Rapport Technique PSI

De l'utilisation d'OBD pour la sélection de variables

Philippe Leray[(1)], Patrick Gallinari[(2)]

(1) INSA Rouen
ASI / PSI
BP 08 - Av. de l'Université
76801 St-Etienne du Rouvray Cedex

Philippe.Leray@insa-rouen.fr

(2) Université Paris 6
LIP6
8 rue du Capitaine Scott
75015 Paris – France

Patrick.Gallinari@lip6.fr

RESUME : La sélection de variables est un problème difficile à résoudre. Comment choisir l'ensemble des variables pertinentes pour résoudre une tâche fixée ? La sélection de variables neuronale essaye de résoudre le problème pendant l'apprentissage du réseau de neurones. Parmi les méthodes utilisées avec les réseaux de neurones de type perceptron multi-couches, certaines sont issues d'une technique d'élagage des poids, OBD (Optimal Brain Damage), proposée par LeCun et al. en 1990. Après avoir rappelé ces différentes méthodes, cet article montre comment essayer de les améliorer en suivant quelques principes simples. Une étude comparative situera ces différentes méthodes par rapport à d'autres techniques statistiques ou neuronales.

ABSTRACT: Feature Selection is a complex problem. How to determine the set of relevant features according to a fixed task ? Neural Feature Selection try to solve the problem during the neural network learning. Some frequently used methods are derived from a pruning technique OBD proposed in 1990 by LeCun and al. This article review these methods and propose some enhancements by using some simple rules. A study will then compare the previous methods and other classical ones.

MOTS-CLES : Perceptron Multi-couches, Elagage

KEYWORDS : Multilayer Perceptron, Pruning

2ème soumission à *Revue d'Intelligence Artificielle*, le 09 octobre 2000

1 Introduction

Les méthodes de sélection de variables sont composées généralement de trois composantes : un critère d'évaluation des variables, une procédure de recherche pour explorer l'espace des différentes combinaisons de variables et un critère d'arrêt.
Nous allons aborder ici les méthodes de sélection de variables neuronales dont le critère d'évaluation est dérivé de la technique d'élagage des poids OBD (Optimal Brain Damage) proposée par [LEC 90]. Dans une étude précédente [LER 97, 99], une comparaison des différentes composantes des méthodes de sélection de variables nous a permis de passer en revue les techniques de sélection de variables dérivées de OBD et de proposer quelques améliorations à ces méthodes.

Après avoir fait quelques rappels à propos des différents critères mis en œuvre lors de la sélection de variables neuronale, nous présenterons les méthodes dérivées de OBD ainsi que les variantes proposées dans la littérature. Nous proposerons aussi plusieurs améliorations à ces méthodes. Dans le cadre de la classification, une étude comparative sur plusieurs problèmes artificiels ou réels situera ces diverses méthodes par rapport à d'autres techniques classiques de sélection de variables neuronales. Pour finir, quelques remarques et perspectives concluront cette étude.

2 Sélection de variables et réseaux de neurones

La détermination des variables importantes est un problème essentiel dans l'identification de modèles. De nombreuses publications essayent de procéder à un état de l'art des différentes méthodes utilisées [LER 99] [ZAP 99].
Il est pratique de voir les techniques de sélection de variables sous trois angles différents, trois composantes qu'il est possible de régler séparément :

- un critère d'évaluation des variables, pour comparer différents sous-ensembles de variables et en retenir un,
- une procédure de recherche, pour explorer l'espace des différentes combinaisons de variables,
- un critère d'arrêt, pour stopper la procédure de recherche ou déterminer l'ensemble de variables à sélectionner.

Nous allons nous intéresser à ces trois critères pour les méthodes de sélection de variables neuronales.

2.1 Critère d'évaluation (mesure de pertinence)

Les mesures de pertinence associées aux méthodes de sélection de variables neuronales sont souvent basées sur des heuristiques calculant l'importance individuelle de chaque variable dans le modèle obtenu après apprentissage. Ces heuristiques sont nombreuses, mais peuvent être classées selon leurs similarités en quatre grandes familles :

- Les mesures d'ordre zéro (i.e. utilisant les valeurs des paramètres du réseau),
- Les mesures du premier ordre (i.e. utilisant les dérivées du premier ordre des paramètres du réseau),

- Les mesures du second ordre (i.e. utilisant les dérivées du second ordre des paramètres du réseau).
- Les termes de régularisation permettant de pénaliser les variables inutiles pendant l'apprentissage.

Seules les mesures du second ordre nous intéresserons dans ce document. Le lecteur pourra se référer à [LER 99] pour une revue des trois premières mesures ou à [GRA 98] pour le lien entre régularisation et sélection de variables.

2.2 Procédure de recherche

En général, le critère d'évaluation utilisé pour la sélection de variables n'est pas monotone. Il faudrait donc examiner l'ensemble des $2^k - 1$ sous-ensembles possibles de k variables. Cette solution, combinatoire par rapport au nombre de sous-ensembles, est inapplicable pour des valeurs, même modérées, de k.

Les procédures de recherche couramment utilisées sont donc très souvent des heuristiques basées sur des parcours séquentiels de recherche (cf. e.g. [KIT 86]) *forward* ou *backward*. D'autres méthodes plus complexes et pour la plupart sous-optimales sont issues de techniques de parcours de graphes.
Une grande partie des méthodes de sélection de variables neuronales sont des méthodes *backward*, où les variables sont éliminées grâce à des considérations sur les paramètres du réseau et/ou sur les données. Il existe aussi des méthodes de construction incrémentales de réseaux de neurones qui peuvent être considérées comme des méthodes de sélection *forward* où des neurones sont itérativement ajoutés en entrée [MOO 94] [GOU 97].

