

Projet RNA :

Réseau à fonction de base radial

Algorithme

RANDRENARIZO Tsiry

Info 05

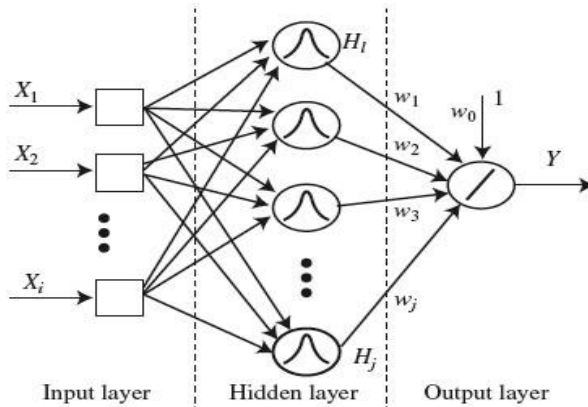
INTRODUCTION

Dans le milieu de réseau de neurones artificiels, les « réseau à fonction de base radial » sont parmi les plus intéressants, avec le « perceptron multi couches », il fait partie des mieux connus, des plus intelligibles et des plus utilisés évidemment. C'est un réseau à modèles connexionnistes, Il prend des valeurs d'entrée et génère des valeurs de sorties.

Son avantage principal est qu'il est plus rapide et souvent plus efficace. Le champ d'utilisation des réseaux RBF est très varié. A l'origine, ce modèle a été conçu surtout pour des applications de classification, cependant ces réseaux ont été utilisés avec succès dans des domaines variés telle que l'approximation des fonctions, la reconnaissance de la parole, la reconnaissance de la parole, et d'autres encore.

A) Structure

La structure générale du réseau est comme suit :



Il y a les entrées de données (X), des couches cachées (H) et un ou plusieurs sorties(Y). Chaque entrée est reliée à chaque couche cachée, ensuite les couches cachées font le traitement des données pour au final être relié à chaque sortie.

B) Les paramètres :

Le réseau radial utilise le jeu de configuration suivant pour faire le traitement des données

1. Le centre de gravité

En premier lieu, il y a « le centre de gravité », c'est une matrice dont le nombre de ligne correspond au nombre de nœuds cachés et le nombre de colonnes de chaque ligne correspond au nombre d'entrées.

2. L'écart-types

Le second paramètre est : « l'écart-types » c'est la largeur de ces gaussiennes., il y a un écart-type pour chaque unité de traitement des nœuds cachés. Donc, bien sûr son nombre est égal au nombre de nœuds cachés.

3. Les poids

Le troisième ensemble de paramètre sont les poids. C'est une matrice à 2 dimensions dont le nombre de ligne est égal au nombre de nœuds cachés et chaque ligne compte le même nombre de colonne que de sorties.

4. Les biais

Le quatrième est le « biais » appelé aussi la valeur de décalage, dont le nombre est égal au nombre de sorties .

A par ces 4 paramètres de base, bien sur les entrées, les sorties et les couches cachées.

C) Fonctions de bases et calculs

Chaque couche cachée utilise la fonction gaussienne suivante

$$\phi_j(x) = e^{-\frac{\|x - \mu_j\|^2}{2\sigma_j^2}}$$

Cette fonction calcule la sortie pour chaque nœud avec :

phi= c'est la sortie de chaque nœud caché

j = l'indice du nœud

U = centre de gravité

X = vecteur d'entrée

Sigma= écart-type

Les deux barres représentent la distance géométrique entre X et Uj

La combinaison des sorties des nœuds cachés pour donner les sorties et calculer par l'équation est la suivante

$$y_k(x) = \left(\sum_{j=0}^{M-1} w_{jk} * \phi_j(x) \right) + b_k$$

Y = la matrice de sortie a une dimension de sortie

k= indice de la sortie

M = nombre de nœud cachés

w(jk)= le poids du nœud caché à l'indice j et k

b(k)= la valeur de décalage (biais)

D) L'Apprentissage

: L'apprentissage consiste à déterminer les 4 paramètres à partir des données brutes, pour cela, on prend des données brutes, c'est-à-dire des données tirées d'une base de données

Ces données sont ensuite mises dans un tableau ou une matrice. A partir de ces données, on constitue ce qu'on appelle « des données d'entraînement » et « des données de teste ». Et en suite, à partir de ces 2 tableaux ou matrice on entraîne le réseau pour enfin avoir les 4 paramètres cités plus haut.

1. Création des données d'entraînement et de testes

La première étape consiste donc à prendre les données de base, ensuite, les données seront échantillonnées, une grosse partie de ces données serviront « de données d'entraînements » et une petite partie pour en faire des « données de testes »

Pour les données d'entraînements et de testes, on prend respectivement 80% et 20% des données brutes en prenant les données au hasard.

2. Génération des centroïdes

L'étape après consiste à générer les centroïdes à partir des données d'entraînements. Il détermine les valeurs « x » d'entrées représentatives . Il existe de nombreuses possibilités ici. Une approche courante consiste à utiliser un algorithme de clustering « k-means » ou « k-medoids » . Les données sont itérativement assignées, réaffectées et regroupées. Ensuite chaque cluster aura un membre de données représentatives. Et celles-ci serviront comme centroïdes du RBF.

