Reconnaissance de caractères

...

26 février 2019

Chapitre 1

Introduction

Le coeur du projet est d'écrire un code permettant de reconnaître automatiquement des caractères manuscrits, le but étant de se familiariser avec des méthodes usuelles de reconnaissance de forme en les implémentant pour un cas particulier et en en comprenant la théorie.

Les caractères à analyser seront ici des chiffres de 1 à 9 sous forme de vecteurs de taille $(28 \times 28, 1)$ à valeurs entières comprise entre 0 et 255 : chaque vecteur représente une image de taille 28×28 en nuance de gris ($0 =$ noir, blanc $= 255$). Nous disposons d'une base de données de 70000 chiffres manuscrits labelisés stockés sous le format décrit précédemment.

Nous avons choisi comme langage de programmation pour notre projet le Python.

Chapitre 2

Pré-traitement des données

La base de donnée étant un fichier matlab, nous avons dans un premier temps du la traduire dans un format utilisable sous python. Nous avons choisi comme format de stockage une liste de liste de vecteurs : le kième éléments de cette liste est une liste des vecteurs correspondants au chiffre k.

Les algorithmes étudiés relèvent du domaine de l'apprentissage automatique : après une phase d'apprentissage où l'on donne à l'algorithme des images et les chiffres respectifs comme références, on a une phase de traitement où l'algorithme doit reconnaître le chiffre à partir de l'image seule. Pour avoir des résultats pertinents, il faut donc distinguer clairement les données utilisées pour l'apprentissage et celles utilisées pour le traitement. (à compléter)

Nous avons ensuite divisé la base de donnée afin que chacun des membre du groupe puisse effectuer des tests indépendants.

Chapitre 3

Première méthode : calcul des écarts à la moyenne

3.1 Théorie

La première méthode étudiée est une approche assez intuitive du problème. Dans un premier temps, à partir du set d'entrainement, pour chaque chiffre on crée une image moyenne de toutes les image du chiffre : le barycentre (terme exacte ?)de tous les vecteurs de ce chiffre.
Une image inconnue est alors classée en cherchant l'image moyenne telle que l'écart de ressemblance entre l'image à tester et cette image soit minimum.

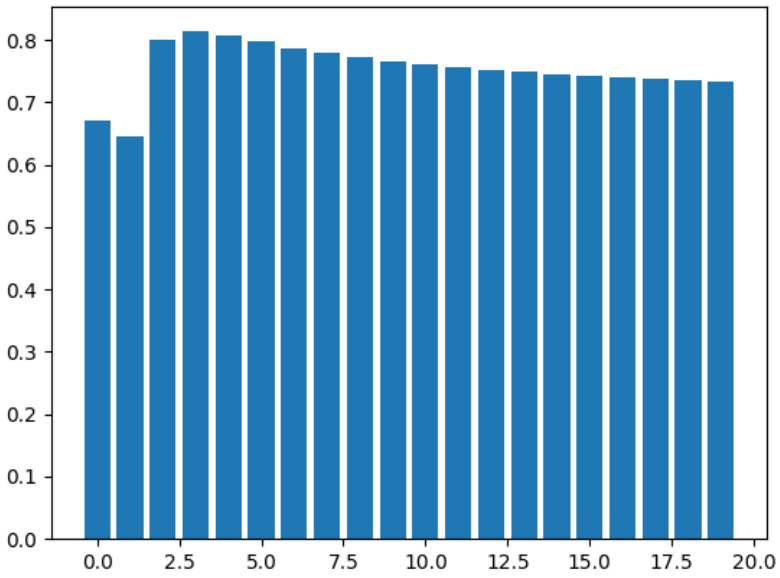
On définit l'écart de ressemblance entre deux images comme étant la distance dans une certaine norme choisie entre les deux vecteurs x et y de $\mathbf{R}^{784}$ correspondant aux images :

$$d(x, y) = \|x - y\|$$

Le choix de la norme est son influence sur les résultat est détaillé dans la partie pratique.

3.2 Pratique

Pour analyser l'efficacité de notre programme nous avons choisi d'observer deux types de données, le taux de réussite global de notre programme pour différentes normes, et le taux de réussite de notre programme pour chaque chiffre avec une norme fixe.
On a alors rapidement constaté grâce au graphe ci dessous, que les normes infini et 1 sont nettement moins efficaces que les autres, la norme la plus efficace semblant être la norme 3, puis on peut constater que l'efficacité jusqu'à atteindre asymptotiquement celle de la norme infinie.



Pour essayer de comprendre ce résultat il est intéressant de regarder plus en détails sur quels chiffres notre programme échoue pour les normes 1,2 et infini.

Chapitre 4

Deuxième méthode : utilisation de la décomposition en valeurs singulière (SVD)

4.0.1 Théorie

4.0.2 Pratique

Chapitre 5

Conclusion

Bibliographie

Annexe A

Annexe 1

...