

MARMARA ÜNİVERSİTÄT
DEUTSCHSPRACHIGE ABTEILUNG FÜR WIRTSCHAFTSINFORMATIK



WIFO-PROJEKT
KLASSIFIZIERUNG VON EEG-SIGNALEN MITTELS ANN UND CNN

138417056 ÖZGÜRAZAD ÇELİK

Betreuer: Doç.Dr. SELÇUK KIRAN

ISTANBUL, 2022

INHALT

I.	EINLEITUNG	4
II.	Anatomie des Gehirns	4
1.	Serebrum	5
2.	Kleinhirn.....	5
3.	Hirnstamm	5
4.	Frontal lobe.....	6
5.	Parietal lobe	6
6.	Occipital lobe	7
7.	Temporal lobe.....	7
III.	ANATOMIE EINES NEURONS.....	7
1.	Dendriten	8
2.	Soma (Zellkörper)	9
3.	Axon	9
4.	Myelinscheide	9
5.	Axon Terminals	10
IV.	BCI	12
A.	Invasive Methoden	12
1.	ECoG.....	13
B.	Nicht-invasive Methoden	13
1.	Magnetoencephalography (MEG).....	14
2.	Functional Magnetic Resonance Imaging (fMRI).....	14
3.	Electroencephalography (EEG).....	15
V.	WAS IST EEG UND WIE FUNKTIONIERT ES?.....	15
A.	EEG im Vergleich zu anderen bildgebenden Verfahren des Gehirns	16
B.	EEG Frequenzbereiche / Frequenzbänder	17
C.	EEG-Elektroden	21
1.	Elektrodenarrays und Platzierung.....	22
D.	Steuersignaltypen in BCI-Systemen.....	24

1. Visual Evoked Potentials (VEPs).....	25
2. Slow Cortical Potentials (SCPs).....	27
3. P300 Evoked Potentials.....	27
4. Sensorimotor Rhythms (mu and beta rhythms).....	28
E. Arten von BCIs.....	31
F. Komponenten von BCI.....	34
1. Signal Acquisition	35
2. Signal Processing.....	35
3. Feature Translation and Feedback.....	60
VI. Klassifizierung von EEG-Signalen durch ANN und CNN.....	60
1. Material	60
2. Testprotokoll	61
3. Vorläufig	62
4. Systeminstallation.....	64
VII. FAZIT.....	72
VIII. Kaynakça.....	73

Figure 1 The brain has three main parts: the cerebrum, cerebellum and brainstem.	4
Figure 2 The cerebrum is divided into four lobes: frontal, parietal, occipital and temporal.	6
Figure 3 Anatomy of a Neuron.....	8
Figure 4 The Action Potential	11
Figure 5 Brainwave Frequencies	19
Figure 6 EEG frequency spectrum and its frequency bands.....	20
Figure 7 Sample EEG data recorded from ‘F3–C3’ location.	20
Figure 8 Scheme of the Ag/AgCl electrodes (a) typical Ag/AgCl electrode with a Vycor tip and (b) Ag/AgCl electrode with a tiny perforation to accelerate the diffusion of the internal electrolyte into the test solution: 1, Ag wire; 2, AgCl layer; 3, KCl solution;	22
Figure 9 The EEG electrodes placement system “10–20”.....	23

Figure 10 a) ERD/ERS components in an observed brain signal. b) Topographical map of ERD/ERS components in a hand movement motor imagery	29
Figure 11 Examples of ongoing EEG recorded during right finger movement.....	30
Figure 12 Conceptual illustration of an exogenous BCI	32
Figure 13 Stimuli for a P300 speller setting. The combination of.....	33
Figure 14 Framework for asynchronous and synchronous control protocols.....	34
Figure 15 Basic design and operation of a BCI system	34
Figure 16 Example Covariance Matrix	38
Figure 17 Dimension reduction using PCA.....	39
Figure 18 Common Spatial Pattern Maps.....	41
Figure 19 Wavelet transform illustration.....	42
Figure 20 Difference between LDA and PCA.....	45
Figure 21 Principle of support vector machine.....	47
Figure 22 Kernel Method in SVM.....	47
Figure 23 Steps of the K-nearest neighbor	49
Figure 24 Mathematical model of an artificial neural network (ANN)	50
Figure 25 Perceptron and ADALINE models	52
Figure 26 MLP model.....	52
Figure 27 Comparing of classifying ability of 3 methods	53
Figure 28 Feedforward Backpropagation Neural Network architecture.	54
Figure 29 Activation Functions	56
Figure 30 The overall architecture of the Convolutional Neural Network (CNN) includes an input layer, multiple alternating convolution and max-pooling layers, one fully-connected layer and one classification layer.	57
Figure 31 Illustration of Max Pooling and Average Pooling.....	59
Table 1 Summary of control signals.....	24
Table 2 Features of VEP modulations: t-VEP, f-VEP and, c-VEP.	26
Table 3 Main differences between exogenous and endogenous BCI.	31
Table 4 Main differences between synchronous and asynchronous BCIs.	32

I. EINLEITUNG

In dieser Studie werden die Funktionsweise der BCI-Systeme, ihr Verwendungszweck und die grundlegenden Komponenten, aus denen das BCI-System besteht, allgemein untersucht. Bei der Betrachtung von BCI-Systemen wird betont, wie die im System verwendeten Signale verarbeitet werden. In BCI-Systemen hergestellte Signalverarbeitungsverfahren wurden untersucht, und daher wird versucht, die Leistungsfähigkeit künstlicher neuronaler Netze, die ein moderner Ansatz sind, und Faltungs-Neuralnetzsysteme, die eine Art künstlicher neuronaler Netze sind, bei der Signalverarbeitung zu messen. Darüber hinaus werden Methoden der Artificial Neural Nets und Convolutional Neural Nets verglichen. Der Hauptgrund für die Wahl dieser Systeme ist, dass sie für die Merkmalsextraktion und -klassifizierung geeignet sein sollten, ohne dass eine Vorverarbeitung in Signalen erforderlich ist. Heutzutage ist mit der Zunahme der Verarbeitungsleistung von Computern die Verwendung dieser Systeme einfacher geworden.

II. ANATOMIE DES GEHIRNS

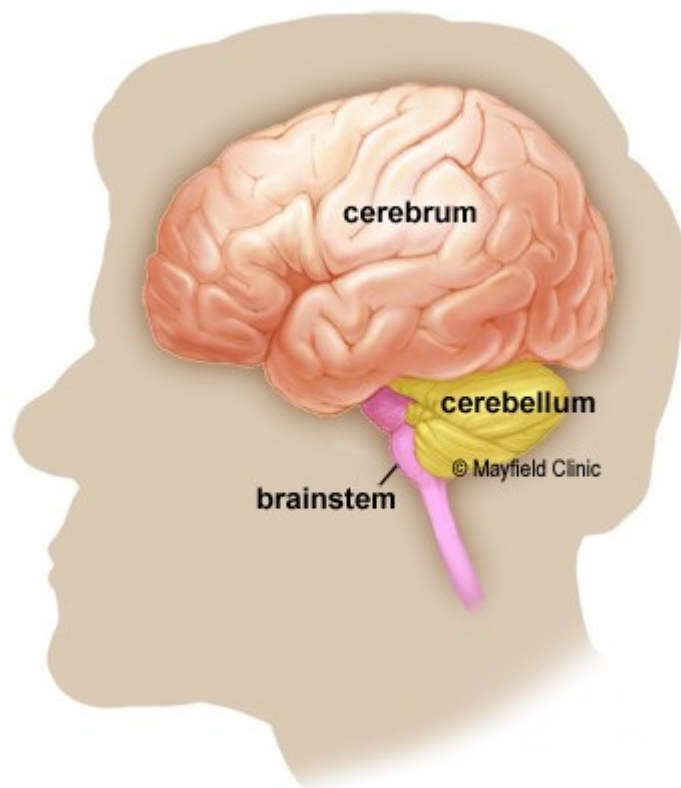


Figure 1 The brain has three main parts: the cerebrum, cerebellum and brainstem.

1. Serebrum

Es ist der größte Teil des Gehirns und besteht aus der rechten und linken Hemisphäre. Es erfüllt höhere Funktionen wie Berührung, Sehen und Hören sowie die präzise Kontrolle von Sprache, Argumentation, Emotionen, Lernen und Bewegung.

2. Kleinhirn

Es befindet sich unterhalb des Großhirns. Seine Funktion besteht darin, Muskelbewegungen zu koordinieren, die Körperhaltung aufrechtzuerhalten und das Gleichgewicht zu halten.

3. Hirnstamm

Es fungiert als Relaiszentrum, das Großhirn und Kleinhirn mit dem Rückenmark verbindet. Führt viele automatische Funktionen wie Atmung, Puls, Körpertemperatur, Wach- und Schlafzyklen, Verdauung, Niesen, Husten, Erbrechen, Schlucken aus (*Brain Anatomy, Anatomy of the Human Brain | Mayfield Brain & Spine Cincinnati, Ohio*, n.d.).

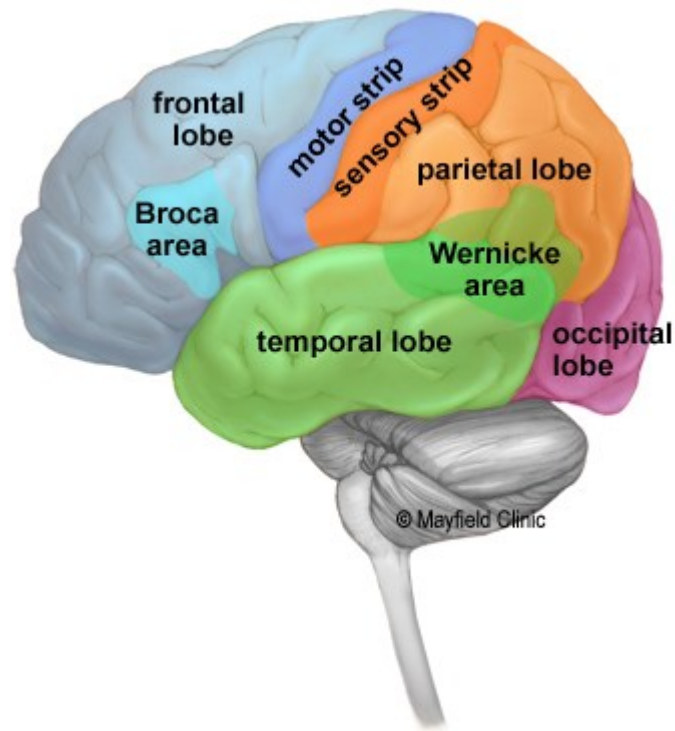


Figure 2 The cerebrum is divided into four lobes: frontal, parietal, occipital and temporal.

4. Frontal lobe

- Persönlichkeit, Verhalten, Emotionen
- Beurteilung, Planung, Problemlösung
- Sprache: Sprechen und Schreiben (Broca-Areal)
- Körperbewegung (Motorspur)
- Intelligenz, Konzentration, Selbstbewusstsein

5. Parietal lobe

- Interpretiert Sprache, Wörter
- Berührungsempfindung, Schmerz, Temperatur (Sensorstreifen)
- Interpretiert Signale von Sehen, Hören, Motorik, Sensorik und Gedächtnis

- Räumliche und visuelle Wahrnehmung

6. Occipital lobe

- Interpretiert das Sehen (Farbe, Licht, Bewegung)

7. Temporal lobe

Sprache verstehen (Wernicke-Areal)

- Erinnerung
- Hören
- Ranking und Organisation(*Brain Anatomy, Anatomy of the Human Brain | Mayfield Brain & Spine Cincinnati, Ohio, n.d.*)

III. ANATOMIE EINES NEURONS

Das Neuron enthält das Soma (Zellkörper), das das Axon (eine Nervenfaser, die elektrische Impulse vom Soma weg überträgt) und Dendriten (baumartige Strukturen, die Signale von anderen Neuronen empfangen) verlängert. Die Myelinscheide ist eine isolierende Schicht, die sich um das Axon bildet und eine schnellere Übertragung von Nervenimpulsen entlang des Axons ermöglicht.

Neuronen berühren sich nicht und zwischen dem Axon eines Neurons und dem Dendriten eines anderen gibt es eine Lücke, die als Synapse bezeichnet wird(Guy-Evans, 2021).

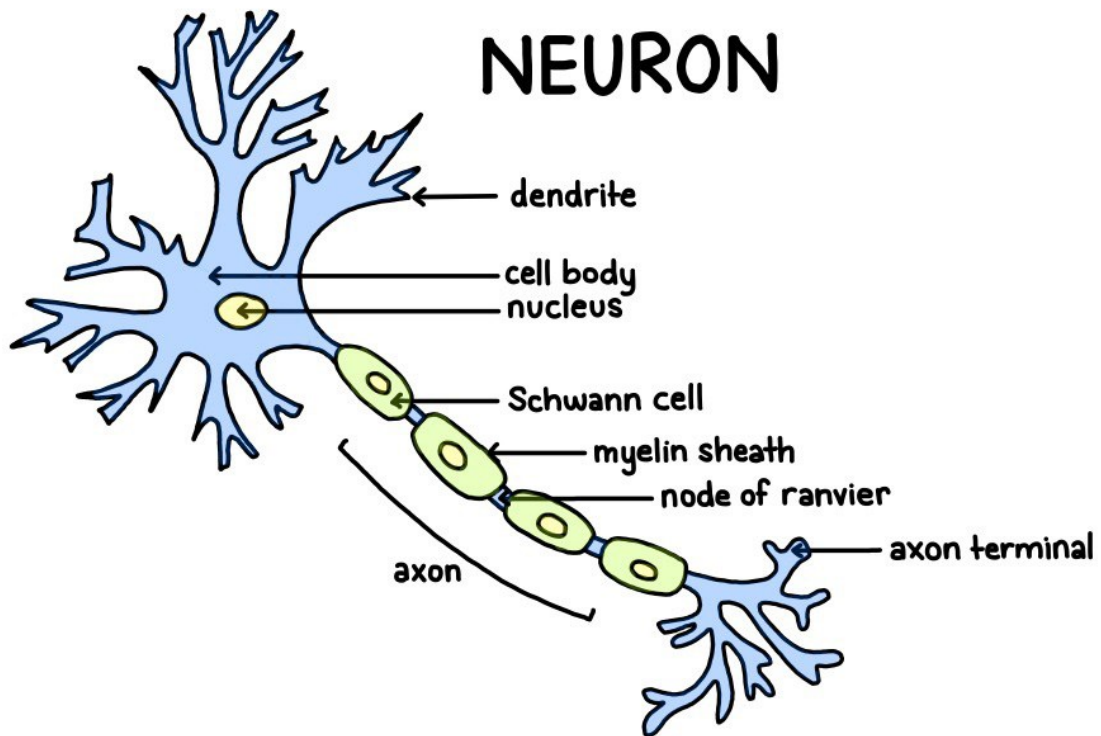


Figure 3 Anatomy of a Neuron

1. Dendriten

Dendriten sind der wurzelförmige Teil des Neurons, der normalerweise kürzer und zahlreicher ist als die Axone. Ihr Zweck ist es, Informationen von anderen Neuronen zu empfangen und elektrische Signale an den Zellkörper zu übermitteln.

Dendriten sind mit Synapsen ausgekleidet, die es ihnen ermöglichen, Signale von anderen Neuronen zu empfangen. Einige Neuronen haben kurze Dendriten, während andere längere Dendriten haben.

Neuronen im Zentralnervensystem sind lang und haben komplexe Verzweigungen, die es ihnen ermöglichen, Signale von vielen anderen Neuronen zu empfangen.

Zum Beispiel haben Zellen namens Purkinje-Zellen im Kleinhirn hochentwickelte Dendriten, um Signale von Tausenden anderer Zellen zu empfangen (Guy-Evans, 2021).

2. Soma (Zellkörper)

Das Soma oder der Zellkörper ist im Wesentlichen der Kern des Neurons. Die Funktion des Somas besteht darin, die Zelle zu schützen und die effiziente Funktion des Neurons sicherzustellen.

Das Soma ist von einer Membran umgeben, die es schützt, ihm aber auch ermöglicht, mit seiner unmittelbaren Umgebung zu interagieren.

Das Soma enthält einen Zellkern, der genetische Informationen produziert und die Synthese von Proteinen steuert. Diese Proteine sind für die Funktion anderer Teile des Neurons von entscheidender Bedeutung (Guy-Evans, 2021).

3. Axon

Das Axon, auch Nervenfasern genannt, ist eine schwanzartige Struktur des Neurons, die den Zellkörper an einer als Axonstigma bezeichneten Verbindungsstelle verbindet.

Die Funktion des Axons besteht darin, Signale vom Zellkörper zu Endknöpfen zu leiten, um elektrische Signale an andere Neuronen zu übertragen.

Die meisten Neuronen haben nur ein Axon, dessen Größe von 0,1 Millimeter bis über 3 Fuß reichen kann. Einige Axone sind mit einer Fettschicht namens Myelin bedeckt, die das Axon isoliert und die Signalübertragung beschleunigt.

Axone sind lange Nervenfortsätze, die sich verzweigen können, um Signale in viele Bereiche zu übertragen, bevor sie an Kreuzungen, den sogenannten Synapsen, enden (Guy-Evans, 2021).

4. Myelinscheide

Die Myelinscheide ist eine Schicht aus Fettmaterial, die die Axone von Neuronen bedeckt. Sein Zweck ist es, eine Nervenzelle von einer anderen zu isolieren und dadurch zu verhindern, dass

der Impuls eines Neurons den Impuls des anderen stört. Die zweite Funktion der Myelinscheide besteht darin, die Übertragung von Nervenimpulsen entlang des Axons zu beschleunigen.

Eingehüllt in Zellen, die als Gliazellen bekannt sind (auch bekannt als Oligodendrozyten und Schwann-Zellen), bilden Axone die Myelinscheide.

Die Myelinscheide, die diese Neuronen umgibt, hat den Zweck, das Axon zu isolieren und zu schützen. Aufgrund dieses Schutzes ist die Leitungsgeschwindigkeit zu anderen Neuronen viel schneller als zu marklosen Neuronen.

Die Myelinscheide besteht aus segmentierten Räumen, die Ranvier-Knoten genannt werden. Elektrische Signale können zwischen Ranvier-Knoten springen und dabei helfen, die Übertragung von Signalen zu beschleunigen (Guy-Evans, 2021).

5. Axon Terminals

Axonterminals (Terminalknöpfe), die sich am Ende des Neurons befinden, sind für die Übertragung von Signalen an andere Neuronen verantwortlich.

Am Ende der Terminaltaste befindet sich ein Raum, der als Synapse bekannt ist. Endknöpfe halten Gefäße, die Neurotransmitter enthalten.

Neurotransmitter werden von den Endknöpfen an die Synapse abgegeben und dazu verwendet, Signale über die Synapse zu anderen Neuronen zu übertragen. Elektrische Signale werden dabei in chemische Signale umgewandelt.

Es ist dann die Aufgabe der Endknöpfe, überschüssige Neurotransmitter abzurufen, die nicht an das nächste Neuron übertragen werden (Guy-Evans, 2021).

a. Was ist Aktionspotential?

Ein Aktionspotential ist ein schneller Anstieg, gefolgt von einer Abnahme der Spannung oder des Membranpotentials über eine Zellmembran mit einem charakteristischen Muster. Es ist ausreichend Strom erforderlich, um eine Spannungsantwort über eine Zellmembran zu initiieren; Ein Aktionspotential wird nicht ausgelöst, wenn der Strom nicht ausreicht, um die Membran auf den Schwellenwert zu depolarisieren. Beispiele für Zellen, die durch Aktionspotentiale signalisieren, sind

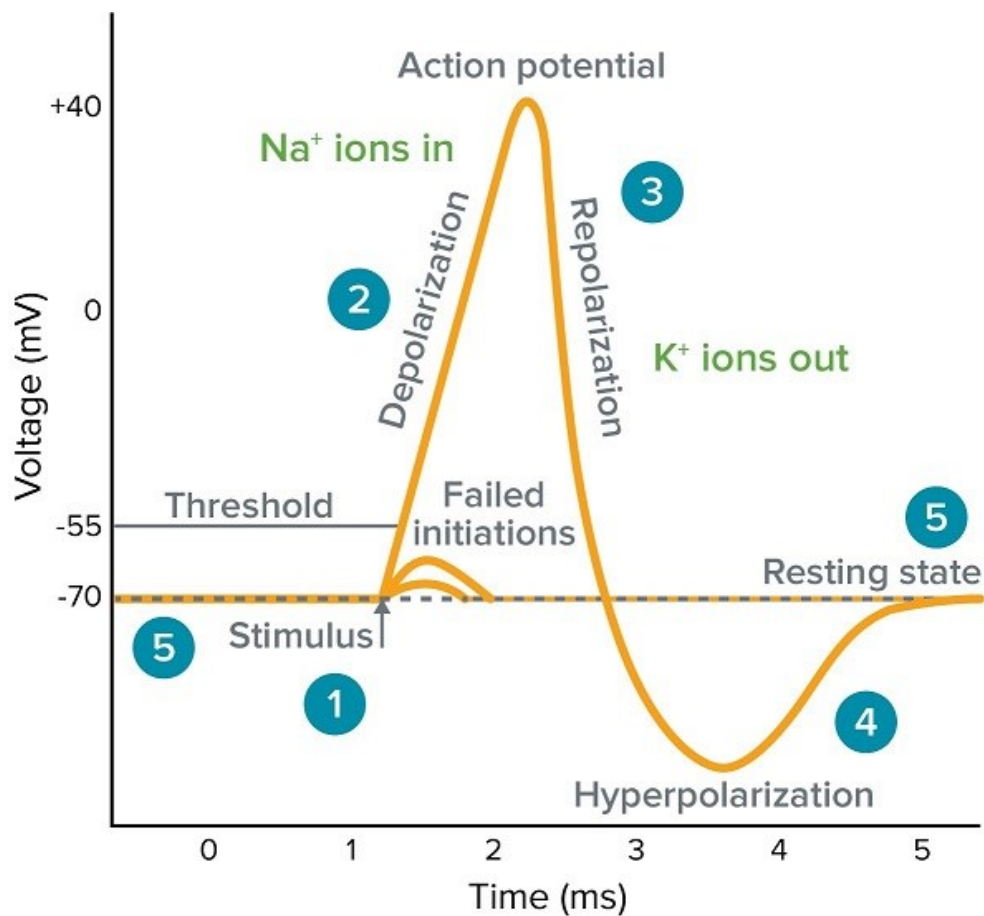


Figure 4 The Action Potential

1. Der Stimulus löst eine schnelle Änderung der Spannung oder des Aktionspotentials aus. Im Patch-Clamp-Modus muss ausreichend Strom an die Zelle angelegt werden, um die Spannung über die Schwellenspannung anzuheben, um die Membrandepolarisation einzuleiten.
2. Depolarisation wird durch einen schnellen Anstieg des Membranpotentials von Natriumkanälen in der Zellmembran verursacht, was zu einem großen Einstrom von Natriumionen führt.
3. Die Membranrepolarisation resultiert aus einer schnellen Natriumkanalinaktivierung und einem großen Einstrom von Kaliumionen aus aktivierten Kaliumkanälen.
4. Hyperpolarisation ist ein niedriges Membranpotential, das aus dem Abfluss von Kaliumionen und dem Schließen von Kaliumkanälen resultiert.

5. Der Ruhezustand ist, wenn das Membranpotential auf die Ruhespannung zurückkehrt, die vor dem Auftreten des Stimulus auftrat(*What Is an Action Potential, Action Potential Chart, Membrane Potential | Molecular Devices*, n.d.).

IV. BCI

Brain-Machine-Interface (BCI) ist ein Hard- und Softwaresystem, das es dem Menschen ermöglicht, ohne Nerven oder Muskeln mit seiner Umwelt zu kommunizieren, indem es Steuersignale verwendet, die durch verschiedene Methoden aus Gehirnaktivitäten gewonnen werden. BCI verwendet einen Kommunikationskanal, um Computern und Hilfsgeräten mitzuteilen, womit Menschen interagieren möchten. Der Wegfall von Muskel- und Nervenbelastung ist ein großer Vorteil für Patienten, die keine motorischen Aktivitäten ausführen können. Daher ist die Verwendung von BCI effektiv, da sie die Lebensqualität dieser Patienten verbessert und die Pflegekosten senkt.

BCI ist ein künstliches Intelligenzsystem, das aus verschiedenen Komponenten besteht, die sinnvolle Daten in Gehirnsignalen erkennen und interpretieren können. Es besteht aus fünf Komponenten wie Signalerfassung, Signalvorverarbeitung und Signalverbesserung, Merkmalsextraktion, Klassifizierung und Steuerschnittstelle(Fernando Nicolas-Alonso & Gomez-Gil, 2012).

