TP : Réseau de Neurones

Indications générales :

1. Le TP se fait impérativement en groupe de 2 à 3 personnes.
2. Le travail doit être démarré durant la séance de TP, à terminer chez soi pour être remis sur Campus avant le 03/12/2024 à 23h55.
3. Un compte rendu obligatoire en format PDF doit être soumis par chaque groupe avant le 03/12/2024 à 23h55.
4. Dans le compte rendu vous présentez le code utilisé pour résoudre chaque partie ainsi que les résultats obtenus et l'interprétation détaillée des résultats le cas échéant.
5. 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.
6. Les codes sont donnés en Python et/ou R-Studio. Mais si vous êtes plus à l'aise avec un autre langage, n'hésitez pas à l'utiliser.

Problème I : Réseau de neurones (à la main)

Dans cet exercice, l'objectif est de mettre à jour les paramètres d'un perceptron simple (pas de couches cachées) en considérant les paramètres initiaux suivants : $W = [1; 1; 1; 1]$ pour le vecteur des poids et un biais $b = 0$ avec une fonction d'activation de type hardlim (voir diapositive 9 du cours). Cela vise à résoudre le problème de classification suivant :

Observations	X_1	X_2	X_3	X_4	Y (target t_i)
P_1	1	1	1	0	1
P_2	0	1	0	0	0
P_3	1	1	1	1	1
P_4	0	0	0	1	0
P_5	0	0	0	0	0

Pour effectuer la mise à jour des paramètres, suivez la même méthode proposée dans le cours, comme décrit dans les diapositives 10 et 11, en injectant les

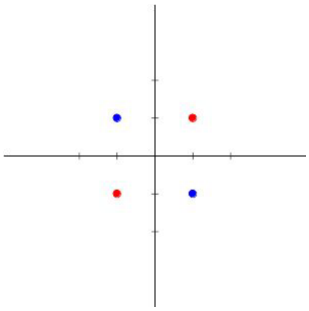
observations $(P_i, i = 1, ..., 5)$ une par une dans le réseau et en corrigeant les paramètres en fonction de l'erreur calculée (en considérant un taux d'apprentissage $\alpha = 1$).

L'objectif est d'appliquer manuellement le principe de la descente du gradient dans le contexte de l'optimisation des performances d'un réseau de neurones.

Continuez à mettre à jour le réseau jusqu'à parvenir à un réseau de neurones ayant un vecteur de poids et un biais qui donnent une classification parfaite des observations $P_i, i = 1, ..., 5$.

Problème II : Problème de classification par un ou plusieurs « Simple Linear Perceptron » (à la main)

Dans cet exercice, il s'agit de résoudre le problème de classification suivant à l'aide de perceptrons (un ou plusieurs) simple (pas de couches cachées) avec deux neurones au niveau de la couche d'entrée et une fonction d'activation linéaire.



N.B. La solution doit être faite **« à la main »** sans aucune implémentation sous R ou Python et avec le moins de calcul possible, même sans faire de calcul. Le plus important est d'expliquer le raisonnement d'une manière claire et logique et de dessiner l'architecture du/des réseau/x considéré/s.

Problème III : Régression Logistique (à la main)

Nous cherchons à expliquer une variable binaire Y (supposée suivre une distribution de Bernoulli $B(p)$) à travers une variable quantitative continue notée X. On suppose que ${}^{X}/_{Y=1} \sim N(\mu_1, \sigma^2)$ et que ${}^{X}/_{Y=0} \sim N(\mu_0, \sigma^2)$. Notez que la fonction de densité de probabilité d'une variable aléatoire suivant une loi normale $N(\mu, \sigma^2)$ est donnée par :

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \text{avec } x \in \mathbb{R}.$$

1. Déterminez la fonction de densité de probabilité de X (appelée la distribution de mélange gaussien).

2. Calculez la probabilité conditionnelle

$$p(x) = \mathbb{P}\left[{}^{Y=1}/_{X=x}\right].$$

3. En déduire une expression linéaire de $\ln\left[\frac{p(x)}{1-p(x)}\right]$ (de la forme $\beta_0 + \beta_1 x$). Donnez aussi l'expression de $p(x)$ en fonction de $(\beta_0 + \beta_1 x)$.
4. Expliquez pourquoi, en pratique, la détermination de $p(x)$ pour une valeur donnée de x est difficile en utilisant l'expression de la partie (3).
5. Sous l'hypothèse que :

$$\ln\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \beta_1 x.$$

Proposez une méthode pour estimer les paramètres β_0 et β_1 en se basant sur un ensemble de données de la forme $(x_i, y_i)_{i=1,\ldots,n}$ (n représente le nombre des observations dans la base de données). Veuillez expliquer, en détail, les différentes étapes de la méthode sans résoudre le système à deux équations que vous allez obtenir.
Indication : Méthode du Maximum de Vraisemblance (voire diapositive 34 du cours).

