

Determinarea fluxului neuronal din semnale electroencefalografice (EEG) in timpul rostirii cuvintelor izolate

Proiect 3

Coordonator stiintific: Drd. Ing. Luiza Rusnac

Student 1: Chisalescu Bogdan

Student 2: Chivu Gheorghe-Iulian

January 28, 2021

1 Scopul lucrarii

1.1 Scopul

Prezenta lucrare urmareste dezvoltarea unei solutii software orientata catre agenti inteligenti, capabila sa poata determina/recunoaste fluxul neuronal bazandu-se pe o serie de semnale electroencefalografice. In cazul de fata, fluxul neuronal este determinat de rostirea cuvintelor izolate.

1.2 Solutia propusa

Pentru a indeplini scopul lucrarii, am propus dezvoltarea unui algoritm de inteligenta artificiala supervizat, din clasa retelelor neuronale. Arhitectura retelei neuronale ce dorim sa o dezvoltam este de tipul clasificatorului multi-clasa.

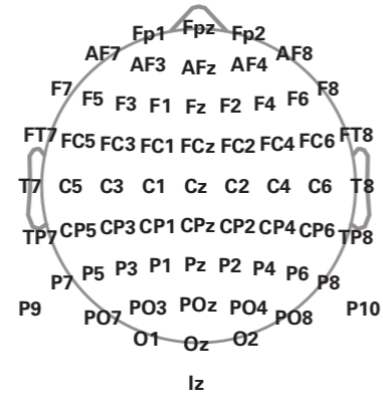
2 Semnalele EEG

Semnalele electroencefalografice (EEG) sunt o serie de monitorizari temporale asupra activitatii electrice a creierului uman. Activitatea ce este urmarita consta din fluctuatiile campului electromagnetic produs de activitatea sincronizata a populatiilor de neuroni din diverse sectiuni ale creierului. Inregistrarea si captarea semnalelor EEG este utila in analiza electroencefalografica deoarece stimuleaza abilitatea de a vizualiza si surprinde in timp schimbarile aparute atunci cand un subiect uman dezvoltă procese cognitive.

Pentru a putea capta semnalele EEG, se monteaza pe capul subiectului o casca ce contine o serie de dispozitive sensibile la fluctuatiile campului electric, numite electrozi. Subiectul este apoi instruit sa efectueze anumite activitati cognitive, timp in care se inregistreaza activitatea neuronală. Electrozii sunt dispusi in asa fel incat sa fie acoperita o arie cat mai mare a creierului si sunt de obicei grupati astfel incat un grup de electrozi sa inregistreze activitatea unui lob cerebral. Fiindca semnalele captate au valori mici, de ordinul microvoltilor (μV) acestea sunt amplificate, mai apoi digitalizate si filtrate. Fiecare electrod corespunde unui canal electroencefalografic.



(a) Subiectul uman in timpul inregistrării EEG



(b) Amplasamentul electrozilor

Din punct de vedere spectral putem considera semnalele EEG ca fiind compuse din 5 game/benzi de frecventa: Delta (1-4 Hz), Theta (4-7 Hz), Alpha (7-12 Hz), Beta(12-30 Hz) si Gamma (30-100 Hz).

