NYCU Introduction to Machine Learning, Homework 1

112550077, 劉逢穎

Part. 1, Coding (60%):

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

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

```
2025-10-04 02:51:22.319 | INFO     | __main__:main:98 - LR_CF.weights=array([  2.85501274,   1.01785863,   0.47202168
,   0.19608925,
       -33.69559829]), LR_CF.intercept=-33.6956
```

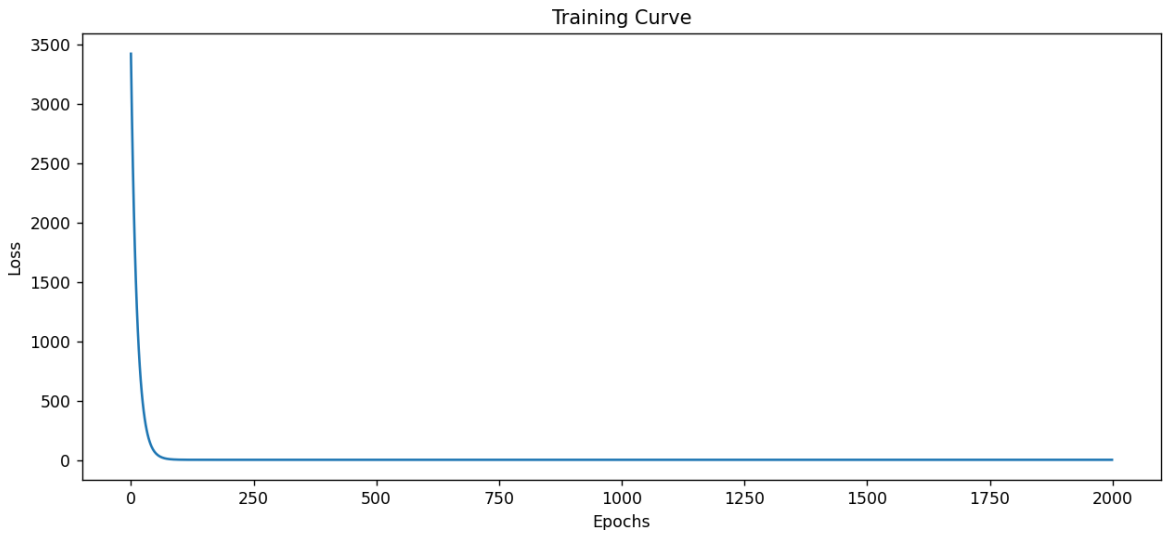
(40%) Linear Regression Model - Gradient Descent Solution

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

```
LR_GD = LinearRegressionGradientdescent()
losses = LR_GD.fit(train_x, train_y, learning_rate=0.02, epochs=2000)
```

```
2025-10-04 02:51:22.624 | INFO     | __main__:main:107 - LR_GD.weights=array([2.85501274, 1.01785863, 0.47202168, 0.1
9608925]), LR_GD.intercept=-33.6956
```

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.

```
2025-10-04 02:51:24.785 | INFO     | __main__:main:124 - Prediction difference: 0.0000
2025-10-04 02:51:24.787 | INFO     | __main__:main:129 - mse_cf=4.2903, mse_gd=4.2903. Difference: 0.000%
```

(10%) Code Check and Verification

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

```
PS D:\ML\hw1> flake8 main.py
PS D:\ML\hw1>
```

```
PS D:\ML\hw1> pytest ./test_main.py -s
========================================== test session starts ==========================================
platform win32 -- Python 3.8.10, pytest-8.3.5, pluggy-1.5.0
rootdir: D:\ML\hw1
collected 2 items

test_main.py (100, 1)
2025-10-04 02:56:49.474 | INFO     | test_main:test_regression_cf:30 - model.weights=array([[3.],
       [4.]]), model.intercept=array([4.])
.(100, 1)
2025-10-04 02:56:49.480 | INFO     | main:fit:68 - EPOCH 0, loss=30622.0606, learning_rate=0.0200
2025-10-04 02:56:49.527 | INFO     | main:fit:68 - EPOCH 1000, loss=0.0000, learning_rate=0.0200
2025-10-04 02:56:49.574 | INFO     | test_main:test_regression_gd:44 - model.weights=array([3.]), model.intercept=3.9
99999999999992
.

========================================== 2 passed in 1.90s ==========================================
```

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 si ngle hyperplane/affine function solves it.

   a. because it can only represent linear relationships and cannot capture complex nonlinear f eatures in data.
   b. For example, classifying data arranged in concentric circles cannot be solved by a single linear model, since no straight line can separate the classes.

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

   We add a regularization term to prevent overfitting by penalizing large weights.
   L2 regularization adds the sum of squared weights to the loss. It shrink weights smoothly t oward zero but rarely making them exactly zero.
   L1 regularization adds the sum of absolute weights to the loss. In addition to preventing lar ge weights, it can also perform feature selection, meaning it allows some weights to be exa ctly zero.

3. (15%)

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

  Overfitting occurs when a model fits the training data too closely, leading to low training e rror but high testing error.

Conditions under which overfitting can occur:

1. The model is too complex relative to the amount of training data.
2. The training data is too small.

Ways to alleviate overfitting:

1. Add regularization
2. Use more training data or apply data augmentation.