

# Outils pour les réseaux de neurones et contenu du CD-Rom

---

Depuis le développement théorique des réseaux de neurones à la fin des années 1980-1990, plusieurs outils ont été mis à la disposition des utilisateurs. Les fonctionnalités et les statuts de ces outils sont très variables. On consultera avec intérêt le site [www.aiaccess.net/f\\_ww.htm](http://www.aiaccess.net/f_ww.htm), pour un large panorama des produits disponibles.

Dans la catégorie des outils libres, fournis avec une licence de type GNU, ou analogue, on trouve principalement le travail de chercheurs de l'Université de Stuttgart, SNNS, disponible sous forme de code compilable sur le site [www-ra.informatik.uni-tuebingen.de/SNNS/](http://www-ra.informatik.uni-tuebingen.de/SNNS/).

La plupart des autres outils disponibles relèvent du monde commercial. Les grands éditeurs de logiciels de statistiques, comme SAS Institute ou SPSS, incluent des modules de réseaux de neurones dans leur offre.

Les éditeurs de logiciels de calcul scientifique ou de Data Mining à usage général proposent, la plupart du temps, une boîte à outils Réseaux de neurones. Dans ce cas, l'intérêt du logiciel réside dans l'accumulation des nombreuses possibilités. Mais chacune de ces possibilités prises séparément n'est pas optimisée.

Les logiciels dédiés, tel Neuro One proposé ici en version d'évaluation, sont spécialisés et offrent des fonctionnalités beaucoup plus proches des derniers résultats théoriques. Consultez à ce propos le site [www.netral.com/](http://www.netral.com/).

Le contenu du CD-Rom de cet ouvrage est le suivant :

- une version d'évaluation de Neuro One, valide 30 jours ;
- cinq exemples de modèles, avec données et codes source ;
- une bibliothèque de modélisation non linéaire NDK\_0 libre ;
- un compilateur C pour Windows.

Parmi les logiciels dédiés, Neuro One, édité par Netral, est l'un des plus anciens, et celui qui a le plus évolué pour rester au fait des derniers développements. C'est aussi, à notre connaissance, le seul qui offre un calcul des intervalles de confiance et des leviers sur les modèles développés.

Neuro One fournit un produit annexe, Neuro Code, qui permet de convertir un modèle neuronal en code source C. Avec ce code, il devient possible d'utiliser le modèle neuronal sous tout système d'exploitation qui admet un compilateur C. Ce code permet également l'apprentissage dans le nouvel environnement.

Les exemples de codes source présentés dans le CD-Rom ont été réalisés avec Neuro Code.

## Installer Neuro One

La configuration minimale requise pour l'installation de Neuro One est la suivante :

- processeur Pentium 2, équivalent ou supérieur ;
- fréquence supérieure à 400 MHz ;
- Windows NT4, 2000, XP ou plus récent ;
- espace disque disponible : 40 Mo ;
- mémoire vive disponible : 100 Mo.

Neuro One 6.10.7 est un outil fonctionnant sous Windows (Windows NT, Windows 2000, Windows XP).

Dans le répertoire NeuroOne, lancez le fichier SetUp.exe. Vous pouvez également cliquer sur Installer Neuro One à la page d'accueil du CD-Rom. Cliquez ensuite sur Ouvrir.

La boîte de dialogue d'introduction suivante apparaît alors, elle vous permet de choisir la langue du programme d'installation.

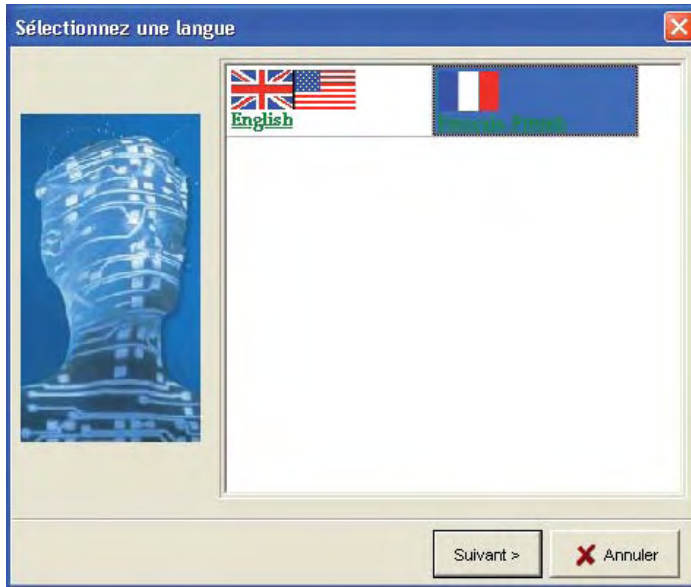


Figure A-1

Cliquez sur Suivant pour afficher la fenêtre d'introduction.