Un problème se pose lorsque sont utilisés conjointement une évaluation individuelle des variables et un parcours séquentiel : les dépendances entre les variables ne sont pas prises en compte explicitement. De plus, à cause de la non-linéarité des RN, la corrélation entre variables n'est plus un indicateur satisfaisant de leur dépendance.
Certaines méthodes de sélection de variables ignorent simplement ce problème, d'autres proposent d'éliminer une variable à la fois et de réapprendre ensuite le nouveau réseau ainsi obtenu avant d'évaluer l'importance des variables restantes. Cette solution permet de tenir compte des dépendances entre variables que le réseau aura découvert grâce au ré-apprentissage. Le problème de l'initialisation des poids se pose aussi au moment du ré-apprentissage : faut-il partir des poids obtenus avec le réseau précédent ou les ré-initialiser (à zéro ou aléatoirement) ? Ce problème reste ouvert, mais il semble judicieux de ré-initialiser les poids aléatoirement à chaque étape pour obtenir de meilleures performances, mais avec un apprentissage souvent plus long.

2.3 Critère d'arrêt

Une fois que la méthode d'évaluation et celle de recherche ont été fixées, certaines méthodes de sélection de variables examinent tous les sous-ensembles fournis par la méthode de recherche.
Une bonne heuristique, dont la complexité est suffisamment raisonnable dans la plupart des applications, est d'estimer l'erreur en généralisation pour les différents sous-

ensembles de variables sélectionnés. L'ensemble de variables idéal est celui qui donne les meilleures performances.
L'erreur de généralisation peut être estimée grâce à un ensemble de validation, par validation croisée ou par d'autres estimations algébriques comme FPE (Final Prediction Error, [AK 70]). Plusieurs mesures ont été proposées en statistiques [GUS 95] ou pour les réseaux de neurones |MOO 91], [LAR 94].

La plupart des méthodes de sélection de variables utilisent des techniques assez rudimentaires pour arrêter la sélection : certaines méthodes fixent un seuil par rapport au critère de pertinence, d'autres classent juste les variables en fonction de l'estimation de l'erreur en généralisation. [VAN 97] propose un autre critère d'arrêt basé sur un test statistique. Il effectue l'équivalent d'un test de Student sur les erreurs quadratiques (en régression) ou les taux d'erreur (en classification) de deux modèles pour décider si leurs moyennes sont statistiquement égales. Pour cela, il est nécessaire d'effectuer la sélection de variables jusqu'au bout (i.e. éliminer ou sélectionner les k variables) et choisir le plus petit modèle obtenu avec une erreur statistiquement proche de l'erreur minimale. Nous proposons un critère d'arrêt similaire en §3.3.

3 Méthodes du second ordre

Plusieurs méthodes de sélection de variables sont inspirées des techniques d'élagage des poids dans le réseau. La décision de supprimer un poids est faite selon un critère de pertinence. Une connexion est coupée si sa pertinence est faible.
Après avoir présenté une technique d'élagage précise (Optimal Brain Damage), nous passerons en revue les différentes méthodes de sélection de variable qui en ont découlé. Nous proposerons aussi des variantes à ces méthodes à partir de considérations générales issues de [LER 97] et [LER 99].

3.1 L'élagage des poids par Optimal Brain Damage (OBD)

[LEC 90] a proposé une technique d'élagage appelée OBD, où il définit la pertinence d'une connexion par :

$$Pertinence\ (w_j) = \frac{1}{2} H_{jj} w_j^2 = \frac{1}{2} \frac{\partial^2 MSE}{\partial w_j^2} w_j^2 \qquad [1]$$

où le Hessien de la fonction de coût est utilisé pour calculer la dépendance du modèle par rapport aux poids.

Pour utiliser cette mesure de pertinence d'une connexion comme critère de sélection d'une variable, il faut calculer la pertinence d'un neurone de la couche d'entrée en utilisant l'approximation suivante :

$$Pertinence\ (x_i) = \sum_{j \in fan-out(i)} Pertinence\ (w_j) \qquad [2]$$

où *fan-out(i)* représente l'ensemble des poids partant de la variable *i*.

3.2 La sélection de variables par Optimal Cell Damage (OCD)

OCD a été proposé dans [CIB 94, 96], une méthode équivalente étant proposée au même moment dans [MAO 94]. Cette méthode généralise la technique d'élagage OBD à la sélection de variables.

En utilisant [1] et [2], nous obtenons :

$$S_i = Pertinence\,(x_i) = \sum_{j \in fan-out(i)} \frac{1}{2} \frac{\partial^2 MSE}{\partial w_j^2} . w_j^2 \qquad [3]$$

OBD et OCD considèrent que H, le Hessien de la fonction de coût (MSE) est une matrice diagonale (i.e. les termes croisés d'un développement de Taylor du second ordre sont négligés). Cette hypothèse revient à supposer que la fonction de coût est minimale et localement quadratique autour du minimum local. Dans le cas d'un PMC, le Hessien « diagonal » peut alors être estimé en O(N). Dans le cas des réseaux récurrents, le calcul est plus difficile [PED 95].

Cibas propose l'algorithme de sélection de variable *backward* basé sur [3] :

Algorithme OCD
0. Atteindre un minimum local (fixé par un seuil θ)
1. Calculer la pertinence de chaque entrée grâce à [3]
2. Trier les entrées par ordre croissant de pertinence
   Soit S_i la liste des pertinences classées par ordre croissant, on peut définir
   La pertinence cumulée par :

$$S_i' = \sum_{j=1}^{i} S_i$$

3. Supprimer les entrées dont la pertinence cumulée est inférieure à un seuil fixé q
4. Recommencer en 0 tant que les performances estimées sur une base de test ne chutent pas.

Les seuils θ et q sont déterminés par validation croisée.

