

.....

Un Nouvel Algorithme d'Apprentissage pour les Réseaux Probabilistes

Vers une Réduction des PNN

Abdelhadi Lotfi

Département Tronc Commun
Institut National des Télécommunication et des Technologies d'Information et de la
Communication INTTIC 31000 ORAN ALGERIE
alotfi@ito.dz

Redouane Tlemsani

Département de transmission
Institut National des Télécommunication et des Technologies d'Information et de la
Communication INTTIC 31000 ORAN ALGERIE
rtlemsani@ito.dz

Abdelkader Benyettou

Département d'Informatique
Université des Sciences et de la Technologie d'Oran mohamed Boudiaf USTO-MB 31000
ORAN ALGERIE
ae.k.benyettou@univ-usto.dz

.....

RÉSUMÉ. Parmi les différents types de réseaux de neurones, les réseaux probabilistes représentent une classe de réseaux très utilisés dans les problèmes de classification. Dans ce papier, un nouvel algorithme d'apprentissage pour les réseaux de neurones probabilistes (PNN) est présenté. L'algorithme proposé se concentre sur l'un des principaux inconvénients des réseaux de neurones probabilistes qui est la taille du réseau après apprentissage. Avec un algorithme d'apprentissages modifiés, le nombre de neurones cachés est réduit après apprentissage au nombre des échantillons les plus représentatifs de l'ensemble d'apprentissage. L'architecture du réseau n'est pas affectée par cette modification et ainsi on garde les principaux avantages des PNNs. Les performances du nouveau réseau sont comparées aux performances des réseaux probabilistes standards pour différentes bases de données de référence. Les résultats montrent un gain important dans la taille du réseau, la vitesse de classification et la généralisation du réseau.

ABSTRACT. Among the different types of neural networks, probabilistic networks represent a class of networks widely used in classification problems. In this paper, a new training algorithm for

probabilistic neural network (PNN) is presented. The proposed algorithm focuses on one of the main disadvantages of probabilistic neural networks which is the size of the network after training. With a modified training algorithm, the number of hidden neurons is reduced after training to the number of the most representative samples from the training set. The network architecture is not affected by this change and so we keep the main advantages of PNNs. Performance of the new network is compared to standard PNNs for various referenced databases. The results show a significant gain in network size, speed of classification and generalization of the network.

MOTS-CLÉS : apprentissage machine, réseaux de neurones probabilistes, optimisation, reconnaissance des formes.

KEYWORDS: machine learning, probabilistic neural networks, optimization, pattern recognition.

1. Introduction

Dans la reconnaissance des formes, la classification est l'un des problèmes majeurs. Le réseau de neurones probabiliste (PNN) a été introduit par D. F. Specht dans son article [1]. En raison de leurs propriétés pour les problèmes de classification, ils sont vite devenus une référence dans le domaine de classification neuronale. Les PNNs offrent de nombreux avantages: ils ne souffrent pas du problème des minima locaux [2] comme pour les MLP par exemple, l'apprentissage est très rapide puisque le réseau est créé après un seul passage sur l'ensemble d'apprentissage [1] [3], ils peuvent être utilisés interactivement et le principe lui-même a une base mathématique très solide [4]. D'autre part, il ya quelques inconvénients importants: le nombre de neurones cachés est égal au nombre d'échantillons de l'apprentissage. Cela peut être très restrictif pour certains problèmes avec un ensemble apprentissage très grand qui contient dans la majorité du temps beaucoup de redondances. Un autre problème est le choix du paramètre de lissage qui peut affecter sérieusement la généralisation du réseau [5].

