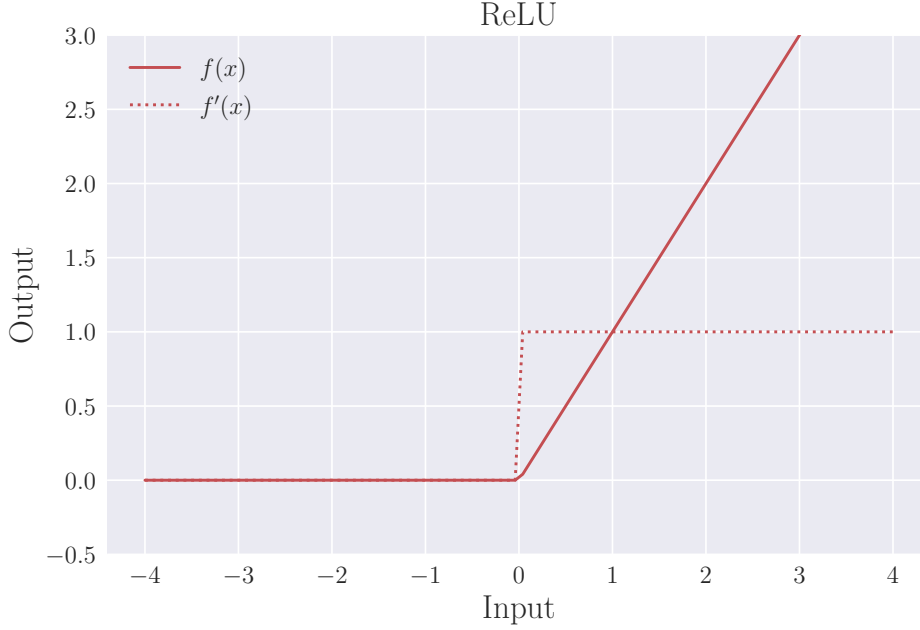


Figure 3.4: ReLU tangent activation function.

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

From Figure ??, 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 technique used in neural 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 propagation technique 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 considering a total of L layers within the neural network, with each layer identified by an index 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 equation setting up the algorithm is the definition of the error δ_i^l of the i -th neuron in the l -th layer:

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

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:

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

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

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

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

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

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 ?. This requires to know the expression for the derivative of both the cost function $C(\mathbf{W}) = C(\mathbf{a}^L)$ and the activation function $f(z)$.
4. **"Backpropagate" the error:** use equation ?? to propagate the error backwards and calculate δ_j^l for all layers.
5. **Calculate gradient:** use equation ?? and ?? 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:**

$$\begin{aligned} w_{jk}^l &= w_{jk}^l - \eta \delta_j^l a_k^{l-1} \\ b_j^l &= b_j^l - \eta \delta_j^l \end{aligned}$$

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 ??, we see that we encounter such scenarios when the input value is considerably small or large.

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

The Inverse Problem

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

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

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

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

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

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

Chapter 4

Dipole Source Localization using Neural Networks

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

Feed Forward Neural Networks

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

Convolutional Neural Networks

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

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

4.0.1 Neural Networks

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

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

$$a = f(\sum_{i=1}^n w_i x_i + b_i) = f(z) \quad (4.1)$$

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

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

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

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

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

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

4.1.1 DiLoc

The FFNN that are trained to solve the inverse problem of ours has an input layer of 231 neurons, corresponding to the $M = 231$ electrode measures of the potentials. The input layer is followed by three hidden layers with 120, 84 and 16 hidden neurons, respectively. When the aim is to estimate the localization of the current dipole, *only*, the final output layer holds the predicted x-, y- and z- position of the desired dipole source. However, if the interest lies in determining the size of the dipole population, an alternative output layer is incorporated in the network architecture providing the radius

4.1. FEED-FORWARD NEURAL NETWORK APPROACH FOR LOCALIZING SINGLE DIPOLE SOURCE

of the dipole(s), in addition to the location coordinates. This enables the model to provide a more comprehensive understanding of the dipole source(s) being analyzed.

