

Une étude des algorithmes de construction d'architecture des réseaux de neurones multicouches.

Norbert Tsopzé^{1*,**} Engelbert Mephu Nguifo*
Gilbert Tindo**

*CRIL-CNRS, IUT de Lens, SP 16 Rue de l'Université 62307 Lens Cedex
{tsopze,mephu}@cril.univ-artois.fr

**Département d'Informatique - Université de Yaoundé I BP 812 Yaoundé
tsopze.norbert@gmail.com, gtindo@uycdc.uninet.cm

Résumé. Le problème de choix d'architecture d'un réseau de neurones multicouches reste toujours très difficile à résoudre dans un processus de fouille de données. Ce papier recense quelques algorithmes de recherche d'architectures d'un réseau de neurones pour les tâches de classification. Il présente également une analyse théorique et expérimentale de ces algorithmes. Ce travail confirme les difficultés de choix des paramètres d'apprentissage (modèle, nombre de couches, nombre de neurones par couches, taux d'apprentissage, algorithme d'apprentissage,...) communs à tout processus de construction de réseaux de neurones et les difficultés de choix de paramètres propres à certains algorithmes.

1 Introduction

Un réseau de neurones est un ensemble de neurones interconnectés qui communiquent entre eux et avec l'extérieur. Un réseau de neurones se présente comme un graphe où les noeuds sont les différentes unités de réseau et les arcs représentent les connexions entre ces unités. Le nombre de couches, le nombre de neurones par couche et les interconnexions entre les différentes unités du réseau définissent l'architecture (encore appelée topologie) de celui-ci. Un neurone peut être appelé unité ou cellule. Comme tout système d'apprentissage supervisé, les systèmes d'apprentissage supervisé à base des réseaux de neurones fonctionnent en deux phases : la phase d'apprentissage qui consiste à construire à partir des observations (exemples présentés sous forme (x, y) où y représente l'observation de la fonction f en x) un système capable d'approximer la fonction f dont l'expression analytique n'est pas facile à trouver ; la phase de classement qui utilise le modèle construit en phase d'apprentissage pour produire des décisions (prédire un nouvel exemple qui ne faisait pas partie des observations de la base d'apprentissage). Définir la structure du réseau pour de tel système n'est pas une tâche évidente (J.Han et Hamber, 2001; A.Cornuéjols et Miclet, 2002). En effet, il n'existe aucune méthode permettant de définir et de justifier la structure d'un réseau de neurones (J.Han et Hamber, 2001).

¹Le Service de Coopération et d'Action Culturelle (SCAC) de l'ambassade de France à Yaoundé (Cameroun) a financé le séjour du premier auteur au CRIL pendant la réalisation de ce travail. Ce travail est partiellement financé par le ministère français des affaires étrangères.

Etude des algorithmes de construction d'architecture des réseaux de neurones.

La définition de l'architecture du réseau de neurones multicouches pour la résolution d'un problème donné reste un problème ouvert. Outre les méthodes génétiques (D.Curran et O'Riordan, 2002), ce problème est souvent résolu en utilisant deux approches : la première consiste à ajouter successivement des neurones et des connexions à une petite architecture, la deuxième quant à elle consiste à supprimer des neurones et des connexions d'une architecture initiale maximale. Ces deux approches ont souvent comme inconvénient le temps d'apprentissage élevé et imprévisible.

Les domaines d'application des réseaux de neurones sont multiples (Dreyfus et al., 2002) : la biologie moléculaire (analyse des séquences d'ADN (Shavlik et Towell., 1994)), prédiction, classification, traitement d'images, le génie logiciel (estimation des coûts de logiciel (S.Mbarki et al., 2004)), etc. Aucune explication ne justifie à notre connaissance la définition des architectures utilisées. Pour les problèmes de classification en particulier, plusieurs méthodes ont été développées et sont proposées dans la littérature (J.Yang et al., 1999; Yang et al., 1996; Parekh et al., 1997b). On peut classer ces méthodes en deux catégories : celles qui construisent l'architecture en utilisant un ensemble de connaissances de domaine (exemple de KBANN (Shavlik et Towell., 1994)) et les autres qui définissent cette architecture sans aucune connaissance (J.Yang et al., 1999; Parekh et al., 1997b; Yang et al., 1996; Parekh et al., 1995). Les algorithmes de construction des réseaux de neurones artificiels que nous avons rencontrés dans la littérature produisent des réseaux ayant les caractéristiques suivantes (Parekh et al., 1995, 1997a, 2000) : architecture minimale, habile à trouver le compromis entre les mesures de performances telles que le temps d'apprentissage, habilité à généraliser, ...etc. Ces méthodes constructives de réseau de neurones diffèrent par les facteurs suivants (Parekh et al., 1997a, 2000) : restriction des entrées (type de données en entrée), circonstances d'ajout d'une nouvelle unité, initialisation des poids de connexion de cette unité et son apprentissage.

