

**Mini projet à rendre au plus tard pour le 18 septembre 2016**

- Dans ce mini projet, on vous demande de comparer les résultats obtenus sur le problème "sunspot" (voir les fiches de TP : **Initiation à Matlab** (Exercice 2) et **Régression par perceptron multicouches** (page 8)) pour les cas où l'on supprime 6 variables

1) avant l'apprentissage

2) après l'apprentissage

en utilisant un réseau avec une couche cachée.

- Dans toutes les expériences qui suivent :

↪ utiliser un réseau avec une couche cachée

↪ utiliser **3** neurones cachés

↪ utiliser une fonction d'activation linéaire ('linear') à la couche de sortie

↪ utiliser l'algorithme d'apprentissage 'scg'

- Après avoir constitué les ensembles  $D^{App}$ ,  $D^{Val}$  et  $D^{Test}$ , faire les expériences suivantes :

↪ voir les fiches de TP : **Initiation à Matlab** (Exercice 2) et

**Régression par perceptron multicouches** (page 8)

**Expérience 1**

Pour prédire une valeur  $x(t)$ , utiliser les 12 valeurs précédentes (comme en TP):

$x(t-12), x(t-11), x(t-10), x(t-9), x(t-8), x(t-7), x(t-6), x(t-5), x(t-4), x(t-3), x(t-2), x(t-1)$

↪ soit **Net1** le réseau obtenu à la fin de l'apprentissage.

**Expérience 2**

Pour prédire une valeur  $x(t)$ , utiliser les 6 valeurs suivantes :

$x(t-12), x(t-11), x(t-8), x(t-3), x(t-2), x(t-1)$

↪ soit **Net2** le réseau obtenu à la fin de l'apprentissage.

**Expérience 3**

Reprendre le réseau **Net1** obtenu à la fin de l'apprentissage dans **Expérience 1**

1. Supprimer les 6 variables suivantes **en une seule fois**

$x(t-10), x(t-9), x(t-7), x(t-6), x(t-5), x(t-4)$

**Remarque** : ce qui revient à ne garder que les variables utilisées dans **Expérience 2**

2. Continuer l'apprentissage avec les variables restantes (réentraîner le réseau) pour compenser l'effet de la suppression

Soit **Net3** le réseau obtenu

**Expérience 4**

- Reprendre le réseau **Net1** obtenu à la fin de l'apprentissage dans **Expérience 1**
- Refaire l'expérience 3 mais en supprimant les 6 variables **une seule à la fois**. A chaque suppression d'une variable, il faut réentraîner le réseau (faire quelques itérations d'apprentissage) pour compenser l'effet de la suppression.

↪ Il faut tester les 2 ordres de suppression suivants :

**Ordre 1**  $x(t-10), x(t-9), x(t-7), x(t-6), x(t-5), x(t-4)$

↪ C'est à dire :

utiliser le réseau **Net1**

supprimer  **$x(t-10)$**  et réentraîner le réseau (pour compenser l'effet de la suppression)

supprimer une deuxième variable  **$x(t-9)$**  et réentraîner le réseau

supprimer une troisième variable  **$x(t-7)$**  et réentraîner le réseau

supprimer une quatrième variable  **$x(t-6)$**  et réentraîner le réseau

supprimer une cinquième variable  **$x(t-5)$**  et réentraîner le réseau

supprimer une sixième variable  **$x(t-4)$**  et réentraîner le réseau

Soit **Net4** le réseau obtenu

**Ordre 2** ordre inverse de ordre 1 :  $x(t-4), x(t-5), x(t-6), x(t-7), x(t-9), x(t-10)$

Soit **Net5** le réseau obtenu

### Expérience 5

Refaire les expériences 1 à 4 en utilisant 20 neurones cachés.

**Travail à rendre au plus tard pour le 18 septembre 2016**

Envoyer par email à [meziane.yacoub@cnam.fr](mailto:meziane.yacoub@cnam.fr)

↪ un petit rapport (**format pdf**) contenant les informations suivantes :

1. Les différents résultats obtenus sous forme d'un tableau.

		ARV sur $D^{App}$	ARV sur $D^{Val}$	ARV sur $D^{Test}$
3 neurones cachés	<b>Net1</b>			
	<b>Net2</b>			
	<b>Net3</b>			
	<b>Net4</b>			
	<b>Net5</b>			
20 neurones cachés	<b>Net1</b>			
	<b>Net2</b>			
	<b>Net3</b>			
	<b>Net4</b>			
	<b>Net5</b>			

2. Vos commentaires sur ces résultats.
3. Un dossier compressé contenant vos programmes

↪ **Je ne regarde les scripts matlab (fichiers .m) qu'après avoir lu votre rapport (.pdf) contenant les résultats de vos expériences et vos commentaires.**

Bon courage.