Apprentissage de concepts à l'aide des réseaux de neurones : une approche de construction du réseau basée sur les exemples positifs

Gilbert TINDO
Université de Yaoundé I
Faculté des sciences
Département d'informatique
B.P. 812 Yaoundé
CAMEROUN
gtindo@uycdc.uninet.cm

Engelbert MEPHU NGUIFO
Université d'Artois, IUT de Lens
CRIL – CNRS FRE 2499
Rue de l'université SP,62307 Lens cedex
FRANCE
mephu@cril.univ-artois.fr

RÉSUMÉ. Dans ce papier, nous proposons une approche hybride pour la construction des réseaux de neurones en vue de l'apprentissage de concepts. Nous montrons qu'à partir des hypothèses couvrant chacune un seul exemple positif, il est possible de construire une hypothèse couvrant tous les exemples positifs telle que le réseau de neurones qu'on pourrait en déduire minimise sensiblement le nombre de neurones et de connexions entre neurones. On obtient ainsi un réseau de neurones, où la phase de reconnaissance est accélérée et présentant de meilleures performances que celles des réseaux construits à l'aide des techniques classiques.

ABSTRACT. In this paper we propose an hybrid approach to build neural networks for concept learning. Using hypothesis that cover one positive example, we show that it is possible to build an hypothesis covering all positive examples such that the generated neural network minimize the number of neurons and the number of connexions between neurons. Our model has a rapid learning phase, and has good performance compared to classical neural networks.

MOTS-CLÉS : apprentissage de concepts, réseau de neurones, topologie de réseau.

KEYWORDS: concept learning, neural network, network architecture.

1. Introduction

Les réseaux de neurones sont des outils mathématiques introduits suite aux travaux de W. Pitts et W.S. McCulloh [1], visant à produire un modèle du cerveau humain. Il s'agit en fait d'une fonction mathématique produisant des résultats à partir des données d'un problème et possédant des paramètres ajustables. L'ajustement des paramètre appelé généralement apprentissage se fait à partir de quelques données du problème. Une fois les paramètres bien ajustés on peut utiliser le modèle dans la phase de reconnaissance, c'est-à-dire qu'on lui fournit une donnée et le modèle retourne un résultat. Les réseaux de neurones sont utilisés aujourd'hui de façon industrielle dans de nombreux domaines tels que la reconnaissance de formes, l'économie ou la finance, …etc . Ils sont une excellente alternative pour l'apprentissage artificiel [8], et sont même préférés dans des situations où on veut construire un modèle non linéaire d'un problème pour lequel on ne dispose pas d'une très grande quantité de données pour ajuster les paramètres du modèle[6].

Pour réaliser un modèle à l'aide des réseaux de neurones, le concepteur doit résoudre deux grands problèmes : le choix de la topologie du réseau et le choix des poids de connexion entre neurones. En ce qui concerne le choix des poids de connexion de nombreux algorithmes supervisés ou non supervisés existent dans la littérature, pour ajuster ces poids à partir de valeurs initiales aléatoires et en fonction des données. Le choix de la topologie suppose que le concepteur doit fixer le nombre de couches du réseau, le nombre de neurones par couche et enfin les connexions entre neurones. En général un réseau de neurones de trois couches est suffisant pour apprendre n'importe quelle concept booléen : le problème majeur à résoudre est celui du nombre de neurones cachés et de leur connexion avec les autres neurones. Deux stratégies sont généralement utilisées pour résoudre ce problème:

1) Commencer avec un petit nombre de neurones dans la couche cachée : faire l'apprentissage et valider le réseau. Si les résultats sont bons, alors retenir l'architecture, sinon augmenter un neurone caché et des connexions et recommencer le processus.
2) Commencer avec un grand nombre de neurones cachés : faire l'apprentissage jusqu'à convergence et relever les performances de l'architecture. Ensuite, diminuer quelques neurones et reprendre l'apprentissage. Si les performances restent bonnes diminuer encore un neurone et reprendre le processus. L'architecture retenue est celle ayant fournit le meilleur taux d'apprentissage et comportant le plus petit nombre de neurones.

Dans chacun des deux cas, le processus qui mène à une architecture acceptable peut être très long. Pour l'apprentissage de concepts, si on a N exemples du problème, on peut construire un réseau de neurones de trois couches ayant N neurones dans la couche cachée capable de reconnaître à 100% les exemples d'apprentissage. Mais ce nombre N n'est pas l'optimum et le réseau ainsi construit a tendance à faire un apprentissage par coeur.

