

Hochschule für angewandte Wissenschaften Ansbach

Fakultät Technik

Studiengang Applied Research in Engineering Sciences (APR)

Masterarbeit

Thema: EEG- und MEG-Quelllokalisierung mithilfe von simulationsbasierten Deep Learning Methoden

Vor- und Zuname: Jakob Matthias Winkler

ausgegeben am: 01.02.2025

abgegeben am: 01.07.2025

Erstprüfer: Prof. Dr. Christian Uhl

Zweitprüfer: Prof. Dr. Stefan Geißelsöder

Erklärung

„Ich versichere, dass ich die Arbeit selbständig angefertigt, nicht anderweitig für Prüfungszwecke vorgelegt, alle benützten Quellen und Hilfsmittel angegeben sowie wörtliche und sinngemäße Zitate gekennzeichnet habe.“

Ort, Datum _____ Unterschrift _____

Zusammenfassung

Diese Masterarbeit untersucht die Lösung des inversen Problems der EEG-Quelllokalisierung mittels eines physikbasierten Deep Learning Ansatzes. Ausgangspunkt ist ein kombinierter MEG-Datensatz von sieben Probanden mit individuellen Finite-Element Kopfmodellen.

Es wurde ein Trainingsdatensatz von 250000 synthetischen Samples mit EEG-Signal mit additivem Rauschen, variabler Gaußverteilung und zugrundeliegender neuronaler Aktivität simuliert. Ein Fully-connected-neuronales Netz mit Skip-connections wird mithilfe einer Weights and Biases Hyperparameteroptimierung trainiert.

Innovativ ist die Verwendung der MEG-Dipol Scan Positionen als Referenz während der Hyperparameteroptimierung und Evaluation, wodurch Modelle besser als mit bisherigen Lösungen trainiert und optimiert werden können.

Die Arbeit widerlegt die Annahme, dass mehrere verschiedene Kopfmodelle im Trainingsdatensatz nötig sind, um auf mehreren Probanden gute Vorhersagen zu treffen. Des Weiteren schlägt die Arbeit einen Benchmark Datensatz vor, der zukünftige Vergleiche zwischen Modellen erleichtert.

Schlüsselwörter

Elektroenzephalographie – Magnetoenzephalographie – Deep Learning – Quelllokalisierung – Dipol Scan – inverses Problem – Datensimulation – Hyperparameteroptimierung

Abstract

This master thesis investigates the solution of the inverse problem of EEG source localization with a physics-based deep-learning approach. The starting point is a combined MEG data set from seven subjects, each including an individual finite-element head model.

A training dataset of 250000 synthetic EEG samples was simulated, including additive noise, variable Gaussian source profiles and the underlying neuronal activity. A fully connected neural network with skip connections was trained with a comprehensive hyperparameter optimization with the Weights and Biases package.

A key innovation is the use of MEG dipole-scan positions as a reference during both hyperparameter tuning and evaluation, enabling models to be trained and optimized more effectively than with previous approaches.

The study demonstrates that achieving accurate predictions across multiple subjects does not require an extensive variety of head models in the training dataset. In addition, a benchmark data set is proposed to facilitate future model comparisons.

Keywords

Electroencephalography – Magnetoencephalography – Deep Learning – Source Localization – Dipole Scan – Inverse Problem – Data Simulation – Hyperparameter Optimization

Danksagungen

An erster Stelle möchte ich mich bei meiner Freundin, meinem besten Freund und meiner Familie bedanken, die mich während der Anfertigung dieser Arbeit immer unterstützt und motiviert haben.

Ein besonderer Dank gilt der Forschungsgruppe von Prof. Dr. Carsten Wolters am Institut für Biomagnetismus und Biosignalanalyse Münster, vor allem Prof. Dr. Carsten Wolters und Dr. Tim Erdbrügger, die mich in häufigem Austausch vor allem im Bereich des EEGs fachlich beraten und die benötigten Daten zur Verfügung gestellt haben.

Des Weiteren bedanke ich mich für die finanzielle Unterstützung durch das Bundesministerium für Bildung und Forschung (BMBF, Förderkennzeichen: 05M20WBA, Verbundprojekt 05M2020 - DyCA).

Zu guter Letzt möchte ich mich bei meinen beiden Betreuern Prof. Dr. Christian Uhl und Prof. Dr. Stefan Geißelsöder bedanken, die immer ein offenes Ohr für mich hatten, mich jederzeit fachlich unterstützt haben und mich in meinem Forschungswillen bestärkt haben.

Vielen Dank!

Inhaltsverzeichnis

Erklärung	2
Zusammenfassung	3
Abstract	4
Danksagungen	5
Inhaltsverzeichnis	6
Abkürzungsverzeichnis	8
Abbildungsverzeichnis	9
1. Einleitung	11
1.1. Motivation	11
1.2. Relevanz der EEG-Quelllokalisierung für Forschung und Praxis	11
1.3. Aufbau der Arbeit	12
2. Theorie	12
2.1. EEG und Quelllokalisierung	12
2.1.1. Grundlagen des EEG	12
2.1.2. Das inverse Problem in der EEG- Analyse	22
2.1.3. Klassische Methoden der Quelllokalisierung	23
2.2. Deep Learning	24
2.2.1. Grundlagen Deep Learning	24
2.3. Potentiale von Deep Learning in der EEG-Quelllokalisierung	28
3. Stand der Forschung und Zielsetzung	29
3.1. Stand der Forschung bei EEG-Quelllokalisierung mit Deep-Learning	29
3.2. Zielsetzung der Arbeit	30
4. Methoden	30
4.1. Datenquellen und Aufbereitung realer EEG-Daten	30
4.1.1. Beschreibung der verfügbaren Daten	30
4.1.2. Vorverarbeitung der realen Daten	31
4.1.3. Herausforderungen bei der Datenvalidierung	32
4.1.4. Validierung anhand des MEG-Dipole Scans	32
4.2. Simulation eines geeigneten Datensatzes	34
4.2.1. Hinzufügen von Rauschen	34
4.2.2. Skalierung der Daten	35
4.2.3. Gaußsche Propagation von Aktivität	36
4.2.4. Variationen von Gaußglocken und verschiedene Verteilungen	38
4.2.5. Quelltiefe	38

4.3. Entwicklung und Training der Deep Learning Modelle	39
4.3.1. Wahl der Grundarchitektur	39
4.3.2. Netzwerkverbesserungen	39
4.3.3. Trainingsmethodik.....	40
4.3.4. Implementierung technischer Optimierungen	40
4.3.5. Hyperparameter tuning	40
4.4. Reproduzierbarkeit der Experimente	41
5. Ergebnisse und Diskussion	41
5.1. Ergebnisse der Hyperparameteroptimierung	41
5.2. Evaluation auf simulierten Daten	44
5.3. Validierung auf realen EEG-Daten	46
5.4. Inferenzzeit	50
6. Grenzen der Arbeit und Ausblick	50
7. Fazit.....	51
7.1. Zusammenfassung der wesentlichen Erkenntnisse.....	51
7.2. Beitrag der Arbeit zur EEG-Quelllokalisation	52
8. Literaturverzeichnis	53

Abkürzungsverzeichnis

EEG	Elektroenzephalographie
MEG	Magnetoenzephalographie
ERP	ereigniskorrelierte Potentiale (event related potentials)
MNE	Minimum Norm Estimation
fMRT	funktionelle Magnetresonanztomographie
PSP	postsynaptisches Potential
SQUID	supraleitendes Quanteninterferenzgerät (superconducting quantum interference device)
DTI	Diffusions-Tensor-Bildgebung (Diffusion-tensor-imaging)
BEM	Boundary element method
FEM	Finite element method
sLORETA	standardized low-resolution brain electromagnetic tomography
NN	Neuronales Netz (Neural network)
FCNN	Fully-connected-neural-network
MAE	Mean absolute error
MSE	Mean squared error
SGD	Stochastischer Gradientenabstieg (Stochastic gradient descent)
ADAM	Adaptive moment estimation
ReLU	Rectified linear unit
SiLU	Sigmoid weighted linear unit
GELU	Gaussian error linear unit
SNR	Signal-Rausch-Verhältnis (Signal-to-noise-ratio)
CNN	Convolutional neural network
LE	Lokalisationsfehler (Localization error)

Abbildungsverzeichnis

Abbildung 1: Schematische Darstellung einer Nervenzelle mit den wesentlichen Bestandteilen Zellkern, Dendriten, Soma, Axon, Myelinscheide, Ranvierscher Schnürring und den Endknöpfchen (Synapse) [17].	13
Abbildung 2: Darstellung des Einflusses von inhibitorischen und exzitatorischen präsynaptischen Signalen auf ein Neuron. Nach [2].	14
Abbildung 3: Potential an der Kopfoberfläche für einen tangentialen Dipol (gelber Pfeil). In rot wird das positive Potential angezeigt, das negative in blau [2].	15
Abbildung 4: Beispiele der 3 häufig genutzten Grenzflächen bei der BEM-Methode: Gehirn-Schädelknochen, Schädelknochen-Kopfhaut und Kopfhaut-Luft. Die Flächen werden mithilfe von Dreiecken realisiert. [28].	20
Abbildung 5: Darstellung von 2 verschieden feinen FEM-Modellen (mitte, rechts), die aus einem Originalmodell (links) berechnet wurden. Mitte und rechts zeigen den rot umrahmten Bereich in links [30].	21
Abbildung 6: Graphische Darstellung der Aktivierungsfunktionen ReLU (blau), Leaky ReLU (gelb), SiLU (grün) und GELU (rot) im Bereich [-2, 2].	28
Abbildung 7: Gemittelte EEG- (links) und MEG-Daten (rechts) des ersten Probanden mit der Amplitude y und der Zeit in ms auf der x-Achse.	30
Abbildung 8: 3D-Positionen der 54 Elektroden für Proband 1.	32
Abbildung 9: EEG- (links) und MEG-Dipole Scan (rechts) des SEP von Proband 1. Der MEG-Dipole Scan zeigt dabei einen deutlich kleineren Bereich und Fehler bei der optimalen Lösung.	33
Abbildung 10: Exemplarische Darstellung der Gauß Verteilung. Der Sigmawert ist veränderbar und kann zu Beginn angegeben werden.	37
Abbildung 11: Beispielhafte Darstellung eines $1r2$ (blau) und eines $1r3$ (orange) Verlaufs.	37
Abbildung 12: Konvexe Hülle von Proband 1 zur Tiefenbestimmung von simulierten Quellen. ...	38
Abbildung 13: Weights and Biases Übersicht der Parameterkombinationen der 10 Trainingsdurchläufe.	42
Abbildung 14: Scatter plot des Zusammenhangs zwischen Trainings Loss und Distanz (in mm) mit einer linearen Regression zum Darstellen der Korrelation.	43
Abbildung 15: Darstellung der Netzarchitektur mit 6 hidden Layern, wobei in jeder hidden Layer Batch-Normalisierung und Dropout enthalten ist. Zwischen den Layern stellen Skip-Connections einen Identitätspfad her. Die Größe der Layer ist proportional zur Anzahl der Neuronen im Layer.	44

Abbildung 16: Durchschnittliche Distanz (in mm) der Lokalisierung verschiedener Methoden zur tatsächlichen ground truth der Simulation.....45

Abbildung 17: Durchschnittliche Distanzen der Vorhersagen der KI-Modelle zur Vorhersage des EEG-Dipol Scans mit der jeweiligen individualisierten Leadfield-Matrix.47

Abbildung 18: Durchschnittliche Distanzen der Vorhersagen der KI-Modelle zur Vorhersage des MEG-Dipol Scans mit der jeweiligen individualisierten Leadfield-Matrix.....48

Abbildung 19: Durchschnittliche Distanz der KI-Modelle zum Target auf simulierten und realen Daten über alle Probanden. Bei den simulierten Daten ist die zufällig gewählte Quelle für die Simulation vorhanden und die Distanz kann einfach berechnet werden. Für die realen Daten wird das Ergebnis des MEG-Dipol Scans als Target verwendet.49

1. Einleitung

1.1. Motivation

Die Elektroenzephalographie (EEG) ist eine weit verbreitete, nicht-invasive Methode zur Messung neuronaler Prozesse [1]. Dazu werden auf der Kopfoberfläche Elektroden angebracht, um elektrische Spannungsunterschiede an verschiedenen Positionen auf dem Schädel aufzunehmen. Diese Spannungen entstehen durch postsynaptische Signale aktiver Neuronen. Um ein messbares Potential zu erzeugen, muss ein ganzer Patch von Neuronen gleichzeitig aktiv sein. Durch die hohe Ausbreitungsgeschwindigkeit der Spannungen ist die zeitliche Auflösung des EEG sehr gut. Jedoch ist die räumliche Auflösung aufgrund von Volumenleitereffekten gering. Die gemessenen Potentiale stellen daher nur einen indirekten Hinweis auf die zugrundeliegenden neuronalen Quellen dar [2], [3]. Deshalb wurde EEG lange Zeit vorrangig eingesetzt, um ereigniskorrelierte Potentiale (ERP) und zeitliche Dynamiken neuronaler Aktivität zu beobachten [4], [5].

Dementsprechend wuchs das Interesse an einer genaueren räumlichen Auflösung und Lokalisierung der zugrunde liegenden Quellen, um genauere Informationen für etwa klinische oder forschungsorientierte Arbeiten zu erhalten. Die räumliche Auflösung kann man mithilfe von Algorithmen zur Suche der zugrundeliegenden neuronalen Quelle, Quelllokalisierung, verbessern. Die zwei wichtigsten Ansätze zur klassischen Quelllokalisierung sind die nicht parametrischen Methoden, die keine feste Anzahl an Dipolen erwarten und die parametrischen Methoden. Die meisten der Methoden basieren dabei auf der Minimum Norm Estimation (MNE) oder Beamforming Techniken [6], [7], [8].

Die Lokalisierung einer neuronalen Quelle beim EEG ist aufgrund der Nicht-Eindeutigkeit und dem Einfluss von Rauschen jedoch ill-posed. Dies bedeutet, dass es unendlich viele mathematische Lösungen zum Erfüllen der Gleichungsbedingungen gibt [9].

Mit dem inversen Problem wird sich in Kapitel 2.1. *EEG und Quelllokalisierung* weiter auseinandergesetzt.

Die in dieser Arbeit entwickelte Methode zur Quelllokalisierung kann sowohl für EEG- als auch für MEG-Daten genutzt werden, im Rahmen der Arbeit wird jedoch nur die EEG-Quelllokalisierung angewandt, da dann die MEG-Daten zur Validierung verwendet werden können.

1.2. Relevanz der EEG-Quelllokalisierung für Forschung und Praxis

Die EEG-Quelllokalisierung wird hauptsächlich in der neurowissenschaftlichen Forschung und der klinischen Anwendung verwendet. In beiden Bereichen führt eine bessere Ortsauflösung zu präziseren Schlussfolgerungen und mehr Möglichkeiten.

Eine präzise Lokalisierung bietet der Neurowissenschaft die Möglichkeit, aktivitätsbedingte Signale nicht nur groben Hirnarealen, sondern spezifischen Neuronenpatches zuzuordnen. Dadurch lassen sich Interaktionen und dynamische Verschaltungen von neuronalen Netzen detaillierter untersuchen.

Ein weiterer Vorteil wären geringere Kosten für Forschungsteams, die anstelle teurer fMRT-Studien mehr EEG basierte Experimente durchführen könnten. Ein Beispiel hierfür ist das LORETA-Neurofeedback, bei dem die mithilfe eines EEG gemessene Aktivität einzelner Gehirnbereiche

aktiv vom Probanden beeinflusst wird, um langfristige Änderungen hervorzurufen, wofür sonst häufig fMRTs verwendet werden [10], [5], [11].

Die im Forschungsumfeld gewonnenen Erkenntnisse und erforschten Methoden bilden die Basis für die medizinische Praxis mit präziseren Diagnosen und Therapieverfahren.

In der neurologischen Diagnostik und Therapie ermöglicht eine höhere Ortsauflösung genauere Befunde und mehr Behandlungsmöglichkeiten. So können psychische und neurodegenerative Erkrankungen wie Schizophrenie, Depression, Alzheimer oder Parkinson durch exaktere Zuordnung von Aktivitätsmustern zuverlässiger und früher erkannt werden. Ein Feld, bei dem die EEG-Quelllokalisierung häufig angewendet wird, ist Epilepsie. Weltweit sind laut der Weltgesundheitsorganisation WHO etwa 50 Millionen Menschen davon betroffen [12] [13]. Mithilfe der Quelllokalisierung lässt sich der Epilepsieherd vor einer Operation identifizieren, was Eingriffe gezielter und mit weniger Risiko für anderes Gewebe ermöglicht.

1.3. Aufbau der Arbeit

Die vorliegende Arbeit gliedert sich in 7 Kapitel, deren Aufbau hier kurz erläutert wird.

Zunächst werden die biologischen und mathematischen Grundlagen der EEG-Quelllokalisierung und die wichtigsten angewendeten Deep Learning Methoden erklärt. Danach werden aktuelle Ansätze zur Quelllokalisierung mit Deep Learning Methoden analysiert. Auf dieser Basis wird die Forschungslücke herausgearbeitet und das Ziel der Arbeit gesetzt. Weiterhin werden die in der Arbeit verwendeten Methoden beschrieben. Kapitel 5 beschreibt die Ergebnisse der Experimente, so dass diese daraufhin diskutiert und eingeordnet werden können. Zum Schluss wird ein Ausblick gegeben und die Arbeit zusammengefasst.

2. Theorie

Die folgenden Abschnitte beleuchten die theoretischen Grundlagen der EEG-Quelllokalisierung sowie die Anwendung von Deep Learning in diesem Kontext.

2.1. EEG und Quelllokalisierung

2.1.1. Grundlagen des EEG

2.1.1.1. Physikalische und biologische Grundlagen des EEG

Um die Grundlagen des EEG zu verstehen, wird zuerst das Gehirn und vor allem das einzelne Neuron in Hinsicht auf die Signalentstehung betrachtet.

Das Gehirn bildet die zentrale Basis für Wahrnehmung, Kognition und Verhalten des Menschen. Es besteht aus etwa 86 Milliarden Neuronen, deren Aufgabe die Aufnahme und Weiterverarbeitung von Informationen ist [14]. Über spezialisierte Verbindungen (Synapsen) werden elektrische und chemische Signale weitergeleitet. Informationen werden dabei durch Aktionspotentiale und synaptische Übertragung kodiert und weitergegeben. Obwohl die Komplexität einzelner Neuronen beschränkt ist, führt das Zusammenspiel der großen Anzahl

vernetzter Neuronen durch ein Phänomen der Emergenz zu unserem Bewusstsein, Denken und Erleben [15], [16].

Ein Neuron hat alle grundlegenden Bestandteile einer lebenden Zelle wie Zellkern, Mitochondrien oder Zellkörper. Das Besondere an einem Neuron ist seine elektrische Aktivität und Leitfähigkeit durch spezifische Permeabilität und Integration von Pumpen für Ionen. So empfängt es an den Dendriten, kleinen Fortsätzen am Soma (Zellkörper), elektrische Signale. Diese werden dann innerhalb der Zelle am Axonhügel summiert und falls das Potential einen Schwellenwert überschreitet, wird dann ein Aktionspotential ausgelöst. Dieses wird dann über das Axon zu den Endknöpfchen/Synapsen weitergeleitet. Um das Axon herum liegen häufig isolierende Myelinscheiden, die die Übertragungsgeschwindigkeit zusammen mit den Ranviersche Schnürringen durch ihre isolierende Eigenschaft erhöhen. Die Synapsen sind wiederum am nächsten Neuron angebunden und können so das Aktionspotential weitergeben [15], [16]. *Abbildung 1* zeigt den schematischen Aufbau einer Nervenzelle.

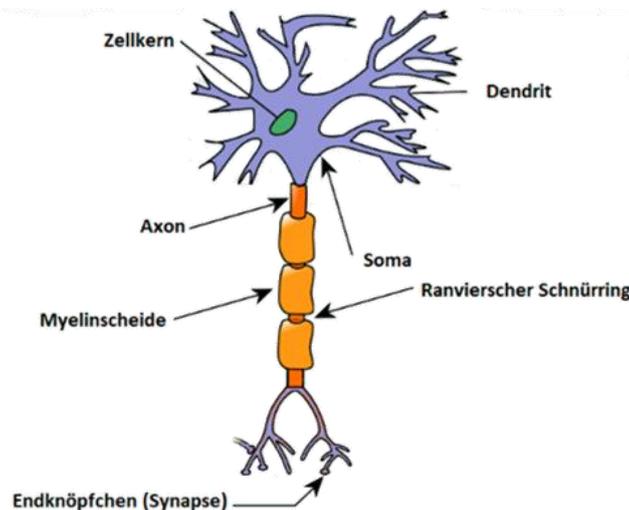


Abbildung 1: Schematische Darstellung einer Nervenzelle mit den wesentlichen Bestandteilen Zellkern, Dendriten, Soma, Axon, Myelinscheide, Ranvierscher Schnürring und den Endknöpfchen (Synapse) [17].

Grundsätzlich halten Neuronen in Ruhe ein Membranpotential von etwa -70 mV. Dies wird durch die Permeabilität der Zellmembran und spezifische Ionenpumpen aufrechterhalten. Ändert sich das Verhalten der Membran, so können entsprechende Ionen entlang den Konzentrations- und elektrischen Gradienten über die Membran strömen und das Membranpotential ändern.

Erreicht diese Depolarisierung einen Schwellenwert von etwa -55 mV, so wird ein Aktionspotential ausgelöst. Das Aktionspotential beginnt mit der Öffnung von spannungsgesteuerten Na⁺ Kanälen, die eine weiter Depolarisation erzeugen. Direkt danach werden diese Kanäle wieder geschlossen und K⁺ Kanäle geöffnet. Dadurch wird die Zelle repolarisiert und kommt zurück in den Grundzustand von -70 mV. Das Aktionspotential bildet sich dabei meist am Axonhügel und setzt sich von dort aus über das Axon zu den Synapsen fort.

