PROJECT 2

Name: Eshwar Malalakere Renukaradhya

Student ID: 101645866

## INTRODUCTION:

In this project we are making use of a multilayer perceptron model to detect the handwritten digits of the MNIST dataset. MNIST database is used to train and test the model. It contains 10000 testing images and 60000 training images.

The MNIST database is in the CSV format. This database consists of 28 x 28 images (N = 784 elements per image).

## METHODOLOGY:

MLP has three layers, they are input layer, hidden layer and output layer. Each layer has a row of neurons. Input layer is the first layer which takes the input from the given dataset. It just passes the input to the next layer. The next layer is called as Hidden layer because it is not in direct contact with the given input. Here we are making use of only one hidden layer. The last layer is the output layer and it outputs the computed value or vector.

Scaling is applied to the image pixel data. It is rescaled to the range between 0 and 1(Normalization). Sigmoid function is used as the activation function. In this project '**Stochastic Gradient Descent**' method is used as the training algorithm. One row of data/pattern is fed to the input layer at a time. The network processes the input, activates the neurons and gives the output. This is known as **forward pass**. The output is compared with the desired output and the error is calculated. The error is **propagated back** through the network and the weights are updated according to their contribution to the error. This process is repeated for all the patterns.

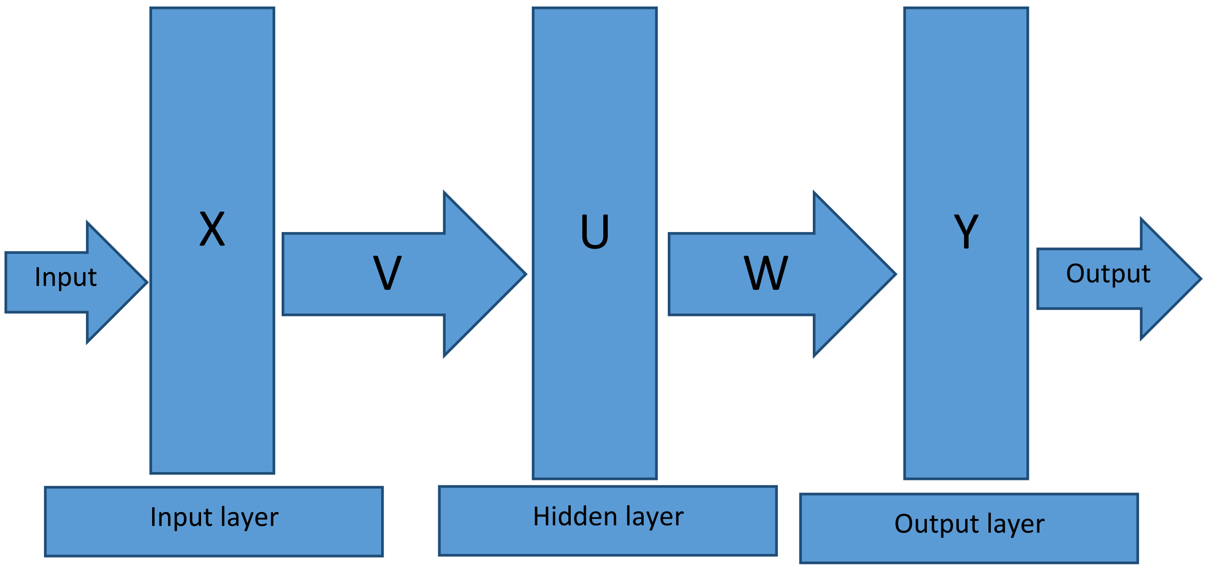



Figure 1. Layers of Multilayer Perceptron

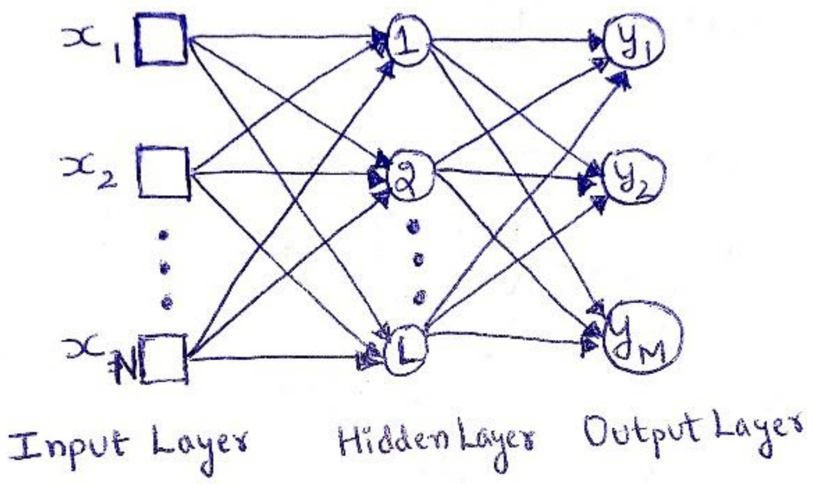



Figure 2. MLP network

**IMPORTANT EQUATIONS:**

<u>Weight change of output Layer:</u>

$$\Delta W_{jk} = \eta \delta_j y_j (1-y_j) u_k$$

<u>Weight change of hidden Layer:</u>

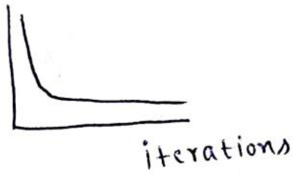
$$\Delta V_{ki} = \eta \delta_k u_k (1-u_k) x_i$$

<u>Back propagation:</u>

$$\delta_k = \sum_{j=1}^{M} w_{jk} \delta_j \qquad k = 1, 2, \ldots L$$

$$W_{jk}(t+1) = w_{jk}(t) + \Delta W_{jk}(t)$$

$$V_{ki}(t+1) = V_{ki}(t) + \Delta V_{ki}(t) \qquad \forall j, k, i$$

<u>MSE:</u> $E = \frac{1}{2} \sum_{s=1}^{P} \sum_{j=1}^{M} (d_j - y_j)^2$ $\quad E =$ 

