

Scala & Java

Perceptron multi couches

Vérifiez les conventions à suivre pour l'écriture du code (cf Moodle).

Vous allez utiliser votre librairie perceptron dans ce TD, apprendre à utiliser Gnuplot et explorer la réponse du réseau.

Récupérez l'utilitaire gnuplot sur internet : cet outil permet de tracer simplement des graphiques. Il existe une version windows sans installation (dans un zip). Normalement gnuplot est intégré de base à la majorité des distributions linux.

A partir de votre programme, faites apprendre le XOR par votre perceptron une seule couche cachée de 4 neurones et un pas d'apprentissage de 0.1, 10000 rétropropagations, et sauvegardez dans un fichier erreur.dat les erreurs commises pour chaque pas d'itération d'apprentissage. Votre fichier erreur.dat aura la forme :

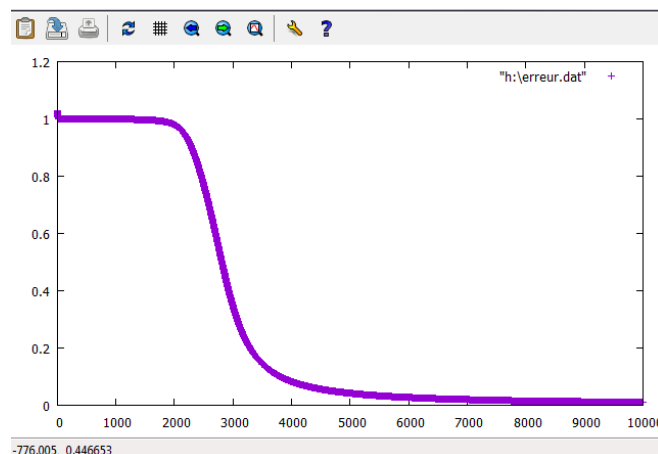
```
0 0.955667
1 0.5674
2 0.52345
```

La coordonnée X du graphique, suivie d'espaces et la coordonnée y (un couple par ligne)

Si vous avez besoin d'utiliser scala.tools (cf cours), n'oubliez pas de rajouter scala.compiler.jar à vos librairies. Rappel : `scala.tools.nsc.io.File("erreur.dat").writeAll("hello")`
Ouvrez Gnuplot et fabriquez le graphisme avec la commande :

```
gnuplot> plot "h:\erreur.dat"
```

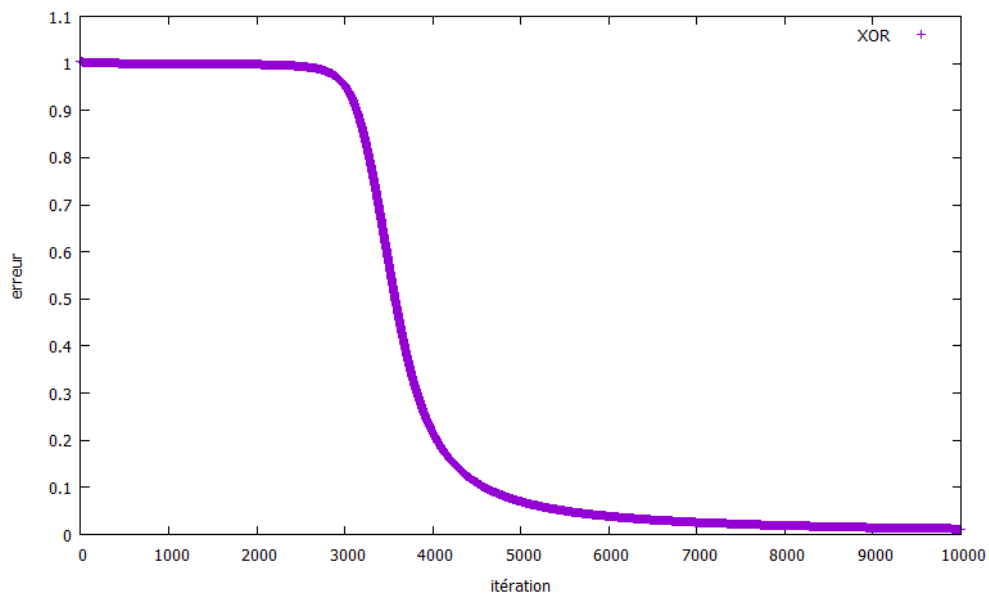
Vous devriez obtenir (un graphe qui ressemble à) cette courbe :



Gnuplot est un outil simple et extrêmement puissant et particulièrement adapté quand on souhaite visualiser des résultats numériques 2D ou 3D sans pour autant programmer une interface graphique en Java ou Scala : il suffit de sauvegarder un fichier dans un format simple et d'utiliser les commandes pour fabriquer une courbe : il est possible ensuite d'exporter cette courbe en format image (png), vectoriel (svg) ou pdf etc..

