

Chapitre V : Les réseaux de Neurones à base Radiale (RNFRB) ou Radial Basis Function (RBF)

Les réseaux de neurones dits *_Feedforward_* (RNF) et les réseaux de neurones à fonctions radiales de base (RNFRB) constituent deux classes de modèles paramétriques largement utilisées en identification de systèmes non linéaires. En effet, ces réseaux avec une seule couche cachée peuvent approximer n'importe quelle fonction continue ayant un nombre fini de discontinuités sur tout compact (Théorème de Cybenko). Un net regain d'intérêt pour les RNFRB a été constaté durant les années 1990 à 2005 dans divers domaines d'application, tel que le traitement du signal, le contrôle et le diagnostic d'erreurs.

Pour un problème donné, l'utilisation d'un RNFRB conduit généralement à une structure de modèle (nombre d'unités de la couche cachée) moins complexe que celle produite par un RNF. La complexité de calcul induite par leur apprentissage est moindre que celle induite par l'apprentissage des RNF (ex : MPC) grâce à l'existence d'algorithmes hybrides. Les performances d'un tel réseau dépendent, pour un choix de fonctions de base [1], du nombre de fonctions constituant la base de fonctions radiales (nombre d'unités de la couche cachée) et de l'estimation des paramètres du réseau. Le choix du nombre optimal d'unité de la couche cachée et l'estimation des paramètres du réseau sont effectués lors d'une phase d'apprentissage au cours de laquelle un ensemble de paires entrée-sortie expérimentales est utilisé pour permettre aux RNFRB d'acquiescer une relation entrée-sortie non linéaire.

4.1 Architecture du RNFRB

Introduit par Powell et Broomhead (Powell et Broomhead, 199), le réseau RBF (Radial Basis Functions) fait partie des réseaux de neurones supervisés. Il est constitué de trois couches Figure 4.1. une couche d'entrée qui retransmet les entrées sans distorsion, une seule couche cachée qui contient les neurones RBF qui sont généralement des gaussiennes 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 (Mrabti et Serodo, 2009).

Le réseau, Figure 4.1, est constitué de N neurones d'entrée, M neurones cachés et J neurones de sortie. La sortie du *mième* neurone de la couche cachée est donnée par :

