

Université de ROUEN - STIM M2

TP Réseau de neurones

Compte rendu

HAO Chang
08/03/2016

Introduction

Ce TP a pour but de vous faire développer un réseau de neurones à une couche cachée et d'apprendre à contrôler son apprentissage.

Les données traitées ici sont le fichier IRIS de 150 éléments avec X de 4 attributs et une sortie Y de 3 classe, classée de 0 à 2.

En utilisant la fonction ***splitiris*** qu'on a créé, on sépare les données en trois parties 1/2 pour la base d'apprentissage (75 éléments), 1/4 pour la base de validation (36 éléments) et 1/4 pour la base de test (39 éléments).

Dans le cadre de ce TP, on se limitera à un réseau de type MLP, avec une couche d'entrée (les données), une couche cachée, et une couche de sortie de type ***softmax***.

Choix des paramètres

Le résultat et la performance d'un réseau neurone dépend de plusieurs paramètres. Dans ce projet, nous allons comparer les résultats pour différents paramètres suivants :

- Les data sont centré réduit ou non
- Fonction activation : sigmoïde, tanh et softmax pour la couche de sortie
- Nombre de neurones J en couche cachée
- Pas d'apprentissage

La complexité est que les paramètres sont liés entre eux, quand on change le nombre de neurones en couche cachée, on doit aussi modifier le pas d'apprentissage, et le même pour la fonction d'activation.

Difficultés

1. Le temps d'apprentissage est long. Du ce fait, on a utilisé les données IRIS parce qu'il est plus petit et moins de l'attribut et moins de classe. En plus quand les sorties du neurone est proche de 0 ou 1, les poids ne changent plus et l'algorithme peut s'arrêter. J'ai perdu beaucoup de temps pour résoudre ce problème mais le résultat n'est pas idéal.
2. Le résultat n'est pas toujours acceptable.
3. Il est difficile de définir la structure du réseau de neurone. Le choix des paramètres sont compliqués.
4. La base d'apprentissage est important. On ne sait pas si les données sont des exemples typiques.

De plus, la fonction du coût dépend de la couche de sortie. Les erreurs entre la sortie prédit et le résultat désiré sont calculés de manière différente pour une sortie de type softmax.

Résultats obtenus

En fait, le réseau neurone est instable, des fois nous pouvons obtenir un bon résultat, des fois il ne marche pas. L'initialisation des vecteurs W_{je} et W_{sj} peut influencer le résultat. Nous avons exécuté plusieurs fois pour les mêmes paramètres pour obtenir un résultat acceptable. Les résultats sont montrés ci-dessous.

➤ $f_1 = f_2 = \text{sigmoïde}$

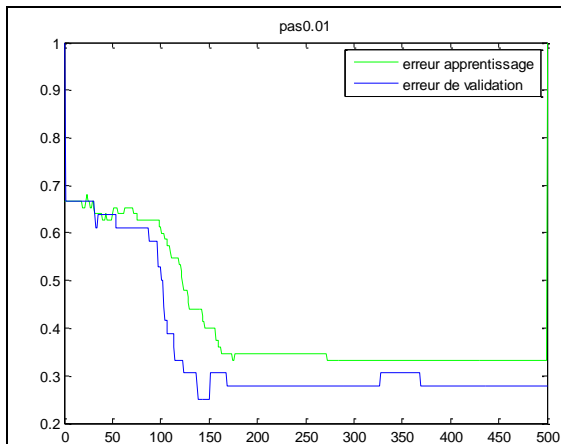


Figure 1: data centré-réduit $J = 10$, $\text{pas} = 0.01$, erreur de test : 0.37

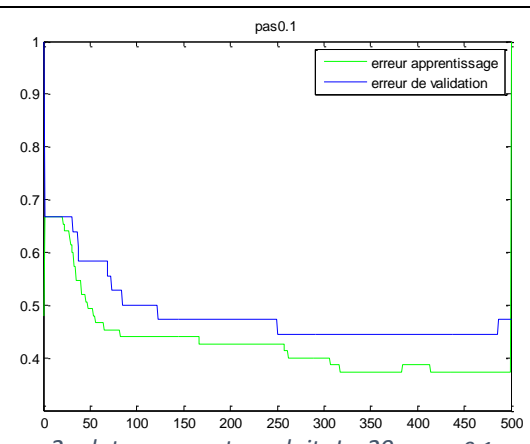


Figure 2 : data non centré-réduit, $J = 20$, $\text{pas} = 0.1$, erreur de test : 0.384615

