Calcul Scientifique

Cours 2: Manipulation de tableau numpy

Alexis Lechervy

Sommaire

reshape

reshape

La méthode reshape permet de changer la forme d'un tableau numpy. Il permet par exemple de passer d'un tableau (6,4) à un tableau (8,2).

Fonctionnement de reshape

```
M=np.array([[1,2,3,4],[5,6,7,8],[9,10,11,12]])
M2= M.reshape( (6,2) )
print(M2)
-> [[1 2]
    [3 4]
    [5 6]
    [7 8]
    [9 10]
    [11 12]]
```

Remarques

- Le nombre d'élément d'un tableau est conservé avec reshape. Il n'est pas possible de transformer un tableau (5,2) en tableau (3,1).
- La valeur -1 permet d'indiquer qu'il faut choisir un nombre de dimensions tel que la fonction reshape puisse fonctionner. Par exemple (6,4) -> (8,-1) et équivalent à (8,2).

np.transpose

Principe

La fonction permet de réorganiser les axes d'un tableau numpy.

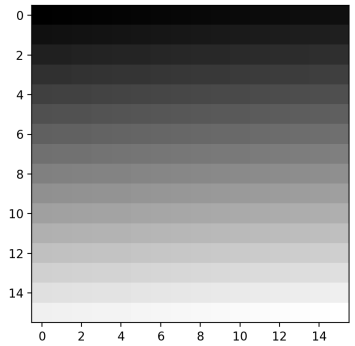
Syntaxe de np.transpose

```
M = np.ones((12,4,2))
print(M.shape)
-> (12,4,2)
M2 = np.transpose(M,(2,1,0))
print(M2.shape)
-> (2,4,12)
```

Exemple : création d'un dégradé rouge

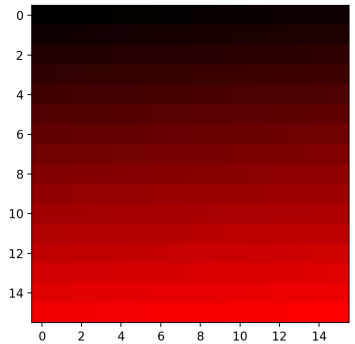
Création d'un dégradé noir et blanc sur les lignes

```
degNB = np.arange(256)
degNB = degNB.reshape((16,16))
```



Création d'un dégradé rouge sur les lignes

```
degRouge = np.zeros((16,16,3))
degRouge[ :, :,0] = degNB
```



Exemple : création d'un dégradé rouge

Transformation d'un dégradé sur les lignes en dégradé sur les colonnes

degRouge2 = np.transpose(im,(1,0,2))

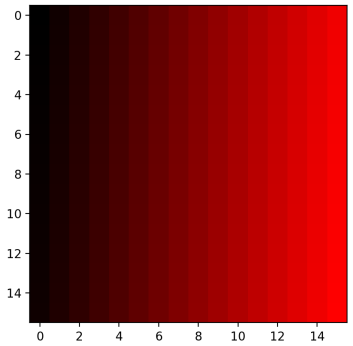
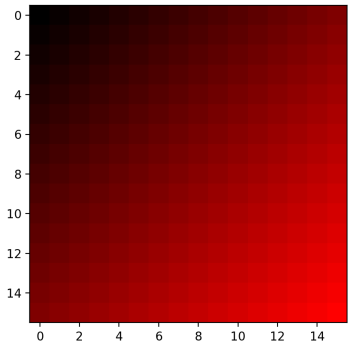


Image finale : dégradé rouge sur les lignes et les colonnes

im = (degRouge1+degRouge2)//2



np.concatenate

Présentation

La fonction concatenate permet de concaténer plusieurs tableau numpy.

