Application de l'algorithme incrémental Monoplan à deux problèmes de classification

Juan 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
Manuel.Torres@cea.fr, Mirta.Gordon@cea.fr

1 Introduction

Un réseau de neurones (RN) avec une seule couche cachée peut approcher toute fonction des entrées, mais le nombre d'unités cachées nécessaires est inconnu. Les bornes fournies par la dimension VC, de même que le nombre de données d'après la théorie PAC, sont excessifs, inutilisables pour les applications. Hormis le cas du perceptron, notre compréhension du problème de la généralisation est encore insuffisante, et malgré le développement récent de techniques neuronales d'apprentissage, il existe un important décalage entre les prédictions théoriques et les performances des algorithmes. Certains algorithmes, comme la Retropropagation du Gradient (BP), introduisent le nombre et la connectivité des unités cachées *a priori*, et déterminent les poids par minimisation d'un coût. Le réseau obtenu est éventuellement élagué, ce qui, en termes d'inférence non paramétrique[1], permet de diminuer la variance. Avec une approche incrémentale on apprend au même temps le nombre d'unités et les poids. Commençant avec une seule unité cachée, on réduit le biais par l'introduction successive de neurones cachés, afin de produire des représentations internes (RI) fidèles. Ce procédé pourrait engendrer des réseaux avec un nombre excessif de neurones (surapprentissage) produisant de mauvaises généralisations. Dans ce travail nous présentons les résultats obtenus avec l'algorithme incrémental Monoplan sur deux problèmes connus, et nous les comparons, sur la base des erreurs de généralisation, aux meilleurs résultats trouvés dans la littérature. Nous constatons que les meilleures généralisations sont produites par des RN. Les algorithmes incrémentaux ne présentent pas de surapprentissage ; ils engendrent souvent des réseaux plus petits que ceux nécessaires aux algorithmes non-incrémentaux (après élagage) pour atteindre les mêmes performances. Monoplan 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 incrémental Monoplan

Nous voulons construire un RN avec une seule couche cachée connectée aux entrées, et un neurone de sortie connecté aux unités cachées. Pendant l'apprentissage, le nombre de neurones de la couche cachée grandit jusqu'à ce que le neurone de sortie classe correctement tout l'ensemble d'apprentissage. En fin d'apprentissage, les H unités cachées ont des poids J_{ki} ($0 \le i \le N$, $1 \le k \le H$) et la sortie a des poids W_k. N est le nombre de neurones d'entrée, P le nombre d'exemples, $\vec{\xi}^{\mu} = \left(1, \xi_1^{\mu}, \ldots, \xi_N^{\mu}\right)$ ($\mu=1,2,\ldots,P$) les états d'entrée, (le neurone 0 représente le seuil), $\tau^{\mu}=\pm 1$ sont les cibles à apprendre. Quand on présente une entrée $\vec{\xi}$, les états $\vec{\sigma} = (1, \sigma_1, \sigma_2, \ldots, \sigma_{NH})$ des unités cachées, et la sortie du réseau ζ, sont donnés respectivement par :

* Centre National de la Recherche Scientifique (CNRS)

$$\sigma_k = \text{signe}\left(\sum_{i=0}^{N} J_{ki}\,\xi_i\right);\ 1 \le k \le H \quad \text{et} \quad \zeta = \text{signe}\left(\sum_{k=0}^{H} W_k\,\sigma_k\right)$$

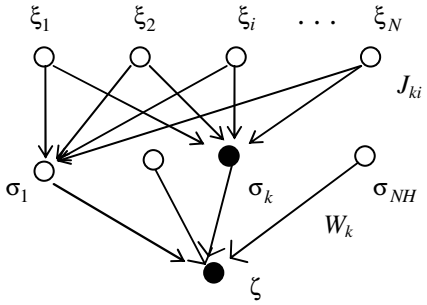

Figure 1 Réseau à une couche cachée (Les neurones seuils sont omis)

