**Perceptron Algorithm from Scratch:**

Perceptron Algorithm is a type of neural network which consists of single neuron that iterates through rows to predict its respective dependent variable. It is a two-class binary classification machine algorithm. As Perceptron is a linear classification algorithm it separates the objects in two separate classes using a decision boundary. Multiple inputs are fed to the algorithm along with the bias. The weighted sum of these inputs is then calculated for the Activation.

**Activation = Weights \* Inputs + Bias**

If the Activation score is greater than 0, the algorithm will classify the label positive or 1 else negative or -1.

**Activation > 0.0 (Classify as 1)**

**Activation <= (Classify as -1)**

**Pseudo code:**

Perceptron\_train(D, iterations)

wd = 0 for all d=1..D //initialise weights

bias = 0 //initialise bias

for iterations = 1…iterations do

for all (x,y) belongs to D do

a = wd + xd + bias // Activation

if ya <= 0 then

wd = wd + y\*xd for all d = 1.. D // Update weights

bias = bias + y //Update bias

end if

end for

end for

return w0, w1,…..wd, bias

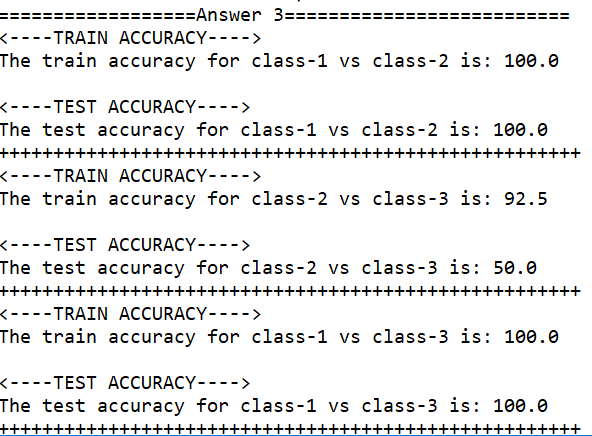
Perceptron\_test(w0,….,wd, bias, xtest)

a = wd + xtest + bias //compute activation

return a

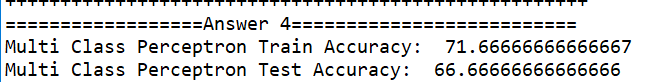
Binary perceptron to train classifiers to discriminate between  
• class 1 and class 2,  
• class 2 and class 3

• class 1 and class 3.



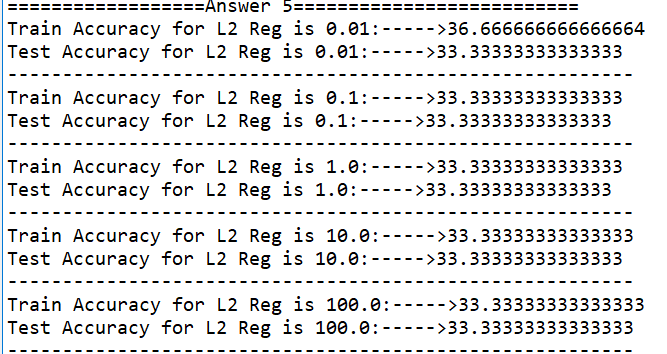
From the above classification after 20 iterations it can be deduced that class 1 vs class 2 and class 1 vs class 3 are linearly separable while class 2 vs class 3 are most difficult to separate.

Multi-class Perceptron:



The Multiclass algorithm was run for the following combination class 1 vs class 2 and class3, class 2 vs class 1 and class3, class 3 vs class 1 and class 2. Predictions are compared to find highest values for classification.

Multi-class Perceptron with L2 regularisation:



Multiclass classifier is trained with coefficients 0.01, 0.1, 1.0, 10.0, 100.0. The train and test classification accuracies remained the same throughout the 20 iterations.