Universitatea POLITEHNICA din București
Facultatea de Electronică, Telecomunicații și Tehnologia Informației

Sistem de clasificare a semnalelor EEG folosind rețele neurale adânci

Lucrare de licență

Prezentată ca cerință parțială pentru obținerea
titlului de *Inginer*
în domeniul *Electronică, Telecomunicații și Tehnologia Informației*
programul de studii *Electronică Aplicată*

Conducător științific
Prof. Dr. Ing. Corneliu Burileanu
As. Univ. Drd. Ing. Ana Neacșu

Absolvent
Maria-Cristina Andronache

Anul 2020

Declaraţie de onestitate academică

Prin prezenta declar că lucrarea cu titlul *Sistem de clasificare a semnalelor EEG folosind rețele neurale adânci*, prezentată în cadrul Facultăţii de Electronică, Telecomunicaţii şi Tehnologia Informaţiei a Universităţii "Politehnica" din Bucureşti ca cerinţă parţială pentru obţinerea titlului de *Inginer* în domeniul Inginerie Electronică şi Telecomunicaţii/ Calculatoare şi Tehnologia Informaţiei, programul de studii *Electronică Aplicată* este scrisă de mine şi nu a mai fost prezentată niciodată la o facultate sau instituţie de învăţământ superior din ţară sau străinătate.

Declar că toate sursele utilizate, inclusiv cele de pe Internet, sunt indicate în lucrare, ca referinţe bibliografice. Fragmentele de text din alte surse, reproduse exact, chiar şi în traducere proprie din altă limbă, sunt scrise între ghilimele şi fac referinţă la sursă. Reformularea în cuvinte proprii a textelor scrise de către alţi autori face referinţă la sursă. Înţeleg că plagiatul constituie infracţiune şi se sancţionează conform legilor în vigoare.

Declar că toate rezultatele simulărilor, experimentelor şi măsurătorilor pe care le prezint ca fiind făcute de mine, precum şi metodele prin care au fost obţinute, sunt reale şi provin din respectivele simulări, experimente şi măsurători. Înţeleg că falsificarea datelor şi rezultatelor constituie fraudă şi se sancţionează conform regulamentelor în vigoare.

Bucureşti, Iunie 2020.
Absolvent: Maria-Cristina Andronache

.....

Universitatea "Politehnica" din Bucureşti
Facultatea de Electronică, Telecomunicaţii şi Tehnologia Informaţiei
Departamentul **EAII**

Anexa 1

TEMA PROIECTULUI DE DIPLOMĂ

a studentului **ANDRONACHE N. Maria-Cristina , 441B-ELA.**

1. Titlul temei: Sistem de clasificare a semnalelor EEG folosind rețele neurale adânci

2. Descrierea contribuţiei originale a studentului (în afara părţii de documentare) și specificații de proiectare:
Scopul acestei teze este clasificarea semnalelor de tip EEG de la un subiect folosind rețele neurale adânci. Semnalele vor fi captate cu ajutorul unei căști cu electrozi (Neuroscan). Pentru acest proiect se vor analiza semnalele obținute de la 10 senzori plasați în zona lobului occipital. Baza de date va conține semnale achiziţionate în timp ce subiectul privește un anumit stimul. Vor exista 2 stimuli care sunt prezentaţi simultan pe un ecran, iar subiectul va fi instruit căruia să îi acorde atenție (mai precis, pe un ecran negru vor fi 2 dreptunghiuri colorate diferit care pâlpâie simultan cu frecvenţe diferite). Obiectivele principale ale acestei lucrări includ:
• realizarea unei paradigme experimentale - datorită naturii semnalului generat în aceste condiții este important să am o rezoluţie de 1 ms pentru stimuli
• achiziția unei baze de date – datele vor fi achiziționate în mai multe sesiuni pentru a evita obosirea subiectului.
• preprocesarea semnalului – se vor utiliza diverse tehnici matematice de extragere a trăsăturilor
• dezvoltarea și testarea a diverse arhitecturi de reţele neurale pentru a obţine o acurateţe de clasificare cât mai bună.

3. Resurse folosite la dezvoltarea proiectului:
NeuroScan, Curry7, Matlab, eeglab, Python, Tensorflow, Keras, ecran 4k Samsung, staţii de lucru

4. Proiectul se bazează pe cunoştinţe dobîndite în principal la următoarele 3-4 discipline:
Electronică şi informatică medicală, Decizie şi estimare în prelucrarea informaţiilor, Imagistică me

5. Proprietatea intelectuală asupra proiectului aparţine: U.P.B.

6. Data înregistrării temei: 2019-11-24 12:49:04

Conducător(i) lucrare,
Prof. dr. ing. Corneliu BURILEANU

Student,

NEACȘU Ana-Antonia

Director departament,
Prof. dr. ing Sever PAŞCA

Decan,
Prof. dr. ing. Mihnea UDREA

Cod Validare: **c8e3d027c7**

Cuprins

Lista figurilor iii

Lista tabelelor iv
Lista acronimelor v

Introducere 1
Motivaţie 1
Obiective 2
Aplicabilitate 2

1. Noţiuni teoretice 5
1.1. EEG - Aspecte Generale 5
1.2. Starea artei 8
1.3. Rețele neurale – noțiuni introductive 11

2. Setup experimental 17
2.1. NeuroScan 17
2.2. Curry 7 18
2.3. Paradigma experimentală 19

3. Metoda propusă 21
3.1. Schema generală a proiectului 21
3.2. Algoritmul CCA 23
3.3. Rezultate 24

4. Concluzii 33
4.1. Concluzii generale 33
4.2. Contribuții personale 34
4.3. Dezvoltări ulterioare 35
Bibliografie 37

Lista figurilor

1.1. *Hans Berger în anii 1930* 5
1.2. *Prima înregistrare EEG la om; primul semnal reprezintă semnalul EEG iar cel de al doilea este un semnal de sincronizare* 6
1.3. *Structura unui neuron* 7
1.4. *Potenţialul membranei* 8
1.5. *Paradigma experimentală pentru reabilitare* 9
1.6. *Modele simplificate ale formelor de undă* 10
1.7. *P300 speller* 11
1.8. *Arhitectura generală pentru o reţea neurală* 13
1.9. *Sigmoid* 13
1.10. *Tanh* 14
1.11. *Arctan* 14
1.12. *ReLU* 15
1.13. *Leaky ReLU* 15
2.1. *Casca cu electrozi* 17
2.2. *Amplificator Neuvo* 17
2.3. *Curry7* 18
2.4. *Lob occipital* 18
2.5. *Configurare Curry 7* 19
2.6. *Distribuție electrozi* 20
2.7. *Reprezentarea grafică a impedanței de contact* 20
3.1. *Succesiunea de etape din cadrul proiectului* 21
3.2. *SGD fără impuls la stânga şi cu impuls la dreapta* 25
3.3. *Ilustrarea fenomenelor de underfitting și overfitting* 28

Lista tabelelor

1.1. Distribuția pe frecvențe a semnalelor EEG 6
3.1. Distribuţia înregistrărilor în funcţie de stimul 22
3.2. Rezultate–tanh (a) 26
3.3. Rezultate–tanh (b) 26
3.4. Rezultate–tanh (c) 27
3.5. Rezultate–tanh (d) 27
3.6. Rezultate–tanh (e) 28
3.7. Rezultate–tanh (f) 28
3.8. Rezultate–tanh (g) 29
3.9. Rezultate–sigmoid (a) 29
3.10. Rezultate–sigmoid (b) 29
3.11. Rezultate–sigmoid (c) 29
3.12. Rezultate–sigmoid (d) 30
3.13. Rezultate–arctan (a) 30
3.14. Rezultate–arctan (b) 30
3.15. Rezultate–arctan (c) 30
3.16. Rezultate–arctan (d) 31
3.17. Rezultate–arctan (e) 31
3.18. Rezultate–arctan (f) 31
3.19. Rezultate–arctan (g) 31

Lista acronimelor

BCE = Binary Cross Entropy
BCI = Brain Computer Interface
CCA = Canonical Correlation Analysis
CT = Computer Tomograf
DNN = Deep Neural Network
DWT = Discrete Wavelet Transform
ECoG = Electrocochleografie
EEG = Electroencefalogramă
ERP = Event-Related Potential
FIR = Finite Impulse Response
FMA = Fugl-Meyer Assessment
fMRI = functional Magnetic Resonance Imaging
GD = Gradient Descent
ICA = Independent Component Analysis
IIR = Infinite Impulse Response
LDA = Linear Discrimant Analysis
LDS = Linear Dynamical System
MAE = Mean Absolute Error
MEG = Magnetoencefalografie
MI = Motor Imagery
ML = Machine Learning
MRC = Medical Research Council
MRI = Magnetic Resonance Imaging
MSE = Mean Square Error
NN = Neural Network
NREM = Non-Rapid Eye Movement
NIHSS = National Institute of Health Stroke Scale
PCA = Principal Component Analysis
PET = Positron Emission Tomography
PReLU = Parametric Rectified Linear Unit
ReLU = Rectified Linear Unit
REM = Rapid Eye Movement
RNN = Recurrent Neural Network
SGD = Stochastic Gradient Descent
SNN = Spiking Neural Network
SPECT = Single-Photon Emission Computed Tomography

SRC = Sparse Representation Classifier
SSVEP = Steady State Evoked Potential
SWLDA = Stepwise Linear Discriminant Analysis
SMR = Sensorimotor Rhythm

Introducere

Motivaţie

Deşi modelul computaţional pentru reţele neurale a fost propus în anii 1940, abia în anii 1980 a luat amploare acest domeniu. Motivul nu a fost lipsa instrumentelor matematice, ci necesitatea unui calculator suficient de complex şi rapid care să poată face calculele care se realizează în timpul antrenării. Momentul de contitură a fost reprezentat perioada în care s-a reuşit creşterea numărului de tranzistoare într-un circuit digital. Evoluţia integrării de tranzistoare pe cip, conform legii lui Moore[1], a permis apariţia unor performanţe impresionante pentru calculatoare. Odată ce a fost atinsă capacitatea de calcul necesară şi antrenarea reţelelor neurale a devenit realizabilă, domeniul de inteligenţa artificială folosind NN (Neural Networks) a crescut rapid în popularitate. Azi reţele neurale sunt folosite pentru aproape orice tip de aplicaţie. De exemplu sunt utilizate în: recunoaştere vocală [1], detecţia cutremurelor [2], predicţia preţului petrolului [3], predicţia stării vremii [4], clasificarea frunzelor de plante bolnave [5], clasificarea de semnale biomedicale (aşa cum îşi propune şi lucrarea de faţă), etc. Prin urmare sunt foarte puţine problemele pentru care nu s-a încercat încă o rezolvare bazată pe NN, dacă aceste probleme nu au deja o abordare deterministă standard. Motivul popularităţii este, pe lângă rezultatele care tind să depăşească performanţa algoritmilor existenţi, modul natural în care se scrie o reţea neurală, aceasta putând fi înţeleasă făcând analogia cu creierul uman. Asemănător modului de funcţionare al acestui organ sunt neuronii şi modelarea tăriei legăturilor dintre aceştia (în cazul creierului legăturile pot fi excitatoare sau inhibitoare). Neuronii din NN sunt unităţi de calcul şi poartă acelaşi nume iar tăria legăturii este reprezentată de valoarea ponderilor.

Aplicațiile de tip BCI (Brain Computer Interface) deschid posibilitatea unei comunicări directe cu calculatorul, nefiind necesară nicio antrenare prealabilă a subiectului uman. Așa cum sugerează și denumirea, paradigma BCI se referă la controlul dispozitivelor electronice doar prin unde cerebrale, fără utilizarea membrelor sau mușchilor, unde care se pot obține folosind electroencefalografia (EEG). Interfața creier-calculator reprezintă o direcție promițătoare de cercetare care s-a dezvoltat semnificativ în ultimii 30 de ani și continuă să fie în plină ascensiune. Deși încă nu s-a atins punctul în care să poată fi citite gândurile [2], progresele actuale în domeniu permit realizarea unor BCI-uri utile și funcționale (ex. proteze inteligente).

De fapt domeniul Brain-Computer Interface este atât de căutat și popular încât încă din 2010 s-a înființat *THE ANNUAL BCI AWARD* [7]. Acesta este un premiu ce se acordă echipelor

[1] Legea lui Moore spune că la fiecare 2 ani numărul de tranzistoare pe circuit integrat se dublează.

[2] cel mai apropiat studiu de această direcție pe care l-am găsit în timpul documentării se referă la clasificarea melodiilor la care se gândește subiectul. Folosind bază de date în care sunt imaginate 12 melodii, autorii studiului [6] au reușit să ajungă la o acuratețe de 37.92%.

de cercetare care reușesc să aducă un plus de inovație în domeniu. În fiecare an 12 proiecte sunt acceptate în competiție și 3 dintre acestea sunt declarate câștigătoare.

Obiective

Acest proiect de licență își propune realizarea următoarelor obiective:

- crearea paradigmei experimentale;
  Acest pas implică realizarea unui program care să genereze un video cu 2 stimuli care să pâlpâie cu frecvențe constante și diferite între ele. Paradigma este una specifică aplicațiilor de tipul SSVEP (Steady-State Visually Evoked Potential).

- colectarea semnalelor EEG;
  Pentru această etapă se vor achiziționa semnalele EEG folosind casca QuikCap conectată la amplificatorul Neuvo. Colectarea datelor va fi interfațată de aplicația software Curry7.

- filtrarea encefalogramelor;
  Pasul acesta este necesar pentru a curăța semnalul de zgomotul cauzat de rețeaua de alimentare de 50Hz. În acest sens se va folosi un filtru de tip *notch* implementat în limbajul de programare MATLAB.

- segmentarea semnalelor;
  Pentru a face o clasificare cât mai rapidă și pentru a mări numărul de exemple care va fi oferit rețelei neurale se vor segmenta datele în porțiuni de câte 2 secunde. Conform [8], în aplicații similare cu cea a acestei lucrări de licență, o fereastră de această dimensiune este suficientă pentru a da rezultate bune. Dacă s-ar folosi o perioadă mai lungă acuratețea nu ar mai crește liniar, întrucât la 2 secunde începe să se atingă valoarea de saturație. De asemenea, pentru a augmenta setul de exemple, segmentarea se va face folosind ferestre suprapuse (suprapunerea fiind de o secundă).

- aplicarea algoritmului CCA (Canonical Correlation Analysis);
  CCA este un algoritm comun în aplicații care folosesc paradigma SSVEP [8], [9], [10], [11] și implică extragerea trăsăturilor din semnal sub forma unor corelații între acesta și semnalul de referință. Referinţa se va construi ca fiind o combinaţie de sinuşi şi cosinuşi având atât frecvența unui stimul din cei 2 prezentați cât și câțiva multipli ai acesteia.

- antrenarea diverselor arhitecturi de rețele neurale;
  În această etapă se vor varia hiperparametrii rețelei și se vor contoriza rezultatele. Variind numărul de neuroni, de straturi și activările ar trebui ca rezultatele obținute să se apropie într-o oarecare măsură de cazul ideal, adică acurateațe de clasificare maximă.

Aplicabilitate

Proiectul de faţă îşi propune realizarea unui sistem de clasificare care să identifice zona de pe ecran căreia subiectul uman i-a acordat atenţie folosind doar undele electrice provenite de la creier. Un clasificator de acest tip poate fi util în următoarele scenarii:

- comunicarea cu pacienţii care nu se pot mişca şi/sau nu pot vorbi;
  Pacientul poate sta cu câţiva[3] electrozi lipiţi pe scalp în zona lobului occipital şi sa aibă

[3] în cadrul acestei lucrări s-au folosit 10 electrozi

lângă acesta un ecran cu stimuli. Fiecărui stimul îi corespunde un mesaj uzual pe care ar putea dori să îl comunice subiectul (de exemplu faptul că îi este sete sau foame). În acest mod pacientul poate comunica cu asistenţa medicală chiar dacă acesta este într-o condiţie care nu îi permine să vorbească sau să facă semne cu mâna.

- realitate augmentată şi jocuri;
  În acest caz modurile în care poate fi utilizat un clasificator ca cel prezentat în această lucrare de licenţă sunt relativ nelimitate. O variantă de implementarea ar fi aceea ca stimulii să reprezinte butoanele de control ale jocului/simulatorului. De pildă dacă este controlată o maşină virtuală stimulul din stângă să reprezinte comanda de viraj la stânga şi analog pentru stimulul din dreapta. Totuşi ideea de a folosi semnale SSVEP în jocuri nu este una nouă; în [12] autorii studiului au implementat un joc de tip *shooter*, iar în [13] a fost realizat un joc în care trebuie scoasă o minge dintr-un labirint.