**PROGRAM OUTLINE:**

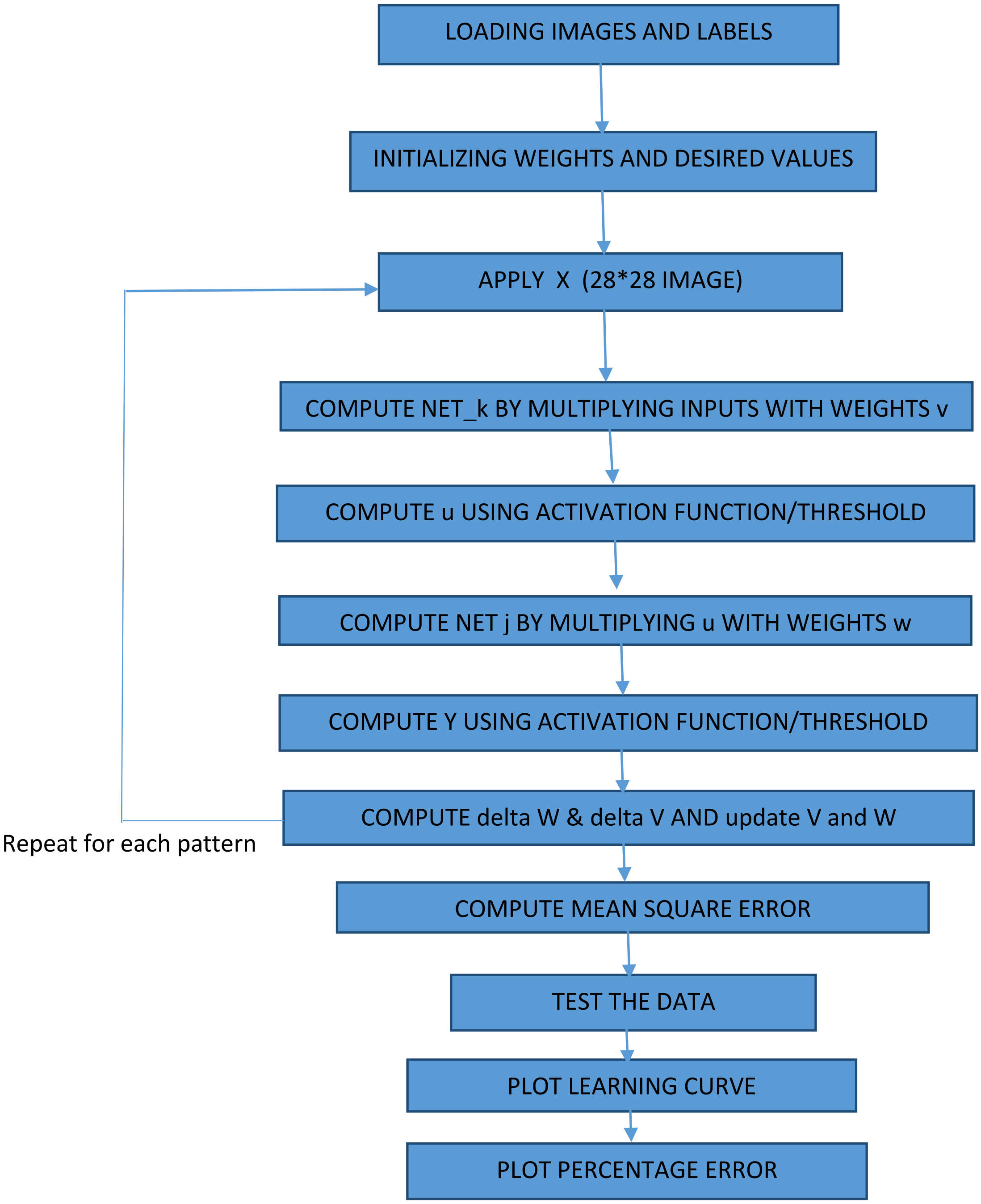



Figure 3. Program flow chart

**RESULTS:**

**Task 1**

**Different learning parameters – 1000 training images :**

Learning curve that illustrates MSE vs iterations is plotted for different learning rate parameters. 1000 images are used for training and 300 images are used for testing. The network is trained for 200 iterations for each learning rate. Three learning rates are used for the training they are 0.001, 0.002 and 0.003. The observation made in this task is that the learning curve is changing abruptly with increase in learning rate.

