7주차(3/3)

Adaline Gradient Descent

파이썬으로 배우는 기계학습

한동대학교
김영섭 교수

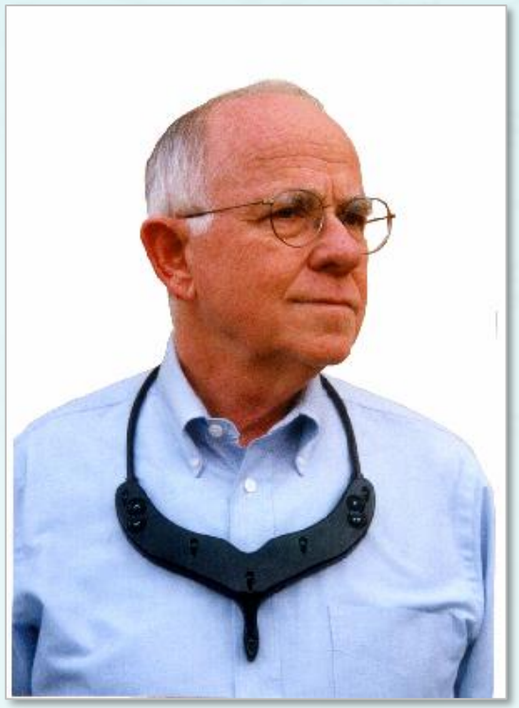
Adaline and Gradient Descent

- Objectives
  - Differences between Adaline and Perceptron Algorithm
  - Minimizing Error Using Cost Function
  - Finding Minimal Using Gradient Descent

- Contents
  - Adaline Algorithm
  - Cost Function
  - Gradient Descent

1. Adaline and Gradient Descent Comparison

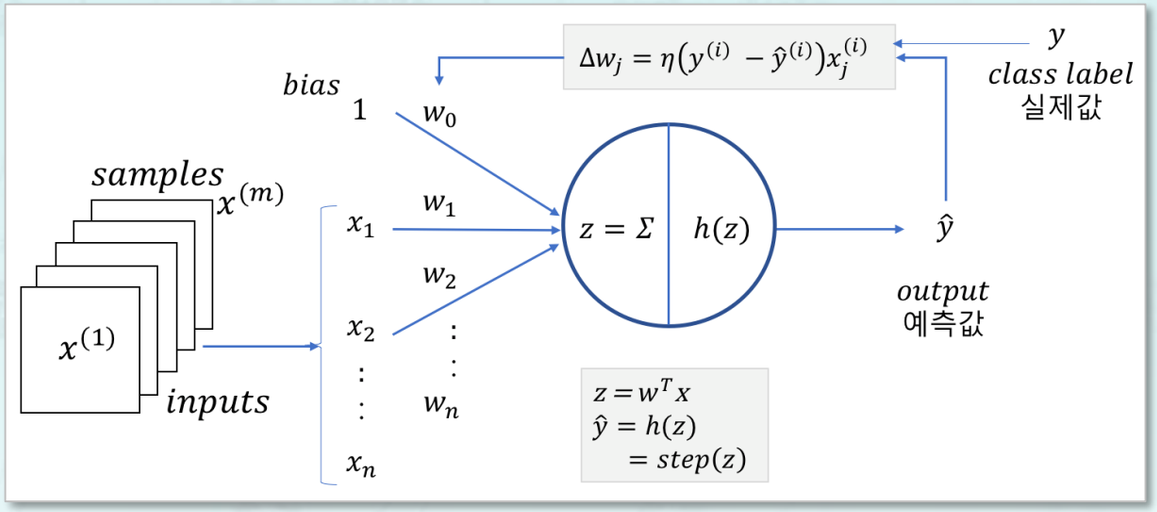
- **Adaline Algorithm**
  - Adaptive Linear Neuron
  - Adaline



Bernard Widrow

1. Adaline and Gradient Descent Comparison

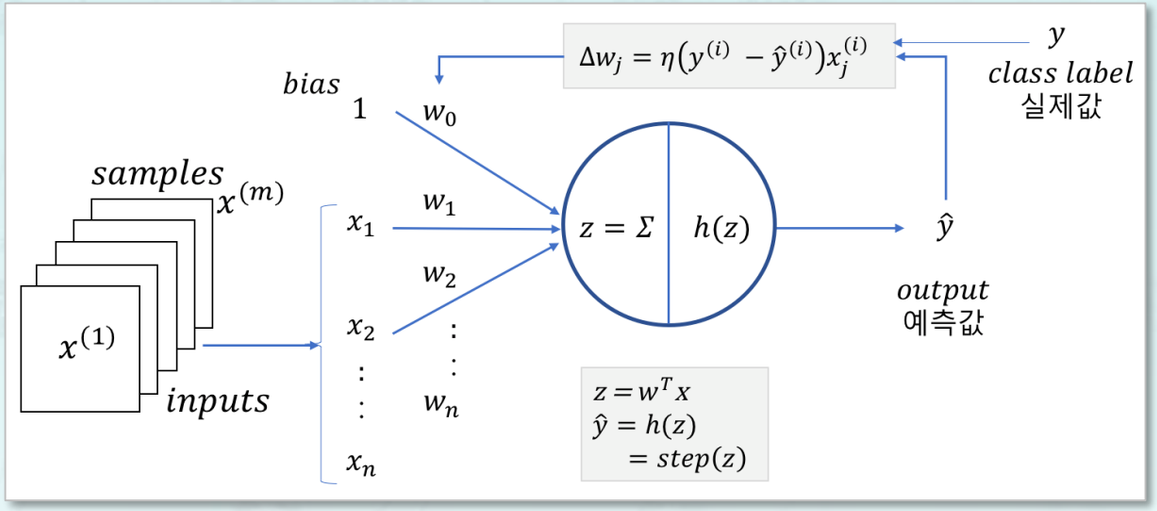
- **Adaline Algorithm**
  - **Perceptron**



