M1.B4: Programming Assignment on NumPy Basics and Matrix Manipulations

New Attempt

Due May 30 by 10:59pm   **Points** 100   **Submitting** a text entry box or a file upload
Available after May 23 at 11pm

This assignment is intended to help you practice with some basic routines in NumPy and prepare you to manipulate image and tabular data.

Instructions:

- Create an iPython notebook to work on the following problems. [How To link: **https://www.dataquest.io/blog/jupyter-notebook-tutorial/** **(https://www.dataquest.io/blog/jupyter-notebook-tutorial/)**]
- Clearly put the problem numbers in appropriate headers and subheaders on the notebook.
- Do not display information that is not being sought.
- Images or data files, if any, should be kept in folders such as   "./images/im_name.jpg" or "./data/data_file.cvs."

PROBLEMS

1. Write a function trace1D() that finds the trace of a matrix given by a one dimensional NumPy array. You can not use the built-in function np.trace() or any other builtin function in any way. **Display the output** of print("Trace of the suggested matrix is: {}".format(trace1D(np.arange(0,100,1))))

2. Read an appropriate RGB-image of your choice in a 3-D Tensor named myRGB. Perform the following operations on this tensor: (A) Pad the image by 5 pixels on all sides. This operation is akin to putting a dark frame around the image. Display this image. (B) Use a sliding window of a 3x3 matrix, K (referred to as a kernel), to perform an operation called convolution on the original image. Display some of these images after convolution. Note that you only need to use appropriate slicing of the source image, element wise product, and np.dot in loops to perform this. Display the images after convolutions using K=[1 0 -1;0 0 0;-1 0 1] and K = [0 -1 0; -1 4 -1; 0 -1 0]. Indicate what these convolutions have achieved. To know more about convolution and how to achieve it, read Section 9.2 in **https://www.deeplearningbook.org/contents/convnets.html** **(https://www.deeplearningbook.org/contents/convnets.html)** .

3. IRIS flower dataset is one of the widely used resources. Load the IRIS data by using:
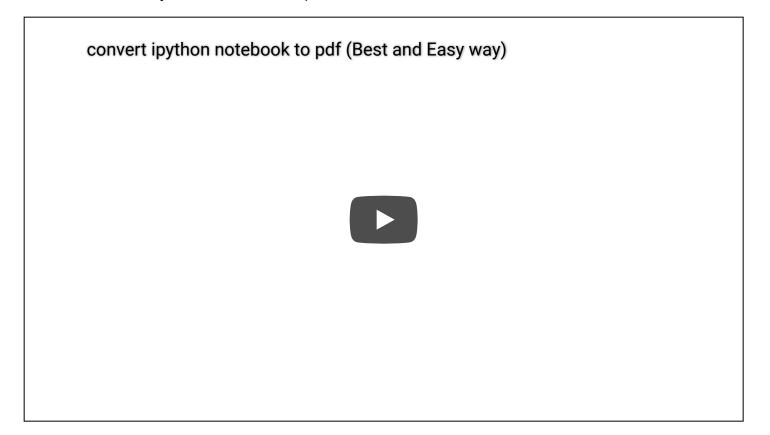
        # Python code

         from sklearn import datasets

iris = datasets.load_iris().data

Normalize this data by using z-scoring ($z = (x - x\_mean)/std$). Don't use loops. You can use np.mean(), np.std() and basic matrix operation with broadcasting. Visualize the distribution of this data using Matplotlib or other packages.

In your submission, you must:

- Provide the console output of the routines that you have implemented in the text input box in the format as specified in the problems.
- Provide a pdf file of the iPython notebook of your code. Check the following link for a convenient way to convert an iPython notebook to a pdf document:

convert ipython notebook to pdf (Best and Easy way)

Programming Assignment Grading Rubric

Criteria	Ratings				Pts
Completeness and Accuracy Programming output is complete and accurate.	**100 pts** **Exceeds Expectations** Work is complete and accurate.	**80 pts** **Somewhat Meets Expectations** Output is not as expected but most of the necessary steps were taken in programming; minor errors.	**60 pts** **Below Expectations** Work is only partially complete and/or contains many errors.	**0 pts** **No Points** Assignment not submitted.	100 pts
				Total Points: 100	