Dans ce papier, nous proposons un algorithme pour déterminer le nombre de neurones de la couche cachée d'une architecture neuronale de trois couches en vue de l'apprentissage de concepts dont l'espace de représentation est également binaire. Le réseau obtenu n'est pas en général complètement connecté. La suite du papier est organisée de la manière suivante. D'abord nous présentons notre approche, ensuite nous appliquons cette approche à la construction d'un réseau de neurones de trois couches pour la reconnaissance d'une fonction booléenne : la fonction majorité.

2. Un algorithme de construction des réseaux de neurones.

L'apprentissage de concepts binaires peut se faire en utilisant les techniques classiques de constructions de réseau de neurones. Pour la plupart de ces techniques le choix du nombre de couches, des connexions entre neurones et l'initialisation des connexions se font de façon aléatoire. L'une des méthodes qui essaient de tirer profit des connaissances a priori sur le problème à apprendre est la méthode dite KBANN introduite dans[3,4]. Il s'agit d'une approche de construction de réseaux de neurones dans laquelle le problème à résoudre est décrit par un ensemble hiérarchique de règles d'inférences. Dans cette description, la conclusion finale est le neurone de sortie, les conclusions intermédiaires sont les neurones cachées et les prémices des règles de bas niveau sont les neurones d'entrée. Les dépendances sont représentées par les liens de connexion entre neurones. La fixation des poids de connexion entre un neurone et les neurones de qui il reçoit des informations se fait de la manière suivante :

1) Considérer la règle d'inférence correspondante dans la base des connaissances. Les connexions correspondant aux dépendances excitatrices, c'est-à-dire celles dont les variables associées dans la règle d'inférence ne sont pas complémentées, ont des poids w, et les autres connexions ont des poids – w, où w est une constante positive
2) le seuil d'excitation du neurone est choisi égal à n*w -ϕ, où n est le nombre de dépendances excitatrices, ϕ une constante choisie de sorte que la fonction de transfert du neurone retourne une valeur proche de 0.9 si tous les antécédents excitateurs sont à 1 et retourne une valeur proche de 0.1 dans la cas contraire. Dans [3,4], on propose de choisir w égal à 3 et ϕ égal à 2.3 .
3) Ajouter à tous les poids de connexion et les seuils d'excitation des quantités aléatoires proches de zéro pour perturber le réseau et faire l'apprentissage avec un algorithme d'apprentissage des réseaux de neurones tel que l'apprentissage du gradient.

Il faut noter que le réseau obtenu à l'étape 2 ci-dessus mémorise par cœur tous les données d'apprentissage. Il les reconnaît à 100%, mais a des capacités de généralisation médiocres. En perturbant le réseau comme indiqué à l'étape 3, on obtient un réseau de neurones dont les capacités de généralisation sont meilleures. Sur des exemples réels, les expérimentations faites dans [3], permettent d'affirmer qu'un réseau construit ainsi a de bonnes performances que celles obtenues avec des réseaux initialisés de façon aléatoire. Notre approche consiste à déduire une hypothèse couvrant tous les exemples positifs

disponibles pour l'apprentissage, et à rechercher une hypothèse équivalente plus compacte à partir de laquelle on déduit un réseau de neurones en utilisant la technique KBANN. Il s'agit en fait d'éliminer un maximum de redondances dans les règles d'inférences décrivant le problème à résoudre, permettant ainsi de construire un réseau de neurones ayant moins de neurones cachés et de connexions.

