Classification par apprentissage : une étude comparative

J. Manuel Torres Moreno et Mirta B. Gordon*
Département de Recherche Fondamentale sur la Matière Condensée - SPSMS
CEA/Grenoble - 17, rue des Martyrs - 38054 Grenoble Cedex 9, France
Mirta.Gordon@cea.fr, Manuel.Torres@cea.fr

Résumé : Nous comparons la performance en généralisation de l'algorithme incrémental Monoplan, qui construit un réseau de neurones à une couche cachée d'unités binaires, à celle d'autres algorithmes, pas nécessairement neuronaux, sur trois problèmes bien connus. En général, les réseaux de neurones, et plus particulièrement ceux construits avec des approches incrémentales, produisent les meilleures généralisations. Les réseaux construits avec Monoplan ont un très petit nombre d'unités, grâce aux performances de Minimerror, l'algorithme utilisé au niveau de l'apprentissage des neurones individuels. Cette petite taille explique, au moins en partie, les excellents résultats obtenus avec Monoplan en généralisation.

Mots-clés : Apprentissage incrémental, généralisation, réseaux de neurones multicouches.

1 Introduction

Ces dernières années, de nouvelles techniques neuronales d'apprentissage ont été développées. Cependant, il existe un important décalage entre les prédictions théoriques et les performances des algorithmes. Par exemple, bien qu'un réseau de neurones avec une seule couche cachée puisse approximer toute fonction des entrées, le nombre d'unités cachées n'est pas connu. Les bornes fournies par la dimension de Vapnik-Chervonenkis, de même que le nombre de données nécessaires d'après la théorie d'apprentissage PAC, sont excessifs, donc inutilisables pour les applications. Hormis le cas du perceptron, notre compréhension du problème de la généralisation est encore insuffisante. Nous sommes donc réduits à comparer les algorithmes d'apprentissage sur des problèmes étalon. L'apprentissage avec des réseaux de neurones se fait actuellement suivant deux approches. Certains algorithmes, comme la Retropropagation du Gradient (BP), ont besoin d'introduire a priori le nombre et la connectivité des unités cachées, et déterminent les poids des connexions par minimisation d'un coût. Le réseau obtenu est éventuellement élagué (Le Cun *et al*. 89 ; Hassibi *et al*. 93), ce qui, en termes d'inférence non paramétrique (Geman *et al*. 92), permet de diminuer la variance. Avec une approche constructive ou incrémentale (Torres-Moreno *et al*. 95), on apprend au même temps le nombre d'unités et les poids, dans le cadre d'une architecture fixée, commençant généralement avec une seule unité. Ce fort biais est réduit par l'introduction successive de nouvelles unités, afin de produire une application des entrées sur des représentations internes (RI) qui soit fidèle. Ce procédé pourrait engendrer des réseaux avec un nombre excessif de neurones, produisant de mauvaises généralisations. Les exemples présentés dans ce travail montrent que ce phénomène, appelé surapprentissage, n'est pas intrinsèque à l'approche incrémentale.

Nous présentons les résultats obtenus avec l'algorithme incrémental Monoplane (Torres-Moreno *et al*. 95) sur trois problèmes connus, et nous les comparons aux meilleurs résultats trouvés dans la littérature. Cette comparaison est faite sur la base des erreurs de généralisation. D'autres critères, comme le temps d'apprentissage, ou la complexité

* Centre National de la Recherche Scientifique (CNRS)

algorithmique, beaucoup plus difficiles à mesurer, ne seront pas considérés. Sur ces bases de données, les réseaux de neurones généralisent mieux que les autres techniques d'apprentissage. Nous constatons aussi que les algorithmes incrémentaux ne présentent pas de surapprentissage, mais au contraire, ils engendrent souvent des réseaux plus petits que ceux dont les algorithmes non-incrémentaux (même avec élagage) ont besoin pour atteindre les mêmes performances. Monoplane permet d'obtenir les meilleurs taux de généralisation dans la plupart des cas étudiés, grâce à la qualité de l'algorithme Minimerror, utilisé pour l'apprentissage des neurones individuels.

