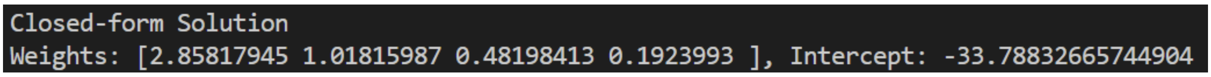
NYCU Introduction to Machine Learning, Homework 1

110550108 施柏江

Part. 1, Coding (50%):

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

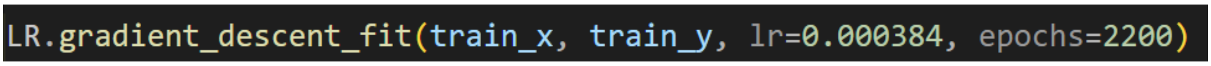
1. (10%) Show the weights and intercepts of your linear model.



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

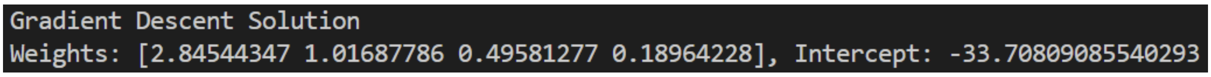
(40%) Linear Regression Model - Gradient Descent Solution

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



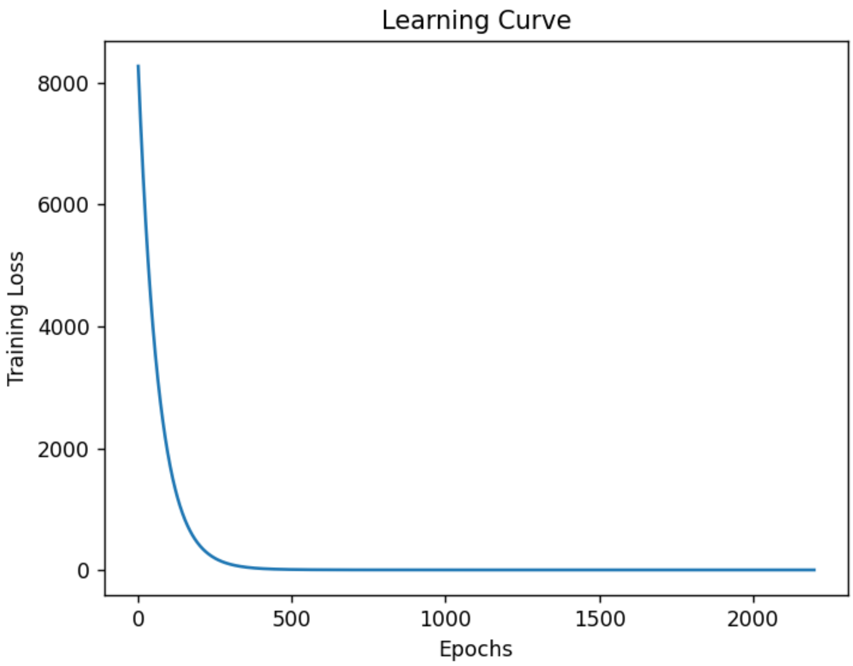
```
LR.gradient_descent_fit(train_x, train_y, lr=0.000384, epochs=2200)
```

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

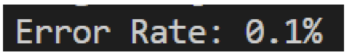


```
Gradient Descent Solution
Weights: [2.84544347 1.01687786 0.49581277 0.18964228], Intercept: -33.70809085540293
```

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: 0.1%
```

Part. 2, Questions (50%):

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

   When the learning rate is set to a small value, the algorithm proceeds cautiously with tiny parameter updates. This results in a smooth convergence process and reduces the risk of overshooting the cost function's minimum. However, the downside is that training can become slow, and the algorithm may become susceptible to getting trapped in local minima. On the other hand, if the learning rate is too large, the algorithm takes bold steps during parameter updates, leading to rapid convergence during the initial training stages. However, it may cause the algorithm to overshoot the minimum, ultimately resulting in divergence.

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

   Gradient descent can face challenges when dealing with cost functions that exhibit significant irregularities, non-convex shapes, and multiple local minima, maxima, and saddle points. These complexities can impede the algorithm's ability to locate the global minimum efficiently. Moreover, the choice of the learning rate plays a pivotal role and must be carefully selected to avoid convergence issues.

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 possesses the property of convexity, indicating it boasts a single global minimum. This characteristic renders it well-suited for gradient descent, ensuring they converge to a singular solution. MSE also exhibits differentiability concerning model parameters, making it compatible with gradient descent. The gradients provide a clear direction for parameter updates.

   However, MSE is sensitive to outliers because it squares the errors. When dealing with datasets containing outliers that require exclusion, MSE can yield suboptimal results.

In such instances, alternative loss functions like Huber loss, which is less sensitive to outliers, may be more appropriate. If we employ linear regression for tasks involving classification, MSE is an unsuitable choice because it does not align with the probabilistic interpretation required for classification. In such instances, it is more customary to use loss functions like logistic loss, which are specifically designed for addressing classification problems.

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

4.1.

Not necessarily always better or worse. Regularization can have varying effects on model performance. In certain scenarios, it can enhance performance by reducing overfitting and enhancing the model's ability to generalize to new data. Conversely, it may not produce a substantial impact or might even degrade performance, especially when overfitting was not a significant issue from the outset or when the regularization term is excessively strong.

4.2.1

When the regularization parameter λ is set to an extremely small or zero value, the regularization term in linear regression becomes negligible compared to the data fitting term. Consequently, the model essentially undergoes training without any regularization constraints. This situation increases the risk of overfitting, as the model can closely fit the training data, potentially capturing noise within it. Consequently, this overfit model may stru ggle to generalize to unseen data, resulting in poor performance on validation or test datasets. Additionally, the absence of regularization makes the model more sensitive to outliers within the training data, as it tries to minimize the squared error term without any constraints.

4.2.2

Setting an excessively large regularization parameter λ strongly penalizes model parameters, driving coefficients close to zero and causing over-simplification. This can lead to underfitting, where the model is too simplistic to capture data patterns, resulting in high bias and poor performance. Additionally, excessive λ may cause the model to discard important features by reducing their coefficients to nearly zero, compromising its ability to learn from the data.