Capitolul 1
Noţiuni teoretice

1.1 EEG - Aspecte Generale

Prima oară când a fost identificată activitatea electrică la nivelul cortexului a fost în 1875 de către doctorul Richard Caton. Totuși, primele analize de tip EEG-uri efectuate asupra creierului uman aveau să apară abia în 1924, întrucât Caton a studiat răspunsul electric doar pentru animale. Mai precis, acesta a folosit maimuțe, pisici și iepuri în cadrul experimentelor sale[14]. Caton anestezia animalele, făcea o incizie în craniu și în *dura mater*[1] iar apoi plasa electrozii. Un electrod era plasat pe aria studiată și un electrod pe o altă parte a cortexului sau pe craniu. Deși a întâmpinat dificultăți în a face animalele să coopereze pentru experimentele care implicau mișcări ale membrelor, a reușit totuși să identifice câteva aspecte. De exemplu, când animalul analizat își mișca un membru, activitatea celebrala era identificată pe partea opusă a creierului față de acel membru. În 2 iepuri (din 7) și o maimuță a observat un punct lângă zona masticatorie care își modifica potențialul înainte să înceapă masticația. Caton a crezut că acest lucru se datorează percepției mirosului mâncării. De asemenea, a observat potențiale crescute în zone care erau atribuite mișcării ochilor atunci când excita retina cu lumină puternică[14].

Figura 1.1: *Hans Berger în anii 1930*

Așa cum am menționat mai sus, prima encefalogramă la om a fost realizată în 1924. Descoperirea îi este atribuită psihiatrului german Hans Berger fig. 1.1, acesta fiind și persoana care i-a dat numele de *„elektrenkephalogramm”* [15]. Prima encefalogramă a fost făcută unui băiat de 17 ani în timpul unei operații pe creier [16]. Acest lucru a fost necesar pentru că prima metoda era una invazivă, craniul fiind dat la o parte în zona studiată. După 4 ani Berger reușește să construiască un sistem de achiziție atât de sensibil încât EEG-ul se putea realiza în mod neinvaziv. Astfel că până în primăvara anului 1929 făcuse

[1] dura mater este una dintre cele 3 membrane ale meningelui

sute de encefalografii, folosind drept subiecți pacienți cu defecte craniene, angajați ai spitalului unde lucra, fiului său și chiar lui însuși. De asemenea, Berger a înregistrat EEG-uri pentru pacienții cu epilepsie, demență, tumori pe creier și alte boli neurologice [17].

În ceea ce privește complexul EEG, Berger a sugerat că acesta este împărțit în unde α și unde β. Undele α erau cele cu perioada mai mare și erau corelate cu conștiința iar undele β cu perioada mai mică erau asimilate activităților metabolice ale țesutului cortical [17]. Chiar dacă legăturile dintre aceste semnale și fenomenul fiziologic care îl determină au suferit anumite modificări, clasificarea încă rămâne valabilă. În prezent undele α mai sunt numite și unde Berger în onoarea descoperitorului lor.

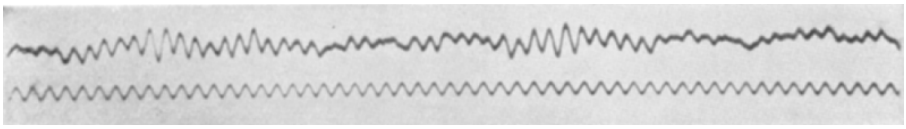

Figura 1.2: *Prima înregistrare EEG la om; primul semnal reprezintă semnalul EEG iar cel de al doilea este un semnal de sincronizare*

Deşi EEG-urile nu mai sunt folosite pentru identificarea bolilor neurologice (datorită apariţiei scanărilor CT - Computer Tomograf), până în anii 1970 acestea au reprezentat un instrument important atât pentru diagnosticare cât şi pentru cercetare. În prezent electroencefalograma se foloseşte în principal în cercetare.

O clasificare mai detaliată asupra undelor EEG se poate regăsi în tabelul de mai jos[2]:

Denumirea undei	Domeniul specific de frecvențe
Delta	0.1 - 4 Hz
Teta	4 - 7 Hz
Alfa	7 - 15 Hz
Mu	7.5 - 12.5 Hz
SMR	13 - 15 Hz
Beta	16 - 30 Hz
Gama	>30 Hz

Tabela 1.1: Distribuția pe frecvențe a semnalelor EEG

Semnificații:

- Unde `Delta` - sunt atribuite stagiului de somn profund (mai precis, stagiu de somn 3 NREM - Non-Rapid Eye Movement)[18].
- Unde `Teta` - de obicei sunt generate în hipocamp şi în principal sunt observate în 2 stări: în interacţiunea activă cu mediul (alergat, mers, etc.) sau în timpul somnului REM[19].
- Unde `Alfa` - sunt create predominant în lobul occipital în timp ce subiectul stă relaxat cu ochii închişi [20].
- Unde `Mu` - sunt generate în cortexul motor iar când o persoană execută o mişcare sau priveşte pe altcineva făcând aceeaşi mişcare aceste unde sunt puternic atenuate [21].

[2]Clasificarea poate varia cu ±1 Hz în funcție de autor

- Unde `SMR` (sensorimotor rhythm) - deşi aceste unde nu sunt pe deplin înţelese din punct de vedere experimental s-a observat că amplitudinea undelor SMR creşte când zona senzorimotoare corespunzătoare este în repaus. Astfel că acestea scand în amplitudine când subiectul se mişcă sau se gândeşte că se mişcă [22].
- Unde `Beta` - sunt asocitate cu gândirea şi concentrarea activă; de asemenea, s-a observat că cu puţin înainte dar şi în timpul unei mişcări aceste unde sunt atenuate [23].
- Unde `Gama` - acest tip de unde este corelat cu atenţia, memoria şi în general cu stările de conştiinţă [24].

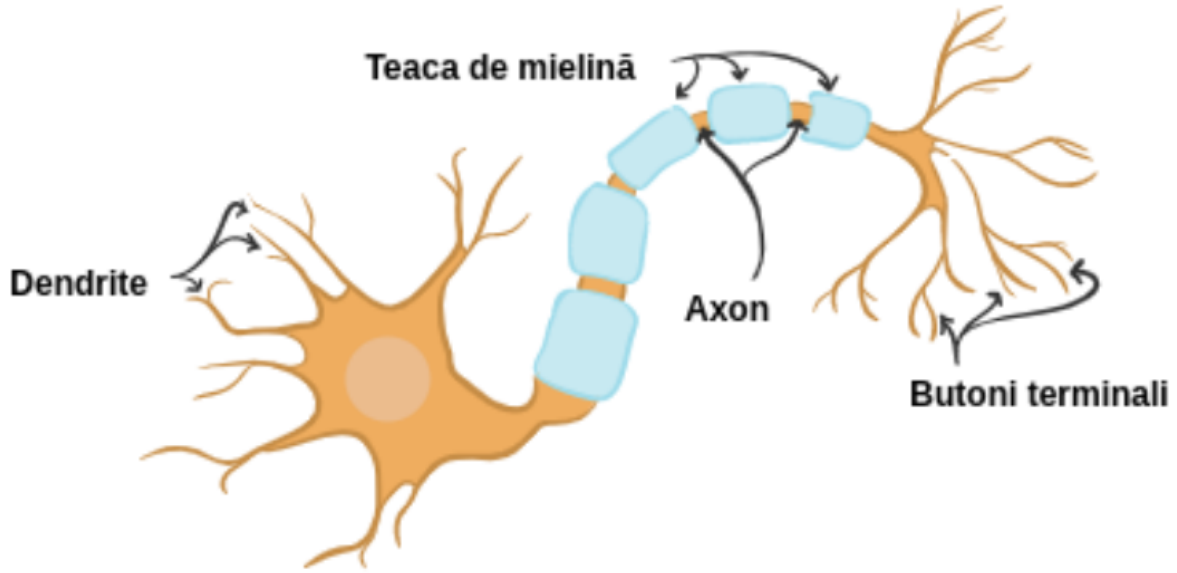

Figura 1.3: *Structura unui neuron*

Dar cum reuşeşte creierul să genereze câmpul electric care este măsurat prin intermediul encefalogramei? Din punct de vedere structural informaţia se propagă cu ajutorul neuronilor. Neuronul prezintă următoarele componente: *dendrite* – au rolul de a primi semnalul de la neuronii vecini; *axon* – transmite semnalul prin neuron; *butoni terminali* – transmit semnalul către dentritele altor neuroni; *teaca de mielină* – măreşte viteza de transmitere prin axon. Toate acestea asigură buna transmisie a informaţiei prin creier. Pentru a putea genera câmp electric este necesară existenţa unui gradient de concentraţii ionice între interiorul celulei şi exteriorul ei.

La repaus (când celula nu este excitată de niciun stimul), neuronii au un gradient de concentraţie negativ. Acest lucru înseamnă că sunt mai mulţi ioni pozitivi în exteriorul celulei decât în interior. În această stare sunt mai mulţi ioni de sodiu (Na^+) în exteriorul neuronului şi mai mulţi ioni de potasiu (K^+) în interioriorul acestuia. Chiar dacă concetraţiile nu sunt stabile pentru că ionii traversează membrana încercând să egalizeze concentraţia, neuronii reuşesc să îşi menţină o tensiune relativ constantă între $-40mV$ şi $-90mV$. Există 3 factori care generează modificarea constantă a tensiunii: faptul că membrana este foarte permeabilă relativ la ionii de potasiu, faptul că este parţial permeabilă pentru ionii de sodiu şi prezenţa unor *porţi* cu funcționalitate de pompă, care trag potasiul înapoi în celulă şi scot natriul în afara ei.

Când un potenţial de acţiune apare, celula începe să se depolarizeze. Pe lângă porțile mai sus menţionate care sunt deschise tot timpul, mai există alte 2 tipuri. Un fel de poartă se deschide

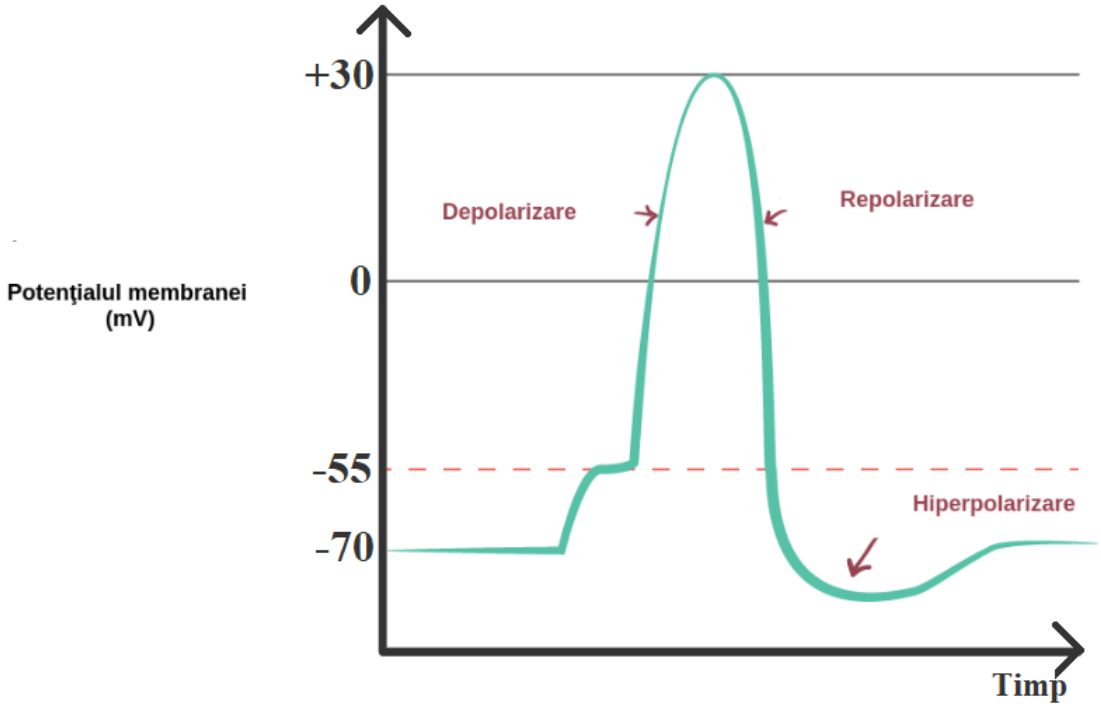

Figura 1.4: *Potenţialul membranei*

doar în momentul în care primeşte stimul din exterior şi alta care se închide când primeşte stimul. Odată apărut potenţialul de acţiune, ionii pozitivi încep să pătrundă în celulă, astfel se depolarizează dinspre dendrite către axon. Dacă la nivelul axonului celula devine suficient de depolarizată, porţile de potasiu se deschid şi potenţialul de acţiune este dat mai departe. Ulterior, celula încearcă să se întoarcă la starea iniţială prin fenomenul numit *repolarizare*. În această etapă, porţile de sodiu se închid şi cele de potasiu se deschid. La acest moment concentrația de K^+ din interior crește semnificativ, deci aceşti ioni vor „inunda" exteriorul celulei. Prin urmare, celula începe să îşi recapete potenţialul inițial (din momentul repausului). Canalele de potasiu stau deschise puţin mai mult decât ar trebui şi astfel ionii pozitivi vor părăsi celula pentru mai mult timp. Prin urmare, neuronul va deveni şi mai negativ. Această etapă poartă denumirea de *hiperpolarizare*. În final porțile de $K^+ - Na^+$ restabiliesc tensiunea iniţială [25].

1.2 Starea artei

Motivul pentru care clasificarea de semnale EEG s-a bucurat de atât de multă atenție în domeniul cercetării încă de la descoperirea lor este aplicabilitatea lor directă în domeniul BCI (Brain Computer Interface). Un BCI este un sistem de comunicare care nu necesită nicio formă de activitate a mușchilor periferici și le permite subiecților să transmită comenzi dispozitivelor eletronice doar prin activitatea celebrală [26]. Avantajul major al unui astfel de sistem este acela că ar putea fi folosit pentru a facilita interacțiunea oamenilor care au dizabilități motorii cu exteriorul. De cele mai multe ori BCI-ul funcționează identificând clasa de apartenență pentru un anumit set de trăsături extras din semnale EEG, pe baza unui algoritm de clasificare [27]. Ținând cont că fiecarei clase îi este atribuită o comandă, se poate face traducerea între unde celebrale și acțiunea dorită a fi întreprinsă de dispozitivul electric.

Majoritatea clasificatorilor care sunt dezvoltați la momentul actual se bazează pe tehnici de învățare automată (*eng. Machine Learning – ML*) [28]. Abordarea folosind ML a reprezentat

un progres în materie de performanţă şi a înlocuit algoritmii folosiţi în mod clasic. Printre algoritmii care erau uzual folosiți amintim: LDA (*Linear discriminant analysis*), SVM (*Support vector machine*), HMM (*Hidden Markov Model*), kNN (*K Nearest Neighbors*) [29].

Aplicațiile principale pe care se concentrează literatura de specialitate în materie de clasificare de semnale EEG sunt următoarele:

- Motor Imagery;
- P300;
- SSVEP (Steady-State Visually Evoked Potential) [3];
- Identificare epilepsie;
- Identificare de emoții.

Motor Imagery (MI) este procedeul prin care subiectul își imaginează că face anumite mișcări (fără să le facă propriu-zis), iar clasificatorul trebuie să identifice ce gest a dorit persoana să execute. În [30] se propune o nouă arhitectură de tipul DNN (Deep Neural Network). Aceasta conține straturi convoluționale pentru început, apoi sunt folosite autoencodere suprapuse (*eng. stacked autoencoders*). Clasificatorul trebuie să deducă dacă persoana în cauză a vrut să miște mâna stângă sau mâna dreaptă (deci o clasificare binară). Autorii au folosit o bază de date în care au fost înregistrați 9 subiecți și au obținut o acuratețe maximă de 95%. Pe lângă genul acesta de aplicații care pot fi translatate în crearea de proteze inteligente, MI folosit în sisteme BCI mai are un avantaj. Conform [31], acestea se pot folosi cu succes în reabilitarea persoanelor care au probleme motorii severe în urma accidentelor vasculare cerebrale și se află în faza subacută[4]. Pentru acest experiment 28 de pacienți au fost împărțiți în 2 grupuri. Un grup, numit de control, s-a antrenat doar cu MI, iar cel de-al doilea grup a făcut aceleași exerciții doar că de această dată aveau suport BCI și putea să vadă în timp real rezultatul încercării de a întinde degetele sau de a strânge pumnul (vezi imaginea 1.5). În urma experimentului s-a constatat faptul că grupul care a folosit MI cu suport BCI a avut o recuperare semnificativ mai bună comparativ cu grupul de control. Mai precis, au fost analizați 3 indicatori considerați relevanți pentru astfel de situații: FMA - Fugl-Meyer Assessment, scara MRC - Medical Research Council pentru puterea muşchilor, NIHSS - National Institute of Health Stroke Scale.