Un concept binaire dont l'espace de représentation est également binaire peut s'écrire sous forme d'une expression booléenne dans la logique des propositions. Soient x_1, x_2, ..., x_N les attributs décrivant un objet. Un monôme formé d'une conjonction des N attributs complémentés ou non, est une hypothèse couvrant un et un seul des exemples positifs ou négatifs du concept. En faisant une disjonction de toutes les hypothèses couvrant les différents exemples positifs, on obtient une hypothèse plus générale que chacune des hypothèses formées à partir d'un seul exemple. Cette hypothèse est cohérente, vis-à-vis des données d'apprentissage. On peut construire un réseau de neurones pour apprendre cette expression, en associant un neurone caché à chaque monôme. Le réseau de neurones obtenu a trois couches avec un neurone sur la couche de sortie, M neurones sur la couche cachée (où M est le nombre d'exemples dans les données d'apprentissage), et N neurones d'entrée (où N est le nombre d'attributs). Les connexions entre neurones de la couche i et ceux de la couche i-1 sont complètes. En utilisant l'approche décrite ci-dessus pour fixer les poids et les seuils d'excitation, on obtient un réseau de neurones qui présente les avantages des réseaux construits grâce à la méthode KBANN. Dans certains cas, cette architecture peut être améliorée. Nous proposons pour cela de construire le réseau de neurones à partir d'une hypothèse cohérente équivalente à celle obtenue en faisant une disjonction des hypothèses couvrant un seul exemple positif. Nous disons que deux hypothèses sont équivalentes si elles couvrent les mêmes exemples d'apprentissage. Le problème à résoudre ici est celui de trouver une stratégie pour obtenir des hypothèses cohérentes équivalentes, c'est-à-dire des stratégies pour éliminer les redondances dans une hypothèse pour obtenir une hypothèse plus compacte. Nous proposons pour cela d'utiliser une démarche analogue aux techniques algébriques de simplification de fonctions booléennes. En effet, un exemple positif ou négatif du concept est complètement décrit par une conjonction des N attributs possibles complémentés ou non complémentés. L'attribut est complémenté s'il a pour valeur 0 et est non complémenté s'il a pour valeur 1. Un monôme $x_1 \wedge x_2 \wedge \ldots \wedge x_i \wedge \ldots \wedge x_N$, (où x_i $1 \le i \le N$ est le $i^{ème}$ attribut) est une hypothèse couvrant un seul exemple. Considérons deux hypothèses couvrant chacune un seul exemple, $H1=x_1 \wedge x_2 \wedge \ldots \wedge x_i \wedge \ldots \wedge x_N$ et $H2=y_1 \wedge y_2 \wedge \ldots \wedge y_i \wedge \ldots \wedge y_N$. La disjonction $H=x_1 \wedge x_2 \wedge \ldots \wedge x_i \wedge \ldots \wedge x_N \vee y_1 \wedge y_2 \wedge \ldots \wedge y_i \wedge \ldots \wedge y_N$ est une hypothèse couvrant exactement deux exemples. L'hypothèse H est plus générale que chacune des hypothèses H1 et H2. Nous notons H=*généralisé*(H1,H2). S'il existe i tel que $x_i \neq y_i$ et $x_j = y_j$, quel que soit j différent de i, alors l'hypothèse H peut se réécrire de la façon suivante $H'=x_1 \wedge x_2 \wedge \ldots \wedge x_{i-1} \wedge x_{i+1} \wedge \ldots \wedge x_N$. Nous interprétons cela en disant que l'hypothèse H' est équivalente à l'hypothèse H et est aussi une généralisation des hypothèses H1 et H2. L'hypothèse H' est plus compacte que l'hypothèse H, en ce sens qu'elle fait intervenir moins d'attributs.

En combinant deux à deux les hypothèses initiales ne couvrant qu'un seul exemple d'apprentissage, nous obtenons des hypothèses couvrant chacune deux exemples d'apprentissage. En itérant ce processus, plusieurs fois, on finit par obtenir quelques hypothèses seulement, plus compactes que les hypothèses initiales couvrant chacune un certain d'exemples de l'ensemble d'apprentissage. On arrête le processus dès qu'on ne peut plus générer une nouvelle hypothèse plus générale que deux hypothèses quelconques de l'ensemble courant des hypothèses. L'hypothèse la plus générale à retenir pour construire le réseau de neurones est la disjonction du plus petit nombre d'hypothèses de l'ensemble courant dont l'union des ensembles couverts est égale à l'ensemble d'apprentissage. L'algorithme est le suivant :

$M=\{H_0,H_1,\ldots,H_p\}$, où H_i est l'hypothèse couvrant un seul exemple d'apprentissage, l'objet O_i. Nous rappelons qu'une hypothèse appartenant à M est le produit de N attributs complémentés ou non des objets du concept. Nous appelons poids d'une hypothèse H noté *poids*(H) le nombre d'attributs non complémentés entrant dans son expression booléenne.

Début :

E ← M

- Créer les files associées à chaque hypothèse (la file d'une hypothèse contient les numéros des hypothèses dont elle est une généralisation. Au départ la file de chaque hypothèse contient uniquement le numéro de cette hypothèse).