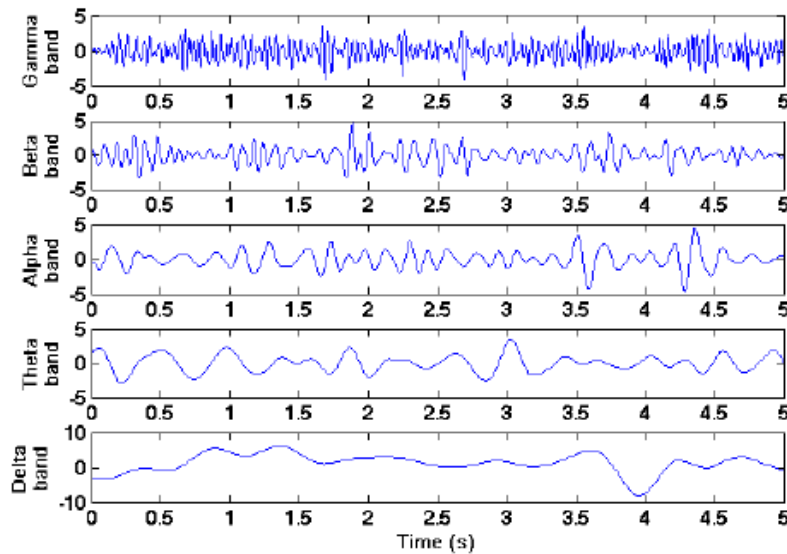


Figure 1: Benzile semnalelor EEG

3 Setul de date

Pentru aceasta lucrare ne-am folosit de setul de date pus la dispozitie de Departamentul de Stiinta Calculatoarelor a Universitatii din Toronto: The KARA ONE Database. Acest set de date a fost achizitionat in urma unui studiu bazat pe aceleasi principii ca cele descrise in sectiunea semnalelor EEG. O casca speciala a fost plasata pe capul subiectilor, ce sacneaza 64 de canale si are electrozii plasati dupa sistemul 10-20; toate datele au fost amplificate folosind un amplificator SynAmps RT si esantionate la o frecventa de esantionare $F_S = 1$ KHz. Timp de 30 pana la 40 de minute subiectii si-au imaginat rostirea dar au si rostit 7 foneme si 4 cuvinte.

In aceasta lucrare am preluat semnalele EEG filtrate ce au rezultat in urma studiului. Au rezultat astfel un numar de 993 de observatii si 64 de canale pentru fiecare observatie. Timpul alocat pentru o observatie a fost de 5 secunde. Stiind frecventa de esantionare de 1 KHz ne rezulta aproximativ 5000 de esantioane pentru fiecare canal.

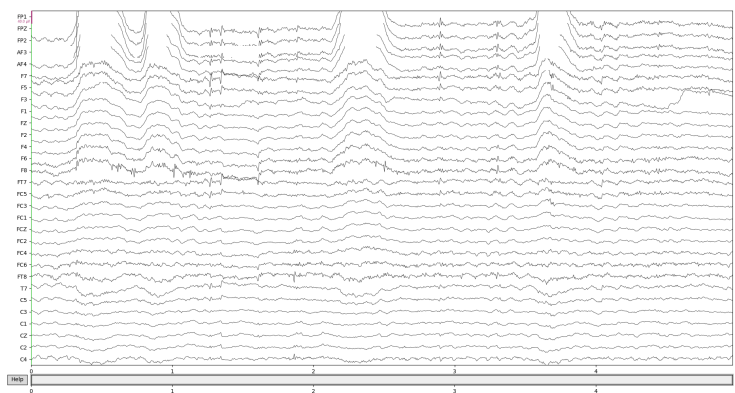


Figure 2: O parte din canalele EEG ale unei observatii

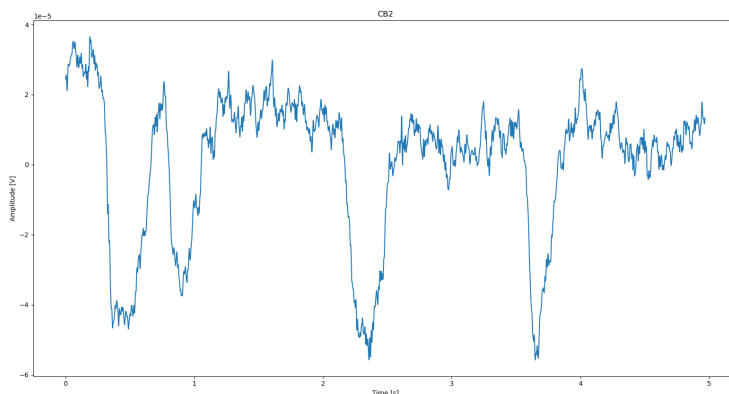


Figure 3: Canalul 'CB2' al unei observatii intins pe durata de aproximativ 5 secunde

Avand la dispozitie toate datele necesare, putem construi structura pe care o vor avea datele de intrare ale clasificatorului. Numarul de observatii este de 993, iar numarul de canale EEG pentru fiecare observatie este de 64, avand in vedere ca la intrarea clasificatorului vom introduce cate o observatie, vom considera toate cele 64 de canale ca trasaturi, canalele fiind aici reprezentate de esantioanele lor. Un canal contine aproximativ 5000 de esantioane, insa acest numar nu este constant pe intreg setul de date. Considerand faptul ca, in timpul studiului subiectul a desfasurat aceeasi activitate cognitiva de mai multe ori in decursul celor 5 secunde ale unei observatii, am luat decizia de a prelua esantioanele de la indicele 499 pana la 4499, deci un numar total de 4000 de esantioane.

Cele 64 de canale, acum a cate 4000 de esantioane fiecare, vor fi liniarizate. In acest caz datele de intrare vor avea forma $(993 \times 4000 \cdot 64) = (993 \times 256,000)$, adica 993 de observatii a cate 256,000 de trasaturi.