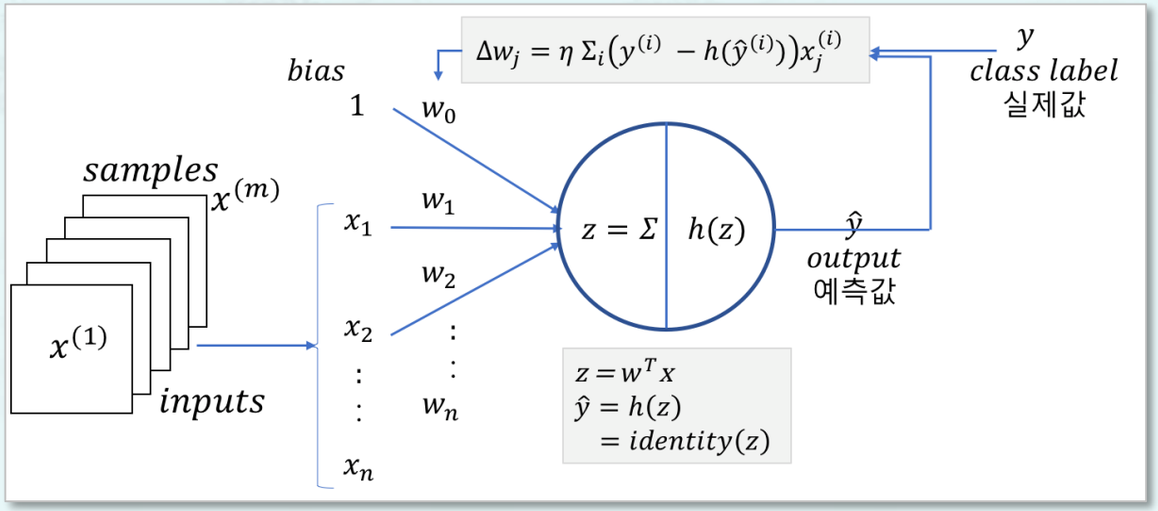
  - **Adaline**

1. Adaline and Gradient Descent Comparison

- **Adaline Algorithm**
  - **Perceptron**

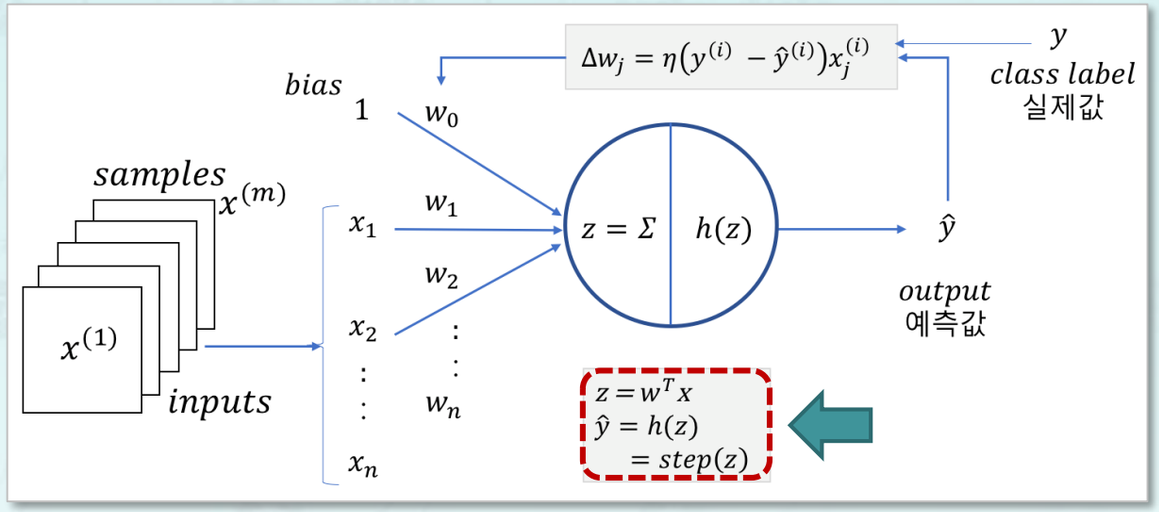


  - **Adaline**

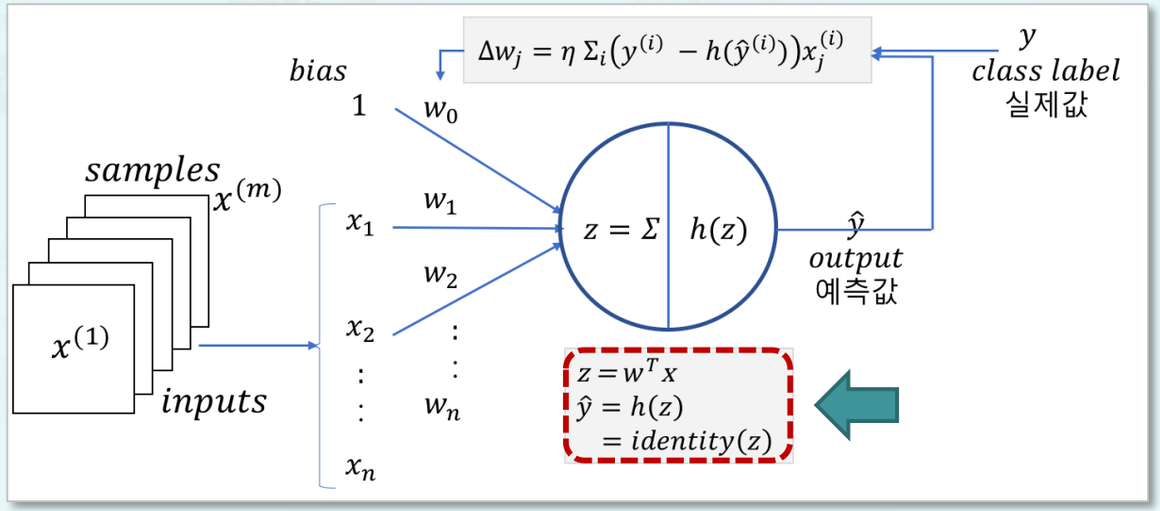


1. Adaline and Gradient Descent Comparison

- **Adaline Algorithm**
  - **Perceptron**
    - **Activation Function : Step Function**

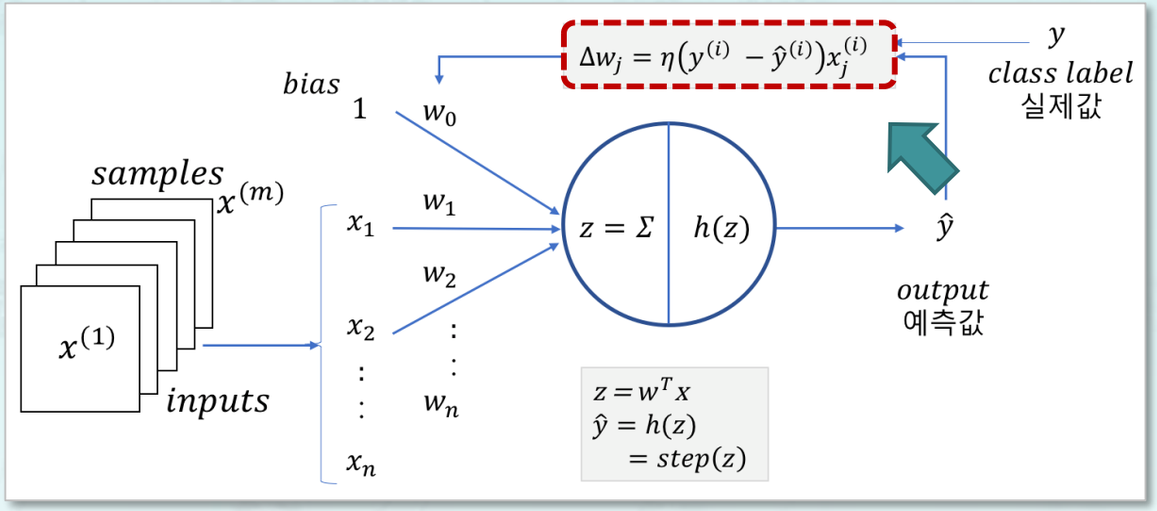


  - **Adaline**
    - **Activation Function : Identity Function**

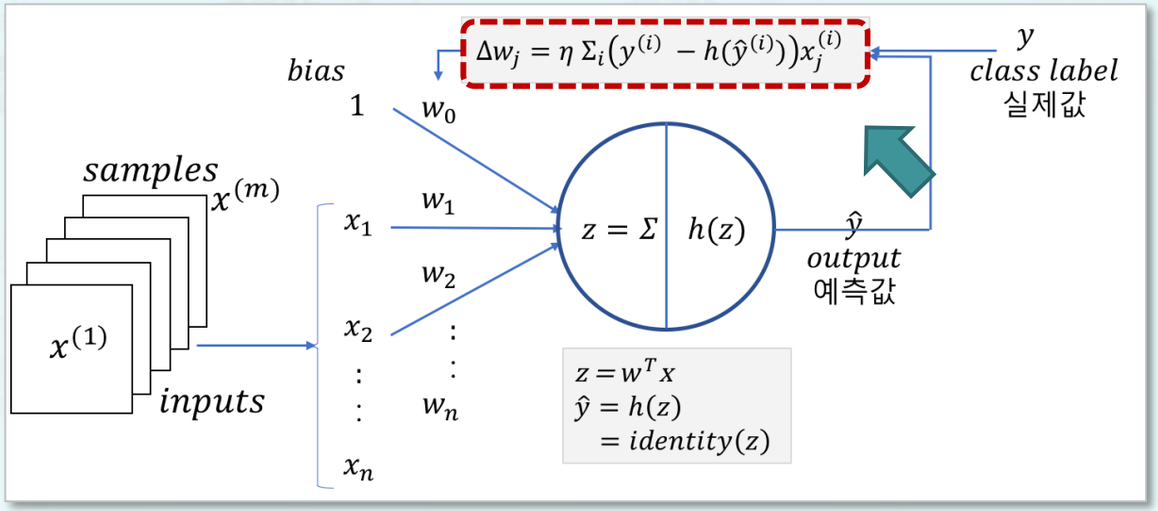


1. Adaline and Gradient Descent Comparison

- **Adaline Algorithm**
  - **Perceptron**
    - **Activation Function : Step Function**
    - **Delta W : Every Sample**

