Assignment 3 Introduction to Deep Learning – CAP 4613

Due: Sunday, February 20

Instructions [Failure to follow instructions will be penalized at least by 1 point]

- All the solutions must be based on the methods taught in class. Otherwise, it will not be graded. This is an important note to the students who do not study the lectures and learn from other online resources.

- The Colab link must be shared.

- Make sure that it is possible to copy the Colab link from your pdf.

- Make sure that your pdf is not cutting the text.

- Provide the full answers in your pdf file with all the outputs of your code.

- Only one file must be submitted electronically.

- Combine the handwritten and Python parts before submission.

- The paper-based problems must be with no coding/python etc.

- The paper-based problems/parts can be typed, but if handwritten, it must be clean and legible.

- Show your detailed work to each problem.

Problem 1) [Paper-based] Perceptron learning: Consider the following set of data points:

input		desired
x_1	x_2	label
1	1	1
1	0	1
0	1	0
-1	-1	0
-1	0	0
-1	1	0

As the above table shows, the data points are categorized (labeled) in two groups specified by the labels "1" and "0".

a) Use the perceptron learning rule to train a single neuron perceptron on the data points given above. Show all the steps in the iteration.
Assume the learning rate is 1 (i.e., $\eta = 1$) and use the hard-limiter activation function (If $v \geq 0$ output 1; otherwise output 0), i.e.:

$$\varphi(v)=\begin{cases}1 & if\ v\geq 0\\ 0 & if\ v<0\end{cases}$$

HINT: If you start with a weight vector of (0,0,0), you must complete 3 complete iterations until the perceptron learning converges.

b) Draw the schematics of the learned perceptron labeling the learned weights.
c) Provide the equation of the trained model (i.e., the classifier line).
d) Plot the given data points with two different markers for each group.
e) Plot the classifier line to the plot in (d).
f) Use the trained perceptron and classify the test data points given in the table below. Show your work for each data point. Provide the predicted labels (0 or 1) in a table like below.

input x_1	input x_2	desired label	local field	predicted label
2	0	1		
2	1	0		
0	0	1		
-2	0	0		

g) Provide the confusion matrix for part (f) and calculate the accuracy. Consider Class "1" as the positive class.

Problem 2) [Python] Perceptron learning in Python:

a) Create class NeuralNetwork(): that creates a single neuron, train it, and test it. This class should have the following function:
   i. def __init__(self): that initializes a 3x1 weight vector randomly and initializes the learning rate to 1.
   ii. def hard_limiter(self, x): that performs the hard-limiter activation on the nx1 vector x.
   iii. def forward_propagation(self, inputs): that performs the forward propagation by multiplying the inputs by the neuron weights and passing the output through the hard_limiter activation function.
   iv. def train(self, inputs, labels, num_train_iterations=10): that performs the perceptron learning rule for num_train_iterations times using the inputs and labels.
   v. def pred(self,inputs): classifies the inputs to either class 0 or 1 by multiplying them by the neuron weights, passing the output through the hard_limiter activation function and thresholding.
b) Use the perceptron learning rule to train a single neuron on the data points given in problem 1:

i. Create an np array of the shape of 6x2 that contains the inputs, and another array with the shape of 6x1 that contains the labels.
ii. Add the bias to the input array to have a 6x3 shape.
iii. Create the network with one perceptron using the class NeuralNetwork(), then train it using train(inputs, labels,100) function.

c) Plot the given data points with two different markers for each group.
d) Use the trained perceptron weight and plot the classifier line in the same plot in (c).
e) Use the trained perceptron and classify the test data samples given in the table below by calling the pred() function.

input		desired label	predicted label
x_1	x_2		
2	0	1	
2	1	0	
0	0	1	
-2	0	0	