Урок 2

Матрицы и матричные операции. Часть 1

```
import numpy as np
```

Практическое задание. Часть 1

1. Установите, какие произведения матриц AB и BA определены, и найдите размерности полученных матриц:

а) A — матрица 4×2, B — матрица 4×2;

б) A — матрица 2×5, B — матрица 5×3;

в) A — матрица 8×3, B — матрица 3×8;

г) A — квадратная матрица 4×4, B — квадратная матрица 4×4.

Решение

Правильно:

Произведением матрицы $A=\|a_{ij}\|$, имеющей порядки m и n, и матрицы $B=\|b_{ij}\|$, включающей порядки n и k, называется матрица $C=\|c_{ij}\|$ с порядками m и k:

$$C=\begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1k} \\ c_{21} & c_{22} & \cdots & c_{2k} \\ \cdots & \cdots & \ddots & \cdots \\ c_{m1} & c_{m2} & \cdots & c_{mk} \end{pmatrix},$$

наполненная элементами, определяемыми формулой:

$$c_{ij}=\sum_{p=1}^{n} a_{ip}b_{pj}.$$

Матрицу A можно умножить не на любую матрицу B: **важно, чтобы число столбцов матрицы A было равно числу строк матрицы B.**

Проверим это условие на предложенных вариантах:

а) A — матрица 4×2, B — матрица 4×2; - **AB - 2 не равно 4, перемножить нельзя, BA - тоже нельзя перемножить, так как 2(стобла) не равно количеству строк - 4**

б) A — матрица 2×5, B — матрица 5×3; - **AB - перемножить можно, размерность 2 x 3, BA - перемножить нельзя, так как 3 не равно 2** в) A — матрица 8×3, B — матрица 3×8; - **AB перемножить можно, размерность 8 x 8, BA перемножить можно, размерность 3 x 3**

г) A — квадратная матрица 4×4, B — квадратная матрица 4×4. - **AB, BA - перемножить можно, размерность 4 x 4**

2. Найдите сумму и произведение матриц $A=\begin{pmatrix}1 & -2\\ 3 & 0\end{pmatrix}$ и $B=\begin{pmatrix}4 & -1\\ 0 & 5\end{pmatrix}$.

2. Решение

Сумма:

$$\begin{pmatrix}1 & -2\\ 3 & 0\end{pmatrix}+\begin{pmatrix}4 & -1\\ 0 & 5\end{pmatrix}=\begin{pmatrix}5 & -3\\ 3 & 5\end{pmatrix}$$

```
#проверка

a = np.array([[1, -2], [3, 0]])
b = np.array([[4, -1], [0, 5]])
a + b
```

```
array([[ 5, -3],
       [ 3,  5]])
```

$$\begin{pmatrix}1 & -2\\ 3 & 0\end{pmatrix}*\begin{pmatrix}4 & -1\\ 0 & 5\end{pmatrix}=\begin{pmatrix}4 & -11\\ 12 & -3\end{pmatrix}$$

```
np.dot(a, b)
```

```
array([[  4, -11],
       [ 12,  -3]])
```

найдем BA, так как матрицы квадратные:

$$\begin{pmatrix}4 & -1\\ 0 & 5\end{pmatrix}*\begin{pmatrix}1 & -2\\ 3 & 0\end{pmatrix}=\begin{pmatrix}1 & -8\\ 15 & 0\end{pmatrix}$$

```
np.dot(b, a)
```

```
array([[ 1, -8],
       [15,  0]])
```

3. Из закономерностей сложения и умножения матриц на число можно сделать вывод, что матрицы одного размера образуют линейное пространство. Вычислите линейную комбинацию $3A-2B+4C$ для матриц $A=\begin{pmatrix}1 & 7\\ 3 & -6\end{pmatrix}$, $B=\begin{pmatrix}0 & 5\\ 2 & -1\end{pmatrix}$, $C=\begin{pmatrix}2 & -4\\ 1 & 1\end{pmatrix}$.

Решение

$$3*\begin{pmatrix}1 & 7\\ 3 & -6\end{pmatrix}-2*\begin{pmatrix}0 & 5\\ 2 & -1\end{pmatrix}+4*\begin{pmatrix}2 & -4\\ 1 & 1\end{pmatrix}=\begin{pmatrix}3 & 21\\ 9 & -18\end{pmatrix}+\begin{pmatrix}0 & -10\\ -4 & 2\end{pmatrix}+\begin{pmatrix}8 & -16\\ 4 & 4\end{pmatrix}=\begin{pmatrix}11 & -5\\ 9 & -12\end{pmatrix}$$

```
#проверка
a = np.array([[1, 7], [3, -6]])
b = np.array([[0, 5], [2, -1]])
c = np.array([[2, -4], [1, 1]])
3 * a + (-2) * b + 4 * c
```

```
array([[ 11,  -5],
       [  9, -12]])
```

4. Дана матрица $A=\begin{pmatrix}4 & 1\\ 5 & -2\\ 2 & 3\end{pmatrix}$. Вычислите $A\,A^T$ и $A^T\,A$.

