Corrigé

Examen ING2 EILCO - Ingénierie Mathématique
Rattrapage

Janvier 2025

Nom :
Prénom :

Total: 31 points
Durée: 2h
Instructions générales: Aucun matériel autorisé, pas de calculatrice autorisée
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, pour un taux d'apprentissage suffisamment petit et un nombre suffisamment élevé d'itérations, la descente de gradient donnera toujours une solution qui annule la fonction coût*
Vrai / Faux	*Dans la décomposition biais variance, l'erreur quadratique moyenne diminue lorsque le biais diminue*
Vrai / Faux	*La dérivée de la fonction sigmoïde satisfait $\sigma' = \sigma(\sigma - 1)$*
Vrai / Faux	*Dans la décomposition biais variance, l'erreur quadratique moyenne diminue lorsque la complexité du modèle augmente*
Vrai / Faux	*La régression Ridge correspond à supposer une distribution Gaussienne pour les résidus*
Vrai / Faux	*Dans le cadre de l'astuce du noyeau, la mise à jour est donnée par* $\lambda_j \leftarrow \lambda_j + \frac{2\eta}{N}\left(t^{(j)} - \sum_{i=1}^{N} \lambda_i (\varphi^{(i)})^\intercal \varphi^{(j)}\right)$
Vrai / Faux	*L'a priori de type Laplace utilisé dans le modèle LASSO est donné par* $\frac{1}{2\lambda}\exp\left(-\frac{\lvert x\rvert}{\lambda}\right)$
Vrai / Faux	*La régularisation ne peut être utilisée qu'en régression, pas en classification*

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

   1) *En scikit-learn, quel est le rôle du paramètre α du modèle* Ridge*?*
      - A. *Il détermine le poids de la norme ℓ_1.*
      - B. *Il contrôle l'équilibre entre la fidélité aux données et l'amplitude des coeffients de régression.*
      - C. *Il conduit certains des coefficients à s'annuler exactement.*
      - D. *Il permet de contrôler le taux d'apprentissage*

   2) *Que se passe-t-il si le paramètre α du modèle* Lasso *est fixé à zéro?*

A. *Le modèle se réduit à un modèle de régression linéaire simple*
B. *Le modèle se réduit à un modèle de type Ridge*
C. *Tous les coefficients du modèle s'annulent*
D. *Tous les coefficients du modèle s'annulent sauf le biais.*

3) *En scikit-learn, quelle fonction est utilisée pour diviser les données en un ensemble de test et un ensemble d'entrainement?*
A. data_split()
B. fit_transform()
C. split_data()
D. train_test_split()
E. cross_val_score()

4) *Quel type de frontière de décision le modèle* LogisticRegression *de scikit-learn produit-il?*
A. *Quadratique*
B. *Linéaire*
C. *Non Linéaire*
D. *Polynomiale*
E. *Circulaire*

5) *Que fait la fonction* np.random.normal *de* numpy*?*
A. *Elle génère des entiers aléatoirement dans un intervalle donné*
B. *Elle génère des échantillons issus d'une distribution uniforme*
C. *Elle génère des échantillons Gaussiens*
D. *Elle génère un tableau de valeurs équidistantes*

6) *Parmi les solveurs suivants, lesquels ne sont pas implémentés par le modèle* MLPClassifier *de* Scikit-learn*?*
A. lbfgs
B. sgd
C. adam
D. gradient_descent
E. newton

3. [4*pts*] *On considère le modèle de régression linéaire suivant*

$$\hat{t} = \beta_0 + \beta_1 x \tag{1}$$

(a) *Soit les données* $\{x^{(i)}, t^{(i)}\}_{i=1}^N$ *et soit* $\mathsf{SXX} = \sum_{i=1}^N (x^{(i)} - \overline{x})^2$ *où* $\overline{x}$ *est la moyenne empirique,* $\overline{x} = \frac{1}{N}\sum_{i=1}^N x^{(i)}$. *Montrer que*

$$\mathsf{SXX} = \sum_{i=1}^N \left(x^{(i)} - \overline{x}\right) x^{(i)} \tag{2}$$

