第05讲 矩阵操作 `课堂互动答案`

作者：欧新宇（Xinyu OU）

本文档所展示的测试结果，均运行于：Intel Core i7-7700K CPU 4.2GHz

【课堂互动一】 `矩阵的加法`

1. 在使用numpy进行矩阵加法运算的时候，要求执行加法的两个元素必须具有相同的形态。

A. 对
B. 错

答案及解析： A

两个矩阵执行加法运算需要满足两个条件，一是同型矩阵，二是对应元素相加。

2. 设存在矩阵$A=\begin{bmatrix}1.2 & 3.4 & 5.6\\ 7.8 & 9.1 & 0.7\end{bmatrix}$，和标量c=10，请使用*numpy*计算库求：*A* + c。

A. 无法直接对矩阵和标量进行计算

B. $A=\begin{bmatrix}11.2 & 13.4 & 15.6\\ 17.8 & 19.1 & 10.7\end{bmatrix}$

C. $A=\begin{bmatrix}11.2 & 13.4 & 15.6\\ 7.8 & 9.1 & 0.7\end{bmatrix}$

D. $A=\begin{bmatrix}11.2 & 3.4 & 5.6\\ 17.8 & 9.1 & 0.7\end{bmatrix}$

答案及解析： B

标量与矩阵相加的结果为标量与矩阵所有元素相加的和。

```python
import numpy as np
A = np.array([[1.2,3.4,5.6],[7.8,9.1,0.7]])
c = 10

print('A+c=\n {}'.format(A+c))
```

```
A+c=
 [[11.2 13.4 15.6]
 [17.8 19.1 10.7]]
```

【课堂互动二】 `矩阵乘法`

1.给定以下两个矩阵A和B，以下求“A的2倍与B的3倍的和”的Python描述正确的是（　　）。

$$A=\begin{bmatrix}150&250&50\\250&500&100\\300&700&120\\450&850&80\end{bmatrix},B=\begin{bmatrix}180&350&60\\300&550&120\\350&850&150\\500&850&100\end{bmatrix}$$

A. 2A+3B
B. 2*A+3*B
C. 2·A+3·B
D. 2×A+3×B
E. 2 ⊙ A+3 ⊙ B

答案及解析： B

选项A，一般用于文本书写中表达线性组合；选项B，为标准矩阵乘法表达；选项C，为向量内积；选项D，为向量外积；选项E，哈达玛乘积，即元素积。

2. 下列关于矩阵运算正确的包括（　　）。
A. $A^T+B^T=(A+B)^T$
B. $(\alpha+\beta)A^T=\alpha A^T+\beta A^T$
C. $(AB)^T=(BA)^T$
D. $A^T+B^T=(AB)^T$

答案及解析： AB

矩阵的乘法不满足交换律。

3. 给定向量 *u*, *v*，及标量 a,b，下列运算规律正确的是（　　　）。
A. a(*b***u**)=b(a**u**)
B. a(***u***+***v***)=a***u***+a***v***
C. a(***uv***)=a(***vu***)
D. $(a+b)$***v*** = a***v*** + b***v***

答案及解析： ABD

矩阵的乘法不满足交换律。

4. 按照下列代码，输出的结果正确的一个是（）。

```
import numpy as np

A=np.array([[1,2],[3,4]])
x=np.array([[1,1]]).T
print(np.dot(A,x))
```

A. $[7,$
$3]$
B. $[[3]$
$[7]]$
C. [[3, 7]]
D. [3, 7]

答案及解析： B

$A_{2\times2}$的矩阵与$x_{2\times1}$的矩阵相乘，结果矩阵的形态为2×1的矩阵。

```
import numpy as np

A=np.array([[1,2],[3,4]])
x=np.array([[1,1]]).T
print(np.dot(A,x))
```

```
[[3]
 [7]]
```

【课堂互动三】 `矩阵的秩和迹`

1.给定矩阵$A=\begin{bmatrix}1&2&3&6\\1&3&4&8\\3&5&6&12\\1&2&5&10\end{bmatrix}$，则矩阵A的秩为( )。