An einer Synapse angekommen führt das Aktionspotential zum Öffnen spannungsabhängiger Ca²⁺ Kanäle, was zur Vesikel Fusion und Neurotransmitterfreisetzung in den synaptischen Spalt führt. Die Neurotransmitter diffundieren nun über den synaptischen Spalt und binden dann an

Rezeptoren des postsynaptischen Neurons, was dort wiederum eine Polarisierung auslöst. Dies wird postsynaptisches Potential (PSP) genannt [2].

Bei den postsynaptischen Potentialen unterscheidet man zwischen exzitatorischen postsynaptischen Signalen (EPSP) und inhibitorischen postsynaptischen Potentialen (IPSP). Ein exzitatorisches postsynaptisches Potential wird durch die Aktivierung von Glutamat-Synapsen, die meist an den apikalen Dendriten der Pyramidenzellen liegen, erzeugt. Es strömen positive Ionen wie Na^+ oder Ca^{2+} in das Zellinnere und erzeugen eine lokale Depolarisation.

Treten hingegen durch GABA-Synapsen, die meist am Zellkörper oder den proximalen Dendriten der Pyramidenzellen liegen, negativ geladene Ionen (Cl^-) in die Zelle oder positive Ionen (K^+) aus der Zelle heraus, so wird die Zellmembran hyperpolarisiert. Dies beschreibt ein inhibitorisches postsynaptisches Potential [15], [18]. Der Einfluss dieser Signale wird in *Abbildung 2* dargestellt.

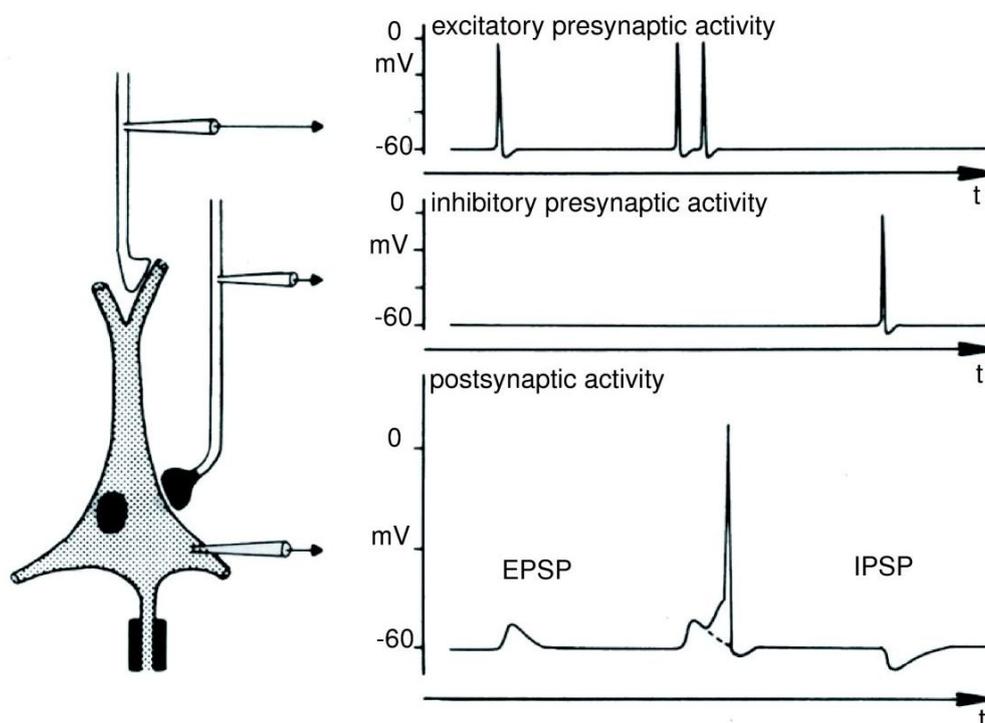


Abbildung 2: Darstellung des Einflusses von inhibitorischen und exzitatorischen präsynaptischen Signalen auf ein Neuron. Nach [2].

Bei einem exzitatorischen postsynaptischen Potential wird durch den Einstrom von positiv geladenen Ionen das Zellinnere depolarisiert. Dadurch sinkt allerdings die Konzentration dieser Ionen im extrazellulären Raum und der Raum wird negativer geladen. Von außen betrachtet verhält sich dieser Effekt wie ein elektrisches Dipolmoment. Bei einem inhibitorischen postsynaptischen Potential ist das elektrische Feld des Dipols entgegengesetzt zu dem eines EPSP-Dipols. *Abbildung 3* zeigt das entstehende Potential eines tangentialen Dipols.

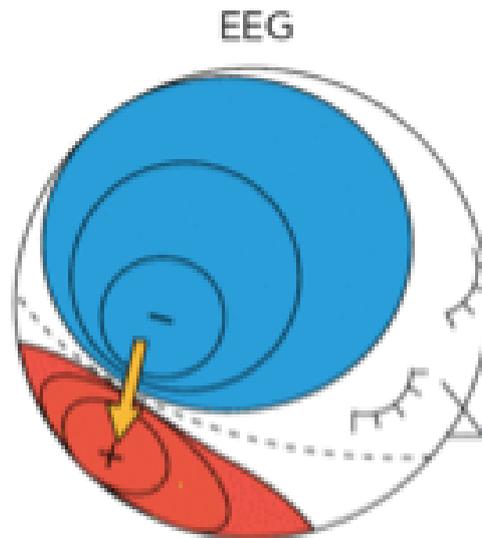


Abbildung 3: Potential an der Kopfoberfläche für einen tangentialen Dipol (gelber Pfeil). In rot wird das positive Potential angezeigt, das negative in blau [2].

Der entstehende Dipol eines einzelnen Neurons reicht allerdings nicht aus, um ein messbares Signal auf der Kopfoberfläche zu erzeugen. Die Spannung eines einzelnen exzitatorischen postsynaptischen Potential innerhalb der Zelle liegt meist im Bereich von $0,1 - 1 \text{ mV}$ und das resultierende elektrische Feld wirkt außerhalb der Zelle nicht signifikant. Sind jedoch viele Neuronen in einer Region synchron aktiv, so summieren sich die Dipolmomente und erzeugen so ein mit dem EEG messbares Feldpotential auf der Kopfoberfläche im Bereich mehrerer Mikrovolt. Um dies zu erreichen, müssen grob 10000 oder mehr Neuronen aktiv sein. Je nach Gehirnbereich bedeutet dies ein anderes Volumen. Eine Studie zeigt allein im Temporalen Neokortex eine Neuronendichte von 10000 bis 55000 Neuronen pro mm^3 je nach betrachteter Schicht [19]. Weitere Arbeiten bestätigen diese Dichten über den gesamten Neokortex [20], [21]. Dies würde für 10000 Neuronen ein Volumen von $\approx 0,2 - 1 \text{ mm}^3$ bedeuten.

Entsteht nun durch die synchrone Aktivität mehrerer Neuronen ein genügend großes Signal, so breitet sich das durch den Stromfluss entstehende elektrische Feld durch die umliegenden Gewebe aus. Dies sind meist das Gehirn selbst (bestehend aus grauer Substanz, weißer Substanz und Liquor), Schädelknochen und die Kopfhaut. Jedes dieser Gewebe hat dabei eine unterschiedliche Leitfähigkeit. Zusammen bilden diese Gewebe ein dreidimensionales, inhomogenes Volumenleiter-System, in dem sich die elektrischen Felder ausbreiten. In diesem Volumenleiter-System können sich die Felder über größere Distanzen ausbreiten und so an verschiedenen Elektroden gleichzeitig gemessen werden. So kann ein Signal an einer Elektrode von nahen, schwachen Neuronen-Patches genauso beeinflusst werden, wie von weiter entfernten Neuronen-Patches. Dieses Problem nennt man Volumenleitereffekt (Volume Conduction Effect) [10].

Die Elektroden des EEGs messen dann die ankommenden Spannungsdifferenzen zwischen den Elektroden. Die Spannung ist dabei der Potentialunterschied zwischen mehreren Elektroden, weshalb EEG-Systeme immer mindestens aus 2 Elektroden bestehen. Die absoluten Ladungen können dabei nicht direkt ausgelesen werden [5].

Aufgrund der Messeigenschaften des EEGs tragen nicht alle Neuronen-Typen und somit nicht alle Gehirnareale gleich zum gemessenen Signal bei. Besonders die Aktivität der Pyramidenzellen in

der Großhirnrinde ist in EEG-Messungen gut erkennbar. Dies liegt an der Anatomie und Ausrichtung der Pyramidenzellen, die optimal zu EEG-Messungen passen. Die parallele Anordnung der Dendriten in kortikalen Säulen erleichtert die Synchronisation über viele Neuronen. Zudem liegen die Pyramidenzellen in der Großhirnrinde und haben so einen geringen Abstand zur Kopfoberfläche. Dies verringert die Auswirkungen von umliegendem Gewebe und so wird das Signal bis zu den EEG-Elektroden weniger abgeschwächt und verändert.

Das EEG misst die Spannungsunterschiede an der Kopfoberfläche. Dadurch sind radiale Dipole sehr gut zu erkennen, da diese starke Potentiale an der Oberfläche erzeugen. Tangentiale Dipole führen durch Volumenströme ebenfalls zu messbaren Potentialverteilungen, sind im Vergleich zu radialen Dipolen aber weniger stark ausgeprägt. So sind vor allem oberflächennahe radiale Quellen besonders gut mit dem EEG messbar.

2.1.1.2. Unterschiede zur Magnetoenzephalographie

Das MEG erfasst mithilfe von SQUID-Sensoren das von intrazellulären Strömen erzeugte Magnetfeld außerhalb des Kopfes. Radiale Dipole erzeugen in einem sphärischen Leiter aufgrund der Symmetrie kein externes Magnetfeld, die Feldlinien bleiben im Volumen geschlossen. Deshalb sind radiale Quellen im MEG kaum detektierbar. Tangentiale Quellen hingegen bilden durch den Stromweg parallel zur Oberfläche gut messbare Felder außerhalb des Kopfes. So bilden die beiden Methoden eine gute Synergie zusammen [10]. Es ist auch möglich, in einem Experiment gleichzeitig MEG- und EEG-Daten des Probanden zu erfassen. Dieses Verfahren heißt EMEG.

2.1.1.2. Mathematische Grundlagen

Aus mathematischer Sicht wird der biologisch-physikalische Vorgang der synchronen Neuronen-Aktivität und die entstehenden elektrischen Felder als Vorwärtsproblem betrachtet. Das Vorwärtsproblem beschreibt dabei mathematisch den Zusammenhang von aktiven Quellen im Gehirn zu den dazu gemessenen Signalen an den Elektroden.

Dabei ist das Vorwärtsproblem die Grundlage für die Lösung des inversen Problems zur Lokalisierung von neuronalen Quellen im EEG. Zudem findet es Anwendung in der Simulation von Daten in dieser Arbeit.

Für das Vorwärtsproblem wird ein möglichst realistisches Kopfmodell des Probanden benötigt. Dies beinhaltet alle möglichen Punkte, an denen ein aktiver Dipol gelegen sein kann. Der Raum aller möglichen Dipole ist der Quellraum. Zusätzlich sind in einem Kopfmodell noch die verschiedenen Leitfähigkeiten der verschiedenen Gewebe dargestellt. Ein persönliches Kopfmodell wird meist anhand von Magnetresonanztomographie (MRT) Aufnahmen und Diffusion-Tensor-Imaging (DTI) Bildern erstellt. Liegen keine spezifischen Daten über einen Probanden vor, so gibt es eine Reihe von Durchschnittsmodellen.

Eine aktive Quelle im Gehirn, bestehend aus vielen synchron aktiven Neuronen wird dabei vereinfacht als Punktquelle oder Punktdipol beschrieben. Im Optimalfall liegt die Punktquelle im Zentrum der aktiven Neuronen. Ein Dipol besteht dabei aus einer Position und einem Moment, bestehend aus Amplitude d und Orientierung e [13].

Der Dipol-Moment \mathbf{m} wird dabei wie folgt berechnet:

$$\mathbf{m} = d\mathbf{e} \quad (1)$$

Der Quellraum besteht aus allen möglichen Quellpunkten, also Punkten, an denen solch ein Dipol entstehen könnte.

Das aus der Aktivität von Neuronen entstehende elektromagnetische Feld besteht zum einen aus dem elektrischen Feld \mathbf{E} und zum anderen aus dem magnetischen Feld \mathbf{B} .

Um den genauen Zusammenhang zwischen der Spannung und den elektromagnetischen Feldern zu beschreiben, betrachten wir die 4 Maxwell-Gleichungen:

1. Gaußsches Gesetz der Elektrizität:

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\varepsilon_0} \quad (2)$$

Stellt den Zusammenhang zwischen elektrischer Feldstärke \mathbf{E} und der räumlichen Ladungsdichte ρ her.

2. Gaußsches Gesetz des Magnetismus:

$$\nabla \cdot \mathbf{B} = 0 \quad (3)$$

Es existieren keine magnetischen Monopole und das Magnetfeld \mathbf{B} ist quellenfrei.

3. Faradaysches Induktionsgesetz:

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \quad (4)$$

Beschreibt, dass die zeitliche Änderung eines Magnetfeldes ein elektrisches Feld \mathbf{E} induziert.

4. Ampèresches Gesetz mit Maxwell-Korrektur:

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{J} + \mu_0 \varepsilon_0 \frac{\partial \mathbf{E}}{\partial t} \quad (5)$$

Hier wird der Zusammenhang zwischen dem rotatorischen Anteil des Magnetfeldes \mathbf{B} , der elektrischen Stromdichte \mathbf{J} und der zeitlichen Änderung des elektrischen Feldes \mathbf{E} hergestellt [2], [22], [23].

Beim Modellieren von neurologischen Aktivitäten im Gehirn nimmt man üblicherweise die quasistatische Approximation der Maxwell-Gleichungen an. Das bedeutet, dass alle Felder und Ströme zu jedem Zeitpunkt als quasistatisch betrachtet werden. Man vernachlässigt dabei die induzierten Magnetfelder (zeitliche Änderung des Magnetfeldes) im Faraday'schen Gesetz, sodass für Gleichung (4) nun gilt:

$$\nabla \times \mathbf{E} = 0 \quad (6)$$

Im Ampèreschen Gesetz lässt man den Anteil des Displacement-Stroms (Verschiebungsstrom) weg. Unter diesen Annahmen reduziert sich das Ampèresche Gesetz (5) zu:

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{J} \quad (7)$$

In der Realität breiten sich Felder nicht unendlich schnell aus und auch die Gehirnaktivität ändert sich stetig, da jedoch die Änderung der neuronalen Aktivität deutlich langsamer als die Ausbreitungsgeschwindigkeit ist, ist der Fehler, der durch die quasistatische Näherung entsteht, so gering, dass er praktisch vernachlässigt werden kann [8], [24], [25], [26].

Um die bekannten Leitfähigkeiten der Kopfgewebe zu nutzen, betrachten wir zusätzlich das Ohmsche Gesetz. Hier wird die Stromdichte \mathbf{J} in A/m^2 durch die Multiplikation des elektrischen Feldes \mathbf{E} in V/m mit dem Leitfähigkeitstensor σ (im isotropen Fall Skalar) in S/m berechnet:

$$\mathbf{J} = \sigma \mathbf{E} \quad (8)$$

Die Divergenz der Stromdichte stellt die Stromquellendichte dar:

$$\nabla \cdot \mathbf{J} = I_m \quad (9)$$

Die Leitfähigkeit kann dabei isotrop und anisotrop sein. Im Gehirn sind isotrope Gewebe vor allem graue Substanz, Kopfhaut, Liquor und Blut. Hier sind die Leitfähigkeiten in alle Raumrichtungen gleich. Dies kommt bei grauer Substanz und Kopfhaut vor allem durch die ungeordnete Verteilung der Zellkörper. Der hohe Wassergehalt führt zudem zu weiterer Isotropie, da es die Diffusion in alle Richtungen erhöht und so ähnlicher macht. Liquor und Blut sind beides Flüssigkeiten, weshalb sich die Ionen wieder in alle Richtungen frei ausbreiten können.

Anisotrope Gewebe sind im Gehirn vor allem die Schädelknochen, die weiße Substanz und die äußere Hirnhaut. Der Schädel besteht aus zwei Knochenschichten und einem schwammartigen Knochen (Spongiosa) dazwischen. Der feste Teil des Schädels hat dabei quer zur Oberfläche eine deutlich geringere Leitfähigkeit als entlang der Faserrichtung. Daher leitet der Schädel den Strom eher seitlich entlang der Knochenstruktur als direkt durch den Knochen. Dies reduziert das gemessene Signal an der Kopfoberfläche stark, jedoch verteilen sich die Potentiale über große Flächen und können so immer noch gut gemessen werden. Der große isolierende Effekt des Schädels ist auch der Grund, wieso bei Anwendungen mit sehr hoher Anforderung an die Präzision der Lokalisierung Elektroden innerhalb des Schädels angebracht werden. Die weiße Substanz besteht aus myelinisierten Axonen, die Neuronen über große Distanzen hinweg verbinden. Das Myelin wirkt dabei stark isolierend und verhindert das seitliche Austreten von Strömen, im Inneren des Axons ist die Signalweiterleitung entlang der Faserrichtung optimal. Durch diese Unterschiede in der Leitfähigkeit abhängig von der Raumrichtung ist weiße Substanz

also stark anisotrop. Dies erschwert die Interpretation von EEG-Daten weiter und muss so auch bei der Analyse beachtet werden [27].

Bei anisotropen Geweben ist die Leitfähigkeit richtungsspezifisch, weshalb σ dann statt als Skalar, als 3x3 Tensor angewendet wird:

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} \quad (10)$$

Wobei die Diagonalelemente die Leitfähigkeiten in den drei Raumachsen beschreiben und die Nichtdiagonalelemente die Kopplungen zwischen den verschiedenen Richtungen. Das bedeutet, dass eine elektrische Feldkomponente in einer Richtung eine Stromdichte in einer anderen Richtung induzieren kann [8], [13].

Um die aus den neuronalen Quellen resultierende Verteilung des elektrischen Potentials V im Volumenleitermodell des Kopfes zu berechnen und somit das Vorwärtsproblem des EEG zu lösen, verwendet man die Poisson-Gleichung:

$$\nabla \cdot (\boldsymbol{\sigma} \cdot \nabla V) = I_m \text{ in } \Omega \quad (11)$$

Wobei $\boldsymbol{\sigma}$ die Leitfähigkeitsverteilung im Kopf ist, I_m die Stromquellendichte der neuronalen Quellen darstellt und Ω das Volumenleitermodell des Kopfes repräsentiert

Beim Modellieren von Kopfmodellen und den damit verbundenen Volumenleitermodellen sind die Randbedingungen (Boundary Conditions) von entscheidender Bedeutung beim Vorwärtsproblem. Sie definieren das Verhalten von elektrischen Strömen und Potentialen an den Grenzen des Modells [8], [13].

Die Neumann Randbedingung setzt die Ableitung der Funktion des Volumenleitermodells auf den Grenzen des Modells fest. Dies entspricht in der Realität einem elektrischen Fluss. Der Fluss darf sich an der Grenze nicht ansammeln, sondern muss nach der Neumann Bedingung nach dem Verlassen eines Mediums weiterfließen.

Die ganze elektrische Spannung, die ein Gewebe mit der Leitfähigkeit σ_1 verlässt, wandert über die Grenze in eine Region mit der Leitfähigkeit σ_2 [8]:

$$\mathbf{J}_1 \cdot \mathbf{e}_n = \mathbf{J}_2 \cdot \mathbf{e}_n \quad (12)$$

$$(\boldsymbol{\sigma}_1 \cdot \nabla V_1) \cdot \mathbf{e}_n = (\boldsymbol{\sigma}_2 \cdot \nabla V_2) \cdot \mathbf{e}_n \quad (13)$$

Wobei \mathbf{e}_n die Normalkomponente an der Grenzfläche darstellt.

Bei homogenen Räumen gilt ein spezieller Fall der Neumann-Randbedingung. Im menschlichen Gehirn bedeutet dies, dass durch die extrem niedrige Leitfähigkeit von Luft außerhalb des Kopfes, keine Spannung vom Kopfgewebe in die Luft fließen kann [8], [13]. Dies bedeutet:

$$\mathbf{J}_1 \cdot \mathbf{e}_n = 0 \quad (14)$$

$$(\boldsymbol{\sigma}_1 \cdot \nabla V_1) \cdot \mathbf{e}_n = 0 \quad (15)$$

Elektrische Potentiale sind immer relativ und beim Messen mit Elektroden werden nur die Unterschiede gemessen. Deshalb hat die durch die Poisson-Gleichung dargestellte Potentialverteilung unendlich viele Lösungen durch additive Konstanten. Um die additiven Konstanten zu entfernen, wird eine Referenzelektrode gewählt und auf einen festen Wert gesetzt [13]:

$$V_{ref} = 0 \quad (16)$$

Nun werden alle berechneten Potentiale relativ zu diesem Referenzpunkt angegeben.

Aus den mathematischen Grundlagen und Bedingungen kann man nun ein mathematisches Modell des Volumenleiters Kopf erstellen. Hierzu wird bei fehlenden Informationen ein Standard/Durchschnitts-Modell verwendet. Um genauere und an den Probanden angepasste Modelle zu erstellen, wird zuerst anhand einer MRT-Aufnahme die Gewebeverteilung im Kopf betrachtet. Danach wird den verschiedenen Geweben unterschiedliche Leitfähigkeiten zugewiesen und die Randbedingungen und Übergangsformeln definiert. Das daraus resultierende angepasste Kopfmodell beschreibt dann mathematisch, wie sich elektrische Aktivität an beliebigen Stellen ausbreiten würde und vor allem welches Signal an der Kopfoberfläche gemessen werden würde. Es ist also ein exakt an den Probanden angepasstes Vorwärtsmodell.

