

# Examen ING2 EILCO - Ingénierie Mathématique

Décembre 2024

Nom :

Prénom :

**Total:** 32 points

**Durée:** 2h

**Instructions générales:** L'examen comprend 2 parties (Chacune de ces parties reprenant différentes sous-questions). Vous êtes libres de rédiger vos réponses sur des pages supplémentaires en veillant toutefois à bien indiquer le numéro de chaque question. Une fois l'examen terminé, Assurez vous de bien écrire votre nom (de façon lisible) sur chacune des pages. Répondez à un maximum de questions, en commençant par les questions qui vous semblent les plus abordables.

## Question 1 (18pts)

1. [5pts] Indiquer si les affirmations suivantes sont vraies ou fausses

Vrai / Faux      En régression linéaire, l'estimateur de maximum de vraisemblance suppose que les données sont indépendantes et identiquement distribuées

Vrai / Faux      Soit le noyau  $K$  défini par la matrice  $K(x, y) = \tanh(\alpha(\mathbf{x}^\top \mathbf{y}) + \mathbf{1}^\top \mathbf{x})$  où  $\mathbf{1}$  représente le vecteur  $\mathbf{1} = [1, \dots, 1]$  et  $\tanh$  est la tangente hyperbolique. Le noyau  $K$  est un noyau valide.

Vrai / Faux      Lorsque l'a priori est uniforme, l'estimateur de maximum a posteriori se réduit à un estimateur de maximum de vraisemblance.

Vrai / Faux      Le modèle de régression logistique nécessite que les caractéristiques suivent une distribution Gaussienne.

Vrai / Faux      Dans l'algorithme du perceptron, la mise à jour est donnée par  $\beta \leftarrow \beta + \eta t^{(i)} \tilde{\mathbf{x}}^{(i)}$  où les  $t^{(i)}$  sont les valeurs cibles et les  $\tilde{\mathbf{x}}^{(i)} = [1, \mathbf{x}^{(i)}]$  sont les vecteurs caractéristiques

Vrai / Faux      Dans le cas du modèle de régression logistique, la fonction de vraisemblance peut prendre des valeurs négatives lorsque les vecteurs caractéristiques sont suffisamment différents

Vrai / Faux      Un réseau de neurones entraîné via la mise à jour ADAM convergera toujours vers un minimum global de la fonction de coût

2. [4pts] Déterminer quelles sont, parmi les affirmations suivantes, celles qui sont correctes:

1) En scikit-learn, quelle fonction est utilisée pour diviser les données en un ensemble de test et un ensemble d'entraînement?

- A. `data_split()`
- B. `fit_transform()`
- C. `split_data()`

- D. `train_test_split()`  
 E. `cross_val_score()`
- 2) Que fait la fonction `StandardScaler()` en `scikit-learn`?
- Elle rééquilibre les caractéristiques de façon à ce que chaque caractéristique soit de moyenne nulle et de variance 1.
  - Elle transforme les caractéristiques binaires en valeurs réelles.
  - Elle rééquilibre les caractéristiques de façon à ce que toutes ces caractéristiques soit comprises dans l'intervalle  $[0,1]$
  - Elle supprime les caractéristiques qui contiennent des valeurs NaN ou qui ne sont pas définies dans certaines données.
- 3) Un fois le modèle de régression linéaire entraîné en `scikit-learn`, quelle ligne permet de récupérer les coefficients du modèle?
- `model.coefficients_`
  - `model.coef_`
  - `model.weights_`
  - `model.params_`
- 4) En `scikit-learn`, quelle est la classe utilisée pour entraîner un modèle de régression de type Ridge?
- `LinearRegression()`
  - `RidgeRegression()`
  - `Ridge()`
  - `RidgeClassifier()`
  - `Ridge_Regression()`
- 5) Quel paramètre du modèle `MLPClassifier` contrôle le nombre de neurones dans chaque couche?
- `n_neurons`
  - `hidden_neurons`
  - `hidden_layer`
  - `hidden_layer_sizes`
  - `network_size`
- 6) Quel est l'effet du paramètre `random_state` en `scikit-learn`?
- Il permet d'initialiser la descente de gradient de façon aléatoire
  - Il permet de sélectionner de manière aléatoire un sous-ensemble d'entraînement parmi les données.
  - Il permet de spécifier la fraction des données utilisée pour l'entraînement et pour le test
  - Il permet d'assurer la reproductibilité des résultats en fixant le paramètre d'initialisation du générateur de nombres pseudo-aléatoires.
  - Il permet d'ajouter une perturbation aléatoire aux données d'entraînement
- 7) Parmi les fonctions d'activation suivantes, quelle est celle qui n'est pas implémentée par le modèle `MLPClassifier`?
- `linear`
  - `logistic`
  - `relu`
  - `heaviside`
  - `tanh`
3. [3pts] On suppose qu'on dispose d'un vecteur caractéristique  $x$  pouvant appartenir soit à la classe  $C_0$  soit à la classe  $C_1$  (On suppose que les classes sont mutuellement exclusives). On définit les activations

$$a_1 = \log P(x|M_1) + \log P(M_1) \quad (1)$$

$$a_2 = \log P(x|M_2) + \log P(M_2) \quad (2)$$

Montrer que la probabilité a posteriori du modèle  $M_1$  peut se réécrire à l'aide de la fonction sigmoïde de la manière suivante:

$$P(M_1|x) = \sigma(a_1 - a_2) = \frac{1}{1 + \exp(-(a_1 - a_2))} \quad (3)$$

(Indice: Etant donné deux événements mutuellement exclusifs  $A$  et  $B$ ,  $p(x) = p(x|A)p(A) + p(x|B)p(B)$ )

4. [4pts] On considère un problème de classification à  $K$  classes.
- (a) Donner les nombres de modèles linéaires devant être entraînés dans le cas de chacune des approches “un contre un” et “un contre tous”.
  - (b) En quelques mots (sans donner le pseudo-code), expliquer la différence entre les deux approches.
  - (c) En quelques mots, expliquer comment ces deux approches peuvent conduire à une ambiguïté quant à la classe de certaines régions de l'espace des caractéristiques.
  - (d) Finalement, donner une alternative permettant de résoudre cette ambiguïté
5. [2pts] Donner l'expression de la fonction objectif dans le cas d'une régularisation de type Ridge.

