# Examen ING2-INFO EILCO - Ingénierie Mathématique 3

Automne 2023 (Rattrapage)

| |
|---|
| Nom : |
| Prénom : |

**Total:** 27 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.

**Partie 1 (13pts)**

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

| | |
|---|---|
| *Vrai / Faux* | *Les réseaux de neurones à action directe (feedforward) sont constitués de connections entre neurones orientées uniquement de l'entrée vers la sortie.* |
| *Vrai / Faux* | *Le gradient* $\partial L/\partial \boldsymbol{w}$ *de la fonction d'entropie binaire croisée pour une seule observation* $(\boldsymbol{x}^{(i)}, t^{(i)})$ *et un réseau de neurones consistant en un unique neurone de paramètres* $(\boldsymbol{w}, b)$ *est donné par* $(\sigma(\boldsymbol{w}^T\boldsymbol{x}^{(i)} + b) - t^{(i)})\sigma'(\boldsymbol{w}^T\boldsymbol{x}^{(i)} + b)\boldsymbol{x}^{(i)}$ |
| *Vrai / Faux* | *Lorsqu'il est utilisé pour un ensemble d'observations à* $K$ *classes, le modèle de classification "un contre tous" requiert l'entrainement de* $K$ *modèles de classification binaires au minimum* |
| *Vrai / Faux* | *L'analyse discriminante Gaussienne est un exemple de modèle discriminant* |
| *Vrai / Faux* | *L'apprentissage d'un réseau de neurones par backpropagation nécéssite le calcul de la dérivée de la fonction d'activation* |
| *Vrai / Faux* | *Le bias d'une famille de modèle encode la sensibilité de ces modèles à une perturbation des données d'entrainement tandis que la variance représente la capacité du modèle à capturer la tendance des données d'entrainement.* |
| *Vrai / Faux* | *Dans un réseau de neurones, un neurone muni d'une fonction d'activation sigmoïdale et dont la valeur de préactivation est proche de zéro peut être approximé par un modèle linéaire* |

2. [4*pts*] *On considère les observations représentées à la figure 1 ci-dessous.*

   (a) [1*pts*] *On souhaite entrainer un modèle de régression logistique de la forme* $p(t = 1|\boldsymbol{x}; \boldsymbol{w}) = \sigma(w_0 + w_1x_1 + w_2x_2)$*, par maximum de vraissemblance, sur base des données de la figure 1. Représenter, sur la figure, une possible frontière de décision pour un tel modèle (après entrainement).*

   (b) [1*pts*] *On souhaite maintenant ajouter une pénalité sur le coefficient* $w_0$*. Pour ce faire, on considère la formulation suivante*

$$\underset{\boldsymbol{w}}{\text{argmin}} \quad -\ell\left(\left\{\boldsymbol{x}^{(i)}, t^{(i)}\right\}_{i=1}^{m}; \boldsymbol{w}\right) + \lambda w_0^2 \tag{1}$$

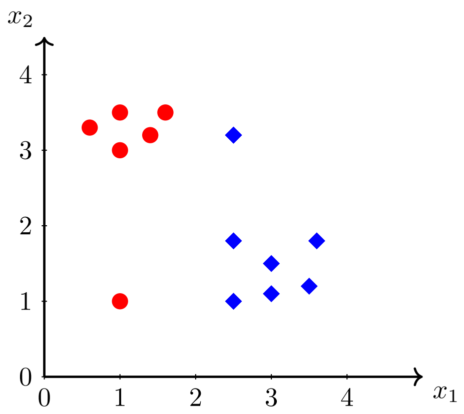


Figure 1: Données utilisées pour la question 2.

*où $\ell(\boldsymbol{x}; \boldsymbol{w})$ représente la fonction de log-vraissemblance. Supposons que $\lambda$ est un paramètre positif très large. Représenter la frontière de décision correspondante.*

(c) [1pts] *On suppose à présent que l'on pénalise des valeurs trop importantes du coefficient $w_1$. La fonction de coût dans ce second cadre se réécrit donc*

$$\underset{\boldsymbol{w}}{\text{argmin}} \quad -\ell\left(\left\{\boldsymbol{x}^{(i)}, t^{(i)}\right\}_{i=1}^{m}; \boldsymbol{w}\right) + \lambda w_1^2 \tag{2}$$

*Représenter la frontière de décision pour ce second modèle.*

(d) [1pts] *On décide finalement de régulariser par rapport au coefficient $w_2$, i.e.*

$$\underset{\boldsymbol{w}}{\text{argmin}} \quad -\ell\left(\left\{\boldsymbol{x}^{(i)}, t^{(i)}\right\}_{i=1}^{m}; \boldsymbol{w}\right) + \lambda w_2^2 \tag{3}$$