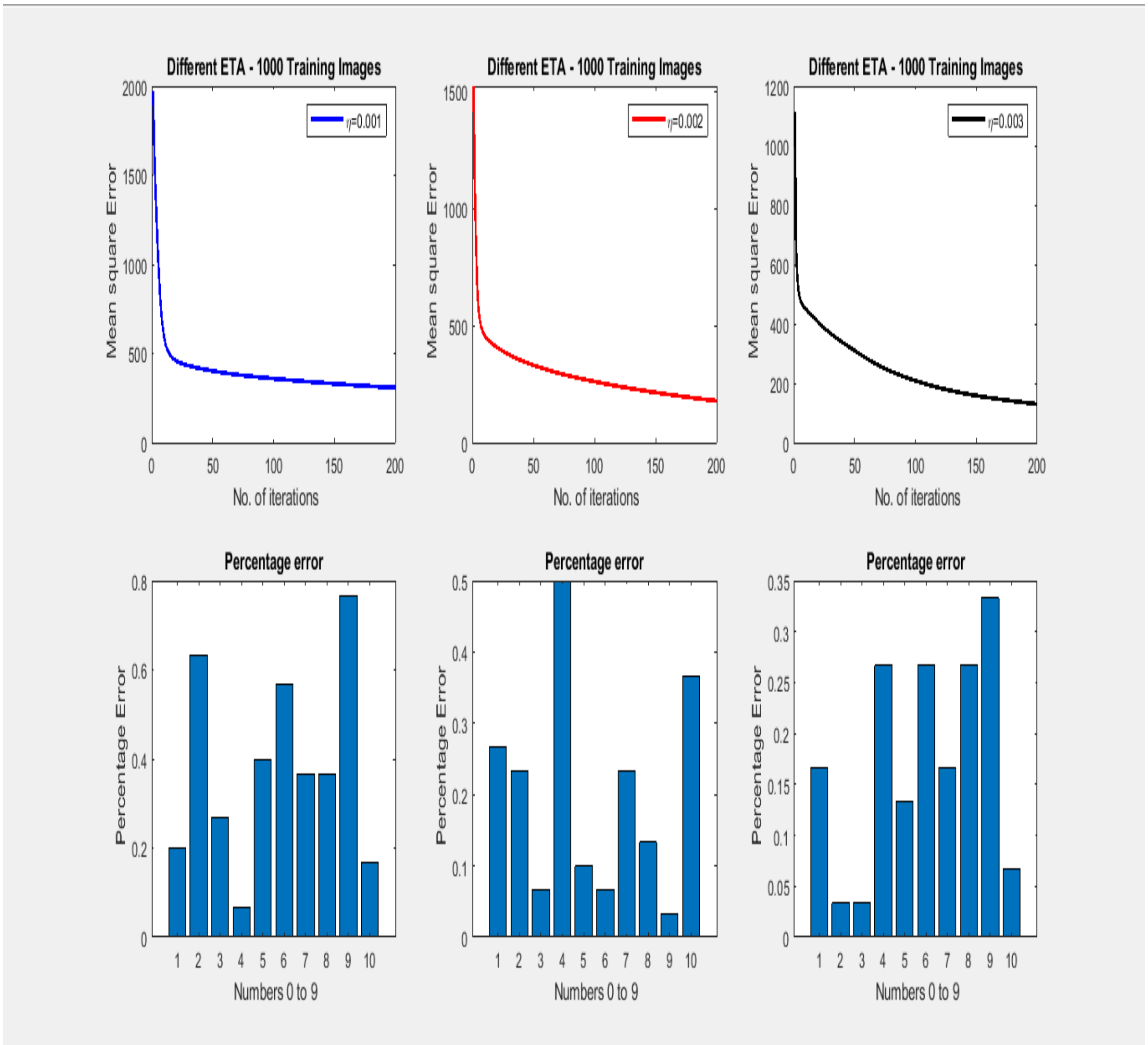



Figure 4. Task_1 output

**Task 2**

**Different L values – 1000 training images :**

Learning curve that illustrates MSE vs iterations is plotted for different L values (number of hidden nodes). 1000 images are used for training and 300 images are used for testing. The network is trained for 200 iterations for each value of L. Three L values are used for the training they are 15, 20 and 30. The observation made in this task is that the decrease of percentage with increase in the value of L.

