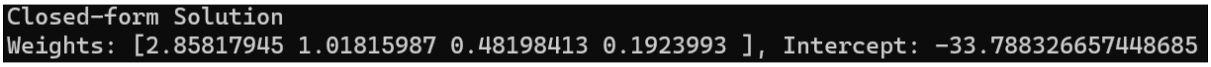
NYCU Introduction to Machine Learning, Homework 1

110652021, 龔大承

Part. 1, Coding (50%):

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

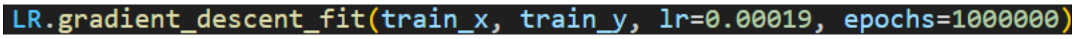
1. (10%) Show the weights and intercepts of your linear model.



```
Closed-form Solution
Weights: [2.85817945 1.01815987 0.48198413 0.1923993 ], Intercept: -33.788326657448685
```

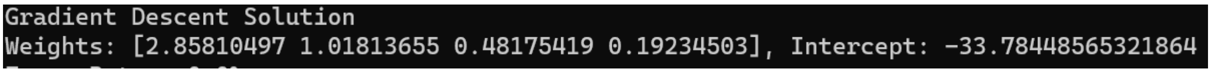
(40%) Linear Regression Model - Gradient Descent Solution

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



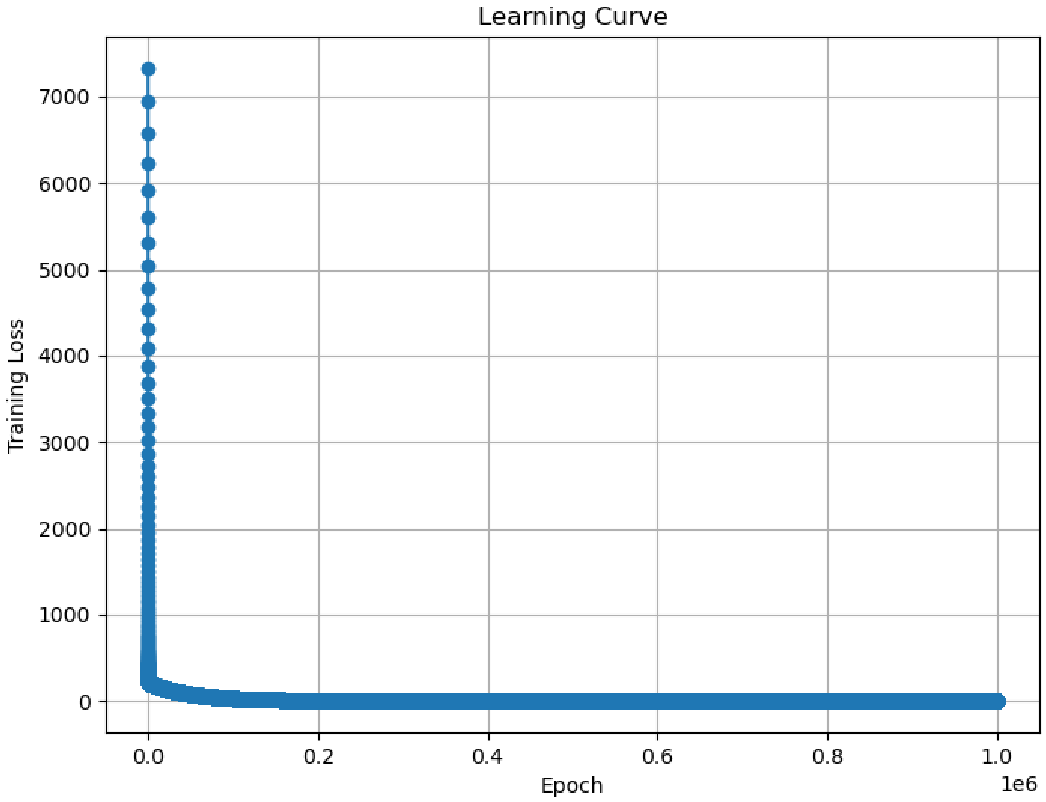
```
LR.gradient_descent_fit(train_x, train_y, lr=0.00019, epochs=1000000)
```

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

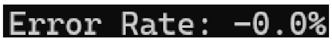


```
Gradient Descent Solution
Weights: [2.85810497 1.01813655 0.48175419 0.19234503], Intercept: -33.78448565321864
```

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 can be considered as the the length of each steps, that is, if the learning rate is very large, then the process of gradient descent algorithm may be diverge or not precise enough. Conversely, if the learning rate is very small, then it will cost much more time to complete the algorithm, moreover, if the learning is not big enough to jump out of the local minimum, the process will stuck and converge at a wrong solution, so the choosing of learning rate should be done carefully.

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

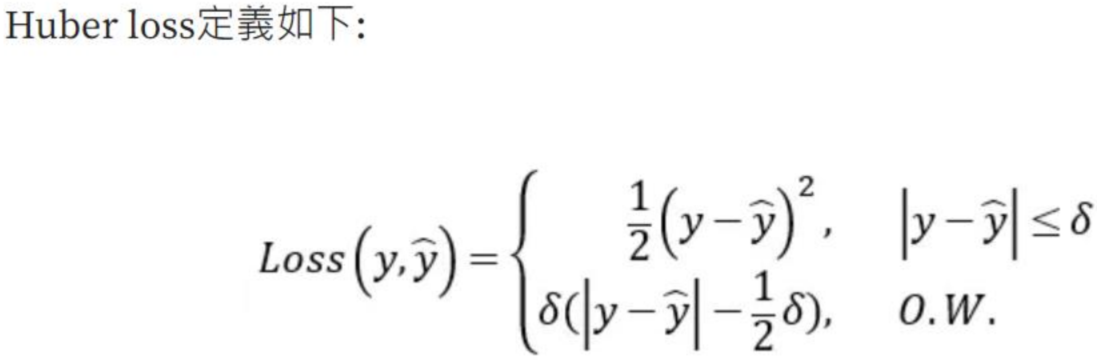
   (1) Epochs too small and learningrate too small:
   If epoches is small(such epochs = 1 ~ 50), in some cases, the gradient descent algorithm may not converge due to there are no sufficient steps to complete the learning.

   (2) Too large learningrate:
   If the learning rate is too large, then the gradient descent process will be diverge or oscillate near the minimum, which lead to a wrong answer.

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.

   (1) MSE is a suitable choice for simple linear regression models in many cases because it penalizes large errors quadratically. This means that MSE is particularly effective when outliers are rare and we want to prioritize small errors. MSE is also suitable for gradient descent since it is differentiable.

(2) Since MSE square the errors, so it is highly sensitive with outlier, to solve this problem Huber loss function can be considered as a solution.



Huber loss定義如下:

$$Loss\left(y,\widehat{y}\right)=\begin{cases}\frac{1}{2}\left(y-\widehat{y}\right)^2, & \left|y-\widehat{y}\right|\le\delta\\ \delta(\left|y-\widehat{y}\right|-\frac{1}{2}\delta), & O.W.\end{cases}$$

MSE assume data are normally distributed, if they are not, then MSE will not produce a correct result, in this case, log-likelihood loss function can be considered.

If the data has varying error variance, weighted MSE can be considered, it assign weight to each data points so that the model can fit the data more accurately.

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."

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 lambda is too small, the regularization term has very little effect so that it doesn’t affect the model overfit.

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 lambda is too large, the regularization term dominate the loss function, which leads to a underfitting, so the model become too simple and will perform poorly on training and validation data.