Решение

$$A^T=\begin{pmatrix}4 & 5 & 2\\ 1 & -2 & 3\end{pmatrix}$$

$$A\,A^T=\begin{pmatrix}4 & 1\\ 5 & -2\\ 2 & 3\end{pmatrix}*\begin{pmatrix}4 & 5 & 2\\ 1 & -2 & 3\end{pmatrix}=\begin{pmatrix}17 & 18 & 11\\ 18 & 29 & 4\\ 11 & 4 & 13\end{pmatrix}$$

$A^{T}A = \begin{pmatrix} 4 & 5 & 2\ 1 & -2 & 3\end{pmatrix} * \begin{pmatrix} 4 & 1\ 5 & -2\ 2 & 3 \end{pmatrix} = \begin{pmatrix} 45 & 0\ 0 & 14 \end{pmatrix} $

```
A = np.array([[4, 1], [5, -2], [2, 3]])
A.T
```

```
array([[ 4,  5,  2],
       [ 1, -2,  3]])
```

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

```
array([[17, 18, 11],
       [18, 29,  4],
       [11,  4, 13]])
```

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

```
array([[45,  0],
       [ 0, 14]])
```

__5*.__ Напишите на Python функцию для перемножения двух произвольных матриц, не используя NumPy.

```
"""Класс для операций сложения и умножения матриц"""
class Matrix:
    def __init__(self, matrix):
        self.matrix= matrix
```

```
    def __str__(self):
        return ','.join(map(str, self.matrix)).replace(']', '\
n').replace('[', '').replace(',', '')

    def __add__(self, other):
        if len(self.matrix) == len(other.matrix):
            for row1, row2 in zip(self.matrix, other.matrix):
                if len(row1) != len(row2):
                    return f'Количество столбцов не равно, эти матрицы
нельзя сложить'
            result = [[a + b for a, b in zip(row1, row2)] for row1,
row2 in zip(self.matrix, other.matrix)]
            return ','.join(map(str, result)).replace(']', '\
n').replace('[', '').replace(',', '')
        else:
            return f'Количество строк не равно, эти матрицы нельзя
сложить'

    def __mul__(self, other):
        result = []
        try:
            for row in self.matrix:
                row_result = []
                for i in range(len(other.matrix[0])):
                    num = 0
                    for idx, el in enumerate(row):
                        num += (el * other.matrix[idx][i])
                    row_result.append(num)
                result.append(row_result)
            return ','.join(map(str, result)).replace(']', '\
n').replace('[', '').replace(',', '')
        except:
            return f'Умножение матриц данной размерности невозможно'



m_1 = Matrix([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
m_2 = Matrix([[10, 20, 30], [40, 50, 60], [70, 80, 90]])
print(m_1)
print(m_2)
print(m_1 + m_2)

m_3 = Matrix([[4, 1], [5, -2], [2, 3]])
m_4 = Matrix([[ 4,  5,  2], [ 1, -2,  3]])

print(m_1 * m_2)
print(m_3 * m_4)
print(m_4 * m_3)
```

```
1 2 3
4 5 6
7 8 9

10 20 30
40 50 60
70 80 90

11 22 33
44 55 66
77 88 99

300 360 420
660 810 960
1020 1260 1500

17 18 11
18 29 4
11 4 13

45 0
0 14
```

```
#проверка работы класса
a = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
b = np.array([[10, 20, 30], [40, 50, 60], [70, 80, 90]])
np.dot(a, b)
```

```
array([[ 300,  360,  420],
       [ 660,  810,  960],
       [1020, 1260, 1500]])
```

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

```
array([[17, 18, 11],
       [18, 29,  4],
       [11,  4, 13]])
```

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

```
array([[45,  0],
       [ 0, 14]])
```

Практическое задание. Часть 2

1. Вычислите определитель:

а)

$$\left| \begin{matrix} sinx & -cosx \\ cosx & sinx \end{matrix} \right|;$$

б)

$$\left| \begin{matrix} 4 & 2 & 3 \\ 0 & 5 & 1 \\ 0 & 0 & 9 \end{matrix} \right|;$$

в)

$$\left| \begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix} \right|.$$

Решение

а)

$$\left| \begin{matrix} sinx & -cosx \\ cosx & sinx \end{matrix} \right| = sin^2 x + cos^2 x = 1$$

б)

$$\left| \begin{matrix} 4 & 2 & 3 \\ 0 & 5 & 1 \\ 0 & 0 & 9 \end{matrix} \right| = 4*(5*9-1*0)-2(0*9-1*0)+3(0*0-5*0)=4*45=180;$$

```
b = np.array([[4, 2, 3], [0, 5, 1], [0, 0, 9]])
np.linalg.det(b)
```

```
180.0
```

в)

$$\left| \begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix} \right| = 1*(5*9-6*8)-2*(4*9-7*6)+3*(4*8-5*7)=0$$

```
c = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
round(np.linalg.det(c), 2)
```

```
0.0
```

2. Определитель матрицы A равен 4. Найдите:

а) $det(A^2)$;

б) $det(A^T)$;

в) $det(2A)$.

