

Deep Into Deep

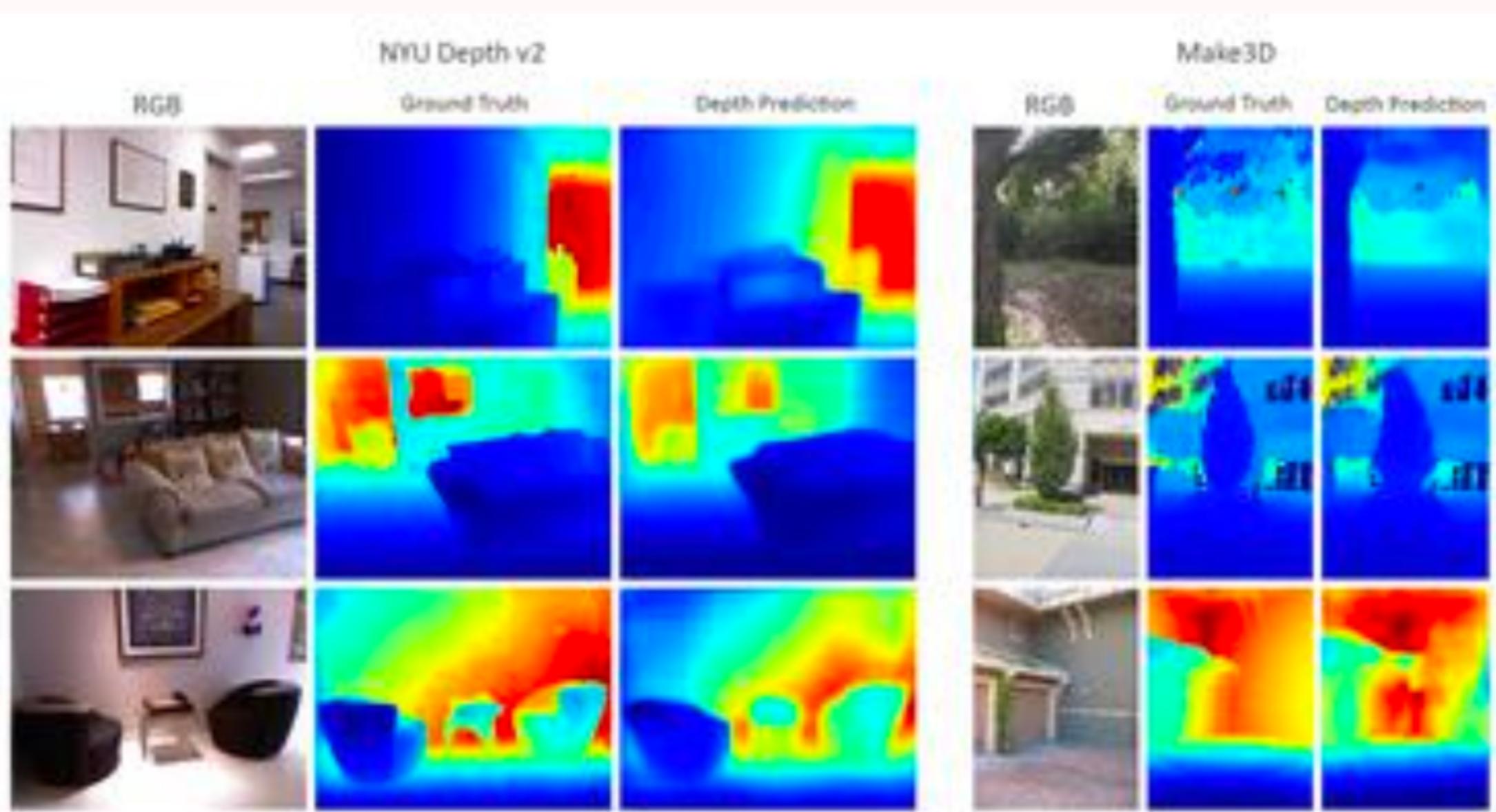
김성찬

Contents

1. Regression
2. Linear Regression
3. Cost Function
4. Gradient Descent
5. Classification
6. Logistic Regression

1. Regression

Regression vs. Classification



<https://paperswithcode.com/task/depth-estimation>



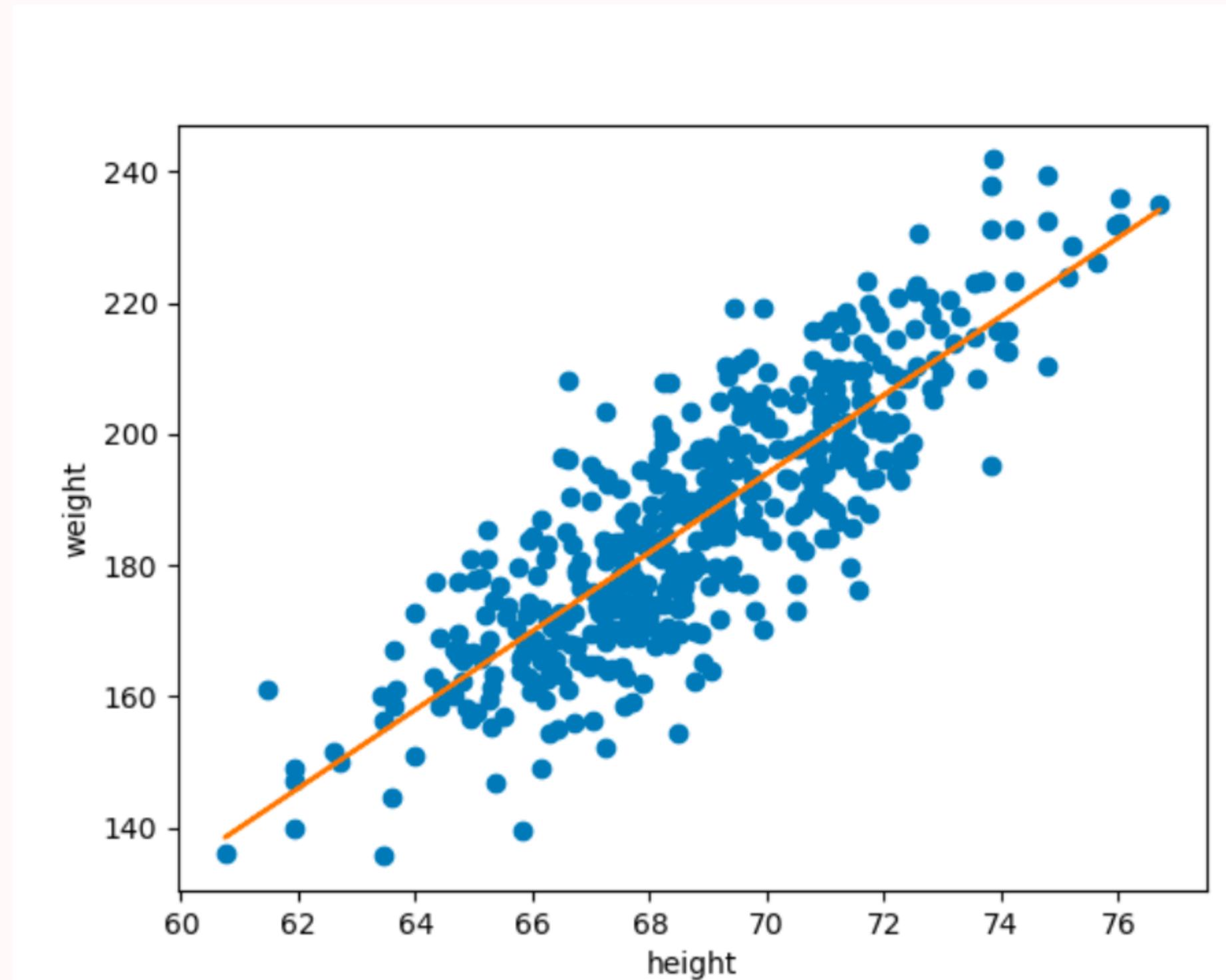
<https://paperswithcode.com/task/image-classification>

1. Regression

Regression is
to **relate** input variables
to the output variable,

to either **predict** outputs
for new inputs, and/or

to **understand** the effect of the input
on the output.



1. Regression

Regression에서, 데이터셋은 다음과 같이 표현한다.

$$D = \{(x_n, y_n)_{n=1}^N\}$$

이때, y_n 은 n 번째 레이블이고, x_n 은 n 번째 입력값이다. N 은 데이터셋의 크기다.

x 가 d 개의 값으로 이루어져 있을 때 d 는 x 의 dimensionality라고 한다.

x : feature, covariates, independent variables, explanatory variables

y : target, label, response, outcome, dependent variable

1. Regression

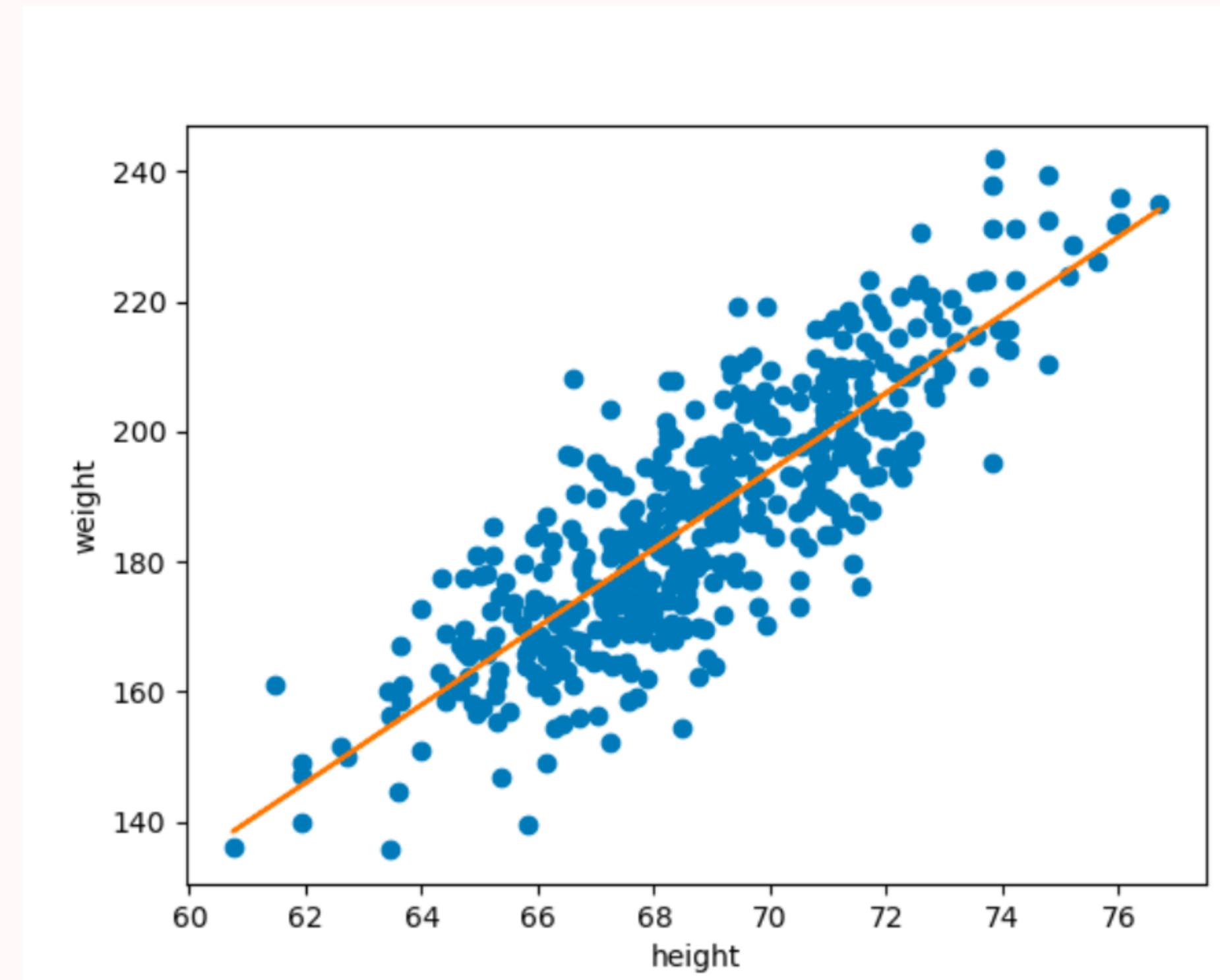
Regression의 두 가지 목표

1. Prediction:

새로운 입력에 대응하는 출력을 예측한다.

2. Interpretation:

출력에 대한 입력의 영향을 이해한다.

