

TRAVAUX PRATIQUES
RESEAX DE NEURONES TYPE « Feedforward »
Simulation en utilisant Scilab-ANN

II.5. TESTER LE RESEAU

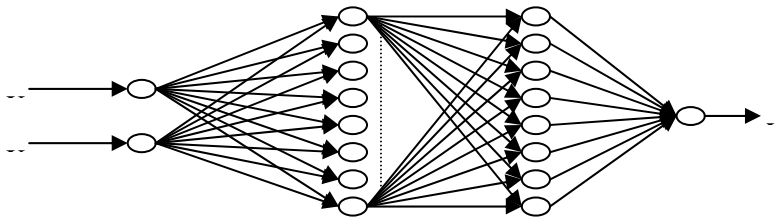
Le test du réseau consiste simplement en exécuter le réseau avec des vecteurs entrée-cibles créés et sauvegardés dans `x_t.dat` et `y_t.dat`. Pour cela, utiliser un pas d'échantillonnage différent de celui utilisé dans `tp1` pour générer les données d'entrée `x`. Le résultat peut ensuite être testé en comparant `y` (sauvegardé par `tp1`) à `y_t`. Les résultats peut être testé en comparant les sorties du réseau aux valeurs des sorties cibles (données par `y_t.dat`). Deux fichiers séparés peuvent accomplir cette tâche « `test_compare.sce` » exécute à la fois le réseau courant et le dernier réseau sauvegardé en utilisant les vecteurs de test et les sauvegarder dans « `result2.dat` ». « `plot_change.sci` » saisira alors « `results2.dat` » et tracera ensuite les sorties courantes, précédentes, cibles et les erreurs $e = y - c$.

SIMULATION DU PROBLEME

Effectuer plusieurs simulations en modifiant les paramètres d'apprentissage (un paramètre à la fois) puis en modifiant la topologie du réseau (nombre de couches cachées, nombre de neurones dans ces couches). Évidemment un réseau avec deux entrées et une sortie simple a été exigé.

Les meilleurs résultats peuvent être obtenus avec plusieurs couches cachées. (Combien de couches sont optimale ?) Chacune de ces couches un nombre de neurones N_h , faisant le vecteur de réseau :

$$N=[2,N_h,\dots,N_h,1];$$



L'ensemble de jeu d'apprentissage selon l'exemple donné dans (`tp1`) est un total de 200 exemples distribués à la gamme $x_1=0$ à 1, $x_2=0$ à 1. Dans chaque exécution du fichier d'apprentissage (algorithme de retro-propagation), la boucle d'apprentissage peut être exécutée avec un nombre d'epoches (e.g. `epoches=100`).

Le taux d'apprentissage (α dans le cours) peut être choisi initialement à une petite valeur aléatoire ($0 < \alpha \leq 1$), (e.g. [1, 0.05, 0.5, 0.1]), mais le premier paramètre de taux d'apprentissage peut être divisé par un facteur de 1.1 à chaque itération pour trouver `W` en évitant trop d'oscillations.

Le graphe suivant montre un exemple de comparaison entre les valeurs cibles et les valeurs de la sortie courante.

