

Comparaison de méthodes de classification réseau RBF, MLP et RVFLNN¹

Mrabti Fatima²

Seridi Hamid³

Résumé

Dans ce papier, nous présentons une étude comparative de trois modèles neuronaux, comme étant des outils de décision destinés à la classification d'objets dans le domaine de la reconnaissance des formes, tels que : le réseau RBF (Radial Basis Functions), le MLP (Multi Layer Perceptron) et le RVFLNN (Random Vector Functional Link Neural Network). Le but de ce travail est d'étudier leurs architectures, leurs algorithmes d'apprentissage et d'apercevoir leurs performances de capacité de classification. Pour comparer leurs performances, nous utilisons deux bases de données : Iris et texture. La qualité de décision de chaque classifieur dépend du choix de son architecture et de sa mise en œuvre.

Mots clés – Apprentissage supervisé, apprentissage non supervisé, classification, reconnaissance des formes, le réseau RBF, le MLP, le RVFLNN.

¹ For the paper in Arabic see pages (351-352).

² Faculté de Sciences et d'ingénierie, Département de Génie électrique, Université de Guelma

³ Maître de conférence université de Guelma, Faculté de Sciences et d'ingénierie, Département d'informatique, Université de Guelma

Introduction

Au cours de ces dernières décennies est apparu un ensemble de disciplines fortement interdépendantes, portant sur le traitement de l'information, la théorie de la décision et les méthodes de reconnaissances des formes en l'occurrence les réseaux de neurones. Ces modèles constituent une méthode connexionniste intéressante pour l'analyse et la synthèse des systèmes demandant un traitement parallèle et un processus d'adaptation à un environnement changeant. Leurs champs d'applications est beaucoup élargi et étendu à plusieurs domaines, en particulier dans: la reconnaissance des formes, l'approximation des fonctions, le traitement d'image, la reconnaissance de la parole, la classification... Ainsi ils représentent des outils de calcul très puissants mais avec quelques inconvénients à aviser comme: la mauvaise maîtrise de leur fonctionnement en particulier au niveau de l'initialisation où il n'existe pas de méthodologie pour le choix d'une topologie de réseau adéquate au problème donné. D'autre part, ils sont incapables à expliquer les résultats qu'ils fournissent où ils sont considérés comme des boîtes noires. L'objectif de cet article est de présenter trois architectures des réseaux de neurones et de comparer leurs performances dans le domaine de la classification supervisée.

1. Classification par le réseau RBF

1.1. Architecture

Introduit par Powell et Broomhead [1], le réseau RBF (Radial Basis Functions) fait partie des réseaux de neurones supervisés. Il est constitué de trois couches (figure 1): une couche d'entrée qui retransmet les entrées sans distorsion, une seule couche cachée qui contient les neurones RBF qui sont généralement des gaussiennes [1] [2] [3] et une couche de sortie dont les neurones sont généralement animés par une fonction d'activation linéaire. Chaque couche est complètement connectée à la suivante et il n'y a pas de connexions à l'intérieur d'une même couche.

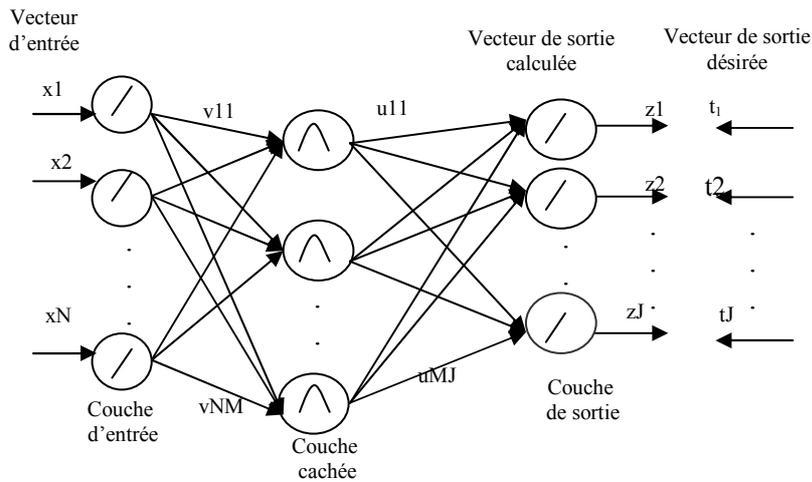


Figure 1. Présentation schématique d'un réseau RBF

Ce réseau est constitué de N neurones d'entrée, M neurones cachés et J neurones de sortie.