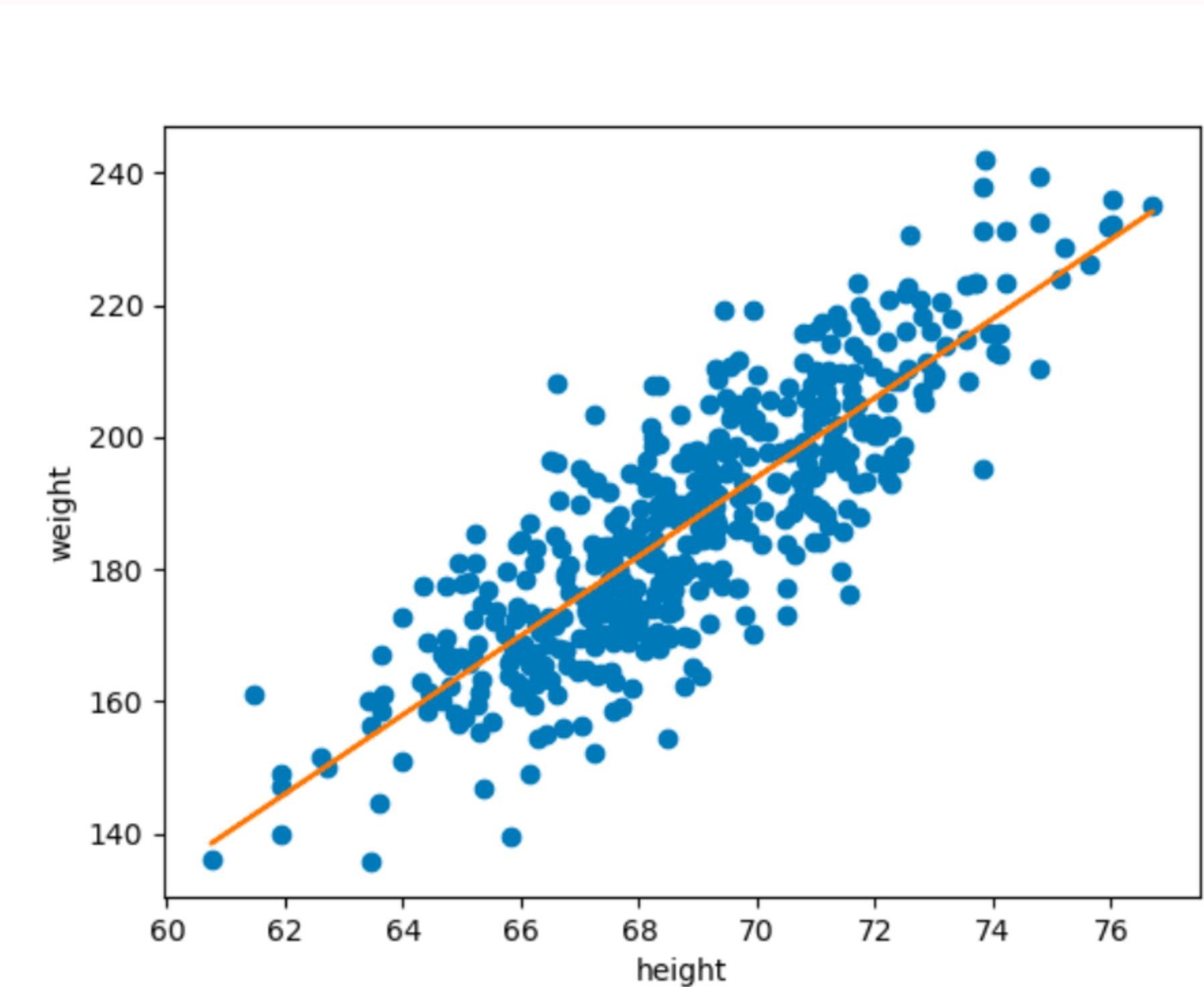


1. Regression

Regression Function (Hypothesis)

find a **mapping function**
that approximates the output
"well enough" given inputs

데이터를 잘 **근사**하는 함수란 무엇일까?

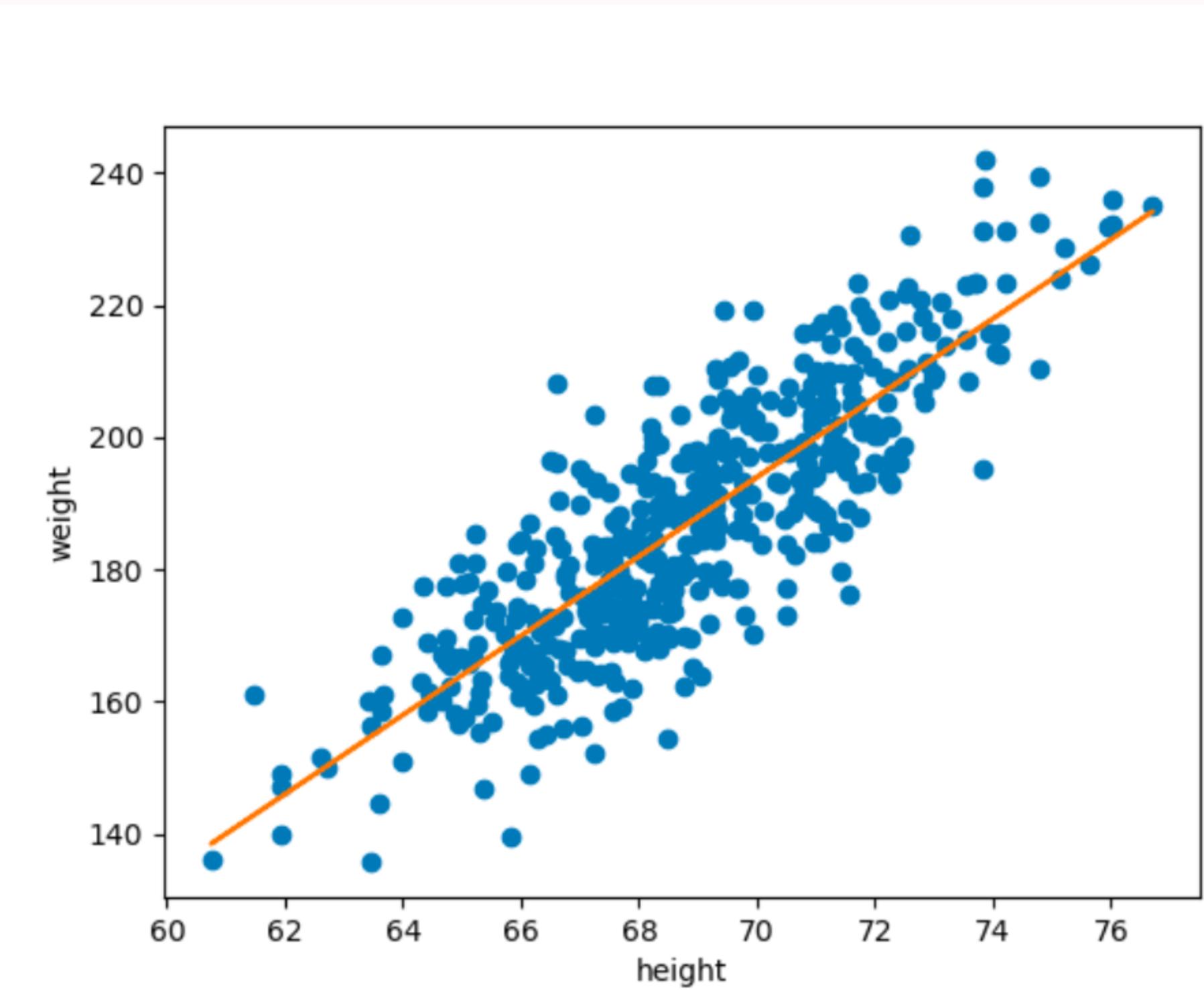


1. Regression

Not Causation But Correlation

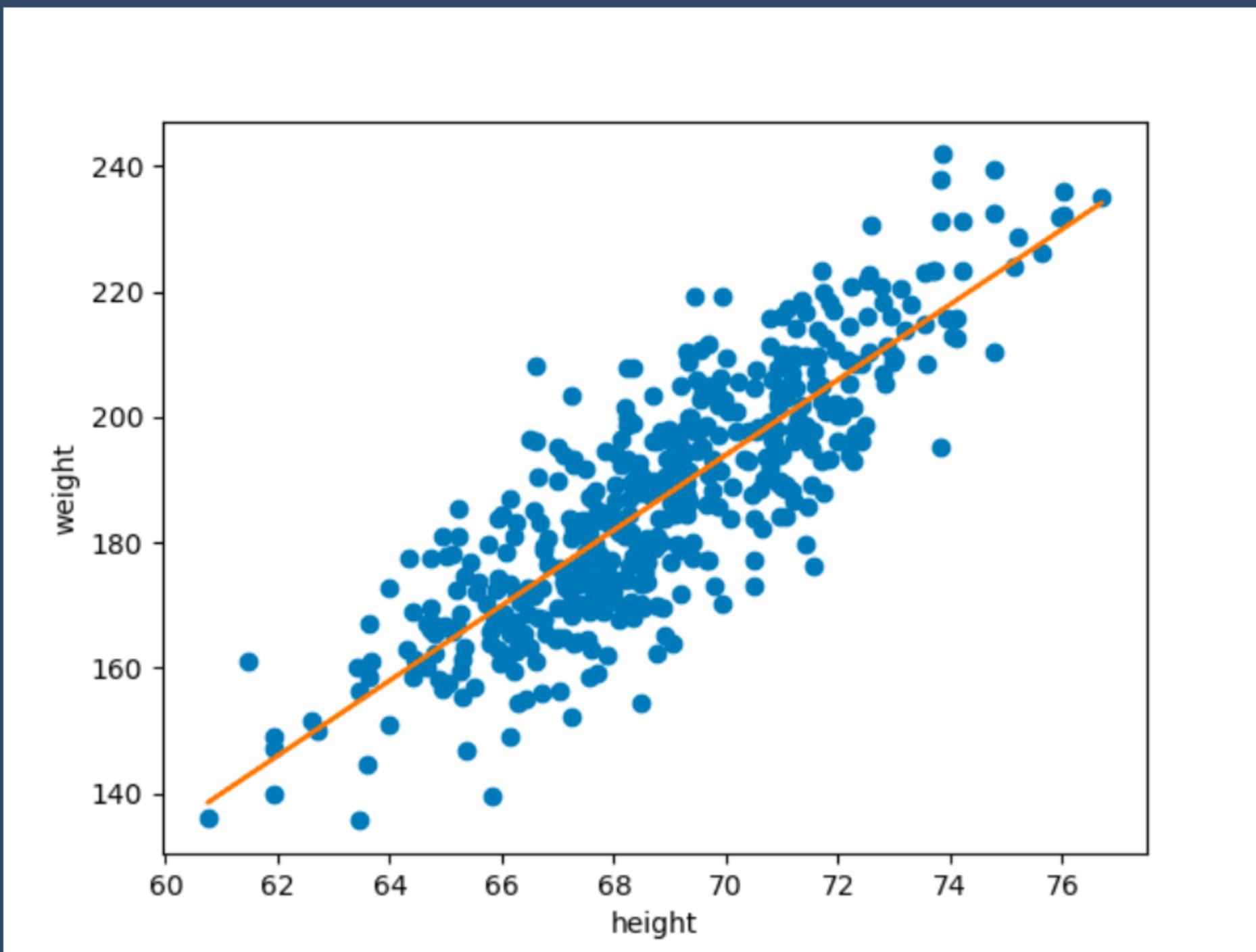
연관성을 찾는 것이지
원인을 찾는 것이 아니다.

따라서 입력과 출력 사이의 관계에 대해
잘못 해석하면 안된다.



2. Linear Regression

선형 회귀는 입력과 출력 사이에 선형적인 관계를 가정한다.



2. Linear Regression

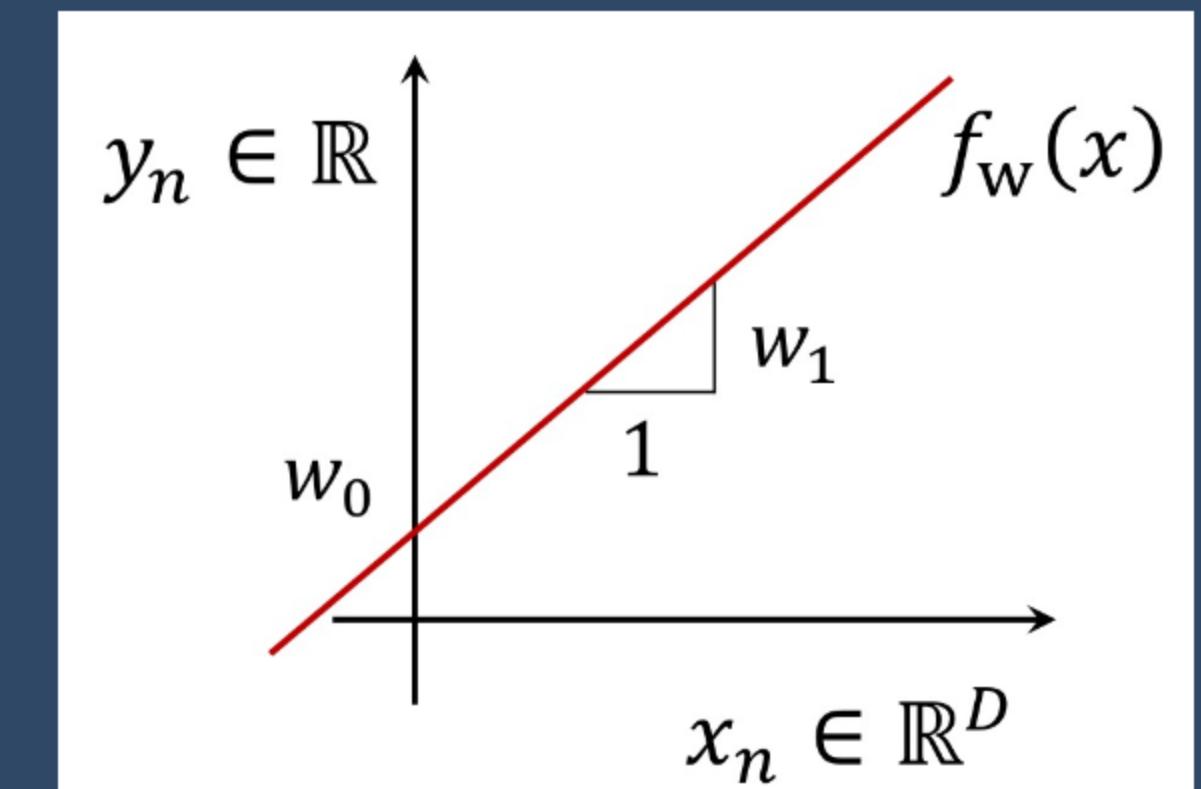
가장 간단한 형태의 선형 회귀는 1차원 입력에 대한 선형 회귀이다. 2차원 상의 직선의 방정식과 같다.

$$y_n \approx f(x_n) = f_W(x_n) = W_0 + W_1 x_n$$

이때, $W = (W_0, W_1)$ 은 모델 파라미터이고, $f_W(x)$ 를 모델이라고 한다.|

W 를 찾는 과정을
learning, estimating the parameter
또는 fitting the model이라고 한다.