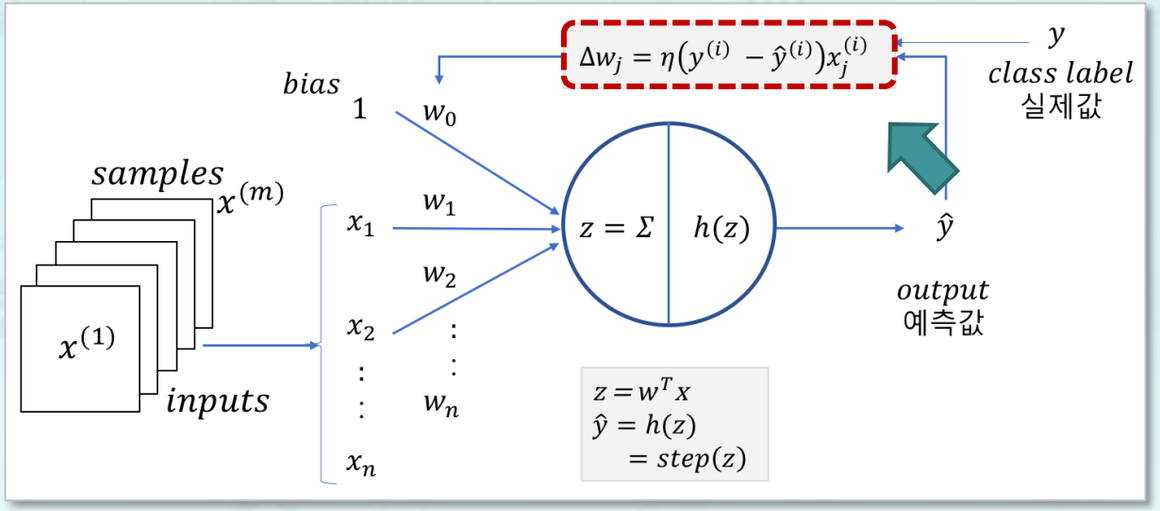


  - **Adaline**
    - **Activation Function : Identity Function**
    - **Delta W : All Samples Once**

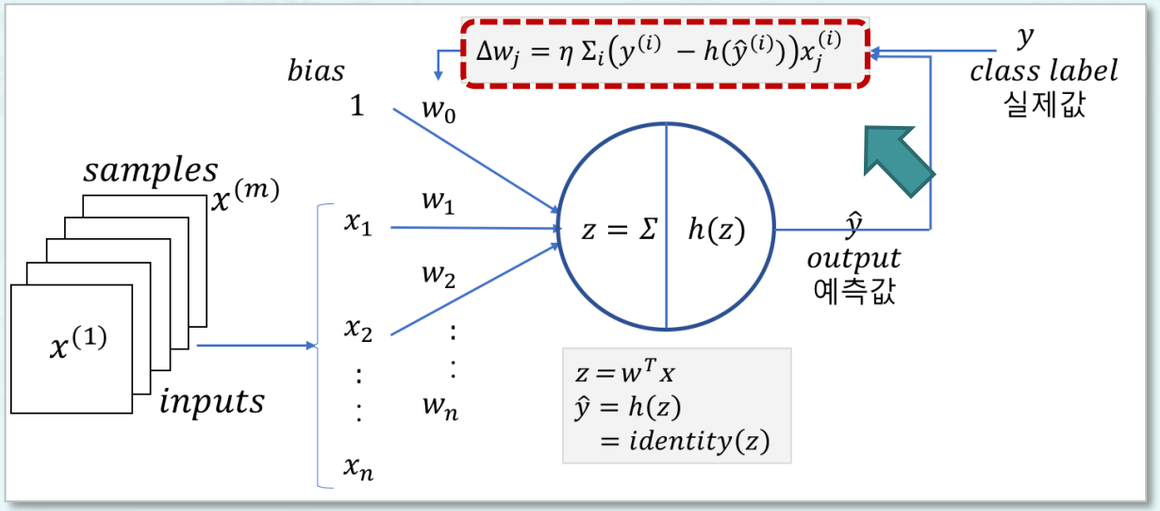


1. Adaline and Gradient Descent Comparison

- **Adaline Algorithm**
  - **Perceptron**
    - Activation Function : Step Function
    - Delta W : Every Sample
    - Error : (-2, 0, -2)



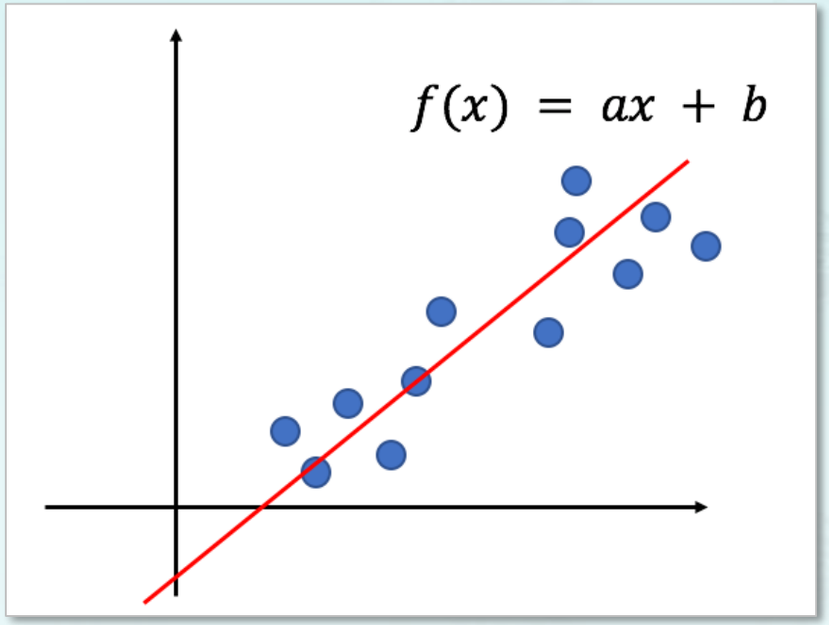
  - **Adaline**
    - Activation Function : Identity Function
    - Delta W : All Samples Once
    - Error : Quantized Error