Figura 1.5: *Paradigma experimentală pentru reabilitare*

Unda P300 este un ERP (Event-Related Potential) endogen care este cauzat în mod natural de stimulii consideraţi importanţi de către subiect. De obicei are forma prezentată în figura 1.6a. Aceasta apare în lobii frontal, temporal şi occipital, dar de obicei este măsurată în lobul parietal [32]. P300 prezintă 2 subcomponente `P3a` şi `P3b` (figura 1.6b). `P3a` provie din regiunea frontală corelată cu atenţia din timpul procesării sarcinii. `P3b` îşi are originea

[3]SSVEP este un răspuns cortical vizual generat de un stimul repetitiv consistând dintr-o sursă de lumină care pâlpâie la o frecvenţă constantă mai mare de $4Hz$

[4]faza subacută începe la circa 2 săptămâni după ce a avut loc accidentul vascular cerebral;

(a) *P300* (b) *P3a şi P3b*

Figura 1.6: *Modele simplificate ale formelor de undă*

în activitatea temporal-parietală asociată cu atenţia şi pare să fie corelată cu procesarea ulterioară a memoriei [33]. Semnalul util din P300 conține informație în amplitudine[5] şi latenţă[6]. În funcţie de aceşti 2 parametri se pot face analize referitoare la starea psihică a subiectului. De exemplu în cazul schizofreniei amplitudinea P300 este mai mică comparativ cu a unui subiect sănătos. De asemenea amplitudinea este redusă şi dacă subiectul are autism sau depresie severă [34]. În ceea ce priveşte latenţa, aceasta va fi crescută dacă persoana are demenţă, dislexie sau este un copil cu diferite disfuncţionalităţi cognitive [34]. Dar utilitatea P300 nu se limitează doar la cercetarea asupra bolilor neurologice. Ţinând cont că acest tip de undă este creat atunci când apare stimulul pe care subiectul a fost instruit să îl caute, se poate face progres în ceea ce priveste o tastatură controlată într-un mod pur BCI. De exemplu, în [35] s-a reușit o clasificare a 64 simboluri distincte, litere şi cifre. Subiecţii priveau un ecran ca în figura 1.7 iar rândurile şi apoi coloanele se accentuau pe rând. Acest lucru înseamna că literele/cifrele de pe acea linie/coloană se colorau galben, se roteau cu un unghi aleator între $-90°$ şi $90°$, se măreau de până la 4 ori şi pâlpâiau cu o anumită frecvenţă constantă. Având informaţii legate de linia şi coloana care a generat unda P300 se poate identifica stimulul. Autorii studiului au folosit o bază de date cu 13 subiecţi şi au obţinut o acurateţe de clasificare maximă de 89.96% şi acurateţe medie de 74.24%. Partea de detecţie a fost făcută cu algoritmul SWLDA (Stepwise linear discriminant analysis), iar în urma acestuia fiecărui rând şi fiecărei coloane îi era atribuit un scor. În funcţie de aceste scoruri se putea identifica litera/cifra căutată.

Motivul pentru care a fost inclusă şi pâlpâierea în articolul menţionat mai sus, este acela că s-a dorit încorporarea a 2 direcţii de cercerare din domeniul clasificării de semnale EEG. Pe lângă clasificarea cu unde P300 a fost făcut şi un sistem hibrid având atât P300 cât şi unde SSVEP (Steady-State Visually Evoked Potential) drept date de intrare. Pentru procesarea semnalului SSVEP a fost folosit algoritmul CCA (Canonical Correlation Analysis) care va fi detaliat în capitolul 4. Când a fost folosită doar componenta SSVEP, acurateţea maximă a fost 84.60% şi cea medie 68.53%. Metoda hibridă a oferit rezultate mult mai bune: 97.77% maxim şi 91.33% mediu. Totuşi paradigma clasică pentru SSVEP este aceea de a prezenta stimulii ca având aceeaşi formă şi doar frecvența de pâlpâire poate varia. De exemplu, în [36] au fost folosiţi 3 stimuli de 13Hz, 17Hz şi 21Hz. Cu un algoritm de clasificare bazat pe geometria riemanniană, au ajuns la o performanță de 87.7% acurateţe medie pentru 12 subiecţi având 4 clase (una fiind clasa care reprezintă lipsa pâlpâirii).

O altă direcţie comună în materie de clasificare de semnale EEG o reprezintă detecția de epilepsie. Epilepsia este o boală cronică care se manifestă prin convulsii, aceasta fiind şi cea

[5]definită ca diferenţa dintre tensiunea medie de dinainte de apariţia stimului şi vârful pozitiv dintr-o anumită perioadă de timp;

[6]definită ca perioada de timp dintre începutul stimului şi momentul la care se atinge vârful undei.

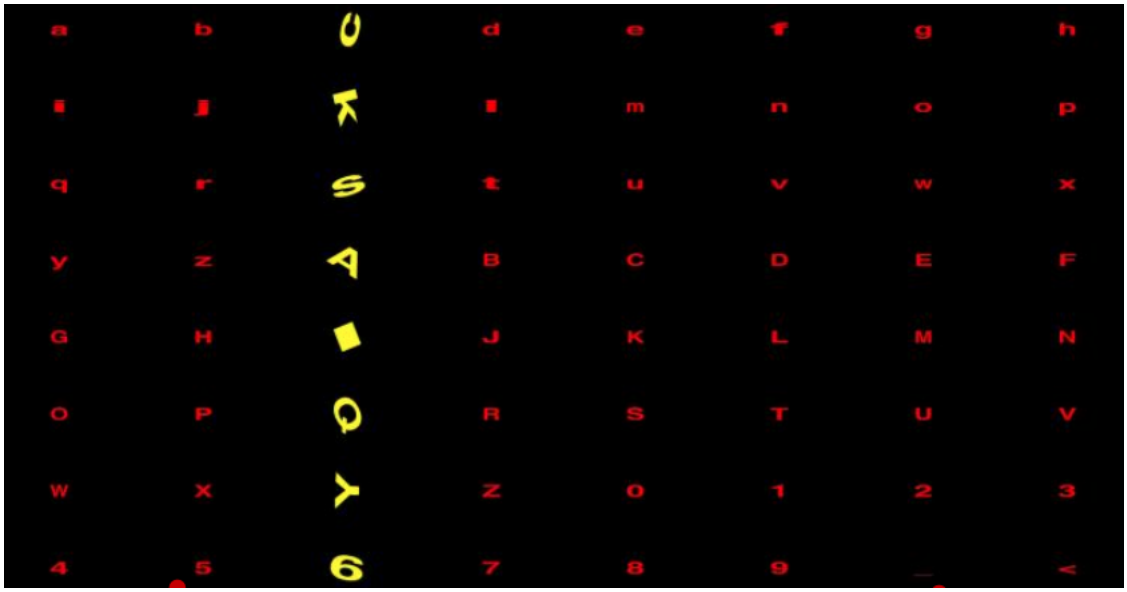

Figura 1.7: *P300 speller*

mai comună afecţiune neurologică (aprox. 65 de milioane de oameni din lume sunt afectaţi) [37]. Din punct de vedere al formei EEG-urilor, în cazul epilepticilor acestea prezintă anumite vârfuri (*eng. spikes*) specifice. Ţinând cont că personalul medical capabil să identifice aceste vârfuri este mic ca număr, se doreşte automatizarea procesului prin intermediul unui clasificator inteligent. O problemă care apare în procesul de clasificare este rata crescută de fals pozitiv. Aceasta se datorează faptului că este foarte ușor de confundat un vârf indicator de epilepsie cu un artefact. În lucrarea [38] se propune un clasificator bazat pe transformata Slantlet[7] (pentru partea de extracţie de trăsături) şi SRC - Sparse Representation Classifier pe post de detector. Aceştia reuşesc să scadă numărul de alarme false, să scadă timpul necesar detecţiei şi să crească acurateţea clasificării. Mai precis, ajung la o valoare de 98.2% pentru o fereastră de semnal de o secundă.

Odată cu dezvoltarea electrozilor uscaţi[8] şi cu evoluţia domeniului de reţele neurale, aplicaţiile BCI prind din ce în ce mai multă popularitate în cercetare întrucât EEG-urile pot fi colectate într-un mod mult mai facil. Pe lângă direcţiile prezentate mai sus, mai putem menţiona şi clasificarea de emoții. În [39] se face discriminarea între emoţii negative şi emoţii pozitive care apar la vizualizarea unor filme. Aceştia colectează date de la 6 subiecţi care sunt puşi să spună ce au simţit în urma vizionării şi cât de puternică a fost emoţia provocată. Scopul celei de a doua cerinţe este de a asigura faptul că emoţiile au fost suficient de pregnante pentru a crea o diferenţă sesizabilă în înregistrarea EEG. Pentru a nivela semnalul şi a-l curăţa de artefacte autorii au folosit algoritmul LDS (*Linear Dynamical System*). Mai departe dimensionalitatea spaţiului este redusă prin mecanismul LDA (*Linear Discriminat Analysis*). Autorii acestui studiu au obţinut o acurateţe maximă de clasificare egală cu 91.77%.

1.3 Rețele neurale – noțiuni introductive

Reţelele neurale (sau *neuronale*) reprezintă o paradigmă de soluţionare a problemelor diferită de metoda clasică în care maparea dintre datele de intrare şi cele de ieşire este făcută în mod determinist. Principiul acestei abordări constă în dezvoltarea unui algoritm care să *înveţe* funcţia

[7]este un tip de transformată DWT - Discrete Wavelet Transform

[8]electrozi care nu necesită gel electroconductor

de transfer ce alocă datelor de intrare valoarea la ieşire, fără ca programatorul să cunoască apriori această relaţie. Modalitatea în care se realizează acest lucru este analog modului de funcţionare al creierului. Creierul conține neuroni interconectaţi ce procesează informaţia şi în timp sunt capabili să creeze legături între ei de aşa natură astfel încât să poată generaliza pe baza exemplelor învățate – deci dându-se anumite date să ofere un răspuns adecvat. În mod similar şi reţeaua neuronală are neuroni – unităţi funcţionale de calcul – şi legăturile dintre aceştia sunt modelate de valori numerice numite ponderi. Printr-un algoritm iterativ de optimizare valorile acestor ponderi se modifică, deci am putea spune că legaturile dintre neuroni se modifică. Pentru a putea învăţa este necesar ca reţeaua să primească suficiente exemple, pentru care se poate cunoaşte ieşirea – învățare *supervizată* sau nu – învățare *nesupervizată*.

În cadrul reţelelor neurale, neuronii sunt organizaţi pe straturi. Aceste straturi pot fi: de intrare, ascunse sau de ieşire. Stratul de intrare are rol de buffer pentru date şi nu modifică în niciun fel valorile de la intrare. Urmează stratul/straturile ascuns(e) care manipulează datele iar apoi stratul de ieşire care face o ultimă transformare asupra valorilor oferite de stratul/straturile ascuns(e) şi oferă ulterior răspunsul final.

Un aspect foarte important pentru arhitectura reţelei neurale îl constituie distribuţia numărului de neuroni pe fiecare strat. Dimensionarea straturilor ascunse este cea mai provocatoare etapă întrucât o structură adecvată nu se poate determina analitic ci doar experimental. Însă dimensiunile straturilor de intrare şi de ieşire sunt uşor de aflat deoarece sunt dictate de aplicaţie. De exemplu, să presupunem că avem o imagine de 32×32 pixeli alb-negru şi trebuie să spunem dacă în acea imagine se află fie o pisică, fie un câine, fie o pasăre. În această situaţie stratul de intrare va avea dimensiunea de $32 \times 32 = 1024$ de neuroni (cate unul pentru fiecare pixel din imagine) iar stratul de ieşire va avea dimensiune 3, corespunzătoare numărului posibil de clase.

O ilustrare asupra arhitecturii generale a unei reţele neurale este prezentată în figura 1.8. Se poate observa că datele de intrare, $x_1, x_2, x_3, \ldots, x_n$ sunt date stratului de intrare care predă mai departe aceste valori stratului ascuns. Stratul ascuns continuă procesarea datelor şi acestea trec mai departe către stratul ascuns. Ieşirea din stratul ascuns este reprezentată de valorile $y_1, y_2, y_3, \ldots, y_n$. De obicei, în cazul problemelor de clasificare acestea reprezintă probabilităţi de apartenenţă.

În cadrul fiecărui neuron au loc 2 operaţii: una *liniară* şi una *neliniară*. Cea liniară este de tipul $W^T X$ unde X reprezintă vectorul de date de la intrarea neuronului şi W vectorul de ponderi specific acelui neuron. Etapa neliniară o constituie aplicarea unui operator neliniar numit *activare*. Câteva dintre funcţiile de activare folosite uzual sunt prezentate mai jos.

Funcţia sigmoid (numită și funcție logistică) este ilustrată în figura 1.9 şi are formula:

$$sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{1.1}$$

După cum se poate observa, aceasta ia valori care tind spre 0 când x tinde la $-\infty$ şi valori care tind către 1 când x tinde la $+\infty$. În vecinătatea lui 0.5 funcţia prezintă pantă maximă iar valoarea în acest punct este 0. Aspectul în formă de „S” ajută în probleme de clasificare întrucât creşte separabilitatea datelor de intrare. Din acest motiv funcţia de activare sigmoid se bucură de popularitate în rândul acestor tipuri de probleme [40].

Funcţia tangentă hiberbolică (tanh) – vezi figura 1.10 – are o formă asemănătoare cu cea a sigmoidului deoarece are tot formă de `S` şi, prin urmare, este folosită în aplicaţii similare. Totuşi, în practică, *tanh* tinde să fie mai bună decât acesta. Tangenta hiberbolică ia valori în intervalul $[-1; 1]$, acest lucru făcând ca valorile date la intrare să fie puternic polarizate (datele negative sunt transformate în valori care tind la -1, datele pozitive sunt transformate în valori care tind la +1 iar valorile în jurul lui 0 sunt mapate tot în vecinătatea lui 0). Formula cu care

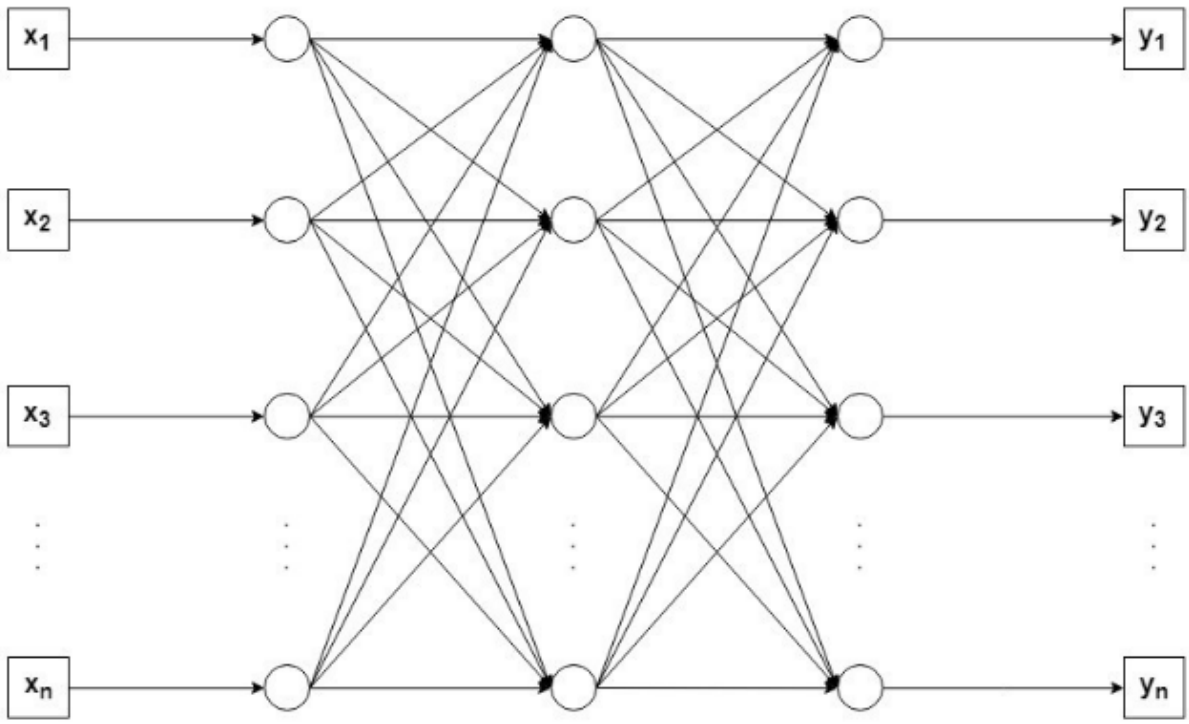

Figura 1.8: *Arhitectura generală pentru o reţea neurală*

se poate calcula *tanh* este dată mai jos:

$$tanh(x) = \frac{sinh(x)}{cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^{2x} - 1}{e^{2x} + 1} \tag{1.2}$$

