Assignment-1

Based on NumPy

Q1: Questions on Basic NumPy Array

(a) Reverse the NumPy array: arr = np.array([1, 2, 3, 6, 4, 5]) arr[::-1]

(b) Flatten the NumPy arr: array1 = np.array([[1, 2, 3], [2, 4, 5], [1, 2, 3]]) using any two NumPy in-built methods array.flatten()/.ravel()

(c) Compare the following numpy arrays:

arr1 = np.array([[1, 2], [3, 4]]) .array_equal(arr1,arr2)
arr2 = np.array([[1, 2], [3, 4]])

(d) Find the most frequent value and their indice(s) in the following arrays: .bincount(x)
.argmax(countx)
.where(x=maxcount)

i. x = np.array([1,2,3,4,5,1,2,1,1,1])
ii. y = np.array([1, 1, 1, 2, 3, 4, 2, 4, 3, 3,])

(e) For the array gfg = np.matrix('[4, 1, 9; 12, 3, 1; 4, 5, 6]'), find

i. Sum of all elements .sum(mat)
ii. Sum of all elements row-wise .sum(mat,axis=1)
iii. Sum of all elements column-wise .sum(mat,axis=0)

(f) For the matrix: n_array = np.array([[55, 25, 15],[30, 44, 2],[11, 45, 77]]), find

i. Sum of diagonal elements .trace(mat)
ii. Eigen values of matrix .linalg.eig(mat) - return eigval, eigmat both
iii. Eigen vectors of matrix .linalg.inv(mat)
iv. Inverse of matrix .linalg.det(mat)
v. Determinant of matrix

(g) Multiply the following matrices and also find covariance between matrices using NumPy:

i. p = [[1, 2], [2, 3]] .dot(mat1,mat2)
.cov(mat1,mat2)
q = [[4, 5], [6, 7]] .dot(x,y.T)
ii. p = [[1, 2], [2, 3], [4, 5]] .outer(x.flatten(),y.flatten()
q = [[4, 5, 1], [6, 7, 2]] .array(np.meshgrid(x.flatten(), y.flatten())).T.reshape(-1, 2)

(h) For the matrices: x = np.array([[2, 3, 4], [3, 2, 9]]); y = np.array([[1, 5, 0], [5, 10, 3]]), find inner, outer and cartesian product?

Q2: Based on NumPy Mathematics and Statistics

(a) For the array: array = np.array([[1, -2, 3],[-4, 5, -6]])

i. Find element-wise absolute value .abs(array)
ii. Find the 25th, 50th, and 75th percentile of flattened array, for each column, for each row. .percentile(flattened_array, [25, 50, 75]) percentile(array, [25, 50, 75], axis=0/1)
iii. Mean, Median and Standard Deviation of flattened array, of each column, and each row .mean .median .std

(b) For the array: a = np.array([-1.8, -1.6, -0.5, 0.5,1.6, 1.8, 3.0]). Find floor, ceiling and truncated value, rounded values

floor_a = np.floor(a)
ceiling_a = np.ceil(a)
truncated_a = np.trunc(a)
rounded_a = np.round(a)

Q3: Based on Searching and Sorting

(a) For the array: array = np.array([10, 52, 62, 16, 16, 54, 453]), find

i. Sorted array .sort(array)
ii. Indices of sorted array .argsort(array)
iii. 4 smallest elements .partition(array, 4)[:4]
iv. 5 largest elements .partition(array, -5)[-5:]

(b) For the array: array = np.array([1.0, 1.2, 2.2, 2.0, 3.0, 2.0]), find

i. Integer elements only
ii. Float elements only

```
# i. Integer elements only
integer_elements = array_b[np.mod(array_b, 1) == 0]
# ii. Float elements only
float_elements = array_b[np.mod(array_b, 1) != 0]
```

Q4:

(a) Write a function named img_to_array(path) that reads an image from a specified *path* and save it as text file on local machine? (Note: use separate cases for RGB and Grey Scale images)

(b) Load the saved file into jupyter notebook?