Es werden invasive, semi-invasive und nicht-invasive Messtechniken verwendet, um Daten zu erhalten, die in BCI-Systemen verwendet werden sollen.

A. Invasive Methoden

Invasive Methoden sind Methoden, die einen chirurgischen Eingriff erfordern und schwierig anzuwenden sind. Elektrokortikographie (ECoG)- und Local Field Potentials (LFP)-Methoden werden als invasive Methoden in BCI-Systemen verwendet(BEKİRYAZICI, 2020).

Bei der LFP-Technik hingegen wird die Potentialdifferenz gemessen, die durch den elektrischen Strom entsteht, der über benachbarte Neuronen im Nervengewebe fließt.

1. ECoG

Bei der ECoG-Methode wird direkt über der Großhirnrinde gemessen. Daher können die erhaltenen Signale eine hohe Spannung erzeugen. Dadurch wird sichergestellt, dass die Verzerrungen im Signal geringer sind. Dies ermöglicht eine einfachere Interpretation von Signalen und einen geringeren Bedarf an Vorverarbeitungsprozessen (BEKİRYAZICI, 2020).

Im Vergleich zum EEG liefert ECoG ein Signal mit höherer Amplitude. Außerdem ist es einfacher zu implementieren und liefert höhere räumliche und zeitliche Informationen. Das bei dieser Methode eingesetzte Implantat kann jedoch Gesundheitsgefahren verursachen. Es wird zur Messung von Alpha-, Beta- und Gammawellen verwendet, die durch motorische Aktivitäten beim Menschen verursacht werden (Fernando Nicolas-Alonso & Gomez-Gil, 2012).

B. Nicht-invasive Methoden

Als nicht-invasive Messverfahren kommen bei BCI-Systemen die Elektroenzephalographie (EEG), die funktionelle Magnetresonanztomographie (fMRT) und die Magnetoenzephalographie (MEG) zum Einsatz. Bei der MEG-Methode wird das Gehirn kartiert, indem das durch Gehirnaktivitäten verursachte Magnetfeld gemessen wird. (BEKİRYAZICI, 2020).

Bei der EEG-Methode messen trockene oder feuchte Elektroden auf dem Schädel die Gehirnaktivität. Obwohl dieses Verfahren einfach anzuwenden ist, treten Verzerrungen in den Signalen auf, da die Signalspannungen zu den Sensoren niedrig sind. In diesem Fall müssen Vorkehrungen am Signal getroffen werden und es wird versucht, die Geräusche zu entfernen.

Die aus nicht-invasiven Methoden gewonnenen Daten können aussagekräftig genug gemacht werden. Darüber hinaus werden sie häufig bevorzugt, da sie keine chirurgischen Eingriffe erfordern.

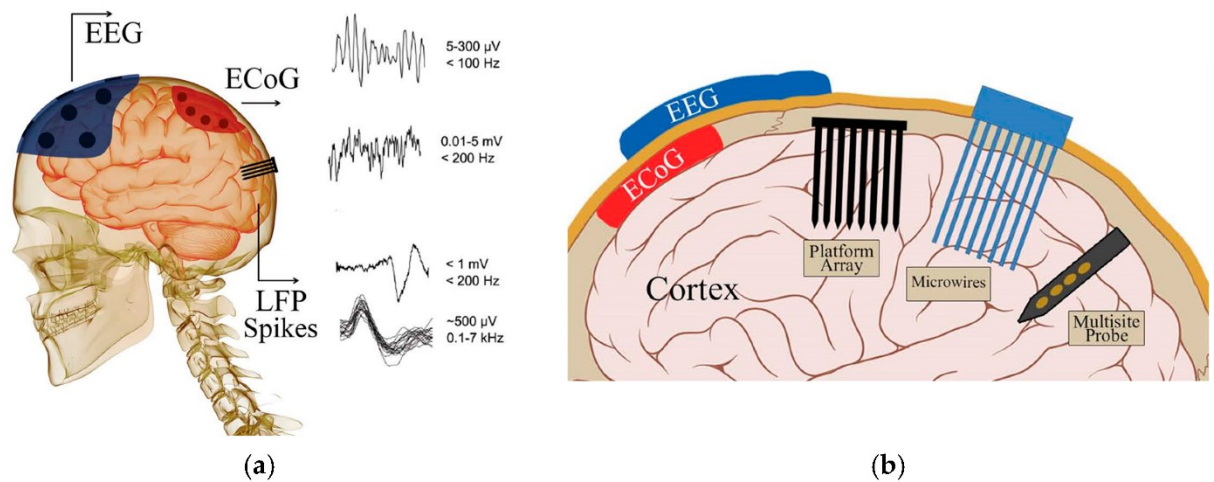


Figure 5 BCI measurement types.

1. Magnetoencephalography (MEG)

Die Magnetoenzephalographie ist ein nicht-invasives Messverfahren, mit dem das durch Gehirnaktivitäten erzeugte Magnetfeld durch magnetische Induktion gemessen wird. MEG misst das Magnetfeld, das innerhalb der Zelle in den Dendriten über dem Schädel erzeugt wird. Der Vorteil des MEG-Systems besteht darin, dass es von den Geräuschen am Schädel nicht beeinflusst wird. Magnetfelder werden von supraleitenden Quantengeräten gemessen, die extrem empfindlich auf Rauschen von neuronalen Aktivitäten reagieren. MEG-Geräte müssen in einem magnetfeldgeschützten Bereich verwendet werden. Supraleiter hingegen benötigen sehr tiefe Temperaturen. Ihr Einsatz ist teuer. Es liefert Daten mit höherer Auflösung als das EEG-Verfahren in einer großen zeitlichen und räumlichen Weise. Es trainiert das BCI-System in kürzerer Zeit und sorgt für eine zuverlässige Kommunikation mit dem System. Es kann auch leicht aktive Bereiche im Gehirn identifizieren (Fernando Nicolas-Alonso & Gomez-Gil, 2012).

2. Functional Magnetic Resonance Imaging (fMRI)

Dieses System misst mithilfe von elektromagnetischen Feldern Änderungen des Blutvolumens und des Sauerstoffgehalts der regionalen Großhirnrinde nach neuronalen Aktivitäten. Der Vorteil dieser Methode ist, dass sie die Regionen auf dem Gehirn gut trennen und die Aktivität in den aktiven Regionen gut messen kann. Die zeitliche Auflösung ist jedoch gering. Es gelingt nicht, schnell mit dem BCI-System zu kommunizieren. Das System kann nicht in Echtzeit arbeiten und Ergebnisse können erst nach Stunden aus den Daten gewonnen werden. Da die erhaltenen Daten aus Blutdruck-

und Sauerstoffmengenmessungen statt aus direkten neuronalen Aktivitäten stammen, geben sie nicht immer gesunde Informationen. Es ist eine alte Methode und wird nicht bevorzugt, weil sie teuer ist(Fernando Nicolas-Alonso & Gomez-Gil, 2012).

3. Electroencephalography (EEG)

Neurale Aktivitäten treten mit den elektrischen Aktivitäten von Nervenzellen im Gehirn auf. Die Elektroenzephalographie ist ein Messverfahren zur Messung und Aufzeichnung dieser elektrischen Aktivitäten mit auf den Schädel aufgesetzten Ag-AgCl-Elektroden(BEKİRYAZICI, 2020).

Für Elektroenzephalographiedaten wird die Zeit- und Frequenzreihenanalyse angewendet, die allgemein in der Datenanalyse verwendet wird. Die durchgeführten Messungen werden in einem bestimmten Zeitraum aufgezeichnet. Normalerweise werden EEG-Daten von Sensoren erhalten, die auf der Haut angebracht sind, aber bei der IEEG-Methode werden Signale aus dem Gehirn empfangen(Hosseini et al., 2021).

Bisher wurden EEG-Signale vor allem bei Hirnerkrankungen wie Epilepsie, Schlaganfall und Schlafstörungen eingesetzt. Heutzutage wird untersucht, um Objekte ohne den Einsatz von Muskeln zu steuern, indem Gehirnwellen zusammen mit maschinellen Lernalgorithmen verwendet werden.EEG-Gehirnwellenformen werden ausgedrückt, indem sie in bestimmte Frequenzbereiche unterteilt werden. Diese Bereiche werden durch die Namen Delta, Theta, Alpha, Beta und Gamma ausgedrückt. Jeder Bereich hat seine eigenen Eigenschaften. Die geringen Kosten des EEG-Systems im Vergleich zu anderen Methoden sind ein wichtiger Vorteil. Ihr Nachteil ist die geringe räumliche Auflösung und das hohe Signalrauschen(Hosseini et al., 2021).

V. WAS IST EEG UND WIE FUNKTIONIERT ES?

Die Elektroenzephalographie oder EEG ist die physiologische Methode der Wahl, um die elektrische Aktivität aufzuzeichnen, die vom Gehirn über Elektroden erzeugt wird, die auf der Kopfhautoberfläche platziert werden. Für eine schnellere Anwendung werden die Elektroden in elastischen Kappen ähnlich wie Badekappen montiert, sodass die Datenerfassung von überall an identischen Kopfhautpositionen möglich ist.

Da die an den Elektroden gemessenen Spannungsschwankungen sehr klein sind, werden die aufgezeichneten Daten digitalisiert und an einen Verstärker gesendet. Die verstärkten Daten können dann als Folge von Spannungswerten angezeigt werden.

Preisunterschiede bei EEG-Systemen sind typischerweise auf die Anzahl der Elektroden, die Qualität der Digitalisierung, die Qualität des Verstärkers und die Anzahl der Schnappschüsse zurückzuführen, die das Gerät pro Sekunde aufnehmen kann (das ist die Abtastrate in Hz).

Das EEG ist eines der schnellsten verfügbaren Bildgebungsverfahren, da es häufig eine hohe Abtastrate aufweist. Vor 100 Jahren wurde der zeitliche Verlauf eines EEGs auf Papier aufgezeichnet – heute werden die Daten (zum Glück) digital als kontinuierlicher Spannungsverlauf auf einem Bildschirm dargestellt (Farnsworth, 2021)

A. EEG im Vergleich zu anderen bildgebenden Verfahren des Gehirns

- Die Magnetoenzephalographie (MEG) zeichnet die Magnetfelder auf, die durch neurale Aktivität erzeugt werden. Wie das EEG hat das MEG eine hervorragende Zeitauflösung und wird oft als viel besser als das EEG angesehen, um tiefere neuronale Aktivitäten zu erfassen. MEG-Scanner sind groß, stationär und teuer. Sie erfordern umfangreiche technische Wartungs- und Schulungsressourcen.
- Die funktionelle Magnetresonanztomographie (fMRI) misst Veränderungen im Blutfluss, die mit Veränderungen in der neuronalen Aktivität einhergehen. Erhöhtes neuronales Feuern erfordert Sauerstoff, der durch Blut zugeführt wird, und die magnetischen Eigenschaften von sauerstoffreichem Blut unterscheiden sich von denen von nicht sauerstoffreichem Blut. Diese Eigenschaft wird durch fMRT als Verzerrung des von Wasserstoffprotonen erzeugten Magnetfelds gemessen. fMRI hat eine ausgezeichnete räumliche Auflösung, während gleichzeitig die Zeitauflösung des EEG fehlt.
- Die Positronen-Emissions-Tomographie (PET) ist ein nukleares Bildgebungsverfahren, das auf Gammastrahlung basiert, die durch zerfallende Radionuklide verursacht wird, die in den Körper des Probanden eingeführt werden. Mit PET können Sie die Stoffwechselaktivität (z. B. Glukosestoffwechsel) von Neuronen während der kognitiven Aktivität überwachen. Während PET-Scans viel robuster gegenüber Bewegungsartefakten sind, fehlt ihnen die hohe Zeitauflösung von EEG-Aufzeichnungen (Farnsworth, 2019)

B. EEG Frequenzbereiche / Frequenzbänder

b. Delta (0.5 – 4 Hz)

Die Spannungsamplituden von Deltawellen liegen im Bereich von 20–400 μV und ihre Frequenz liegt im Bereich von 0,5 Hz. Diese Wellenform ist häufiger in Zuständen von ruhendem klarem Bewusstsein, tiefer Meditation, Träumen und Koma(BEKİRYAZICI, 2020).

Deltawellen werden untersucht, um die Schlaftiefe zu verstehen. Je stärker der Rhythmus der Deltawellen, desto tiefer der Schlaf. Es wurde auch beobachtet, dass das Vorhandensein von Deltawellen auch mit zunehmender Konzentration verbunden ist(Farnsworth, 2021).

c. Theta (4 – 8 Hz)

Theta-Wellen haben Spannungsamplituden von 5–100 μV und Frequenzen im Bereich von 4–8 Hz. Es wird im Ruhezustand von klarem Bewusstsein, tiefer Meditation, Träumen, Stress beobachtet(BEKİRYAZICI, 2020).

Tritt bei Schläfrigkeit auf, die in den frühen Schlafphasen wie N1 und N2 auftritt(Buyukgoze, n.d.).

Theta ist mit einer Vielzahl von kognitiven Prozessen verbunden, wie z. B. der Kodierung und dem Abruf von Erinnerungen sowie der kognitiven Arbeitsbelastung. Wenn der Geist mit schwierigen Aufgaben konfrontiert wird, werden Theta-Wellen stärker ausgeprägt. Theta wird auch mit erhöhter Ermüdung in Verbindung gebracht.

d. Alpha (8 – 13 Hz)

Die Spannungsamplituden der Alphawellen liegen im Bereich von 2-10 μV und die Frequenz im Bereich von 8-13 Hz. Alphawellen sind normalerweise sinusförmig geformt. Ihre Messungen werden normalerweise durch den Hinterhauptslappen durchgeführt. Diese Wellenform wird oft mit motorischen Bewegungen in Verbindung gebracht. Es offenbart sich, wenn es wach ist, in einem Zustand völliger geistiger und körperlicher Ruhe(BEKİRYAZICI, 2020).

Sie sind mit dem Schließen der Augen sowie mit Hemmung und Aufmerksamkeit verbunden.

e. Beta (13 – 30 Hz)

Die Spannungsamplituden von Betawellen liegen im Bereich von 1-5 μ V und der Frequenzbereich im Bereich von 13-30 Hz. Die Messung dieser Wellenform erfolgt normalerweise durch die parietalen und frontalen Lappen. Sie werden beobachtet, wenn das zentrale Nervensystem aktiv arbeitet, wenn Aufmerksamkeit und Konzentration erhöht sind (BEKİRYAZICI, 2020).

Die Beta-Wellenform ist die am häufigsten beobachtete Form bei normalen Erwachsenen und Kindern. Bei Schläfrigkeit erhöht der N1-Schlaf sofort die Amplitude, und dann nimmt seine Amplitude in den Phasen N2 und N3 ab (Buyukgoze, n.d.).

In den motorischen Regionen werden die Beta-Frequenzen verstärkt, wenn die Bewegungen eines Körperteils geplant oder ausgeführt werden. Dieser Anstieg der Beta macht sich auch bemerkbar, wenn die Körperbewegungen anderer Menschen beobachtet werden. Unser Gehirn ahmt die Bewegungen nach, die es auf der anderen Seite sieht, und Beta-Wellen koordinieren dies (Farnsworth, 2019).

f. Gamma (>30 Hz)

Der Frequenzbereich der Gammawellenform liegt im Bereich von 30-50 Hz und ihre Amplitude ist sehr niedrig. Die Messung erfolgt über den Frontallappen (BEKİRYAZICI, 2020).

Einige Forscher argumentieren, dass Gammawellen einen sorgfältigen Fokus widerspiegeln und als Trägerfrequenzen dienen, um den Datenaustausch zwischen Gehirnregionen zu erleichtern. Einige andere Forscher assoziieren Gammawellen mit schnellen Augenbewegungen, die als "Mikrosakkaden" bezeichnet werden und als integrale Bestandteile für die sensorische Verarbeitung und Informationsgewinnung gelten (Farnsworth, 2021).

Darüber hinaus können Gammawellen eine Frequenz von bis zu 100 Hz haben, aber auf diesen Ebenen kann keine signifikante Aktivität beobachtet werden. Filter, die auf Signale angewendet werden, verhindern auch die Signalübertragung auf diesen Ebenen..

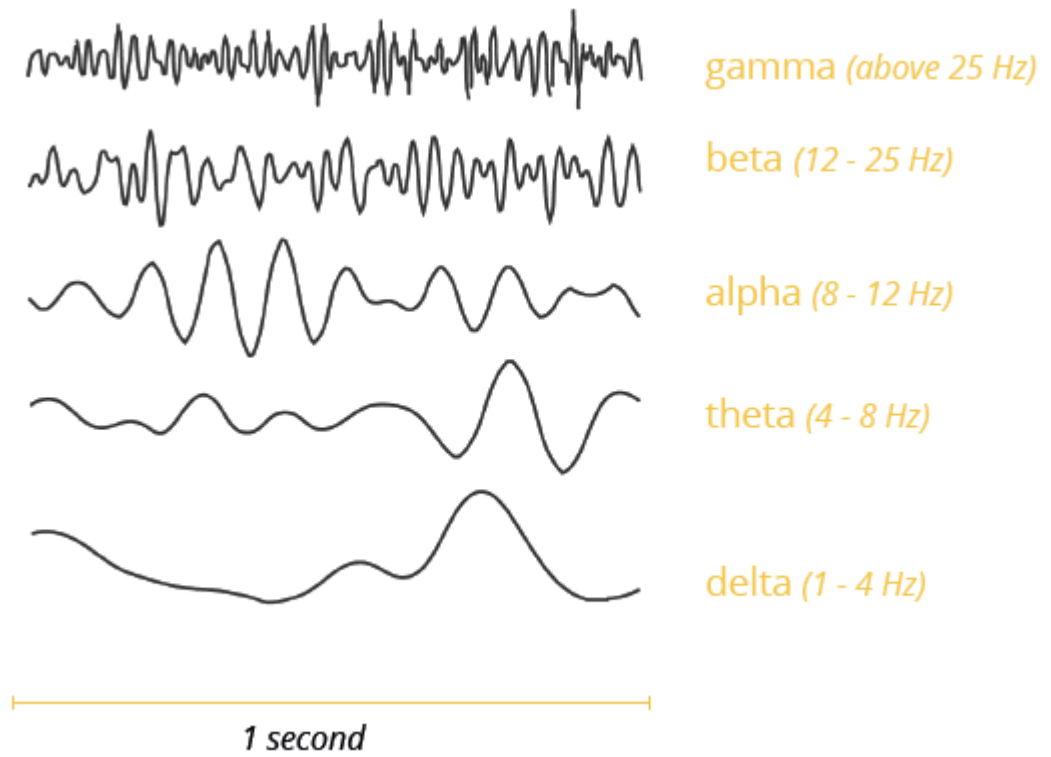


Figure 6 Brainwave Frequencies

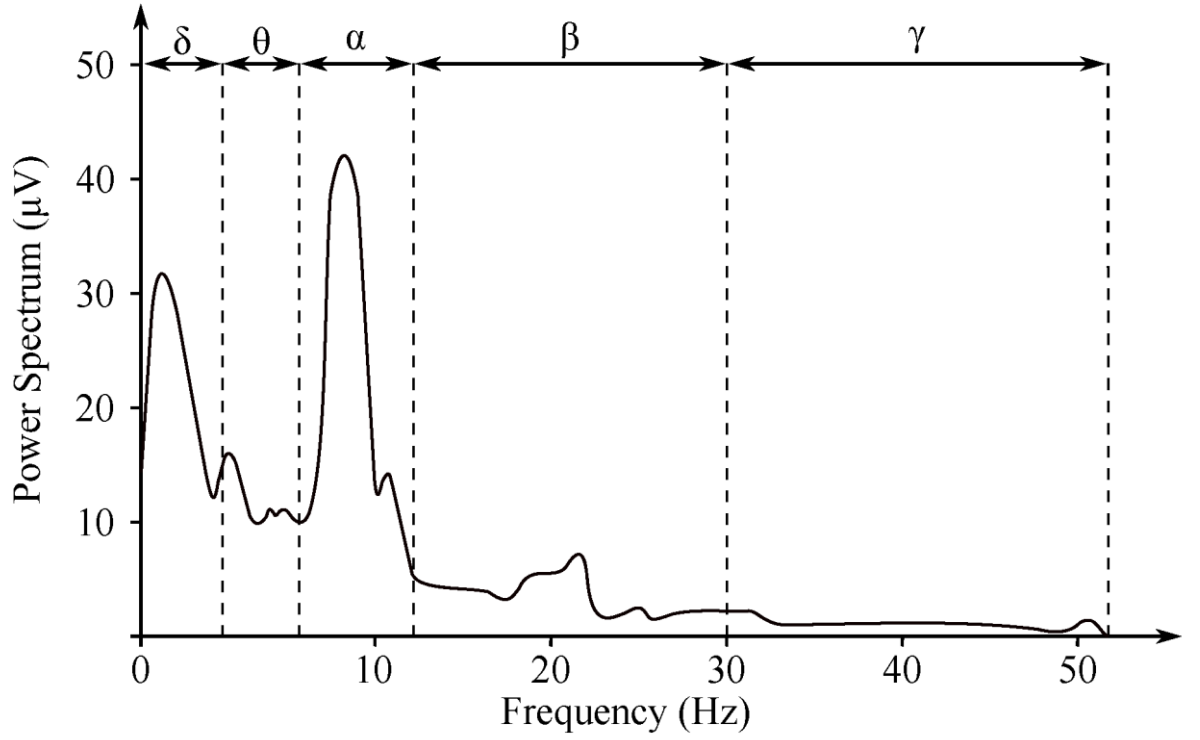
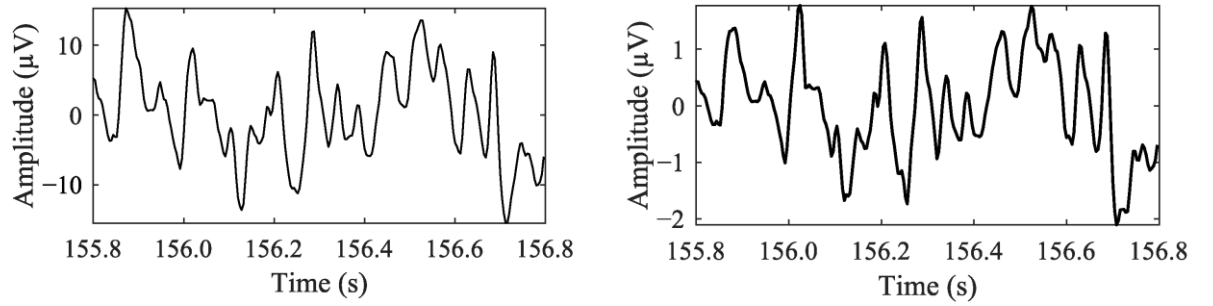
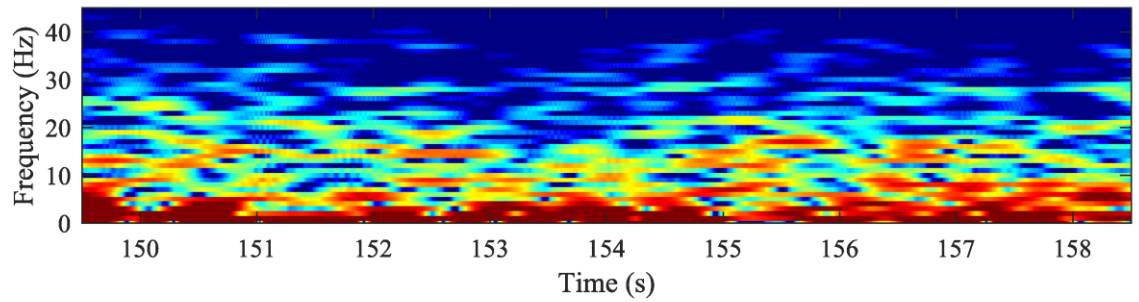


Figure 7 EEG frequency spectrum and its frequency bands



(a) Sample (1 s).

(b) Sample (1 s)—normalized.



(c) Sample (10 s)—spectrogram.

Figure 8 Sample EEG data recorded from 'F3-C3' location.

C. EEG-Elektroden

EEG-Systeme verwenden Elektroden, um bioelektrische Potentiale zu messen, die als Folge neuronaler Aktivität auftreten. Diese Elektroden werden verwendet, um Markierungen zu unterscheiden und einen bestimmten Bereich zu stimulieren. Elektroden werden verwendet, um Elektronenstrom in Ionenstrom oder Ionenstrom in elektrischen Strom umzuwandeln.