4.1.2 Validation accuracy

In Figure ?? we have provided the validation accuracy, 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:

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

The coefficient of determination is given as follows:

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

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

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

4.1.3 Activation functions, Batchsize and Optimization

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

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

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

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

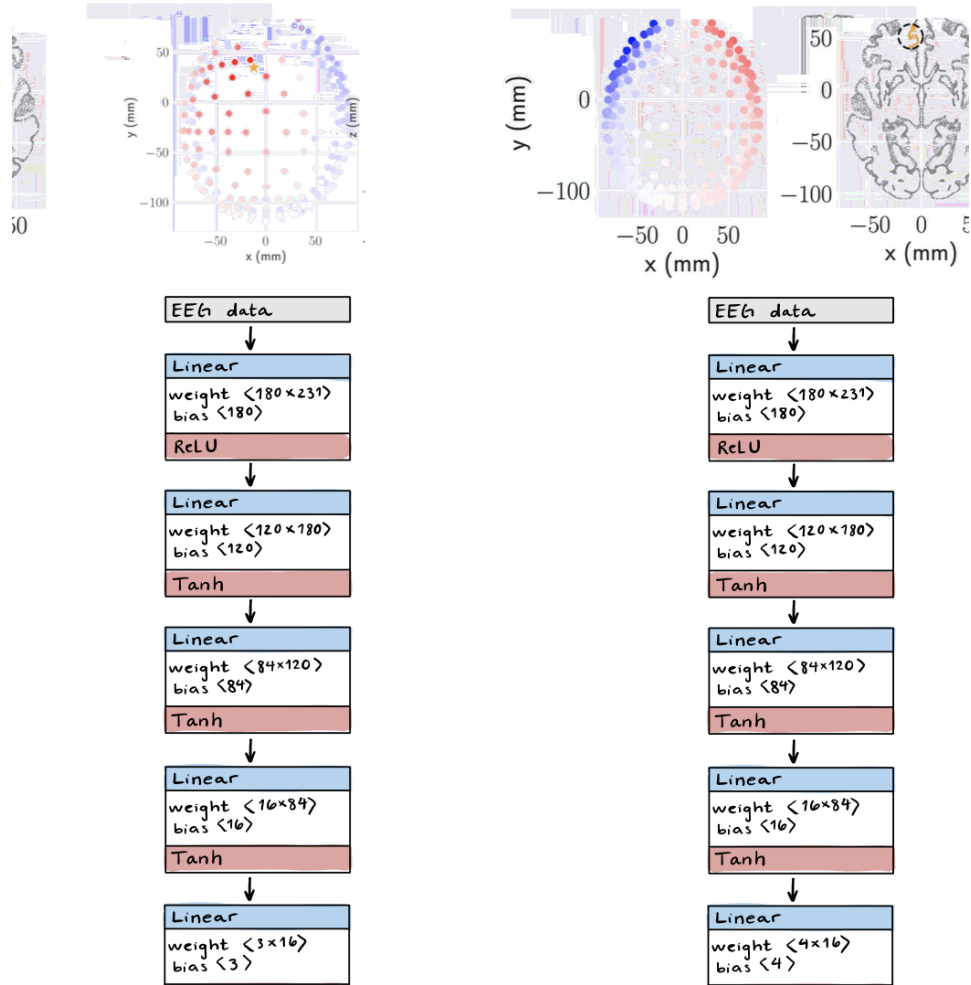


Figure 4.1: A caption here is needed.

4.1. FEED-FORWARD NEURAL NETWORK APPROACH FOR LOCALIZING SINGLE DIPOLE SOURCE

4.1.4 Training, testing and evaluation

In order to make an ANN that generalizes well to new data we split our data into training and testing sets. Randomly selecting 80 percent of the rows in the full dataset, we put this into a separate one and call it our training set. The remaining 20 percent is put into the test set. In practice, the training data set consists of pairs of an input vector with EEG signals and the corresponding output vector, where the answer key is the x-, y- and z coordinate of the dipole source. The neural network is then feed with the training data and produces an estimation of the localization of the dipole. The estimation is found by the network through optimizing the parameters β minimizing the cost function, or said in other words, through finding parameters for the function that produces the smallest outcomes, meaning the smallest errors. The result provided by the network is then compared with the target, for each input vector in the training data. Adjustment of parameters...

