

Manuel utilisateur

PMD

Perceptron multicouche didactique

Projet de semestre de 1ère année

EIVD classe EI2A – Avril - Juillet 2002

Frédéric JUNOD *frederic.junod@eivd.ch*

Mathieu BORNOZ *mathieu.bornoz@eivd.ch*

Table des matières

1 Informations générales	3
1.1 But du logiciel et généralités	3
1.2 Installation	3
1.2.1 Plate-forme Windows	3
1.2.2 Plate-forme GNU/Linux	3
2 Mode d'emploi	4
2.1 Problèmes lors du lancement	4
2.2 Utilisation étape par étape	4
2.2.1 Base de données d'entraînements	5
2.2.2 Réglage des paramètres	6
2.2.3 Initialisation du réseau	6
2.2.4 Entraînement du réseau	6
2.2.5 Graphe d'erreurs	7
2.2.6 Présentation d'une forme	7
2.2.7 Analyse des résultats	8
3 En résumé	9
3.1 Récapitulatif des différentes étapes	9
3.2 Informations complémentaires	9

1 Informations générales

1.1 But du logiciel et généralités

Ce programme a pour but de présenter un réseau de neurones de manière didactique. Ce modèle est plus particulièrement un perceptron multicouche paramétrable (pour savoir ce qu'est un réseau de neurones et plus particulièrement un perceptron multi-couche, nous vous conseillons vivement de lire le document général du projet. Ce dernier vous expliquera à quoi ils peuvent servir et vous aidera à comprendre le but de cette application et les explications qui vont suivre).

Tout l'intérêt de cette application réside dans le fait que le réseau est modulaire, ce qui est l'une des principales difficultés du perceptron multicouche. Le fait de pouvoir choisir les paramètres et affiner les réglages du réseau permet à l'utilisateur d'effectuer différents tests de validité et, en fonction des résultats, modifier les paramètres pour se rapprocher du but fixé.

1.2 Installation

Ce programme est portable (uniquement recompilation) sur les deux plate-formes ci-dessous :

1.2.1 Plate-forme Windows

Ce programme a été entièrement codé en Ada 95 et il est associé à GTK(ADA) pour l'interface graphique. Pour lancer l'application il vous faudra donc contrôler que la librairie `GtkAda` a été préalablement installée sur votre système.

Cette librairie est sous licence GPL et elle est disponible à l'adresse suivante :

<http://libre.act-europe.fr/GtkAda/>

A l'heure où nous écrivons ce manuel, l'exécutable pour l'installation de `GtkAda` a pour nom **GtkAda-1.3.12.exe**.

Notez qu'il ne faut pas oublier de copier les dll présentes dans le dossier généré par le programme d'installation, dans le répertoire `system` de Windows.

1.2.2 Plate-forme GNU/Linux

L'installation pour GNU/Linux n'est pas plus compliquée ; la librairie est disponible à l'adresse suivante :

<http://libre.act-europe.fr/GtkAda/>

Vous trouverez soit les sources au format compressé **tar.gz** ou encore pour les utilisateurs de la distribution Red-Hat ou Suse les sources au format **rmp**.

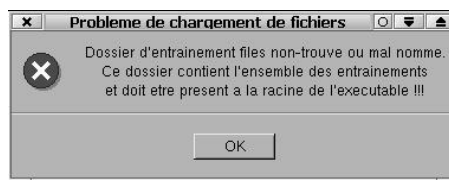
Il est peut-être plus prudent d'installer la version présente sur les CD de votre distribution ou sur les sites internet de mise à jour des paquetages pour des distribution comme Debian.