La sortie du m^{ième} neurone de la couche cachée est donnée par :

$$y_m^{(q)} = \exp\left[-\frac{\|x^{(q)} - v_m\|^2}{2\sigma_m^2}\right] \tag{1}$$

v_m est le centre du m^{ième} neurone de la couche cachée où du m^{ième} neurone gaussien et σ_m est la largeur du m^{ième} gaussienne.

La sortie du j^{ième} neurone de la couche de sortie est donnée par:

$$z_j^{(q)} = \left(\frac{1}{M}\right) \left[\sum_{(m=1,M)} w_{mj} y_m^{(q)} \right] \tag{2}$$

$m=1, \dots, M$ et $j=1, \dots, J$.

w_{mj} sont les poids reliant la couche cachée à celle de la sortie.

1.2. Algorithme d'apprentissage du réseau RBF

L'apprentissage du réseau RBF a été présenté la première fois par Moody et Darken [2]. Il consiste à régler quatre paramètres principaux: le nombre de neurones dans l'unique couche cachée ou le nombre des gaussiennes, la position des centres de ces gaussiennes, la largeur de ces gaussiennes et les poids de connexions entre les neurones cachés et le(s) neurone(s) de sortie. Le réseau RBF consiste à minimiser l'erreur quadratique totale E calculée entre les sorties obtenues du réseau et celles désirées :

$$E = \sum_{q=1}^Q \sum_{j=1}^J \left(t_j^{(q)} - z_j^{(q)} \right)^2 \quad (3)$$

Pour le réseau RBF, l'ajustement des poids w_{mj} reliant la couche cachée à celle de la sortie est réalisé par la règle de Widrow-Hoff. Il se fait comme suit :

$$w_{mj}^{(i+1)} = w_{mj}^{(i)} + \eta \cdot (t_j - z_j) \cdot y_m \quad (4)$$

t_j est la sortie du $j^{\text{ième}}$ neurone désirée, z_j est la sortie du $j^{\text{ième}}$ neurone calculée, y_m est la sortie du $m^{\text{ième}}$ neurone de la couche cachée et η est le pas d'apprentissage dont sa valeur est comprise entre 0 et 1.

Dans les tâches de classification, le nombre et les centres des gaussiennes sont choisis par des techniques de regroupement de données de type K-means [4]. Les largeurs des gaussiennes sont calculées par la valeur moyenne des distances séparant tous les exemples au centre correspondant. Les K-means [4] [5] sont des méthodes itératives permettant de séparer une série de vecteurs en différents clusters et chaque cluster est représenté par un centre.

1.3. Algorithme des K-means

L'algorithme des K-means se présente comme suit [4] :

- Choisir au hasard le centre de chacun des K clusters.
- Attribuer chaque objet au cluster dont le centre lui est le plus proche.
- Recalculer les positions des nouveaux centres.
- Répéter les étapes 2 et 3 jusqu'à convergence, c'est-à-dire jusqu'à ce qu'un nombre d'itérations maximal soit atteint.

L'algorithme des K-means présente des avantages tels que : la convergence rapide, la facilité de mise en œuvre et la possibilité de traiter des grandes

bases de données. Son inconvénient réside dans la nécessité de la connaissance à priori du nombre de clusters.

2. Classification par le MLP

2.1. Architecture

Un MLP (Multi Layer Perceptron) est composé d'une couche d'entrée, une couche de sortie et une ou plusieurs couches cachées [6] [7]. Les éléments de deux couches adjacentes sont interconnectés par des poids assurant la liaison des différentes couches comme indiqué sur la figure 2.