2. Cost Function

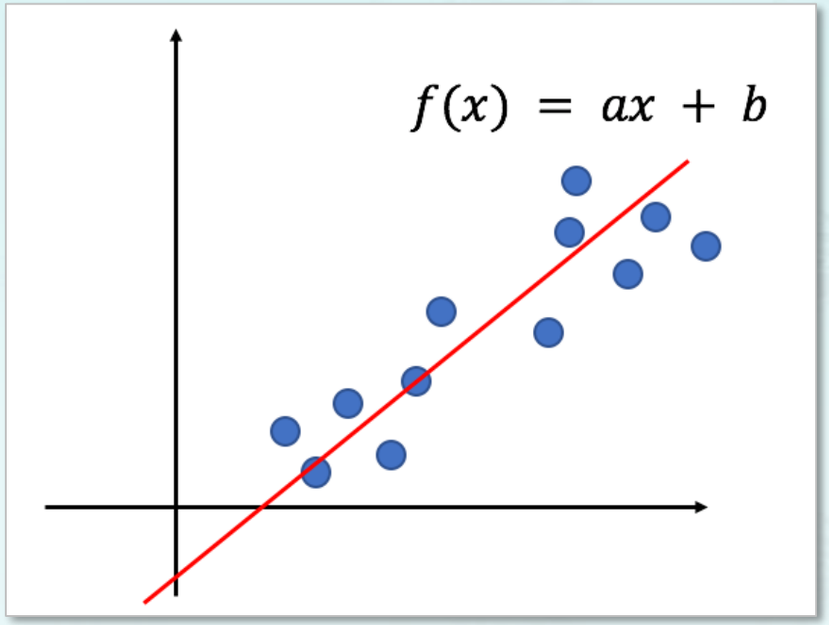
2. Cost Function

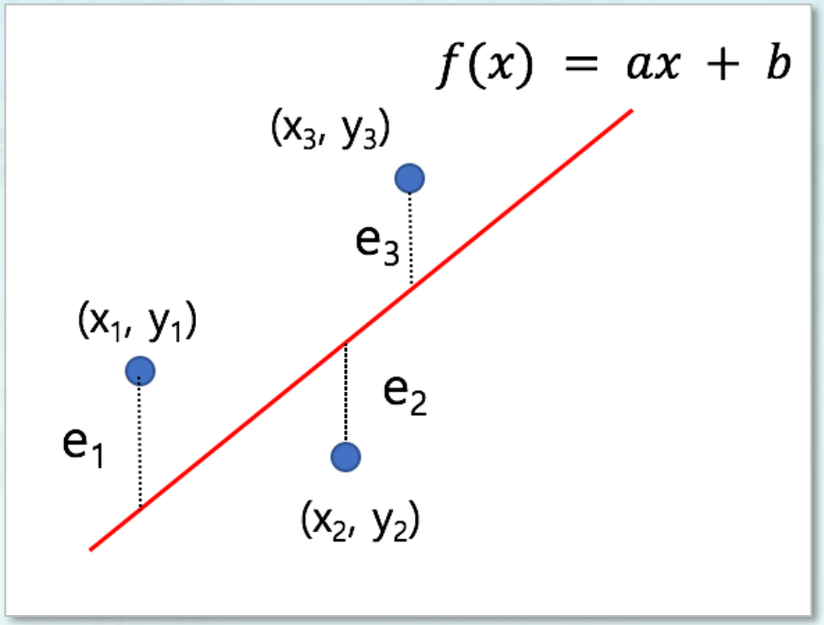
- **Sum of Squared Error (SSE)**



2. Cost Function

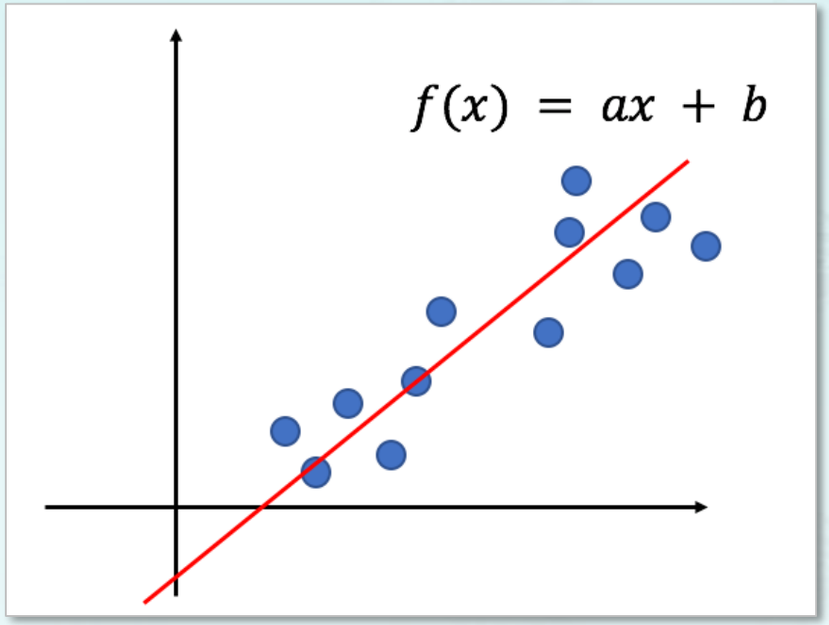
- **Sum of Squared Error (SSE)**

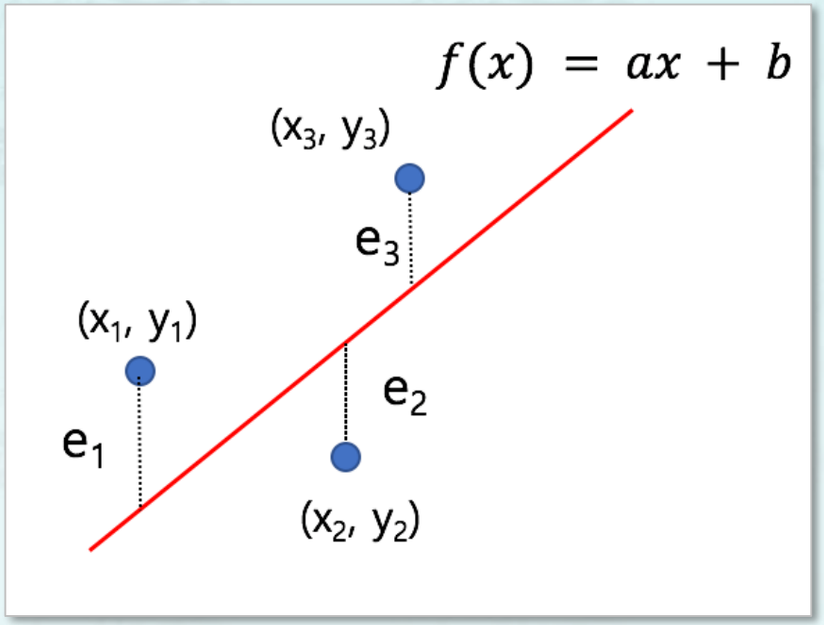




2. Cost Function

- **Sum of Squared Error (SSE)**





$$E(a,b) = \sum_{i=1}^{n} \left(y_i - (ax_i + b)\right)^2$$

2. Cost Function

- **Cost Function J(w) Using Mean Squared**

$$J(w) = \frac{1}{2}\sum_{i=1}^{m}\left(y^{(i)} - \hat{y}^{(i)}\right)^2$$

$$E(a,b) = \sum_{i=1}^{n}\left(y_i - (ax_i + b)\right)^2$$

2. Cost Function

- **Cost Function Using Mean Squared**

$$J(w) = \frac{1}{2}\sum_{i=1}^{m}\left(y^{(i)} - \hat{y}^{(i)}\right)^2$$

2. Cost Function

- **Cost Function Using Mean Squared**

$$J(w) = \frac{1}{2}\sum_{i=1}^{m}\left(y^{(i)} - \hat{y}^{(i)}\right)^2$$

2. Cost Function

- **Cost Function Using Mean Squared**

$$J(w) = \frac{1}{2} \sum_{i=1}^{m} \left(y^{(i)} - \hat{y}^{(i)} \right)^2$$

2. Cost Function

- **Cost Function Using Mean Squared**

$$J(w) = \frac{1}{2}\sum_{i=1}^{m}\left(y^{(i)} - \hat{y}^{(i)}\right)^2$$

$$= \frac{1}{2}\sum_{i=1}^{m}\left(y^{(i)} - h\left(z^{(i)}\right)\right)^2 \quad \cdots\cdots\cdots\cdots \text{식(1)}$$

$$= \frac{1}{2}\sum_{i=1}^{m}\left(y^{(i)} - z^{(i)}\right)^2$$

2. Cost Function

- **Cost Function Using Mean Squared**

$$J(w) = \frac{1}{2}\sum_{i=1}^{m}\left(y^{(i)} - \hat{y}^{(i)}\right)^2$$

$$= \frac{1}{2}\sum_{i=1}^{m}\left(y^{(i)} - h\left(z^{(i)}\right)\right)^2 \quad \cdots\cdots\cdots\cdots \quad 식(1)$$

$$= \frac{1}{2}\sum_{i=1}^{m}\left(y^{(i)} - z^{(i)}\right)^2$$

$$= \frac{1}{2}\sum_{i=1}^{m}\left(y^{(i)} - \sum_{j=1}^{n}\left(w_j x_j^{(i)}\right)\right)^2 \quad \cdots\cdots\cdots\cdots \quad 식(2)$$

2. Cost Function

- **Cost Function Using Mean Squared**

$$J(w) = \frac{1}{2}\sum_{i=1}^{m}\left(y^{(i)} - \hat{y}^{(i)}\right)^2$$

$$= \frac{1}{2}\sum_{i=1}^{m}\left(y^{(i)} - h\left(z^{(i)}\right)\right)^2 \quad \cdots\cdots\cdots\cdots \quad 식(1)$$

$$= \frac{1}{2}\sum_{i=1}^{m}\left(y^{(i)} - z^{(i)}\right)^2$$

$$= \frac{1}{2}\sum_{i=1}^{m}\left(y^{(i)} - \sum_{j=1}^{n}\left(w_j x_j^{(i)}\right)\right)^2 \quad \cdots\cdots\cdots\cdots \quad 식(2)$$