Expérimentez (et consultez l'aide en ligne) pour rajouter titre et label :

```
gnuplot> set xlabel "itération"
gnuplot> set ylabel "erreur"
gnuplot> plot "h:\erreur.dat"
gnuplot> plot "h:\erreur.dat" title "XOR"
```

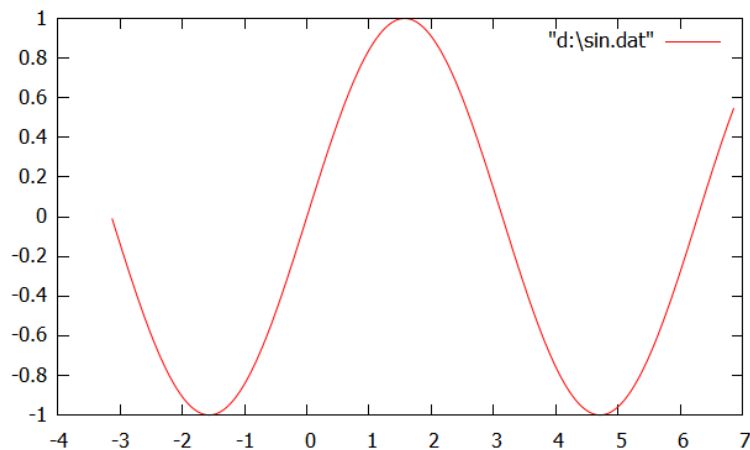


Utilisez CTRL-roue de la souris pour zoomer. Pressez la touche 'h' pour voir apparaître la liste des commandes interactives dans la console. Expérimentez.

Fabriquez maintenant la courbe qui représente un sinus dans le fichier sin.dat (montrez plusieurs périodes) :

```
plot "sin.dat" lt rgb "red" with lines
```

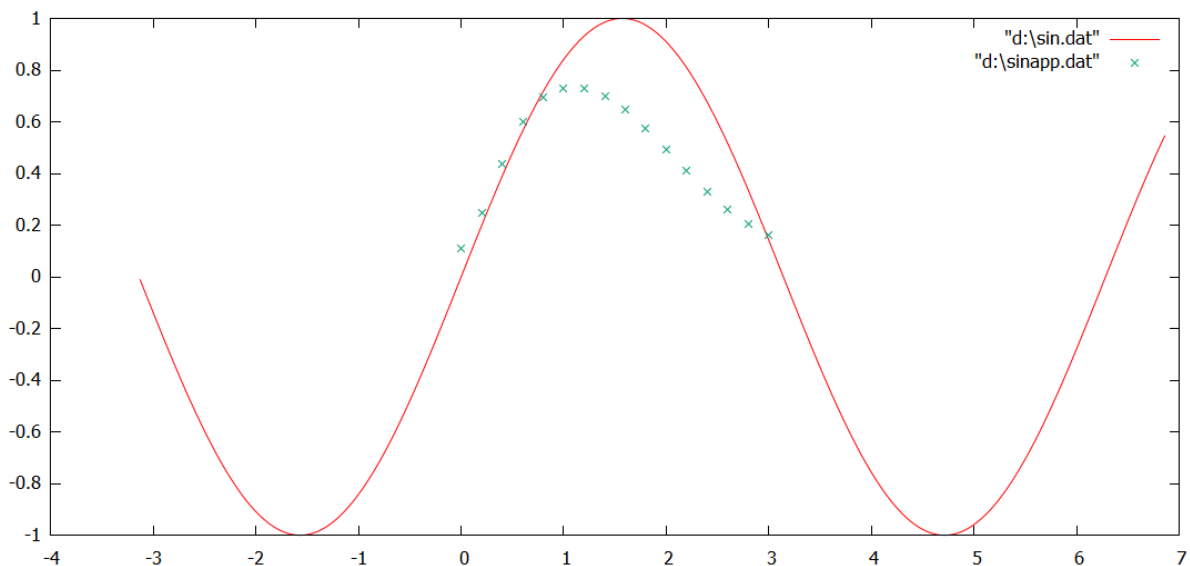
(regardez la doc de gnuplot)



Vous allez maintenant utiliser votre réseau pour tenter d'apprendre une partie de la courbe sinus. Vous allez prendre certaines valeurs de cette courbe pour constituer l'ensemble d'apprentissage : normalement le réseau devrait apprendre les couples de (valeur, $\sin(\text{valeur})$).

