**Q1]**

* **System Specification:**

**Epoch = 100 of 4000 training and 1000 test records.**

**η = 0.0000001**

**momentum= 0.08**

**initial weights = randomly between -1 to 1**

**hidden layers= 1**

**hidden layer neurons= 100**

**activation function= Sigmoid**

**Testing threshold = 1 for s > 0.75**

1. **for s < 0.30**

**Test output= Winner take all**

**Training stopping criteria = Error saturation.**

**Lowest Error (training)= 0.96**

I tried for various η and Epoch.

* For η = 0.01 ,momentum= 0.8 and Epoch = 400
* For η = 0.001 , momentum= 0.5 and Epoch = 500
* For η = ,0.0001 momentum= 0.5 ,and Epoch = 200
* For η = 0.00001 momentum= 0.08 and Epoch = 100
* Activation function considered: Sigmoid, ReLU, Tanh
* Weights considered: 0 to 0.9, -inf to inf range, Xavier

Various learning rate were implemented along with the momentum to find optimal output. The lower learning rate was chosen as it did not oscillate the error fractions. Also, momentum values did seem to push the error fractions to higher end and so a lower momentum was chosen. Activation function was chosen as sigmoid after trials on tanh and ReLU. Various weights were used including -inf to inf range and Xavier for tanh activation.

* **Results**:

1. Confusion matrix on test set: Correct identification of numbers 0 to 9(row and column) in form of matrix. Diagonal representing the correct classification.

A picture containing calendar

Description automatically generated

*Fig 1.1 Confusion matrix on test set.*

1. Confusion matrix on training set:

Calendar

Description automatically generated

*Fig 1.2 Confusion matrix on training set.*

1. Graph denoting error fraction: Error fraction on train and set run for 100 epochs. Every 10th epoch is denoted by red marker.

Chart, line chart

Description automatically generated

*Fig 1.3 Graph of error fraction of test and training set on same epoch.*

* **Analysis of Result:**

The neural network was observed to work best with 1 hidden layer of 100 neurons. The activation function used was sigmoid function with initial random initialization. The weights and activation function had high dependency and as I finalized sigmoid activation function, I used the weights mentioned in system specification. The test and training error before training was very high and constant near to 100%. The weights were updated at every record. The training data translated into 0 1 with H/L values of 0.75/30 respectively. While the test set was evaluated with winner take all and thus the testing error is less than the train error. Confusion matrices show pretty well classification for both sets of data.