README -Tema 2 MN

Mihai Alexia-Andra, 315CC

TASK 1

Compresia imaginilor folosind descompunerea redusă a valorilor singulare (SVD)

Funcţia de la Task-ul 1 comprimă o imagine într-o formă aproximativă, reţinând doar cele mai semnificative valori singulare.

REZOLVARE:

- Am convertit matricea photo în tipul de date double, pentru a permite calculele precise cu valorile singulare.
- Am aplicat descompunerea SVD asupra matricei photo, utilizând funcţia din Octave svd. Matricele U, S şi V rezultate reprezintă factorizarea matricei photo.
- Am selectat doar primele k coloane din matricea U, primele k valori de pe diagonala matricei S şi primele k coloane din matricea V. Acestea reprezintă noile matrice reduse U, S şi V utilizate pentru aproximarea matricii iniţiale photo.
- Am calculat matricea new_X folosind matricele reduse U, S şi V prin înmulţirea lor şi transpunerea rezultatului.
- Am convertit new_X la tipul de date uint8, pentru a reprezenta o imagine validă.

TASK 2

Compresia imaginilor folosind analiza componentelor principale- metoda SVD

Funcţia de la Task-ul 2 realizează compresia unei imagini utilizând analiza principalelor componente (PC).

REZOLVARE:

- Am convertit matricea photo în tipul de date double, pentru a permite calculele precise cu valorile singulare.
- Am normalizat matricea inițială photo prin scăderea mediei fiecărui rând din acesta. Am stocat mediile calculate în vectorul q. Acest pas asigură centrarea datelor în jurul originii.
- Am construit matricea Z utilizând formula de normalizare a datelor în PCA, unde fiecare coloană a matricei Z reprezintă un vector de caracteristici normalizate.
- Am aplicat descompunerea SVD asupra matricei Z, utilizând funcția svd.
- Am construit matricea W din primele pcs coloane ale matricei V, rezultând subspațiul format din principalele componente.
- Am calculat matricea Y prin înmulțirea transpusei lui W cu matricea inițială. Aceasta reprezintă proiecția matricei inițiale în subspațiul definit de principalele componente.
- Am aproximat matricea inițială, utilizând produsul matricelor W și Y și adăugând mediile rândurilor, calculate anterior.
- Am convertit new_X la tipul de date uint8, pentru a reprezenta o imagine validă.

TASK 3

Compresia imaginilor folosind analiza componentelor principale- algoritm bazat pe matricea de covarianță

Funcția de la Task-ul 3 realizează compresia unei imagini utilizând analiza componentelor principale (PCA) și matricea de covarianță.

REZOLVARE:

- Am convertit matricea photo în tipul de date double, pentru a permite calculele precise cu valorile singulare.
- Am normalizat matricea inițială photo prin scăderea mediei fiecărui rând din acesta. Am stocat mediile calculate în vectorul q. Acest pas asigură centrarea datelor în jurul originii.
- Am calculat matricea de covarianță Z utilizând formula matricei de covarianță.

- Am calculat vectorii şi valorile proprii ale matricei de covarianţă Z utilizând funcţia eig, obţinând matricea diagonală S (valorile proprii) şi matricea V (vectorii proprii).
- Valorile proprii sunt ordonate descrescător, iar vectorii proprii sunt rearanjaţi în aceeaşi ordine. Astfel, matricea V este formată din vectorii proprii corespunzători valorilor proprii ordonate descrescător.
- Am păstrat doar primele pcs coloane ale matricei V, care reprezintă componentele principale. Acestea sunt cele mai importante componente care asigură o compresie bună a datelor.
- Am calculat matricea Y prin înmulţirea transpusei lui W cu matricea iniţială.
- Am aproximat matricea iniţială, utilizând produsul matricelor W şi Y şi adăugând mediile rândurilor, calculate anterior.
- Am convertit new_X la tipul de date uint8, pentru a reprezenta o imagine validă.

OBSERVAȚII:

Am aplicat descompunerea SVD pe o imagine şi am redus dimensiunea imaginii prin păstrarea a unui număr specific de componente principale. Adica, am selectat doar k coloane din U, k randuri din V şi k valori de pe diagonala lui S.

Cu cat k ales este mai mare, adică numărul de componente principale este mai mare, cu atât imaginea aproximată va fi mai asemănătoare cu cea originală. Calitatea generală a imaginii va fi mai înaltă.

Cu cât k este mai mic, aproximarea va fi mai slabă, imaginea mai neclară, mai estompată. Dar memoria necesară pentru a stoca datele va fi mai puţină şi timpul de procesare mai scurt.

Cu cât se creşte numărul de componente principale claritatea imaginii creşte, dar de la un anumit prag încolo diferenţa nu poate fi sesizată de ochiul uman aşa că pot fi eliminate.

TASK 4- Recunoașterea cifrelor scrise de mână

prepare_data

Funcţia prepare_data pregăteşte datele pentru antrenarea unui model utilizând un set de imagini şi etichete.

