WHAT I HAVE DONE:

- First I implemented the program by basic logic in colab.
- After that I implemented with sequential neural network in pytorch and trained the network with large data other than test input by varying the parameters.
- After training the network I applied test input manually and checked the performance of the model by comparing with the true output(implemented by basic logic).
- After getting descent errors I saved my model in my drive and downloaded from there to laptop.
- I wrote another code to test the saved model by loading it, by taking input through command line instruction.
- I included my basic code in the above mentioned code and shown the output in two text files.

RESULTS:

For 10000 epochs:

	Fixed learning rate		Variable learning rate	
No. of hidden units	Training loss	Test errors	Training loss	Test errors
10	0.9809	45	0.8405	44
100	0.0073	3	0.0007	5
500	0.0029	1	0.0004	0

For 1000 epochs:

	Fixed learning rate		Variable learning rate	
No. of hidden units	Training loss	Test errors	Training loss	Test errors
10	1.0319	45	1.1290	55
100	0.7010	35	0.0210	2
500	0.2164	5	0.0080	2

OBSERVATIONS:

- After finishing my coding part, I started varying parameters of the network.
- I observed that when I am increasing the iterations(epochs) network training well by decreasing training loss and when I checked for test input I am getting less number of errors.
- Similarly by increasing number of layers and weight matrix size(no. of hidden units) training and test errors are decreasing.
- When I changed weight_decay in stochastic gradient descent from 0 to decay_rate(=learning rate/epoch+1) with momentum as 0.8 training and test errors are decreased to very small values.
- One more thing I observed was when I increased epochs converging rate is slow.