

TASK_1

Această funcție, denumită task1, primește două argumente: o matrice photo reprezentând o imagine și un număr întreg k. Funcția are ca scop realizarea unei aproximări a matricei inițiale photo utilizând algoritmul SVD (Descompunerea Valorilor Singulare).

Astfel, rezultatul final al funcției task1 este matricea new_X, care reprezintă aproximarea matricii inițiale photo folosind primele k componente singulare din descompunerea SVD.

TASK_2

Acest algoritm implementează metoda SVD (Descompunerea Valorilor Singulare) pentru a calcula componentele principale ale unei matrice. Algoritmul este scris în limbajul Octave și are ca rezultat aproximarea matricei inițiale utilizând un număr specificat de componente principale.

Pentru a utiliza algoritmul, trebuie să apelați funcția `task2(photo, pcs)`, unde:

- `photo` reprezintă matricea imaginii de intrare.
- `pcs` reprezintă numărul de componente principale dorite.

Rezultatul returnat de funcție este matricea aproximată `new_X`, care reprezintă matricea A_k conform algoritmului.

TASK_3

Acest algoritm are ca scop calcularea componentelor principale ale unei matrici de imagine utilizând matricea de covarianță. Rezultatul final este o matrice care reprezintă o aproximare a matricii inițiale cu ajutorul componentelor principale.

- se inițializează matricea `new_X` ca o matrice nulă cu aceeași dimensiune ca și matricea de imagine `photo`
- matricea de imagine `photo` este convertită la tipul de date `double` pentru a permite calculele matematice necesare.
- se calculează media fiecărui rând al matricii de imagine `photo` utilizând funcția `mean` și se obține un vector de medii denumit `row_means`.
- matricea inițială `photo` este normalizată prin scăderea mediei fiecărui rând din matricea respectivă. Rezultatul este o nouă matrice denumită `norm_photo`.
- se calculează matricea de covarianță `cov_mat` utilizând formula matricii de covarianță, care implică înmulțirea matricii normalizate `norm_photo` cu transpusa sa `t_norm_photo`, împărțită la $n - 1$, unde n este numărul de coloane al matricii `photo`.
- se calculează vectorii și valorile proprii ale matricii de covarianță utilizând funcția `eig`. Vectorii proprii sunt stocați în matricea `V`, iar valorile proprii sunt stocate în matricea diagonală `D`.
- Valorile proprii obținute sunt sortate descrescător, iar indicii corespunzători sunt stocați în vectorul `indici`. Apoi, se reordonează coloanele matricii `V` în funcție de indicii sortați, astfel încât prima coloană să corespundă celui mai mare vector propriu, a doua coloană celui de-al doilea cel mai mare vector propriu și așa mai departe.
- Se selectează primele `pcs` coloane din matricea `V`, reprezentând componentele principale, unde `pcs` este numărul de componente principale specificat ca argument în funcția `task3`.

TASK_4. visualise_image

Această funcție primește două argumente: `train_mat`, care reprezintă matricea de antrenament, și `number`, care reprezintă numărul liniei din matricea de antrenament pe care dorim să o vizualizăm ca imagine.

1. Se inițializează o matrice goală `im` cu dimensiunea `28x28` pentru a reține imaginea finală.
2. Se citește linia din matricea de antrenament cu numărul specificat (`number`).
3. Linia citită este transformată într-o matrice `28x28` prin utilizarea funcției `reshape`. Această matrice este apoi transpusă pentru a avea orientarea corectă a imaginii.
4. Matricea rezultată este convertită în tipul de date `uint8` pentru a fi o imagine validă.
5. Imaginea rezultată este afișată utilizând funcția `imshow`.

TASK_4. `prepare_data`

Funcția `prepare_data` primește două argumente: `name`, care reprezintă numele unui fișier de date, și `no_train_images`, care specifică numărul de imagini de antrenament ce trebuie prelucrate.

Funcția `prepare_data` are rolul de a pregăti datele pentru antrenamentul unui model, extrăgând un set de imagini de antrenament și etichetele corespunzătoare dintr-un fișier specificat.

TASK_4. magic_with_pca

Această funcție, numită "magic_with_pca", implementează o tehnică numită Analiză Componentelor Principale (PCA) pentru a efectua o reducere a dimensionalității datelor și pentru a reconstrui o aproximare a matricei de antrenament inițiale.

Rezultatele funcției sunt:

- "train" - matricea reconstruită, care reprezintă o aproximare a matricei inițiale, obținută prin proiecția datelor pe componentele principale selectate.
- "miu" - vectorul medie al coloanelor matricei de antrenament inițiale.
- "Y" - matricea rezultată din proiecția datelor pe noile componente principale.
- "Vk" - matricea formată din vectorii proprii corespunzători componentelor principale selectate.

Această funcție poate fi utilă în procesarea datelor și analizei exploratorii pentru a reduce dimensionalitatea datelor și pentru a extrage caracteristici relevante.

TASK_4. prepare_photo

Această funcție primește o imagine ca argument și pregătește imaginea pentru a fi utilizată într-un algoritm sau într-o analiză ulterioară. Mai exact, funcția face următoarele acțiuni:

1. Inițializează un șir (vector) de dimensiune 1x784 cu valori de zero.
2. Inversează pixelii imaginii. Aceasta se realizează prin scăderea fiecărui pixel al imaginii de la valoarea maximă a unui pixel (255 în acest caz). Astfel, pixelii albi devin negri și invers.
3. Transpune imaginea. Aceasta înseamnă că coloanele și rândurile pixelilor sunt inversate.
4. Transformă imaginea într-un vector liniar (un șir) de dimensiune 1x784. Acest lucru se realizează prin "răsucirea" matricei imaginii într-un singur rând, astfel încât fiecare element al șirului reprezintă un pixel al imaginii.

În final, funcția returnează șirul rezultat, care reprezintă imaginea pregătită pentru utilizare ulterioară.

TASK_4. KNN

Acesta functie reprezintă o implementare a algoritmului K-Nearest Neighbors (KNN) pentru clasificare.

Algoritmul KNN primește un set de antrenare cu etichete, calculează distanțele euclidiene între vectorul de test și fiecare exemplu din setul de antrenare, sortează distanțele și extrage k etichete corespunzătoare celor mai mici distanțe. Apoi, se prezice clasa vectorului de test prin luarea celei mai frecvente etichete dintre vecinii cei mai apropiați.

TASK_4. `classifyImage`

Această funcție clasifică o imagine utilizând algoritmul KNN și analiza componentelor principale (PCA). Ea preprocesează setul de antrenament prin PCA, scade media fiecărei coloane și apoi utilizează algoritmul KNN pentru a prezice clasa imaginii de clasificat.