

Laborator 6: Aplicații

1. Scrieți un program Matlab (fișier script) în care să implementați antrenarea unui perceptron cu funcția de transfer heaviside care să clasifice fără eroare datele mulțimii de antrenare liniar separabile de mai jos, formată din perechi de forma: vectori de intrare bidimensionali și etichete corespunzătoare de 1 și 0:

$$S = \left\{ \left((-0.5, -0.5)^T, 1 \right), \left((-0.5, +0.5)^T, 1 \right), \left((-0.3, -0.5)^T, 0 \right), \left((-0.1, +1)^T, 0 \right), \left((-40, 50)^T, 1 \right) \right\}$$

Cerințe:

- a. Să se definească un perceptron cu regula de învățare "learnp" și să se antreneze folosind funcția de antrenare "adapt" cât timp eroarea de clasificare este nenulă.
- b. Să se afișeze punctele din mulțimea de antrenare și dreptele de separare obținute după fiecare pas din procesul de antrenare.

Indicație:

Pentru a reprezenta numai dreapta de separare de la iterația curentă se poate folosi funcția $H = \text{plotpc}(W, B, H)$ (vezi help-ul) cu trei argumente pentru a șterge dreapta de separare de la pasul anterior.

Pentru a vizualiza dreapta de decizie de la pasul curent se poate folosi funcția "pause(nr_secunde)".

- c. Să se afișeze pasul (iterația) la care s-a obținut dreapta care clasifică fără eroare datele mulțimii de antrenare.
- d. Punctele a, b, c de mai sus, folosind un perceptron cu regula de învățare "learnpn".
- e. Comparați rezultatele obținute (privind convergența algoritmilor) folosind un perceptron cu regula de învățare "learnp" și respectiv unul cu regula de învățare normalizată "learnpn".

2. Implementați în MATLAB varianta **batch** a algoritmului de antrenare a perceptronului (fără a utiliza funcțiile MATLAB **adapt** sau **train**) pentru următoarele mulțimi de antrenare:

i.

$$S = \left\{ \left((0, -0.5)^T, -1 \right), \left((0, 0)^T, -1 \right), \left((0, 0.5)^T, -1 \right), \left((0.5, 0)^T, -1 \right) \right\} \cup \left\{ \left((0.5, 0.5)^T, 1 \right), \left((1, 0)^T, 1 \right), \left((1, 0.5)^T, 1 \right), \left((1, 1)^T, 1 \right) \right\}$$

ii.

$$S = \left\{ \left((0, -0.5)^T, -1 \right), \left((0, 0)^T, -1 \right), \left((0, 0.5)^T, -1 \right), \left((0.5, 0)^T, -1 \right) \right\} \cup \left\{ \left((0.5, 0.5)^T, 1 \right), \left((1, 0)^T, 1 \right), \left((1, 0.5)^T, -1 \right), \left((1, 1)^T, 1 \right) \right\}$$

- a. Repetați algoritmul de antrenare de un număr maxim de epoci (pași) de 1000. În momentul determinării ponderilor care dau direcția dreptei de separare, care clasifică corect punctele din mulțimea de antrenare, încheiați procesul de antrenare și plotați punctele din mulțimea de antrenare și dreapta de separare.
- b. Pe baza antrenării într-un număr maxim de 1000 de epoci, afișați dacă mulțimea de antrenare este liniar separabilă (dacă s-a determinat dreapta care clasifică corect punctele) sau neliniar separabilă (dacă după 1000 de epoci nu a fost găsită dreapta care clasifică corect punctele) folosind funcția **disp**. Dacă mulțimea de antrenare nu este liniar separabilă determinați dreapta care misclasifică cele mai puține puncte și afișați punctele misclasificate.
- c. Studiați convergența algoritmului perceptronului, reprezentând grafic iterația versus valoarea cosinusului dintre doi vectori de ponderi obținuți la iterații consecutive.

3. Implementați în Matlab algoritmul de învățare al perceptronului varianta batch și determinați dreptele de separare care clasifică fără eroare datele mulțimilor de antrenare liniar separabile de mai jos:

- a. Se consideră ca date de intrare $m=50$ ($m=100$) de puncte bidimensionale distribuite uniform pe $[-1,1] \times [-1,1]$ repartizate în două clase astfel:

- dacă punctul se află în semiplanul pozitiv în raport cu dreapta $2x - y = 0$ îi asociem clasa 1
- dacă punctul se află în semiplanul negativ în raport cu dreapta $2x - y = 0$ îi asociem clasa - 1.

b. Considerăm aceeași mulțime de antrenare ca mai sus, presupunând, în plus, că punctele se află la o distanță strict mai mare de 0.25 față de dreapta $2x - y = 0$.

Comparați rezultatele obținute (privind convergența algoritmilor) pentru mulțimile de antrenare considerate la punctele a) si b). (Comparați numărul de epoci necesare pentru a obține dreptele de separare care clasifică fără eroare datele din mulțimile de antrenare de mai sus?)