

Consignes

- Le TP se fait en groupe de 2 ou 3 personnes
- Le travail doit être démarré durant les séances de TP, et terminé en autonomie
- Un compte-rendu **obligatoire** au format **pdf** doit être soumis pour chaque groupe avant le **Jeudi 04/12/2025**, 23h55.
- Dans le compte rendu vous présenterez le code utilisé pour résoudre chaque partie ainsi que les résultats obtenus et l'interprétation détaillée des résultats le cas échéant. Le détail de vos approches et vos tentatives (potentiellement infructueuses) sera valorisé.
- En plus du compte-rendu, un ou plusieurs notebooks pourront être déposés.
- Pour la partie II, le langage de programmation est laissé au choix. Je conseille cependant d'utiliser **python** (version ≥ 3.7) ou **R**

Partie I Problèmes à la main

I.1 Régression logistique (très important)

Soit Y une variable aléatoire binaire. On connaît la distribution de X , conditionnellement à la valeur de Y :

$$\begin{aligned} X \mid Y = 0 &\sim \mathcal{N}(\mu_0, \sigma^2) \\ X \mid Y = 1 &\sim \mathcal{N}(\mu_1, \sigma^2) \end{aligned}$$

1. Quelle loi peut-on utiliser pour modéliser Y ?
2. On rappelle que $\varphi(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ est la densité de probabilité d'une variable aléatoire distribuée selon $\mathcal{N}(\mu, \sigma^2)$. Déterminer la densité de probabilité de X $p(x)$.
3. Quelle est la principale difficulté pour calculer $p(x)$?
4. Déterminez β_0 et β_1 tels que le logarithme des rapports des chances (*log odds ratio*) puisse s'exprimer comme

$$\ln\left(\frac{\mathbb{P}[Y = 1 \mid X = x]}{\mathbb{P}[Y = 0 \mid X = x]}\right) = \beta_0 + \beta_1 x \quad (1)$$

On posera $\mathbb{P}[Y = 1 \mid X = x] = \pi(x)$, la probabilité que $Y = 1$ étant donné que $X = x$.

5. Supposons les valeurs de β_0 et β_1 connues, comment peut-on prendre une décision à partir de tout ça ?
6. Représenter graphiquement (par ordinateur) la densité de probabilité de $X \mid Y = 1$, celle de $X \mid Y = 0$, et $\pi(x)$ pour $\mu_0 = 0$, $\mu_1 = 5$ et
 - $\rightarrow \sigma = 1$, puis
 - $\rightarrow \sigma = 3$
7. La décision associée change-t-elle ? Commentez.

8. On suppose que l'on dispose d'un jeu de données $\{(x_i, y_i)\}_{1 \leq i \leq n}$, c'est-à-dire un ensemble de n paires x et y . On fera l'hypothèse que les (x_i, y_i) sont i.i.d. et distribués selon la loi jointe de X et Y

- Que signifie l'hypothèse "i.i.d." ? Qu'est ce que cela représente concrètement pour notre jeu de données ?
- Comment estimer β_0 et β_1 à l'aide de la méthode du maximum de vraisemblance ?

I.2 Perceptron

On a les données suivantes

$$\begin{aligned} f([0, 0]) &= 0 & f([0, 1]) &= 1 \\ f([1, 0]) &= 1 & f([1, 1]) &= 0 \end{aligned}$$

1. Représenter graphiquement ce problème
2. Quelle fonction logique essayons nous de représenter ? Pourquoi ne pouvons nous pas trouver un perceptron simple pour la modéliser ?
3. Proposer un perceptron à **couches cachées** pour résoudre ce problème de classification (une piste étant de penser ce problème à l'aide de portes logiques).