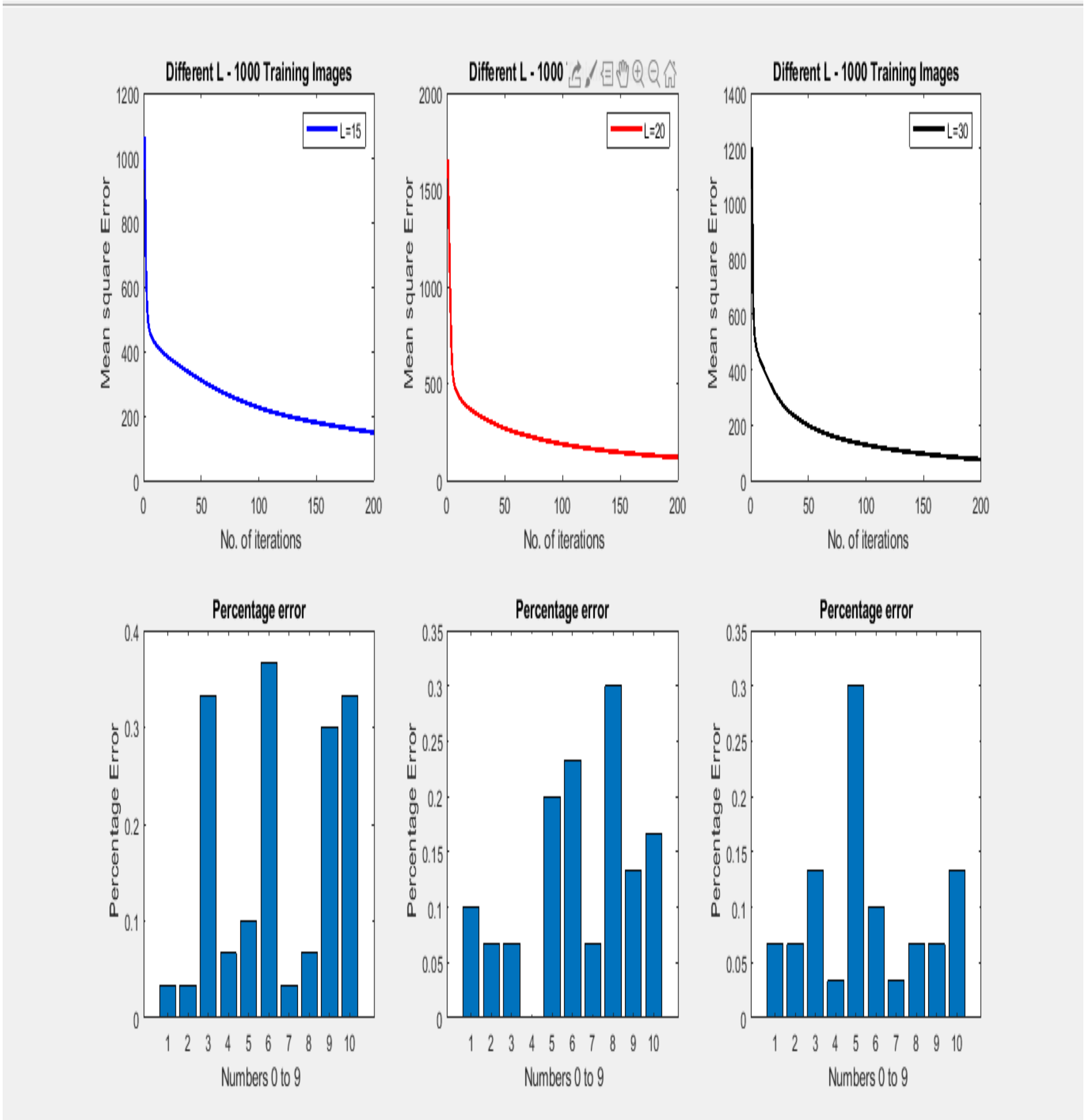



Figure 5. Task_2 output

**Task 3_1**

**Different learning parameters – 3000 training images :**

Learning curve that illustrates MSE vs iterations is plotted for different learning rate parameters. 3000 images are used for training and 1000 images are used for testing. The network is trained for 200 iterations for each learning rate. Three learning rates are used for the training they are 0.001, 0.002 and 0.003. The observation made in this task is that the decrease of percentage with increase in learning rate.