Nous avons développé un algorithme[2] qui se propose de bâtir un RN du type machine à parité : la sortie est la parité de la RI associée à l'entrée. Notre algorithme apprend la séparation des RI, en rajoutant des neurones dans la seule couche cachée si besoin est. Pour construire la couche cachée, Monoplan[2] commence par apprendre les sorties voulues à un perceptron. Si le problème n'est pas linéairement séparable (LS) il y aura des erreurs d'apprentissage : on gèle alors les poids du perceptron, qui sera le premier neurone de la couche cachée. On ajoute une deuxième unité cachée, ses poids sont appris pour donner une sortie $\tau_2^\mu = 1$ aux exemples bien appris par le neurone précédent, et $\tau_2^\mu = -1$ aux autres. La procédure est répétée tant qu'il reste des erreurs. Les RI ainsi engendrées sont fidèles[13]: des exemples de classes différentes ont des RI distinctes. Des théorèmes de convergence[3] existent tant pour des entrées binaires que réelles. Une fois tous les exemples appris, on connecte le neurone de sortie aux unités cachées. Il doit apprendre l'ensemble $\{\vec{\sigma}^\mu, \tau^\mu\}$. Si les RI sont LS, la procédure se termine. Autrement, il faut augmenter la dimension des RI afin de les rendre séparables. Pour cela on rajoute une unité cachée et on lui apprend $\tau^\mu = 1$ pour les exemples correctement classés par le perceptron de sortie, et $\tau^\mu = -1$ pour les autres. Éventuellement, on continue à rajouter des neurones jusqu'à ce que le neurone de sortie puisse séparer tous les exemples. Monoplan construit donc un réseau à une seule couche cachée (figure 1), et 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[4], 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 $\vec{\mathbf{w}} = (w_0, w_1, \ldots, w_N)$; $\gamma^\mu = \tau^\mu \vec{\mathbf{w}} \cdot \vec{\xi}^\mu / \sqrt{\vec{\mathbf{w}} \cdot \vec{\mathbf{w}}}$ est la stabilité ou marge des entrées. 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[4] et vérifié numériquement[5] que si l'ensemble d'apprentissage est LS, Minimerror permet d'obtenir une probabilité de généralisation maximale. Autrement, il minimise le nombre d'erreurs d'apprentissage.

3 Résultats

Nous présentons nos résultats, ainsi que ceux de différents auteurs, sur deux bases d'apprentissage souvent utilisées comme étalon pour l'étude des performances des algorithmes d'apprentissage[10]: le problème des formes d'ondes de Breiman et le diagnostic du cancer du sein (données de l'Hôpital de l'Université de Wisconsin). Afin que les

comparaisons ne soient pas biaisées, nous avons réalisé nos tests dans les mêmes conditions que les autres auteurs. Les résultats sont présentés sous forme graphique. Les différents algorithmes, triés par leurs erreurs de généralisation ε_g, 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.

3.1 Formes d'ondes de Breiman[14]

Ce problème a été introduit pour vérifier la méthode *CART* (*arbres de classification et de régression*). Il faut apprendre à classer des ondes $\vec{x}(t)$, synthétisées à partir de trois ondes de base, $h_1(t)$, $h_2(t)$ et $h_3(t)$, montrées sur la figure 2. Chaque classe consiste en une combinaison convexe aléatoire de deux de ces ondes de base, échantillonnées sur les entiers t=1,...,21.

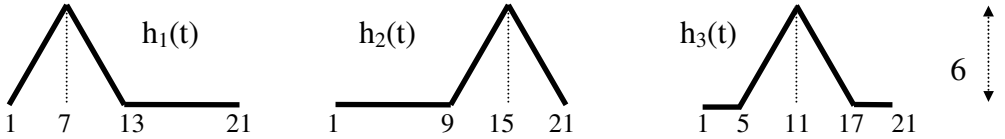

Figure 2. Formes d'onde de base

Les classes sont définies par:

Classe 1: $x_t = u h_1(t) + (1-u) h_2(t) + \varepsilon_t$, t=1,...,21
Classe 2: $x_t = u h_1(t) + (1-u) h_3(t) + \varepsilon_t$, t=1,...,21
Classe 3: $x_t = u h_2(t) + (1-u) h_3(t) + \varepsilon_t$, t=1,...,21

où $0 \le u \le 1$ est une variable aléatoire de distribution uniforme et ε_t est un bruit gaussien de distribution N(0,1). Le problème n'est donc pas déterministe. Il en résulte une borne supérieure à la probabilité de reconnaissance des formes d'onde, qui a été estimée à 86%[14]. Puisque Monoplan fait des séparations binaires, nous avons divisé le problème en trois sous problèmes : nous avons construit trois réseaux, chacun dédié à la séparation d'une classe par rapport aux deux autres. L'ensemble d'apprentissage a été le même pour les trois réseaux. La classe attribuée à chaque exemple est celle dont la sortie a la plus grande somme ponderée (*Winner-Take-All*) parmi les sorties des 3 réseaux. Nous avons réalisé nos tests dans les mêmes conditions que le groupe SYMENU[8] : les bases d'apprentissage contiennent 300 exemples et la base de test 5000, dont 33% de chaque classe. Les barres d'erreur sur la figure 3 correspondent à des moyennes de ε_g obtenus avec dix ensembles d'apprentissage distincts. L'excellente performance des algorithmes neuronaux a été attribuée à la quasi-séparabilité linéaire du problème. Le faible ε_g obtenu par un perceptron simple avec Minimerror corrobore cette conclusion.