Elektroden enthalten Elektronen, die Ströme tragen. Im Elektrolytteil tragen Anionen und Kationen die Ströme. Die Ionen-Elektron-Umwandlung tritt auf, wenn die Elektrodenatome Elektronen verlieren und zu Kationen in der Elektrolytlösung werden und sich die Kationen in der Elektrolytlösung auf der Oberfläche ansammeln, indem Elektronen aus der Elektrode entnommen werden. Solange das System nicht unter Strom steht, hält das System weiterhin sein eigenes Gleichgewicht.

Wenn sich Ladung an der Grenzfläche ansammelt, ändert sich das elektrische Potential des Elektrolyten. Die Potentialdifferenz unterscheidet sich je nach Material der Elektrode sowie der Ionenaktivität. Dieser Potentialunterschied wird als Quasi-Zell-Potential bezeichnet.

Elektroden werden in polarisierte und unpolarisierte Elektroden unterteilt. Bei vollpolarisierten Elektroden ist der Strömungsfluss durch die Elektrolytelektrode begrenzt. Wenn Strom durch die polarisierten Elektroden fließt, findet im Elektrolyt-Elektroden-Bereich keine Ladungsübertragung statt. Große Potentialdifferenzen sind erforderlich, um einen aussagekräftigen Strom in polarisierten Elektroden zu leiten. Elektroden, durch die Strom bidirektional fließen kann, ohne die Grenzfläche möglicherweise zu verringern, werden als reversible Elektroden bezeichnet. In reversiblen Elektroden kann Strom frei aus dem Elektroden-Elektrolyt-Bereich fließen und benötigt keine Energie.

Ag-AgCl-Elektroden werden in EEG-Systemen bevorzugt. Die Silberelektrode ist mit AgCl beschichtet, um eine geräuschlosere Messung zu ermöglichen. Diese Elektroden können trocken oder geliert verwendet werden. Wenn das Gel verwendet wird, nimmt der Oberflächenwiderstand (Impedanz) zum Gesicht ab. So werden stärkere Signale erhalten. Bei Messungen sollte die Kontaktimpedanz zwischen Elektrode und Haut auf ein Minimum reduziert werden. Moderne Studien verwenden bioelektrische Verstärker mit hohem Eingangswiderstand, um diese Impedanz zu reduzieren (BEKİRYAZICI, 2020).

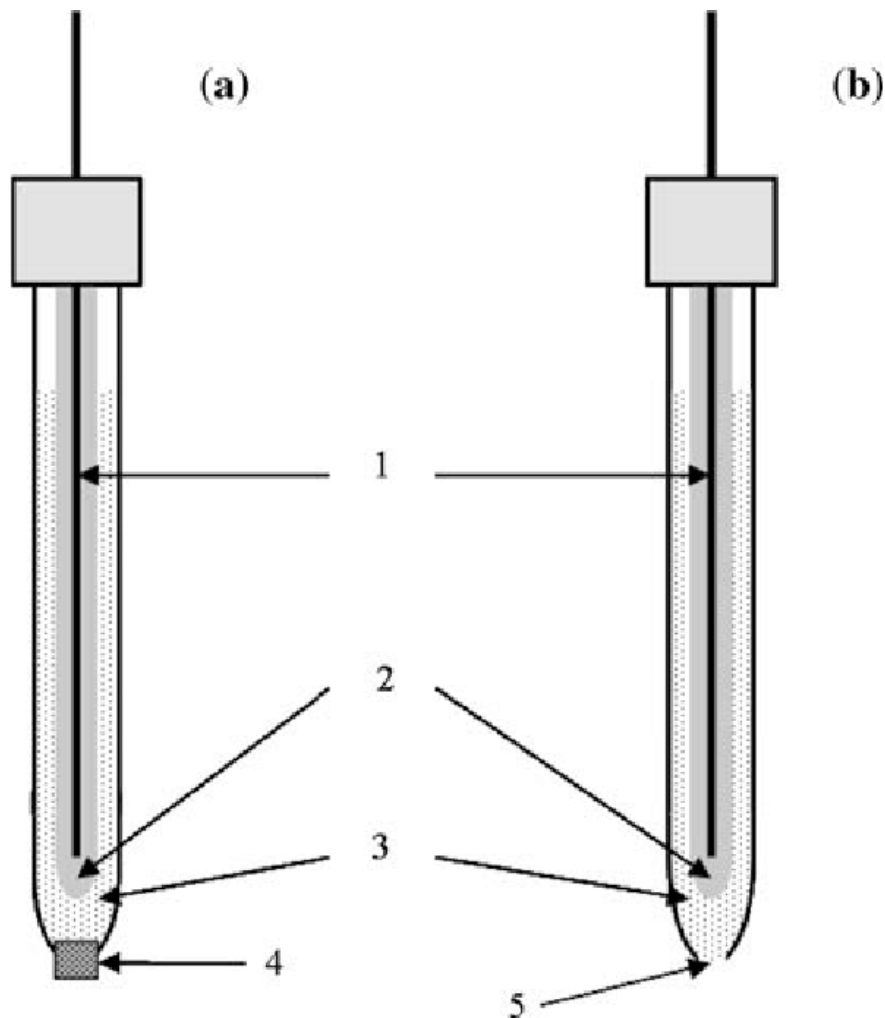


Figure 9 Scheme of the Ag/AgCl electrodes (a) typical Ag/AgCl electrode with a Vycor tip and (b) Ag/AgCl electrode with a tiny perforation to accelerate the diffusion of the internal electrolyte into the test solution: 1, Ag wire; 2, AgCl layer; 3, KCl solution;

1. Elektrodenarrays und Platzierung

Die gebräuchlichsten Systeme zur Identifizierung und Benennung von Elektrodenpositionen entlang der Kopfhaut wurden von der American Encephalographic Society (1994) und Oostenveld & Praamstra (2001) bereitgestellt. Zum Einsatz kommen Systeme wie 10-10 oder 10-5, insbesondere das 10-20 System. Im 10-20-System werden die Elektroden an 10% und 20% Punkten entlang der Längen- und Breitengradlinien platziert.

Zu den Highlights des 10-20-Systems gehören:

Nasion (Nz): Depression zwischen den Augen im oberen Teil der Nase.

- Ion (Narbe): Schwellung am Hinterkopf.

Linke und rechte präaurikuläre Flecken: Wenn Sie Ihren Mund öffnen und schließen, können Sie diese Vertiefungen mit Ihren Fingern direkt vor Ihren Ohren spüren.

Die vertikale Linie, die Nasion (vorne) und inion (hinten) verbindet, und die horizontale Linie, die den linken und rechten vorderen Ohrpunkt verbindet, sind nun in zehn gleiche Abschnitte unterteilt.

Ebenso wird der Äquator in 10% und 20% unterteilt (Farnsworth, 2021).

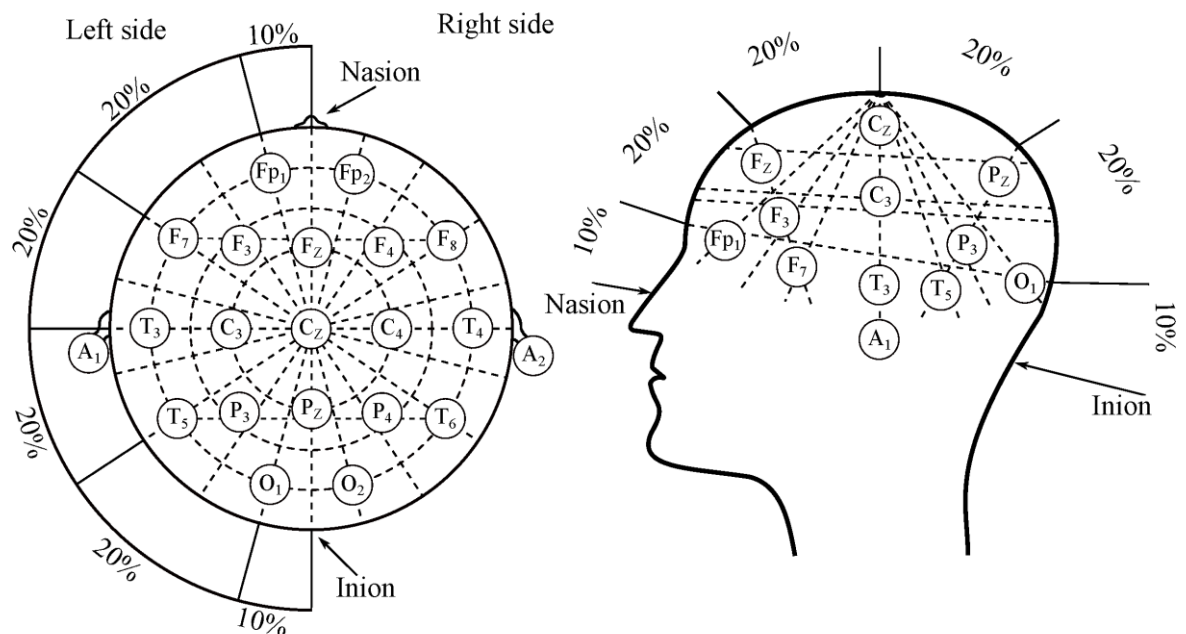


Figure 10 The EEG electrodes placement system "10-20"

D. Steuersignaltypen in BCI-Systemen

Das Ziel eines BCI-Systems ist es, die neuronale Aktivität in der Großhirnrinde zu überwachen und zu interpretieren. Das Gehirn enthält eine große Anzahl von Fällen im Zusammenhang mit kognitiven Aktivitäten gleichzeitig. Viele dieser Phänomene im Gehirn sind noch ungeklärt. Einige Gehirnsignale wurden jedoch verstanden und diese Signale werden verwendet, um die Gehirnaktivität zu interpretieren. Diese Signale werden in BCI-Systemen als mögliche Steuersignale bezeichnet.

Mit den durchgeführten Studien wurden große Signalgruppen identifiziert, die als Steuersignale in BCI-Systemen verwendet werden können. Visuelle evozierte Potentiale, langsame kortikale Potentiale, P300-evozierte Potentiale und sensomotorische Rhythmen sind Gruppen, die zur Beschreibung von Steuersignalen verwendet werden (Fernando Nicolas-Alonso & Gomez-Gil, 2012).

Tablo 1 Summary of control signals

1	Signa	Psiological phenoma	Anzahl der Auswahlmöglichkeiten	Ausbildung	Informationsübertragungsrate
	VEP	Signalmodulationen des Gehirns im visuellen Kortex	Hoch	Nein	60–100 bits/min
	SCP	Langsame Spannungsverschiebung in den Gehirnsignalen	Niedrig (2 oder 4, sehr schwierig)	Ja (Begriffsklärung)	5–12 bits/min
	P300	Positive Spitzen durch seltene Reize	Hoch	Nein	20–25 bits/min
	Sensomotorische Rhythmen	Modulationen in sensomotorischen Rhythmen, die mit motorischen Aktivitäten synchronisiert sind	Niedrig (2, 3, 4, 5)	Ja	3–35 bits/min

1. Visual Evoked Potentials (VEPs)

VEPs sind Modulationen der Gehirnaktivität, die im visuellen Kortex auftreten, nachdem sie einen visuellen Reiz erhalten haben. Die Amplitude der VEPs nimmt stark zu, wenn sich der Reiz dem zentralen Sehraum nähert. Daher sind diese Modulationen relativ einfach zu erkennen. VEPs können nach drei verschiedenen Kriterien klassifiziert werden.

a. Nach der Morphologie optischer Reize

Nach den Kriterien der Morphologie optischer Reize können VEPs mit Hilfe von Blitzstimulation oder grafischen Mustern wie Schachbrettnetz, Tür und zufälliger Punktkarte erzeugt werden.

TVEPs können mit jeder Veränderung im Gesichtsfeld aufgerufen werden. Die am häufigsten verwendeten TVEPs sind:

- Blitz-TVEPs mit blinkenden Lichtern
- TVEPs, die durch das plötzliche Auftreten eines Musters auf einem verstreuten Hintergrund entstehen
- TVEPs mit umgekehrtem Muster, die durch Umkehrung der Phase eines Musters verursacht werden

Tatsächlich werden TVEPs normalerweise nicht für BCI verwendet (Fernando Nicolas-Alonso & Gomez-Gil, 2012).

b. Nach Häufigkeit der visuellen Warnung

Nach Häufigkeit können VEPs auch als Transiente VEPs (TVEPs) und Steady State VEPs (SSVEPs) klassifiziert werden. TVEPs treten auf, wenn die visuelle Stimulationsfrequenz unter 6 Hz liegt, während SSVEPs als Reaktion auf höherfrequente Reize auftreten. SSVEP-basierte BCIs ermöglichen es Benutzern, ein Ziel mit einem Auge auszuwählen. Der Benutzer richtet seine Aufmerksamkeit visuell auf ein Ziel und definiert das Ziel durch BCI, SSVEP-Feature-Analyse (Fernando Nicolas-Alonso & Gomez-Gil, 2012).

SSVEP-basierte BCIs können je nach verwendeter spezifischer Reizsequenzmodulation in drei Kategorien eingeteilt werden.

- Zeitmodulierte VEP (t-VEP) BCIs
- Frequenzmodulierte VEP (f-VEP) BCIs
- Gefälschte zufällige Code-modulierte VEP (c-VEP) BCIs

Tablo 2 Features of VEP modulations: t-VEP, f-VEP and, c-VEP.

VEP modulation	Funktionen
t-VEP	<ul style="list-style-type: none"> • Relativ niedrige Informationsübertragungsrate (<30 Bit/min) • Synchronsignal ist notwendig • Keine Benutzerschulung erforderlich
f-VEP	<ul style="list-style-type: none"> • Hohe Informationsübertragungsrate (30–60 Bit/min) • Einfache Systemkonfiguration • Keine Benutzerschulung erforderlich • Besser geeignet für Anwendungen mit wenigen Optionen
c-VEP	<ul style="list-style-type: none"> • Sehr hohe Informationsübertragungsrate (>100 Bit/min) • Synchronsignal ist notwendig • Anwenderschulung erforderlich • Besser geeignet für die Anwendung mit vielen Optionen

c. Mit Feldstimulation

Schließlich können VEPs nach dem Feldanregungskriterium in All-Area-VEPs, Halbfeld-VEPs und Teilflächen-VEPs unterteilt werden, abhängig von der Fläche des Stimulus auf dem Bildschirm (Fernando Nicolas-Alonso & Gomez-Gil, 2012).

2. Slow Cortical Potentials (SCPs)

SCPs sind langsame Spannungsverschiebungen, die in EEGs auftreten und von einer Sekunde bis zu einigen Sekunden dauern. SCPs entsprechen dem Sub-1-Hz-Anteil der EEG-Signale. SCPs sind mit Veränderungen des Niveaus der kortikalen Aktivität verbunden. Negative SCPs korrelieren mit erhöhter neuronaler Aktivität, während positive SCPs mit einer Abnahme der Aktivität in Zellen korrelieren. Diese Gehirnsignale können sich selbst regulieren, um externe Geräte über einen BCI zu steuern. SCPs können verwendet werden, um einen Cursor zu bewegen und Ziele auszuwählen, die auf einem Computerbildschirm angezeigt werden.

Das Thought-Translation-Gerät ist ein Werkzeug, das verwendet wird, um den Benutzer zu trainieren, indem audiovisuelle Hinweise an den Benutzer gesendet werden, damit der Benutzer den SCP steuern kann. Das Gedankenübersetzungsgerät enthält typischerweise einen Cursor auf dem Bildschirm, der die Amplitude der SCP-Verschiebungen widerspiegelt. Der Erfolg im SCP-Training hängt von zahlreichen Faktoren ab, wie z. B. der psychischen und physischen Verfassung des Patienten, der Motivation, dem sozialen Kontext oder der Beziehung zwischen Ausbilder und Patient. Für den Betrieb eines BCI-Systems, das auf SCP basiert, ist eine relativ lange Schulung erforderlich, und die Benutzer müssen möglicherweise mehrere Monate lang kontinuierlich implementiert werden (Fernando Nicolas-Alonso & Gomez-Gil, 2012).

3. P300 Evoked Potentials

P300 evozierte Potentiale sind positive Spitzen im EEG, selten aufgrund auditiver, visueller oder somatosensorischer Reize. P300-Reaktionen treten etwa 300 ms auf, nachdem sie sich einem ungewöhnlichen Reiz unter den folgenden häufigen Reizen angeschlossen haben. Als Ergebnis einiger Studien wurde nachgewiesen, dass je geringer die Wahrscheinlichkeit des Stimulus für P300-Antworten ist, desto größer ist die Amplitude des Antwortpeaks.

Für die Verwendung von P300-basierten BCIs ist keine Schulung erforderlich. Wenn sich der Benutzer jedoch an den seltenen Reiz gewöhnt, wird die P300-Amplitude indirekt reduziert und daher kann die Leistung abnehmen. Eine typische BCI-Anwendung, die auf visuellen P300-evozierten Potentialen basiert, besteht aus einer Matrix von Buchstaben, Zahlen oder anderen Symbolen oder Befehlen. Während das EEG-System überwacht wird, blinken die Zeilen oder Spalten dieser Matrix zufällig. Der Benutzer schaut auf das gewünschte Symbol und zählt, wie oft die Zeile oder Spalte mit

der gewünschten Auswahl blinkt. Der P300 wird nur belichtet, wenn die gewünschte Zeile oder Spalte angezeigt wird. Das BCI-System verwendet diese Methode, um das beabsichtigte Symbol zu bestimmen.

Aufgrund des niedrigen Signal-Rausch-Verhältnisses in EEG-Signalen ist es ziemlich schwierig, Zielsymbole mit einem einzigen Versuch zu erkennen. Für jede Auswahl müssen Zeilen oder Spalten mehrmals blinken. Je mehr Blitze auftreten, desto besser wird die Wahl getroffen. Dies verlängert jedoch die Wahlperiode. Obwohl visuelle Reize in den meisten P300-stimulierten potentialbasierten Anwendungen verwendet werden, wurden auditive Reize auch für Menschen mit Sehbehinderungen verwendet.

BCI-Systeme, die mit dem P300 ausgeführt werden, haben wenig Genauigkeit und erfordern mehr Signale, um die Genauigkeit zu erhöhen. Daher ist die Übertragungsgeschwindigkeit des Systems langsam. Die Geschwindigkeit des Systems hängt auch von Parametern wie der Größe und Farbe der Symbole ab.

Die P300-Antwort wird nicht wesentlich davon beeinflusst, ob das Subjekt direkt auf das Ziel schaut, im Gegensatz zur größeren VEP-Antwort, die größer ist, wenn das Ziel markiert ist. Diese Unterscheidung ist für die klinische Praxis wichtig, da Augenbewegungen in der Zielpopulation oft beeinträchtigt sind oder verloren gehen. Wenn die Probanden jedoch auf die gewünschte Marke schauen, wird die Leistung eines P300-basierten BCI deutlich verbessert. Daher hängt die Leistung visueller P300-basierter BCIs nicht nur vom mit dem P300 induzierten Potenzial ab, sondern auch von der VEP-Antwort, die stark von der Blickrichtung abhängt (Fernando Nicolas-Alonso & Gomez-Gil, 2012).

4. Sensorimotor Rhythms (mu and beta rhythms)

Sensomotorische Rhythmen umfassen Mu- und Beta-Rhythmen, bei denen es sich um Schwingungen der Gehirnaktivität handelt, die sich im Mu-Band (7-13 Hz) bzw. im Beta-Band (13-30 Hz) befinden.

Obwohl beide Rhythmen unabhängig voneinander sein können, sind einige Mu- und Beta-Rhythmen harmonisch miteinander verbunden.

Die Amplitude sensomotorischer Rhythmen ändert sich, wenn die zerebrale Aktivität mit einer motorischen Aufgabe zusammenhängt, obwohl motorische Bewegung nicht direkt notwendig ist, damit sich die Amplitude der sensomotorischen Rhythmen ändert.

Sensomotorische Rhythmen können sich auf zwei Arten der Amplitudenmodulation stützen, die als ereignisbezogene Desynchronisation (ERD) und ereignisbezogene Synchronisation (ERS) bekannt sind und durch sensorische Stimulation, motorisches Verhalten und mentale Bilder erzeugt werden (Fernando Nicolas-Alonso & Gomez-Gil, 2012).

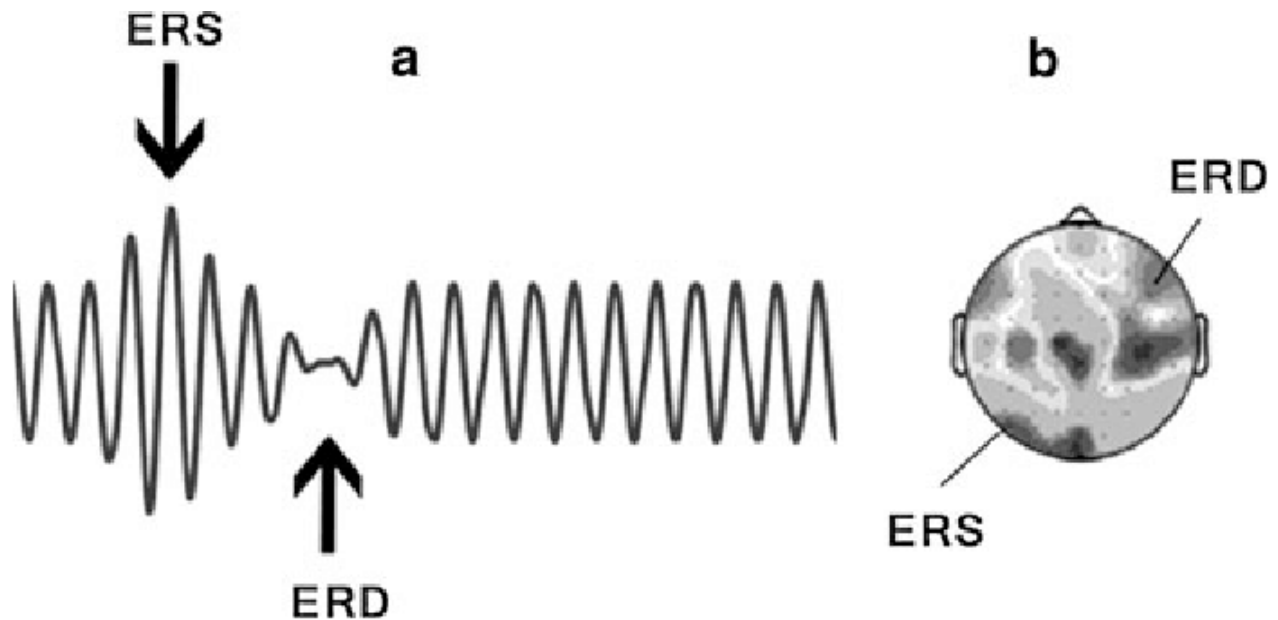


Figure 11 a) ERD/ERS components in an observed brain signal. b) Topographical map of ERD/ERS components in a hand movement motor imagery

ERD beinhaltet die Amplitudenreduktion des Rhythmus, und ERS bezieht sich auf Amplitudenamplitudenerhöhung.

Das Mu-Band ERD startet 2,5 s vor Beginn der Bewegung, erreicht kurz nach Bewegungsbeginn die maximale ERD und kehrt innerhalb weniger Sekunden zu seinem ursprünglichen Niveau zurück.

Im Gegensatz dazu zeigt der Beta-Rhythmus eine kurze ERD zu Beginn der Bewegung an, gefolgt von ERS, die nach der Ausführung der Bewegung ihr Maximum erreicht. Dies tritt auf, wenn der ERS-, Mu-Rhythmus noch geschwächt ist. Darüber hinaus ist die Gamma-Oszillation (36-40 Hz) ein weiterer Rhythmus, der mit motorischen Aufgaben zusammenhängt. Gamma-Rhythmen lösen kurz vor Beginn der Bewegung eine ERS aus. Darüber hinaus zeigt die folgende Abbildung, dass simultane ERD und ERS an verschiedenen Kopfhautpositionen möglich sind (Fernando Nicolas-Alonso & Gomez-Gil, 2012).

