NYCU Introduction to Machine Learning, Homework 1

111550129, 林彥亨

Part. 1, Coding (60%):

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

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

```
2025-09-28 23:36:27.214 | INFO     | __main__:main:124 - LR_CF.weights=array([2.85501274, 1.01785863, 0.47202
168, 0.19608925]), LR_CF.intercept=-33.6956
```

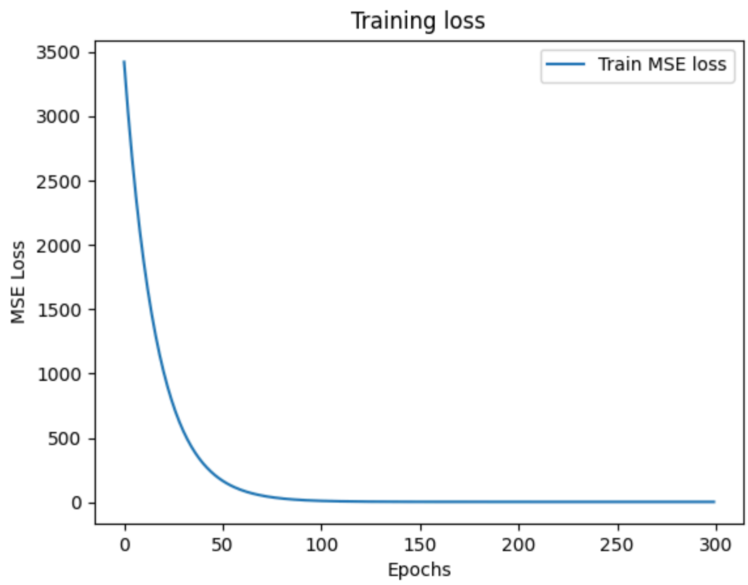
(40%) Linear Regression Model - Gradient Descent Solution

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=0.015, epochs=300)
```

```
2025-09-28 23:36:27.243 | INFO     | __main__:main:134 - LR_GD.weights=array([2.85456156, 1.01773842, 0.47207
595, 0.19619369]), LR_GD.intercept=-33.6918
```

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.

```
2025-09-28 23:42:32.025 | INFO     | __main__:main:150 - Prediction difference: 0.0059
2025-09-28 23:42:32.026 | INFO     | __main__:main:155 - mse_cf=4.2903, mse_gd=4.2905. Difference: 0.004%
```

(10%) Code Check and Verification

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

```
(ml) heng@heng:~/Desktop/ml_hw1$ flake8 main.py
(ml) heng@heng:~/Desktop/ml_hw1$
```

```
● (ml) heng@heng:~/Desktop/ml_hw1$ pytest ./test_main.py -s
============================================ test session starts ============================================
platform linux -- Python 3.11.13, pytest-8.4.2, pluggy-1.6.0
rootdir: /home/heng/Desktop/ml_hw1
collected 2 items

test_main.py (100, 1)
2025-09-28 23:45:56.782 | INFO     | test_main:test_regression_cf:30 - model.weights=array([[3.]]), model.inter
cept=array([4.])
.(100, 1)
2025-09-28 23:45:56.783 | INFO     | main:fit:92 - EPOCH 0, loss=30622.0606, lr=0.0001
2025-09-28 23:45:56.875 | INFO     | main:fit:92 - EPOCH 10000, loss=560.6383, lr=0.0001
2025-09-28 23:45:56.965 | INFO     | main:fit:92 - EPOCH 20000, loss=10.2643, lr=0.0001
2025-09-28 23:45:57.054 | INFO     | main:fit:92 - EPOCH 30000, loss=0.1879, lr=0.0001
2025-09-28 23:45:57.144 | INFO     | main:fit:92 - EPOCH 40000, loss=0.0034, lr=0.0001
2025-09-28 23:45:57.233 | INFO     | main:fit:92 - EPOCH 50000, loss=0.0001, lr=0.0001
2025-09-28 23:45:57.323 | INFO     | main:fit:92 - EPOCH 60000, loss=0.0000, lr=0.0001
2025-09-28 23:45:57.413 | INFO     | test_main:test_regression_gd:43 - model.weights=array([2.99999751]), model
.intercept=np.float64(3.999996678539041)
.

============================================= 2 passed in 1.00s =============================================
```

Part. 2, Questions (40%):

1. (10%) Linear models $y = w^{\top} x + b$ have limited fitting power.
    a. In one sentence, explain why a single linear model is limited.
    b. Give one concrete task that a single linear model cannot solve, and state why no single hyperplane/affine function solves it.

    (a) Because it can only denote in straight-line relationships and cannot represent complex nonlinear patterns in the data.

    (b) We use the function sin(2πx) to generate the training data, since the sine function wave is nonlinear and periodically changed. However, the single hyperplane/affine function cannot bend and non-monotonic, so it cannot solve the sine function well. Because the sine function has a lot of slope

2. (15%) Why do we add a regularization term in linear regression? What are the differences between L2 regularization (Ridge) and L1 regularization (Lasso)? <u>Please explain in detail</u>.

    (1) Because root mean squares can overfit (high variance). Adding a penalty on the coefficients shrinks them toward zero, which can reduce variance.

    (2)

	Ridge	Lasso
Penalty term	$\lambda \sum_{j} \omega_{j}^{2}$	$\lambda \sum_{j} \lvert \omega_{j} \rvert$
Character	Smooth shrinkage and usually lower prediction error when signals are dense.	Cut variance via variable selection if the true signal is sparse.

Solutions	has a closed form	has no closed form
In Bayesian view	Equivalent to assuming a Gaussian prior on weights.	Equivalent to assuming a Laplace prior on weights.

3. (15%)

- What is overfitting? Under what conditions can a model overfit? (List two) How can overfitting be alleviated? (List two)

(1) The model has well fitting in the training data, but has poor fitting in testing.
(2) i. The model is too complex relative to data. For example, high-degree polynomial regression with only a few data points.
ii. Training data is too limited or noisy. With very few samples or lots of random noise, the model picks up the patterns that don't generalize.
(3) Regularization and Cross-validation.