

README

Namoianu Petre - 315CC

Task 1:

Scopul taskului 1 este sa obtinem compresia unei imagini folosind descompunerea redusa a valorilor singulare. Pentru aceasta, intai dam cast matricei initiale la tipul "double", apoi ii aplicam funtia svd pentru a obtine matricele U, S, V. Pentru a calcula compresia luam din U doar primele k (parametru primit) coloane, din S primele k linii si k coloane, iar din V^t primele k linii. In final inmultim aceste 3 noi matrice pentru a obtine noua imagine.

Task 2:

Cu ajutorul taskului 2 calculam componentele principale cu metoda SVD. Mai intai aflam vectorul u, care cu ajutorul comenzii $\text{mean}(A, 2)$ calculeaza media fiecarui rand. Apoi se normalizeaza matricea, scazand din fiecare coloana a lui A vectorul u. Se construiesc matricea Z dupa formula data, asupra careia va fi aplicata functia svd, rezultand matricele U, S, V. Matricea W va fi formata prin extragerea primelor pcs coloane ale lui V. Cu ajutorul transpusei lui W vom calcula matricea $Y = W^t * A$. In final rezulta aproximatia matricei initiale, cu formula $A_k = W * Y + u$;

Task 3:

Primii pasi ai taskului 3 sunt la fel cu cei de la taskul precedent, calculandu-se vectorul u si apoi scazandu-l din fiecare coloana a lui A. Se calculeaza matricea Z de covarianta cu formula data, pe care aplicam functia eig pentru a afla vectorii si valorile proprii. Se ordoneaza, in continuare, descrescator valorile propria si vectorii propria corespunzatori valorilor. Cu ajutorul functiei sort extragem indicii valorilor ordonate descrescator si ne folosim de acestia pentru a reordona vectorii proprii in V si valorile in S. Se calculeaza din nou matricele Y si A_k , dupa care se mai aduna la A_k inca o data vectorul u si se returneaza matricea, cu aceeasi semnificatie ca cea de la taskul precedent.

Task 4:

Visualize_image:

Pentru prima functie de la acest task luam linia number din matricea de antrenament si o transformam intr-o matrice A de dimensiune [28 28], pe care o returnam, ea reprezentand o imagine de antrenament.

Prepare_data:

Functia da load tabelului folosit pentru citire si cu ajutorul functiei d.train salveaza in train_mat si train_val primele no_train_images linii din tabelul de imagini de antrenament, respectiv primele valori ale vectorului de etichete.

Magic_with_pca:

Pentru functia aceasta parcurgem initial aceeasi pasi de la taskul 3. Pornind de la matricea initiala(A), calculam vectorii si valorile proprii ale matricei de covarianta si le ordonam, pastrand doar primii pcs vectori(coloane) in matricea V_k . Apoi se mai creeaza si matricele $Y = A * V_k$ si aproximatia matricei initiale $\text{train} = Y * V_k^t$.

Prepare_photo:

Functia primeste o imagine, ii inverseaza pixelii, o transpune, si apoi o transforma intr-un vector linie.

KNN:

Functia primeste o imagine pe baza careia calculeaza distanta Euclidiană pentru fiecare rand dintre acesta si vectorul test cu ajutorul functiei norm. Apoi sortam crescator vectorul de distante si selectam in altul doar primele k elemente, urmand sa scoatem de aici predictia, ca fiind mediana celor k valori.

Classify_image:

Primul pas din functia de clasificare este reprezentat de apelarea functiei magic_with_pca pe matricea de antrenament si numarul pcs. Mai apoi, scadem din vectorul imagine miu, media fiecarei coloane din train_mat. Prin inmultirea cu V_k schimbam baza lui A pentru a o pregati de KNN. In final aplicam KNN cu argumentele train_mat, Y, test($A * V_k$) si 5, ce va returna un numar reprezentand predictia cifrei care se afla in imaginea initiala.

Observatii:

Calitatea imaginii este strans legata de numarul de componente principale alese, scazand odata cu un numar mai mic, si apropiindu-se de cea initiala pentru un numar cat mai mare.