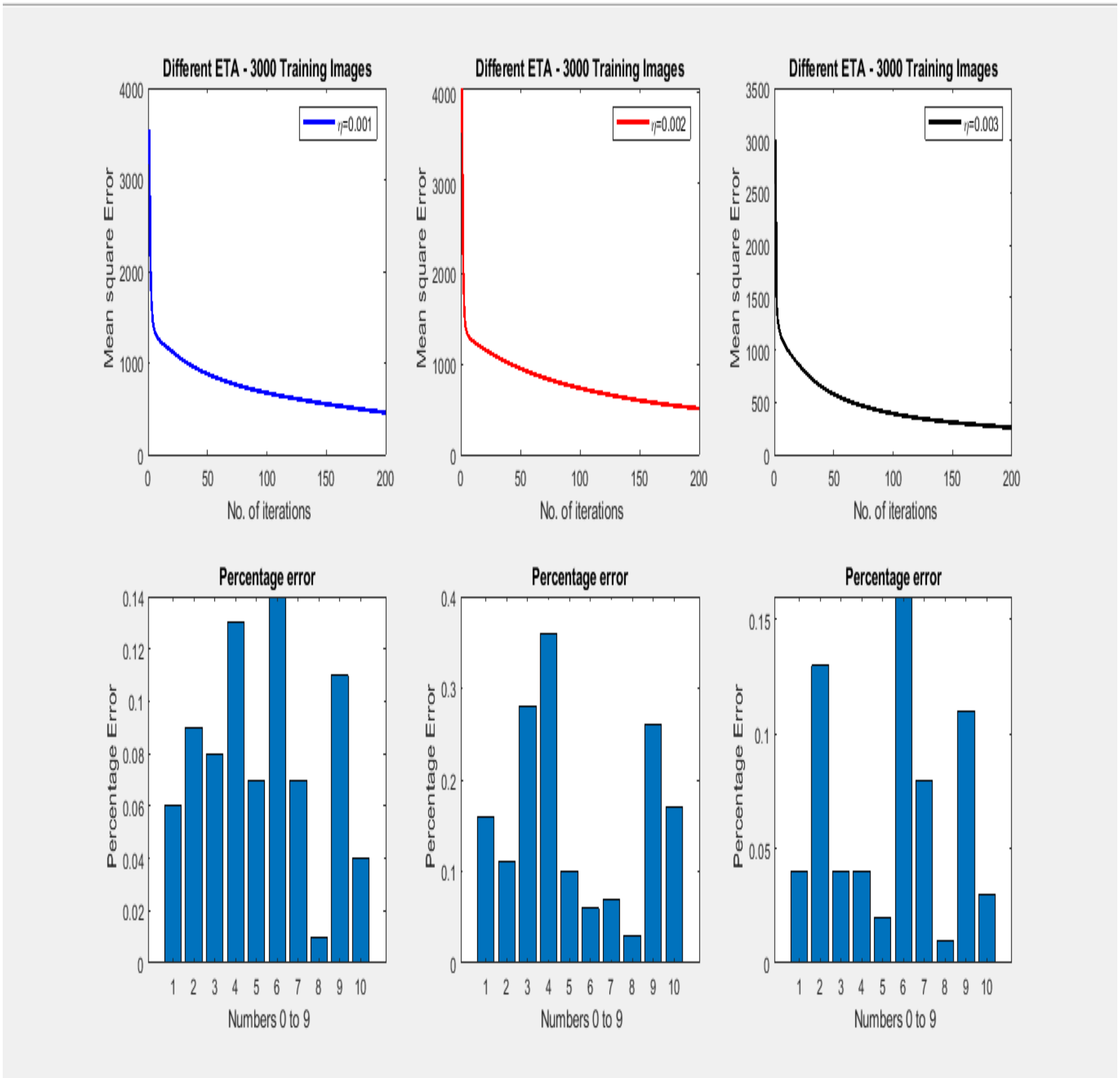



Figure 6. Task_3_1 output

**Task 3_2**

**Different L values – 3000 training images :**

Learning curve that illustrates MSE vs iterations is plotted for different Lvalues(number of hidden nodes). 3000 images are used for training and 1000 images are used for testing. The network is trained for 200 iterations for each value of L. Three L values are used for the training they are 15, 20 and 30. The observation made in this task is that the decrease of percentage with increase in the value of L.