Dans ce travail, notre intérêt porte sur les méthodes de recherche d'architecture des réseaux de neurones multicouches feed-forward (les informations circulent des entrées vers les sorties, sans retour) pour la résolution des problèmes de classification. Les principaux paramètres de mesure de performance traités sont : la taille du réseau (nombre de neurones, nombre de couches...), la complexité en temps et la capacité de généralisation. Certaines méthodes de recherche d'architecture de réseaux de neurones ont été évaluées sur des données de taille relativement petite et la qualité des résultats varie d'un ensemble de données à l'autre (Parekh et al., 1997a). D'autre part, une comparaison théorique des ces algorithmes n'a pas à notre connaissance été faite. Notre étude portera essentiellement sur la comparaison de ces algorithmes d'après les mesures de performances citées ci-dessus et des résultats expérimentaux sur les données tirées de la base UCI (Newmann et al., 1998). Les opérations supplémentaires de prétraitement de données telles que projection, binarisation, la normalisation et autres ne seront pas abordées dans cette étude.

Le reste du papier est organisé comme suit : la section suivante présente les réseaux de neurones multicouches, et quelques notions (définitions et apprentissage) liées aux réseaux de neurones multicouches ; la troisième section recense les algorithmes de construction d'architecture neuronale. Les analyses expérimentales et théoriques feront l'objet de la quatrième section.

2 Les réseaux de neurones multicouches et algorithmes d'apprentissage

2.1 Généralités

Ce sont des réseaux de neurones dont les architectures vérifient les propriétés suivantes :

1. les cellules (neurones ou unités) sont réparties de façon exclusive sur les différentes couches.
2. la première couche ou couche d'entrée est composée des cellules d'entrée qui correspondent aux n variables d'entrée ; elle a généralement un nombre d'unités égal au nombre d'attributs des exemples.
3. la couche cachée est composée des unités qui effectuent des calculs intermédiaires entre les entrées et les sorties. Elle peut être composée de plusieurs autres couches.
4. la dernière couche est celle de décision ; elle peut avoir aussi un nombre d'unités égal au nombre de classes.

Les poids de connexion des réseaux multicouches sont généralement modifiés par rétropropagation (Rumelhart et al., 1986a,b). Cet algorithme produit des bons résultats lorsque l'architecture est appropriée, il est également utilisé lorsque l'architecture du réseau reste statique (D.Curran et O'Riordan, 2002). Des outils tels que WEKA (Witten et Frank, 2005) et SNNS (Stuttgart Neural Network Simulator)² offrent aux utilisateurs la possibilité de définir la structure de leur réseau, ces outils apprennent ces réseaux par rétropropagation. La difficulté majeure est de trouver cette architecture (A.Cornuéjols et Miclet, 2002).

2.2 Algorithme d'apprentissage des poids de connexion

L'apprentissage dans les systèmes neuronaux peut se faire par unité ou par couche. Le principal algorithme d'apprentissage des unités neuronales est le perceptron ; lorsque les données ne sont pas linéairement séparables, il est remplacé par l'une de ses variantes : pocket with racket, barycentric,...etc. L'apprentissage d'une couche peut être généralisé à toutes ses unités ou se faire suivant le principe du Winner Take All (WTA). Afin de montrer l'influence de l'algorithme d'apprentissage sur la complexité du système, nous présentons (voir tableau 1) sans entrer dans les détails les complexités en temps de ces algorithmes. Les détails sur ces algorithmes d'apprentissage sont présentés dans (Parekh et al., 1999, 2000), (Gallant, 1990) et (Frean, 1992a). Théoriquement, ces algorithmes utilisent de manière similaire l'espace mémoire. Le tableau 1 présente les complexités en temps de ces algorithmes d'apprentissage dans lequel les variables désignent : n le nombre d'objets, M le nombre de classes, m le nombre d'attributs et Nit le nombre d'itérations dans l'algorithme d'apprentissage.