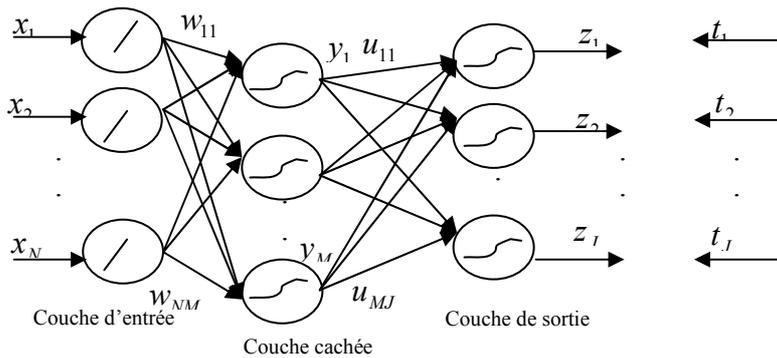


Figure 2. Présentation schématique d'un MLP

w_{nm} représente les poids reliant la couche d'entrée avec la couche cachée ;

u_{mj} représente les poids reliant la couche cachée avec celle de la sortie.

Les composants du vecteur d'entrée X du MLP seront pondérés par les poids w_{nm} et ensuite transmis à la couche cachée selon les équations suivantes :

$$c_m = \sum_{n=1}^N x_n w_{nm} \tag{5}$$

$$y_m = f(c_m) \tag{6}$$

Les sorties de la couche cachée seront aussi pondérées par les poids u_{mj} et transmis ensuite à la couche de sortie selon les équations suivantes :

$$o_j = \sum_{m=1}^M y_m u_{mj} \quad (7)$$

$$t_j = g(o_j) \quad (8)$$

f et g représentent respectivement les fonctions d'activations des neurones de la couche cachée et celle de la de sortie. Pour notre cas elles sont des sigmoïdes.

La fonction sigmoïde est : $f(x) = \frac{1}{1 + e^{-\alpha x}}$ (9)

Sa dérivée est : $f'(x) = \alpha \cdot f(x)[1 - f(x)]$ (10)

2.2. Algorithme d'apprentissage

L'apprentissage consiste à adapter les valeurs des poids afin d'obtenir le comportement désiré du MLP. Pour cela, on se crée une base d'exemples où chaque exemple est constitué d'un vecteur d'entrée et d'un vecteur de sortie approprié. Un algorithme d'apprentissage connu sous le nom d'algorithme de rétropropagation du gradient a été mis en place en 1985 par Rumelhart et Al. [6].

Ainsi l'adaptation des poids u_{mj} qui relie la couche cachée à celle de la sortie est comme suite :

$$u_{mj}^{(i+1)} = u_{mj}^{(i)} + \eta_1 \cdot (t_j - z_j) g'(o_j) y_m \quad (11)$$

La mise à jour des poids w_{nm} reliant la couche d'entrée à celle cachée est comme suite :

$$w_{nm}^{(i+1)} = w_{nm}^{(i)} + \eta_2 \cdot \left\{ \sum_{j=1, i=1}^J (t_j - z_j) g'(o_j) u_{mj} \right\} (f'(c_m))(x_n) \quad (12)$$

η_1 et η_2 sont les pas d'apprentissage.

3. Classification par le RVFLNN

3.1. Architecture

Le RVFLNN (Random Vector Functional Link Neural Network) est un modèle du perceptron [8] développé par Pao et al [9]. Il consiste comme un réseau ayant les sorties de la couche cachée du MLP comme des entrées supplémentaires (figure 3). Le réseau possède donc (N+M) entrées, selon Pao les (M) entrées fournies par la couche cachée forme un ensemble d'entrées optimisées.

