CS 559 - Fall 2021
Homework #3

Krishna Garg
kgarg8@uic.edu

(f) $n = 50, \eta = 1, \epsilon = 0$ [Fig. 1(a)]

Epochs taken: 12

Test Misclassifications: 4560 or 45.6%

Even though the training converges to 0 misclassifications, we get 45.6% misclassifications on the test set. This discrepancy is due to the insufficient training samples to learn good weights for the generalization. This phenonmenon is also called *overfitting*.

(g) $n = 1000, \eta = 1, \epsilon = 0$ [Fig. 1(b)]

Epochs taken: 373

Test Misclassifications: 1692 or 16.92%

Again, we could learn the weights such that we could get 0 misclassifications on the training set, but the test set still reports 16.92% misclassifications. This percentage is quite less than part (e) since the number of training samples increased from 50 to 1000 which helps to better learn the features of the dataset but it is still not very accurate.

(h) $n = 60000, \eta = 1, \epsilon = 0$ [Fig. 2(a)]

The training does not converge even after 575 epochs. The misclassifications initially reduce from 54k to around 10k but as the training progresses, the misclassifications keep on fluctuating between 8k and 17k.

(i.1) $n = 60000, \eta = 1, \epsilon = 0.15$ [Fig. 2(b)]

Epochs taken: 32

Test Misclassifications: 1426

(i.2) **Results with different seed values:** [Fig. 3]

$seed = 112, n = 60000, \eta = 0.50, \epsilon = 0.15$

Epochs taken: 32

Test Misclassifications: 1426

$seed = 100, n = 60000, \eta = 0.50, \epsilon = 0.15$

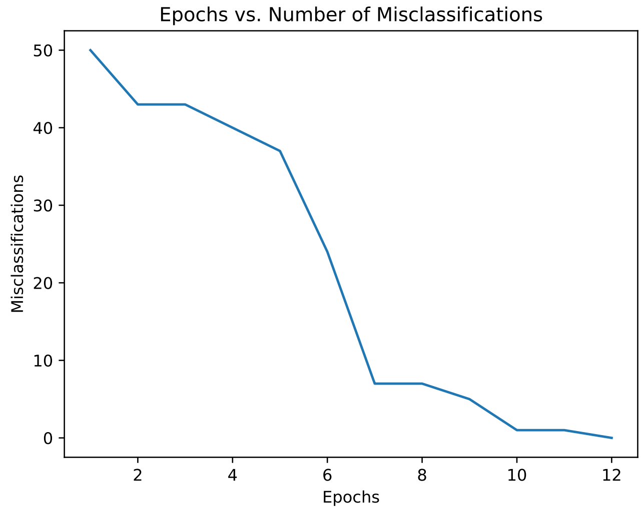
Epochs taken: 34

Test Misclassifications: 1538

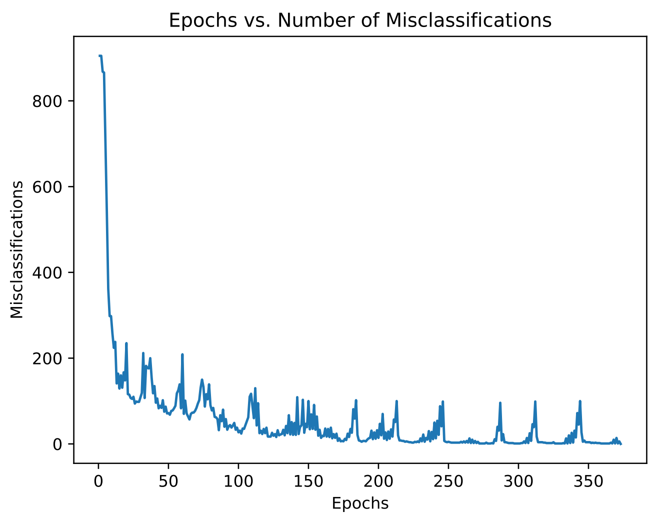
$seed = 250, n = 60000, \eta = 0.50, \epsilon = 0.15$