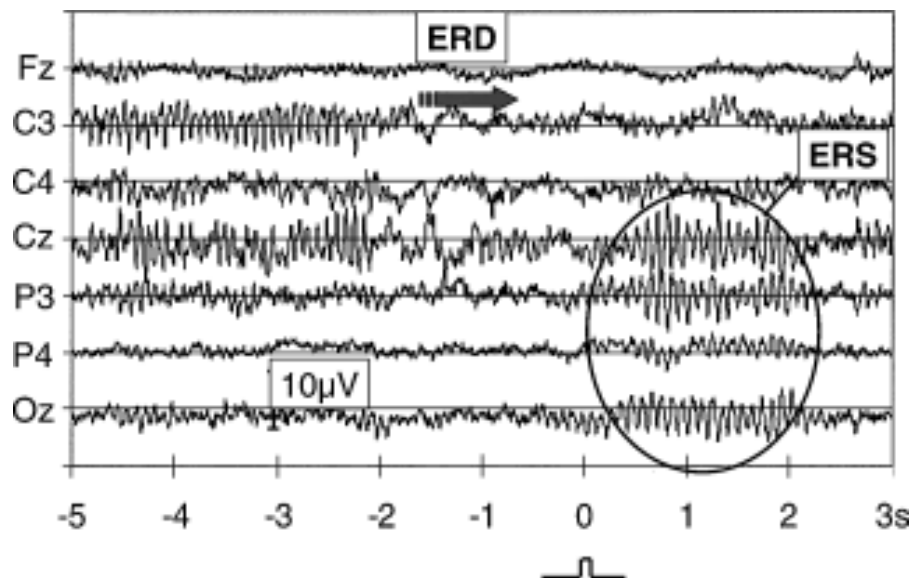


Figure 12 Examples of ongoing EEG recorded during right finger movement

Sensomotorische Rhythmen sind mit motorischen Bildern ohne tatsächliche Bewegung verbunden. Auf diese Weise ist es möglich, sensomotorische Rhythmen in endogenen BCI-Systemen zu verwenden, die nützlicher sind als exogene BCI-Systeme.

Darüber hinaus sind sensomotorische Rhythmen nicht leicht selbst zu kontrollieren, und die meisten Menschen haben Schwierigkeiten, motorische Bilder zu machen. Die Muster sensomotorischer Rhythmen unterscheiden sich von motorischen Bildern. Aber wenn Menschen versuchen, sich visuelle Bilder von realen Bewegungen vorzustellen, stellen sie sich Bilder vor, die nicht nützlich genug sind.

Die Benutzerschulung sollte durch kinästhetische Erfahrungen und nicht durch visuelle Darstellungen von Aktionen erfolgen. Motorisches Bildtraining findet traditionell durch visuelles oder auditives Feedback statt. Das Training der motorischen Bilder erfordert von den Benutzern, eine bestimmte motorische Bildaufgabe auszuführen, und dann werden sensomotorische Rhythmen extrahiert und klassifiziert, indem sie mit einer Referenz verglichen werden. Schließlich wird dem Teilnehmer ein visuelles oder auditives Feedback entsprechend dem Erfolg des Ergebnisses gegeben. Motorisches Bildtraining ist weit verbreitet, obwohl seine Wirksamkeit in der Regel nicht sehr hoch ist.

Jüngste Fortschritte auf dem Gebiet der BCI, die auf sensorisch-motorischen Rhythmen basieren, haben gezeigt, dass es möglich ist, menschliche willkürliche Bewegungen vorherzusagen, bevor sie auftreten, basierend auf Modulationen in sensorisch-motorischen Rhythmen. Außerdem kann

diese Schätzung bereitgestellt werden, ohne dass der Benutzer irgendwelche Schritte unternimmt (Fernando Nicolas-Alonso & Gomez-Gil, 2012).

E. Arten von BCIs

BCIs können in zwei Hauptgruppen eingeteilt werden:

1. Exogene und endogene BCIs
2. Synchron (mit der gleichen Geschwindigkeit wie die Markierung) oder Asynchron (mit eigener Geschwindigkeit)

Die Vor- und Nachteile von BCI-Arten sind in den folgenden Tabellen aufgeführt.

Tablo 3 Main differences between exogenous and endogenous BCI.

Annähern	Gehirnsignale	Vorteile	Nachteile
Exogene BCI	<ul style="list-style-type: none"> • SSVEP • P300 	<ul style="list-style-type: none"> • Minimaler Schulungsaufwand • Einfacher und schneller Steuersignalaufbau • Hohe Bitrate (60 Bit/min) • Nur ein EEG-Kanal erforderlich 	<ul style="list-style-type: none"> • Permanente Aufmerksamkeit für äußere Reize • Kann bei einigen Benutzern Müdigkeit verursachen
Endogene BCI	<ul style="list-style-type: none"> • SCPs • Sensorimotor rhythms 	<ul style="list-style-type: none"> • Unabhängig von jeglicher Stimulation • Kann nach freiem Willen betrieben werden • Nützlich für Benutzer mit betroffenen Sinnesorganen • Geeignet für Cursorsteuerungsanwendungen 	<ul style="list-style-type: none"> • Sehr zeitaufwendiges Training (Monate oder Wochen) • Nicht alle Benutzer sind in der Lage, die Kontrolle zu erlangen • Mehrkanal-EEG-Aufzeichnungen für gute Leistung erforderlich • Niedrigere Bitrate (20–30 Bit/min)

Tablo 4 Main differences between synchronous and asynchronous BCIs.

Annähern	Vorteile	Benachteiligungen
Synchrone BCI	<ul style="list-style-type: none"> Einfacheres Design und Leistungsbewertung Der Benutzer kann die Generierung von Artefakten vermeiden, da er Blinzeln und andere Augenbewegungen ausführen kann, wenn Gehirnsignale nicht analysiert werden 	<ul style="list-style-type: none"> Bietet keine natürlichere Art der Interaktion
Asynchrone BCI	<ul style="list-style-type: none"> Keine Notwendigkeit, auf externe Hinweise zu warten Bietet eine natürlichere Art der Interaktion 	<ul style="list-style-type: none"> Viel komplizierteres Design Schwierigere Bewertung

BCIs können je nach Art des Eingangssignals als exogen oder endogen klassifiziert werden.

Exogene BCIs nutzen die Neuronenaktivität, die im Gehirn durch einen externen Reiz wie VEPs oder auditiv stimulierte Potenziale erzeugt wird. Exogene Systeme erfordern relativ wenig Training, da Steuersignale SSVEPs und P300-Signale einfach installiert werden können. Darüber hinaus können die Steuersignale über einen einzigen EEG-Kanal verwendet werden und eine Übertragungsrate von bis zu 60 Bit pro Minute erreichen.

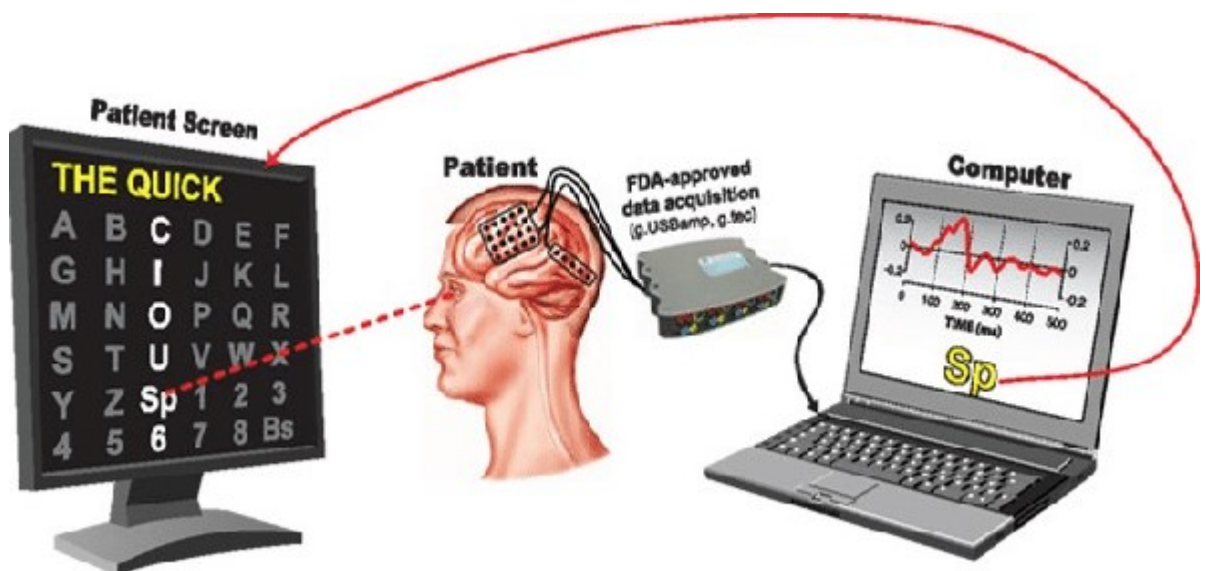


Figure 13 Conceptual illustration of an exogenous BCI

Endogene BCIs hingegen basieren auf der Selbstregulation von Gehirnrhythmen und -potentialen, ohne äußere Reize. Durch Training lernen die Benutzer, spezifische Gehirnmuster zu erzeugen, deren Code durch BCI dekodiert werden kann, wie z. B. sensomotorische Rhythmen und Modulationen in SCPs. Bei endogenen BCIs kann der Benutzer die BCI aus freien Stücken ausführen und einen Cursor auf dem Bildschirm zu einem beliebigen Punkt bewegen. Im Gegensatz dazu können exogene BCIs den Benutzer auf bestimmte Optionen beschränken. Es ist auch nützlich für Benutzer, die fortgeschrittene Stadien der endogenen BCI, ALS oder deren Sinnesorgane betroffen sind.

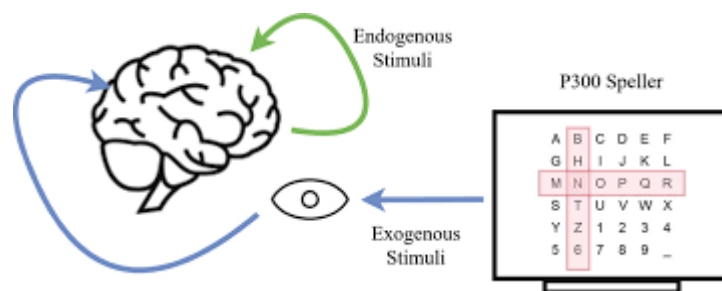


Figure 14 Stimuli for a P300 speller setting. The combination of

Entsprechend der Methode, mit der Eingaben verarbeitet werden, können BCIs als synchron oder asynchron klassifiziert werden. Synchrone BCIs analysieren Gehirnsignale während bestimmter Zeitrahmen. Gehirnsignale außerhalb der definierten Zeit werden nicht verarbeitet. Der Benutzer kann Befehle nur während des auf dem System angegebenen Zeitrahmens senden. Der Vorteil dieser Art von System ist das Wissen um die Anfangs- und Endmomente geistiger Aktivitäten.

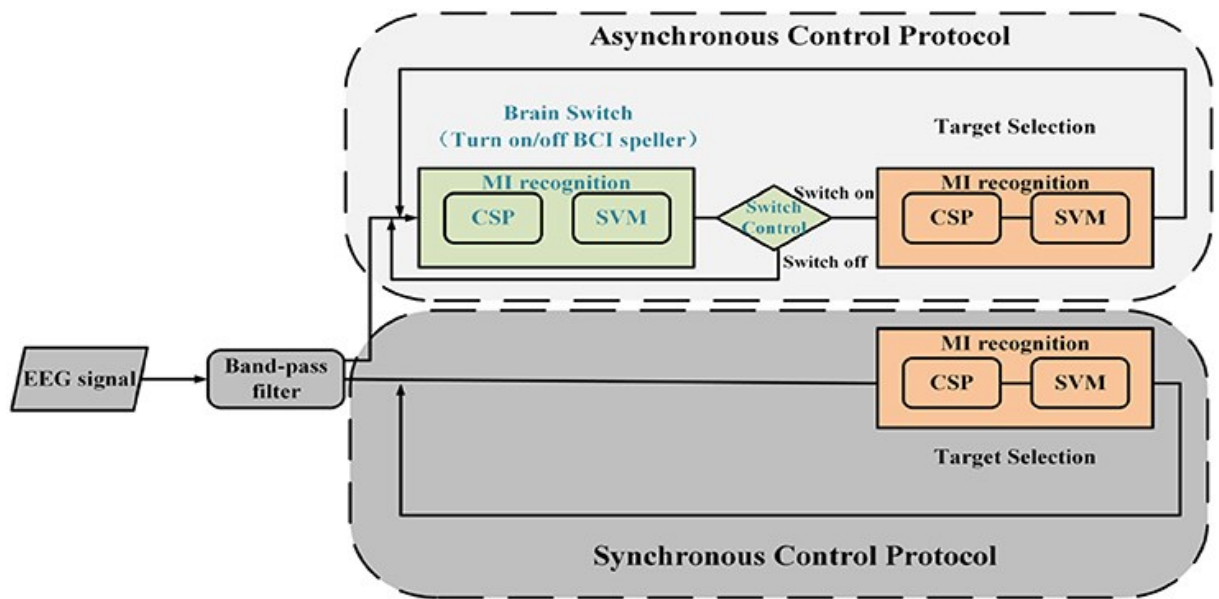


Figure 15 Framework for asynchronous and synchronous control protocols.

Asynchrone BCIs analysieren kontinuierlich Gehirnsignale, unabhängig vom Zeitpunkt des Bewegungsbeginns. Sie bieten eine natürlichere Mensch-Maschine-Interaktion als das synchronisierte BCI-System. Asynchrone BCIs erfordern jedoch mehr Berechnungen und sind komplex (Fernando Nicolas-Alonso & Gomez-Gil, 2012).

F. Komponenten von BCI

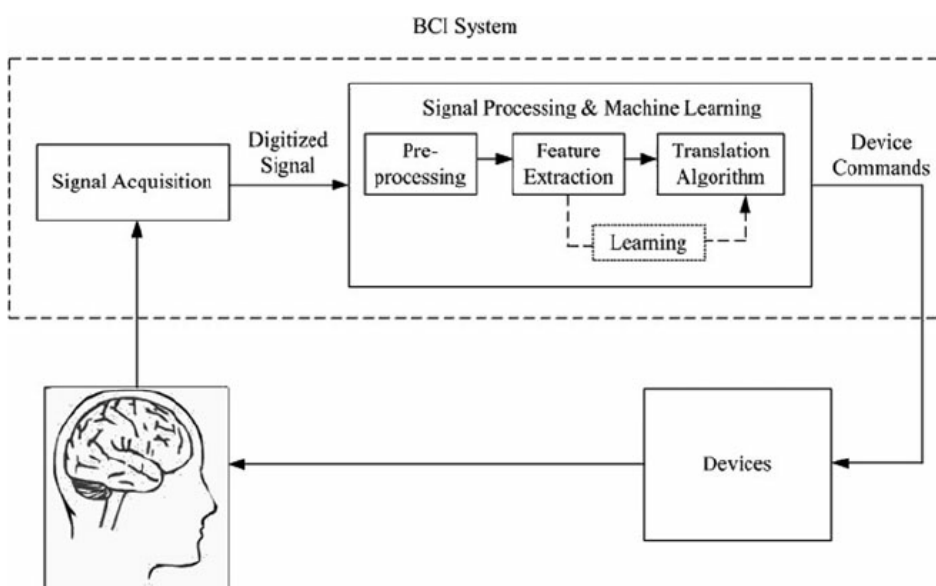


Figure 16 Basic design and operation of a BCI system

1. Signal Acquisition

Rohe EEG-Signale werden aufgezeichnet, indem Elektroden auf die Kopfhaut gelegt werden.

Da EEG-Signale durch nicht-invasive Methoden erhalten werden, sind die Amplitudenwerte niedrig. Daher werden die Signale durch die Vorverstärkungsstufe geleitet. Ein Bandpassfilter wird verwendet, um die gewünschten Frequenzbereiche aus den verstärkten Signalen zu erhalten.

Schwache Signale, die den Filter passieren können, werden mit dem Verstärkungsverstärker verstärkt. In Systemen, in denen EEG-Signale gesammelt werden, wird der Isolierung im Schaltungsdesign Bedeutung beigemessen, um vor elektrischen Schlägen zu schützen, die durch Leckagen verursacht werden können, die von Netzteilen oder Messgeräten auftreten können (BEKİRYAZICI, 2020).

2. Signal Processing

a. Pre-Processing

EEG-Signale können leicht durch physikalische und Umgebungsgeräuschquellen beeinflusst werden, da sie aufgrund ihrer Struktur kleine Amplitudenpegel aufweisen. Bei der Sammlung von EEG-Signalen werden Muskelbewegungen wie Bewegung, Blinzeln, Atmung als störender Effekt zu den Signalen hinzugefügt. Vorverarbeitungsprozesse werden durchgeführt, um die zerstörerische Wirkung all dieser physikalischen oder Umgebungsrauschquellen auf die Signale zu beseitigen (BEKİRYAZICI, 2020).

Die Vorverarbeitung von Gehirnsignalen spielt bei BCIs eine wichtige Rolle. Die Komponenten von Gehirnsignalen, die die Absichten des Benutzers widerspiegeln, sind nützlich, während die verbleibenden Komponenten als Rauschen betrachtet werden können. Im BCI-System, das auf dem P300 basiert, ist beispielsweise das P300-Potential die nützliche Komponente, und zu den Rauschkomponenten gehören Hochfrequenzsignale, Alphawellen, EMG, EOG und Interferenzen von Netzteilen und anderen Geräten. Da eine effektive Vorverarbeitungsmethode das Signal-Rausch-Verhältnis und die räumliche Auflösung verbessert, ist für die meisten BCI-Systeme eine Vorverarbeitung erforderlich, die anschließend die Leistung von BCIs verbessert. Räumliche und

zeitliche Filterung sind gängige Vorverarbeitungsmethoden, die in den meisten BCIs verwendet werden (Li et al., 2009).

b. Feature Extraction

Es ist der Prozess der Gewinnung der Attributmatrix durch Extrahieren der Attribute, die die in EEG-Signalen enthaltenen Informationen am besten ausdrücken können. Die Merkmalsextraktion ist einer der wichtigsten und kritischsten Schritte in BCI-Systemen. Dieser Prozessschritt erfordert das Extrahieren von Attributen, um Daten aus verschiedenen Klassen optimal unterscheiden zu können. Der Erfolg der extrahierten Attribute wirkt sich direkt auf die Klassifizierungsleistung und den Gesamterfolgsprozentsatz des Systems aus.

In vielen Studien in der Literatur wird es bevorzugt, viele Feature-Gruppen zusammen zu verwenden, anstatt ein einzelnes Feature zu verwenden, das verschiedene Klassen voneinander unterscheiden kann. Jedes entfernte Attribut kann die Markierung identifizieren, aber es kann keine 100%ige Genauigkeit im Klassifizierungsprozess bieten. Daher ist es notwendig, verschiedene Funktionen zusammen zu verwenden, um den Klassifizierungserfolg des Systems zu erhöhen.

Signale in EEG-basierten BCI-Systemen; Varianz, Standardabweichung, Maximalwert, Minimalwert, Mittelwert, Ebenheitskoeffizient (Kurtose), Schiefe, Hjorth-Parameter (Aktivität, Mobilität, Komplexität), Fourier-Transformationskoeffizienten, Wavelet-Transformationskoeffizienten, Mittelwert der Ableitung, Hilbert-Transformationskoeffizienten usw. werden in der Literatur häufig verwendet (BEKİRYAZICI, 2020).

Das Signal-Rausch-Verhältnis kann verbessert werden, indem Filter mit Rauschunterdrückung verwendet und unerwünschte Daten und Artefakte ignoriert werden. Viele räumliche, spektrale und zeitliche Filter werden verwendet, um den Vorverarbeitungsprozess durchzuführen. Damit die aufgezeichneten Signale angemessen genutzt werden können, sollten einige Merkmale, die das Signal am besten ausdrücken, bestimmt und bearbeitet werden. Diese Eigenschaften werden in der Literatur als "Attribute" bezeichnet und in einem Attributvektor gespeichert. Kurz gesagt, der Prozess der Attributextraktion wird als die Umwandlung von Signalen in eine Form ausgedrückt, die ihre Klassifizierung erleichtert. Es gibt verschiedene Attribute: die Zeitdomäne, die Frequenzdomäne und die räumliche Domäne (Buyukgoze, n.d.).

c. Extraction Methodik

- Principal Component Analysis (PCA)
- Independent Component Analysis (ICA)
- Common Spatial Pattern (CSP)
- AutoRegressive Components (AR)
- Matched Filtering (MF)
- Wavelet Transform (WT)

i. Principle Component Analysis (PCA)

PCA ist ein Verfahren zum Reduzieren der Größe des Features. Da die Anzahl der Spalten kleiner ist als die Anzahl der Eigenvektoren, ist die Größe der projizierten Ausgabedaten kleiner als die Größe der Eingabedaten. Diese Reduzierung der Dimensionalität kann die Komplexität des nächsten Klassifikationsschritts in einem BCI-System reduzieren. Da die am besten unterscheidenden Komponenten möglicherweise nicht zu den größten Hauptkomponenten gehören, garantiert PCA nicht immer eine gute Klassifizierung. PCA reduziert die Datengröße, indem nach einer neuen optimalen Darstellung der Daten in Bezug auf den minimalen durchschnittlichen quadratischen Fehler zwischen der Darstellung und den Originaldaten gesucht wird. Es garantiert nicht, dass die Unterscheidungsmerkmale für die Klassifizierung optimal sind. Trotz dieses Mangels hat sich PCA als zuverlässige Methode zur Rauschunterdrückung erwiesen. Hinsichtlich der Anwendungen von PCA in BCI-Systemen wurde PCA eingesetzt, um künstliche Komponenten recht erfolgreich in EEG-Signalen zu identifizieren und Signale ohne künstliche Komponenten zu rekonstruieren. Damit PCA auf diese Weise funktioniert, dürfen jedoch keine Artefakte mit dem EEG-Signal verknüpft werden. PCA wird auch verwendet, um die Dimensionalität des Feature-Raums zu reduzieren (Fernando Nicolas-Alonso & Gomez-Gil, 2012).

$$\begin{bmatrix} cov(x_1, x_1) & \dots & cov(x_1, x_n) \\ \dots & \dots & \dots \\ cov(x_n, x_1) & \dots & cov(x_n, x_n) \end{bmatrix}$$

Figure 17 Example Covariance Matrix

Die lineare Transformation erzeugt aus den Eingabedaten eine Reihe von Komponenten, die nach ihrer Varianz so sortiert sind, dass die erste Hauptkomponente die höchstmögliche Varianz aufweist. Diese Varianz ermöglicht es PCA, die Gehirnsignalisierung in verschiedene Komponenten zu zerlegen. Die Kovarianzmatrix ist Σ eine reelle und symmetrische Matrix $d \times d$, so dass Σ verschiedene Eigenvektoren und Eigenwerte d hat. Durch Eigenwerte ist es möglich zu wissen, welche Eigenvektoren die wichtigsten Informationen im Datensatz darstellen. Die Eigenvektoren mit dem höchsten Eigenwert stellen die Kernkomponenten des Trainingsdatasets p dar. Diese ausgewählten Eigenvektoren dienen dazu, eine Projektionsmatrix A zu erstellen, die verwendet wird, um den q -Attributvektor aus den Testdaten zu extrahieren. Die Eigenvektoren K sind in Spalten unterteilt, so dass in der Matrix A die erste Spalte von A dem größten Eigenwert entspricht. Schließlich berechnet PCA den Attributvektor v aus den Daten in Matrix A , indem es die Testdaten q in den neuen Unterraum projiziert (Fernando Nicolas-Alonso & Gomez-Gil, 2012).

