Homework #3

I have found below observations using multicategory Perceptron Training Algorithm (PTA) for digital classification

a)

Downloading the MNIST data from http://yann.lecun.com/exdb/mnist/

b)

Training set images are stored in Xtrain
Training set labels are stored in ytrain
Test set images are stored in Xtest
Test set labels are stored in ytest

c)

As per the input
Size of Xtrain is [60000,784]
Size of ytrain is [60000,1]
Size of Xtest is [10000,784]
Size of ytest is [10000,1]

So the size of w is [10,784]

d)

Uniformly randomly assigning values to w[10,784] using *np.array(np.random.uniform(-1,1,size = [10,784]))*

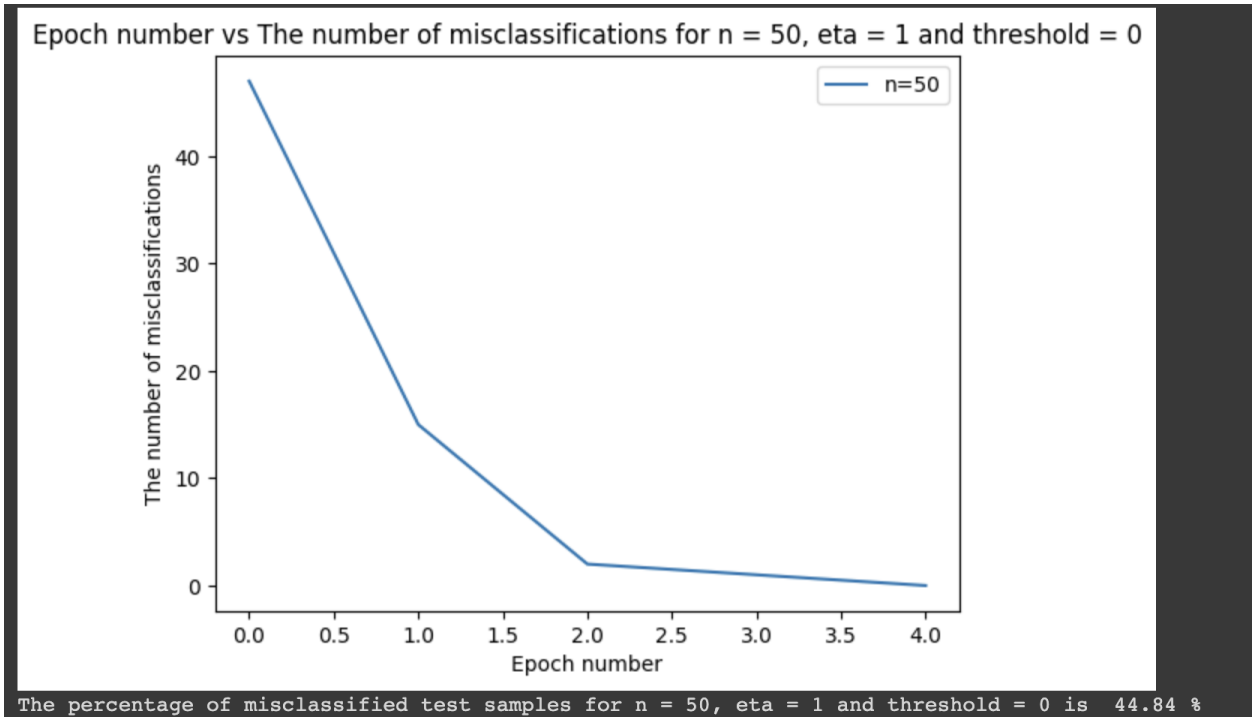
For training set data, **perceptron_training_algo(eta,threshold,n,epoch,w,Xtrain,ytrain,limit)** function is used to update the weights
Calculating the misclassifications for each epoch after updating the weights. If $error/n <= threshold$ then the loop of updating the weights is broken.

e)

For checking errors in testing set data, **applying_perceptron_training_algo(w, Xtest, ytest)** function is used.
Calculating the errors obtained by using the weights given by **perceptron_training_algo**

f)

For n = 50 (taking only 1st 50 training samples), η = 1 and ϵ = 0, we can see from the below graph that after 4 epochs the number of misclassifications converged to 0.