Fonctionnement de concatenate

np.concatenate((M,M2,M3) , axis=0)

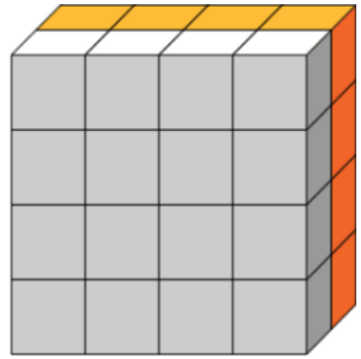
$$\text{->} \begin{bmatrix} M1 \\ M2 \\ M3 \end{bmatrix}$$

np.concatenate((M,M2,M3) , axis=1)

-> [M,M2,M3]

Empiler des matrices

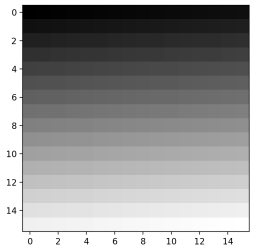
np.concatenate((M[:, :,np.newaxis],M2[:, :,np.newaxis]) , axis=2)



Exemple : Création d'un dégradé multicolore

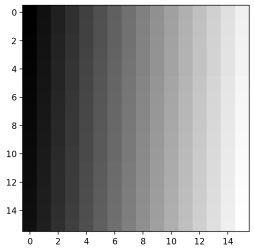
Création du canal rouge

```
imR = np.arange(256)
imR = imR.reshape((16,16))
```



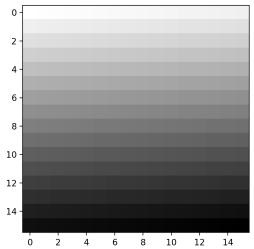
Création du canal vert

```
imG = np.arange(256)
imG = imG.reshape((16,16)).T
```



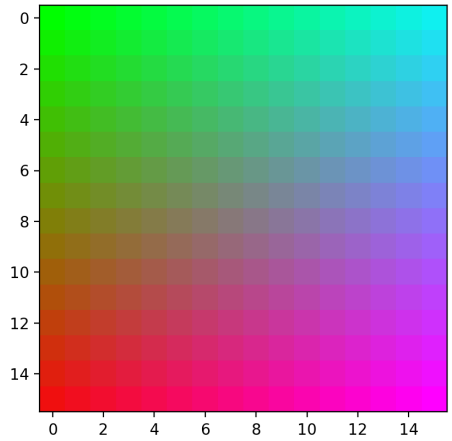
Création du canal bleu

```
imB = 255-np.arange(256)
imB = imB.reshape((16,16))
```



Assemblage des trois canaux

```
im = np.concatenate(
(imR[ :, :,np.newaxis],
 imB[ :, :,np.newaxis],
imG[ :, :,np.newaxis]),
axis=2)
```



Sommaire

Le brodcast de calcul

Principes

Lors d'un calcul entre deux tableaux numpy de dimension différentes, numpy va dans certaines conditions étendre le plus petit des tableaux par recopie pour obtenir deux tableaux de même dimension et effectuer le calcul.

Exemple simple : Opération entre une matrice et un scalaire

M+s est équivalent à M+s*np.ones(M.shape)

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + 5 \longrightarrow \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + \begin{bmatrix} 5 & 5 \\ 5 & 5 \end{bmatrix}$$

Règles générales

Lors d'une opération entre deux tableau, numpy commence par comparer les dimensions des deux tableaux dimensions par dimensions. L'opération sera possible si :

1. toute les dimensions sont identiques,
2. les dimensions sont identiques ou un des tableaux à une dimension à 1.

Dans le deuxième cas, le tableau le plus petit sera recopié suivant la dimension à 1 jusqu'à avoir la même taille que l'autre.

Opération entre une matrice (n,m) et un vecteur (m,)

Principe

Soit M un array de taille (n, m) et v un array de taille $(m,)$. L'opération numpy $M + v$ correspond à construire une matrice de taille (n, m) où le vecteur v est recopié m fois à la sommer avec M.

Exemple 1

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + \begin{bmatrix} 11 & 22 \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + \begin{bmatrix} 11 & 22 \\ 11 & 22 \end{bmatrix}$$

Exemple 2

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} + \begin{bmatrix} 11 & 22 \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} + \begin{bmatrix} 11 & 22 \\ 11 & 22 \\ 11 & 22 \end{bmatrix}$$

Exemple 3

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} + \begin{bmatrix} 11 & 22 & 33 \end{bmatrix} \longrightarrow \text{Erreur}$$

Opération entre une matrice (n,m) et un vecteur (1,m)

Principe

Cette opération est équivalente à la précédente.

