NYCU Introduction to Machine Learning, Homework 1

111550151, 徐嘉亨

Part. 1, Coding (60%):

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

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

```
2024-10-01 15:56:09.819 | INFO     | __main__:main:79 - LR_CF.weights=array([2.8491883 , 1.0188675 , 0.48562739, 0.1937254 ]), LR_CF.intercept=-33.82
23
```

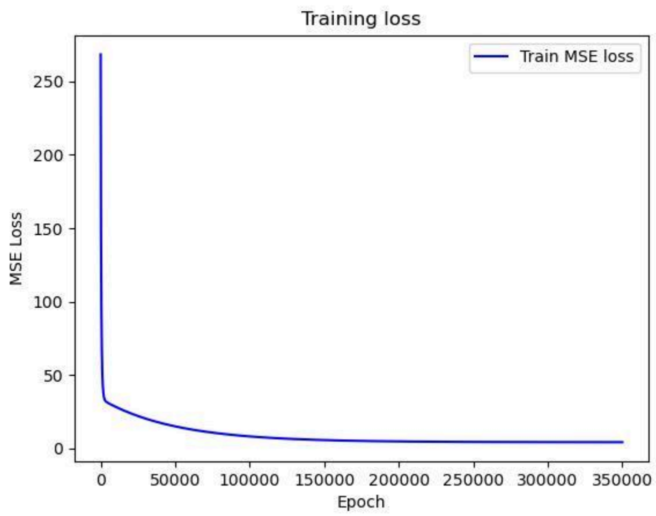
(40%) Linear Regression Model - Gradient Descent Solution

2. (10%)
   - Show the hyperparameters of your setting (e.g., learning rate, number of epochs, batch size, etc.).
   - Show the weights and intercepts of your linear model.

```
LR_GD.fit(train_x, train_y, learning_rate=1.92e-4, epochs=350000)
```

```
2024-10-01 15:56:57.717 | INFO     | __main__:main:84 - LR_GD.weights=array([2.83018421, 1.01290317, 0.4261732 , 0.18010143]), LR_GD.intercept=-32.83
56
```

3. (10%) Plot the learning curve. (x-axis=epoch, y-axis=training loss)



4. (20%) Show your MSE.cf, MSE.gd, and error rate between your closed-form solution and the gradient descent solution.

```
2024-10-01 15:56:57.719 | INFO     | __main__:main:93 - Mean prediction difference: 0.1317
2024-10-01 15:56:57.719 | INFO     | __main__:main:98 - mse_cf=4.1997, mse_gd=4.2101. Difference: 0.246%
```

(10%) Code Check and Verification

5. (10%) Lint the code and show the PyTest results.

```
============================================================== test session starts ==============================================================
platform linux -- Python 3.9.19, pytest-7.4.4, pluggy-1.0.0
rootdir: /home/jxea666/Intro2ML/hw1
collected 2 items

test_main.py 2024-10-01 16:00:26.390 | INFO     | test_main:test_regression_cf:27 - model.weights=array([3.]), model.intercept=4.0000000000000036
.2024-10-01 16:00:27.514 | INFO     | test_main:test_regression_gd:39 - model.weights=array([3.]), model.intercept=3.999996478786354
.

=============================================================== 2 passed in 1.42s ===============================================================
```

Part. 2, Questions (40%):

1. (10%) How does the presence of outliers affect the performance of a linear regression model? How should outliers be handled? <u>List at least two methods</u>.

   Outliers can lead to a biased estimate of slope and intercept. This results in a poor fit for the majority of data points, reducing the overall accuracy of the model. For the first method to handle outliers, it can be simple. We can just detect the outliers and remove them from the training data. For the second method, we can use logarithmic to transform the training data. It can reduce the effect of outliers on the model.

2. (15%) How do different values of learning rate (too large, too small…) affect the convergence of optimization? <u>Please explain in detail</u>.

   If the learning rate is too small, there are some possible cases. First, it might finally converge but it will take a long time. Second, it might lead to the gradient stuck in the local minimum of the loss function. On the other hand, if the learning rate is too large, the parameter might take a too large step in the update, and make the loss larger..

3. (15%)
   - What is the prior, likelihood, and posterior in Bayesian linear regression. [Explain the concept in detail rather than writing out the mathematical formula.]
   - What is the difference between Maximum Likelihood Estimation (MLE) and Maximum A Posteriori Estimation (MAP)? (Analyze the assumptions and the results.)

   Prior is the initial belief of the assumption about the coefficients and intercept of the linear regression model before training. It is a probability distribution of the uncertainty you have about these parameters based on prior knowledge or assumptions. Likelihood is the probability of the observed data under a specific set of parameters of a model. It tells us how well the model can explain the observed data with its parameters. Posterior is the belief of the parameters of the model after training. It is a combination of both the prior and likelihood, and represents the probability of the parameters under observed data.

   In the assumption part, MLE doesn’t consider the prior information about the parameters, and the goal of MLE is to find the parameters that maximize the likelihood of the observed data. However, MAP considers the prior information of the parameters, and its goal is to find the parameters that maximize the posterior distribution. In the result part, since MLE doesn't consider the prior information of the parameters, which means it only relies on the observed data. It might have high variance and bias on the estimate. However, MAP considers the prior information of the parameters, which means it not only relies on the observed data. It can still enjoy lower variance and bias and more stable estimate under noisy data.