Eine weit verbreitete Methode, um in der Realität die Leitfähigkeiten des Kopfes in einem mathematischen Modell darzustellen, ist die Boundary Element Methode (BEM). Hier werden typischerweise die 3 Grenzen Gehirn-Schädelknochen, Schädelknochen-Kopfhaut und Kopfhaut-Luft des Kopfes, wie in *Abbildung 4* abgebildet, verwendet. Alles Gewebe zwischen den Grenzflächen wird als homogen und isotrop angenommen. Berechnungen werden durch Triangulierung der Grenzflächen vorgenommen. Durch diese Annahmen ist die BEM relativ schnell zu berechnen, erzeugt aber auch einen vergleichsweise großen Modellierungsfehler [28], [29].

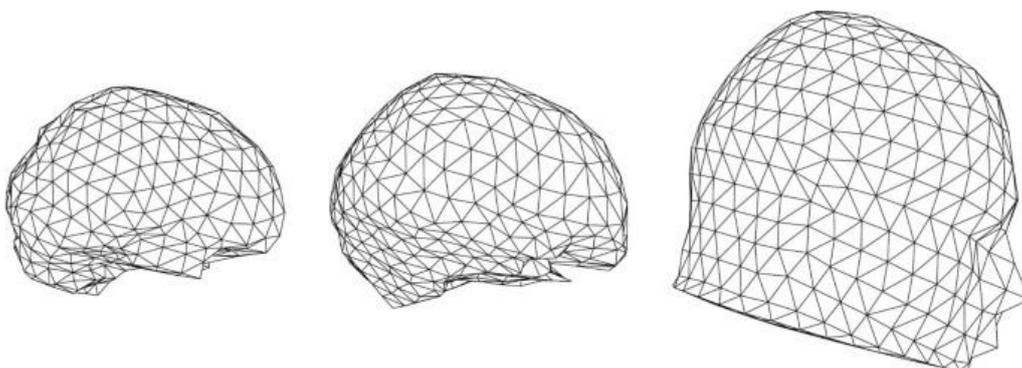


Abbildung 4: Beispiele der 3 häufig genutzten Grenzflächen bei der BEM-Methode: Gehirn-Schädelknochen, Schädelknochen-Kopfhaut und Kopfhaut-Luft. Die Flächen werden mithilfe von Dreiecken realisiert. [28]

Die Finite Element Methode (FEM) löst die Poisson-Gleichung (11) in einem realistischen Kopfmodell, indem das Volumen in 3D-Elemente (meist lineare Tetraeder, seltener Hexaeder) wie in *Abbildung 5* gezeigt, unterteilt wird. Jedem Element sind dabei lokale polynomielle Basisfunktionen zugeordnet, mit denen das Potential innerhalb des Elements approximiert wird. So entsteht ein sehr großes, aber spärlich besetztes lineares Gleichungssystem.

Durch die deutlich feinere Auflösungsmöglichkeit können verschiedene Gewebearten mit je eigenen Leitfähigkeiten versehen und auch anisotrope Leitfähigkeiten verwendet werden. Dadurch steigt die Präzision des Kopfmodells gegenüber BEM Modellen, jedoch auch die zur Berechnung benötigten Ressourcen [28].

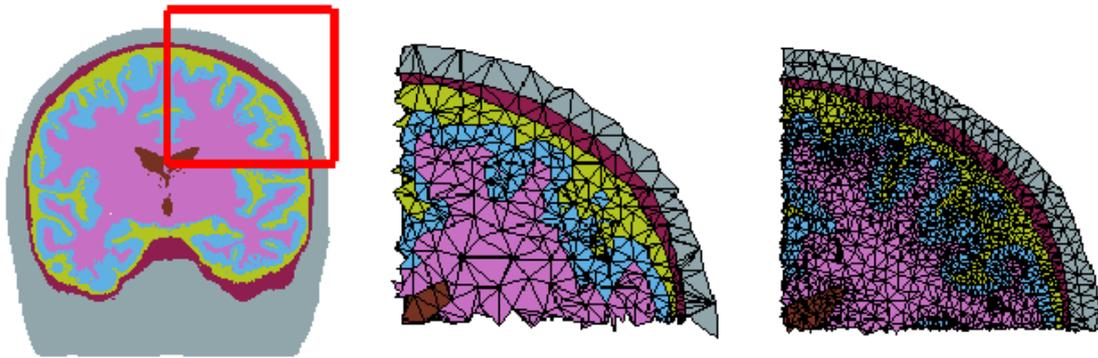


Abbildung 5: Darstellung von 2 verschiedenen feinen FEM-Modellen (mitte, rechts), die aus einem Originalmodell (links) berechnet wurden. Mitte und rechts zeigen den rot umrahmten Bereich in links [30].

Anhand des erstellten Kopfmodells, welches die verschiedenen Gewebe darstellt und mathematisch beschreibt, kann man nun die Leadfield-Matrix berechnen. Die Leadfield-Matrix $\mathbf{L} \in R^{q \times p}$ mit q Anzahl der Elektroden und p Anzahl der Dipole beschreibt dabei den Fluss der elektrischen Spannung von jedem Dipol zu jeder Elektrode. Dabei gilt:

$$\mathbf{M} = \mathbf{L}\mathbf{D} \tag{17}$$

Mit der Potentialverteilung \mathbf{M} , der Leadfield-Matrix \mathbf{L} und der Quellaktivität der Dipole \mathbf{D} .

Für einen spezifischen Zeitpunkt t gilt dann:

$$\mathbf{M}_t = \underbrace{\begin{bmatrix} \mathbf{M}(elec_1) \\ \vdots \\ \mathbf{M}(elec_q) \end{bmatrix}}_{q \times 1} = \underbrace{\begin{bmatrix} I(\mathbf{r}_1, \mathbf{r}_{source_1}) & \cdots & I(\mathbf{r}_1, \mathbf{r}_{source_p}) \\ \vdots & \ddots & \vdots \\ I(\mathbf{r}_q, \mathbf{r}_{source_1}) & \cdots & I(\mathbf{r}_q, \mathbf{r}_{source_p}) \end{bmatrix}}_{q \times p} \underbrace{\begin{bmatrix} d_1 \mathbf{e}_1 \\ \vdots \\ d_p \mathbf{e}_p \end{bmatrix}}_{p \times 1} \tag{18}$$

Wobei $\mathbf{r}_1, \dots, \mathbf{r}_q$ die Positionen der Elektroden sind und $\mathbf{r}_{source_1}, \dots, \mathbf{r}_{source_p}$ die Positionen der Dipole im Quellraum. Nach [6], [13].

2.1.1.3. Signalaufzeichnung

Die zuverlässige Erfassung der elektrischen Hirnaktivität bildet die Basis sämtlicher Analysen in der EEG-Quelllokalisierung. Die Qualität des aufgenommenen Signals ist dabei abhängig von der Wahl und Platzierung der Elektroden, sowie dem Auftreten von Artefakten. Die Qualität begrenzt dabei auch die Aussagekraft aller daraus resultierenden Analysen.

Die Magnetoenzephalografie (MEG) misst statt der elektrischen Potentiale die von den elektrischen Strömen erzeugten Magnetfelder. Da Magnetfelder deutlich weniger von den im Gehirn auftretenden Geweben abgeschirmt werden, sind MEG -Aufnahmen weniger anfällig für Leitfähigkeitsartefakte. Darüber hinaus ist MEG tendenziell auch für Muskel- oder Bewegungsartefakte weniger empfindlich.

Neben der korrekten Elektrodenplatzierung gibt es noch weitere mögliche Artefakte, die bei EEG- oder MEG-Messungen auftreten können. Diese beinhalten Oszillationen durch Augenbewegungen (EOG), Muskelaktivität (EMG), Herzaktivität (EKG) und externe elektromagnetische Störungen [32].

2.1.2. Das inverse Problem in der EEG- Analyse

2.1.2.1. Definition des inversen Problems

Beim Vorwärtsproblem werden aus bekannten Quellen und einem definierten Kopfmodell die Potentialverteilungen an den Elektroden berechnet. Das inverse Problem beschreibt dabei den umgekehrten Prozess, also aus den an der Kopfoberfläche gemessenen elektrischen Potentialverteilungen (dem „Output“) auf die zugrunde liegenden neuronalen Aktivitätsquellen (den „Input“) im Gehirn zu schließen.

Das inverse Problem ist dabei „ill-posed“. Das bedeutet, dass mehrere unterschiedliche Quellkonfigurationen dieselben Potentiale an der Kopfoberfläche erzeugen können.

Dies liegt zum einen an der begrenzten Anzahl der EEG-Elektroden im Vergleich zur Vielzahl möglicher Quellen im dreidimensionalen Gehirn:

$$p \gg q \quad (\text{in dieser Arbeit } 14909 \gg 54)$$

Dadurch entsteht ein großer Nullraum in der Leadfield-Matrix.

Zum anderen misst das EEG nur die 2D Projektion des elektrischen Potentials auf der Kopfoberfläche, während die Quellen im 3D-Volumen verteilt sind. So bleibt das Gesamtsystem zudem unterbestimmt. Ohne zusätzliche Einschränkungen lässt sich somit keine eindeutige Lösung ableiten [8].

Ein weiteres Problem bei realen EEG-Messungen ist das Rauschen, welches mehrere Quellen haben kann und nicht völlig zu entfernen ist. Wegen des Rauschens werden häufig Regularisierungsverfahren verwendet, um die Rekonstruktion robuster zu machen. Diese haben zusätzlich die Eigenschaft, dass sie meist die Unstetigkeit des inversen Problems mindern.

Zu den gängigen Ansätzen, um robuste Lösungen zu erzielen, zählen vor allem dipolbasierte Modelle, bei denen ein begrenzter Satz von aktiven Quellen angenommen wird, sowie verteilte

Verfahren wie sLORETA (standardized low-resolution brain electromagnetic tomography) oder Bayesianische Methoden, die die Lösung durch Regularisierung stabilisieren [7], [8].

Eine gute Lösung des inversen Problems hängt also von der Qualität der Messdaten, der Genauigkeit des Volumenleitermodells und angewendeten Regularisierungen ab.

2.1.2.2. Mathematische Darstellung des inversen Problems

Die Formel des Vorwärtsproblems beschreibt, wie sich die Potentialverteilung \mathbf{M} aus der Aktivität der Dipole \mathbf{D} , der Leadfield-Matrix \mathbf{L} und meist noch einem Rauschen \mathbf{n} zusammensetzt:

$$\mathbf{M} = \mathbf{L}\mathbf{D} + \mathbf{n} \quad (19)$$

Beim inversen Problem ist die Potentialverteilung \mathbf{M} durch die Messungen der EEG-Elektroden bekannt. Die Leadfield-Matrix wird entweder aus einem Durchschnitts-Kopfmodell oder einem persönlichen Kopfmodell entnommen. Gesucht ist die Konfiguration der aktiven Dipole im Quellraum:

$$\mathbf{D} = \mathbf{L}^{-1}(\mathbf{M} - \mathbf{n}) \quad (20)$$

Da die Matrix \mathbf{L} jedoch nicht quadratisch ist ($p \gg q$) und das Problem unterbestimmt ist (ill-posed), lässt sich \mathbf{L} nicht einfach direkt invertieren.

Üblicherweise wird die Lösung deshalb als Regularisierungsproblem dargestellt. Ein weit verbreiteter Ansatz ist die Tikhonov-Regularisierung, bei der ein Minimierungsproblem aufgestellt wird [6], [13].

2.1.3. Klassische Methoden der Quelllokalisierung

Es gibt viele verschiedene klassische Methoden zur Quelllokalisierung. Diese beinhalten Dipol-Modelle, verteilte Ansätze oder bayesianische Ansätze [2], [6], [7]. In dieser Arbeit wird von den klassischen Methoden der Dipol Scan verwendet, weshalb dieser kurz erklärt wird:

Der Dipol Scan ist eine Methode, um eine oder eine kleine Anzahl an aktiven Quellen zu finden, die die größte Menge an Varianz des Signals vorhersagen. Dafür wird eine Grid-Search genutzt, die für jeden Punkt im Quellraum die Differenz zwischen simulierten und den gemessenen Werten berechnet. Der oder die Punkte mit der niedrigsten Differenz werden als Quelle der Aktivität interpretiert [13].

2.2. Deep Learning

2.2.1. Grundlagen Deep Learning

Methoden des Maschinellen Lernens sind heute in den meisten Bereichen in Forschung und Anwendung etabliert. Besonderes Potential bietet dabei das Deep Learning, eine Unterkategorie des Maschinellen Lernens. Deep Learning bezieht sich dabei meistens auf "tiefe", mehrstufige künstliche neuronale Netze, die in der Lage sind, beliebige Muster und Merkmale zu erkennen. Im Folgenden werden die theoretischen Grundlagen der Architektur und des Trainings dieser Netze erläutert.

Das künstliche Neuron (Perzeptron) ist das kleinste vollständige Bauteil eines neuronalen Netzes und bildet wie auch im biologischen Gehirn eine einfache Berechnung ab.

Dabei wird der Input x_1, \dots, x_m jedes Eingangssignal mit einem entsprechenden Gewicht w_1, \dots, w_m multipliziert und auf die Summe dann ein „Bias“ b aufaddiert. Diese Gewichte werden im Trainingsvorgang so angepasst, dass die Vorhersage des Netzes möglichst nah an die „ground truth“, also den tatsächlichen Wert der Zielvariable herankommt.

$$z = \sum_{i=1}^m w_i x_i + b \quad (21)$$

Um auch komplexere Funktionen zusammenstellen zu können, wird auf den Output z dann eine Aktivierungsfunktion a angewendet:

$$y = a(z) \quad (22)$$

Durch die nicht-linearität der Aktivierungsfunktionen eines Neurons lassen sich dann mit einem gesamten neuronalen Netz auch komplexere nichtlineare Funktionen beschreiben. Bei der Wahl der Aktivierungsfunktion gibt es verschiedene mögliche nicht-lineare Funktionen zur Auswahl [33] [34].

Schaltet man nun mehrere Neuronen in Reihe, so spricht man von einem Multi-Layer-Perzeptron (MLP). Dabei ist jedes Neuron einer Schicht (Layer) mit jedem Neuron der unmittelbar vorhergehenden Schicht verbunden. Erreicht das Multi-Layer-Perzeptron eine ausreichende Größe, spricht man von einem "fully-connected feed-forward Netz" (FCNN).

Ein Neuron j im Layer l erhält einen Eingangswert pro Neuron im vorhergehenden Layer $l - 1$. Damit liegt auch die Zahl lerner Parameter für das Neuron bei $n_{l-1} + 1$, mit n_{l-1} Gewichten und einem Bias [35].

$$z_j^l = \sum_{i=1}^{n_{l-1}} w_{ij}^l a_i^{l-1} + b_j^l \quad (23)$$

Um die Qualität und Korrektheit einer Vorhersage eines neuronalen Netzes zu quantifizieren, werden Verlustfunktionen verwendet. Diese beschreiben die Abweichung der Vorhersage vom optimalen gewünschten Ergebnis. Während des Trainings soll diese Funktion, also die Abweichung vom optimalen Ergebnis, iterativ minimiert werden. Die am häufigsten verwendeten

Verlustfunktionen für Regressionsprobleme sind der Mean Absolute Error (MAE) und der Mean Squared Error (MSE):

$$MAE(\mathbf{x}, \mathbf{y}) = \frac{1}{n} \sum_{i=1}^n |x_i - y_i|$$

(24)

Dadurch beschreibt der MAE die durchschnittliche absolute Abweichung von der Vorhersage \mathbf{x} und dem Zielwert \mathbf{y} . Die Anzahl der Trainingsbeispiele ist mit n gegeben.

$$MSE(\mathbf{x}, \mathbf{y}) = \frac{1}{n} \sum_{i=1}^n (x_i - y_i)^2.$$

(25)

Der MSE ist die durchschnittliche, quadrierte Abweichung des vorhergesagten \mathbf{x} vom tatsächlichen Wert \mathbf{y} [33], [34].

Um die gewählte Kostenfunktion auf den Trainingsdaten zu minimieren, werden die trainierbaren Parameter (Gewichte und Biases) iterativ optimiert. Dies wird mithilfe des Gradientenabstiegsverfahrens und des Backpropagation Algorithmus erreicht.

Das Gradientenabstiegsverfahren bewegt die Parameter in Richtung des negativen Gradienten der Kostenfunktion, um diese möglichst zu verringern. Backpropagation liefert die nötigen Gradienten und damit die Updates für die einzelnen Gewichte und Biases [35]. Dafür werden mithilfe der Kettenregel die partiellen Ableitungen der Kostenfunktion nach allen Gewichten und Biases berechnet [33], [34]. Beim Mean Squared Error heißt diese Ableitungsfunktion:

$$\frac{\partial}{\partial w_i} MSE = \frac{2}{n} \sum_{s=1}^n x_i^s (\mathbf{w}^T \mathbf{x}^s - \mathbf{y}^s)$$

(26)

Der berechnete Gradient wird dann mit einer Lernrate α multipliziert, die den Einfluss des aktuellen Gradienten $\nabla L(\mathbf{w}^i)$ skaliert. Parameter \mathbf{w}^{i+1} folgt dann aus:

$$\mathbf{w}^{i+1} = \mathbf{w}^i - \alpha \nabla L(\mathbf{w}^i)$$

(27)

Beim Training eines neuronalen Netzes werden nun die Gewichte und der Biases mit jedem Update so angepasst, dass die Verlustfunktion minimiert wird. Der Gradient für die Änderung der Gewichte und Biases ergibt sich aus der Berechnung des Backpropagation-Algorithmus. Nun gibt es verschiedene Optimierungsstrategien, um mithilfe des Gradienten eine möglichst optimale Konvergenz der Gewichte zu erreichen.

Ein sehr weit verbreiteter Algorithmus ist Stochastic Gradient Descent (SGD). Weitere häufig verwendete Algorithmen sind Momentum, ADAGrad, RMSProp oder ADAM [36], [37].

Zum Start des Trainings werden die Werte aller Gewichte und Biases einmalig initialisiert. Die Initialisierung beeinflusst dabei direkt den ersten Datendurchlauf und damit die Gradienten vor dem ersten Update.

Bei der Initialisierung der Gewichte ist darauf zu achten, dass diese nicht mit 0 initialisiert werden dürfen, da sich sonst alle Neuronen genau gleich verhalten und keine individuellen Merkmale lernen können (Symmetrieproblem). Ebenso müssen die Gewichte untereinander unterschiedlich sein, damit jedes Neuron im Verlauf einen individuellen Anteil der gesamten abzubildenden Funktion lernen kann.

Beim Training werden die Updates der Gewichte mithilfe der Kettenregel vom „Ende“ bis zum „Anfang“ des Netzes geleitet. Werden die Gradienten dabei merklich kleiner oder größer als 1, so multipliziert sich dies auf und die Gradienten werden für Werte <1 sehr klein und für Werte >1 zu groß. Dies führt im ersten Fall zu vanishing Gradienten und stagnierendem Lernen in frühen Layern. Bei sehr großen Gradienten wird das Lernen instabil und die Gewichte machen zu große Sprünge bis hin zur Divergenz des Loss. Die geeignete Wahl einer Initialisierung kann auch Effekte wie exploding oder vanishing Gradienten abschwächen, indem es die erwartete Norm nahe bei 1 setzt.

Häufig werden zur Initialisierung zufällig normalverteilte Werte mit einem Durchschnitt von 0.0 gewählt. Varianten davon sind etwa die Xavier Initialisierung für ähnlichere Varianzen oder die Kaiming-He Initialisierung, die oft bei der Aktivierungsfunktion ReLU angewendet wird [33], [37], [38], [39].

Mit steigender Modellkomplexität wird es schwieriger, dass das trainierte Modell auch bei unbekanntem Daten gute Vorhersagen liefert, also gut generalisiert.

Eine Methode, um die Generalisierungsfähigkeit eines Netzes zu erhöhen, ist die Anwendung von Dropout. Hier werden in jedem Vorwärtsdurchlauf der Datensamples eine feste Anzahl an Neuronen pro Layer zufällig ausgewählt und für diesen Durchlauf deaktiviert. Durch das zufällige Deaktivieren von Neuronen, muss das Netz die Verarbeitungsschritte innerhalb des Netzes auf mehrere Neuronen aufteilen. Dies führt dazu, dass das Modell sich nicht zu sehr auf die genauen Daten anpasst, sondern eher die zugrundeliegende Verteilung der Daten lernt und so besser auf vorher ungesehenen Daten abschneidet. Dropout wird dabei nur während dem Training des Netzes angewendet [34], [40].

Um einen stabileren Lernverlauf zu erreichen, kann man zwischen den Layern des neuronalen Netzes Batch-Normalisierung einsetzen. Durch verschiedene Inputs in jedem Mini-Batch beim Training verschieben sich die Eingangsverteilungen für spätere Layer ständig, was das Lernen langsam macht und instabile Gradienten erzeugt. Dieses Phänomen wird „internal covariate shift“ genannt. Mithilfe von Batch-Normalisierung werden die Eingaben pro Mini-Batch und Neuron jeweils auf Mittelwert 0 und Varianz 1 skaliert, was zu gleichmäßigeren Aktivierungsverteilungen führt. Dadurch wird außerdem verhindert, dass extrem hohe oder niedrige Aktivierungen in den Sättigungsbereich der Aktivierungsfunktion gelangen. Dies führt zu gleichmäßigeren Gradienten, stabilerem Lernen und häufig zu schnellerer Konvergenz [41], [42].

Besonders in tieferen neuronalen Netzen tritt das Problem von exploding und vanishing Gradienten auf. Um auch in tieferen Netzschichten noch stabile Gradienten zu erhalten, ist es möglich zwischen den Layern Skip-Connections zu schalten. Skip-Connections fügen zwischen den Schichten des neuronalen Netzes Identitäts-pfade hinzu, z. B.:

$$\mathbf{y}_{l+1} = F_l(\mathbf{y}_l) + \mathbf{y}_l \tag{28}$$

Über diese Identität kann der Gradient dann ohne Abschwächung in weitere Schichten zurückfließen, was vor allem das Problem von vanishing Gradienten erleichtert [42].

