

**Inteligență Artificială**  
**Lucrare de laborator – Varianta 1**  
**8 iunie 2024**

În această lucrare veți dezvolta și antrena modele pentru clasificare de semnale provenite de la senzorii de tip accelerometru a 4 telefoane, obținute în timpul utilizării telefoanelor.

În directorul **data**, veți găsi un subdirector numit **train** ce conține 1000 de semnale de antrenare, fiecare stocat într-un fișier **.txt** separat. Aceste fișiere conțin 3 coloane, anume câte o coloană pentru fiecare axă pentru care accelerometrul raportează valori ale forței G: x, y, z. Pe fiecare linie, se găsesc 3 valori reprezentând descompunerea forței G pe cele 3 axe la un anumit moment de timp. Numărul de linii din fiecare fișier este dat de numărul de valori raportate într-o perioadă de 1.5 secunde.

Subdirectorul **test**, conține 400 de exemple de testare reprezentate în același format (câte un exemplu de test în fiecare fișier).

Semnalele au o lungime diferită unul față de celălalt, deoarece frecvența de raportare nu este constantă. Pentru a le putea folosi în exercițiile de mai jos, este nevoie să normalizați lungimea acestora, fie prin eliminarea valorilor de la final din semnalele mai lungi, fie prin adăugarea de zero-uri la final pentru semnalele mai scurte.

Fișierul **train.txt** conține pe fiecare linie numele unui fișier ce conține un semnal de antrenare, urmat de eticheta (de la 0 la 3) semnalului respectiv. Cele două valori sunt separate prin virgulă. Prima linie conține denumirea în engleză a celor două coloane.

Fișierul **test.txt** conține numele fișierelor de test (câte unul pe rând), fără etichete.

***Rezolvați următoarele cerințe:***

**1. (2p)** Antrenați și testați o rețea neuronală feed-forward (cu maxim 3 straturi) folosind optimizatorul Adam. Pentru a obține punctajul acordat, trebuie să implementați corect modelul și să generați un fișier cu predicțiile pe datele de test (conform observațiilor de la final).

**1p** – acuratețe minimă pe datele de test = 80%

**2p** – acuratețe minimă pe datele de test = 85%

**2. (2.5p)** Implementați o funcție care să calculeze matricea de tranziție Markov pentru fiecare din cele 3 axe (x, y, z) ale unui semnal. Primul pas în calcularea acestor matrici este discretizarea semnalelor. În acest sens, trebuie să stabiliți o mulțime de k intervale de valori disjuncte și de lungime egală pentru fiecare dintre cele 3 axe (x, y, z), astfel încât valorile de pe fiecare axă să fie incluse într-unul din intervale. Apoi, puteți înlocui fiecare valoare din semnal cu indexul intervalului corespunzător. Pentru fiecare axă, trebuie să calculați o matrice  $A$  de dimensiune  $k \times k$ , unde componenta  $A(i, j)$  conține probabilitatea de a trece din intervalul  $i$  în intervalul  $j$ . Parcurgând un semnalul discretizat de pe o anumită axă de la prima la ultima valoare din semnal, la fiecare pas  $t$ , incrementați celula corespunzătoare din matricea  $A$ . Ulterior, pentru a transforma valorile din  $A$  în probabilități, puteți normaliza matricea  $A$  prin împărțirea elementelor de pe fiecare linie la suma elementelor de pe linia respectivă. În final, funcția implementată trebuie să

returneze un vector obținut prin liniarizarea și concatenarea celor 3 matrici corespunzătoare axelor x, y, z.

Aplicați funcția pe datele de antrenare și test utilizând  $k = 6$ .

**Exemplu:**

Calculul matricei de tranziție pentru o singură axa dintr-un semnal, folosind  $k = 4$ .

Semnalul original:

$$S = [5.2, 4.1, 3.1, 6.8, 6.8, 2.5, 5.2, 1.4, 4.2, 4.2]$$

Intervalele:

$$[1.4 \ 2.75), [2.75, 4.1) [4.1, 5.45), [5.45, 6.8]$$

Semnalul discretizat:

$$D = [2, 2, 1, 3, 3, 0, 2, 0, 2, 2]$$

Matricea nenormalizată:

$$A = \begin{pmatrix} 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 2 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

Matricea A normalizată:

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0.25 & 0.25 & 0.5 & 0 \\ 0.5 & 0 & 0 & 0.5 \end{pmatrix}$$

În final, liniarizăm matricea și obținem următorul vector de trăsături:

$$X = [0, 0, 1, 0, 0, 0, 0, 1, 0.25, 0.25, 0.5, 0, 0.5, 0, 0, 0.5]$$

**3. (2.5p)** Antrenați și testați metoda celor mai apropiați vecini folosind vectorii de la exercițiul anterior. Pentru calculul distanței dintre exemple folosiți distanța Manhattan între vectori.

**1p** – acuratețe minimă pe datele de test = 92%

**2p** – acuratețe minimă pe datele de test = 93%

**2.5p** – acuratețe minimă pe datele de test = 95%

**4. (2.5p)** Antrenați și testați un model SVM cu parametrul kernel setat la valoarea 'precomputed'. Folosiți funcția nucleu intersecție pentru a crea matricile kernel de antrenare și test pe baza vectorilor rezultați de la exercițiul 2.

**1p** – acuratețe minimă pe datele de test = 94%

**2p** – acuratețe minimă pe datele de test = 95%

**2.5p** – acuratețe minimă pe datele de test = 96%

**5. (1.5p)** Creați un raport al experimentelor însoțit de evaluarea pe un set de validare a diferite combinații de hiperparametri pentru modelele de la punctele 1, 3 și 4. Raportul poate conține tabele sau grafice.

### **1p - Oficiu**

#### ***Observații importante:***

După implementarea cerințelor de mai sus, trebuie să trimiteți într-un folder denumit {Nume}\_{Prenume}\_{Grupa}\_{Varianta}:

a) Cel mult 1 submitie pentru setul de testare cu metodele de la punctul 1; cel mult 3 submitii pentru setul de testare cu fiecare din metodele de la punctele 3 și 4. O submitie constă într-un fișier .txt denumit:

`{Nume}_{Prenume}_{Grupa}_subiect{i}_solutia_{j}.txt`

unde i este numărul subiectului (1, 3 sau 4) și j este numărul submitiei (1, 2 sau 3), în care se află un denumirele fișierelor corespunzătoare exemplelor de test și etichetele prezise pentru toate exemplele de test, în același format ca fișierul **train.txt**.

b) Codul aferent pentru antrenarea modelelor și obținerea soluțiilor trimise. Pentru fiecare submitie, codul trebuie organizat într-un singur fișier .py denumit:

`{Nume}_{Prenume}_{Grupa}_subiect{i}_solutia_{j}.py`

unde i este numărul subiectului (1, 2, 3 sau 4) și j este numărul submitiei (1, 2 sau 3).

c) Raportul de la punctul 5.

Exemplu:

Denumire director: Popa\_Marian\_231\_1

Prima submitie pentru subiectul 3: Popa\_Marian\_subiect3\_solutia1.txt

Codul care a generat submitia de mai sus: Popa\_Marian\_subiect3\_solutia1.py