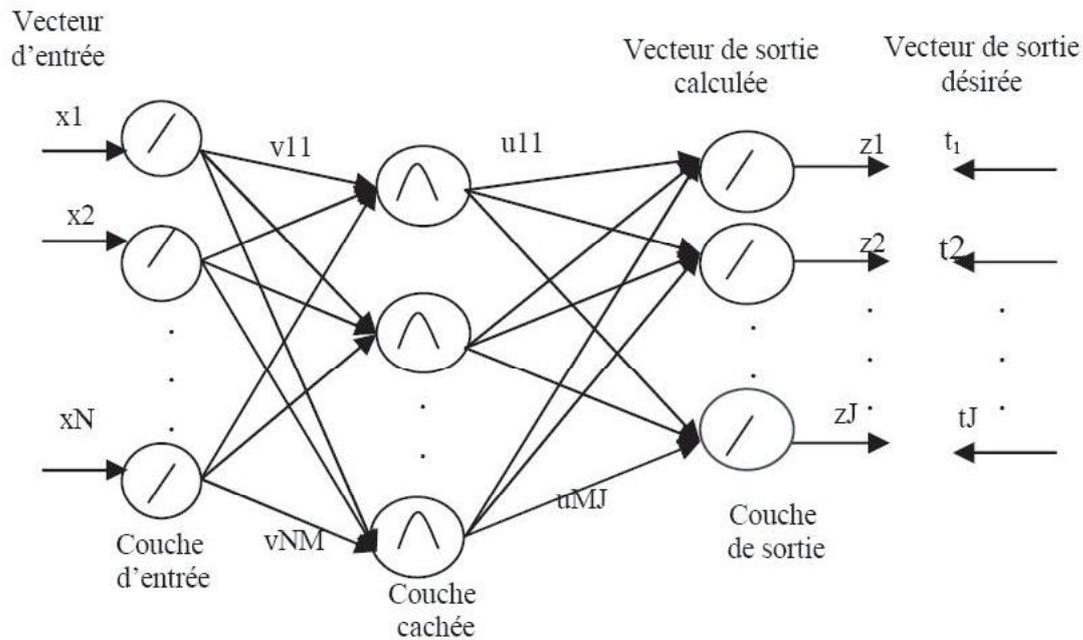


Figure 4.1 Réseaux a fonctions radiales de base

$$y_m^q = \exp \left[-\frac{\|x^q - v_m\|^2}{2\sigma_m^2} \right] \quad (4.1)$$

v_m est le centre du m^{ieme} neurone de la couche cachée ou du m^{ieme} neurone Gaussien et σ_m est la largeur de la m^{ieme} Gaussienne. La sortie du j^{ieme} neurone de la couche de sortie est donnée par :

$$Z_j^q = \left(\frac{1}{m} \right) \left[\sum_{m=1}^M w_{mj} y_m \right] \quad (4.2)$$

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

Un exemple de la valeur de σ sur la forme de la Gausienne est donné par la Figure 4.2.

D'autre part la nature de la séparation introduite par le RNFBR est de type locale par rapport a l'aspect globale de la séparation introduite par un PMC pas exemple. La Figure 4.3, explique les deux types de séparations.

4.2. Algorithme d'apprentissage du réseau RBF

L'apprentissage du réseau RBF a été présenté la première fois par Moody et Darken (Moody et Darken,). 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 EQT calculée entre les sorties obtenues du réseau et celles désirées :