Nachdem innerhalb eines künstlichen Neurons die gewichtete Summe der Eingaben zusammen mit einem Bias berechnet wurde, wird auf das Ergebnis eine Aktivierungsfunktion angewendet. Diese ist vor allem dafür da, um Nichtlinearitäten modellieren zu können. Ohne sie könnte das gesamte Netz nur lineare Abbildungen lernen. Die gebräuchlichsten Aktivierungsfunktionen sind etwa Rectified Linear Unit (ReLU) [43]:

$$a(z) = \max(0, z) \tag{29}$$

ReLU gibt für negative Werte von z genau 0 und für positive Werte genau z zurück.

Leaky ReLU [44]:

$$a(z) = \max(\alpha z, z) \tag{30}$$

Wobei $\alpha \ll 1$. Dadurch ist die Ableitung für $z < 0$ nicht direkt immer 0.

Sigmoid:

$$a(z) = \frac{1}{1 + e^{-z}} \tag{31}$$

Sigmoid weighted Linear Unit (SiLU) [45], [46]:

$$a(z) = z \cdot \text{sigmoid}(z) \tag{32}$$

Welches einen konstanten Output hat, für $z < 0$ nicht direkt 0 ist und mittlerweile häufig Anwendung findet.

Gaussian Error Linear Unit (GELU) [47]:

$$a(z) = zP(Z \leq z) \tag{33}$$

GELU hat einen ähnlichen Verlauf wie die SiLU-Funktion und wird beispielsweise häufig in Tranformer-Architekturen verwendet. Eine graphische Darstellung der einzelnen Funktionen ist in *Abbildung 6* gegeben.

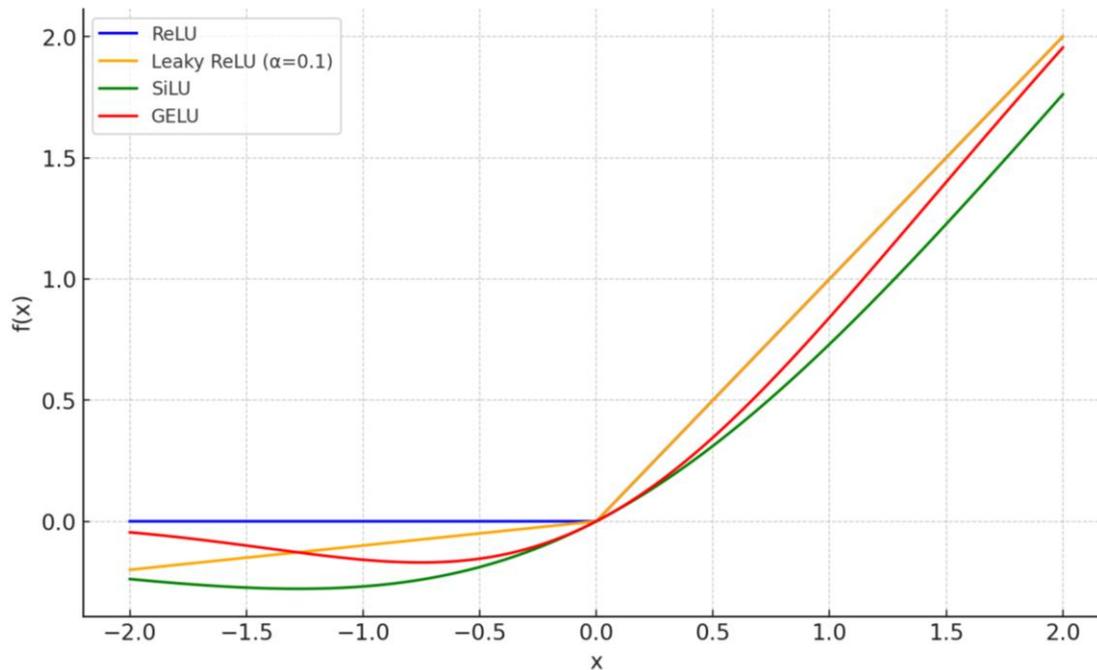


Abbildung 6: Graphische Darstellung der Aktivierungsfunktionen ReLU (blau), Leaky ReLU ($\alpha=0.1$) (gelb), SiLU (grün) und GELU (rot) im Bereich $[-2, 2]$.

2.3. Potentiale von Deep Learning in der EEG-Quelllokalisierung

Im Gegensatz zu klassischen Methoden, bei denen Merkmale und Vorwissen explizit eingebracht werden müssen, ermöglicht Deep Learning eine vollständig automatisierte Vorhersage direkt aus minimal vorverarbeiteten Daten. Es eröffnet neue Möglichkeiten, komplexe Zusammenhänge in den Daten zu erkennen und robust auf Rauschen und Störquellen zu reagieren.

Ein zentrales Merkmal von Deep-Learning ist das end-to-end-Learning. neuronale Netze lernen rein aus den Inputdaten alle notwendigen Repräsentationen und Transformationen, um das Target zu berechnen. In diesem Anwendungsfall lernt es allein aus den simulierten Elektroden Daten und der ground truth (Aktivität der Quellen) ohne weitere Angaben die zugrundeliegenden Quellen zu lokalisieren. So kann das Netz sowohl die Effekte der Vorwärtsmodellierung als auch der Artefakte und Rauscheigenschaften direkt lernen, solange diese Effekte ausreichend in den Trainingsdaten repräsentiert sind.

Tiefe neuronale Netze neigen ohne geeignete Regularisierung leicht zum Overfitting. Die eingesetzten Techniken Batch-Normalisierung oder Dropout wirken regularisierend, indem sie die Verteilung der Aktivierungen stabilisieren und zufälliges Maskieren einzelner Neuronen erzwingen.

Neben den internen Regularisierungen kann man auch in den Trainingsdaten selbst realistische Rauschmuster einbringen. Indem man die simulierten EEG-Daten mit synthetischem Rauschen versieht, lernt das Netz, zwischen Signal und Rauschkomponenten zu unterscheiden. Dies führt zu einer besseren Robustheit gegen Rauschquellen in echten Datensätzen.

Studien zeigen, dass klassische Lokalisationsverfahren bei idealen, rauschfreien Simulationen sehr gut abschneiden, unter schwierigeren Signal-to-noise-ratios (SNR) aber deutlich an Genauigkeit verlieren. Die Performance von neuronalen Netzen sinkt dabei deutlich langsamer und bleibt auch bei niedrigen SNRs noch stabil [6], [48], [48].

Die Fähigkeit zum end-to-end-Learning von neuronalen Netzen lässt sich nutzen, um den Simulationsdatensatz beliebig zu erweitern. So lassen sich theoretisch neben synthetischem Rauschen auch beliebig viele, verteilte oder interagierende Quellen, verschiedene Leitfähigkeiten (etwa von pathologischem Gewebe), Hintergrundaktivität im Gehirn oder andere Effekte einbinden. Durch das Hinzufügen spezieller Effekte können Netze möglicherweise auch Konstellationen erkennen, bei denen klassische Methoden aktuell noch an ihre Grenzen kommen.

Ein weiterer theoretischer Vorteil vom Einsatz neuronaler Netze ist die Rechenanforderung. Das Erstellen genügend großer Datensätze und das Trainieren und Optimieren der Netzparameter ist sehr rechenaufwändig. Ist ein Modell jedoch fertig trainiert, so sind die Kosten und Dauer einer Inferenz sehr gering und bleiben bei einem fertig trainierten Modell auch konstant, unabhängig etwa von der genauen Anzahl der Quellen in einem Sample. BEM- oder FEM- basierte Methoden benötigen dagegen für jede neue Messung aufwändige Berechnungen, die mit der Anzahl der möglichen Quellen schnell ansteigen.

Diese theoretischen Potentiale bilden die Basis für den Einsatz von Deep-Learning-Methoden in der EEG-Quelllokalisierung, wie sie in dieser Arbeit weiterverfolgt werden.

3. Stand der Forschung und Zielsetzung

3.1. Stand der Forschung bei EEG-Quelllokalisierung mit Deep-Learning

Bereits sehr früh gab es schon verschiedene erste Ansätze, künstliche neuronale Netze bei der Lokalisierung von EEG- oder MEG-Quellen zu verwenden. Durch den damaligen Stand sind die Ergebnisse heute nur begrenzt anwendbar (vereinfachtes Kopfmodell, kleine Datenmengen, einfache neuronale Netze) [49], [50], [51], [13].

Die meisten aktuellen Ansätze verwenden CNN-Modelle, die eine 2D Topographie Karte des EEG Signals als Input nehmen [3], [52], [53]. In diesen Arbeiten ist der Quellraum der möglichen Quellen jedoch beschränkt, es wurden nur Durchschnittsmodelle als Grundlage für die BEM oder FEM vorwärts Berechnung gewählt. Die BEM/FEM Modelle sind nicht ganz auf dem aktuellen Stand. Eine aktuelle Arbeit mit genauerem Kopfmodell und aktueller Architektur ist "M/EEG source localization for both subcortical and cortical sources using a convolutional neural network with a realistic head conductivity model" [54].

Einige Arbeiten verwenden auch fully-connected-Netze entweder vollständig oder in Kombination mit anderen Architekturen, um beispielsweise besonders schnelle Inferenzzeiten zu erreichen [55]. Doch auch diese Arbeiten behalten Limitierungen, wie das Einschränken der möglichen Dipolrichtungen [56].

Die meisten Arbeiten validieren ihre Ergebnisse mit realen Messungen von Probanden. Dabei werden meist ein [3], [53], [56] bis drei Probanden verwendet [13], [54].

Bisher wurden alle Deep Learning Modelle auf möglichst gute Performance auf simulierten Daten hintrainiert. Noch keine Forschungsarbeit hat den MEG-Dipole Scan als ground truth für Evaluation oder Training angewendet.

3.2. Zielsetzung der Arbeit

Ziel der Arbeit ist es, diese Forschungslücke zu schließen und ein Deep Learning Modell zu trainieren, das auf mehr den Messungen von als 3 Probanden evaluiert wird. Als Evaluations- und Optimierungsmethode soll der MEG-Dipole Scan genutzt werden. Die Dauer für die Berechnung der Quellverteilung eines Samples soll im Bereich der Echtzeit-Anwendung liegen. Es soll für jeden Quellpunkt die Aktivität vorhergesagt werden und nicht nur die Koordinaten des Maximus. Es soll jede mögliche Dipolrichtung als Input nehmen können.

4. Methoden

Im Folgenden werden die verwendeten Methoden der realen und simulierten Daten, sowie des Trainings der Modelle behandelt.

4.1. Datenquellen und Aufbereitung realer EEG-Daten

4.1.1. Beschreibung der verfügbaren Daten

Die Forschungsgruppe am Institut für Biomagnetismus und Biosignalanalyse unter Leitung von Prof. Dr. Carsten Wolters hat einen anonymisierten Datensatz von 19 Probanden mit individuell an MRT-Aufnahmen angepassten 3-Compartment-FEM-Kopfmodellen und simultanen EEG- und MEG-Aufnahmen bereitgestellt.

Für die Datengenerierung wurde eine elektrische Stimulation des Medianervs durchgeführt (P20/N20). Die Stimuli bestanden aus monophasischen Rechteckimpulsen mit 0,5 ms Dauer in zufälligen Intervallen zwischen 350 und 450 ms. Die Stimulus Stärke wurde so eingestellt, dass sich der Daumen deutlich bewegte. Insgesamt wurden 2200 Stimuli gemittelt, um das signal-to-noise-ratio zu verbessern. Siehe auch [57]. Die gemittelten Signale von Proband 1 sind in *Abbildung 7* dargestellt.

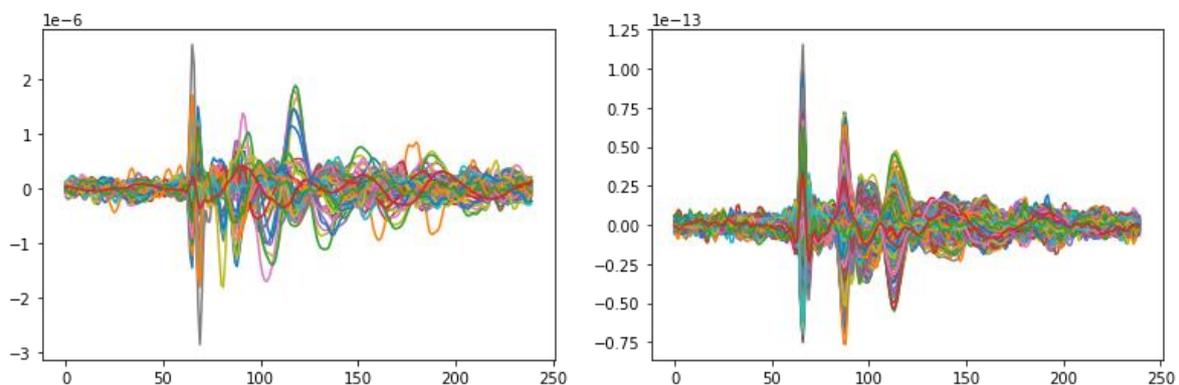


Abbildung 7: Gemittelte EEG- (links) und MEG-Daten (rechts) des ersten Probanden mit der Amplitude y und der Zeit in ms auf der x-Achse.

Nun wird das Rauschlevel der realen Daten betrachtet.

Tabelle 1: Durchschnittliche Rauschlevel der realen Daten über alle Probanden gemittelt.

Durchschnittliches Rauschen vor Onset	Durchschnittliches Rauschen nach Onset
2.48 db	23.48 db

Da in den Simulationen eine konkrete neuronale Aktivität simuliert werden soll, wird sich für die Simulationen am durchschnittlichen Rauschen nach Onset des P20/N20 orientiert. Das Signal ist nach Onset deutlich besser, da höhere Amplituden erzeugt werden und diese im Vergleich zum bestehenden Rauschen mehr des Signals ausmachen (siehe *Tabelle 1*). Das Rauschlevel der Simulation kann über einen Parameter genau eingestellt werden und wird deshalb nicht weiter betrachtet.

4.1.2. Vorverarbeitung der realen Daten

Bei den Datensätzen der 19 Probanden wurden „bad channels“, also Kanäle mit zu viel Rauschen oder anderen Problemen entfernt. So haben die Messungen alle unterschiedlich viele Datenpunkte abhängig von der Anzahl der verbliebenen Sensoren (Elektroden, SQUID-Sensoren). Da die Modelle jedoch eine feste Anzahl an Eingaben voraussetzen, müssen die Daten alle auf dieselbe Anzahl an Sensoren gebracht werden. Da final eine Elektrodenlänge von 54 angewendet wird, wurden von den 19 Probanden nur die 7 ausgewählt, die 54 oder mehr Messpunkte haben, um nur herunterskalieren und nicht interpolieren zu müssen. So bleiben die Daten in konstanter Qualität und werden kaum verfälscht.

Zunächst wird von Python MNE die Montage, die bei der Datengenerierung verwendet wurde (bei diesem Datensatz „easycap-M1“) geladen. Diese definiert für jeden Kanalnamen eine 3D-Koordinate auf der Kopfoberfläche. Die vorhandenen Kanäle haben bereits individuell vermessene Koordinaten, welche in das Standardmodell übernommen werden. Für alle fehlenden Kanäle werden die Template-Positionen übernommen. So erhält man eine vollständige Liste von Kanalnamen und den jeweiligen Koordinaten für jeden Probanden.

Die vorliegenden EEG-Zeitreihen werden dann in ein Array der neuen Größe überführt, wobei dann fehlende Werte für die neuen Kanäle entstehen. Aus den kombinierten Kanalpositionen und den vorhandenen Potentialverteilungen wird ein MNE Montage-Objekt erstellt. Mithilfe der Lösung der Laplace-Gleichung auf der Sphäre (Kopf) werden die fehlenden Potentiale so berechnet, dass sie an den bekannten Elektroden genau mit den gemessenen Werten übereinstimmen und über die gesamte Kopfoberfläche möglichst glatt verlaufen [58].

Aus den 74 interpolierten Datenpunkten werden dann die in den Daten am häufigsten auftretenden 54 Stück ausgewählt, um die Daten möglichst original zu halten. Die final genutzten Kanäle nach der Interpolation sind: Fp1, Fp2, T8, Fz, Cz, Oz, FC1, CP1, F1, F2, C1, C2, P1, P2, AF3, AF4, FC3, FC4, CP3, PO3, PO4, F5, F6, C5, C6, AF7, AF8, FT7, TP7, PO8, Fpz, AFz, CPz, POz, F9, FT9, F10, FCz, FC2, P5, FT8, Iz, T7, CP2, TP10, TP8, FT10, TP9, CP4, PO7, PO9, PO10, P6 und P10. Die Positionen dieser Elektroden bei Proband 1 sind in *Abbildung 8* dargestellt.

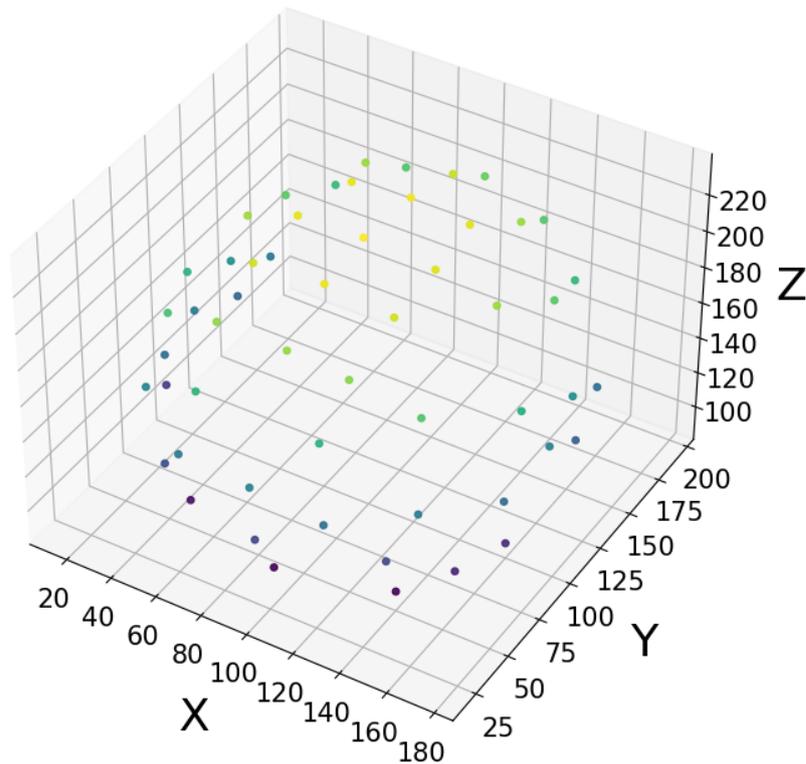


Abbildung 8: 3D-Positionen der 54 Elektroden für Proband 1.

4.1.3. Herausforderungen bei der Datenvalidierung

Bei realen gemessenen Daten von Probanden gibt es das Problem der fehlenden ground truth. Man kann zwar wertvolle Messungen der Signale an der Kopfoberfläche durchführen, es ist jedoch in den allermeisten Fällen nicht möglich die genaue tatsächliche Quelle dieser Signale herauszufinden. Besondere Datensätze, die eine genaue Lokalisierung ermöglichen, wie operativ entferntes Gewebe bei Epilepsie, sind zu klein, um sie für Deep Learning Methoden zu verwenden. Deshalb wird, wie in allen Deep Learning Quelllokalisierungen eine Simulation zum Trainieren der Modelle verwendet.

4.1.4. Validierung anhand des MEG-Dipole Scans

Eine mögliche Herangehensweise ist die Verwendung des MEG-Dipole Scan. In vielen Fällen ist die Quelllokalisierung von MEG-Daten genauer und robuster als die von EEG-Methoden.

Dadurch, dass im vorliegenden Datensatz EEG und MEG gleichzeitig aufgenommen wurden (EMEG), ist die zugrundeliegende neuronale Aktivität sicher gleich. Trainiert man nun ein neuronales Netz für die Lokalisierung anhand von EEG-Daten, kann man diese auf den Messdaten von realen Messungen validieren. Da der Dipole Scan der MEG-Daten die tatsächliche Quelle hier besser abbildet, ist es möglich, diesen auch als ground truth für die gemessenen EEG-Daten zu verwenden [59], [60].

Gerade für die im vorhandenen Datensatz gemessenen P20 Signale, ist eine bessere Lokalisierung durch MEG-Daten bereits bekannt [61], [62], [63]. *Abbildung 9* zeigt das Ergebnis eines Dipol Scan

4. Methoden

auf den EEG- und MEG-Daten. Die Verteilung der Ergebnisse der MEG-Daten ist dabei deutlich genauer und weniger ausgebreitet.

