MAE 263F: Homework 3

Zhicheng Ren

12.10.2024

1 Abstract

This assignment primarily focuses on the problem of regression fitting. In terms of models, it mainly explores linear regression and nonlinear regression. Methodologically, it investigates the influence of hyperparameters—specifically the number of epochs and the learning rate—on model fitting effectiveness when using gradient descent and backpropagation methods.

2 Linear model

When performing linear regression fitting on the data, the parameters to be fitted are m and b, and the formula is as follows:

$$y = mx + b$$

Mean Squared Error(MSE) loss function:

$$\text{Loss}(m, b) = \frac{1}{N} \sum_{i=1}^{N} \left(y_i - (mx_i + b)\right)^2$$

The gradient of the loss function with respect to m and b are:

$$\frac{\partial \,\text{Loss}}{\partial m} = -\frac{2}{N} \sum_{i=1}^{N} x_i \left(y_i - (mx_i + b)\right)$$

$$\frac{\partial \,\text{Loss}}{\partial b} = -\frac{2}{N} \sum_{i=1}^{N} \left(y_i - (mx_i + b)\right)$$

Using gradient descent, m and b is updated as follows:

$$m \leftarrow m - \eta \frac{\partial \,\text{Loss}}{\partial m}$$

$$b \leftarrow b - \eta \frac{\partial \,\text{Loss}}{\partial b}$$

2.1 Fitting Result

Initially, set the hyperparameters $Epochs = 10000$ and $LearningRate = 0.001$. The fitting results are shown in the figure below:

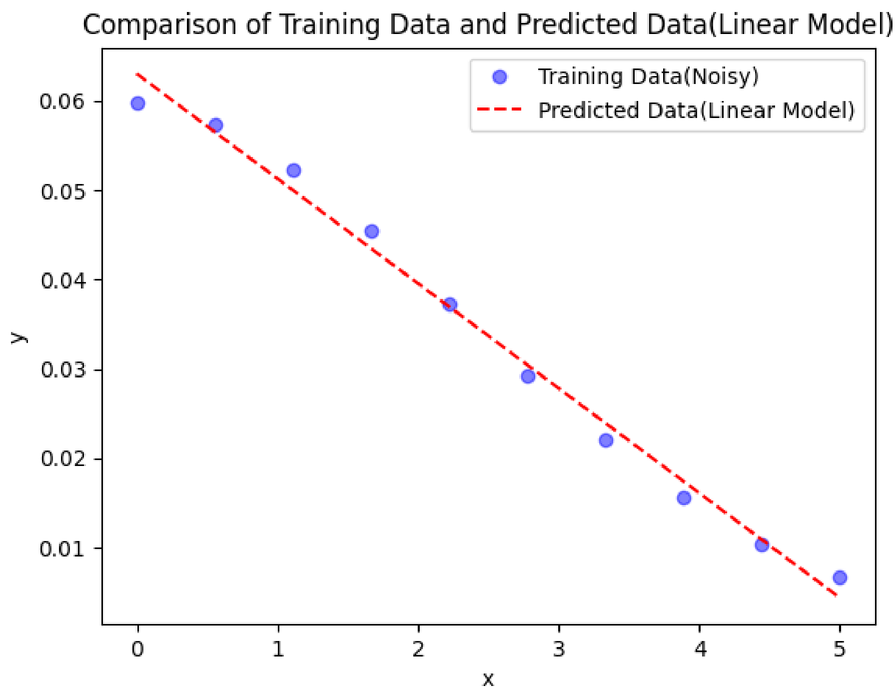
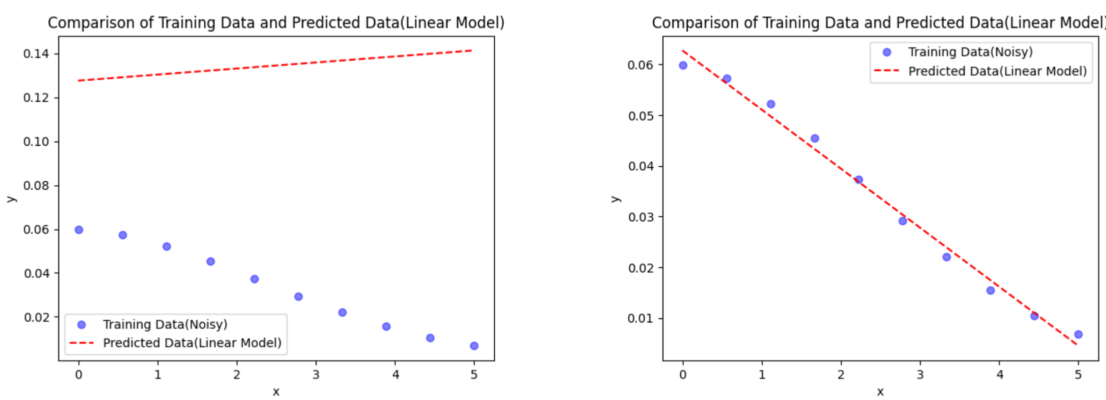


