hw1_report

Regression equation in the basic part

1. Matrix Inversion

$$\text{training\_bias} = \begin{bmatrix} 1 & x_1^1 & x_2^1 & \dots & x_m^1 \\ 1 & x_1^2 & x_2^2 & \dots & x_m^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^n & x_2^n & \dots & x_m^n \end{bmatrix}$$

$$\text{training\_bias\_inverse} = \left(\text{training\_bias}^T \cdot \text{training\_bias}\right)^{-1} \cdot \text{training\_bias}^T$$

$$\text{weights} = \text{training\_bias\_inverse} \cdot \text{validation\_dataset}$$

1. Gradient Descent

$$\text{training\_bias} = \begin{bmatrix} 1 & x_1^1 & x_2^1 & \dots & x_m^1 \\ 1 & x_1^2 & x_2^2 & \dots & x_m^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^n & x_2^n & \dots & x_m^n \end{bmatrix}$$

$$\text{training\_bias\_inverse} = \left(\text{training\_bias}^T \cdot \text{training\_bias}\right)^{-1} \cdot \text{training\_bias}^T$$

$$\text{weights} = \text{training\_bias\_inverse} \cdot \text{validation\_dataset}$$

Variables used in the advanced part (Gradient Descent)

```
training_dataset_adv = [] # dataset for features
validation_dataset_adv = [] # dataset for labels
testing_dataset_adv = []
trained_weights_adv = []
trained_bias_adv = 0

learning_rate_adv = 0.00001
num_iteration_adv = 100000
```

- 1 feature in basic training_dataset / 4 features in advanced training_dataset
- 1 feature in basic testing_dataset / 4 features in advanced testing_dataset
- 1 coefficients in basic weights / 4 coefficients in advanced weights
- learning rate in basic part: 0.0001 / learning rate in advanced part: 0.00001
- number of iterations in basic part: 50,000 / number of iterations in advanced part: 100,000

Difficulty encountered

- solve the combination of learning rate and number of iterations
  - solution: just try...
- I would like to change the data type in numpy array from numpy.str_ to float each by each element, i.e. element[i][j] = float(element[i][j]), but failed.
  - solution: need to change the data type of the container but not elements, i.e. element = elements.astype(float)
- the shape of weights and datasets doesn't match so that I cannot use dot function
  - solution: reshape their size

Reflections

- learn more abt numpy