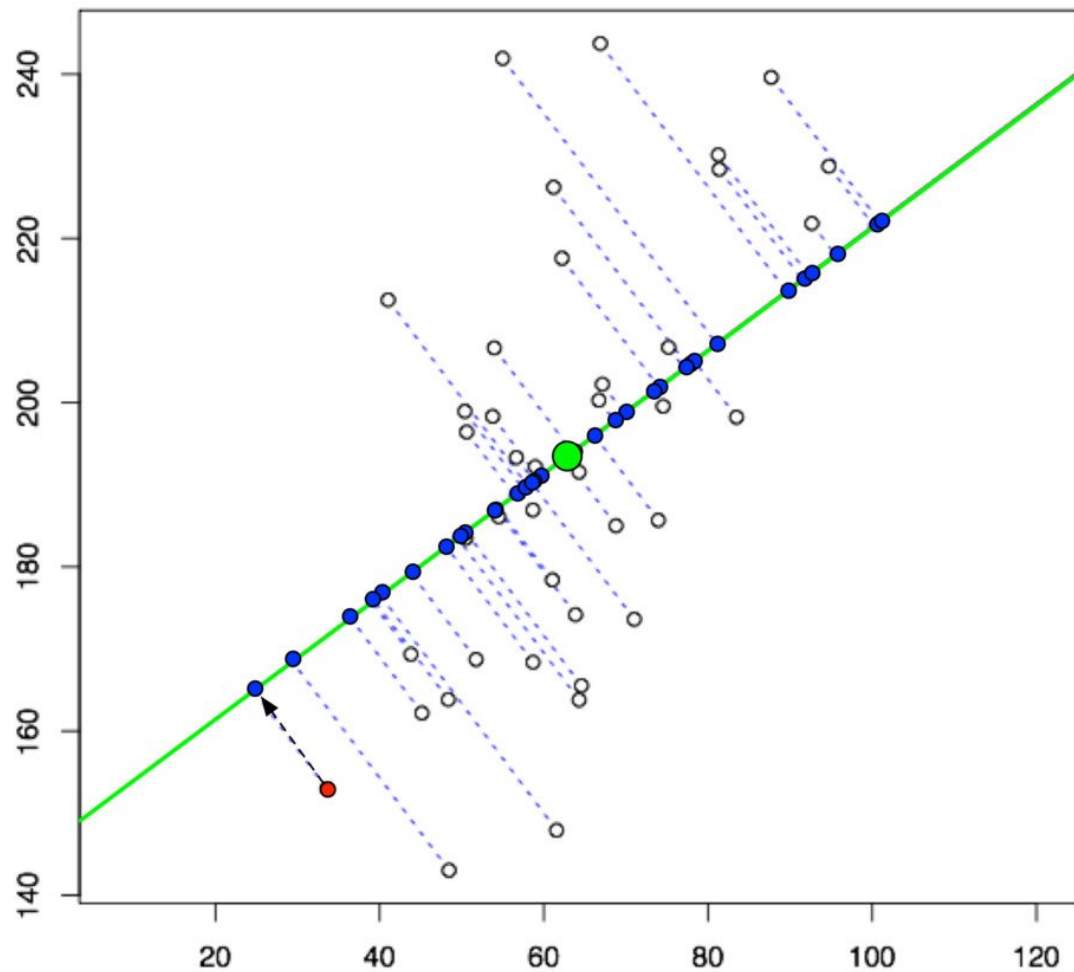


Figure 18 Dimension reduction using PCA

ii. Common Spatial Pattern (CSP)

Im Gegensatz zu PCA, das die Varianz der ersten Komponente im konvertierten Raum maximiert, maximieren gemeinsame räumliche Modelle (CSP) das Varianzverhältnis zweier Bedingungen oder Klassen. Mit anderen Worten, der CSP findet eine Transformation, die die Varianz von Instanzen einer Bedingung maximiert und gleichzeitig die Varianz von Instanzen der anderen

Bedingung minimiert. Diese Funktion macht CSP zu einem der effektivsten räumlichen Filter für die BCI-Signalverarbeitung, vorausgesetzt, dass die Absicht des Benutzers in der Varianz oder Stärke des zugehörigen Gehirnsignals codiert ist. BCIs, die auf Motorbildern basieren, sind typische Beispiele für solche Systeme. Der CSP-Algorithmus benötigt nicht nur Trainingsproben, sondern auch Wissen darüber, zu welcher Bedingung die Stichproben gehören, und berechnet die lineare Transformationsmatrix. Im Gegensatz dazu benötigen PCA und ICA diese zusätzlichen Informationen nicht. Daher sind PCA und ICA unüberwachte Methoden, während CSP eine überwachte Methode ist und Kenntnisse der Bedingung oder des Klassenlabels für jede Trainingsstichprobe erfordert. Für eine detailliertere Erklärung des CSP-Algorithmus stellen wir die Hypothese auf, dass $W \in \mathbb{R}^{n \times n}$ eine CSP-Matrix ist. Die Werte der Spalten in der Matrix stellen den Beitrag der CSP-Komponenten zu den Kanälen dar und können daher verwendet werden, um die topografische Verteilung der CSP-Komponenten zu visualisieren. Die topographischen Verteilungen dieser Komponenten entsprechen der erwarteten kontralateralen Aktivität der sensomotorischen Rhythmen, die durch motorische Bildaufgaben induziert werden. Das heißt, linksmotorische Bilder induzieren sensorisch-motorische Aktivitätsmuster (ERD / ERS) über den gemeinsamen sensomotorischen Bereichen, während rechte motorische Bilder zu Aktivitätsmustern über den linken sensomotorischen Bereichen führen. CSP und PCA basieren beide auf der Diagonalisierung von Kovarianzmatrizen. Während PCA jedoch eine Kovarianzmatrix diagonalisiert, diagonalisiert CSP gleichzeitig zwei Kovarianzmatrizen R_1 und R_2 , die zwei verschiedenen Klassen entsprechen. Für PCA reicht es aus, das Eigenwertproblem zu lösen. Für CSP muss das verallgemeinerte Eigenwertproblem mit R_1 und R_2 so gelöst werden, dass die W -Transformationsmatrix entsteht, die gleichzeitig zwei Kovarianzmatrizen diagonalisiert. Der grundlegende CSP-Algorithmus kann nur zwei Bedingungen verarbeiten (Li et al., 2009).

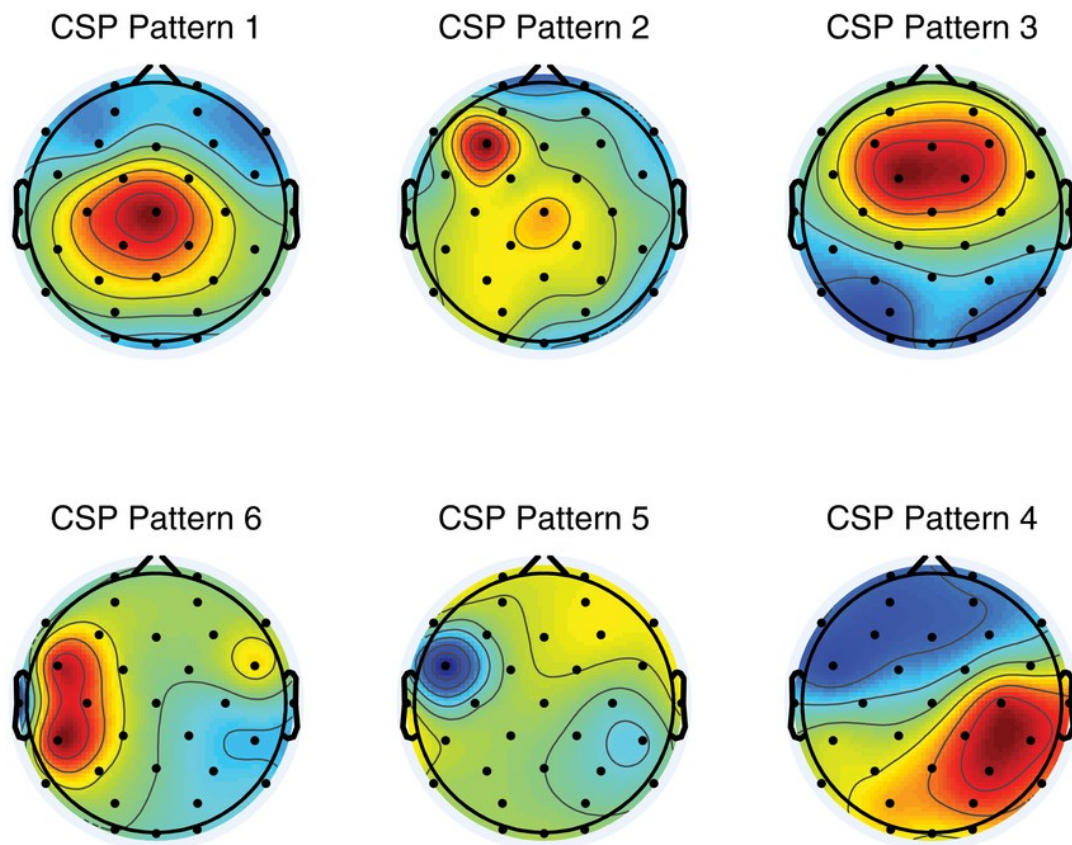


Figure 19 Common Spatial Pattern Maps

iii. Wavelet Transform (WT)

WT ist ein weit verbreitetes mathematisches Werkzeug, um Informationen aus vielen verschiedenen Arten von Daten zu extrahieren, wie z. B. Audio- oder Videodaten. WT eignet sich besonders dann, wenn Signale nicht stationär sind, da es eine flexible Möglichkeit bietet, die Zeitfrequenz eines Signals darzustellen.

Wavelets sind im Gegensatz zu anderen Methoden der Signalanalyse wie der Fourier-Transformation (FT) Funktionen mit variabler Frequenz und begrenzter Zeit, die es dem Signal ermöglichen, gleichzeitig im Zeit- und Frequenzraum zu arbeiten. Die FT liefert lediglich eine Analyse der Signalaktivität im Frequenzfeld. Die FT gibt Auskunft über den Frequenzgehalt, wird aber nicht von Informationen darüber begleitet, wann diese Frequenzen aufgetreten sind. Um dieses Manko der Fourier-Analyse zu beheben, wurde die kurzfristige Fourier-Transformation (STFT) vorgeschlagen. STFT unterteilt das Signal in aufeinanderfolgende Zeitfenster und wendet die FT auf jede Periode des Signals in der Zeit an. Bei diesem Ansatz ist die Gestaltung der Fensterlänge eine

Änderung, da kleinere Fenster zu einer höheren zeitlichen Auflösung sowie einer geringeren Frequenzauflösung führen. WT überwindet diesen Nachteil, indem es das Signal in mehreren Auflösungen sowohl im Zeit- als auch im Frequenzraum zerlegt, wobei ein moduliertes Fenster verwendet wird, das das Signal auf verschiedenen Skalen verschiebt. Die kontinuierliche Wavelet-Transformation (CWT) ist definiert als die Faltung des $x(t)$ -Signals mit der Wavelet-Funktion.

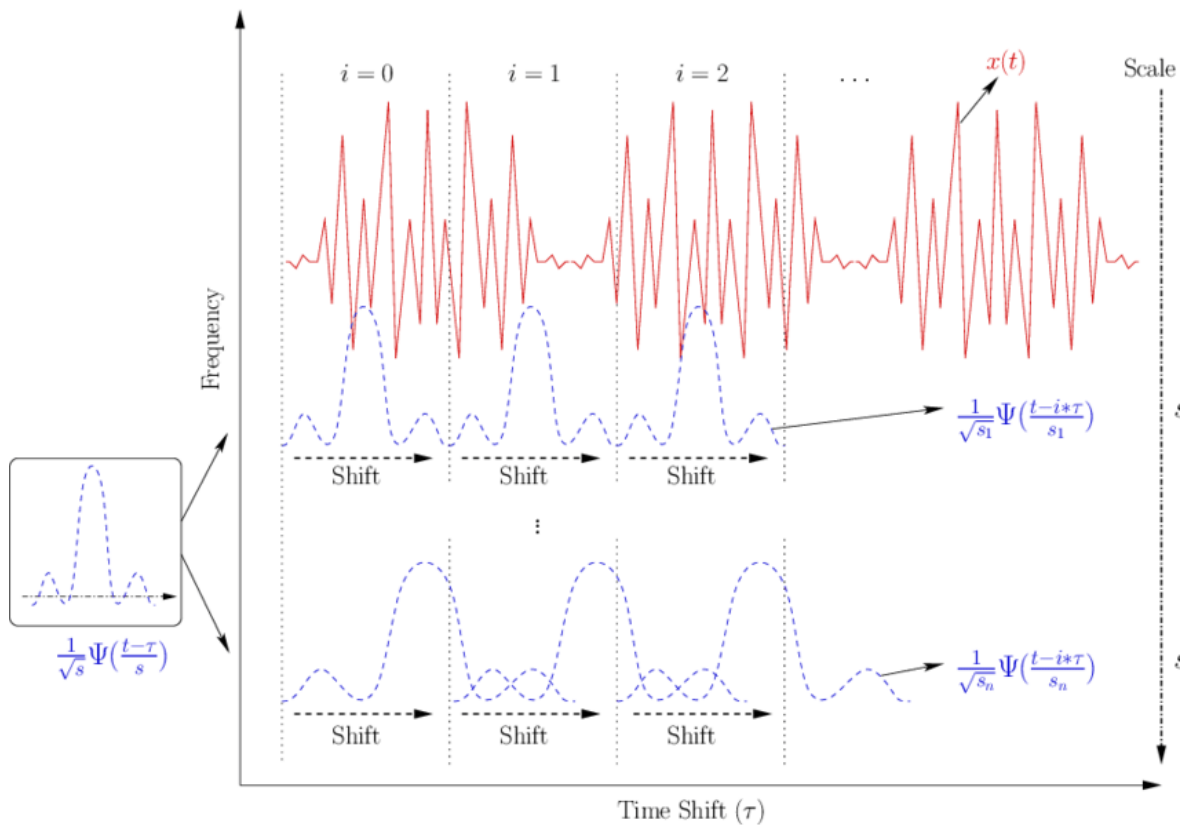


Figure 20 Wavelet transform illustration.

CWT ist eine Art Mustervergleich ähnlich einem Matching-Filter, bei dem die Kreuzvarianz zwischen dem Signal und einer vordefinierten Wellenform berechnet wird. Der Vorteil von CWT gegenüber herkömmlichen Modellabgleichsmethoden liegt in den besonderen Eigenschaften des Wavelet-Modells. Wavelets eignen sich für die transiente Signalanalyse, bei der sich die spektralen Eigenschaften des Signals im Laufe der Zeit ändern. WT ist ein leistungsfähiges Werkzeug zur Zerlegung von temporalen Gehirnsignalen, basierend auf einer Kombination von Kriterien wie Frequenz und zeitlicher Position. Signale im gleichen Frequenzbereich können durch zeitliche Position unterschieden werden. Ebenso ist es möglich, Prozesse, die sich zeitlich überschneiden, durch den unterschiedlichen Frequenzgehalt zu trennen. CWT bietet viel Redundanz und Komplexität, da es die Analyse eines sehr hochfrequenten Signals mit einer großen Anzahl von Erweiterungen und

Verschiebungen bei der Hauptwellenlänge beinhaltet. Um diese Redundanz und Komplexität zu reduzieren, wurde die diskrete Wavelet-Transformation (DWT) eingeführt. RWD übersetzt und erweitert das Master-Wavelet nur bei bestimmten diskreten Werten. Die Forscher demonstrierten einen Modellerkennungsansatz zur Klassifizierung kortikaler Potentiale, die mit einer einzigen experimentellen Aktion verbunden sind, bei der der Merkmalsraum aus den Koeffizienten einer separaten Wavelet-Transformation konstruiert wird. Obwohl DWT weniger unnötig und weniger komplex ist als CWT, wird CWT immer noch verwendet, um Features aus P300 und SCP zu extrahieren, da es feine Informationen schärfen kann, die DWT nicht extrahieren kann. Die Verwendung von WT erfordert die Auswahl eines Haupt-Wavelets. Viele verschiedene Haupt-Wavelets können in BCI-Anwendungen gefunden werden, und die Wahl hängt davon ab, welche Art von Eigenschaften aus dem Signal extrahiert werden müssen. Das Mexican Line Wavelet ist im Zeitraum gut lokalisiert und wird verwendet, um ERP-Komponenten im Laufe der Zeit zu lokalisieren. Das Morlet-Wavelet ist im Frequenzfeld gut lokalisiert und wurde für die Analyse der Gammaaktivität verwendet. Das Double-Scale-Wavelet wurde erfolgreich verwendet, um ein asynchrones BCI zu entwerfen, das auf der Erkennung imaginärer Bewegungen im Frequenzbereich von 1-4 Hz basiert. Auch die Daubechies-Welle, ein sehr bekanntes Haupt-Wavelet, wurde verwendet, um SCPs zu klassifizieren (Fernando Nicolas-Alonso & Gomez-Gil, 2012).

d. Feature Selection

Die Anzahl der von einem BCI-System aufgezeichneten Kanäle kann groß sein. Die Anzahl der Attribute, die aus jedem Kanal extrahiert werden, kann ebenfalls groß sein. Im Beispiel von 128 Kanälen und Features, die von einem 6-reihigen AR-Modell abgeleitet wurden, würde die Gesamtzahl der Features, dh die Größe des Feature-Bereichs, 6-mal 128 betragen. Das ist eine ziemlich große Menge an Funktionen. Während davon ausgegangen werden kann, dass Daten, die von mehreren Kanälen aufgezeichnet wurden, mehr Informationen enthalten, sind einige wahrscheinlich unnötig und sogar irrelevant in Bezug auf den Klassifizierungsprozess. Außerdem gibt es mindestens zwei Gründe, warum die Anzahl der Funktionen nicht sehr groß ist. Erstens kann die Rechenkomplexität zu groß sein, um die Echtzeitanforderungen eines BCI zu erfüllen. Zweitens kann eine Vergrößerung des Feature-Speicherplatzes zu einer Abnahme der Leistung des BCI-Systems führen. Dies liegt daran, dass die in BCI-Systemen verwendeten Mustererkennungsmethoden mithilfe von Trainingsdaten abgestimmt (trainiert) werden und das BCI-System stark von diesen unnötigen und sogar irrelevanten Dimensionen des Feature-Raums beeinflusst wird. Aus diesem Grund wählen BCIs in der Regel eine Teilmenge von Eigenschaften für die spätere Verarbeitung aus. Diese Strategie wird als Feature- oder Variablenauswahl bezeichnet. Durch die Verwendung der Feature-Auswahl kann die Größe des

Feature-Bereichs effektiv reduziert werden. Die Auswahl von Merkmalen kann auch in der BCI-Forschung nützlich sein. Bei der Verwendung neuer mentaler Aufgaben oder neuer BCI-Paradigmen ist möglicherweise nicht immer klar, welche Kanalpositionen und welche Eigenschaften am besten geeignet sind. In solchen Fällen kann die Feature-Auswahl helfen.

Zunächst werden viele Merkmale aus verschiedenen Methoden und Kanälen extrahiert, die alle potenziell nützlichen Gehirnbereiche abdecken. Anschließend wird die Feature-Auswahl durchgeführt, um optimale Kanalpositionen und Feature-Extraktionsmethoden zu finden. In diesem Abschnitt werden zwei Aspekte der Featureauswahl für BCIs beschrieben: die Kanalauswahl und die Frequenzbandauswahl(Li et al., 2009).

e. Selection Methods

- Genetic Algorithm (GA)
- Sequential Selection

f. Feature Classification

Es ist der Prozess der Zuordnung der relevanten Marke zu einer vorgegebenen Klasse als Ergebnis der Bewertung der aus den EEG-Marken extrahierten Attribute. Im Klassifizierungsprozess wird eine Attributmatrix erstellt, indem zunächst bestimmte Attribute aus jeder Trainingsdaten extrahiert werden. Als nächstes wird der Klassifikator mit der Feature-Matrix und den Labels der entsprechenden Signale trainiert. Während der Testphase wird die aus den Testdaten extrahierte Attributmatrix dem trainierten Klassifikatorialgorithmus hinzugefügt und entschieden, zu welcher Klasse das Experiment gehört(BEKİRYAZICI, 2020).

g. Classification Algorithms

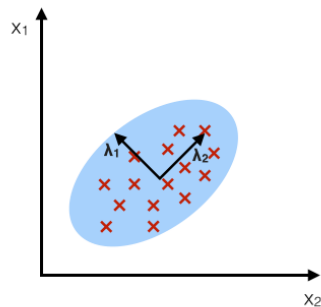
- Linear Discriminant Analysis (LDA)
- Support Vector Machine (SVM)
- K-Nearest Neighbor Classifier (k-NNC)
- Artificial Neural Network (ANN)

i. Linear Discriminant Analysis (LDA)

Die Linear Separation Analysis (LDA) ist ein Algorithmus, der entscheidet, ob die Durchschnittswerte der Daten signifikante Unterschiede in Klassifizierungsproblemen mit zwei oder mehr Klassen aufweisen. Diese Methode bestimmt die Variablen, die am meisten dazu beitragen, die Klassen voneinander zu trennen. Variablen mit einem niedrigen Deckungsbeitrag werden kombiniert, um eine neue Variable zu erzeugen. Die Differenz zwischen der neuen Variablen und den Variablen mit hohem Deckungsbeitrag ist auf höchstem Niveau. Die LDA-Analyse wird häufig in BCI-Anwendungen bevorzugt, da sie ein einfacher und schneller Klassifikator ist (BEKİRYAZICI, 2020).

PCA:

component axes that maximize the variance



LDA:

maximizing the component axes for class-separation

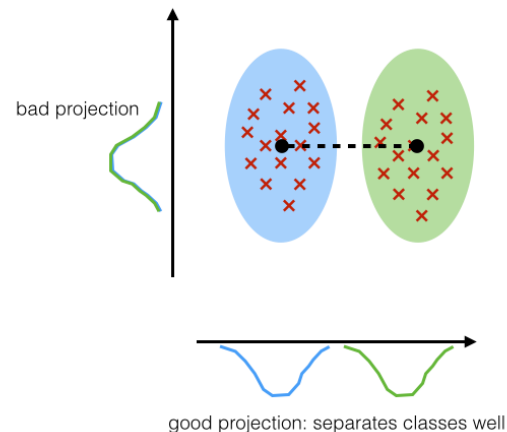


Figure 21 Difference between LDA and PCA

Die lineare Diskriminanzanalyse (LDA) wird als Dimensionsreduktionstechnik in der Vorverarbeitungsphase für Machine Learning-Anwendungen eingesetzt. Ziel ist es, eine Überanpassung zu verhindern und gleichzeitig die Kalkulationskosten zu senken.

Obwohl LDA im Allgemeinen PCA ähnelt, besteht die Arbeitslogik von LDA darin, den Abstand zwischen den Klassen zu maximieren. Es gibt kein Klassenkonzept in PCA. PCA versucht nur, den Abstand zwischen Datenpunkten zu maximieren.

Zusammenfassend lässt sich sagen, dass der Zweck von LDA darin besteht, die Größe des Datasets zu reduzieren, indem der Unterschied zwischen den Klassen maximiert wird.

ii. Support Vector Machine (SVM)

SVM ist eine Unterkategorie des überwachten Lernens, die verwendet wird, um Daten für die Klassifizierungs- und Regressionsanalyse zu analysieren. Ziel ist es, Punkte im Raum so abzubilden, dass die Stichproben der Zielkategorie durch den größtmöglichen Zähler dividiert werden. Dadurch wird sichergestellt, dass SVM als Klassifikator einen geringeren Generalisierungsfehler aufweist. Das Ziel ist es, eine Hyperebene oder eine Menge von Hyperebenen in einem n -dimensionalen Raum zu finden. Stützvektoren sind Datenpunkte, die näher an einer bestimmten Hyperebene liegen. Sie maximieren den Rand des Klassifikators, indem sie die Position und Ausrichtung der Hyperebene ändern. Darüber hinaus ist es innerhalb dieser Lücke möglich, dass die Punkte aufgrund der Position der Daten nicht linear getrennt sind. Um dieses Problem zu beheben, kann SVM die generierten Quorumfunktionen oder das Dataset verwenden, besser bekannt als "Kernel-Tricks". Dieser Trick beinhaltet die Konvertierung des vorhandenen Algorithmus von einem niedrigdimensionalen Datensatz in einen höheren Datensatz. Die Informationsmenge bleibt gleich, aber es ist möglich, einen linearen Klassifikator in diesem höherdimensionalen Raum zu erstellen. Jedem Punkt sind mehrere Kernel zugeordnet, die helfen, die optimale Hyperebene für den neu transformierten Feature-Bereich zu bestimmen. Mit ausreichenden K-Funktionen ist eine präzise Trennung möglich (Hosseini et al., 2021).

