NYCU Introduction to Machine Learning, Homework 1

Part. 1, Coding (50%):

(10%) Linear Regression Model - Closed-form Solution

1. (10%) Show the weights and intercepts of your linear model.

```
Closed-form Solution
Weights: [2.85817945 1.01815987 0.48198413 0.1923993 ], Intercept: -33.78832665744904
```

(40%) Linear Regression Model - Gradient Descent Solution

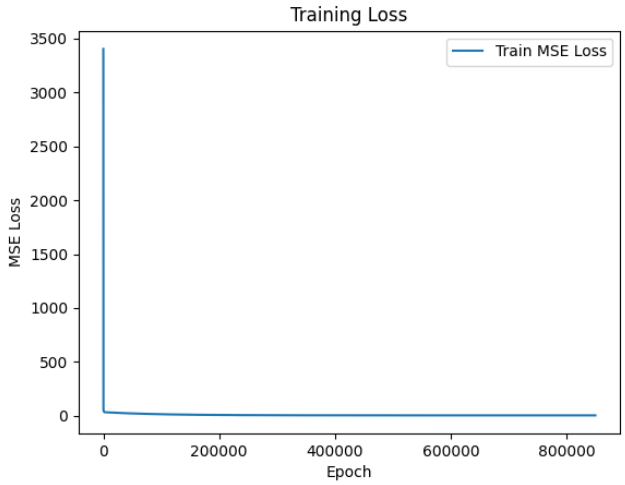
2. (0%) Show the learning rate and epoch (and batch size if you implement mini-batch gradient descent) you choose.

```
LR.gradient_descent_fit(train_x, train_y, lr=0.0001, epochs=850000)
```

3. (10%) Show the weights and intercepts of your linear model.

```
Gradient Descent Solution
Weights: [2.85086581 1.01587004 0.45940509 0.18707053], Intercept: -33.41115292343488
```

4. (10%) Plot the learning curve. (x-axis=epoch, y-axis=training loss)



5. (20%) Show your error rate between your closed-form solution and the gradient descent solution.

```
Error Rate: 0.0%
```

Part. 2, Questions (50%):

1. (10%) How does the value of learning rate impact the training process in gradient descent? Please explain in detail.

The value of learning rate determines the size of the steps in each iteration of gradient descent.

With a smaller learning rate, the algorithm takes smaller steps. It is beneficial for convergence since it avoids making dramatic changes to the model parameters during each iteration. However, if the learning rate is too small, it may lead to slow convergence and thus a longer training time.

In contrast, a larger learning rate causes converging to the minimum point more quickly. But the convergent process isn't that stable due to the huge movement.

Hence, it is important to find an appropriate learning rate that strikes a balance between convergence speed and training efficiency.

2. (10%) There are some cases where gradient descent may fail to converge. Please provide at least two scenarios and explain in detail.

I. Too Large Learning Rate

If the learning rate is set too large, chances are that the steps will exceed the minimum point each iteration. So the algorithm keeps missing the optimal solution every time, causing convergence to fail.

II. Encountering Local Minima

Gradient descent may encounter some points with zero gradients, but they are not necessarily the global minimum. In these cases, the algorithm gets stuck in such local extrema, causing convergence to stop before reaching the optimal solution.

This scenario won't occur in our case.

3. (15%) Is mean square error (MSE) the optimal selection when modeling a simple linear regression model? Describe why MSE is effective for resolving most linear regression problems and list scenarios where MSE may be inappropriate for data modeling, proposing alternative loss functions suitable for linear regression modeling in those cases.

Yes, MSE is a commonly used loss function in linear regression problems.

The advantage of MSE is that, it is a convex function in linear regression. There are no local minima, but only the global one. Thus, gradient descent is guaranteed to converge arbitrarily close to the global minimum effectively. Additionally, since the

MSE's mechanism involves squaring the error, it emphasizes the impact of larger errors. It benefits in making these large errors more significant in the overall loss calculation. However, this also brings drawback in some scenarios that I'll mention in the next part.

MSE may be inappropriate in some scenarios, including:

I. Outliers

MSE is sensitive to outliers because it squares the error. In other words, the error will be amplified, causing us to overestimate the model's badness. However, what we have to do is ignore these outliers and aim to construct a general model.

The alternative for this case is MAE, which won't prone to outliers since its core concept is applying absolute value on the errors. It puts equal weight to all errors so it is less influenced by outliers.

II. Non-Gaussian Errors

Linear regression is based on the assumption of Gaussian distribution, that is, most errors are close to zero, and extreme errors are very rare. When the data doesn't follow a Gaussian distribution, MSE may not produce accurate results.

The alternative is GLM, which is designed to handle a variety of error distributions. We can choose either distribution that best fits the data.

4. (15%) In the lecture, we learned that there is a regularization method for linear regression models to boost the model's performance. (p18 in linear_regression.pdf)

Add a regularization term helps alleviate over-fitting

$$E_D(\mathbf{w}) + \lambda E_W(\mathbf{w})$$

4.1. (5%) Will the use of the regularization term always enhance the model's performance? Choose one of the following options: "Yes, it will always improve," "No, it will always worsen," or "Not necessarily always better or worse."

Not necessarily always better or worse.

4.2. We know that λ is a parameter that should be carefully tuned. Discuss the following situations: (both in 100 words)

4.2.1. (5%) Discuss how the model's performance may be affected when λ is set too small. For example, λ=10^(-100) or λ=0

If λ is set too small, the regularization term becomes insignificant. The model behaves like the original one without

regularization, which is prone to overfitting if there are noise and outliers.

4.2.2. (5%) Discuss how the model's performance may be affected when λ is set too large. For example, λ=1000000 or λ=10^100

If λ is set too large, the error function is dominated by the regularization term. This can lead to excessive shrinkage of the weights, causing the model to underfit the data. That is, the model becomes too simple and cannot capture the patterns in the training data.