Princess Tara Zamani
Adv Opt for Machine Learning
Homework 3
Gradient Descent for Logistic Regression

The objective of this assignment was to apply gradient descent to solve the logistic regression problem defined as:

$$P(y_i = 1 \mid x_i, w, b) = h_{w,b}(x_i), \; where \; h_{w,b}(x_i) = 1 / (1 + exp(-(w^T x_i + b)))$$

where $w \in R^d$, $b \in R$. W and b are the parameters of the model to be learned.

The gradient for the definitions for w and b:

$$\nabla f_w(x) = \frac{1}{n}\sum(h_{w,b}(x_i) - y_i)\, x_i$$

$$\nabla f_b(x) = \frac{1}{n}\sum(h_{w,b}(x_i) - y_i)$$

The entries for x and y were provided by a large dataset. Before learning, the data set was normalized by subtracting the mean and dividing by the std of each feature. The normalized dataset was broken up into two subsets for training (2,656 samples) and testing (1,000 samples).

The training set was used for training the algorithm. A learning rate of *1e-3* was chosen and w_0 and b_0 were initialized randomly. The gradient algorithm implementation is iterative and takes the following form for both parameters, w and b:

$$x_{k+1} = x_k - \eta\nabla f(x_k), \text{ where } \eta \text{ is the learning rate.}$$

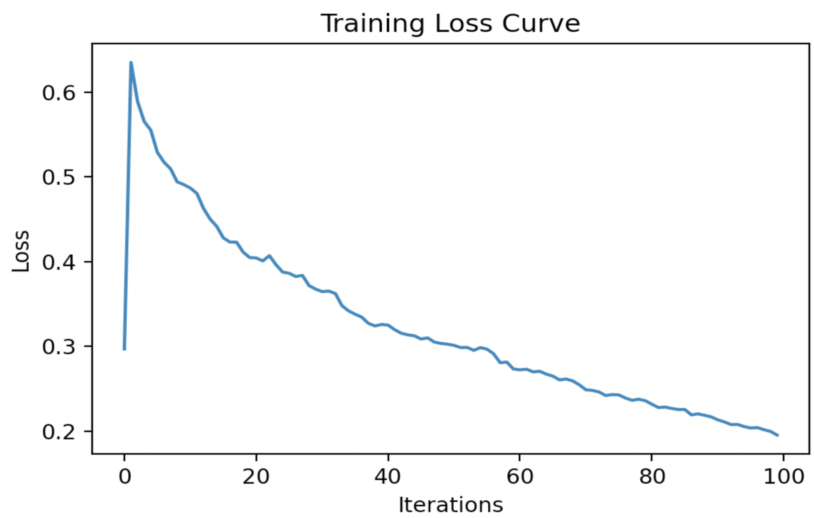
After training, the learned *w* and *b* parameters were used to predict the labels of the test dataset using the logistic model in which,

$$P(y_i = 1 \mid x_i, w, b) = h_{w,b}(x_i) \qquad P(y_i = 0 \mid x_i, w, b) = 1 - h_{w,b}(x_i)$$

The highest predicted probability determines the predicted label.

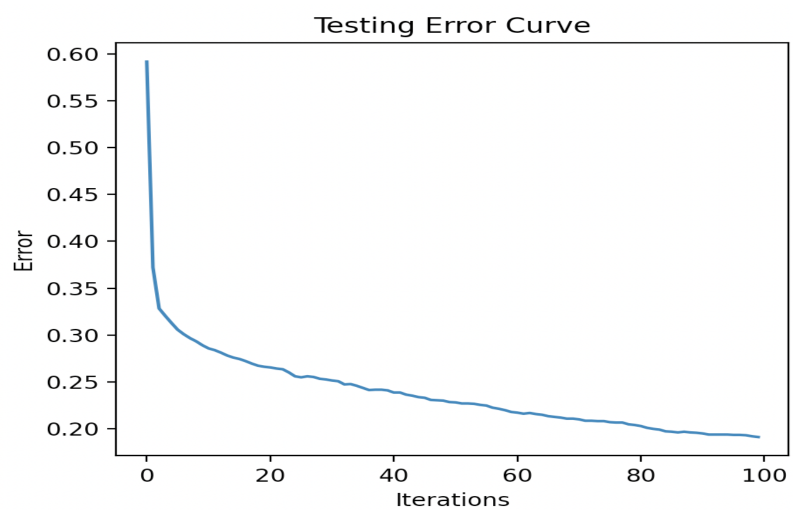
The results of the gradient descent for this assignment are shown below.

The Training Loss Curve



The training loss curve follows an expected trajectory. Other than the initial spike, which I presume is from the randomization of the variable initializations, the loss decreases as iterations go on.

The Testing Error Curve



The error curve is based on the error between the predicted label value and the actual label value. We can see that the error also decreases as more iterations occur. Without any parameter tuning for further improvement, the final accuracy when the model was used on the test data was ~83% on average.