



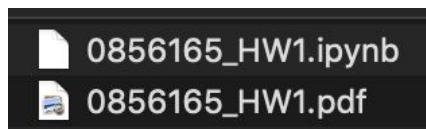
# Introduction to Machine Learning

## Homework 1 announcement

TA: 楊証琨 Jimmy  
Ph.D. student at National Taiwan University  
[d08922002@csie.ntu.edu.tw](mailto:d08922002@csie.ntu.edu.tw)

# Homework 1

- **Deadline: Oct. 25, Tue. at 23:59**
  1. Code assignment (60%): Implement linear regression using only numpy
  2. Short answer questions (40%): Write your answer on **pdf**
- Submit the **code** (.py/.ipynb) and **answers** (.pdf) on [E3](#)
  - Include your answers of code/question in the pdf
  - [HW1 questions](#)
  - [Sample Code](#)
- Compress your file in **.zip**



Compress



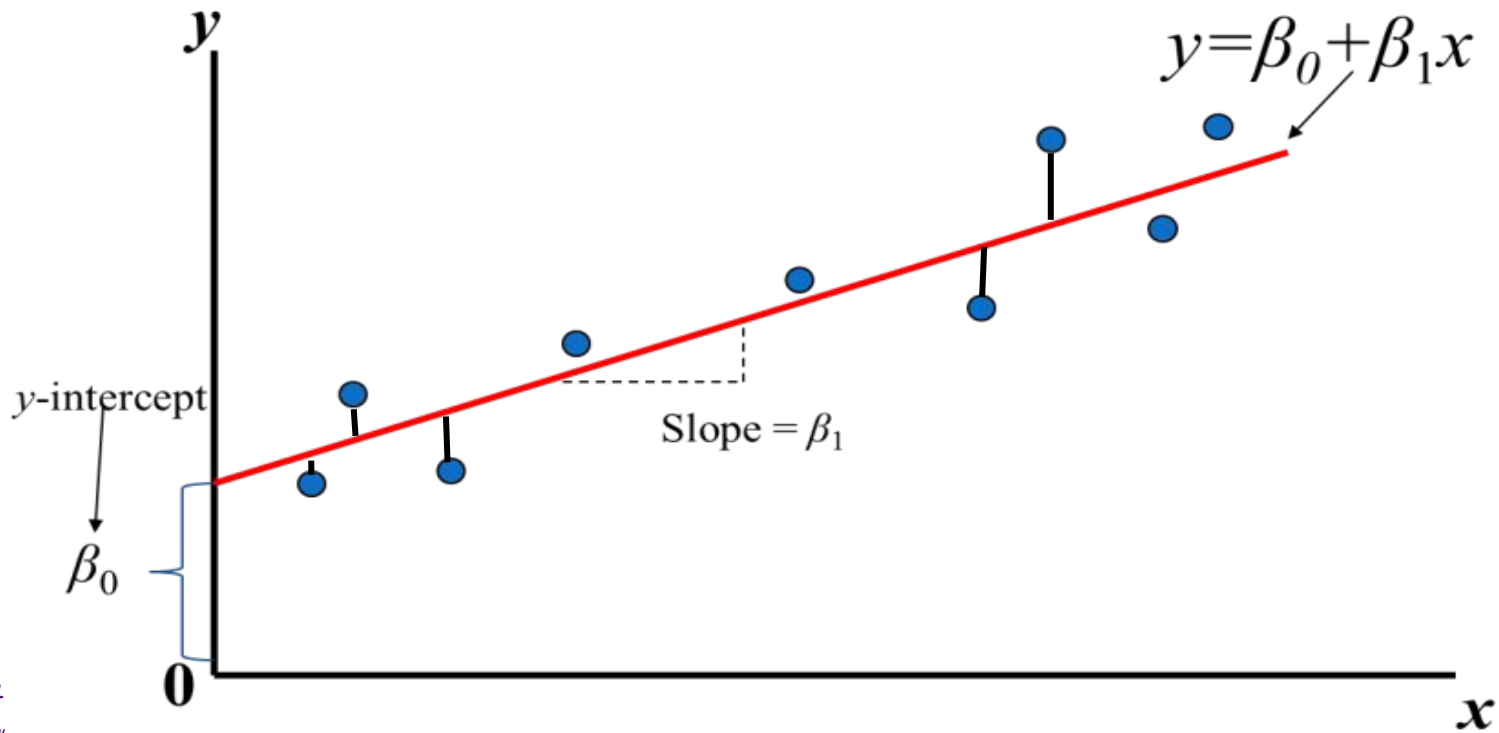
submit

[E3](#)