Figure 1: Actual vs Predicted y Values($Epochs = 10000, lr = 0.001$)

2.2 Explore the Epochs

When keeping $LearningRate = 0.001$ and the initial values of m, b unchanged, adjusting the number of epochs yields the following results:



(a) $Epochs = 100, lr = 0.001$

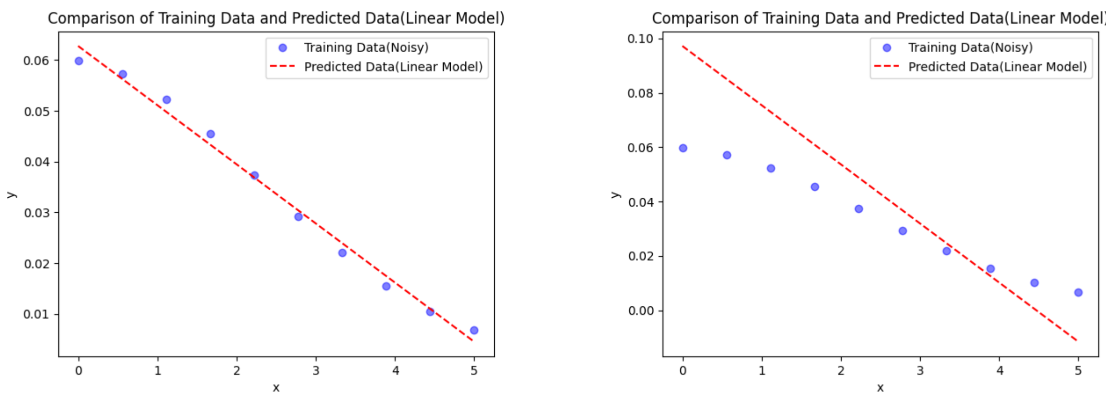
(b) $Epochs = 1000000, lr = 0.001$

Figure 2: Actual vs Predicted y Values (different Epochs)

From Figure 2(a) and the output loss values, it can be seen that when the learning rate is fixed and the number of epochs is too small, the simulation cannot reach the convergence solution, resulting in poor fitting performance. From Figure 2(b) and the output loss values, it can be observed that although the number of epochs increases, the actual fitting performance does not improve significantly.

2.3 Explore the learning rate

When keeping $Epochs = 10000$and the initial values of m, b unchanged, adjusting the learning rate, the results are as follows:



(a) $Epochs = 10000, lr = 0.1$

(b) $Epochs = 10000, lr = 0.0001$

Figure 3: Actual vs Predicted y Values (different learning rate)

When the number of epochs is fixed and the learning rate is set to 1, the loss value returns NaN after iterations, likely due to issues in gradient computation. Adjusting the learning rate to 0.1 yields the result shown in Figure 3(a), which indicates a good fitting performance. When the learning rate is adjusted to 0.0001, the results are shown in Figure 3(b). Combined with the output loss values, it can be seen that the convergence solution has not yet been reached, resulting in poor fitting performance.

2.4 Conclusion

Based on the above results, it can be concluded that the learning rate affects the speed of model convergence, while the number of epochs influences whether the model can reach the convergence solution. Therefore, to achieve better results, it is necessary to set an appropriate learning rate and number of epochs. This requires balancing between avoiding too slow a speed and preventing overshooting the convergence solution. By observing the changes in loss values and the final fitting result graphs, the optimal number is $Epochs = 100$, and $LearningRate = 0.1$.