Créez un réseau 1,5,1 avec une entrée, une couche cachée et une sortie et faites un apprentissage sur les valeurs entre 0 et 3.14 avec un pas de 0.2. Faites 10000 itérations avec un pas de 0.1. Sortez les valeurs obtenues dans le fichier sinapp.dat (n'effacez pas sin.dat) et affichez les deux courbes sur le même graphe de la manière suivante :

`plot "d:\sin.dat" lt rgb "red" with lines , "d:\sinapp.dat"`



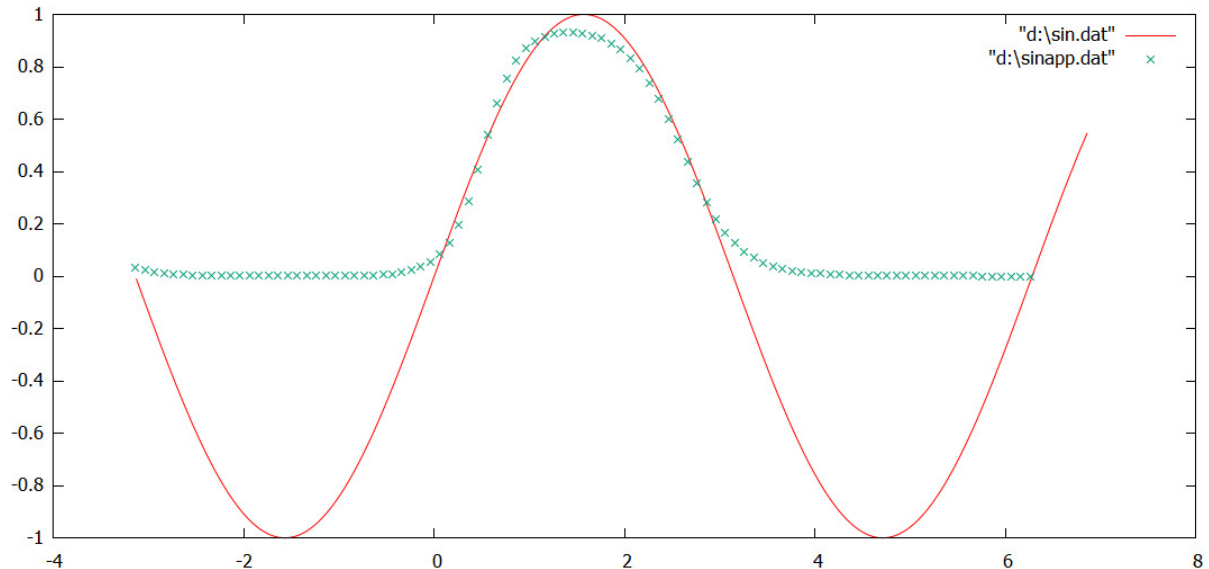
Dans un premier temps, vous allez visualiser sur ce graphe la position des points appris (et vérifier que ces points se trouvent bien sur la courbe du sinus).

Refaites le même apprentissage avec une (seule) couche cachée de 3 neurones puis de 4, 5 neurones, 10 neurones.

Dessinez les courbes. Que remarquez-vous ?

Augmentez le nombre de cycles d'apprentissage. Comparez les courbes.

Maintenant, histoire de voir ce que le réseau a appris, vous allez interroger le réseau sur des valeurs qu'il n'a pas appris : vous allez afficher la réponse du réseau entre -3.14 et 2×3.14 avec un pas de 0.1 entre chaque valeur.



Est-ce normal ? Pourquoi ?

Affichez sur la courbe du sinus. Expérimentez avec des tailles de couche cachées différentes.

