ROAR ACADEMY EXERCISES

Lecture 11. Files

1. WRITE A MOTTO

Please program a code to perform the following steps:

1. Create a file named **motto.txt**, write the first line as:

   Fiat Lux!

2. Properly close the file, then reopen it to print out its content.

3. Append the file with the second line:

   Let there be light!

2. IMAGE MANIPULATION

Use the image class in matplotlib module to load the original lenna image (512-by-512 color image). Then load another image of your national flag, such as from:

https://www.countryflags.com/

Please code a program that replaces pixels values from the top right corner of the lenna image with those from the full color image of the national flag image, and display the modified lenna image. Note that after the modification, the bigger image of lenna must include the full image of the smaller national flag image.

ROAR ACADEMY EXERCISES

Lecture 11. Files

3. SENSE AND SENSIBILITY

Jane Austen's novel *Sense and Sensibility* is freely available online at: https://www.fulltextarchive.com/pdfs/Sense-and-Sensibility-by-Jane-Austen.pdf. Below we provide some sample code to read one-page text from a pdf file:

```
pip3 install PyPDF2 —user
>>> import PyPDF2
>>> file_handle = open('Sense-and-Sensibility-by-Jane-Austen.pdf', 'rb')
>>> pdfReader = PyPDF2.PdfFileReader(file_handle)
>>> page_number = pdfReader.numPages   # this tells you total pages
>>> page_object = pdfReader.getPage(0)    # We just get page 0 as example
>>> page_text = page_object.extractText()   # this is the str type of full page
```

Please solve:

Assemble all the distinct words used by Jane Austen in *Sense and Sensibility* (from all pages and then all the lines in each page) in a dictionary called **frequency_table**. The keys in the dictionary are the distinct words, and the respective values are the number of times the key words were used in the novel.

Please note that in creating this dictionary, we will only document the usage frequency of English words in the novel. For example, we shall ignore some patterns:

(1) Numerical numbers such as 1, 2, 3

(2) Punctuation and space

(3) Headlines such as "CHAPTER 1"

(4) Page numbers such as "page 1 / 559"

Hint: use string.split('\n') to split a text string to lines; use string.split() to split a text string to words.

ROAR ACADEMY EXERCISES

Lecture 12. NumPy

1. ARRAY DOT PRODUCT

Given a numpy array

```
v = np.array([2., 2., 4.])
```

Please write a program that projects the array onto three 3D coordinate axes e0, e1, e2. The expected projection values should be 2., 2., and 4., respectively

2. MATRIX DOT PRODUCT

Please write python code to compute the following matrix scalar product and dot product using numpy:

1. $$2 \cdot \begin{bmatrix} 6 & -9 & 1 \\ 4 & 24 & 8 \end{bmatrix}$$

2. $$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 6 & -9 & 1 \\ 4 & 24 & 8 \end{bmatrix}$$

3. $$\begin{bmatrix} 4 & 3 \\ 3 & 2 \end{bmatrix} \cdot \begin{bmatrix} -2 & 3 \\ 3 & -4 \end{bmatrix}$$

3. SLICING

Define a numpy array **arr** as follows:

0	1	2	3	4	5
10	11	12	13	14	15
20	21	22	23	24	25
30	31	32	33	34	35
40	41	42	43	44	45
50	51	52	53	54	55

Please use slicing methods to print out the data formed in four sub-arrays, as highlighted in Pink, Green, Blue, and Orange colors.

ROAR ACADEMY EXERCISES

Lecture 12. NumPy

4. SWAP MATRIX ROWS AND COLUMNS

Please write two functions **swap_rows(M, a, b)** and **swap_cols(M, a, b)**, where M is a NumPy matrix. swap_rows() swaps the a-th row and b-th row of M, and swap_cols() swaps the a-th column and the b-th column of M.

Note: Please in your code include sufficient argument sanity check to make sure to be able to handle potential corner cases.

5. CREATE A MATRIX ARRAY

Please write a function **set_array(L, rows, cols)**, where L is a list-type variable with rows*cols total number of elements. set_array returns a numpy array of rows-by-cols dimension.

Note: The implementation can pick either row-order or column-order as the default order to fill the array with L elements. Alternatively, one can create a third input argument known as **order**.

ROAR ACADEMY EXERCISES

Lecture 13. Visualization

1. A BETTER CLOCK INTERFACE

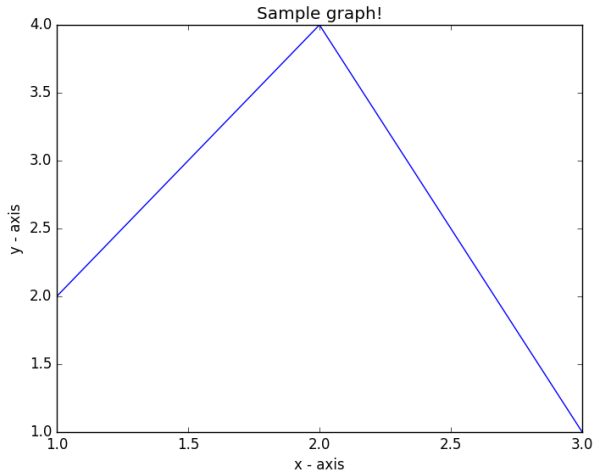
In the Lecture, we discussed rendering a clock interface that display the real-time system clock in the sample code: matplotlib_clock.py. In this exercise, we will attempt to improve the graphics of the system clock.