3 Algorithmes de construction des RNA

Cette section présente un résumé des algorithmes MTiling (Yang et al., 1996), MTower (Gallant, 1990; Parekh et al., 1995, 1997a), MUpstart (Parekh et al., 1997b), Distal (J.Yang

²simulateur des réseaux de neurones disponibles à l'adresse Internet <http://www-ra.informatik.uni-tuebingen.de/SNNS/>

Etude des algorithmes de construction d'architecture des réseaux de neurones.

Algorithmes	Complexité en temps
Perceptron	$n \times m \times M$
Pocket with racket modification	$n \times m \times M \times Nit$
Thermal Perceptron	$n \times m \times M \times Nit$
Barycentric Correction	$n \times m \times M \times Nit$

TAB. 1 – Complexité en temps des algorithmes d'apprentissage

et al., 1999). La convergence de ces algorithmes est démontrée dans (Parekh et al., 1995) et dans (Parekh et al., 1997a).

3.1 L'algorithme Mtiling (Yang et al., 1996)

Mtiling est une adaptation de l'algorithme Tiling (Mezard et Nadal., 1989) à la classification multiclassées. Cette méthode construit un réseau de neurones multicouches dans lequel les unités d'un niveau (couche) reçoivent des unités du niveau inférieur (immédiat). En cas de mauvais classement du réseau courant, la procédure détermine le neurone ayant fait le maximum d'erreurs, augmente un certain nombre de neurones (auxillaires) à la couche de sortie courante, elle augmente également une nouvelle couche de M neurones au réseau et connecte les entrées de cette couche aux sorties des unités de la couche de sortie ; cette couche (ajoutée) devient la nouvelle couche de sortie. Les neurones ajoutés sont appris individuellement par les algorithmes pocket ou une variante. Le nombre maximal de couches cachées H est spécifié par l'utilisateur. La couche de sortie fonctionne suivant le principe de WTA afin d'assurer qu'une seule classe soit active pour un exemple donné.

3.2 L'algorithme Mtower (Parekh et al., 1997a)

Cette méthode construit le réseau sous forme de tour comme la méthode originale Tower (Gallant, 1990). L'architecture finale du réseau est telle que : les neurones au sommet (sortie) sont connectés à tous les neurones d'entrée et un neurone de la couche k reçoit de l'information de tous les neurones de la couche $k - 1$ (immédiatement connectés à la couche k). La tour est construite en ajoutant successivement les couches de M unités au réseau. Ces unités sont apprises par l'une des variantes du perceptron. La couche ajoutée est complètement connectée à la couche de sortie et à la couche d'entrée ; cette couche devient ainsi la nouvelle couche de sortie. La modification du réseau est répétée jusqu'à l'obtention de la précision (de classement) fournie par l'utilisateur, l'algorithme peut aussi d'arrêter le nombre maximal de couche est atteint. La couche de sortie fonctionne aussi suivant le principe du WTA. La méthode Mpyramid (Parekh et al., 1997a) est semblable à Mtower à la seule différence que la couche ajoutée reçoit l'information de toutes les couches précédentes.

3.3 L'algorithme MUpstart(Parekh et al., 1997b)

Mupstart est une version de l'algorithme Upstart (Frean, 1992b) pour la classification multiclassées. Comme Upstart, cet algorithme est basé sur une correction d'erreur produite par le

réseau courant. Ce réseau a une couche d'entrée de $N + 1$ unités et une couche de sortie de M unités. Le fils gauche, X est augmenté au nœud en cas de "wrongly on" et le fils droit Y en cas de "wrongly off". On parle de "Wrongly on" (resp. "Wrongly off") lorsque la sortie obtenue est "1" (resp. "0") alors que l'on désirait obtenir plutôt "0" (resp. "1"). Le neurone ajouté pour corriger l'erreur produite à cette étape est appris par l'algorithme du perceptron ou une de ses variantes. Ces neurones ajoutés sont appris avec pour sorties attendues (T_x et T_y) définies dans le tableau 2 ; dans ce tableau, y (resp. o) représente la sortie désirée (resp. sortie obtenue), T_x (resp. T_y) est la sortie attendue pour l'apprentissage des unités x (resp. y) pour la correction de la différence entre y et o .

