

Reconnaissance de caractères par un perceptron

Sujet proposé par Jean-Christophe Filliâtre

`mailto:Jean-Christophe.Filliatre@lri.fr`

Difficulté : dans la moyenne (★★)

URL de suivi : <http://www.enseignement.polytechnique.fr/profs/informatique/Jean-Christophe.Filliatre/09-10/INF431/perceptron/>

Le but de ce projet est de réaliser un programme reconnaissant des caractères manuscrits à l'aide d'un *perceptron*. On se limitera ici à la reconnaissance des dix chiffres 0 à 9, dont une application possible est la lecture automatique des codes postaux pour le tri du courrier.

1 Perceptron

Étant donné un vecteur w de \mathbb{R}^n , on définit la fonction f_w de \mathbb{R}^n dans l'ensemble $\{0, 1\}$ des booléens de la manière suivante :

$$f_w(x) = \begin{cases} 1 & \text{si } \sum_i w_i x_i > 0 \\ 0 & \text{sinon} \end{cases}$$

Autrement dit, la fonction f_w sépare les éléments de \mathbb{R}^n situés des deux côtés de l'hyperplan défini par w . Dans le domaine des réseaux de neurones, on appelle une telle fonction un *perceptron*. Dans toute la suite, on considère que la n -ième composante des vecteurs d'entrée est toujours égale à 1, ce qui n'est pas une restriction en pratique (il suffit d'ajouter une dimension).

Étant donné un sous-ensemble fini D de $\mathbb{R}^n \times \{0, 1\}$, appelé *échantillonnage*, on cherche à déterminer s'il existe un perceptron en accord avec cet échantillon, et à le construire effectivement le cas échéant. Autrement dit, on cherche un vecteur w tel que pour tout élément (x, y) de D , on a $f_w(x) = y$. Pour cela on utilise l'algorithme suivant :

1. $w \leftarrow (0, \dots, 0)$
2. tant qu'il existe un élément (x, y) de D tel que $f_w(x) \neq y$,
 - (a) faire $w_i \leftarrow w_i + (y - f_w(x))x_i$ pour tout i

S'il existe une solution w , alors l'algorithme ci-dessus termine, sur une telle solution.

2 Application à la reconnaissance de caractères

Soit un caractère manuscrit donné sous la forme d'une image en noir et blanc. Voici un exemple de telle image, de 50 pixels de large et de 70 pixels de haut :



On commence par déterminer la plus petite zone rectangulaire contenant tous les pixels noirs (*bounding box*).



Puis on la subdivise en w colonnes et h lignes, pour deux paramètres w et h fixés. Ainsi, pour $w = 3$ et $h = 5$, on obtient le découpage suivant :



Enfin, on calcule pour chaque case de la grille la proportion de pixels noirs. On obtient $w \times h$ réels compris entre 0 et 1. En ajoutant la valeur constante 1, on obtient un vecteur de \mathbb{R}^n avec $n = 1 + w \times h$. Dans l'exemple ci-dessus, on a $n = 13$ et le vecteur suivant

$$X = (0.833, 0.486, 0.417, 0.156, 0.573, 0.069, 0.156, \\ 0.069, 0.486, 0.521, 0.000, 0.547, 0.347, 0.564, 0.260, 1.000)$$

si les cases sont énumérées de gauche à droite, puis de bas en haut.

À partir d'une collection d'images représentant plusieurs variantes de chacun des dix caractères 0 à 9, on construit dix perceptrons p_0, \dots, p_9 reconnaissant chacun un caractère. Pour chaque caractère i , le perceptron p_i est construit en considérant toutes les images représentant i comme devant être acceptées et toutes les autres images que devant être refusées.

Ensuite, pour reconnaître un caractère, on le présente indépendamment aux dix perceptrons. Si un seul l'accepte, on considère que l'on a identifié correctement le caractère (mais le résultat peut néanmoins être incorrect). Sinon, on échoue, soit parce qu'aucun perceptron n'accepte le caractère, soit parce que plusieurs perceptrons l'acceptent (ambiguïté).

3 Travail demandé

Le travail demandé se compose de plusieurs tâches :

1. proposer un ou plusieurs outils pour élaborer facilement une base d'apprentissage, c'est-à-dire un ensemble d'images correspondant à des caractères manuscrits déjà identifiés. Exemples : interface graphique utilisant la souris comme crayon (même s'il est relativement difficile d'écrire avec une souris), récupération des images à partir d'un document scanné, etc.
2. construire une telle base d'apprentissage, si possible à partir de l'écriture manuscrite de plusieurs personnes, et les dix perceptrons correspondant.
3. tester la reconnaissance sur de nouveaux caractères manuscrits, en réutilisant autant que possible les outils développés pour la tâche 1.

L'objectif est d'analyser l'efficacité de cette méthode et l'influence des différents paramètres, en particulier les entiers w et h et la taille de la base d'apprentissage. On répétera donc les tâches 2 et 3 pour différentes valeurs de ces paramètres. On pourra également s'intéresser à d'autres paramètres, comme par exemple l'influence de la largeur du trait, ou encore proposer une définition différente du vecteur d'entrée du perceptron à partir de l'image.

Références

- [1] Page de suivi du projet. <http://www.enseignement.polytechnique.fr/profs/informatique/Jean-Christophe.Filliatre/09-10/INF431/perceptron/>