2 Mode d'emploi

2.1 Problèmes lors du lancement

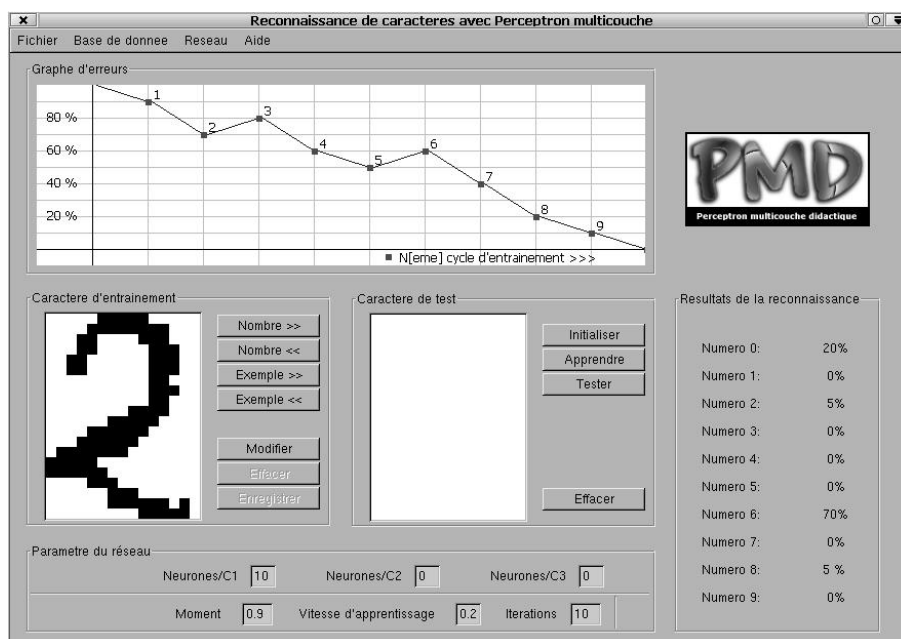
La seule erreur que vous pouvez rencontrer lors du lancement hormis une mauvaise installation de la librairie GTK est le fait de ne pas avoir les fichiers d'entraînements à la racine de l'exécutable (dans le répertoire `files`).

Cette erreur vous sera signalée par le biais d'une boîte de dialogue (voir ci-dessous) et le programme s'arrêtera.



Message d'erreur

2.2 Utilisation étape par étape

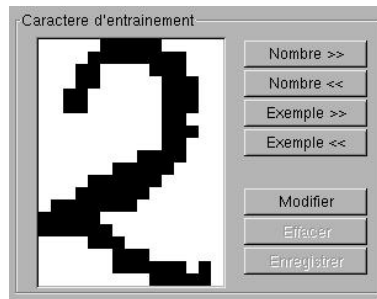


Interface complète du programme PMD

Ci-dessous nous décrivons les différentes étapes d'utilisation de ce programme. Le but final étant de faire reconnaître au réseau une forme représentant un nombre entre 0 et 9. Ce dernier est capable d'exprimer à quelle hauteur (pourcentage) il peut affirmer que c'est un 0, un 3 ou encore un 5.

Pour faire apprendre le réseau, une base de données de 100 exemple, (10 par nombre de 0 à 9) vous est fournie. Il vous est possible de modifier les exemples afin de tester différentes configurations de réseau.

2.2.1 Base de données d'entraînements



Ces différentes formes d'entraînement sont consultable par le biais de la zone graphique ci-dessus. Comme dit précédemment un mode édition vous est fourni afin de personnaliser et sauvegarder vos propres exemples.

Les modifications se font directement à l'aide de la souris (bouton gauche pour dessiner, bouton droit pour effacer). Une fois la forme modifiée appuyer sur enregistrer, et le tour est joué.

Notons que les différents exemples sont enregistrés dans des fichiers de la manière présentée ci-dessous, il est donc bien évidemment possible de les modifier directement ou à l'aide d'une petite moulinette de votre cru.

