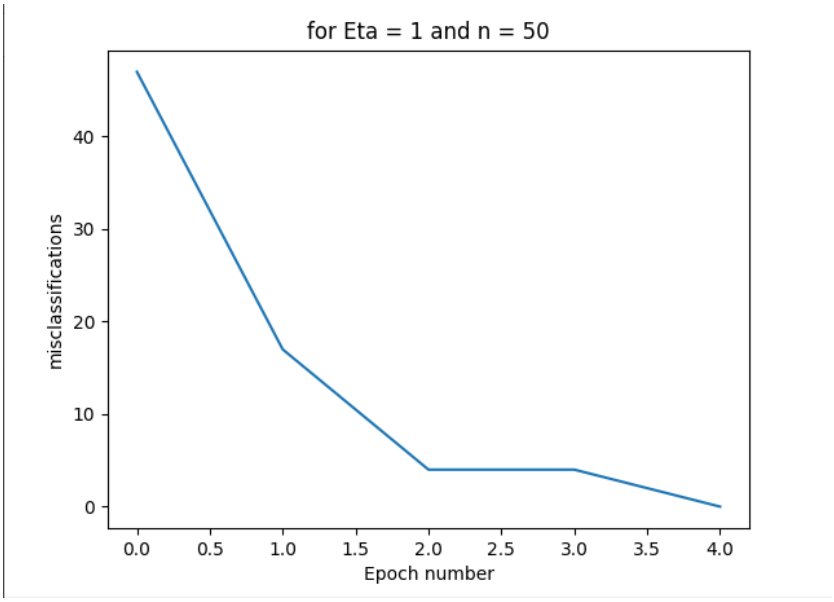
Assignment 3

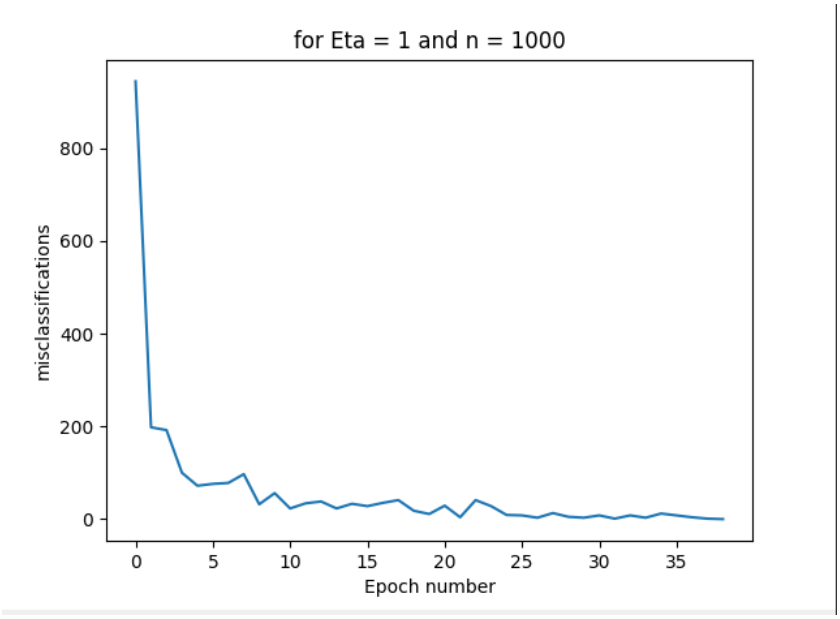
Plot for eta = 1 and n = 50 (epsilon value is zero):



The number of misclassifications reduces with each epoch until convergence.

The percentage of errors on the testing sample is 45.6%. This discrepancy is because the model has been trained with very few numbers of images and so it cannot accurately classify all the images in the training set properly.

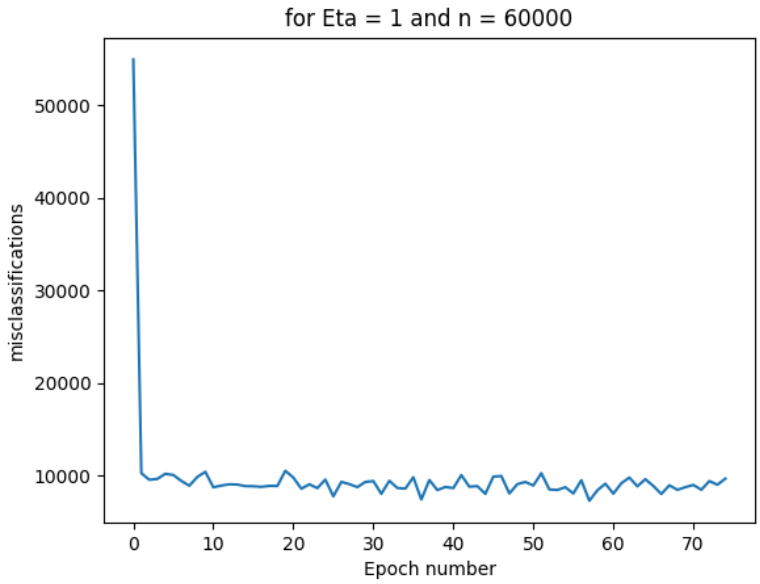
Plot for eta = 1 and n = 1000 (epsilon is zero):



This takes a total of 38 epochs to converge. This is because the dataset is much larger than before and thus takes time for convergence.

The percentage of errors this time is 17.63%. This is because the model has been trained on a much larger data set which has many more types of images, and this can classify the testing set much more accurately.

