lab01(numpy)

October 25, 2023

NumPy : fundamental package for scientific computing Module "Machine Learning", MST IASD/S1 2023-2024 (M. AIT KBIR-FST de Tanger)

[1]:
```
import numpy as np
```

Créer une matrice 5x5 (éléments initialisés aléatoirement avec des valeurs dans [0 et 1]), voir aussi np.ones(...), np.eye(...), np.random.zeros(...), np.random.randn()

[2]:
```
m1=2*np.random.rand(5,5)-1
print(m1)
print(m1.dtype)
```

```
[[-0.74857486 -0.95097669  0.29558864  0.5024464  -0.8543243 ]
 [ 0.04257913 -0.9348289  -0.29743934  0.94321114  0.8896065 ]
 [ 0.81765201 -0.93016468 -0.33666796 -0.26333607 -0.44676822]
 [-0.61446411  0.30427583 -0.05692167 -0.05057912  0.36142993]
 [ 0.7455125  -0.79840106  0.34194193  0.88009032  0.68805841]]
float64
```

Membres

[3]:
```
print(m1.shape) # tuple of ints
print(m1.ndim)  # Dimension
print(m1.itemsize) # Taille de chaque élément
print(m1.max())    # Calcul du maximum
print(m1.max(axis=1))    # Calcul du maximum selon le premier axe
print(m1.std())    # l'écart type de toutes les valeurs
print(m1.std(axis=1))
```

```
(5, 5)
2
8
0.943211137607302
[0.5024464  0.94321114 0.81765201 0.36142993 0.88009032]
0.6403608467289927
[0.61932554 0.71589278 0.57398012 0.34812263 0.61127749]
```

Afficher les trois premiers éléments de la dernières ligne, puis tous les éléments de la 3 ème colonne

[4]:
```
print(m1[-1,0:3]) #-1 pour désigner la dernière ligne
print(m1[:,2])
```

```
[ 0.7455125  -0.79840106  0.34194193]
[ 0.29558864 -0.29743934 -0.33666796 -0.05692167  0.34194193]
```

Créer un vecteur de valeurs entières entre 1 et 25

[5]:
```
m2=np.arange(1,26)
print(m2)
```

```
[ 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
 25]
```

Redimensionner en une matrice 5x5

[6]:
```
m3=m2.reshape(5,5)
print(m3)
```

```
[[ 1  2  3  4  5]
 [ 6  7  8  9 10]
 [11 12 13 14 15]
 [16 17 18 19 20]
 [21 22 23 24 25]]
```

30 éléments dans [0,10] linéarement espacés

[7]:
```
m4=np.linspace(0,10,30)
print(m4)
print(m4.shape)
```

```
[ 0.          0.34482759  0.68965517  1.03448276  1.37931034  1.72413793
  2.06896552  2.4137931   2.75862069  3.10344828  3.44827586  3.79310345
  4.13793103  4.48275862  4.82758621  5.17241379  5.51724138  5.86206897
  6.20689655  6.55172414  6.89655172  7.24137931  7.5862069   7.93103448
  8.27586207  8.62068966  8.96551724  9.31034483  9.65517241 10.        ]
(30,)
```

Operations

[8]:
```
A = np.array([[1, 2],[3, 1]])
B = np.array([[1, 0],[4, 1]])
print(A*B)
print(A@B)
print(A.dot(B)) # Même chose
```

```
[[ 1  0]
 [12  1]]
[[9 2]
 [7 1]]
[[9 2]
 [7 1]]
```

[9]:
```
A*=3
print(A)
print(np.sin(B))
```

```
[[3 6]
 [9 3]]
[[ 0.84147098  0.        ]
 [-0.7568025   0.84147098]]
```

Voir : https://numpy.org/doc/stable/user/basics.creation.html

[ ]:
```

```