Please solve:

1. Remove the Axes ('top', 'bottom', 'left', 'right') displayed together with the clock.
2. In the sample code, the hour and minute hands only point to full hour and minute numbers, respectively. In other words, their positions are not updated following the second hand movement, which is not realistic. Improve the code to have the hour and minute hands to update their positions synchronized with the movement of the second hand.
3. Add a fourth clock hand, called the GMT hand. The GMT hand should be rendered in yellow color, and it will point to the hour of the GMT time zone but utilize the clock face to indicate the 24 hours instead of 12 hours as in displaying the local time. Namely, 0 o'clock GMT hour hand will point to the 0 degree , and 12 o'clock GMT hour hand will point to the 180 degree.

2. CREATE THE GRAPH

Please code a program that generates the graph below. Please pay attention to the provide graph title and x-axis and y-axis labels.



ROAR ACADEMY EXERCISES

Lecture 14. Gradient Descent

1. LEARNING RATES

In the Lecture, the sample code gradient_descent.py illustrates the process of gradient descent search for a globally optimal solution that minimizes the penalty function f(). The effectiveness of the search is determined to a large extent by the user-defined learn_rate.

In this exercise, please choose a new initial position aa, and demonstrate the effect of different learn_rate to cause the gradient descent search to behave differently. We are looking for three types of behaviors:

(1) slow convergence;

(2) fast convergence;

(3) oscillation and/or divergence.

ROAR ACADEMY EXERCISES

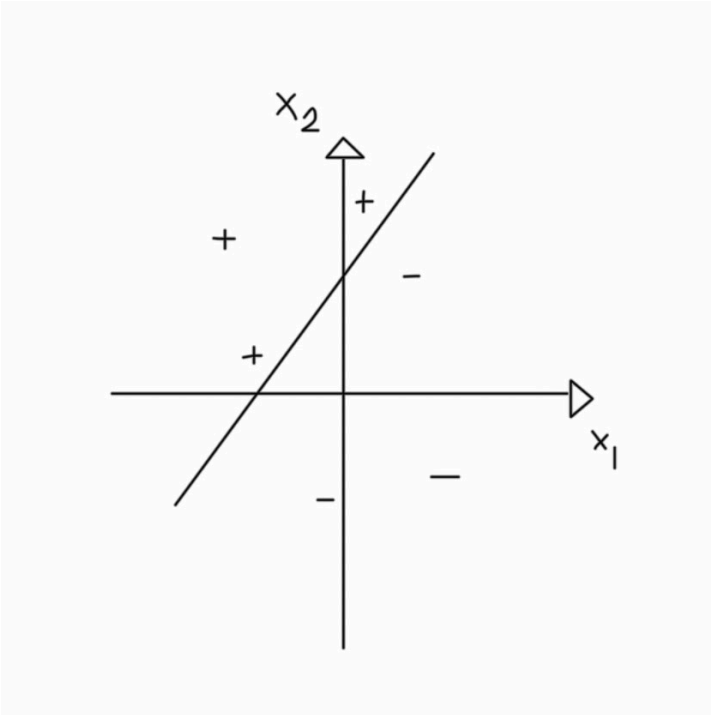
Lecture 15. Perceptron

1. PERCEPTRON WEIGHTS

Recall a single perceptron models the following state function:

$$z = [w_1 \quad w_2 \quad \cdots \quad w_n \quad w_0] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ 1 \end{bmatrix}$$

In the linear separation example below, what can be the values of weights w0, w1, and w2 for the perceptron whose decision surface is illustrated in the figure? **Assume the surface crosses the x1 axis at -2 and the x2 axis at 3.**