Here, test samples size is 10000 (all test samples are taken)

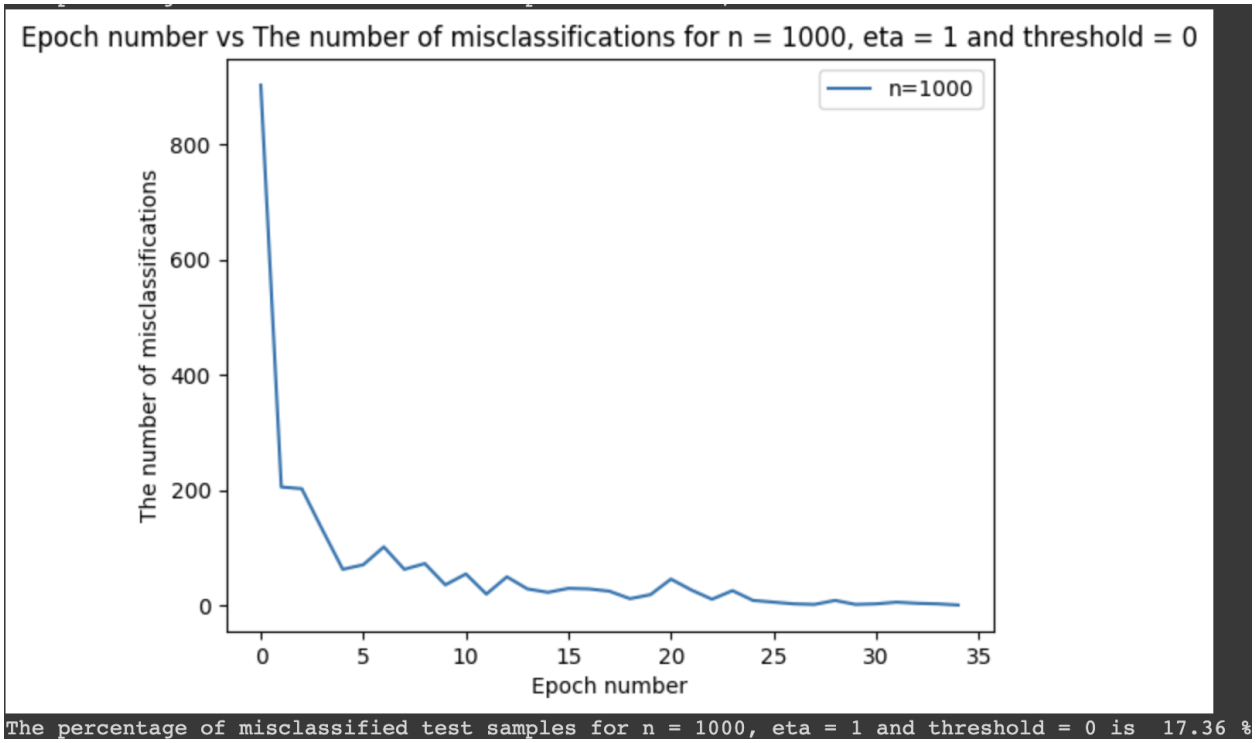
So, after updating the weights till the number of misclassifications converged to 0 for training set, the error rate on test samples was 44.84%

Error rate on training samples = 0% (as at end the number of misclassifications is 0)
Error rate on test samples = 44.84%

As the training set is small (50 samples) compared to the test set (10,000), it does not represent the full range of variability in the data, so PTA is unable to learn accurately (all the underlying patterns) to generalize the test set. Hence, we are getting 44.84% error rate for test samples instead of 0% error rate.

g)

For n = 1000 (taking only 1st 1000 training samples), η = 1 and ϵ = 0, we can see from the below graph that after 34 epochs the number of misclassifications converged to 0.



Here, test samples size is 10000 (all test samples are taken)

So, after updating the weights till the number of misclassifications converged to 0 for the training set, the error rate on test samples was 17.36%.