y	o	T_x	T_y
0	0	0	0
0	1	1	0
1	0	0	1
1	1	0	0

TAB. 2 – Tableau présentant les sorties attendues des unités ajoutées

3.4 L'algorithme Distal (J.Yang et al., 1999)

C'est un algorithme qui est basé sur le calcul de distance entre exemple (ou entre attribut). La distance peut être euclidienne, ou autre. Cette procédure calcule, trie d'abord les distances entre exemples (ou attributs) et les stocke dans une matrice D . La procédure Distal construit à partir de la matrice des distances la couche cachée du réseau. Cette couche cachée est construite de la manière suivante : un neurone est ajouté pour apprendre k exemples tel que k soit l'indice de la classe ayant la plus grande plage d'exemples consécutifs dans D appartenant à la même classe. Les neurones de la couche cachée dans Distal sont à seuil sphérique (neurone actif si $\theta_{low} \leq WX \leq \theta_{high}$ et inactif sinon, où WX est le résultat en sortie) et les poids de connexion entre l'unité ajoutée et les unités d'entrée sont initialisés par les attributs de l'exemple i à partir duquel k a été trouvé. A l'étape suivante, tous les poids du réseau entre la couche cachée et celle de sortie sont multipliés par deux. Les unités de la couche fonctionnent suivant le principe de WTA.

4 Complexités - Evaluations théoriques

Les algorithmes de recherche d'architectures de réseau de neurones peuvent être classés suivant l'approche de construction du réseau en deux grands groupes :

1. l'approche descendante, le réseau final est obtenu par élimination des neurones et de connexions. L'architecture initiale comporte un nombre suffisant de neurones et de connexions capables de bien classer tous les exemples.
2. L'approche ascendante, le réseau est obtenu par ajout de neurones et de connexions. Les ajouts se font lorsque le réseau considéré produit des erreurs. L'architecture initiale

Etude des algorithmes de construction d'architecture des réseaux de neurones.

comporte un nombre assez réduit de neurones, généralement deux couches (la couche d'entrée et la couche de sortie).

Tous les algorithmes présentés précédemment construisent le réseau par l'approche ascendante. Seul Mtiling (Yang et al., 1996) permet de combiner les deux approches en recherchant par l'approche ascendante une architecture maximale (architecture ayant le maximum H) de couches et ensuite élaguant cette architecture.

Asymptotiquement, les algorithmes que nous avons présentés ont un comportement semblable. La complexité de ces algorithmes est calculée et présentée (tableau 3) au pire des cas : pour Distal par exemple, en général le nombre d'exemples est toujours supérieur au nombre d'attributs, c'est ce qui justifie que le temps maximal est obtenu lorsque la distance entre exemples est considérée. Le tableau 3 présente les complexités en temps et en espace mémoire des algorithmes tandis que le tableau 4 présente d'autres aspects non négligeables de ces algorithmes ; la projection consiste à ajouter un attribut supplémentaire ayant pour valeur la somme des carrés des autres attributs de l'exemple ($\sum_i x_i^2$). Les particularités de ces algorithmes sont

Algorithmes	Complexité en temps	Complexité en espace
Distal	N^3	N^2
Mupstart	$H \times Nit \times m \times N$	$\max(H \times (m + 1), H \times M^2)$
Mtiling	$(M + m)H \times n \times m \times Nit$	$\max((M + m)m, (M + m)^2 H)$
Mtower	$H \times M \times n \times m \times Nit$	$\max((m + 1) \times M, H \times M^2)$
Mpyramid	$H \times M \times n \times m \times Nit$	$\max((n + 1) \times M^2, H \times M)$
P.-Cascade	$H \times Nit \times n \times m \times N$	$\max(H \times (m + 1), H \times M^2)$