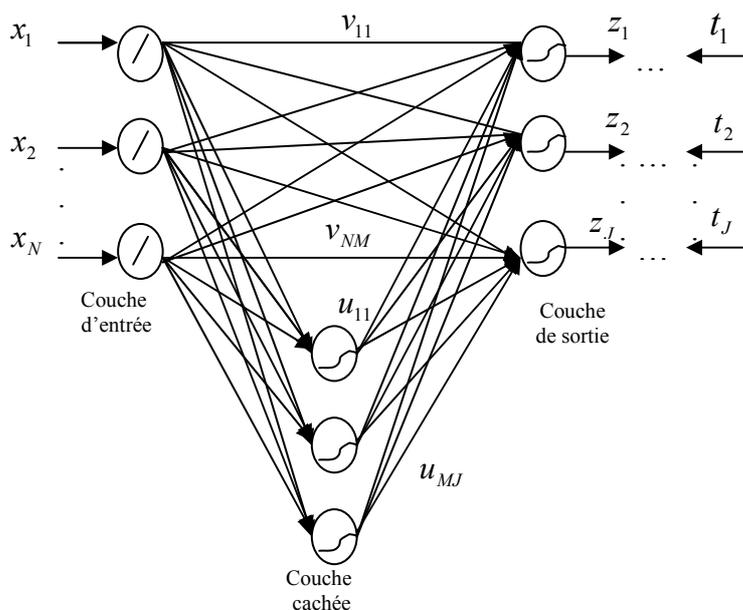


Figure 3. Présentation schématique d'un RVFLNN

La fonction d'activation utilisée pour chaque neurone cachée et de sortie est une sigmoïde.

3.2. Algorithme d'apprentissage du RVFLNN

L'apprentissage du RVFLNN consiste à ajuster seulement les poids v_{nj} reliant la couche d'entrée avec celle de la sortie et les poids u_{mj} reliant la

couche cachée avec celle de la sortie. Les poids W_{nm} reliant la couche d'entrée avec la couche cachée sont non ajustables [9].

La $j^{\text{ième}}$ sortie du réseau est donnée par:

$$z_j = g(o_j) = g\left(\sum_{n=1}^N x_n v_{nj} + \sum_{m=1}^M y_m u_{mj}\right) \quad (13)$$

Ainsi la mise à jour des poids de connexions à l'itération (i+1) reliant la couche cachée avec celle de la sortie est comme suite:

$$u_{mj}^{(i+1)} = u_{mj}^{(i)} + \eta \sum_{q=1}^Q (t_j^{(q)} - z_j^{(q)}) g'(o_j) y_m \quad (14)$$

L'adaptation des poids de connexions reliant la couche d'entrée avec celle de la sortie est :

$$v_{nj}^{(i+1)} = v_{nj}^{(i)} + \eta \sum_{q=1}^Q (t_j^{(q)} - z_j^{(q)}) g'(o_j) x_n \quad (15)$$

η est le pas d'apprentissage.

4. Tests et résultats

Notre travail est réalisé sur un ordinateur de type Pentium 4 et de vitesse de 2.4 GHZ. Les programmes utilisés pour la comparaison de la performance et la capacité de classification des trois modèles neuronaux présentés précédemment: le réseau RBF, le MLP et le RVFLNN sont exécutés en MATLAB en utilisant deux bases de données : Iris (<http://mllearn.ics.uci.edu/databases/iris>) et texture (réalisé par le professeur Noel Bonnet de Leri [10]). L'architecture de chaque réseau neuronal est caractérisée par: une couche d'entrée ayant un nombre de neurone égal au nombre des caractéristiques; une couche cachée dont le nombre de neurone est déterminée expérimentalement et une couche de sortie ayant un nombre de neurone égal au nombre des classes à discriminer et chaque sortie étant dédié à une classe donnée. Pour valider la cohérence et la stabilité des résultats, nous utilisons la méthode de validation croisée (cross-validation). La première base de données utilisée est iris .Elle est constituée de 150 échantillons de fleurs: 135 échantillons pour l'apprentissage et 15 pour le test. Elle est répartie en trois classes. Chaque vecteur est décrit par 4 caractéristiques.

Dans le tableau 1, nous donnons les moyennes statistiques d'apprentissages, de tests et les temps d'apprentissage pour les modèles

neuronaux : réseau RBF, MLP et RVFLNN sur la base de données Iris après l'utilisation de la validation croisée d'ordre 10.

