Python科学计算库Numpy之

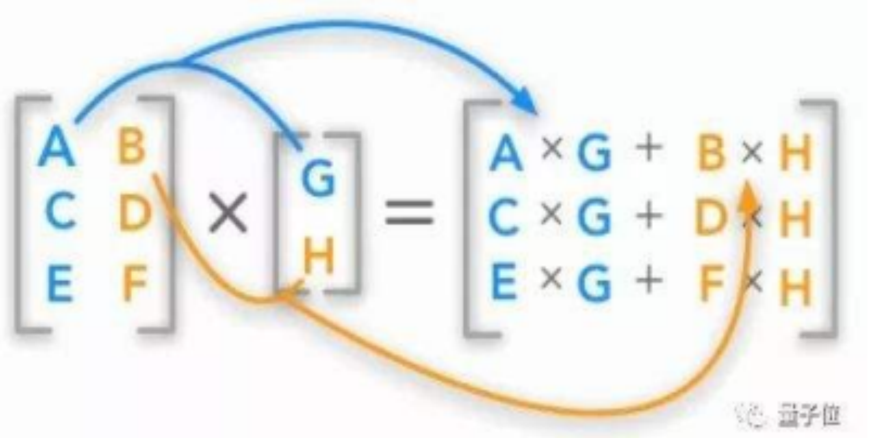
09-代数运算

海豚数据科学实验室

知识结构图

海豚数据科学实验室

- 代数运算
  - 方程组求解
    - Ax = b
    - numpy.linalg.solve(A, b)
  - 矩阵求逆
    - 单位矩阵
    - 逆矩阵
    - np.linalg.inv()：矩阵求逆
  - 矩阵行列式
    - numpy.linalg模块
    - numpy.linalg.det() 函数
  - 问题导入
    - 数组乘法
    - 矩阵乘法

