

TP7: matrices, images matricielles

Le document de référence doit être apporté en TP.

Créer un dossier "TP7" dans votre répertoire personnel. Pour l'exercice 1, dans Pyzo, enregistrer un fichier nommé "exo1.py" dans le dossier "TP7". Pour l'exercice 2, dans Pyzo, enregistrer un fichier nommé "exo2.py" dans le dossier "TP7"..... Pour chaque exercice, le fichier python sera régulièrement sauvegardé et exécuté intégralement avec la commande "Run file as script" du menu Run (Ctrl+Shift+E). Les instructions d'affichage seront saisies sur le fichier python. Au fur et à mesure de l'avancée de l'exercice, les instructions devenant inutiles et gênantes pourront être désactivées en utilisant le caractère #.

Matrices :

On appelle matrice de taille (n, p) un tableau de nombres rectangulaire à n lignes (numérotées de 0 à $n - 1$) et p colonnes (numérotées de 0 à $p - 1$). Pour tout $(i, j) \in [[0, n - 1]] \times [[0, p - 1]]$, on appelle coefficient d'indice (i, j) de la matrice le nombre situé sur la ligne i et la colonne j .

Exemple : $M = \begin{pmatrix} 1 & 5 & 2 \\ 3 & 2 & 7 \\ 6 & 8 & 5 \\ 4 & 2 & 1 \end{pmatrix}$ matrice de taille $(4, 3)$. Le coefficient d'indice $(2, 1)$ de M est 8.

Matrices en python avec le module numpy renommé np :

$M = np.array([[a_{0,0}, a_{0,1}, \dots, a_{0,p-1}], [a_{1,0}, a_{1,1}, \dots, a_{1,p-1}], \dots, [a_{n-1,0}, a_{n-1,1}, \dots, a_{n-1,p-1}]])$

permet d'affecter à M la matrice $\begin{pmatrix} a_{0,0} & a_{0,1} & \dots & a_{0,p-1} \\ a_{1,0} & a_{1,1} & \dots & a_{1,p-1} \\ \vdots & \vdots & & \vdots \\ a_{n-1,0} & a_{n-1,1} & \dots & a_{n-1,p-1} \end{pmatrix}$

$(n, p) = np.shape(M)$ permet d'affecter au couple (n, p) la taille de la matrice.

$M[i, j]$ permet d'obtenir le coefficient d'indice (i, j) de la matrice.

$M[i, j] = x$ permet de réaffecter au coefficient d'indice (i, j) de la matrice la valeur x

Exercice 1. Recopier et faire exécuter la suite d'instructions suivante et examiner les affichages afin de s'assurer d'avoir compris le fonctionnement.

```
import numpy as np                                print(n, p)
M = np.array([[1, 5, 4, 2], [3, 2, 5, 7], [1, 3, 4, 6]]) print(M[1, 2])
print(M)                                           M[2, 1] = 7
(n, p) = np.shape(M)                             print(M)
```

Exercice 2.

1. Affecter à la variable M la matrice $\begin{pmatrix} 2 & 7 \\ 1 & 4 \\ 5 & 9 \end{pmatrix}$. Faire afficher M .

2. Ecrire une fonction "sommeli" ayant pour argument M (matrice) et i (indice de ligne) et retournant la somme des coefficients de M situés sur la ligne i . On contrôlera la précondition $0 \leq i \leq n - 1$ où n est le nombre de lignes. Faire afficher $sommeli(M, 0)$, $sommeli(M, 1)$, $sommeli(M, 2)$.

Exercice 3. Coller la fonction "licon" après l'avoir copiée dans le fichier "listes.py".

Recopier et faire exécuter la suite d'instructions suivante et examiner les affichages afin de s'assurer d'avoir compris le fonctionnement.

```
import numpy as np
(n, p) = (3, 5)
print(licon(p, 0))

print(licon(n, licon(p, 0)))
M = np.array(licon(n, licon(p, 0)))
print(M)
```

On appelle matrice identité de taille 1, matrice identité de taille 2, matrice identité de taille 3, matrice identité de taille 4,... les matrices (1) , $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$, $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$,

Exercice 4. Ecrire une fonction nommée "matid" ayant pour argument n (entier naturel non nul) et retournant la matrice identité de taille n . Faire afficher `matid(5)`.

Exercice 5 (Traitement d'images matricielles en niveau de gris).

On appelle image matricielle de taille (n, p) une image rectangulaire quadrillée en n lignes (numérotées de 0 à $n - 1$) et p colonnes (numérotées de 0 à $p - 1$) de sorte que pour tout $(i, j) \in [[0, n - 1]] \times [[0, p - 1]]$, l'intersection de la ligne i et la colonne j est un petit carré unicolore appelé le pixel d'indices (i, j) . L'image est dite en niveau de gris ssi les couleurs des pixels appartiennent à un nuancier de gris, chaque nuance de gris étant caractérisée par un coefficient de luminance, qui est un nombre compris entre 0 à 255 (les valeurs extrêmes 0 et 255 correspondent au noir et au blanc). Dans ce cas, l'image est représentée par la matrice de (n, p) telle que pour tout $(i, j) \in [[0, n - 1]] \times [[0, p - 1]]$, le coefficient d'indices (i, j) de la matrice est égal au coefficient de luminance du pixel d'indices (i, j) .

