



首都师范大学

# 人工智能数学基础Python实践

## 第五章 线性代数基础

张苗苗

信息工程学院

## □在Python中实现以下操作：

- 向量和矩阵的创建
- 特殊矩阵
- 基本操作

## □所需的python库：

- **numpy**

**import numpy      # 使用numpy.函数名**

**import numpy as np    # 使用np.函数名**

**from numpy import \*    # 直接使用函数名**

## □ 直接创建:

```
A = [1,2,3,4,5,6]
```

```
B = numpy.array(A)
```

## □ 随机生成:

```
numpy.random.random() # 随机浮点数
```

```
numpy.random.randint() # 随机整数
```

`numpy.zeros()` # 零矩阵

`numpy.eye()` 或 `numpy.identity()` # 单位矩阵

`numpy.diag()` # 对角矩阵

`numpy.triu()` # 上三角矩阵

`numpy.tril()` # 下三角矩阵

`numpy.allclose()` # 矩阵相等

**A = [1,2,3,4,5,6]**

**B = numpy.array(A)**

**C = B.reshape(3,2) # 改变矩阵大小**

**C.shape** 或者 **numpy.shape(C)** # C的大小

**C[0]** # C的第0行

**C[:,0]** # 第0列

**C[0:2]** # C的前2行

**C[[0,2]]** # C的第0和第2行

**C[2,1]** # C的第2行第1列

# 矩阵的基本操作



**$A \pm B$  # 矩阵加减法**

**数字\*A # 矩阵的数乘运算**

**numpy.dot(A,B) # 矩阵乘法**

**numpy.multiply(A,B) # 矩阵对应元素相乘  $A*B$**

**$A**3$  #  $A$ 是matrix形式,  $A$ 是方阵,  $A*A*A$**

**$A**3$  #  $A$ 是array形式,  $A$ 是任意大小,  $A$ 对应元素3次方**

**numpy.transpose() # 转置**

**numpy.linalg.inv() # array求逆**

**numpy.linalg.det() # 求行列式**

**numpy.linalg.matrix\_rank() # 求秩**

# 练习1-- 求解线性方程组



$$\begin{cases} x + y + z = 2 \\ x + 2y + 4z = 3 \\ x + 3y + 9z = 5 \end{cases}$$



# 练习1-- 求解线性方程组



$$\begin{cases} x + y + z = 2 \\ x + 2y + 4z = 3 \\ x + 3y + 9z = 5 \end{cases}$$

$$AX = B$$

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{bmatrix}$$

$$X = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$B = \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix}$$

# 练习1-- 求解线性方程组



$$AX = B$$

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{bmatrix}$$

$$X = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$B = \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix}$$

方法1:  $X = A^{-1}B$

方法2: 用`numpy.linalg.solve()` 求解

方法3: 用`sympy.solve()` 求解



首都师范大学

# 第六章 特征值与矩阵分解

张苗苗

信息工程学院

## □在Python中实现以下操作：

- 特征值与特征向量
- SVD分解

## □所需的python库：

- `numpy`
- PIL中的Image模块

# 练习1-- 利用SVD对图像进行压缩

使用SVD对图像进行压缩步骤：

## 一、读取图像

```
from PIL import Image  
img = Image.open( 'filename' , ' r' )  
R0 = img[:, :, 0]  
G0 = img[:, :, 1]  
B0 = img[:, :, 2]
```

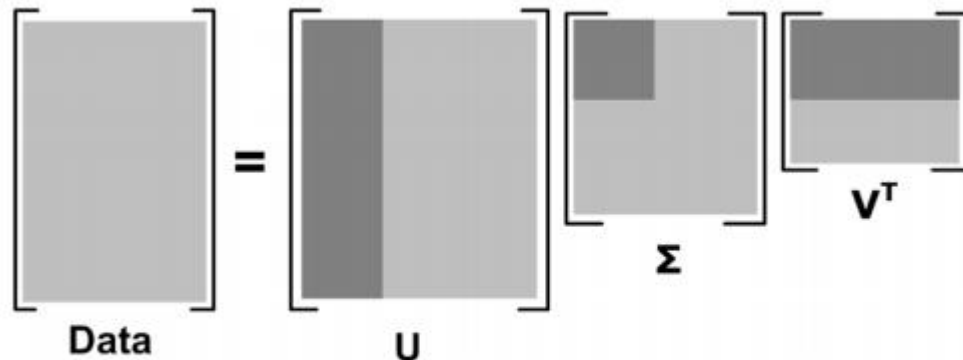


# 练习1-- 利用SVD对图像进行压缩

使用SVD对图像进行压缩步骤：

二、对每个颜色通道分别进行SVD分解并压缩

`numpy.linalg.svd()`



SVD的示意图。矩阵Data被分解。浅灰色区域是原始数据，深灰色区域是矩阵近似计算仅需要的数据

# 练习1-- 利用SVD对图像进行压缩



使用SVD对图像进行压缩步骤：

三、将压缩后的三个矩阵叠加起来，重构图像

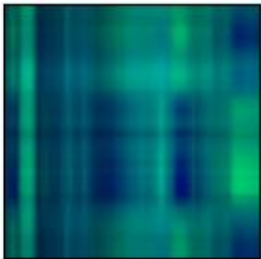
```
I = numpy.stack((R, G, B), 2)  
Image.fromarray(I).save('filename')
```

# 练习1-- 利用SVD对图像进行压缩

20%



40%



60%



80%



100%





# 练习1-- 利用SVD对图像进行压缩



显示图像：

```
import matplotlib.pyplot as plt
plt.figure()
k = 0
for i in range(20, 120, 20):
    k+=1
    img = Image.open(str(i) + "1.jpg", 'r')
    plt.subplot(2, 5, k)
    plt.imshow(img)
    plt.xticks([])
    plt.yticks([])
for i in range(20, 120, 20):
    k+=1
    img = Image.open(str(i) + "2.jpg", 'r')
    plt.subplot(2, 5, k)
    plt.imshow(img)
    plt.xticks([])
    plt.yticks([])
plt.show()
```

## 6.7小节



首都师范大学

谢谢!