

Inteligență Artificială

Lucrare de laborator – Varianta 2

29 aprilie 2023

În această lucrare veți antrena modele pentru a prezice clasele din care fac parte o serie de hieroglife (reprezentate sub formă de imagini).

În directorul **curent**, veți găsi datele de antrenare (trainImages.npy), etichetele corespunzătoare (trainLabels.npy) și datele de testare (testImages.npy). În fișierul trainImages.npy se găsește un tensor ce conține 1200 de imagini care reprezintă imaginile de antrenare. În fișierul testImages.npy se găsește un tensor care conține 500 de imagini de testare.

Rezolvați următoarele cerințe:

1. **(2p)** Implementați o funcție care estimează similaritatea „Patch Frequency” între două imagini binarizate. Pentru binarizare, folosiți un prag egal cu T . Similaritatea „Patch Frequency” se bazează pe funcția nucleu intersecție între histograme ce conțin frecvențele de apariție ale patch-urilor (sub-imagini) de dimensiune $P \times P$. Pentru fiecare imagine, se va construi o histogramă care reține frecvența fiecărui patch de dimensiune $P \times P$. Un patch este identificat unic prin valorile pixelilor săi. Pentru eficiență, histogramele pot fi calculate înainte și folosite ca date de intrare.

2. **(1p)** Arătați (empiric / programatic) că pentru o imagine de 28×28 pixeli cu background negru (pixeli de valoare 0) și un pătrat alb de 8×8 pixeli în centrul imaginii (pixeli de valoare 1), similaritatea „Patch Frequency” cu $P=4$ este invariantă la translații pe verticală sau orizontală cu 2 pixeli.

3. **(2p)** Implementați metoda celor mai apropiați vecini folosind similaritatea „Patch Frequency” cu $T=40$. Pentru a obține punctajul acordat, trebuie să implementați corect modelul și să generați 3 submisii / fișiere cu predicțiile pe datele de test.

1p – acuratețe minimă pe datele de test = 72%

2p – acuratețe minimă pe datele de test = 75%

4. **(1p)** Implementați o metodă care să returneze matricea kernel dintre două mulțimi de exemple, folosind similaritatea „Patch Frequency”. Apelați metoda pentru a calcula matricile kernel pentru antrenare și testare. De exemplu, dacă mulțimea de antrenare are 1200 de exemple și mulțimea de testare are 500 de exemple, atunci matricea kernel pentru antrenare va avea 1200×1200 componente, iar matricea kernel pentru testare va avea 500×1200 componente.

5. **(2p)** Antrenați un model KRR pe mulțimea de antrenare, folosind matricea kernel precalculată de la punctul 3, specificând opțiunea kernel=“precomputed” (*). Pentru a obține punctajul maxim, trebuie să găsiți parametrii optimi pentru modelul dat și să

generați maxim 3 submisii / fișiere cu predicțiile pe datele de test. În lipsa etichetelor de test, puteți păstra o parte din mulțimea de antrenare pentru validare.

0.5p – acuratețe minimă pe datele de test = 80%

1p – acuratețe minimă pe datele de test = 84%

2p – acuratețe minimă pe datele de test = 90%

(*) în lipsa utilizării opțiunii `kernel="precomputed"`, punctajul maxim pentru punctul 4 este **1p**.

6. **(1p)** Creați un raport al experimentelor însoțit de evaluarea pe un set de validare a diferite combinații de hiperparametri, atât pentru modelul de la punctul 2 cât și pentru cel de la punctul 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 3 submisii pentru setul de testare cu metoda de la punctul 2 și cel mult 3 submisii pentru setul de testare cu metoda de la punctul 4. O submisie constă într-un fișier txt denumit:

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

unde *i* este numărul subiectului (2 sau 4) și *j* este numărul submisiei (1, 2 sau 3), în care pe fiecare linie se află predicția pentru câte un exemplu de test. Fișierul va avea în consecință 500 de linii cu câte un număr, aferente celor 500 de exemple de test.

Exemplu submisie:

Nume fișier: `Creanga_Ion_123_subiect_4_solutie_1.txt`

Conținut:

1
0
1
3
2
...

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 (2, 3, 4 sau 5) și *j* este numărul submisiei (1, 2 sau 3).

c) Raportul de la punctul 5.