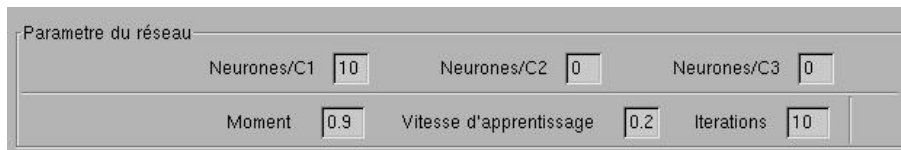
```

.....XX .....XX.. .....XXXXXXXX
.....XXXXX.. .XXXXXXXXXXXXX.. ..XXXXXXXXXX..
.XXXXX.XXX. .... .XXX..X.. ..XX.....
.....X... .XXX..... ..X...
..XXX..... ..XX.... .XXX.....
.....XXX.... ..XX..... ..XX....
..XX...XXX.. ..XX.... .XXXXXXXXXX..
.....XX. .XXX.XXXXXX. ....XX.
.....XX. ....XX. ....XXX
.....XX. .X.....XX .....XX..
..XX.....XXX ..X....XXX.. ..XX....XXXX
..XX...XXX.. ..XXX....XXX. ....XXX.XXX...
...XXXX.XXXXX. ....XXXX.... ..XXXXXXXXX...
.....XX.....
    
```

Remarques :

Attention la proportion des nombres dessinés doit être respectée (15x20 pixels) car nous n'avons pas traité le recadrage des formes présentées. Le réseau ne sera donc pas capable d'apprendre si les exemples sont de taille différentes.

La lecture des fichiers ne comporte pas de gestion d'erreurs approfondie, donc si vous modifiez les exemples directement dans les fichiers, veillez à respecter le format et ne pas rajouter des lignes inutiles ou des formes supplémentaires.

2.2.2 Réglage des paramètres

Parametre du réseau					
Neurones/C1	10	Neurones/C2	0	Neurones/C3	0
Moment	0.9	Vitesse d'apprentissage	0.2	Iterations	10

La zone du programme ci-dessus permet le réglage de certains paramètres comme la vitesse d'entraînement, le nombre de couches cachées ou encore le nombre de fois que vous voulez soumettre les exemples d'entraînement au réseau.

Nous nous sommes fixés sur un réseau pouvant aller de 0 à 3 couches cachées qui peuvent chacune posséder de 0 à 99 neurones. Le paramètre moment (plus connu sous le nom de momentum) est relativement subtil même si nous l'avons implémenté dans ce programme, nous n'en connaissons pas encore réellement l'effet (il semblerait qu'il agit sur la vitesse d'apprentissage).

Remarques :

Nous vous conseillons vivement de fixer votre attention sur le nombre de couche et de neurones car il s'agit là des principaux paramètres. En ce qui concerne les paramètres **Moment** et **Vitesse d'apprentissage** nous vous conseillons de laisser les valeurs par défaut et par la suite de les tester pour évaluer leur impact sur les résultats.

2.2.3 Initialisation du réseau

Cette partie consiste à initialiser le réseau avec des petites valeurs aléatoires, puis lui présenter en entrée et en sortie des couples de donnée ; une forme provenant des fichiers et ce qu'il doit reconnaître (apprentissage dit supervisé).

A partir de ce moment, le réseau est prêt à apprendre, une étape que nous allons vous décrire ci-dessous.

2.2.4 Entraînement du réseau

L'entraînement est la partie la plus importante pour la configuration du réseau. Elle consiste à lui faire apprendre les différentes formes et leurs significations (un 1, un 4,

etc ...). Pour ce faire nous allons lui présenter une à une les formes d'entraînement présentes dans les fichiers jusqu'à qu'il ai acquit un sens de la reconnaissance suffisamment développé (Voir partie graphe d'erreur).

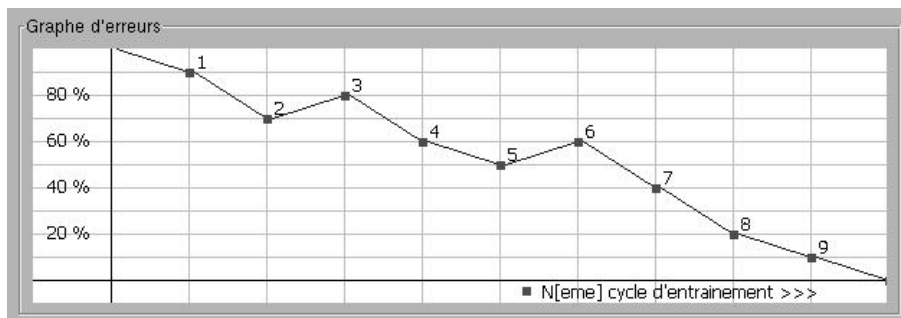
Pour fixer le nombre de fois qu'on veut lui faire apprendre les différentes formes, utilisez le paramètre itération dans la zone des paramètres. Il est bien sûr possible de relancer l'entraînement plusieurs fois de suite, le graphe vous aidera à voir l'évolution du réseau au cours des entraînements.