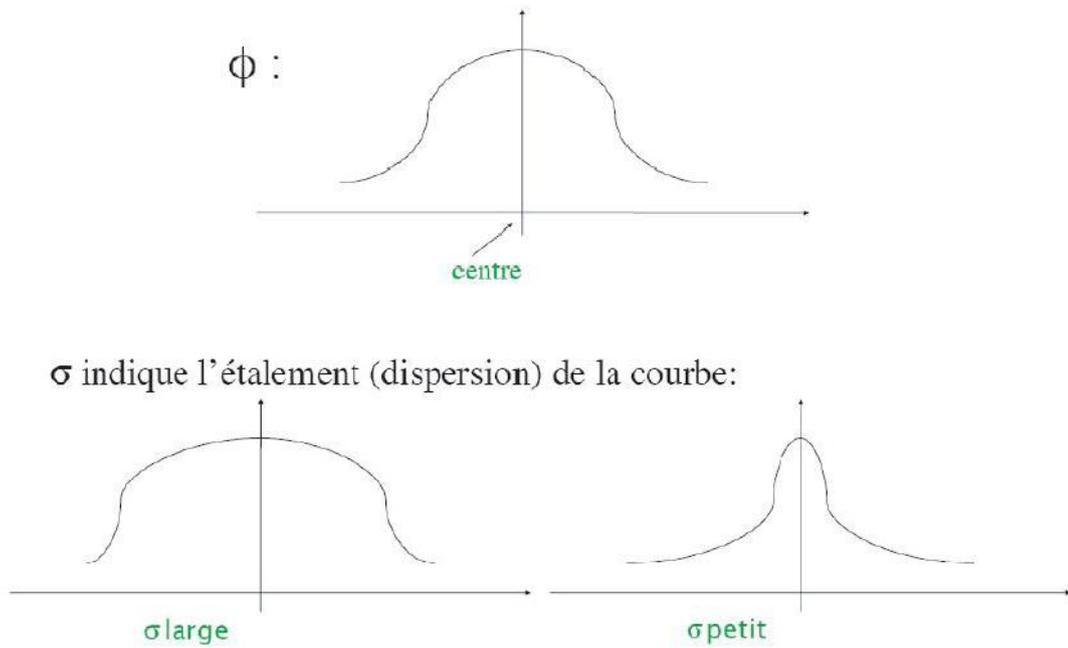


Figure 4.2 fonctions radiales de base

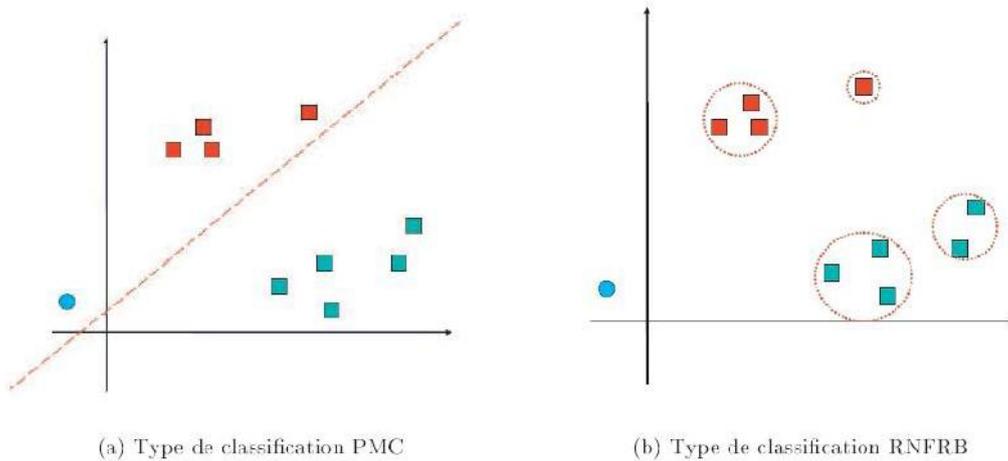


Figure 4.3 – Type de classification

$$EQT = \sum_{q=1}^Q \sum_{j=1}^J (t_j^q - z_j^q)^2 \quad (4.3)$$

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

$$w_{mj}^{i+1} = w_{mj}^i + \mu \cdot (t_j - z_j) y_m \quad (4.4)$$

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

Les paramètres à régler (pour RBF gaussienne) sont :

- les centres v_i
- les dispersions σ_i
- et les poids w_{knj}

L'avantage principal des RBF est qu'il est possible de simplifier l'apprentissage en divisant le travail en trois phases :

- positionnement des centres
- détermination de la largeur des noyaux gaussiens
- et adaptation des poids

Il existe plusieurs stratégies d'apprentissage selon la façon dont les centres sont spécifiés (Ghosh and Nag, 2000) :

- Centres fixes sélectionnés de façon aléatoire : choisir les centres et les dispersions suite à une analyse des données, les poids sont la solution des équations linéaires.
- Centres obtenus par apprentissage non-supervisé : estimer les centres par catégorisation kmeans, les dispersions peuvent être choisies comme la distance moyenne entre les centres voisins, et les poids sont la solution des équations linéaires.
- Centres obtenus par apprentissage supervisé : tous les paramètres sont ajustés pour minimiser une fonction d'erreur

Dans les tâches de classification, le nombre et les centres des gaussiennes sont choisis par des techniques de regroupement de données de type Kmeans. Les largeurs des gaussiennes sont calculées par la valeur moyenne des distances séparant tous les exemples au centre correspondant.

Les K-means sont des méthodes itératives permettant de séparer une série de vecteurs en différents clusters et chaque cluster est représenté par un centre.

L'utilisation des RNFRB nécessite la prise de deux décisions majeures :

- Quel doit être la fonction de transfert ? et si c'est une gaussienne comment en choisir les paramètres ? Il n'y a pas de réponses définitives à ses questions et les solutions restent empiriques. La largeur de la Gaussienne doit généralement être plus large que la distance entre deux points de l'espace mais plus petite que la largeur du diamètre du cluster.
- Combien de centres dois-je utiliser et où les placer ? Généralement on utilise le moins possible de centres, bien sûr en essayant de garder le plus possible de précision. On peut facilement approcher l'erreur zéro pour les données d'apprentissage, mais la validation doit se faire sur des données autres, comme pour l'apprentissage du PCM vu précédemment.