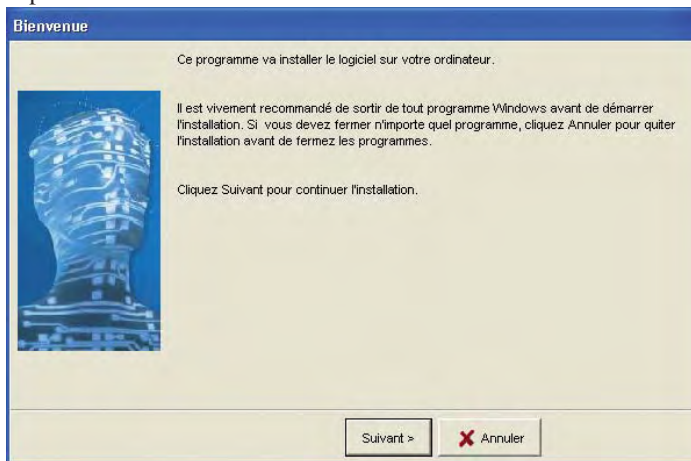


Figure A-2

Cliquez à nouveau sur Suivant et la boîte de dialogue du contrat de licence de Neuro One s'affiche.

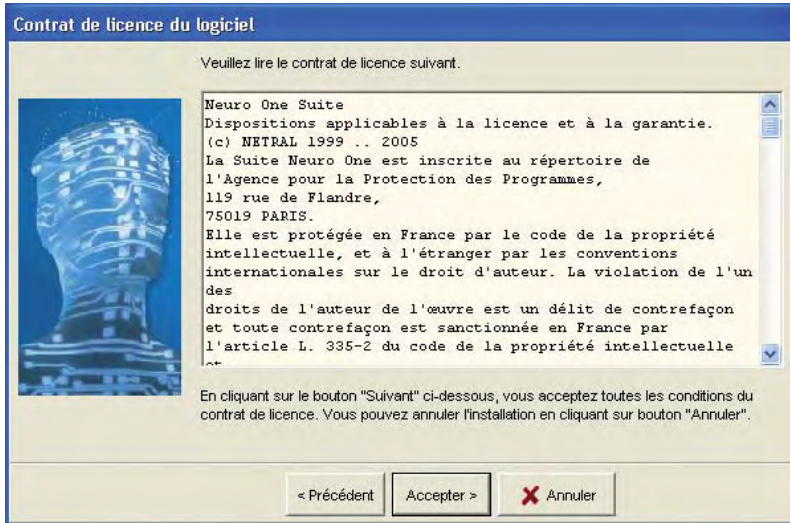


Figure A-3

Lisez attentivement ce contrat. Si vous n'êtes pas d'accord avec les termes de ce contrat, cliquez sur Annuler. Dans ce cas, vous ne pouvez pas utiliser Neuro One. Si vous êtes d'accord avec les termes de ce contrat, cliquez sur Accepter.

Lorsque la boîte de dialogue suivante apparaît, entrez votre nom, votre organisation, et le chiffre 0 (zéro), puis cliquez sur le bouton Pour une évaluation 30 jours.



Figure A-4

Puis, cliquez sur Suivant, la boîte de dialogue suivante apparaît.

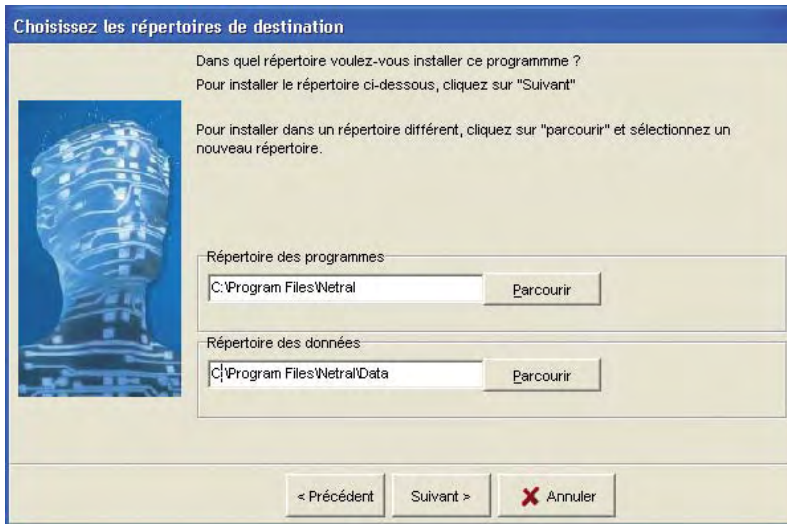


Figure A-5

Choisissez à présent le répertoire où seront enregistrés tous les fichiers de travail de Neuro One et cliquez sur Suivant pour accéder à la prochaine fenêtre.

