

Inteligență Artificială

Lucrare de laborator – Varianta 3

27 aprilie 2024

În această lucrare veți dezvolta și antrena modele pentru clasificarea de imagini. Fiecărei imagini de antrenare îi corespunde o clasă.

În directorul **curent**, veți găsi datele de antrenare (trainImages.npy), clasele imaginilor de antrenare (trainLabels.npy), datele de testare (testImages.npy), o mulțime de filtre (filterBank.npy).

În fișierul trainImages.npy se găsește un tensor ce conține 1000 de imagini care reprezintă imaginile de intrare. În fișierul trainLabels.npy se găsește un tensor ce conține cele 1000 de clase de antrenare corespunzătoare exemplilor din trainImage.npy.

În fișierul testImages.npy se găsește un tensor care conține 500 de imagini de testare. În fișierul filterBank.npy se găsește un tensor care conține 40 de filtre de dimensiune 14×14 .

Rezolvați următoarele cerințe:

1. **(2p)**. Antrenați o rețea neuronală feed-forward (cu maxim 3 straturi) pe datele disponibile pentru antrenare. 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 = 90%

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

2. **(2.5p)**. Implementați operația de convoluție pe imaginile de antrenare și de test folosind fiecare filtru din fișierul filterBank.npy. Operația de convoluție între o imagine și un filtru constă în calculul produsului Hadamard (pe elemente) pentru fiecare regiune (matrice) din imaginea inițială, care are dimensiunile egale cu dimensiunile filtrului, anume 14×14 , și însumarea valorilor rezultate în urma calculului produsului Hadamard. Rezultatele acestor calcule vor fi stocate într-o nouă matrice respectând poziția spațială a regiunilor inițiale. În general, pentru o imagine de $N \times N$ pixeli și un filtru de $F \times F$, se obține o matrice rezultat de dimensiune $N-F+1$. De exemplu, aplicând un filtru de dimensiune 3×3 pe o imagine de dimensiune 5×5 , rezultatul operației este o matrice de dimensiune 3×3 , așa cum este ilustrat în exemplul de mai jos.

Pentru fiecare matrice rezultată în urma convoluției dintre o imagine și un filtru, returnați indecșii pentru linia și coloana corespunzătoare valorii maxime. În cazul în care sunt mai multe valori egale cu maximul, se poate returna oricare din pozițiile corespunzătoare.

Creați o funcție care aplică mulțimea de 40 de filtre și returnează un vector cu 80 de componente având următorul conținut:

idx_row_f ₁	idx_col_f ₁	idx_row_f ₂	idx_col_f ₂	...	idx_row_f ₄₀	idx_col_f ₄₀
------------------------	------------------------	------------------------	------------------------	-----	-------------------------	-------------------------

unde:

- idx_row_f_k este indexul liniei corespunzătoare valorii maxime din matricea rezultată în urma aplicării filtrului f_k
- idx_col_f_k este indexul coloanei corespunzătoare valorii maxime din matricea rezultată în urma aplicării filtrului f_k

2	4	9	1	4		1	2	3		51	66	
2	1	4	4	6		-4	7	4				
1	1	2	9	2	x	2	-5	1	=			
7	3	5	1	3								
2	3	4	8	5								
Imagine						Filtru				Rezultat		

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

1p – acuratețe minimă pe datele de test = 90%

2p – acuratețe minimă pe datele de test = 91%

2.5p – acuratețe minimă pe datele de test = 92%

4. **(2.5p)** Antrenați un model KRR cu parametrul kernel setat la valoarea 'precomputed'. Aplicați funcția RBF peste distanța Manhattan pentru a crea matricele kernel de antrenare și test. Distanțele vor fi calculate între vectorii rezultați de la exercițiul 2.

1p – acuratețe minimă pe datele de test = 91%

2p – acuratețe minimă pe datele de test = 92%

2.5p – acuratețe minimă pe datele de test = 94%

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}.npv

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