МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ

Отчет по лабораторной работе №3 по дисциплине: технологии распознавания образов

Выполнил:
студент группы ПИЖ-б-о-21-1
Прокопов Дмитрий Владиславович

Проверил:
доцент кафедры инфокоммуникаций
Романкин Р.А.

Ставрополь, 2023 г.

ВЫПОЛНЕНИЕ

Примеры

```
v_hor_np=np.array([1,2])
print(v_hor_np)
```

```
[1 2]
```

```
v1=np.zeros((5,))
print(v1)
```

```
[0. 0. 0. 0. 0.]
```

```
v2=np.ones((5,))
print(v2)
```

```
[1. 1. 1. 1. 1.]
```

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

```
print(v_vert)
```

```
[[1]
 [2]]
```

```
v_vert_z=np.zeros((5,1))
print(v_vert_z)
```

```
[[0.]
 [0.]
 [0.]
 [0.]
 [0.]]
```

```
v_vert_o=np.ones((5,1))
print(v_vert_o)
```

```
[[1.]
 [1.]
 [1.]
 [1.]
 [1.]]
```

```
m_sqr_arr=np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
print(m_sqr_arr)
```

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

```
m=np.matrix('1 2 3; 4 5 6; 7 8 9')
diag=np.diag(m)
print(diag)
```

```
[1 5 9]
```

```
md=np.diag(np.diag(m))
print(md)
```

```
[[1 0 0]
 [0 5 0]
 [0 0 9]]
```

```
m_eye=np.eye(5)
print(m_eye)
```

```
[[1. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0.]
 [0. 0. 1. 0. 0.]
 [0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 1.]]
```

```
m_z=np.zeros((3,3))
print(m_z)
```

```
[[0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]]
```

```
a=np.matrix('1 2 3; 4 5 6')
print(a)
```

```
[[1 2 3]
 [4 5 6]]
```

```
a_t=a.transpose()
print(a_t)
```

```
[[1 4]
 [2 5]
 [3 6]]
```

```
print(a.T)
```

```
[[1 4]
 [2 5]
 [3 6]]
```

```
a=np.matrix('1 2; 3 4')
b=np.matrix('5 6; 7 8')
l=(a.dot(b)).T
r=(b.T).dot(a.T)
print(l)
print(r)
```

```
[[19 43]
 [22 50]]
[[19 43]
 [22 50]]
```

```
a=np.matrix('1 2 3; 4 5 6')
k=3
l=(k*a).T
r=k*(a.T)
print(l)
print(r)
```

```
[[ 3 12]
 [ 6 15]
 [ 9 18]]
[[ 3 12]
 [ 6 15]
 [ 9 18]]
```

```
a=np.matrix('1 6 3; 8 2 7')
b=np.matrix('8 1 5; 6 9 12')
c=a+b
print(c)
```

```
[[ 9  7  8]
 [14 11 19]]
```

```
a=np.matrix('1 2 3; 4 5 6')
b=np.matrix('7 8; 9 1; 2 3')
c=a.dot(b)
print(c)
```

```
[[31 19]
 [85 55]]
```

```
a=np.matrix('-4 -1 2; 10 4 -1; 8 3 1')
print(a)
np.linalg.det(a)
```

```
[[-4 -1  2]
 [10  4 -1]
 [ 8  3  1]]

-14.000000000000009
```

```
a=np.matrix('-4 -1 2; 0 0 0; 8 3 1')
print(a)
np.linalg.det(a)
```

```
[[-4 -1  2]
 [ 0  0  0]
 [ 8  3  1]]

0.0
```

```
a=np.matrix('-4 -1 2; 10 4 -1; 8 3 1')
print(a)
k=2
b=a.copy()
b[2,:]=k*b[2,:]
print(b)
det_a=round(np.linalg.det(a), 3)
det_b=round(np.linalg.det(b), 3)
det_a*k
det_b
```

```
[[-4 -1  2]
 [10  4 -1]
 [ 8  3  1]]
[[-4 -1  2]
 [10  4 -1]
 [16  6  2]]

-28.0
```

```
a=np.matrix('1 -3; 2 5')
a_inv=np.linalg.inv(a)
a_inv_inv=np.linalg.inv(a_inv)
print(a)
print(a_inv_inv)
```

```
[[ 1 -3]
 [ 2  5]]
[[ 1. -3.]
 [ 2.  5.]]
```

```
print(m_eye)
```

```
[[1. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0.]
 [0. 0. 1. 0. 0.]
 [0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 1.]]
```

```
rank=np.linalg.matrix_rank(m_eye)
print(rank)
```

```
5
```

