

Info. - TD : Feuille 4

Exercice 1 – Bases numpy et *broadcasting*

Aide: `np.random.randint(a, b, sha)` crée un tableau (de taille `sha`, une shape) de nombres entiers tirés au hasard entre `a` (inclus) et `b` (exclus).

```
1 import numpy as np
2 def magic(arr):
3     return arr * 100 + np.random.randint(0, 10, (5, 8))
4
5 e = np.zeros((5, 8))
6 f = magic(e)
7 g1 = magic(f)
8 g2 = f + np.arange(0, 800, 100).reshape( (1, -1) )
9 g3 = f + 100*np.arange(8).reshape( (1, -1) )
10 h = f + 100*np.arange(5).reshape( (-1, 1) )
11 i = np.mean(f, axis=1)
12 j = np.mean(f, axis=0, keepdims=True)
```

Note : en cas de valeurs aléatoires, donner un contenu possible (pas trop particulier, qui montre que vous avez compris comment sont choisies les valeurs) puis utiliser ces valeurs par la suite.

Q1) Que contient la variable `e` ?

Q2) Que contient alors `f` ?

Q3) Que contient alors `g1` ?

Q4) Que contient alors `g2` ?

Q5) Que contient alors `g3` ?

Q6) Que contient alors `h` ?

Q7) Que contient alors `i` ?

Q8) Que contient alors `j` ?

Q9) On veut « normaliser » `f` par ligne, c'est à dire que l'on veut que les valeurs de chaque ligne somment à 1. En pratique, nous voulons calculer `k` en divisant chaque valeur de `f` par la somme des valeurs de sa ligne. Quelle instruction permet de calculer `k` ?

Q10) Comment calculer un tableau `l` pour qu'il contiennent `f`, « normalisé » par colonne ?

Q11) On veut maintenant faire que toutes les valeurs soit comprises entre 0 et 1, en travaillant à nouveau par ligne. Pour cela, à chaque valeur on va soustraire le minimum de la ligne et diviser par la différence entre le maximum et le minimum (selon la formule mathématique $\frac{v-min}{max-min}$). Comment calculer dans une variable `m`, cette opération appliquée à `f` ?

Q12) Écrire trois fonctions `norm1_ligne`, `norm1_colonne` et `normmax_ligne` qui reprennent les 3 questions précédentes. Chaque fonction doit recevoir un tableau en paramètre et renvoyer le nouveau tableau.

Exercice 2 – Indices et Tranches

On considère le tableau `f` de l'exercice précédent, mais vous devez donner des réponses les plus générales possibles. Par exemple, on évitera de supposer que `f` a 5 lignes et 8 colonnes.

Q13) Comment extraire les valeurs d'indices 4 à 7 (inclus) de la seconde ligne du tableau ?

Q14) Comment remplacer toutes les valeurs d'indices 4 à 7 (inclus) de la seconde ligne du tableau par 111 ?

Q15) Comment remplacer toutes les valeurs d'indices 4 à 7 (inclus) de la seconde ligne du tableau par celles de la troisième ligne ?

Q16) Comment extraire les 4 dernières valeurs de la seconde ligne du tableau ?

Q17) Comment extraire une colonne sur deux ?

Q18) Comment extraire une valeur sur deux de la colonne d'indice 3 ?

Exercice 3 – Tableaux de booléens

En Python, les deux constantes booléennes `True` (vrai) et `False` (faux), sont interprétées comme des entiers. En particulier `True` vaut `1` alors que `False` vaut `0`. Ainsi, bien que l'on évitera de l'utiliser, on peut écrire `True + 10` et cette expression aura la valeur `11`.

Q19) Qu'affiche la ligne `print( np.sum(np.array([10, 20, 30])) )` ?

Q20) Qu'affiche la ligne `print( np.sum(np.array([True, False, True])) )` ?

En numpy, les opérateurs de comparaison (`==`, `!=` (différent), `>`, `<`, `>=` (supérieur ou égal), ...) se comportent élément par élément, exactement comme les autres opérateurs. Ces opérateurs calculent alors un tableau de booléen.

On considère le tableau `f` de l'exercice précédent, et on exécute les instructions suivantes :

```
1 b1 = f == 0
2 b2 = f <= 2
3 b3 = f > -10
```

Q21) Que contient alors `b1` ?

Q22) Que contient alors `b2` ?

Q23) Que contient alors `b3` ?

Q24) Qu'affiche alors la ligne `print( np.sum(b1) )` ?

Q25) Qu'affiche alors la ligne `print( np.sum(b2) )` ?

Q26) Qu'affiche alors la ligne `print( np.sum(b3) )` ?

La fonction `np.count_nonzero` permet de compter le nombre de valeurs non-nulles dans un tableau numpy.

Q27) Qu'affiche alors la ligne `print( np.count_nonzero(f) )` ?

Q28) Qu'affiche alors la ligne `print( np.count_nonzero(f != 0) )` ?

Q29) Qu'affiche alors la ligne `print( np.count_nonzero(f == 0) )` ?

Q30) Qu'affiche alors la ligne `print( np.count_nonzero(b1) )` ?

Q31) Qu'affiche alors la ligne `print( np.count_nonzero(b2) )` ?

Q32) Qu'affiche alors la ligne `print( np.count_nonzero(b3) )` ?

Q33) Qu'affiche alors la ligne `print( np.sum(f == 0) )` ?

Exercice 4 – Accès par tableaux de booléens (masques)

Q34) Que contient `a = np.linspace(0, 1, 6).reshape((2, 3))` ?

Numpy permet d'accéder (en lecture et en écriture) à une sous partie d'un tableau en passant en tant qu'indice un tableau de booléens ayant la même taille (shape). Par exemple, si `b = np.array( [ [True, True, False], [True, False, True] ] )`, alors `a[b]` vaut `[ 0. , 0.2, 0.6, 1. ]`. Notons que le résultat est un tableau à une dimension. De la même façon, on pourra changer les valeurs (0, 0.2, 0.6 et 1) de `a` en écrivant `a[b] = 789`.

On considère le tableau `f` de l'exercice précédent.

Q35) Comment calculer la somme des valeurs de `f` qui sont inférieures (strictement) à 3 ?

Q36) Comment (dans `f`) remplacer les 0 par 124 ?

Q37) Comment (dans `f`) remplacer les multiples de 3 par leurs carrés (à chacun) ?

Q38) Comment (dans `f`) remplacer les multiples de 2 par leur moyenne (commune) ?