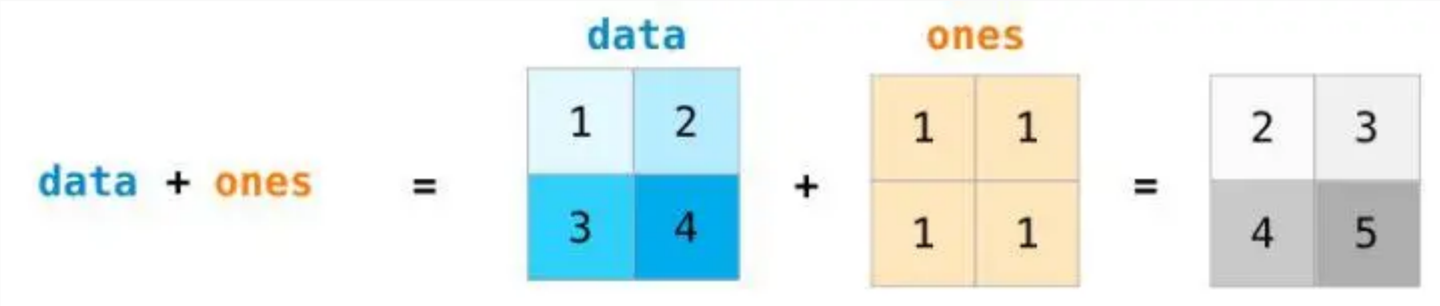


  - 点积运算
    - np.dot(A, B)
    - 注意：A的维度 m x k
      B的形状 k x n
      计算结果：m x n

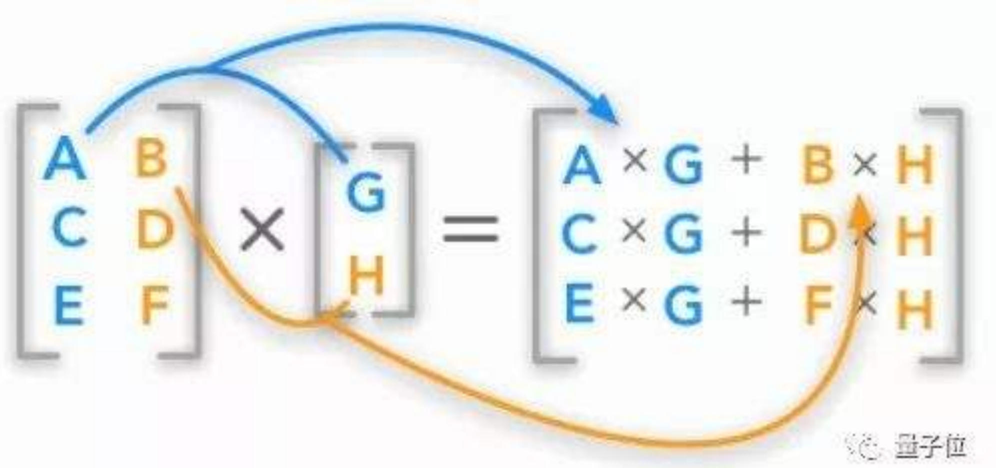
数组乘法与矩阵乘法

海豚数据科学实验室

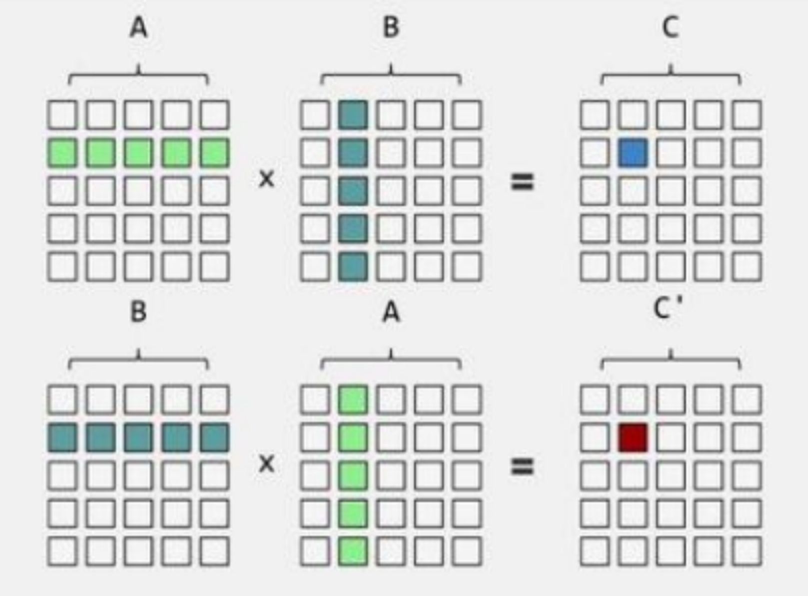
如果两个矩阵的大小相同，我们可以使用算术运算符（+ – * /）来进行数组计算。NumPy将对应元素进行运算：

