Inteligență Artificială
Lucrare de laborator – Varianta 1
21 iunie 2025

În această lucrare veți dezvolta și antrena modele pentru clasificarea de documente text. Fiecărui document de antrenare îi corespunde o clasă.

În directorul curent, veți găsi datele de antrenare (train_sentences.txt), etichetele corespunzătoare (train_labels.npy), și datele de testare (test_sentences.txt).
În fișierul train_sentences.txt se găsește un tensor ce conține 1025 de exemple care reprezintă textele de antrenare. În fișierul test_sentences.txt se găsește un tensor care conține 776 de exemple de testare.

În fișierul words.txt se găsește o listă a celor mai importante 500 3-grams din setul de antrenare.
În fișierul mapping.txt sunt stocate pe linii diferite perechi de caracter și număr. Elementele fiecărei perechi sunt separate de virgulă.

Rezolvați următoarele cerințe:

1. **(3.5p)**. Antrenați un model Bayes Naiv pe datele disponibile pentru antrenare. Reprezentați datele folosind metoda „Bag of Words" aplicată la nivel de caracter. 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 = 50%
2p – acuratețe minimă pe datele de test = 60%
3p – acuratețe minimă pe datele de test = 65%
3.5p – acuratețe minimă pe datele de test = 70%

2. **(2.5p)**. Implementați operația de convoluție pe documentele de antrenare și de test folosind fiecare 3-gram din fișierul words.txt. Înaintea aplicării operației de convoluție este nevoie să transformați textele (atât 3-gramele cât şi documentele) în vectori numerici. Utilizați maparea din fișierul "mapping.txt" pentru a înlocui fiecare litera cu numărul corespunzător. Operația de convoluție între un document și un 3-gram constă în calculul produsului scalar pentru fiecare regiune (subvector) din documentul inițial, care are dimensiunile egale cu dimensiunile 3-gram-ului, anume 3. Pentru a obține un rezultat mai bun este nevoie să normalizați produsul scalar prin împărțirea la produsul normelor celor 2 vectori implicați. Rezultatele acestor calcule vor fi stocate într-un nou vector respectând ordinea regiunilor inițiale. În general, pentru un document cu L litere și un filtru (n-gramă) de n litere, se obține un vector rezultat de dimensiune $L - n + 1$. De exemplu, aplicând un filtru (n-gramă) de dimensiune $n = 3$ pe un document cu 5 caractere, rezultatul operației este un vector de dimensiune 3, așa cum este ilustrat în exemplul de mai jos.

Pentru fiecare vector rezultat în urma convoluției dintre un document și un 3-gram, returnați numărul de valori din vector care depășesc limita $t = 0.9$.

Creați o funcție care aplică mulțimea de 500 de 3-grams și returnează un vector cu 500 de componente având următorul conținut:

count_ f_1	count_ f_2	...	count_ f_{500}

unde:

- count_f_k este numărul de valori care depășesc limita $t = 0.9$ din vectorul rezultat în urma operației de convoluție cu al k-lea filtru, $k = \overline{1,500}$.

Exemplu convolutie:
Document: „ana are mere”
3-gram: „ana”
Pași:

1. Aplicarea mapării stocate în „mapping.txt”:
   „ana are mere” -> [37, 20, 37, 18, 37, 7, 24, 18, 30, 24, 7, 24]
   „ana” - > [37, 20, 37]
2. Aplicarea operației de convoluție:
   [37, 20, 37, 18, 37, 7, 24, 18, 30, 24, 7, 24] * [37, 20, 37] = [1.0 , 0.837, ...]

3. **(2.5p)** Implementați și antrenați metoda Kernel Ridge folosind vectorii de la exercițiul 2. Folosiți funcția kernel liniară în implementare.

1p – acuratețe minimă pe datele de test = 65%
2p – acuratețe minimă pe datele de test = 70%
2.5p – acuratețe minimă pe datele de test = 78%

4. **(2.5p)** Antrenați un model SVM cu parametrul kernel setat la valoarea ‘precomputed’. Aplicați funcția kernel Hellinger pentru a crea matricele kernel de antrenare și test. Funcția kernel va fi aplicată pe vectorii rezultați de la exercițiul 2.

1p – acuratețe minimă pe datele de test = 77%
2p – acuratețe minimă pe datele de test = 80%
2.5p – acuratețe minimă pe datele de test = 81%

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 submisie pentru setul de testare cu metodele de la punctul 1; cel mult 3 submisii pentru setul de testare cu fiecare din metodele de la punctele 3 și 4. O submisie constă într-un fișier .npy denumit:

{Nume}_{Prenume}_{Grupa}_subiect{i}_solutia_{j}.npy

unde i este numărul subiectului (1, 3 sau 4) și j este numărul submisiei (1, 2 sau 3), în care se află un tensor ce conține imaginile de ieșire pentru toate exemplele de test.

b) Codul aferent pentru antrenarea modelelor și obținerea soluțiilor trimise. Pentru fiecare submisie, 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 submisiei (1, 2 sau 3).

c) Raportul de la punctul 5.

Exemplu:
Denumire director: Popa_Marian_231_1
Prima submisie pentru subiectul 3: Popa_Marian_subiect3_solutia1.txt
Codul care a generat submisia de mai sus: Popa_Marian_subiect3_solutia1.py