Informatique 2ème année

L'objectif de ce TP est d'écrire un programme qui, en voyant une image d'un chiffre manuscrit comme : «  », sera capable de dire de quel chiffre il s'agit.

Nos images ont 8×8 pixels, chacune sera donc une liste de 64 nombres pris dans $[\![0, 16]\!]$.

Question 1 : Écrire une fonction `dist(a,b)` qui prend en argument deux listes de même longueur, et renvoie le carré de la distance euclidienne entre ces deux listes vues comme des éléments de $\mathbb{R}^n$ $\left(\text{rappel : ce résultat est donné par la formule } \sum_{i=0}^{n-1}(a_i - b_i)^2\right)$

Question 2 : Sans utiliser la fonction prédéfinie `min`, écrire une fonction `indice_min(L)` qui prend en argument une liste de nombres et renvoie l'indice de son plus petit élément.

Notre stratégie pour reconnaître un chiffre manuscrit inconnu sur une image `a` consiste à s'appuyer sur un jeu de données d'entraînement `D_E` contenant un grand nombre d'images pour lesquelles le chiffre est connu. On choisit l'image de `D_E` qui ressemble le plus à `a` et on parie que le chiffre écrit est le même.

Question 3 : Pour importer le jeu d'entraînement, importer le module `csv`, puis faire `D_E_str = list(csv.reader(open( chemin )))` , en remplaçant «*chemin*» par le chemin d'accès au fichier «donnees_entrainement.csv», qui se trouve dans le dossier «Documents en Consultation» de la classe.

Question 4 : La variable `D_E_str` ainsi créée est une liste de 3823 listes de 64 chaînes de caractères (chaque liste de taille 64 décrit les 64 pixels d'une des 3823 images). Créer une variable globale `D_E` contenant les mêmes informations, mais avec des données de type `int`.

Question 5 : Pour vérification : quel est le carré de la distance euclidienne entre les images d'indices 15 et 51 dans le jeu de données ? .....

Question 6 : Affichons une image du jeu d'entraînement, par exemple la numéro 2. Pour cela importer le module `matplotlib.pyplot` et utiliser la fonction `imshow` de ce module. On lui donnera deux arguments :

- les 64 pixels de l'image, sous la forme d'une liste de 8 listes de taille 8
- l'argument facultatif `cmap="gray_r"` , pour demander une image en niveau de gris, et inversée (16=noir, 0=blanc)

Question 7 : Lire le jeu de données de test (dans le fichier «donnees_test.csv») de la même manière que le jeu d'entraînement. On le désignera par la variable globale `D_T` .

Question 8 : Écrire une fonction `toutes_distances(a)` qui prend en argument une liste `a` et renvoie la liste des distances entre `a` et les images du jeu d'entraînement `D_E` .

Question 9 : Pour lire les chiffres des images du jeu d'entraînement, faire : `C_E_str=list(open(chemin))` , avec le chemin du fichier «chiffres_entrainement.txt». Comme précédemment, convertir cette liste de chaînes de caractères en une liste d'entiers. Désigner cette liste par la variable globale `C_E` .

Question 10 : En utilisant les fonctions du début, comparer la première image du jeu de test à chaque image du jeu d'entraînement, et trouver l'indice de celle qui lui ressemble le plus : ....
En utilisant `C_E` , déterminer quel chiffre est représenté sur cette image : .....

Question 11 : Écrire une fonction `predire_chiffre(a)` qui renvoie une prédiction du chiffre dans l'image `a` en utilisant cette stratégie.

Question 12 : Calculer la liste des prédictions faites par cet algorithme pour les 100 premières images du jeu de test. Donner les 10 dernières valeurs de cette liste : .....

À partir du temps de calcul que vous observez, estimer le temps de calcul nécessaire pour faire une prédiction pour tout le jeu de test. Expliquer le raisonnement : .....

.....

.....

Question 13 : Pendant que Python calcule la liste des prédictions pour l'ensemble du jeu de données, préparer un programme pour :

- lire le fichier «chiffres_test.txt» de la même manière que «chiffres_entrainement.txt»
- calculer la *Matrice de Confusion* : une matrice `M` de taille 10×10, telle que l'élément `M[i][j]` est le nombre de fois où l'algorithme a prédit le chiffre j alors que l'image représentait le chiffre i.