이를 위해 optimization algorithm이 필요하다.



3. Cost Function

어떻게 학습할 것인가?

- 학습을 위해서는 현재의 상태를 알아야 한다. 여기서 상태라 함은 모델의 성능을 의미한다.

모델의 성능을 어떻게 알 수 있을까?

- 모델의 성능을 평가하기 위한 척도가 필요하다. 이를 위해 우리는 Cost Function을 정의한다.

Cost Function이란?

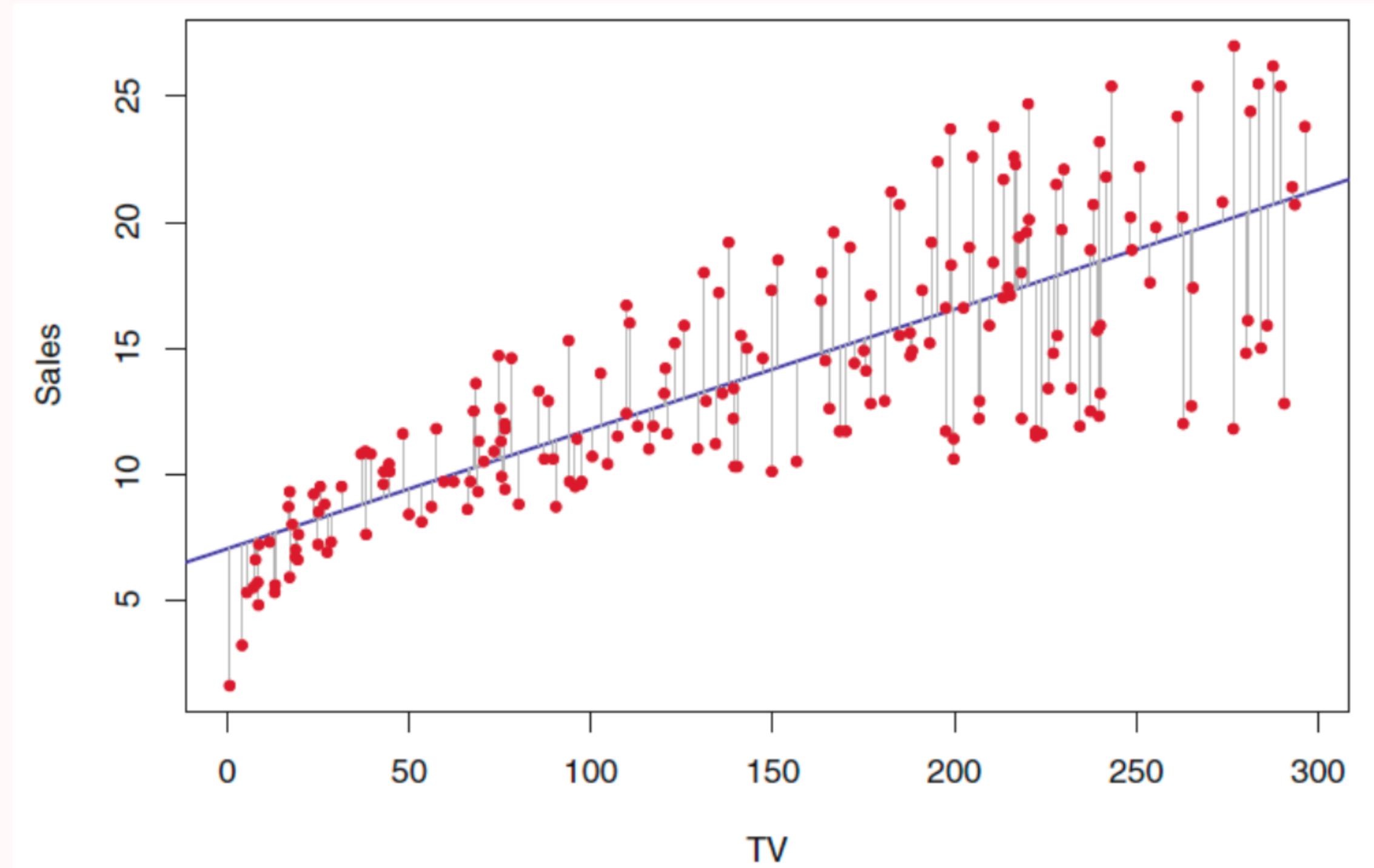
- 모델이 데이터를 얼마나 잘 표현하는가 또는 모델의 입력에 대한 출력,
즉 예측값이 얼마나 레이블과 유사한가를 측정하기 위한 함수이다.

3. Cost Function

W의 변화가
오차에 주는 영향을
분석하면서
어떻게 오차를 줄일 수 있을지
즉, 어떻게 성능을 올릴 수 있을지
고민해보자.

1. 기울기가 변화하는 경우

2. 편향이 변화하는 경우



3. Cost Function

$$\min_W \frac{1}{N} \sum_{n=1}^N (f_W(x_n) - y_n)^2$$

학습 데이터셋 $\{(x_n, y_n)\}_{n=1}^N$ 에 대하여
 y 에 가장 가까운 $f_W(x)$ 를 표현하는
 $W = (W_0, W_1)$ 을 선택한다.

cost function은 $f(x_n)-y_n$
loss는 그것들 sigma

$$L(W) = \frac{1}{N} \sum_{n=1}^N C(f_W(x_n), y_n)$$

$$W^* = \arg \min_W L(W)$$

3. Cost Function

Cost Function은 데이터를 잘 표현하는 파라미터를 학습하기 위해 사용된다.
다른 표현으로 Loss, Training object라고도 한다.

Cost Function은 모델이 얼마나 데이터를 잘 설명하는지, 또는 얼마나 잘 설명하지 못하는지를
양적으로 표현한다.



3. Cost Function

$$f_W(x) = W_1 x$$

$$W = W_1$$

$$L(W) = \frac{1}{N} \sum_{n=1}^N (f_W(x_n) - y_n)^2$$

$$W^* = \arg \min_W L(W)$$

3. Cost Function

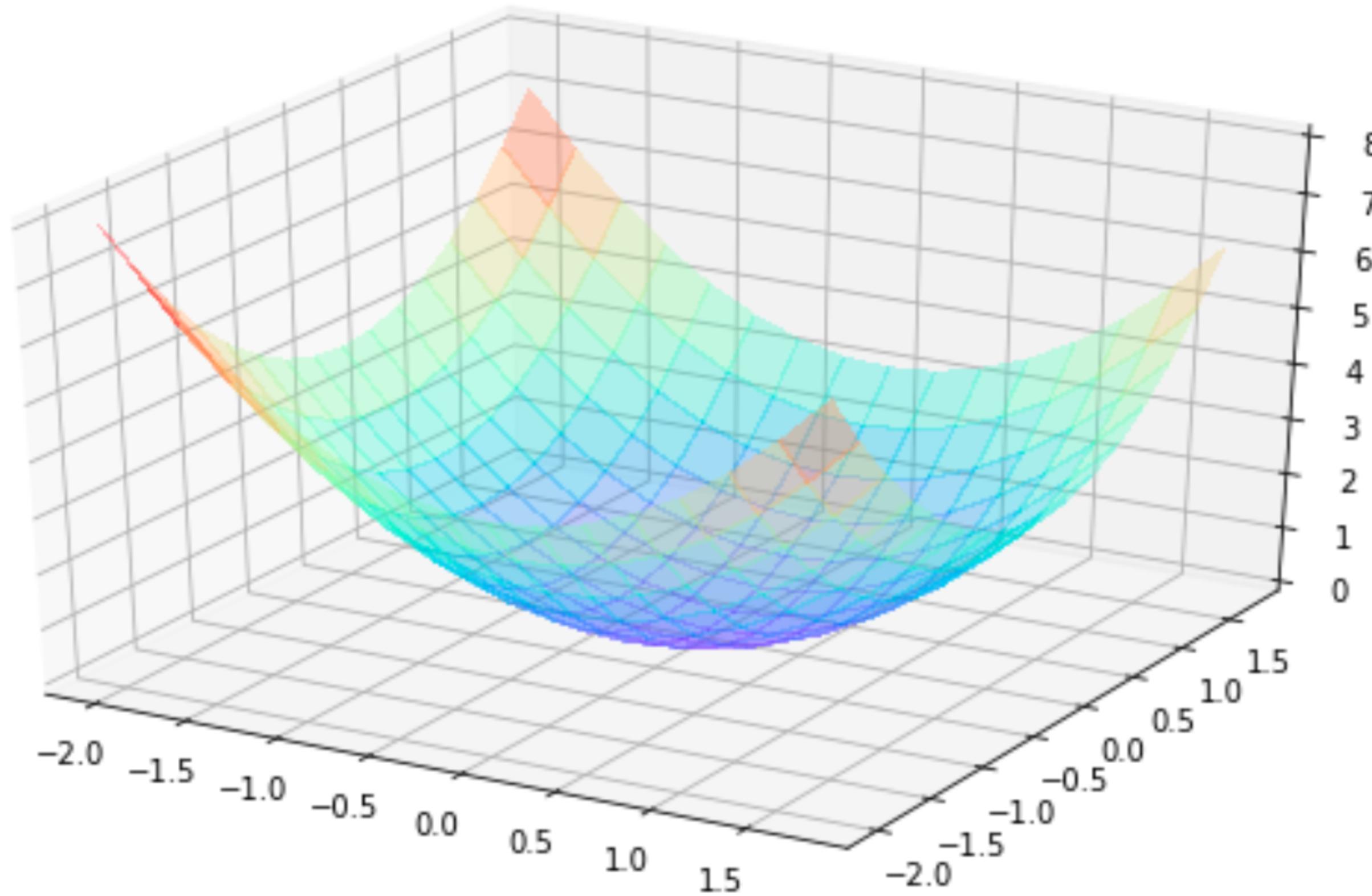
$$f_W(x) = W_0 + W_1 x$$

$$W = (W_0, W_1)$$

$$L(W) = \frac{1}{N} \sum_{n=1}^N (f_W(x_n) - y_n)^2$$

$$W^* = \arg \min_W L(W)$$

3. Cost Function



2개의 파라미터를 가지는
선형 회귀의 경우
만들어질 수 있는
오차 함수의 그래프

3. Cost Function

Cost Function의 종류

- Mean Square Error (MSE)

$$L_{MSE}(W) = \frac{1}{N} \sum_{n=1}^N (f_W(x_n) - y_n)^2$$

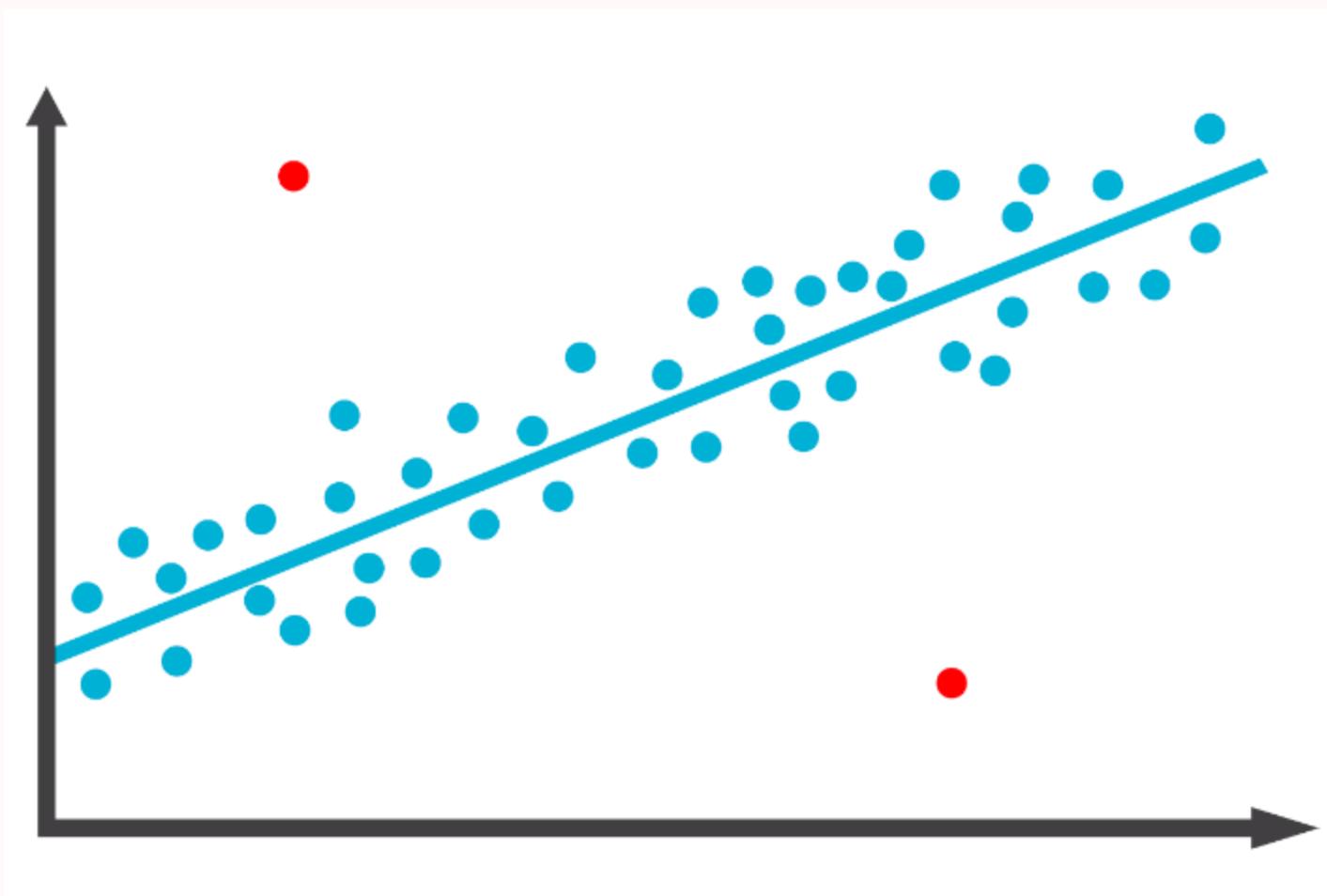
- Mean Absolute Error (MAE)

$$L_{MAE}(W) = \frac{1}{N} \sum_{n=1}^N |f_W(x_n) - y_n|$$

3. Cost Function

Cost Function의 이상적인 특성

1. 레이블의 값이 실수인 경우, 0에 대하여 대칭인 것이 좋다.
 > 양의 값과 음의 값이 동일하게 평가된다.
2. 큰 오류와 매우 큰 오류를 동일하게 평가하는 것이 좋다.
 > Outlier를 매우 큰 오류로 평가하는 경우 제대로 된 학습이 어렵다.



