Assignment 4

Calvin Tran

All code for this assignment can be found in the file named Assignment\_4\_Exercise\_X.ipynb found with this assignment.

# Exercise 1

The implementation of the logistic regression algorithm can be found in Assignment\_4\_Exercise\_1.ipynb. A step size of 0.1 was used and a tolerance of 10e-7 was used for convergence criteria.

The plot for test error and training error vs the number of epochs can be found below:

A graph with numbers and a line

Description automatically generated

The plot for test loss and training loss vs the number of epochs can be found below:

A graph with numbers and a line

Description automatically generated

The final test error using logistic regression is 0.57%. Using the sklearn implementations of the other classifiers, we can quickly determine the test errors of the other classifiers:

* MED: 0.61%
* GED: 0.38%
* kNN: 0.28%

The classifier that performs the best (and has the lowest test error) is the kNN classifier. We can briefly discuss each classifier and assess its suitability when used against the MNIST dataset.

The MED classifier uses Euclidean distance as a way to assign samples to classes. However, because it uses Euclidean distance, high-dimensional data like MNIST can cause the classifier to suffer from the curse of dimensionality.

The GED classifier is similar to the MED classifier, but uses covariance matrices instead. Although the GED classifier likely performs better than the MED classifier when classifying the MNIST dataset due to its ability to capture data distributions, the GED’s performance is also limited in high-dimensional spaces.

Logistic regression is often considered a good choice for binary classification problems, which in this case, we have updated the dataset to only consider two classes. However, when considering the whole MNIST dataset, logistic regression may not be the best choice.

Finally, the kNN classifier assigns new samples to the majority class among its k-nearest neighbors. This classifier is often effective in image classification problems, like this one. Although it may become computationally expensive in large datasets, for the MNIST dataset, the kNN classifier is ideal, which explains why it performed the best out of the four classifiers.

# Exercise 2