



Introduction to Machine Learning

Homework 1 Announcement

Presenter: TA Jui-Che (Ben)
Lastest update: 2023/09/27 13:00

Homework 1

- Deadline: 23:59, Oct. 10th (Tue), 2023
- Coding (50%): Implement linear regression by only using **numpy**.
 - Submit your python file (.py).
 - Answer the questions (by screenshots) in the report (.pdf).
- Handwritten Questions (50%): Answer questions about linear regression.
 - Answer the questions (handwritten, typed, digital, etc.) in the report.

Links

- Questions: [Link](#)
- Sample code: [Link](#)
- Dataset: [Link](#)
- Report template: [Link](#)

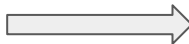
Environment

- Python version: 3.8 or newer
- Tips
 - We recommend that you use **virtual environments** when implementing your homework assignments.
 - Here are some popular virtual environment management tools:
 - [Conda](#)
 - [Miniconda](#)
 - [virtualenv](#)

Numpy

- Build-in array operations.
- Numpy Tutorial: [Link](#)

```
a = np.array([1, 2, 3])  
b = np.array([4, 5, 6])  
for i in range(a.shape[0]):  
    a[i] *= b[i]  
print(a)  
# a = [ 4 10 18]
```



```
a = np.array([1, 2, 3])  
b = np.array([4, 5, 6])  
a *= b  
print(a)  
# a = [ 4 10 18]
```

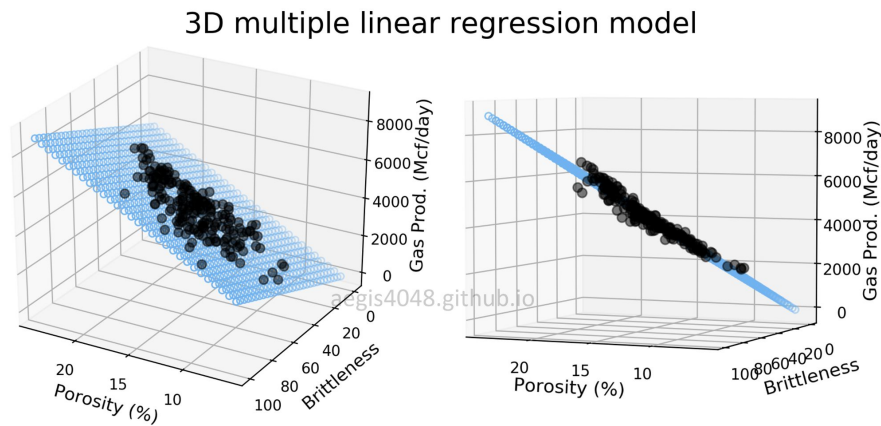
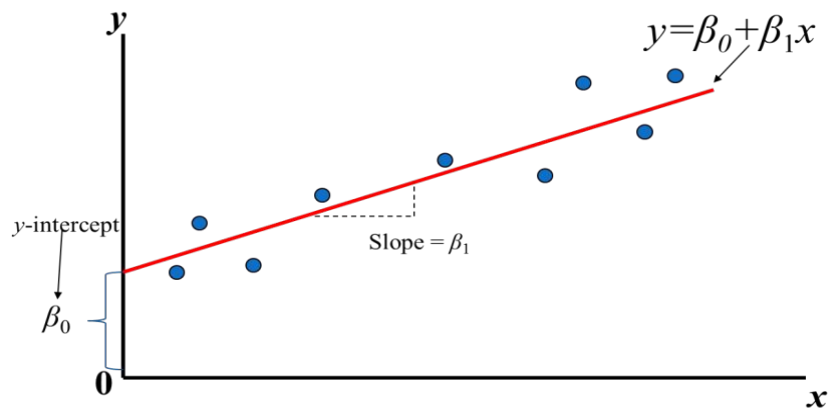
```
import math  
a = np.array([1, 4, 9])  
for i in range(a.shape[0]):  
    a[i] = math.sqrt(a[i])  
print(a)  
# a = [1 2 3]
```



```
a = np.array([1, 4, 9])  
a = np.sqrt(a)  
print(a)  
# a = [1 2 3]
```

Linear Regression

- Find the best value of the weights and the intercept



How to find β_0 and β_1 ?