Epochs taken: 34

Test Misclassifications: 1330



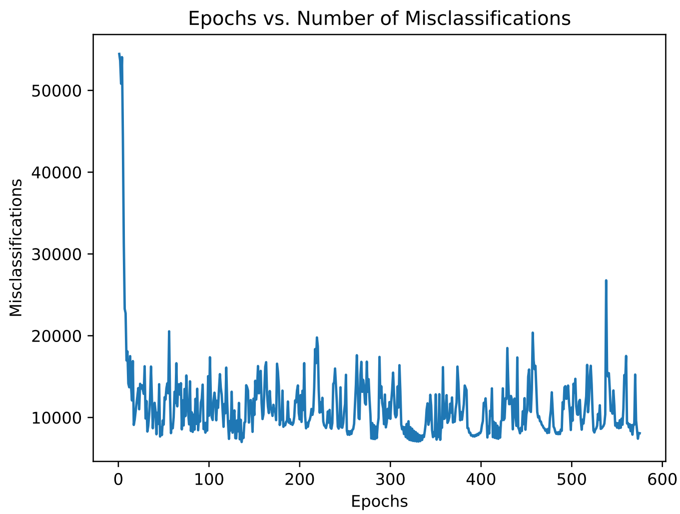
(a) N = 50



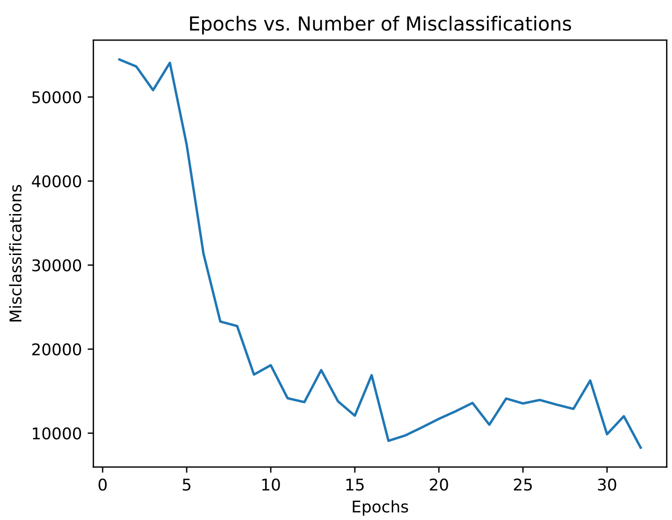
(b) N = 1000

Figure 1: Graphs for Epochs vs. Number of Misclassifications for different η values, N = 100

We can see from the above results that the different initializations of the weights W take different number of epochs for convergence and the number of misclassifications also vary according to the different seed values.