(b) *Soit la quantité* SXY *définie par* $\mathsf{SXY} = \sum_{i=1}^N (x^{(i)} - \overline{x})(t^{(i)} - \overline{t})$. *Montrer que*

$$\mathsf{SXY} = \sum_{i=1}^n (x^{(i)} - \overline{x}) t^{(i)} \tag{3}$$

(c) *En se basant sur les résultats* (2) *et* (3), *si* $\hat{\beta}_1$ *et* $\hat{\beta}_0$ *sont les solutions du problème de minimisation par moindres carrés*

$$\hat{\beta}_1, \hat{\beta}_0 = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^n \left(t^{(i)} - (\beta_0 + \beta_1 x^{(i)})\right)^2 \tag{4}$$

montrer que

$$\hat{\beta}_1 = \frac{\mathsf{SXY}}{\mathsf{SXX}} \tag{5}$$

Question 2 (13pts)

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

Vrai / Faux	*La limite quand p tend vers l'infini des normes ℓ_p correspond à l'hypercube* $\max_i \lvert x_i \rvert \leq 1$
Vrai / Faux	*L'estimateur de Maximum de Vraisemblance produit un modèle qui maximise la probabilité d'observer les données*
Vrai / Faux	*La fonction d'entropie binaire croisée est équivalente à une fonction de log-vraisemblance dont les probabilités de classe sont définies par la sortie du modèle*
Vrai / Faux	*Dans le cadre de la régression linéaire, ajouter une pénalité de type Ridge, de coefficient associé $\lambda > 0$, correspond à translater les valeurs propres de la matrice $\boldsymbol{X}^\intercal \boldsymbol{X}$ de λ.*
Vrai / Faux	*Dans l'algorithme de backpropagation, dans l'expression des $\frac{\partial L}{\partial a_i}$ de la couche ℓ en fonction des $\frac{\partial L}{\partial a_i}$ de la couche $\ell + 1$, le nombre de termes dans l'expression dépend du nombre de neurones de la couche $\ell + 1$*
Vrai / Faux	*Lorsque l'a priori est uniforme, l'estimateur de maximum a posteriori se reduit a un estimateur de maximum de vraissemblance*
Vrai / Faux	*Dans l'algorithme du perceptron, la fonction de coût à minimiser est donnée par* $\ell(\beta) = \sum_{i=1}^{N} t^{(i)}(\beta_0 + \beta_1 x_1^{(i)} + \ldots + \beta_D x_D^{(i)})$
Vrai / Faux	*Si un jeu de données est linéairement séparable par un modèle de régression logistique de la forme $\sigma(\boldsymbol{w}^\intercal \boldsymbol{x} + b)$ où $\sigma(a) = 1/(1 + e^{-a})$, alors il reste linéairement séparable par tout modèle de la forme $\sigma(\boldsymbol{w}_c^\intercal \boldsymbol{x} + b_c)$ où $\boldsymbol{w}_c = c\boldsymbol{w}$ et $b_c = cb$.*

2. [4pts] *On considère les extraits de code donnés en page 5. Pour chaque extrait, donner le résultat de la dernière (ou des deux dernières) commandes* `print`

3. [4pts] *On considère un réseau de neurones à 2 couches prenant en entrée un vecteur caractéristique de taille 3 et dont les deux couches contiennent respectivement 3 neurones (pour la première couche) et 1 neurone (pour la couche de sortie). La sortie du réseau est donnée par le scalaire $y(\boldsymbol{x}) \in \mathbb{R}$. Les matrices de poids et biais pour chacune des couches sont données par*

$$\boldsymbol{W}_1 = \begin{bmatrix} 1 & -1 & 0 \\ 0.5 & 2 & -1 \\ -1 & 1 & 1 \end{bmatrix}, \quad \boldsymbol{b}_1 = \begin{bmatrix} 0.5 \\ -1 \\ 1 \end{bmatrix} \tag{6}$$

$$\boldsymbol{W}_2 = \begin{bmatrix} 1 & -1 & 0 \end{bmatrix}, \quad \boldsymbol{b}_2 = -0.5 \tag{7}$$

La $i^{\text{ème}}$ ligne de la matrice $\boldsymbol{W}_\ell$ correspond donc aux poids $\boldsymbol{w}_i^{(\ell)} = [w_{i1}^{(\ell)}, w_{i2}^{(\ell)}, w_{i3}^{(\ell)}]$ du $i^{\text{ème}}$ neurone de la couche ℓ. On suppose que tous les neurones sont munis d'une fonction d'activation de type ReLU (Unité Linéaire Rectifiée), $\sigma(\boldsymbol{x}) = ReLU(x)$ définie par

$$ReLU(x) = \begin{cases} x & si\ x \geq 0 \\ 0 & si\ x < 0 \end{cases} \tag{8}$$

Calculer la valeur de la sortie du réseau, $y(\boldsymbol{x})$, pour un vecteur caractéristique d'entrée donné par $\boldsymbol{x} = [2, -1, 0]$.