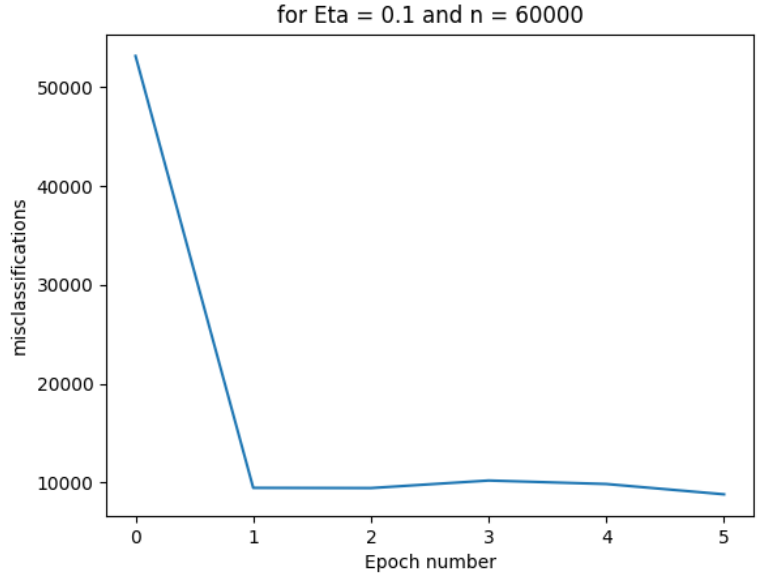
Plot for eta = 1 and n = 60000 (epsilon is zero):



The model never converges since epsilon can never reach zero. I have let it run for 75 epochs and after a few epochs, the epsilon value settled at around 0.15. This is because there is a very large and diverse data set and thus it cannot accurately cover all images. We would need even more layers for this.

The percentage of errors with testing set this time is 15.61 %. Here the epsilon and percentage values are very similar as more data is used to train the model and thus the same amount of accuracy is shown for both training data and testing data. The result of training on all training data is that the error percentage has reduced from training only with 1000 images.

Based on the previous results, the epsilon value should now be 0.15. I will also change the eta value to 0.1 and run the training. The plot for that is as below:



It only took 5 epochs to converge to epsilon of 0.15, as the number of times the weights are updated per epoch has increased to 60000.

The percentage of errors on testing set this time is 15.44% which is similar to the value before. The minima of the energy function is reached with these many updates from a lot of data.

Code:

```
import numpy as np

import random as rand

import idx2numpy

import matplotlib.pyplot as plt


rand.seed(1)


def step(input):
	return_array = np.empty(np.shape(input))
	for i in range(len(input)):
		if input[i] < 0:
			return_array[i] = 0
		else:
			return_array[i] = 1


	return return_array


def testing(w, testing_set, label_set):
	errors = 0


	for i in range(len(testing_set)):
		v = np.matmul(w, testing_set[i].flatten())
		if np.argmax(v) != label_set[i]:
			errors+=1


	print('percent of testing errors: ' + str((errors/len(testing_set)) * 100))


def training(eta, epsilon, n, training_set, label_set):
```

```
w = np.random.rand(10, 784)
#print(w)
epoch = 0
errors = []
while epoch == 0 or (errors[epoch - 1]/n > epsilon and epoch < 75):

    errors.append(0)

    for i in range(n):
        #print(label_set[i])
        v = np.matmul(w, training_set[i].flatten())
        if np.argmax(v) != label_set[i]:
            errors[epoch] += 1

    print('epoch: '+str(epoch))
    print('curr epsilon: '+str(errors[epoch]/n))
    print(errors[epoch])
    epoch += 1

    for i in range(n):
        d = np.zeros([10, 1])
        d[label_set[i]] = 1
        #w = w + eta*(d + step(np.matmul(w, training_set[i].flatten())))
        #np.matmul(np.subtract(d, step(np.matmul(w, training_set[i].flatten()))), training_set[i].flatten().T)
        #print(np.shape(training_set[i].reshape([1, 784])))
        w = np.add(w, eta * np.matmul(np.subtract(d, step(np.matmul(w, training_set[i].flatten())).reshape([10, 1])), training_set[i].reshape([1, 784])))

print(errors)
plt.plot(errors)
plt.xlabel('Epoch number')
```

```
        plt.ylabel('misclassifications')
        plt.title('for Eta = ' + str(eta) + ' and n = ' + str(n))
        plt.show()
        return w

training_set = idx2numpy.convert_from_file('C:/Users/visha/OneDrive/Documents/nn codes/mnist set/train-images.idx3-ubyte')
label_set = idx2numpy.convert_from_file('C:/Users/visha/OneDrive/Documents/nn codes/mnist set/train-labels.idx1-ubyte')
testing_set = idx2numpy.convert_from_file('C:/Users/visha/OneDrive/Documents/nn codes/mnist set/t10k-images.idx3-ubyte')
testing_label_set = idx2numpy.convert_from_file('C:/Users/visha/OneDrive/Documents/nn codes/mnist set/t10k-labels.idx1-ubyte')

#print(np.shape(training_set))

w = training(1, 0, 50, training_set, label_set)
testing(w, testing_set, testing_label_set)

w = training(1, 0, 1000, training_set, label_set)
testing(w, testing_set, testing_label_set)

#w = training(1, 0, len(training_set), training_set, label_set)
#testing(w, testing_set, testing_label_set)

w = training(0.1, 0.15, len(training_set), training_set, label_set)
testing(w, testing_set, testing_label_set)
```