3.2 Diagnostic du cancer du sein[6]

Depuis 1988, l'Université de Winsconsin alimente une base avec des données médicales concernant des prélèvements cytologiques[9] (épaisseur, uniformité de la taille et de la forme des cellules, etc.) avec leurs diagnostics : *bénin* ou *malin*. Chaque cas est décrit par N=9 attributs qui prennent des valeurs entre 1 et 10. La base contient 699 cas, dont nous avons enlevé 16 qui ont des attributs manquants. Les 683 exemples restants correspondent à 65.5% de cas *bénins* et 34.5% *malins.* La figure 4 présente l'erreur de généralisation ε_g de différents algorithmes[9,10,11,12] ayant appris avec des ensembles d'apprentissage de tailles variables. Les RN sont les plus performants, en grande partie parce que ce problème est séparable avec un faible nombre d'hyperplans. Minimerror a trouvé que dix ensembles d'apprentissage de 75 exemples tirés au hasard sont tous LS. Sur la figure 4, on voit que, à nombre d'exemples donné, Monoplan généralise mieux que les autres algorithmes et nécessite moins de ressources (nombre de poids).

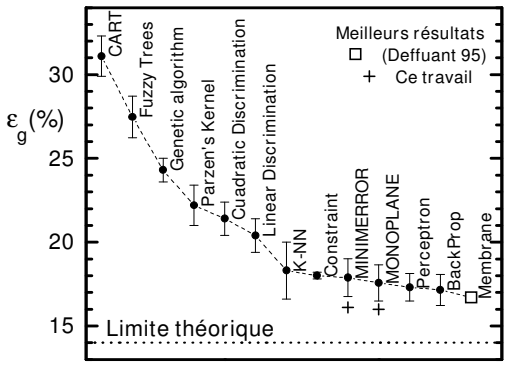

Figure 3. Classification des formes d'ondes

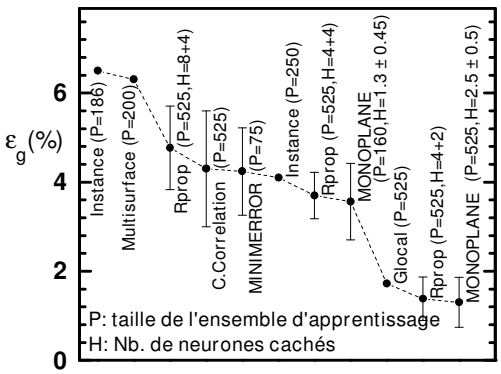

Figure 4. Diagnostic du cancer du sein

4 Conclusion

Les algorithmes neuronaux semblent mieux généraliser que d'autres sur les deux tâches de classification étudiées, très probablement parce qu'ils découvrent efficacement que les classes sont séparables avec un petit nombre d'hyperplans. L'heuristique de Monoplan s'est avérée très puissante. Elle consiste à construire une machine de parité, qui dispose d'un grand choix de représentations internes (RI), qui permettent 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 bonne performance de Minimerror est à la base du succès de Monoplan.

5 Références

[1] S. Geman, E. Bienenstock and R. Doursat: Neural networks and the bias/variance dilemma. Neural Comp 4, 1-58 (1992).

[2] J.M. Torres-Moreno, P. Peretto and M. B. Gordon: An evolutive architecture coupled with optimal perceptron learning for classification. in: ESANN'95, M. Verleysen ed. pp.365-370, (1995).

[3] M. B. Gordon. A convergence theorem for incremental learning with real-valued inputs. Proceedings ICNN'96 à paraître. (1996).

[4] M. B. Gordon and D. Berchier: Minimerror: A Perceptron Learning Rule that Finds the Optimal Weights, in: ESANN'93, M. Verleysen ed. pp.105-110, (1993).

[5] B. Raffin and M. B. Gordon: Learning and generalization with Minimerror, a temperature dependent learning algorithm. Neural Comp 7, 1206-1224. (1995)

[6] P.M. Murphy & D.W. Aha. Données de http://www.ics.uci.edu//~mlearn/MLRepository.html

[7] G. Deffuant: An algorithm for building regularized piecewise linear discrimination surfaces: the perceptron membrane. Neural Comp 7, 380-398 (1995).

[8] SYMENU (article collectif, O. Gascuel coordonateur des travaux) : Méthodes symbolique-numériques de discrimination, 5èmes Journées Nationales du PRC-IA (Nancy), Teknea, 29-76, 1995.

[9] W. H. Wolberg and O. L. Mangasarian: Multisurface method of pattern separation for medical diagnosis applied to breast citology. Proc.of the National Academy of Sciences 87, 9193-9196 (1990).

[10] L. Prechelt: Proben1 - A set of neural network benchmark problem and benchmarking rules. Technical Report 21/94 Fakultät für Informatik, Universität Karlsruhe, Germany (1994).

[11] J. Zhang: Selecting typical instances in instance-based learning. Proc. of the Ninth International Machine Learning Conference. Aberdeen Scotland: Morgan Kaufman, pp. 470-479 (1992).

[12] J. Depenau: A Global-Local learning algorithm. Proceedings of the World Congress on Neural Networks, Washington. Vol.1, pp. 587-590 (1995).

[13] D. Martinez and D. Estève: The Offset Algorithm: Building and Learning Method for Multilayer Neural Networks. Europhys. Lett., 18, 95-100 (1992)

[14] L. Breiman, H. J. Friedman, R. Olsen and C. Stone: Classification and Regression Trees. Wadsworth Inc., California (1984).