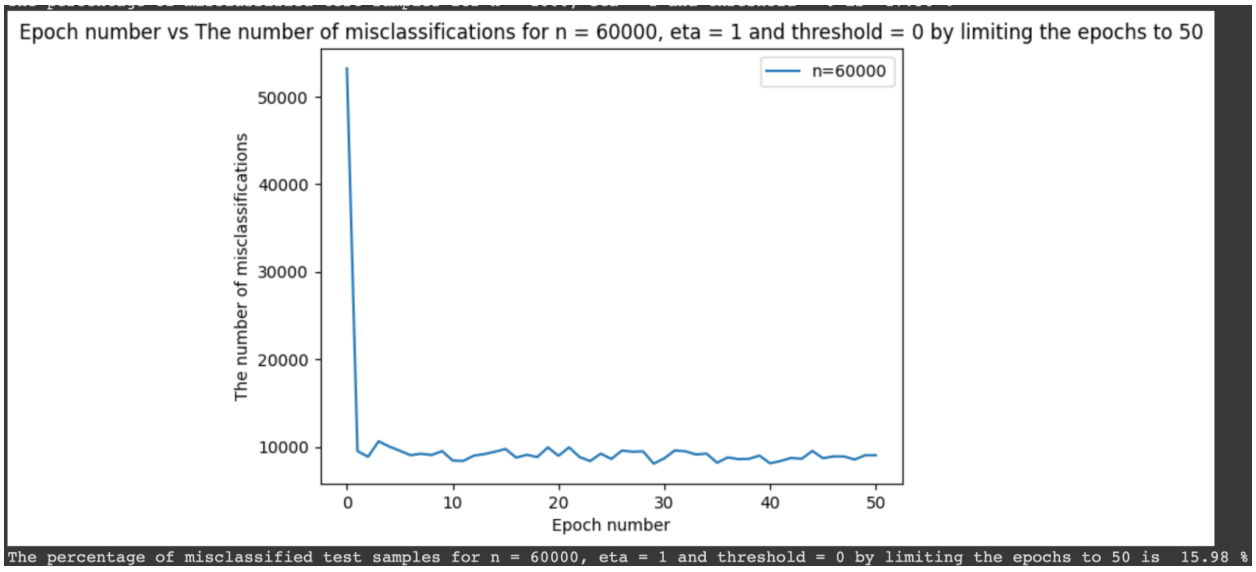
Error rate on training samples = 0% (as at end the number of misclassifications is 0)
Error rate on test samples = 17.36%

As the training set is large (1000 samples) compared to the previous training set (50), it increased the range of variability in the data. So, the error rate for the test samples by using the weights obtained from 1000 training samples (17.36%) is less than 50 training samples(44.84%). So, it increased the efficiency in the test samples.

But still the training set is small (1000 samples) compared to the test set (10,000), so it does not represent the full range of variability in the data. Hence, PTA is unable to learn accurately (all the underlying patterns) to generalize the test set, so we are not getting a 0% error rate for test samples.

h)

For n = 60000 , η = 1 and ϵ = 0, we can see from the below graph that the number of misclassifications is not converging to 0. So, I have set the number of epoch limits to 50, to plot the graph.



Here, test samples size is 10000 (all test samples are taken)

So, after updating the weights till the number of epochs is 50, the error rate on test samples was 15.98%