Beaucoup de travail a été fait pour résoudre le problème de généralisation ainsi que celui du choix des paramètres dans les réseaux de neurones probabilistes [7]. M. Kim dans son article [6] a proposé une nouvelle architecture pour améliorer la généralisation du PNN standard. Dans [8], un algorithme LVQ a été utilisé pour l'apprentissage d'un PNN afin de rendre la taille du réseau plus petite. D'autres exemples de variantes PNNs sont le PNN floue [7] et l'optimisation stochastique pour le choix des paramètres du PNN [9]. Toutes ces approches ne permettent qu'une résolution partielle du problème et soulève en général un autre problème. En fait, en modifiant l'algorithme apprentissage de certaines façons, le réseau perd certains de ses atouts majeurs. Par exemple, RKPNNs (PNNs avec noyau rotationnel, voir réf. [10]) ont des bonnes qualités de généralisation que l'on prétend être comparé aux SVM (dans le cas d'un nombre réduit de classes [11]) mais l'apprentissage est très lent, même pour les petits ensembles d'apprentissage et le réseau ne supporte pas l'ajout de nouvelles classes et / ou des échantillons, dans ce cas tout le processus d'apprentissage doit être repris avec le

nouvel ensemble. L'application de ce type de réseaux d'une manière interactive semble impossible. L'algorithme d'apprentissage présenté dans ce papier réduit le nombre de neurones cachés, sans affecter l'architecture du réseau, ceci signifie que l'ajout de nouveaux échantillons ou de classes est possible à tout moment et sans refaire l'apprentissage du début. Cet algorithme d'apprentissage peut être utilisé pour un large éventail de problèmes tels que la biométrie, la reconnaissance de caractères et la reconnaissance vocale.

2. Réseaux de Neurones Probabilistes

2.1. Principe

Contrairement aux réseaux MLP, les réseaux à fonction radiale RBF (y compris les réseaux de neurones probabilistes) utilisent des fonctions radiales plutôt que des fonctions d'activation sigmoïdes pour construire une fonction de décision locale centrée sur un sous-ensemble de l'espace d'entrée [8]. La fonction de décision globale est la somme de toutes les fonctions locales [6] [12]. De cette façon, le problème des minima locaux est résolu.

Dans un contexte de classification des formes, chaque vecteur observé x (x est vecteur de dimension d) est placé dans une des classes de cluster C_i , $i=1, 2, \dots, m$ prédéfini; où m est le nombre de classes possibles, dans lequel x peut appartenir. L'efficacité du classificateur est limitée par la dimension du vecteur x d'entrée et le nombre de classes possibles m . Le classificateur de Bayes met en œuvre le modèle de la règle de Bayes de probabilité conditionnelle que la probabilité $P(C_i/x)$ pour x d'appartenir à la classe C_i . Cette probabilité est donnée par:

$$P(C_i|x) = \frac{P(x|C_i)P(C_i)}{\sum_{j=1}^m P(x|C_j)P(C_j)} \quad (1)$$

Où $P(C_i/x)$ est la fonction de densité de probabilité conditionnelle de x sachant C_i et $P(C_j)$ est la probabilité de choisir un échantillon de la classe C_j .

Un vecteur en entrée x est classifié comme appartenant à C_i si :

$$P(C_i|x) > P(C_j|x); \quad \forall j=1, 2, \dots, m; j \neq i \quad (2)$$

L'estimation de ces probabilités à partir de l'ensemble d'apprentissage est réalisé en utilisant la technique de fenêtrage de Parzen pour l'estimation des fonctions de densités (pdf) [6]. L'estimateur utilisé pour les réseaux PNN est :

$$f_A(x) = \frac{1}{2\pi^{p/2}} \frac{1}{\sigma^p} \frac{1}{m} \sum_{i=1}^m \exp \left[-\frac{(X - X_{ai})'(X - X_{ai})}{2\sigma^2} \right] \quad (3)$$

X_{ai} est le $i^{\text{ème}}$ échantillon appartenant à la classe C_A et σ est un paramètre de lissage.

2.2 Architecture d'un réseau probabiliste

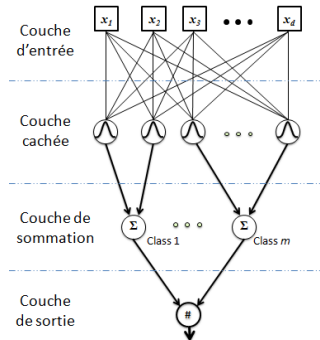


Figure 1. Architecture générale d'un réseau probabiliste.