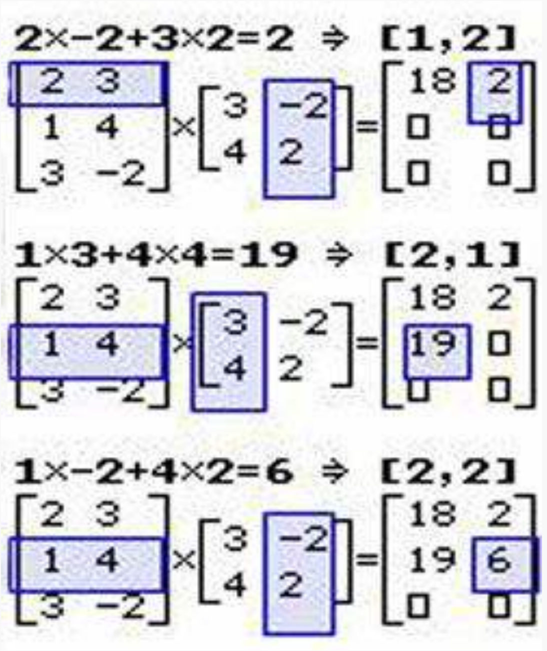


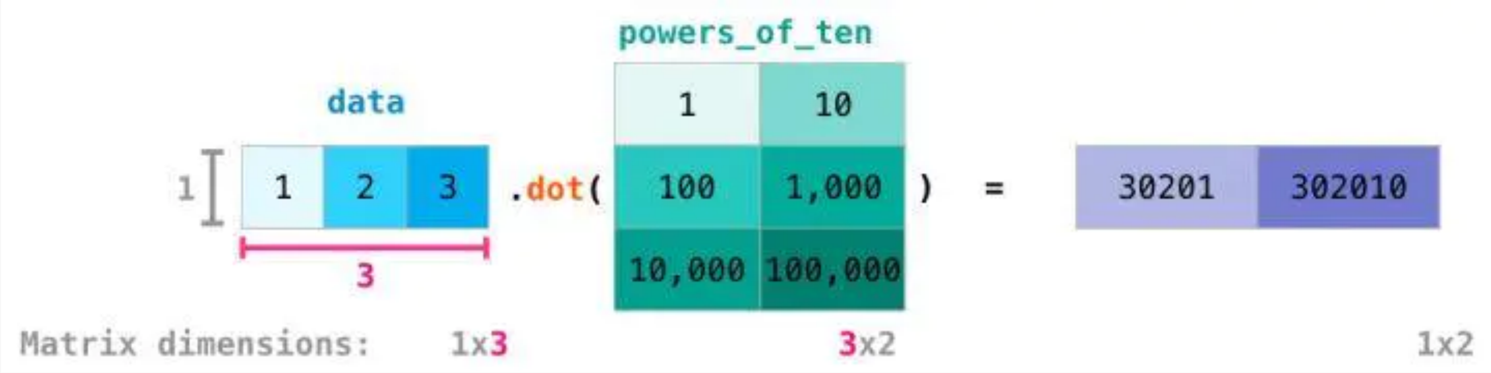
线性代数中矩阵乘法（点积）的定义： np.dot(A, B)



矩阵乘法







两个矩阵在它们彼此面对的
一侧必须具有相同的尺寸
（左图底部红色的数字）

点积运算


```python
import numpy as np
```

```python
a = np.array([[1,2,3,4],[5,6,7,8]])
a
```

```
array([[1, 2, 3, 4],
       [5, 6, 7, 8]])
```

```python
a.shape
```

```
(2, 4)
```

```python
b = np.array([[1,2,3],[4,5,6],[7,8,9],[10,11,12]])
b
```

```
array([[ 1,  2,  3],
       [ 4,  5,  6],
       [ 7,  8,  9],
       [10, 11, 12]])
```

```python
b.shape
```

```
(4, 3)
```

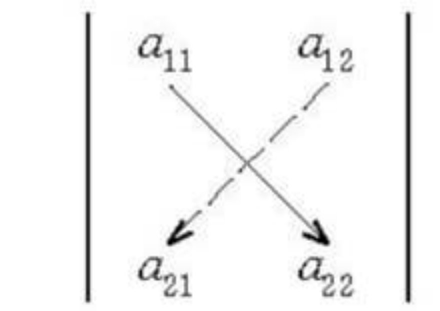
```python
c = np.dot(a, b)
c
```

```
array([[ 70,  80,  90],
       [158, 184, 210]])
```

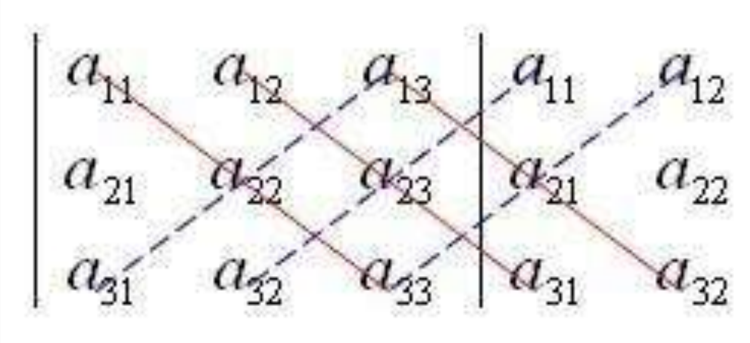
```python
c.shape
```

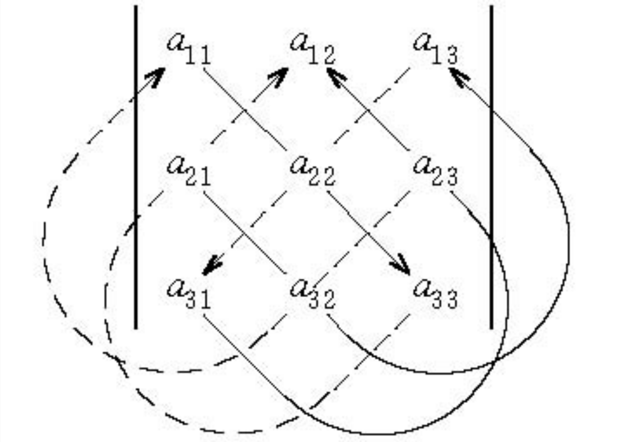
```
(2, 3)
```

矩阵行列式



$$=a_{11}a_{22}-a_{12}a_{21}$$





$$=a_{11}a_{22}a_{33}+a_{12}a_{23}a_{31}+a_{13}a_{21}a_{32}$$

$$-a_{13}a_{22}a_{31}-a_{12}a_{21}a_{33}-a_{11}a_{23}a_{32}$$

矩阵行列式

$$\begin{vmatrix} 1 & 0 & 1 \\ 2 & 1 & 1 \\ 1 & 3 & 2 \end{vmatrix}.$$

$$\begin{vmatrix} 1 & 0 & 1 \\ 2 & 1 & 1 \\ 1 & 3 & 2 \end{vmatrix} = 1\times1\times2+0\times1\times1+1\times2\times3-1\times1\times3-0\times2\times2-1\times1\times1=4.$$

