hw1_51215903008_陈诺

作业链接

https://www.wolai.com/mathskiller/diWoQZ6fB2ZKq4iFqJrAXE?theme=light

习题部分

习题 1.矩阵 $A^2 = A, B^2 = B$,并且 B 的列是 A 的列的线性组合。证明 $AB = B$ 。

$$
\begin{gathered}
\text{由}B\text{的列是}A\text{的列的线性组合，得}B = AX \\
\because A^2 = A \\
\therefore (A - I)A = 0 \\
(A - I)AX = 0 \\
AAX = AX \\
AB = B
\end{gathered}
$$

习题 2.已知 $\beta = (1, 2, 1, 1)^T$ ，以及

$\alpha_1 = (1, 1, 1, 1)^T, \alpha_2 = (1, 1, -1, -1)^T, \alpha_3 = (1, -1, 1, -1)^T, \alpha_4 = (1, -1, -1, 1)^T.$

试将向量 β 表示成 $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ 的线性组合。

$$
\begin{gathered}
\begin{pmatrix} 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} x = \begin{pmatrix} 1 \\ 2 \\ 1 \\ 1 \end{pmatrix} \\
\begin{pmatrix} 1 & 1 & 1 & 1 & | & 1 \\ 1 & 1 & -1 & -1 & | & 2 \\ 1 & 1 & 1 & -1 & | & 1 \\ 1 & -1 & -1 & 1 & | & 1 \end{pmatrix} \\
\begin{pmatrix} 1 & 1 & 1 & 1 & | & 1 \\ 0 & 1 & 0 & 1 & | & 0 \\ 0 & 1 & 1 & 0 & | & 0 \\ 0 & 0 & 2 & 2 & | & -1 \end{pmatrix} \\
\begin{pmatrix} 1 & 1 & 1 & 1 & | & 1 \\ 0 & 1 & 0 & 1 & | & 0 \\ 0 & 0 & 1 & -1 & | & 0 \\ 0 & 0 & 0 & 1 & | & -\frac{1}{4} \end{pmatrix}
\end{gathered}
$$

$$
\begin{gathered}
x = \begin{pmatrix} \frac{5}{4} \\ \frac{1}{4} \\ -\frac{1}{4} \\ -\frac{1}{4} \end{pmatrix} \\
\beta = \frac{5}{4}\alpha_1 + \frac{1}{4}\alpha_2 - \frac{1}{4}\alpha_3 - \frac{1}{4}\alpha_4
\end{gathered}
$$

习题 3.设 A, B 为任意两个 n 阶方阵，证明： AB 和 BA 具有相同的特征多项式，即 $|\lambda E - AB| = |\lambda E - BA|$ 。

$$
\begin{gathered}
\text{由特征多项式}|\lambda E - A| = |\lambda E - B| <=> \lambda_A = \lambda_B, \\
\text{故只需证明}AB\text{的特征值是}BA\text{的特征值}
\end{gathered}
$$

$$1^\circ \text{当} \mathrm{A} = \mathrm{O}\text{或}\mathrm{B} = \mathrm{O}\text{时}, \mathrm{AB} = \mathrm{BA} = \mathrm{O}$$
$$\text{显然}\mathrm{AB}\text{与}BA\text{有相同的特征值}$$
$$2^\circ \text{当}A \neq O\text{且}B \neq O\text{时}$$
$$\text{设}\lambda\text{是AB的特征值}, \alpha \neq \vec{0}\text{是对应的特征向量}$$
$$i)\text{当}\lambda \neq 0\text{时}, AB\alpha = \lambda\alpha$$
$$BA \cdot B\alpha = \lambda \cdot B\alpha$$
$$\text{记}\beta = B\alpha, \text{得}BA\beta = \lambda\beta,$$
$$\text{由于}\lambda \neq 0, \alpha \neq \bar{0}, \text{得}AB\alpha = \lambda\alpha \neq \bar{0}, \text{可知}B\alpha = \beta \neq \bar{0}$$
$$\text{所以}\lambda\text{也是}BA\text{的特征值，}\beta\text{是对应的特征向量}$$
$$ii)\text{当}\lambda = 0\text{时}, AB\alpha = \lambda\alpha = \vec{0}, \text{又}\alpha \neq \bar{0}, \text{即}ABx = \vec{0}\text{有非零解即}$$
$$|AB| = |A| \cdot |B| = |BA| = 0,$$
$$\text{所以}BAx = \bar{0}\text{也有非零解，不妨记为}\beta$$
$$\text{即}BA\beta = \vec{0} = 0 \cdot \beta, \text{所以}\lambda = 0\text{也是}BA\text{的特征值.}$$
$$\text{综合}i)ii)\text{得：AB的特征值是BA的特征值}$$
$$\text{得证}$$

实操部分

习题 4.编写 Python 代码将你自选的一张图片旋转一定角度。该题的答案需包含三部分：

(1)代码；

(2)原来的图片；

(3)旋转后的图片.

以下为图片和实现细节，要翻转的是这只猫

```python
from PIL import Image
import matplotlib.pyplot as plt
import numpy as np
img=np.array (Image.open ('test.jpg') ) #读取图像
#旋转参数
theta=30/180*np.pi # 旋转角度
cos_theta=np.cos(theta)
sin_theta=np.sin(theta)
center_i=len (img) /2# 中心点位置
center_j=len(img[0])/2
```
Python

先做前向变换，旋转像素方式的数学表示是

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} \cos\delta & -\sin\delta \\ \sin\delta & \cos\delta \end{pmatrix} \begin{pmatrix} x_1 - s_1 \\ x_2 - s_2 \end{pmatrix} + \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}$$

```python
imgr=np.zeros_like (img) # 前向变换结果位置
for i in range(len(img)):
  for j in range(len(img[0])):
    yi=int(cos_theta*(i-center_i)-sin_theta*(j-center_j)+center_i)
    yj=int(sin_theta*(i-center_i)+ cos_theta*(j-center_j)+center_j)
    if (yi<0 or yj<0 or yi>=len(img)or yj>=len(img[0])):
      continue
    for k in range(3):
      imgr[yi] [yj] [k]=img[i][j][k]
# 输出结果图像
fig1=plt.figure()
ax=plt.subplot(111)
```

```python
ax.imshow(imgr)
ax.axis('off')
plt.show()
```

可见通过前向变换得到的图片有噪点

再做前向变换，旋转像素方式的数学表示是

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \cos\delta & \sin\delta \\ -\sin\delta & \cos\delta \end{pmatrix} \begin{pmatrix} y_1 - d_1 \\ y_2 - d_2 \end{pmatrix} + \begin{pmatrix} s_1 \\ s_2 \end{pmatrix}$$

```python
imgR=np.zero8_1ike (img) # 后向变换结果位置
for i in range(len(img)):
  for j in range(len(img[0])):
    xi= int(cos_theta*(i-center_i)+sin_theta*(j-center_j)+center_i)
    xj= int(-sin_theta*(i-center_i)+cos_theta*(j-center_j)+center_j)
    if xi<0 or xj<0 or xi>=len(img)or xj>=len(img[0]):
      continue
    for k in range(3):
      imgR[i][j][k]=img[xi] [xj] [k]
# 输出结果图像
fig2=plt.figure()
ax=plt.subplot(111)
ax.imshow(imgR)
ax.axis('off')
plt.show()
```