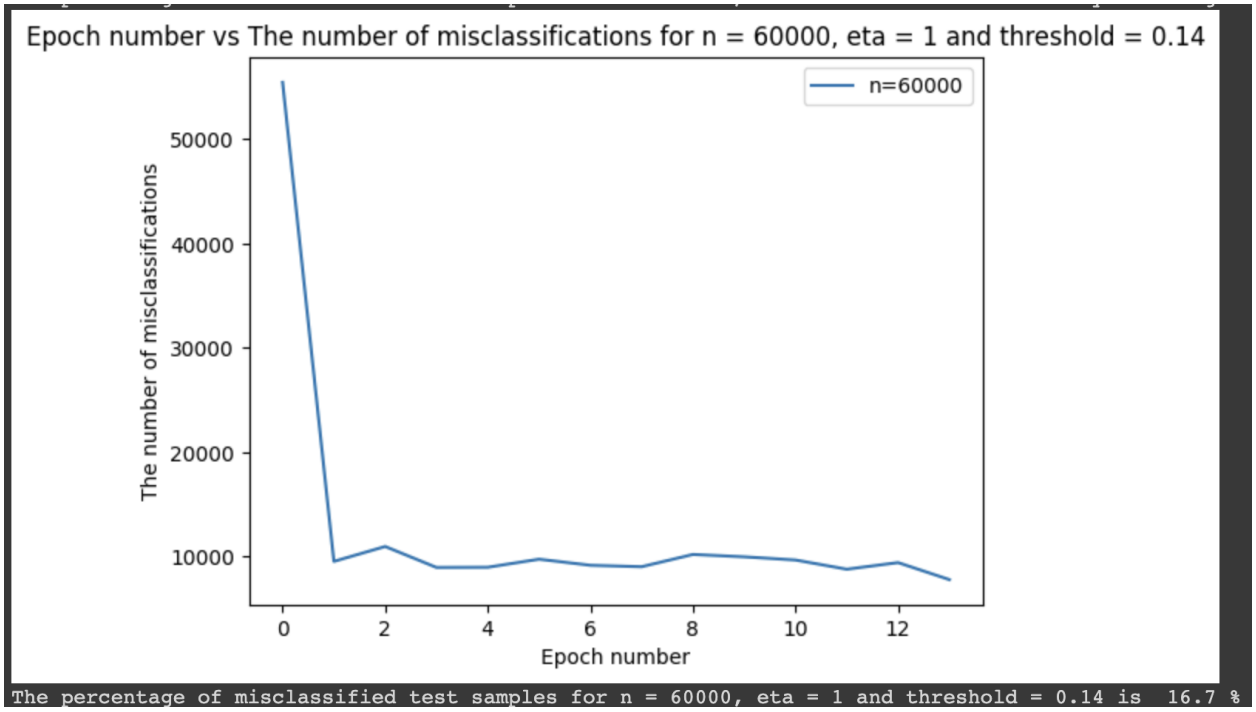
The number of misclassifications were not converging and oscillating between certain thresholds. [*Threshold is calculated by using (*$misclassifications(epoch - 1)/n$*)*]. Here, in this case it is oscillating between 0.13 and 0.17.

i)

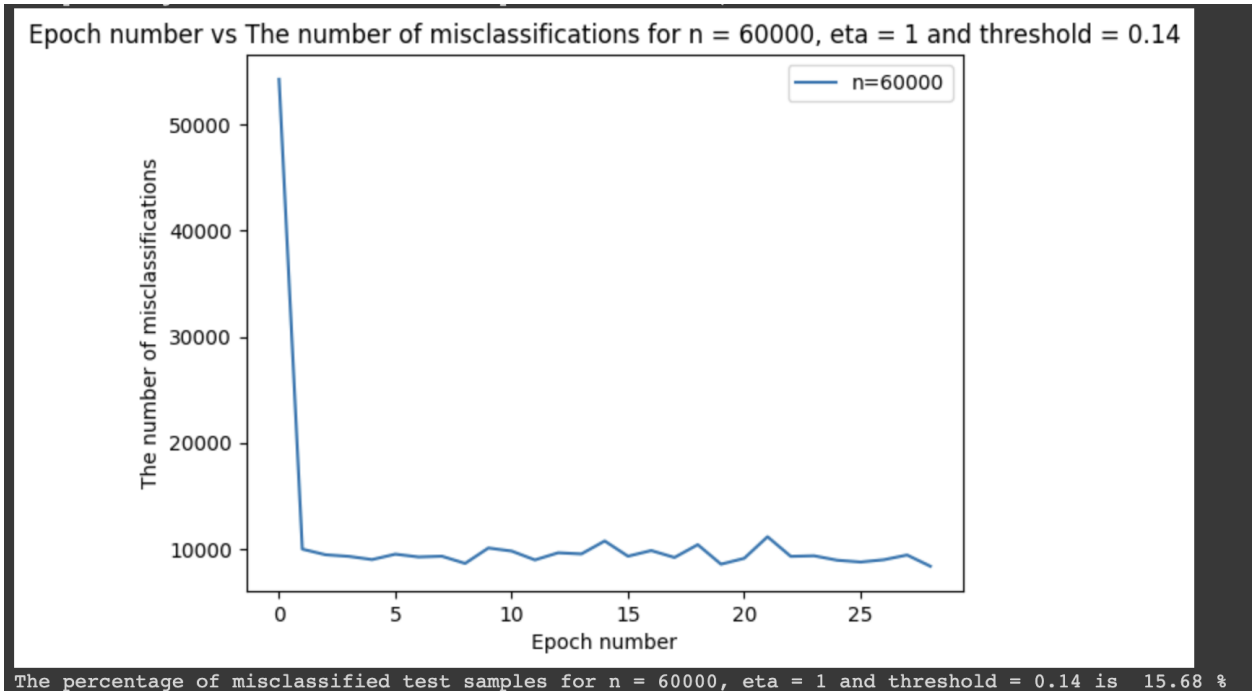
As in this case, the number of misclassifications were oscillating between 0.13 and 0.17 thresholds, so I have taken threshold as 0.14

