# Opérations matricielles et recherche d'information

#### 1 décembre 2020

—Lexicologie, terminologie, dictionnairique—

## Créer et utiliser des matrices avec numpy

Un vecteur est une matrice de dimensions  $1 \times n$ , où une liste numpy : a = np.array([1,2,5])

Le *i*=ième élément du vecteur : a[i]

Accéder aux éléments avec indice  $\geq i$  du vecteur : a[i :]

Accéder aux éléments avec indice < i du vecteur : a[:i]

Une matrice (dimensions  $n \times m$ ) est une liste de listes : b = np.array([[3,2,9],[1,5,1],[0,3,4]])

Accéder à la ligne i de la matrice : b[i] ou b[i, :]

Accéder à la colonne j de la matrice : b[:, j]

L'élément i, j de la matrice est celle de l'i-ième ligne, j-ième colonne : b[i, j]

Créer une matrice de dimensions  $5 \times 6$  avec des valeurs zéro partout : c = np.zeros([5,6])

Transposée d'une matrice A :  $A_{ij}$  devient  $B_{ji}$ , soit on échange les lignes et les colonnes.

#### Multiplication de deux vecteurs :

avec le même nombre n de dimensions (produit scalaire, dot product, inner product):

$$u = x \times y = \sum_{i=1}^{n} x_i y_i = x_1 y_1 + x_2 y_2 \dots x_n y_n$$
 (1)

Par exemple:

a = np.array([1, 0, 3])b = np.array([0, 9, 9])

$$c = a \times b = 1 \times 0 + 0 \times 9 + 3 \times 9 = 27 \tag{2}$$

Multiplication vecteur-vecteur, vecteur-matrice, matrice-matrice dans numpy:

$$c = np.dot(a, b) (3)$$

ou

$$c = np.matmul(a, b) (4)$$

# **Multiplication matrice-vecteur:**

la multiplication d'une matrice avec un vecteur donne un vecteur, dont le i-ième élément est le produit scalaire de la ligne i de la matrice et du vecteur :

$$\begin{bmatrix} 1 & 3 \\ 4 & 0 \\ 2 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 5 \end{bmatrix} = \begin{bmatrix} 1x1 + 3x5 = 16 \\ 4x1 + 0x5 = 4 \\ 2x1 + 1x5 = 7 \end{bmatrix}$$

I. Multiplication de matrice avec un vecteur de colonne.

La matrice a les dimensions  $m \times n$ , et le vecteur  $n \times 1$  (ou  $1 \times n$ ), donc le nombre des colonnes de la matrice = la dimensionnalité du vecteur.

$$\begin{bmatrix} 1 & 3 \\ 4 & 0 \\ 2 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 5 \end{bmatrix} = \begin{bmatrix} 16 \\ 4 \\ 7 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 3 \\ 4 & 0 \\ 2 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 5 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 16 & 4 & 7 \end{bmatrix}$$

## Exercices I.

- 1. A la main : calculez le résultat.
  - vecteur vecteur :

— matrice - vecteur :

$$\begin{bmatrix} 0 & 1 & 0 & 1 \\ 4 & 3 & 2 & 1 \\ 5 & 6 & 9 & 9 \end{bmatrix} \quad X \quad \begin{bmatrix} 3 \\ 7 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} \\ \\ \end{bmatrix}$$

- 2. Créez une matrice A  $15 \times 10$  peuplée avec les nombres entiers naturels impairs en ordre croissant de gauche à droite et de haut en bas. (Tip : vous pouvez utiliser np.arange, puis np.reshape pour modifier les dimensions une fois que vous avez la suite des valeurs).
  - Sélectionnez colonnes avec indice 2-5, dans une matrice B.
  - Sélectionnez les lignes 0-4, dans une matrice C.
  - Créez un vecteur z de dimension 10, avec une valeur de 1 à l'indice 4, zéro ailleurs.
  - Calculez le produit scalaire de la première ligne de A avec z.
  - Calculez le produit de la matrice C avec z.

#### **Multiplication matrice-matrice:**

produits des matrice A de taille  $m \times n$  et matrice B de taille  $n \times p$  (nombre des colonnes dans A = nb des lignes dans B) :

$$C_{ij} = \sum_{k=1}^{n} A_{ik} B_{kj} \tag{5}$$

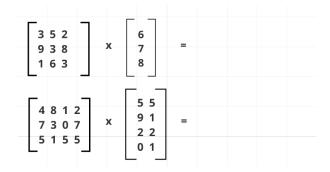
donc on multiplie les lignes de A avec les colonnes de B. Intuitivement : on prend les colonnes de la matrice B un par un et on fait le même calcul que dans I. multiplication de matrice avec un vecteur de colonne.

 $C_{ij}$  contient donc le produit des vecteurs de *l'i-ième ligne de A et de la j-ième colonne de B*.

$$\begin{bmatrix} 0 & 1 & 0 & 1 \\ 4 & 3 & 2 & 1 \\ 5 & 6 & 9 & 9 \end{bmatrix} \times \begin{bmatrix} 0 & 1 \\ 3 & 3 \\ 9 & 2 \\ 5 & 1 \end{bmatrix} = \begin{bmatrix} 8 & 4 \\ 32 & 18 \\ 144 & 50 \end{bmatrix}$$

#### **Exercices II.**

1. A la main : calculez le résultat.



- 2. https://github.com/fastai/numerical-linear-algebra/blob/master/
   README.md
  - Le premier exercice sous "Matrix and Tensor Products" (AIDS)
  - Le deuxième exercice sous "Matrix and Tensor Products" (shopping)