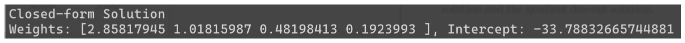
NYCU Introduction to Machine Learning, Homework 1

110550088 李杰穎

1 Part. 1, Coding (50%)

1.1 (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: -33.78832665744881
```

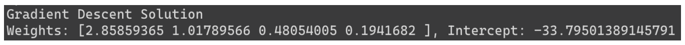
Figure 1: The weights and intercepts given by closed-form solution.

1.2 (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.

   - Learning rate: 2×10^4
   - Epoch: 50
   - Mini-batch size: 10 (with random sampling)

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



```
Gradient Descent Solution
Weights: [2.85859365 1.01789566 0.48054005 0.1941682 ], Intercept: -33.79501389145791
```

Figure 2: The weights and intercepts given by mini-batch gradient descent solution.

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

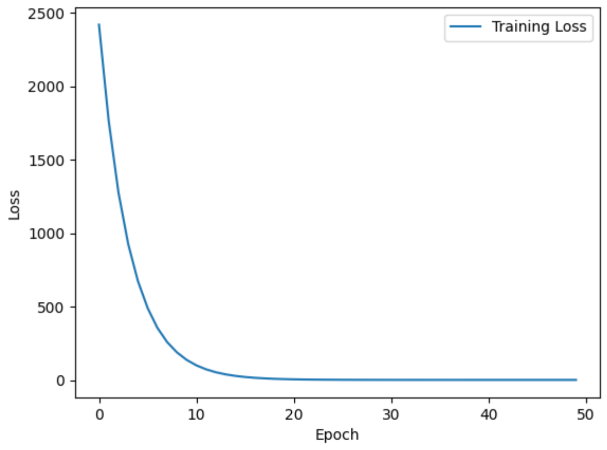
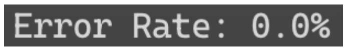


Figure 3: The plot of training loss to epoch.

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



```
Error Rate: 0.0%
```

Figure 4: The error rate between closed-form solution and gradient descent solution.

I think it's worth noticing that I actually normalize the training data before performing gradient descent. I found that the value of "Previous Scores" is much greater than other features, this will make gradient descent unstable, and will take lots of time to converge to optimal solution. Therefore, I normalized each feature by,

$$X_i' = \frac{X_i - \mu_i}{\sigma_i}$$

, where μ_i is the mean of feature i, and σ_i is the standard deviation of feature i. This is actually standardization. With this normalization, we can quickly approach optimal solution in only 50 epochs.

And because the normalization operation is linear, after we get the optimal weights β' and intercepts β'_0 under X', we can convert β' and β'_0 back to β and β_0 under distribution of X by,

$$\beta_i = \frac{\beta'_i}{\sigma_i}$$

and

$$\beta_0 = \beta'_0 - \sum_i \beta'_i \cdot \frac{\mu_i}{\sigma_i}$$

2 Part. 2, Questions (50%)

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

   If the learning rate is too large, then it may cause gradient exploration, made the weights and intercept goes to infinity, leading to divergence. Even if the model doesn't diverge, large learning rate may cause the algorithm missed the minima of loss function, and can't converge to the optimal solution.

   If the learning rate is too small, then it has greater chance to stuck in local minima, which will make us not be able to get the optimal solution. And also, small learning rate leads to longer convergence time.

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

   - Linear regression is not applicable to the data: If the data can not be approximated using linear equation, then the surface of loss function may not have a significant minima. This may make gradient descent fail to converge.
   - Different scale across each feature: If the features is significantly difference in scale (or let's say, the range of value), this will make gradient descent very unstable, so that we can only use a smaller learning rate, but this will arise another problem: slow training. Therefore, normalize the data, as I've done in this homework, before apply gradient descent is very important.

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.

   Let's first talk about the advantages of using MSE as loss function:

   - Differentiable: MSE is differentiable everywhere, therefore, we can use gradient descent method to optimize the parameters. If we're using something like L1 Loss, then it's not differentiable at $x = 0$. Make it a little bit harder to perform gradient descent.
   - Least square solution: In linear algebra, we have a closed-form method to find the least square solution of a linear equation. And least square solution is actually equivalent to optimize MSE loss. Therefore, we can compare the effect of using gradient descent with closed-form solution.

   The disadvantages are:

   - Overfitting: If the data is high-dimensional, then using pure MSE loss may lead to overfitting problem. To deal with this problem, it maybe suitable to introduce the regularization term, as mentioned in class.
   - Outliers: Because of the math characteristic of MSE, it's not very robust to outliers, a very large or small value will make the loss blows up, because of the square term. To deal with the outliers problem, we can use Huber loss. The definition of Huber loss is,

$$L_\delta(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{for } |y - f(x)| \le \delta \\ \delta \cdot \left(|y - f(x)| - \frac{1}{2}\delta\right), & \text{otherwise} \end{cases}$$

.

As we can see, when $|y - f(x)| \leq \delta$, the term is same as MSE. But if $|y - f(x)| > \delta$, it becomes something like L1 Loss. This make Huber loss more robust to the outliers.

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

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

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."

I think it's "Not necessarily always better or worse.".

4.2. We know that λ is a parameter that should be carefully tuned. Discuss the following situations:

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

If λ is set too small, then the regularization term $E_W(\mathbf{w})$ is not significant, make the process of regularization useless.

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

If λ is set too large, then the most significant term in loss function is $E_W(\mathbf{w})$. This will make the gradient descent methods only focus on minimize the regularization term, make the overall model performance very bad.