4 Analiza Componentelor Principale

Deoarece datele de intrare sunt semnificative ca dimensiune, iar pentru aceasta lucrare am dispus de resurse relativ modeste de calcul, am luat decizia de a selecta doar canalele EEG pe care le-am considerat importante si mai apoi de a aplica o analiza a componentelor principale .

Canalele EEG pe care le-am considerat cele mai importante si de care am dispus in setul de date sunt urmatoarele: C5, C3, C1, CZ, C2, C4, C6, CP5, CP3, CP1, CPZ, CP2, CP4, CP6, P7, P5, P3, PZ, P2, P4, P6, P8, FC5, FC3, FC1, FCZ, FC2, FC4, FC6, F7, F5, F3, F1, FZ, F2, F4, F6, F8, AF3, AF4. In urma acestei selectii, din cele 64 de canale initiale vor ramane 40, ceea ce ne modifica structura datelor de la $(993 \times 256,000)$ la $(993 \times 160,000)$.

Analiza componentelor principale (PCA) este o metoda prin care se poate reduce dimensionalitatea datelor de intrare pastrand in acelasi timp cea mai mare cantitate din informatie. Algoritmul PCA reprezinta o modalitate de implementare a transformatei Karhunen–Loève. Pentru a reduce dimensionalitatea datelor dar si de a pastra cat mai multa informatie este necesara o transformare liniara cu proprietati de unitaritate, conservare si compactare a energiei. Transformata Karhunen–Loève ne ofera aceste proprietati. Avand la dispozitie un vector $\xi = [\xi(0) \ \xi(1) \ \dots \ \xi(N-1)]^T \in \mathbb{R}_{(1 \times n)}$ reprezentand datele ce vor fi transformate, putem gasi reprezentarea vectorului in spatiul transformat ca fiind $\eta = L\xi$, unde L este o transformare liniara. Pentru a determina transformarea ne vom intoarce in spatiul original: $\tilde{\xi} = L^T\eta$, transformarea inversa este transpusa lui L datorita proprietatii de unitaritate iar $\tilde{\eta}$ este o aproximatie a lui η deoarece ne dorim ca in urma transformarii sa reducem din componentele vectorului η . In acest mod, putem determina eroarea de transformare in sensul erorii patratice medii ca fiind $\varepsilon = \sqrt{\|\xi - \tilde{\xi}\|^2}$. Putem gasi transformarea L daca minimizam expresia erorii astfel incat $\epsilon = \epsilon_{min}$. Se poate demonstra matematic ca transformarea L este formata din vectorii propri ai matricii de

covarianta a vectorului ξ asezati pe coloana si ordonati descrescator dupa valorile proprii corespunzatoare.

Transformata Karhunen–Loève, gaseste o baza vectoriala de reprezentare astfel incat datele sa se grupeze de-a lungul a cat mai multe axe pe care varianta este maxima. Pe acele axe pe care varianta nu este maxima, putem inlocui valorile din setul de date cu media axei respective, iar in cazul in care media a fost extrasa inaintea transformarii, devenind astfel zero, putem elimina componentele respective.

In cazul nostru, vectorul ξ este un vector ale carui elemente sunt la randul lor vectori n -dimensionali, deci vom aplica algoritmul PCA asupra unei matrici, a carei forma este determinata de setul de date. **Deoarece dispunem de sisteme de calcul cu capacitate de stocare relativ modesta, am luat decizia de a media cele 4000 de esantioane ale fiecarui canal EEG pe ferestre de cate 10 esantioane fara suprapunere.** In urma medierii, structura setului de date se modifica de la (993 x 160,000) la (993 x 16,000).

Dupa multiple incercari esuate, am ajuns la concluzia ca setul de date este inca mult prea mare pentru a-l putea procesa cu puterea de calcul si capacitatea de stocare de care dispunem, astfel ca am luat decizia de a clasifica semnalele pentru 2 clase in loc de 11 cum prevedea setul de date initial. Astfel, structura datelor se modifica de la (993 x 16,000) la (182 x 16,000), dimensiunea de 182 fiind justificata de numarul de observatii insumat pentru cele doua clase.

In urma aplicarii algoritmului PCA asupra setului de date cu structura de (182 x 16,000) am putut reduce numarul de componente de la 16,000 la 137. Structura finala a setului de date ce vor reprezenta datele de intrare pentru clasificator va avea forma (182 x 137).

