ARF Examen - 2ème session

Exercice 1 (5 points) – Questions indépendantes

Q 1.1 Soit un réseau de neurones à deux entrées, une couche cachée de deux neurones et un neurone de sortie. On suppose toutes les fonctions d'activation linéaire.

Q 1.1.1 Dessinez ce réseau en précisant tous les poids du réseau.

Q 1.1.2 Est-il possible de représenter ce réseau par un perceptron ? Si oui, donner son équivalent.

Q 1.2 Répondre par vrai ou faux **en justifiant** votre réponse en une ligne :

1. La regression logistique apprend des frontières non-linéaires car la fonction logistique est non linéaire.
2. Sur des données séparables linéairement, l'algorithme du perceptron converge vers des minimas locaux du fait de l'initialisation aléatoire des poids.
3. Les arbres de décision sont plus expressifs que les k-plus proches voisins.
4. L'estimation de densité est un préalable à tout algorithme de classification.

Q 1.3 Un étudiant décide d'utiliser de l'apprentissage par renforcement pour jouer au loto (6 numéros à trouver dans l'ordre parmi 49). A-t-il raison et que peut-il espérer ?

Q 1.4 Donner un réseau de neurones capable de classifier le plan 2D en deux classes : en positif l'ensemble des points dans le carré centré en $(0, 0)$ et de coté de longueur 2, en négatif tous les autres points du plan.

Exercice 2 (6 points) – Multi-classes

Nous considérons dans cet exercice un problème d'apprentissage multi-classes à K classes. Soit $\mathcal{Y} = \{y_1, \ldots, y_K\}$ un ensemble de K labels et $E = \{(\mathbf{x}^1, y^1), \ldots, (\mathbf{x}^N, y^N)\} \in \mathbb{R}^d \times \mathcal{Y}$ un ensemble d'apprentissage du problème.

Q 2.1 Donner une méthode usuelle pour construire un classifieur multi-classes à partir d'un algorithme de classification binaire.

Q 2.2 Nous supposons dans la suite que la probabilité a posteriori d'une classe k est donnée par $P(Y = k|X = \mathbf{x}) = \frac{e^{<\mathbf{w}_k, \mathbf{x}>}}{\sum_{i=1}^{K} e^{<\mathbf{w}_i, \mathbf{x}>}}$, avec $\mathbf{w}_i \in \mathbb{R}^d$ caractérisant chaque classe (nous ne considérerons pas de biais pour simplifier les notations).

Q 2.2.1 Montrer que cette expression définit bien une probabilité.

Q 2.2.2 Peut-on apprendre les paramètres $\{\mathbf{w}_i, i \in \{1, \ldots, K\}\}$ de manière indépendante ?

Q 2.3 On s'intéresse dans cette question à l'apprentissage des paramètres par maximum de vraisemblance.

Q 2.3.1 Quelle quantité veut-on optimiser ?

Q 2.3.2 Donner la formulation mathématique.

Q 2.3.3 En introduisant le logarithme de la quantité précédente, proposer un algorithme itératif de résolution et les mises-à-jour des paramètres.

Q 2.3.4 Donner le code python associé.

Q 2.4 Lors de l'utilisation de ce modèle, on remarque une tendance au sur-apprentissage. Peut-on modifier la fonction objectif afin de calibrer mieux l'espace de recherche ? Expliciter précisément l'idée.

Exercice 3 (4 points) – Bayes naif et perceptron

Les données suivantes sont disponibles au service marketing d'une entreprise, l'objectif étant de prédire l'achat ou non du produit :

ID	âge	revenu (en milliers)	étudiant	credit	achat
1	27	48	non	moyen	non
2	25	45	non	bon	non
3	32	47	non	moyen	oui
4	42	30	non	moyen	oui
5	43	20	oui	moyen	oui
6	48	15	oui	bon	non
7	33	25	oui	bon	oui
8	24	33	non	moyen	non
9	28	22	oui	moyen	oui
10	45	32	oui	moyen	oui
11	28	30	oui	bon	oui
12	39	31	non	bon	oui
13	37	50	oui	moyen	oui
14	41	34	non	bon	non

Q 3.1 Un data scientist propose d'utiliser les données brutes avec un modèle bayésien naïf. Est-ce possible ? Que conseillez vous comme transformation des données dans le cas contraire ? Donner le nombre de paramètres du modèle proposé. Expliciter la prédiction pour l'individu 14 (et le détails des calculs).

Q 3.2 Un autre data scientist propose d'utiliser un perceptron. Est-ce possible sur les données brutes ? Proposer une transformation des données sinon. Illustrer sur quelques itérations le comportement de l'algorithme.

Q 3.3 Un troisième propose un réseau de neurones. Les deux autres lui disent que c'est une mauvaise idée : pourquoi ? Argumenter sur les bénéfices et les inconvénients des trois modèles.

Exercice 4 (5) – Boosting

Soit $f(\mathbf{x}) = a\mathbf{1}_{<\mathbf{w},\mathbf{x}>+b>0} + c$, avec a, b, c des réels, $\mathbf{w}$ un vecteur de la dimension de $\mathbf{x}$ et $<\mathbf{w},\mathbf{x}>$ le produit scalaire ; soit $\mathcal{D} = \{\mathbf{x}^i, y^i\}_{i=1}^n$ un ensemble d'apprentissage et soit la fonction de coût suivante : $L(f) = \sum_{i=1}^n \gamma^i (y^i - f(\mathbf{x}^i))^2$ avec $\gamma^i > 0$ et $\sum_{i=1}^n \gamma^i = 1$.

Q 4.1 Rappeler le principe d'un algorithme de boosting et le rôle des γ^i.
On suppose dans la suite le vecteur $\mathbf{w}$ fixé (en pratique il est tiré au hasard à chaque itération).

Q 4.2 Calculer les dérivées partielles de f par rapport à a et c.

Q 4.3 Calculer les dérivées partielles de $L(f)$ par rapport à a et c. Faites intervenir l'ensemble $X^{\mathbf{w}^+} = \{\mathbf{x}^i | \ <\mathbf{w},\mathbf{x}^i> +b > 0\}$.

Q 4.4 Déduire la valeur optimale pour c. Donner la valeur optimale de a. Commenter.

Q 4.5 Comment calculer b ?