Etiq :

- Trier les éléments de M par ordre croissant de poids et attribuer à chaque hypothèse un numéro
- E ← M
- M' ← ∅ (l'ensemble vide)
- Pour i allant de 1 à cardinal (M) fait
  - Pour j allant de i+1 à cardinal(M) fait
    - Si *poids* (H_i)+1=*poids*(H_j) alors
      - Calculer h ← *généralisé*(H_i,H_j)
      - Si h est définie et h ∉ M alors
        - M' ← M' ∪ {h}
        - file(h) ← file(H_i) ∪ file(H_j)
      - fsi
    - fsi
  - fpour

fpour

- supprimer de M toute hypothèse dont le numéro apparaît dans la file d'attente d'un élément de M'.
- M' ← M' ∪ M
- si M=M' alors aller à fin fsi
- M ← M'
- Aller à etiq

Fin : R ← ∅

Label :

4) Choisir l'hypothèse h qui a la plus grande file et supprimer la de M
5) $R \leftarrow R \cup [h]$
6) Supprimer de la file des hypothèses non retenues tous les éléments de la file de l'hypothèse h (l'hypothèse h est plus général que les hypothèses de M dont les numéros sont supprimés)
7) Supprimer de M toutes les hypothèses dont les files sont devenues vides
8) Si M est non vide aller à label
9) L'hypothèse finale est la disjonction des éléments de R

Remarque 1. Au pire des cas, le réseau obtenu par notre méthode est égal au réseau initial ayant autant de neurones cachés que d'exemples à apprendre.

Remarque 2. L'algorithme peut s'étendre au cas où l'espace de représentation n'est pas binaire mais reste numérique. Il suffit pour cela de redéfinir la fonction *poids*(H). On pourrait par exemple choisir *poids*(H)= nombre d'attributs dont la valeur est supérieure à un certain seuil. Dans la fonction *généralisé*(H1,H2), un attribut sera supprimé si sa valeur est supérieure au seuil dans H1 et est inférieure au seuil dans H2. Les expérimentations sont en cours avec ces redéfinitions de *poids* et *généralisé*.

3. Illustration de l'algorithme

Considérons un concept complètement décrit par une fonction F de cinq attributs binaires a,b,c,d,e et supposons que l'on soit en possession des seize exemples positifs listés ci-dessous après conversion en base 10 (on suppose que a est le bit de poids fort et e est le bit de poids faible) : 26, 28, 30, 29, 27, 23, 15, 31, 7, 11, 13, 14, 19, 21, 25 et 22. Les huit premiers vont servir pour l'apprentissage et donc pour la construction du réseau. Les huit derniers vont servir pour la validation.

La fonction majorité est une fonction booléenne dont la valeur est toujours égale à celle de la majorité des attributs. Une liste d'exemples négatifs de la fonction majorité est : 0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 16, 17, 18, 20 et 24. Ces contre exemples vont nous permettre de vérifier si notre modèle est correct, c'est-à-dire vérifier s'il rejette tous les exemples négatifs.

Les hypothèses couvrant un seul exemple positif déduites des huit premiers exemples ci-dessus sont (Nous représentons les négations des attributs par des lettres majuscules ; les compléments de a,b,c,d et e sont respectivement A,B,C,D et E):

H0=a∧b∧C∧d∧E, H1=a∧b∧c∧D∧E, H2=a∧b∧c∧d∧E,
H3=a∧b∧c∧D∧e,
H4=a∧b∧C∧d∧e, H5=a∧B∧c∧d∧e, H6=A∧b∧c∧d∧e, H7=a∧b∧c∧d∧e,

Où ∧ représente la conjonction, c'est-à-dire le produit booléen. L'hypothèse plus générale couvrant tous les huit exemples d'apprentissage est :

H= H0∨H1∨H2∨H3∨H4∨H5∨H6∨H7

où ∨ représente la disjonction, c'est-à-dire la somme booléenne. L'hypothèse simplifiée produite par l'algorithme de ce papier est :

H'=a∧b∧d ∨ a∧b∧c ∨ a∧c∧d∧e ∨ b∧c∧d∧e

Trois réseaux de neurones ont été construits pour apprendre les concepts décrits par les hypothèses H et H'. Le premier est un réseau complètement connecté dont les poids de connexion initiaux sont choisis de façon aléatoires. Le deuxième réseau est initialisé par l'approche KBANN, en utilisant la règle H. Le troisième est initialisé avec l'approche KBANN aussi, mais en utilisant l'hypothèse H' plus compacte produite par notre algorithme.