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

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

Chapter 5

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 fed to the different networks.

5.1 Simulation of EEG Signals

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

locations within the cortex.

5.1.1 The Effect of dipole location and orientation

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

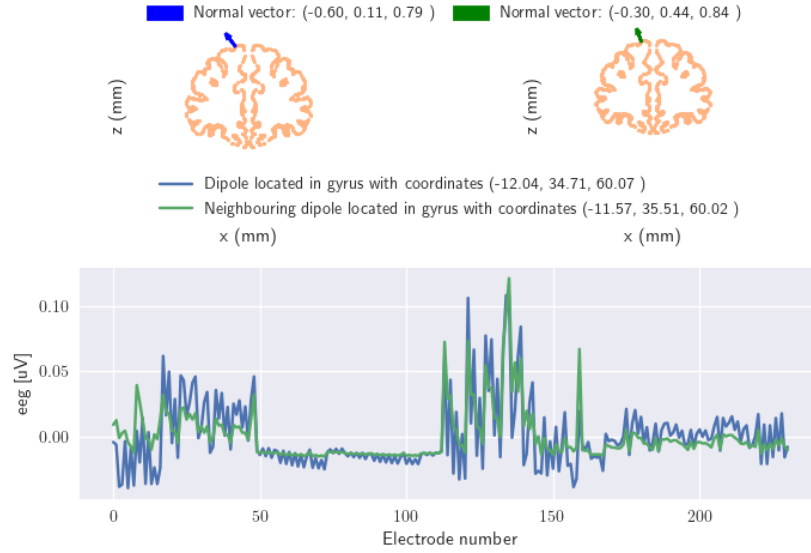


Figure 5.1: EEG signal for neighbouring dipoles.

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

pattern, as shown in Figure ??C.

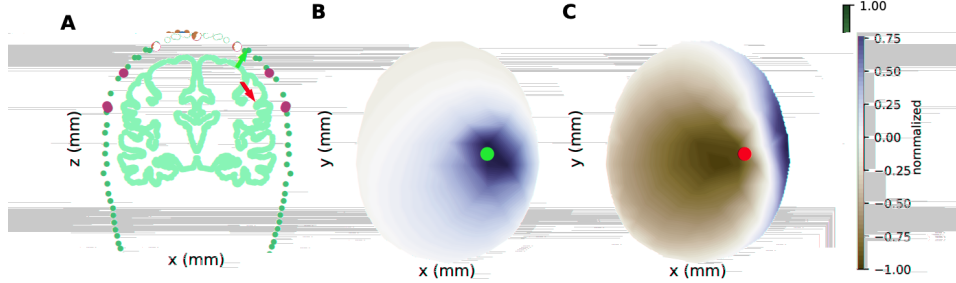


Figure 5.2: A: Two selected dipole locations in the New York head model: one in a gyrus (red) and one in a sulcus (blue). The head model is viewed from the side (x, z-plane). Close to the chosen cross-section plane, EEG electrode locations are marked in light blue. Available dipole locations near the cortical cross-section form an outline of the cortical sheet and are marked in pink. The current dipole moment for all cases was 10^7 nA μ m. B: Interpolated color plot of EEG signal from the gyrus dipole, viewed from the top (x, y-plane). The plotted EEG signal is normalized, with a maximum value of 1.1 μ V. C: Interpolated color plot of EEG signal from the sulcus dipole. The plotted EEG signal is normalized, with a maximum value of 0.7 μ V. [naess2021biophysically]

With other words, the EEG outcome is genuinely influenced by the orientation of the current dipole moment generating the signal, as variations in dipole orientation can impact both the direction and distribution of electrical potentials. In Figure ??, we present the EEG signals obtained from four identical current dipoles with different orientations, situated in distinct hypothetical folding patterns of the cortical surface. Firstly, Figure ??A and ??C provide an expanded illustration of the aforementioned scenarios, incorporating additional dipole moments located in a gyrus and a sulcus, respectively. In Figure ??B, 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 ??A and ??B, as depicted in Figure ??D. Lastly, Figure ??E demonstrates the minimal EEG contribution observed when the dipoles are divided between two opposing sulci.