3.3 N-OCD, notre variante d'OCD

Suite aux travaux de [CIB 96], nous avons proposé une variante de OCD (N-OCD) qui distingue bien les critères mis en œuvre dans la sélection de variable en essayant d'améliorer l'évaluation de la mesure de pertinence et celle du critère d'arrêt.

Évaluation de la mesure de pertinence

OCD donne de bons résultats mais possède aussi un inconvénient: si le seuil q est trop élevé, l'algorithme peut supprimer des variables significatives. De même, si le nombre de variables de pertinence faible est élevé, cela ne signifie pas que toutes ces variables sont inutiles et à éliminer. C'est la raison pour laquelle nous proposons une autre version

d'OCD où nous n'utilisons pas le seuil q. Nous supprimons les variables une par une en ré-apprenant le RN et en ré-estimant les mesures de pertinence à chaque fois.

Critère d'arrêt

Un autre problème se pose alors : quel critère d'arrêt utiliser ? L'estimation de l'erreur en généralisation avec un ensemble de test fournit un critère d'arrêt non monotone, qui oscille au fur et à mesure de la sélection. Il est assez brutal de s'arrêter dès que les performances en test diminuent.
Notre variante de OCD va donc supprimer toutes les variables jusqu'à la dernière et déterminer ensuite quel sous-ensemble de variables sélectionner grâce à un test statistique.

Algorithme N-OCD
0. Pour p = k jusqu'à 1 (nombre de variables)
1. Atteindre un minimum local (déterminé grâce à un ensemble de validation)
2. Estimer l'erreur *MSE(p)* (sur un ensemble de test)
3. Calculer la pertinence de chaque entrée grâce à [3]
4. Supprimer la variable la moins pertinente
5. Retourner en 0.

6. $M° = min(MSE(p))$
7. $\{p_i\} = \{ p / MSE(p) \approx M° \text{ au sens de Fisher } \}$
8. $p_0^* = min\{p_i\}$

Supposons que nous sommes dans un problème de régression à k variables, grâce à N-OCD nous obtenons un ensemble de k modèles ayant de moins en moins de variables. Notons $F_{(p)}$ (de $p=1$ à k variables) ces réseaux et $MSE(p)$ l'erreur estimée sur des données de test.
La solution classique est de sélectionner le réseau $F(p_o)$ qui obtient la plus petite erreur. Malheureusement, le nombre de données servant à estimer l'erreur est limité, il existe donc une incertitude sur l'estimation du critère de choix.
Pour tenir compte de ce problème, nous allons chercher parmi tous nos modèles $F_{(p)}$ ceux qui sont statistiquement proches de $F(p_o)$ à l'aide d'un test de Fisher (cf. [SAP 90]). Nous obtenons ainsi un ensemble de modèles tels que $MSE(p_i) \approx MSE(p_0)$. En posant comme hypothèse que nous cherchons le plus petit ensemble de variables possible, il suffit de prendre $p_0^* = min\{p_i\}$, i.e. le plus petit modèle statistiquement proche du modèle obtenant une erreur en test minimale.[1]
Cette méthode, testée sur les mêmes problèmes que OCD, permet ainsi d'obtenir à chaque fois un modèle ayant de bonnes performances avec un nombre plus faible de variables.

[1] L'évaluation des performances en généralisation et le choix du meilleur ensemble de variables ne devraient théoriquement pas être effectués sur le même ensemble de test.

3.4 D'autres variantes

Revenons à l'hypothèse de diagonalité du Hessien faite par OBD et OCD : elle permet de calculer le Hessien rapidement mais au prix d'approximations fortes. [HAS 93] propose une autre technique d'élagage OBS qui calcule le Hessien en entier (calcul en $O(N^2)$).
Cette méthode, qui n'est pas envisageable pour un grand nombre de poids, a cependant un avantage : elle permet de mettre à jour immédiatement les poids du réseau lorsqu'une connexion est supprimée. De plus, [HAS 94] insistent sur le fait qu'un apprentissage utilisant OBD peut conduire à une baisse de performances en généralisation, ce qui n'est pas le cas d'OBS.
De la même manière qu'OCD utilisait le calcul de pertinence des poids donné par OBD, Unit-OBS proposé dans [STA 97] se sert du calcul de pertinence des poids donné par OBS pour supprimer des variables. L'avantage de cette méthode est qu'il n'est plus nécessaire de recalculer le Hessien à chaque élimination d'un poids, mais à chaque suppression d'une variable.

Quelque soit la méthode utilisée (OBD ou OBS), la pertinence d'une connexion (ou d'une variable) est calculée à partir des mêmes données que celles utilisées pour l'apprentissage. [PED 96] proposent deux nouvelles méthodes d'élagage γOBD et γOBS qui calculent la pertinence en fonction d'une approximation de l'erreur en généralisation obtenue grâce au critère FPE "Final Prediction Error" [AKA 70]. Comme OBD et OBS, γOBD et γOBS pourraient aussi être transformées en techniques de sélection de variables.

3.5 Une autre variante : Early Cell Damage (N-ECD)

L'hypothèse de base de OBD et de OBS est que le réseau a atteint un minimum local. En pratique, l'apprentissage du réseau de neurones est arrêté par *Early Stopping*, avant que le minimum local ne soit atteint.
[TRE 97] propose donc deux nouvelles variantes de OBD et OBS : EBD (Early Brain Damage) et EBS (Early Brain Surgeon). À partir de considérations heuristiques, il ajoute deux nouveaux termes dans le calcul de la pertinence des poids pour prendre en compte le fait que la dérivée de la fonction de coût n'est pas nulle à la fin de l'apprentissage.