TAB. 3 – Complexité en temps et en espace mémoire

Algorithmes	Nombre de couches cachées	Nombre de neurones dans la couche cachée	opération supplémentaire
Distal	1	N	Distance
Mupstart	1 ou H	$H \times M$	Projection
Mtiling	H	$H \times M$	Projection
Mtower	H	$H \times M$	Projection
Mpyramid	H	$H \times M$	Projection
Perceptron-Cascade	H	$H \times M$	Projection

TAB. 4 – Recapitulatif de certains aspects.

les suivantes :

La particularité de l'algorithme Mtiling, la correction d'erreur se fait par ajout d'une couche de M neurones ("maîtres"), cette couche devient la nouvelle couche de sortie ; ajout de k neurones à la couche cachée immédiatement connectée à la couche de sortie. Les connexions sont complètes entre couches adjacentes.

Mtower ajoute tout simplement une nouvelle couche de M neurones. Cette couche devient également la nouvelle couche de sortie et est complètement connectée à la couche d'entrée.

Distal est plus spécifique que les autres méthodes ; elle construit un réseau ayant une seule couche cachée, chaque neurone de la couche cachée permet de séparer un sous ensemble d'exemples. le réseau obtenu garantit le bon classement des exemples sans être appris.

La méthode MUpstart quant à elle ajoute à chaque correction un seul neurone ; ce neurone est appris individuellement et connectée au neurone ayant produit le plus grand nombre d'erreurs.

5 Expérimentations

Ces algorithmes ont été testés sur les données de la base UCI (Newmann et al., 1998), les attributs de ces ensembles sont tous numériques et sans valeur manquante. Les expérimentations précédentes sont présentées dans (Parekh et al., 1997a; J.Yang et al., 1999) Nos expériences ont porté sur les algorithmes Distal, Mupstart, Mtiling, Perceptron-Cascade, Mpyramid, Mtower. Ces expériences étaient validées par validation croisée d'ordre 10 (10-Cross-Validation). Dans nos expériences, la distance choisie (sans aucune justification) entre exemples pour Distal est la distance euclidienne. Les unités ajoutées au cours de la construction du réseau avec les autres ont été apprises par "Perceptron with racket modification". Les résultats obtenus après le classement que nous avons obtenus ne sont pas aussi meilleurs que ceux présentés dans (Newmann et al., 1998).

Données	Nombre d'exemples	Nombre d'attributs	Nombre de classes
Spambase	4601	57	2
pendigits	7494	17	10
Opdigits	3823	64	10

TAB. 5 – Description des données utilisées pour les expérimentations

Au cours de ces expériences, l'attention s'est surtout portée pour chaque algorithme sur les aspects suivants : le temps nécessaire pour la construction du réseau, le nombre total de neurones du réseau construit et la capacité de généralisation.

Nos expérimentations sont classées en deux groupes :

Distal est la seule méthode qui construit son réseau sans faire d'apprentissage ; pour cette raison, nous l'avons évalué séparément des autres. Cette méthode présente des très bons résultats expérimentaux en présence des données de petite taille. Mais lorsque les données atteignent une certaine taille, Distal ne produit plus de résultat. Cela est due à un besoin excessif de l'espace mémoire. La figure 5 montre l'évaluation de Distal en fonction de la taille de données.

Les autres algorithmes construisent un réseau de neurones en modifiant simultanément la structure de celui-ci et les poids de connexion. Leurs évaluations expérimentales sont très coûteuses en temps. L'apprentissage des unités ajoutées lors de la construction du réseau s'est fait avec l'algorithme "Perceptron with racket modification" avec 500 itérations, un maximum de 350 (choix arbitraire) neurones par couches et une précision d'apprentissage et de test souhaitée à 100%. Le temps affiché ici est en secondes, mais représente le temps obtenu pour le meilleur système en validation croisée.

Etude des algorithmes de construction d'architecture des réseaux de neurones.

