

INF7370

Réseaux à fonctions de base radiale

bouguessa.mohamed@uqam.ca

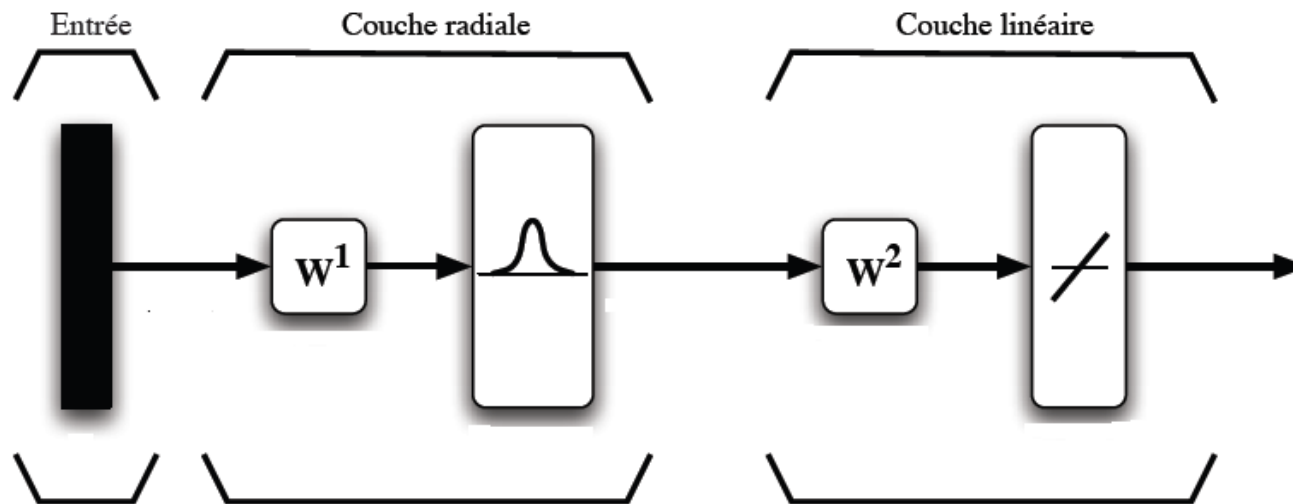


Réseaux à fonction radiale de base

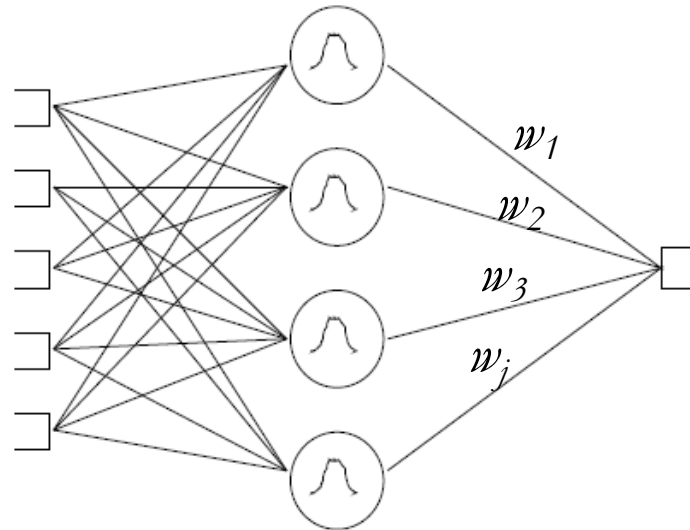
- Ce sont des réseaux à deux couches:
 1. La première couche consiste à faire des opérations non linéaire sur données d'apprentissage à l'aide d'une fonction à base radiale (RBF: Radial Basis Function).
 2. La deuxième couche (la sortie) consiste à faire une combinaison linéaire des résultats de la première couche.

Architecture des réseaux RBF

→ Ces réseaux peuvent se voir comme une extension du modèle linéaire : la sortie est une combinaison linéaire des vecteurs d'entrées, transformés non linéairement.



Réseaux RBF



- À gauche la couche d'entrée : chaque entrée correspond à vecteur d'une données d'apprentissage x_i .
- Au centre les neurone à base de RBF.
- À droite la sortie \hat{y} .
- Pour traiter les problèmes de discrimination à C classes, on peut utiliser C sorties.



Réseaux RBF

- Les deux couches du réseaux réalise la fonction suivante :

$$\hat{y}_k(x_i) = \sum_{j=1}^M w_{kj} \Phi(x_i)$$

- x_i : un élément de l'ensemble d'apprentissage.
- M : le nombre de neurones dans la couche cachée.
- k : indice de la classe ($k = 1, \dots, C$).
- w_{kj} le poids de la connexion entre le neurone j de la couche caché (la couche à base de RFB) et le neurone de sortie associée à la classe k .
- Φ fonction à base radiale.



Réseaux RBF

Un exemple de la fonction de base radiale : la gaussienne

$$\phi_j(x_i) = \exp\left(-\frac{(x_i - \mu_j)^T Q_j^{-1} (x_i - \mu_j)}{2}\right)^2$$

μ_j le centre

Q_j matrice de covariance

$$Q_j = \frac{1}{n_i} \sum_{i=1}^{n_i} (x_i - \mu_j)(x_i - \mu_j)^T$$

→ Chaque neurone de la couche caché représente une gaussienne qui contient n_i élément avec μ comme centre et Q comme matrice de covariance.



Apprentissage des réseaux RBF

Après avoir initialiser les poids w_{kj} des connections entre la couche cachée et la couche de sortie, on doit effectuer les opérations suivantes :

1. Identification des paramètres μ_j et Q_j des fonctions Φ_j
 - On va regrouper les données d'apprentissage de tel sorte que les données proches les unes des autres seront placée dans un même cluster (groupe) → dans chaque groupe les données doivent êtres le plus similaire possible alors que les données qui appartiennent aux différents groupes doivent être les plus dissimilaire possible.
 - Pour réaliser cette tâche, on peut utiliser un algorithme d'apprentissage non supervisé comme le k-means



Apprentissage des réseaux RBF

1. Identification des paramètres μ_j et Q_j des fonctions Φ_j (suite)
 - Le nombre de groupe (cluster) identifié = M , le nombre de neurone dans le couche caché.
 - Pour chaque groupe identifié : on calcule le centre μ_j et la matrice de covariance Q_j



Apprentissage des réseaux RBF

2. Une fois les paramètres μ_j et Q_j sont déterminés, on calcule la sortie \hat{y}_i pour chaque données d'apprentissage dans $S = \{x_1, x_2, \dots, x_n\}$ avec la formule

$$\hat{y}_k(x_i) = \sum_{j=1}^M w_{kj} \Phi(x_i)$$

3. Mise à jour des poids : la mise à jour peut être effectué comme dans le cas du perceptron (réseaux monocouche).