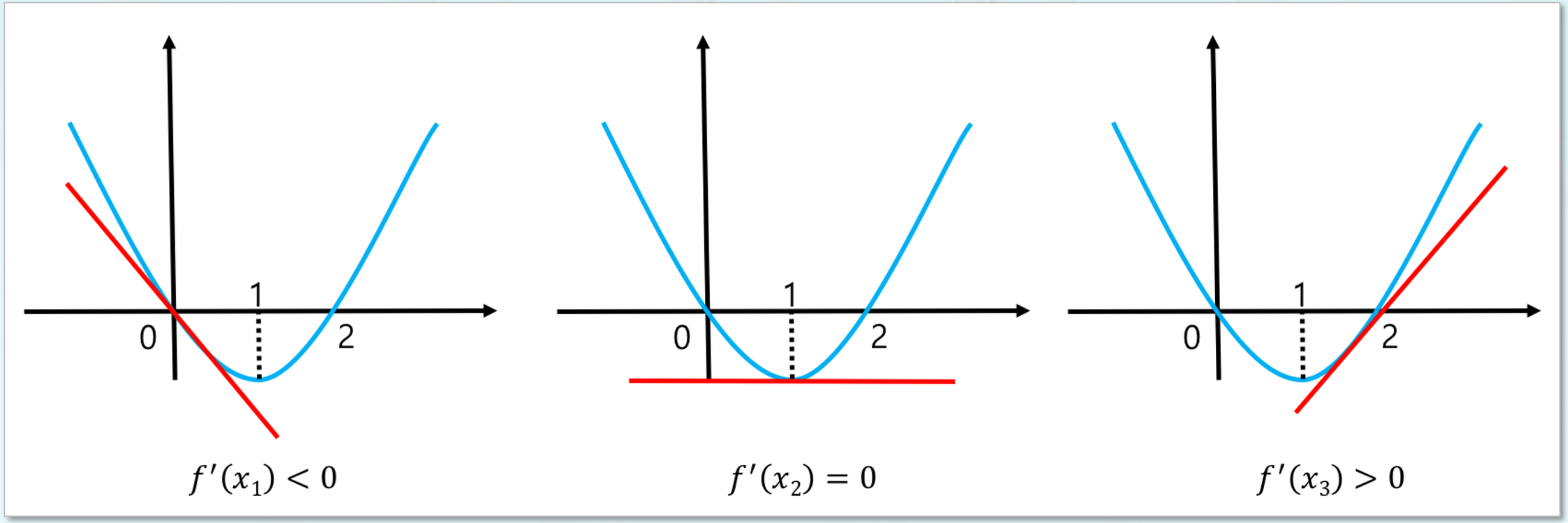
2. Cost Function

- **Cost Function Using Mean Squared**
- **Minimum in Cost Function ?**

$$J(w) = \frac{1}{2}\sum_{i=1}^{m}\left(y^{(i)} - \sum_{j=1}^{n}\left(w_j x_j^{(i)}\right)\right)^2$$

2. Cost Function

- **Gradient Descent**
  - **Derivative of 2nd order function**
  - $f(x) = x^2 - 2x$



2. Cost Function

- **Cost Function Using Mean Squared**

$$J(w) = \frac{1}{2}\sum_{i=1}^{m}\left(y^{(i)} - \hat{y}^{(i)}\right)^2$$

$$= \frac{1}{2}\sum_{i=1}^{m}\left(y^{(i)} - h\left(z^{(i)}\right)\right)^2 \quad \cdots\cdots\cdots\cdots \quad 식(1)$$

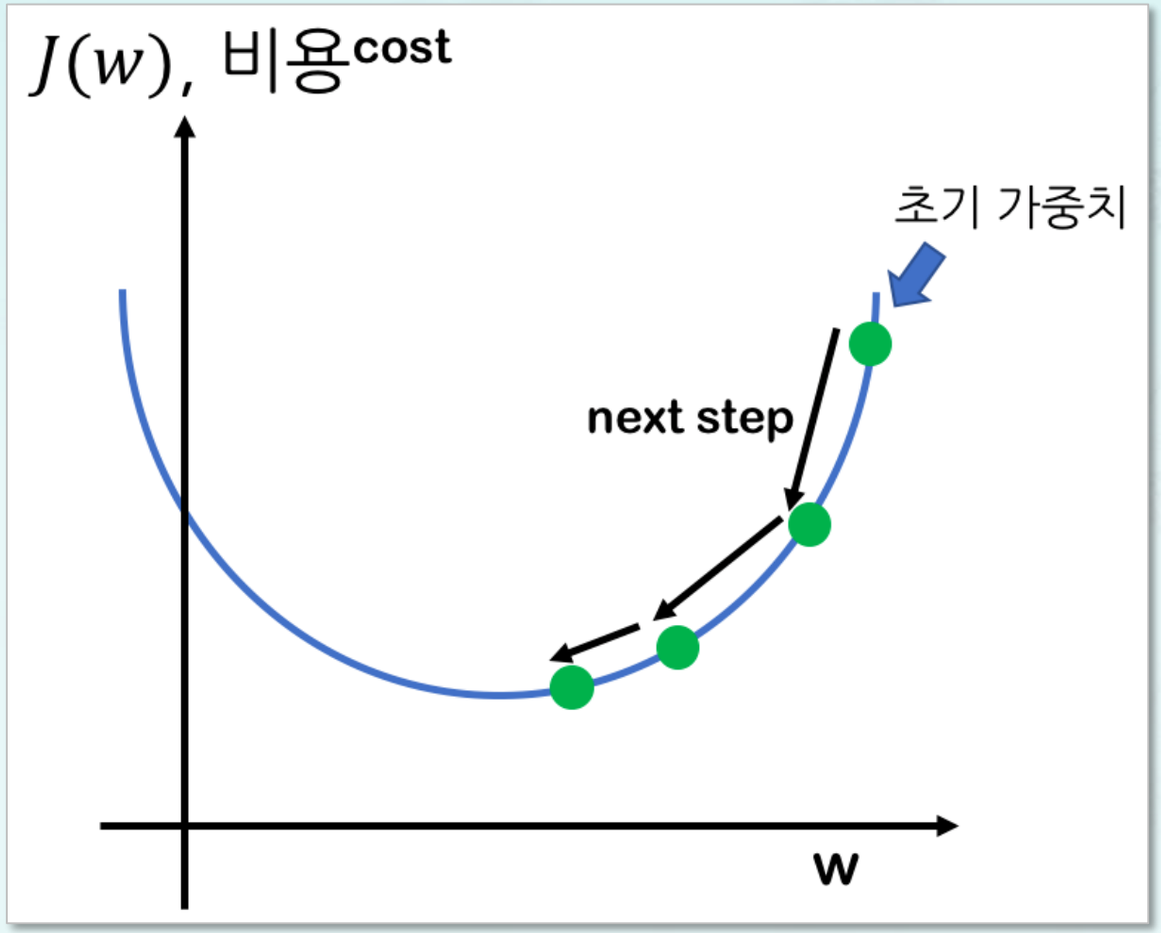
$$= \frac{1}{2}\sum_{i=1}^{m}\left(y^{(i)} - z^{(i)}\right)^2$$

$$= \frac{1}{2}\sum_{i=1}^{m}\left(y^{(i)} - \sum_{j=1}^{n}\left(w_j x_j^{(i)}\right)\right)^2 \quad \cdots\cdots\cdots\cdots \quad 식(2)$$