A. 1
B. 2
C. 3
D. 4

答案及解析： C

对于方阵来说，它的秩等于`行秩`和`列秩`中的较小数，也就是**线性无关**行（或列）的较小数，即: rank(A)=min(rank(row),rank(column))。

```
import numpy as np
A = np.array([[1,2,3,6],[1,3,4,8],[3,5,6,12],[1,2,5,10]])

print(A)
print('rank={}'.format(np.linalg.matrix_rank(A)))
```

```
[[ 1  2  3  6]
 [ 1  3  4  8]
 [ 3  5  6 12]
 [ 1  2  5 10]]
rank=3
```

2.给定矩阵$A=\begin{bmatrix}1.2&-3.6\\2.3&-1.48\end{bmatrix}$，则矩阵A的迹为( )。

A. -0.28
B. 1.7
C. 3.5
D. -2.08

答案及解析： C

矩阵的迹等于其主对角线上所有元素的和。

```
import numpy as np
A = np.array([[1.2, -.6], [2.3, -1.48]])

print(A)
print('trace = {}'.format(np.trace(A)))
```

```
[[ 1.2  -0.6 ]
 [ 2.3  -1.48]]
trace = -0.28
```

3.设矩阵A的迹trace(A)=3，矩阵B的迹trace(B)=4，求：$3A + 2B - 4B^T$。

A. -0.28
B. 11
C. -22.84
D. 无法计算

答案及解析： C

矩阵的迹满足基于数乘、加法和转置运算的不变性，因此对于线性组合来说，可以单独求出每个矩阵的迹，然后直接使用迹运算进行计算。即：$Tr(3A + 2B - 4B^T) = 3Tr(A) + 2Tr(B) - 4Tr(B)$

```
import numpy as np
A = np.array([[1.2, -.6], [2.3, -1.48]])
B = np.array([[3,6],[3,8]])


print('trace = {:.2f}'.format(np.trace(3*A+2*B-4*B.T)))
tra_A = np.trace(A)
tra_B = np.trace(B)
print('trace = {}'.format(3*tra_A+2*tra_B-4*tra_B.T))
```

```
trace = -22.84
trace = -22.84
```

【课堂互动四】 矩阵分块和张量的常用操作

1. 对于下列使用矩阵分块方法将方程式转换成向量形式 $a_1x_1 + a_2x_2 + a_3x_3 + a_4x_4 = b$，正确的一个是：（ ）。

$$\begin{cases} x_1 + 2x_2 + 3x_3 & = 2 \\ 2x_1 - 3x_2 + 4x_4 & = -2 \\ 2x_2 + 3x_3 + 5x_4 & = 5 \end{cases}$$

A. $\begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix} x_1 + \begin{bmatrix} 2 \\ -3 \\ 2 \end{bmatrix} x_2 + \begin{bmatrix} 3 \\ 0 \\ 3 \end{bmatrix} x_3 + \begin{bmatrix} 0 \\ 4 \\ 5 \end{bmatrix} x_4 = \begin{bmatrix} 2 \\ -2 \\ 5 \end{bmatrix}$

B. $\begin{bmatrix} 1 \\ 2 \end{bmatrix} x_1 + \begin{bmatrix} 2 \\ -3 \\ 2 \end{bmatrix} x_2 + \begin{bmatrix} 3 \\ 3 \end{bmatrix} x_3 + \begin{bmatrix} 4 \\ 5 \end{bmatrix} x_4 = \begin{bmatrix} 2 \\ -2 \\ 5 \end{bmatrix}$

答案及解析： A

利用矩阵分块原理进行方程式的改写需要对齐未知数，对于不存在未知数的等式以补零方式构建向量。

2. 给定矩阵M，请问以下python语句可以实现调整矩阵维度顺序的语句是（），可以实现将矩阵M拉成行向量的语句是（）。

A. M.transpose(1,2,0) M.reshape(-1,1)
B. M.concatenate(1,2,0) M.reshape(1,-1)
C. M.concatenate(1,2,0) M.reshape(-1,1)
D. M.transpose(1,2,0) M.reshape(1,-1)

答案及解析： D

M.transpose(1,2,0)将矩阵的第0列调整到最后一个维度，并将第1，2列调整到第0和第1个维度；M.reshape(row, col)可以实现将矩阵重新调整为row行，col列，当值为`-1`时，表示跟随其他维度进行最大化调整；concatenate(M,axis=0)可以实现将矩阵M按照第0个维度进行合并。