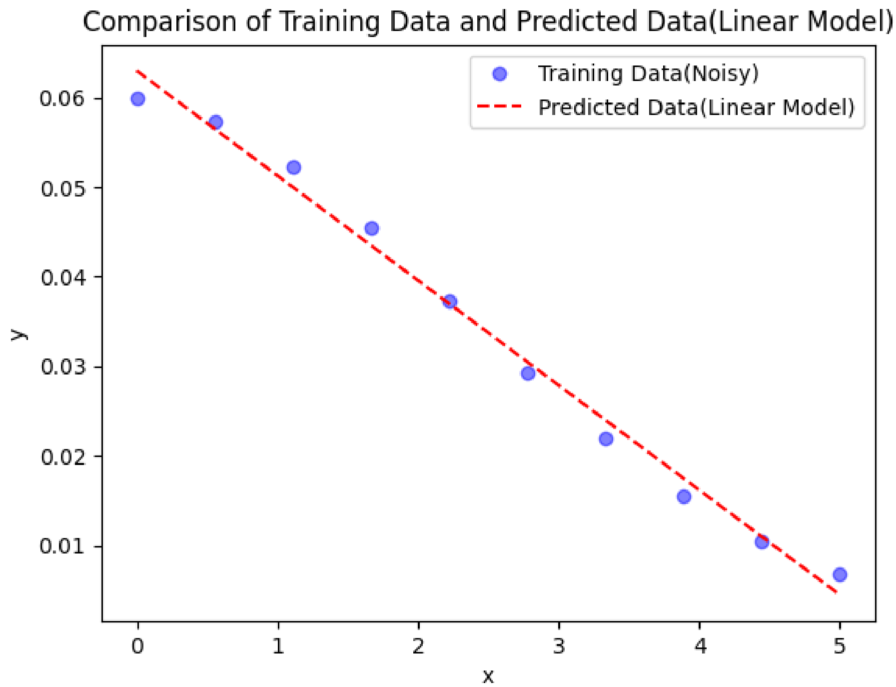


Figure 4: Actual vs Predicted y Values($Epochs = 100, lr = 0.1$)

3 Nonlinear model

When performing nonlinear fitting on the data, the parameters to be fitted include n, a, m and b. The formula is as follows:

$$y = n \cdot \exp\left(-a \cdot y_{\text{int}}\right), \quad \text{where} \quad y_{\text{int}} = (mx + b)^2$$

Mean Squared Error(MSE) loss function:

$$\text{Loss}(n, a, m, b) = \frac{1}{N} \sum_{i=1}^{N} \left(y_i - n \cdot \exp\left(-a \cdot (mx_i + b)^2\right)\right)^2$$

The gradient of the loss function with respect to n, a, m and b are:

$$\frac{\partial \, \text{Loss}}{\partial n} = -\frac{2}{N} \sum_{i=1}^{N} \left(y_i - n \cdot \exp\left(-a \cdot y_{\text{int}}\right)\right) \cdot \exp\left(-a \cdot y_{\text{int}}\right)$$

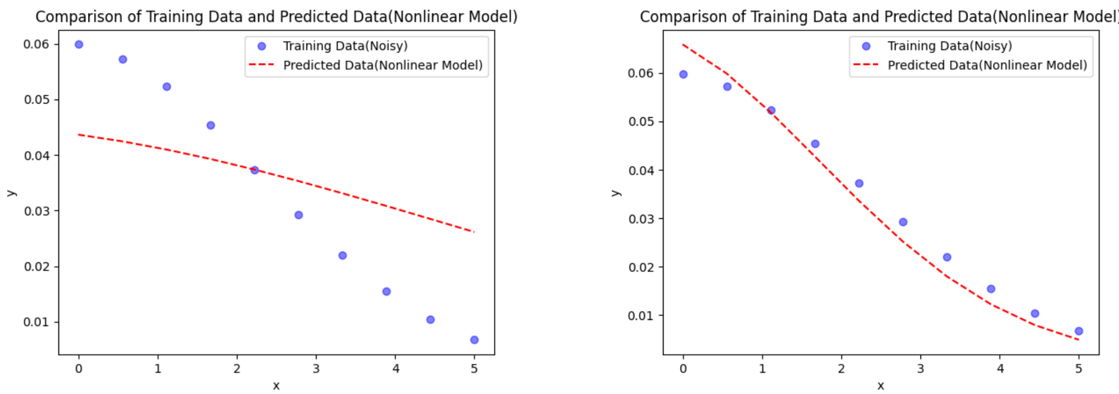
$$\frac{\partial \, \text{Loss}}{\partial a} = \frac{2}{N} \sum_{i=1}^{N} \left(y_i - n \cdot \exp\left(-a \cdot y_{\text{int}}\right)\right) \cdot n \cdot \exp\left(-a \cdot y_{\text{int}}\right) \cdot \left(-y_{\text{int}}\right)$$