5 Clasificatorul

In prima parte a lucrarii ne-am propus sa dezvoltam un clasificator multi-clasa, insa avand in vedere dificultatile tehnice intalnite ne-am rezumat la dezvoltarea unui clasificator ce poate distinge intre doua clase. Arhitectura retelei neuronale devine in acest fel de tipul unui perceptron multi-nivel (MLP). Perceptronul multi-nivel sau multi-strat se compune dintr-o multitudine de straturi computationale succesive formate la randul lor din unitati computationale individuale denumite neuroni. MLP-ul este format dintr-un strat de intrare ce se conecteaza cu urmatorul prin intermediul unor ponderi, urmatoarele straturi dupa cel de intrare se vor denumi straturi ascunse iar ultimul strat va fi cel de iesire. In arhitectura aleasa de noi toti neuronii de pe stratul l sunt in totalitate conectati cu cei de pe stratul $l + 1$. La nivelul neuronilor au loc doua tipuri de computatii, una liniara, urmata mai apoi de una neliniara. Vom denumi si nota in continuare: L = numarul total de straturi; $n^{[l]}$ = numarul de neuroni de pe stratul l ; $w_{ji}^{[l]}$ = ponderea ce conecteaza neuronul i de pe stratul $l - 1$ cu neuronul j de pe stratul l . $b_i^{[l]}$ = termenul liber i de pe stratul l ; $E(\hat{y}, y)$ = functia de eroare; $J(w, b)$ = functia de cost.

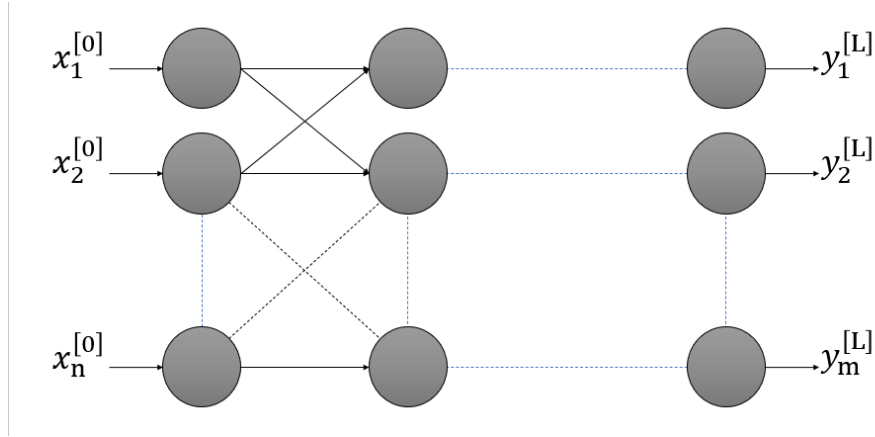


Figure 4: Arhitectura generala a clasificatorului multi-clasa

Algoritmul de invatare al rețelei cuprinde doua etape: etapa de forward propagation si etapa de backpropagation, prima reprezinta computatia efectuata de retea pentru a putea clasifica valorile de pe stratul de intrare, iar cea de-a doua este necesara pentru corectia ponderilor astfel incat predictia/clasificarea rețelei sa se imbunatateasca.

Pentru a putea beneficia de avantajul de timp oferit de algoritmii de calcul paraleli, am decis sa vectorizam operatiile de calcul ale rețelei. Am grupat astfel ponderile si termenii liberi in seturi de matrici corespunzatoare computatiilor efectuate la propagarea intre straturi. Partea de calcul neliniara efectuata de neuroni este descrisa de o functie de activare, in aceasta lucrare am folosit ca functii de activare: Sigmoid, tangenta hiperbolica si ReLU (Rectified Linear Unit). Ca metoda de optimizare am ales metoda gradientului descendent. Deoarece dorim sa clasificam doua clase, vom considera simbolurile ce le vor reprezenta ca 0 si 1. Ne-am definit deci, o functie de eroare a stratului de iesire in concordanta cu entropia canalului informational binar:

$$E(\hat{y}, y) = y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})$$

Pentru a afla eroarea asupra intregului set de date am definit functia de cost ca fiind:

$$J(w, b) = -\frac{1}{m} \sum_{i=1}^m (y^{(i)} \log(a^{[L](i)}) + (1 - y^{(i)}) \log(1 - a^{[L](i)}))$$

Din pacate nu am reusit sa facem retea sa functioneze pe actualul set de date. Chiar daca funtia de cost are aliura dorita, rata sa de descrestere nu este semnificativa.

