**Comp 337 Assignment 1 Report**

Student Name:

Student ID:

1. **Perceptron Algorithm Pseudocode**
2. **Perceptron( Data, n, MaxInter ):**
3. b = 0
4. **for** all weight w :

w(i) = 0

1. **end for**
2. **for** iter = 1 … MaxInter do:
3. **for** all Data:( 1….t :Input & Output ) **do:**
4. y(t) = sign[ w(t)T \* Data:Input(t) + b ]
5. **If** y(t) \* Data:TargetOutput(t) <= 0 **do**:
6. w(t + 1) = w(t) + n \* Data:TargetOutput(t) \* Data:Input(t)
7. b = b + Data:TargetOutput(t)
8. **end if**
9. **end for**
10. **end for**
11. **return** b, w(0,1,2,….)

**Explanation:**

**Line 1 :** The perceptron algorithm that takes 3 parameters Data (containing data in the form input – output), n (learning rate), and MaxInter (max iteration times).

**Line 2 – Line 4** : Initialization stage, initialize the parameters in the network to zero, that includes the bias b and the weights w.

**Line 5**: Training the network MaxInter times, starting from time episode 1.

**Line 6**: In each time episode, train the network with all input & output data within the dataset.

**Line 7**: Compute the output of the network with input data t. output = 1 if sum of product of weights and inputs is

**Line 8**: If the sing of expected output y(t) and our actual output Data:TargetOutput(t) is not the same, i.e. y(t) \* Data:TargetOutput(t) <= 0, update the weights and bias.

**Line 9 - 10:** Updating parameters in the network.

**Line 14:** Return the trained network parameters.

1. **Binary Implementation**

Please see attached python file called “perceptron.py”, the detailed running instruction is given in “README” as well.

1. **Accuracy Report**
2. Training dataset ( After 20 Iterations, learning rate is set to be 1 )

*Class 1 & 2 : 100 %*

*Class 2 & 3 : 50 %*

*Class 1 & 3 : 100 %*

1. Testing dataset ( After 20 Iterations, learning rate is set to be 1 )

*Class 1 & 2 : 100 %*

*Class 2 & 3 : 50 %*

*Class 1 & 3 : 100 %*

The above data was obtained by the output of the program which can be verified by running the program again. It is shown that the pairs of data consist of class 2 and class 3 are most difficult to separate.

1. **Multi-class Classification**

Please see attached python file called “perceptron\_1vsR.py”, the detailed running instruction is given in “README” as well. Please find in below the accuracy report of 1vR multi-class prediction perceptron.

1. Training dataset ( After 20 Iterations, learning rate is set to be 1 )

*66.6%*

1. Testing dataset ( After 20 Iterations, learning rate is set to be 1 )

*66.6%*

Comment: As expected, since we can tell from the dataset, the class 2 & 3 are not linear separable.

1. **Regularization**

Remines Unimplemented.