## Question 2 (14pts)

1. [5pts] Indiquer si les affirmations suivantes sont vraies ou fausses

- Vrai / Faux L'algorithme de rétropropagation ne fonctionne que pour des réseaux de neurones dont les couches sont complètement connectées
- Vrai / Faux Une famille de modèles dont l'erreur sur les données d'entraînement est nulle est nécessairement associée à une variance faible.
- Vrai / Faux En régression logistique, les points abhérents peuvent avoir un effet important sur l'estimateur des coefficients de régression
- Vrai / Faux Les modèles Ridge et Lasso supposent tous les deux que les erreurs entre les données exactes et le modèle linéaire sous-jacent suivent une loi Gaussienne
- Vrai / Faux Un niveau de régularisation élevé a tendance à augmenter le biais de la famille de modèles associés.
- Vrai / Faux L'astuce du noyau est particulièrement utile pour entraîner des modèles sur des jeux de données de grande taille.
- Vrai / Faux En validation croisée à  $K$  compartiments, les données sont séparées en  $K$  sous-ensembles et chaque sous-ensemble est utilisé une fois comme ensemble de test, tandis que les  $K - 1$  sous-ensembles restants sont utilisés comme ensembles d'entraînement.
- Vrai / Faux Si un jeu de données est linéairement séparable par un modèle de régression logistique de la forme  $\sigma(\mathbf{w}^T \mathbf{x} + b)$  ou  $\sigma(a) = 1/(1 + e^{-a})$ , alors il reste linéairement séparable pour tout modèle de la forme  $\sigma(\mathbf{w}_c^T \mathbf{x} + b_c)$  où  $\mathbf{w}_c = c\mathbf{w}$  et  $b_c = cb$ .

2. [4pts] On considère l'extrait de code donné à la figure 1. Compléter la fonction `my_linear_regression()` de façon à satisfaire les spécifications. On supposera que l'ordinateur ne dispose pas de la librairie `scikit-learn`. On veillera à bien détailler chaque étape.
3. [5pts] On considère le problème de classification suivant. On dispose de vecteurs caractéristiques constitués de deux caractéristiques  $x_1$  et  $x_2$ , ainsi que d'une valeur cible binaire  $t \in \{0, 1\}$ . Pour chaque vecteur caractéristique, la valeur cible est définie de la manière suivante:

$$t(\mathbf{x}) = \begin{cases} 1 & \text{si } x_2 \geq |x_1| \\ 0 & \text{sinon} \end{cases} \quad (4)$$

(a) [1pt] La fonction  $t(\mathbf{x})$  peut-elle être représentée par un modèle de régression logistique ou par un perceptron? Justifier.

(b) [2pts] On considère une simplification de la fonction (4) donnée par

$$t(\mathbf{x}) = \begin{cases} 1 & \text{si } x_2 \geq x_1 \\ 0 & \text{sinon.} \end{cases} \quad (5)$$

Déterminer l'expression ( $\mathbf{w} = (w_1, w_2)$  et  $b$ ) d'un modèle de perceptron de la forme  $\sigma(\mathbf{w}^T \mathbf{x} + b)$ , ou  $\sigma(a)$  est la fonction d'activation

$$\sigma(a) = \begin{cases} 1 & \text{si } a \geq 0 \\ 0 & \text{sinon} \end{cases}, \quad (6)$$

permettant de représenter  $t(\mathbf{x})$ ,

(c) [2pts] On souhaite à présent revenir aux cibles introduites en (4). Donner l'expression  $(\alpha_1, \alpha_2, \alpha_0, \mathbf{w}, \tilde{\mathbf{w}}, b, \tilde{b})$  d'un réseau de neurones à deux couches, i.e.

$$y(\mathbf{x}) = \sigma(\alpha_1 \sigma(\mathbf{w}^T \mathbf{x} + b) + \alpha_2 \sigma(\tilde{\mathbf{w}}^T \mathbf{x} + \tilde{b}) + \alpha_0) \quad (7)$$

```

In [ ]: 1 import numpy as np
        2 import matplotlib.pyplot as plt
        3
        4
        5 def my_linear_regression(X, t, eta):
        6
        7     '''la fonction doit renvoyer le vecteur des coefficients de régression d'un
        8     modèle entraîné sur base des données X, t'''
        9
        10    # Entrées:  X : Matrice caractéristique.
        11    #           t : vecteur des valeurs cibles
        12    #           eta : taux d'apprentissage
        13    # Sortie:  beta : Vecteur des coefficients de régression
        14
        15
        16    '''À compléter'''
        17
        18
        19
        20
        21    return beta
        22
        23

```

Figure 1: Extrait de code utilisé à la question 2.2.

*permettant de représenter la fonction. Indice: On pourra ré-utiliser l'expression du neurone calculée en (3b) pour l'un des neurones de la couche cachée.*