Réseaux de neurones : TD n°2

La plupart des assistants personnels et des téléphones portables intègre un logiciel permettant de reconnaître l'écriture manuscrite. Dans cette séance de TD, on vous propose de travailler sur cette problématique. L'objectif est de créer un réseau de neurones capable de reconnaître des chiffres manuscrits.

FIGURE 1 – Application Palm pour la reconnaissance de l'écriture manuscrite.

Partie A
Mise en route

Avant de pouvoir utiliser NETLAB, il est nécessaire de configurer MATLAB :

1. recopiez les fichiers du TD dans un répertoire accessible de l'ordinateur,
2. ouvrez MATLAB,
3. ajoutez le chemin du répertoire de NETLAB dans le path de MATLAB à l'aide de la commande `File->SetPath->AddFolder`,
4. ouvrez le fichier `visualisation.m`,
5. exécutez ce script avec la commande `Debug->Run` (ou `F5`) et acceptez le changement du répertoire courant proposé par MATLAB,
6. vous devez voir s'afficher une figure.

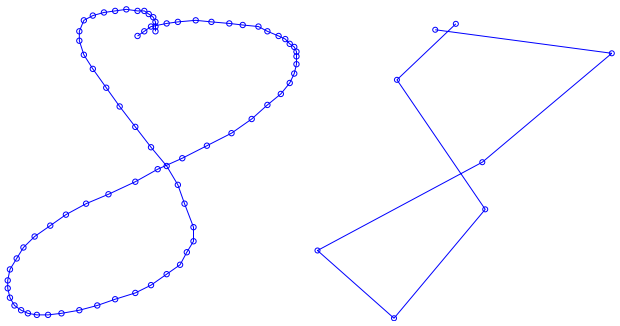
Partie B
Description des données

Les données utilisées pour cette séance de TD ont été proposées en 1998 par Alpaydin et Alimoglu de l'université d'Istanbul (voir le fichier `pendigits_names.txt` pour plus de détails).

Cette base de donnée recueille 250 échantillons de chiffres manuscrits de 44 personnes différentes. Les échantillons de 30 personnes sont utilisés pour l'apprentissage, et les chiffres écrits par les 14 autres sont utilisés pour la validation.

Les échantillons ont été recueillis à l'aide d'une tablette graphique sur laquelle on écrit avec un stylet. Chaque personne est invité à écrire 250 chiffres dans un ordre aléatoire à l'intérieur d'un cadre. Si la personne fait une erreur, sort du cadre ou n'est pas satisfaite de son geste, elle peut pour effacer le contenu et recommencer. Les dix premiers chiffres enregistrés sont systématiquement retirés de la base car la plupart des personnes n'est pas familière avec ce type de périphérique d'entrée.

Les données brutes enregistrées se composent des coordonnées des points du tracé de chaque chiffre. Avant d'être utilisées pour l'apprentissage, ces données doivent être traitées. Une technique couramment utilisée et conduisant à de bons résultats est de ré-échantillonner les points, soit temporellement (points régulièrement espacés dans le temps), soit spatialement (points régulièrement espacés sur le tracé).



(a) Tracé brut (b) Tracé ré-échantillonné

FIGURE 2 – Traitement des données.

La base de donnée étudiée utilise un ré-échantillonnage spatiale. Chaque chiffre est représenté par les coordonnées de 8 points régulièrement espacés sur le tracé brut (soit 16 valeurs). Les tracés sont également mis à l'échelle de manière être inscrits dans un carré. Ainsi, le fichier `pendigits_data.txt` contient sur chaque ligne, 16 valeurs comprises entre 0 et 100 représentant le tracé, plus une valeur entre 0 et 9 indiquant le chiffre correspondant.

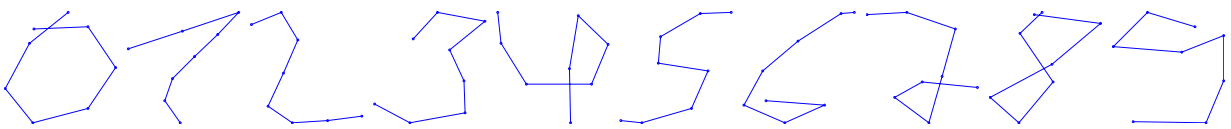


FIGURE 3 – Exemples de tracés après traitement.

B.1. Ouvrez le script visualisation.m *et utilisez-le pour visualiser quelques tracés*

Partie C
Reconnaissance d'un chiffre manuscrit

Nous allons focaliser notre étude sur la reconnaissance d'un seul chiffre. Par exemple, on souhaite savoir si un tracé correspond à un 3 ou non. Par ailleurs, pour permettre un apprentissage du réseau de neurones, les données doivent être normées et centrées.

C.1. Lisez attentivement les premières lignes du script apprentissage.m *pour y localiser ces modifications.*

C.2. Choisissez le chiffre qui vous plaira et modifier le script apprentissage.m *pour que la sortie Y soit égale à 1 pour ce chiffre et 0 pour les autres chiffres.*

C.3. Remplacer les points d'interrogation dans le script apprentissage.m *pour qu'il s'exécute convenablement avec 1 neurones cachés.*

On veut maintenant déterminer le nombre de neurones cachés à l'aide d'une validation simple.

C.4. A l'aide du script validation.m *calculez le critère sur l'ensemble de validation après apprentissage sur l'ensemble d'apprentissage.*

C.5. Répétez l'opération pour différentes valeurs du nombre de neurones cachés.

Pour chaque nombre de neurones cachés, on notera dans un tableau :
- le nombre d'itérations effectué,
- la valeur du critère pour les données d'apprentissage,
- le nombre de détections correctes pour les données d'apprentissage,
- le nombre de non détections pour les données d'apprentissage,
- le nombre de fausses détections pour les données d'apprentissage,
- la valeur du critère pour les données de validation,
- le nombre de détections correctes pour les données de validation,
- le nombre de non détections pour les données de validation,
- le nombre de fausses détections pour les données de validation.

C.6. Tracez différentes courbes de comparaison en fonction du nombre de neurones cachés. Proposez enfin un nombre de neurones cachés suffisant et évitant le sur-apprentissage.

Le compte-rendu (pdf) et les scripts (m) sont à rendre au plus tard le mercredi 16 février 12h00 par mail : `guillaume.laurent@ens2m.fr`