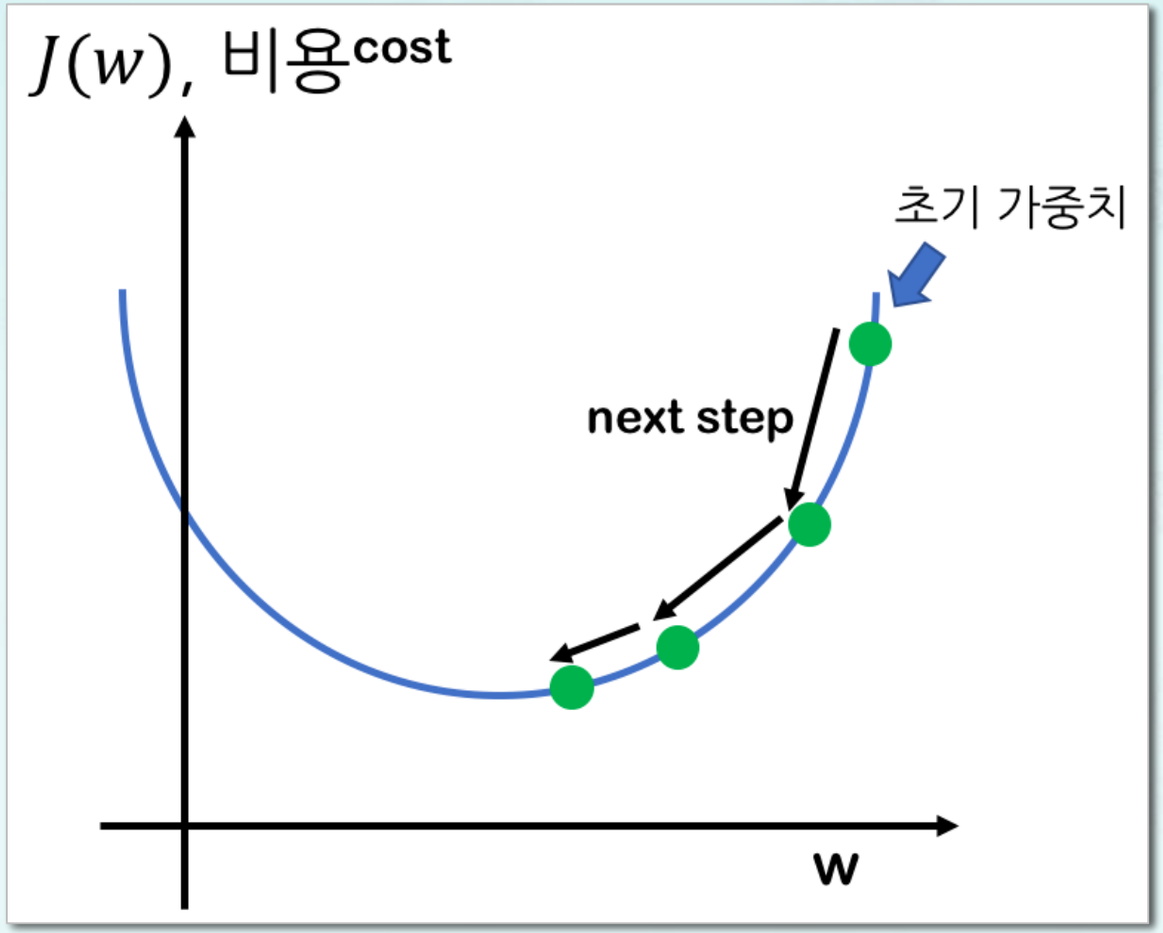
3. Gradient Descent

- Numerical method, not Algebraic method

3. Gradient Descent

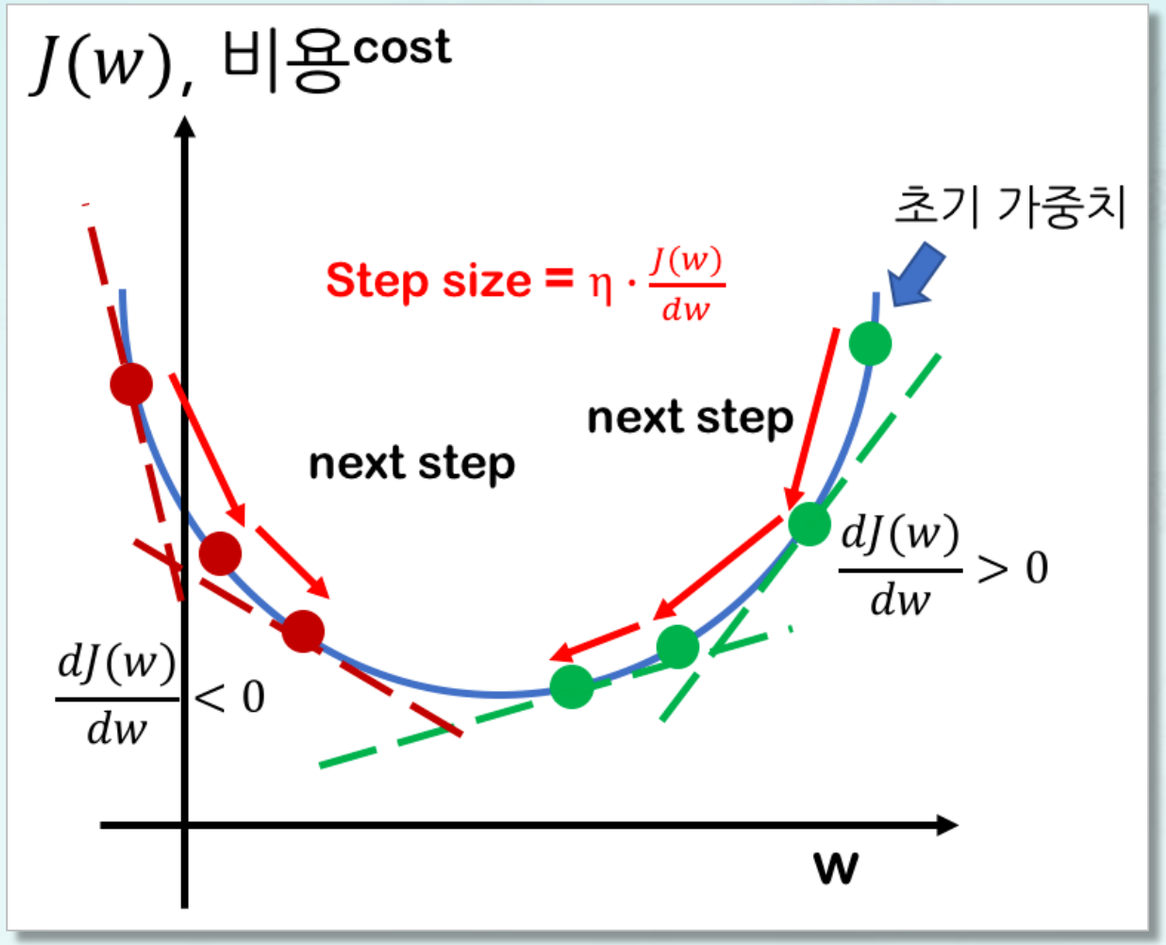


3. Gradient Descent



- **Step Direction :** $-\frac{dJ(w)}{dw}$

3. Gradient Descent



- **Step Direction :** $-\frac{dJ(w)}{dw}$
- **Step Size :** $-\eta \cdot \frac{dJ(w)}{dw}$

Adaline and Gradient Descent

- Summary
  - Adaline Algorithm
  - Adaline and Perceptron Algorithm Difference
  - Cost Function
  - Gradient Descent