O altă funcţie de activare este reprezentată de **arctangentă (arctan)**, aceasta fiind şi inversa tangentei hiperbolice. Valorile pe care le poate lua se află în intervalul $[-\pi/2; \pi/2]$. Se poate observa că *arctan* este puţin mai plată decât *tanh*, aspect care o face să diferenţieze mai bine între valori similare la intrare.

$$arctan(x) = tan^{-1}(x) = \int_0^x \frac{1}{z^2 + 1} dz \tag{1.3}$$

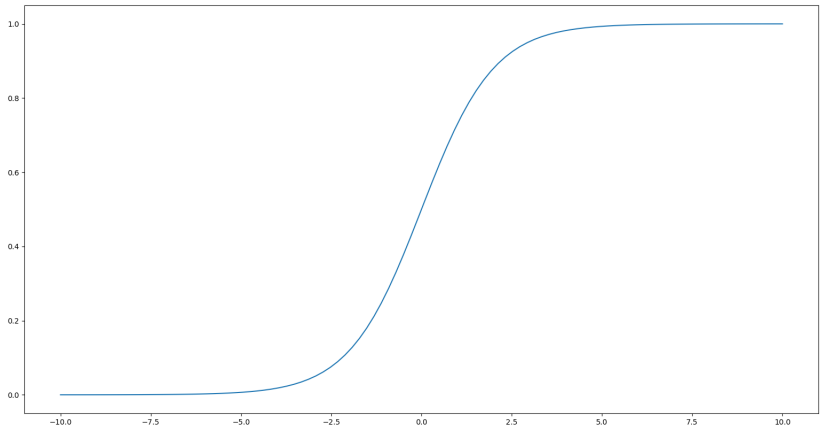

Figura 1.9: *Sigmoid*

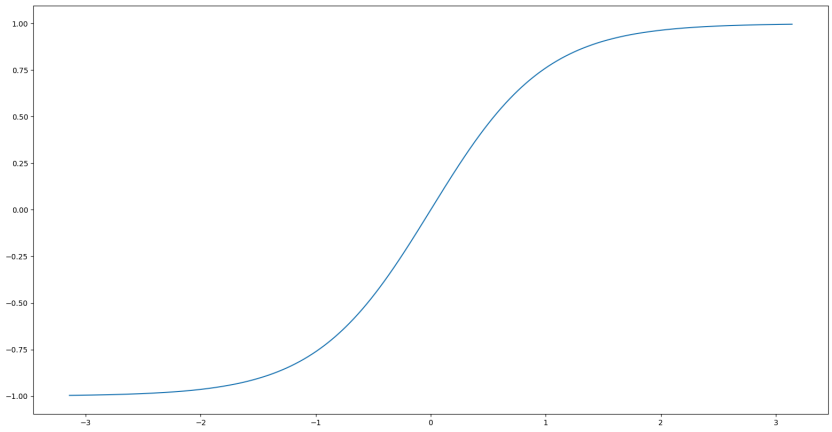

Figura 1.10: *Tanh*

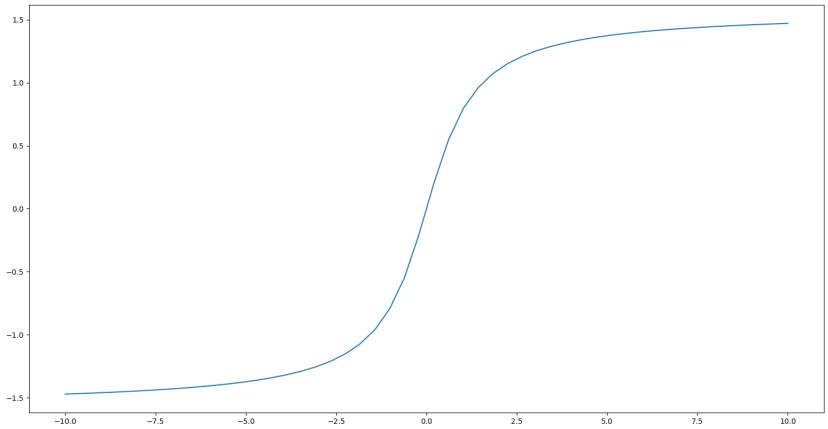

Figura 1.11: *Arctan*

O opţiune foarte populară în rândul funcţiilor de activare este **ReLU (Rectified Linear Unit)** [41]. Acesta este folosit în principal în probleme de aproximare/estimare [42].

ReLU se comportă conform formulei:

$$ReLU(x) = max(0, x) \tag{1.4}$$

Un avantaj major al acestei funcţii este faptul că scad şansele să apară fenomenul de *„vanishing gradient”* (gradient[9] ale carui valori tind puternic către 0) [43]. De asemenea, simplitatea funcţiei ajută în procesul de antrenare.

O variaţie asupra funcţiei ReLU este reprezentată de **Leaky ReLU**. Aşa cum îi sugerează şi numele această variantă lasă o parte din valorile negative să se „scurgă” la ieşire. Comparativ cu forma originală, Leaky ReLU are un gradient pozitiv relativ mic pentru valori negative la intrare şi, în acest caz, niciun gradient nu va avea valoarea 0. Totuşi acest aspect nu este întru totul un avantaj deoarece matricele din cadrul algoritmului nu vor mai fi rare, ,îngreunând astfel procesul de calcul. Pentru a calcula valoarea la ieşirea lui Leaky ReLU se foloseste următoarea formulă:

$$LeakyReLU(x) = \begin{cases} x & \text{,dacă } x \geq 0 \\ 0.01x & \text{,dacă } x < 0 \end{cases} \tag{1.5}$$

Dacă în loc de valoarea de 0.01 de pe a doua ramură a funcţiei este o altă valoare α, atunci funcţia de activare se numeşte **Parametric ReLU (PReLU)**.

[9]utilitatea gradientului în cadrul rețelelor neuronale va fi discutată ulterior

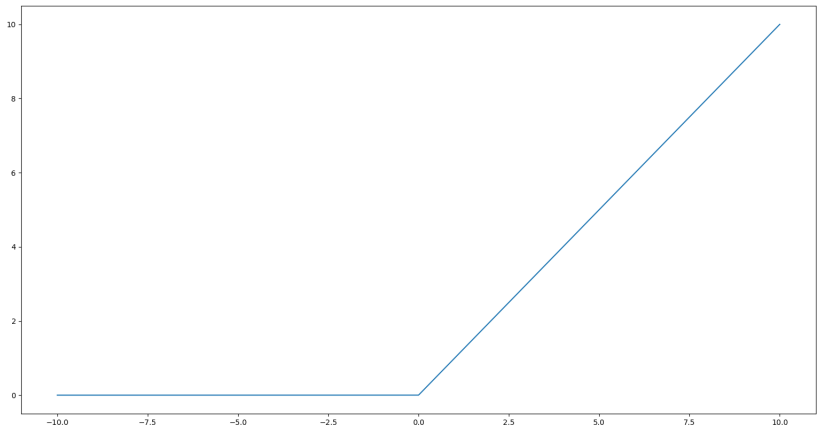

Figura 1.12: *ReLU*

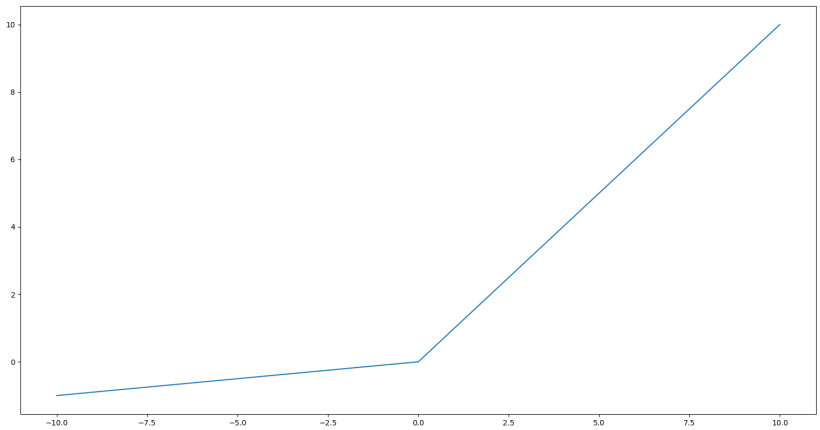

Figura 1.13: *Leaky ReLU*

Deşi în această lucrare nu s-a folosit nici ReLU, nici Leaky ReLU, acestea au fost amintite deoarece sunt foarte populare în domeniul reţelelor neurale.

În mod uzual, pentru stratul de ieşire, funcţia de activare folosită este *softmax*. Aceasta se întâlneşte, în special, în probleme de clasificare [44]. Specificitatea funcţiei este aceea că transpune valorile vectorului de la intrare în intervalul $(0, 1)$. De asemenea suma componentelor de la ieşire va fi egală cu 1.

$$softmax(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}} \tag{1.6}$$

Așa cum am amintit mai sus, pe lângă etapa neliniară, există și cea liniară. Aceasta se scrie astfel:

$$\xi = W^T X = \sum_{i=1}^{N} x_i \cdot w_i + b, \tag{1.7}$$

unde x_i sunt elementele vectorului de intrare X, iar w_i sunt elementele vectorului de ponderi W. Termenul liber, b, are rolul de a prepolariza neuronul. De obicei acest termen apare în componența lui W. Odată calculată suma ponderată în modul prezentat mai sus se aplică etapa neliniară și ieșirea din neuron va arăta astfel:

$$y = g(\xi), \tag{1.8}$$

unde g reprezintă o funcție de activare aleasă conform specificității aplicației.

Pentru a măsura performanța, în procesul antrenare se utilizează un algoritm de optimizare

iterativ ce are ca scop micșorarea unei funcții de cost[10], definită ca o măsura a similarității dintre datele de ieșire ale rețelei și ceea ne dorim să obținem. Ca și în cazul funcțiilor de activare și aici există o varietate de formule matematice pentru a calcula costul. Câteva exemple sunt prezentate în continuare.

MSE – Mean Square Error – de obicei este folosit în aplicații de regresie liniară întrucât ajută la găsirea dreptei care minimizează distanța dintre fiecare punct și aceasta. În general se folosește formula:

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - \hat{y}^{(i)})^2, \tag{1.9}$$

unde n este numărul de exemple pentru care se calculeaza eroarea, $y^{(i)}$ este ieșirea ideală pentru cel de al i-lea exemplu, iar $\hat{y}^{(i)}$ este ieșirea calculată pentru cel de al i-lea exemplu.

MAE – Mean Absolute Error – este similar cu MSE, însă este mai robust la elemente care sunt relativ îndepartate de cluster-ul principal.

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n} \mid y^{(i)} - \hat{y}^{(i)} \mid^2 \tag{1.10}$$

L2 - din punct de vedere matematic este asemănător cu MSE, doar ca de această dată nu se mai face împărțirea la numărul de exemple.

$$\mathcal{L} = \sum_{i=1}^{N}(y^{(i)} - \hat{y}^{(i)})^2 \tag{1.11}$$

L1 – analog lui L2, L1 este asemănător cu MAE, doar că în acest caz nu se mai împarte la numărul de exemple.

$$\mathcal{L} = \sum_{i=1}^{n} \mid y^{(i)} - \hat{y}^{(i)} \mid^2 \tag{1.12}$$

BCE – Binary Cross Entropy – este o funcție de cost folosită în probleme de clasificare binară sau când datele trebuie binarizate [45]. De asemenea este tipul de cost care a fost folosit în cadrul acestei lucrări.

$$\mathcal{L} = -\frac{1}{n}\sum_{i=1}^{N} y_i \cdot log(\hat{y}) + (1 - y_i) \cdot log(1 - \hat{y}) \tag{1.13}$$

Odată ce valoarea costului a fost calculată, următorul pas este acela de a-l minimiza. Un algoritm comun în domeniul rețelelor neurale pentru a face acest lucru este *backpropagation*. În cadrul acestei metode, după fiecare parcurgere în sens direct (de la intrare către ieșire), se face și parcurgerea în sens invers modificându-se ponderile. Această modificare se face în funcție de gradientul $\frac{\partial \mathcal{L}}{\partial w}$. Mai precis, se poate scrie relația dintre valoarea ponderii anterioare și cea la iterația curentă astfel:

$$w_i(t+1) = w_i(t) - \eta\frac{\partial \mathcal{L}}{\partial w_i} \tag{1.14}$$

Ținând cont de faptul că gradientul arată direcția de creștere a unei funcții, scăzând la fiecare iterație $\eta\frac{\partial \mathcal{L}}{\partial w}$, ne vom îndrepta către minimul costului $\mathcal{L}$. Parametrul η se numește rată de învățare (*eng. learning rate*) și determină cât de mare va fi variția lui w la fiecare pas. Acesta poate fi static sau poate să își modifice valoarea în timpul antrenării.

[10]se mai numește și funcție de pierdere *eng. loss function* - de unde provine și notarea ulterioară cu $\mathcal{L}$

Capitolul 2
Setup experimental

2.1 NeuroScan

Instalația pentru achiziție constă în casca cu electrozi *Quik-Cap*, figura 2.1, amplificatorul de semnale EEG: *Neuvo*, figura 2.2, și calculatorul pe care este instalat software-ul *Curry 7*.

Casca Quik-Cap este special concepută pentru a fi compatibilă cu amplificatorul Neuvo și a putea fi conectată direct la acesta. Este prevazută cu 128 de electrozi, făcând-o ideală pentru aplicații care au nevoie de o rezoluție spațiala mare. Electrozii sunt atașați pe un material elastic din lycra și sunt plasați conform standardului internațional 10-20. Acest mod de aranjare se bazează pe legătura dintre zona unde este plasat electrodul și regiunea din cortexului celebral aferentă. Numerele „10” și „20” se referă la distanța dintre 2 electrozi adiacenți care poate să fie 10% sau 20% din distanța totală frontal-anterior sau stânga-dreapta a craniului. Pentru a putea face achiziția, între electrozi și scalp este introdusă pastă electroconductoare. Pasta are rolul de a micșora impedanța de contact între subiect și senzori. În plus, pentru a asigura o măsurătoare cât mai precisă, electrozii sunt din *Ag/AgCl* sinterizat, aceștia fiind recunoscuți internațional ca prezentând stabilitate mare și zgomot redus.

Neuvo are 128 de canale pentru EEG și este conectat direct la casca cu electrozi. Acesta este alimentat direct de la priză. Frecvența sa de eșantionare poate fi variată între $100Hz$ și $10kHz$; pentru lucrarea de față s-a folosit o frecvență de eșantionare de $200Hz$. Amplificatorul

Figura 2.1: *Casca cu electrozi*

Figura 2.2: *Amplificator Neuvo*

are impedanța de intrare de minim 10GΩ și *CMMR* (Common Mode Rejection Ratio) de minim $110dB$. Datele sunt convertite din analog în digital cu un CAD (Convertor Analogic Digital) cu rezoluție de 24 de biți. Mai departe, datele convertite în format digital sunt trimise prin protocolul *Ethernet* către calculatorul cu software-ul *Curry 7* instalat.

Curry 7 înregistrează și salvează semnalele *EEG* cu posibilitatea de a le vizualiza pe ecran în timp real. Procesarea datelor astfel achiziționate va fi detaliată în Capitolului 3.

2.2 Curry 7

Figura 2.3: *Curry7*

Curry 7 este aplicația software oferită de Compumedics pentru a gestiona atât achiziția cât și procesarea ulterioară a datelor biomedicale. Programul are suport pentru o varietate de semnale fiziologice: EEG, ECoG (electrocochleografie) [1], MEG (magnetoencefalografie)[2], MRI (Magnetic Resonance Imaging)[3], fMRI (functional Magnetic Resonance Imaging)[4], CT (Computer Tomograf)[5], PET (Positron Emission Tomography)[6] și SPECT (Single-Photon Emission Computed Tomography)[7].

Pentru a putea face achiziția, aplicația trebuie configurată conform specificității experimentului. Primul pas este reprezentat de alegerea tipului de amplificator și de cască. În cadrul aceleași ferestre de configurare se poate selecta și frecvența de eșantionare (vezi figura 2.5). Pentru această lucrare s-a folosit o rată de eșantionare de 200 Hz, această valoare fiind un compromis între dimensiunea datelor ce vor fi colectate și surprinderea evenimentelor relevante din semnal. Celelalte configurații sunt lăsate la valorile implicite, respectiv filtru trece jos de valoare 80 Hz și acuratețe de 24 nV/LSB[8].

