

LELEC2870: séance d'exercices 3

Radial basis functions networks

9 novembre 2012

1 Objectifs

Cette séance traite de l'approximation de fonctions en utilisant des radial basis functions networks (RBFNs). Ces réseaux neuronaux peuvent être vus comme une extension des modèles linéaires ; en effet, leur sortie est une combinaison linéaire des entrées, transformées de façon non-linéaire. Une des avantages des RBFNs est que leur apprentissage peut être simplifié en le divisant en trois étapes distinctes. Durant cette séance, vous implémenterez un RBFN et le testerez sur des données. L'archive contient les données et un script SciLab.

2 Description de la séance

Au cours de cette séance, vous implémenterez les trois étapes de l'apprentissage des RBFNs, qui sont détaillées dans les deux sections suivantes. Ensuite, vous devrez utiliser votre RBFN pour approximer les deux fonctions bivariées

$$f : \mathbb{R}^2 \rightarrow \mathbb{R} : f(\mathbf{x}) = \frac{\sin \|\mathbf{x}\|_2}{\|\mathbf{x}\|_2}$$

où $\|\mathbf{x}\|_2$ est la norme euclidienne de \mathbf{x} et

$$f : \mathbb{R}^2 \rightarrow \mathbb{R} : f(\mathbf{x}) = e^{-\frac{1}{2} \|\mathbf{x}\|_2^2}.$$

Pour chaque fonction, un dataset contenant quelques instances d'entraînement est fourni.

Une fois votre RBFN implémenté, entraînez-le grâce aux données fournies et visualisez le résultat en utilisant la fonction `visualise`. Quelle est l'influence du nombre de centres M et du width scaling factor h ? Pouvez-vous expliquer les résultats que vous observez pour de petites, moyennes et grandes valeurs de ces paramètres ? Y-a-t-il un lien avec les phénomènes d'*overfitting* et d'*underfitting* ?

Pour chaque dataset fourni, extrayez quelques instances pour construire un nouveau training set plus petit, entraînez un nouveau RBFN et visualisez le résultat. Observez-vous une différence ? En général, quel l'impact pensez-vous que le nombre d'instances aura sur l'apprentissage ?

3 Radial basis functions networks

Le but de la procédure d'apprentissage est de construire un modèle capable de produire une approximation de la fonction. Typiquement, la fonction est uniquement connue à travers un training set d'observations $\{(\mathbf{x}^p, t^p) | p = 1 \dots P\}$. En particulier, un RBFN peut être vu comme une extension du modèle linéaire

$$y = \sum_{i=1}^D w_i x_i + w_0$$

où D est la dimensionalité de \mathbf{x} . Au lieu de travailler avec les entrées \mathbf{x} , nous allons utiliser une transformation non-linéaire de celles-ci : la sortie du RBFN est

$$y = \sum_{i=1}^M w_i \phi_i(\mathbf{x}) + w_0$$

où

$$\phi : \mathbf{x} \in \mathbb{R}^D \rightarrow \phi(\mathbf{x}) \in \mathbb{R}^M$$

est une transformation non-linéaire et M est le nombre de composantes (ou basis functions). Pour cette séance, nous utiliserons une fonction gaussienne de la forme

$$\phi_i(\mathbf{x}, \mathbf{c}^i, \sigma^i) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}^i\|^2}{2\sigma^{i2}}\right)$$

où l'index $i = 1, \dots, M$ indique la composante de la fonction ϕ . Deux paramètres sont associés à chaque fonction et doivent être appris : le centre \mathbf{c}^i , appartenant au même espace que \mathbf{x} , et une largeur de gaussienne σ^i .

4 Procédure d'apprentissage

Un RBFN peut être appris en trois étapes :

- déterminer la position des centres ;
- déterminer la largeur des noyaux gaussiens ;
- adapter les poids de la couche de sortie.

Une façon de déterminer ces paramètres serait de calculer le gradient de l'erreur par rapport à ceux-ci et de réaliser une descente de gradient pour minimiser l'erreur. Cependant, dans le cas d'un RBFN, cette procédure doit être uniquement utilisée pour déterminer les poids optimaux. Pour choisir les positions et largeurs des noyaux gaussiens, des procédures plus simples et moins coûteuses seront utilisées ici. Ces procédures sont non-supervisées (quels sont les avantages et inconvénients d'un tel type de procédure?).

4.1 Position des centres

Pour la première phase de l'apprentissage, le but est d'étaler les centres de façon à imiter la densité de probabilités des données d'entraînement. Pour faire cela efficacement, vous pouvez utiliser une des méthodes de vector quantization que vous avez implémentées au cours de la séance précédente.

4.2 Détermination des largeurs des noyaux gaussiens

La largeur des noyaux gaussiens doit être liée à la densité des données d'apprentissage (localement près de chaque centre). Une solution simple consiste à utiliser la moyenne des distances entre le centre et les points situés dans la zone de Voronoï du centre considéré, multipliée par un facteur de lissage h (width scaling factor), commun à tous les noyaux.

4.3 Adaptation des poids

Tout comme nous l'avons vu pour le modèle linéaire, il est possible d'optimiser les poids du RBFN en minimisant un critère d'erreur. Si le critère des moindres carrés est choisi, l'optimisation peut être réduite à la résolution d'un système linéaire ; il est possible de montrer que les poids optimaux sont données par

$$\mathbf{w} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{t}$$

où les entrées transformées sont placées dans la matrice

$$\Phi = \begin{pmatrix} 1 & \phi(\mathbf{x}^1) \\ 1 & \phi(\mathbf{x}^2) \\ \vdots & \vdots \\ 1 & \phi(\mathbf{x}^P) \end{pmatrix} = \begin{pmatrix} 1 & \phi_1(\mathbf{x}^1) & \phi_2(\mathbf{x}^1) & \cdots & \phi_M(\mathbf{x}^1) \\ 1 & \phi_1(\mathbf{x}^2) & \phi_2(\mathbf{x}^2) & \cdots & \phi_M(\mathbf{x}^2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \phi_1(\mathbf{x}^P) & \phi_2(\mathbf{x}^P) & \cdots & \phi_M(\mathbf{x}^P) \end{pmatrix} \quad (1)$$

et les valeurs à prédire sont placées dans le vecteur (colonne)

$$\mathbf{t} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_P \end{pmatrix}. \quad (2)$$

La première colonne de cette matrice Φ correspond au terme de biais w_0 .