Dans la suite de cet exercice, le terme image désigne une image matricielle en niveau de gris.

Question 1. Recopier le fichier 'fleur.png' dans le dossier "TP7". Effectuer les instructions :

```
from PIL import Image
I=Image.open('fleur.png')
I.show()

import numpy as np
M=np.array(I)
print(M)
```

On appelle inverse d'un coefficient de luminance x le coefficient de luminance $255 - x$. On appelle négatif d'une image l'image de même taille obtenue en inversant les coefficients de luminance de tous les pixels.

Question 2. Ecrire une fonction nommée "negatif" ayant pour argument M (matrice représentant une image) et retournant N la matrice représentant le négatif de l'image. Affecter à la variable N la valeur `negatif(M)`. Faire afficher N . Effectuer les instructions `J = Image.fromarray(N)` et `J.show()`.

Question 3. Ecrire une fonction nommée "retournerho" ayant pour argument M (matrice représentant une image) et retournant N la matrice représentant l'image obtenue par retournement horizontal. Affecter à la variable N la valeur `retournerho(M)`. Faire afficher N . Effectuer les instructions `J = Image.fromarray(N)` et `J.show()`.

Question 4. Ecrire une fonction nommée "pivoter90g" ayant pour argument M (matrice représentant une image) et retournant N la matrice représentant l'image obtenue par rotation de 90 degrés vers la gauche. Affecter à la variable N la valeur `pivoter90g(M)`. Faire afficher N . Effectuer les instructions `J = Image.fromarray(N)` et `J.show()`.

(Indication : notons (n, p) la taille de M . A quelle partie de M correspond la première colonne de N ? la dernière colonne de N ? une colonne d'indice j de N ? la première ligne de N ? la dernière ligne de N ? une ligne d'indice i de N ?)

On appelle bord d'une image l'ensemble des pixels situés sur la première ligne, la première colonne, la dernière ligne ou la dernière colonne.

Considérons $C = \begin{pmatrix} C[0,0] & C[0,1] & C[0,2] \\ C[1,0] & C[1,1] & C[1,2] \\ C[2,0] & C[2,1] & C[2,2] \end{pmatrix}$ une matrice carrée de taille 3.

On appelle filtre de matrice de convolution C l'opérateur qui à une image (appelée l'image d'entrée) associe une autre image (appelée l'image de sortie) de la manière suivante :

-L'image de sortie a même taille que l'image d'entrée.

-Les pixels du bord de l'image de sortie sont identiques aux pixels du bord de l'image d'entrée.

-Considérons un pixel de l'image d'entrée n'appartenant pas au bord. Notons (i, j) le couple d'indices de ce pixel. Notons M la matrice représentant l'image d'entrée. Dans l'image d'entrée, ce pixel et ses voisins ont pour coefficients de luminance :

$$\begin{array}{ccc} M[i-1, j-1] & M[i-1, j] & M[i-1, j+1] \\ M[i, j-1] & M[i, j] & M[i, j+1] \\ M[i+1, j-1] & M[i+1, j] & M[i+1, j+1] \end{array}$$

Considérons la matrice symétrique de C par rapport à son coefficient central :

$$\begin{pmatrix} C[2,2] & C[2,1] & C[2,0] \\ C[1,2] & C[1,1] & C[1,0] \\ C[0,2] & C[0,1] & C[0,0] \end{pmatrix}$$

Le pixel d'indices (i, j) de l'image de sortie a pour coefficient de luminance :

$$\begin{array}{l} C[2,2]M[i-1, j-1] + C[2,1]M[i-1, j] + C[2,0]M[i-1, j+1] \\ + C[1,2]M[i, j-1] + C[1,1]M[i, j] + C[1,0]M[i, j+1] \\ + C[0,2]M[i+1, j-1] + C[0,1]M[i+1, j] + C[0,0]M[i+1, j+1] \end{array}$$

(Le coefficient de luminance du pixel d'indice (i, j) de l'image de sortie est une combinaison linéaire des coefficients de luminances du pixel d'indice (i, j) de l'image d'entrée et de ses voisins.)

Question 5. Ecrire une fonction nommée "filtre" ayant pour arguments C (matrice de convolution d'un filtre) et M (matrice représentant l'image d'entrée) et retournant N la matrice représentant l'image de sortie.

Question 6. Affecter à la variable C la matrice $\begin{pmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{pmatrix}$ (matrice de convolution d'un filtre passe-bas, floutant). Affecter à la variable N la valeur `filtre(C, M)`. Faire afficher N . Effectuer les instructions `J = Image.fromarray(N)` et `J.show()`.

Question 7. Affecter à la variable C la matrice $\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$ (matrice de convolution d'un filtre passe-haut, détectant les contours). Affecter à la variable N la valeur `filtre(C, M)`. Faire afficher N . Effectuer les instructions `J = Image.fromarray(N)` et `J.show()`.