$$\frac{\partial \, \text{Loss}}{\partial m} = \frac{2}{N} \sum_{i=1}^{N} \left(y_i - n \cdot \exp\left(-a \cdot y_{\text{int}}\right)\right) \cdot n \cdot \exp\left(-a \cdot y_{\text{int}}\right) \cdot (-a) \cdot 2 \cdot (mx_i + b) \cdot x_i$$

$$\frac{\partial \, \text{Loss}}{\partial b} = \frac{2}{N} \sum_{i=1}^{N} \left(y_i - n \cdot \exp\left(-a \cdot y_{\text{int}}\right)\right) \cdot n \cdot \exp\left(-a \cdot y_{\text{int}}\right) \cdot (-a) \cdot 2 \cdot (mx_i + b)$$

3.1 Fitting Result

Initially, set the hyperparameters $Epochs = 10000$ and $LearningRate = 0.001$. The fitting results are as follows:



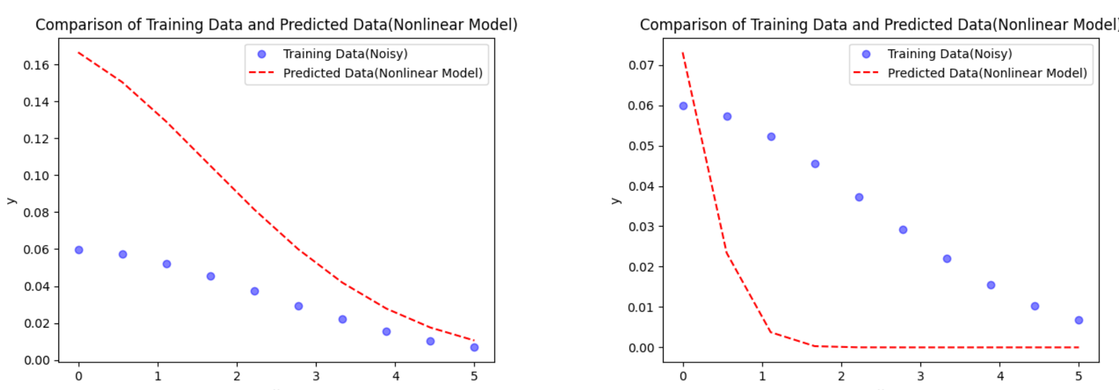
(a) $Epochs = 10000, lr = 0.001$ (b) $Epochs = 10000, lr = 0.001$

Figure 5: Actual vs Predicted y Values (different initial value)

During the experiment, it was found that setting different initial values for n, a, m, and b had a significant impact on the final fitting performance. Therefore, when setting the initial values, multiple calls to $np.random.rand()$ (with a fixed random seed) were used to try different initial values. The final results are shown in Figure 5. As can be seen from Figure 4, the fitting performance in (b) is significantly better than in (a).

3.2 Explore the Epochs

When keeping $LearningRate = 0.001$ and the initial values of n, a, m, b unchanged, adjusting the number of epochs yields the following results:



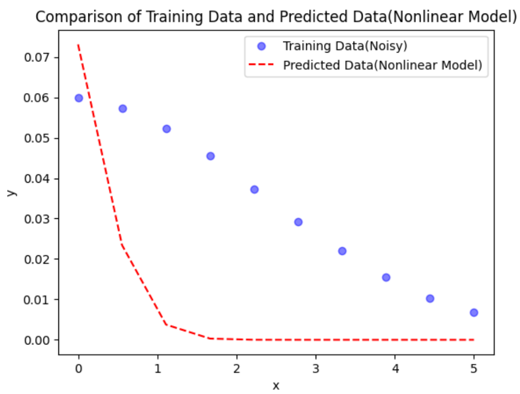
(a) $Epochs = 100, lr = 0.001$ (b) $Epochs = 1000000, lr = 0.001$

Figure 6: Actual vs Predicted y Values (different Epochs)