I.3 Régression ReLU

On rappelle que la fonction d'activation ReLU est donnée par $\text{ReLU}(x) = \max(0, x)$. On se donne les fonctions f et g affines par morceaux:

$$f(x) = \begin{cases} 2x - 1 & \text{si } x \leq 1 \\ 1 & \text{si } x > 1 \end{cases} \quad g(x) = \begin{cases} 0 & \text{si } x < 0 \\ x & \text{si } 0 < x \leq 1 \\ 1 & \text{si } x > 1 \end{cases} \quad (2)$$

1. Représenter graphiquement f . puis l'écrire comme $f = a_2 \circ \text{ReLU} \circ a_1$, où a_1 et a_2 sont des fonctions affines ($a_i : x \mapsto \alpha_i x + \beta_i$) que l'on explicitera. Faire de même pour g (a_2 sera probablement une fonction à plusieurs variables $a_2 : \mathbf{x} \mapsto \langle \boldsymbol{\alpha}_2, \mathbf{x} \rangle + \beta_2$)



2. Écrire les fonctions définies graphiquement à l'aide de composition de fonctions ReLU et de fonctions affines (possiblement à plusieurs variables).

3. On cherche à approcher la fonction $h : x \mapsto \begin{cases} \sin(\pi x) & \text{si } x \in [0, 1] \\ 0 & \text{sinon} \end{cases}$

- (a) On pose, pour $\alpha > 0$: $g_\alpha(x) = \begin{cases} 2\alpha x & \text{si } x \in [0, 1/2] \\ -2\alpha x + 2\alpha & \text{si } x \in [1/2, 1] \end{cases}$. À l'aide des questions précédentes, écrire g_α comme composition de fonctions ReLU et de fonctions affines.
- (b) Quel est le paramètre α optimal pour approcher h à l'aide de g_α , au sens de la norme L_2 ? On rappelle que $\|h - g_\alpha\|_{L_2}^2 = \int_{\mathbb{R}} (h(x) - g_\alpha(x))^2 dx$.

I.4 CNN

I.4.1 Opérateur linéaire

On définit la convolution 1D comme l'opération \star définie pour un vecteur $y \in \mathbb{R}^n$ et un noyau $k \in \mathbb{R}^p$ par $(y \star k) \in \mathbb{R}^{n-p+1}$, où la i ème composante est donnée par

$$(y \star k)_i = \sum_{j=1}^p y_{i+j-1} k_j$$

1. On s'intéresse à un noyau de convolution 1D que l'on note $k = [\alpha, \beta, \gamma]$. Écrire la matrice C_k tel que

pour un vecteur $y \in \mathbb{R}^n$, $C_k y = \begin{bmatrix} (y \star k)_1 \\ (y \star k)_2 \\ \vdots \\ (y \star k)_{n-p+1} \end{bmatrix}$

2. On considère la matrice

$$D = \frac{1}{h} \begin{bmatrix} -1 & +1 & 0 & 0 & \dots & 0 \\ 0 & -1 & +1 & 0 & \dots & 0 \\ & & & \vdots & & \\ 0 & 0 & \dots & 0 & -1 & +1 \end{bmatrix} \quad (3)$$

Écrire le noyau de convolution associé. Si on suppose que le vecteur d'entrée $y \in \mathbb{R}^n$ est défini par $y^T = [f(0), f(\frac{1}{n-1}), f(\frac{2}{n-1}), \dots, f(1)]$ pour $f : [0, 1] \rightarrow \mathbb{R}$ dérivable et $n > 1$, $h = \frac{1}{n-1}$. Que représente Dy ?

