NYCU Introduction to Machine Learning, Homework 1

109550198，卜銳凱

Part. 1, Coding (50%):

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: -33.788326657448984
```

Linear Regression Model - Gradient Descent Solution

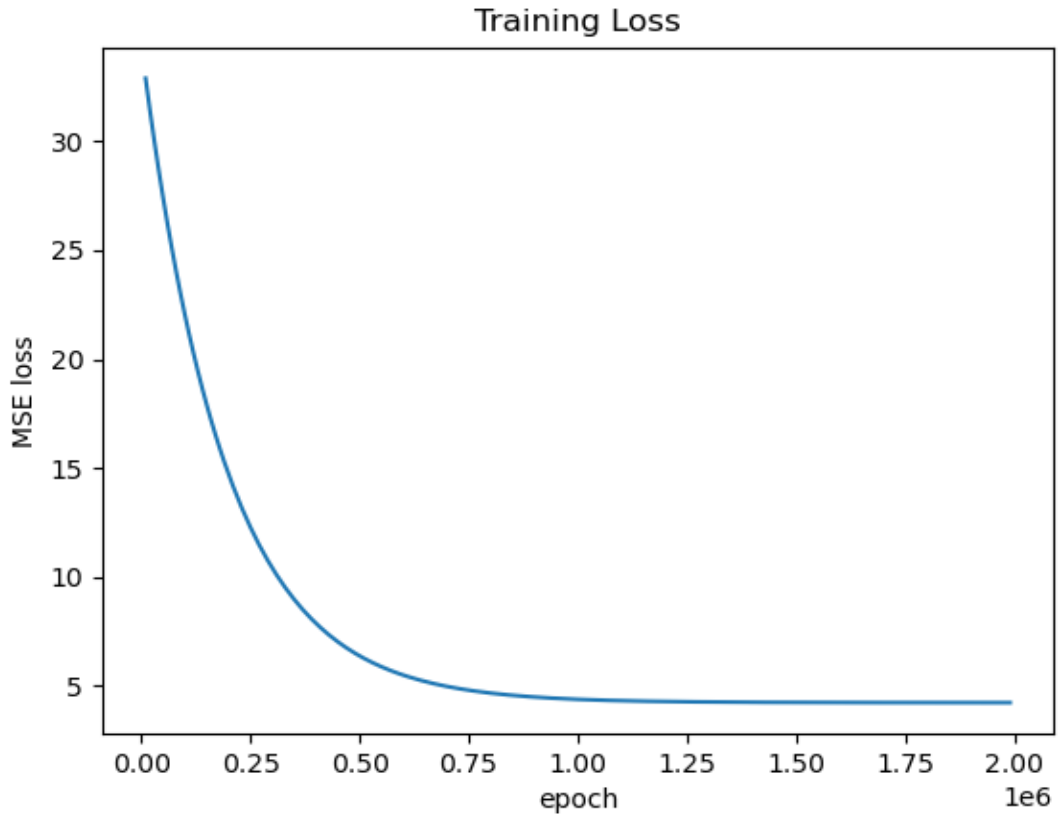
2. (0%) Show the learning rate and epoch (and batch size if you implement mini-batch gradient descent) you choose.

```
LR.gradient_descent_fit(train_x, train_y, lr=0.0001, epochs=2000000)
```

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