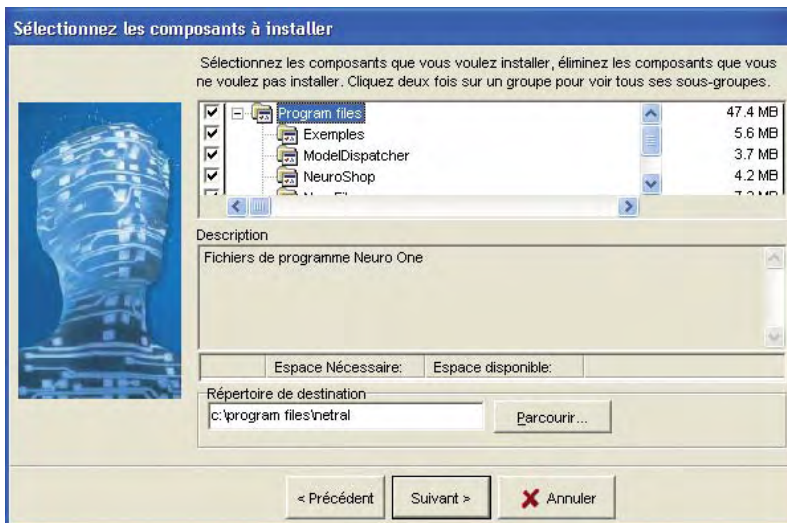


Figure A-6

Sélectionnez Par défaut, et cliquez sur Suivant pour passer à la fenêtre qui suit.

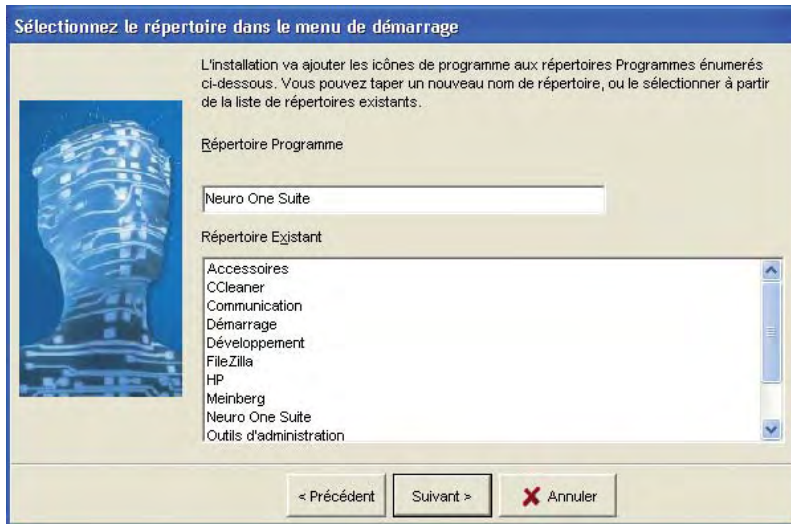


Figure A-7

Sélectionnez un répertoire de programme, et cliquez sur Suivant.

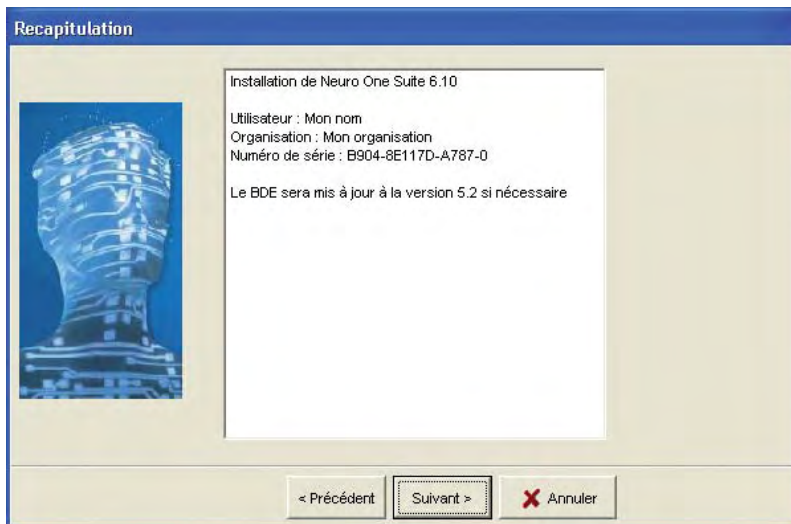


Figure A-8

Cette fenêtre présente un résumé de vos demandes. Si vous êtes satisfait, cliquez sur Suivant. Sinon, revenez en arrière en cliquant sur Précédent, corrigez l'erreur, et revenez en cliquant autant que nécessaire sur Suivant.