Nous proposons d'étendre cette méthode d'élagage pour obtenir une nouvelle mesure de pertinence donnée en [4], en continuant à utiliser notre test statistique comme critère d'arrêt. Soit N-ECD (ECD pour Early Cell Damage) la méthode ainsi obtenue.

$$S_i = Pertinence(x_i) = \sum_{j \in fan-out(i)} \frac{1}{2} \frac{\partial^2 MSE}{\partial w_j^2} . w_j^2 - \frac{\partial MSE}{\partial w_j} . w_j + \frac{1}{2} \frac{\left(\frac{\partial MSE}{\partial w_j} \right)^2}{\frac{\partial^2 MSE}{\partial w_j^2}} \qquad [4]$$

Algorithme N-ECD
0. Pour p = k jusqu'à 1 (nombre de variables)
1. Atteindre un minimum local (déterminé grâce à un ensemble de validation)
2. Estimer l'erreur *MSE(p)* (sur un ensemble de test)
3. Calculer la pertinence de chaque entrée grâce à [4]
4. Supprimer la variable la moins pertinente
5. Retourner en 0.

6. $M°=min(MSE(p))$
7. $\{p_i\} = \{ p / MSE(p) \approx M° \text{ au sens de Fisher } \}$
8. $p_0^* = min\{p_i\}$

Comparons N-OCD et N-ECD sur le problème des vagues de Breiman à 40 variables (décrit dans le paragraphe 4). La figure 1 montre que N-OCD et N-ECD ont un comportement assez proche : le taux d'erreur des modèles obtenus successivement par les deux méthodes stagne autour de 15% jusqu'au moment où il ne reste plus assez de variables pour résoudre correctement le problème. Par contre, le taux d'erreur des modèles obtenus par N-ECD est toujours inférieur à ceux de N-OCD.
Cette figure illustre aussi l'utilité du test statistique comme critère d'arrêt : il permet de trouver, sur cet exemple, le modèle le plus intéressant, avec le moins de variables possibles et de bonnes performances.

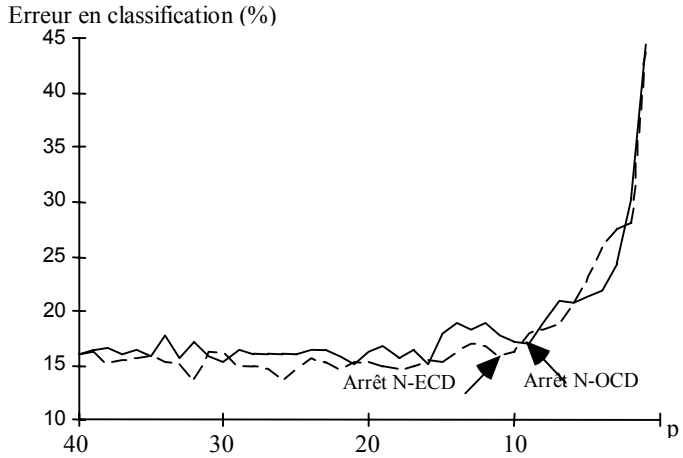

Fig. 1 : Comparaison des performances des réseaux obtenus progressivement par N-OCD (en trait plein) et N-ECD (en pointillé) sur le problème des vagues de Breiman à 40 variables. L'axe horizontal représente le nombre de variables restantes, l'axe vertical représente le taux d'erreur du modèle obtenu.

4 Etude comparative

Nous allons présenter les résultats obtenus avec des méthodes de sélection de variables couvrant la plupart des techniques présentées dans [LER 99] :

- Une méthode statistique (Stepdisc) basée sur une mesure de pouvoir discriminant,
- Une méthode issue de la théorie de l'information : MIFS [BON 94] [BAT 94] basée sur l'information mutuelle,
- Des méthodes neuronales de différents ordres :
  - Ordre 0 : HVS [YAC 97]
  - Ordre 1 : SBP [MOO 92], *Ruck* [RUC 90], *Dorizzi* [DOR 96], *Czernichow* [CZE 96]
  - Ordre 2 : N-OCD (§3.3), N-ECD (§3.5)

Il est difficile de comparer quantitativement les différentes méthodes de sélection de variables, il n'y a pas de mesure de pertinence idéale, et la précision de la sélection dépend fortement du type de parcours utilisé et du critère d'arrêt. Dans le cas des méthodes connexionnistes, il serait évidemment possible d'échanger chacun de ces critères d'une méthode à l'autre. Le tableau 1 essaye de récapituler les différentes caractéristiques des méthodes utilisées dans cette étude.

Méthode	**Parcours**	**Critère d'arrêt**
Stepdisc	Stepwise	Test statistique
MIFS (Bonnlander)	Forward	seuil (0.99)
HVS (Yacoub)	Backward	Variation des performances en test
SBP (Moody)	Backward (sans réapprentissage)	Variation des performances en test
(Ruck)	Backward (sans réapprentissage)	Seuil (moyenne des pertinences)
(Dorizzi)	Backward (sans réapprentissage)	Seuil (moyenne des pertinences)
(Czernichow)	Backward (sans réapprentissage)	Seuil (moyenne des pertinences)
N-OCD (Cibas, Leray)	Backward	Test statistique
N-ECD (Leray)	Backward	Test statistique

Table 1 : Récapitulatif des composantes (mesure de pertinence, type de parcours et critère d'arrêt) des différentes méthodes comparées dans cette étude**.**

Nous avons choisi d'appliquer ces méthodes à différents problèmes artificiels de classification permettant de mettre à jour leurs qualités ou défauts respectifs :

- comportement face à un problème non-linéaire (§4.1),
- comportement face à des variables inutiles (§4.2),
- problème du choix du critère d'arrêt (§4.3),
- comportement face à des données très corrélées (§4.4).

Nous présenterons ensuite les résultats obtenus par N-OCD pour un problème réel de diagnostic (§4.5).