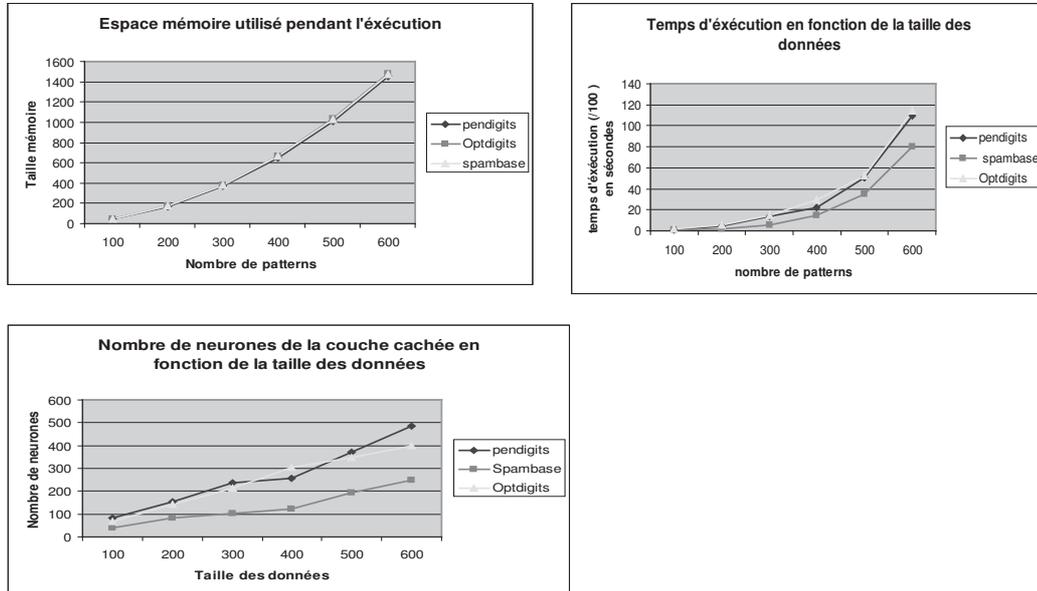


FIG. 1 – Evaluation expérimentale de l'algorithme Distal

Les tableaux 6 et 7 présentent les résultats expérimentaux de ces algorithmes sur la base "spambase" avec respectivement comme nombre maximum de couches cachées 2 et 24. Ces tableaux montrent l'influence du nombre de couches sur le taux de généralisation. En effet malgré un temps plus élevé, nous obtenons une bonne capacité de généralisation lorsque le nombre de couches est élevé.

Les tableaux 8 et 10 présentent les résultats obtenus de l'expérimentation sur les données présentées plus haut. Le nombre maximal de couches cachées choisies au cours de ces expériences est 2 (ce choix est arbitraire et nous a permis d'avoir les résultats avec un temps relativement raisonnable). Le symbole "-" signifie que soit l'expérience a échoué (généralement un débordement de la pile), soit le nombre maximal de couches a été atteint avec une précision inférieure à celle souhaitée. Notons que les données "pendigits" et "optdigits" ont pour particularité par rapport à "spambase" le fait que les exemples appartiennent à 10. Le comportement des algorithmes sur les données ci-dessus n'est pas très appréciable, mais ne saurait être généralisé à toute la classification multiclassées. Les tableaux 9 et 11 montrent une amélioration de la capacité de généralisation lorsque le nombre maximum de couches passe de 2 à 24 ; ce qui explique l'influence du nombre de couches sur la capacité de généralisation.

6 Discussion - Conclusion

Ces méthodes de construction des réseaux de neurones sont presque semblables ; ces algorithmes initialisent le réseau à une structure minimale et modifient ce réseau par ajout des

Algorithmes	Nombre de neurones dans la couche cachée	Temps	Capacité de généralisation
Cascade	1	8,42	48,45
MPyramid	2	48,8	0
MTiling	1	24,79	21,78
MTower	1	507,79	22,02
MUpstart	1	8,5	45,12

TAB. 6 – Expérimentations des algorithmes sur la base de données "Spambase" avec un nombre maximal de couches égal à 2

Algorithmes	Nombre de neurones dans la couche cachée	Temps	Capacité de généralisation
Cascade	1	18,89	64.07
MPyramid	24	294.69	30.34
MTiling	24	345.80	63.55
MTower	24	143.34	55.78
MUpstart	1	87,5	87.49

TAB. 7 – Expérimentations des algorithmes sur la base de données "Spambase" avec un nombre maximal de couches égal à 24

unités au fur et à mesure que les erreurs surviennent. La question qui se pose est de savoir comment diriger le choix d'un concepteur face au problème d'architecture du réseau de neurones. Le choix de l'architecture de réseau de neurones devra être orienté vers la satisfaction de l'utilisateur final du système c'est-à-dire que ce choix devrait aboutir à la production d'un réseau de neurones ayant une bonne capacité de généralisation.