2.3. Algorithme d'apprentissage standard

Pour l'apprentissage d'un réseau probabiliste, il faut d'abord que chaque vecteur x en entrée soit normalisé de façon que :

$$\sum_{i=1}^n x_i^2 = 1 \quad (9)$$

Le nombre de neurones d'entrée est égal au nombre des paramètres (variables) de la forme à classifier [3]. Pour chaque échantillon de la base d'apprentissage, on crée un neurone dans la couche cachée (couche de motifs) avec les connexions correspondantes avec les neurones d'entrée de façon que $w_k = x_k$ pour $k=1, 2, \dots, n$. Ensuite une seule connexion est créée vers le neurone de la couche de classification correspondant à la classe de l'échantillon [10]. Ce processus est illustré dans l'algorithme suivant :

Début

Initialisation

$j = 0$; $n =$ nombre d'échantillons ;

Faire

$j \leftarrow j+1$;

Normalisation :

$$x_{jk} \leftarrow \frac{x_{jk}}{\sqrt{\sum_i x_{jk}^2}} ;$$

Apprentissage des poids : $w_{jk} \leftarrow x_{jk}$;

Si $x \in w_j$ **alors** $a_{ic} \leftarrow 1$;

Jusqu'à $j = n$;

Fin

3. Algorithme d'apprentissage modifié

Afin d'exclure les neurones qui sont assez similaires dans leurs réponses (qui ont la même zone d'activation), un apprentissage standard du réseau est réalisé avec

l'algorithme standard d'apprentissage des PNNs présenté dans la section 2.3. Notez ici qu'avec cet algorithme, nous pouvons ajouter ou supprimer des unités cachées (même pour des nouvelles classes) facilement en ajoutant des neurones cachés et leurs connexions respectives. Après la création de ce réseau, un autre passage est nécessaire pour supprimer les neurones de faible importance. Le nouvel algorithme devient :

1^{ère} étape (Construction du PNN initial) : Utiliser l'algorithme standard d'apprentissage des réseaux probabilistes présenté dans la section 2.2 pour construire un premier PNN initial. Dans ce réseau, le nombre de neurones cachés est égal au nombre d'échantillons d'apprentissage.

2^{ème} étape (Réduction de la taille du réseau) : Le réseau initial de la première étape est transformé en un réseau réduit PNN* (voire figure 2).

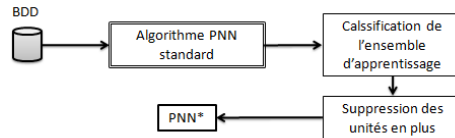


Figure 2. Algorithme d'apprentissage modifié

Début

Initialisation

$k = 0$; x = premier échantillon d'apprentissage;

Faire

$k \leftarrow k+1$;

Suppression du $k^{\text{ème}}$ neurone de la couche cachée ;

Classifier le $k^{\text{ème}}$ échantillon dans le nouveau réseau (PNN*) ;

Si la classification est bonne **alors** Garder le nouveau réseau ;

Sinon Reprendre le PNN de l'étape $k-1$;

Jusqu'à $k = n$;

Fin

4. Expériences, Résultats et Analyses

4.1 Test sur un ensemble bidimensionnel

Dans cette expérience, trois classes sont représentées par 39 points de données. Deux réseaux sont créés pour séparer le plan ($100 * 100$ points) en utilisant les 39 échantillons étiquetés. Chaque classe est représentée par une couleur (noir, gris et blanc).

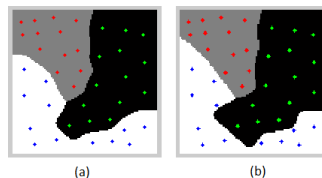


Figure 3. Classification de ($100 * 100$) points de données avec un algorithme PNN standard (a) et un algorithme modifié (b).