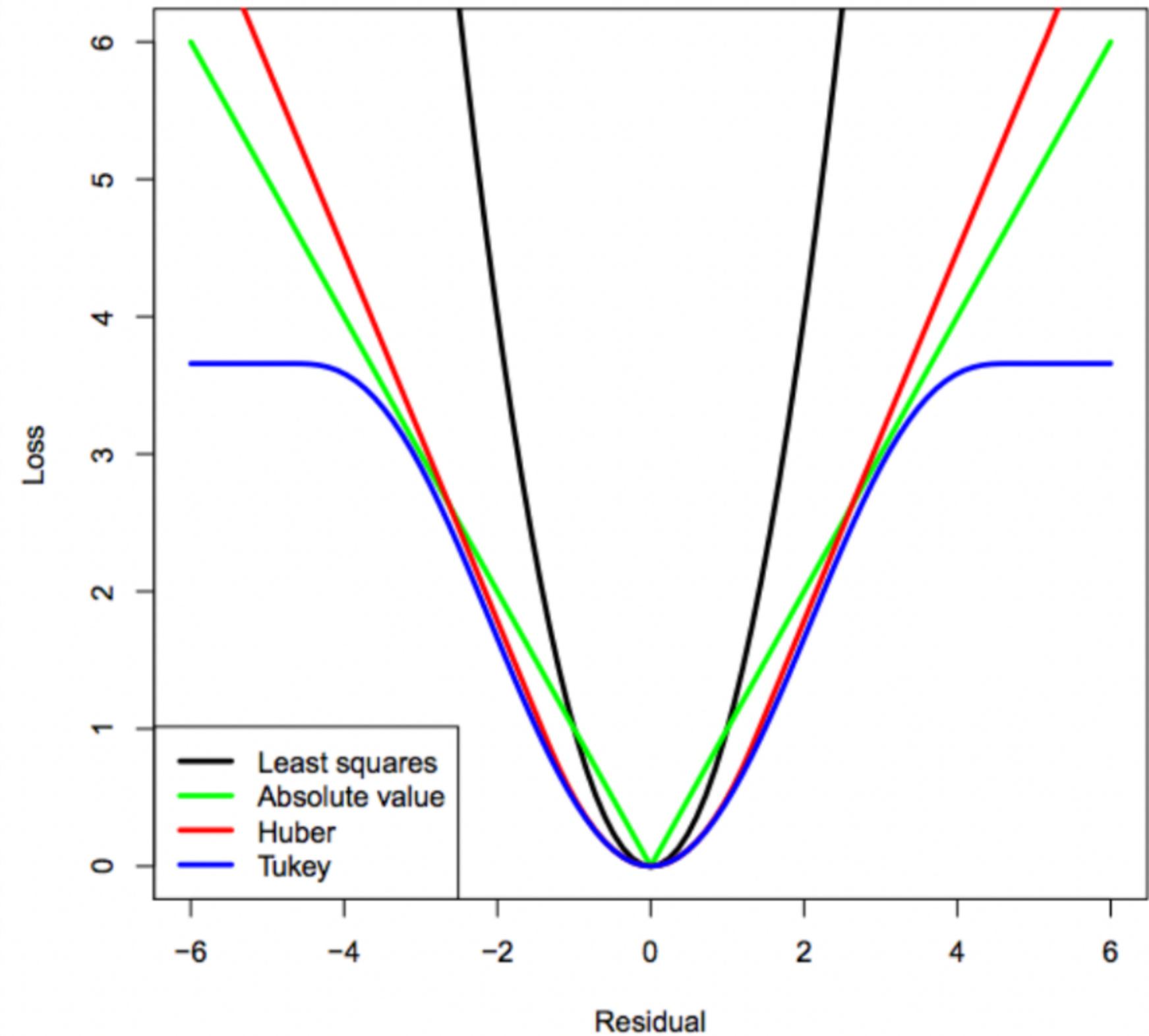
3. Cost Function

Statistical vs. Computational Trade-off

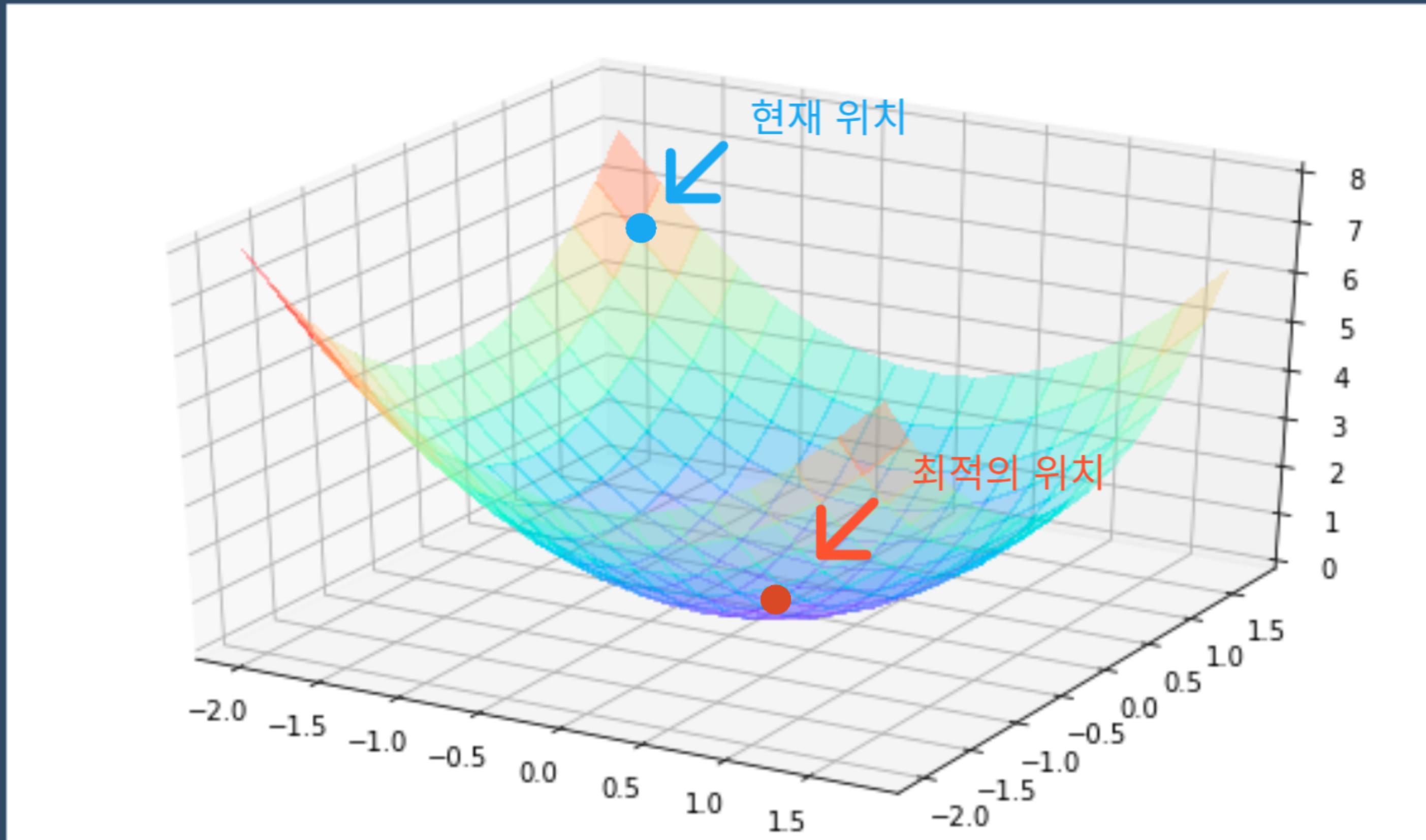
좋은 statistical properties를 얻기 위해서는 computational properties를 포기해야 한다.

- 이상치에 대한 오류를 제거하기 위해 조금 느린 방식을 택할 것인가?

tukey는 지수함수와 computation이 많음. 그대신 좋은 statistical properties를 가짐.



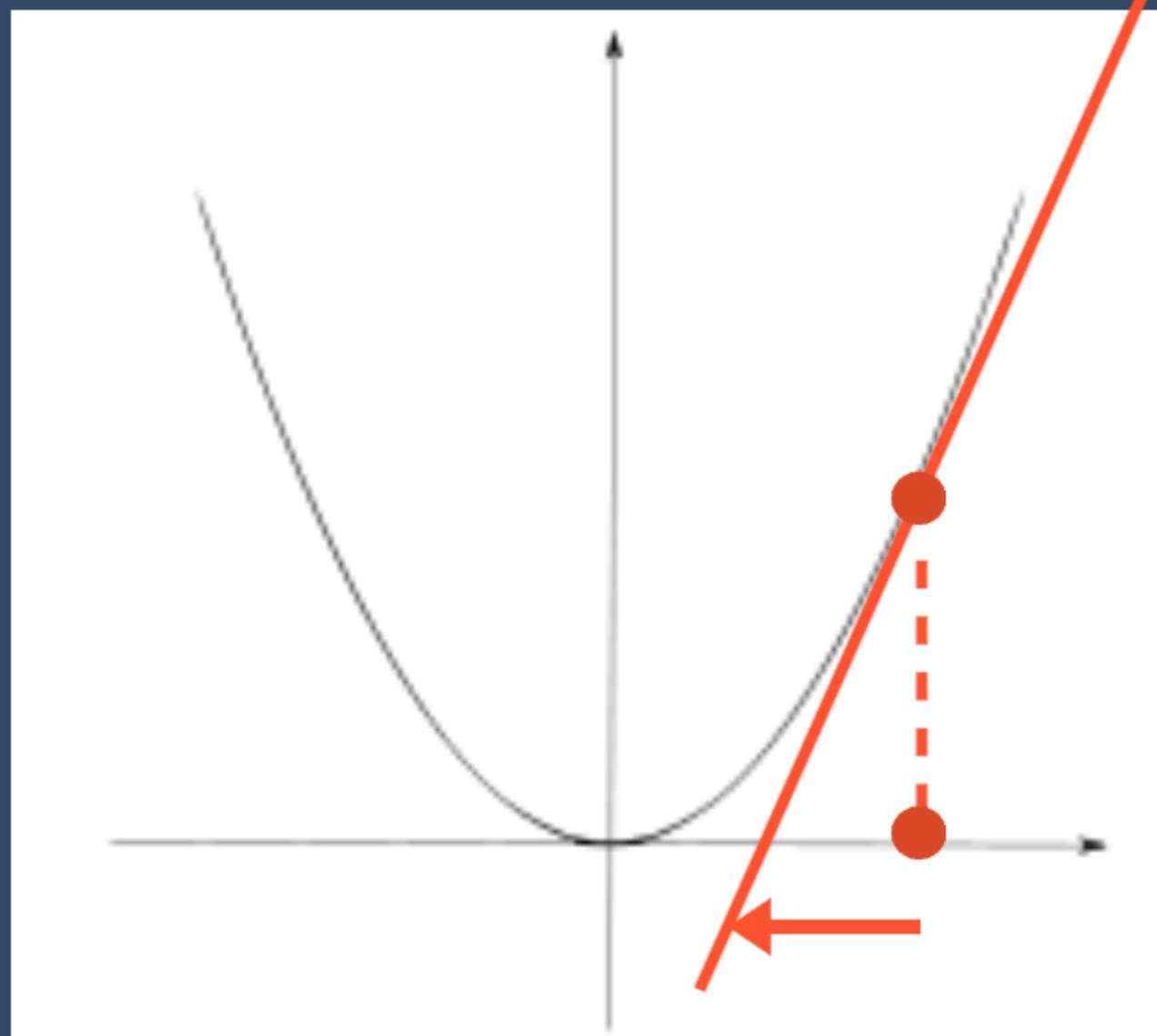
4. Gradient Descent: 어떻게 학습할 것인가?



4. Gradient Descent

Grid Search는 좋지 않다.

Gradient를 따라 이동하기: Gradient는 함수가 가장 크게 증가하는 방향을 가리킨다.



Cost Function을 줄이는 방향으로 학습하기 위해서
Gradient Descent의 방향은
Gradient의 반대 방향이다.

4. Gradient Descent

1. 어디에서 시작할 것인가?