Here for every case, n = 60000 , η = 1 and ϵ = 0.14

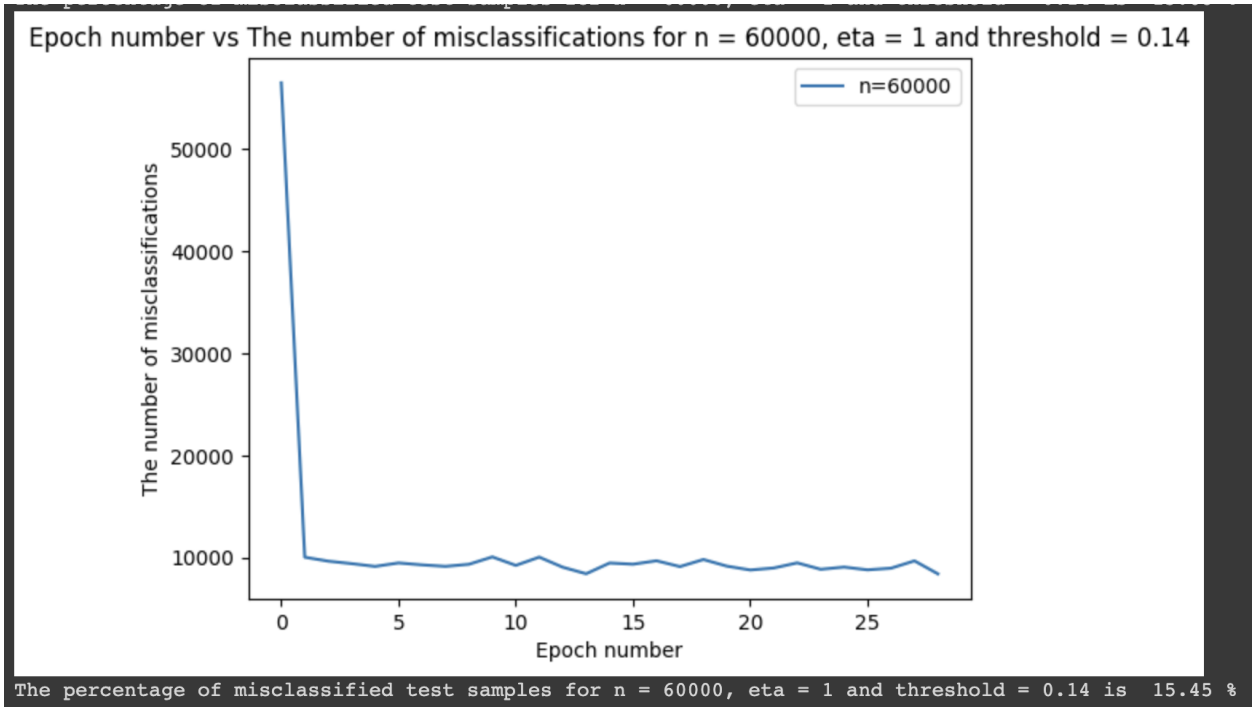
By taking the 1st random set of weights, below graph of number of misclassifications vs epochs is obtained.



By taking the 2nd random set of weights, below graph of number of misclassifications vs epochs is obtained.



By taking the 3rd random set of weights, below graph of number of misclassifications vs epochs is obtained.



We can observe from the above graphs that for different weights it converges at the end to the threshold defined but the number of epochs would be different. Even the test error rate is almost similar but they are different for different weights.