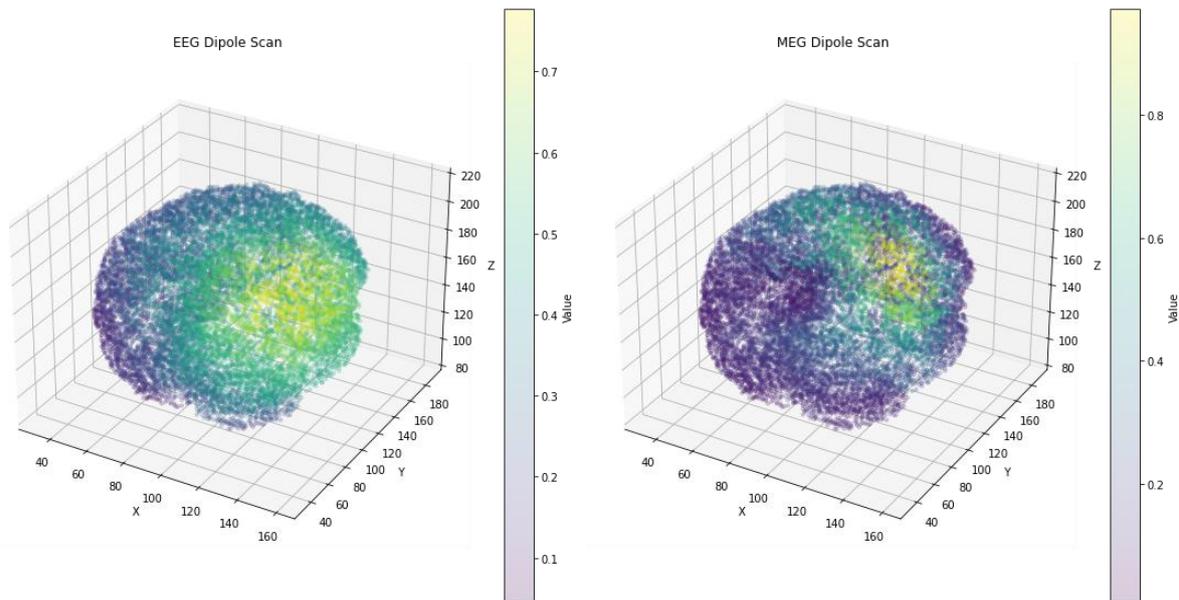


Abbildung 9: EEG- (links) und MEG-Dipole Scan (rechts) des SEP von Proband 1. Der MEG-Dipole Scan zeigt dabei einen deutlich kleineren Bereich und Fehler bei der optimalen Lösung.

Diese Methode der Validierung wurde in der Literatur der EEG-Deep Learning Quelllokalisierung nach ausführlicher Suche noch nicht angewendet [64].

Dabei bietet diese Methode viele mögliche Vorteile. Einerseits ist die Zugänglichkeit sehr gut. Nach Berechnen der MEG-Dipole Scan Positionen kann der Datensatz mit ground truth einfach von anderen Forschungsgruppen verwendet werden, die dann die Möglichkeit haben, ihre Modelle auf den gleichen Datensätzen zu testen, was auch die Vergleichbarkeit der verschiedenen Modelle erhöht. Zudem ist das Durchführen des Dipole Scan nicht aufwändig, was viele andere Forscher motivieren könnte, ihre Daten auch zur Verfügung zu stellen. Dies könnte mittelfristig zu einem großen Benchmark Datensatz führen, der bei ausreichender Größe vielleicht in der Zukunft auch direkt fürs Training verwendet werden könnte. Außerdem erlaubt es eine quantitative Bewertung der Modelle während dem Training, was das sinnvolle Einsetzen von Hyperparameteroptimierung für neuronale Netze ermöglicht. Auf den Simulationen Hyperparameteroptimierung durchzuführen ist nicht zwingend sinnvoll, da die Performance auf simulierten und realen Daten weit auseinander gehen kann. Die Optimierung auf den realen Daten ist auf jeden Fall sinnvoll. Diese Vorteile sollen in dieser Arbeit getestet werden.

Allerdings muss auch beachtet werden, dass der MEG-Dipole Scan ebenfalls eine Lokalisierungsmethode ist und dadurch immer fehlerbehaftet ist und nicht 100% der Quellen abdeckt. Zwar kann man die Abdeckung beim Dipole Scan auslesen, sie dadurch aber nicht erhöhen. Zudem gilt die bessere Lokalisierung mit MEG-Daten nicht für alle möglichen Quellen, sondern muss vorher ausführlich evaluiert werden.

4.2. Simulation eines geeigneten Datensatzes

Für das Training von Deep-Learning Methoden bei der EEG-Quelllokalisierung, wird ein entsprechender Datensatz benötigt. Soll ein Netz aus den originalen Messdaten direkt die aktiven Quellen vorhersagen, so muss der Trainings-Datensatz einerseits aus dem Input \mathbf{x} mit allen Messungen an der Kopfoberfläche und andererseits aus dem Target \mathbf{y} bestehen, welches für jeden möglichen Dipol im Quellraum eine Aktivität angibt.

Einen solchen Datensatz in ausreichender Größe, Präzision und Varianz mit realen Messungen zu erstellen ist aktuell unmöglich. Man müsste eine sehr große Anzahl an Probanden eine große Anzahl an Experimenten machen lassen. Zusätzlich bietet eine eigene Simulation viele flexible Parametrisierungsmöglichkeiten. Aus diesen Gründen wird eine selbst erstellte Simulation verwendet.

Für die Datensimulation wird die Leadfield-Matrix eines Probanden aus Kapitel 4.1.1. *Beschreibung der verfügbaren Daten* verwendet. Für das Erstellen der Leadfield-Matrix wurde DUNEuro verwendet.

Zuerst wird ein Punkt zufällig aus dem Quellraum ausgewählt und diesem ein zufälliger Dipolvektor zugewiesen. Der Quellraum besteht aus 14909 möglichen Quellpunkten. Die der ausgewählten Quelle entsprechende Zeile der Leadfield-Matrix wird daraufhin mit den Dipolstärken in X-, Y- und Z-Richtung multipliziert. Daraus resultieren die Signalwerte an der Kopfoberfläche. In dieser Simulation werden 54 Kopf-Elektroden verwendet.

Dies entspricht der Vorwärts-Berechnung (17) in 2.1.1.2. *Mathematische Grundlagen*:

$$\mathbf{M} = \mathbf{L}\mathbf{D}$$

(17)

4.2.1. Hinzufügen von Rauschen

Um die Simulation realistischer zu machen, wird Rauschen zu den simulierten EEG-Signalen hinzugefügt. Bei realen Aufnahmen gibt es viele mögliche Rauschquellen wie die Augenbewegungen, Muskelaktivität oder externe Störungen (siehe: 2.1.1.3. *Signalaufzeichnung*). Zusätzlich tritt auch immer instrumentelles Rauschen und das konstant vorhandene Hintergrundrauschen durch Aktivität des menschlichen Gehirns auf [2], [28].

Außerdem führt Rauschen in den Trainingsdaten zu mehr Varianz und erhöht so auch allgemein die Generalisierungsfähigkeiten des neuronalen Netzes.

Das Rauschen wird berechnet, indem die durchschnittliche Leistung des Signals bestimmt und dann basierend auf einer definierten Signal-Rausch-Verhältnis (SNR) ein entsprechendes additives, gaußsches Rauschen generiert wird:

$$P_{signal} = \frac{1}{N} \sum_{i=1}^N x_i^2$$

(34)

Die Rauschleistung P_{noise} ergibt sich aus dem definierten SNR:

$$P_{noise} = \frac{P_{signal}}{10^{SNR/10}} \quad (35)$$

Die additiven Rauschwerte werden dann aus einer Normalverteilung mit Mittelwert 0 und Varianz σ_n^2 gezogen.

$$\sigma_n = \sqrt{P_{noise}} \quad (36)$$

$$n_i \sim N(0, \sigma_n^2) \quad (37)$$

Das verrauschte Signal \mathbf{x}_{noisy} wird berechnet durch Addition des Rauschens \mathbf{n} zu den ursprünglichen sauberen Signalwerten \mathbf{x} :

$$\mathbf{x}_{noisy} = \mathbf{x} + \mathbf{n} \quad (38)$$

Dies entspricht dann Formel (19) in Kapitel 2.1.2.2. *Mathematische Darstellung des inversen Problems.*

Mit dieser Methode können dann Datensätze mit verschiedensten SNRs generiert werden.

4.2.2. Skalierung der Daten

Beim Anwenden des neuronalen Netzes auf reale Daten ist es wichtig, dass die Skalierung der Daten ähnlich ist, um eine gute Performance zu ermöglichen. Die realen Daten haben über alle Probanden gemittelt folgende Größen:

Tabelle 2: Durchschnittliche Werte der realen Messungen über alle Probanden.

Mittelwert μ_{real}	$5.74 \cdot 10^{-25}$
Standardabweichung σ_{real}	$3.16 \cdot 10^{-7}$
95% Perzentil	$4.17 \cdot 10^{-6}$

Die Standardabweichung ist damit etwa $3.0 \cdot 10^{-7}$ und der Mittelwert ≈ 0.0 . Um die simulierten Daten möglichst realistisch zu halten, werden diese nach der Simulation einmal global standardisiert:

$$\mu_{sim} = \frac{1}{N} \sum_{i=1}^N x_i, \quad \bar{\mathbf{x}} = \mathbf{x} - \mu_{sim} \quad (39)$$

$$\sigma_{sim} = \sqrt{\frac{1}{N} \sum_{i=1}^N \bar{x}_i^2}, \quad \hat{\mathbf{x}} = \bar{\mathbf{x}} \frac{\sigma_{real}}{\sigma_{sim}} \quad (40)$$

Dadurch werden die Werte der simulierten Daten grob an die Skalierung der realen Daten angepasst und haben auch einen Mittelwert von 0 und eine Standardabweichung von $\approx 3.0 \cdot 10^{-7}$. Allgemein darf die exakte Standardabweichung aufgrund von „data leakage“ nicht verwendet werden, wobei ein ähnlicher Wert in der gleichen Größenordnung, wie hier angewendet, in Ordnung ist. Data leakage tritt auf, wenn während dem Training Informationen in das Netz einfließen, die in der realen Anwendung nicht zur Verfügung stehen dürfen. Da Mittelwert und Standardabweichung bei ähnlichen Daten mit dem gleichen Messverfahren immer ähnlich sind, tritt noch kein data leakage auf.

Vor dem Anwenden der simulierten Daten als Trainingsdaten für ein neuronales Netz, werden die simulierten Daten dann nochmal global angepasst mit:

$$\mathbf{x}_{train} = \frac{\hat{\mathbf{x}} - \mu_{real}}{\sigma_{real}} \quad (41)$$

Dies führt dazu, dass die Werte für das Training eine Varianz von ≈ 1 haben, was zu stabileren Gradienten und einem konstanteren Training führt. Beim Anwenden eines neuronalen Netzes auf reale Daten muss (41) dann auch angewendet werden.

4.2.3. Gaußsche Propagation von Aktivität

In realen Umgebungen gibt es fast nie den Fall, dass genau eine einzelne Punktquelle aktiv ist. Tatsächlich breitet sich die Aktivität meist über einen größeren Bereich aus. Die genaue Ausbreitungsform ist dabei sehr unterschiedlich je nach Gehirnregion, Tiefe der Quelle, Stimuli und vielen weiteren Einflussfaktoren [65], [66], [67]. Um trotz der immensen Komplexität der realen Ausbreitung eine räumliche Ausbreitung in der Simulation zu erzeugen, wird, wie auch in ähnlichen Arbeiten, eine gaußsche Propagation angenommen [3], [13], [68].

Eine Gaußsche Verteilung ermöglicht eine realistische Modellierung der räumlichen Ausbreitung der Aktivität und vermeidet unnatürliche Übergänge, die bei Punktquellen sonst auftreten könnten.

Beim Erstellen der simulierten Datensätze wird dann für alle Punkte innerhalb des Radius eine Aktivität simuliert. Diese ist abhängig von der Stärke der originalen Quellen und der Distanz eines Punktes zur Quelle, beziehungsweise der Wert der Gaußsche Verteilung an der Distanz. Der Radius wird so berechnet, dass er 90% der Gaußfunktion abdeckt. Die Verteilung wird durch einen wählbaren Sigma-Wert (Standardabweichung) verändert. Dadurch ist die summierte neuronale Aktivität eines Datensamples nur abhängig von der Stärke der originalen Quellen und der Breite der Gauß Glocke. Dies führt im Gegensatz zu normierten Verteilungen zu mehr Variabilität im Datensatz, was die Generalisierungsfähigkeit des Modells unterstützt. Eine mögliche Verteilung ist in *Abbildung 10* gezeigt.

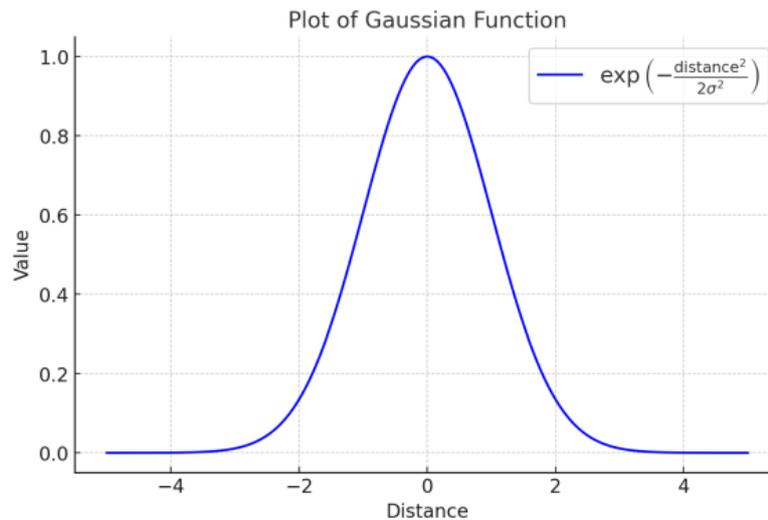


Abbildung 10: Exemplarische Darstellung der Gauß Verteilung. Der Sigmawert ist veränderbar und kann zu Beginn angegeben werden.

Durch die Ausbreitung der Aktivität gibt es für die Fehlerfunktion des neuronalen Netzes deutlich konstantere Werte, da die Distanz der geschätzten Verteilung zur tatsächlichen Verteilung gut messbar gemacht werden kann, wobei bei Punktquellen normale Fehlerfunktionen nicht wirklich funktionieren würden und man für die Fehlerfunktion die Distanz zwischen Vorhersage und tatsächlichem Wert separat berechnen müsste.

Eine alternative Methode der Propagation ist die $1/r^2$ Abnahme. In einem idealen homogenen Volumenleiter folgt das Potential eines punktförmigen Dipols genau der $1/r^2$ Abnahme. Durch verschiedene Gewebeschichten und Anisotropie folgt die tatsächliche Ausdehnung in realem Gewebe allerdings eher einer $1/r^3$ Abnahme. Beide Funktionen sind beispielhaft in *Abbildung 11* dargestellt. Bei ersten Tests führte diese Propagationmethode allerdings zu deutlich schlechterer Performance auf den realen Daten, weshalb für den weiteren Verlauf die Gauß-Propagation gewählt wird.

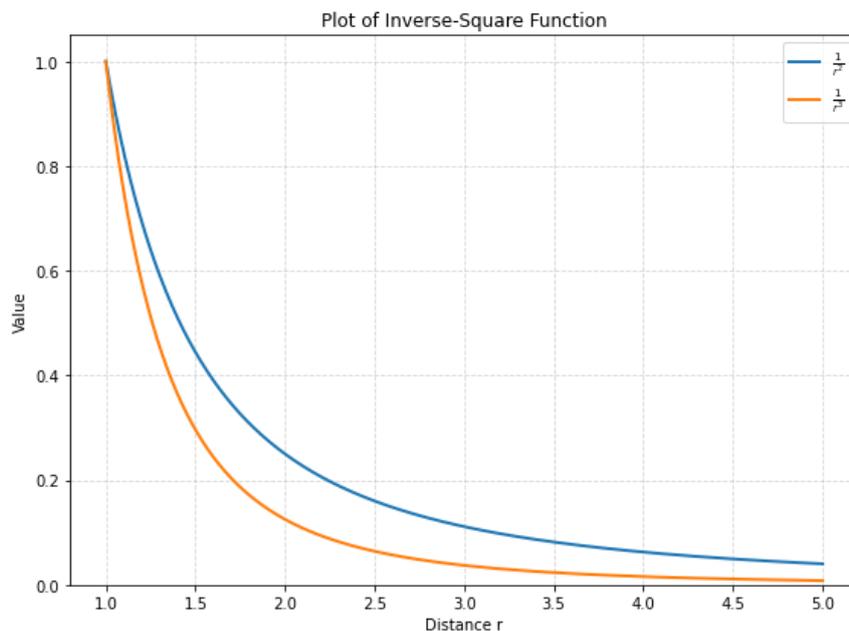


Abbildung 11: Beispielhafte Darstellung eines $1/r^2$ (blau) und eines $1/r^3$ (orange) Verlaufs.

4.2.4. Variationen von Gaußglocken und verschiedene Verteilungen

Die in der Simulation verwendeten Gaußglocken für die gaußsche Propagation sind abhängig vom gewählten Wert Sigma. Um mehr Varianz in den Datensatz zu bringen und um die Wahrscheinlichkeit zu erhöhen, dass eine ähnliche Verteilung in den realen Daten und den simulierten Daten existiert, wird für jedes Sample zufällig ein Sigma aus einem vorgegebenen Intervall ausgewählt. Dieses Sigma Intervall orientiert sich dabei grob an realistischen Ausbreitungen.

4.2.5. Quelltiefe

Mithilfe einer konvexen Hülle über alle verfügbaren Quellpunkte im Quellraum kann für jeden möglichen Quellpunkt eine Tiefe bestimmt werden. Die konvexe Hülle von Proband 1 wird in *Abbildung 12* gezeigt.

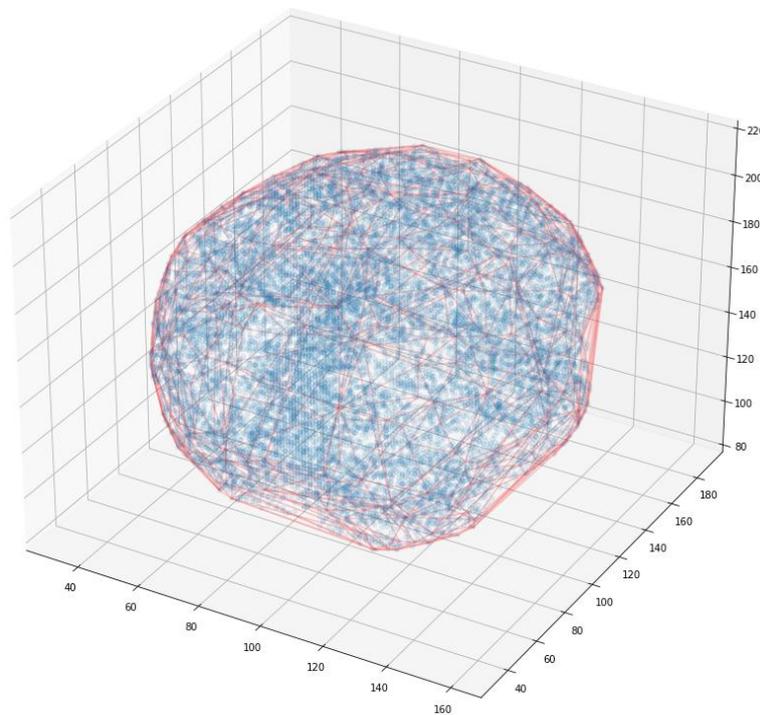


Abbildung 12: Konvexe Hülle von Proband 1 zur Tiefenbestimmung von simulierten Quellen.

Eine Übersicht über die Quelltiefenverteilung von Proband 1 ist in *Tabelle 3* aufgelistet.

Tabelle 3: Quelltiefenverteilung des Quellraums von Proband 1.

Metrik	Wert
Durchschnitt	1.58 cm
Standardabweichung	1.16 cm
Min	0.0 cm
25%	0.72 cm
50%	1.26 cm
75%	2.19 cm
Max	6.01 cm

Daraus lässt sich erkennen, dass sich die meisten möglichen Quellpunkte relativ nah an der Kortexoberfläche befinden. Wobei hier die konvexe Hülle hier auch unter dem Kortex verläuft. Das liegt daran, dass die Quellpunkte nur in Geweben liegen, die auch Aktionspotentiale erzeugen können. Dies ist hauptsächlich die graue Substanz, welche an der Oberfläche des Kortex liegt und etwa 2-4 mm tief ist. Durch die Faltung der Hirnrinde befindet sich die graue Substanz aber eben auch teilweise so bis einigen Zentimetern im Gehirn. Im Hirninneren besteht viel Gewebe aus weißer Substanz oder Liquor, welche beide keine Aktionspotenziale erzeugen können.

4.3. Entwicklung und Training der Deep Learning Modelle

4.3.1. Wahl der Grundarchitektur

Bei der Wahl der Grundarchitektur wurde ein Fully-Connected-Feed-Forward Netz gewählt. Einerseits koppelt ein FC-Layer jeden Sensor mit jeder Quellkomponente. Diese volle Kopplung ist auch bei der Leadfield-Matrix gegeben und kann aufgrund des beschränkten rezeptiven Feldes von Convolutional-Layern bei CNNs nur indirekt über die Kombination mehrerer Layer erreicht werden. Ein weiterer Vorteil gegenüber CNNs ist, dass keine Annahme über die Nachbarschaft getroffen werden muss. Während 2D-CNNs Nachbarschaften auf einer Fläche betrachten, sind die Elektroden unregelmäßig auf einer gekrümmten Fläche angeordnet. Zudem muss für die Anwendung von CNNs aus den originalen Daten zuerst eine Topologie-Karte berechnet werden, während FCNN direkt die Elektrodenwerte entgegennehmen können. Außerdem ist ein FCNN leicht erweiterbar durch weitere Funktionalitäten wie Skip-Connections oder individuelle Loss-Funktionen. Natürlich haben andere Architekturen auch individuelle Vorteile, aufgrund der genannten Punkte wird in dieser Arbeit jedoch eine FCNN-Architektur verwendet.

Zur Implementierung wurde Python 3.8.3 und das Python Deep Learning Framework Pytorch verwendet [69], [70].

4.3.2. Netzwerkverbesserungen

Wie in Kapitel 4.3.1. *Wahl der Grundarchitektur* beschrieben, ist ein neuronales Netz leicht durch weitere Funktionen und Algorithmen erweiterbar.

Eine Methode, die die Leistung deutlich erhöht hat, ist das Nutzen von den in Gleichung (28) beschriebenen Skip-Connections. Hier wird die Information mithilfe einer Identitätsgleichung zusätzlich zu den normalen Layern weitergegeben. Dies führt zu stabileren Gradienten und besserer Performance.