5.1.2 Noise

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

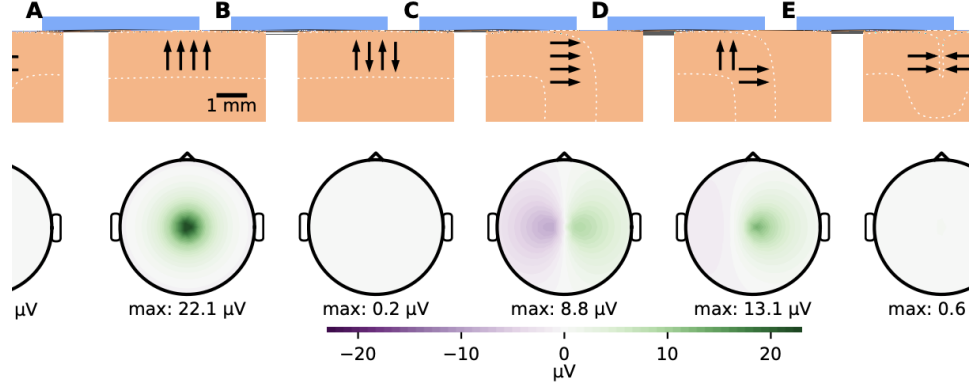


Figure 5.3: Different folding patterns of the cortical surface are represented by white dashed lines. EEG signals are calculated from four identical current dipoles with varying orientations. A: Dipoles aligned in the same direction within a gyrus. B: Dipoles pointing in opposite directions within a gyrus. C: Dipoles aligned in the same direction within a sulcus. D: Dipoles distributed between a gyrus and a sulcus, pointing towards the cortical surface. E: Dipoles divided between opposing sulci, pointing towards the cortical surface. All panels have a dipole moment magnitude of 10 nAm, and the dipoles are positioned at the centers of the arrows in the top row.

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

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

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

In our approach, we recognize that the introduction of noise to the simulated EEG data is an essential step to enhance the robustness of the trained

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

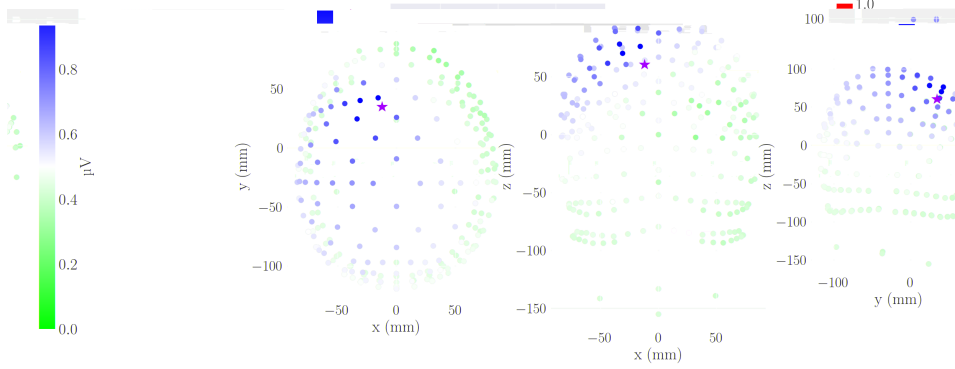


Figure 5.4: EEG for a sample containing one single current dipole source at a random position within the cerebral cortex. As for all samples, 10 percent of normally distributed noise has been added to the original signal. The EEG measure is seen from both sides (x-, z-plane and y-, z-plane) and above (the x-, y-plane). EEG electrode locations are presented as filled circles, 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.