D'après notre étude, nous pouvons séparer ces algorithmes en deux catégories : Les algorithmes qui construisent les réseaux ayant une seule couche cachée ; dans cette catégorie, peut être classé Distal. Cet algorithme a aussi pour avantage la faible intervention de l'utilisateur ; mais il y a un besoin énorme en espace mémoire. Les autres algorithmes construisent des réseaux pouvant avoir H couches cachées ; l'utilisateur doit fournir à l'algorithme en plus du nombre maximal (H) de couches cachées, la précision souhaitée, l'algorithme d'apprentissage. A partir des tables 6 et 7, on remarque une variation de taux généralisation en fonction du nombre maximal de couches. On peut également noter le mauvais comportement de ces algorithmes sur les données "pendigits" et "optdigits" mais nous ne pouvons pas généraliser sur toutes les grandes bases de données. Pour généraliser, nous devons tenir compte de l'influence de la taille du réseau, de la répartition des données...

Quelque soit la méthode de construction des réseaux de neurones, le choix des paramètres (modèle du réseau, nombre de couches, nombre de neurones par couches, définition des connexions, taux d'apprentissage,...) reste toujours très problématique : pour la méthode

Etude des algorithmes de construction d'architecture des réseaux de neurones.

Algorithmes	Nombre de neurones dans la couche cachée	Temps	Capacité de généralisation
Cascade	10	64,37	10
MPyramid	10	63,70	20
MTiling	10	64,23	0
MTower	10	64,26	10
MUpstart	-	-	-

TAB. 8 – Expérimentations des algorithmes sur la base de données "Pendigits" avec un nombre maximal de 2 couches cachées

Algorithmes	Nombre de neurones dans la couche cachée	Temps	Capacité de généralisation
Cascade	-	-	-
MPyramid	240	4648,9	35,6
MTiling	240	2775,4	18,7
MTower	240	3064,26	22,8
MUpstart	-	-	-

TAB. 9 – Expérimentations des algorithmes sur la base de données "Pendigits" avec un maximum de 24 couches

distal, il faut trouver l'influence du choix de calcul de distance entre exemples ou entre attributs ; nos expériences confirment son comportement appréciable sur les données de petite taille, et un mauvais comportement sur des grands volumes de données ; cette défaillance est due à son besoin énorme en espace mémoire. Pour les méthodes Mupstart, Mtiling, Mtower, Pyramid, Perceptron-Cascade, trouver l'influence de l'algorithme d'apprentissage des unités ajoutées et le choix du nombre maximal de couches cachées reste problématique. Nous étudions actuellement l'application de ces méthodes sur de grandes bases de données. Il est aussi envisageable de trouver un moyen de traiter des données résidentes sur le disque dur avec ces algorithmes.

Références

- A.Cornuéjols et L. Miclet (2002). *Apprentissage Artificiel : Concepts et algorithmes*. Eyrolles.
- D.Curran et C. O'Riordan (2002). Applying evolutionary computation to designing neural networks: A study of the state of the art. *department of Information Technology, NUI Galway*.
- Dreyfus, G., M. Samuelides, J.M.Martinez, M. Gordon, F.Badran, S.Thiria, et L.Hérault (2002). *Réseaux de Neurones : Méthodologie et applications*. Eyrolles.