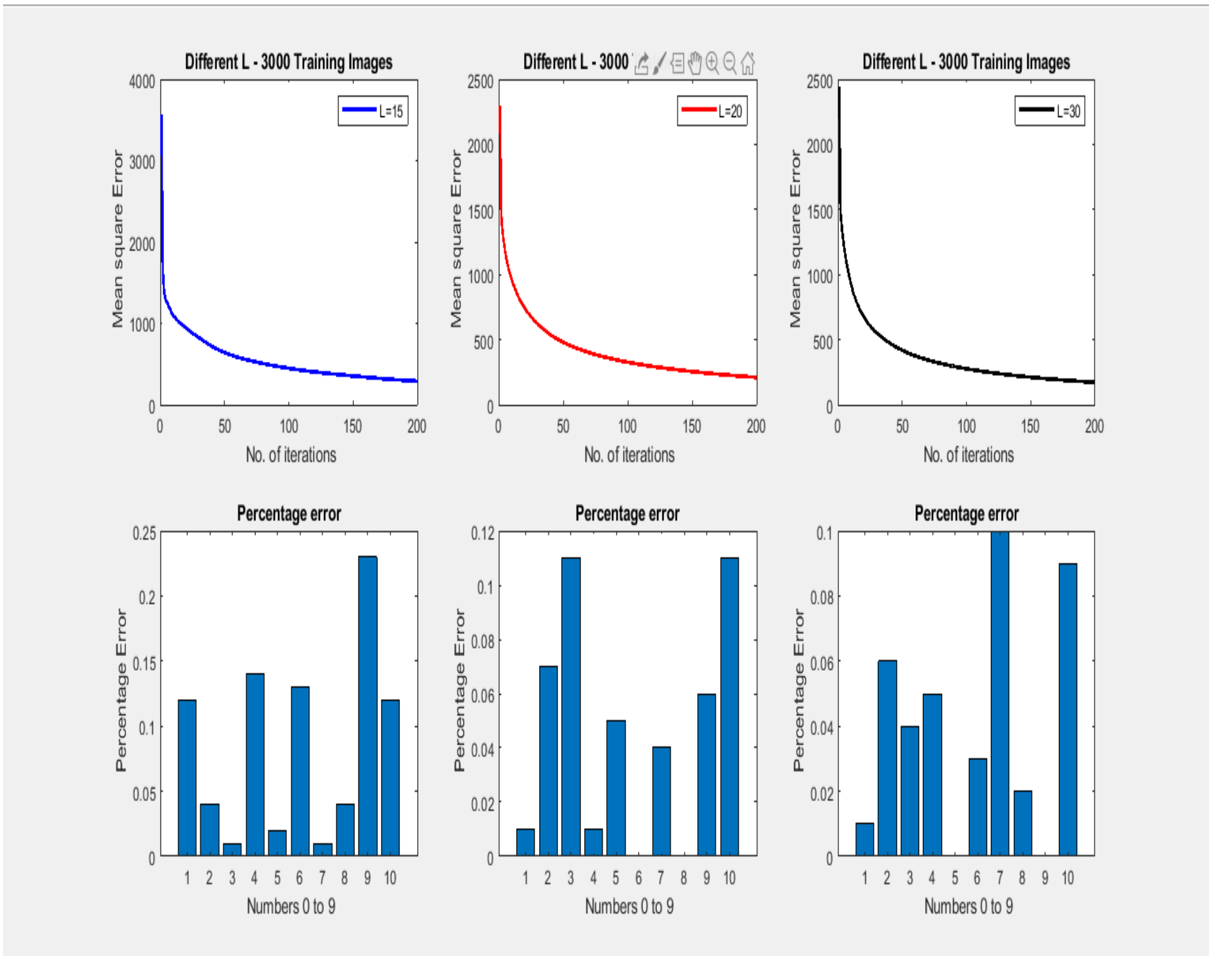



Figure 7. Task_3_1 output

**CONCLUSION**

The multilayer perceptron has been realized and simulated with the help of MNIST dataset using MATLAB. The behaviour of the multilayer perceptron network changes along with learning rate parameter and L (number of hidden nodes).

**REFERENCES**


[1] https://pjreddie.com/projects/mnist-in-csv/

[2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE*, 86(11):2278-2324, November 1998.

[3] https://machinelearningmastery.com/neural-networks-crash-course/