Le processus d'installation est alors engagé et se poursuit.

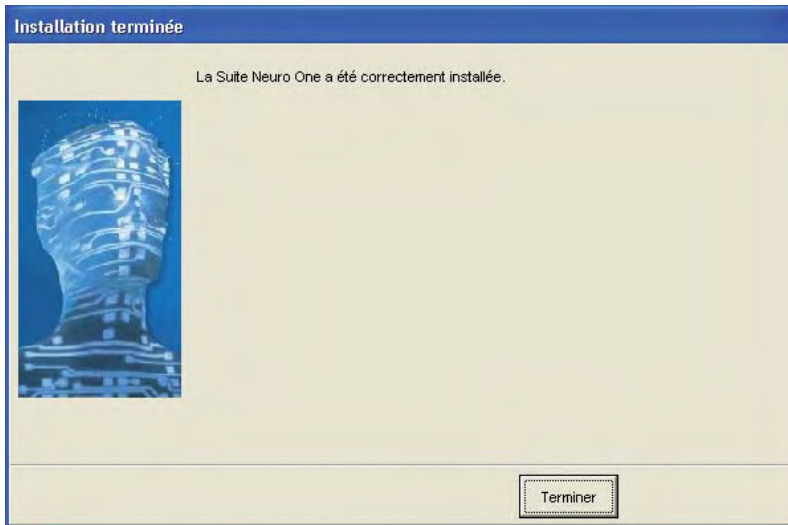


Figure A-9

Cliquez alors sur Terminer.

En cas de problème lors de l'installation, ou dès le premier démarrage de Neuro One, reportez-vous au site web de NETRAL [www.netral.com/index-fr.html](http://www.netral.com/index-fr.html), à la page support technique.

## Présentation des exemples

### Exemple 1

Voici un exemple académique, avec une entrée et une sortie :

Une entrée  $X$  varie de  $-1$  à  $+1$ .

Une sortie  $Y$  et une sortie  $Y_b$  bruitées sont disponibles.

Le modèle cherche à retrouver  $Y$  en utilisant les données  $X$  et  $Y_b$ .

Les modèles présentés comportent 0, 1, 2, 3, 5 et 10 neurones cachés.

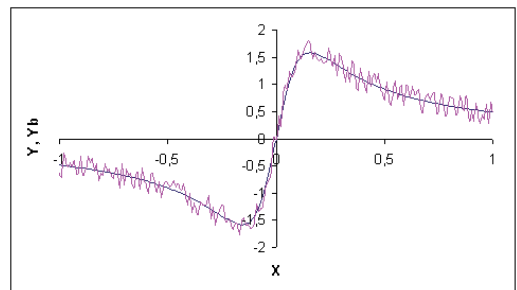


Figure A-10

### Exemple 2

Il s'agit de la construction d'un modèle du LogP de molécules chimiques en fonction de quelques descripteurs de la molécule.

Cet exemple est tiré de la publication :

*Toward a Principled Methodology for Neural Network Design and Performance Evaluation in QSAR. Application to the Prediction of LogP.* A. F. Duprat, T. Huynh et G. Dreyfus. J. Chem. Inf. Comput. Sci. 1998, 38, 586-594.

Elle est utilisée ici avec l'aimable autorisation de A. F. Duprat.

Les entrées sont des caractéristiques des molécules qui sont au nombre de 7. La sortie représente le LogP.

Les modèles présentés comportent 2, 4, 6 et 7 neurones cachés.

### Exemple 3

Cet exemple modélise la température de liquidus de verres binaires Lithium/Silicium.

L'entrée est la fraction molaire de  $\text{LiO}_2$ . La sortie est la température de liquidus.

Les modèles présentés comportent 2, 4, 5 et 6 neurones cachés.

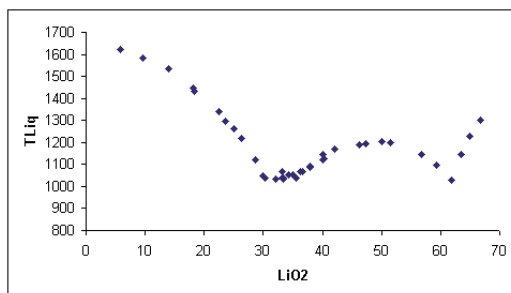


Figure A-11

### Exemple 4

Cet exemple modélise la température de liquidus de verres ternaires Aluminium/Potassium/Silicium.

Les entrées sont les fractions molaires de  $\text{Al}_2\text{O}_3$  et  $\text{K}_2\text{O}$ . La sortie est la température de liquidus.

Les modèles présentés comportent 2, 4, 6, 8 et 10 neurones cachés.

