NYCU Introduction to Machine Learning, Homework 1

110550085房天越

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

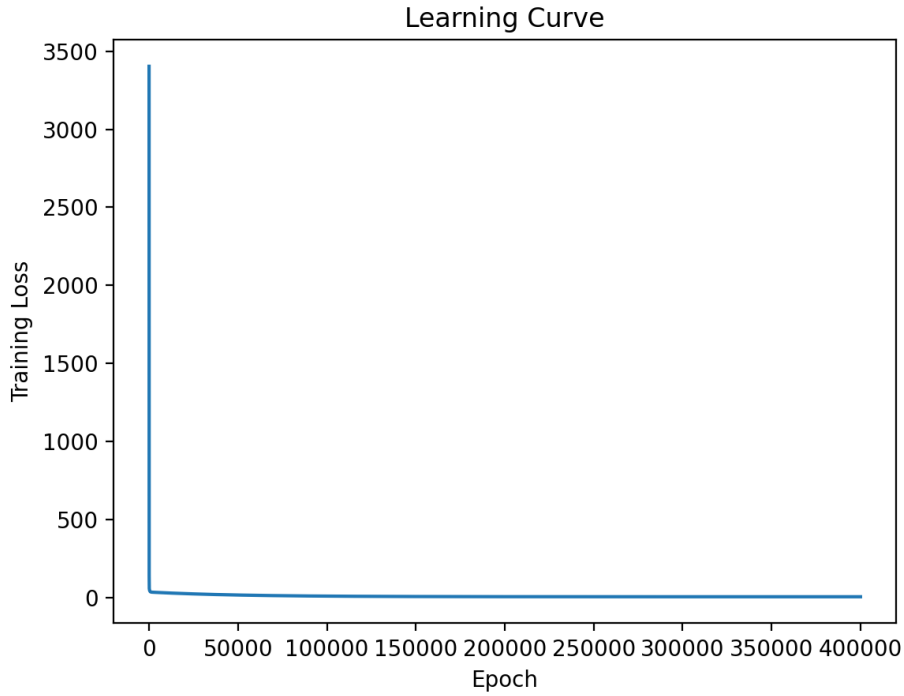
2. (0%) Show the learning rate and epoch (and batch size if you implement mini-batch gradient descent) you choose.

```
LR.gradient_descent_fit(train_x, train_y, lr=0.00019, epochs=400000)
```

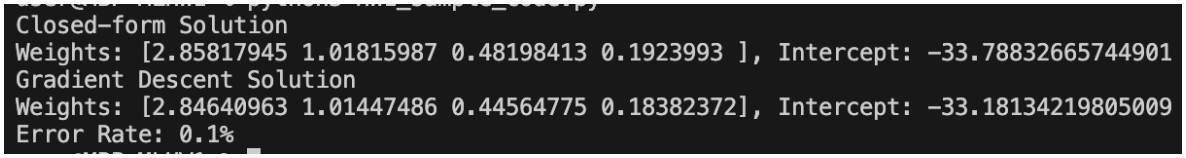
3. (10%) Show the weights and intercepts of your linear model.

```
Gradient Descent Solution
Weights: [2.84640963 1.01447486 0.44564775 0.18382372], Intercept: -33.18134219805009
```

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.



Part. 2, Questions (50%):

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

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E_n$$

According to this equation, we can see that the learning rate plays an important role in gradient descent, if the learning rate is too big, the model may not be able to converge, and if the learning rate is too small, the model may need a lot of epochs to converge, which increases the cost to do that.

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

The first scenario is that the learning rate is too big. When updating the weight vertor, it updates too much and make it farther and farther to the correct solution.

The second scenario is that if the cost function is highly nonlinear and has many local minima, then the model might stuck at a local minima rather than going to the real solution.

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.

Not necessarily, MSE is indeed a good choice when modeling a simple linear regression model, but it is not appropriate to say that it is always “Optimal”.

MSE is effective in most linear regression problems because it is quite simple in math, it is easy to optimize, and it is a nice approach when the problem we want to solve is to minimize the prediction error.

However, there are some situations that might make MSE inappropriate. The first is that there are a lot of outliers, since MSE gives a large weight to large errors, it is very sensitive to outliers. Huber loss might be able to deal with this kind of problem. Another one is that if the data is highly skewed, it may not be the optimal solution because it doesn’t account for asymmetry, we may try to use pinball loss to deal with that problem.

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)

Add a regularization term helps alleviate over-fitting

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

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

Not necessarily always better or worse.

When λ is set too small, the regularization term might be negligible, and the model’s performance would not be improved by this adding term, it would behave hust like the ordinary model. In this situation, the model might be able to fit perfectly to the training data, but deal poorly to the testing data.

When λ is set too big, then the whole model might be dominated by this regularization term, which might result in the inability to capture the underlying terms in the data, and the model become unable to reach the correct solution we want.