# Linear Regression

- Find the value of  $\beta_0$  and  $\beta_1$



# How to find $\beta_0$ and $\beta_1$ ?



TRY  
and  
Error

$$\beta_0 = -2, -1, 0, 1, 2, \dots$$
$$\beta_1 = 1, 2, 3, 4, 5, \dots$$

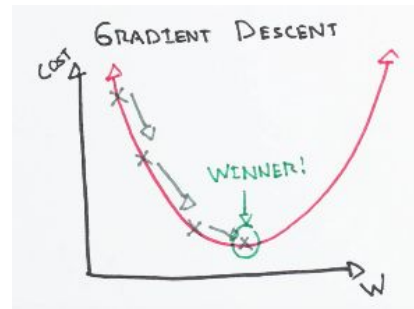


Closed  
form  
solution

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

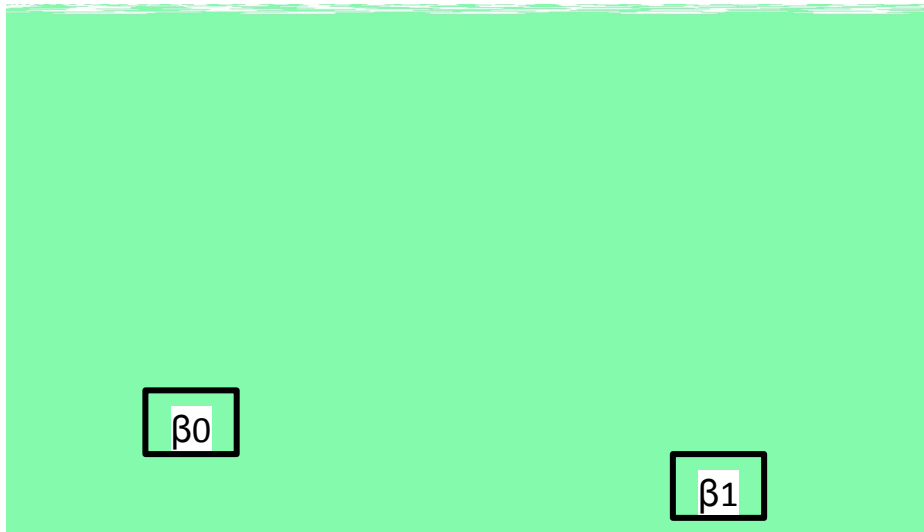


Gradient  
Descent



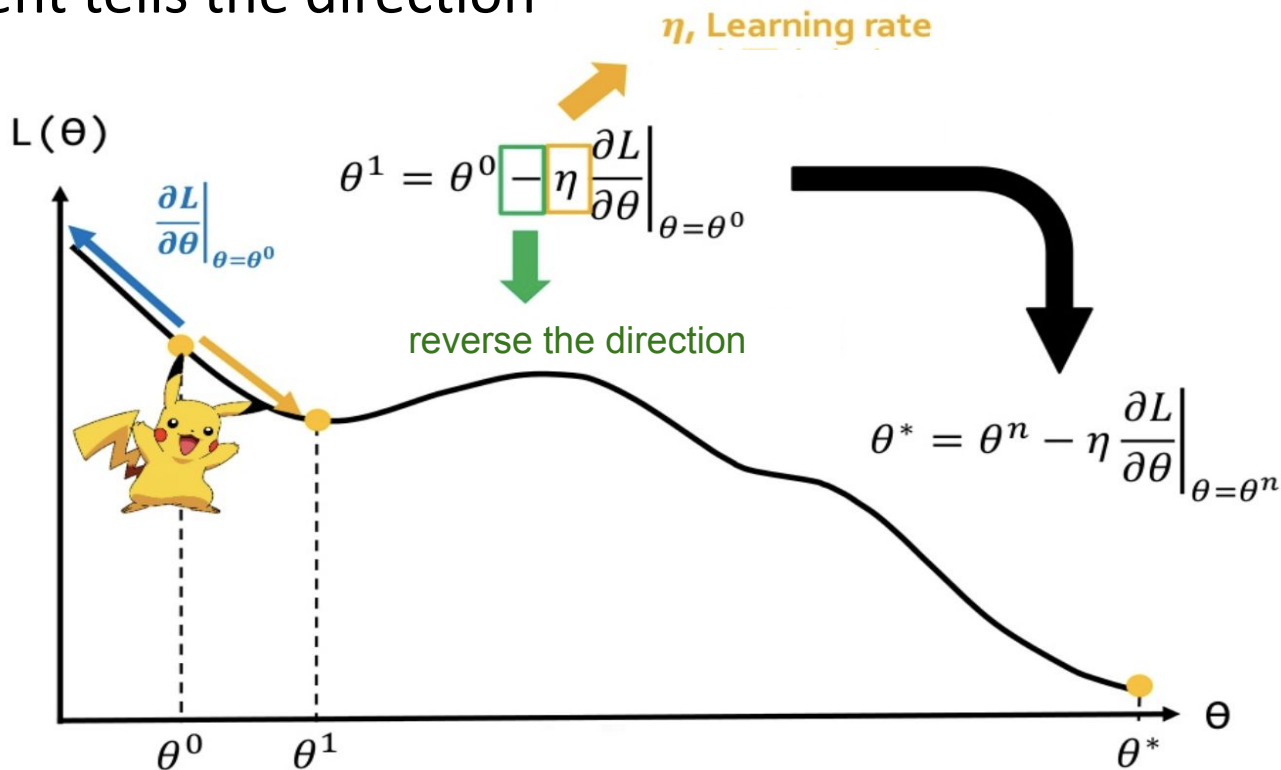
# Gradient Descent

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



# Gradient Descent

- Gradient tells the direction



# Gradient Descent: Pseudo code

## Algorithm

1. Initialize weights randomly  $\sim N(0, \sigma^2)$
2. Loop until convergence:
  - i. Pick batch of B data points
  - ii. Compute gradient.  $\frac{\partial J(\theta)}{\partial \theta} = \frac{1}{B} \sum_{k=1}^B \frac{\partial J_k(\theta)}{\partial \theta}$
  - iii. Update weights  $\theta \leftarrow \theta - \eta \frac{\partial J(\theta)}{\partial \theta}$
3. Return weights

- Supplementary materials: [Andrew NG: Gradient Descent](#)



# Code readability

- Write beautiful Python code with [PEP8 guidelines](#) for readability
- Base requirement: use **whitespace** correctly!
- Useful package: [Pylance](#)

Python

```
# Recommended
def function(default_parameter=5):
    # ...

# Not recommended
def function(default_parameter = 5):
    # ...
```

Python

```
# Recommended
my_list = [1, 2, 3]

# Not recommended
my_list = [ 1, 2, 3, ]
