REPUBLIQUE DE COTE D'IVOIRE
Union – Discipline – Travail

Institut National Polytechnique Félix Houphouët-Boigny

REPUBLIC OF IVORY COAST
Union – Discipline – Work

National Polytechnic Institute Houphouët-Boigny Félix

Master Data Science - Big Data

Les réseaux de neurones

Travaux Pratiques

Questions

Q1. Écrivez un script Python qui permet de calculer les séries de Taylor suivantes (approximant 6 fonctions de x) pour N termes (jusqu'à n = N) :

$$f_1(x) = \frac{1}{1-x} = \sum_{n=0}^{\infty} x^n \qquad f_2(x) = \exp(x) = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

$$f_3(x) = \cos(x) = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!} \qquad f_4(x) = \sin(x) = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!}$$

$$f_3(x) = \ln(1+x) = \sum_{n=0}^{\infty} (-1)^{n+1} \frac{x^n}{n} \qquad f_6(x) = tan^{-1}(x) = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)}$$

Pour chacune des séries, comparez l'approximation obtenue (avec N=50 termes) à la valeur réelle de la fonction, pour x = $\pi/8$ (calculez la différence absolue entre l'approximation et la valeur réelle pour chaque fonction f(x)).

Q2. L'équation de Colebrook est une équation implicite qui permet de calculer le coefficient de friction (f) dans une conduite (en hydraulique), afin de calculer la perte de charge (diminution de la pression par unité de longueur de conduite). Plus le facteur de friction sera élevé, plus la perte de charge par unité de longueur de conduite sera élevée.

$$\frac{1}{\sqrt{f}} = -2log_{10}\left[\frac{\varepsilon/d}{3.7} + \frac{2.51}{(Re)\sqrt{f}}\right]$$

Où f est le coefficient de friction, (ε/d) est la rugosité relative de la conduite (qui dépend par exemple du matériau de la conduite) et (Re) est un nombre adimensionnel (le nombre de Reynolds).

Vous considérez développer une fonction dans Python utilisant la méthode de Newton- Raphson pour pouvoir calculer le coefficient f en fonction des paramètres d'entrée (les paramètres (ε/d) et (Re)). Pour se faire vous considérez utiliser la fonction suivante et sa dérivée (pour la méthode de Newton-Raphson):

$$f(f) = \frac{1}{\sqrt{f}} + 0.8696ln\left[\frac{\varepsilon/d}{3.7} + \frac{2.51}{(Re)\sqrt{f}}\right] = 0$$

$$f'(f) = -\frac{f^{-3/2}}{2}\left[1 + \left(\frac{2.1826}{(Re)}\right)\left(\left[\frac{\varepsilon/d}{3.7} + \frac{2.51}{(Re)\sqrt{f}}\right]^{-1}\right)\right]$$

À noter que dans la première équation, le logarithme en base de 10 a été transformé en logarithme naturel *(ln)*.

Q2a) Développez la fonction fColebrook(epsilon_sur_d,Re,f0,erreur,nmax), qui permettra de trouver la valeur du facteur de friction (f) (sortie de la fonction). Utilisez la méthode de Newton-Raphson à l'intérieur de la fonction.

Testez la fonction en trouvant la valeur du facteur de friction *f* pour (ε/d) = 4x10^{-4}, (Re) = 1x10^{7}, f0=1x10^{-4}, erreur=1x10^{-6}, nmax=100). Vous devriez obtenir une valeur de f autour de 0.0159.

Q2b) Utilisez la fonction développée (fColebrook) à l'intérieur d'une boucle itérative « for » afin de calculer le facteur de friction pour différentes valeurs de (Re). Vous allez donc « appeler » la fonction fColebrook à l'intérieur de la boucle « for ».

Valeurs à utiliser pour (Re) : [1x10^{4}, 5x10^{4}, 1x10^{5}, 5x10^{5}, 1x10^{6}, 5x10^{6}, 1x10^{7}, 5 x10^{7}, 1x10^{8}]

Portez en graphique les valeurs du facteur de friction f en fonction de (Re), et ce pour trois différents matériaux (valeurs de (ε/d) différentes):

- Plastique : (ε/d)=6x10^{-5}
- Acier commercial : (ε/d)=1.8x10^{-4}
- Fonte : (ε/d)=1x10^{-3}

Utilisez une échelle logarithmique pour l'axe des x (plt.xscale('log')). Vous devriez obtenir trois (3) courbes : une pour le plastique, une pour l'acier commercial et une pour la fonte. Utilisez une légende pour identifier les courbes.

Note: Pour la fonction fColebrook, utilisez toujours f0=1x10^{-4}, erreur=1x10^{-6} et nmax=100.

Q3. Données d'une érablière

Le fichier suivant (DonneesErabliere.csv) contient des données liées à l'opération d'une érablière (données simulées).

