

TP Numpy

Exercice 1 : Création et manipulation de tableaux NumPy

1. Créez un tableau NumPy de 10 éléments allant de 0 à 9.
2. Transformez ce tableau en une matrice 2x5.
3. Inversez l'ordre des colonnes de la matrice.
4. Calculez la somme de chaque colonne.
5. Sélectionnez tous les éléments impairs du tableau d'origine.

Exercice 2 : Opérations mathématiques sur des tableaux

1. Créez un tableau aléatoire de taille 3x3 avec des nombres entiers entre 1 et 20.
2. Trouvez le minimum et le maximum du tableau.
3. Normalisez le tableau (transformez les valeurs pour qu'elles soient entre 0 et 1).
4. Calculez la moyenne, la variance et l'écart-type des éléments du tableau.
5. Remplacez les valeurs impaires du tableau par leur carré.
6. Créez un masque qui renvoie "True" pour les éléments de la matrice supérieur à 10.

Exercice 3 : Recherche du plus proche voisin

Écrire une fonction `find_nearest(arr, x)` qui prend en entrée un tableau NumPy `arr` et une valeur `x`, puis renvoie l'élément de `arr` qui est le plus proche de `x`.

Indice : Utilisez l'algèbre vectorielle pour comparer les écarts entre les valeurs.

Exercice 4 : Calculer la matrice de corrélation sans `np.corrcoef`

Écrire une fonction `correlation_matrix(data)` qui calcule la matrice de corrélation pour un ensemble de données sans utiliser `np.corrcoef`.

Indice : Utilisez la formule de la corrélation de Pearson entre chaque paire de colonnes.

La formule est la suivante :

$$r_{XY} = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum (X_i - \bar{X})^2} \cdot \sqrt{\sum (Y_i - \bar{Y})^2}}$$

Exercice 5:

1. Calculer $\sum_{i=0}^n i$ sachant que $n = 1\,000\,000$ sans utiliser numpy.

Chronométrer le temps nécessaire pour le calcul précédent, par exemple en utilisant `time.perf_counter()`.

NB : Le module **time** : Permet de mesurer le temps écoulé et le temps CPU (le temps passé par un processus en mémoire centrale) :

time() : mesure le temps ordinaire, celui de l'horloge.

perf_counter() : permet de mesurer la durée d'exécution d'une portion de code de manière précise.

2. Utiliser un tableau numpy et la méthode `sum` pour calculer à nouveau la somme proposée.
3. Comparer le temps de calcul avec la méthode précédente.