Lab 4 19-08-2022

August 24, 2022

[2]:

```
import numpy as np
```

0.1 8

0.1.1 Write a numpy program to access array by column

[3]:

```
array = np.array([[1,2,3],[4,8,11],[30,5,9]])
print(array)
print("accessing 3rd column: ")
print(array[:, 2])
print("accessing 2nd column: ")
print(array[:, 1])
print("accessing 1st column: ")
print(array[:, 0])
```

```
[[ 1  2  3]
 [ 4  8 11]
 [30  5  9]]
accessing 3rd column:
[ 3 11  9]
accessing 2nd column:
[2 8 5]
accessing 1st column:
[ 1  4 30]
```

0.2 9

0.2.1 Write a numpy program to compute

#### a) multiplication of two given matrices. #### b) addition of two matrices

[4]:

```
m1 = np.array([[1,2,3],[4,8,11],[30,5,9]])
m2 = np.array([[5,4,3],[1,1,1],[2,4,5]])
print("multiplication is: \n",m1@m2)
print("Addition is: \n",m1+m2)
```

```
multiplication is:
 [[ 13  18  20]
 [ 50  68  75]
 [173 161 140]]
Addition is:
 [[ 6  6  6]
 [ 5  9 12]
 [32  9 14]]
```

0.3 10

0.3.1 Write a numpy program to eigenvalues and eigenvectors of a given matrix

[8]:

```
m = np.array([[3,1],[2,2]])
w,v = np.linalg.eig(m)
print("Eigen values: ",w)
print("Eigen vectors: \n",v)
```

```
Eigen values:  [4. 1.]
Eigen vectors:
 [[ 0.70710678 -0.4472136 ]
 [ 0.70710678  0.89442719]]
```

0.4 11

0.4.1 Write a program to find transpose, determinant and inverse of a matrix.

[12]:

```
a = np.array([[10,2,3],[4,8,11],[1,5,9]])
print("determinant is: ",np.linalg.det(a))
print("transpose is: \n",np.transpose(a))
print("inverse is: \n",np.linalg.inv(a))
```

```
determinant is:  156.0
transpose is:
 [[10  4  1]
 [ 2  8  5]
 [ 3 11  9]]
inverse is:
 [[ 0.10897436 -0.01923077 -0.01282051]
 [-0.16025641  0.55769231 -0.62820513]
 [ 0.07692308 -0.30769231  0.46153846]]
```

[13]:

```
b = np.arange(1, 10)
print(b)
np.savetxt('file.txt', b, delimiter=' ')
w=np.loadtxt("file.txt")
q=np.loadtxt(open("file.txt","rb"),delimiter=" ",dtype=np.int32)
print(w)
print(q)
```

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

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

[14]:

```
data = np.array( [[10,20,30], [40,50,60], [70,80,90]] )
print(data)
np.savetxt("test.csv", data, delimiter=",")
#k=np.loadtxt("test.csv")
#print(k)
np.loadtxt(open("test.csv","rb"),delimiter=",")
```

```
[[10 20 30]
 [40 50 60]
 [70 80 90]]
```

[14]:

```
array([[10., 20., 30.],
       [40., 50., 60.],
       [70., 80., 90.]])
```