NYCU Introduction to Machine Learning, Homework 1

111550088, 張育維

Part. 1, Coding (60%):

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

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

```
2024-10-04 19:40:32.503 | INFO     | __main__:main:88 - LR_CF.weights=array([2.8491883 , 1.0188675 , 0.48562739, 0.1937254 ]), LR_CF.intercept=-33.8
223
```

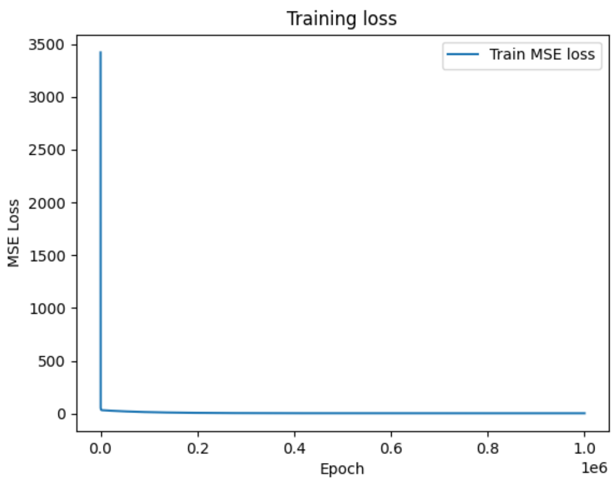
(40%) Linear Regression Model - Gradient Descent Solution

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.

```
losses = LR_GD.fit(train_x, train_y, learning_rate=1e-4, epochs=1000000)
```

```
2024-10-04 19:41:08.354 | INFO     | __main__:main:93 - LR_GD.weights=array([2.84575631, 1.01779039, 0.47489042, 0.19126502]), LR_GD.intercept=-33.6
441
```

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.

```
2024-10-04 19:48:09.541 | INFO     | __main__:main:103 - Mean prediction difference: 0.0238
2024-10-04 19:48:09.542 | INFO     | __main__:main:108 - mse_cf=4.1997, mse_gd=4.1978. Difference: 0.047%
```

(10%) Code Check and Verification

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

```
(machine_learning) D:\user\Desktop\for_school\machine_learning\hw1>pytest ./test_main.py -s
============================================================== test session starts ==============================================================
platform win32 -- Python 3.11.9, pytest-8.3.3, pluggy-1.5.0
rootdir: D:\user\Desktop\for_school\machine_learning\hw1
collected 2 items

test_main.py 2024-10-04 20:26:01.383 | INFO     | test_main:test_regression_cf:27 - model.weights=array([[3.]]), model.intercept=array([4.])
.2024-10-04 20:26:02.099 | INFO     | test_main:test_regression_gd:39 - model.weights=array([3.]), model.intercept=np.float64(3.9999966785390386)
.

=============================================================== 2 passed in 2.41s ===============================================================
```

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 distort the regression line because they have large residuals, which will increase the sum of squared residuals. Additionally, they can mislead the estimation of the coefficients, reducing the model's predictive power. To handle outliers, we can apply several methods. One approach is to set a threshold to identify outliers and limit their values to a specified maximum or minimum. We can also apply a log transformation, which can reduce the impact of outliers by making the data more symmetric.

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

   Too large: When the learning rate is too large, the algorithm may overshoot the minimum, causing the loss to increase instead of decrease. This can result in oscillations, preventing the model from converging to a solution. Additionally, a large learning rate can make the algorithm unstable, causing the optimization path to bounce around the minimum and exhibit erratic behavior. Therefore, if the learning rate is too large, it may cause oscillations or even lead to divergence.

   Too small: When the learning rate is too small, the algorithm will converge less efficiently because it requires more iterations (epochs) to approach the minimum, which wastes computational resources. Furthermore, if the learning rate is too small, the model may get stuck in a local minimum, making it difficult to find the global minimum.

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: The prior represents our beliefs or knowledge about the parameters of the model before observing any data, such as the regression coefficients (weights). Priors allow us to incorporate external knowledge into the model.

Likelihood: The likelihood represents how probable the observed data is, given specific values of the model parameters. In Bayesian linear regression, it quantifies how well the model explains the observed data based on those parameters.

Posterior: The posterior combines the prior and the likelihood to provide a new distribution that reflects the model parameters after observing the data. According to Bayes' Theorem, it is proportional to the product of the prior and the likelihood, meaning it reflects both our prior beliefs and how well the observed data supports those beliefs. In summary, the posterior is the result we use to make more informed predictions.

Assumptions: MLE assumes that we have no prior knowledge about the parameters and relies purely on the observed data. In other words, it focuses solely on the likelihood. MAP, on the other hand, incorporates prior information, using both the likelihood of the data and the prior distribution, applying Bayes' Theorem to find the new distribution.

Results: MLE does not account for prior information, so the resulting estimate is based entirely on the data. However, because it maximizes the likelihood of the data, MLE may overfit when data is sparse or noisy, typically requiring large datasets to perform well. MAP, on the other hand, regularizes the results through the use of priors, which can help prevent overfitting in noisy or insufficient data. However, if the dataset is too small, MAP results can be biased due to the strong influence of the prior. Like MLE, MAP also benefits from large datasets to ensure convergence.