Experiment No 8: Perceptron learning using python

Name: Mushrifah Hasan

Roll no: CS116006

Date: 2nd April 2019

**CODE:**

# Make a prediction with weights

def predict(row, weights):

activation = weights[0]

for i in range(len(row)-1):

activation += weights[i + 1] \* row[i]

return 1.0 if activation >= 0.0 else 0.0

# Estimate Perceptron weights using stochastic gradient descent

def train\_weights(train, l\_rate, n\_epoch):

weights = [0.0 for i in range(len(train[0]))]

for epoch in range(n\_epoch):

sum\_error = 0.0

for row in train:

prediction = predict(row, weights)

error = row[-1] - prediction

sum\_error += error\*\*2

weights[0] = weights[0] + l\_rate \* error

for i in range(len(row)-1):

weights[i + 1] = weights[i + 1] + l\_rate \* error \* row[i]

print('>epoch=%d, lrate=%.3f, error=%.3f' % (epoch, l\_rate, sum\_error))

return weights

# Calculate weights

dataset = [[2.7810836,2.550537003,0],

[1.465489372,2.362125076,0],

[3.396561688,4.400293529,0],

[1.38807019,1.850220317,0],

[3.06407232,3.005305973,0],

[7.627531214,2.759262235,1],

[5.332441248,2.088626775,1],

[6.922596716,1.77106367,1],

[8.675418651,-0.242068655,1],

[7.673756466,3.508563011,1]]

l\_rate = 0.1

n\_epoch = 3

weights = train\_weights(dataset, l\_rate, n\_epoch)

print(weights)

**OUTPUT:**