Odată ce a fost selectat tipul de cască se poate vizualiza și distribuția electrozilor pe aceasta, iltustrată în figura 2.6. Pentru lucrarea de față s-au folosit electrozii cu numerele: 42, 43, 44, 45, 68, 69, 70, 71, 97, 98. Toţi electrozi sunt identificaţi şi cu nume formate din litere. Pentru electrozii de mai sus numele alternative sunt, în ordine: I1, OI1, POO3, PO1, I2, OI2, POO4, PO2, PPO8, P6. Aceștia au fost aleși deoarece se află în zona lobului occipital (este colorat cu galben în imaginea 2.4), zonă care este responsabilă cu procesarea vizuală la om [46].

Figura 2.4: *Lob occipital*

Înainte de începerea achiziției propriu-zise trebuie asigurat faptul că impedanța de contact între scalp și electrozi este de o valoare acceptabilă. Dacă impedanța este prea mare atunci nu se va putea înregistra semnalul util, care are o amplitudine foarte mică (de obicei între $10\mu V$ și $100\mu V$ [47]). Curry 7 oferă opțiunea de a vedea în

[1]reprezintă o metodă de înregistrare a potențialul electric generat de urechea internă în momentul în care este prezentat un stimul auditiv

[2]reprezintă o metodă de analiză a activității cerebrale măsurând câmpul magnetic generat de curenții electrici care traversează creierul

[3]reprezintă o tehnică care presupune realizarea de imagini ale organelor folosind câmpi magnetici puternici

[4]este o metodă de a măsura activitatea cerebrală prin monitorizarea schimbării volumului de sânge din acesta

[5]este o metodă în care se realizează imagini 3D a diverselor organe, făcând o serie de măsurători cu raze X

[6]reprezintă metoda în care se folosesc substanțe radioactive pentru a genera imagini aferente proceselor metabolice

[7]reprezintă o tehnică de imagistică medicală nucleară în care se folosesc raze gama

[8]Least Significant Bit

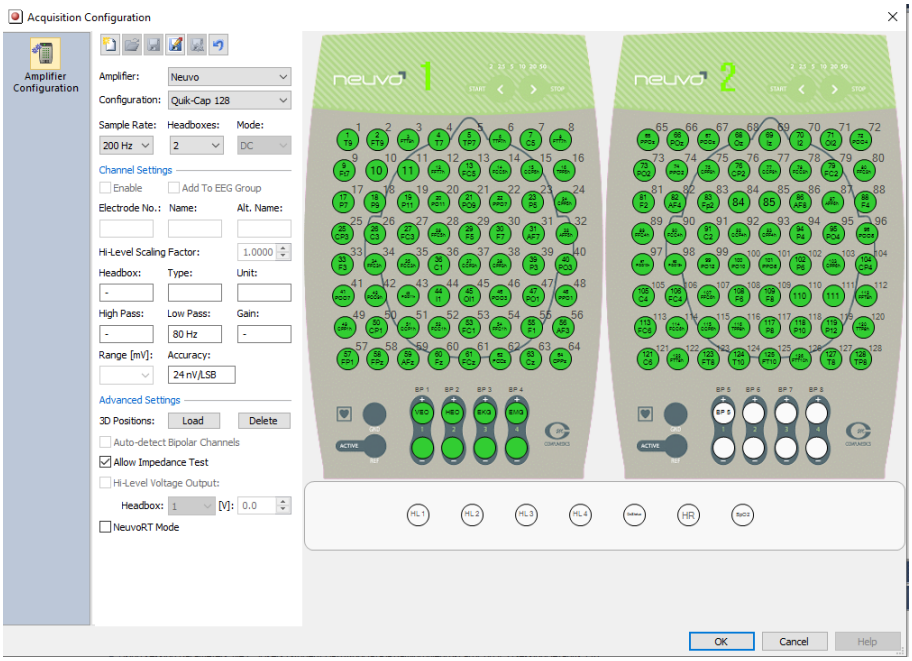

Figura 2.5: *Configurare Curry 7*

timp real impedanța de contact a electrozilor. Această facilitate se prezintă din punct de vedere grafic ca în figura 2.7. Imaginea ilustrează impedanțele înaintea aplicării gelului electrolitic. Cu ajutorul acestui mod se poate ajusta cantitatea de substanță conductoare necesară astfel încât să se obțină o achiziție de calitate.

2.3 Paradigma experimentală

Pentru achiziția bazei de date s-a folosit un singur subiect. Acesta este bărbat, în vârstă de 21 de ani și cu părul scurt. Subiectul nu prezintă afecțiuni care ar putea altera rezultatele sau care l-ar putea pune în pericol în timpul procesului de achiziție (de ex. epilepsie). Subiectul a stat așezat pe un scaun confortabil la aproximativ 2 metri de un ecran de rezoluție $4K$ şi diagonală de 194 cm și a fost instruit cărui stimul de pe ecran să îi acorde atenție. Pe ecran erau prezenți simultan 2 stimuli pe un fundal negru: în partea stângă un dreptunghi verde care pâlpâia cu frecvența de aproximativ $10.63Hz$ iar în partea dreaptă un dreptunghi roșu care pâlpâia cu frecvență de aproximativ $4Hz$.

Mențiunea de „aproximativ” se referă la limitările sistemului de calcul care în domeniul acesta de frecvențe de afișare tinde să aibă întârzieri care depășesc $1ms$ (cât este considerat acceptabil pentru o analiză EEG). O soluție de compromis a fost aceea de a monitoriza constanța frecvenței de pâlpâiere pentru mai multe frecvențe și a găsi o valoare care rămâne cât mai stabilă în timp. Am observat faptul că pâlpâirea de pe ecran rămânea relativ stabilă pentru ambii stimuli la frecvențe de aproximativ $4Hz$ și $10.63Hz$.

În timpul achiziției a fost foarte important ca subiectul să nu obosească, fapt ce putea duce la alterarea semnalelor colectate. În acest sens achiziția a fost realizată în două reprize, chiar dacă timpul total de achiziție este de 7 minute și 30 secunde. În plus, subiectul a fost consultat cu privire la începerea și oprirea înregistrării astfel încât acesta să nu își piardă atenția în timpul achiziției. Au fost făcute 11 înregistrări cu stimulul verde însumând 3 minute și 41 secunde; pentru stimulul roșu au fost făcute 13 înregistrari însumând 3 minute și 48 de secunde. S-a încercat păstrarea unui echilibru între semnalele achiziționate pentru cei 2 stimuli.

Pentru a asigura un contact cât mai bun între scalp și electrozi subiectul a fost spălat pe cap înaintea achiziției. La suprafața pielii tinde să se formeze un strat de celule moarte care

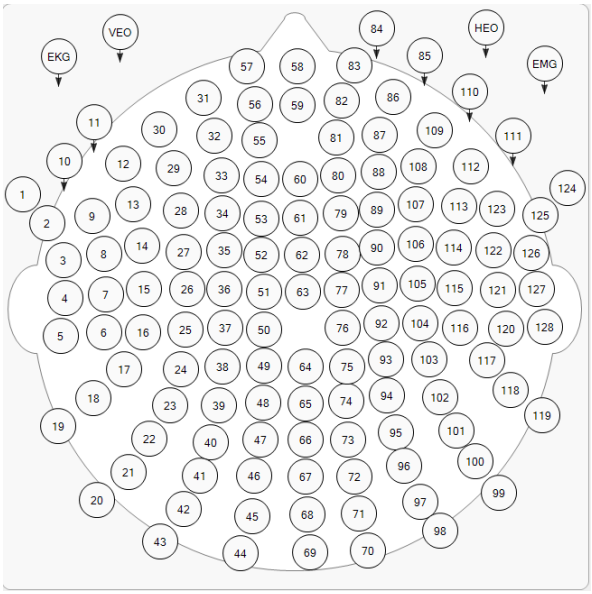

Figura 2.6: *Distribuție electrozi*

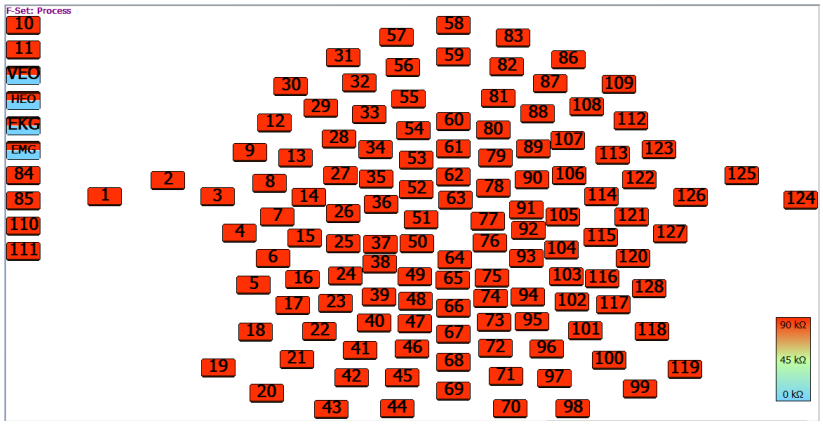

Figura 2.7: *Reprezentarea grafică a impedanței de contact*

confera pielii un efect capacitiv ceea ce ar îngreuna procesul de colectare a semnalelor EEG. Pe lângă curățare, este imperios necesară folosirea gelului electroconductor. Acesta a fost inserat cu ajutorul unei seringi fără ac prin orificiul din electrozi. *Curry 7* are opțiunea de vizualizare în timp real a impedanței fiecarui electrod de pe cască astfel că este ușor de evaluat dacă a fost aplicată suficientă pastă electroconductoare.

La finalul achiziției electrozii au fost curățați pentru a le conserva proprietățile în timp. În caz contrar gelul se usucă și este mult mai greu de curățat. De asemenea, există riscul de oxidare a argintului, ceea ce ar duce la compromiterea electrozilor.

Capitolul 3
Metoda propusă

3.1 Schema generală a proiectului

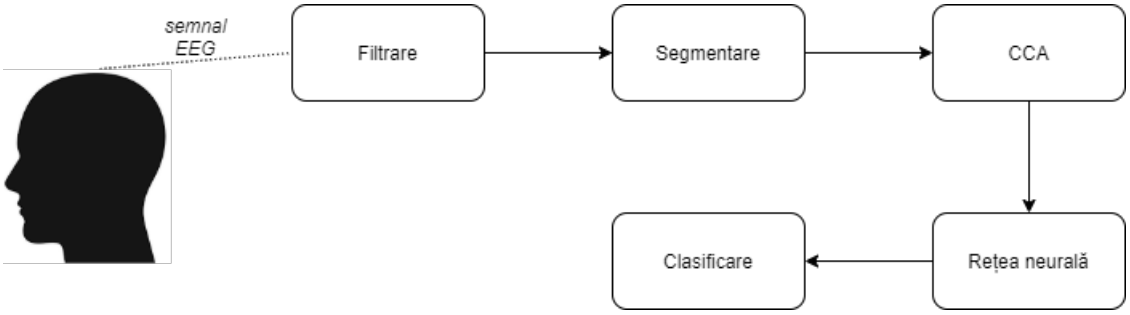

Figura 3.1: *Succesiunea de etape din cadrul proiectului*

Odată ce semnalele au fost achiziționate conform metodei detaliate în subcapitolul *3.2 Paradigmă experimentală* urmează ca acestea să fie procesate după o succesiune de etape ilustrată în figura 3.1.

În primul rând, semnalele EEG trebuie filtrate. Pasul acesta este imperios necesar deoarece peste datele achiziționate este suprapus un semnal parazit de 50Hz. Acesta provine de la rețeaua națională de alimentare cu energie electrică și poate fi atât de puternic încât să „înece" semnalul util și să facă clasificarea ulterioară aproape imposibilă. Așa cum am menționat anterior Neuroscan se conectează direct la priză în timpul achiziției și astfel apare artefactul de 50Hz. În scopul eliminării zgomotului de la rețeau de alimentare s-a folosit un filtru *IIR notch* implementat în MATLAB. Un filtru *notch* este un filtru cu banda de oprire foarte îngustă iar acronimul IIR provine de la **I**nfinite **I**mpulse **R**esponse. Alternativa clasei IIR este clasa FIR - **F**inite **I**mpulse **R**esponse. Printre diferențele dintre acestea se numară:

- IIR are un raspuns în timp mai rapid comparativ cu FIR [48];
- IIR defazează diferit componentele pe frecvențe diferite față de FIR care generează o defazare constantă (IIR are fază neliniară și FIR are fază liniară) [48];
- FIR este întotdeauna stabil în timp ce IIR este uneori instabil [48];

În contextul acestui proiect de licență faza neliniară nu este un impediment deoarece informația de interes (prin intermediul căreia se va face clasificarea) nu se află în componenta de

fază ci în cea de frecvență. Astfel că filtrul ales pentru această primă etapă de procesare a fost din clasa IIR. Pentru a asigura un semnal filtrat stabil, din fiecare înregistrare a fost eliminată prima secundă. Această perioadă de timp este aleasă experimental și are rolul a asigura faptul că nu se iau în considerare momentele de timp în care filtrul încă este în faza de stabilizare.

După ce semnalele EEG au fost filtrate, următoarea etapă este cea de segmentare. Așa cum am menționat în *3.2 Paradigma experimentală*, în timpul achiziției datelor EEG este foarte important ca subiectul să nu obosească și în acest sens s-a făcut colectarea datelor în 2 reprize separate de câteva săptămâni. În plus, înregistrarile sunt separate în funcție de stimul iar ordinea de prezentare a stimulilor a fost aleatoare, tocmai pentru a lăsa subiectul să se odihnească. Distribuția fișierelor cu semnale EEG este prezentată în tabelul 3.1 cu mențiunea că ordinea din tabel nu reflectă ordinea de achiziție.

Nr. crt.	Culoare stimul	Durată [secunde]
1	verde	28.5
2		47.2
3		11.1
4		11.9
5		19.4
6		15.7
7		16
8		15.9
9		13.8
10		14
11		27.9
12	roșu	23.2
13		25.5
14		11.6
15		11.8
16		13.5
17		23.6
18		13.3
19		19.7
20		21.3
21		15.5
22		13.6
23		16.1
24		20.4

Tabela 3.1: Distribuţia înregistrărilor în funcţie de stimul

După cum se poate observa durata totală de achiziție este de 7 minute și 30 secunde. Pentru antrenarea unei rețele neurale este nevoie de cât mai multe date de antrenare iar înregistrările disponibile sunt destul de puține și scurte pentru genul acesta de abordare de clasificare. Pentru a compensa dimensiunea bazei de date am împărțit datele filtrate în segmente de 2 secunde. Această perioadă de timp este suficientă pentru a clasifica semnale EEG de tipul SSVEP [8], în contextul utilizării algoritmului CCA pentru extragere de trăsături. În plus pentru a augmenta datele de antrenare înregistrările au fost împărțite cu suprapunere de o secundă între segmente.

În continuare, segmentelor li s-a aplicat algoritmul CCA (vezi *4.2 CCA* pentru detalii asupra algoritmului în sine). Pentru aplicarea acestei metode a fost folosit un pachet scris în limbajul de programare Python numit `Pyrcca` [49]. Datorită naturii metodei CCA este necesară existența

unui semnal de referință, iar algoritmul va indica cât de corelate sunt datele de intrare în raport cu referința oferită. În momentul dezvoltării paradigmei experimetale a fost observat faptul că frecvența de pâlpâiere a stimulului roşu (cel de 4Hz) este mai stabilă comparativ cu stimului verde de 10.63Hz. Ca urmare a acestui aspect, referința a fost construită în jurul frecvenței de 4Hz. Mai exact referința este o matrice de dimensiune 400×8. Numărul de linii este numărul de eșantioane necesar pentru a acoperi o perioadă de 2 secunde la frecvența de eșantionare de 200Hz. Pe coloane sunt vectori ce reprezintă, în această ordine: sinus de 4Hz, cosinus de 4Hz, sinus de 8Hz, cosinus de 8Hz, sinus de 12Hz, cosinus de 12Hz, sinus de 16Hz, cosinus de 16Hz. Așa cum am detaliat în capitolul 2, în cazul semnalelor SSVEP, atât frecvența stimulului cât și multiplii acesteia se regăsesc în răspunsul generat de creier. Motivul pentru care am folosit și cosinuși este acela de a acoperi o plajă cât mai mare de defazări a semnalului răspuns. Datele de ieșire ale algoritmului sunt reprezentate de vectori de dimensiune 8 (dimensiunea maximă pe care o poate oferi algoritmul, acesta fiind minimul dintre numărul de trăsături ale matricei de intrare și numărul de trăsături ale matricei referință). Matricea cu care a fost comparată matricea de referință este de dimensiune 400×10. Valoarea 10 reprezintă numărul de electrozi folosiți iar 400 numărul de eșantioane colectate la o frecvență de eșantionade de $200Hz$ timp de 2 secunde. Practic răspunsul algoritmului CCA va indica o informație de tipul: dacă semnalul provine ca urmare a stimulului de 4Hz sau nu.

