MAE 263F: Homework 3

Sarah Enayati

I. Assignment Part 1

A. Assignment Guidelines/Expectations

This assignment tasks us with fitting data to both linear and nonlinear models using gradient descent and backpropagation. We begin by generating a set of synthetic data points based on a given function and adding noise to simulate variability. For the first part, we fit the data to a linear model by optimizing the slope and intercept using gradient descent, where we iteratively adjust the parameters to minimize the error between predicted and actual values.

In the second part, we use a nonlinear model, requiring us to optimize additional parameters using the same iterative approach. We will experiment with different learning rates and numbers of training iterations to understand how these settings influence the performance and convergence of the model. Finally, we will compare our predicted values to the actual data through visualization and analyze how well our models perform.

B. Predicted Values vs. Actual Data (Given Parameters)

We start by performing gradient descent and backpropagation to fit the data to the linear model using the given parameters (epochs = 10000 and learning rate = 0.001.) We simulate the predicted y-values and compare to the given data:

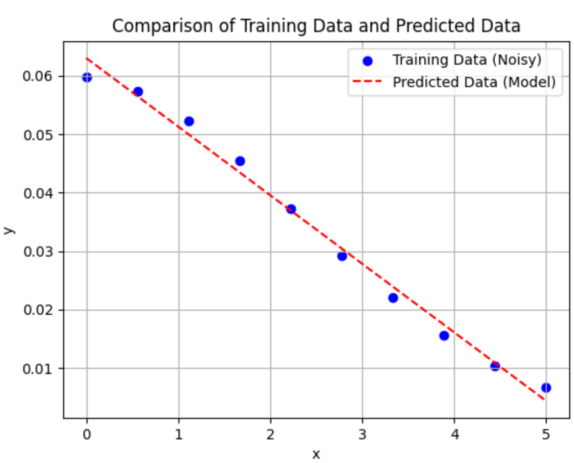

Figure 1: Predicted y-values versus actual data for epochs = 10000 and learning rate = 0.001.

C. Experimenting with Parameters

We want to experiment with the learning rate and the number of epochs to minimize the loss while ensuring the model converges. The learning rate determines the size of the steps taken towards minimizing the loss function. If the learning rate is too small, the model should converge very slowly, requiring a large number of epochs to reach an optimal solution. On the other hand, if the learning rate is too large, the model might overshoot the optimal solution or fail to converge entirely, leading to unstable or oscillating results. The number of epochs specifies how many times the entire dataset is used to update the model's parameters. A small number of epochs might result in underfitting, where the model does not learn the underlying patterns in the data. However, too many epochs can lead to overfitting, where the model fits the training data too closely and performs poorly on new data.

Using the previously defined parameters in section *B*, we experienced a final loss = 3e-6. We will refer to this case as our "initial case."

We decrease the learning rate to 0.0001, keeping the number of epochs to 1000:

Figure 2: Predicted y-values versus actual data for epochs = 10000 and learning rate = 0.0001.

We experience a final loss = 3.8e-4, significantly higher than our initial case. This makes sense intuitively as we are decreasing the rate at which the algorithm can learn while keeping the number of epochs constant. We are failing to converge and missing the optimal solution. We conclude that our previous learning rate was better. We now keep learning rate constant at 0.001, and increase our number of epochs to 20000:

Figure 3: Predicted y-values versus actual data for epochs = 20000 and learning rate = 0.001.

Our final loss = 3e-6, which matches our initial results. This implies that increases the number of epochs past 10000 will not result in a lower loss, but instead might run the risk of overfitting to our data. We conclude that our initial case had optimal parameters.

II. Assignment Part 2

A. *Assignment Guidelines/Expectations*

We repeat part 1, but for a nonlinear case.

B. *Predicted Values vs. Actual Data (Given Parameters)*

Using a learning rate of 0.001 and epochs = 10000, we obtain the following results:

Figure 4: Nonlinear fit for learning rate = 0.001 and epochs = 10000.

C. *Experiments*

Using the same logic as before, we experiment with the number of epochs and the learning rate. We begin by increasing the number of epochs to 20,000:

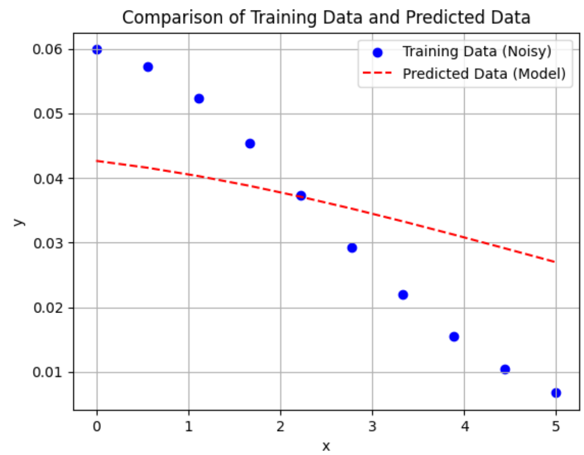

Figure 5: Nonlinear fit for learning rate = 0.001 and epochs = 20000.

We note that this did not increase the accuracy of the run. We now decrease the learning rate to 0.0001:

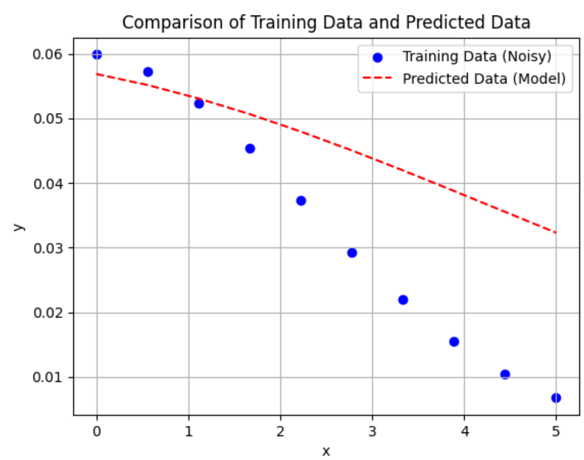

Figure 6: Nonlinear fit for learning rate = 0.0001 and epochs = 10000.

This looks like it might be slightly better than before, so we further decrease it 0.00001:

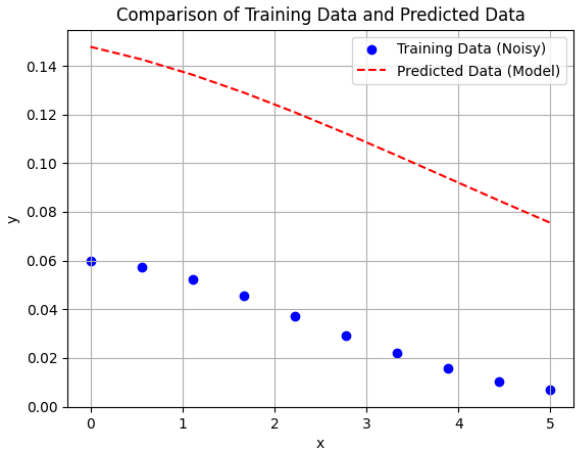

Figure 7: Nonlinear fit for learning rate = 0.00001 and epochs = 10000.

This only increased the loss. We conclude that our initial results were the best.