*Représenter la frontière de décision de ce troisième et dernier modèle.*

3. [4pts] *On dispose de l'extrait de code donné à la figure 2 et dont le résultat est donné à la figure 3.*

   (a) [2pts] *En modifiant une seule ligne, montrer comment obtenir le résultat donné à la figure 4 (haut).*

   (b) [2pts] *Au lieu de modifier le code comme au point 3a, on souhaite à présent repartir de l'extrait donné à la figure 2 et introduire de la régularisation afin d'obtenir le résultat donné à la figure 4 (bas). Expliquer comment procéder et fournir les lignes de code correspondant à la modification.*

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

x = np.linspace(-2,2,10)
beta = [.1, .1, 1]
t = beta[0] + beta[1]*x + beta[2]*x**2
tnoisy = t + np.random.normal(0,.2, len(x))


maxDegree = 12
myPolynomialFeatures = PolynomialFeatures(maxDegree)
Xtilde = myPolynomialFeatures.fit_transform(x.reshape(-1,1))
xtest = np.linspace(-2,2,100)
XtildeTest = myPolynomialFeatures.fit_transform(xtest.reshape(-1,1))

prediction = np.zeros((len(xtest), 1))

myLinearRegression = LinearRegression()
myLinearRegression.fit(Xtilde, tnoisy)


prediction = myLinearRegression.predict(XtildeTest)