Pour chaque problème, nous indiquons les performances en test d'un réseau de neurones de type perceptron multicouche (avec une seule couche cachée de 10 neurones), l'ensemble de variables sélectionnées par chacune des méthodes et les performances du réseau de neurones correspondant.
L'intervalle de confiance à α % est donné par la formule suivante, expliquée dans [BEN 92] avec N nombre d'exemples, T performance du classifieur et Z_α=1.96 pour α=95%.

$$I(\alpha, N) = \frac{T + \frac{Z_\alpha^2}{2N} \pm Z_\alpha \sqrt{\frac{T(1-T)}{N} + \frac{Z_\alpha^2}{4N^2}}}{1 + \frac{Z_\alpha^2}{N}}$$

4.1 Non-linéarité

Prenons un problème de classification simple dans R^{20} avec des variables non corrélées et une frontière de décision non linéaire. Pour cela, il suffit de prendre deux gaussiennes de matrice de variance/covariance respectives *4*I* et *I* (où *I* est la matrice identité dans R^{20}).
Pour que l'importance des variables soit progressive, nous avons choisi de placer les centres des gaussiennes de telle manière que le premier axe soit perpendiculaire à la frontière de décision et que les autres axes s'éloignent de plus en plus de cette perpendiculaire. Ainsi, nous avons pris μ_1*=(0,...,0)* et μ_2*=(0,1,2,...,19)/*α. α sert de critère de recouvrement des deux classes : s'il est très grand la première classe (dont la variance est la plus grande) recouvre complètement la seconde; s'il est proche de 0, les deux classes sont disjointes. Dans les deux cas, le problème de classification n'est pas intéressant. Nous avons alors fixé arbitrairement α pour avoir $||\mu_1\mu_2|| = 2$.
Pour nos diverses études, les bases d'apprentissage, de validation et de test ont respectivement 2500, 2500 et 5000 éléments.
Dans ce problème, l'importance des variables est progressive : x_1 est inutile, x_2 est moins utile que x_3, etc. Ainsi les dernières variables sont bien plus importantes que les premières, ce que l'on retrouve dans la table 2.
Cette table nous montre aussi que Stepdisc n'est pas adapté aux cas de frontières non linéaires, elle est la seule méthode à sélectionner x_1 qui est une variable inutile pour ce problème !
La figure 2 donne la répartition des méthodes de sélection de variable selon les performances des modèles obtenus (axe y) et le pourcentage de variables sélectionnées (axe x). Les meilleurs méthodes sont celles qui donnent le modèle le plus performant possible avec un nombre de variable le plus faible possible.

Grâce à cette figure, il est possible de noter plusieurs comportements caractéristiques des différentes méthodes. De manière générale, La méthode proposée par Bonnlander (MIFS) avec un parcours *forward* ne sélectionne pas assez de variables. De même, le critère d'arrêt proposé par Yacoub (HVS) est trop brutal et ne supprime pas assez de variables.
Comme nous avons défini le problème pour que les variables ne soient pas corrélées, le ré-apprentissage entre chaque suppression de variables n'est pas utile. Les méthodes qui ré-estiment la mesure de pertinence à chaque étape (Yacoub, Leray, Cibas) semblent légèrement pénalisées par rapport aux autres (Dorizzi, Ruck, Czernichow).

Méthode	**p***	**Variables sélectionnées**	**Performances**
Aucune	20	11111111111111111111	94.80 % [94.15 - 95.35]
Stepdisc	17	10001111111111111111	94.88 % [94.23 - 95.43]
(Bonnlander)	5	00010000000000011011	90.60 % [89.76 - 91.38]
(Yacoub)	18	01011111111111111111	94.86 % [94.21 - 95.44]
(Moody)	9	01000100011000110111	92.94 % [92.20 - 93.62]
(Ruck)	10	00000000101101111111	94.86 % [94.21 - 95.44]
(Dorizzi)	11	00000000101111111111	94.66 % [94.00 - 95.25]
(Czernichow)	9	00000000001101111111	94.02 % [93.33 - 94.02]
(Cibas)	14	01001110010111111111	94.62 % [93.96 - 95.21]
(Leray)	15	01011011101110111111	94.08 % [93.39 - 94.70]

Table 2 : Résultats comparatifs de plusieurs méthodes de sélection de variables pour un problème non linéaire de classification à deux classes.

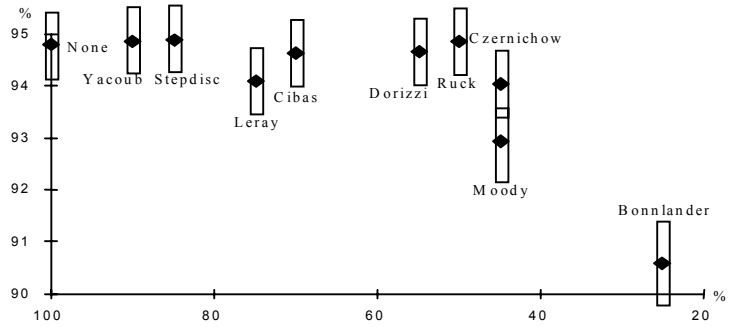

Figure 2 : Comparaison des performances (avec intervalle de confiance) de différentes méthodes de sélection de variables pour un problème de classification non linéaire à deux classes : pourcentage de variables sélectionnées (axe horizontal) vs. pourcentage de bonne classification (axe vertical).

4.2 Variable inutiles

Nous présentons dans le tableau 3 les résultats obtenus sur un problème de classification comportant un nombre important de variables inutiles.

Ce problème a été proposé par [BRE 84] et repris dans une variante bruitée par [DEB 91]. C'est un problème à 3 classes avec un ensemble de 21 variables de différents degrés de pertinence et 19 variables supplémentaires (bruit gaussien centré réduit) inutiles à la classification. Les performances du classifieur de Bayes (performances optimales) ont été estimées par Breiman à 86 % de bonne classification.

