

Informatique S6 - CC1

Exercice 1 (Perceptron (11.0 points)).

On va construire un **Perceptron**. Il s'agit d'un modèle paramétrique qui permet d'approcher une fonction. Il s'agit du modèle le plus simple de réseau de neurones (très important en IA). On va construire une classe pour gérer ces modèles. Un perceptron est une fonction du type

$$f_{\theta}(x) = \sigma((X, W) + b)$$

avec $X \in \mathbf{R}^d$, $W \in \mathbf{R}^d$ et $b \in \mathbf{R}$. On appelle la fonction $\sigma()$ une fonction d'activation. Elle est un paramètre de la méthode.

Question (4.5 points). Ecrire la classe perceptron qui contient comme attribut :

- un pointeur W de type à déterminer, qui contiendra le tableau des coefficients W
- un double b ,
- la taille des données d (type à déterminer),
- deux pointeurs de fonctions "sigma()" et "d_sigma()" qui contiendront la fonction d'activation et sa dérivée.

Écrivez les constructeurs par défaut (0.75pts), par copie (1pts), le destructeur (0.75 pts) et un constructeur qui prend en entrée "d", "sigma()" et "d_sigma()" (1Pts).

Question (2.0 points). Ecrire les mutateurs et accesseurs pour les poids W et b .

Question (1.0 points). Ecrire des accesseurs qui appliquent "sigma()" et "d_sigma()" à un double "x" et renvoie le resultat.

Question (1.5 points). Surcharger l'opérateur "="

Question (2.0 points). Ecrire un exemple ou on initialisera les poids avec comme fonction d'activation la sigmoïde : $\sigma(x) = \frac{1}{1+e^{-x}}$ et $\sigma'(x) = \sigma(x)(1 - \sigma(x))$

Exercice 2 (Gradient (9.0 points)).

On va construire la méthode de gradient utilisée pour résoudre des problèmes de minimisation. Elle est notamment utilisée pour calculer les poids de modèles paramétriques (comme les perceptron).

On ne l'appliquera pas dans ce cas la mais dans un cas plus facile. On considère la fonction :

$$f(x) = \phi(wx + b)$$

avec ϕ une fonction, $w, b \in \mathbf{R}$ de points. On souhaite ici trouver w et b qui minimiser

$$J(w, b) = \sum_{i=1}^n f(x_i).$$

Pour cela on va donc implémenter une méthode de gradient.

Conseils : les structures des classes de l'exo 1 et 2 sont similaires. En recopiant et modifiant la classe une vous irez plus vite.

Question (4.5 points). Ecrire la classe "gradient" qui contient comme attribut :

- un pointeur X de type a à déterminer, qui contiendra les (x_1, \dots, x_n)
- un double b , un double w
- le nombre des données n (type à déterminer),
- deux pointeurs de fonctions "phi()" et "d_phi()" qui contiendront la fonction d'activation et sa dérivée.

Écrivez les constructeurs par défaut (0.75pts), par copie (1pts), le destructeur (0.75 pts) et un constructeur qui prend en entrée "d", "phi()" et "d_phi()" (1Pts).

Question (1.5 points). Ecrire les accesseurs pour les poids W et b et un mutateur pour X .

Question (1.5 points). Ecrire une méthode qui calcul le gradient :

$$\partial_w J(w, b) = \sum_{i=1}^n \phi'(wx_i + b)x_i$$

et

$$\partial_b J(w, b) = \sum_{i=1}^n \phi'(wx_i + b)$$

Question (1.5 points). Ecrire une méthode qui calcul les poids avec le processus itératif :

$$w = w - \eta \partial_w J(w, b)$$

et

$$b = b - \eta \partial_b J(w, b)$$

η est un paramètre de la méthode et on se donne aussi un nombre max d'itération en paramètre.