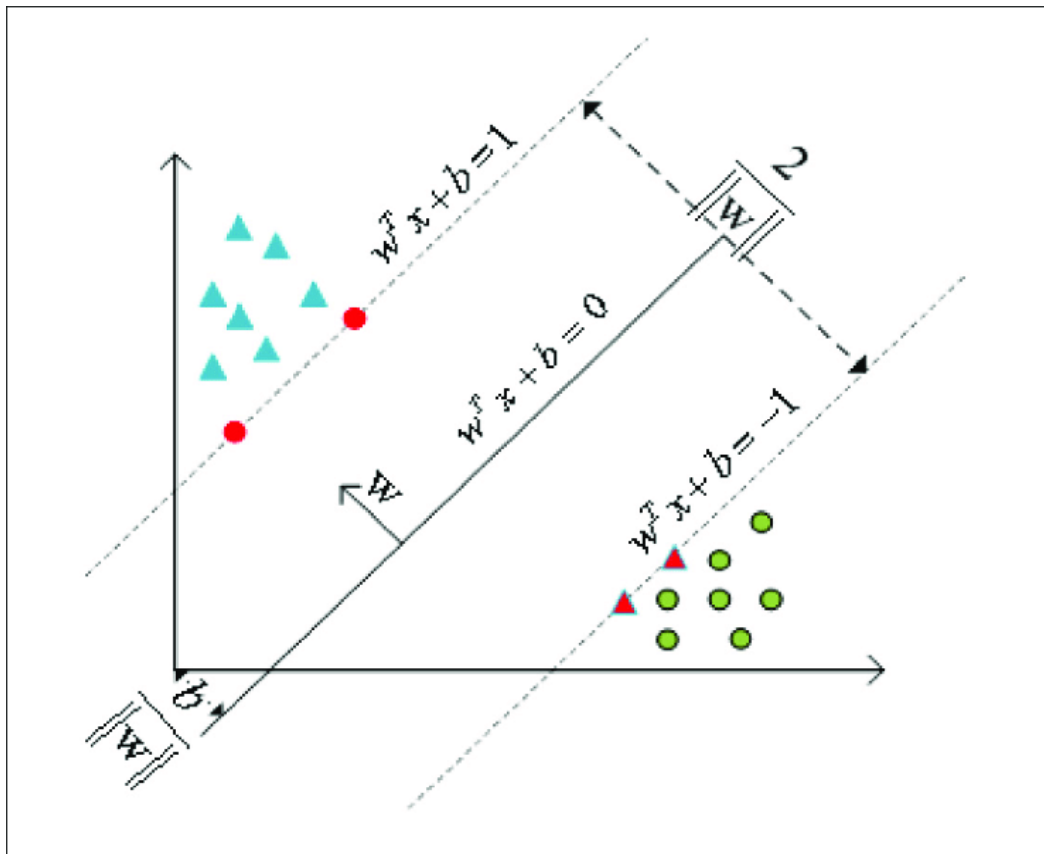


Figure 22 Principle of support vector machine.

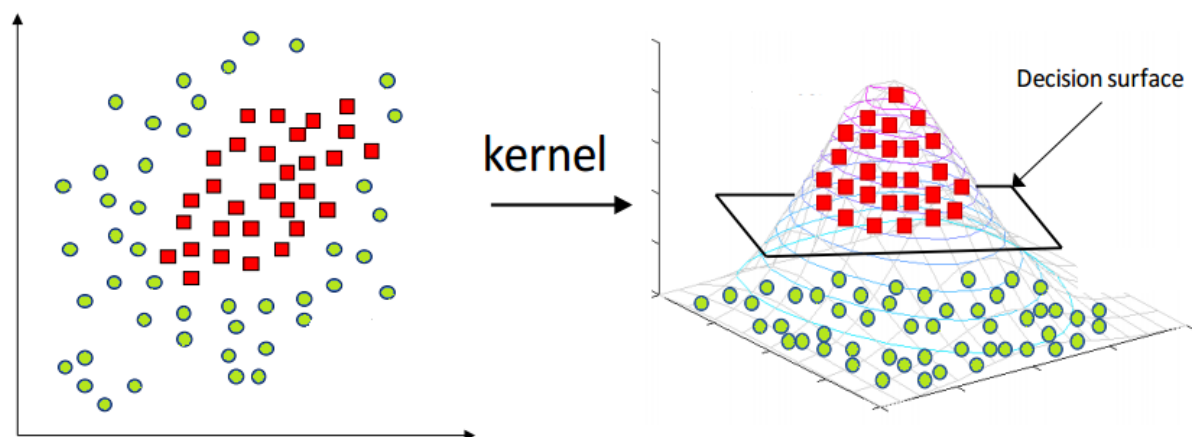
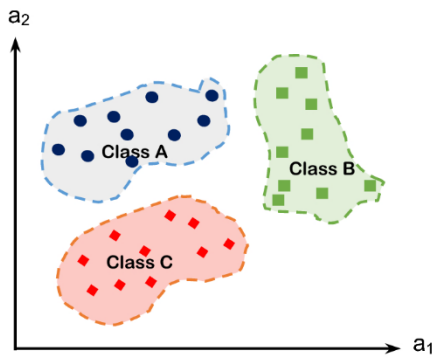


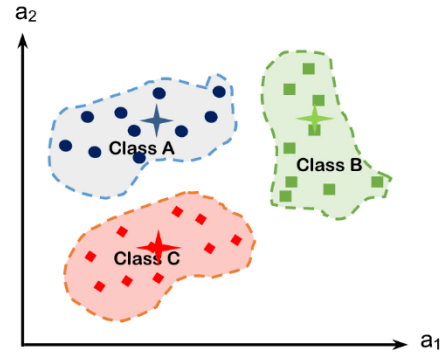
Figure 23 Kernel Method in SVM

iii. K-Nearest Neighbor Classifier (k-NNC)

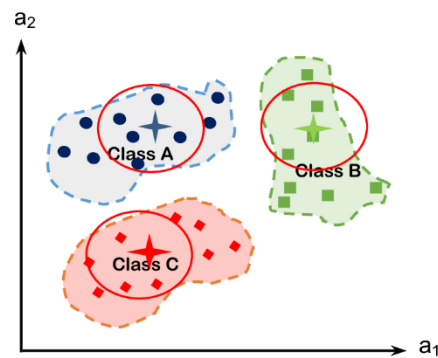
KNN ist einer der überwachten Algorithmen für maschinelles Lernen. Beim überwachten Lernen ist die Beziehung zwischen Input und Output für den Trainingsdatensatz bereits hergestellt, d.h. der Output für einen bestimmten Input ist bereits bekannt. Überwachtes Lernen wird als Regression und Klassifizierung klassifiziert. KNN kann sowohl für die Klassifizierung als auch für die Regression verwendet werden. Die Eingabe für Klassifizierung und Regression ist identisch, aber die Ausgabe ist unterschiedlich. Für nicht trainierte Datensätze werden Beispiel-Input-Output-Paare verwendet, um die Ausgabe vorherzusagen. KNN klassifiziert die Eingabe entsprechend der Klassifizierung seiner K-Nachbarn. Um die nächsten Nachbarn zu finden, wird die euklidische Entfernung oder Mahalanobis-Entfernung vom Eingang zu allen bekannten Datenpunkten berechnet. Nach der Berechnung der Entfernung werden die K-nächsten Nachbarn ausgewählt. Anschließend klassifiziert es die Eingabe nach den Ähnlichkeiten zwischen der Eingabe und ihren K-Nachbarn. Die Wahl von K hängt von der Größe des Datensatzes ab. Die Quadratwurzel der Größe des Datensatzes wird genommen, und wenn das Ergebnis eine gerade Zahl ist, wird 1 addiert oder subtrahiert. Das Ergebnis wird dann als K für diesen Datensatz ermittelt. K wird als ungerade Zahl ausgewählt, um Verzerrungen bei der Eingabeschätzung zu vermeiden (Hosseini et al., 2021).



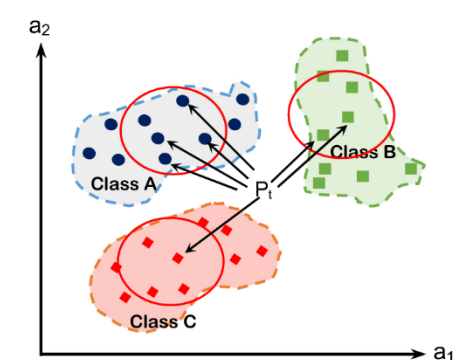
1- The training data is presented in the feature space.



2- Calculate the class center of each class and the distance between the center and each point in the class.



3- Calculate the average distance in each class.



4- Classify the target point.

Figure 24 Steps of the K-nearest neighbor

iv. Artificial Neural Network (ANN)

Deep Learning, das seinen Entwicklungsprozess mit der Entstehung des Konzepts der künstlichen neuronalen Netze in den frühen 1900er Jahren begann, hat seinen Platz in jedem Aspekt unseres heutigen Lebens mit dem Fortschritt der Technologie eingenommen. Deep Learning, das eine Teilmenge der Konzepte der künstlichen Intelligenz und des maschinellen Lernens ist, hat in vielen Studien in der Literatur einen großen Beitrag zur Lösung komplexer Probleme mit hoher Genauigkeit geleistet. Deep-Learning-Methoden; Es spielt eine wichtige Rolle in vielen verschiedenen Bereichen wie Klassifizierung, Vorhersage, Vorhersage, Klangverarbeitungsanwendungen, Anwendungen zur Verarbeitung natürlicher Sprache, Mustererkennung. Deep-Learning-Methoden erstellen mathematische Modelle, die den Lernprozess basierend auf der Arbeitslogik des menschlichen Gehirns durchführen. Lernprozesse können in Form von überwachtem Lernen und unbeaufsichtigtem Lernen

Lernen durchgeführt werden. Überwachte Lernalgorithmen beinhalten die Zuordnung von Eingaben im Datensatz zu bekannten Ausgaben. In unüberwachten Lernalgorithmen zielt es darauf ab, die gewünschte Operation ohne die Hilfe einer bekannten Ausgabe aus den Eingaben im Datensatz auszuführen. Unüberwachte Lernalgorithmen werden häufig verwendet, um Daten zu visualisieren, die Beziehung zwischen Daten zu verstehen, Daten zu komprimieren und das Rauschen in den Daten zu reduzieren (BEKİRYAZICI, 2020).

01. Neuronale Netze und ihr mathematisches Modell

Das menschliche Gehirn hat viele Fähigkeiten, wie Training, Lernen, Vorhersagen, Klassifizieren und Visualisieren. Das Konzept der neuronalen Netze ist für die Realisierung dieser Fähigkeiten durch unabhängige Systeme entstanden.

Eine Nervenzelle im menschlichen Gehirn besteht aus Dendriten, Zellkernen, Axonen und Synapsen. Dendrit überträgt Impulse vom zentralen Nervensystem oder anderen Nervenzellen an den Zellkörper. Der Zellkörper enthält den Zellkern und die Organellen. Dank dieser Komponenten machen die Informationen aus den Dendriten Sinn. Das Axon ist dafür verantwortlich, diese Information an andere Nervenzellen weiterzugeben. Synapsenverbindungen verbinden Axonklemmen mit Dendriten in anderen Zellen. Mit künstlichen neuronalen Netzen werden mathematische Modelle erstellt, die die Funktion dieser Teile der Nervenzelle übernehmen können.

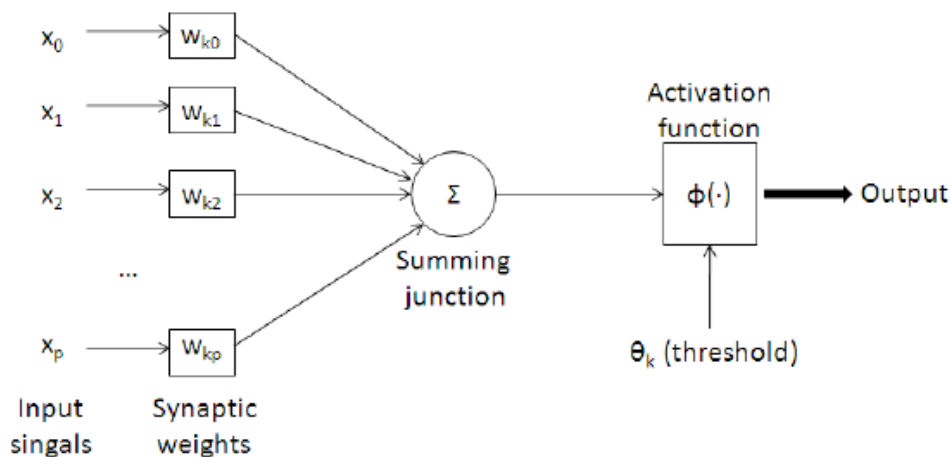


Figure 25 Mathematical model of an artificial neural network (ANN)

Die Eingangssignale, die zu den Axonenden kommen, werden mit ihren Gewichtungskoeffizienten multipliziert und im Zellkörper gesammelt. Die resultierende Summe

durchläuft die Aktivierungsfunktion durch Addieren des Bias-Wertes und wird auf die Ausgabe übertragen(BEKİRYAZICI, 2020).

02. *Perceptron*

Das Sensormodell, das 1958 vom Psychologen Frank Rosenblatt entwickelt wurde, um "einige der Grundmerkmale intelligenter Systeme zu simulieren", basiert auf dem Prinzip, dass ein Neuron mehrere Eingaben aufnimmt und eine Ausgabe erzeugt. Die Ausgabe des Netzwerks erhält man, indem man einen Schwellenwert mit der gewichteten Summe der Eingabewerte vergleicht. Wenn er größer oder gleich dem Gesamtschwellenwert ist, ist der Ausgabewert 1, und wenn er kleiner ist, ist er 0. Rosenblatt entwickelte eine Lernregel für das Sensortraining, die Mustererkennungsprobleme löst. Er bewies, dass, wenn es Gewichte gibt, die das Problem lösen, diese Regel immer zu den richtigen Gewichten konvergieren wird. Als Ergebnis ihrer tiefen mathematischen Forschung an Sensoren zeigten Marvin Minsky und Seymour Papert in ihrem Buch "Perceptrons" deutlich, dass Sensoren auf sehr engem Raum eingesetzt werden können und dass es zu viele Klassen von Problemen gibt, die der Detektor nicht lösen kann. Ein Beispiel für ein Problem, das Sensoren nicht lösen können, ist das XOR-Problem. Diese Einschränkung der Sensoren wurde durch die Entwicklung des mehrschichtigen Sensormodells in den 1980er Jahren überwunden(Arı et al., 2017).

03. *Adaline Model*

Bernard Widrow begann in den späten 1950er Jahren mit der Arbeit an neuronalen Netzen, als Frank Rosenblatt den Sensor entwickelte. Im Jahr 1960 entwickelten Widrow und der Doktorand Marcian Hoff eine Lernregel namens Smallest Average Square Algorithmus mit dem ADALINE-Netzwerk. Dieses Neuronenmodell mit dem expliziten Namen "ADaptive LINEar NEuron" oder "ADaptive LINEar Element" unterscheidet sich strukturell nicht vom Sensor. Wenn ADALINE jedoch eine lineare Funktion verwendet, verwendet der Sensor die Schwellenwertfunktion als Aktivierungsfunktion. Beide Modelle können nur Lösungen für linear trennbare Probleme liefern. Der Least Squares-Algorithmus, auch Widrow-Hoff-Regel genannt, ist leistungsfähiger als die Lernregel des Sensors. Selbst wenn die Sensorlernregel die Konvergenz zu einer Lösung garantiert, kann sie aufgrund der Nähe der Trainingsmuster zur Grenzlinie empfindlich auf Rauschen reagieren. Der Algorithmus der kleinsten Quadrate versucht, Trainingsmodelle so weit wie möglich von der Berandungslinie entfernt zu halten, um den durchschnittlichen Framefehler zu minimieren. Widrow und Hoff entwickelten auch das neuronale Netzwerkmodell MADALINE, das mehrere adaptive Elemente enthält(Arı et al., 2017).

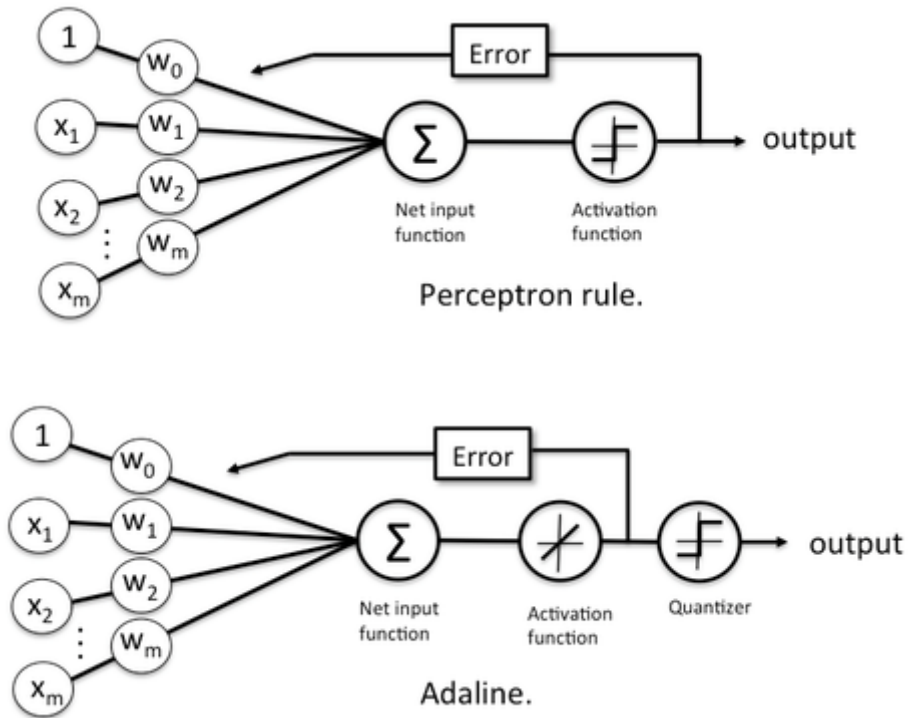


Figure 26 Perceptron and ADALINE models

04. Multilayer Perceptron

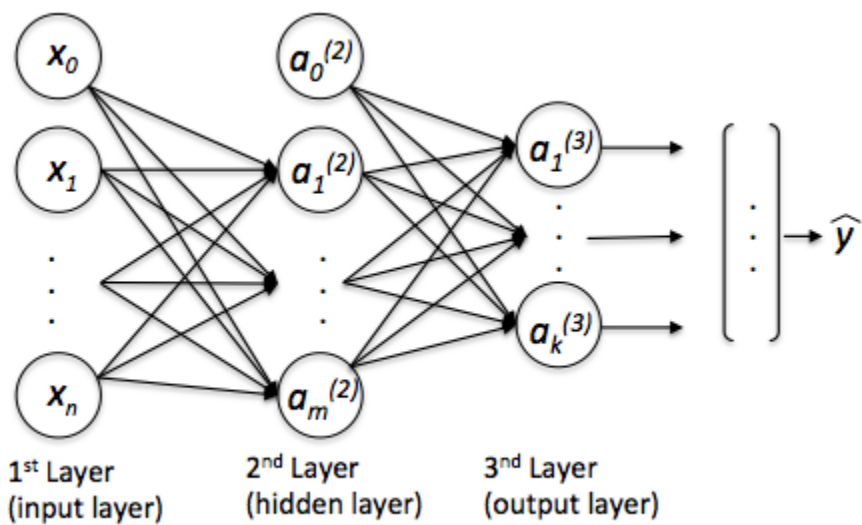


Figure 27 MLP model

Mehrschichtsensoren (MLP), die entwickelt wurden, nachdem einschichtige Sensoren nichtlineare Probleme nicht lösen konnten, bestehen aus einer Eingabeschicht, einer oder mehreren verborgenen Schichten und einer Ausgabeschicht. In MLP gibt es Übergänge zwischen Schichten, die als Vorwärts- und Rückwärtsspanne bezeichnet werden. In der Vorwärtsausbreitungsphase werden die Ausgabe des Netzwerks und der Fehlerwert berechnet. Während der Backpropagationsphase werden die Gewichtungswerte zwischen den Layer-Links aktualisiert, um den berechneten Fehlerwert zu minimieren. Das MLP-Modell verwendet den Backpropagation-Lernalgorithmus, der eine Verallgemeinerung des Algorithmus der kleinsten Quadrate im linearen Sensor darstellt.(Arı et al., 2017).

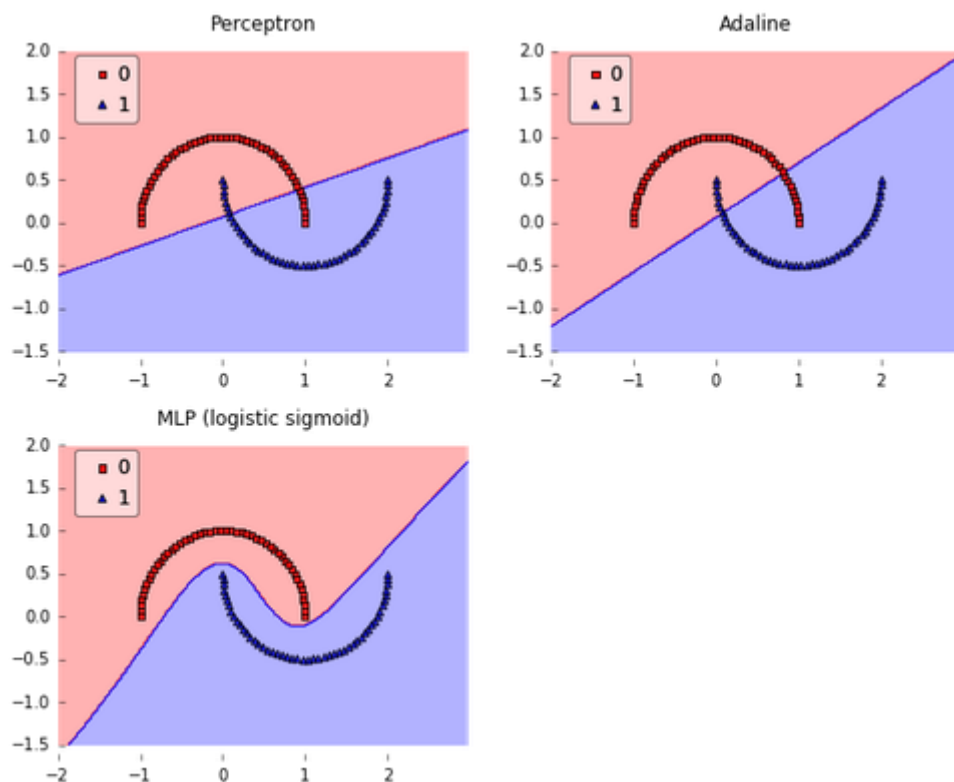


Figure 28 Comparing of classifying ability of 3 methods

05. Vorwärts- und Rückwärtsausbreitungsalgorithmen in neuronalen Netzen

Damit das Lernen in künstlichen neuronalen Netzen stattfinden kann, werden Vorwärts- und Rückwärtsausbreitungsoperationen unter Verwendung von Eingangssignalen, Gewichtungskoeffizienten

und Bias-Werten durchgeführt. Im generierten neuronalen Netzwerkmodell; Mathematische Operationen, von Eingangssignalen bis hin zu Ausgangswerten, werden als Vorwärtsausbreitung bezeichnet. Während der Entwurfsphase des neuronalen Netzes werden die Gewichtswerte in der Regel zu Beginn zufällig ausgewählt. Nach der weiteren Weitergabe wird der Ausgabewert mit dem tatsächlichen Beschriftungswert der Daten verglichen. Mathematische Operationen zum Aktualisieren der Gewichts- und Bias-Werte, um den Fehlerwert dazwischen zu minimieren, werden als Backpropagation definiert.

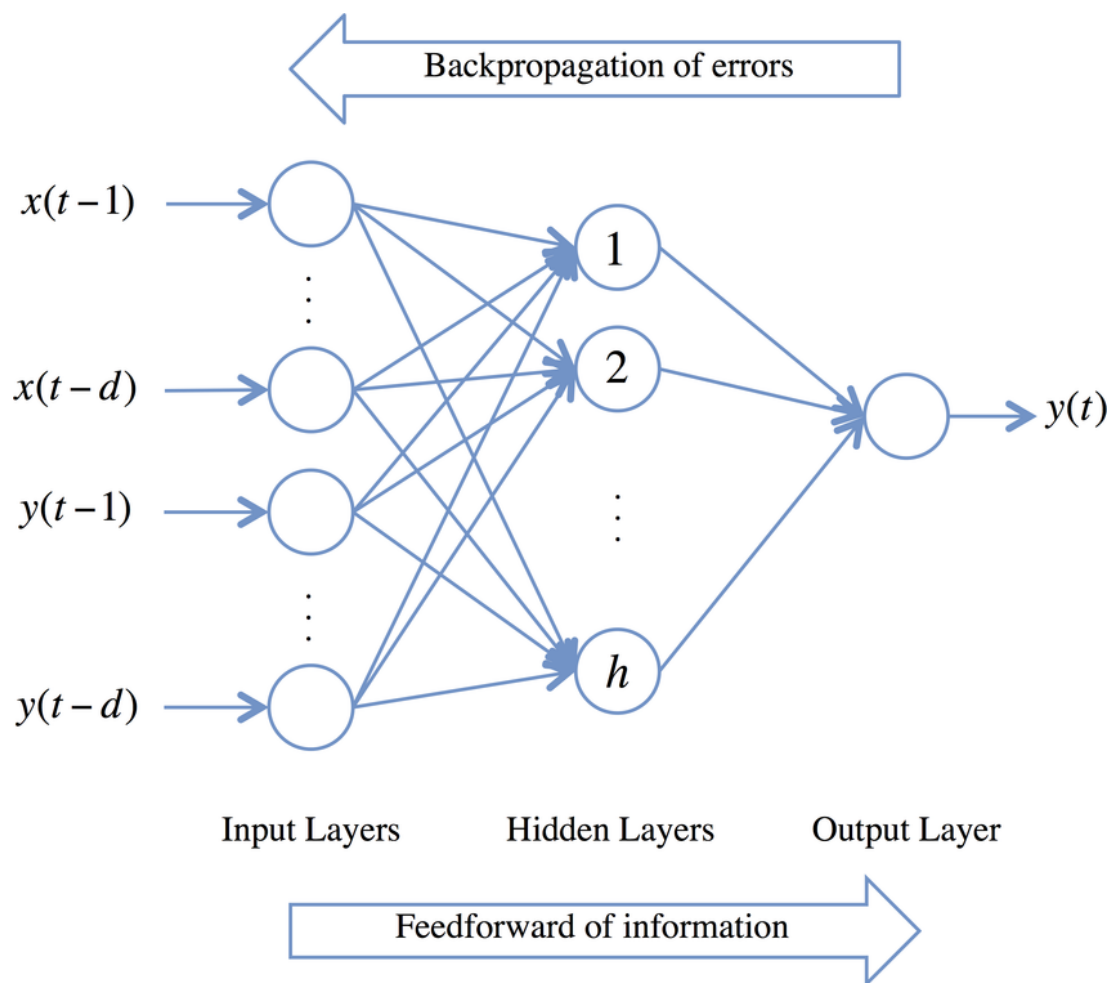


Figure 29 Feedforward Backpropagation Neural Network architecture.