Решение

а) $det(A^2)$;

Воспользуемся правилом:

9. Для двух квадратных матриц одинакового размера:

$$det(AB) = det\,A \cdot det\,B.$$

$det(A^2) = 4 * 4 = 16$;

б) $det(A^T)$;

Воспользуемся правилом:

1. Определитель транспонированной матрицы равен определителю исходной:

$$det\,A^T = det\,A.$$

$$det(A^T) = 4$$

в) $det(2A)$.

Воспользуемся правилом:

2. Умножение строки или столбца матрицы на число λ приведёт к умножению определителя матрицы на то же число.

Таким образом, если каждая строка увеличится в 2 раза, определитель всей матрицы увеличится в 2^n раз, где n - количество строк

$det(2A) = 2^n * 4 = 2^{n+2}$.

3. Докажите, что матрица:

$$\begin{pmatrix} -2 & 7 & -3 \\ 4 & -14 & 6 \\ -3 & 7 & 13 \end{pmatrix}$$

вырожденная.

Решение

Матрица является вырожденной, если ее определитель равен нулю

По правилу:

6. Если две строки (столбца) матрицы линейно зависимы, то определитель этой матрицы равен нулю.

Так как вторая строка матрицы равна первой строке, умноженной на -2, то эти две строки линейно зависимы, соответственно

определитель матрицы равен нулю, что и означает, что матрица вырожденная

Для верности посчитаем определитель матрицы:

$$\begin{vmatrix} -2 & 7 & -3 \\ 4 & -14 & 6 \\ -3 & 7 & 13 \end{vmatrix} = -2(-14*13-6*7)-7(4*13+6*3)-3(4*7-(14*3))=448-490+42=0$$

```
c = np.array([[-2, 7, -3], [4, -14, 6],[-3, 7, 13]])
np.linalg.det(c)
```

```
0.0
```

4. Найдите ранг матрицы:

а) $\begin{pmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \\ 2 & 3 & 4 \end{pmatrix};$

б) $\begin{pmatrix} 0 & 0 & 2 & 1 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 4 & 3 \\ 2 & 3 & 5 & 6 \end{pmatrix}.$

Решение

а) $\begin{pmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \\ 2 & 3 & 4 \end{pmatrix};$

Отнимем из второй строки первую:

$\begin{pmatrix} 1 & 2 & 3 \\ 0 & -1 & -2 \\ 2 & 3 & 4 \end{pmatrix};$

Теперь отнимем из третьей строки первую, умноженную на 2

$\begin{pmatrix} 1 & 2 & 3 \\ 0 & -1 & -2 \\ 0 & -1 & -2 \end{pmatrix};$

Так как вторая и третья строки равны, удалим третью строку.

$\begin{pmatrix} 1 & 2 & 3 \\ 0 & -1 & -2 \end{pmatrix};$

Получилась верхне-треугольная матрица, таким образом, ранг матрицы равен 2.

```python
#проверка
a = np.array([[1, 2, 3], [1, 1, 1],[2, 3, 4]])

np.linalg.matrix_rank(a)
```

2

б) $\begin{pmatrix} 0 & 0 & 2 & 1 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 4 & 3 \\ 2 & 3 & 5 & 6 \end{pmatrix}$.

Переместим последную строку наверх на место первой

$$\begin{pmatrix} 2 & 3 & 5 & 6 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 4 & 3 \end{pmatrix}.$$

Из четвертой строки отнимем вторую, умноженную на 2

$$\begin{pmatrix} 2 & 3 & 5 & 6 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Получилась верхне-треугольная матрица.

Правило:

7. Определитель матрицы треугольного вида равен произведению элементов, стоящих на её главной диагонали:

В нашем случае произведение (и определитель) равны нулю, так как во второй строке стоит ноль на главной диагонали.

Если убрать вторую строку, то ранг матрицы получается равен 3.

```python
#проверка
a = np.array([[0, 0, 2, 1], [0, 0, 2, 2], [0, 0, 4, 3], [2, 3, 5, 6]])

np.linalg.matrix_rank(a)
```

3