6. Une fois que les estimateurs $\widehat{\beta_0}$ et $\widehat{\beta_1}$ des paramètres β_0 et β_1 sont calculés, expliquez comment, en pratique, le modèle logistique est utilisé pour classifier un nouvel individu x_{new} en 0 ou 1.

Problème IV : Classification

On suppose que R-Studion et/ou Python sont installés. Nous commençons par importer les librairies utiles.

Dans cet exercice, il s'agit d'implémenter **un réseau de neurones** de haute performance pour classifier des personnes, selon plusieurs caractéristiques, en risque élevé (« 1 ») ou risque faible (« 0 ») d'avoir une cardiopathie. Ainsi, à partir du répertoire en ligne "Échantillons de données" de Campus, considérez l'échantillon de données intitulé « HD_Complete_Data ». Respecter les consignes suivantes :

1. Commencez par effectuer une analyse exploratoire descriptive de votre base de données (typologie des variables, valeurs manquantes, distributions des observations pour les différentes variables, boxplots, etc.).
2. Expliquer comment vous avez réglé le problème des données catégorielles.
3. Diviser votre base de données en 80 % pour l'apprentissage et 20 % pour le test. (Indication : la proportion des classes « 1 » et « 0 » de la variable dépendante doit être la même pour les données d'apprentissage et les données de test).
4. Vous êtes libre de choisir les paramètres de votre modèle afin d'optimiser la performance (Vous serez évalué sur les résultats de votre modèle appliqué aux **données test**).

5. Expliquer comment vous avez fait pour réduire le surapprentissage.
6. Analyser et interpréter, en détail, la performance de votre modèle (sur les données test) :
   (a) Erreurs de classification.
   (b) Matrice de confusion détaillée.
   (c) Courbe ROC et AUC.
   (d) Etc.
7. Donner une conclusion générale sur votre modèle comportant vos recommandations et remarques.
8. Enfin, comparez la performance de votre modèle avec un modèle de régression logistique classique, tel que développé dans le Problème III, et interprétez les résultats.

Problème V : Analyse d'une série temporelle

Dans cet exercice, il s'agit d'implémenter un **réseau de neurones de type RNN** de haute performance pour prédire le taux d'ammonium mensuel dans l'eau du Danube à un point géographique bien défini. Ainsi, à partir du répertoire en ligne "Échantillons de données" de Campus, considérez l'échantillon de données intitulé « Danube ammonium level Time Series ». Cet échantillon représente le taux d'ammonium dans l'eau du Danube à un point géographique bien défini au début de chaque mois de 01/01/1996 jusqu'à 01/12/2017 c.à.d. $12 \times 22 = 264$ observations. Construisez votre modèle en respectant les consignes suivantes :

1. Considérer les observations de 01/01/1996 jusqu'à 01/12/2016 pour l'apprentissage ($12 \times 21 = 252$ observations) et de 01/01/2017 jusqu'à 01/12/2017 pour le test ($12 \times 1 = 12$ observations pour la prédiction). De plus, normaliser les données pour que les observations soient dans $[0, 1]$.
2. Avant de construire le modèle, préparer votre échantillon pour qu'il soit utilisable dans des réseaux de neurones de type RNN (voir diapositive 51 du cours) . On vous propose de représenter les données sous deux colonnes : la première pour le taux d'ammonium à un instant t et la deuxième pour le taux à l'instant $t + 1$. (Cette représentation sera faite pour les données d'apprentissage et celles de test)
3. Penser à remodeler (Reshape) l'ensemble des données d'entrées (c.à.d. première colonne de l'échantillon obtenu dans le point 2) dans un format qui convient le mieux au modèle RNN.
4. Vous êtes libre de choisir les paramètres de votre modèle afin d'optimiser la performance (Vous serez évalué sur les résultats de votre modèle appliqué aux **données test**). N'oublier pas de retransformer (l'inverse de la normalisation qui a été faite dans le point 1) les résultats obtenus en prédiction afin d'obtenir des vrais taux d'ammonium.

5. Analyser et interpréter, en détail, la performance de votre modèle (ex : par des mesures de performance classiques : MSE, RMSE, etc. et par des analyses graphiques)
6. Comparer la performance de votre modèle à un modèle de série chronologique classique comme ARMA, ARIMA etc. après l'avoir bien calibré.
7. Donner une conclusion générale sur votre travail comportant vos recommandations et remarques.