> Initialization

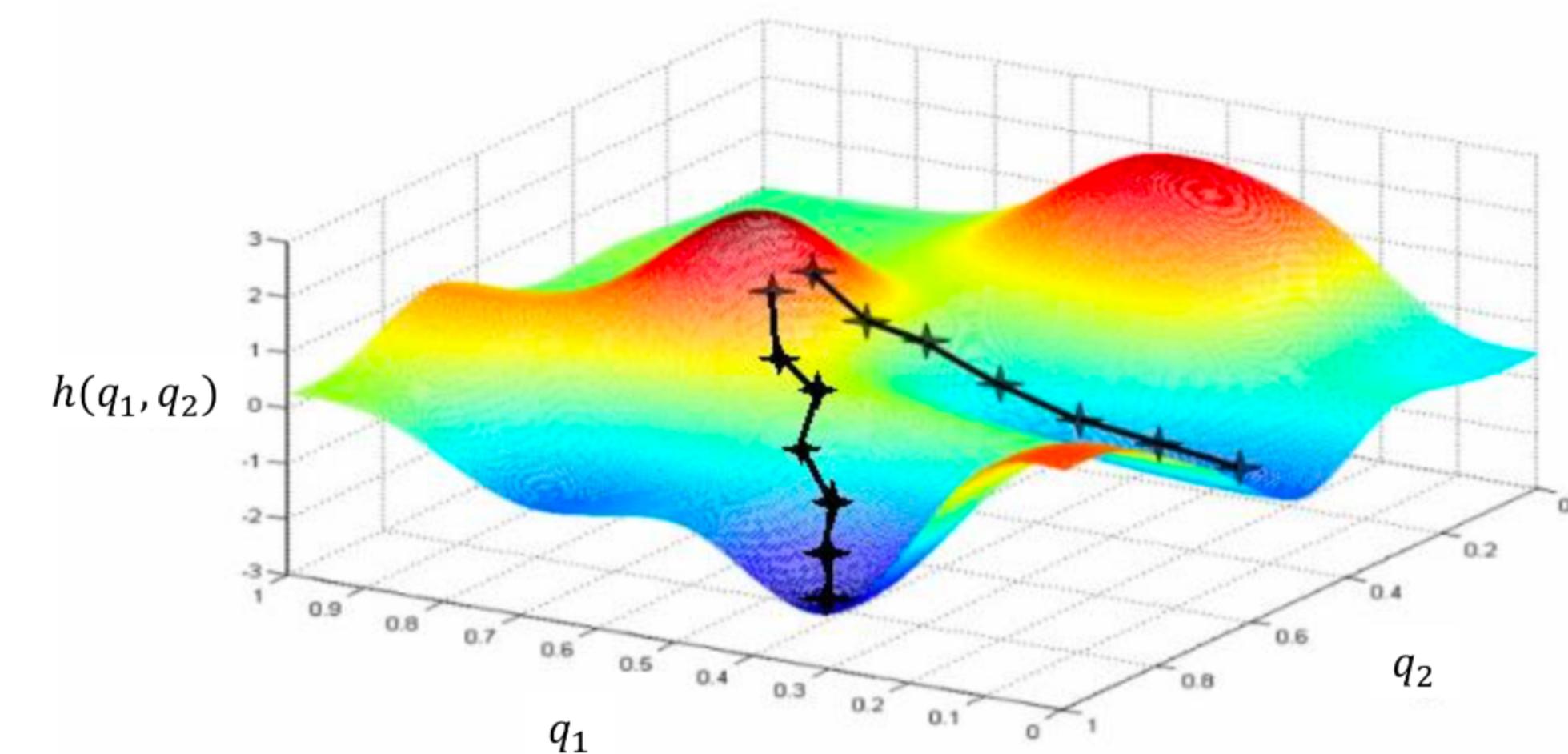
2. 어느 방향으로 이동할 것인가?

> Gradient

3. 얼마나 이동할 것인가?

> Learning rate

Non-convex Example



4. Gradient Descent

Gradient Descent Algorithm: Repeat until convergence

$$W_0 \leftarrow W_0 - \alpha \frac{\partial}{\partial W_0} L(W)$$

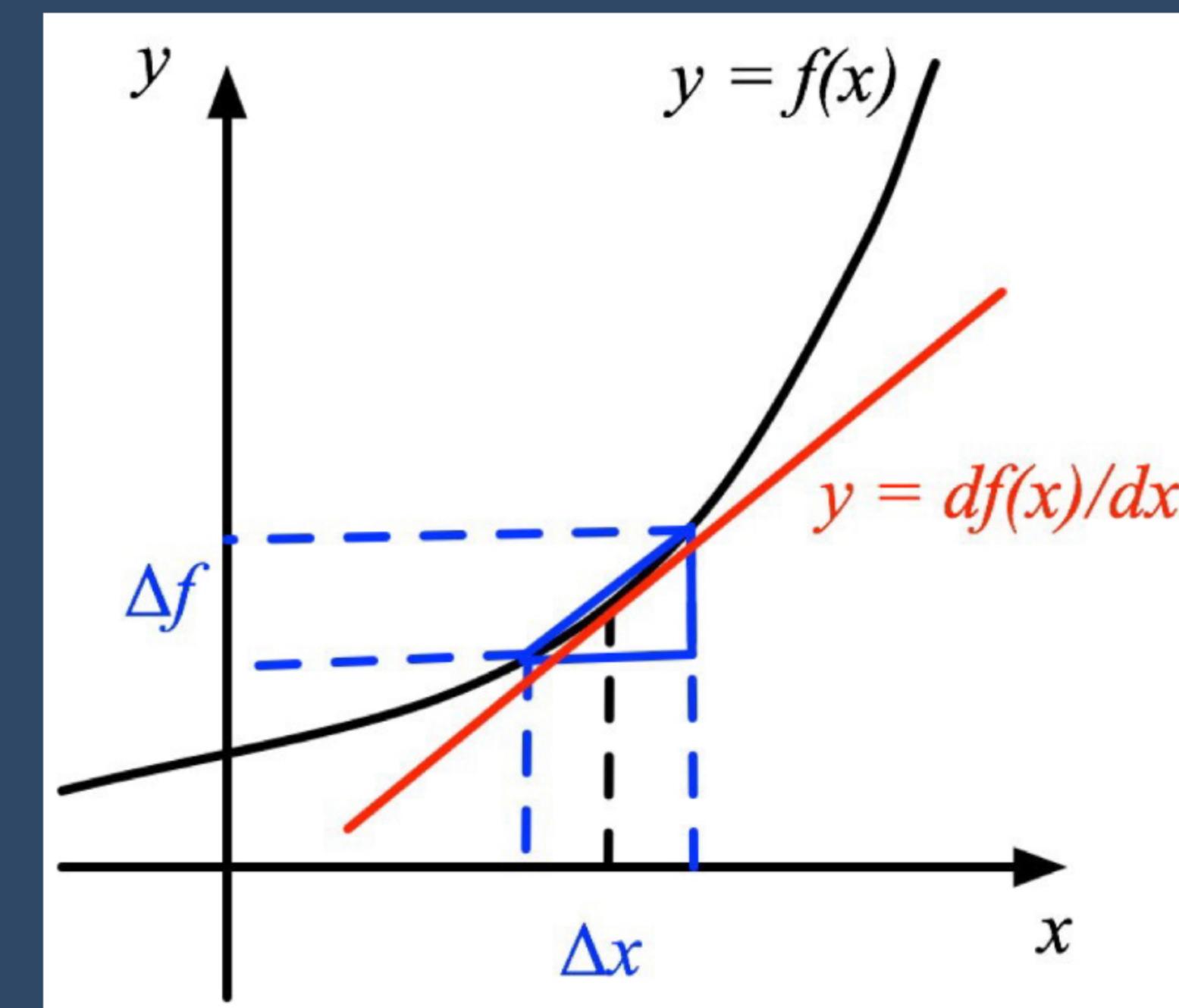
$$\cancel{W_0} \leftarrow W_1 - \alpha \frac{\partial}{\partial W_1} L(W)$$

4. Gradient Descent

Numetrical Gradients vs. Analytic Gradients

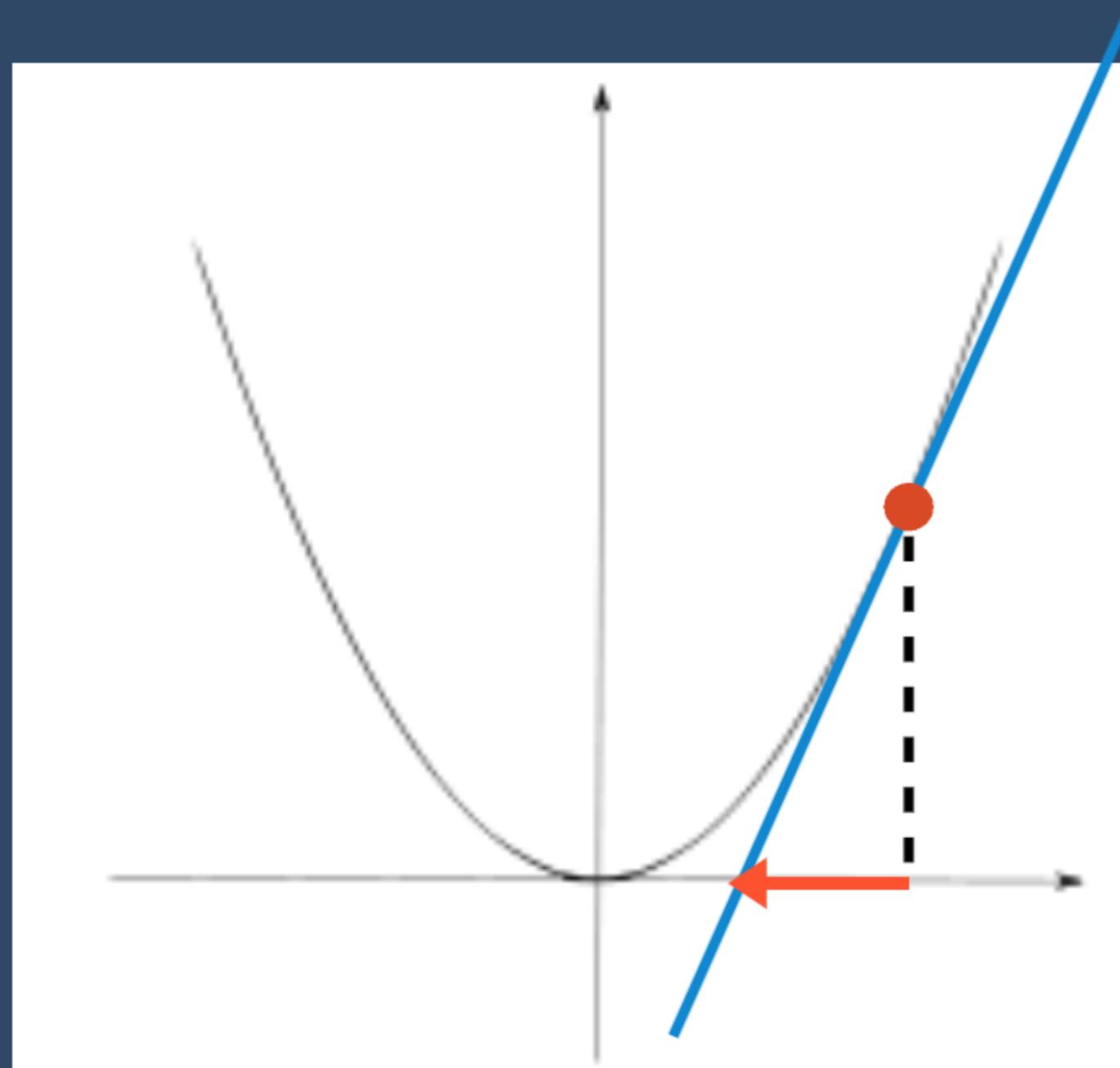
Numetrical Gradients : 평균변화율

Analytic Gradients : 순간변화율



4. Gradient Descent

Gradient와 Parameter Update의 관계



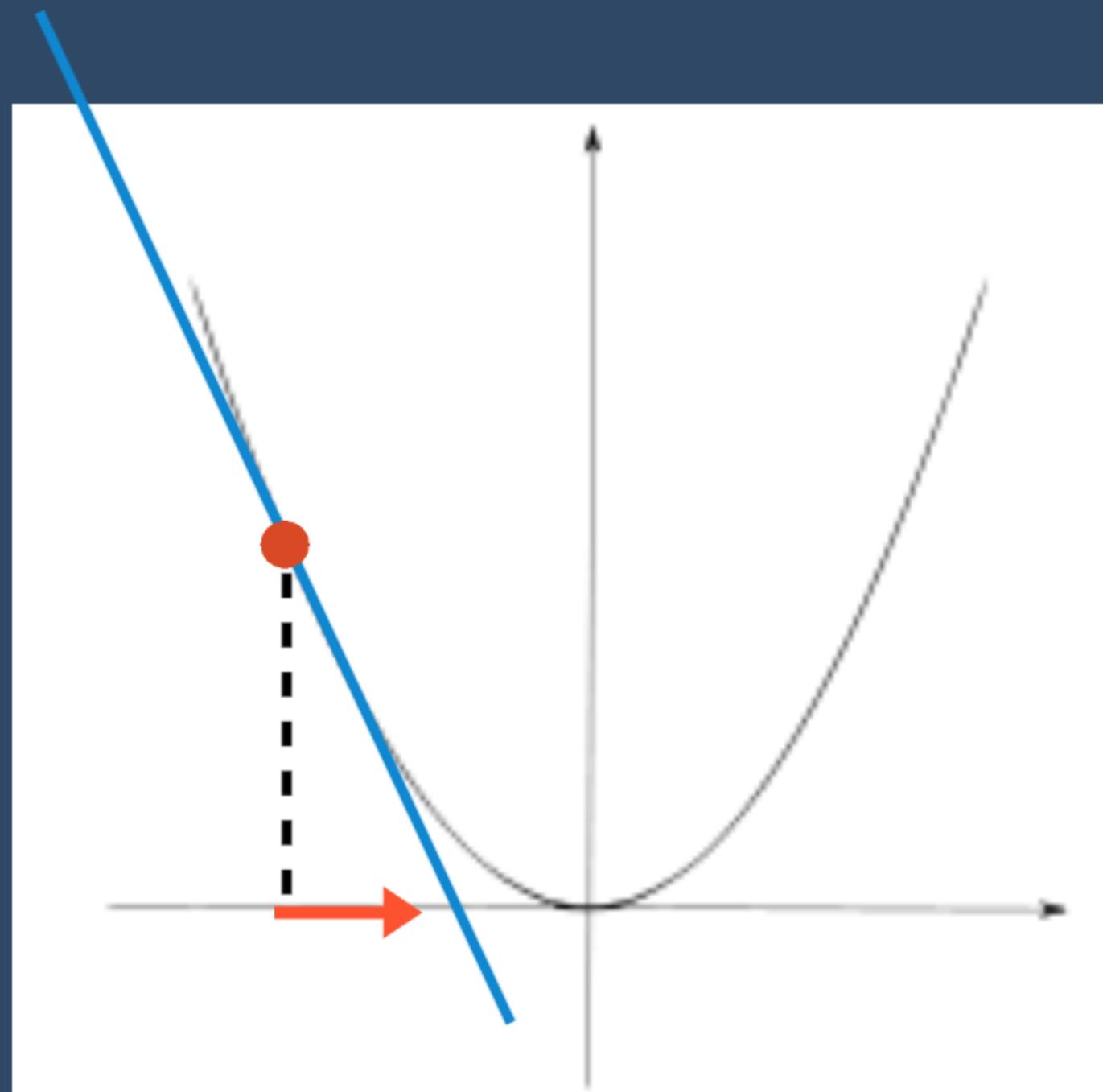
Gradient가 양수인 경우
왼쪽으로 이동하면 Loss가 감소한다.

