21100118

REPORT - PART 1

Approach

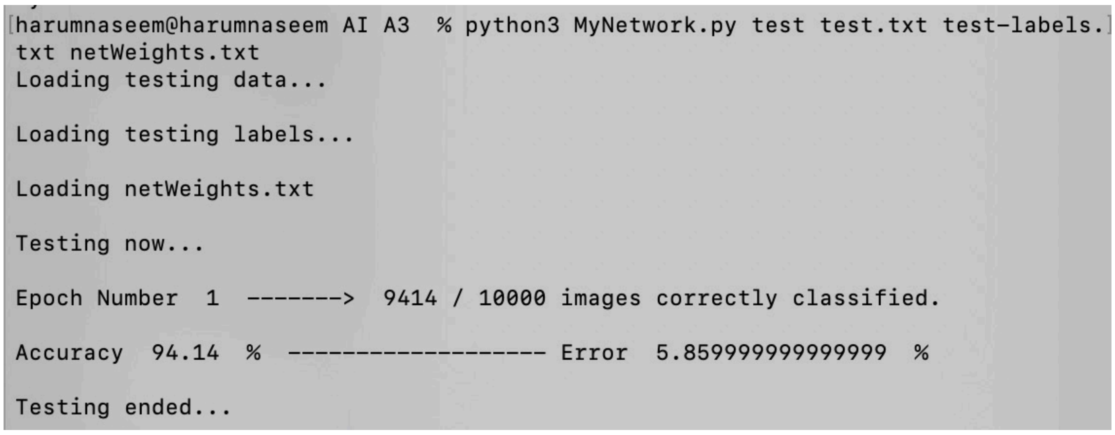
Initially I initialized weights by generating random values and formed two matrices of the dimensions 784x30 (Input to Hidden Layer) and 30x10 (Hidden to Output Layer). Forward propagating (by passing to the sigmoid function) these give us an Output Layer matrix (1x10) which is essentially a prediction. After this, I calculated the error for the weights from hidden to output. These errors are then used to calculate the error value and subtracted to get better end results and to minimize these errors.

The accuracy I got ranged from 92-95% and the error ranges between 8-5%.

Methods used to improve accuracy

- The test and training data were normalized in order to prevent overflow in the sigmoid function.
- In order to prevent overtraining, I calculated the error using the cross-entropy method. If the error computed is less than 0.0000002, no more training is done.
- As mentioned before, the error was multiplied with learning rate and the weights are updated.

Output



```
harumnaseem@harumnaseem AI A3  % python3 MyNetwork.py test test.txt test-labels.
txt netWeights.txt
Loading testing data...

Loading testing labels...

Loading netWeights.txt

Testing now...

Epoch Number  1  ------->  9414 / 10000 images correctly classified.

Accuracy  94.14  %  ------------------- Error  5.859999999999999  %

Testing ended...
```