Q3a) Chargez les données dans Python (ex : en utilisant pd.read_csv) et créez un tableau avec le module pandas (DataFrame). Utilisez la colonne « Date » comme index. Enlevez les colonnes suivantes : Statut données et ID usager.

Q3b) Écrivez un script Python qui permet de porter en graphique la quantité de sirop obtenue en litres (variable y) versus l'eau d'érable recueillie en litres (variable x). Utilisez plt.scatter(x,y) du module matplotlib.pyplot (où x et y sont des séries).

Q3c) Écrivez un script Python qui permet de calculer le % de sucre moyen dans l'eau d'érable pour chacune des années.

Q3d) Écrivez un script Python qui permet de calculer le volume total (cumulatif) d'eau d'érable recueillie et de sirop d'érable produit pour chacune des années.

Q4. Optimisation d'une fonction

Optimisez la fonction f(x) sur l'intervalle [0,10] : trouvez le maximum (ou maxima) et le minimum (ou minima) de la fonction. Portez en graphique le résultat avec la discrétisation utilisée (pour N=50 et 200 points).

$$f(x) = 2 + 3(x^3 - 2x^2)e^{\left(-\frac{x}{2}\right)} + \sin(2x)\, e^{(-x/4)}$$

Q5. Intégrale par la somme de Riemann

Développez une fonction dans Python nommée sommeRiemannG(fonc,N,x0,xn) qui permet d'intégrer une fonction en utilisant la méthode de Riemann par la gauche. La fonction doit également afficher le graphique montrant la discrétisation utilisée pour l'estimation de l'intégrale. Les paramètres d'entrée de la fonction : fonc (fonction à intégrer), N est le nombre d'incréments, x0 est la borne inférieure de l'intégrale, xn est la borne supérieure. La fonction sommeRiemannG(fonc,N,x0,xn) retourne l'estimé de l'intégrale.

Testez la fonction sommeRiemannG(fonc,N,x0,xn) sur la fonction suivante (en valeur absolue), avec N=200, x0=0 et xn=10 :

$$f(x) = \left|5\cos(3x)e^{(-x/2)}\right| = \cos(3x)\, e^{(-x/2)}$$

Q6 – Système de gestion des ventes

Une entreprise cherche à développer un système de gestion des ventes et des inventaires. Un gabarit avec certaines classes préliminaires (exemple hypothétique) est fourni (voir Q6Gabarit.py).

Q6a) Complétez dans le gabarit le script Python afin de définir les 3 classes considérées : Article, LigneArticle et Vente. Les attributs pour chaque classe sont :

- Article : identifiant, nom, categorie, unite, prixcoutant, prixvente
- LigneArticle : article (instance de la classe Article), quantite
- Vente : identifiant, date (chaîne de caractères), client, listeLignesArticles

(liste d'instances de la classe LignesArticles) Complétez la définition des méthodes constructeurs.

Q6b) Créez les instances (4) de la classe Article, à l'aide des données du tableau suivant :

Instance	Identifiant	Nom	Catégorie	Unité	Prix achat	Prix de vente
bois2x4x8	1	2x4x8	Bois construction	$/unité	5.48	7
bois2x6x8	2	2x6x8	Bois construction	$/unité	7.87	10
cimentGradeBase	3	Ciment Grade base	Ciment	$/m3	147.53	175
cimentGradeSup	4	Ciment Grade supérieur	Ciment	$/m3	284.76	310

Q6c) Créez les instances (2) de la classe Vente, à l'aide des données dans les tableaux suivants :

Vente 1	
Identifiant	20220001
Date	10 mars 2022
Client	VCH Entrepreneur expert
listeLignesArticles :	
article	**quantité**
bois2x4x8	530
bois2x6x8	320
cimentGradeBase	25
cimentGradeSup	-

Vente 2	
Identifiant	20220002
Date	14 mars 2022
Client	PLT Contsruction Pro
listeLignesArticles :	
article	**quantité**
bois2x4x8	430
bois2x6x8	250
cimentGradeBase	-
cimentGradeSup	50

Q6d) Développez la méthode *profits()* dans la classe Vente afin de pouvoir calculer le profit associé à une vente. Le profit peut être calculé avec la démarche suivante, pour « *n* » LigneArticles dans une vente:

$$profits = \sum_{i=1}^{n\ LigneArticle} \big(prix\ vente(article) - prix\ achat(article)\big) * quantité$$

La méthode profits() devrait imprimer dans la console l'identifiant de la vente et le montant des profits. Par exemple :

Le profit de la vente *X* est de *Y*$

Calculez le profit associé aux ventes 20220001 et 20220002 avec la méthode *profits()*

(les ventes #1 et #2 des tableaux fournis).