따라서 파라미터는
감소시키는 방향으로 Update해야 한다.

$$W \leftarrow W - \alpha \frac{\partial}{\partial W} L(W)$$

4. Gradient Descent

Gradient와 Parameter Update의 관계



Gradient가 음수인 경우
오른쪽으로 이동하면 Loss가 감소한다.

따라서 파라미터는
증가시키는 방향으로 Update해야 한다.

$$W \leftarrow W - \alpha \frac{\partial}{\partial W} L(W)$$

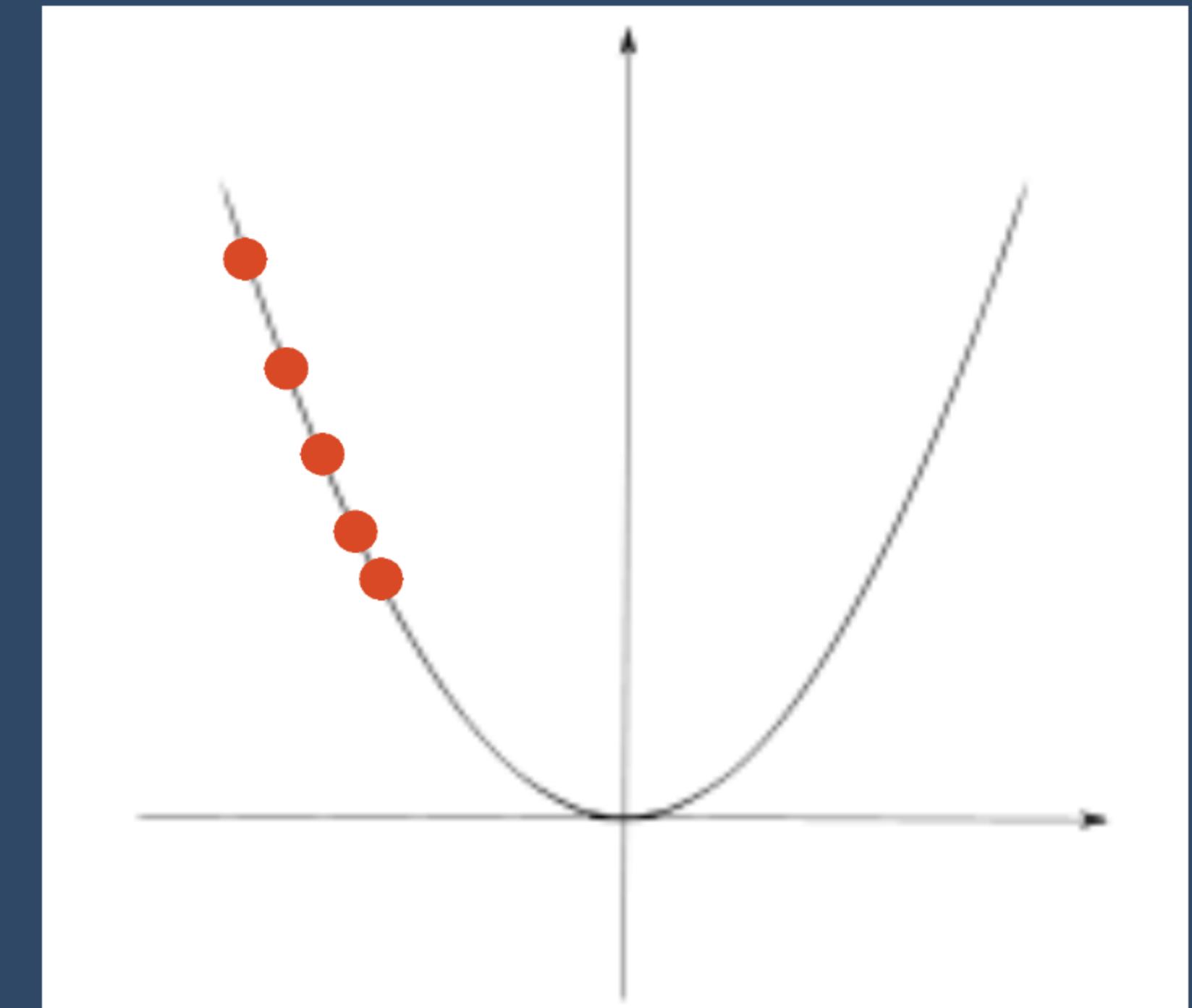
4. Gradient Descent

$$W \leftarrow W - \alpha \frac{\partial}{\partial W} L(W)$$

alpha는 learning rate, 학습률이라고 한다.

학습률이 너무 작은 경우, 학습이 너무 느리고 때문에 학습이 제대로 이루어지지 못할 수 있다.

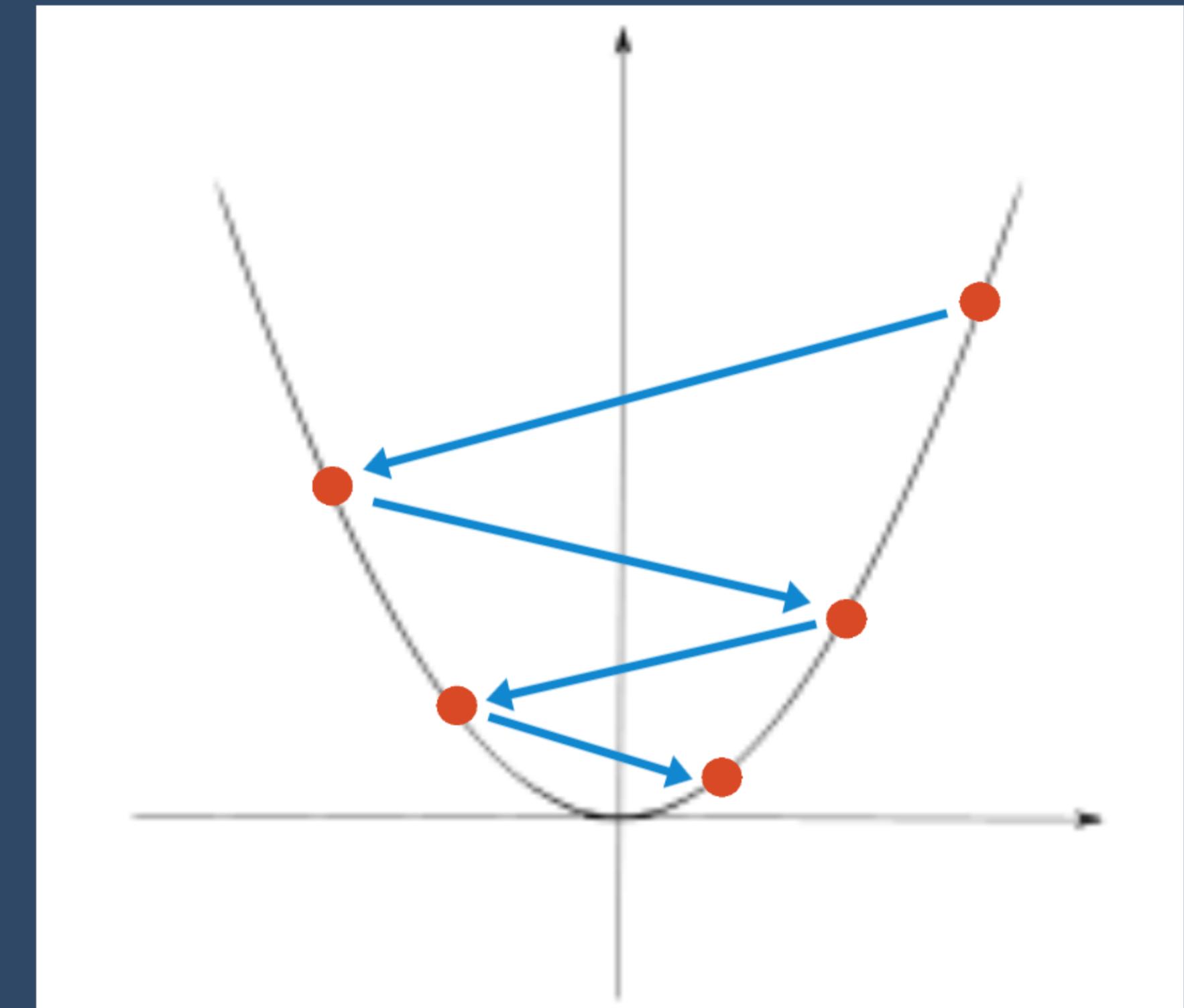
> 일반적으로 underfitting이라고 한다.



4. Gradient Descent

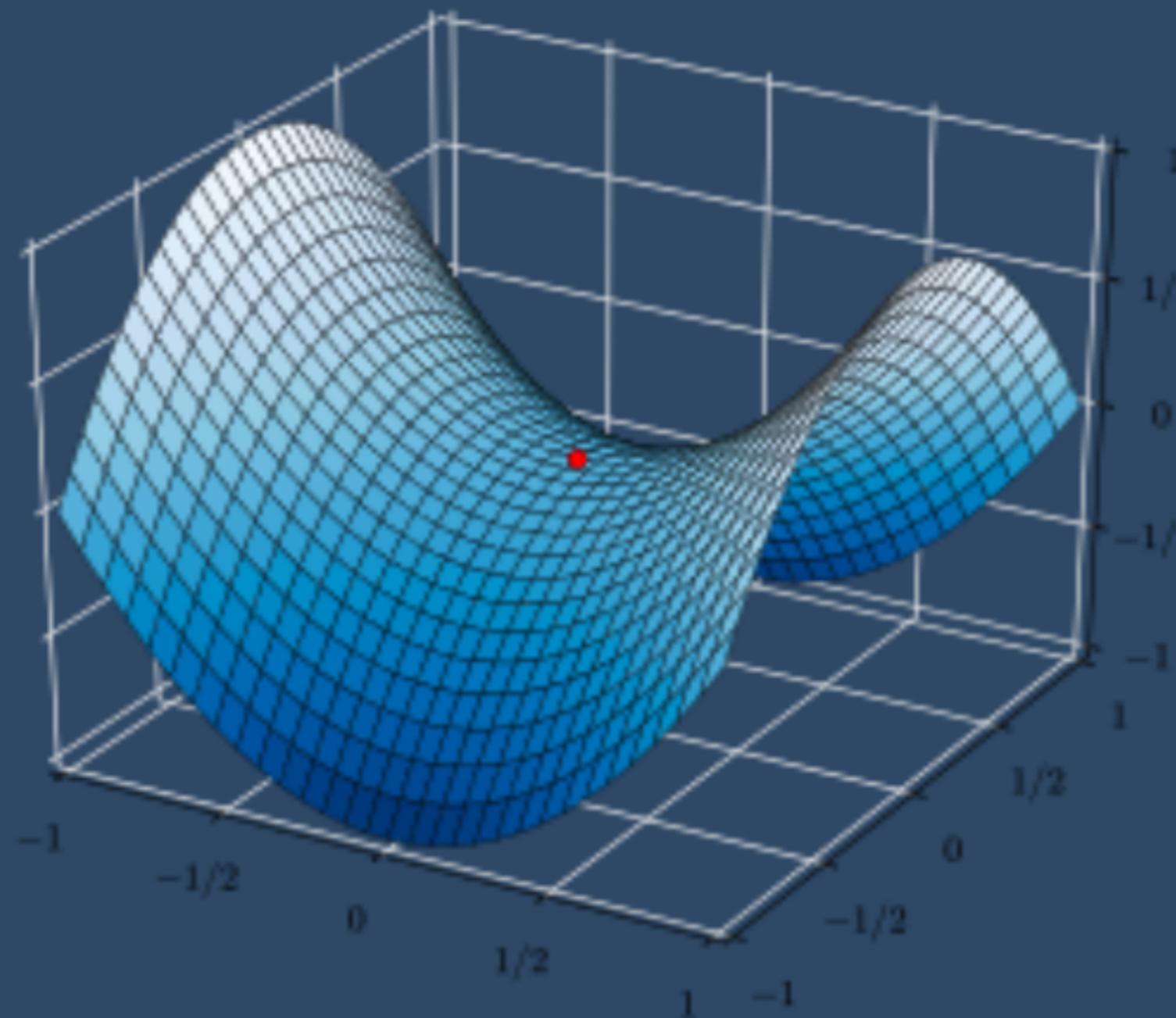
$$W \leftarrow W - \alpha \frac{\partial}{\partial W} L(W)$$

학습률이 너무 큰 경우,
학습이 제대로 이루어지지 않고
적절한 해에 수렴하지 못할 뿐 아니라
발산할 수 있다.



