Assignment 1 Introduction to Deep Learning – CAP 4613

Due: Wednesday, Jan 27 before 11:59 pm.

Instructions:

• The entire assignment has to be submitted "electronically on Canvas" before the deadline.

• Each assignment should have the following information on the first page: assignment number, student name and znumber, and a shareable link to the final version of your Python code in Colab.

> In your Colab notebook, click 'Share' on the upper right corner to get a shareable link, then click 'Get shareable link' and copy the link.

• The Python submission should include the codes and the generated outputs.

> To generate the PDF submission file: in Colab, go to 'File'=>'Print', then change the 'Destination' to 'Save as PDF' and save.

• **ONLY one file** has to be submitted electronically. Combine the handwritten and Python parts before submission.

> You can upload the pictures of your handwritten answers to the drive and then import and show them in a Colab notebook using the matplotlib library.

• Filename for electronic submission: Student_Name_Assignmentxx.pdf or doc.

Problem 1) Simple Calculator: In Python, implement a simple calculator that does the following operations: summation, subtraction, multiplication, division, mod, power, exp, natural log, and abs.

a) Follow the instructions below:
- To work with the calculator, the program asks the user to enter the first number, then the operation, and finally the second number if required.
- Your code has to recognize the need for the second number and ask for it if required.
- After performing one operation, the calculator prints the output of the operation.
- After performing one operation, the calculator must not exit. It has to start again and ready for the next operation.
- The calculator will be closed if the user writes 'x' as an input.
- Use functions to perform the operations and the appropriate conditions to prevent common errors such as entering characters as one of the numbers etc.

b) Run your code and provide the results for at least one example per operation.

Problem 2) Threshold-based Classifier – We have a two-class classification problem (i.e., C1 and C2). Each data sample is represented by two attributes (x,y). The three data samples in class C1 are {(2, 2), (3, 2), (2, 3)} and {(1, 2), (1, 1), (2, 1)} in class C2. Perform the followings in Python:

a) Plot the data samples. The data points in classes C1 and C2 have to be in two different colors and shapes. Label the axes and add legends as appropriate.
b) The code asks the user to enter two thresholds th_x and th_y.
c) Your code calculates and prints the training accuracy based on the user-entered thresholds. To do so, assume that for any data point (x,y) with x> th_x and y> th_y, the data sample belongs to class C1, and C2 if otherwise. Using this rule and the user-entered thresholds, the code calculates the classification accuracy for the six data samples. The classification accuracy is defined as the number of correctly classified data points over the total number of data points (6 in here).
d) The code asks the user to enter a new data sample by entering both x and y.
e) Plot the new data sample on the plot in part (a).
f) The code determines and prints the class (C1 or C2) of the new data sample using the user-entered thresholds.
g) The user can enter new data points, and the code repeats parts (d)-(f) unless the user writes 'x' as the input.
h) The code asks the user for new threshold values and starts from part (b)
i) Select the best thresholds that will give the highest accuracy. Run part (c) using these thresholds. What classification accuracy did you achieve? Was it the highest possible accuracy for the given six data samples?

Problem 3) MNIST dataset - The MNIST dataset is divided into two sets - training and test. Each set comprises a series of images (28 x 28-pixel images of handwritten digits) and their respective labels (values from 0 - 9, representing which digit the image corresponds to).

a) Use mnist function in keras.datasets to split the MNIST dataset into the training and testing sets. Print the following: The number of images in each training and testing set, and the image width and height.
b) Write a function (with images of ten digits and labels as the input) that plots a figure with 10 subplots for each 0-9 digits. Each subplot has the number of the handwritten digit in the title.
c) Create a loop to call the plot function in (b) with images from each set to create three figures. Note: the code has to select the images randomly. Include all the 10 digits in each figure. Show the results of your code.
d) In machine learning, we usually divide the training set into two sets of training and validation sets to adjust a machine learning model parameters. In your code, randomly select 20% of the training images and their corresponding labels and name them as x_valid and y_valid, respectively. Name the remaining training images and their labels as x_train and y_train, respectively. Print the number of images in each training and validation set. Note: that there are no overlaps between the two sets.