**Artificial Intelligence**
Assignment# 02

Implementation:

I have implemented this assignment using both Radial Basis Function and Sigmoid Function.
Sigmoid gives Accuracy much better than RBF. The following Tables Below shows accuracy
and time of execution at different learning rates in both the implementations:

Learning Rate	ExecutionTime (RBF) (sec)	ExecutionTime (Sigmoid) (sec)	Accuracy (RBF)	Accuracy (Sigmoid) (%)
0.03	20.234	25.137	49.1	90.9
0.07	20.312	25.572	47.2	91.56
0.1	21.913	27.363	53.2	91.16
0.2	20.180	26.863	55.4	89.9
0.3	19.987	27.530	57	87.7

Since, Sigmoid is giving better accuracy, I have used Sigmoid and will explain that in later part.

Following graph plots the execution time versus learning rate graph.
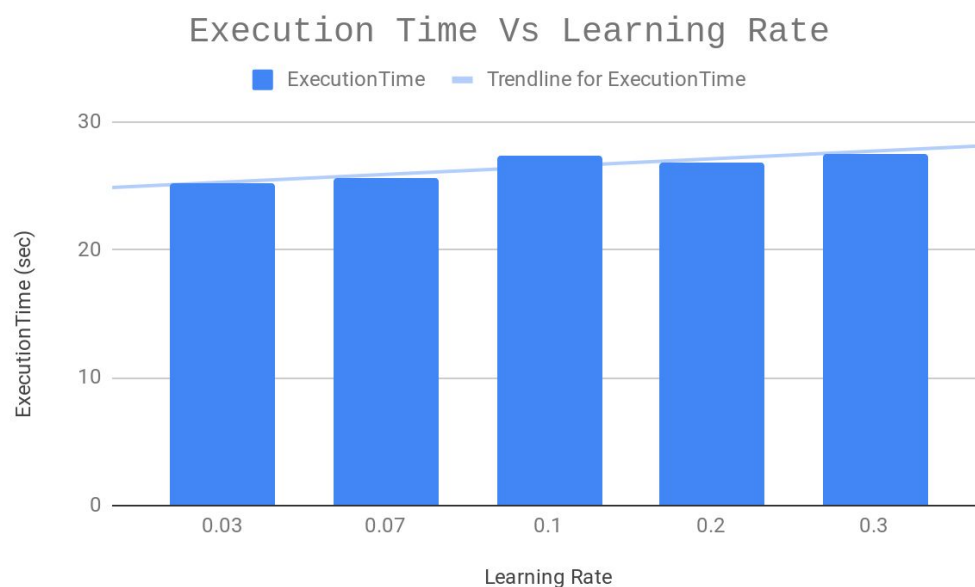
Techniques

1. First, all the data points are normalized because otherwise they data points were spread from 0-255. Normalizing makes the data points more meaningful.
2. Initial weights are chosen random that are uniformly distributed over -1 and 1.
3. Learning Rate is 0.07 for my final weights because this gives maximum accuracy.
4. Back Propagation is used for Error propagation.
5. In testing phase, I have normalized data points because this does not matter as we have already trained and now we just have to use weights.
6. Accuracy can be improved by increasing number of epochs, but this will be at the cost of time.

How to Run:

I have used python2.7 following libraries:
Numpy, sys

The command to train is:
    pyhton Network.py train <trainingData> <Labels> <LearningRate>
The command to test is:
    Python Network.py test <testData> <labels> netWeights.txt