2 L'algorithme incremental Monoplane

Monoplane (Torres-Moreno *et al*. 95 ; Gordon 96) génère un réseau de neurones avec une seule couche cachée d'unités binaires, connectée à un neurone binaire de sortie. Chaque unité cachée ajoutée corrige des erreurs d'apprentissage de l'unité précédente, suivant une heuristique engendrant une machine de parité : les sorties voulues sont la parité des représentations internes (RI). Si le neurone de sortie détecte que les RI ne sont pas linéairement séparables (LS), la dimensionalité de la couche cachée est augmentée jusqu'à ce qu'elles le soient. Monoplane réduit le problème à celui de l'apprentissage par des perceptrons. Sa performance est dont contrôlée par celle de l'algorithme d'apprentissage utilisé pour ces unités. Nous utilisons Minimerror (Gordon *et al*. 93), qui minimise la fonction de coût :

$$C = \frac{1}{2}\sum_{\mu=1}^{P}\left[1-\tanh\left(\gamma^{\mu}/2T\right)\right] \qquad (1)$$

dans l'espace des poids $\vec{\mathbf{w}} = (\mathrm{w}_0, \mathrm{w}_1, \ldots, \mathrm{w}_N)$. P est le nombre d'exemples, $\gamma^{\mu} = \tau^{\mu}\vec{\mathbf{w}}\cdot\vec{\xi}^{\mu} / \sqrt{\vec{\mathbf{w}}\cdot\vec{\mathbf{w}}}$ est la stabilité ou marge des entrées $\vec{\xi}^{\mu} = \left(1, \xi_1^{\mu}, \ldots, \xi_N^{\mu}\right)$ (le neurone 0 représente le seuil), N est le nombre de neurones d'entrée, τ^{μ} sont les cibles à apprendre. La minimisation est faite par une descente de gradient simple combinée avec un recuit déterministe (la "température" T est diminuée durant l'apprentissage). La fonction de coût (1) représente une mesure bruitée du nombre d'erreurs d'apprentissage. Il a été montré théoriquement, avec l'approche de la physique statistique et la méthode des répliques et vérifié numériquement (Gordon *et al*. 93 ; Raffin *et al*. 95) que Minimerror permet d'obtenir une probabilité de généralisation maximale si l'ensemble d'apprentissage est LS, et que si non, il minimise le nombre d'erreurs d'apprentissage.

3 Résultats

Nous présentons nos résultats, ainsi que ceux de différents auteurs, sur trois bases d'apprentissage : le problème de Monk, celui des formes d'ondes de Breiman et le diagnostic du cancer du sein des données de l'Hôpital de l'Université de Wisconsin. Les résultats sont présentés sous forme graphique. Les différents algorithmes, triés par leurs erreurs de généralisation, sont portés en abscisses. La ligne unissant les points successifs, qui n'est qu'un guide pour les yeux, est donc toujours décroissante : plus un algorithme est performant, plus il est à droite et en bas sur les figures.

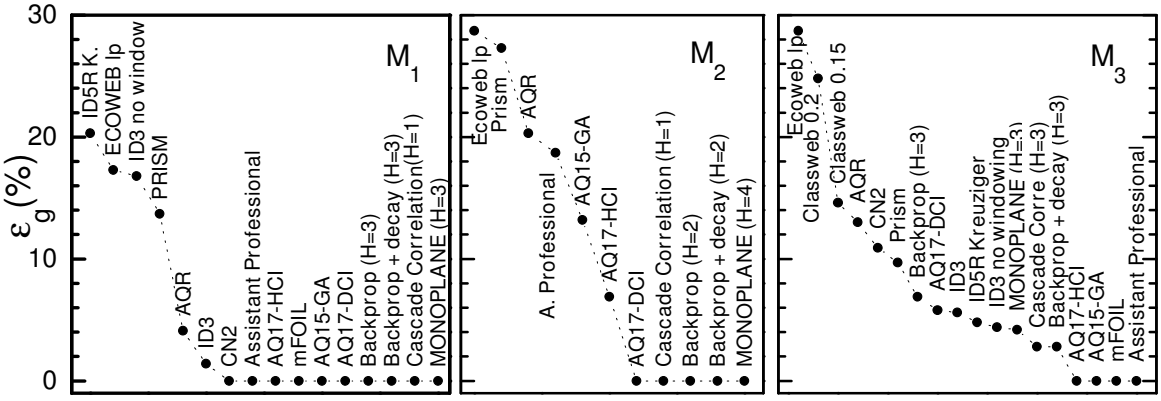

Figure 1 : Les problèmes de Monk. (H : nombre de neurones cachés)

3.1 Problèmes de Monk M_1, M_2, et M_3 (Thrun *et al*. 91)

Tous les algorithmes neuronaux généralisent correctement M_1 et M_2. Le nombre de neurones cachés oscille entre 2 et 4 suivant l'algorithme. Parmi les algorithmes non-neuronaux, environ la moitié échouent sur M_1, et un seul généralise correctement M_2. Cette tendance semble se renverser sur le problème M_3, dont l'ensemble d'apprentissage est bruité tandis que celui de test ne l'est pas. Un test non biaisé devrait se faire avec un ensemble contenant la même proportion d'erreurs, environ 5%, que l'ensemble d'apprentissage. Dans ces conditions, il existe une limite inférieure théorique à l'erreur de généralisation (Amari *et al*. 93), $\varepsilon_g \geq \varepsilon_t$. Les réseaux de neurones, et en particulier Monoplane, font effectivement environ 5% d'erreurs de généralisation, mais certains algorithmes d'intelligence artificielle spécialement conçus pour des situations bruitées, obtiennent $\varepsilon_g = 0$. Cependant, aucun de ces algorithmes ne généralise correctement les tâches non bruitées M_1 et M_2 : ils sont incapables de reconnaître un problème bruité d'un problème qui ne l'est pas.