➤ $f_1 = \tanh$, $f_2 = \text{sigmoïde}$

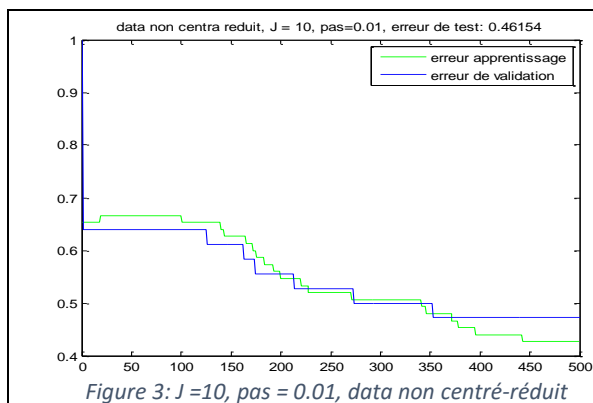


Figure 3: $J = 10$, $\text{pas} = 0.01$, data non centré-réduit

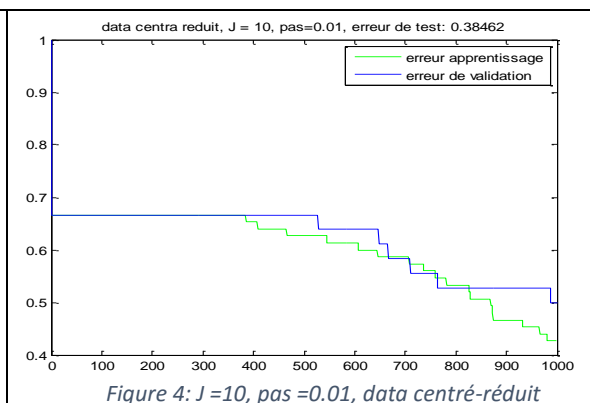


Figure 4: $J = 10$, $\text{pas} = 0.01$, data centré-réduit

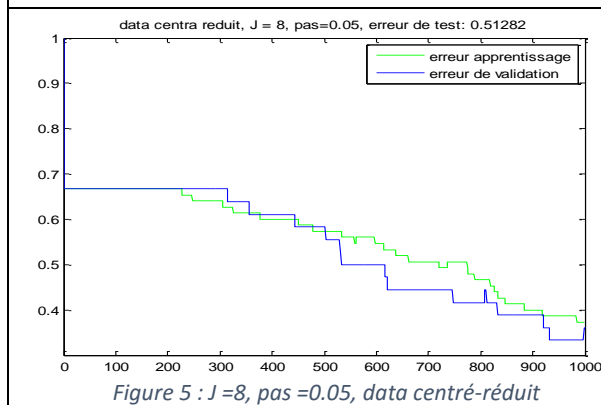


Figure 5 : $J = 8$, $\text{pas} = 0.05$, data centré-réduit

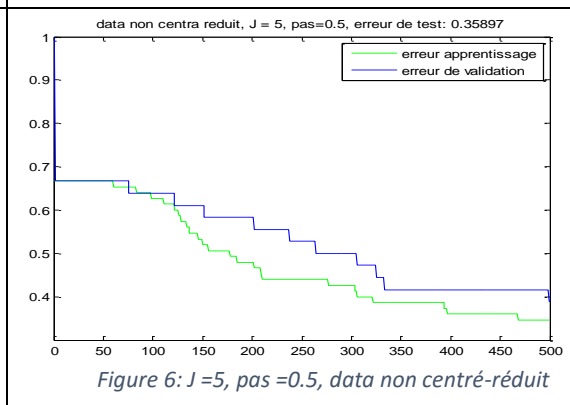
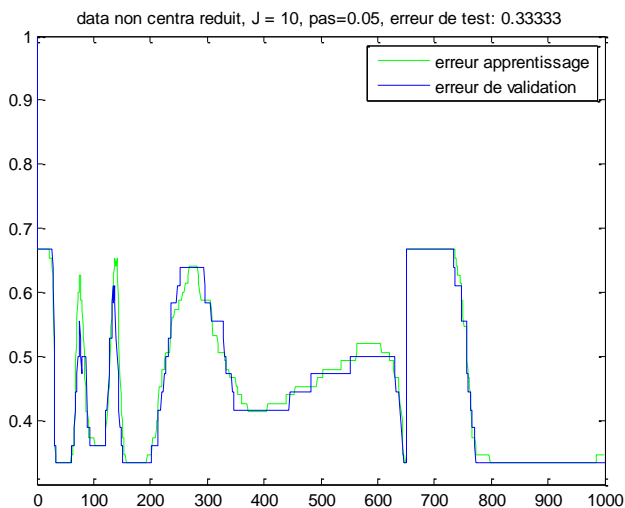
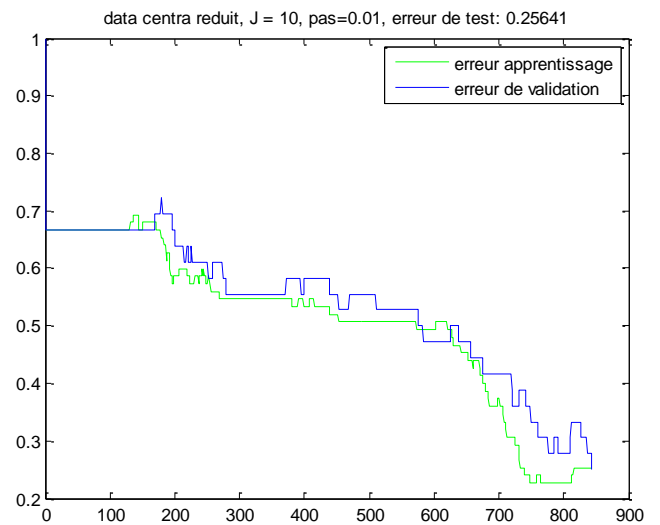
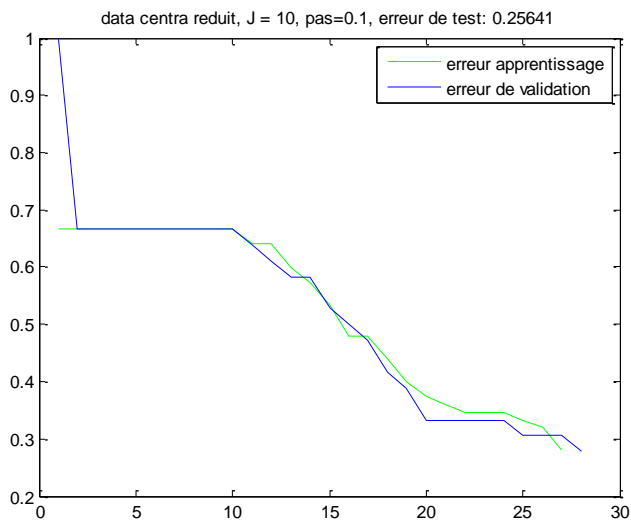


Figure 6: $J = 5$, $\text{pas} = 0.5$, data non centré-réduit

➤ f1 = sigmoid, f2 = softmax



Conclusion

En comparant les résultats de différentes fonctions d'activation, nous observons que le réseau avec une couche de sortie de type softmax est mieux que sigmoïde. La fonction d'activation a une grande influence sur le taux de reconnaissance et la vitesse de convergence.

Le nombre de neurone J n'a pas de l'influence signifiante sur le taux de reconnaissance mais sur le coût de calcul et temps de calcul.

Le choix du pas est aussi important, ce paramètre influe la vitesse de convergence et si le réseau peut converger ou non. Si le pas est très petit, le réseau converge mais lentement. Si le pas est très grand, le réseau peut ne converger pas.