Ebenfalls angewendet wird das Dropout, was beim Training des neuronalen Netzes zufällig einzelne Neuronen deaktiviert, um die Informationsverarbeitung breit über alle Neuronen eines Layers zu verteilen. Dies erhöht prinzipiell die Generalisierungsfähigkeiten des Modells.

Mithilfe von Batch-Normalisierung werden die Inputs eines Layers über einen Mini-batch standardisiert, was wiederum zu stabileren Gradienten und konstanterem Training führt.

4.3.3. Trainingsmethodik

Die generierten Daten werden in Trainings-, Validierungs- und Testdaten aufgeteilt. Dabei wird eine Aufteilung von 80% Trainingsdaten, 10% Validierungsdaten und 10% Testdaten angewandt. Diese TensorDatasets werden dann beim Training der neuronalen Netze in DataLoader gepackt. Hier muss die Wahl der Batchsizes getroffen werden. So gehört die Wahl der Trainings-Batchsize zu den zu optimierenden Hyperparametern. Die Wahl der Validierungs-Batchsize beeinflusst die Modellqualität nicht und wird üblicherweise auf ein Vielfaches der Train-Batchsize gesetzt. Die Batchsize beschreibt die Anzahl an Datensamples, die zusammen betrachtet und durch das Netz propagiert werden, bevor ein Update der Gewichte stattfindet. Bei kleineren Batchsizes werden so mehr spezifische Updates gemacht, bei größeren Batchsizes tendenziell weniger Updates, die über mehr Datenpunkte gemittelt wurden und so generell einen konstanteren Trainingsverlauf erzeugen.

Als Kosten- oder Loss Funktionen sind Mean Absolute Error (MAE) (siehe Formel (24)), Mean Squared Error (MSE) (siehe Formel (25)), L1 Loss und Huber Loss verfügbar.

Nach der Berechnung des Loss von einem Batch gibt es verschiedene Strategien, die Gewichte im neuronalen Netz aufgrund des Losses anzupassen. Implementierte Optimizer sind ADAM, Stochastic Gradient Descend (SGD) und Adagrad.

Aktuell stehen die Initializer xavier_normal und kaiming_normal zur Verfügung.

Beim Training müssen zudem noch die Parameter der Learningrate und der Anzahl der Epochen betrachtet werden. Die Learningrate ist ein Parameter des Optimizers und skaliert die berechneten Updates der Optimierungsfunktion. So soll verhindert werden, dass einzelne Batches die Gewichte des Netzes komplett verändern. Die Anzahl der Epochen beschreibt die Nummer der Durchläufe des gesamten Datensatzes. Innerhalb einer Epoche werden alle Batches einmal in zufälliger Reihenfolge durch das Netz geleitet. Die Anzahl der Epochen ist abhängig von der Größe des Datensatzes, der verfügbaren Zeit und der Learningrate.

4.3.4. Implementierung technischer Optimierungen

Um die hohen Trainingszeiten zu optimieren, wurde das Training auf einer GPU durchgeführt. Für das Training wurde ein spezifischer DeviceDataLoader geschrieben, der gesamte DataLoader auf die GPU laden kann, statt wie üblich jeden Batch einzeln auf die GPU zu verschieben. So entsteht weniger Wartezeit auf der GPU und das Training kann beschleunigt werden. Beim Erstellen des DataLoader werden mehrere Worker verwendet. Dies bedeutet, dass eine entsprechende Anzahl an Subprozessen gestartet werden, die die Daten parallel laden. So können etwa folgende Daten vorbereitet werden, während die GPU die aktuellen Batches trainiert.

Zusätzlich wird Mixed-Precision Training verwendet, was die Effizienz des Trainings von Deep Learning-Modellen verbessert, indem sowohl 16-Bit- (Half Precision) als auch 32-Bit-Gleitkommazahlen (Single Precision) genutzt werden. Hier wurde PyTorch „Automatic Mixed Precision“ gewählt [71].

4.3.5. Hyperparametertuning

Hyperparameteroptimierung ist eine Methode, um die optimale Parameterkombination für ein Optimierungsproblem zu finden. Im Falle von neuronalen Netzen wird diese Methode häufig

genutzt, um die wählbaren Hyperparameter für das Netz auf bestmögliche Performance zu optimieren.

In dieser Arbeit gibt es folgende, im Trainingsprozess relevanten Parameter:

Tabelle 4: Alle relevanten Hyperparameter, angeordnet nach Kategorie.

Kategorie	Parameter
Datengenerierung	Anzahl der Samples
	Sigma Intervall der Quellausbreitung
	Signal-to-noise-ratio (SNR)
	Trainings-Batchsize
Neuronales Netz	Anzahl der Layer
	Anzahl der Neuronen pro Layer
	Aktivierungsfunktion
	Dropout-Faktor
	Initialisierungsstrategie
Training	Kostenfunktion
	Optimierungsfunktion
	Anzahl der Epochen
	Learning Rate

Die Anwendung von Hyperparameteroptimierung auf die Aufgabenstellung ist jedoch problematisch. Ziel des Projektes ist eine auf reale Daten anwendbare KI zu trainieren. Um die Hyperparameter zu optimieren, muss es eine Methode geben, um die Performance eines Modells zu quantifizieren. Dafür wird normalerweise der Validierungs Loss eines Modells verwendet. In diesem Fall bedeutet aber eine bessere Leistung auf den simulierten Daten nicht unbedingt eine bessere Leistung auf den realen Daten. Eine Möglichkeit ist die in Kapitel 4.1.4. *Validierung anhand des MEG-Dipole Scans* vorgestellte Methode, die Distanz der neuronalen Netz Lösung zur MEG-Dipole Scan Lösung zu verwenden. Diese Methode wurde in dieser Arbeit angewandt.

Die Hyperparameteroptimierung wurde mithilfe der Python Bibliothek „Weights and Biases“ durchgeführt [72].

4.4. Reproduzierbarkeit der Experimente

Um die Ergebnisse der Arbeit reproduzierbar zu machen, wurde bei jedem Skript ein `random_seed` gesetzt, der dann bei allen zufallsbasierten Methoden (wie etwa die Wahl der aktiven Dipole) immer das gleiche Ergebnis liefert. Im Sinne von Open-Science ist der gesamte Code außerdem unter https://github.com/HS-Ansbach-CCS/ai_eeg_source_localization verfügbar.

5. Ergebnisse und Diskussion

5.1. Ergebnisse der Hyperparameteroptimierung

Zuerst wird eine Hyperparameteroptimierung der Netzparameter durchgeführt. Nach jedem Training wird das neuronale Netz mithilfe einer Evaluationsfunktion bewertet. Diese berechnet

5. Ergebnisse und Diskussion

mit dem aktuellen Modell auf allen 7 Datensätzen den Lokalisationsfehler (LE), wobei als r_{true} das Ergebnis des MEG-Dipol Scans auf dem jeweiligen Datensatz genutzt wird.

$$LE = \|r_{true} - r_{pred}\|_2 \quad (42)$$

Trainiert wird das Netz nur auf simulierten Daten mit der Leadfield-Matrix von Proband 1. Eine Validierung über die Simulationen weiterer Probanden wird dann nach herausfinden der optimalen Parameter durchgeführt.

Nach 10 Trainingsdurchläufen zur Minimierung des LE wurden folgende in *Abbildung 13* dargestellten Parameterkombinationen getestet:

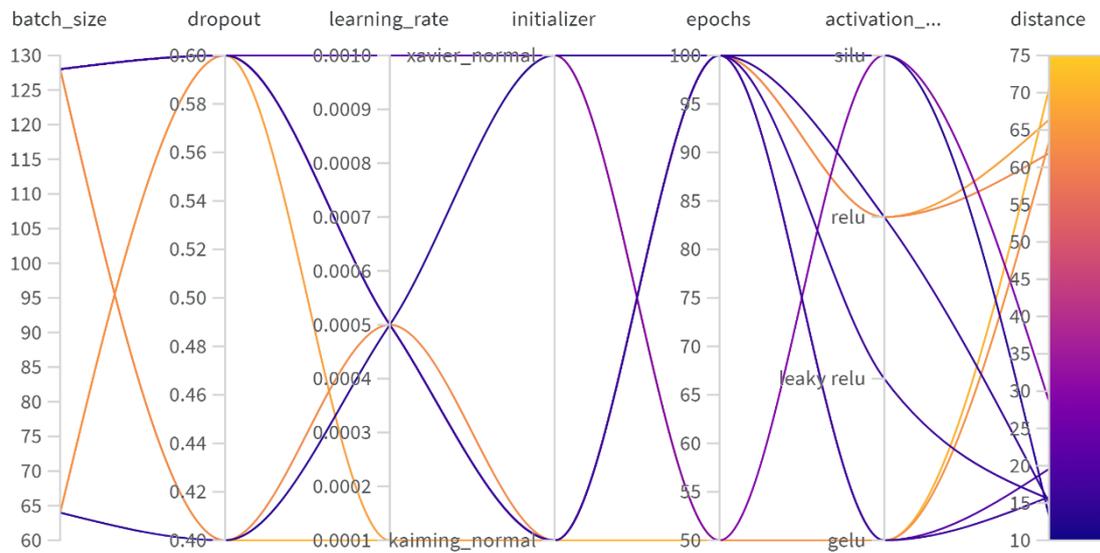


Abbildung 13: Weights and Biases Übersicht der Parameterkombinationen der 10 Trainingsdurchläufe.

Die resultierenden Parameter für das Netz mit dem geringsten durchschnittlichen Abstand zur MEG-Dipol Scan Lösung sind in *Tabelle 5* aufgelistet:

Tabelle 5: Netzparameter des besten Modells nach der Hyperparameteroptimierung. Alle grün hinterlegten Parameter wurden für die Optimierung mit einbezogen, die restlichen Parameter waren für jedes Training gleich oder werden nach dem Training berechnet.

Datensatz		
	Anzahl der Samples	250000
	Sigma Intervall	[1, 20]
	Signal-to-noise-ratio	20 db
	Standardisierung (μ, σ)	$0.3 \cdot 10^{-7}$
	Batchsize Training	64
	Batchsize Validierung	512
	Split	0.8, 0.1, 0.1
	Propagationsmethode	Gaussian
Training		
	Kostenfunktion	MSE

Optimizer	ADAM
Epochen	100
Learningrate	0.0005
Initialisierungsmethode	Xavier normal
Dropoutfaktor	0.4
Aktivierungsfunktion	SiLU
Neuronen pro Layer	54, 64, 1024, 2048, 2048, 4096, 8192, 14909

Die resultierende Parameterkombination wird im folgenden Parameter-MEG-Distanz-optimiert genannt.

Eine interessante Erkenntnis aus den Resultaten der Hyperparameteroptimierung ist der Zusammenhang des Trainings Loss und der Distanz zur MEG-Dipol Scan Lösung wie in *Abbildung 14* gezeigt. Eine bessere Performance auf den Trainingsdaten korreliert leicht mit einer geringeren Distanz. Dies validiert zu einem kleinen Grad die genutzte Datensimulation. Allerdings folgen nicht alle Netze diesem Verlauf und es gibt auch Netze mit guten Ergebnissen auf dem Training und schlechter Leistung auf den MEG-Dipol Scans. Dies erlaubt die Hypothese, dass die Performance auf simulierten Daten nicht konstant die Leistung auf realen Daten vorhersagen kann. Diese Hypothese wird in den folgenden Experimenten weiterverfolgt.

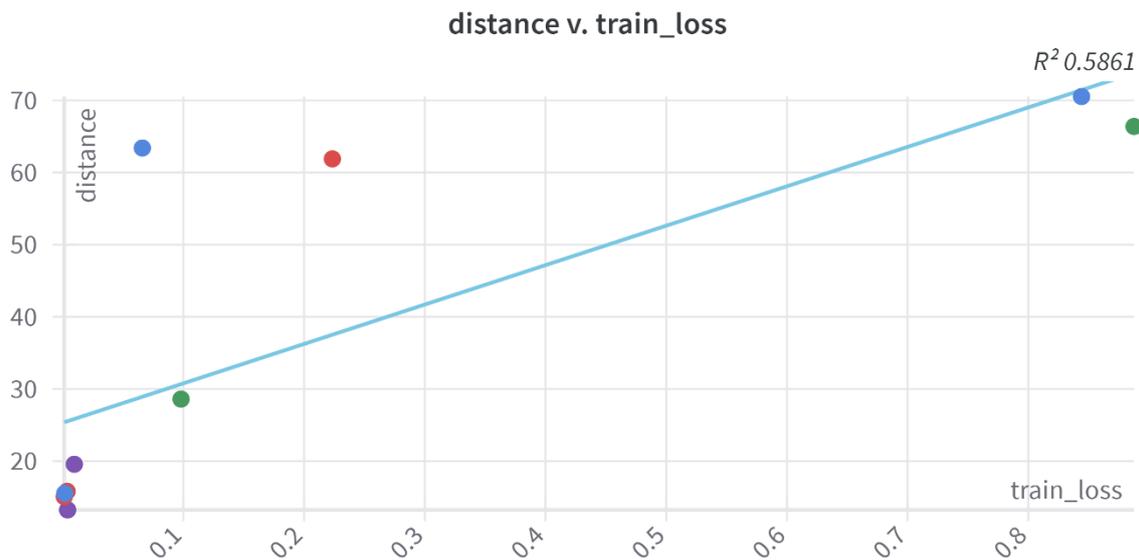


Abbildung 14: Scatter plot des Zusammenhangs zwischen Trainings Loss und Distanz (in mm) mit einer linearen Regression zum Darstellen der Korrelation.

Die finale Netzarchitektur ist in *Abbildung 15* dargestellt:

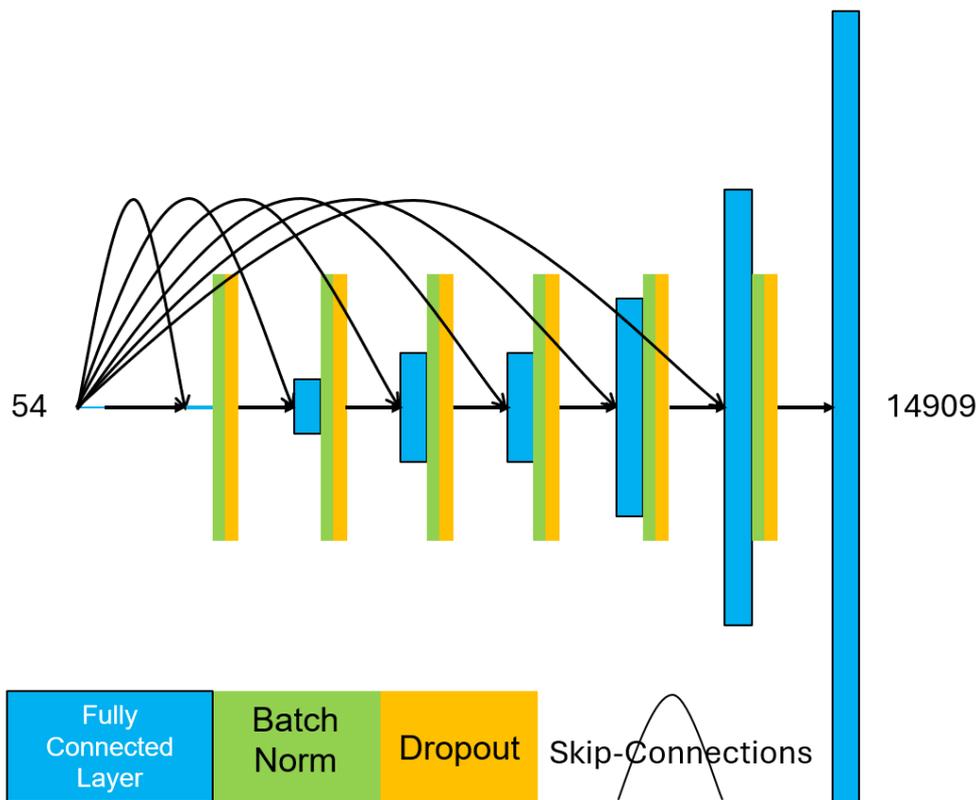


Abbildung 15: Darstellung der Netzarchitektur mit 6 hidden Layern, wobei in jeder hidden Layer Batch-Normalisierung und Dropout enthalten ist. Zwischen den Layern stellen Skip-Connections einen Identitätspfad her. Die Größe der Layer ist proportional zur Anzahl der Neuronen im Layer.

Für eine ausreichende Validierung wurden mit den gefundenen Parametern 3 Netze trainiert. Ein Netz wurde mit simulierten Daten eines Probanden trainiert, ein Netz mit 2 Probanden und eins mit den Daten von 3 Probanden. In den folgenden Kapiteln soll so eine ausreichende Validierung sichergestellt werden und der Einfluss der gewählten Simulation überprüft werden.

5.2. Evaluation auf simulierten Daten

Um die Methode zu validieren, sollte man diese mit einer bekannten Methode mit ähnlichen Voraussetzungen vergleichen. Zur Evaluation wird in dieser Arbeit deshalb der Dipole Scan verwendet.

Voraussetzung für den Dipole Scan ist ein vorhandenes Vorwärtsmodell. Dies ist hier durch die Leadfield-Matrix gegeben. Beim Dipole Scan wird für jede mögliche Quelle iteriert und die Vorwärtslösung berechnet. Die Quelle, deren Vorwärtslösung die gemessenen Daten am besten beschreibt, ist dann die berechnete Lösung.

Der Dipole Scan wurde als Vergleichsmethode gewählt, da es weit verbreitet, einfach zu berechnen ist und das gleiche Vorwissen benötigt wie das neuronale Netz. In beiden Fällen wird das Vorwärtsmodell in Form der Leadfield-Matrix verwendet. Beim neuronalen Netz ist es nicht direkt gegeben, wird aber verwendet, um die Samples der Trainingsdatensätze zu generieren. So lernt das Netz indirekt auch von der Leadfield-Matrix und keine der beiden Methoden hat einen Informationsvorteil. Die berechnete Lösung des neuronalen Netzes ist der Punkt mit der höchsten Vorhergesagten Aktivität.

Zum Testen wird ein Datensatz mit den Parametern von *Tabelle 6* für jeden Probanden erstellt:

Tabelle 6: Parameter für die Datengenerierung der Tests auf simulierten Daten

Anzahl der Samples	100
Sigma Intervall	[1, 20]
Signal-to-noise-ratio	20 db

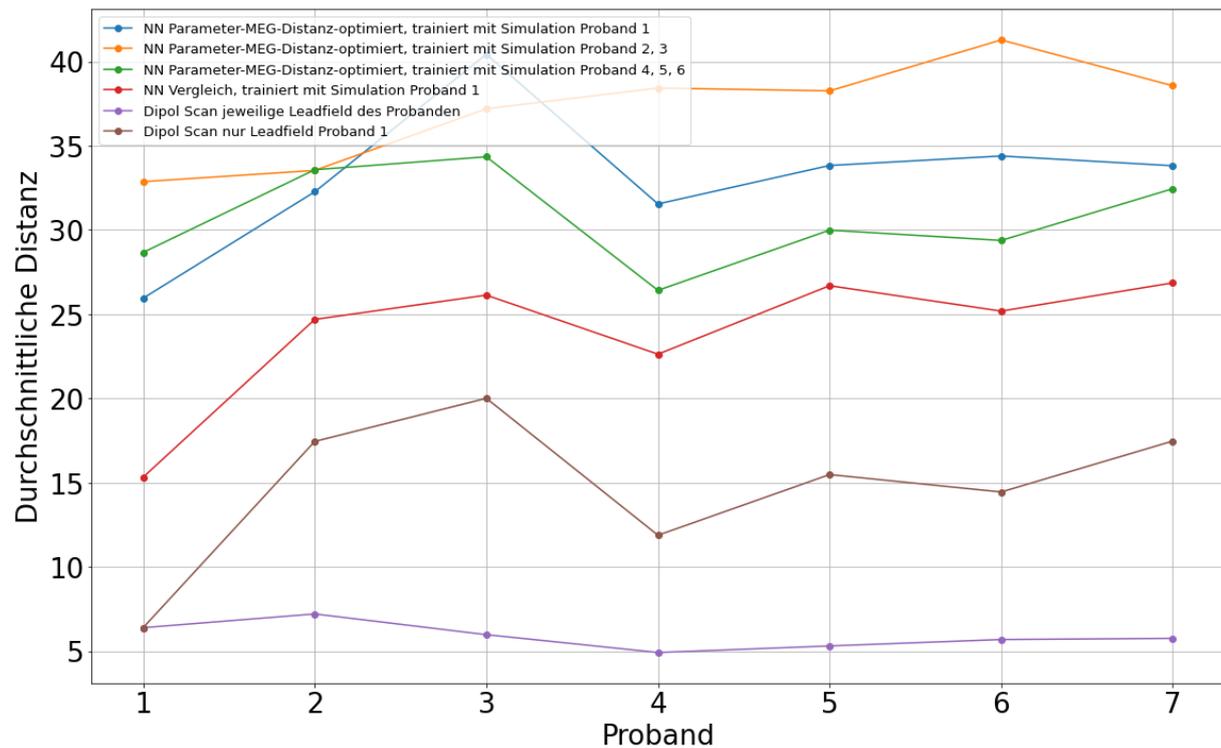


Abbildung 16: Durchschnittliche Distanz (in mm) der Lokalisierung verschiedener Methoden zur tatsächlichen ground truth der Simulation.