3.2 Formes d'ondes de Breiman

Le groupe SYMENU (Gascuel *et al*. 95) a étudié en détail la performance de différents algorithmes sur ce problème, et trouve que celui qui présente les meilleurs taux de généralisation est la BP, le seul algorithme neuronal étudié. Cette performance y est expliquée par la quasi-séparabilité linéaire du problème. En effet, le meilleur ε_g de Minimerror est très légèrement supérieur à celui du meilleur réseau construit avec Monoplane. Ces résultats sont comparés à ceux d'autres algorithmes (Deffuant 95) sur la figure 2.

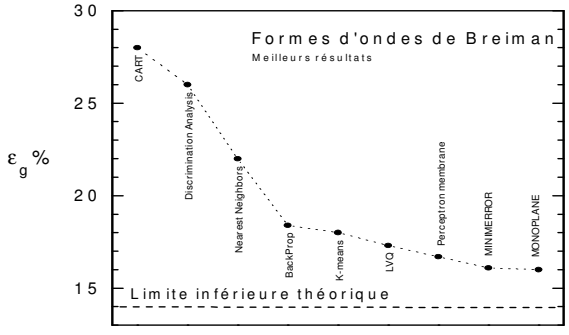

Figure 2 : Le problème des formes d'ondes

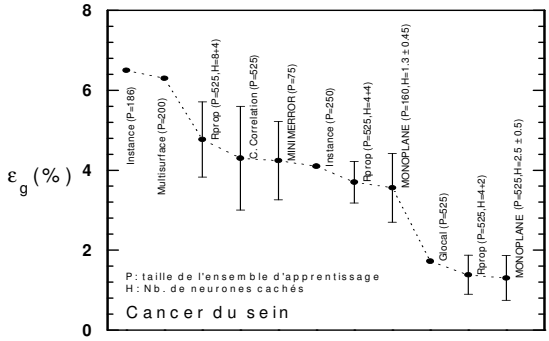

Figure 3 : Le problème du cancer du sein

3.3 Base de données de cancer du sein (Murphy *et al*)

Nous avons éliminé de cette base les 16 exemples avec attributs manquants. La figure 3 présente les résultats de différents algorithmes (Wolberg *et al*. 90 ; Prechelt 94 ; Zhang 92 ;

Depenau 95) ayant appris avec des ensembles d'apprentissage de tailles variables. Les algorithmes neuronaux sont les plus performants, en grande partie parce que ce problème est quasi LS. En effet, les dix ensembles d'apprentissage de 75 exemples que nous avons tirés au hasard se sont avérés être tous LS. Les perceptrons simples générés par Monoplane (avec l'étiquette Minimerror sur la figure 3) généralisent mieux que d'autres algorithmes nécessitant plus de ressources.

4 Conclusion

Les algorithmes neuronaux semblent mieux généraliser que d'autres sur les trois tâches de classification étudiées. Très probablement, la raison en est que les classes sont séparables avec un petit nombre d'hyperplans. Mais cela, il fallait encore le découvrir! L'heuristique de Monoplane s'est avérée très puissante. Elle consiste à construire une machine de parité, qui dispose d'un grand choix de représentations internes (RI), ce qui permet de séparer les classes avec peu d'unités cachées. Les RI ainsi engendrées n'épuisent pas l'espace des états cachés, et sont souvent linéairement séparables sans besoin d'augmenter leur dimension. La performance de Minimerror permet d'apprendre ces RI très efficacement. Ce qui explique qu'à nombre égal d'exemples, Monoplane généralise mieux que les autres algorithmes, et nécessite moins de ressources.