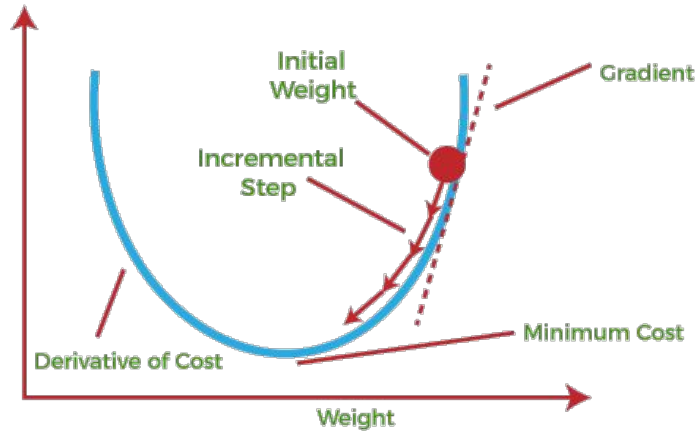
- Closed-form solution

$$\hat{\beta} = (X^T.X)^{-1}X^T.Y$$

- How about a large dataset?
 - high dimensional data
 - huge amount of data

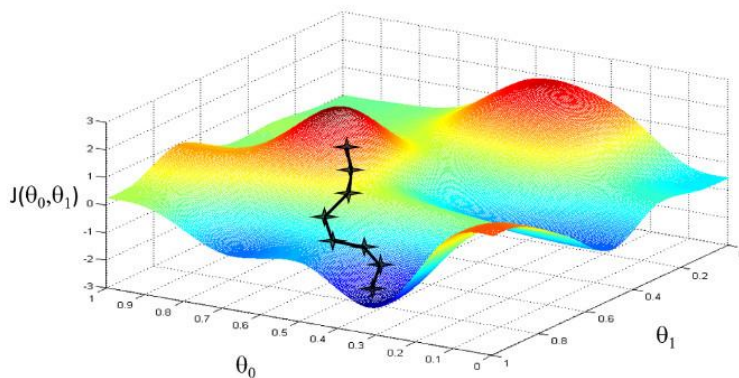
How to find β_0 and β_1 ?

- Gradient Descent



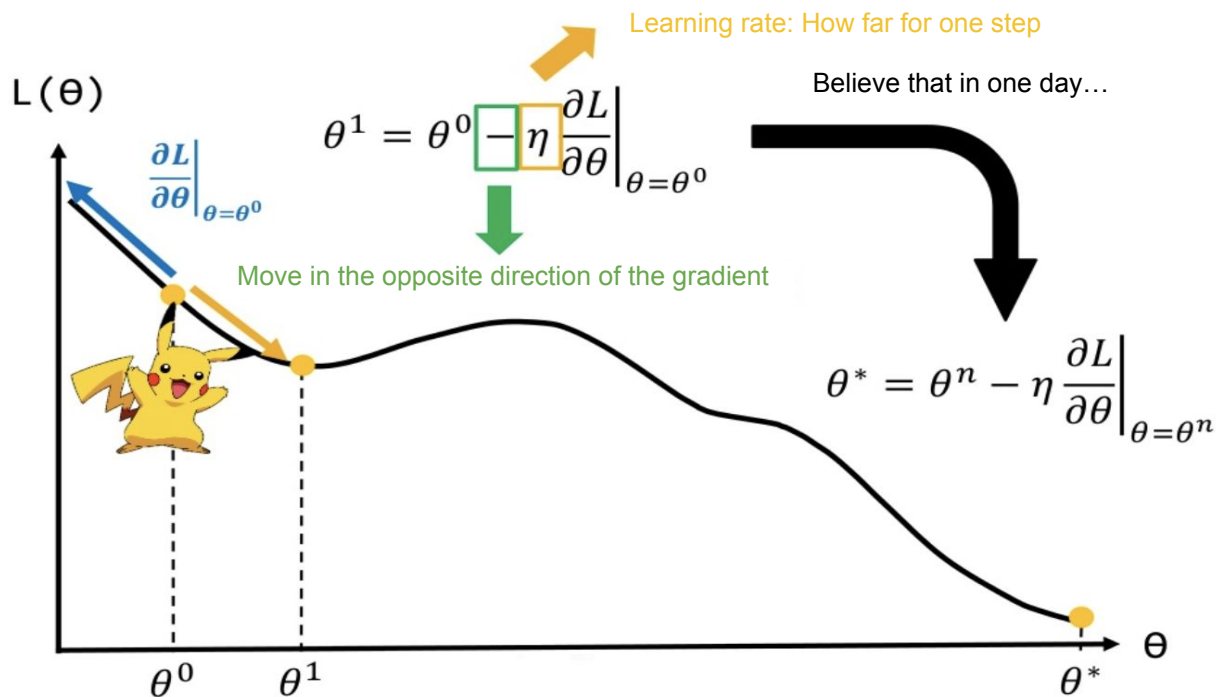
Gradient Descent

- x-axis and y-axis: the value of **weights**
- z-axis: the **value of loss** of the corresponding weights
- Goal: Find the weights that **minimize** the value of loss