Dans la figure 2 (a), le nombre de neurones cachés est égal au nombre d'échantillons d'apprentissage. Qui est 39 neurones cachés. Dans la figure 2 (b), le nombre de neurones cachés utilisés pour la classification est 12 neurones. Nous pouvons voir que la classification est réduite à environ 30% de la taille d'un PNN standard. C'est parce que les points de données adjacents sont simplement remplacés par un seul neurone dans le réseau.

4.2 Description des bases de données

Pour illustrer l'efficacité de l'algorithme proposé, nous avons utilisé différents ensembles de données depuis le dépôt de bases de données UCI. Les ensembles de données sont choisis avec des propriétés différentes (nombre de classes, nombre d'entrées ...) pour tester le nouvel algorithme pour un large éventail de situations. On a utilisé les bases de données suivantes :

- Base de données de reconnaissance des lettres (Letter)
- Base de données d'identification de verre (Glass)
- Base de données balance (Balance)
- Base de données cancer du sein (Breast)
- Base de données d'iris (Iris)

Base de données	Taille des entrées	Nombre d'échantillons
Letter	16	20000
Balance	4	625
Breast	9	286
Glass	9	214
Iris	4	150

Tableau 1. *Propriétés des ensembles de données utilisés dans les expériences.*

4.3 Résultats

Les deux algorithmes d'apprentissage (standard et modifié) sont appliqués sur les ensembles de données séparément. Nous nous sommes concentrés sur les performances du réseau (taux de classification) et le nombre de neurones créés dans la couche cachée. Environ 2 / 3 des échantillons sont utilisés pour l'entraînement et le reste (1 / 3) des échantillons sont utilisés pour le test. Les échantillons de l'ensemble de test sont nouveaux (observés pour la première fois) pour le réseau et ils ont été choisis au hasard de la base de données entière pour rendre la situation aussi réaliste que possible. Il n'est pas nécessaire de présenter le taux de classification pour la base d'apprentissage, puisque les deux réseaux peuvent classer tous les échantillons avec succès.

Base de données	# échantillons	Algorithme standard		Nouvel algorithme	
		% correct	% erreur	% correct	% erreur
Letter	20000	95.96	4.04	93.06	6.94
Balance	625	94.87	5.12	91.66	8.33
Breast	286	96.18	3.81	95.03	4.96
Glass	214	76.59	23.40	78.72	21.27
Iris	150	96.42	3.57	100.00	0.00

Table 2. Taux de classification pour toutes les BDD.

Base de données	# échantillons d'apprentissage	Algorithme standard		Nouvel algorithme	
		# neurones cachés	t (s)	# neurones cachés	t (s)
Letter	15000	15000	9472.75	2433	1443.74
Balance	469	469	24.8078	171	8.9744
Breast	437	437	42.1040	50	4.8678
Glass	167	167	2.7767	91	1.5131
Iris	85	85	0.6776	36	0.2886

Table3. Performance en terme taille du réseau (nombre de neurones cachés) et temps nécessaire pour la classification.

4.4 Discussion

Les résultats du tableau 2 montrent que pour toutes les bases de données, les performances de l'algorithme standard et du nouveau sont très proches. Par exemple, les performances pour les bases de données : Cancer, Lettre et Balance sont mieux avec le PNN standard. D'autre part, les performances pour la base de données de Glass et Iris sont meilleures avec le nouvel algorithme. Peu importe le nombre d'échantillons ou la taille de l'espace d'entrée, les deux réseaux ont tout à fait le même comportement et les résultats sont presque similaires.

Le tableau 3 donne des détails importants sur la taille de chaque réseau après apprentissage (nombre d'unités cachées créées) et le temps nécessaire pour la classification de tous les échantillons de test en secondes (t(s)). Nous pouvons constater que les réseaux créés avec le nouvel algorithme sont tous plus petits que ceux créés avec un algorithme standard d'apprentissage à PNN. Pour la reconnaissance des lettres, le nouveau réseau est 6 fois plus petit en taille avec presque les mêmes performances. Pour la base de données d'identification de verre (qui est une base de données avec beaucoup de perturbations), la taille est réduite à 54% avec un taux de classification meilleur pour le nouvel algorithme. La réduction pour la base de données Balance et Iris est d'environ 36% et 42% respectivement, avec de meilleures performances dans le cas de la base de données de l'iris. Dans le cas de la base de données du cancer du sein (Brest), il y a une grande réduction dans le nombre d'unités cachées. En effet, seulement 11% des unités cachées ont été créés.