Pour ce problème bruité, toutes les méthodes éliminent les variables purement bruitées. Exception faite la méthode proposée par Cibas, toutes les autres donnent des résultats similaires par rapport aux performances (autour de 85%) et au nombre de variables sélectionnées (entre 11 et 14 pour Stepdisc, Bonnlander et Leray; et entre 16 et 18 pour Yacoub, Moody, Ruck, Dorizzi et Czernichow).
Stepdisc donne de bons résultats par rapport au problème précédent : ici les frontières sont presque linéaires et les données sont uni-modales.

Méthode	**p***	**Variables sélectionnées**	**Performances**
Aucune	40	11111111111111111111 (+19 vb)	82.51 % [81.35 - 83.62]
Stepdisc	14	00011011111111011100 (+0 vb)	85.35 % [84.26 - 86.38]
(Bonnlander)	12	00001110111111110000 (+0 vb)	85.12 % [84.02 - 86.15]
(Yacoub)	16	00011111111111111100 (+0 vb)	85.16 % [84.07 - 86.19]
(Moody)	16	00011111111111111100 (+0 vb)	85.19 % [84.10 - 86.22]
(Ruck) (Dorizzi)	18	01111111111111111100 (+0 vb)	85.51 % [84.43 - 86.53]
(Czernichow)	17	01011111111111111100 (+0 vb)	85.67 % [84.59 - 86.69]
(Cibas)	9	00000111110111000000 (+0 vb)	82.26 % [81.09 - 83.37]
(Leray)	11	00000111111111100000 (+0 vb)	84.56 % [83.45 - 85.61]

Table 3 : Résultats comparatifs de plusieurs méthodes de sélection de variables pour le problème des vagues de Breiman à 40 variables. Sont indiquées dans la colonne "Variables sélectionnées" celles retenues parmi les 21 variables informatives ainsi que le nombre de variables de bruit pur sélectionnées (vb).

4.3 Choix du critère d'arrêt

Reprenons le problème original des vagues de Breiman avec uniquement les 21 premières variables, plus ou moins utiles à la classification. Pour ce problème, non bruité, l'ordre des méthodes change : les méthodes utilisant un seuil comme critère d'arrêt ne sélectionnent plus assez de variables ou trouvent des modèles avec de moins bonnes performances.

La figure 3 permet une nouvelle fois de noter plusieurs comportement caractéristiques des différentes méthodes. Comme en §4.1, la méthode proposée par Bonnlander (MIFS) avec un parcours forward ne sélectionne pas assez de variables. De même, le critère d'arrêt proposé par Yacoub (HVS) est trop brutal et ne supprime pas assez de variables.

Cette figure montre cette fois-ci que notre méthode est satisfaisante en sélectionnant peu de variables avec de très bonnes performances. Notre critère d'arrêt basé sur un test statistique semble plus intéressant que les seuils fixés par l'utilisateur.

Méthode	**p***	**Variables sélectionnées**	**Performances**
Aucune	21	111111111111111111111	85.28 % [84.19 - 86.31]
Stepdisc	14	00111010111111011100	84.19 % [83.07 - 85.25]
(Bonnlander)	8	00000110011101010000	83.05 % [81.90 - 84.14]
(Yacoub)	18	01111111111111111100	85.46 % [84.38 - 86.48]
(Moody)	16	00011111111111111100	85.63 % [84.55 - 86.65]
(Ruck) (Dorizzi)	12	00011110111111010000	84.65 % [83.54 - 85.70]
(Czernichow)	10	00011010111111010000	82.58 % [81.42 - 83.68]
(Cibas)	15	00101111111111110100	85.23 % [84.14 - 86.26]
(Leray)	13	00001111111111110000	85.67 % [84.59 - 86.69]

Table 4 : Résultats comparatifs de différentes méthodes de sélection de variables pour le problème des vagues de Breiman avec 21 variables**.**

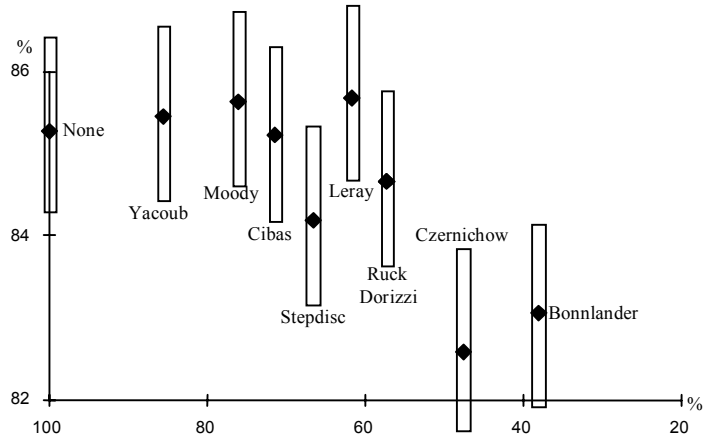

Figure 3 : Comparaison des performances (avec intervalle de confiance) de différentes méthodes de sélection de variables pour un problème original des vagues de Breiman : pourcentage de variables sélectionnées (axe horizontal) vs. pourcentage de bonne classification (axe vertical).

4.4 Variables Corrélées

Prenons maintenant le problème à deux classes utilisé en §4.1 mais en le modifiant pour obtenir quatre groupes de cinq variables successives corrélées.
Cette variante reprend les caractéristiques du problème précédent en remplaçant la matrice Identité utilisée pour Σ_1 et Σ_2 par une matrice diagonale par bloc (chaque bloc est de dimension 5x5). Ainsi le nouveau problème possède quatre groupes de cinq variables successives corrélées.

