
Réseaux de neurones : TD n°1

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 `approximation.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 trois figures.

Partie B

Approximation de fonction

L'objectif est d'approximer une fonction réelle bruitée à l'aide d'un réseau de neurones. Les observations (mesures) sont sauvegardées dans le fichier `A_data.txt` (consultez le fichier `A_names.txt` pour connaître sa structure).

B.1. Lisez attentivement le script `approximation.m` pour en comprendre le déroulement.

Remarques :

- un réseau de neurones est défini à l'aide de la commande `mlp`.
- l'optimisation des paramètres du réseau (apprentissage) est réalisé par la commande `netopt`.
- pour obtenir de l'aide sur une commande, on peut taper par exemple `>>help plot` dans la fenêtre de commande,
- toutes les commandes peuvent être testées dans la fenêtre de commande avant d'être ajoutées dans un script.

B.2. A l'aide de la courbe d'évolution du critère des moindres carrés, déterminez un nombre d'itération suffisant pour qu'un minimum soit bien atteint (ce réglage devra être effectué à chaque fois que vous changez le nombre de neurones cachés).

On s'intéresse maintenant à la comparaison entre sortie du réseau et les données d'apprentissage. La commande `y=mlpfd(net,x)` permet de calculer la sortie du réseau pour toutes les observations d'un vecteur `x`.

B.3. Déterminer « à l'oeil » le nombre idéal de neurones cachés.

Partie C

Sélection de modèle

On veut maintenant déterminer le nombre de neurones cachés à l'aide d'une validation simple. Cette méthode nécessite deux jeux d'observations : un ensemble d'apprentissage et un ensemble de validation.

Les données d'apprentissage sont fournies dans le fichier `A_data.txt`. Les données de validation sont fournies dans le fichier `A_valid.txt`.

La commande `mlperr(net,x,y)` permet de calculer le critère total pour le vecteur d'observation d'entrée `x` et le vecteur d'observation de sortie `y`.

C.1. A l'aide du script `validation.m` calculez le critère sur l'ensemble de validation après apprentissage sur l'ensemble d'apprentissage.

C.2. Répétez l'opération pour différentes valeurs du nombre de neurones cachés, puis tracez les courbes des deux critères en fonction du nombre de neurones cachés. Proposez enfin un nombre de neurones cachés suffisant et évitant le sur-apprentissage.

Remarque : pour tracer une courbe, il suffit d'entrer la commande `plot([x1 x2 ...],[y1 y2 ...]);` dans la fenêtre de commande.

Partie D

Approximation de fonction II

On souhaite maintenant approximer une deuxième fonction. Les données d'apprentissage sont fournies dans le fichier `B_data.txt`. Les données de validation sont fournies dans le fichier `B_valid.txt`.

D.1. Adaptez les scripts `approximation.m` et `validation.m` pour qu'ils fonctionnent avec ce nouveau jeu de données.

D.2. Proposez un nombre de neurones cachés suffisant et évitant le sur-apprentissage à l'aide d'une validation simple.