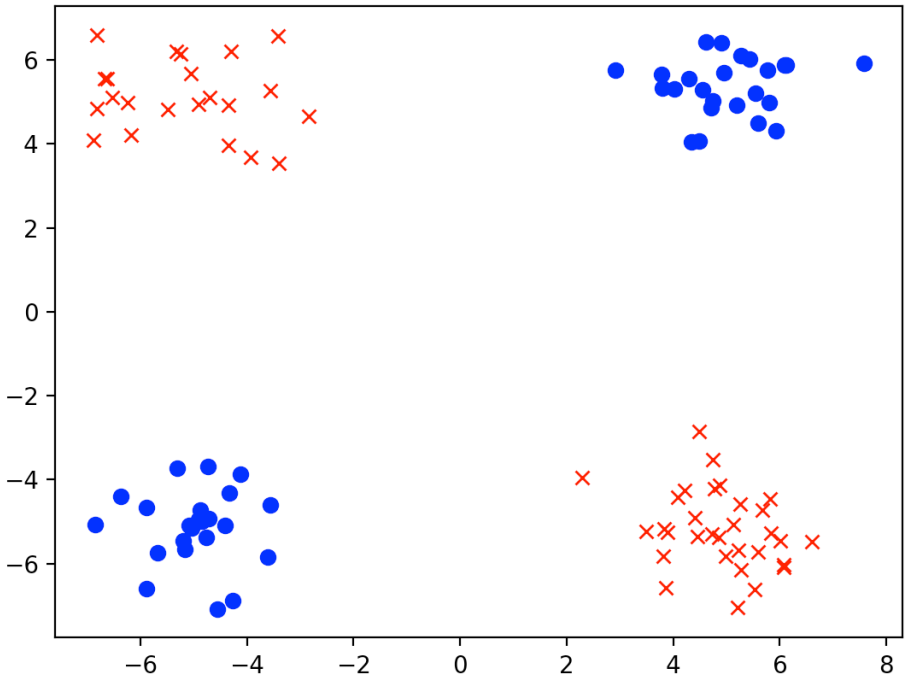
ROAR ACADEMY EXERCISES

Lecture 16. Deep Neural Networks

1. MULTI-LAYER PERCEPTRON FOR LINEARLY NONSEPARABLE DATA

In the MLP sample code, we demonstrated by changing the sigmoid activation function to ReLU, a two-layer MLP model can effectively classify the "XOR" linearly non separable sample points (shown below) with 100% accuracy.

Please propose and train an alternative MLP model, whereby the two class labels are assigned as [1, 0] and [0, 1], and the hidden layers will keep the sigmoid activation, but you may be able to add more perceptrons per layer or add more layers. Please show which new MLP model you may find can effectively classify the sample points with 100% accuracy.



2. SPECIAL XOR CLASSIFIER WITH TWO PERCEPTRONS

For the above problem, if we use ([1, 1], [-1, -1]) to identify the two blue clusters within the first group, and ([1, -1], [-1, 1]) to identify the two red clusters within the second group, then it is possible to implement a one-layer perceptron network with just two perceptrons to train on the output vector [out_1, out_2], and there is no additional hidden layers. Please implement a feasible solution to the above setup of the XOR classification problem.

ROAR ACADEMY EXERCISES

Lecture 16. Deep Neural Networks

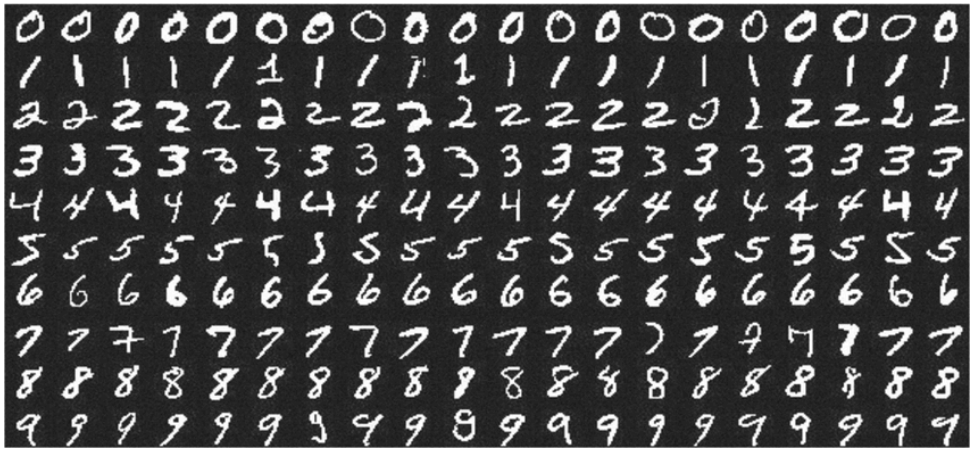
3. MNIST CLASSIFIER

In the Lecture, we introduced the classification problems using a classical hand-written digit database, called MNIST. MNIST is often the first practical image database used in learning neural networks.

In tensorflow and keras, a tutorial is provided to train and classify MNIST database using a simple MLP network. Specifically, with a two-layer ReLU network, MLP achieves 97%-98% accuracy.

Please review the tutorial and code your implementation based on:

https://www.tensorflow.org/datasets/keras_example



ROAR ACADEMY EXERCISES

Lecture 19.
Reinforcement Learning

1. PID CONTROLLER FOR CARTPOLE GAME

In the PID lecture, we have learned about the PID control law for minimizing error of a system state by applying control to update the state. In the exercise, please reference the Cart-Pole DQN implementation, but instead replace the DQN control by a simpler PID control.

The program shall first create a simulation environment using gym.make('CartPole-v1'), then apply action = 0 or action = 1 based on the PID control signal derived from the pole angle, where the ideal pole angle state is zero, indicating the pole remains upright. Therefore, nonzero pole angle is the error that can be compensated by applying action = 0 or action = 1 to always drive the nonzero pole angle towards zero value.