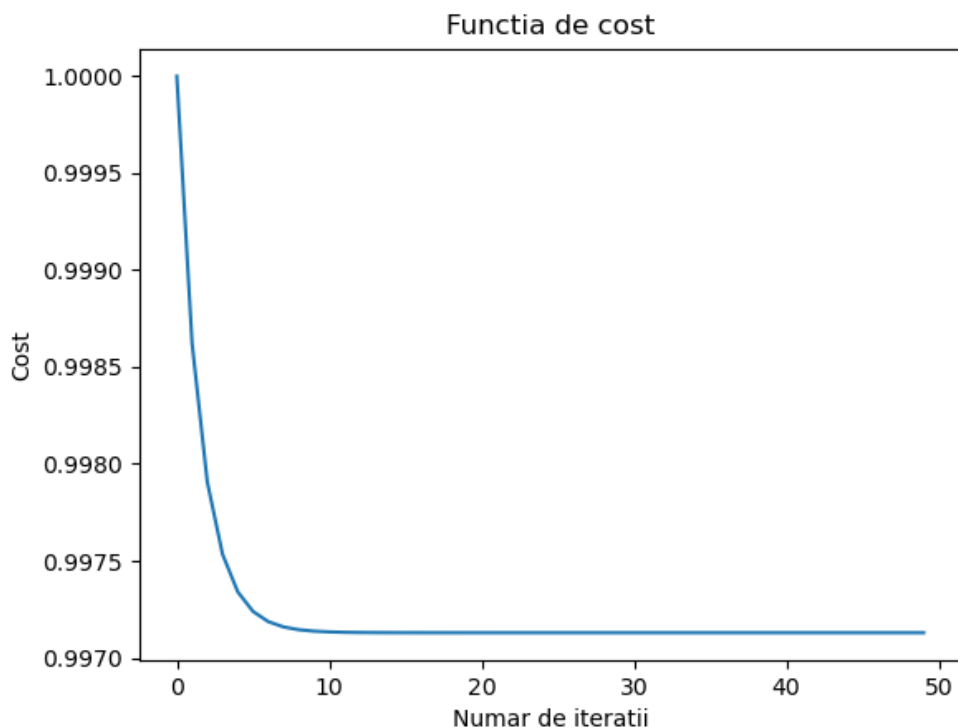


Figure 5: Descresterea functiei de cost pentru intregul set de date

6 Bibliografie

References

- [1] Cursul de Decizie si Estimare in Prelucrarea Informatiei, Prof. dr. ing. Mihai Ciuc, Facultatea de Electronica, Telecomunicatii si Tehnologia Informatiei.
- [2] Cursul de Recunoasterea Formelor si Inteligenta Artificiala, Prof. dr. ing. Victor Neagoe, Facultatea de Electronica, Telecomunicatii si Tehnologia Informatiei.
- [3] Cursul de Tehnici de Optimizare in Programare, Prof. dr. ing. Ovidiu Grigore, Facultatea de Electronica, Telecomunicatii si Tehnologia Informatiei.
- [4] Prelucrarea statistică a semnalelor , Mihai Ciuc, Constantin Vertan , MatrixRom 2005

- [5] Cursul "Neural Networks and Deep Learning" , Andrew Ng,
<https://www.coursera.org/learn/neural-networks-deep-learning?specialization=deep-learning>
- [6] Analyzing Neural Time Series Data, Theory and Practice, Mike X Cohen, The MIT Press Cambridge, Massachusetts London, England
- [7] <http://www.cs.toronto.edu/~complingweb/data/karaOne/karaOne.html>
- [8] <https://imotions.com/blog/eeg-books/>
- [9] <https://pypi.org/project/mne/>
- [10] https://mne.tools/dev/auto_tutorials/raw/plot_10_raw_overview.html#time-sample-number-and-sample-index
- [11] https://en.wikipedia.org/wiki/Cross-covariance_matrix
- [12] https://en.wikipedia.org/wiki/Principal_component_analysis#Computing_PCA_using_the_covariance
- [13] <https://numpy.org/doc/stable/reference/generated/numpy.linalg.eig.html>
- [14] pmaweb.caltech.edu/physlab/lab_21_current/Ph21_5_Covariance_PCA.pdf
- [15] <https://numpy.org/doc/stable/reference/generated/numpy.sort.html>
- [16] https://www.youtube.com/channel/UCUR_LsXk7IYyueSnXcNextQ/videos
- [17] https://www.researchgate.net/figure/The-five-frequency-bands-of-EEG-signal_fig1_275830679