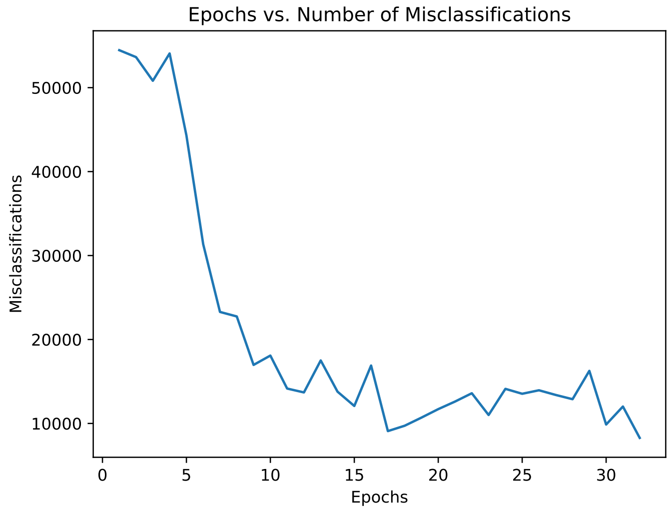


(a) N = 50

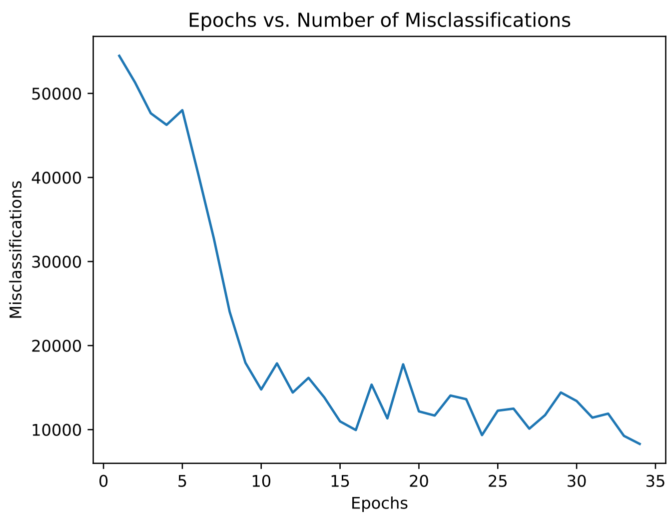


(b) N = 1000

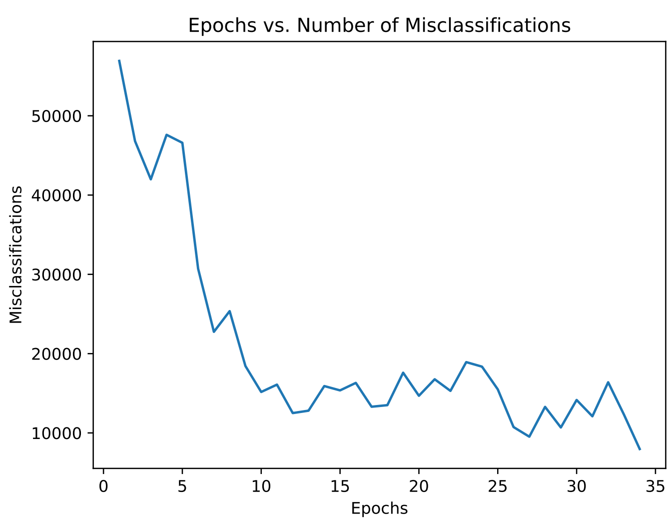
Figure 2: Graphs for Epochs vs. Number of Misclassifications for different η values, N = 100



(a) N = 50



(b) N = 1000



(c) N = 1000

Figure 3: Graphs for Epochs vs. Number of Misclassifications for different η values, N = 100

```python
import idx2numpy, numpy as np, matplotlib.pyplot as plt, pdb

seed = 250
np.random.seed(seed)

# Load train and test sets
file = 'data/train-images-idx3-ubyte'
train_inputs = idx2numpy.convert_from_file(file)
file = 'data/train-labels-idx1-ubyte'
train_labels = idx2numpy.convert_from_file(file)
file = 'data/t10k-images-idx3-ubyte'
test_inputs = idx2numpy.convert_from_file(file)
file = 'data/t10k-labels-idx1-ubyte'
test_labels = idx2numpy.convert_from_file(file)

W = np.random.uniform(-1, 1, (10, 784)) # initialize weights
n = 60000                             # no. of samples used for training
eta = 0.5                             # learning rate
eps = 0.15                            # threshold

print('seed: {}, #train_samples: {}, lr: {}, threshold: {}'.format(seed,
    n, eta, eps))

# Training
epoch = 0
errors = []
while(1):
   cur_epoch_errors = 0
   update = 0
   for i in range(n):
      x_i = train_inputs[i].reshape((-1, 1))
      v = W @ x_i
      pred = np.argmax(v)
      if pred != train_labels[i]:
         cur_epoch_errors += 1

      d_xi = np.zeros((10,1)) # 10 classes
      d_xi[train_labels[i]] = 1 # One-hot encoding of desired label
      update += eta * (d_xi - np.sign(np.maximum(v, 0))) @ x_i.T

   print('Epoch: {}, errors: {}, cur_threshold: {}'.format(epoch,
       cur_epoch_errors, cur_epoch_errors/ n))

   W += update               # update weights after the epoch
   errors.append(cur_epoch_errors)
   epoch += 1

   if cur_epoch_errors/ n <= eps: # threshold condition
      break

print('Train Misclassifications: ', errors)
print('Epochs taken:', epoch)

# Plot epochs vs. misclassifications
epoch_arr = list(range(1, epoch+1))
plt.figure()
plt.xlabel('Epochs')
plt.ylabel('Misclassifications')
plt.plot(epoch_arr, errors)
plt.title('Epochs vs. Number of Misclassifications')
plt.savefig('N_{}_eps_{}_eta_{}_seed_{}.pdf'.format(n, eps, eta, seed))

# Testing
errors = 0
```

```python
for i in range(test_inputs.shape[0]):
   x_i = test_inputs[i].reshape((-1, 1))
   v = W @ x_i
   pred = np.argmax(v)
   if pred != test_labels[i]:
      errors += 1
print('Test Misclassifications: ', errors)
```