

Laboratoires d'Évaluation d'algorithmes pour l'identification de visages statiques

SYS828 – Systèmes biométriques

Responsable et enseignant: Eric GRANGER

Auxiliaire de laboratoire: George Ekladios

Session: A2019

Laboratoire 5 – Classification– RBF et Combinaison de classificateurs

L'objectif de ce laboratoire est de se familiariser avec les étapes d'apprentissage, de validation et d'évaluation de la performance d'un système de classification. Les classificateurs choisis pour l'expérimentation sont le K plus proche voisin, les réseaux de neurones à fonctions de base radiale et les machines à vecteur de support qui devront être testés pour l'identification de visage. Alors que le kNN et SVM ont été présenté dans le laboratoire précédent, ce laboratoire présente le réseau de neurones à fonctions de base radiale et la combinaison des classificateurs.

5.1 RBF (Radial Basis Function):

Les réseaux de neurones à fonctions de base radiale calculent la distance entre les entrées, c'est-à-dire que la fonction de combinaison renvoie la norme euclidienne du vecteur issu de la différence vectorielle entre les vecteurs d'entrées. Il s'agit de trouver une surface multidimensionnelle qui correspond le mieux avec les données d'entraînement et utiliser cette surface pour interpoler les données de test.

Un RBF est constitué uniquement de 3 couches (figure 1):

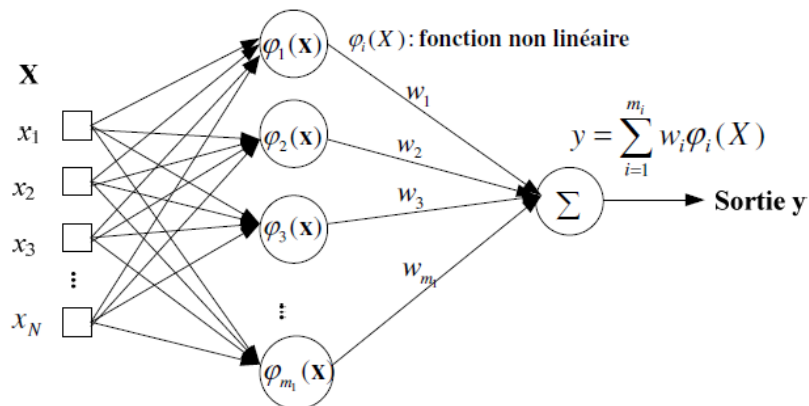


Figure 1: Illustration d'un réseau de neurones à fonctions de base radiale (RBF)

Entrée retransmet les inputs (vecteur des caractéristiques) sans distorsion

Cachée transformation non linéaire $x_i \rightarrow \phi(x_i)$ des entrées, par exemple chaque neurone RBF contient une gaussienne $\phi_j(x) = \exp(-\frac{\|x - t_j\|^2}{2\sigma_j^2})$. Pour une entrée donnée, la sortie du neurone RBF est la hauteur de la gaussienne en ce point (voir figure 2).

Sortie transformation linéaire $\phi(x_i) \rightarrow y_i$ des fonctions cachées. Une combinaison linéaire des sorties des neurones RBF multipliés par le poids de leur connexion respective $y_k(x) = \sum_{j=1}^M w_{kj} \cdot \phi_j(x)$. Pour un problème de régression la sortie consiste à un neurone représentant la valeur à estimer. Pour un problème de classification on aura autant de sorties que de classes et chaque sortie est interprétée comme une probabilité a posteriori.

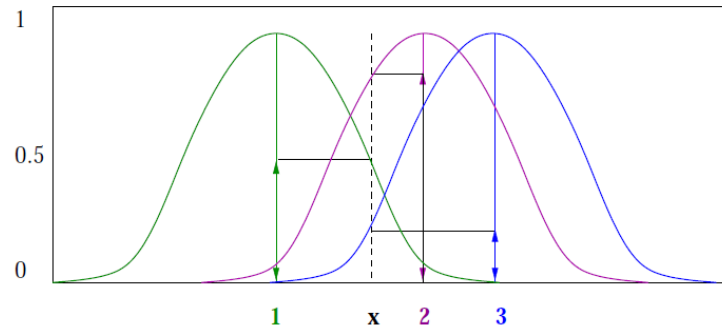


Figure 2: Activation des neurones cachés: 3 unités cachées de même largeur ayant pour centres 1, 2, 3 respectivement et une entrée x à classer. Plus x est proche du centre d’une gaussienne, plus l’activation de l’unité est grande (hauteur)

Les paramètres à régler (pour RBF gaussienne) sont: les centres t_i , les dispersion σ_i , et les poids w_{kj} . 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:

- A) 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.
- B) Centres obtenus par apprentissage non-supervisé: estimer les centres par catégorisation k-means, les dispersions peuvent être choisies comme la distance moyenne entre les centres voisins, et les poids sont la solution des équations linéaires.
- C) Centres obtenus par apprentissage supervisé: tous les paramètres sont ajusté pour minimiser une fonction d’erreur

5.2 Expérimentations

5.2.1 Exploration avec les données synthétiques

- A) écrire un `exp5_rbf.m` pour entraîner le réseau de neurones RBF en utilisant un apprentissage avec validation “hold-out” et les différentes bases de données (modifier le script du laboratoire précédent pour remplacer SVM par RBF).
- B) Optimiser le nombre de centre du réseau RBF en variant sa valeur et en mesurant l’erreur (pour mieux choisir ce paramètre, vous pouvez tracer une courbe qui montre l’erreur en fonction de variations du paramètre).
- C) Incorporez le code pour entraîner et afficher le classificateur kNN afin de comparer les deux classificateurs.

Fonction utile (PRTools) : `rbnc`, `plotc`.

5.2.2 Pour le projet de laboratoire

- A) Comparez les performances de kNN et du RBF avec la base de données AT&T, qui a été organisée pour permettre un apprentissage avec validation croisée en 5 blocs et une évaluation des performances.
- B) Refaire l'étape précédente en utilisant des caractéristiques définies par les algorithmes d'extraction de caractéristiques vus durant le laboratoire 2 (PCA et LDA).
- C) Faire varier le nombre de caractéristiques.

5.3 Combinaison de classificateurs

La combinaison de classificateurs est devenue une sous-discipline active dans le domaine de la reconnaissance biométrique. Combiner plusieurs classificateurs au lieu de chercher un classificateur unique peut servir à améliorer les performances de classification et à réduire la complexité des modèles. La combinaison parallèle des classificateurs est parmi les approches les plus utilisées en pratique. Selon cette approche, la fusion peut être faite à plusieurs niveaux : au niveau de l'extraction de caractéristiques, au niveau de la classification, au niveau de la décision.

Nous allons expérimenter avec quelques techniques de fusion au niveau de classificateurs. La forme à reconnaître est présentée simultanément à l'ensemble de E classificateurs et les réponses de chaque classificateur sont combinées avec des règles de fusion fixes, e.g., moyenne pondérée, minimum, etc. Essayer le script `exp5_comb.m`, puis changer le générateur des données, les classificateurs, et leurs paramètres pour voir l'effet de la fusion au niveaux de la classification. (voir `help stacked, classc`).

Pour le présent projet, nous feront une combinaison de classificateurs en utilisant un vote majoritaire (voir `help votec`). Cette méthode simple consiste à définir une prédiction comme étant la classe prédite par le plus grand nombre de classificateurs.