5.1.3 The dataset

The training dataset for a simple feed-forward neural network comprises 50,000 rows, where each row corresponds to a single sample or patient. Within the dataset, there are 231 columns representing the features, which denote the EEG measurements recorded at each electrode. Consequently, the design matrix has a size of 50,000 x 231.

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

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

5.2 Localizing Single Dipole Sources

The first problem we want to introduce for our neural network, DiLoc, is simply the standard inverse problem. By feeding the network EEG data corresponding to the electrical activity from dipoles distributed randomly in the cerebral cortex for different samples, we want the network to output the x-, y- and z-coordinates of the localization of the current dipoles. In figure ?? we have provided the loss of our network as a function of training epochs. It is apparent that the the network picks up on the patterns of the data, as we clearly can observe the trend of decreasing loss with an increasing number of epochs. Moreover, we can see that the both training and validation loss stabilizes at around 400 epochs.

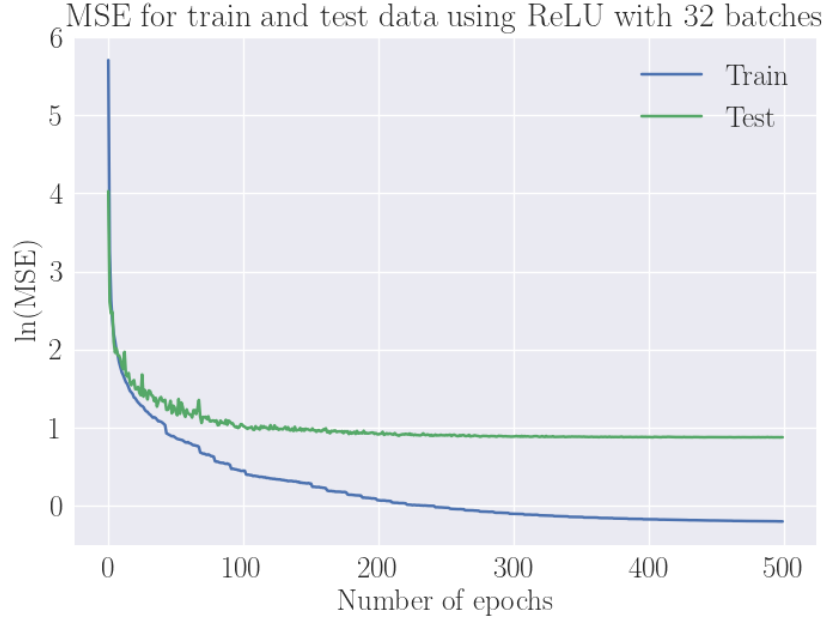


Figure 5.5: The loss for the simple Feed Forward Neural Network with 50 000 samples and ReLU as activation function.

In Table ?? we have provided the performance of the network by considering different error metrics. The mean absolute error (MAE) values for the x-, y-, and z- coordinates range from 0.8300 mm to 0.8998 mm. This means that,

on average, the network’s predictions have an error smaller than 1 mm in each coordinate. Considering the range of the coordinates, the MAE values represent a reasonable level of accuracy. The mean squared error penalizes larger errors/outliers more severely than MAE since it involves squaring the differences. In our case the MSE values for the different coordinates range from 1.2134 mm to 1.4110 mm. The higher MSE values suggest that the predictions of the network may have larger errors in some cases, resulting in a higher average squared difference. However, the magnitude of the mean squared errors is still within a reasonable range when analyzing them in the context of the coordinates ranges. Finally the root mean squared error provides a measure of the standard deviation of the errors and helps to understand the spread of errors around the mean. The RMSE values of ours are slightly lower than the corresponding MSE values with a range from 1.1016 mm to 1.1878 mm. The table also presents the error metrics calculated for the euclidean distance. For both MAE, MSE and RMSE the value is higher than the individual coordinate errors, indicating that the errors in the x, y, and z coordinates are not perfectly aligned and contribute to the overall distance. It is worth mentioning that specific points in the cortex matrix may potentially contribute more to the errors. Further investigation could be performed to identify any specific patterns or regions in the cortex that exhibit higher error rates. However, overall the results indicate that the network is able to predict the dipole location with reasonable accuracy. While there are some errors in the predictions, the errors are generally within an acceptable range.

