Partiel

Exercice 1 (4 points) – Questions indépendantes

Q 1.1 Commenter les phrases suivantes : vrai,faux, ou partiellement vrai ou faux et pourquoi

1. On peut obtenir plusieurs optima locaux lors de la résolution d'une régression linéaire par les moindres carrés dans le cas général.
2. Il est plus facile de sur-apprendre avec un k-nn qu'une régression logistique
3. Il est plus facile de sur-apprendre avec peu de dimensions qu'avec beaucoup
4. Les réseaux de neurones optimisent une fonction convexe, peuvent être utilisés en régression et classification.
5. Utiliser une validation croisée nous assure de ne jamais sur-apprendre.
6. Pour la fonction $f(x) = max(2, |x^2 + 1|) - 1$, l'algorithme de descente du gradient converge toujours vers le minimum de la fonction.

Q 1.2 Le(s)quel(s) de ces classifieurs peuvent atteindre une erreur nulle en apprentissage pour n'importe quel ensemble linéairement séparable : Arbres de décision, 2-NN, 15-NN, Perceptron, réseaux de neurones.

Q 1.3 Expliquer en quelques lignes le principe de la régression aux moindres carrés et logistique. Est-il plus adapter de faire de la classification avec la régression logistique ou aux moindres carrés ? Pourquoi ?

Exercice 2 (4 points) – Réseau de neurones

On considère un réseau de neurones à une couche cachée de trois entrées (x_1, x_2, x_3), deux neurones (h_1, h_2) et deux sorties y. On utilise une fonction d'activation $g_1(z) = max(0, z)$ pour la couche cachée et $g_2(x) = tanh(x)$ pour la couche de sortie. Soit W et V les poids de la première couche et de la deuxième couche. On utilise un coût aux moindres carrés pour l'optimisation.

Q 2.1 Dessiner le réseau, donner les dimensions de W et V.

Q 2.2 Donner l'expression de la sortie y en fonction de (x_1, x_2, x_3) et des poids.

Q 2.3 Donner l'expression exacte des dérivées partielles du coût en fonction des poids.

Q 2.4 Donner l'algorithme du gradient pour l'apprentissage de ce réseau de neurones.

Exercice 3 (4 points) – Risque

Soit le résultat d'une analyse médicale exprimé par un réel $x \in \mathbb{R}$, soit deux classes y_-, y_+ pas malade et malade. On sait que $P(y = y_+|x) = \begin{cases} 0 & \text{si } x \leq 0 \\ x & \text{si } 0 \leq x \leq 1 \\ 1 & \text{si } x \geq 1 \end{cases}$.

On utilise un classifieur de type $f_\theta(x) = \begin{cases} y_+ & \text{si } x > \theta \\ y_- & \text{si } x \leq \theta \end{cases}$.

Q 3.1 Donner l'expression du coût 0-1 et donner le risque en un point x^0 en fonction de θ.

Q 3.2 On suppose le résultat du test x uniformément répartie dans $[-1, 1]$. Quel est le classifieur optimal ? Quelle est la valeur du risque minimale ?

Q 3.3 On sait qu'il est 3 fois plus coûteux de classer un malade en pas malade que l'inverse. Donner une fonction de coût associée, la nouvelle formulation du risque et le classifieur optimal.

Q 3.4 Pourrait-on faire mieux avec un classifieur bayésien ? Un classifieur bayésien naïf ?

Exercice 4 (4 points) – **Metric learning**

Soit $p_A : \mathbb{R}^d \to \mathbb{R}^d$ une projection linéaire et $A \in \mathbb{R}^{d \times d}$ sa matrice associée, $Ax \in \mathbb{R}^d$ le point projeté et un ensemble d'apprentissage $E = \{(x^i, y^i) \in \mathbb{R}^d \times \{-1, 1\}\}, i \in \{1, \ldots, n\}$ de n exemples. Le domaine du *Metric Learning* s'attache à trouver une projection A telle qu'il soit plus facile de séparer les projections $\{(p_A(x^i), y^i)\}_{i=1}^n$ que la base initiale E avec un algorithme du type k-nn.

