

TP4 : Introduction à l'apprentissage machine (2h)

Exercice 1 –Regression linéaire

A l'aide d'un système à base d'arduino et capteurs, nous cherchons à trouver une relation linéaire entre la vitesse d'un train électrique et la consommation électrique de son moteur. Nous avons collectés les données pour trouver cette relation lorsque le train circule avec 50 passagers à bord :

Yi: Vitesse (Km/h)	Xi Kw
80	20
60	50
50	42
30	25
90	70
110	?

Q 1.1 Représenter graphiquement l'hypothèse linéaire $h_{\theta}(x)$ (vitesse du train en fonction la puissance moteur)

Q 1.2 Ecrire le critère J et rappeler l'algorithme de descente de gradient pour calculer les paramètres (θ_1, θ_0) .

Q 1.3 Ecrire un code C Arduino Due pour l'apprentissage (θ_1, θ_0) . Le taux d'apprentissage $\alpha = 1/100$ et la précision de l'algorithme est 10^{-3} . Il est préférable d'ajouter la fonction d'apprentissage dans le fonction **setup()**

Q 1.4 Utiliser la table ci-dessus pour l'alimenter la fonction d'apprentissage et exécuter l'apprentissage de θ_0 et θ_1 sur l'arduino Due. Tester d'autres valeurs de $0 < \alpha < 1$ et conclure.

Q 1.5 Compléter le code Arduino en ajoutant la fonction de prédiction dans **loop()**. Exécuter le code pour afficher la vitesse prédite en fonction de la puissance du moteur. Conclure

Exercice 2 –Regression logistique

Nous cherchons à mettre en place un système de prédiction à deux états 0 ou 1. Nous collectons un dataset présenté dans la table ci-dessous. Nous proposons de réaliser cette prédiction à partir d'une hypothèse de regression logistique binaire.

X_i	Sortie Y_i
5	1
3,5	1
3	0
2	?
0,5	0
1,5	0
0	?
1	0

Q 2.1 Rappeler le critère **J** et l'algorithme de descente de gradient pour calculer les paramètres(θ_1, θ_0) dans le cas d'une régression logistique binaire.

Q 2.2 Ecrire un code Arduino pour l'apprentissage des paramètres de régression logistique. Le taux d'apprentissage $\alpha = 1/50$ et la précision de l'algorithme est 10^{-3} . Il est intéressant de mettre la fonction « Apprentissage » dans le **setup()** du code arduino.

Q 2.3 Utiliser la table ci-dessus pour alimenter la fonction d'apprentissage et exécuter le code sur

l'arduino Due. Afficher dans le moniteur série le nombre d'itération maximum pour avoir les meilleurs valeurs (θ_1, θ_0)

Q 2.4 Ajouter au code arduino une fonction pour afficher l'hypothèse $h_\theta(x)$ dans le moniteur série.

Exécuter encore une fois le code sur l'arduino Due.

Q 2.5 Ajouter la fonction de prédiction binaire dans **loop()** du code C arduino. Le taux d'apprentissage est toujours $\alpha = 1/50$ et la précision de l'algorithme est 10^{-3} .

Q 2.6 Proposer Un seuil de décision associé l'hypothèse pour afficher 0 ou 1.

Exercice 3 –SVM

Nous espérons concevoir un système de prédiction à partir des données étiquettés +1 ou -1. Nous avons classé les les mesures dans 03 tables avec les étiquettes.

x_1^i	x_2^i	Y_i	x_1^i	x_2^i	Y_i	x_1^i	x_2^i	Y_i
0	1	+1	3	3	+1	0	-1	-1
2	1	+1	1	0	+1	3	-2	-1
-1	0	+1	1	-2	-1	-4	-3	-1
-2	2	+1	-2	-2	-1	0	-4	-1

Nous avons utilisé la SVM pour réaliser la prédiction à deux état -1 et +1 . La fonction hypothèse considérée pour cette prédiction est donnée par :

$$h_\theta(x) = \frac{1}{1 + e^{-(\theta_2 x_2 + \theta_1 x_1 + \theta_0)}}$$

Q3.1 Nous utilisons le critère simplifié ci-dessous pour résoudre le problème de prédiction :

$$J(\theta) = C \sum_{i=1}^m y^{(i)} \log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)})) + \frac{1}{2}(\theta_1^2 + \theta_2^2)$$

Déduire l'algorithme d'apprentissage pour calculer les paramètres $(\theta_2, \theta_1, \theta_0)$.

Note : $\frac{\partial}{\partial \theta_i} \log(h_{\theta}(x^{(i)})) = \frac{1}{h_{\theta}(x^{(i)})} \frac{\partial}{\partial \theta_i} h_{\theta}(x^{(i)}) = \frac{1}{h_{\theta}(x^{(i)})} h_{\theta}(x^{(i)}) (1 - h_{\theta}(x^{(i)})) = 1 - h_{\theta}(x^{(i)})$

Q 3.2 Ecrire le code Arduino pour l'apprentissage des paramètres de SVM dans la fonction **setup()**.

Le taux d'apprentissage $\alpha = 1/50$ et $C = \frac{1}{m\lambda}$. La précision de l'algorithme est 10^{-3} et la constante $\lambda = 0.1$. Utiliser les trois tables lors de la phase l'apprentissage SVM.

Q 3.3 Ajouter à votre code arduino une fonction pour afficher $h_{\theta}(x)$ avec 03 paramètres dans le moniteur série. Exécuter encore une fois le code sur l'arduino Due et afficher les meilleurs valeurs de θ

Q 3.4 Ajouter à votre code arduino une fonction pour calculer et afficher la largeur de la SVM. Exécuter encore une fois le code sur l'arduino Due et afficher dans le moniteur série la largeur SVM.

Q 3.5 Ajouter la fonction de prédiction dans **loop()** du code C arduino. Tester la fonction de prédiction avec les données de la table ci-dessous. Conclure.

x_1^i	x_2^i	Y_i
2	3	?
-1	4	?
-1	-1	?
-2	-2	?
0	0	?