3. On considère un noyau de convolution $k = \begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}$, et on s'intéresse à la convolution en 2D d'une image assimilée à une matrice $x \in \mathbb{R}^{3 \times 3}$. Après avoir écrit l'image comme un vecteur colonne, déterminer la matrice C_k tel que $y = C_k x$ soit le vecteur colonne correspondant à la convolution 2D de x par k . On appliquera une convolution sans *padding* et avec un *stride* de 1.
4. Même question pour un noyau de convolution de taille 3×3 , et pour une matrice de taille $n \times n$?
5. On définit le *sparsity index* s d'une matrice comme le nombre d'éléments nuls d'une matrice divisé par le nombre total d'éléments de la matrice. Quel est le *sparsity index* de C_k ?
6. Quel est le nombre d'éléments et le *sparsity index* de la matrice associée à une couche dense de neurones pour passer d'un vecteur de taille 9 à un vecteur de taille 4

I.4.2 Équivariance

Soit $k : \mathbb{R}^2 \rightarrow \mathbb{R}$. Pour $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, on considère l'opérateur de convolution "formel":

$$(f \star k)(\mathbf{x}) = \int_{\mathbb{R}^2} f(\mathbf{z})k(\mathbf{z} - \mathbf{x}) d\mathbf{z} \quad (4)$$

1. On définit l'opérateur de translation φ_y (qui s'applique sur des fonctions), défini par $\varphi_y(f)(\mathbf{x}) = f(\mathbf{x} - \mathbf{y})$. Montrer que $\varphi_y(f) \star k = \varphi_y(f \star k)$.
2. Discuter du rôle de la convolution et du pooling dans les CNN

On suppose que l'on a entraîné les poids d'un noyau de convolution k , qui permet de détecter l'emoji "🐶". On définit l'opération \diamond comme

$$(f \diamond k)(\mathbf{x}, \theta) = \int_{\mathbb{R}^2} f(\mathbf{z})k(R_{-\theta}(\mathbf{z} - \mathbf{x})) d\mathbf{z} \quad (5)$$

où $R_\theta = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$

1. Calculer R_0 , $R_{2\pi}$, R_π , $R_{\theta_1}R_{\theta_2} - R_{\theta_1+\theta_2}$ pour $\theta_1, \theta_2 \in \mathbb{R}$.
2. Que permettent de faire les opérations suivantes
 - $\mathbf{x} \mapsto \max_{0 \leq k \leq 3}(f \diamond k)(\mathbf{x}, k\pi/2)$
 - $\mathbf{x} \mapsto \frac{1}{2\pi} \int_0^{2\pi} (f \diamond k)(\mathbf{x}, \theta) d\theta$
 - Quels(s) frein(s) ou difficulté(s) pouvez vous envisager à l'implémentation de cette dernière transformation en pratique ?

I.5 Softmax

1. On considère une version modifiée du softmax qui dépend d'un paramètre $T > 0$:

$$S_T : z = (z_1, \dots, z_n) \mapsto \text{Softmax}(z/T) = \left(\frac{e^{z_1/T}}{\sum_{i=1}^n e^{z_i/T}}, \dots, \frac{e^{z_n/T}}{\sum_{i=1}^n e^{z_i/T}} \right) \quad (6)$$

- (a) Soit un vecteur $z \in \mathbb{R}^n$, et $T > 0$. On pose $(s_1, \dots, s_n) = S_T(z)$. Calculer $\sum_{i=1}^n s_i$.
- (b) Calculer $S_T(z)$ pour $z = (-2, 3, 4, 1, 0)$ et $T = 1$, puis $T = 0.2$ et enfin pour $T = 10$
- (c) Que vaut $S_T(z)$ quand $T \rightarrow +\infty$
- (d) Que vaut $S_T(z)$ quand $T \rightarrow 0^+$
- (e) On souhaite tirer aléatoirement un des éléments de z en suivant les poids donnés par $S_T(z)$. Discuter l'influence de T sur cet échantillonnage.

Partie II Application à un jeu de données

Dans le compte rendu, en plus des réponses rédigées, vous présenterez:

- le code utilisé pour résoudre chaque partie
- les résultats obtenus

L'évaluation est principalement sur votre capacité d'analyser, de critiquer et d'interpréter les résultats. Ainsi, il est essentiel d'expliquer clairement vos conclusions.