Exemple 1

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + \begin{bmatrix} 11 & 22 \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + \begin{bmatrix} 11 & 22 \\ 11 & 22 \end{bmatrix}$$

Exemple 2

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} + \begin{bmatrix} 11 & 22 \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} + \begin{bmatrix} 11 & 22 \\ 11 & 22 \\ 11 & 22 \end{bmatrix}$$

Opération entre une matrice (n,m) et un vecteur (n,1)

Principe

Soit M un array de taille (n, m) et v un array de taille $(n, 1)$. L'opération numpy $M + v$ correspond à construire une matrice de taille (n, m) où le vecteur v est recopié m fois à la sommer avec M. Le vecteur est recopié sur les colonne non sur les lignes.

Exemple 1

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + \begin{bmatrix} 5 \\ 6 \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + \begin{bmatrix} 5 & 5 \\ 6 & 6 \end{bmatrix}$$

Exemple 2

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} + \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} + \begin{bmatrix} 7 & 7 \\ 8 & 8 \\ 9 & 9 \end{bmatrix}$$

Exemple : Pourcentage de glucide/lipide/protéine

Objectif

Calculer le pourcentage de glucide, Lipide et Protéine d'aliment connaissant la quantité pour 100g de glucide, Lipide et Protéine de ces aliments.

	Pomme	Oeuf	Lait	Boeuf
Glucide	14	0.7	5	0
Lipide	0.2	11	1	15
Protéine	0.3	13	3.4	26

Calcul de la quantité de glucide+lipide+protéine pour chaque aliment

s = np.sum(M,axis=0) –> [14.5 24.7 9.4 51.]

Pourcentage de glucide+lipide+protéine pour chaque aliment

print(M.shape) –> (3,4)
print(s.shape) –> (4,)
100*M/s

	Pomme	Oeuf	Lait	Boeuf
Glucide	96.55	2.83	53.19	0.
Lipide	1.38	44.53	10.64	29.41
Protéine	2.07	52.63	36.17	70.59

Exemple : Ajouter un vignettage à une photo

Objectif

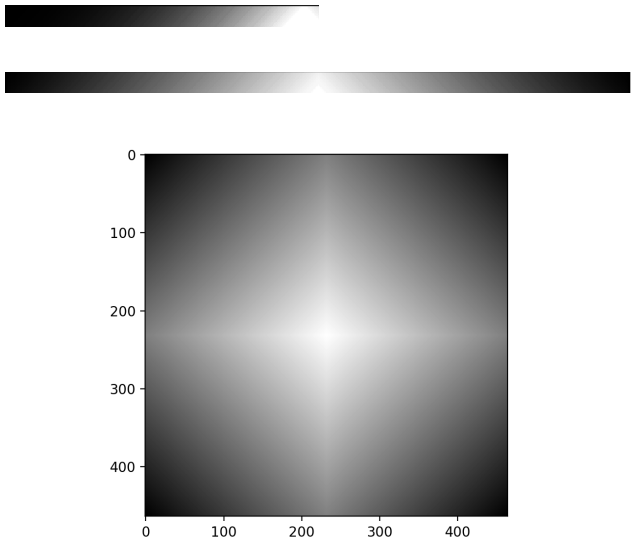
Ajouter du vignettage à une image. Cela consiste à assombrie les bords d'une photo.

Construire un filtre

v = np.arange(im.shape[0]//2)/(im.shape[0]//2)

line = np.concatenate((v,1-v))

masque = (line[:,np.newaxis]+line[np.newaxis, :])/2



Exemple : Ajouter un vignettage à une photo

On applique le masque

```
imFinal = masque[ :, :,np.newaxis]*im
imFinal = imFinal.astype('ubyte')
```

Avant :

Après