Méthode	**p***	**Variables sélectionnées**	**Performances**
Aucune	20	11111111111111111111	90.58 % [89.74 - 91.36]
Stepdisc	11	00001101011010110111	91.96 % [91.17 - 92.68]
(Bonnlander)	5	00001001010000100001	88.48 % [87.57 - 89.34]
(Ruck)	10	00011001011110100011	91.06 % [90.24 - 91.82]
(Leray)	7	00000010101010100011	90.72 % [89.88 - 91.49]

Table 4 : Résultats comparatifs de plusieurs méthodes de sélection de variables pour un problème non linéaire de classification à deux classes avec des variables corrélées**.**

La Table 4 donne les résultats de quelques méthodes représentatives pour ce problème. Stepdisc et la méthode de Ruck donnent de très bonnes performances, mais sélectionnent un grand nombre de variables corrélées.

Comme pour les autres exemples, la méthode de Bonnlander retient très peu de variables et obtient des performances légèrement plus faibles que les autres méthodes.
Notre méthode trouve un modèle possédant à la fois un faible nombre de variables (7 par rapport aux 10 et 11 de Ruck et Stepdisc) et de bonnes performances. En effet, le ré-apprentissage du réseau entre chaque sélection de variables permet de prendre en compte la corrélation entre variables, ce que ne fait pas la méthode de Ruck.

4.5 Problème réel de diagnostic

Dans [LER98], nous avons appliqué notre méthode de sélection de variable (N-OCD) à un problème de diagnostic dans le réseau téléphonique français. Pour cela, nous avions proposé une architecture modulaire dont le premier niveau devait traiter les données provenant de chaque centre de transit du réseau téléphonique afin de reconnaître 5 types de surcharges (dont 2 fortement semblables).
Pour chaque centre de transit, nous avons à notre disposition un ensemble de 36 variables (indicateurs de trafic, …). Certaines variables sont ou inutiles pour notre tâche, ou fortement corrélées.

Méthode	**p***	**Variables sélectionnées**	**Performances**
Aucune	36	111111111111111111111111111111111111	75.00 % [73.00 – 77.00]
Stepdisc	17	10010001011011001101010000000111111	73.60 % [71.60 – 75.60]
(Bonnlander)	9	00000000110010110101000000000000100	74.70 % [72.70 – 76.70]
(Leray)	8	00000100110110110000000000000000010	76.70 % [74.70 – 78.70]

Table 5 : Résultats comparatifs de plusieurs méthodes de sélection de variables pour un problème de reconnaissance de surcharges dans le réseau téléphonique.

La Table 5 nous donne les résultats de la sélection de variables (et d'un PMC appris avec ces variables) pour le problème de reconnaissance de surcharge pour un centre de transit donné, en utilisant la méthode statistique (stepdisc), la méthode basée sur l'information mutuelle et la méthode N-OCD.
Le classifieur neuronal construit à partir des variables sélectionnées par N-OCD obtient des performances légèrement supérieures au classifieur utilisant toutes les variables à un coût moindre (moins de paramètres dans le réseau,...). Il obtient aussi des résultats supérieurs aux autres méthodes puisqu'il a été construit pour obtenir de bonnes performances avec moins possible de variables (alors que les performances du classifieur n'entrent pas en compte pour les méthodes comme Stepdisc et celle de Bonnlander).

5 Conclusion

Les études comparatives précédentes portent sur différents problèmes de classification, même si la plupart des méthodes présentées ici (dont nos deux méthodes N-OCD et N-ECD) s'appliquent aussi dans le cadre de la régression.
Ces comparaisons permettent de tirer quelques conclusions sur les méthodes de sélection de variables. Tout d'abord, il n'existe pas de méthode de sélection qui soit meilleure que les autres. Les méthodes dérivées des techniques d'élagage comme OBD ne sont pas meilleures que les autres. Par contre, les résultats vont dépendre de la politique choisie par rapport aux différents critères utilisés :

- Les critères de pertinence basés sur des hypothèses de linéarité ou de distribution unimodale sont mal adaptés aux autres problèmes (§4.1).
- L'évaluation du critère de pertinence est liée aux paramètres du modèle. La suppression d'une variable dans le réseau de neurones change automatiquement la valeur de ses paramètres optimaux. Ne pas ré-estimer les paramètres du modèle signifie que l'on considère toutes les variables indépendantes. Un réapprentissage est nécessaire si l'on désire prendre en compte la corrélation entre les variables (§4.4).
- Le rôle du critère d'arrêt est déterminant : un critère basé uniquement sur les variations de performances peut s'avérer trop brutal et stopper trop tôt la sélection (ou l'élimination) des variables (§4.3).
- Pour les problèmes de taille « raisonnable », il semble intéressant de faire à la fois la sélection de variables et l'apprentissage du réseau de neurones (§4.5)

L'observation de ces différents phénomènes nous a mené à proposer deux règles permettant d'améliorer à un moindre coût les méthodes dérivées d'OBD comme la plupart des méthodes de sélection de variables neuronales existantes :

- Il faut réapprendre le réseau à chaque étape, avant de ré-estimer la pertinence des variables.
- Le choix du meilleur ensemble de variables peut se faire grâce à l'estimation des performances sur un ensemble de test et à l'utilisation d'un test statistique pour déterminer l'ensemble de variables minimal.

Bibliographie

[AKA 70] Akaike, H. 1970. Statistical Predictor Identification, Ann. Inst. Statist. Math. 22 :203-217.

[BAT 94] Battiti, R. 1994. Using Mutual Information for Selecting Features in Supervised Neural Net Learning. IEEE Transactions on Neural Networks 5(4) :537-550.

[BEN 92] Bennani, Y. 1992. Approches connexionnistes pour la reconnaissance du locuteur : modélisation et identification. Thèse de Doctorat de l'Université d'Orsay, janvier 1992.

[BON 94] Bonnlander, B.V. and Weigend, A.S. 1994. Selecting Input Variables Using Mutual Information and Nonparametric Density Evaluation. In Proceedings of ISANN'94, Tainan, Taiwan. 42-50.