### Exemple 5

Cet exemple modélise la température de liquidus de verres quaternaires Sodium/Calcium/Aluminium/Silicium.

Les entrées sont les fractions molaires de  $\text{CaO}$ ,  $\text{Na}_2\text{O}$ ,  $\text{Al}_2\text{O}_3$ . La sortie est la température de liquidus.

Les modèles présentés comportent 2, 4, 6, 8 et 10 neurones cachés.

## Installation des exemples

Les exemples fournis peuvent être copiés dans un répertoire quelconque de votre machine, en respectant les termes de la licence rappelée à la fin de cette annexe à la section Licence. Pour effectuer l'installation, copiez le répertoire ncode, avec la totalité de son contenu et ses sous-répertoires sur votre machine.

#### Attention

Le répertoire ncode du CD-Rom doit être entièrement copié, avec son arborescence, sous peine de dysfonctionnements des compilations. Certaines inclusions de fichiers, nécessaires lors de la compilation, ont des adresses relatives. L'arborescence doit donc être conservée.

## Compiler le code source

Pour chaque projet, il existe deux fichiers nommés `makefileuse` et `makefiletrain`. Ces fichiers sont des fichiers Makefile destinés à la compilation des deux applications disponibles, dont les fichiers principaux sont : `xxxmainuse.c`<sup>1</sup> et `xxxmaintrain.c`<sup>1</sup>.

La compilation pourra être différente en fonction du compilateur C que vous utilisez. Si le compilateur GCC est disponible sur votre machine, le nom de la commande `make` est : `mingw32-make`.

Placez-vous dans le répertoire où sont situées les sources.

Pour compiler l'application d'utilisation, lancez la commande :

```
mingw32-make -f makefileuse
```

L'exécutable créé porte le nom de `xxxuse.exe`.

Pour compiler l'application d'apprentissage, lancez la commande :

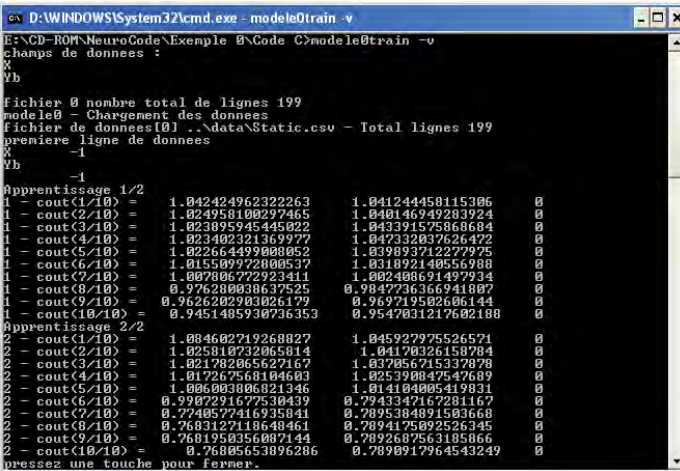
```
mingw32-make -f makefiletrain
```

L'exécutable créé porte le nom de `xxxttrain.exe`.

## Exécuter le code source

Les exécutables compilés peuvent être produits. Chacun nécessite un fichier de description nommé `ndesc.txt`. Lisez attentivement le fichier d'aide concernant les fichiers : `lisezmoidesc.txt`, dans le répertoire `doc`.

Ces fichiers permettent d'indiquer au programme comment lire les données, insérer les résultats ou ce qu'il faut calculer. Pour chaque exemple, un fichier de description a été proposé dans le répertoire des sources. Un exemple de lancement d'un programme d'apprentissage est donné ci-après :



```

D:\WINDOWS\System32\cmd.exe - modele0train -v
E:\CD-ROM\NeuroCode\Exemple 0\Code C>modele0train -v
champs de donnees :
X
Yb
Yb
Fichier 0 nombre total de lignes 199
modele0 - Chargement des donnees
Fichier de donnees[0] ..\data\static.csv - Total lignes 199
premiere ligne de donnees
X
-1
Yb
-1
Apprentissage 1/2
1 - cout(1/10) = 1.042424962322263 1.041244458115306 0
1 - cout(2/10) = 1.024958100297465 1.040146949283924 0
1 - cout(3/10) = 1.023895945445022 1.043391575868684 0
1 - cout(4/10) = 1.023402321369777 1.047332037626472 0
1 - cout(5/10) = 1.022664499000652 1.039893712277975 0
1 - cout(6/10) = 1.015509722000537 1.031092140556988 0
1 - cout(7/10) = 1.007806722923411 1.002408691497934 0
1 - cout(8/10) = 0.976280038637525 0.9847736366941007 0
1 - cout(9/10) = 0.9626202903026179 0.969719502666144 0
1 - cout(10/10) = 0.9451485730736353 0.9647031217602188 0
Apprentissage 2/2
2 - cout(1/10) = 1.004602719268827 1.045927975526571 0
2 - cout(2/10) = 1.025816732065814 1.04170326158784 0
2 - cout(3/10) = 1.021702065229197 1.037065715337070 0
2 - cout(4/10) = 1.017267568104603 1.025390847547689 0
2 - cout(5/10) = 1.006003806821346 1.014104005419831 0
2 - cout(6/10) = 0.9907291677530439 0.7943347167281167 0
2 - cout(7/10) = 0.7740577416935041 0.709304091503660 0
2 - cout(8/10) = 0.7683127118648461 0.7094175092526345 0
2 - cout(9/10) = 0.7681950356007144 0.7092687563185866 0
2 - cout(10/10) = 0.76805653896286 0.7090917964543249 0
apprenez une touche pour fermer.

