

## Test 2 Laborator

*Redenumiți folderul în care lucrați cu numele vostru și grupa (spre exemplu AlexeBogdan-232). Veți scrie codul Matlab al soluției voastre în fișierul soluție.m. La sfârșit veți pune pe stick acest folder conținând soluția voastră.*

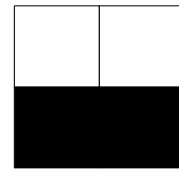
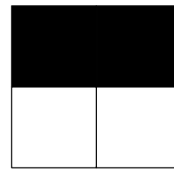
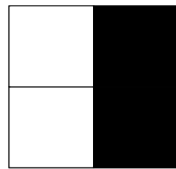
Considerăm imaginile pătratice binare de dimensiuni  $n \times n$  în care fiecare pixel ia două valori : 0 (alb) sau 1 (negru). Fiecare imagine cu  $n^2$  pixeli poate fi reprezentată printr-un vector cu  $n^2$  elemente.

Pentru un  $n$  fixat, considerăm mulțimea  $X_n^0$  formată din toate imaginile care reprezintă o linie verticală: există o coloană (dintre cele  $n$ ) cu toți pixelii ce au valoarea = 1, restul pixelilor au valoarea = 0. Etichetăm fiecare exemplu din  $X_n^0$  cu 0. Similar, considerăm mulțimea  $X_n^1$  formată din toate imaginile ce reprezintă o linie orizontală: există o linie (dintre cele  $n$ ) cu toți pixelii ce au valoarea = 1, restul pixelilor au valoarea = 0. Etichetăm fiecare exemplu din  $X_n^1$  cu 1. În cele ce urmează vrem să construim rețele care discriminează între cele două clase.

Pentru  $n = 2$ , avem exemplele din cele două clase reprezentate mai jos:



Clasa 0



Clasa 1

- (1 punct) scrieti codul Matlab care genereaza multimea de exemple  $X_2 = X_2^0 \cup X_2^1$  si etichetele  $t_2$  corespunzătoare.  $X_2$  ar trebui să fie o matrice  $4 \times 4$  (4 imagini, fiecare imagine are 4 pixeli).
- (1 punct) antrenati un perceptron de tip Rosenblatt pentru 1000 de epoci care să separe cele două clase.
- (1 punct) clasificați fiecare exemplu din  $X_2$  într-una din cele două clase. Este mulțimea  $X_2$  liniar separabilă? Justificați în comentarii.
- (2 puncte) antrenați o rețea cu mai mulți perceptroni pe stratul ascuns, 1 perceptron pe stratul de ieșire (încercați mai multe

arhitecturi). Simulați comportamentul rețelei pe  $X_2$  clasificând output-ul rețelei într-una din cele două clase în funcție de pragul 0.5. Există o rețea ce separă exemplele din mulțimea de antrenare?

Considerăm acum cazul general, în care imaginea binară are  $n^2$  pixeli.

- e. (1 punct) scrieți funcția **genereazaMultimeAntrenare(n)** care generează mulțimea de exemple  $X_n = X_n^0 \cup X_n^1$  și etichetele  $t_n$  corespunzătoare.  $X_n$  ar trebui să fie o matrice  $n^2 \times 2n$  ( $2n$  imagini, fiecare imagine are  $n^2$  pixeli).
- f. (1 punct) scrieți funcția **perturbaExemple(X,sigma)** care perturba fiecare exemplu (coloana) din  $X$  cu un zgomot repartizat normal de medie 0 și deviație standard sigma aplicat independent fiecărui pixel. Fiecare pixel afectat de zgomot ce primește o valoare  $< 0$  va fi setat la 0, fiecare pixel afectat de zgomot ce primește o valoare  $> 1$  va fi setat la 1.
- g. (2 puncte) considerați  $n = 25$ . Antrenați o rețea de tip *patternnet* care discriminează între cele două clase. Folosiți o mulțime de antrenare cu 500 de exemple obținută prin generarea unor exemple de antrenare perturbate din cele inițiale. Folosiți un zgomot cu  $\sigma = 0.1$ . Folosiți o tehnică pentru a evita fenomenul de overfitting. Care este rata de misclasare pe mulțimea de exemple de antrenare a exemplilor neperturbate?

***Se acordă 1 punct din oficiu. Timp de lucru: 75 de minute.***