4. Gradient Descent

Gradient Descent의 한계



기울기가 0인 점에 도달하면
더 이상 학습이 이루어지지 않는다.

하지만, Saddle Point와 같은 점에서는 아무리
기울기가 0이어도 더 학습할 수 있고

Local Minimum에 도달하는 경우도
좋은 해가 아닐 수 있다.

4. Gradient Descent

Multiple Variables

$$f_W(x_n) = W_0 + W_1 x_{n1} + W_2 x_{n2} + \cdots + W_d x_{nd}$$

$$x_n = [x_{n0}, x_{n1}, \dots, x_{nd}]^T \in \mathbb{R}^{d+1}$$

$$W = [W_0, W_1, \dots, W_d]^T \in \mathbb{R}^{d+1}$$

$$f_W(x_n) = W^T x_n$$

4. Gradient Descent

Gradient Descent for Multiple Variables

$$W_i \leftarrow W_i - \alpha \frac{\partial}{\partial W_i} L(W)$$

$$W_i \leftarrow W_i - \alpha \frac{\partial}{\partial W_i} \frac{1}{N} \sum_{n=1}^N (f_W(x_n) - y_n)^2$$

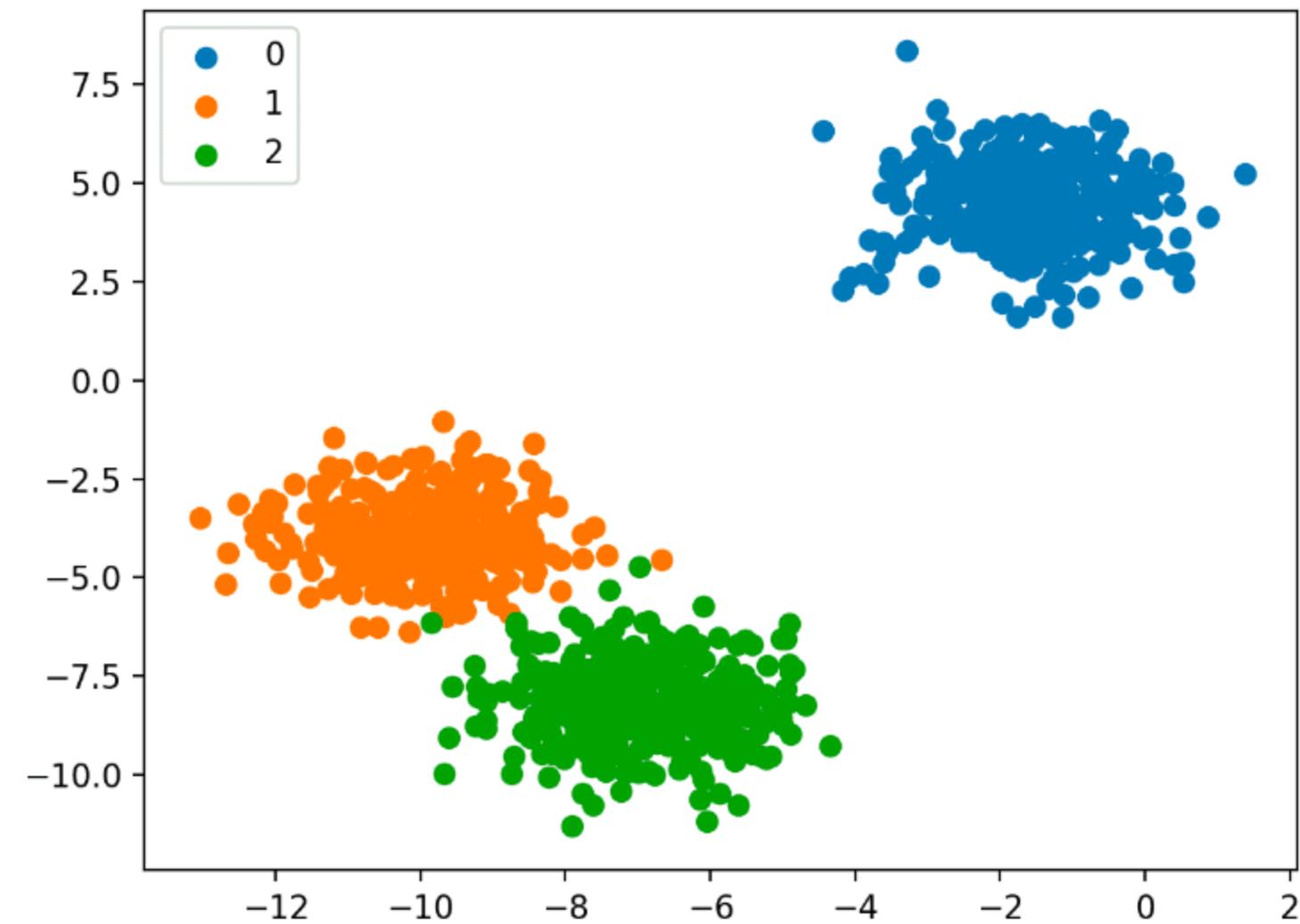
$$W_i \leftarrow W_i - \alpha \frac{2}{N} \sum_{n=1}^N (f_W(x_n) - y_n) x_{ni}$$

5. Classification

Regression과 유사하게,
Classification 또한
입력 변수와 출력 변수 사이의 관계를 찾는 문제이다.

하지만 Regression과 달리,
출력 변수는 이산적인 값을 갖는다.

Regression을 통해
Classification 문제를 해결할 수 있다.

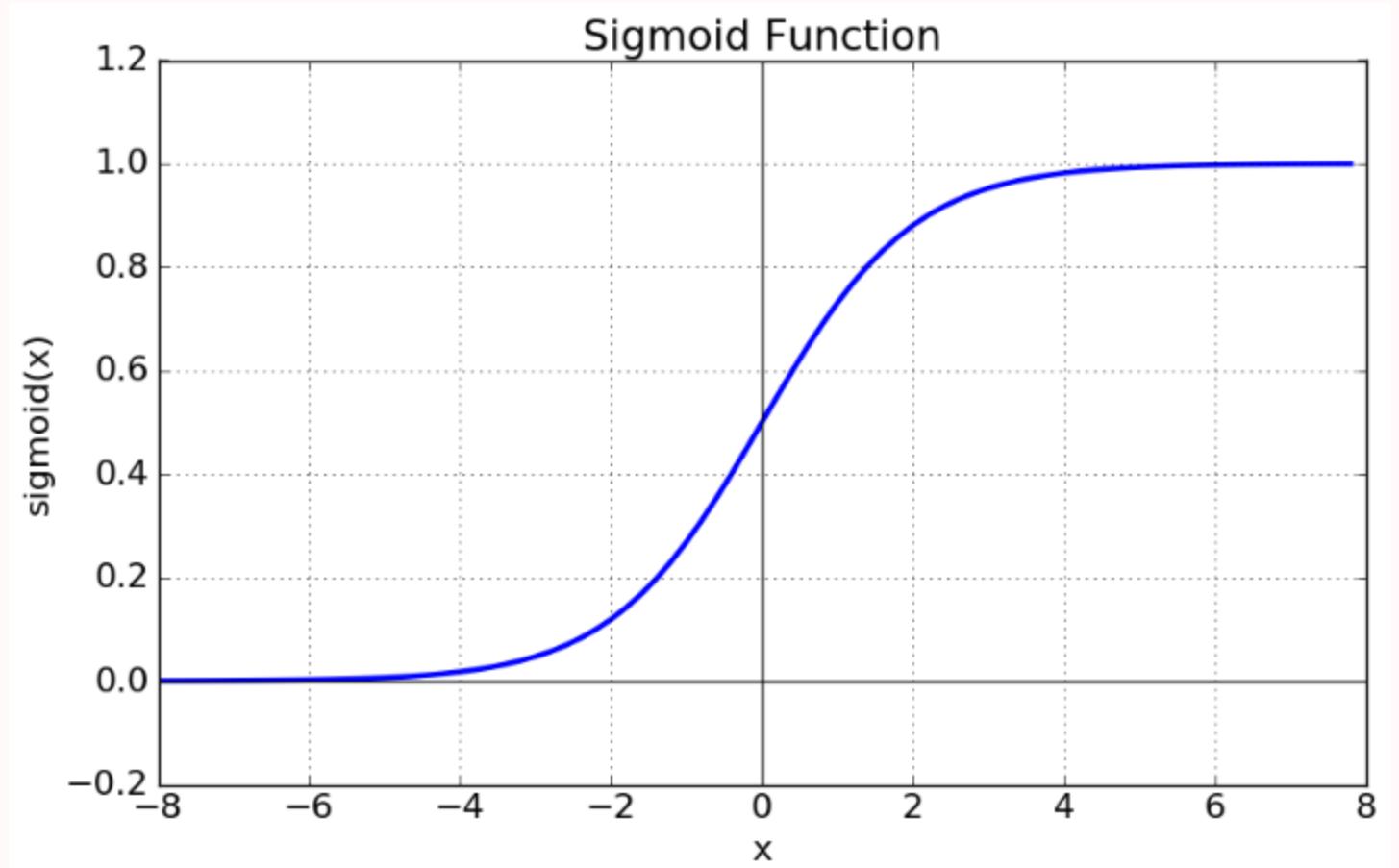


5. Classification

The Classification Function >> Sigmoid function / Logistic Function

$$0 \leq f_W(x) \leq 1$$
$$f_w(x) = \frac{1}{1 + \exp(-W^T x)}$$

$$\begin{cases} y = 1 & \text{if } f_W(x) \geq 0.5, W^T x \geq 0, \\ y = 0 & \text{if } f_W(x) < 0.5, W^T x < 0. \end{cases}$$



5. Classification

분류 함수를 통해 확률을 계산한다.

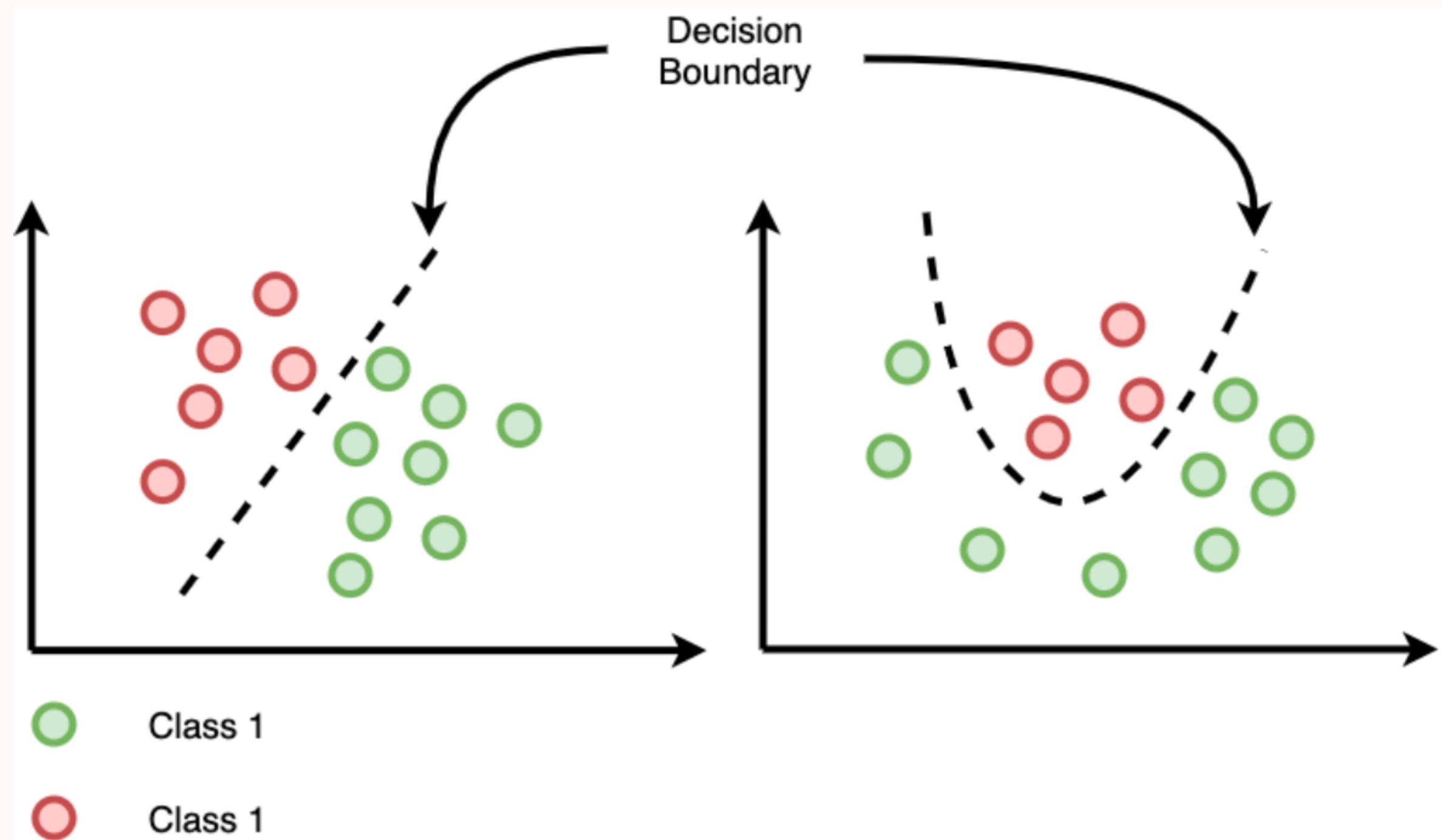
$$f_W(x) = P(y = 1|x, W)$$

$$\begin{aligned}P(y = 0|x, W) + P(y = 1|x, W) &= 1 \\P(y = 0|x, W) &= 1 - P(y = 1|x, W)\end{aligned}$$

$y = 1$ 는 특정 클래스에 포함된다는 의미이고,
따라서 함수는 특정 클래스에 포함될 확률을 계산한다.

