

Dossier de TIPE

Camille Chanial, Tristan Du Castel, Tristan Stérin

05/06/2014

Résumé

caca

Table des matières

| | | |
|----------|---|----------|
| 1 | Introduction | 2 |
| 2 | La justification mathématiques | 2 |
| 3 | L'algorithme de rétropropagation | 2 |
| 4 | L'application pratique : aspects techniques et résultats | 3 |

1 Introduction

2 La justification mathématiques

3 L'algorithme de rétropropagation

On a donc vu que les réseaux de neurones présentent un modèle pertinent dans le cadre de la régression non linéaire.

Il faut désormais mettre en place l'algorithme qui va permettre de trouver, d'apprendre, la combinaison de poids optimales afin de réaliser l'approximation désirée.

On se donne une base d'apprentissage :

$$B = \{(p_1, q_1), \dots, (p_n, q_n)\} \quad \text{avec : } (p_i, q_i) \in A^a \times B^b$$

Où A et B sont les espaces de départ et d'arrivée et a et b leur dimension respective.

Si on note $f : A \rightarrow B$ la fonction du réseau (variable au cours de l'apprentissage), notre but est d'avoir un algorithme tel qu'en un nombre fini d'itérations :

$$\forall i, \quad f(p_i) = q_i$$

On va procéder à cet apprentissage par cycle de présentation de tous les couples de la base au réseau.

Lors d'un cycle, on présente chaque couple au réseau, pour le i ème couple on calcule le vecteur erreur instantanée $\vec{e} = q_i - f(p_i)$.

On se donne comme critère de "bon apprentissage" de notre réseau, comme indice de performance l'erreur quadratique moyenne :

$$F(p_i) = E[\vec{e}^t * \vec{e}]$$

L'espérance en question n'étant pas calculable a priori, on l'approxime l'indice de performance par l'erreur instantanée :

$$\hat{F}(p_i) = \vec{e}^t * \vec{e}$$

Ainsi notre but est de minimiser la fonction \hat{F} . On utilise pour cela la méthode de descente de gradient telle que :

$$\Delta \omega_{i,j}^k = - \frac{\partial \hat{F}}{\partial \omega_{i,j}^k}$$

Avec $\omega_{i,j}^k$ le j ème poids du i ème neurone de la k ème couche, η le taux d'apprentissage.

Le sujet de cet exposé n'étant pas la descente de gradient, on introduira très brièvement à l'oral les notions permettant de comprendre la suite.

Calculer les dérivées partielles de \hat{F} n'est pas un problème simple.

On expliquera à l'oral les principales étapes de ce calcul.

On donne ici uniquement l'expression final qui jsutifie l'algorithme proposé par la suite.

On introduit une quantité intermédiaire appelée sensibilité, associée à chaque couche, définie vectoriellement par :

$$s^k = \frac{\partial \hat{F}}{\partial n^k}$$

s^k est la sensibilité de la couche k , et n^k le vecteur des niveaux d'activation de la $k^{ième}$ couche.

On a : $\Delta W^k = -\eta s^k * {}^t a^{k-1}$

Avec W^k la matrice des poids de la $k^{ième}$ couche et a^k le vecteur sortie de la $k^{ième}$ couche.

4 L'application pratique : aspects techniques et résultats