Algorithmes	Nombre de neurones dans la couche cachée	Temps	Capacité de généralisation
Cascade	10	68,21	0
MPyramid	10	69,43	0
MTiling	10	909,22	0
MTower	10	68,90	0
MUpstart	10	68,55	0

TAB. 10 – Expérimentations des algorithmes sur la base de données "Optdigits" avec un maximum de 2 couches

Algorithmes	Nombre de neurones dans la couche cachée	Temps	Capacité de généralisation
Cascade	-	-	-
MPyramid	240	4648,9	18,5
MTiling	240	3909,22	10,3
MTower	240	3268,90	12,4
MUpstart	-	-	-

TAB. 11 – Expérimentations des algorithmes sur la base de données "Optdigits" avec un maximum de 24 couches

- Frean (1992a). A thermal perceptron learning rule. *Neural computation* 4, 946–957.
- Frean (1992b). The upstart algorithm: A method for constructing and training feed forward neural networks. *Neural computation* 4, 198–209.
- Gallant, S. (1990). Perceptron - based learning algorithms. *IEEE Transactions On Neural networks* 1, 179–191.
- J.Han et M. Hamber (2001). *Datamining: Concepts and Techniques*. Morgan Kauffman Publishers.
- J.Yang, R. Parekh, et V. Honavar (1999). Distal: An inter-pattern distance-based constructive learning algorithm. *Intell. Data Anal* 3, 55–73.
- Mezard, M. et J. Nadal. (1989). Learning feed forward network: the tiling algorithm. *J. Phys A: Math Gen.* 22, 2191–2203.
- Newmann, D., S. Hettich, C. Blake, et C. Merz (1998). (uci)repository of machine learning databases. *Dept. Inform. Comput. Sci., Univ. California, Irvine, CA*, <http://www.ics.uci.edu/AI/ML/MLDBRepository.html>.
- Parekh, Y. R.G., et H. V.G (1997b). Mupstart-a constructive neural network learning algorithm for multi-category patterns classification. *In Proceedings of the IEEE/INNS International Conference on Neural Networks INNS'97*, 81–106.
- Parekh, R., J.Yang, et V. Honavar (1995). Constructive neural networks learning algorithms for

Etude des algorithmes de construction d'architecture des réseaux de neurones.

- multi-category classification. Tech report isu cs tr 95-15, Department of Computer Science Iowa State University. Describes the package natbib.
- Parekh, R., J. Yang, et V. Honavar (1997a). Constructive neural network learning algorithm for multi-category real \mathbb{U} valued pattern classification. Tech report tr 96-14, Department of Computer Science Iowa State University. Describes the package natbib.
- Parekh, R., J. Yang, et V. Honavar (1999). Comparison of performance of variants of single-layer perceptron algorithms on non-separable datasets. *Neural, Parallel, and Scientific Computations*. 3.
- Parekh, R., J. Yang, et V. Honavar (2000). Constructive neural-network learning algorithms for pattern classification. *IEEE TRANSACTIONS ON NEURAL NETWORKS* 11(2), 436–451.
- Rumelhart, D. E., G. E. Hinton, et R. J. Williams (1986a). Learning internal representations by error propagation. *In Parallel Distributed Processing, Cambridge, MA: MIT Press*. 1, 318–362.
- Rumelhart, D. E., G. E. Hinton, et R. J. Williams (1986b). Learning representations by back-propagating errors. *Nature* 323, 533–536.
- Shavlik, J. W. et G. G. Towell. (1994). Kbnns: Knowledge based artificial neural networks. *Artificial Intelligence* 70 (1,2), 119–165.
- S.Mbarki, A. Idri, et A. Abram (2004). Application du réseau de neurones rbf à l'estimation des coûts de logiciel. *Conférence africain sur la recherche en Informatique CA-RI'04*, 405–413.
- Witten, I. H. et E. Frank (2005). *Data Mining: Practical machine learning tools and techniques, 2nd Edition*. Morgan Kaufmann, San Francisco.
- Yang, J. Parekh, et H. V. R. (1996). Mtiling-a constructive neural network learning algorithm for multi-category pattern classification. *Proceedings of the World Congress on Neural Networks*, 182–187.

Summary

The choice of neural network architecture remains a tremendous task for each neural network user. This paper describes different algorithms to build feed-forward multilayer neural networks architecture. These algorithms differ on many features. This paper presents a theoretical study of these algorithms, as well as an experimental study using 10 cross-validation on datasets taken from UCI repository. These experiments point out difficulties to choose networks parameters such as learning algorithm, learning accuracy, number of layers.