Die z-Funktion erhält man, indem man die Gewichtungswerte der Eingangssignale multipliziert und mit dem Bias-Wert summiert. Wenn die Funktion z in die Aktivierungsfunktion eintritt, wird der Ausgabewert \hat{y} berechnet. Alle diese Prozesse werden als Vorwärtsausbreitung

bezeichnet. Um eine Rückweitergabe durchzuführen, wird der Fehlerwert des Modells berechnet, indem die Werte von \hat{y} und y (der tatsächliche Beschriftungswert der Daten) der L-Verlustfunktion unterworfen werden. Die Gewichts- und Verzerrungswerte werden aktualisiert, um den resultierenden Fehlerwert zu minimieren. Das generierte Modell vervollständigt den Lernprozess mit Vorwärts- und Rückwärtsausbreitungsalgorithmen (BEKİRYAZICI, 2020).

06. Aktivierungsfunktionen

In neuronalen Netzwerkmodellen werden Aktivierungsfunktionen verwendet, um die nichtlinearen Merkmale zu beschreiben, die in realen neuronalen Netzen vorhanden sind. In einem einfachen neuronalen Netzwerkmodell werden Eingangssignale mit ihrem Gewichtungswert multipliziert und mit dem Bias-Wert summiert und dann der Aktivierungsfunktion unterzogen. Der als Ergebnis dieses Prozesses erhaltene nichtlineare Ausgang wird als Lernergebnis des Modells oder als Eingangssignal einer anderen Schicht verwendet. Werden Aktivierungsfunktionen in neuronalen Netzmodellen nicht verwendet, erhalten Sie die Ausgangssignale als lineare Funktion. Lineare Funktionen schränken die Lernfähigkeit des Modells ein. Neuronale Netzmodelle erweitern die Lernfähigkeit für nichtlineare Situationen durch Aktivierungsfunktionen. Daher spielt die Aktivierungsfunktion eine wichtige Rolle in künstlichen neuronalen Netzen (BEKİRYAZICI, 2020).







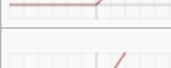


Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1 + e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
TanH		$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2 + 1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU) [2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU) [3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1 + e^x)$	$f'(x) = \frac{1}{1 + e^{-x}}$

Figure 30 Activation Functions

v. Convolutional Neural Network (CNN)

CNNs haben viele Vorteile gegenüber DNNs, einschließlich der Tatsache, dass sie eher einem menschlichen visuellen Verarbeitungssystem ähneln, für die Verarbeitung von 2D- und 3D-Bildern optimiert sind und effektiv beim Lernen und Extrahieren von Abstraktionen von 2D-Features sind. Die "Max-Pooling" -Schicht von CNNs absorbiert effektiv Formänderungen. Außerdem haben spärlich gekoppelte CNNs mit zugehörigen Gewichten deutlich weniger Parameter als ein vollständig verbundenes Netzwerk ähnlicher Größe. Zunächst einmal werden CNNs mit gradientenbasierten Lernalgorithmen trainiert und leiden weniger unter dem abnehmenden Gradientenproblem. Da der gradientenbasierte Algorithmus das gesamte Netzwerk trainiert, um ein Fehlerkriterium direkt zu minimieren, können CNNs hochoptimierte Gewichtungen erzeugen (Alom et al., 2019).

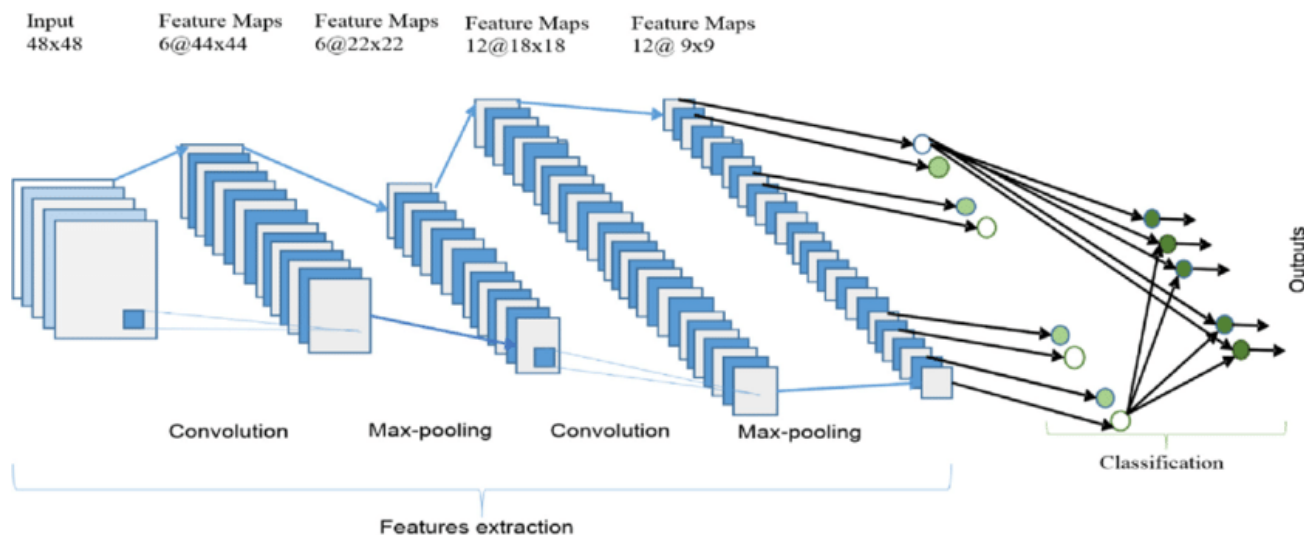


Figure 31 The overall architecture of the Convolutional Neural Network (CNN) includes an input layer, multiple alternating convolution and max-pooling layers, one fully-connected layer and one classification layer.

Bei Attributextraktionsschichten empfängt jede Schicht des Netzwerks die Ausgabe von der vorherigen Schicht als Eingabe und übergibt ihre Ausgabe als Eingabe an die nächste Schicht. Die Architektur von CNN besteht aus einer Kombination von drei Arten von Schichten: Faltung, "maximales Pooling" und Klassifizierung. Auf der unteren und mittleren Ebene des Netzwerks gibt es zwei Arten von Schichten: Faltungsschichten und maximale Pooling-Schichten. Doppelschichten sind für Falten und Einzelschichten für "maximale Pooling" -Operationen. Die Ausgabeknoten der Faltungs- und Maximalpooling-Layer werden in einer 2D-Ebene gruppiert, die als Feature-Mapping bezeichnet wird. Jede Ebene einer Ebene wird normalerweise aus der Kombination einer oder mehrerer Ebenen der vorherigen Ebenen abgeleitet. Die Knoten einer Ebene sind mit einem kleinen

Bereich jeder verbundenen Ebene der vorherigen Ebene verbunden. Jeder Knoten der Faltungsschicht extrahiert Attribute aus Eingabebildern mit Faltungsoperationen auf Eingabeknoten. Eigenschaften auf hoher Ebene werden von Eigenschaften abgeleitet, die von untergeordneten Ebenen ausgehen. Wenn sich Attribute an die höchste Ebene oder Ebene weitergeben, schrumpfen die Attributgrößen in Abhängigkeit von der Größe des Kernels für Faltungs- bzw. maximale Pooling-Operationen. Die Anzahl der Feature-Maps hat sich jedoch oft erhöht, um bessere Features von Eingabebildern darzustellen, um die Klassifizierungsgenauigkeit zu gewährleisten. Die Ausgabe der letzten Schicht von CNN wird als Eingabe eines vollständig verbundenen Netzwerks verwendet, das als Klassifizierungsschicht bezeichnet wird. Vorwärts gespeiste neuronale Netze wurden als Klassifizierungsschicht verwendet, da sie eine bessere Leistung aufweisen(Alom et al., 2019).

01. Convolutional Layer

In diesem Layer werden Feature-Karten aus vorherigen Layern mit erlernbaren Kernen zusammengerollt. Die Ausgabe der Kernel durchläuft lineare oder nichtlineare Aktivierungsfunktionen wie Sigmoid, hyperbolische Tangente, Softmax, rektifizierte lineare und Identitätsfunktionen, um Ausgabe-eigenschaftskarten zu erstellen. Jede der Ausgabe-eigenschaftszuordnungen kann mit mehr als einer Eingabeeigenschaftszuordnung kombiniert werden. Eingabekarten werden mit verschiedenen Kernen "transformiert", um entsprechende Ausgabekarten zu erstellen. Ausgabeabbildungen durchlaufen schließlich eine lineare oder nichtlineare Aktivierungsfunktion (z. B. Sigmoid-, hyperbolische Tangenten-, Softmax-, Gleichrichterlinear- oder Identitätsfunktionen)(Alom et al., 2019).

02. Sub-sampling Layer

Die Downsampling-Schicht führt ein Downsampling auf den Eingabekarten durch. Dies wird allgemein als Pooling-Ebene bezeichnet. Die Anzahl der Eingabe- und Ausgabeattributzuordnungen in diesem Layer ändert sich nicht. Aufgrund des Downsampling-Prozesses wird die Größe jeder Dimension der Ausgabekarten in Abhängigkeit von der Größe der Downsampling-Maske reduziert. Diese Ebene führt hauptsächlich zwei Arten von Vorgängen aus: durchschnittliches Pooling oder maximales Pooling. Beim durchschnittlichen Pooling-Ansatz sammelt die Funktion normalerweise den $N \times N$ -Patch der Eigenschaftszuordnungen im vorherigen Layer und wählt den Durchschnittswert aus. Bei maximalem Pooling hingegen wird der höchste Wert unter den $N \times N$ -Patches der Feature-

Maps ausgewählt. Daher schrumpfen die Abmessungen der Ausgabezuordnung im Laufe der Zeit. In einigen Sonderfällen wird jede Ausgabezuordnung mit einem Skalar multipliziert (Alom et al., 2019).

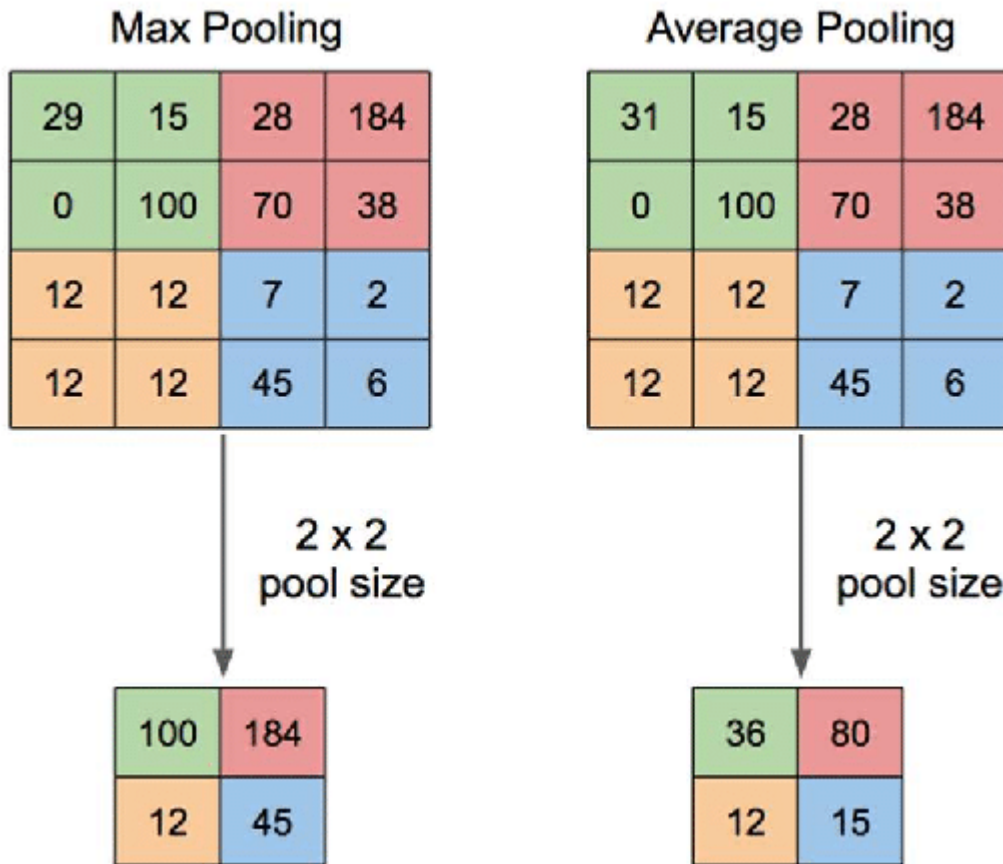


Figure 32 Illustration of Max Pooling and Average Pooling

03. Classification Layer

Dies ist die vollständig verknüpfte Ebene, die die Punktzahl jeder Klasse aus den Eigenschaften berechnet, die in den vorherigen Schritten aus einer Faltungsschicht extrahiert wurden. Endgültige Layer-Eigenschaftskarten werden als Vektoren mit skalaren Werten dargestellt, die an vollständig verbundene Layer übergeben werden. Vollständig gekoppelte vorwärts gespeiste Nervenschichten werden als Soft-Max-Klassifizierungsschicht verwendet. Bei der Rückwärtsausbreitung durch CNNs wird die vollständig verbundene Schicht nach dem allgemeinen Ansatz vollständig verbundener neuronaler Netze (FCNN) aktualisiert. Filter für Faltungs-Layer

werden aktualisiert, indem eine vollständige Faltung in Eigenschaftszuordnungen zwischen dem Faltungs-Layer und dem unmittelbar vorhergehenden Layer durchgeführt wird (Alom et al., 2019).

04. *Popular CNN Architectures*

- LeNet (1998)
- AlexNet (2012)
- ZFNet / Clarifai (2013)
- Network in Network (NiN)
- GoogLeNet (2014)
- VGGNET (2014)
- Residual Network (ResNet in 2015)
- Densely Connected Network (DenseNet)
- FractalNet (2016)

3. Feature Translation and Feedback

Nach dem Entfernen von Funktionen, die die Absicht des Benutzers widerspiegeln, besteht der nächste Schritt darin, diese Unterscheidungsmerkmale in Befehle umzuwandeln, die ein Gerät bedienen. Die in BCIs verwendeten Übersetzungsmethoden konvertieren extrahierte Features in Gerätesteuerebefehle. Bei diesen Befehlen kann es sich um diskrete Werte wie die Buchstabenauswahl oder fortlaufende Werte wie vertikale und horizontale Cursorbewegungen handeln. Übersetzungsmethoden in BCIs verwenden häufig maschinelle Lernansätze, um ein Modell aus den gesammelten Trainingsdaten zu trainieren. Translationsmethoden können grob als Klassifikations- und Regressionsmethoden klassifiziert werden (Li et al., 2009).

VI. KLASSIFIZIERUNG VON EEG-SIGNALEN DURCH ANN UND CNN

1. Material

In dieser Studie wurde ein Datensatz von der PhysioNet-Seite verwendet. Dieser Datensatz besteht aus über 1500 ein- und zweiminütigen EEG-Aufzeichnungen von 109 Freiwilligen.

2. Testprotokoll

Während die Probanden unterschiedliche motorische/bildgebende Aufgaben ausführten, wurde das 64-Kanal-EEG mit dem BCI2000-System (<http://www.bci2000.org>) aufgezeichnet. Jedes Probanden führte 14 experimentelle Durchläufe durch: zwei einminütige Grundlagenstudien (eine mit offenen Augen, eine mit geschlossenen Augen) und drei zweiminütige Durchläufe jeder der folgenden vier Aufgaben:

1. Auf der linken oder rechten Seite des Bildschirms wird ein Ziel angezeigt. Das Subjekt öffnet und schließt seine jeweilige Faust, bis das Ziel verschwindet. Dann entspannt sich das Thema.
2. Auf der linken oder rechten Seite des Bildschirms wird ein Ziel angezeigt. Das Subjekt stellt sich vor, seine jeweilige Faust zu öffnen und zu schließen, bis das Ziel verschwindet. Dann entspannt sich das Thema.
3. Ein Ziel wird oben oder unten auf dem Bildschirm angezeigt. Das Subjekt öffnet und schließt beide Fäuste (wenn das Ziel oben ist) oder beide Füße (wenn das Ziel unten ist), bis das Ziel verschwindet. Dann entspannt sich das Thema.
4. Ein Ziel wird oben oder unten auf dem Bildschirm angezeigt. Das Subjekt stellt sich vor, entweder beide Fäuste zu öffnen und zu schließen (wenn sich das Ziel oben befindet) oder beide Füße (wenn sich das Ziel unten befindet), bis das Ziel verschwindet. Dann entspannt sich das Thema.

Zusammenfassend umfassen experimentelle Studien:

1. Basic, Augen auf
2. Basic, Augen geschlossen
3. Aufgabe 1 (Öffnen und Schließen der linken oder rechten Faust)
4. Aufgabe 2 (Stellen Sie sich vor, Sie öffnen und schließen die linke oder rechte Faust)
5. Aufgabe 3 (beide Fäuste oder beide Füße öffnen und schließen)
6. Aufgabe 4 (Stellen Sie sich vor, Sie öffnen und schließen beide Fäuste oder beide Füße)
7. Aufgabe 1
8. Aufgabe 2
9. Aufgabe 3
10. Aufgabe 4

11. Aufgabe 1
12. Aufgabe 2
13. Aufgabe 3
14. Aufgabe 4

Die Daten werden hier im EDF+-Format bereitgestellt (jeweils 64 EEG-Signale, die in 160 Proben pro Sekunde abgetastet wurden, und einen Erklärungskanal). Für die Verwendung mit der PhysioToolkit-Software hat rdedfann für jede Registrierung eine separate PhysioBank-kompatible Beschreibungsdatei (mit einem .event-Anhang) erstellt. Die EVENT-Dateien und Kommentarkanäle in den entsprechenden EDF-Dateien enthalten dieselben Daten.

Jeder Kommentar enthält einen von drei Codes (T0, T1 oder T2):

- T0 entspricht Rest
- T1 entspricht dem Beginn der Bewegung (real oder imaginär).
 1. linker Punch (in den Läufen 3, 4, 7, 8, 11 und 12)
 2. beide Schläge (in den Läufen 5, 6, 9, 10, 13 und 14)
- T2 entspricht dem Beginn der Bewegung (real oder imaginär).
 1. Rechter Punch (in den Läufen 3, 4, 7, 8, 11 und 12)
 2. beide Füße (in den Läufen 5, 6, 9, 10, 13 und 14)

EEGs wurden von 64 Elektroden nach dem internationalen 10-10-System aufgezeichnet (ohne Nz, F9, F10, FT9, FT10, A1, A2, TP9, TP10, P9 und P10 Elektroden)(Goldberger et al., 2000).

3. Vorläufig

Da der Datensatz für die durchzuführende Studie zu groß ist, wurden in dieser Studie nur die erste und zweite Aufgabe für die Klassifizierung verwendet. Zu diesen Aufgaben gehören das Öffnen der Augen und das Schließen der Augen. In der ersten Studie wurden 16 Dateien von 8 Probanden verwendet. Später wurde die Zahl der untersuchten Probanden auf 20 erhöht. So soll der Effekt des Anstiegs der Stichprobenzahl auf das Training der Systeme gemessen werden.

Da die Dateien im heruntergeladenen Datensatz die Erweiterung (.edf) haben, wurden diese Dateien durch zusätzliche Software in das Format (.csv) konvertiert. Zur Durchführung dieses Vorgangs wurde die Software 'EDF BROWSER' verwendet. Dieses Programm, das viele Funktionen enthält, wird auch für die Verarbeitung von EEG-Signalen verwendet. Die Tags der heruntergeladenen Dateien müssen bearbeitet werden, damit die Dateierweiterungen geändert werden können.

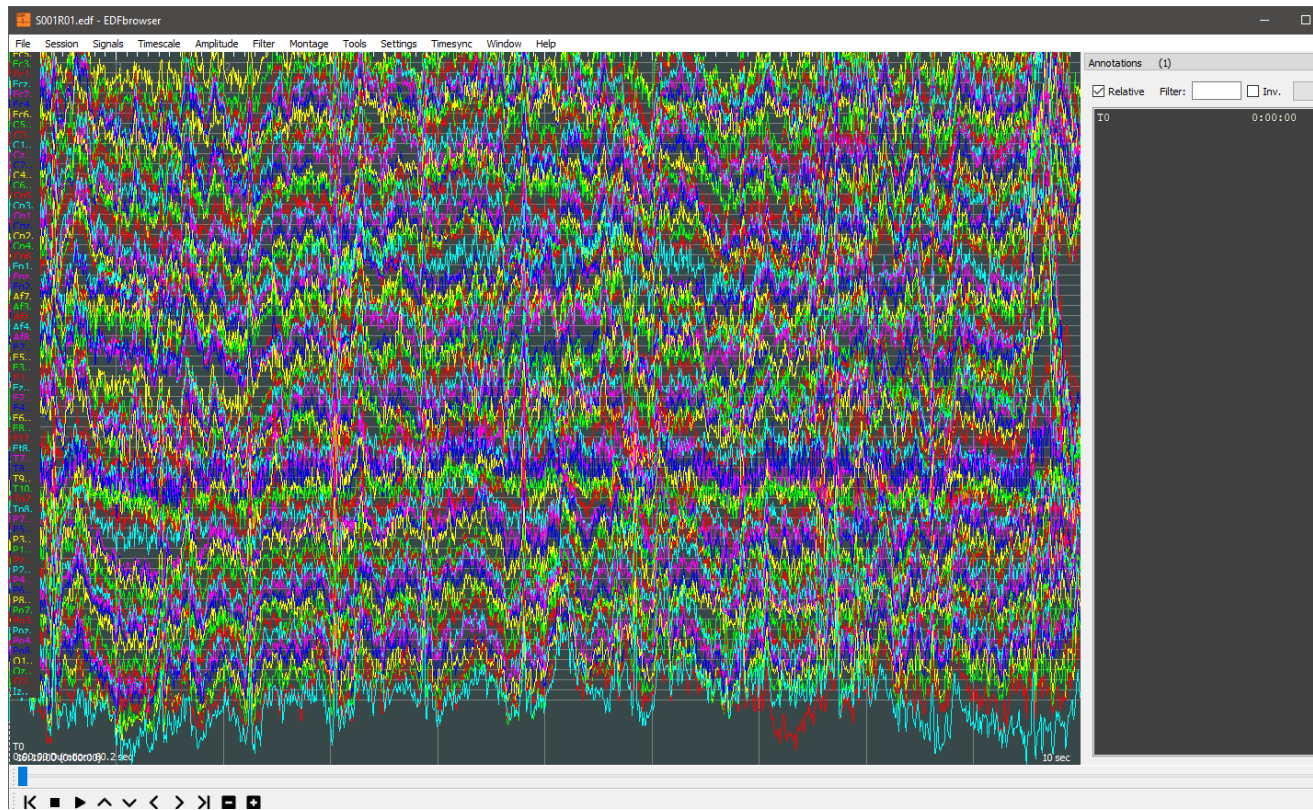


Figure 33 Eine Signalmessdatei, die an das EDF-Browserprogramm angehängt ist

Im Programm werden, wie im Bild gezeigt, alle Elektroden separat ausgedrückt und die Signale in jeder Elektrode werden für den aktuellen Zeitraum angezeigt.