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import PolynomialFeatures

x_class1 = np.asarray([-1,-1,-1])
y_class1 = np.asarray([1,2,3])

x_class2 = np.asarray([1,1,1])
y_class2 = np.asarray([1,2,3])



X1 = np.vstack((x_class1.flatten(), y_class1.flatten())).T
X2 = np.vstack((x_class2.flatten(), y_class2.flatten())).T

X = np.vstack((X1, X2))

print(X)
```

Résultat Extrait 1:

$$X = \begin{bmatrix} -1 & 1 \\ -1 & 2 \\ -1 & 3 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix}$$

```
my_poly = PolynomialFeatures(2)

Xpoly = my_poly.fit_transform(X)

print(Xpoly)
```

Résultat Extrait 2:

$$Xpoly = \begin{bmatrix} 1 & -1 & 1 & -1 & 1 & 1 \\ 1 & -1 & 2 & -2 & 1 & 4 \\ 1 & -1 & 3 & -3 & 1 & 9 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 2 & 1 & 4 \\ 1 & 1 & 3 & 3 & 1 & 9 \end{bmatrix}$$

```
x = np.linspace(1,10,10)
t = np.linspace(1,10,10)

print(x)
print(t)
```

```
[ 1.  2.  3.  4.  5.  6.  7.  8.  9. 10.]
[ 1.  2.  3.  4.  5.  6.  7.  8.  9. 10.]
```

```
from sklearn.linear_model import LinearRegression

my_model = LinearRegression().fit(x.reshape(-1,1), t)

xpred = np.linspace(0,5,5)
print(xpred)

```

```
[0.   1.25 2.5  3.75 5.  ]
```

```
model_prediction = my_model.predict(xpred.reshape(-1,1))