```

Figure A-12

1. Les mentions « xxx » sont remplacées par le nom du modèle Neuro One qui est à l'origine du code.



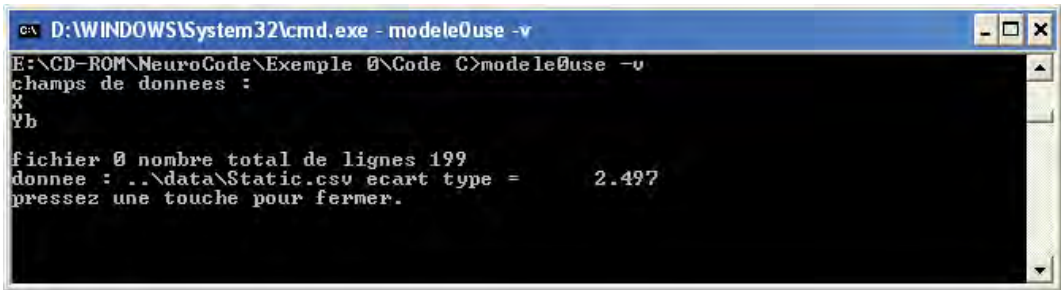
L'option `-v` permet un affichage détaillé. L'analyse du fichier de description donne deux champs de données : X et Yb. Le nombre total de lignes s'élève à 199 dans le fichier « `..\data\static.csv` ». Il y a deux apprentissages, chacun contenant 10 époques.

Pour chaque époque, les deux valeurs affichées sont respectivement l'écart-type d'apprentissage, et l'écart-type de généralisation obtenu par la méthode du Leave-One Out virtuel. Le dernier chiffre entier est le nombre de secondes écoulé depuis le début de l'apprentissage.

Vous pouvez consulter les fichiers créés dans le répertoire de résultat pour obtenir tous les détails de l'apprentissage :

- les fichiers « `xxxhistory.txt` »<sup>1,2</sup> retracent l'histoire de l'apprentissage ;
- les fichiers « `xxxresy.txt` »<sup>1,2</sup> donnent les coûts d'apprentissage et de généralisation, les poids et la matrice de dispersion ;
- le fichier « `xxxweights.txt` »<sup>1,2</sup> donne les poids et la matrice de dispersion de l'apprentissage qui présente le coût d'apprentissage le plus faible.

Le lancement du programme d'utilisation donne une fenêtre qui ressemble à la fenêtre suivante.



```

C:\ D:\WINDOWS\System32\cmd.exe - modele0use -v
E:\CD-ROM\NeuroCode\Exemple 0\Code C>modele0use -v
champs de donnees :
X
Yb

fichier 0 nombre total de lignes 199
donnee : ..\data\Static.csv ecart type = 2.497
pressez une touche pour fermer.
  
```

Figure A-13

Les fichiers de description et de données sont analysés. Les champs X et Yb sont trouvés. Le modèle est appliqué à toutes les données lisibles et complètes du fichier de données, et l'écart-type obtenu est affiché.

Vous pouvez consulter les fichiers créés dans le répertoire de résultat :

- les fichiers « `xxxusehisty.txt` »<sup>1,2</sup> retracent l'historique de l'utilisation ;
- les fichiers « `xxxusey.csv` »<sup>1,2</sup> donnent, pour chaque ligne du fichier de données, le résultat de l'application du modèle neuronal aux données présentées.

## Exécuter le code source Visual Basic

Pour les exemples proposés, un code Visual Basic est également fourni.

Pour compiler et exécuter ce code, il faut effectuer les opérations suivantes :

- ouvrir Excel, et l'éditeur Visual Basic d'Excel ;
- ouvrir un nouveau module ;
- copier le contenu du fichier « `xxx.bas` »<sup>1</sup> dans le code du nouveau module ;
- compiler la macro ;

2. Les mentions y sont remplacées par une valeur entière représentant le numéro de l'apprentissage ou de l'utilisation.

- retourner à Excel.

