

Laborator 5

Rețelele neuronale reprezintă un model matematic inspirat de principiile funcționării unui creier biologic. Creierul uman are în componență zeci de miliarde de *neuroni*, fiecare neuron fiind conectat cu zeci de mii de alți neuroni. Conexiunile dintre neuroni se realizează prin *sinapse*. Simplificând, un neuron funcționează în felul următor (Figura 1 - stânga): neuronul primește *semnale electrice* de la *axonii* (pre-sinapse) altor neuroni, ce trec prin sinapse, prin intermediul *dendritelor* (post-sinapse), procesează aceste semnale iar dacă semnalul electric este suficient de puternic neuronul se *activează* și emite un semnal electric prin intermediul axonilor.

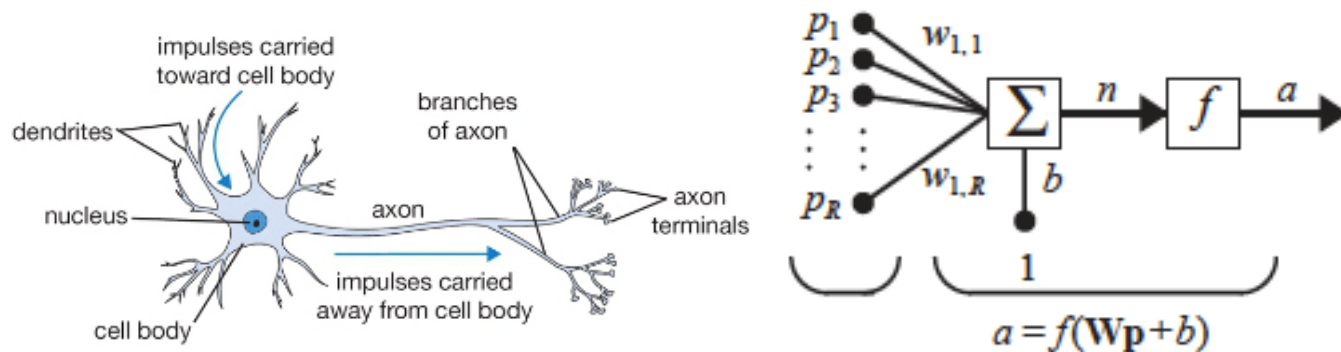


Figura 1. Stânga: structura unui neuron biologic. Dreapta: structura unui neuron artificial (perceptron)

Perceptronul (Figura 1 - dreapta) reprezintă modelul matematic al unui neuron. Semnalele electrice primite de la alți neuroni sunt modelate ca *valori numerice de intrare* (intrări - p_1, p_2, \dots, p_R), procesarea semnalelor este modelată printr-o *funcție de integrare* (de obicei funcția de integrare liniară = suma) aplicată intrărilor care sunt ponderate (cu ponderile $w_{1,1}, w_{1,2}, \dots, w_{1,R}$), activarea neuronului este modelată de o *funcție de transfer* f . Suma poate fi scrisă ca produsul dintre matricea W (conținând ponderile $w_{1,1}, w_{1,2}, \dots, w_{1,R}$) și vectorul coloană p (conținând intrările p_1, p_2, \dots, p_R). Perceptronul are bias-ul b (starea curentă a neuronului), care este adunat intrărilor ponderate, rezultând argumentul funcției de transfer.

Programele demonstrative **nnd2n1** și **nnd2n2** ilustrează un perceptron cu o singură sau două intrări ($R = 1$ respectiv $R = 2$), funcția de integrare liniară și diverse funcții de transfer. Rulați cele două programe familiarizându-vă cu noțiunile prezentate.

Exercițiul 1: Reprezentați grafic funcțiile de transfer (*hardlim*, *hardlims*, *purelin*, *satlin*, *satlins*, *logsig*, *tansig*) exemplificate în programul demonstrativ. Reprezentați de asemenea și alte funcții de tranfer precum: *poslin*, *softmax*, *tribas*, *radbas*, etc.

Programul demonstrativ **nnd3pc** prezintă un exemplu de clasificare folosind o rețea cu un perceptron. O mașină vrea să clasifice fructele (mere sau portocale) care vin pe o banda rulantă echipată cu un senzor ce măsoară trei caracteristici: forma, textura, greutatea. Caracteristicile sunt cuantificate prin valori numerice continue între -1 și 1 (senzorul care măsoară forma unui

fruct returnează o valoare apropiată de 1 dacă fructul are forma rotundă și aproape de -1 dacă fructul are formă eliptică; senzorul care măsoară textura unui fruct returnează o valoare apropiată de 1 dacă fructul are textură fină și aproape de -1 dacă fructul are textură aspră; senzorul care măsoară greutatea unui fruct returnează o valoare apropiată de 1 dacă fructul are o greutate > 0.25 kg și aproape de -1 dacă fructul are o greutate < 0.25 kg). Pe baza caracteristicilor măsurate, mașina vrea să clasifice fructele în una din cele două clase: mere (clasa 1) sau portocale (clasa -1). Un exemplu tipic de portocală are caracteristicile $[1, -1, -1]$ iar un exemplu tipic de măr are caracteristicile $[1, 1, -1]$.

Codul Matlab care implementează rețeaua este următorul:

```
net = perceptron;%creeaza o retea de perceptroni
net = configure(net,[0;0;0],0);%configureaza rețeaua - are ca input vectori  $1 \times 3$  si un output
net.IW{1} = [0 1 0];%setam ponderile pentru intrari
net.b{1} = 0;%setam bias-ul
view(net);%vizualizeaza rețeaua
```

Exercițiul 2: Generați aleator 10 exemple de vectori de caracteristici (considerați fiecare caracteristică între -1 și 1) și simulați comportamentul rețelei (folosiți metoda sim pentru rețeaua cu un singur perceptron *net* definită de codul Matlab de mai sus).

Exercițiul 3: Generați un nor de 1000 de puncte (x_i, y_i) în pătratul de dimensiuni $[-1, 1] \times [-1, 1]$ și etichetați fiecare punct (x_i, y_i) cu eticheta 1 dacă $x_i - y_i \geq 0$ sau cu eticheta 0 altfel. Plotați punctele în plan asociind o culoare punctelor cu aceeași etichetă. Scrieți codul Matlab al unei rețele cu un perceptron care clasifică corect aceste puncte.

Programul demonstrativ **nnd4db** ilustrează o problemă binară de clasificare folosind un perceptron. Rulați acest program și încercați să realizați ce impact are asupra dreptei de separare a celor două clase valorile ponderilor w și a bias-ului b .

Exercițiul 4: Scrieți codul Matlab care implementează un perceptron ce rezolvă corect problema de separare în cazul în care avem 4 puncte $(0,0)$, $(1,0)$, $(0,1)$, $(1,1)$ iar etichetele lor sunt date de operațiile AND sau OR aplicate componentelor punctelor. Puteți scrie un perceptron care rezolvă corect problema pentru cazul XOR?