| | Error for different target values | | | |
|------|-----------------------------------|----------------------|----------------------|----------------------------|
| | x-coordinate [mm] | y-coordinate [mm] | z-coordinate [mm] | Euclidean Distance [mm] |
| MAE | 0.8300 | 0.8998 | 0.8419 | 0.8573 |
| MSE | 1.2638 | 1.4110 | 1.2134 | 1.2961 |
| RMSE | 1.1242 | 1.1878 | 1.1016 | 1.1385 |

Table 5.1: **Evaluation of the network performance utilizing different Error Metrics.**

Network performance on test dataset consisting of 1000 samples. The errors are measured using Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE).

In order to conduct a detailed analysis of the network’s performance, Figure ?? presents the Mean Absolute Error (MAE) for various dipole locations within the New York head model cortex matrix. The figure provides valuable insights into the distribution of errors across different regions of the cortex, with three cross-sections—front, top, and side—depicted for examination.

Theothesx-coordinate

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

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

Surprisingly, the Mean Absolute Error (MAE) for dipoles located in the sulcus is 1.1075 mm, which is even smaller than the MAE for dipoles in the gyrus measuring 1.2458 mm. This finding directly contradicts the initial assumption and suggests that the network exhibits great precision in predicting dipole locations regardless of their placement within the cortex. The lower MAE values specifically for dipoles in the sulcus the network’s remarkable ability to effectively capture the complex features and variations associated with deeper cortical placements. Moreover, the results highlight the network’s robustness and accuracy across different cortical structures, reinforcing its potential for accurate dipole localization within the human brain.