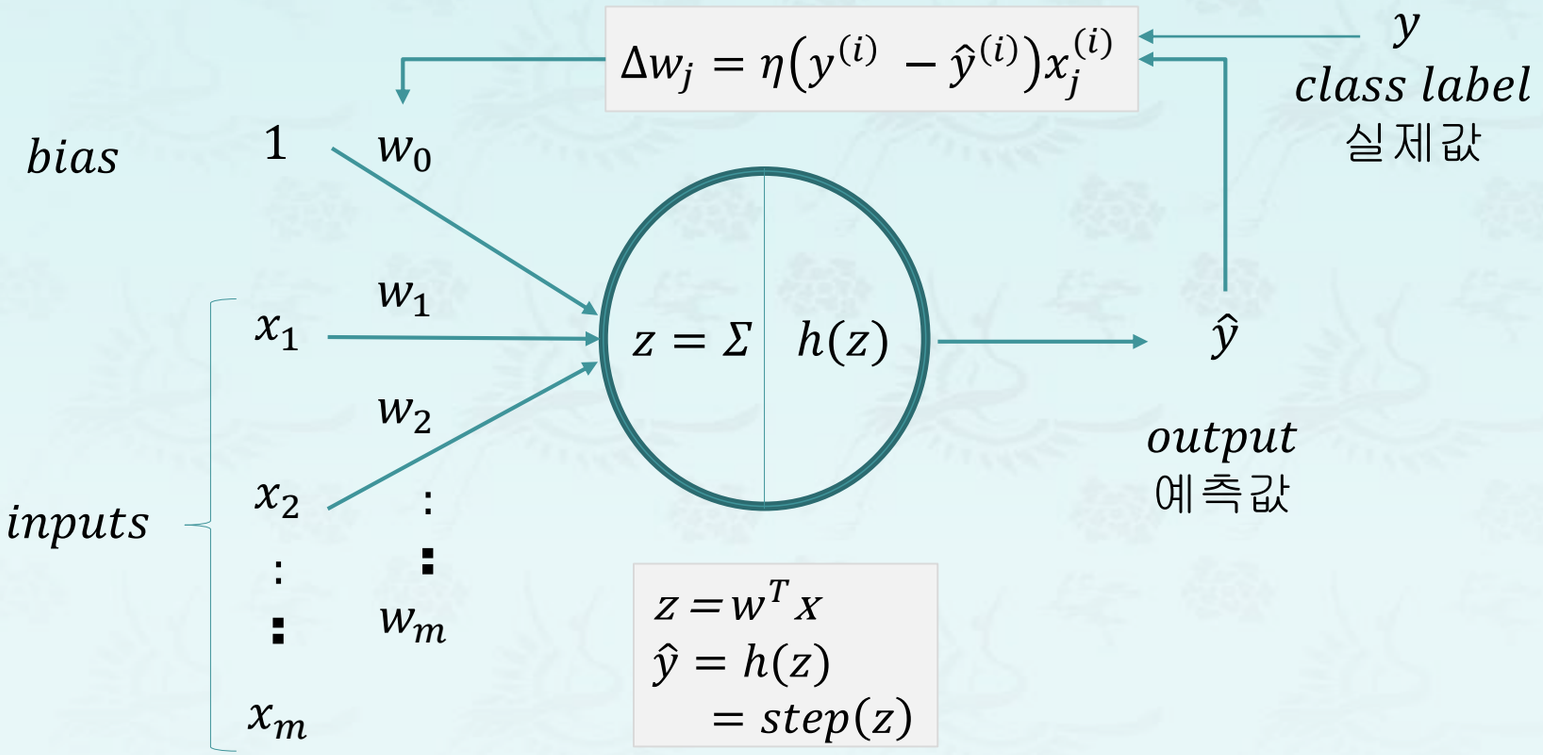
- Next
  - 8.1 Adaline Gradient Descent Implementation

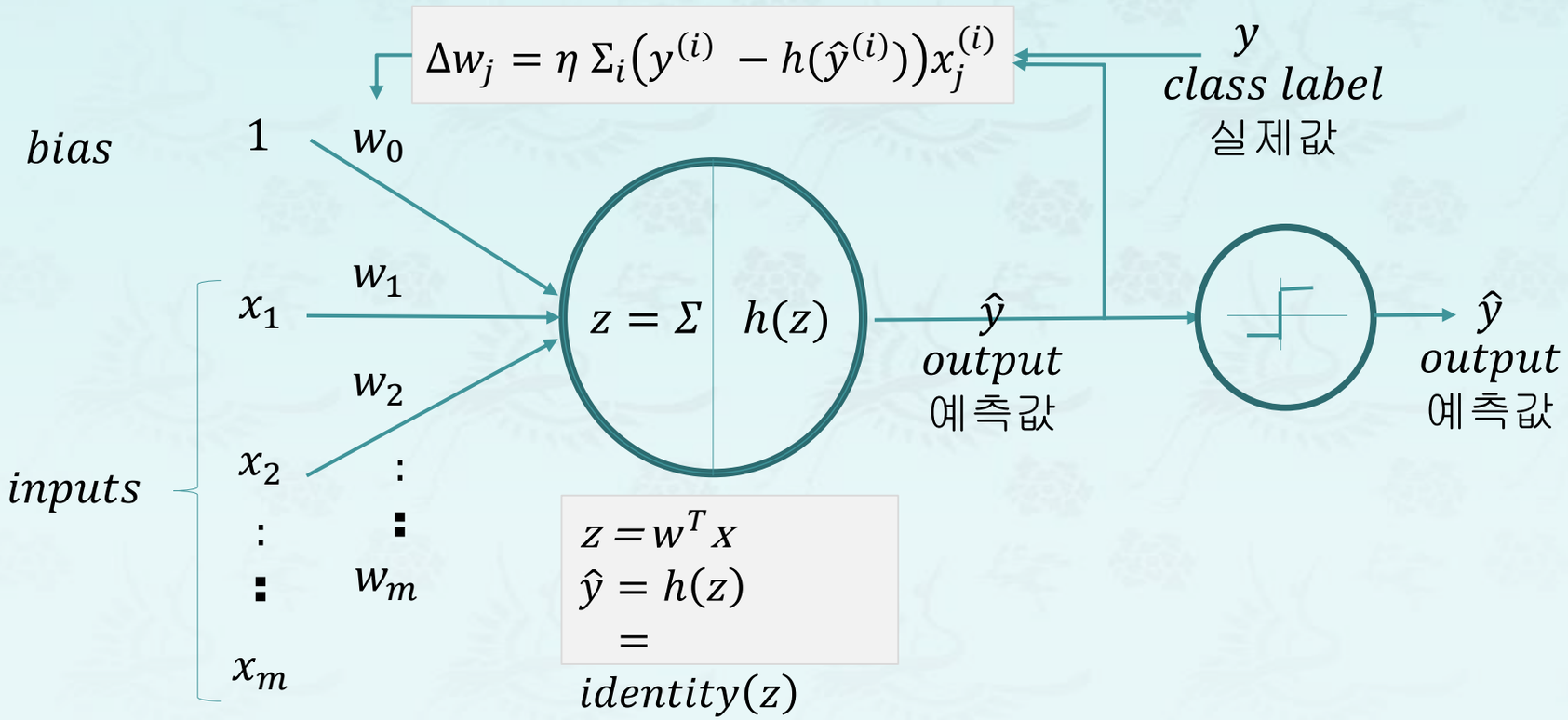
7주차(3/3)