Odată ce am obținut vectorii de corelație de dimensiune 8 pentru fiecare segment, se poate trece la faza urmatoare. Aceasta nouă etapa constă în antrenarea rețelelor neurale. Cu modelul obținut în urma antrenării se poate face clasificarea propriu-zisă a datelor din lotul păstrat pentru test. Justificările asupra variațiilor arhitecturilor încercate din cadrul rețelelor neurale sunt oferite în subcapitolul *4.3 Rezultate*.

3.2 Algoritmul CCA

CCA – Canonical Correlation Analysis reprezintă un algoritm folosit adesea pentru clasificarea semnalelor EEG care au drept stimul lumini pâlpâitoare [8]. Această metodă matematică se bazează pe faptul că frecvența stimulilor este cunoscută și se pot calcula corelații între semnalul EEG colectat și cel de referință. CCA are avantajul că găsește un spațiu în care datele au corelație maximă și acolo se analizează aceste corelații. Comparativ cu calculul de corelație clasic care poate să rateze o legatură reală între date dacă nu se află într-un spațiu adecvat.

Din punct de vedere istoric, CCA a fost dezvoltat de matematicianul Harold Hotelling în mijlocul anilor 1930 [50], [51]. De atunci a fost folosit în multiple domenii, de la analiza făinii în funcție de caracteristicile grâului [52], până la probleme complexe de procesare de semnal [53]. Însă recent s-a remarcat utilitatea domenii precum BCI [9], selecție de trăsături [54], reducerea dimensionalității [55], etc.

Din punct de vedere matematic, algoritmul încearcă să găsească proiecții ale datelor de intrare astfel încât în noul spațiu datele să aibă corelație maximă. Acestă transformare de spațiu este făcută prin intermediul unor ponderi.

Să presupunem că avem 2 seturi de date de intrare X_a și respectiv X_b, cu mențiunea ca X_a și X_b sunt matrici de dimensiunile $n \times p$ respectiv $n \times q$. $\forall k \in \{1, \ldots, p\}$, vectorii rând $x_a^k \in \mathbb{R}^p$ și $x_b^k \in \mathbb{R}^q$ aparținând X_a respectiv X_b denotă seturile de observații. Vectorii coloană $a_i \in \mathbb{R}^n$ ($\forall i \in \{1, \ldots, p\}$) și $b_j \in \mathbb{R}^n$, ($\forall j \in \{1, \ldots, q\}$) reprezintă variabilele vector pentru cele n observații.

Așa cum am menționat anterior aceste date trebuie proiectate într-un alt spațiu. Acest lucru este făcut prin intermediul unei transformări liniare:

$$X_a w_a = z_a \quad \text{și} \quad X_b w_b = z_b, \tag{3.1}$$

unde: $w_a \in \mathbb{R}^p$, $z_a \in \mathbb{R}^n$, $w_b \in \mathbb{R}^q$, $z_b \in \mathbb{R}^p$. w_a, w_b se numesc ponderi (canonice) și z_a, z_b se numesc imagini in spațiul $\mathbb{R}^n$.

Constrângerile algoritmului sunt acelea că vectorii poziție ai imaginilor z_a și z_b sunt vectori normați la unitate și că unghiul dintre ele, $\theta \in [0, \pi/2]$, este minimizat [56]. Cosinusul între cele 2 imagini este dat de formula:

$$cos(z_a, z_b) = \langle z_a, z_b \rangle / \|z_a\| \|z_b\|, \tag{3.2}$$

unde $\langle z_a, z_b \rangle$ reprezintă produsul scalar dintre z_a și z_b, iar prin $\|z_a\|$ și $\|z_b\|$ s-au notat normele celor 2 imagini. După cum am menționat anterior o constrângere este aceea că norma trebuie să fie 1. Rezultă că formula se simplifică la:

$$cos(z_a, z_b) = \langle z_a, z_b \rangle \tag{3.3}$$

În continuare imaginile z_a și z_b ,obținute cu ponderile w_a și w_b, care oferă cel mai mic unghi între ele, notat θ_1, determină prima valoare de corelație canonică. Aceasta este egală cu $cos\theta_1$ [57]. Din punct de vedere matematic scriem astfel:

$$cos\theta_1 = \max_{z_a, z_b \in \mathbb{R}^n} \langle z_a, z_b \rangle \tag{3.4}$$

, cu constrângerea ca: $\|z_a\|_2 = 1$ și $\|z_b\|_2 = 1$

Fie z_a^1 și z_b^1 imaginile pentru care se găsește primul maxim. Pentru a găsi cea de-a doua pereche z_a^2 și z_b^2 care are cel mai mic unghi între ele, θ_2, se calculeaza complementele ortogonale ale primei perechi [58]. Acest procedeu se continuă până când nu se mai găsesc perechi.

Să presupunem că în acest mod se vor găsi r perechi care vor avea r unghiuri între ele $\theta_r \in [0, \pi/2]$, iar $r = 1, 2, \ldots, q$. Când $p > q$ unghiurile se pot găsi recursiv astfel:

$$cos\theta_r = \max_{z_a, z_b \in \mathbb{R}^n} \langle z_a^r, z_b^r \rangle, \tag{3.5}$$

$$\|z_a^r\|_2 = 1 \quad \|z_b^r\|_2 = 1 \tag{3.6}$$

$$\langle z_a^r, z_a^j \rangle = 0 \quad \langle z_b^r, z_b^j \rangle = 0, \tag{3.7}$$

$$\forall j \neq r; \quad j, r = 1, 2, \ldots, min(p, q).$$

Pentru a găsi vectorii poziție w_a și w_b care generează cosinusul maxim se poate folosi o serie de metode. Prima metodă este chiar cea propusă de creatorul algoritmului [51], fiind o metodă bazată pe rezolvarea unei probleme de valori proprii standard. O altă abordare este prin rezolvarea valorilor proprii generalizate [59]. Cea de a treia metodă constă în descompunerea valorilor proprii (*eng. Singular Value Decomposition – SVD*) așa cum a fost dezvoltată în [60].

3.3 Rezultate

Pentru a verifica separabilitatea celor 2 clase s-au folosit diverse arhitecturi de reţele neurale. Aşa cum s-a menţionat anterior trăsăturile oferite ca date de intrare constau în cele 8 corelaţii extrase cu algoritmul CCA. Aceste valori oferă informaţii asupra similitudinii dintre semnalele EEG şi sinuşii/coşinusii aferenţi frecvenţei stimulilor.

Având în vedere că pentru această aplicaţie clasificarea este binară, în cadrul experimentelor s-au folosit arhitecturi de o complexitate relativ redusă. Astfel că au fost testate reţele cu un număr de straturi ascunse variind de la 1 la 7. De asemenea numărul de neuroni pe fiecare strat a fost variat între 8 şi 256 (mai precis au fost folosiţi fie 8, fie 16, fie 32, fie 128, fie 256 de neuroni pe strat).

Activările care au fost testate sunt: *sigmoid*, *tanh* şi *arctan*. Acestea au fost folosite datorită faptului că sunt cunoscute să dea rezultate bune în special în probleme de clasificare [40], [61], dar şi în probleme de estimare [62]. Rezultatele cu funcțiile de activare care neagă componentele negative (de ex. ReLU, Leaky ReLU) nu au fost prezentate în această lucrare deoarece ofereau răspunsuri aleatorii (acuratețe în jurul valorii de 50%). Un motiv posibil pentru acest lucru s-ar putea afla în fenomenul numit *dying ReLU*. Acesta se referă la faptul că dacă prea multe ponderi iau valori sub 0 atunci rezultatul activării va fi 0, deci discriminarea datelor devine imposibilă.

Tehnica de optimizare folosită este gradient stocastic descendent (*eng. Stochastic Gradient Descent – SGD*) cu impuls (*eng. momentum*). Acest algoritm este unul popular în domeniul reţelelor neurale şi oferă, în general, rezultate satisfăcătoare [63], [64], [65], [66]. Metoda este o simplificare de la GD (Gradient Descent) în sensul că în loc să se calculeze gradientul pentru întreg lotul la fiecare iteraţie, se calculează gradientul doar pentru o valoare din lot aleasă aleator [67]. Prin urmare procesul de învăţare este stocastic şi depinde de aceste valori alese întâmplător. În acest mod se doreşte ca ecuaţia 3.8 pentru GD să se comporte relativ la fel ca 3.9 pentru SGD, având avantajul unui volum de calcul mult mai mic.

$$w_{t+1} = w_t - \eta \frac{1}{n} \sum_{i=1}^{n} \nabla_w \mathcal{L}(z_i, w_t) \tag{3.8}$$

$$w_{t+1} = w_t - \eta \nabla_w \mathcal{L}(z_t, w_t) \tag{3.9}$$

În aceste formule z se referă la perechea (x, y) însemnând (valoare de intrare – valoare de ieşire asociată), iar z_t este perechea aleasă la momentul epocii t.

Pe lângă SGD clasic, există şi varianta cu impuls. Această variantă îndreaptă mai repede costul către minim. O reprezentare vizuală a acestui fapt poate fi observată mai jos în figura 3.2. După cum se poate vedea SGD fără impuls oscilează mai mult până gaseşte minimul în timp ce SGD cu impuls ajunge mai repede la optim (într-un număr redus de iteraţii). Din punct de vedere matematic impulsul constă în adăugarea unei fracţii din ponderea precedentă la calculul ponderii actuale. Mai precis algoritmul se comporta dupa formula 3.10.

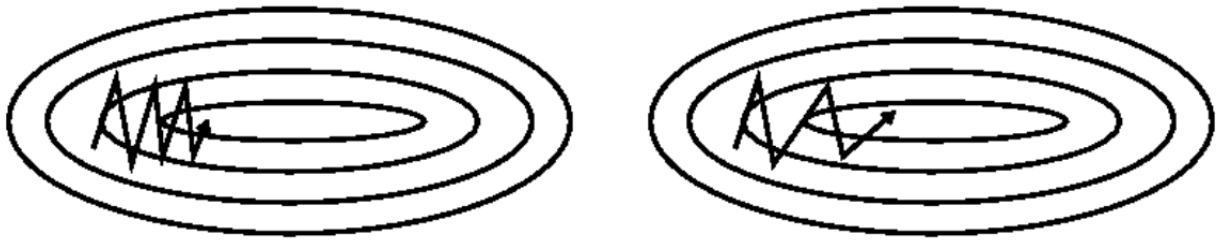

Figura 3.2: *SGD fără impuls la stânga şi cu impuls la dreapta*

$$W_t = W_{t-1} - (\beta V_{t-1} + \eta \nabla_w \mathcal{L}(Z, W)) \tag{3.10}$$

Unde β este parametrul care determină impulsul. De obicei acesta are valoarea 0.9. Dacă folosim formula de mai sus pentru toate datele atunci practic vom actualiza cu suma ponderată a valorilor W. Acest aspect face ca aproape tot timpul SGD cu impuls să fie mai bun decât varianta clasică. Motivul este acela că la SGD original pasul este afectat de zgomot (din cauză că se ia un exemplu aleator ca factor pentru actualizare), deci există riscul să o ia în direcţii greşite. Dacă folosim şi impuls (deci să luam în calcul valorile precedente), acest risc scade considerabil.

De asemenea, datorită dimensiunii relativ scăzute a bazei de date am folosit *batch size* de valoare 1 (un singur exemplu a fost evaluat pentru actualizarea parametrilor). În plus, pentru a asigura faptul că toate arhitecturile au fost testate în aceleaşi condiţii, numărul de epoci a fost întotdeauna 100.

În cele ce urmează vor fi prezentate rezultatele arhitecturilor testate folosind baza de date a cărei creare a fost descrisă în capitolul 4 - *Metoda propusă*. Aceasta a fost împărţită în 2: partea de antrenare dimensiune 350 şi partea de testare de dimensiune 39, echivalând unei împărţiri 90% cu 10%. După cum am menționat în capitolul 2, în timpul antrenării scopul rețelei este acela de a găsi ponderile care alocă valorilor de la intrare etichetele ideale. Datele pe care rețeaua „învață" se numesc lot de antrenare. Pentru a verifica posibilitatea de generalizare a funcției găsite, se păstrează separat câteva valori, numite lot de testare, care sunt trecute prin rețea. Dacă rezultatele sunt considerate bune, putem spune ca rețeaua a învățat și este capabilă să clasifice date noi.

Pentru funcţia de activare tanh au fost încercate arhitecturi cu 1, 2, 3, 4, 5, 6 şi 7 straturi ascunse. Straturile ascunse au fost tot timpul identice şi au avut numărul de neuroni variaţi conform valorilor din tabelele de mai jos (mai precis tabelele 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8). După cum se poate observa cel mai bun cost înregistrat pe test a fost 0.32 cu o acurateţe de 87.18%. Arhitectura care a dat acest rezultat are 5 straturi şi câte 16 neuroni pe strat. Totuşi această acurateţe nu este şi cea maximă la care s-a ajuns folosind tanh. Valoarea acesteia este de 89.74% dar la un cost de 0.36.

1 strat - funcţie de activare tanh

Nr. neuroni	antrenare		testare	
	cost	acurateţe	cost	acurateţe
8	0.52	74.86	0.38	84.62
16	0.54	72.57	0.38	84.62
32	0.53	73.43	0.38	84.62
64	0.52	72.86	0.37	82.05
128	0.51	73.71	0.37	84.62
256	0.52	72.57	0.38	84.62

Tabela 3.2: Rezultate–tanh (a)

2 straturi identice - funcţie de activare tanh

Nr. neuroni	antrenare		testare	
	cost	acurateţe	cost	acurateţe
8	0.54	72.29	0.36	84.62
16	0.56	72.57	0.35	84.62
32	0.53	73.71	0.38	84.62
64	0.53	72.29	0.37	84.62
128	0.51	72.29	0.37	84.62
256	0.52	72.57	0.38	84.62

Tabela 3.3: Rezultate–tanh (b)

Următoarea funcţie de activare testată a fost sigmoid. Numărul de straturi în acest caz a luat valorile: 1, 2, 3 sau 4. Deşi s-a dorit ca testele să fie identice cu cele pentru tanh, ţinând cont că rezultatele nu se mai îmbunătăţeau experimentele au fost oprite la 4 straturi ascunse. Distribuţia neuronilor este identică cu a cazului precedent, straturile ascunse fiind tot timpul

3 straturi identice - funcţie de activare tanh

Nr. neuroni	antrenare		testare	
	cost	acurateţe	cost	acurateţe
8	0.53	72.57	0.35	82.05
16	0.56	72.00	0.35	87.18
32	0.56	70.00	0.34	84.62
64	0.57	72.29	0.36	87.18
128	0.54	72.29	0.36	84.62
256	0.55	72.57	0.36	84.62

Tabela 3.4: Rezultate–tanh (c)

4 straturi identice - funcţie de activare tanh

Nr. neuroni	antrenare		testare	
	cost	acurateţe	cost	acurateţe
8	0.56	74.86	0.35	84.62
16	0.56	72.00	0.36	84.62
32	0.56	72.86	0.36	84.62
64	0.54	71.43	0.34	84.62
128	0.58	73.43	0.34	84.62
256	0.55	71.43	0.35	84.62

Tabela 3.5: Rezultate–tanh (d)

identice (vezi tabelele: 3.9, 3.10, 3.11, 3.12). În acest caz cel mai bun cost pe lotul de testare a fost 0.36 la o acurateţe de 84.62%. De asemenea s-a observat că odată cu creşterea numărului de straturi ascunse şi chiar şi cu a numărului de neuroni, reţeaua începe să se oprească din a învăţa şi să dea rezultate aleatorii pentru datele de antrenare.