plt.plot(xtest, prediction, c='b', alpha=.1)
plt.scatter(x, tnoisy, c='r')
plt.plot(xtest,prediction, color='blue', linestyle='--')
plt.show()
```

Figure 2: Extrait 1.

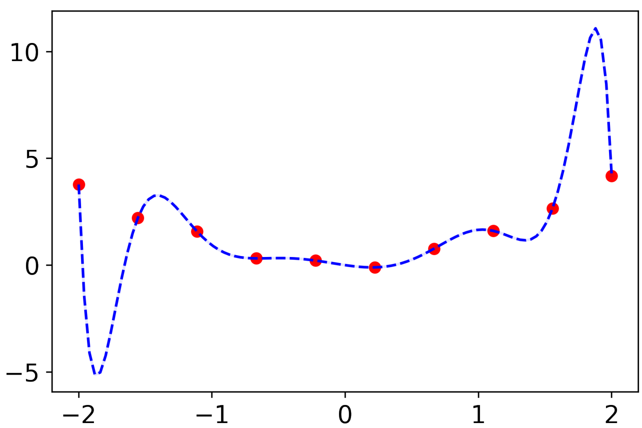


Figure 3: Résultat de l'extrait 1.

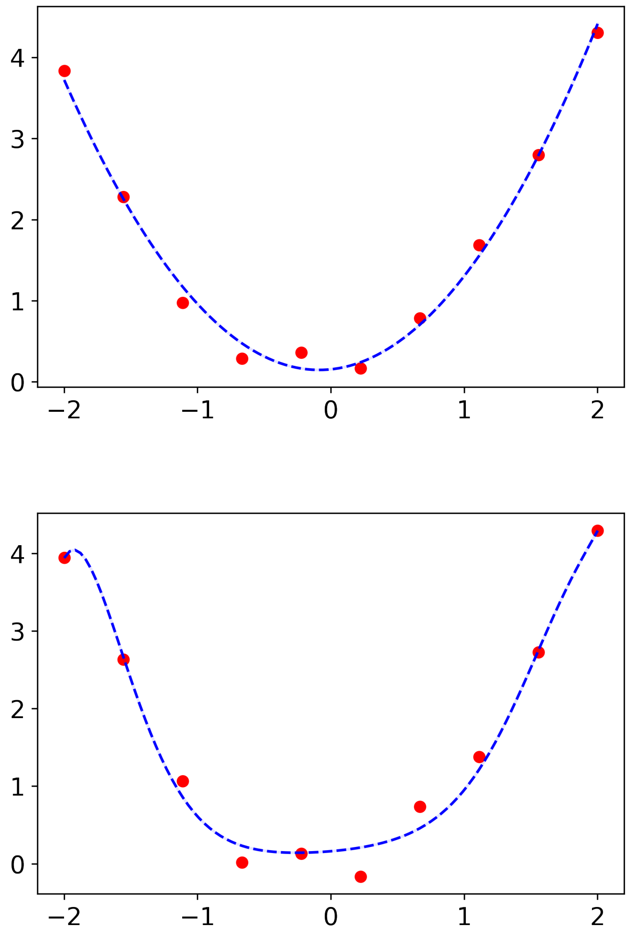


Figure 4: Résultats souhaités pour les questions 3a (haut) et 3b (bas).

**Partie 2 (14pts)**

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

| | |
|---|---|
| *Vrai / Faux* | *La solution des équations normales est donnée par* $\boldsymbol{\beta} = \boldsymbol{X}^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{t}$ *où* $\boldsymbol{t}$ *est le vecteur des valeurs cibles,* $\boldsymbol{X}$ *la matrice des caractéristiques et* $\boldsymbol{\beta}$ *le vecteur des coefficients de régression.* |
| *Vrai / Faux* | *Étant donné une distribution de probabilité* $p(x;\theta)$ *paramétrée par* $\theta$*, l'estimateur de maximum de vraissemblance pour un échantillon de* $m$ *observations indépendantes est donné par* $\theta_{MLE} = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{m} p(x_i;\theta)$ |
| *Vrai / Faux* | *La distribution de Laplace est donnée par* $Lap(x\|\mu, b) = \frac{1}{2b}\exp\left(-\frac{\|x-\mu\|^2}{b}\right)$ |
| *Vrai / Faux* | *Étant donné un ensemble de données comprenant deux classes distinctes, il est toujours possible de séparer les deux classes en considérant un nombre suffisamment grand de caractéristiques polynomiales de degré suffisamment élevé* |
| *Vrai / Faux* | *Le modèle de régression logistique peut séparer des classes qui ne sont pas linéairement séparables pourvu qu'il soit combiné avec des caractéristiques polynomiales* |
| *Vrai / Faux* | *La fonction sigmoïde est donnée par* $\sigma(x) = e^x/(e^x+1)$ |
| *Vrai / Faux* | *La validation croisée à* $k$ *blocs divise un échantillon de données en* $k$ *sous-échantillons parmi lesquels l'un des sous-échantillons est utilisé comme ensemble de validation et les* $k-1$ *sous-échantillons restants sont utilisés comme ensemble d'entrainement* |