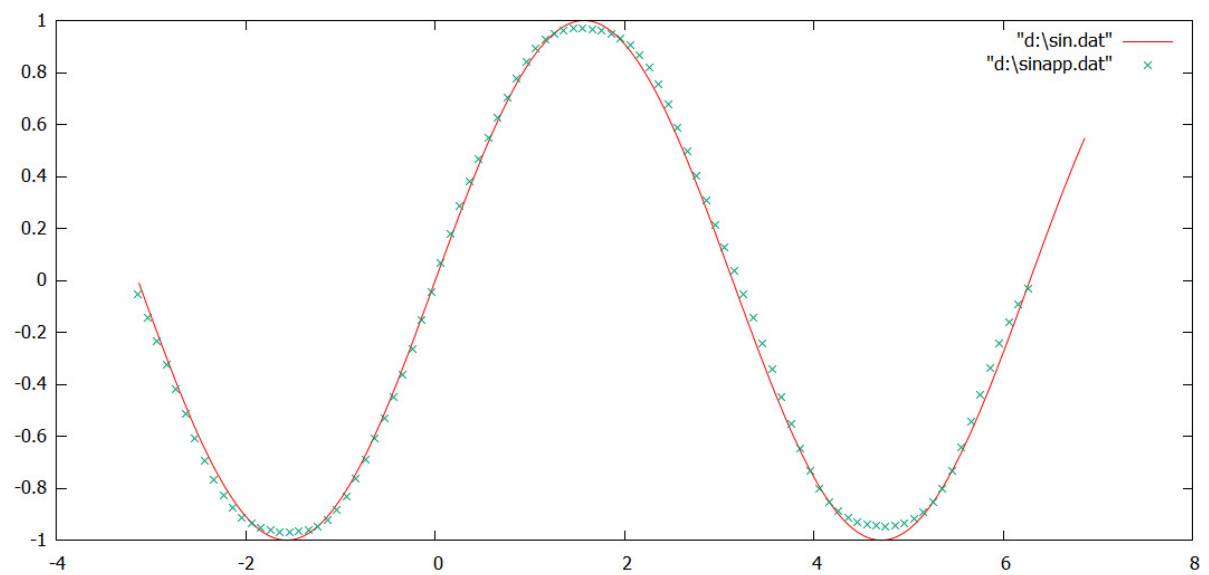
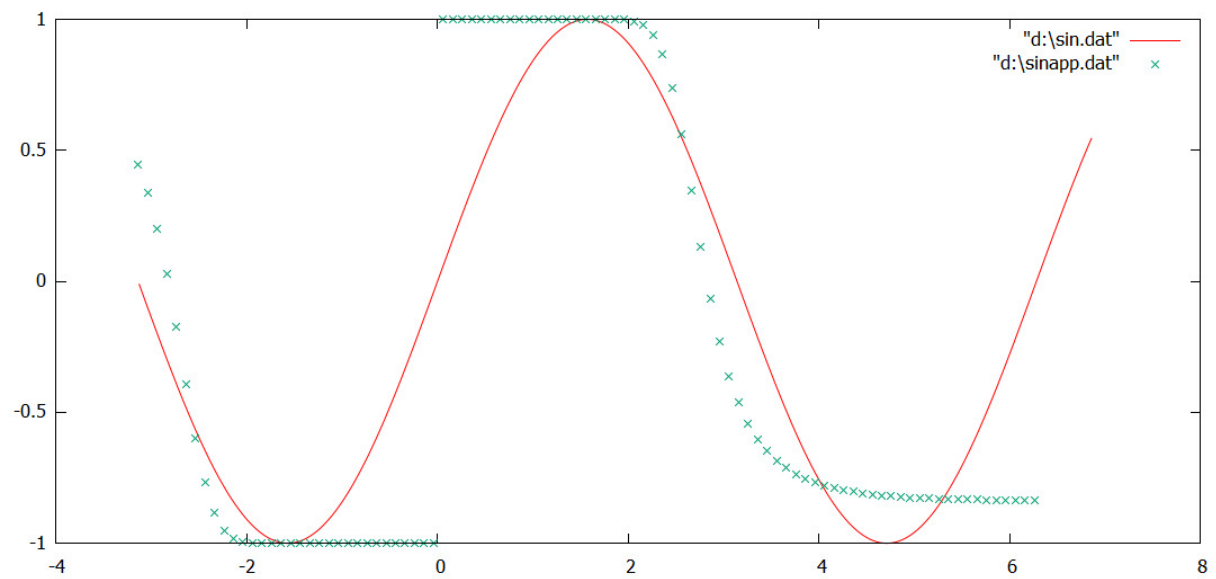
Refaites le test avec plusieurs couches cachées. Changez également le nombre de cycles d'apprentissage.

Refaites les mêmes tests, cette fois en étendant l'ensemble d'apprentissage entre 0 et 2×3.14

Que remarquez-vous sur le résultat ? pourquoi ?

Changez la fonction de transfert et recommencez l'apprentissage (la fonction et sa dérivée sont données dans le cours). Prenez la fonction tangente hyperbolique cette fois. Refaites les courbes. Conclusion ? Que se passe-t-il ? pourquoi ?

Essayez avec une configuration 1, 10, 1 et 1, 10, 5, 1 (avec deux couches cachées)



Notez bien quelle est la partie apprise et sur quelle partie on interroge le réseau. Conclusion ?

Rendez une archive avec vos sources et les graphes de vos expérimentations (en indiquant les paramètres utilisés, pas, apprentissage, ensemble d'apprentissage etc..) et vos remarques.