Abbildung 16 zeigt die Ergebnisse auf simulierten Daten. Verglichen wurden dabei:

- **NN Parameter-MEG-Distanz-optimiert nur Simulation Proband 1**, ein neuronales Netz, das mit den Parametern aus der Hyperparameteroptimierung auf den simulierten Daten von Proband 1 trainiert wurde,
- **NN Parameter-MEG-Distanz-optimiert nur Simulation Proband 2, 3**, welches auf den simulierten Daten von Proband 2 und 3 trainiert wurde,
- **NN Parameter-MEG-Distanz-optimiert nur Simulation Proband 4, 5, 6**, welches auf den simulierten Daten von Proband 4, 5 und 6 trainiert wurde,
- **NN Vergleich nur Simulation Proband 1**, ein neuronales Netz mit Parametern, die manuell ohne Hyperparameteroptimierung angepasst wurden, trainiert auf den simulierten Daten von Proband 1,
- **Dipol Scan jeweilige Leadfields des Probanden**, ein Dipol Scan mit vollen Informationen, das für jeden Probanden die personalisierte Leadfield nutzt und
- **Dipol Scan nur Leadfield Proband 1**, das für jeden Probanden die Leadfield von Proband 1 anwendet.

Auf den simulierten Daten hat der Dipol Scan mit individueller Leadfield-Matrix über alle Probanden die beste Performance. Dies ist zu erwarten, da die jeweilige Leadfield-Matrix der Probanden genutzt wurde, um die Datensamples zu erstellen und der Dipol Scan so volle Informationen hat. Die Distanz von $\approx 5 \text{ mm}$ entsteht wahrscheinlich aus dem hinzugefügten Rauschen, da alle Datensamples mit 20 db additivem Rauschen belegt wurden.

Der Dipol Scan mit der Leadfield-Matrix nur von Proband 1 schneidet über die Probanden 2-7 deutlich schlechter ab, da die Leadfield von Proband 1 nicht genau zu den Kopfmodellen der anderen Probanden passt.

Die drei neuronalen Netze mit optimierten Hyperparametern schneiden auf den simulierten Daten schlechter ab als die Dipol Scans. Auch bei den neuronalen Netzen ist die Distanz für die im Trainingsdatensatz enthaltenen Probanden geringer. Dies führt zur zweiten Hypothese, dass Modelle auf Probanden, deren Kopfmodell in den Trainingsdaten enthalten ist, am besten abschneiden.

Das neuronale Netz zum Vergleich liegt von der Leistung etwa zwischen den Dipol Scans und den optimierten Netzen und zeigt, dass andere Hyperparameter als die auf den MEG-Dipol Scan optimierten auf simulierten Daten bessere Vorhersagen erzeugen können.

Die durchschnittlichen Distanzen pro Kategorie (bekannt, unbekannt) werden in *Tabelle 7* aufgelistet und unterstützen Hypothese 2.

Tabelle 7: Durchschnittliche Distanzen von verschiedenen Methoden jeweils auf bekannten und unbekannt simulierten Daten.

Modell	Mittlere Distanz auf Tests, die in den Trainingsdaten enthalten sind	Mittlere Distanz auf neuen Daten
NN Simulation Proband 1	25.9 mm	34.5 mm
NN Simulation Proband 2, 3	35.3 mm	39.1 mm
NN Simulation Proband 4, 5, 6	28.6 mm	32.3 mm
NN Vergleich Proband 1	15.3 mm	25.4 mm
Dipol Scan alle Probanden	5.9 mm	–
Dipol Scan Proband 1	6.4 mm	16.1 mm

5.3. Validierung auf realen EEG-Daten

Da die Methode für die Quelllokalisierung von realen Messdaten entwickelt wurde, sollte diese auch auf realen Daten validiert werden. Zur Validierung stehen die in Kapitel 4.1.1. *Beschreibung der verfügbaren Daten* genannten Daten zur Verfügung.

Zur Validierung der Netze wird der Lokalisationsfehler sowohl zum Ergebnis des EEG- als auch des MEG-Dipol Scans der P20/N20 Events verwendet. Es werden dieselben KI-Modelle wie in 5.2. *Evaluation auf simulierten Daten* verwendet. Die Dipol Scan Methoden sind nicht enthalten, da diese die ground truth darstellen. Die durchschnittlichen Distanzen der Modelle zum Ergebnis des EEG- und MEG-Dipol Scans für alle Probanden sind in *Abbildung 17* und *Abbildung 18* dargestellt. Die durchschnittlichen Distanzen über alle Probanden sind in *Tabelle 8* aufgelistet.

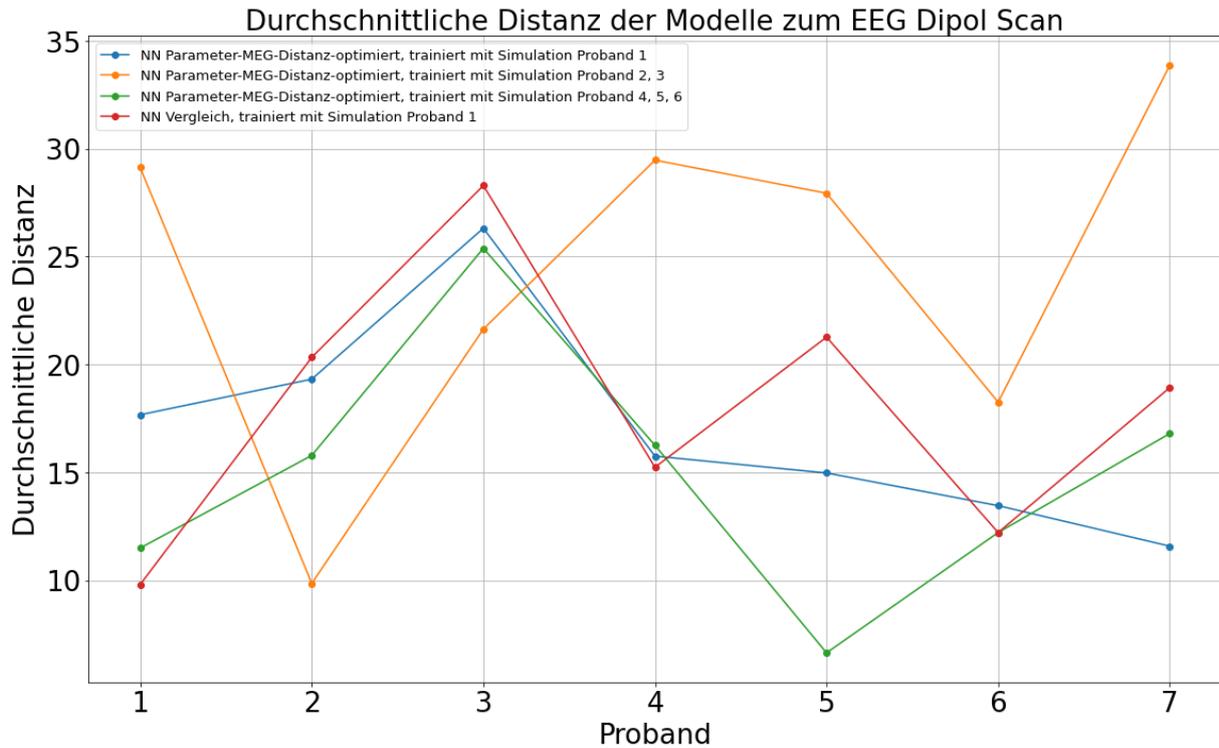


Abbildung 17: Durchschnittliche Distanzen der Vorhersagen der KI-Modelle zur Vorhersage des EEG-Dipol Scans mit der jeweiligen individualisierten Leadfield-Matrix.

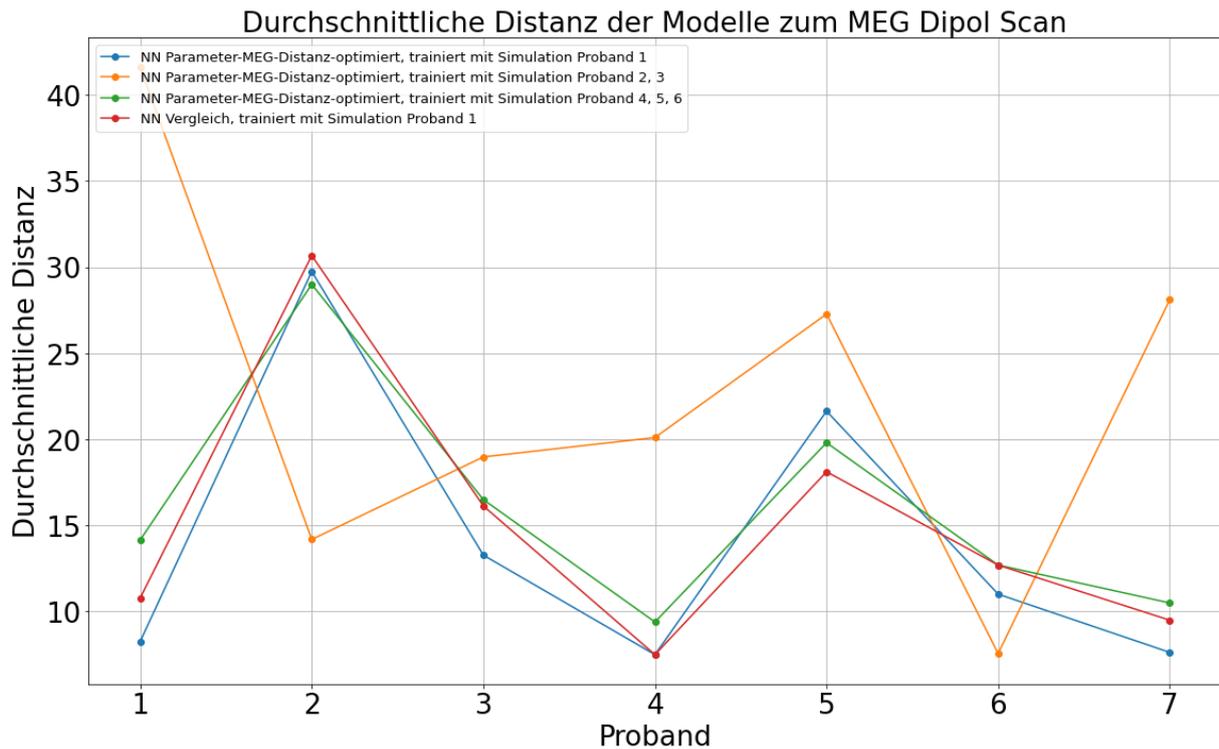


Abbildung 18: Durchschnittliche Distanzen der Vorhersagen der KI-Modelle zur Vorhersage des MEG-Dipol Scans mit der jeweiligen individualisierten Leadfield-Matrix.

Tabelle 8: durchschnittliche Distanzen der Vorhersagen der KI-Modelle zur Vorhersage des EEG- und MEG-Dipol Scans mit der jeweiligen individualisierten Leadfield-Matrix über alle Probanden.

Modell	Durchschnittliche Distanz zum EEG Dipol Scan	Durchschnittliche Distanz zum MEG Dipol Scan
NN Simulation Proband 1	17.0 mm	14.1 mm
NN Simulation Proband 2, 3	24.3 mm	22.5 mm
NN Simulation Proband 4, 5, 6	14.9 mm	16.0 mm
NN Vergleich Proband 1	18.0 mm	15.0 mm

Allgemein haben die Distanzen der Modelle beim EEG- und MEG-Dipol Scan einen ähnlichen Verlauf über die verschiedenen Probanden. Dies ist schlüssig, da es die gleichen Methoden auf den grundsätzlich gleichen neuronalen Aktivitäten sind. Gerade auf den hier vorkommenden P20/N20 Daten erzeugt der MEG-Dipol Scan jedoch bessere Lokalisierungen. Deshalb wird im Folgenden hauptsächlich auf diesen eingegangen.

Die neuronalen Netze Parameter-MEG-Distanz-optimiert Proband 1, Parameter-MEG-Distanz-optimiert Proband 4, 5, 6 und das Vergleich Proband 1 Netz schneiden in beiden Fällen ähnlich zueinander ab. Das Netz mit 2 Probanden schneidet über beide Verfahren fast konstant schlechter ab.

Nun werden die Ergebnisse mit den aufgestellten Hypothesen verglichen:

Hypothese 1: Die Performance auf simulierten Daten korreliert nicht eindeutig mit der Leistung auf realen Daten.

Die vorliegenden Ergebnisse unterstützen Hypothese 1 weiter. Die 3 ähnlich guten Netze auf dem MEG-Dipol Scan haben stark abweichende Performances auf den simulierten Daten. So ist das beste Modell auf ungesehenen simulierten Daten mit Abstand das Vergleichsnetz. Auf den MEG-Dipol Scan Distanzen ist es unter den 3 guten Netzen jedoch nur von mittlerer Performance. Das beste Netz auf dem MEG-Dipol Scan ist unter den 3 ähnlich guten Netzen das schwächste auf ungesehenen Simulationen. *Abbildung 19* zeigt die durchschnittliche Distanz zum Target (ground truth der Simulation und MEG-Dipol Scan). Die Distanzen für Simulation und reale Daten sind für jedes Modell außer das 3 Probanden NN unterschiedlich. Die Rangfolge der besten Modelle ist für beide Metriken komplett unterschiedlich. Dadurch lässt sich das Ergebnis von Hypothese 1 nochmal klar veranschaulichen.

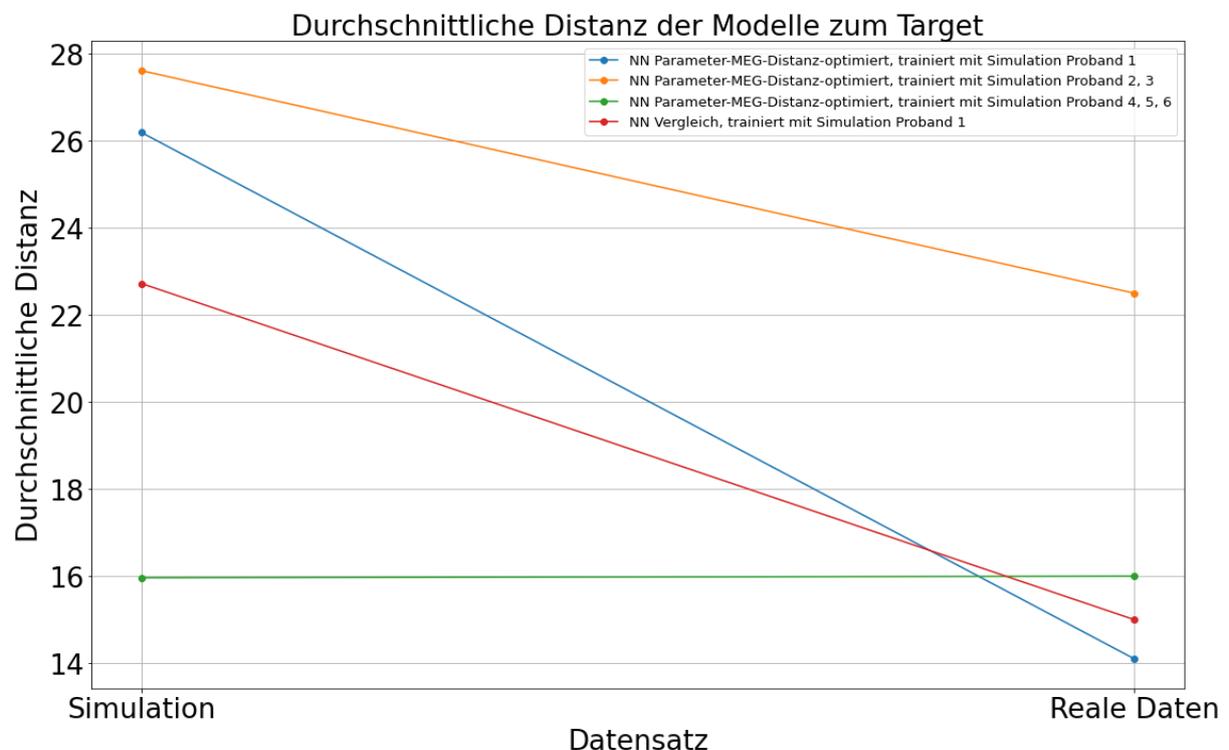


Abbildung 19: Durchschnittliche Distanz der KI-Modelle zum Target auf simulierten und realen Daten über alle Probanden. Bei den simulierten Daten ist die zufällig gewählte Quelle für die Simulation vorhanden und die Distanz kann einfach berechnet werden. Für die realen Daten wird das Ergebnis des MEG-Dipol Scans als Target verwendet.

Bisher ist es üblich, eine neue Deep Learning Methode auf eine möglichst gute Performance auf den Trainingsdaten hin zu optimieren. Hypothese 1 stellt das bisher übliche Verfahren in Frage, da es nur inkonsistent zu guten Modellen auf realen Daten führt. Der in dieser Arbeit angewandte Weg, die Hyperparameter der Netze so anzupassen, dass diese nach dem Training (während dem Training werden die Netze immer noch auf die Trainingsdaten hin angepasst) so nah wie möglich an den Quellen realer Daten, repräsentiert durch die MEG-Dipol Scan Lösung, liegen, führt konstant zu guten Modellen auf realen Daten und ermöglicht auch eine insgesamt bessere Leistung auf realen Daten.

Hypothese 2: Modelle schneiden auf Probanden deren Kopfmodell zum Erstellen des Trainingsdatensatz verwendet wurde am besten ab.

Diese Hypothese kann auf den realen Daten klar widerlegt werden. Hier gilt das Gegenteil: die Modelle mit den Daten von Proband 1 und das Modell mit den Daten der Probanden 4, 5 und 6 schneiden über alle Probanden ähnlich ab. Dies lässt vermuten, dass das Hinzufügen von mehr Kopfmodellen im Training zumindest auf diesem Datensatz keinen großen Mehrwert bringt. Im besten Fall kann so ein Netz, nachdem es einmal trainiert wurde, direkt auf neuen Probanden angewendet werden und muss nicht extra für jeden Probanden trainiert werden.

5.4. Inferenzzeit

Das Training eines Modells hat auf einer NVIDIA GeForce RTX 3070 mit 250000 Samples im Schnitt etwa 4 Stunden gedauert.

Über jeweils 100 Samples haben die optimierten Netze eine durchschnittliche Inferenzzeit pro Sample von 0.0015 Sekunden auf der GPU und 0.019 Sekunden auf der CPU. Damit liegt die Inferenzzeit deutlich unter der von klassischen Methoden und verwendeten Modelle können selbst bei Echtzeitanwendungen eingesetzt werden.

6. Grenzen der Arbeit und Ausblick

Des Weiteren werden nun die Grenzen der Arbeit betrachtet und wie man diese mit zukünftigen Arbeiten reduzieren könnte:

Das beste Modell im Vergleich zum MEG-Dipol Scan ist das optimierte Netz Proband 1. Die Hyperparameteroptimierung hat allerdings auch genau ein Netz mit den Trainingsdaten von Proband 1 optimiert. Es kann also sein, dass die Parameter der optimierten Modelle allgemein besonders für das Kopfmodell von Proband 1 optimiert wurden. Das Netz mit denselben Parametern und den Trainingsdaten von 2 Probanden schneidet auf beiden Metriken am schlechtesten ab. Um dies zu lösen könnte eine zukünftige Hyperparameteroptimierung, statt konstant auf einem Kopfmodell zu trainieren, einen Cross-Validation Ansatz verwenden und so alle Kopfmodelle in die Optimierung miteinbeziehen. Auch der Vergleich von Modellen, die durch Hyperparameteroptimierung auf den Validierungs Loss der Simulation optimiert wurden mit den MEG-Dipol Scan optimierten Netzen wäre hier wertvoll.

Die Größe des Experiments ist auf 7 Probanden und 4 neuronale Netze begrenzt und die Ergebnisse deshalb nur schwer zu verallgemeinern. Zukünftige Arbeiten in diesem Bereich sollten die Ergebnisse auf einem größeren Datensatz mit mehr Probanden validieren. Des Weiteren wäre eine öffentliche Sammlung von EEG-Datensätzen mit entsprechenden MEG-Dipol Scan sehr hilfreich, da so die verschiedenen KI-Modelle untereinander vergleichbar wären. Dieser öffentliche Benchmark Datensatz wäre deshalb besonders hilfreich, da er mit mehr MEG-Messungen einfach zu erweitern ist und bei ausreichender Größe vielleicht auch als zusätzlicher Input für das Training von Netzen fungieren könnte. Hier könnten auch mehrere verschiedene Kopfmodelle wie BEM und FEM miteinander verglichen werden.

Auch die Größe der simulierten Datensätze und KI-Modelle könnte erhöht werden. Dies würde zwar die Trainingsdauer deutlich erhöhen, könnte jedoch auch zu noch besseren Modellen führen.

Die beobachteten Ergebnisse könnten ausschließlich auf der in dieser Arbeit angewandten Simulation auftreten. Dies ist unwahrscheinlich, da die Simulation eng an bestehende Arbeiten anknüpft, was das Risiko allerdings nicht komplett ausschließt. Es wäre möglich, die hier vorgeschlagene Methode der Optimierung auf den MEG-Dipol Scan zu nutzen, um die Simulation möglichst optimal zu parametrisieren. So könnten für verschiedene Simulationsparameter jeweils verschiedene klassische und Deep Learning Methoden angewandt werden und die Simulation iterativ so angepasst werden, dass die Modelle auf der Simulation eine möglichst gute Leistung auf realen Daten erbringen. Diese Methode der Optimierung der Datensimulation könnte für viele verschiedene Modelle und Simulationen verwendet werden.

In dieser Arbeit wird der MEG-Dipol Scan als bestmögliche Annäherung an die Quellen der realen Daten verwendet. Auch wenn dieser auf den hier vorkommenden P20/N20 Daten in der Regel bessere Lokalisierungen erzeugt als der EEG-Dipol Scan, ist der MEG-Dipol Scan auch nicht fehlerfrei. Es könnten in Zukunft auch andere Lokalisierungsmethoden, wie beispielsweise neuartige Dipol Scans aufgrund von den kombinierten EMEG-Daten als ground truth anstatt des MEG-Dipol Scans getestet werden [73]. Die hier vorgestellte, sowie alle neuen Methoden sollten auf realen Daten noch von Experten validiert werden.