4.2 Algorithme des K-means

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

- 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 oeuvre et la possibilité de traiter des grandes bases de données. Son inconvénient réside dans la nécessité de la connaissance a priori du nombre de clusters.

4.3. Exemple d'application

Dans l'exemple figure 4.4, les 50 points de l'échantillon d'apprentissage, représentés sur les 4 courbes, sont générés comme suit : $x = rand(3 : 2; 1 : 6)$; $y = \sin(x) + rand(0; 0 : 3)$. La courbe $y(x)$ est la « vraie » fonction (sinus), et la courbe « RBF » est la sortie du modèle.

Avec 5 centres, l'approximation est correcte (sauf dans la partie droite où l'on a très peu de points). Avec seulement deux centres, l'erreur est supérieure, tandis qu'avec 10 voire 20 centres on note un overfitting important : la solution oscille et est très sensible au bruit.

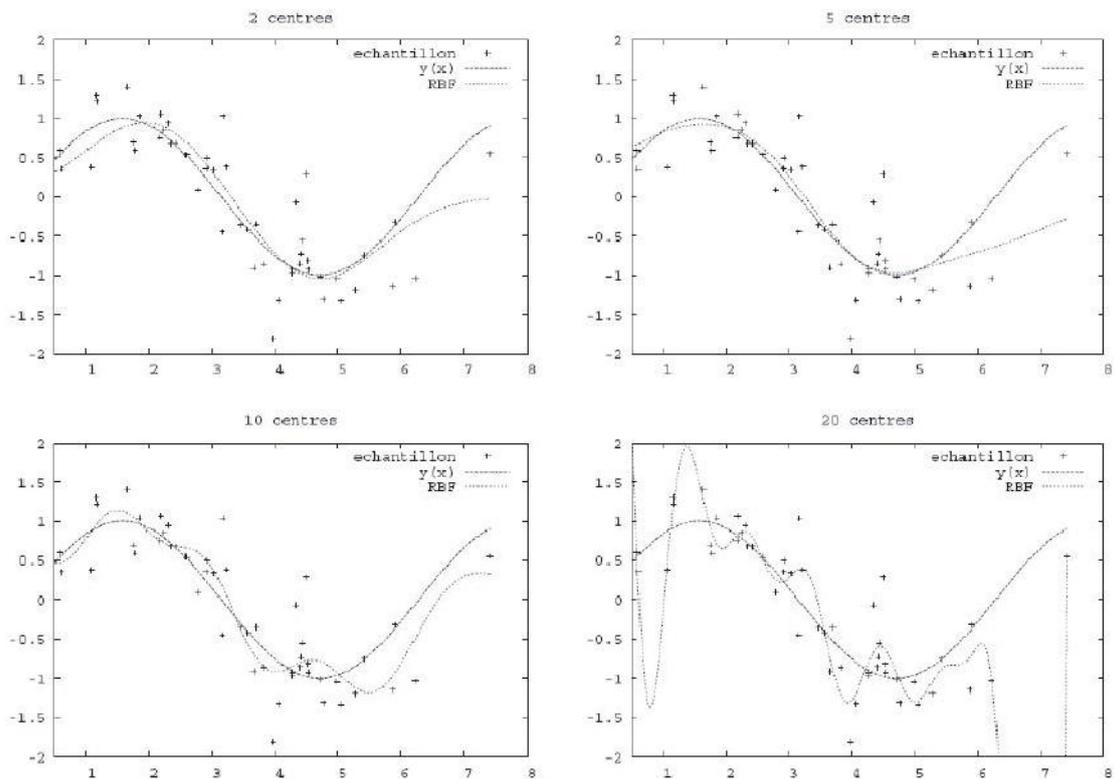


Figure 4.4 Exemple d'approximation de fonctions avec les Réseaux a fonctions radiales de base