

TP réseaux de neurones

Implémentation de l'algorithme de perceptron multicouche

Luong Phat NGUYEN

Remarques préliminaires :

- Ce TP est noté.
- Il faut rendre : vos codes sources (avec commentaires) + compte rendu, par mail :
luong.nguyen@univ-tours.fr

Le but de ce TP est d'implémenter un perceptron multi-couche. Nous allons utiliser l'algorithme de descente de gradient pour l'optimisation.

Algorithme 1 : L'algorithme de perceptron multicouche

Input : L -nombre de couches cachées, \mathbf{X} -données numériques, \mathbf{Y} -Sortie du réseau (une matrice sparse, etc.)

Input : \mathbf{W}^l , $1 \leq l \leq L$, matrice de poids.

Input : \mathbf{b}^l , $1 \leq l \leq L$, vecteur de poids.

Output : Paramètres ajustés \mathbf{W}^l et \mathbf{b}^l

Initialisation ;

for $epoch = 1, 2, \dots$, nombre d'epoch **do**

$\mathbf{a}^0 = \mathbf{X}$;

for $l = 1, 2, 3, \dots, L - 1$ **do**

$\mathbf{z}^l = \mathbf{W}^l \mathbf{a}^{l-1} + \mathbf{b}^l$;

$\mathbf{a}^l = f(\mathbf{z}^l)$;

end

$\mathbf{z}^L = \mathbf{W}^L \mathbf{a}^{L-1} + \mathbf{b}^L$;

$\hat{\mathbf{Y}} = \mathcal{F}(\mathbf{z}^L)$;

 Compute $J(\hat{\mathbf{Y}}, \mathbf{Y})$;

$\mathbf{g} \leftarrow \nabla_{\hat{\mathbf{Y}}} J(\hat{\mathbf{Y}}, \mathbf{Y})$;

for $l = L, L - 1, \dots, 2, 1$ **do**

$\mathbf{g} \leftarrow \mathbf{g} \odot \nabla_{\mathbf{a}^l} f'(\mathbf{a}^l)$, si $l = L$, $f = \mathcal{F}$;

$\nabla_{\mathbf{b}^l} = \mathbf{g}$;

$\nabla_{\mathbf{W}^l} = \mathbf{g} \mathbf{a}^{(l-1)T}$;

$\mathbf{g} \leftarrow \nabla_{\mathbf{a}^{l-1}} J = \mathbf{W}^{lT} \mathbf{g}$;

end

end