Nach dieser Phase wurde die Jupyter-Schnittstelle über die IDE "ANACONDA" ausgeführt.

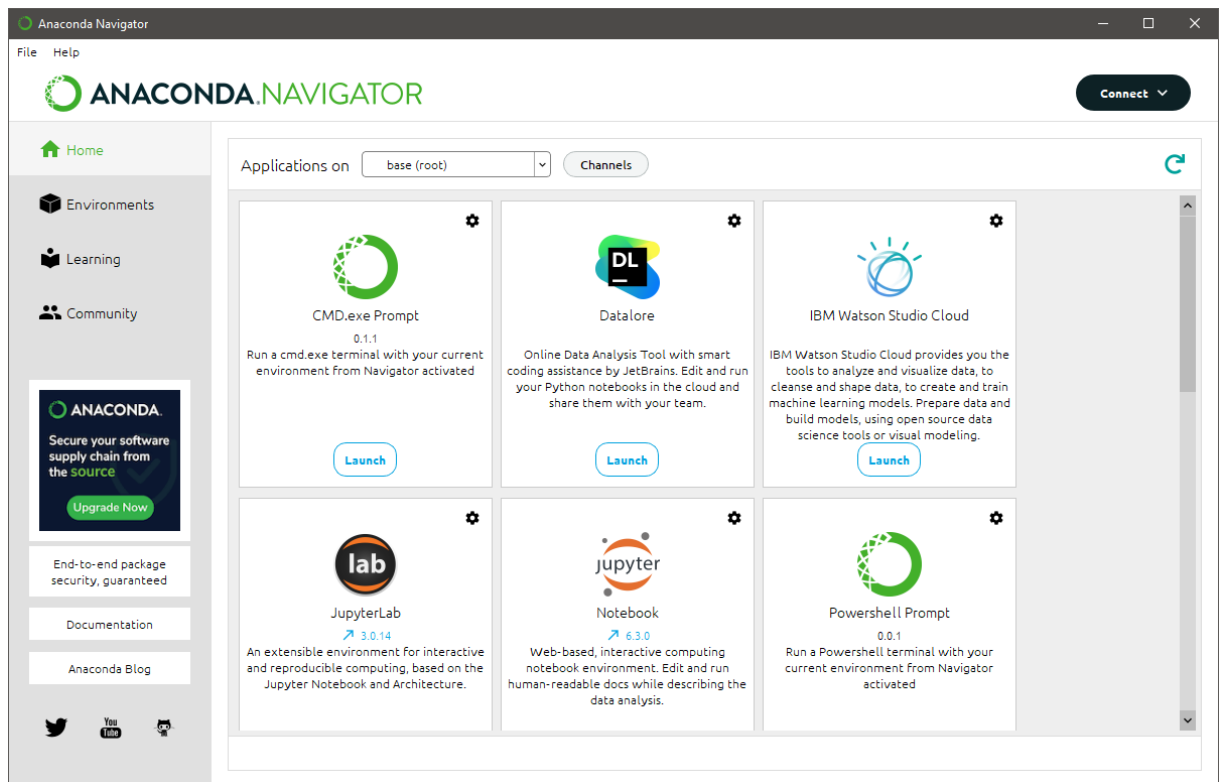


Figure 34 Anaconda Navigator

4. Systeminstallation

i. Hinzufügen von Bibliotheken

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as scs
import re
from numpy import genfromtxt
import tensorflow as tf
import keras
from sklearn.model_selection import KFold
from keras.utils import to_categorical
from keras.layers import LSTM, Conv1D, MaxPooling1D, Flatten, Embedding, GlobalAveragePooling1D
from tensorflow.keras.constraints import max_norm
from keras.models import Sequential
from keras.layers import Dense, Flatten, Conv2D, Dropout, BatchNormalization, MaxPooling2D, Activation, SpatialDropout1D, G
from keras.losses import sparse_categorical_crossentropy
from keras.optimizers import Adam
```

ii. Hinzufügen von Datensätzen und Labels zum System

```
In [2]: data1 = pd.read_csv('C:/Users/AzamaoZ/Desktop/WIFO/Subjects_EYE_open_closed/csv/S001R01_data.csv')
data2 = pd.read_csv('C:/Users/AzamaoZ/Desktop/WIFO/Subjects_EYE_open_closed/csv/S001R02_data.csv')
data3 = pd.read_csv('C:/Users/AzamaoZ/Desktop/WIFO/Subjects_EYE_open_closed/csv/S002R01_data.csv')
data4 = pd.read_csv('C:/Users/AzamaoZ/Desktop/WIFO/Subjects_EYE_open_closed/csv/S002R02_data.csv')
data5 = pd.read_csv('C:/Users/AzamaoZ/Desktop/WIFO/Subjects_EYE_open_closed/csv/S003R01_data.csv')
data6 = pd.read_csv('C:/Users/AzamaoZ/Desktop/WIFO/Subjects_EYE_open_closed/csv/S003R02_data.csv')
data7 = pd.read_csv('C:/Users/AzamaoZ/Desktop/WIFO/Subjects_EYE_open_closed/csv/S004R01_data.csv')
data8 = pd.read_csv('C:/Users/AzamaoZ/Desktop/WIFO/Subjects_EYE_open_closed/csv/S004R02_data.csv')
data9 = pd.read_csv('C:/Users/AzamaoZ/Desktop/WIFO/Subjects_EYE_open_closed/csv/S005R01_data.csv')
data10 = pd.read_csv('C:/Users/AzamaoZ/Desktop/WIFO/Subjects_EYE_open_closed/csv/S005R02_data.csv')
data11 = pd.read_csv('C:/Users/AzamaoZ/Desktop/WIFO/Subjects_EYE_open_closed/csv/S006R01_data.csv')
data12 = pd.read_csv('C:/Users/AzamaoZ/Desktop/WIFO/Subjects_EYE_open_closed/csv/S006R02_data.csv')
data13 = pd.read_csv('C:/Users/AzamaoZ/Desktop/WIFO/Subjects_EYE_open_closed/csv/S007R01_data.csv')
data14 = pd.read_csv('C:/Users/AzamaoZ/Desktop/WIFO/Subjects_EYE_open_closed/csv/S007R02_data.csv')
data15 = pd.read_csv('C:/Users/AzamaoZ/Desktop/WIFO/Subjects_EYE_open_closed/csv/S008R01_data.csv')
data16 = pd.read_csv('C:/Users/AzamaoZ/Desktop/WIFO/Subjects_EYE_open_closed/csv/S008R02_data.csv')

labels = [1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0]
```

iii. Bearbeiten von Daten

```
In [3]: data1 = data1.to_numpy()
data2 = data2.to_numpy()
data3 = data3.to_numpy()
data4 = data4.to_numpy()
data5 = data5.to_numpy()
data6 = data6.to_numpy()
data7 = data7.to_numpy()
data8 = data8.to_numpy()
data9 = data9.to_numpy()
data10 = data10.to_numpy()
data11 = data11.to_numpy()
data12 = data12.to_numpy()
data13 = data13.to_numpy()
data14 = data14.to_numpy()
data15 = data15.to_numpy()
data16 = data16.to_numpy()
```

```
In [12]: data1 = np.expand_dims(data1, axis=0)
data2 = np.expand_dims(data2, axis=0)
data3 = np.expand_dims(data3, axis=0)
data4 = np.expand_dims(data4, axis=0)
data5 = np.expand_dims(data5, axis=0)
data6 = np.expand_dims(data6, axis=0)
data7 = np.expand_dims(data7, axis=0)
data8 = np.expand_dims(data8, axis=0)
data9 = np.expand_dims(data9, axis=0)
data10 = np.expand_dims(data10, axis=0)
data11 = np.expand_dims(data11, axis=0)
data12 = np.expand_dims(data12, axis=0)
data13 = np.expand_dims(data13, axis=0)
data14 = np.expand_dims(data14, axis=0)
data15 = np.expand_dims(data15, axis=0)
data16 = np.expand_dims(data16, axis=0)
```

```

In [13]: data1.shape
Out[13]: (1, 9760, 65)

In [15]: data_all = np.concatenate((data1, data2, data3, data4, data5, data6, data7, data8, data9, data10, data11, data12,
                                     data13, data14, data15, data16),axis=0)

In [16]: data_all.shape
Out[16]: (16, 9760, 65)

In [17]: labels = np.array(labels)

```

iv. Installation des ANN-Modells und der Ergebnisse

```

In [18]: model = Sequential()

model.add(tf.keras.layers.Flatten())
model.add(tf.keras.layers.Dense(800, activation = "relu"))
model.add(tf.keras.layers.Dense(500, activation = "relu"))
model.add(tf.keras.layers.Dense(500, activation = "relu"))
model.add(tf.keras.layers.Dense(500, activation = "relu"))
model.add(tf.keras.layers.Dense(10, activation = "relu"))
model.add(tf.keras.layers.Dense(2, activation = "softmax"))

optimizer = keras.optimizers.Adam(learning_rate = 1e-4)
model.compile(optimizer = optimizer, loss = "sparse_categorical_crossentropy", metrics = ["accuracy"])

model_history = model.fit(data_all, labels, batch_size = 16, validation_split = 0.2, epochs = 50)

```

1/1 [=====] - 0s 121ms/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 237.1502 - val_accuracy: 0.5000
 Epoch 45/50
 1/1 [=====] - 0s 135ms/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 238.1963 - val_accuracy: 0.5000
 Epoch 46/50
 1/1 [=====] - 0s 127ms/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 238.9671 - val_accuracy: 0.5000
 Epoch 47/50
 1/1 [=====] - 0s 132ms/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 239.6689 - val_accuracy: 0.5000
 Epoch 48/50
 1/1 [=====] - 0s 127ms/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 240.3099 - val_accuracy: 0.5000
 Epoch 49/50
 1/1 [=====] - 0s 128ms/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 240.8952 - val_accuracy: 0.5000
 Epoch 50/50
 1/1 [=====] - 0s 129ms/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 241.4278 - val_accuracy: 0.5000

```
In [21]: model.summary()
```

Model: "sequential_1"

Layer (type)	Output Shape	Param #
flatten_1 (Flatten)	(None, 634400)	0
dense_6 (Dense)	(None, 800)	507520800
dense_7 (Dense)	(None, 500)	400500
dense_8 (Dense)	(None, 500)	250500
dense_9 (Dense)	(None, 500)	250500
dense_10 (Dense)	(None, 10)	5010
dense_11 (Dense)	(None, 2)	22
Total params: 508,427,332		
Trainable params: 508,427,332		
Non-trainable params: 0		

v. Neuaufbau des ANN-Modells mit Trennung von Test- und Trainingsdaten

```
In [22]: from sklearn.model_selection import train_test_split
```

```
x_train, x_test, y_train, y_test = train_test_split(data_all, labels, test_size = 0.2)
```

```
In [24]: model1 = Sequential()
```

```
model1.add(tf.keras.layers.Flatten())
model1.add(tf.keras.layers.Dense(800, activation = "relu"))
model1.add(tf.keras.layers.Dense(500, activation = "relu"))
model1.add(tf.keras.layers.Dense(500, activation = "relu"))
model1.add(tf.keras.layers.Dense(500, activation = "relu"))
model1.add(tf.keras.layers.Dense(10, activation = "relu"))
model1.add(tf.keras.layers.Dense(2, activation = "softmax"))
```

```
optimizer = keras.optimizers.Adam(learning_rate = 1e-4)
```

```
model1.compile(optimizer = optimizer, loss = "sparse_categorical_crossentropy", metrics = ["accuracy"])
```

```
model1_history = model1.fit(x_train, y_train, batch_size = 16, validation_data = (x_test, y_test), epochs =
```

```
000
```

```
Epoch 47/50
```

```
1/1 [=====] - 0s 127ms/step - loss: 0.0576 - accuracy: 1.0000 - val_loss: 5.2622 - val_accuracy:
```

```
000
```

```
Epoch 48/50
```

```
1/1 [=====] - 0s 128ms/step - loss: 0.0576 - accuracy: 1.0000 - val_loss: 5.2931 - val_accuracy:
```

```
000
```

```
Epoch 49/50
```

```
1/1 [=====] - 0s 127ms/step - loss: 0.0576 - accuracy: 1.0000 - val_loss: 5.3211 - val_accuracy:
```

```
000
```

```
Epoch 50/50
```

```
1/1 [=====] - 0s 133ms/step - loss: 0.0576 - accuracy: 1.0000 - val_loss: 5.3464 - val_accuracy:
```

```
000
```

vi. Etablierung des CNN-Modells

```
In [28]: model_cnn = Sequential()

model_cnn.add(tf.keras.layers.Conv1D(filters = 16, input_shape = (9760, 65), kernel_size = 20, activation = "relu"))
model_cnn.add(tf.keras.layers.BatchNormalization())
model_cnn.add(tf.keras.layers.MaxPooling1D(pool_size = 2))
model_cnn.add(tf.keras.layers.Dropout(0.2))

model_cnn.add(tf.keras.layers.Conv1D(filters = 32, kernel_size = 20, activation = "relu"))
model_cnn.add(tf.keras.layers.BatchNormalization())
model_cnn.add(tf.keras.layers.MaxPooling1D(pool_size = 2))
model_cnn.add(tf.keras.layers.Dropout(0.2))

model_cnn.add(tf.keras.layers.Conv1D(filters = 64, kernel_size = 20, activation = "relu"))
model_cnn.add(tf.keras.layers.BatchNormalization())
model_cnn.add(tf.keras.layers.Dropout(0.2))

model_cnn.add(tf.keras.layers.Flatten())
model_cnn.add(tf.keras.layers.Dense(10, activation = "relu"))
model_cnn.add(tf.keras.layers.BatchNormalization())
model_cnn.add(tf.keras.layers.Dropout(0.2))
model_cnn.add(tf.keras.layers.Dense(2, activation = "softmax"))

callback = tf.keras.callbacks.EarlyStopping(monitor = "val_loss", patience = 20, restore_best_weights = True)
optimizer = keras.optimizers.Adam(learning_rate = 1e-4)
model_cnn.compile(optimizer = optimizer, loss = "sparse_categorical_crossentropy", metrics = ["accuracy"])

model_cnn_history = model_cnn.fit(x_train, y_train, batch_size = 4, validation_data = (x_test, y_test), epochs = 100)
```

```
Epoch 95/100
3/3 [=====] - 0s 153ms/step - loss: 0.0739 - accuracy: 1.0000 - val_loss: 0.7582 - val_accuracy: 0.0000
Epoch 96/100
3/3 [=====] - 0s 149ms/step - loss: 0.1264 - accuracy: 1.0000 - val_loss: 0.7645 - val_accuracy: 0.0000
Epoch 97/100
3/3 [=====] - 0s 148ms/step - loss: 0.1075 - accuracy: 1.0000 - val_loss: 0.7704 - val_accuracy: 0.0000
Epoch 98/100
3/3 [=====] - 0s 156ms/step - loss: 0.0843 - accuracy: 1.0000 - val_loss: 0.7768 - val_accuracy: 0.0000
Epoch 99/100
3/3 [=====] - 0s 141ms/step - loss: 0.1256 - accuracy: 1.0000 - val_loss: 0.7893 - val_accuracy: 0.0000
Epoch 100/100
3/3 [=====] - 0s 146ms/step - loss: 0.1334 - accuracy: 1.0000 - val_loss: 0.8034 - val_accuracy: 0.0000
```

```
: model_cnn.summary()
```

```
Model: "sequential_3"
```

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 9741, 16)	20816
batch_normalization (Batch Normalization)	(None, 9741, 16)	64
max_pooling1d (MaxPooling1D)	(None, 4870, 16)	0
dropout (Dropout)	(None, 4870, 16)	0
conv1d_1 (Conv1D)	(None, 4851, 32)	10272
batch_normalization_1 (Batch Normalization)	(None, 4851, 32)	128
max_pooling1d_1 (MaxPooling1D)	(None, 2425, 32)	0
dropout_1 (Dropout)	(None, 2425, 32)	0
conv1d_2 (Conv1D)	(None, 2406, 64)	41024
batch_normalization_2 (Batch Normalization)	(None, 2406, 64)	256
dropout_2 (Dropout)	(None, 2406, 64)	0
flatten_3 (Flatten)	(None, 153984)	0
dense_18 (Dense)	(None, 10)	1539850
batch_normalization_3 (Batch Normalization)	(None, 10)	40
dropout_3 (Dropout)	(None, 10)	0
dense_19 (Dense)	(None, 2)	22
Total params: 1,612,472		
Trainable params: 1,612,228		
Non-trainable params: 244		

vii. Neuinstallation von Modellen mit neuen Parametern

Die Anzahl der Probanden wurde von 8 auf 20 erhöht, um festzustellen, ob die Modelle mit mehr Daten funktionierten. Darüber hinaus wurden die Anzahl der ausgeblendeten Ebenen der Modelle geändert, um die Auswirkungen auf die Leistung zu sehen.


```

model1 = Sequential()

model1.add(tf.keras.layers.Flatten())
model1.add(tf.keras.layers.Dense(800, activation = "relu"))
model1.add(tf.keras.layers.Dense(500, activation = "relu"))
model1.add(tf.keras.layers.Dense(500, activation = "relu"))
model1.add(tf.keras.layers.Dense(500, activation = "relu"))
model1.add(tf.keras.layers.Dense(10, activation = "relu"))
model1.add(tf.keras.layers.Dense(2, activation = "softmax"))

optimizer = keras.optimizers.Adam(learning_rate = 1e-4)
model1.compile(optimizer = optimizer, loss = "sparse_categorical_crossentropy", metrics = ["accuracy"])
model1_history = model1.fit(x_train, y_train, batch_size = 16, validation_data = (x_test, y_test), epochs = 50)

y: 0.5000
Epoch 45/50
2/2 [=====] - 2s 1s/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 20.9029 - val_accuracy: 0.5000
Epoch 46/50
2/2 [=====] - 2s 997ms/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 20.9029 - val_accuracy: 0.5000
Epoch 47/50
2/2 [=====] - 2s 1s/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 20.9030 - val_accuracy: 0.5000
Epoch 48/50
2/2 [=====] - 2s 1s/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 20.9031 - val_accuracy: 0.5000
Epoch 49/50
2/2 [=====] - 2s 990ms/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 20.9032 - val_accuracy: 0.5000
Epoch 50/50
2/2 [=====] - 2s 1s/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 20.9032 - val_accuracy: 0.5000

```

x. Ausführen des CNN-Modells mit mehr Daten

```

model_cnn_history = model_cnn.fit(x_train, y_train, batch_size = 4, validation_data = (x_test, y_test), epochs = 100)

Epoch 95/100
8/8 [=====] - 1s 185ms/step - loss: 0.1076 - accuracy: 1.0000 - val_loss: 0.9996 - val_accuracy: 0.5000
Epoch 96/100
8/8 [=====] - 2s 201ms/step - loss: 0.0494 - accuracy: 1.0000 - val_loss: 1.0062 - val_accuracy: 0.5000
Epoch 97/100
8/8 [=====] - 2s 213ms/step - loss: 0.1617 - accuracy: 0.9375 - val_loss: 1.0118 - val_accuracy: 0.5000
Epoch 98/100
8/8 [=====] - 2s 214ms/step - loss: 0.1663 - accuracy: 0.8750 - val_loss: 1.0143 - val_accuracy: 0.5000
Epoch 99/100
8/8 [=====] - 2s 191ms/step - loss: 0.1802 - accuracy: 0.9375 - val_loss: 1.0185 - val_accuracy: 0.5000
Epoch 100/100
8/8 [=====] - 1s 181ms/step - loss: 0.1157 - accuracy: 0.9688 - val_loss: 1.0192 - val_accuracy: 0.5000

```


VII. FAZIT

Die Leistungen der Systeme unter verschiedenen Bedingungen wurden untersucht.

Zuerst wurden die Daten für das ANN-Modell wie gewohnt aufgeteilt, gefolgt von geteilten Trainings- und Testdaten. Bei der Untersuchung der Ergebnisse zeigte sich, dass der Validitätsverlust deutlich reduziert wurde. Andere Ergebnisse sind ähnlich.

Das nächste Ziel ist es, ANN- und CNN-Systeme für 8 Probanden zu vergleichen. Es wurde erwartet, dass das CNN-Modell bei Signalen besser funktioniert. Der Vorteil des CNN-Modells besteht darin, dass es einen geringeren Gültigkeitsverlust aufweist. Die Gültigkeitsüberprüfung blieb bei 0,5 für die ersten drei Modelle. Für ein 2-Klassen-Problem sind diese Ergebnisse unzureichend.

Die Anzahl der Probanden wurde auf 20 erhöht, so dass der Prozentsatz der Gültigkeitsüberprüfung erhöht werden kann. In dem Experiment, in dem die Zwischenschichtdichte und die Trennung der Daten verglichen wurden, zeigte sich, dass der Erfolg des ANN-Modells mit einer Zunahme der Anzahl der Schichten signifikant zunahm. Die Erhöhung der Daten allein, ohne die Schichtdichte zu ändern, führte zu keiner signifikanten Verbesserung.

Schließlich wurde das CNN-Modell mit hohen Daten gemessen. Das Modell, dessen andere Parameter nicht geändert wurden, lieferte eine höhere Validierungsvalidierung als in seiner ursprünglichen Form. Diese Ebene war jedoch nicht so hoch wie der Wert, den das ANN-Modell mit mehr Zwischenschichten bietet. Frühere Ergebnisse deuten darauf hin, dass eine erfolgreichere Klassifizierung durch Änderung der Dichte in der ANN-Schicht (Fully Connected) des CNN-Modells erfolgen kann.

Die Genauigkeit bei allen Modellen wird zu 100% erreicht. Dieser Wert weist jedoch auf übermäßiges Lernen hin, wenn die Validierung keinen sinnvollen Wert erreicht. Dies bedeutet, dass neue Daten nicht gut klassifiziert werden können.

Wenn man die Anzahl der Parameter der installierten Modelle berücksichtigt, sieht man, dass CNN-Modelle mit viel niedrigeren Parametern arbeiten. Wenn also die Verwendung von CNN-installierten Modellen in BCI-Systemen ausreichend erfolgreich ist, kann davon ausgegangen werden, dass dies in Bezug auf den Aufwand viel vorteilhafter sein kann.

VIII. BIBLIOGRAPHIE

1. Abilzade, L. (tarih yok). CLASSIFICATION OF THE MOTOR EEG SIGNALS BY USING DEEP NEURAL NETWORKS.
2. <https://doi.org/10.3390/s21196343> adresinden alındı
3. <https://mayfieldclinic.com/pe-anatbrain.htm> adresinden alındı
4. www.mdpi.com/journal/sensors adresinden alındı
5. www.elsevier.com/locate/clinphCLINPH2001764 adresinden alındı
6. Buyukgoze, S. (tarih yok). BEYİN BİLGİSAYAR ARAYÜZLERİ ve UYGULAMA ALANLARI Algoritma View project.
<https://www.researchgate.net/publication/350581236> adresinden alındı
7. <https://imotions.com/blog/eeg/> adresinden alındı
8. <https://physionet.org/content/eegmmidb/1.0.0/> adresinden alındı
9. <http://hdl.handle.net/11452/15172> adresinden alındı
10. Goodfellow, I., Bengio, Y., & Courville, A. (tarih yok). Deep Learning.
11. Graimann, B., Allison, B., & Pfurtscheller, G. (2010). BRAIN–COMPUTER INTERFACES. Springer. <https://link.springer.com/content/pdf/10.1007/978-3-642-02091-9.pdf> adresinden alındı
12. Mühendisliği, B., Dalı Bilgisayar, A., Tezli, M., & Lisans, Y. (tarih yok). FATİH SULTAN MEHMET VAKIF ÜNİVERSİTESİ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ ÖZNİTELİK ÇIKARIM VE EVRİMSEL ÖZNİTELİK SEÇİM METOTLARININ EEG SİNYALLERİNİN SINIFLANDIRMA BAŞARISINA ETKİLERİ YÜKSEK LİSANS TEZİ Ferda ABBASOĞLU.
13. <https://www.simplypsychology.org/neuron.html> adresinden alındı
14. Thomas, J., Maszczyk, T., Sinha, N., Kluge, T., & Dauwels, J. (2017). Deep Learning-based Classification for Brain-Computer Interfaces.
15. <https://www.moleculardevices.com/applications/patch-clamp-electrophysiology/what-action-potential> adresinden alındı
16. <https://imotions.com/blog/what-is-eeg/> adresinden alındı