În continuare a fost testat cazul în care funcţia de activare este arctan. De această dată numărul de straturi ascunse a fost: 1, 2, 3, 4, 5, 6 şi 7, iar numărul de neuroni a fost variat conform tabelelor: 3.13, 3.14, 3.15, 3.16, 3.17, 3.18 și 3.19. În acest caz costul minim găsit pe lotul de test este de 0.31 la o acurateţe de 84.62% cu 7 straturi ascunse şi 16 neuroni pe fiecare strat. Similar cu cazul tanh, nici această acurateţe nu este cea maximă, aceasta fiind 89.74% pentru un cost de 0.34. Acest lucru era de asteptat întrucât costul nu oferă o informație similară cu acuratețea. Aceasta din urmă fiind calculată conform formulei:

$$acurateţe = \frac{număr\ cazuri\ clasificate\ corect}{număr\ cazuri\ totale,} \tag{3.11}$$

iar funcția de cost folosită este *binary cross entropy* și se comportă după legea:

$$\mathcal{L} = -\frac{1}{n}\sum_{i=1}^{N} y_i \cdot log(\hat{y}) + (1 - y_i) \cdot log(1 - \hat{y}) \tag{3.12}$$

În domeniul rețelelor neuronale există 2 fenomene care pot să apară în timpul antrenării: *underfitting* și *overfitting*. Primul se referă la momentul în care învățarea a fost oprită prea repede raportat la numărul de epoci. După cum se poate observa și în figura 3.3, underfitting apare atunci când costul încă este în scădere, atât pentru lotul de antrenare cât și pentru lotul de test, dar nu s-a ajuns la un cost minim. Prin urmare, zona de subantrenare este caracterizată de acurateţe mică și pe lotul de învățare și cel de testare. Overfitting-ul reprezintă extrema de

5 straturi identice - funcţie de activare tanh

Nr. neuroni	antrenare		testare	
	cost	acurateţe	cost	acurateţe
8	0.65	55.43	0.35	84.62
16	0.58	64.86	0.32	87.18
32	0.58	70.57	0.33	87.18
64	0.57	71.43	0.35	82.05
128	0.58	72.00	0.35	84.62
256	0.54	74.57	0.35	84.62

Tabela 3.6: Rezultate–tanh (e)

6 straturi identice - funcţie de activare tanh

Nr. neuroni	antrenare		testare	
	cost	acurateţe	cost	acurateţe
8	0.62	64.00	0.36	89.74
16	0.64	62.00	0.33	87.18
32	0.60	66.57	0.33	87.18
64	0.58	71.71	0.33	87.18
128	0.58	68.57	0.34	82.05
256	0.59	67.14	0.35	84.62

Tabela 3.7: Rezultate–tanh (f)

sens opus față de underfitting. De această dată rețeaua a fost lăsată să învețe pentru un număr mare de epoci. În acest caz au fost găsiți parametri care se mulează aproape perfect pe datele de antrenare. Problema este că rețeaua nu a învățat o legătură reală dintre semnale, ci doar a reușit să mapeze datele de intrare la valorile aferente de ieșire. În acest caz costul pe lotul de test crește pentru că rețeaua nu are robustețe și nu gestionează bine date noi.

Analizând rezultatele prezentate în această licență, putem spune că în nicio variantă de arhitectură testată nu s-a ajuns la overfit. Totuși trebuie menționat că regula de salvare a modelului a fost: se păstrează cel care are costul cel mai mic pe lotul de test. Prin urmare toate rezultatele prezentate nu sunt afectate nici de underfit nici de overfit.

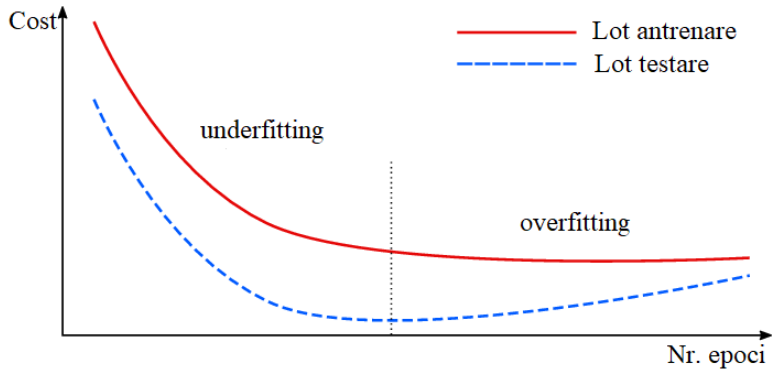

Figura 3.3: *Ilustrarea fenomenelor de underfitting și overfitting*

7 straturi identice - funcţie de activare tanh

Nr. neuroni	antrenare		testare	
	cost	acurateţe	cost	acurateţe
8	0.63	64.00	0.40	84.62
16	0.59	65.71	0.34	87.18
32	0.57	69.71	0.32	87.18
64	0.57	71.71	0.35	87.18
128	0.58	71.43	0.33	87.18
256	0.58	70.86	0.33	84.62

Tabela 3.8: Rezultate–tanh (g)

1 strat - funcţie de activare sigmoid

Nr. neuroni	antrenare		testare	
	cost	acurateţe	cost	acurateţe
8	0.51	72.86	0.38	84.62
16	0.51	74.29	0.38	84.62
32	0.52	72.86	0.38	84.62
64	0.53	72.00	0.38	84.62
128	0.58	70.00	0.38	82.05
256	1.83	52.57	0.72	64.10

Tabela 3.9: Rezultate–sigmoid (a)

2 staturi identice - funcţie de activare sigmoid

Nr. neuroni	antrenare		testare	
	cost	acurateţe	cost	acurateţe
8	0.51	72.00	0.36	84.62
16	0.51	74.00	0.37	84.62
32	0.54	70.29	0.41	84.62
64	0.52	72.57	0.40	79.49
128	0.55	72.00	0.41	84.62
256	0.64	63.43	0.55	71.79

Tabela 3.10: Rezultate–sigmoid (b)

3 staturi identice - funcţie de activare sigmoid

Nr. neuroni	antrenare		testare	
	cost	acurateţe	cost	acurateţe
8	0.70	51.43	0.65	64.10
16	0.70	50.00	0.65	64.10
32	0.70	52.57	0.65	64.10
64	0.71	50.86	0.65	64.10
128	0.70	50.57	0.65	64.10
256	0.70	50.57	0.65	64.10

Tabela 3.11: Rezultate–sigmoid (c)

4 staturi identice - funcţie de activare sigmoid

Nr. neuroni	antrenare		testare	
	cost	acurateţe	cost	acurateţe
8	0.69	53.43	0.65	64.10
16	0.71	48.29	0.65	64.10
32	0.70	49.71	0.65	64.10
64	0.70	55.71	0.65	64.10
128	0.70	50.86	0.65	64.10
256	0.70	48.86	0.65	64.10

Tabela 3.12: Rezultate–sigmoid (d)

1 strat - funcţie de activare arctan

Nr. neuroni	antrenare		testare	
	cost	acurateţe	cost	acurateţe
8	0.50	76.00	0.37	84.62
16	0.53	72.57	0.38	82.05
32	0.52	72.29	0.37	84.62
64	0.52	74.86	0.37	84.62
128	0.50	75.43	0.37	84.62
256	0.51	74.29	0.37	82.05

Tabela 3.13: Rezultate–arctan (a)

2 straturi identice - funcţie de activare arctan

Nr. neuroni	antrenare		testare	
	cost	acurateţe	cost	acurateţe
8	0.53	72.57	0.36	84.62
16	0.53	72.57	0.37	84.62
32	0.55	72.86	0.37	84.62
64	0.53	72.57	0.37	84.62
128	0.53	71.43	0.37	84.62
256	0.53	73.43	0.37	84.62

Tabela 3.14: Rezultate–arctan (b)

3 straturi identice - funcţie de activare arctan

Nr. neuroni	antrenare		testare	
	cost	acurateţe	cost	acurateţe
8	0.54	74.00	0.35	82.05
16	0.58	70.29	0.34	87.18
32	0.54	73.71	0.35	84.62
64	0.55	73.72	0.35	84.62
128	0.52	76.00	0.36	84.62
256	0.55	71.43	0.35	84.62

Tabela 3.15: Rezultate–arctan (c)

4 straturi identice - funcţie de activare arctan

Nr. neuroni	antrenare		testare	
	cost	acurateţe	cost	acurateţe
8	0.55	74.29	0.34	84.62
16	0.55	72.29	0.35	84.62
32	0.55	73.14	0.35	84.62
64	0.55	72.00	0.35	84.62
128	0.53	73.71	0.34	87.18
256	0.51	76.29	0.36	82.05

Tabela 3.16: Rezultate–arctan (d)

5 straturi identice - funcţie de activare arctan

Nr. neuroni	antrenare		testare	
	cost	acurateţe	cost	acurateţe
8	0.55	73.43	0.34	84.62
16	0.57	70.86	0.35	87.18
32	0.54	73.18	0.35	82.05
64	0.57	73.43	0.35	84.62
128	0.56	72.00	0.34	84.62
256	0.56	71.14	0.35	84.62

Tabela 3.17: Rezultate–arctan (e)

6 straturi identice - funcţie de activare arctan

Nr. neuroni	antrenare		testare	
	cost	acurateţe	cost	acurateţe
8	0.57	73.71	0.34	89.74
16	0.54	76.57	0.36	84.62
32	0.56	70.57	0.33	84.62
64	0.55	73.71	0.33	82.05
128	0.55	74.29	0.34	84.62
256	0.56	70.57	0.34	84.62

Tabela 3.18: Rezultate–arctan (f)

7 straturi identice - funcţie de activare arctan

Nr. neuroni	antrenare		testare	
	cost	acurateţe	cost	acurateţe
8	0.61	69.43	0.42	82.05
16	0.61	63.71	0.31	84.62
32	0.55	71.71	0.33	84.62
64	0.55	73.43	0.35	82.62
128	0.54	75.14	0.34	87.18
256	0.58	70.00	0.35	82.05

Tabela 3.19: Rezultate–arctan (g)

Capitolul 4
Concluzii

4.1 Concluzii generale

În cadrul acestui proiect s-a demonstrat faptul că este posibilă discriminarea zonei unui ecran căreia subiectul i-a acordat atenţie folosind doar semnale EEG și tehnici de învățare automată.

O parte importantă a acestei lucrări o constituie studiul detaliat al fenomenului fiziologic care determină undele electrice oscilante. Această documentare prealabilă ajută la determinarea pozițiilor electrozilor relevanți. Căștile de achiziție pentru encefalograme de pe piață au zeci sau chiar sute de electrozi. Dacă se vor folosi toți senzorii atunci dimensiunea datelor va fi inutil de mare și va îngreuna semnificativ întreg procesul de clasificare. O altă informație care este derivată din cunoașterea fenomenului fiziologic este alegerea unei rate de eșantionare adecvate. Astăzi se poate ajunge la frecvențe de eșantionare pentru EEG-uri de până la 20 kHz (frecvenţa maximă de eşantionare a sistemului Neuvo). Bineînțeles că la 20.000 de eșantioane pe secundă dacă se înmulțește cu numărul de electrozi folosiți, apoi se înmulțește cu numărul de exemple și se înmulțește cu perioada unei achiziții se va ajunge la vectori foarte mari și dificil de gestionat. Totuși, dacă se cunoaște frecvența maximă la care găsește semnalul de interes atunci se poate folosi teorema lui Nyquist pentru a afla frecvența de eșantionare adecvată fară a irosi spațiu de stocare pentru encefalogramele colectate.

Pentru a folosi paradigma SSVEP este necesar ca stimulii să varieze cu o frecvență constantă. Abaterea pâlpâirii trebuie să fie, ideal, de cel mult o milisecundă. În timpul efectuării părții practice a acestei licențe am aflat că frecvențele de stimuli care se folosesc uzual sunt dificil de implementat cu acuratețe ridicată. Acest impediment este cauzat de faptul că sistemul de operare face în paralel mult mai multe sarcini decât îi sunt vizibile în mod direct utilizatorului. De asemenea unele limbaje de programare tind să fie mai rapide decât altele. De pildă, `Python`, deși este popular în domeniul *data science*, este destul de lent comparativ cu familia de limbaje `C`. De aceea pentru crearea video-ului cu stimuli am ales să fac un formular descris în C# cu 2 butoane care își schimbă culoarea din negru în verde/roșu la fiecare jumătate de perioadă a stimulului respectiv. Chiar dacă am folosit un limbaj mai rapid, când am monitorizat momentele de timp la care se schimbă culoarea butonului am aflat că frecvența cerută în cod nu era respectată. Prin urmare, ca o soluție de compromis, am lăsat sistemul să se stabilizeze la o frecvență arbitrară și am considerat acele pâlpâiri, pe care le-am găsit relativ constante, ca fiind stimulii experimentului. În acest mod am ajuns la valorile de $4Hz$ și $10.63Hz$, cu mențiunea că prima frecvență este mai stabilă decât cea de a doua. Din acest motiv am folosit ca referință o matrice de sinuși și cosinuși de multipli de $4Hz$.

Înainte de a începe colectarea datelor a trebuit să asigur faptul că impedanţa de contact

între electrozi şi scalp are o valoare suficient de mică. Acest lucru se face prin aplicarea de gel electroconductor. Toţi electrozii sunt prevăzuţi cu orificii prin care se poate aplica substanţa conductoare. Aplicarea a fost făcută cu ajutorul unei seringi fără ac. În timpul achiziţiei s-au folosit 2 tipuri de gel[1]. De asemenea pentru a asigura faptul că subiectul nu oboseşte am luat pauze când acesta a cerut.

După ce am achiziţionat semnalele EEG, pasul următor a fost reprezentat de filtrarea acestora. Interferenţa cauzată de reţeaua de alimentare naţională este atât de puternică încât dacă se reprezintă grafic datele tot ce se poate observa este un semnal sinusoidal având frecvenţa de $50Hz$. Acest lucru este adevărat indiferent de tipul de stimul care a generat semnalul. Pentru a curăţa EEG-urile de acest zgomot s-a folosit un filtru *notch* implementat în limbajul de programare `Matlab`.

Odată ce am avut semnalele „curăţate" s-a putut trece la etapa în care electroencefalogramele se pregătesc pentru a fi date ca valori de intrare reţelelor neuronale. În acest sens am segmentat datele folosind ferestre de 2 secunde cu suprapunere de o secundă. Două secunde sunt suficente pentru a clasifica semnale SSVEP [8] iar suprapunerea a ajutat la augmentarea bazei de date. După segmentare am obţinut 339 de exemple (192 pentru stimulul verde şi 197 pentru stimulul roşu).

O metodă comună de prelucrare a semnalelor colectate conform paradigmei SSVEP este reprezentată de aplicarea algoritmului CCA [8], [68], [69]. *Canonical Correlation Analysis* este o metodă matematică care presupune extragerea de trăsături sub forma unor coeficienţi de corelaţie dintre semnal şi o referinţă. Avantajul SSVEP este acela că generează un răspuns electroencefalografic de aceeaşi frecvenţă (şi multipli) cu a stimulului căruia i-a fost acordat atenţie. Cunoscând acest lucru referinţa a fost construită ca o combinaţie de sinuşi şi cosinuşi având următoarele frecvenţe: 4, 8, 12 şi 16 Hz. După aplicarea algoritmului, pentru fiecare exemplu am obţinut un vector de corelaţii de dimensiune 8.

Ultima etapă a fost reprezentată de testarea diverselor arhitecturi de rețele neurale. În cadrul acestui pas au fost modificați hiperparametrii sistemului. Mai precis, s-a experimentat cu diverse valori pentru numărul de neuroni, de straturi și cu diverse funcții de activare. Acestea din urmă sunt: tanh, arctan și sigmoid. Cele 3 funcții de activare au fost alese deoarece au formă de S, ceea ce le face să crească separabilitatea datelor de la intrare. Cel mai bun rezultat a fost obținut folosind o arhitectură cu 6 staturi ascunse, 8 neuroni pe fiecare strat ascuns și fiecare strat (excluzând stratul de ieșire care a avut *softmax*) având funcția de activare arctan. Acuratețea de clasificare în acest caz pe lotul de test a fost de 89.74%.

În concluzie, această lucrare s-a ocupat cu realizarea unui sistem de clasificare a semnalelor EEG de tip SSVEP bazat pe rețele neuronale adânci. De asemenea întreg procesul, de la realizarea paradigmei experimentale, la achiziția semnalelor, la preprocesarea datele și până la crearea arhitecturilor de rețele neurale a fost acoperit în cadrul proiectului.

4.2 Contribuții personale

Pentru această lucrare de licență am adus următoarele contribuții personale:

- realizarea stării artei pentru clasificarea de semnale EEG;
- realizarea paradigmei experimentale pentru colectarea semnalelor de tip SSVEP;
- colectarea datelor folosind casca Quik-Cap, amplificator Neuvo și software-ul Curry 7;

[1]gelul electroconductor *„NuPrep"* s-a terminat în timpul celei de a doua sesiuni de achiziţie aşa că l-am folosit şi pe cel de la *Compumedics*.

- filtrarea semnalelor EEG;
- segmentarea datelor;
- aplicarea algoritmului CCA;
- testarea diverselor structuri de rețele neuronale.