print(model_prediction)

```

Résultat Extrait 3: model_prediction = [0   1.25   2.5   3.75   5]

Examen ING2 - EILCO - INGÉNIERIE MATHÉMATIQUE (Rattrapage)

Correction

Question 1.3 (a)

On a $S_{xx} = \sum_{i=1}^{N} (x^{(i)} - \bar{x})^2$

$$= \sum_{i=1}^{N} \left(x^{(i)} - \frac{1}{N} \sum_{j=1}^{N} x^{(j)} \right)^2 =$$

$$= \sum_{i=1}^{N} (x^{(i)})^2 + \frac{1}{N^2} \sum_{i=1}^{N} \left(\sum_{j=1}^{N} x^{(j)} \right)^2 - \frac{2}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} x^{(i)} x^{(j)}$$

$$= \sum_{i=1}^{N} (x^{(i)})^2 + \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} x^{(i)} x^{(j)} - \frac{2}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} x^{(i)} x^{(j)}$$

$$= \sum_{i=1}^{N} x^{(i)} \left(x^{(i)} - \sum_{j=1}^{N} \frac{x^{(j)}}{N} \right)$$

$$= \sum_{i=1}^{N} x^{(i)} (x^{(i)} - \bar{x})$$

Question 1.3. (b)

On a $S_{xy} = \sum_{i=1}^{N} (x^{(i)} - \bar{x})(t^{(i)} - \bar{t})$

$$= \sum_{i=1}^{N} x^{(i)} t^{(i)} - \cancel{\frac{1}{N} \sum_{j=1}^{N} x^{(j)} \sum_{i=1}^{N} t^{(i)}}$$

$$+ \cancel{\sum_{i=1}^{N} x^{(i)} \frac{1}{N} \sum_{j=1}^{N} t^{(j)}} - \sum_{i=1}^{N} \frac{1}{N^2} \sum_{j=1}^{N} \sum_{k=1}^{N} x^{(i)} t^{(k)}$$

$$= \sum_{i=1}^{N} x^{(i)} t^{(i)} - \sum_{i=1}^{N} \frac{1}{N^2} \sum_{j=1}^{N} \sum_{k=1}^{N} x^{(i)} t^{(k)}$$

$$= \sum_{i=1}^{N} x^{(i)} t^{(i)} - \sum_{i=1}^{N} t^{(i)} \frac{1}{N} \sum_{j=1}^{N} x^{(j)} = \sum_{i=1}^{N} (x^{(i)} - \bar{x}) t^{(i)}$$

La dernière ligne provient du fait que $\sum_{i=1}^{N} \frac{1}{N^2} \sum_{j=1}^{N} \sum_{k=1}^{N} x^{(i)} t^{(k)} = \sum_{i=1}^{N} t^{(i)} \frac{1}{N} \sum_{j=1}^{N} x^{(j)}$

Question 1.3 (Corrigé) suite

(c) En prenant les dérivées partielles par rapport à β_0, β_1, on trouve les équations normales

$$\frac{\partial}{\partial \beta_1} \sum_{i=1}^N \left(t^{(i)} - (\beta_0 + \beta_1 x^{(i)})\right)^2 = -2 \sum_{i=1}^N \left(t^{(i)} - (\beta_0 + \beta_1 x^{(i)})\right) x^{(i)}$$

$$\frac{\partial}{\partial \beta_0} \sum_{i=1}^N \left(t^{(i)} - (\beta_0 + \beta_1 x^{(i)})\right)^2 = -2 \sum_{i=1}^N \left(t^{(i)} - (\beta_0 + \beta_1 x^{(i)})\right)$$

On a donc
$$\beta_0 = \frac{1}{N} \sum_{i=1}^N t^{(i)} - \frac{1}{N} \sum_{i=1}^N x^{(i)} \beta_1$$

$$\beta_1 \left(\sum_{i=1}^N (x^{(i)})^2 \right) = \sum_{i=1}^N x^{(i)} t^{(i)} - \sum_{i=1}^N \beta_0 x^{(i)}$$

En substituant l'expression de β_0 dans celle de β_1, on trouve

$$\beta_1 \left(\sum_{i=1}^N (x^{(i)})^2 \right) = \sum_{i=1}^N x^{(i)} t^{(i)} - \sum_{i=1}^N x^{(i)} \frac{1}{N} \sum_{j=1}^N t^{(j)} + \sum_{i=1}^N x^{(i)} \frac{1}{N} \sum_{j=1}^N x^{(j)} \beta_1$$

$$\Rightarrow \beta_1 \left[\sum_{i=1}^N (x^{(i)})^2 - \sum_{i=1}^N x^{(i)} \bar{x} \right] = \sum_{i=1}^N x^{(i)} t^{(i)} - \sum_{j=1}^N t^{(j)} \bar{x}$$

$$\Rightarrow \quad \beta_1 = \frac{\sum_{i=1}^N x^{(i)} t^{(i)} - \sum_{i=1}^N t^{(i)} \bar{x}}{\sum_{i=1}^N (x^{(i)})^2 - \sum_{i=1}^N x^{(i)} \bar{x}}$$

Question 2.3 (Corrigé)

En utilisant les matrices de poids et de biais, la sortie de la 1ère couche est donnée par

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \sigma \left(\begin{bmatrix} 1 & -2 & 0 \\ 0.5 & 2 & -1 \\ -1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0.5 \\ -1 \\ 1 \end{bmatrix} \right)$$

Question 2.3 (Corrigé) suite

où $\sigma(x) = \begin{cases} x & x \geq 0 \\ 0 & \text{sinon} \end{cases}$

pour la couche de sortie, on trouve

$$y(x) = \sigma\left([1, -1, 0] \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} - 0.5 \right)$$

En choisissant $x = [2, -1, 0]$ on retrouve

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \sigma\left(\begin{bmatrix} 1 & -1 & 0 \\ 0.5 & 2 & -1 \\ -1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ -1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0.5 \\ -1 \\ 1 \end{bmatrix} \right)$$

$$= \sigma\left(\begin{bmatrix} 3 \\ -1 \\ -3 \end{bmatrix} + \begin{bmatrix} 0.5 \\ -1 \\ 1 \end{bmatrix} \right) = \sigma\left(\begin{bmatrix} 3.5 \\ -2 \\ -2 \end{bmatrix} \right) = \begin{bmatrix} 3.5 \\ 0 \\ 0 \end{bmatrix}$$

En propageant ces données à travers la couche de sortie, on trouve

$$y(x) = \sigma\left([1, -1, 0] \begin{bmatrix} 3.5 \\ 0 \\ 0 \end{bmatrix} - 0.5 \right) = \sigma(3.5 - 0.5) = 3$$