Ответы на вопросы:

1. Виды: двухмерный массивы и вектор (матрица, у которой есть только один столбец или одна строка). Удобным способом создание матриц является библиотека Numpy при помощи np.array()
2. A.transpose() или A.T
3. Свойства:
   - Дважды транспонированная матрица равна исходной матрице
   -Транспонирование суммы матриц равно сумме транспонированных матриц
   -Транспонирование произведения матриц равно произведению транспонированных матриц расставленных в обратном порядке
   -Транспонирование произведения матрицы на число равно произведению этого числа на транспонированную матрицу
   - Определители исходной и транспонированной матрицы совпадают
4. Для транспонированния матрицы используется transpose
5. Виды действий:
   -Умножение матрицы на число
   -Сложение матриц
   -Умножение матриц

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix},$$

$$C = 3 \cdot A,$$

$$C = \begin{pmatrix} 3 & 6 & 9 \\ 12 & 15 & 18 \end{pmatrix}.$$

6.
7. Свойства:
   -Произведение единицы и любой заданной матрицы равно заданной матрице
   -Произведение нуля и любой матрицы равно нулевой матрице, размерность которой равна исходной матрицы
   -Произведение матрицы на сумму чисел равно сумме произведений матрицы на каждое из этих чисел
   -Произведение матрицы на произведение двух чисел равно произведению второго числа и заданной матрицы, умноженному на первое число
   -Произведение суммы матриц на число равно сумме произведений этих матриц на заданное число

$$A = \begin{pmatrix} 1 & 6 & 3 \\ 8 & 2 & 7 \end{pmatrix}, B = \begin{pmatrix} 8 & 1 & 5 \\ 6 & 9 & 12 \end{pmatrix},$$

$$C = A + B,$$

$$C = \begin{pmatrix} 1+8 & 6+1 & 3+5 \\ 8+6 & 2+9 & 7+12 \end{pmatrix} = \begin{pmatrix} 9 & 7 & 8 \\ 14 & 11 & 19 \end{pmatrix}.$$

8. Аналогично и с вычитанием матриц
9. Свойства:
   -Коммутативность сложения. От перестановки матриц их сумма не изменяется
   -Ассоциативность сложения. Результат сложения трех и более матриц не зависит от порядка, в котором эта операция будет выполняться
   -Для любой матрицы существует противоположная ей, такая, что их сумма является нулевой матрицей
10. + и –

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}, \; B = \begin{pmatrix} 7 & 8 \\ 9 & 1 \\ 2 & 3 \end{pmatrix},$$

$$C = A \times B,$$

$$C = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \times \begin{pmatrix} 7 & 8 \\ 9 & 1 \\ 2 & 3 \end{pmatrix} =$$

$$= \begin{pmatrix} 1 \cdot 7 + 2 \cdot 9 + 3 \cdot 2 & 1 \cdot 8 + 2 \cdot 1 + 3 \cdot 3 \\ 4 \cdot 7 + 5 \cdot 9 + 6 \cdot 2 & 4 \cdot 8 + 5 \cdot 1 + 6 \cdot 3 \end{pmatrix} = \begin{pmatrix} 31 & 19 \\ 85 & 55 \end{pmatrix}.$$

11.
12. Свойства:
    -Ассоциативность умножения. Результат умножения матриц не зависит от порядка, в котором будет выполняться эта операция
    -Дистрибутивность умножения. Произведение матрицы на сумму матриц равно сумме произведений матриц
    -Умножение матриц в общем виде не коммутативно. Это означает, что для матриц не выполняется правило независимости произведения от перестановки множителей
    -Произведение заданной матрицы на единичную равно исходной матрице
    -Произведение заданной матрицы на нулевую матрицу равно нулевой матрице
13. A-матрица
    B-вторая матрица
    C=A.dot(B)-произведение матриц
14. Определитель матрицы — это некоторое число, с которым можно сопоставить любую квадратную матрицу.
    Свойства:
    -Определитель матрицы остается неизменным при ее транспонировании
    -Если у матрицы есть строка или столбец, состоящие из нулей, то определитель такой матрицы равен нулю
    -При перестановке строк матрицы знак ее определителя меняется на противоположный
    -Если у матрицы есть две одинаковые строки, то ее определитель равен нулю
    -Если все элементы строки или столбца матрицы умножить на какое-то число, то и определитель будет умножен на это число
    -Если все элементы строки или столбца можно представить как сумму двух слагаемых, то определитель такой матрицы равен сумме определителей двух соответствующих матриц
    -Если к элементам одной строки прибавить элементы другой строки, умноженные на одно и тоже число, то определитель матрицы не