REZOLVARE:

- Am încărcat datele din tabelul specificat în argumentul funcţiei, utilizând funcţia load. Aceasta va încărca structura de date d care conţine câmpurile trainX şi trainY.
- Am copiat primele no_train_images linii din matricea trainX în matricea train_mat, selectând imaginile de antrenament.
- Am copiat primele no_train_images linii din vectorul trainY în vectorul train_val, selectând etichetele corespunzătoare imaginilor de antrenament.
- Am returnat matricele train_mat şi vectorul train_val ca rezultat.

visualise_image

Funcţia visualise_image vizualizează o imagine din matricea de antrenament.

REZOLVARE:

- Am citit linia cu numărul number din matricea de antrenament train_mat şi am stocat-o în variabila linie.
- Am utilizat funcţia reshape pentru a transforma linia citită într-o matrice 28x28, reconstruind imaginea originală.
- Am transpus matricea rezultată pentru a avea orientarea corectă a imaginii.
- Am convertit new_X la tipul de date uint8, pentru a reprezenta o imagine validă.

magic_with_pca

Funcţia magic_with_pca realizează prelucrări speciale folosind analiza componentelor principale asupra matricei de antrenament.

REZOLVARE:

- Am convertit matricea train_mat la tipul de date double.
- Am calculat media fiecărei coloane a matricei train_mat şi am stocat-o în vectorul miu. Apoi am scazut media din matricea iniţială, pentru a centra datele.
- Am calculat matricea de covarianţă cov_matrix.
- Am calculat vectorii şi valorile proprii ale matricei de covarianţă cov_matrix folosind funcţia eig.
- Am sortat valorile proprii în ordine descrescătoare şi am reordonat vectorii proprii corespunzător.
- Am păstrat primele pcs coloane din matricea V obţinută anterior, formând matricea de transformare Vk.
- Am calculat matricea Y prin înmulţirea matricei train_mat cu Vk, obţinând reprezentarea datelor în spaţiul componentelor principale.
- Am calculat matricea train care reprezintă aproximarea matricei de antrenament train_mat folosind matricea Y şi matricea de transformare Vk.

prepare_photo

Funcţia prepare_photo pregăteşte o imagine pentru utilizare într-un model de învăţare automată.

REZOLVARE:

- Am inversat pixelii imaginii im prin scăderea valorii fiecărui pixel din 255. Acest lucru poate fi util în anumite cazuri, cum ar fi prelucrarea imaginilor în care obiectele de interes sunt mai întunecate decât fundalul.
- Am transpus imaginea im, pentru a obţine imaginea în formatul corect.
- Am transformat imaginea transpusă într-un vector linie folosind funcţia reshape, astfel încât toţi pixelii să fie concatenaţi într-un singur

şir. Vectorul rezultat, sir, reprezintă imaginea pregătită şi este returnat ca rezultat.

KNN

Funcţia KNN realizează clasificarea unui vector de test utilizând algoritmul K-Nearest Neighbors (KNN).

REZOLVARE:

- Pentru fiecare vector de antrenament, am calculat distanţa euclidiană faţă de vectorul de test utilizând formula distanţei euclidiene.
- Am ordonat crescător distanţele calculate şi am reţinut primele `k` valori care reprezintă celor mai apropiaţi vecini.
- Am determinat etichetele corespunzătoare acestor vecini şi am calculat predicţia ca mediana acestor etichete.

classify_image

Funcţia classify_image implementează clasificarea unei imagini de test im utilizând setul de date de antrenament train_mat şi etichetele corespunzătoare train_val.

REZOLVARE:

- Clasificarea se realizează prin aplicarea funcţiei `magic_with_pca` asupra setului de date de antrenament pentru a obţine matricea `Y` care reprezintă vectorii de antrenament în noua bază determinată de analiza de componentă principală. Aceasta implică şi calculul mediei fiecărei coloane din imaginea de antrenament şi scăderea acesteia din vectorul imagine im.
- Ulterior, am schimbat baza imaginii de test prin înmulţirea acesteia cu matricea Vk obţinută în urma analizei de componentă principală a setului de date de antrenament.
- În final, am utilizat algoritmul KNN implementat anterior pentru a clasifica imaginea de test.

OBSERVAȚII:

Algoritmul prezentat pentru recunoașterea cifrelor scrise de mână se bazează pe PCA (Principal Component Analysis) și are o acurateţe de aproximativ 93.3%. Deşi această acurateţe poate fi mai mică decât cele obţinute cu metodele bazate pe reţele neuronale, acest algoritm are avantajul de a fi mai simplu de înţeles şi de aplicat.

Utilizarea algoritmului PCA pentru recunoașterea cifrelor scrise de mână permite o compresie semnificativă a datelor. În acest algoritm, imaginea de 28x28 pixeli este reprezentată printr-un vector de lungime 784, iar apoi se utilizează doar primele k componente principale pentru recunoaştere. Astfel, se obţine o reprezentare mai compactă a imaginii, ceea ce poate fi benefic în contextul unor seturi de date mari.

Cu cât alegem un număr mai mic de componente principale (k), cu atât vom avea o calitate mai scăzută a imaginii aproximate. Cu toate acestea, chiar şi cu un număr relativ mic de componente principale, algoritmul poate obţine o recunoaştere acceptabilă a cifrelor scrise de mână.