NYCU Introduction to Machine Learning, Homework 1

109700046, 侯均頲

The screenshot and the figures we provided below are just examples. **The results below are not guaranteed to be correct.** Please make sure your answers are clear and readable, or no points will be given. Please also remember to convert it to a pdf file before submission. You should use English to answer the questions. After reading this paragraph, you can delete this paragraph.

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

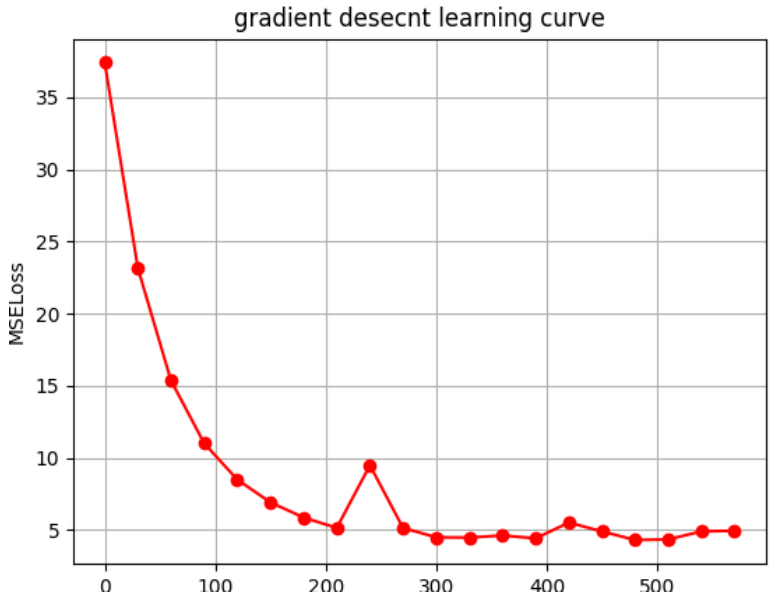
2. (0%) Show the learning rate and epoch (and batch size if you implement mini-batch gradient descent) you choose.

```
LR.gradient_descent_fit(train_x, train_y, lr=0.00002, epochs=600,
```

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

```
Gradient Descent Solution
Weights: [2.83615046 1.01729624 0.477912   0.19571287], Intercept: -33.57690290073857
```

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: -0.2%
```

Part. 2, Questions (50%):

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

   If the learning rate is too large, the weights may sometimes diverge and become excessively large. This can lead to results that significantly differ from the closed-form solution and may even result in overflow issues. Conversely, if the learning rate is too small, the gradual steps may slow down the loss reduction process, even though the direction is correct. In such cases, the final weights may remain far from the closed-form solution.

   In summary, we adjust the learning rate based on our experience, continuously experimenting and learning from errors.

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

   imagine that all possible weights are like a muddy road, and we aim to find the lowest point on this road by taking steps with our feet. If every step is a big stride, we might skip over a very deep hole and end up going far far away, which can be viewed as divergence.

Noise in the data can also hinder convergence. Imagine that we repeatedly select a bad dat a point with a large value in SGD. This can cause the direction to go to the wrong path rat her than the correct path, leading to a failure to converge.

Our loss function may feature numerous local minima. Due to inappropriate initial weights or learning rates, we may easily converge to one of these local minima, but not the global minimum.

3. (15%) Is mean square error (MSE) the optimal selection when modeling a s imple linear regression model? Describe why MSE is effective for resolving most linear regression problems and list scenarios where MSE may be ina ppropriate for data modeling, proposing alternative loss functions suitable for linear regression modeling in those cases.

Yes it is, it is effective because MSE function won't diverge, and minimizing MSE is sa me meaning as maiximizing the likelihood of our weights and observed data,
because MSE is the square of units of different between target value and prediction, it is in terpretable in the same units as the target. What's more MSE is differentiable, so it is sui table for problems related to computing gradients, linear regression is one of them.

inappropriate scenarios:
1. Heteroscedasticity, the error is not constant in those data
-> we use weighted least squares, which assign different weights to those data base on th eir variances when calculating the loss.
2. data with some outliers
-> Huber loss or Tukey's bisquare loss can down weight the impact of outliers.

4. (15%) In the lecture, we learned that there is a regularization method for li near regression models to boost the model's performance. (p18 in linear_r egression.pdf)

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

4.1. (5%) Will the use of the regularization term always enhance the mod el's performance? Choose one of the following options: "Yes, it will a

lways improve," "No, it will always worsen," or "Not necessarily alwa ys better or worse."

4.2. We know that λ is a parameter that should be carefully tuned. Disc uss 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, when the real weights are all huge numbers w ith high dimensions, we are not overfitting the model as we thought we are, In this case th e regularization term is decreasing the model's performance. However, in normal cases, the regularization term can enhance our model' s performance.

4.2.1 If λ is set too small ,we can consider the regularization term like a normal MSE loss but without divided by n, and this case often cause overfitting, the loss of the trained data can be v ery low after training, while the loss of the test data can be extremely large due to overfitting. And our final weights can be large numbers, far from the correct weights.

4.2.2 If λ is set too large, than we are just indicating that we want our weights as closer t o zero as possible, and we' ll finally get a bunches of numbers near zero, the final training loss Should still be high, let alone testing loss. This case can be seem as underfitting, Both the final per formance of training and testing data are poor.