4.3 Dezvoltări ulterioare

Proiectul descris în lucrarea de față este unul scalabil și drept dovadă am propus câteva îmbunătățiri ce pot apărea în viitor:

- achiziții ulterioare de date în aceleași condiții pentru a augmenta baza de date și astfel să crească relevanța experimentelor cu rețele neuronale;
- achiziții de date și cu alți subiecți în încercarea de a crea un sistem care este independent de utilizator;
- repetarea experimentelor cu un număr mai mare de stimuli;
- repetarea achiziției dar fără ca stimulii să pâlpâie; scopul ar fi acela de a face diferența între informația cu culoare (roșu sau verde) și pâlpâire și informația doar cu culoare. Un sistem capabil să facă discriminarea între stimuli fără a integra pâlpâirea ar fi mult mai ușor de folosit pentru că nu ar obosi subiectul la fel de repede;
- folosirea mai multor electrozi pentru a oferi mai multă informație sistemului de clasificare; de exemplu ar putea fi folosiți și electrozii plasați în zona lobului frontal întrucât acesta este responsabil cu cogniția [70];
- încercarea mai multor variante de CCA așa cum propune lucrarea [8];
- testarea altor algoritmi comuni pentru extragerea de trăsături (de exemplu: PCA (Principal Component Analysis), LDA, ICA (Independent Component Analysis), etc.);
- încercarea unor rețele de tip RNN – Recurrent Neural Network;
- încercarea unor rețele de tip SNN – Spiking Neural Network[2];
- colectarea datelor EEG și când subiectul nu privește niciun stimul pentru a putea face diferența dintre intenția de a comunica cu sistemul BCI și simpla privire a mediului înconjurător.

[2]SNN-urile reprezintă o familie separată de rețele neurale care se bucură de popularite în această perioadă - chiar în acest an [71], [72], [73], [74], [75], [76]; acestea sunt mai apropiate modului de funcționare al creierului uman deoarece folosesc impulsuri discrete pentru a procesa și transmite mai departe informația, comparativ cu, spre exemplu, funcțiile de activare specifice NN-urilor clasice care sunt continue. Pentru SNN-uri datele utile se regăsesc atât în momentul impulsului cât și în rata impulsurilor [71].

Bibliografie

[1] George E Dahl, Dong Yu, Li Deng, and Alex Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on audio, speech, and language processing*, 20(1):30–42, 2011.

[2] Thibaut Perol, Michaël Gharbi, and Marine Denolle. Convolutional neural network for earthquake detection and location. *Science Advances*, 4(2):e1700578, 2018.

[3] Rania Jammazi and Chaker Aloui. Crude oil price forecasting: Experimental evidence from wavelet decomposition and neural network modeling. *Energy Economics*, 34(3):828–841, 2012.

[4] Ch Jyosthna Devi, B Syam Prasad Reddy, K Vagdhan Kumar, B Musala Reddy, and N Raja Nayak. Ann approach for weather prediction using back propagation. *International Journal of Engineering Trends and Technology*, 3(1):19–23, 2012.

[5] K Muthukannan, P Latha, R Pon Selvi, and P Nisha. Classification of diseased plant leaves using neural network algorithms. *ARPN Journal of Engineering and Applied Sciences*, 10(4):1913–1919, 2015.

[6] Chuanqi Tan, Fuchun Sun, Tao Kong, Bin Fang, and Wenchang Zhang. Attention-based transfer learning for brain-computer interface. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1154–1158. IEEE, 2019.

[7] `https://www.bci-award.com/Home`, 2020.

[8] Masaki Nakanishi, Yijun Wang, Yu-Te Wang, and Tzyy-Ping Jung. A comparison study of canonical correlation analysis based methods for detecting steady-state visual evoked potentials. *PloS one*, 10(10), 2015.

[9] Lei Cao, Zhengyu Ju, Jie Li, Rongjun Jian, and Changjun Jiang. Sequence detection analysis based on canonical correlation for steady-state visual evoked potential brain computer interfaces. *Journal of neuroscience methods*, 253:10–17, 2015.

[10] Yong Jiao, Yu Zhang, Yu Wang, Bei Wang, Jing Jin, and Xingyu Wang. A novel multilayer correlation maximization model for improving cca-based frequency recognition in ssvep brain–computer interface. *International journal of neural systems*, 28(04):1750039, 2018.

[11] Xiaogang Chen, Yijun Wang, Shangkai Gao, Tzyy-Ping Jung, and Xiaorong Gao. Filter bank canonical correlation analysis for implementing a high-speed ssvep-based brain–computer interface. *Journal of neural engineering*, 12(4):046008, 2015.

[12] Ignas Martišius and Robertas Damaševičius. A prototype ssvep based real time bci gaming system. *Computational intelligence and neuroscience*, 2016, 2016.

[13] Bonkon Koo, Hwan-Gon Lee, Yunjun Nam, and Seungjin Choi. Immersive bci with ssvep in vr head-mounted display. In *2015 37th annual international conference of the IEEE engineering in medicine and biology society (EMBC)*, pages 1103–1106. IEEE, 2015.

[14] Walter Ormerod. Richard caton (1842–1926): pioneer electrophysiologist and cardiologist. *Journal of medical biography*, 14(1):30–35, 2006.

[15] Lindsay F Haas. Hans berger (1873–1941), richard caton (1842–1926), and electroencephalography. *Journal of Neurology, Neurosurgery & Psychiatry*, 74(1):9–9, 2003.

[16] Mario Tudor, Lorainne Tudor, and Katarina Ivana Tudor. Hans berger (1873-1941)–the history of electroencephalography, 2005.

[17] David Millett. Hans berger: From psychic energy to the eeg. *Perspectives in biology and medicine*, 44(4):522–542, 2001.

[18] Christopher J Davis, James M Clinton, and James M Krueger. Microrna 138, let-7b, and 125a inhibitors differentially alter sleep and eeg delta-wave activity in rats. *Journal of Applied Physiology*, 113(11):1756–1762, 2012.

[19] Vladyslav V Vyazovskiy and Irene Tobler. Theta activity in the waking eeg is a marker of sleep propensity in the rat. *Brain research*, 1050(1-2):64–71, 2005.

[20] Kuno Kirschfeld. The physical basis of alpha waves in the electroencephalogram and the origin of the "berger effect". *Biological cybernetics*, 92(3):177–185, 2005.

[21] Fiza Singh, Jaime Pineda, and Kristin S Cadenhead. Association of impaired eeg mu wave suppression, negative symptoms and social functioning in biological motion processing in first episode of psychosis. *Schizophrenia research*, 130(1-3):182–186, 2011.

[22] Rui Zhang, Peng Xu, Rui Chen, Fali Li, Lanjin Guo, Peiyang Li, Tao Zhang, and Dezhong Yao. Predicting inter-session performance of smr-based brain–computer interface using the spectral entropy of resting-state eeg. *Brain topography*, 28(5):680–690, 2015.

[23] Kazutaka Takahashi, Maryam Saleh, Richard D Penn, and Nicholas Hatsopoulos. Propagating waves in human motor cortex. *Frontiers in human neuroscience*, 5:40, 2011.

[24] John R Hughes. Gamma, fast, and ultrafast waves of the brain: their relationships with epilepsy and behavior. *Epilepsy & Behavior*, 13(1):25–31, 2008.

[25] `https://www.khanacademy.org/test-prep/mcat/organ-systems/neuron-membrane-potentials/a/neuron-action-potentials-the-creation-of-a-brain-signal`, 2020.

[26] Jonathan R Wolpaw, Niels Birbaumer, Dennis J McFarland, Gert Pfurtscheller, and Theresa M Vaughan. Brain–computer interfaces for communication and control. *Clinical neurophysiology*, 113(6):767–791, 2002.

[27] Dennis J McFarland, Charles W Anderson, K-R Muller, Alois Schlogl, and Dean J Krusienski. Bci meeting 2005-workshop on bci signal processing: feature extraction and translation. *IEEE transactions on neural systems and rehabilitation engineering*, 14(2):135–138, 2006.

[28] Fabien Lotte, Laurent Bougrain, Andrzej Cichocki, Maureen Clerc, Marco Congedo, Alain Rakotomamonjy, and Florian Yger. A review of classification algorithms for eeg-based brain–computer interfaces: a 10 year update. *Journal of neural engineering*, 15(3):031005, 2018.

[29] Fabien Lotte, Marco Congedo, Anatole Lécuyer, Fabrice Lamarche, and Bruno Arnaldi. A review of classification algorithms for eeg-based brain–computer interfaces. *Journal of neural engineering*, 4(2):R1, 2007.

[30] Yousef Rezaei Tabar and Ugur Halici. A novel deep learning approach for classification of eeg motor imagery signals. *Journal of neural engineering*, 14(1):016003, 2016.

[31] Floriana Pichiorri, Giovanni Morone, Manuela Petti, Jlenia Toppi, Iolanda Pisotta, Marco Molinari, Stefano Paolucci, Maurizio Inghilleri, Laura Astolfi, Febo Cincotti, et al. Brain–computer interface boosts motor imagery practice during stroke recovery. *Annals of neurology*, 77(5):851–865, 2015.

[32] Victor Lafuente, Juan M Gorriz, Javier Ramirez, and Eduardo Gonzalez. P300 brainwave extraction from eeg signals: An unsupervised approach. *Expert systems with applications*, 74:1–10, 2017.

[33] John Polich. Updating p300: an integrative theory of p3a and p3b. *Clinical neurophysiology*, 118(10):2128–2148, 2007.

[34] Terence W Picton. The p300 wave of the human event-related potential. *Journal of clinical neurophysiology*, 9(4):456–479, 1992.

[35] Erwei Yin, Timothy Zeyl, Rami Saab, Tom Chau, Dewen Hu, and Zongtan Zhou. A hybrid brain–computer interface based on the fusion of p300 and ssvep scores. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 23(4):693–701, 2015.

[36] Emmanuel K Kalunga, Sylvain Chevallier, Quentin Barthélemy, Karim Djouani, Eric Monacelli, and Yskandar Hamam. Online ssvep-based bci using riemannian geometry. *Neurocomputing*, 191:55–68, 2016.

[37] Solomon L Moshé, Emilio Perucca, Philippe Ryvlin, and Torbjörn Tomson. Epilepsy: new advances. *The Lancet*, 385(9971):884–898, 2015.

[38] Ali K Ibrahim, Hanqi Zhuang, Nurgun Erdol, and Ali Muhmed Ali. Eeg seizure detection by integrating slantlet transform with sparse coding. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 459–462. IEEE, 2017.

[39] Xiao-Wei Wang, Dan Nie, and Bao-Liang Lu. Emotional state classification from eeg data using machine learning approach. *Neurocomputing*, 129:94–106, 2014.

[40] S-J Han, MH Song, J Kim, W-S Lee, and H-K Lee. Classification of temporal bone pneumatization based on sigmoid sinus using computed tomography. *Clinical radiology*, 62(11):1110–1118, 2007.

[41] Yuanzhi Li and Yang Yuan. Convergence analysis of two-layer neural networks with relu activation. In *Advances in neural information processing systems*, pages 597–607, 2017.

[42] Boris Hanin. Universal function approximation by deep neural nets with bounded width and relu activations. *Mathematics*, 7(10):992, 2019.

[43] Sachin S Talathi and Aniket Vartak. Improving performance of recurrent neural network with relu nonlinearity. *arXiv preprint arXiv:1511.03771*, 2015.

[44] Mingyang Jiang, Yanchun Liang, Xiaoyue Feng, Xiaojing Fan, Zhili Pei, Yu Xue, and Renchu Guan. Text classification based on deep belief network and softmax regression. *Neural Computing and Applications*, 29(1):61–70, 2018.

[45] Antonia Creswell, Kai Arulkumaran, and Anil A Bharath. On denoising autoencoders trained to minimise binary cross-entropy. *arXiv preprint arXiv:1708.08487*, 2017.

[46] Reinhard Werth. Visual functions without the occipital lobe or after cerebral hemispherectomy in infancy. *European Journal of Neuroscience*, 24(10):2932–2944, 2006.

[47] H Aurlien, IO Gjerde, JH Aarseth, G Eldøen, B Karlsen, H Skeidsvoll, and NE Gilhus. Eeg background activity described by a large computerized database. *Clinical Neurophysiology*, 115(3):665–673, 2004.

[48] Louis Litwin. Fir and iir digital filters. *IEEE potentials*, 19(4):28–31, 2000.

[49] Natalia Y Bilenko and Jack L Gallant. Pyrcca: regularized kernel canonical correlation analysis in python and its applications to neuroimaging. *Frontiers in neuroinformatics*, 10:49, 2016.

[50] Harold Hotelling. The most predictable criterion. *Journal of educational Psychology*, 26(2):139, 1935.

[51] Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28(3-4):321–377, 12 1936.

[52] Frederick V Waugh. Regressions between sets of variables. *Econometrica, Journal of the Econometric Society*, pages 290–310, 1942.

[53] Stephan V Schell and William A Gardner. Programmable canonical correlation analysis: A flexible framework for blind adaptive spatial filtering. *IEEE transactions on signal processing*, 43(12):2898–2908, 1995.

[54] Toru Ogura, Yasunori Fujikoshi, and Takakazu Sugiyama. A variable selection criterion for two sets of principal component scores in principal canonical correlation analysis. *Communications in Statistics-Theory and Methods*, 42(12):2118–2135, 2013.

[55] Nan Lin, Guochang Wang, and Baoxue Zhang. Dimension reduction in functional regression using mixed data canonical correlation analysis. *Statistics and its interface*, 6(2):187–196, 2013.

[56] J Dauxois and GM Nkiet. Canonical analysis of two euclidean subspaces and its applications. *Linear Algebra and its Applications*, 264:355–388, 1997.

[57] Åke Björck and Gene H Golub. Numerical methods for computing angles between linear subspaces. *Mathematics of computation*, 27(123):579–594, 1973.

[58] Viivi Uurtio, João M Monteiro, Jaz Kandola, John Shawe-Taylor, Delmiro Fernandez-Reyes, and Juho Rousu. A tutorial on canonical correlation methods. *ACM Computing Surveys (CSUR)*, 50(6):1–33, 2017.

[59] David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664, 2004.

[60] Tetsuro Sakurai. Asymptotic expansions of test statistics for dimensionality and additional information in canonical correlation analysis when the dimension is large. *Journal of multivariate analysis*, 100(5):888–901, 2009.

[61] Lichao Mou, Pedram Ghamisi, and Xiao Xiang Zhu. Deep recurrent neural networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(7):3639–3655, 2017.

[62] Bolin Liao, Chuan Ma, Lin Xiao, Rongbo Lu, and Lei Ding. An arctan-activated wasd neural network approach to the prediction of dow jones industrial average. In *International Symposium on Neural Networks*, pages 120–126. Springer, 2017.

[63] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[64] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pages 315–323, 2013.

[65] Martin Zinkevich, Markus Weimer, Lihong Li, and Alex J Smola. Parallelized stochastic gradient descent. In *Advances in neural information processing systems*, pages 2595–2603, 2010.

[66] Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the twenty-first international conference on Machine learning*, page 116, 2004.

[67] Léon Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer, 2012.

[68] Yun Li, Guangyu Bin, Xiaorong Gao, Bo Hong, and Shangkai Gao. Analysis of phase coding ssvep based on canonical correlation analysis (cca). In *2011 5th International IEEE/EMBS Conference on Neural Engineering*, pages 368–371. IEEE, 2011.

[69] Gido Hakvoort, Boris Reuderink, and Michel Obbink. Comparison of psda and cca detection methods in a ssvep-based bci-system. *Centre for Telematics & Information Technology University of Twente*, 2011.

[70] Edward E Smith and John Jonides. Storage and executive processes in the frontal lobes. *Science*, 283(5408):1657–1661, 1999.

[71] Amirhossein Tavanaei, Masoud Ghodrati, Saeed Reza Kheradpisheh, Timothee Masquelier, and Anthony Maida. Deep learning in spiking neural networks. *Neural Networks*, 111:47–63, 2019.

[72] Abhronil Sengupta, Yuting Ye, Robert Wang, Chiao Liu, and Kaushik Roy. Going deeper in spiking neural networks: Vgg and residual architectures. *Frontiers in neuroscience*, 13, 2019.

[73] Sebastian Brusca, Giacomo Capizzi, Grazia Lo Sciuto, and Gianluca Susi. A new design methodology to predict wind farm energy production by means of a spiking neural network–based system. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, 32(4):e2267, 2019.

[74] Nikola K Kasabov. *Time-space, spiking neural networks and brain-inspired artificial intelligence*. Springer, 2019.

[75] Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019.

[76] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, Yuan Xie, and Luping Shi. Direct training for spiking neural networks: Faster, larger, better. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1311–1318, 2019.