```
Gradient Descent Solution
Weights: [2.85487014 1.01712376 0.47176746 0.18998811], Intercept: -33.617661386785386
```

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.

   Write or type your answer here.

   When the learning rate is too large, the speed of learning process will increase, however the model may have problem with convergence. When the learning rate is too small, training not only becomes slower.

   To summarize, selecting a learning rate is a critical step in training machine learning models using gradient descent. It has an effect on convergence speed, stability, and the ability to find the optimal solution. Choosing an optimal learning rate often requires experimentation and is problem-dependent.

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

   Write or type your answer here.

   If the learning rate is too large gradient descent will overshoot the minimum point and finally fail to converge. If the learning rate is too small, the descent will be small and there will be a delayed or no convergence.

   Gradient vanish: when the model is really deep, the gradient maybe near 0 which cause the gradient vanish problem.

   Gradient descent is based on the assumption that the cost function is convex, with a single, well-defined global minimum. In practice, however, cost functions in large machine learning models, particularly deep neural networks, are frequently non-convex and can contain several local minima, saddle points, and plateaus.

   In conclusion, gradient descent may fail to converge due to difficulties with the learning rate and the cost function's nature. To address these issues, effective hyperparameter tuning, adaptive learning rate methods, and more advanced optimization techniques are frequently used.

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.

   Write or type your answer here.

   The Mean Squared Error (MSE) is perhaps the most basic and commonly used loss function for linear regression and simple linear regression.in general, but whether it is the "optimal" solution is dependent on various aspects, including the specific characteristics of your data and your modeling goals. Consider the following:

   Assumptions: MSE assumes that the errors (residuals) are normally distributed with constant variance. It also presupposes a linear relationship between the independent variable(s) and the dependent variable. If these assumptions hold true for your data, MSE is a viable option.

   Minimizing squared errors: When the primary purpose of your modeling effort is to reduce the squared differences between predicted and actual target values, MSE is appropriate. It penalizes larger errors more severely than smaller ones, which is consistent with the general goal of regression: to reduce prediction errors.

Ease of interpretation: MSE has a straight-forward interpretation because it measures the average squared deviation between predicted and actual values. This simplicity might be useful when discussing the performance of your model to stakeholders.

Convexity: MSE is a convex loss function, which means it has a single global minimum. This condition guarantees that optimization algorithms such as gradient descent will converge to a unique solution.

However, there are some scenarios in which MSE may not be the best option:

If our model makes a single incorrect prediction, the squaring function amplifies the error. However, in many cases, we don't worry about these outliers and instead seek for a well-rounded model that performs well enough on the majority. When the data has outliers , it might be more suitable to use L1 loss than MSE because the MSE might have a higher magnitude and the magnitude of the gradient would become higher which might cause larger influence to the model comparing to L1 loss .

In conclusion, while MSE is a good default choice for simple linear regression and frequently works well in practice, it is not always optimal. The loss function should be chosen based on the characteristics of your data, your modeling objectives, and any possible violations of the assumptions behind MSE. It is essential to determine whether MSE corresponds with your objectives and to consider alternate loss functions as necessary.

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)

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

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."

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

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

Write or type your answer here.

Not necessarily always better or worse:

The inclusion of a regularization component in a linear regression model does not ensure improved model performance. It is dependent on the specific characteristics of the dataset and the modeling aims if it improves performance. Regularization is often beneficial for dealing with difficulties such as overfitting; however, if the model is underfitting or the dataset has very little noise, regularization may not be essential and may even worsen performance.

When is set too small (For example, λ=10^ (-100) or λ=0), the regularization term becomes essentially negligible, and the model acts as if there is no regularization at all. This may have the following consequences:

Overfitting: When the model has a low or absent regularization, it is more likely to overfit the training data, capturing noise and fluctuations. The model gets overly complex, resulting in poor generalization on unseen data. The model's performance on the training data may appear to be good in this scenario, but it is likely to perform poorly on new, unseen data.

when λ is set too large. For example, λ=1000000 or λ=10^100), the regularization term dominates the loss function, which might result in the following consequences:

Underfitting: Excessive regularization drives the model coefficients to be near to zero, resulting in a model that is too simple. It may fail to capture important relationships in the data, leading to underfitting. As a result, both the training and test data performance of the model can suffer dramatically.

The choosing of an appropriate λ (regularization strength) is critical in both circumstances. Cross-validation or other hyperparameter tuning techniques can help in determining the best value of that balances the trade-off between bias and variance, resulting in improved model performance on unseen data.