изменится
-Если строка или столбец матрицы является линейной комбинацией других строк (столбцов), то определитель такой матрицы равен нулю
-Если матрица содержит пропорциональные строки, то ее определитель равен нулю

15. np.linalg.det(“Матрица”)

Обратной матрицей A^{-1} матрицы A называют матрицу, удовлетворяющую следующему равенству:

$$A \times A^{-1} = A^{-1} \times A = E,$$

где – E это единичная матрица.

16.
17. Свойства:
-Обратная матрица обратной матрицы есть исходная матрица
-Обратная матрица транспонированной матрицы равна транспонированной матрице от обратной матрицы
-Обратная матрица произведения матриц равна произведению обратных матриц
18. Функция inv(). Пример: B=np.linalg.inv(A)
19. Самостоятельно изучите метод Крамера для решения систем линейных уравнений. Приведите алгоритм решения системы линейных уравнений

методом Крамера средствами библиотеки NumPy.

Метод Крамера

In [46]:
```
slau = np.random.randint(0, 15, size = (3, 4))
```

In [60]:
```
slau_a = slau[:, 0:3]
slau_b = np.ones((3, 1))
x = np.ones((3, 1))
slau_b[0, 0] = slau[0, 3]
slau_b[1, 0] = slau[1, 3]
slau_b[2, 0] = slau[2, 3]

print("slau_a")
print(slau_a)
print("slau_b")
print(slau_b)
```

```
slau_a
[[ 7 14  4]
 [ 1  0  5]
 [ 7  3 10]]
slau_b
[[8.]
 [6.]
 [5.]]
```

In [61]:
```
#Найдем определитель
A_det = np.linalg.det(slau_a)

if A_det != 0:
    #Найдем дополнительные определители
    for i in range(3):
        A_dop = slau_a.copy()
        A_dop[:, i] = slau_b[:, 0]
        x[i,0] = round(np.linalg.det(A_dop), 3) / round(np.linalg.det(slau_a), 3)
    print("Решения:")
    print(x)
else:
    print("Матрица вырожденная, нельзя продолжить")
```

```
Решения:
[[-2.09338521]
 [ 1.15564202]
 [ 1.61867704]]
```

20. Самостоятельно изучите матричный метод для решения систем линейных уравнений. Приведите алгоритм решения системы линейных уравнений матричным методом средствами библиотеки NumPy

Метод Крамера

In [46]:
```
slau = np.random.randint(0, 15, size = (3, 4))
```

In [60]:
```
slau_a = slau[:, 0:3]
slau_b = np.ones((3, 1))
x = np.ones((3, 1))
slau_b[0, 0] = slau[0, 3]
slau_b[1, 0] = slau[1, 3]
slau_b[2, 0] = slau[2, 3]

print("slau_a")
print(slau_a)
print("slau_b")
print(slau_b)
```

```
slau_a
[[ 7 14  4]
 [ 1  0  5]
 [ 7  3 10]]
slau_b
[[8.]
 [6.]
 [5.]]
```

In [61]:
```
#Найдем определитель
A_det = np.linalg.det(slau_a)

if A_det != 0:
    #Найдем дополнительные определители
    for i in range(3):
        A_dop = slau_a.copy()
        A_dop[:, i] = slau_b[:, 0]
        x[i,0] = round(np.linalg.det(A_dop), 3) / round(np.linalg.det(slau_a), 3)
    print("Решения:")
    print(x)
else:
    print("Матрица вырожденная, нельзя продолжить")
```

```
Решения:
[[-2.09338521]
 [ 1.15564202]
 [ 1.61867704]]
```