MAE263F MS Comprehensive Exam

Che Jin Goh

This report presents the implementation of gradient descent to fit a linear model to generated-data using MATLAB. The dataset simulates a relationship between input and output variables with added noise, modeled as y=mx+b. The model minimizes the Mean Squared Error (MSE) loss by iteratively updating parameters m and b using calculated gradients. Training progress is evaluated by analyzing the loss curve over epochs, and the final model is visualized by comparing the fitted line to the actual data points. Results demonstrate the effectiveness of gradient descent in learning the underlying data patterns.

I. Introduction

This report focuses on the application of gradient descent to fit a linear model to a simulated dataset in MATLAB. The dataset represents a noisy relationship between input and output variables, modeled as y=mx+b. The objective is to train the model by minimizing the Mean Squared Error (MSE) loss function through iterative updates of the parameters m (slope) and b (intercept) based on computed gradients. The training process involves initializing parameters, computing predictions, calculating gradients, and updating parameters to reduce the error. The effectiveness of the approach is evaluated by analyzing the loss curve over training epochs and visualizing the fitted model against the actual data. The experiment demonstrates the practical implementation of gradient descent in optimizing a simple linear regression problem.

II. Equations

a) Mean Squared Error (MSE) Loss Function

$$Loss\,(m, b) = \frac{1}{N}\sum_{i=1}^{N}(y_i - (m * x_i + b))^2$$

b) Gradient of loss function with respect to m

$$\frac{\delta loss}{\delta m} = \frac{-2}{N}\sum_{i=1}^{N} x_i * (y_i - (m * x_i + b))$$

c) Gradient of loss function with respect to b

$$\frac{\delta loss}{\delta b} = \frac{-2}{N}\sum_{i=1}^{N}(y_i - (m * x_i + b))$$

III. Results And Discussion

Problem 1

1)

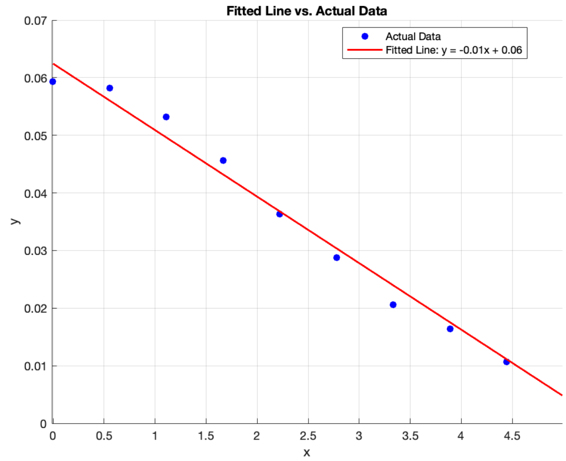

Figure 1: Predicted values vs actual data

2)

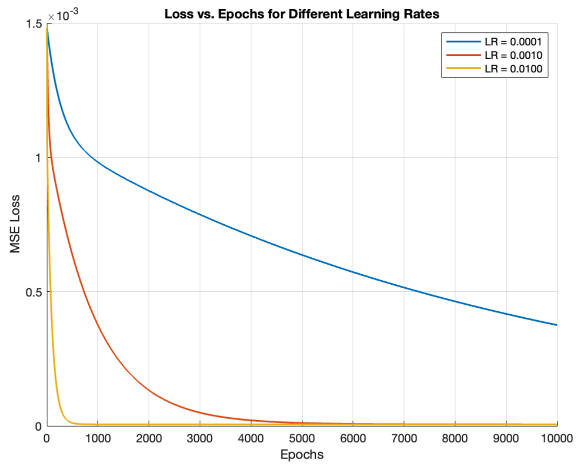

Figure 2: MSE Loss with 0.0001, 0.001, 0.01 learning rates

MAE263F MS Comprehensive Exam

Che Jin Goh

3) Problem 3

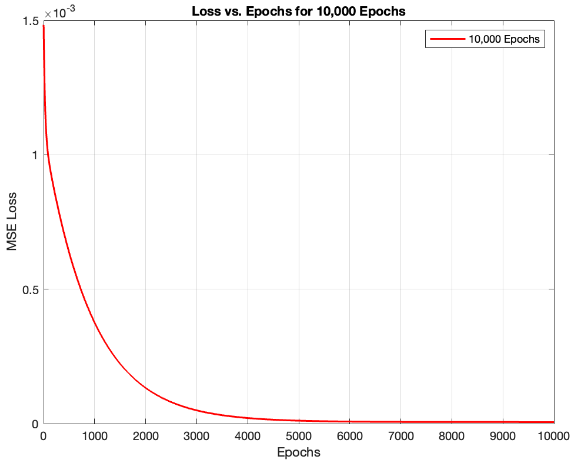

Figure 4: MSE loss with varying epochs

Figure 1 compares the predicted values with the actual data after training the model using gradient descent with a learning rate of η=0.001 and 10,000 epochs. The red fitted line aligns closely with the blue actual data points, demonstrating that the model effectively captures the underlying relationship between the variables. The parameters m (slope) and b (intercept) were optimized to minimize the Mean Squared Error (MSE), resulting in accurate predictions and a well-fitted line.

Figure 2 illustrates the effect of varying learning rates (0.0001, 0.001, 0.01) on the convergence of gradient descent. A high learning rate leads to rapid convergence within the first few hundred epochs, but it risks instability or overshooting if not carefully managed. A moderate learning rate achieves a smooth and reliable optimization, striking a balance between convergence speed and stability. Conversely, a low learning rate results in slow convergence, requiring a significantly larger number of epochs to minimize the loss effectively.

Figure 3 evaluates the effect of the number of epochs on model performance, using 10,000 epochs as the configuration. The loss decreases rapidly during the initial training stages, indicating effective learning. After approximately 5,000 epochs, the loss curve stabilizes, showing diminishing returns with additional training. This demonstrates that while longer training helps refine the model, the majority of improvements occur early in the process, with little added benefit from excessively high epoch counts.

The experiments reveal that the best performance is achieved with a learning rate of η=0.001 and sufficient epochs (5,000–10,000). The findings emphasize the need for careful tuning of hyperparameters to achieve both efficient convergence and accurate model predictions.