함수의 의미는
"주어진 입력 x 와 파라미터 W 에 대하여 $y = 1$ 일 확률"을
의미한다.

5. Classification



Decision Boundary

6. Logistic Regression

학습 데이터셋

$$\{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$$

n번째 샘플

$$x_n = [x_{n0}, x_{n1}, \dots, x_{nd}]^T \in \mathbb{R}^{d+1}, \quad x_{n0} = 1, \quad y_n \in \{0, 1\}$$

mapping function

$$f_W(x) = \frac{1}{1 + \exp(-W^T x)}$$

6. Logistic Regression

엔트로피를 사용해 값을 늘림

$$C(f_W(x_n), y_n) = \begin{cases} -\log(f_W(x_n)) & \text{if } y_n = 1 \\ -\log(1 - f_W(x_n)) & \text{if } y_n = 0 \end{cases}$$

$$C(f_W(x_n), y_n) = -y_n \log(f_W(x_n)) - (1 - y_n) \log(1 - f_W(x_n))$$

6. Logistic Regression

$$\begin{aligned} L(W) &= \frac{1}{N} \sum_{n=1}^N C(f_W(x_n), y_n) \\ &= -\frac{1}{N} \sum_{n=1}^N [y_n \log(f_W(x_n)) + (1 - y_n) \log(1 - f_W(x_n))] \end{aligned}$$

6. Logistic Regression

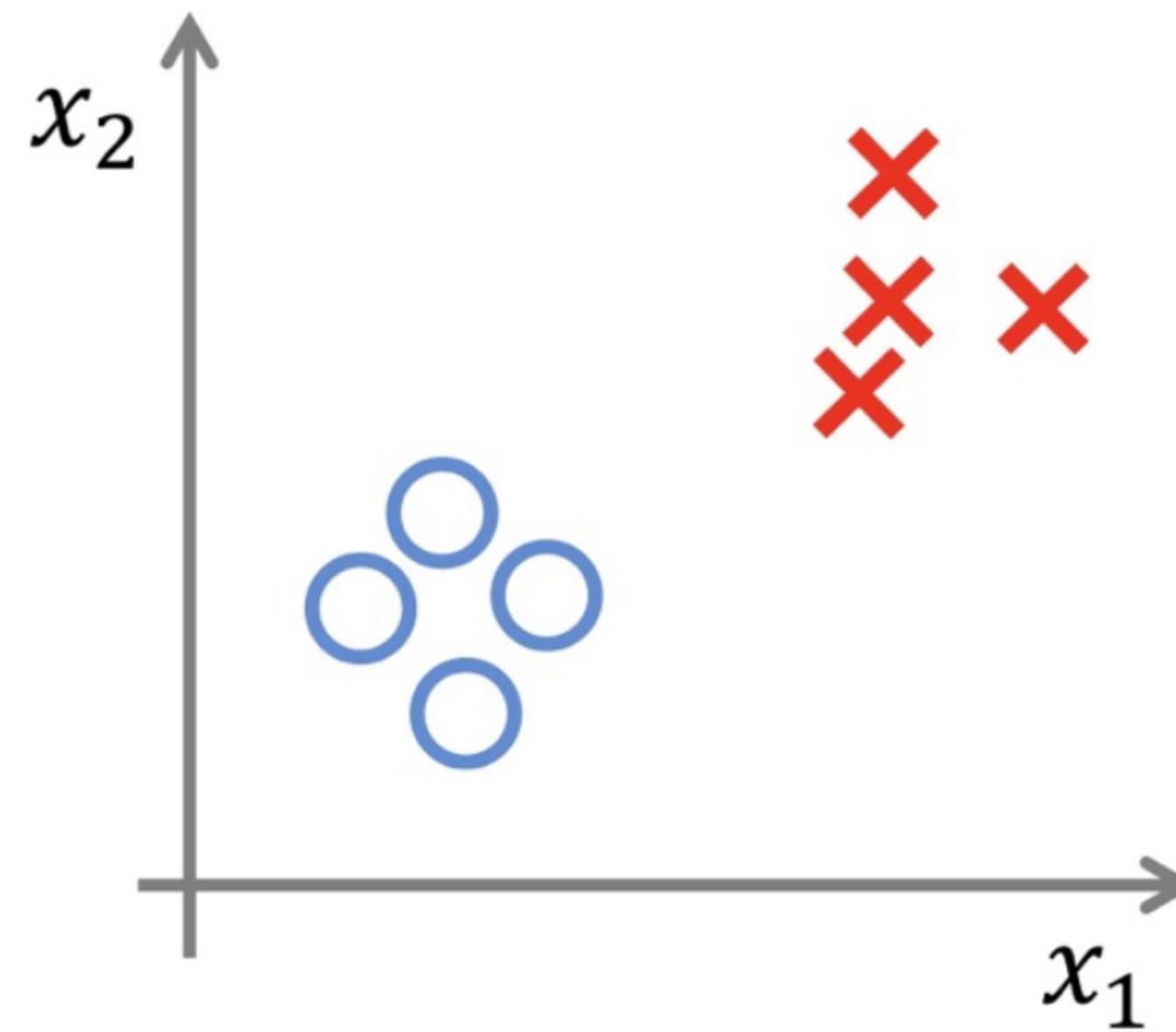
$$W_i \leftarrow W_i - \alpha \frac{\partial}{\partial W_i} L(W)$$

$$W_i \leftarrow W_i - \alpha \frac{1}{N} \sum_{n=1}^N (f_W(x_n) - y_n) x_{ni}$$

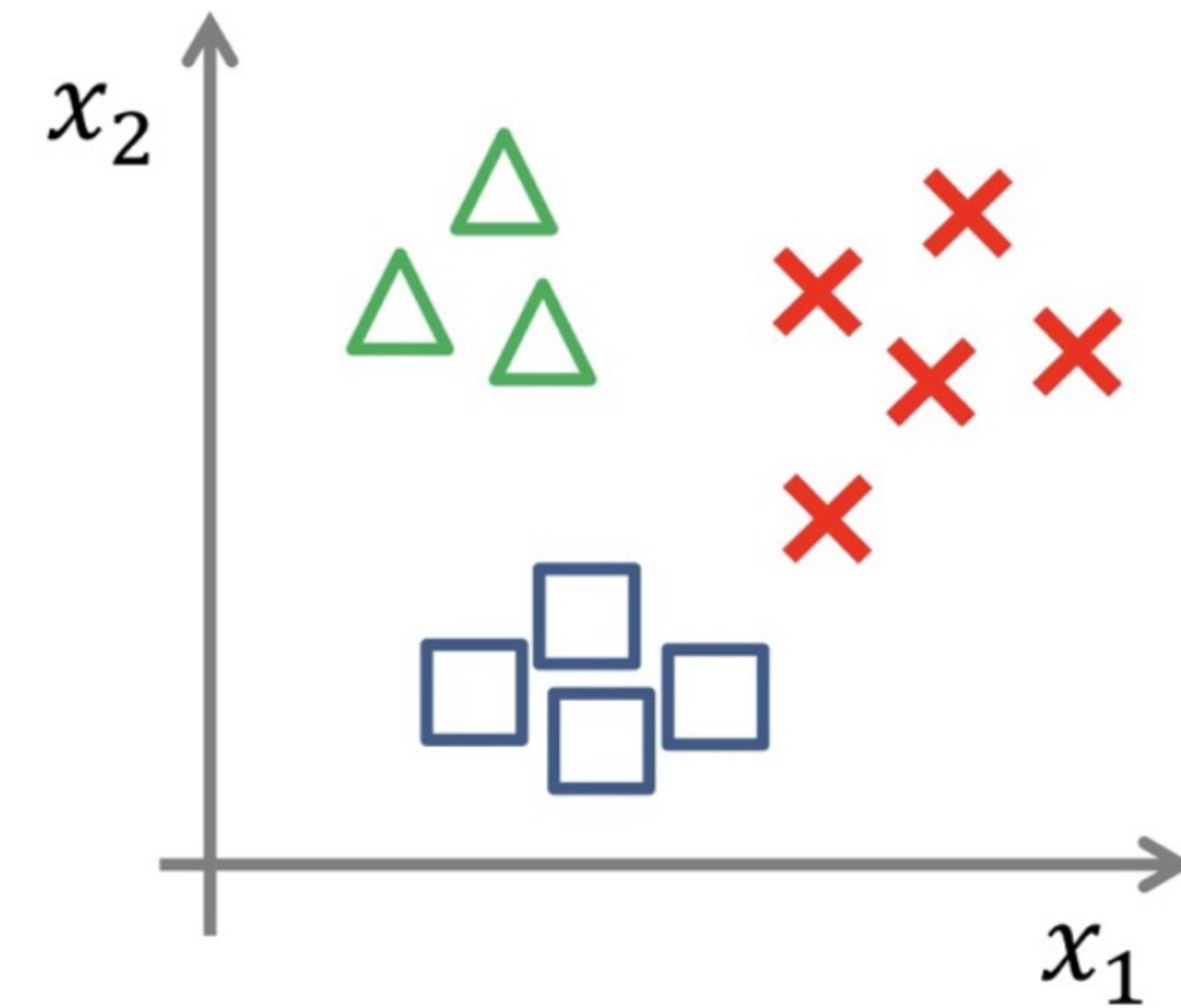
6. Logistic Regression

Multi-Class Classification: One vs. All

Binary Classification

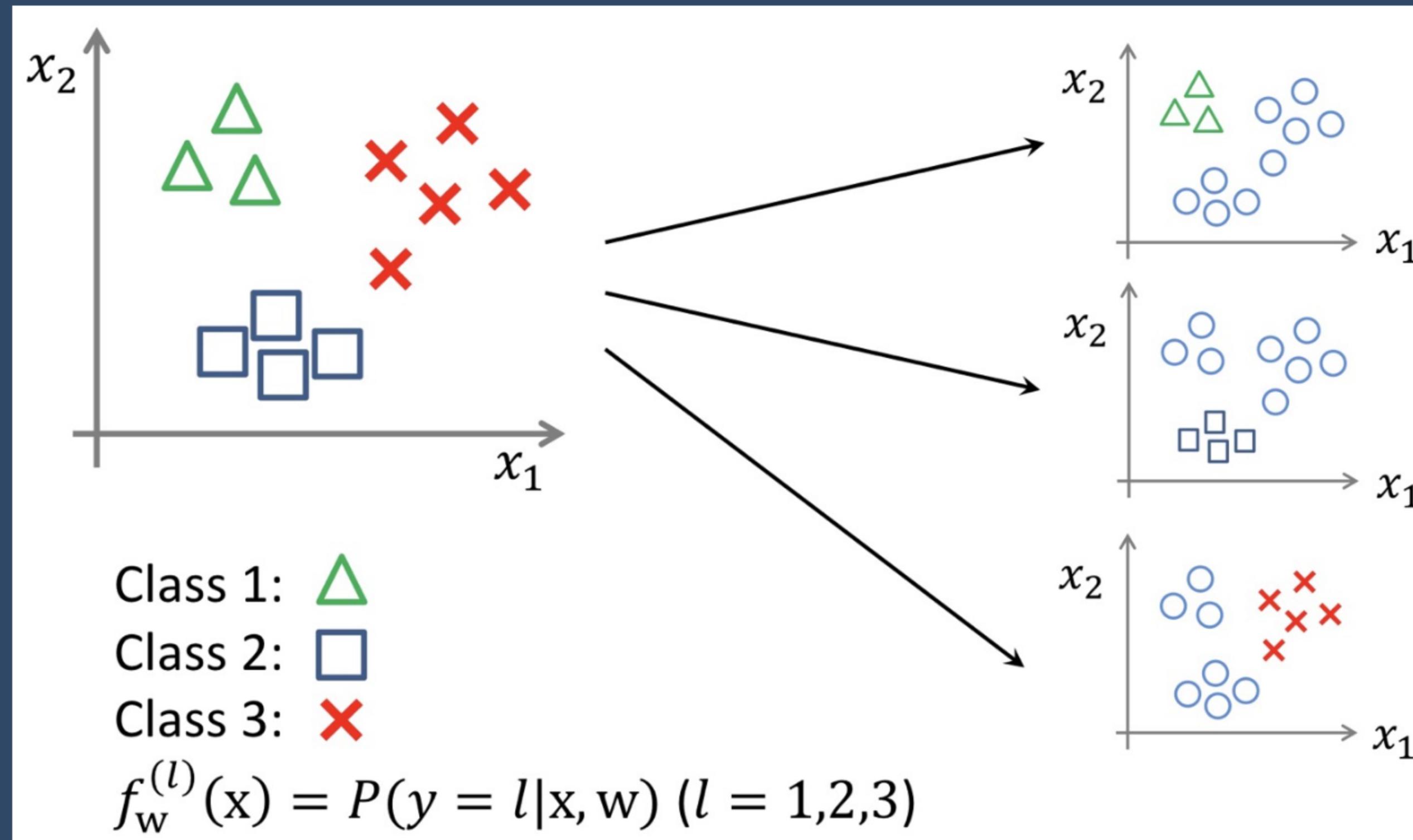


Multi-Class Classification



6. Logistic Regression

Multi-Class Classification: One vs. All



Thank you for listening

Deep Into Deep