Remarques :

Si vous voulez réinitialiser le réseau, appuyer sur le bouton initialiser qui effacera la mémoire du réseau (lavage de cerveau) et le remettra à son état initial.

2.2.5 Graphe d'erreurs

Le graphe est la partie visible de l'iceberg, il représente l'évolution de l'erreur du réseau en fonction du nombre d'entraînements.



A la fin des itérations vous pouvez soit :

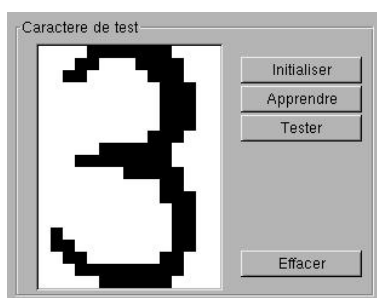
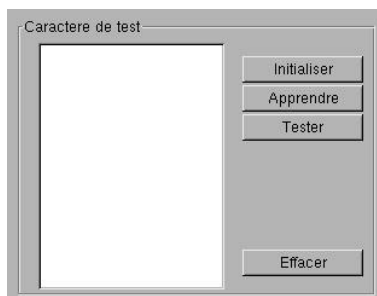
- Changer les paramètres, réinitialiser le réseau et recommencer une série d'entraînements.
- Continuer l'entraînement en espérant améliorer les résultats vus sur le graphe.

Remarques :

Un nombre trop important d'entraînement peut conduire le réseau à un état dit "poubelle", il s'agit donc de trouver les bons paramètres, le bon nombre de couches et le bon nombre d'entraînements (eh oui, c'est pas simple!!!)

2.2.6 Présentation d'une forme

Après toutes ces étapes vous allez enfin savoir ce que vaut votre réseau. Pour ce faire, dessiné un nombre (ressemblant) dans la zone de test (voir ci-dessous) et tester en appuyant sur le bouton créé à cet effet.



2.2.7 Analyse des résultats

Vous trouverez dans la zone (présentée ci-dessous les résultats de la reconnaissance de forme, ce qui va permettre d'évaluer les capacités du réseaux et son aptitude à la reconnaissance.

The screenshot shows a window titled "Resultats de la reconnaissance" containing a list of digits and their corresponding recognition percentages.

Numero 0:	20%
Numero 1:	0%
Numero 2:	5%
Numero 3:	0%
Numero 4:	0%
Numero 5:	0%
Numero 6:	70%
Numero 7:	0%
Numero 8:	5%
Numero 9:	0%

Remarques :

Il est important de tester le réseau avec plusieurs formes, certaines pouvant être mieux reconnues que d'autres. Un bon réseau est capable de s'en sortir avec toute les formes (résultats entre 60 et 95%).

3 En résumé

3.1 Récapitulatif des différentes étapes

- [1] Réglage des différents paramètres (ou utilisation des valeurs par défaut)
- [2] Initialiser le réseau (par défaut au démarrage du programme)
- [3] Lancement de l'entraînement (le graphe doit être très proche de zéro)
- [4] En fonction du pourcentage d'erreur
 - Relancer l'entraînement.
 - Modifier les paramètres puis réinitialiser.
- [5] Dessiner une forme et la présenter au réseau
- [6] Analyser les résultats

3.2 Informations complémentaires

Pour de plus amples informations concernant cette application, nous restons bien sûr disponible et prêt à y répondre, dans la mesure de nos compétence et nos connaissances en matière de réseaux de neurones.

Frédéric JUNOD *frederic.junod@eivd.ch*
Mathieu BORNOZ *mathieu.bornoz@eivd.ch*