Le jeu de données peut se trouver sur Campus (`bike_daily.csv`), ou sur <https://archive.ics.uci.edu/dataset/275/bike+sharing+dataset> (choisir le jeu de données `day.csv`). Une description des colonnes du fichier se trouve tableau 1

II.1 Analyse exploratoire

Effectuer une analyse exploratoire du jeu de données: type des variables, valeurs manquantes, statistiques descriptives, figures informatives etc...

II.2 Classification

Nous cherchons tout d'abord à prédire si le nombre de vélos utilisés au total, donnée par la colonne `cnt` dépasse le seuil de 4000 en utilisant les variables météorologiques: `atemp`, `temp`, `hum`, `windspeed`

Nous allons utiliser un réseau de neurones:

1. Expliquer quelles sont les données que l'on va utiliser en entrée du réseau de neurones (traitement des variables catégorielles ? traitement des variables continues ?).

Feature	Description
<code>yr</code>	Année
<code>mnth</code>	Mois
<code>holiday</code>	Jour férié (1) ou non (0)
<code>weekday</code>	Jour de la semaine (0: Dimanche, ..., 6: Samedi)
<code>workingday</code>	Jour ouvré (1) ou non (0)
<code>weathersit</code>	0: Temps clair; 1: Couvert; 2: Pluie Faible 3: Pluie Intense; 4: Tempête
<code>temp</code>	Température normalisée en Celsius (valeurs divisées par 41)
<code>atemp</code>	Température ressentie normalisée en Celsius (valeurs divisées par 50)
<code>hum</code>	Humidité normalisée (valeurs divisées par 100)
<code>windspeed</code>	Vitesse du vent normalisée (valeurs divisées par 64)
<code>casual</code>	Nombre d'usagers non abonnés louant un vélo
<code>registered</code>	Nombre d'usagers abonnés louant un vélo
<code>cnt</code>	Nombre de vélos loués au total

Table 1: Description du jeu de données

2. Diviser les données pour avoir données d'apprentissage et des données de test (80/20), en vérifiant que les proportions des classes soient les mêmes pour les données d'apprentissage et de test.
3. Justifier la fonction coût que vous utilisez.
4. Valider les performances de votre modèle sur les données de test en regardant détaillant et **commentant** la matrice de confusion, les courbes ROC, AUC et tout ce qui vous semblera pertinent.
5. Utiliser un modèle de régression logistique classique pour la même tâche.
6. Commenter vos différents résultats en comparant les méthodes, nombre de paramètres, et tout ce qui vous paraîtra pertinent.

II.3 Régression de série temporelle (RNN)

On cherche à prédire le nombre de vélos qui vont être utilisés le lendemain à partir des données du jour même, et des jours précédents.

1. Faites une analyse exploratoire de la série temporelle. Représenter graphiquement la série temporelle. Quelles observations pouvez vous faire ?
2. Expliquer quelles sont les données que l'on va utiliser en entrée du réseau de neurones (traitement des variables catégorielles ? traitement des variables continues ?).
3. Diviser les données pour avoir des données d'apprentissage et des données de test (80/20)
4. Créer l'ensemble d'apprentissage, en utilisant un paramètre de longueur de fenêtre `window_length` de 2, c'est-à-dire que l'on va utiliser $[x_n, \dots, x_{n+window_length-1}]$ pour prédire $x_{n+window_length}$.
5. Choisir et entraîner un modèle de réseau récurrent pour résoudre cette tâche.
6. Présenter les choix, les différents essais, les statistiques durant l'entraînement
7. Analyser et interpréter en détail les performances du modèle entraîné (graphiquement, MSE/RMSE etc)
8. Comparer le modèle avec un modèle plus "classique" du type ARMA, ou ARIMA par exemple.
9. Commenter vos différents résultats en comparant les méthodes, nombre de paramètres, et tout ce qui vous paraîtra pertinent.