5.3 Convolution Neural Network Approach for localizing single dipole sources

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

5.3. CONVOLUTION NEURAL NETWORK APPROACH FOR LOCALIZING SINGLE DIPOLE SOURCES

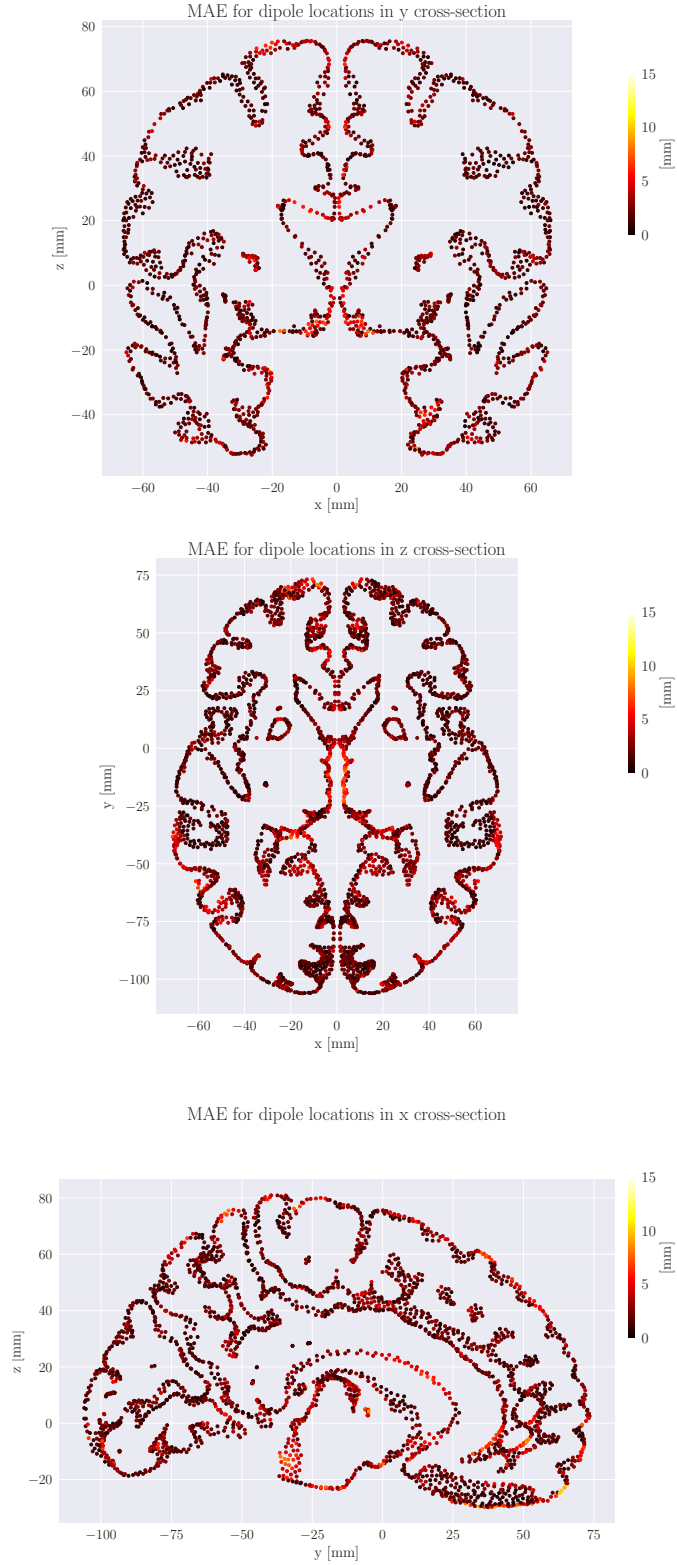


Figure 5.6: Different cross-sections of the cortex from the New York head model, seen from front, top and side. Each point represents a possible position in the cortex matrix. The color of the each point indicates the mean absolute error (MAE) of the neural network when predicting that specific dipole location.

Figure 5.7:
Right: EEG measure for 3 different samples measured in μV .
Middle and Left

5.3. CONVOLUTION NEURAL NETWORK APPROACH FOR LOCALIZING SINGLE DIPOLE SOURCE

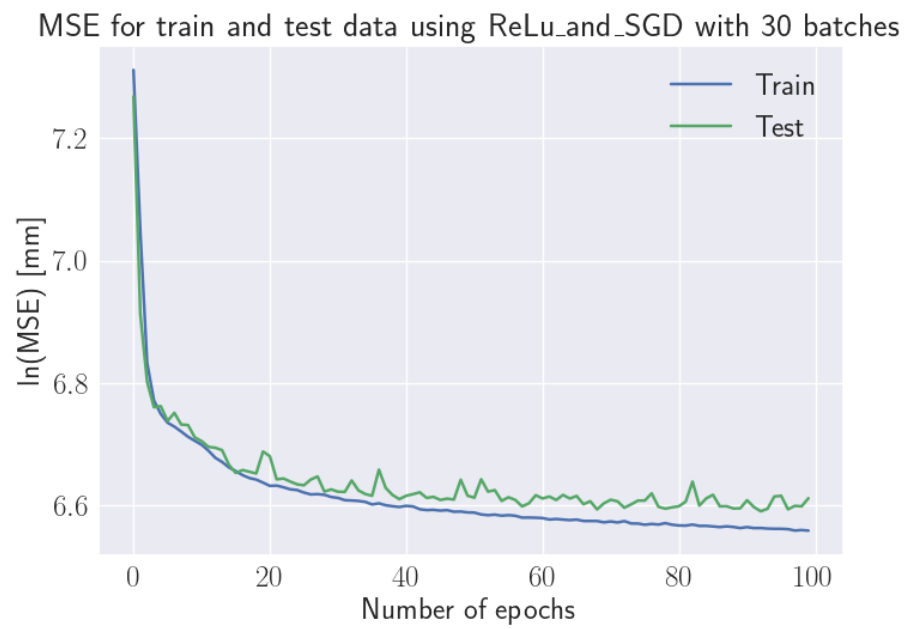


Figure 5.8: The validation accuracy for Convolutional Neural Network with