Réseaux	RBF	MLP	RVFLNN
Apprentissage Utilisé	Non supervisé + supervisé	Supervisé	Supervisé
Taux d'apprentissage moyens (%)	97.48	99.18	95.41
Taux de test moyens (%)	98	95.33	94.67
Temps d'apprentissage moyens (s)	1.2016 + 0.3320	9.30	6.130

Tableau 1: Taux et temps de classification moyens obtenus par les réseaux RBF, MLP et RVFLNN

La deuxième base de données utilisée est la texture. Elle est constituée de 800 pixels: 600 pixels pour l'apprentissage et 200 pour le test. Elle est répartie en deux classes. Chaque vecteur est décrit par 8 caractéristiques. Le tableau 2 représente les moyennes statistiques d'apprentissages, de tests et le nombre d'itérations moyens pour les modèles neuronaux : réseau RBF, MLP et RVFLNN sur la base de données texture après l'utilisation de la validation croisée d'ordre 4.

Réseaux	RBF	MLP	RVFLNN
Apprentissage Utilisé	Non supervisé + supervisé	Supervisé	Supervisé
Taux d'apprentissage moyens (%)	99.95	100	99.62
Taux de test moyens (%)	99.37	99.25	99.12
Nombre d'itérations moyen	30+50	700	2500

Tableau 2. Résultats de la classification obtenus par les réseaux RBF, RVFLNN et le MLP

5. Conclusion

Les tests expérimentaux réalisés sur les bases de données : Iris et texture, avec le réseau RBF , le MLP et le RVFLNN, nous permettent de dégager les points suivants : les résultats obtenus par l'ensemble des modèles sont satisfaisants; les taux d'apprentissage fournis par le MLP sont souvent les meilleurs mais ses performances dépendent de l'initialisation des poids; tandis que les taux de test du réseau RBF sont les meilleurs de ceux des autres réseaux neuronaux , d'où le réseau RBF a bien généralisé les nouvelles données. La phase d'apprentissage du réseau RBF est plus rapide que celle des deux autres modèles neuronaux et il converge pour un nombre réduit d'itérations par rapport aux réseaux MLP et RVFLNN par contre l'apprentissage du réseau RVFLNN est simple. L'utilisation de ces modèles neuronaux dans le domaine de la classification s'est avérée intéressante et encourageante, pour cela nous envisageons l'application d'une autre base de donnée plus élargie et aussi l'utilisation d'autres techniques surtout pour améliorer mieux le taux d'apprentissage du réseau RBF.

References

- [1] Broomhead, D.S., Lowe, D. (1988), Multivariate functional interpolation and adaptive networks, *Complex Systems*, vol 2, p.321-355.
- [2] J.Moody and C.J Darken (1989). Fast Learning in Network for Locally Tuned Processing Units, *Neural Computation*, vol.1, p.281-294.
- [3] T.Poggio and F., Girosi (1990),"Network for approximation and learning, "Proc. IEEE, vol 78, no. 9, pp. 1481 -1497.
- [4] E.Forgy (1965). Cluster Analysis of multivariate data: efficiency versus interpretability of classifications, *Biometric*, 21, 768-776.
- [5] Kohonen, T.K., *Self Organization and Associative memory*.Berlin: Springer -Verlag (1989).
- [6] Rumelhart D.E. McClelland, J.L. (1986), *Parallel Distributed Processing, Explorations in the Microstructure of cognition. I & II*, Cambridge: MIT Press.
- [7] Bishop C.M., *Neural Networks for Pattern Recognition*, Clarendon Press OXFORD 1995.
- [8] Rosenblatt F., The perceptron : a probabilistic model for information storage and organization in the brain , *Psycho.Review*,vol.654,386-408,1958.
- [9]Pao, Y.H., *Adaptive Pattern Recognition and Neural Network*, Addison – Wesley, Reading, MA, 1989.
- [10]Borgi, A., *Apprentissage supervise par génération de règles:le système SUCRAGE*, Thèse de l'université de Paris 6 ,1999.

Received, 30-3-2009.