Q 4.1 Supposons A diagnonale. Quelle est la signification d'un coefficient a_{ii} nul à la i-ème position de la diagonale ? D'un coefficient grand ? Donner un cas de figure où ce type de projection peut être utile.

Q 4.2 Que vaut la distance au carré $\|p_A(x^i) - p_A(x^j)\|^2$ en fonction de la matrice AA^T et $(x^j - x^i)$?

Q 4.3 On se propose d'utiliser la fonction de coût suivante :

$$E(W) = \alpha \sum_{i,j|y^i \neq y^j} \left(1 - min\left(\|p_A(x^i) - p_A(x^j)\|^2, 1\right)\right) + (1 - \alpha) \sum_{i,j|y^i = y^j} \|p_A(x^i) - p_A(x^j)\|^2$$

. A quoi correspond le premier et le deuxième terme de la fonction de coût ? Quelle est l'utilité de la fonction min ?

Q 4.4 Proposer un algorithme pour apprendre A.

Exercice 5 (4 points) – **Modèle AR (Auto-Régressifs)**

Nous nous plaçons dans le cadre de la prédiction de série temporelle, par exemple, la prédiction d'un cours de bourse à partir des précédentes valeurs de ce cours. Nous disposons donc d'une donnée $X_t \in \mathbb{R}$ pour chaque pas de temps t : $\{X_t\}_{t=1,\ldots,T}$ représente la série temporelle.
Un modè auto-régressif (AR) consiste à prédire la prochaine valeur en fonction d'un nombre p de valeurs précédentes appelé ordre. Formellement, un modèle AR(p) d'ordre p est défini de la manière suivante :

$$\hat{X}_t = \varphi_0 + \sum_{i=1}^{p} \varphi_i X_{t-i}$$

où $\varphi_0, \ldots, \varphi_p$ sont les paramètres du modèle. Nous noterons $\hat{X}_t$ les estimations issues de notre modèle et X_t la vérité terrain (celle des données).

Q 5.1 Analyse intuitive du modèle

Q 5.1.1 Si nous choisissons un modèle d'ordre 0, quelle est la forme de la prédiction ?

Q 5.1.2 Pour un modèle d'ordre 0, quelle est l'erreur aux moindres carrés ? Quelle est la valeur de φ_0 qui la minimise ?

Q 5.2 Apprentissage d'un modèle d'ordre p.

Q 5.2.1 Lorsque nous travaillons avec un modèle d'ordre p, le premier échantillon que l'on est capable de prédire est l'échantillon $p+1$ (nécessité de disposer d'un historique). Soit la base de donnée $\{X_t\}_{t=1,\ldots,T}$, de combien d'échantillons disposons nous pour apprendre le modèle AR(p) ? Donner le coût au sens des moindres carrés pour ce problème.

Q 5.2.2 Nous souhaitons réécrire le problème d'apprentissage sous forme matricielle :

$$\begin{bmatrix} \hat{X}_{p+1} \\ \hat{X}_{p+2} \\ \hat{X}_{p+3} \\ \vdots \end{bmatrix} = A \times \begin{bmatrix} \varphi_0 \\ \varphi_1 \\ \varphi_2 \\ \vdots \end{bmatrix}$$

Quelle est la dimension de φ (le vecteur contenant les paramètres du modèle) ? Quelles sont les dimensions de A ? Exprimer le contenu des cellules de la matrice A, a_{ij}.

Q 5.2.3 Donner la formulation matricielle du problème d'optimisation de la fonction coût au sens des moindres carrés. Calculer la valeur optimale de φ.

Q 5.3 Implémentation

Q 5.3.1 Donner le code python de la fonction `learnAR(X,p)` qui prend en argument une série temporelle X et l'ordre p et qui retourne les paramètres φ.
NB : cette méthode n'est pas triviale puisqu'elle inclut la mise en forme de la matrice A.

Q 5.3.2 Donner le code python de la fonction `testAR(X,phi)` qui prend en argument une série temporelle X et les paramètres φ et qui calcule les prédictions du modèle sur X.
Cette méthode calculera et retournera également la performance du modèle au sens des moindres carrés.