Après l'initialisation de ces trois réseaux de neurones, l'apprentissage s'est fait en utilisant l'algorithme classique de rétro-propagation de l'erreur. Le tableau ci-dessous reproduit les résultats obtenus :

Réseau numéro	1	2	3
Taux de reconnaissance des exemples de l'ensemble d'apprentissage	100%	100%	100%
Taux de reconnaissance des exemples de l'ensemble de validation	100%	100%	100%
Taux de reconnaissance des exemples négatifs	37%	93%	93%
Nombre de neurones cachés	8	8	4
Nombre de connexions avec les neurones cachés	48	48	18

Table 1. ***Comparaison des trois réseaux de neurones décrits ci-dessus***

En terme de la simplicité de la topologie, le réseau numéro 3 est le meilleur. En ce qui concerne les taux de reconnaissance lors de l'apprentissage, de la validation et le test des contre exemples, il est encore plus bon que le premier et a des performances identiques à celles du deuxième.

Commentaire . Les bonnes qualités de tous ces trois réseaux lors de l'apprentissage peuvent sans doute s'expliquer par le fait que la fonction majorité est linéairement séparable. En effet, elle peut se définir ainsi : $f(x_1,\ldots,x_n)= 1$ si la somme des $(x_i)_{1\le i\le n}$ est supérieure ou égale à n/2 et égale à 0 dans le cas contraire. L'hyperplan $x_1+x_2+\ldots+x_n=n/2$ est dont une séparatrice linéaire entre les exemples positifs et négatifs de ce problème. Par ailleurs, l'apprentissage s'étant fait avec uniquement des exemples positifs, on peut penser que le réseau de la figure 1 a fait un apprentissage par cœur des propriétés des exemples et a tendance à tout assimiler à ces exemples. C'est ce qui explique le fait qu'il ne rejette pas suffisamment de contre exemples.

4. Conclusion

Nous avons proposé dans ce papier une approche pour la construction de réseaux de neurones en vue de l'apprentissage de concepts dont l'espace de représentation est binaire. Cette approche présente au moins deux avantages : la topologie du réseau obtenue est plus simple. Elle est à trois couches, et les connexions entre neurones ne sont pas complètes. Le deuxième avantage découle du premier : le temps d'utilisation dans la phase de reconnaissance est réduit par rapport à celui qu'on aurait pour le même problème avec des architectures complètement connectées. Par ailleurs, le fait que les poids initiaux et les connexions soient déduits à l'aide de la méthode KBANN, fait que les réseaux obtenus héritent des avantages qu'on a à utiliser cette technique.

Le problème du choix des exemples positifs à utiliser reste à résoudre. En effet l'hypothèse initiale couvrant tous les exemples positifs est fortement liée à ces exemples.

Nous effectuerons d'autres expérimentations en utilisant des techniques de validation couramment utilisées en apprentissage de concepts, telle que la validation croisée [9] afin de caractériser les classes de problèmes où cette nouvelle approche présente des avantages.

5. Bibliographie.

[1] W.S. McCULLOH; W. PITTS: "A logical calculus of the ideas immanent in nervous activity", *Bulletin of Mathematical Biophysics*, 5 (1943), pp115-133.

[2] G. Dreyfus: "Les réseaux de neurones: une technique opérationnelle pour le traitement des données industrielle, économiques et financières », *Mesures*, n° 699, 1997.

[3] G. G. TOWELL ; J. W. SHAVLIK; M. O. NOORDEWIER : "Refinement of Approximately correct domain theories by Knowledge-based neural networks", In Proc. Of the *Eighth National Conf. On Artificial Intelligence*, pp 861-866, Boston, MA, 1990.

[4] M.O. NOORDEWIER; G.G. TOWELL; J.W. SHAVLIK : " Training knowledge-based neural networks to recognize genes in DNA sequences", *Advances in Neural Information Processing Systems*, vol. 3, 1991, M. Kaufmann.

[5] E. FIESLER : "Neural networks topologies", in *Neural Networks handbook*, second edition pp B2.1:1 –B2.8:5

[6] M. B. STINCHCOMBE: "Theoretical considerations for choosing a network topology", in "*Neural Networks Handbook*, second edition, pp B2.10:1-B2.10:5

[7] E. MEPHU NGUIFO :" Une nouvelle approche basée sur le treillis de galois pour l'apprentissage de concepts", *Math. Inf. Sci.hum.* n,° 124, (1993), pp19-38.

[8] A. CORNUEJOLS ; L. MICLET : *Apprentissage Artificiel : Concepts et Algorithmes*, Eyroles, Paris (2002).

[9] T. DIETTERICH : « Approximate Statistical Tests for Comparing Supervised Classification », Research Report, CS Dept, Oregon State University (December 1997).