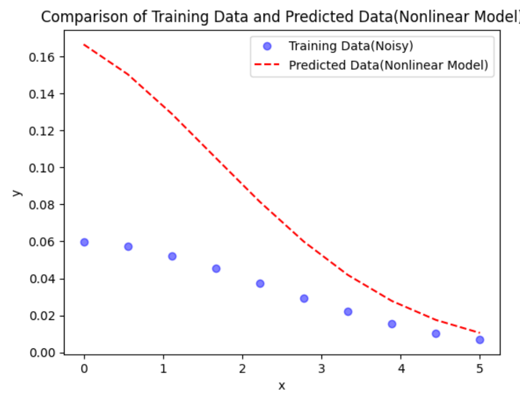
From Figure 6(a), combined with the output loss values, it can be seen that when the learning rate is fixed and the number of epochs is too small, the result has not yet converged, resulting in poor fitting performance. From Figure 6(b), combined with the output loss values, it can be observed that when the learning rate is fixed and the number of epochs is too large, the result overshoots the optimal solution, fitting toward a poorer direction, and the final fitting performance is also poor.

3.3 Explore the Learning Rate

When keeping $Epochs = 10000$ and the initial values of n, a, m, b unchanged, adjusting the learning rate, the results are as follows:



(a) $Epochs = 10000, lr = 0.1$



(b) $Epochs = 10000, lr = 0.0001$

Figure 7: Actual vs Predicted y Values (different learning rate)

From Figure 7(a) and the output loss values, it can be seen that when the number of epochs is fixed and the learning rate is too high, the optimal solution cannot be reached, and the fitting moves toward a poorer direction. From Figure 7(b), it can be observed that when the learning rate is too low, the convergence solution has not yet been reached, resulting in poor fitting performance.

3.4 Conclusion

Based on the above results, it can be concluded that in the nonlinear model fitting process, due to the complexity of the model, the initial values of the parameters to be fitted have a significant impact on the fitting performance. To address this issue, parameter values can be optimized by initializing them with random numbers. The learning rate has an even more pronounced impact on the fitting speed compared to linear models. The number of epochs is also crucial in determining whether the model can stably reach the optimal convergence solution. Therefore, it is essential to combine the output loss values to select an appropriate number of epochs and learning rate. After multiple experiments, the optimal solution was $Epochs = 1000$ and $LearningRate = 0.01$.

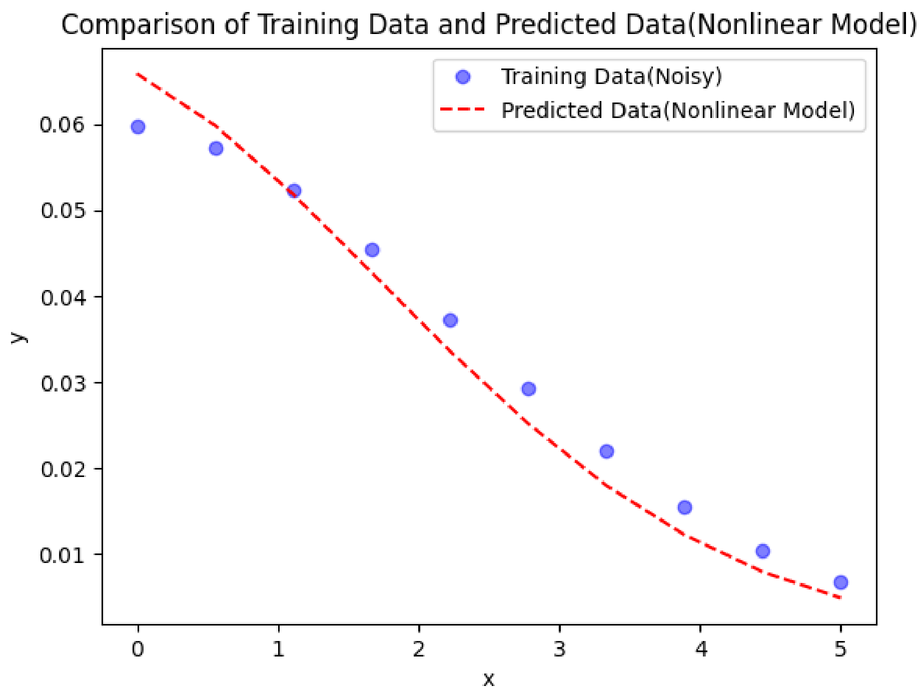


Figure 8: Actual vs Predicted y Values($Epochs = 1000, lr = 0.01$)