线性代数 linear algebra

numpy.linalg模块包含线性代数的函数:

求解行列式、

计算逆矩阵

解线性方程组等

numpy.linalg.det() 函数计算输入矩阵的行列式。

```
a = np.array( [[1,0,1], [2,1,1], [1,3,2]] )
a
```

```
array([[1, 0, 1],
       [2, 1, 1],
       [1, 3, 2]])
```

```
np.linalg.det(a)
```

```
4.000000000000001
```

矩阵求逆

海豚数据科学实验室

若在相同数域上存在另一个n阶矩阵B，使得： AB=BA=E，
则我们称B是A的逆矩阵，而A则被称为可逆矩阵。
注：E为单位矩阵。

n阶方阵，且形如：

$$E_n=\begin{pmatrix}1 & & & 0\\ & 1 & & \\ & & \ddots & \\ 0 & & & 1\end{pmatrix}_{n\times n}$$

称 单位矩阵，记为E_n

$$K=\begin{pmatrix}k & & & 0\\ & k & & \\ & & \ddots & \\ 0 & & & k\end{pmatrix}=kE_n$$ 称为数量矩阵

4

二阶矩阵A的逆矩阵计算公式

$$A=\begin{vmatrix}a & b\\ c & d\end{vmatrix} \quad 则$$

$$A^{-1}=\frac{1}{ad-bc}\begin{vmatrix}d & -b\\ -c & a\end{vmatrix}$$

矩阵求逆

海豚数据科学实验室

np.linalg.inv()：矩阵求逆

矩阵必须是方阵且可逆，否则会抛出LinAlgError异常

若$A=\begin{pmatrix} a & b \\ c & d \end{pmatrix}$,则$A^{-1}=\frac{1}{ad-bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$

本题：$A=\begin{pmatrix} 1 & 2 \\ 1 & 4 \end{pmatrix}$,

则$A^{-1}=\frac{1}{2}\begin{pmatrix} 4 & -2 \\ -1 & 1 \end{pmatrix}=\begin{pmatrix} 2 & -1 \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix}$

```
A = np.array([[1,2], [1,4]])
A
```

```
array([[1, 2],
       [1, 4]])
```

```
B = np.linalg.inv(A)
B
```

```
array([[ 2. , -1. ],
       [-0.5,  0.5]])
```

```
np.dot(A, B)
```

```
array([[1., 0.],
       [0., 1.]])
```

```
np.dot(B, A)
```

```
array([[1., 0.],
       [0., 1.]])
```

方程组求解

例9. 线性方程组的矩阵表示（矩阵方程）

$$
\begin{cases}
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\
a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\
\cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \\
a_{s1}x_1 + a_{s2}x_2 + \ldots + a_{sn}x_n = b_s
\end{cases}
$$

$$
A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{s1} & a_{s2} & \cdots & a_{sn} \end{pmatrix} \quad \Updownarrow \quad x = \begin{pmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{pmatrix} \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \cdots \\ b_s \end{pmatrix}
$$

$$
Ax = b
$$

方程组求解

numpy.linalg中的函数solve可以求解形如 Ax = b 的线性方程组

```
A = np.array([[1, -2, 1], [0, 2, -8], [-4, 5, 9]])
A
```

```
array([[ 1, -2,  1],
       [ 0,  2, -8],
       [-4,  5,  9]])
```

```
b = np.array([0,8,-9])
b
```

```
array([ 0,  8, -9])
```

```
# 调用solve函数求解线性方程
x = np.linalg.solve(A,b)
x
```

```
array([29., 16.,  3.])
```

$$Ax = b$$

```
# 使用dot函数检查求得的解是否正确
np.dot(A, x)   # = b
```

```
array([ 0.,  8., -9.])
```

谢 谢!

海豚数据科学实验室