5. Conclusion

L'algorithme de construction de réseaux probabilistes réduits en taille présenté dans ce papier donne des taux de classification similaires à un algorithme d'apprentissage standard de PNN pour la plupart des cas. Le nombre de neurones cachés dans la deuxième couche est, cependant, très réduit et dépend essentiellement de la nature de la base de données. Les bases de données avec beaucoup de redondances (grandes bases) ont besoin de moins d'unités cachées pour représenter tous les échantillons d'apprentissage. En réduisant le nombre de neurones cachés, la rapidité de classification augmente. L'algorithme proposé présente une solution au problème de la taille du PNN et conserve la même architecture et ainsi les mêmes particularités et avantages qu'un PNN standard. Pour ajouter de nouvelles classes au nouveau PNN, le même processus peut être exécuté pour l'apprentissage des nouveaux échantillons seulement.

6. Bibliographie

- [1] D. F. Specht, « Probabilistic neural networks », *Neural networks*, vol. 3, no. 1, p. 109–118, 1990.
- [2] Y.-qun Deng et P.-ming Wang, « Predicting the shrinkage of thermal insulation mortar by probabilistic neural networks », *Journal of Zhejiang University SCIENCE A*, vol. 11, no. 3, p. 212–222, févr. 2010.
- [3] M. Bazarghan et R. Gupta, « Automated classification of sloan digital sky survey (SDSS) stellar spectra using artificial neural networks », *Astrophysics and Space Science*, vol. 315, no. 1-4, p. 201–210, mai 2008.
- [4] C. M. Bishop, *Neural networks for pattern recognition*. Oxford University Press, 1995.
- [5] A. Lotfi, K. Mezzoug, et A. Benyettou, « Rotated Kernel Neural Networks for Radar Target Detection in Background Noise », *Journal of Applied Sciences*, vol. 10, no. 13, p. 1331–1335, déc. 2010.
- [6] M. W. Kim et M. Arozullah, « Generalized probabilistic neural network based classifiers », in *Neural Networks, 1992. IJCNN., International Joint Conference on, 2002*, vol. 3, p. 648–653.
- [7] V. Georgiou, P. Alevizos, et M. Vrahatis, « Fuzzy Evolutionary Probabilistic Neural Networks », *Artificial Neural Networks in Pattern Recognition*, p. 113–124, 2008.
- [8] P. Burrascano, « Learning vector quantization for the probabilistic neural network », *IEEE Transactions on Neural Networks / a Publication of the IEEE Neural Networks Council*, vol. 2, no. 4, p. 458–461, 1991.
- [9] I. De Falco, A. Della Cioppa, et E. Tarantino, « Facing classification problems with Particle Swarm Optimization », *Applied Soft Computing*, vol. 7, p. 652–658, juin 2007.
- [10] I. Gallese et J. Castellanos, « A rotated kernel probabilistic neural network (RKPN) for multi-class classification », in *Proceedings of the Artificial and natural neural networks 7th international conference on Computational methods in neural modeling - Volume 1*, Berlin, Heidelberg, 2003, p. 152–157.
- [11] I. Gallese et J. Castellanos, « A rotated kernel probabilistic neural network (RKPN) for multi-class classification », *Computational Methods in Neural Modeling*, p. 1040–1040, 2003.
- [12] D. K. Kim, D. H. Kim, S. K. Chang, et S. K. Chang, « Modified probabilistic neural network considering heterogeneous probabilistic density functions in the design of breakwater », *KSCE Journal of Civil Engineering*, vol. 11, no. 2, p. 65–71, mars 2007.