- d'abord, on initialise les tailles des matrices.
- En suite pour chaque nœud d'entraînement :
 - Il faut un tableau d'indice qui servira ensuite à chercher la distance, pour cela, on utilise l'algo « échantillonnage de réservoir » qui consiste à choisir les premiers indices n possibles, puis remplacer les indices initiaux avec les indices possibles restants.
 - Après que cela, on calcule la distance « Avg » entre les nœuds consécutifs du tableau d'indice qu'on a obtenu avec la méthode ci-dessus. Ici la distance moyenne entre les paires adjacentes des candidats est calculée avec une variation de la distance de « Manhattan ».

- Ensuite on fait la moyenne de la somme de toutes ces distances afin de déterminer la distance moyenne
 - et on sauvegarde le résultat de cette moyenne dans un tableau des meilleurs indices
- après avoir obtenu ce « tableau des meilleurs indices » : on crée le centroïde à partir de ce dernier. Le centroïde est une matrice et comprend les données qui ont les mêmes indices que ceux contenu dans le tableau des meilleurs indices.

3. Génération de l'écart-type

L'écart-type, ou valeur de largeur pour chaque nœud caché est calculé à partir de la largeur commune comme la moyenne des distances euclidienne entre toutes les paires possibles du centre de gravité. La distance est la somme des distances euclidienne divisée par le nombre de centroïdes.

4. Détermination des poids, et du biais

Pour calculer les poids de chaque nœuds cachés il existe beaucoup de techniques, mais on va utiliser « l'optimisation d'essaim de particules » pour estimer les meilleures valeurs . Cette technique est abrégée par l'acronyme : « PSO ».

PSO est un comportement de méta heuristiques groupe coordonnée sur la base, comme les vols d'oiseaux ou de bancs de poissons. En PSO, une particule a une position qui représente une solution potentielle (le meilleur jeu de valeurs de poids en l'occurrence). Chaque particule a une vitesse qui détermine la position suivante de la particule. La PSO cherche non seulement les poids mais aussi les biais.

E) Algorithme et structure du futur code programmation.

A La base de l'algorithme utilisés les formules citées plus haut pour faire le traitement à partir des paramètres :

1. Les attributs

nbEntrEes, nbNoeudsCachEs, nbSorties :int

entrEes, ecartsTypes, biais, sorties : Ligne;²

centreDeGravitEe, poids : Matrice

2. Les méthodes

Les méthodes sont toutes des getters et setters qui servent à attribuer la valeur des différents attributs.

3. Le traitement « traiter() »

Cette méthode fait le traitement des données à partir des paramètres, en utilisant les formules plus haut :

- Pour chaque nœud caché :
 - On calcule la distance euclidienne à partir de la matrice d'entrée et du centre de gravité à l'indice courant
 - On calcule la sortie du nœud caché
- Pour chaque sortie :
 - Pour chaque nœud caché
 - ❖ La sortie à l'indice courant est égal à la sortie de la ligne calculée précédemment multipliée par le poids à l'indice du nœud courant et de la sortie courante
 - La sortie finale est égale à la sortie attribuée précédemment plus le biais à l'indice courant

4. Pour l'entraînement :

a) .Création des données d'entrainements et de teste

- On met tous les indices de données brutes dans un tableau
- On échange les indices
- On initialise les tailles : 80% pour l'entraînement et 20% pour le teste
- On crée les matrices d'entraînement et teste
- On remplit les deux matrices

b) Génération des centroïdes

- On déclare les variables : la taille des données d'entraînement, un tableau pour contenir l'indice des meilleures données égales au nombre de nœuds cachés, distance maximum
- Pour i de 0 à la taille des données d'entraînement
 - On crée un tableau d'indice aléatoire à l'aide de l'algo d'échantillonnage de réservoir
 - Pour j de 0 à la taille du tableau d'indice aléatoire -1

- ❖ On fait la somme des valeurs contenues dans l'indice actuel avec celle de la valeur contenue dans l'indice suivant
- A l'aide de la somme calculée plus haut, on fait la moyenne en divisant la somme avec le nombre d'entrées
- Pour i de 0 à nombre d'entrées :
 - Idx = indice contenu dans le tableau des meilleurs indices à l'indice i
 - Pour j de 0 à nombre d'entrées
 - ❖ On assigne le centre de gravité à l'indice i et j avec la matrice des données d'entraînement à l'indice idx et j
- c) Génération des écart types :
- On initialise la somme des distances et le compteur
- Pour i de 0 à nombre de ligne des centres de gravités-2
 - Pour j de i+1 à nombre de ligne des centres de gravités-1
 - ❖ On calcule la distance avec la fonction de la distance euclidienne, distance entre le centre de gravité à l'indice i et le centre de gravité à l'indice j
 - ❖ On ajoute la distance calculée précédemment avec la somme
 - ❖ On incrémente le compteur
- On fait la moyenne à parti de la somme et le compteur
- Pour i de 0 à taille des écart types
 - On assigne la valeur avec la moyenne calculée plus haut

d) . Génération des poids et des biais

La génération des poids et des biais utilise l'algo de "PSO"

CONCLUSION.

Ce petit résumé fut l'occasion de présenter le réseau à fonction de base radial avec sa présentation ; ces avantages et son fonctionnement interne sans trop entrer dans trop de détails. Un algorithme a été toutefois créé et un petit programme a été développé afin de tester le fonctionnement de l'algorithme et l'efficacité de ce type de réseau sur des données concrètes.