[BRE 84] Breiman, L. ; Friedman, J. ; Olshen, R. and Stone C. 1984. Classification and Regression Trees. Wadsworth International Group.

[CIB 94] Cibas, T.; Fogelman Soulié, F.; Gallinari, P. and Raudys, S. 1994. Variable Selection with Optimal Cell Damage. In Proceedings of ICANN'94.

[CIB9 4] Cibas, T.; Fogelman Soulié, F.; Gallinari, P. and Raudys, S. 1996. Variable Selection with Neural Networks. Neurocomputing 12 :223-248.

[CZE 96] Czernichow, T. 1996. Architecture Selection through Statistical Sensitivity Analysis. In Proceedings of ICANN'96, Bochum, Germany.

[DEB 91] De Bollivier, M.; Gallinari, P. and Thiria, S. 1991. Cooperation of Neural Nets and Task Decomposition. In Proceedings of IJCNN'91 (2) :573-576.

[DOR 96] Dorizzi, B.; Pellieux, G.; Jacquet, F; Czernichow, T. and Munoz, A. 1996. Variable Selection Using Generalized RBF Networks : Application to the Forecast of the French T-Bonds. In Proceedings of IEEE-IMACS'96, Lille, France.

[GOU 97] Goutte, C. 1997. Extracting the Relevant Decays in Time Series Modelling, Neural Networks for Signal Processing VII, Proceedings of the IEEE Workshop, Neural Networks for Signal Processing VII, Proceedings of the IEEE Workshop.

[GRA 98] Grandvalet, Y. and Canu, S. 1998. Outcomes of the equivalence of adaptative ridge with the least absolute shrinkage. Neural Information Processing Systems 11.

[GUS 95] Gustafson and Hajlmarsson. 1995. 21 maximum likelihood estimators for model selection. Automatica.

[HAS 93] Hassibi, B. and Stork, D.G. 1993. Second Order Derivatives for Network Pruning : Optimal Brain Surgeon. Neural Information Processing Systems 5 :164-171.

[HAS 94] Hassibi, B.; Stork, D.G. and Wolf, G. 1994. Optimal Brain Surgeon : Extensions and Performance Comparisons. Neural Information Processing Systems 6 :263-270.

[KIT 86] Kittler, J. 1986. Feature Selection and Extraction, Chapter 3 in Handbook of Pattern Recognition and Image Processing. Eds.Tzay Y.Young, King-Sun Fu, Academic Press. 59-83.

[LAR 94] Larsen, J. and Hansen, L.K. 1994. Generalized performances of regularized neural networks models. Proceedings of the 1994 IEEE Workshop on Neural Networks for Signal Processing. 42-51.

[LEC 90] LeCun, Y.; Denker, J.S. and Solla, S.A. 1990. Optimal Brain Damage. Neural Information Processing Systems 2 :598-605.

[LER 97] Leray, P. and Gallinari, P. 1997. Report on Variable Selection. Neurosat Project, Environment and Climate DG III, Science, Research and Development ENV4-CTP96-0314, D1-1-1.

[LER 99] Leray, P. and Gallinari, P. 1999. Feature Selection with Neural Networks. Behaviormetrika (special Issue on Analysis of Knowledge Representation in Neural Network Models. 26(1):145-166.

[LER 98] Leray, P. 1998. Apprentissage et Diagnostic de Systèmes Complexes : Réseaux de Neurones et Réseaux Bayesiens - Application à la gestion en temps réel du trafic téléphonique français. Thèse de Doctorat de l'Université Paris 6, septembre 1998.

[MAO 94] Mao, J.; Mohiuddin, K. and Jain, A.K. 1994. Parsimonious Network Design and Feature Selection Through Node Pruning. In Proceedings of the 12th International Conference on Pattern Recognition. 622-624.

[MOO 91] Moody, J. 1991. Note on generlization, regularization and architecture selection in non linear learning systems. Proceedings of the first IEEE Workshop on Neural Networks for Signal Processing. 1-10.

[MOO 92] Moody, J. and Utans, J. 1992. Principled Architecture Selection for Neural Networks : Application to Corporate Bond Rating Prediction. Neural Information Processing Systems 4.

[MOO 94] Moody, J. 1994. Prediction Risk and Architecture Selection for Neural Networks. in From Statistics to Neural Networks - Theory and Pattern Recognition Applications, Eds V. Cherkassky, J.H. Friedman, H. Wechsler, Springer-Verlag.

[PED 96] Pedersen, M.W.; Hansen, L.K. and Larsen, J. 1996. Pruning with generalisation based weight saliencies : γOBD, γOBS. Neural Information Processing Systems 8.

[RUC 90] Ruck, D.W.; Rogers, S.K. and Kabrisky, M. 1990. Feature Selection Using a MultiLayer Perceptron. In J. Neural Network Comput. 2 (2) :40-48.

[SAP 90] Saporta, G. 1990. Probabilités, Analyse des données et Statistiques, Editions Technip.

[STA 97] Stahlberger, A. and Riedmiller, M. 1997. Fast Network Pruning and Feature Extraction Using the Unit-OBS Algorithm. Neural Information Processing Systems 9 :655-661.

[TRE 97] Tresp, V.; Neuneier, R. and Zimmermann, G. 1997. Early Brain Damage. Neural Information Processing Systems 9:669-675.

[VAN 97] Van de Laar, P.; Gielen, S. and Heskes, T. 1997. Input Selection with Partial Retraining. In Proceedings of ICANN'97.

[YAC 97] Yacoub, M. and Bennani, Y. 1997. HVS : A Heuristic for Variable Selection in Multilayer Artificial Neural Network Classifier. in Proceedings of ANNIE'97. 527-532.

[ZAP 99] Zapranis, A. and Refenes A.P. 1999. Principles of neural model identification, selection and adequacy. Perszpectives in Neural Computing, Springer-Verlag