Gradient Descent

- Learning rate



Dataset

- Student Performance Dataset
- Features
 - Hour Studied
 - Previous Score
 - Sleep Hours
 - Sample Question Papers Practiced
- Target
 - Performance Index (higher means better performance)

Linear Regression – Closed-form Solution

- Requirements
 - Implement Linear Regression by closed-form solution.
- Grading Criteria
 - (10%) Show the weights and intercepts of your linear model.
- Tips
 - There is only one answer, You can check your answer by yourself using third-party libraries (such as scikit-learn).

Linear Regression – Gradient Descent

- Requirements

- Update your weights and intercept by gradient descent (you can implement mini-batch gradient descent or stochastic gradient descent if you want).
- Use MSE (Mean Square Error) as your loss function.

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

- Tune the **learning rate** and **epoch** hyper-parameters (and **batch size** if you implement mini-batch gradient descent) to make your testing MSE loss as closed as the closed-form solution.

Linear Regression – Gradient Descent

- Grading Criteria
 - (10%) Show the weights and intercepts of your linear model.
 - (10%) Plot the learning curve. (x-axis=epoch, y-axis=training loss)
 - (20%) Show your error rate between your closed-form solution and the gradient descent solution.

error rate: $(\text{gradient_descent_loss} - \text{closed_form_loss}) / \text{closed_form_loss} * 100$

Points	error rate
20	< 0.5%
15	< 1%
10	< 3%
5	< 5%
0	>= 5%

Linear Regression – Gradient Descent

- Tips
 - Finding suitable hyper-parameters may cost you some time. Be patient!

Code Output

- Do not modify the main function architecture.
- Your code output will look like this:

```
Closed-form Solution
```

```
Weights: , Intercept: 
```

```
Gradient Descent Solution
```

```
Weights: , Intercept: 
```

```
Error Rate: 0.3%
```


Report

- Please follow the report template format.
- [Link](#)

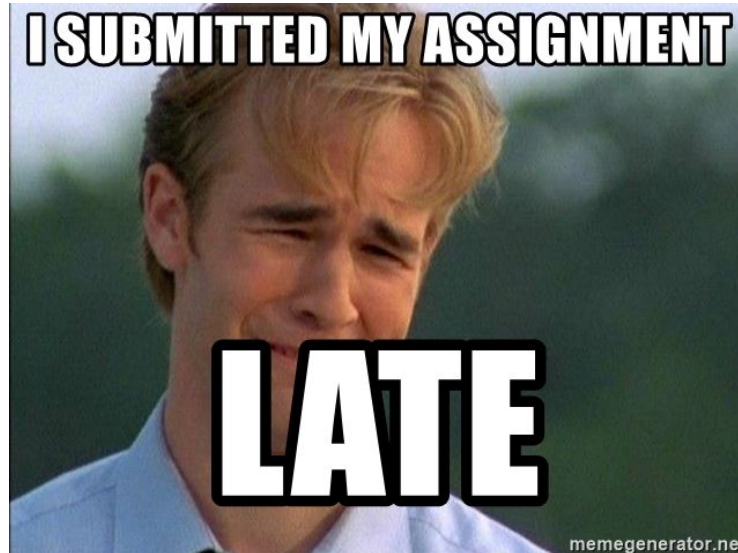
Submission

- Compress your code and report into a **.zip file** and submit it on E3.
- <STUDENT ID>_HW1.zip
 - <STUDENT ID>_HW1.py
 - <STUDENT ID>_HW1.pdf (do not submit .doc, .docx or others format)

```
zip 0716040.zip 0716040_HW1.py 0716040_HW1.pdf
adding: 0716040_HW1.py (stored 0%)
adding: 0716040_HW1.pdf (deflated 10%)
```

Late policy

- We will deduct a late penalty of 20 points per additional late day.
- For example, If you get 90 points but delay for two days, your will get only 50 points!



FAQs

- Why can't my gradient descent model converge?
 - Make sure you calculate the gradients correctly.
 - Use smaller learning rate.
- Can I use deep learning frameworks such as TensorFlow, PyTorch or other library such as math?
 - **No!** In HW1, you are request using **only Numpy** to implement linear regression and gradien descent. You can use matplotlib to plot the results.
- If you have other questions, ask on **E3 forum** first! We will reply as soon as possible.

Have Fun!

