NYCU Introduction to Machine Learning, Homework 1

110612117, 張仲瑜

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: -33.78832665744901
```

(40%) Linear Regression Model - Gradient Descent Solution

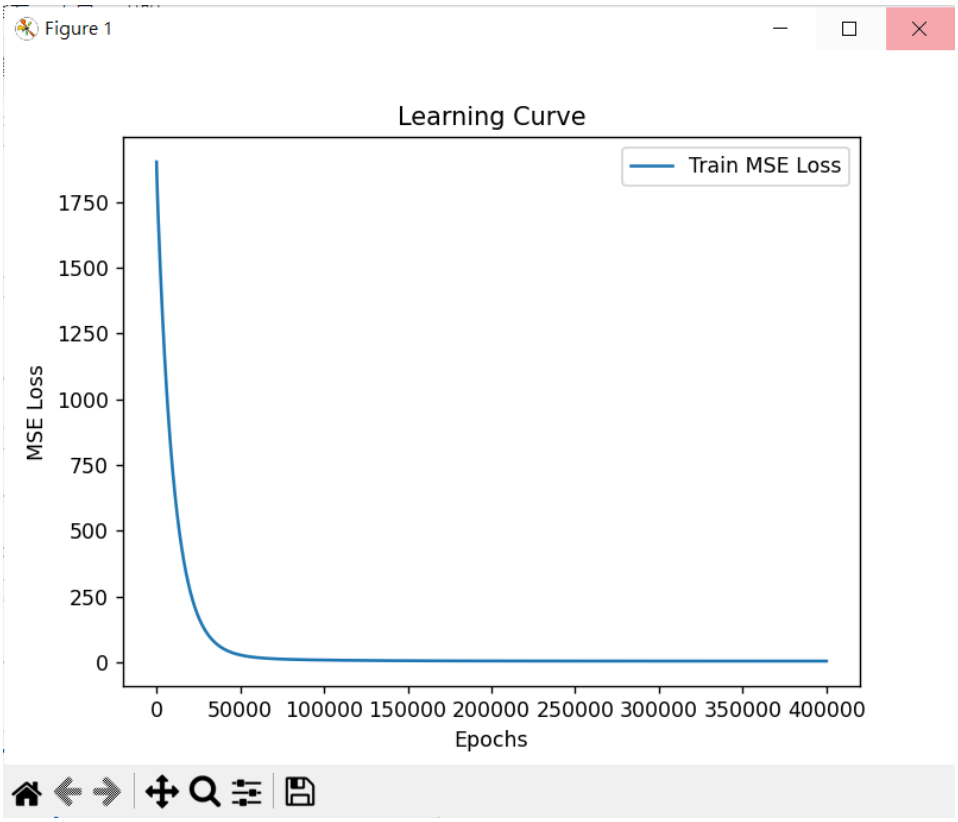
2. (0%) Show the learning rate and epoch (and batch size if you implement mini-batch gradient descent) you choose.

```
LR.gradient_descent_fit(train_x, train_y, lr=0.00019273, epochs=400000)
```

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

```
Gradient Descent Solution
Weights: [2.84706958 1.01468149 0.44768519 0.18430457], Intercept: -33.21537681882141
```

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 i n gradient descent? Please explain in detail.

   Since we update the weight and intercept with
   -learning rate * gradient, learning rate will impact the distance of weigh t movement during the training process. The larger learning rate can take less iterations while stepping further every time; however, overlarge learning rate may lead to divergence while re-crossing the slump of minimum every time due to larger steps. The learning curve of larger learning rate will be more erratic with oscillations. In contrast, smaller learning rate takes more iterations when moving a little every time; nonetheless, over tiny learning rate may cause to the stuck of local minimum since the step is too little to cross the slump. The learning curve of it will be smooth and gradual. In conclusion, selecting a approp riate learning rate is key when performing gradient descent.

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

   (1) The learning rate are too large

   When the learning rate is set too high, gradient descent may fail to converge and even diverge. It happens because the algorithm takes large steps overly during each iteration, overshooting the optimal

solution. As a result, it bounces back and forth across the minimum, and the loss function increases rather than decreases.

(2) Poor parameters initialization

Due to the existence of local minimum, poor parameters initialization may lead to convergence to local minimum instead of global minimum, which makes the result of gradient descent unperfect.

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 fun ctions suitable for linear regression modeling in those cases.

(1) It is one of the optimal selections when modeling a simple linear regression model.

(2) Effective reasons:

a. Convexity: MSE forms a convex loss function, which means it has a single, global minimum. This property ensures that gradient-ba sed optimization methods converge to a unique solution, which is typically the optimal one.

b. Differentiability: MSE is a differentiable loss function, which is crucial for gradient-based optimization algorithms like gradient descent. The gradient of MSE with respect to model parameters can be easily computed, making it suitable for optimization.

c. Statistical Justification: MSE is closely related to the maximum likelihood estimation (MLE) of model parameters under the assumption of normally distributed errors. This statistical foundation provides a strong justification for its use in linear regression.

(3) Scenarios where MSE may be inappropriate and the alternative

a. Due to the formula, MSE is significantly sensitive to the data with large outliers, which will make the update toward that direction, deteriorating the performance. alternative loss functions like Huber loss or Tukey's bisquare loss, which are less sensitive to ou tliers, will be better when this occured.

4. (15%) In the lecture, we learned that there is a regularization method fo r linear regression models to boost the model’ s performance. (p18 in li near_regression.pdf)

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

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

Not necessarily always better or worse

Whether regularization enhances model performance depends o n several factors, including the nature of the data, the presence of multicollinearity, and the choice of the regularization strength

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

When λ is set to an extremely small value or zero, it provides a very weak regularization effect. It's not strong enough to significantly influence the model's parameter estimates. Second, it slightly reduce the variance of the model compared to no regularization, but the difference is typically negligible. In summary, the model behaves similarly to ordinary linear regression. It's prone to overfitting, has high variance, and may not effectively handle multicollinearity or 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

When λ is set to an extremely large value, it dominates the regularization effect, and the model's coefficients are heavily penalized. This leads to excessive bias in the model. The model may underfit the data and perform poorly on both the training and test sets. In sum, it increases bias, reduces variance, and can even lead to feature selection in the case of Lasso, which increase the risk of underfitting.