La macro xxx<sup>1</sup> est alors disponible, sous forme de fonction Excel.

## Visualiser les modèles

Pour chacun des exemples, le modèle est fourni sous forme de fichier xxx.nml<sup>1</sup>. Ces fichiers utilisent le langage XML, avec le fichier de schéma sur le site [www.netral.com/public/xml/xsd/model.xsd](http://www.netral.com/public/xml/xsd/model.xsd).

Ces fichiers sont prévus pour être visualisés par un navigateur Internet sous forme de dessin SVG, en utilisant le fichier de transformation XSLT `neuronnethm.xsl`, téléchargé depuis le site de NETRAL.

Si votre navigateur est capable de lire les fichiers d'images SVG, ouvrez le fichier de modèle NML, et vous obtiendrez le dessin du réseau de neurones du modèle. Sur ce dessin, les traits noirs représentent les synapses normales, susceptibles d'apprentissage. Les traits rouges représentent des synapses fixes, invariables pendant l'apprentissage. Ces synapses assurent la normalisation et le centrage des données, de façon à ce que le réseau de neurones puisse voir des données correctement calibrées.

## La librairie NDK (Neuro Developer Kit)

La librairie NDK est disponible en quatre niveaux de licences :

- niveau 0, `MonaEx70_0.d11` : réseaux de neurones pour régression et classification, réseaux de Kohonen, création de modèles neuronaux, chargement de données, apprentissage ;
- niveau 1, `MonaEx70_1.d11` : niveau 1 plus sauvegarde et chargement des résultats ;
- niveau 2, `MonaEx70_2.d11` : niveau 2 plus leviers, intervalles de confiance, analyse de données, analyse de réseaux, mode inverse, information détaillée, apprentissages multiples, calculs des gradients et hessiens directs et par différences finies ;
- niveau 3, `MonaEx70.d11` : niveau 2 plus modélisation de processus dynamiques, modification quelconque de modèles, modèles de connaissance, modèles compilés, génération de code.

Une version spéciale de `ndk` est capable de traiter les graph machines (modélisation structurée et modèles multiples à paramètres partagés).

Ce CD-Rom propose la librairie `MonaEx70_0.d11` de niveau 0. Cette librairie est gratuite et son utilisation ne peut en aucun cas engager la responsabilité de NETRAL. La documentation de la librairie `MonaEx70` peut être consultée dans le fichier `MonaEx70.chm`.

## Programme de démonstration de la librairie

La librairie NDK est fournie avec un programme de démonstration écrit en Python. Ce programme a été développé et testé avec Python 2.4. Il nécessite la présence des modules `Pylab` et `Numeric`.

Pour le lancer, ouvrez une invite de commande sous Windows. Placez-vous dans le répertoire où se trouve le NDK, et lancez la commande :

```
python.exe demo.py
```

Ce programme crée un modèle neuronal à une entrée, une sortie et deux neurones cachés, charge un jeu de données, lance un apprentissage, et affiche le résultat.