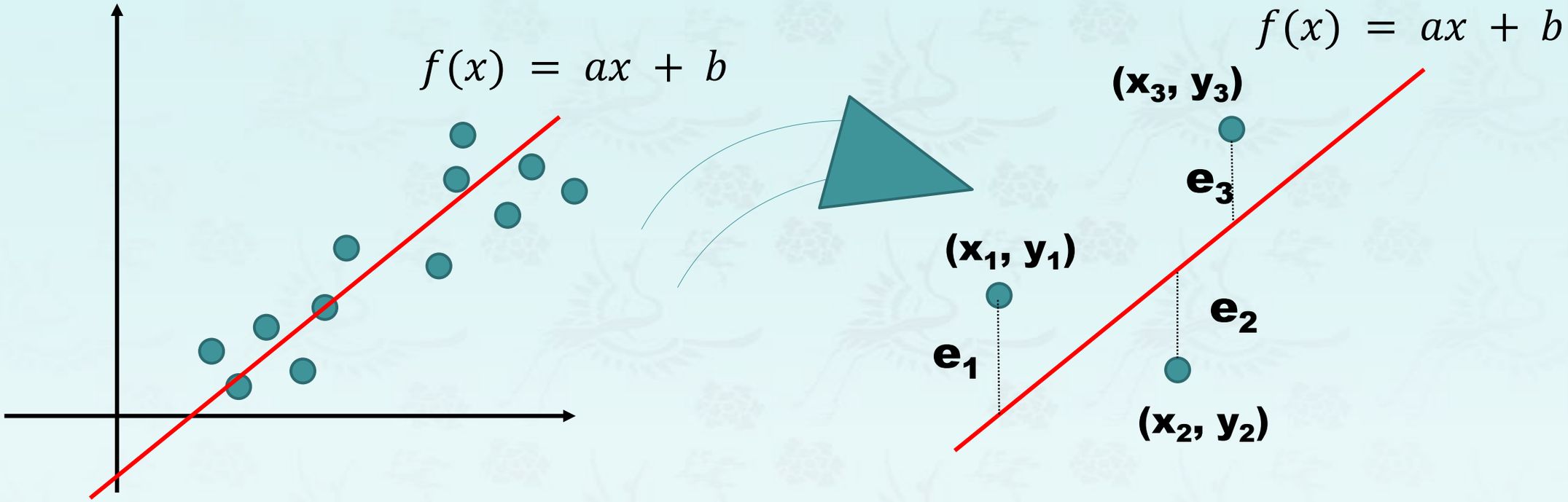
Adaline and Gradient Descent

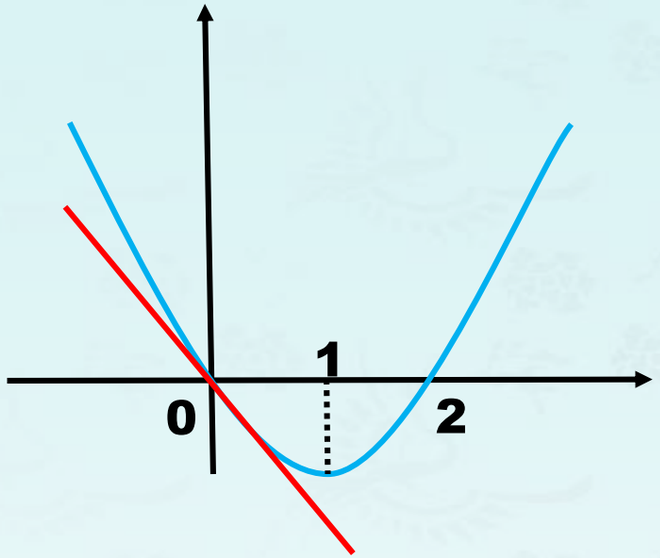
파이썬으로 배우는 기계학습

한동대학교
김영섭 교수

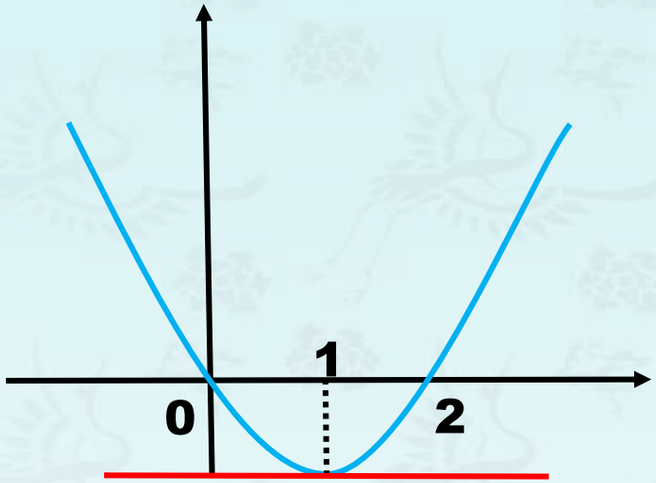

$\Delta w_j = \eta\left(y^{(i)} - \hat{y}^{(i)}\right)x_j^{(i)}$
y
$class\ label$
실제값
$bias$
1
w_0
w_1
x_1
w_2
x_2
w_m
x_m
$inputs$
$z = \Sigma$
$h(z)$
$\hat{y}$
$output$
예측값
$z = w^T x$
$\hat{y} = h(z)$
$= step(z)$


$\Delta w_j = \eta \Sigma_i \left(y^{(i)} - h(\hat{y}^{(i)})\right) x_j^{(i)}$
y
$class\ label$
실제값
$bias$
1
w_0
w_1
x_1
w_2
x_2
$inputs$
x_m
w_m
$z = \Sigma$
$h(z)$
$\hat{y}$
$output$
예측값
$\hat{y}$
$output$
예측값
$z = w^T x$
$\hat{y} = h(z)$
$= identity(z)$

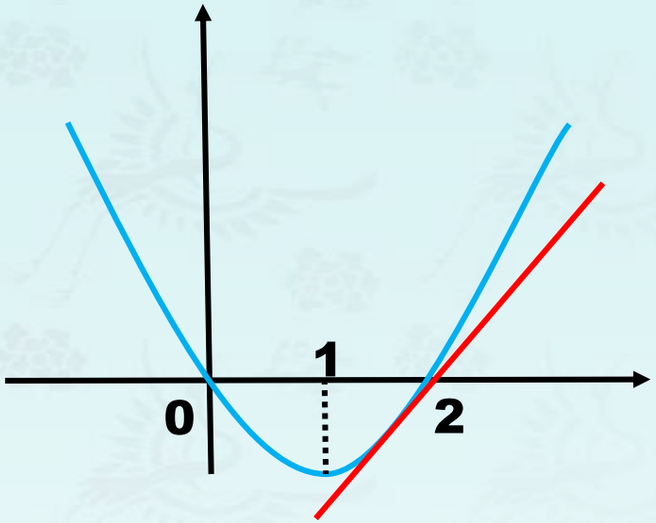

$f(x) = ax + b$
$f(x) = ax + b$
(x_3, y_3)
e_3
(x_1, y_1)
e_2
e_1
(x_2, y_2)



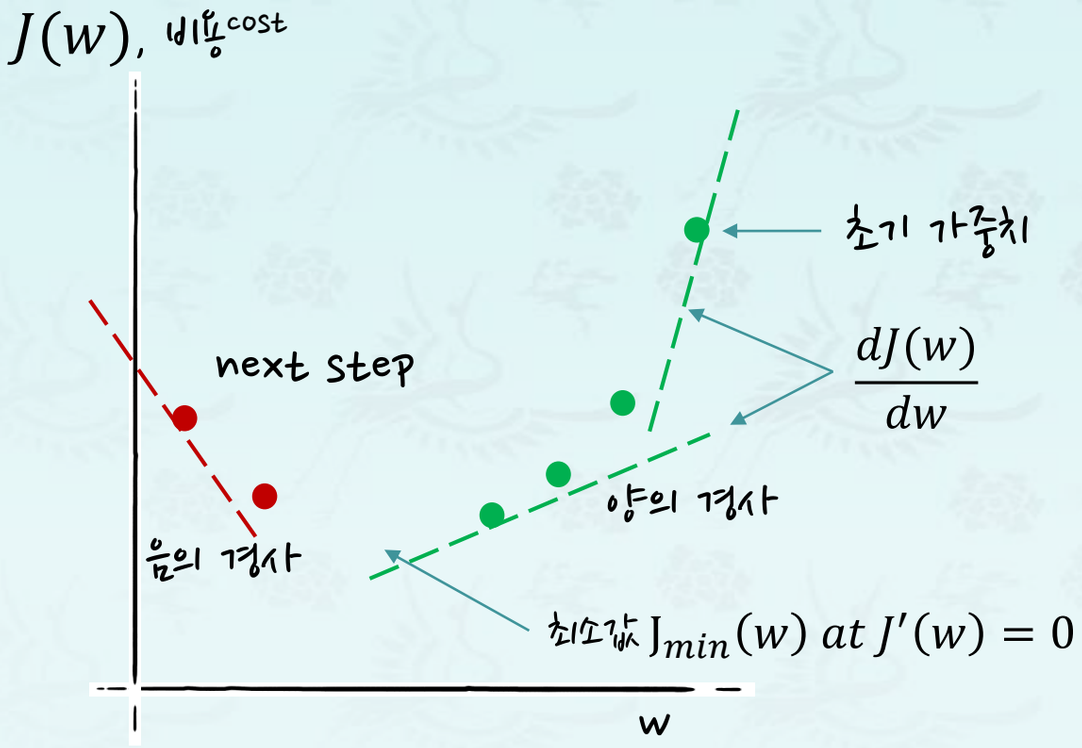
$f'(x_1) < 0$



$f'(x_2) = 0$



$f'(x_3) > 0$


$J(w)$, 비용cost
초기 가중치
$\frac{dJ(w)}{dw}$
next step
음의 경사
양의 경사
최소값 $J_{min}(w)$ at $J'(w) = 0$
w