Table 1. **Algorithme Monoplane**

N : nombre de neurones d'entrée (binaires ou réelles), $\left\{\left(\vec{\xi}^{\mu}, \tau_0^{\mu}\right); 1 \le \mu \le P\right\}$ *: ensemble d'apprentissage*

Begin

H ← 0; $\varepsilon_t \leftarrow P$; **For** μ=1 **to** *P* **do** $y_{H+1}^{\mu} \leftarrow \tau_0^{\mu}$

While (ε_t > 0)

H ← **Couche_cachée**(*N*) ; $\varepsilon_t \leftarrow$ **Sortie**(H)

End

End

Couche_cachée(*N***)**

Begin

h ← H; *nombre d'unités cachées*

Do

$\varepsilon_t \leftarrow P$; h ← h+1; *nouvelle unité cachée*

$\mathcal{L} = \left\{\left(\vec{\xi}^{\mu}, y_h^{\mu}\right); 1 \le \mu \le P\right\}$; *ensemble d'apprentissage*

ε_t , $\left\{z_h^{\mu}; 1 \le \mu \le P\right\} \leftarrow$ **Minimerror**($\mathcal{L}$,*P* ,*N*);

For μ=1 **to** *P* **do** $y_{h+1}^{\mu} \leftarrow y_h^{\mu} z_h^{\mu}$;

Until (ε_t = 0)

Return h

End Couche_cachée

Sortie(H)

Begin

$\mathrm{L} = \left\{\left(\vec{z}^{\mu}, \tau_0^{\mu}\right); 1 \le \mu \le P\right\}$; *où*

$\vec{z}^{\mu} = \left(z_1^{\mu}, \ldots, z_h^{\mu}, \ldots z_H^{\mu}\right)$

ε_t , $\left\{\varsigma^{\mu}; 1 \le \mu \le P\right\} \leftarrow$ **Minimerror**($\mathcal{L}$,*P* ,H);

$y_{H+1}^{\mu} \leftarrow \tau_0^{\mu} \varsigma^{\mu}$;

Return ε_t

End Sortie

Minimerror($\mathcal{L}$, *P* ,*n***)** *P=nombre d'exemples, n=nombre d'entrées*

Begin

Minimiser par gradient simple et recuit déterministe le coût (1)

Return ε_t , $\left\{\varsigma^{\mu}; 1 \le \mu \le P\right\}$ ε_t *=nombre d'erreurs d'apprentissage,* $\left\{\varsigma^{\mu}; 1 \le \mu \le P\right\}$ *: sorties apprises*

End Minimerror

Remerciements. Nous remercions M. O. Gascuel, qui nous a facilité la référence SYMENU. J.M. Torres Moreno remercie CONACYT et UAM-A (México) pour leur soutien (Bourse 65659).

5 Bibliographie

Y. Le Cun, J. S. Denker, S. A. Solla. 1989. Optimal brain damage. *NIPS 2*, pp. 598-605.
Hassibi B., Stork D. 1993. Second order derivatives for network pruning: Optimal Brain Surgeon. *NIPS 5*, pp. 164-171.
Geman S., Bienenstock E., Doursat R. 1992. Neural networks and the bias/variance dilemma. *Neural Comp 4*. 1-58.
Torres-Moreno J.M., Peretto P., Gordon M. B. 1995. An evolutive architecture coupled with optimal perceptron learning for classification. *ESANN'95*, pp.365-370, et références citées.
Gordon M. B. 1996. A convergence theorem for incremental learning with real-valued inputs. *Proceedings ICNN'96 à paraître*.
Gordon M. B., Berchier D. 1993. Minimerror: A Perceptron Learning Rule that Finds the Optimal Weights, *ESANN'93*, pp.105-110.
Raffin B., Gordon, M.B. 1995. Learning and generalization with Minimerror, a temperature dependent learning algorithm. *Neural Comp 7*, 1206-1224.
Murphy P.M., D. W. Aha: UCI Repository of machine learning data bases [http://www.ics.uci.edu//~mlearn/MLRepository.html].
Thrun S. B., et al 1991. The MONK's Problems, a performance comparison of different learning algorithms. *CMU-CS-91-197 Carnegie Mellon University*.
Amari S., Murata N. 1993. Statistical theory of learning under entropic loss criterion. *Neural Comp 5*, 140-153.
Deffuant G. 1995. An algorithm for building regularized piecewise linear discrimination surfaces: the perceptron membrane. *Neural Comp 7*, 380-398.
Gascuel O. *et al*. (SYMENU group). 1995. Supervised classification. A comparison of twelve numerical, symbolic and hybrid methods. Preprint.
Wolberg W. H., Mangasarian O. L 1990. Multisurface method of pattern separation for medical diagnosis applied to breast citology. *Proc. of the National Academy of Sciences 87*, 9193-9196.
Prechelt L. 1994. Proben1 - A set of neural network benchmark problem and benchmarking rules. *Technical Report 21/94 Fakultät für Informatik, Universität Karlsruhe*, Germany.
Zhang J. 1992. Selecting typical instances in instance-based learning. *Proc. of the Ninth International Machine Learning Conference*. *Aberdeen Scotland* pp. 470-479.
Depenau J. 1995. A Global-Local learning algorithm. *Proc. of the World Congress on Neural Networks, Washington*. Vol.1, pp. 587-590.