2. [6pts] *On considère les observations représentées à la figure 7. On souhaite assembler un réseau de neurones dont les frontières de décision soient semblables aux lignes en pointillé (les lignes en biais correspondent aux droites* $x_2 = x_1$ *et* $x_2 = 2 - x_1$*).*

   (a) [3pt] *On commence par définir 4 neurones dont les frontières de décision sont représentées à la figure 7. Donner l'expression de chacun de ces neurones (fonction d'activation + coefficients de régression) en considérant un schéma semblable à celui donné à la figure 5. On supposera que les zones bleues correspondent à la sortie* $1$ *tandis que les zones rouges correspondent à la sortie* $0$*.*

   (b) [2pts] *Afin d'obtenir un réseau permettant de séparer les données rouges (points) des données bleues (losanges), on considère l'architecture représentée à la figure 6. En supposant que les neurones de la première couche sont donnés par les neurones obtenus au point (a), donner l'expression des deux neurones de la seconde couche. Les sorties de ces neurones sont représentées à la figure 8.*

   (c) [1pts] *Finalement, on termine le réseau par un unique neurone dont la sortie sera donnée par la combinaison des sorties des deux neurones de la seconde couche. Le neurone de sortie doit donc implémenter une simple opération 'OU' et renvoyer un '1' si l'un* <u>ou</u> *l'autre des neurones de la seconde couche renvoie un '1'. Donner l'expression de ce dernier neurone.*

3. [3pts] *Expliquer la différence entre régularisation Ridge et régularisation LASSO.*

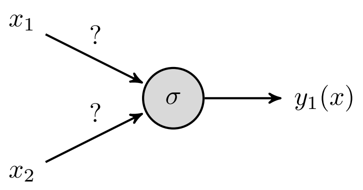


Figure 5: Schéma de neurone utilisé pour la question 2.2

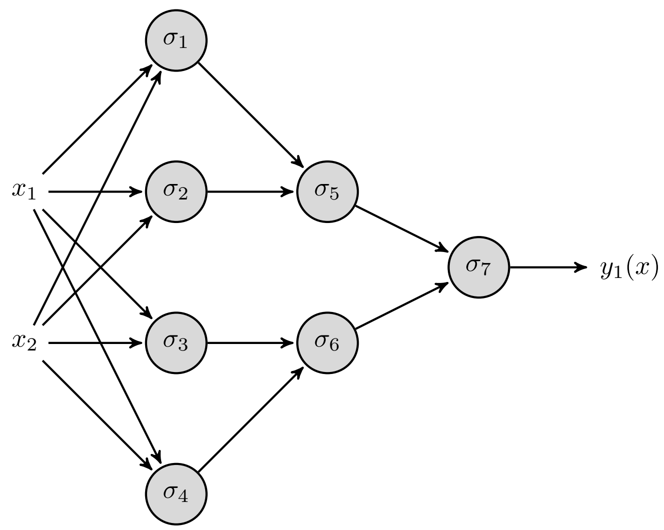


Figure 6: Architecture utilisée pour la question 2.2. Les neurones de la première couche correspondent aux neurones 1 à 4 de la figure 7.

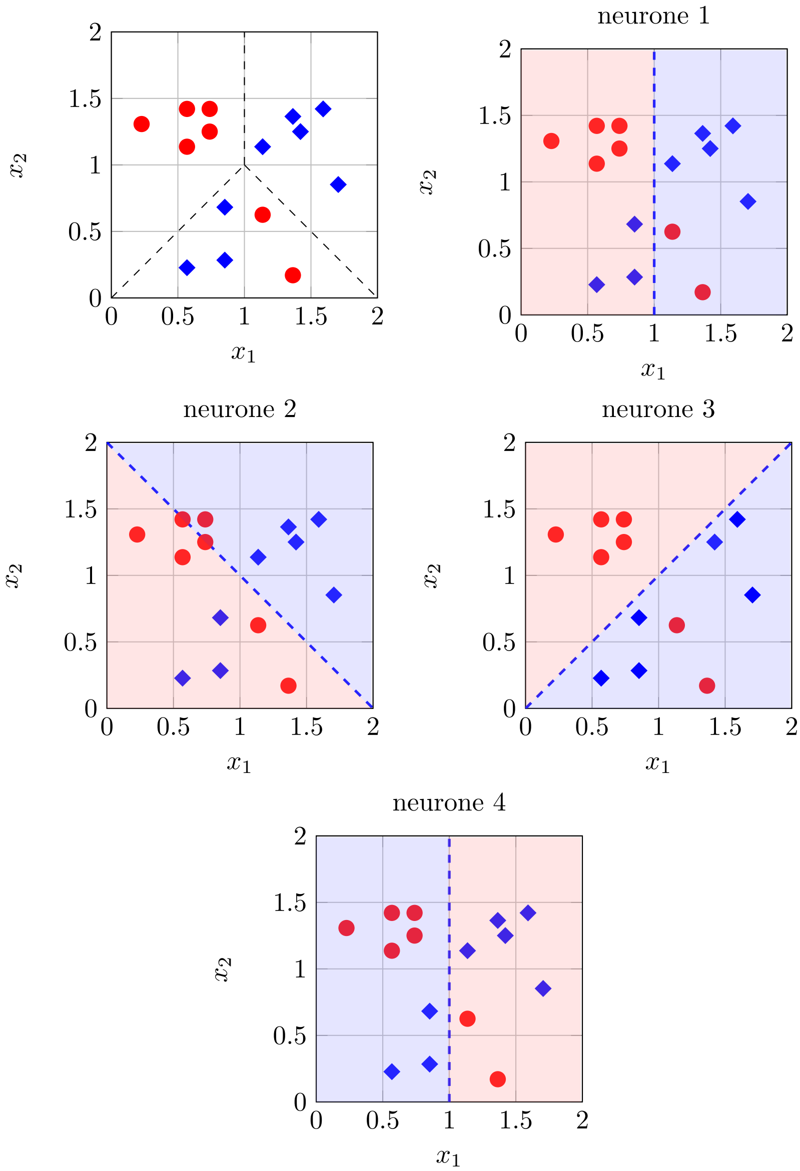


Figure 7: Données et frontières de décision pour les neurones de la première couche (question 2.2).

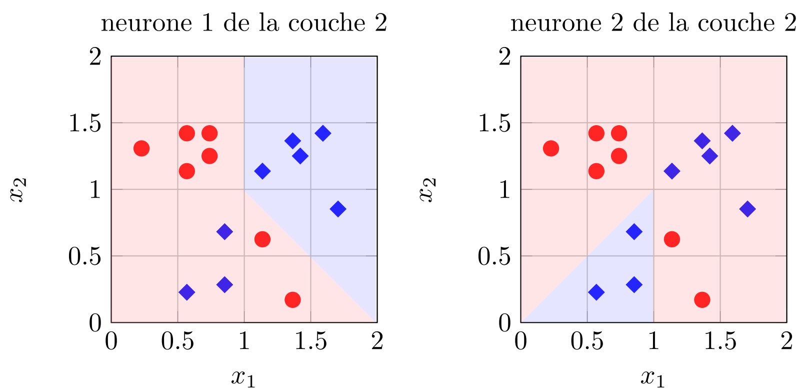


Figure 8: Sorties des neurones de la seconde couche (question 2.2).