```

Python

```
x = 5
y = 6

# Recommended
print(x, y)

# Not recommended
print(x , y)
```



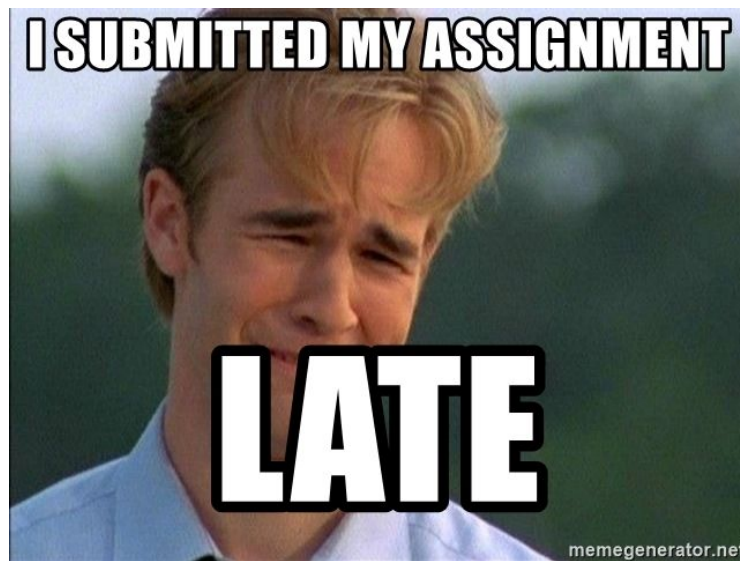




國立交通大學  
National Chiao Tung University

# Late policy

- We will deduct a late penalty of **20 points** per additional late day
- For example, If you get 90 points of HW but delay for two days, your will get only  $90 - (20 \times 2) = 50$  points!



# FAQ

- Why my loss is high and the training can not converge
  - Make sure you calculate the gradients correctly
  - Use smaller learning rate
- Can I use deep learning frameworks such as TensorFlow, PyTorch?
  - **No!** In HW1, you are request using **only Numpy** to implement linear regression and gradien descent. You can use matplotlib to plot the results.
- **DO NOT** copy homework from others! Otherwise, both of you will get 0 points for the homework



# Notice

- Submit your homework on [E3-system](#) !
- Check your email regularly, we will mail you if there are any updates or problems of the homework
- If you have any questions or comments for the homework, please mail me and cc Prof. Lin
  - Prof. Lin, [lin@cs.nctu.edu.tw](mailto:lin@cs.nctu.edu.tw)
  - TA Jimmy, [d08922002@csie.ntu.edu.tw](mailto:d08922002@csie.ntu.edu.tw)
  - TA 政儒, [ace52751208@gmail.com](mailto:ace52751208@gmail.com)
  - TA 季嘉, [jijiawu.cs@gmail.com](mailto:jijiawu.cs@gmail.com)
  - TA 睿哲, [benchiang.cs07@nctu.edu.tw](mailto:benchiang.cs07@nctu.edu.tw)



# Have fun!