7. Fazit

7.1. Zusammenfassung der wesentlichen Erkenntnisse

In dieser Arbeit wurde erstmals ein Deep Learning Framework vorgestellt, das die Lokalisierung von EEG-Quellen mithilfe simulierter Trainingsdaten mit einer auf MEG-Dipol Scan Lösungen basierten Hyperparameteroptimierung betrachtet und auf realen Daten validiert wird.

Die Kombination aus einer physikbasierten Datensimulation und einem optimierten neuronalen Netz führte zu einem Modell mit einem durchschnittlichen Lokalisationsfehler auf den realen Daten von 7 Probanden von 17.0 und 14.1 mm zum EEG- und MEG-Dipol Scan. Die Interferenzzeit unterschreitet deutlich die Echtzeitschwelle.

Als Resultat aus den Experimenten in dieser Arbeit wird die Anwendung des MEG-Dipol Scans für die Lokalisierung von EEG-Daten bei kombinierten EMEG-Aufnahmen als Optimierungstarget für die Optimierung der neuronalen Netze vorgeschlagen. Es wurde gezeigt, dass durch die Optimierung auf die MEG-Dipol Lösungen konstant schon während dem Training quantifizierbare, auf den realen Daten besser abschneidende Netze erzeugt werden können.

Zusätzlich wird der MEG-Dipol Scan als einfache Validierungsmethode für Deep Learning Methoden vorgestellt und angewendet. Der Aufbau eines derartigen Benchmark-Datensatzes wird begründet.

Die aufgestellte Hypothese, dass mehr verschiedene Kopfmodelle im Trainingsdatensatz zu besserer Performance auf unbekanntem Daten führt wurde widerlegt.

Die gesamten Methoden dieser Arbeit sind nur auf kombinierte EMEG-Messungen anwendbar. Weitere Grenzen und mögliche zukünftige Forschungsrichtungen wurden vorgestellt.

7.2. Beitrag der Arbeit zur EEG-Quelllokalisierung

- Neue Evaluationsmetrik: Die Arbeit testet den MEG-Dipol Scan als quantitative Zielgröße für Training und Validierung von Deep Learning Modellen zur EEG-Quelllokalisierung. Dieser neuartige Ansatz erlaubt den einfachen und objektiven Vergleich unterschiedlicher Verfahren.
- Die Analyse widerlegt die Annahme, dass mehr Diversität durch verschiedene Kopfmodelle in den Trainingsdaten zu einer besseren Generalisierung auf realen Daten führt. Stattdessen wird gezeigt, dass bereits ein einziges Kopfmodell in den Simulationen zu robusten Performances führen kann.
- Reproduzierbare Open-Science-Pipeline: Sämtlicher Code, inklusive Datensimulation, Hyperparameteroptimierung, Training und Auswertung ist offen verfügbar. Dies erleichtert die zukünftige Weiterentwicklung.

8. Literaturverzeichnis

- [1] H. Berger, „Über das Elektrenkephalogramm des Menschen“, *Arch. Für Psychiatr. Nervenkrankh.*, Bd. 87, Nr. 1, S. 527–570, Dez. 1929, doi: 10.1007/BF01797193.
- [2] E. Niedermeyer und F. H. L. D. Silva, *Electroencephalography: Basic Principles, Clinical Applications and Related Fields*, 3. Aufl. Baltimore: Lippincott Williams and Wilkins, 1993.
- [3] L. Hecker, R. Rupperecht, L. Tebartz Van Elst, und J. Kornmeier, „ConvDip: A Convolutional Neural Network for Better EEG Source Imaging“, *Front. Neurosci.*, Bd. 15, Juni 2021, doi: 10.3389/fnins.2021.569918.
- [4] S. J. Luck, „An Introduction to the Event-Related Potential Technique“, MIT Press. Zugegriffen: 19. März 2025. [Online]. Verfügbar unter: <https://mitpress.mit.edu/9780262525855/an-introduction-to-the-event-related-potential-technique/>
- [5] C. M. Michel, M. M. Murray, G. Lantz, S. Gonzalez, L. Spinelli, und R. Grave De Peralta, „EEG source imaging“, *Clin. Neurophysiol.*, Bd. 115, Nr. 10, S. 2195–2222, Okt. 2004, doi: 10.1016/j.clinph.2004.06.001.
- [6] R. Grech u. a., „Review on solving the inverse problem in EEG source analysis“, *J. NeuroEngineering Rehabil.*, Bd. 5, Nr. 1, S. 25, Nov. 2008, doi: 10.1186/1743-0003-5-25.
- [7] R. D. Pascual-Marqui u. a., „Low resolution brain electromagnetic tomography (LORETA) functional imaging in acute, neuroleptic-naive, first-episode, productive schizophrenia“, *Psychiatry Res.*, Bd. 90, Nr. 3, S. 169–179, Juni 1999, doi: 10.1016/s0925-4927(99)00013-x.
- [8] R. Grech u. a., „Review on solving the inverse problem in EEG source analysis“, *J. Neuroengineering Rehabil.*, Bd. 5, S. 25, Nov. 2008, doi: 10.1186/1743-0003-5-25.
- [9] Michael D. Nuenz, „Electroencephalography (EEG): neurophysics, experimental methods, and signal processing“, in *ResearchGate*, Bd. Handbook of Neuroimaging Data Analysis (pp.175-197), Chapman & Hall/CRC, 2016. doi: 10.13140/RG.2.2.12706.63687.
- [10] F. Lopes da Silva, „EEG and MEG: Relevance to Neuroscience“, *Neuron*, Bd. 80, Nr. 5, S. 1112–1128, Dez. 2013, doi: 10.1016/j.neuron.2013.10.017.
- [11] R. Coben, D. C. Hammond, und M. Arns, „19 Channel Z-Score and LORETA Neurofeedback: Does the Evidence Support the Hype?“, *Appl. Psychophysiol. Biofeedback*, Bd. 44, Nr. 1, S. 1–8, 2019, doi: 10.1007/s10484-018-9420-6.
- [12] „Epilepsy“. Zugegriffen: 19. März 2025. [Online]. Verfügbar unter: <https://www.who.int/news-room/fact-sheets/detail/epilepsy>
- [13] Athanasios Delatolas, Michaelis Zervakis, und Carsten H. Wolters, „Electromagnetic brain source analysis with statistical and deep learning approaches“, Technical university of crete, crete, 2022.
- [14] S. Herculano-Houzel, „The human brain in numbers: a linearly scaled-up primate brain“, *Front. Hum. Neurosci.*, Bd. 3, S. 31, 2009, doi: 10.3389/neuro.09.031.2009.
- [15] D. Purves, Hrsg., *Neuroscience*, Sixth edition. New York: Oxford University Press, 2018.

- [16] E. R. Kandel, Hrsg., *Principles of neural science*, 5th ed. New York: McGraw-Hill, 2013.
- [17] „Aufbau einer Nervenzelle“. Zugegriffen: 10. Juni 2025. [Online]. Verfügbar unter: <https://www.studienkreis.de/biologie/nervenzelle-aufbau/>
- [18] G. Buzsáki, C. A. Anastassiou, und C. Koch, „The origin of extracellular fields and currents — EEG, ECoG, LFP and spikes“, *Nat. Rev. Neurosci.*, Bd. 13, Nr. 6, S. 407–420, Juni 2012, doi: 10.1038/nrn3241.
- [19] L. Alonso-Nanclares, J. Gonzalez-Soriano, J. R. Rodriguez, und J. DeFelipe, „Gender differences in human cortical synaptic density“, *Proc. Natl. Acad. Sci. U. S. A.*, Bd. 105, Nr. 38, S. 14615–14619, Sep. 2008, doi: 10.1073/pnas.0803652105.
- [20] P. F. M. Ribeiro *u. a.*, „The human cerebral cortex is neither one nor many: neuronal distribution reveals two quantitatively different zones in the gray matter, three in the white matter, and explains local variations in cortical folding“, *Front. Neuroanat.*, Bd. 7, Sep. 2013, doi: 10.3389/fnana.2013.00028.
- [21] S. Herculano-Houzel, C. R. Watson, und G. Paxinos, „Distribution of neurons in functional areas of the mouse cerebral cortex reveals quantitatively different cortical zones“, *Front. Neuroanat.*, Bd. 7, Okt. 2013, doi: 10.3389/fnana.2013.00035.
- [22] „Basic mathematical and electromagnetic concepts of the biomagnetic inverse problem - PubMed“. Zugegriffen: 21. Januar 2025. [Online]. Verfügbar unter: <https://pubmed.ncbi.nlm.nih.gov/3823129/>
- [23] M. Hämäläinen, R. Hari, R. J. Ilmoniemi, J. Knuutila, und O. V. Lounasmaa, „Magnetoencephalography---theory, instrumentation, and applications to noninvasive studies of the working human brain“, *Rev. Mod. Phys.*, Bd. 65, Nr. 2, S. 413–497, Apr. 1993, doi: 10.1103/RevModPhys.65.413.
- [24] R. Plonsey und D. B. Heppner, „Considerations of quasi-stationarity in electrophysiological systems“, *Bull. Math. Biophys.*, Bd. 29, Nr. 4, S. 657–664, Dez. 1967, doi: 10.1007/BF02476917.
- [25] J. C. de Munck und B. W. van Dijk, „Symmetry considerations in the quasi-static approximation of volume conductor theory“, *Phys. Med. Biol.*, Bd. 36, Nr. 4, S. 521, Apr. 1991, doi: 10.1088/0031-9155/36/4/009.
- [26] B. Wang *u. a.*, „Quasistatic approximation in neuromodulation“, *J. Neural Eng.*, Bd. 21, Nr. 4, S. 041002, Aug. 2024, doi: 10.1088/1741-2552/ad625e.
- [27] H. McCann, G. Pisano, und L. Beltrachini, „Variation in Reported Human Head Tissue Electrical Conductivity Values“, *Brain Topogr.*, Bd. 32, Nr. 5, S. 825–858, Sep. 2019, doi: 10.1007/s10548-019-00710-2.
- [28] H. Hallez *u. a.*, „Review on solving the forward problem in EEG source analysis“, *J. NeuroEngineering Rehabil.*, Bd. 4, Nr. 1, S. 46, Nov. 2007, doi: 10.1186/1743-0003-4-46.
- [29] J. Haueisen, A. Böttner, H. Nowak, H. Brauer, und C. Weiller, „The influence of conductivity changes in boundary element compartments on the forward and inverse problem in electroencephalography and magnetoencephalography“, *Biomed. Tech. (Berl)*, Bd. 44, Nr. 6, S. 150–157, Juni 1999, doi: 10.1515/bmte.1999.44.6.150.

- [30] „Effect of realistic human head modelling on brain source distribution“, *ResearchGate*, doi: 10.1049/el.2012.1569.
- [31] T. Erdbrügger u. a., „CutFEM forward modeling for EEG source analysis“, *Front. Hum. Neurosci.*, Bd. 17, S. 1216758, Aug. 2023, doi: 10.3389/fnhum.2023.1216758.
- [32] R. J. Croft und R. J. Barry, „Removal of ocular artifact from the EEG: a review“, *Neurophysiol. Clin. Neurophysiol.*, Bd. 30, Nr. 1, S. 5–19, Feb. 2000, doi: 10.1016/S0987-7053(00)00055-1.
- [33] J. M. Winkler, „Entwicklung eines modellgestützten Deep Learning Ansatzes zur Rekonstruktion multiparametrischer Magnetresonanz-Korrelationsspektren“, Technische Hochschule Ingolstadt, 2023. Zugegriffen: 4. Juni 2025. [Online]. Verfügbar unter: <https://opus4.kobv.de/opus4-haw/frontdoor/index/index/docId/3644>
- [34] I. Goodfellow, Y. Bengio, und A. Courville, *Deep Learning*. Cambridge, Massachusetts: The MIT Press, 2016.
- [35] G. Görz, U. Schmid, J. Schneeberger, und T. Braun, *Handbuch der Künstlichen Intelligenz*, 6. Auflage. Berlin: Walter de Gruyter GmbH, 2021.
- [36] D. P. Kingma und J. Ba, „Adam: A Method for Stochastic Optimization“, 30. Januar 2017, *arXiv*: arXiv:1412.6980. doi: 10.48550/arXiv.1412.6980.
- [37] I. Sutskever, J. Martens, G. Dahl, und G. Hinton, „On the importance of initialization and momentum in deep learning“, in *Proceedings of the 30th International Conference on Machine Learning*, PMLR, Mai 2013, S. 1139–1147. Zugegriffen: 4. Juni 2025. [Online]. Verfügbar unter: <https://proceedings.mlr.press/v28/sutskever13.html>
- [38] X. Glorot und Y. Bengio, „Understanding the difficulty of training deep feedforward neural networks“, in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, JMLR Workshop and Conference Proceedings, März 2010, S. 249–256. Zugegriffen: 4. Juni 2025. [Online]. Verfügbar unter: <https://proceedings.mlr.press/v9/glorot10a.html>
- [39] K. He, X. Zhang, S. Ren, und J. Sun, „Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification“, in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dez. 2015, S. 1026–1034. doi: 10.1109/ICCV.2015.123.
- [40] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, und R. Salakhutdinov, „Dropout: A Simple Way to Prevent Neural Networks from Overfitting“, *J. Mach. Learn. Res.*, Bd. 15, Nr. 56, S. 1929–1958, 2014.
- [41] S. Ioffe und C. Szegedy, „Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift“, in *Proceedings of the 32nd International Conference on Machine Learning*, PMLR, Juni 2015, S. 448–456. Zugegriffen: 4. Juni 2025. [Online]. Verfügbar unter: <https://proceedings.mlr.press/v37/ioffe15.html>
- [42] K. He, X. Zhang, S. Ren, und J. Sun, „Deep Residual Learning for Image Recognition“, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Juni 2016, S. 770–778. doi: 10.1109/CVPR.2016.90.
- [43] A. F. Agarap, „Deep Learning using Rectified Linear Units (ReLU)“, 7. Februar 2019, *arXiv*: arXiv:1803.08375. doi: 10.48550/arXiv.1803.08375.

- [44] B. Xu, N. Wang, T. Chen, und M. Li, „Empirical Evaluation of Rectified Activations in Convolutional Network“, 27. November 2015, *arXiv*: arXiv:1505.00853. doi: 10.48550/arXiv.1505.00853.
- [45] P. Ramachandran, B. Zoph, und Q. Le, „Swish: a Self-Gated Activation Function“, Okt. 2017, doi: 10.48550/arXiv.1710.05941.
- [46] S. Elfving, E. Uchibe, und K. Doya, „Sigmoid-Weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning“, 2. November 2017, *arXiv*: arXiv:1702.03118. doi: 10.48550/arXiv.1702.03118.
- [47] D. Hendrycks und K. Gimpel, „Gaussian Error Linear Units (GELUs)“, 11. November 2018, *arXiv*: arXiv:1606.08415. doi: 10.48550/arXiv.1606.08415.
- [48] J. Winkler, C. Uhl, S. Geißelsöder, T. Erdbrügger, und C. Wolters, „Comparison of Classical EEG Source Analysis with Deep Learning“, in *CONTROLO 2024*, A. P. Aguiar, P. Rocha Malonek, V. H. Pinto, F. A. C. C. Fontes, und R. Chertovskih, Hrsg., Cham: Springer Nature Switzerland, 2025, S. 258–267. doi: 10.1007/978-3-031-81724-3_24.
- [49] G. V. Hoey *u. a.*, „EEG dipole source localization using artificial neural networks“, *Phys. Med. Biol.*, Bd. 45, Nr. 4, S. 997, Apr. 2000, doi: 10.1088/0031-9155/45/4/314.
- [50] S. C. Jun, B. A. Pearlmutter, und G. Nolte, „Fast accurate MEG source localization using a multilayer perceptron trained with real brain noise“, *Phys. Med. Biol.*, Bd. 47, Nr. 14, S. 2547–2560, Juli 2002, doi: 10.1088/0031-9155/47/14/312.
- [51] S. C. Jun und B. A. Pearlmutter, „Fast robust subject-independent magnetoencephalographic source localization using an artificial neural network“, *Hum. Brain Mapp.*, Bd. 24, Nr. 1, S. 21–34, Sep. 2004, doi: 10.1002/hbm.20068.
- [52] A. Razorenova, N. Yavich, M. Malovichko, M. Fedorov, N. Koshev, und D. Dylov, „Deep Learning for Non-invasive Cortical Potential Imaging“, 2020, S. 45–55. doi: 10.1007/978-3-030-66843-3_5.
- [53] R. Sun, A. Sohrabpour, S. Ye, und B. He, „SIFNet: Electromagnetic Source Imaging Framework Using Deep Neural Networks“, 9. Juli 2020, *bioRxiv*. doi: 10.1101/2020.05.11.089185.
- [54] H. Yokoyama *u. a.*, „M/EEG source localization for both subcortical and cortical sources using a convolutional neural network with a realistic head conductivity model“, *APL Bioeng.*, Bd. 8, Nr. 4, S. 046104, Okt. 2024, doi: 10.1063/5.0226457.
- [55] D. Pantazis und A. Adler, „MEG Source Localization via Deep Learning“, 1. Dezember 2020, *arXiv*: arXiv:2012.00588. doi: 10.48550/arXiv.2012.00588.
- [56] C. Wei, K. Lou, Z. Wang, M. Zhao, D. Mantini, und Q. Liu, „Edge Sparse Basis Network: A Deep Learning Framework for EEG Source Localization“, 16. Juni 2021, *arXiv*: arXiv:2102.09188. doi: 10.48550/arXiv.2102.09188.
- [57] T. Erdbrügger *u. a.*, „CutFEM forward modeling for EEG source analysis“, *Front. Hum. Neurosci.*, Bd. 17, Aug. 2023, doi: 10.3389/fnhum.2023.1216758.
- [58] „MNE — MNE 1.9.0 documentation“. Zugegriffen: 24. Juni 2025. [Online]. Verfügbar unter: <https://mne.tools/stable/index.html>

- [59] M. Antonakakis u. a., „Inter-Subject Variability of Skull Conductivity and Thickness in Calibrated Realistic Head Models“, *NeuroImage*, Bd. 223, S. 117353, Dez. 2020, doi: 10.1016/j.neuroimage.2020.117353.
- [60] F. Neugebauer u. a., „Validating EEG, MEG and Combined MEG and EEG Beamforming for an Estimation of the Epileptogenic Zone in Focal Cortical Dysplasia“, *Brain Sci.*, Bd. 12, Nr. 1, S. 114, Jan. 2022, doi: 10.3390/brainsci12010114.
- [61] „Mapping the signal-to-noise-ratios of cortical sources in magnetoencephalography and electroencephalography“, doi: 10.1002/hbm.20571.
- [62] A. Hillebrand und G. R. Barnes, „A Quantitative Assessment of the Sensitivity of Whole-Head MEG to Activity in the Adult Human Cortex“, *NeuroImage*, Bd. 16, Nr. 3, Part A, S. 638–650, Juli 2002, doi: 10.1006/nimg.2002.1102.
- [63] M.-X. Huang u. a., „A novel integrated MEG and EEG analysis method for dipolar sources“, *NeuroImage*, Bd. 37, Nr. 3, S. 731–748, Sep. 2007, doi: 10.1016/j.neuroimage.2007.06.002.
- [64] „EEG source localization MEG dipole scan ground truth - Search Results“, PubMed. Zugegriffen: 24. Juni 2025. [Online]. Verfügbar unter: <https://pubmed.ncbi.nlm.nih.gov/?term=EEG+source+localization+MEG+dipole+scan+ground+truth&size=200>
- [65] M. H. Histed, V. Bonin, und R. C. Reid, „Direct activation of sparse, distributed populations of cortical neurons by electrical microstimulation“, *Neuron*, Bd. 63, Nr. 4, S. 508–522, Aug. 2009, doi: 10.1016/j.neuron.2009.07.016.
- [66] A. Fracasso, S. O. Dumoulin, und N. Petridou, „Point-spread function of the BOLD response across columns and cortical depth in human extra-striate cortex“, *Prog. Neurobiol.*, Bd. 202, S. 102034, Juli 2021, doi: 10.1016/j.pneurobio.2021.102034.
- [67] L. E. Muller, A. Reynaud, F. Chavane, und A. Destexhe, „Propagating waves structure spatiotemporal activity in visual cortex of the awake monkey“, *BMC Neurosci.*, Bd. 14, Nr. Suppl 1, S. O8, Juli 2013, doi: 10.1186/1471-2202-14-S1-O8.
- [68] R. A. Chowdhury, J. M. Lina, E. Kobayashi, und C. Grova, „MEG Source Localization of Spatially Extended Generators of Epileptic Activity: Comparing Entropic and Hierarchical Bayesian Approaches“, *PLOS ONE*, Bd. 8, Nr. 2, S. e55969, Feb. 2013, doi: 10.1371/journal.pone.0055969.
- [69] „Python Release Python 3.8.3“, Python.org. Zugegriffen: 25. Juni 2025. [Online]. Verfügbar unter: <https://www.python.org/downloads/release/python-383/>
- [70] „PyTorch“, PyTorch. Zugegriffen: 25. Juni 2025. [Online]. Verfügbar unter: <https://pytorch.org/>
- [71] M. Dörrich, M. Fan, und A. M. Kist, „Impact of Mixed Precision Techniques on Training and Inference Efficiency of Deep Neural Networks“, *IEEE Access*, Bd. 11, S. 57627–57634, 2023, doi: 10.1109/ACCESS.2023.3284388.
- [72] „Weights & Biases: The AI Developer Platform“, Weights & Biases. Zugegriffen: 25. Juni 2025. [Online]. Verfügbar unter: <https://wandb.ai/site/>

- [73] S. Bastola *u. a.*, „Improved Dipole Source Localization from Simultaneous MEG-EEG Data by Combining a Global Optimization Algorithm with a Local Parameter Search: A Brain Phantom Study“, *Bioengineering*, Bd. 11, Nr. 9, Art. Nr. 9, Sep. 2024, doi: 10.3390/bioengineering11090897.