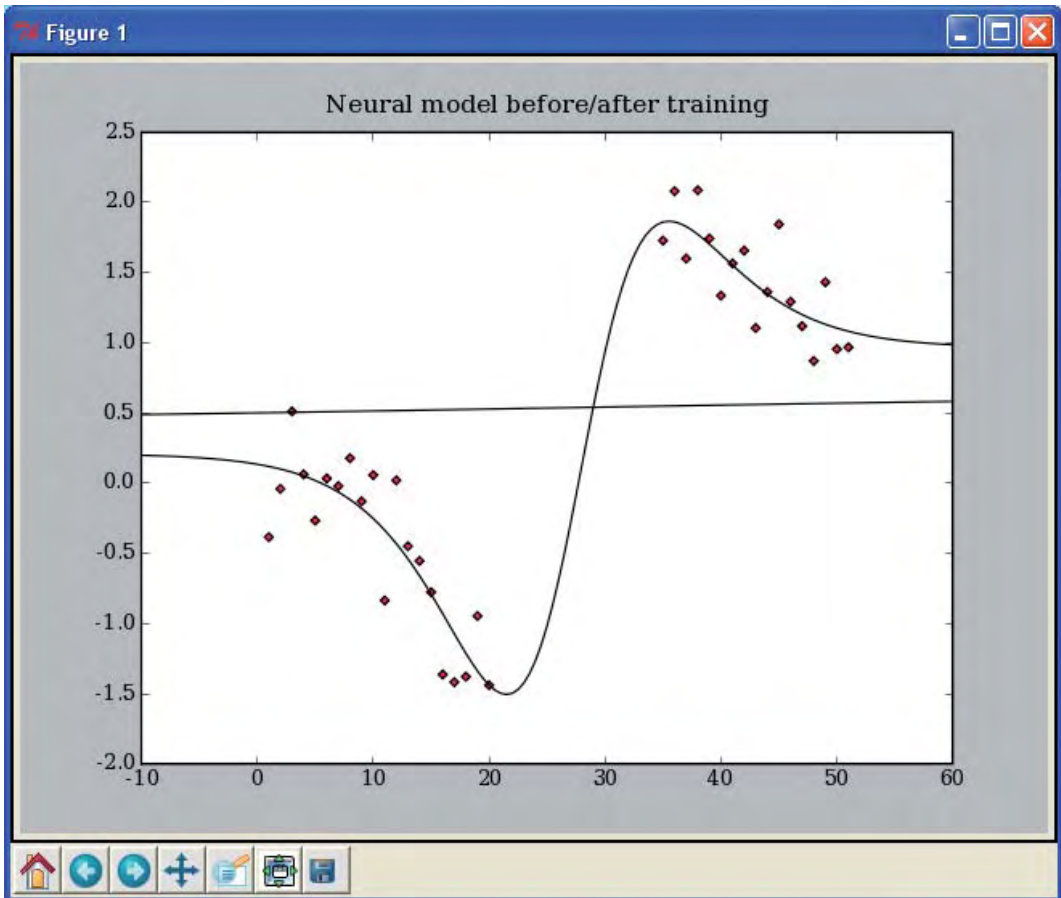


Figure A-14

Les points rouges représentent les exemples présentés, et les deux courbes pleines, la réponse du modèle neuronal avant et après apprentissage.

## Les compilateurs C

La compilation des exemples fournis nécessite la présence d'un compilateur C. Les utilisateurs de Windows trouveront ici deux exemples de compilateurs gratuits :

- GCC, disponible sous licence GNU, dans le répertoire gcc ;
- Turbo C, mis à disposition par Borland, sur le site <http://community.borland.com/museum>.

Pour l'installation d'un compilateur C, il est recommandé de disposer des droits d'administrateur. Pour installer GCC, cliquez sur `MinGW-2.0.0-3-gnuwin.exe` dans le répertoire gcc. Vous pouvez également cliquer sur Installer GCC dans la page d'accueil, puis sur Ouvrir.

À la fin de l'installation du compilateur, assurez-vous que celui-ci peut être appelé depuis tous les répertoires de votre machine. Pour cela, il peut être nécessaire de modifier la variable d'environnement PATH de votre machine en y incluant le chemin du binaire du compilateur.

## Licence

La licence de Neuro One est lisible pendant l'installation du logiciel, et doit être acceptée avant l'installation complète de celui-ci. Les codes source fournis en langage C relèvent de la licence suivante.

**Les présents codes source générés par le progiciel NEURO CODE sont fournis à titre gracieux par la société NETRAL. Ces codes sont protégés tant par les dispositions nationales qu'internationales en matière de droits de la propriété intellectuelle, dont les droits sont détenus, à titre exclusif, par la société NETRAL.**

L'utilisation et la modification de ces codes source sont soumises à un contrat de licence d'utilisation.

Ces codes sont utilisés sous la responsabilité pleine et entière de l'utilisateur. La société NETRAL ne saurait en aucun cas être tenue pour responsable des résultats de cette utilisation, tant sur les machines qui les utilisent que sur les données incluses dans ces machines.

La modification ou la copie même partielle de ce code, est strictement interdite, à l'exception des parties de commentaire et des fichiers de description. L'utilisateur possesseur du CD-Rom est autorisé à faire une unique copie de ce code à des fins de compilation. Toute autre copie de ce code est strictement interdite.

L'utilisation de ce code à des fins commerciales est strictement interdite. On entend, par fin commerciale, toute cession à titre onéreux du code lui-même, ou toute cession, à titre onéreux ou à titre gratuit, des résultats obtenus par l'utilisation de ce code une fois compilé.

**TOUTE EXTENSION DU DROIT D'UTILISATION NON PRÉVUE DANS CE CONTRAT DE LICENCE EST INTERDITE ET SERA CONSTITUTIVE D'UN ACTE DE CONTREFAÇON.**

La contrefaçon est un délit pénal, puni de 2 ans d'emprisonnement et de 150 000 € d'amende.

Le fait de copier le code en vue de sa compilation ou de le compiler sans le copier signifie que vous avez donné votre accord sur les termes de cette licence.

**Pour les fichiers principaux des programmes fournis, dont le nom se termine par « maintrain » et « mainuse », avec les extensions « .h » et « .c », et pour ceux-là seulement, la modification des codes est autorisée.**