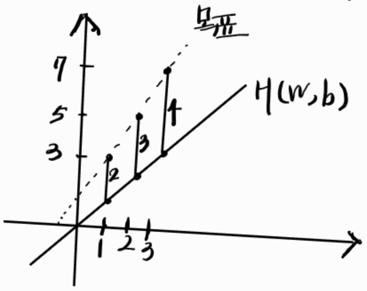
지도학습 $X : [1 \quad 2 \quad 3]$ Q, X=4일 때, Y는?
$Y : [3 \quad 5 \quad 7]$ $f(x) = 2x+1$, $f(4) = 9$ 예측 ⇒ Only 선형 회귀에 대한 내용임

가설함수 $H(w, b) = Wx + b$

목표 $W \Rightarrow 2$, $b \Rightarrow 1$

가설 초기화 $W = 1$, $b = 0$ 얼마나 잘못 되었는가? ⇒ Cost(w,b) 함수

비용함수 $Cost(w, b) = \frac{1}{m} \times \sum_{i=1}^{m} (\underbrace{Wx_i + b}_{가설} - \underbrace{y_i}_{실제})^2$ 최소제곱법



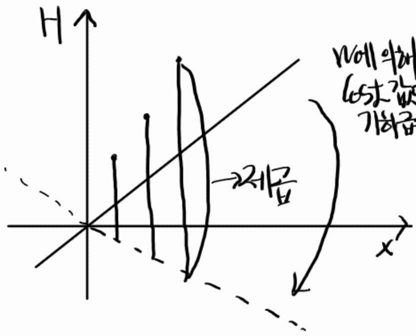
※ 제곱 vs 절댓값 이유

1. 더 강한 Penalty ⇒ 빠른 학습
2. 절댓값은 내부적으로 조건문 사용 ⇒ 느려짐

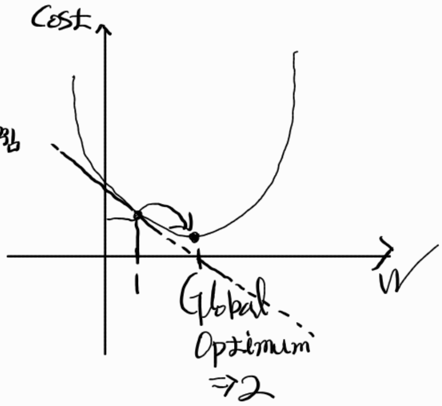
$Cost = \frac{2^2 + 3^2 + 4^2}{3} = \frac{29}{3}$ (초기 상태 비용)

우리가 해결할 문제 $\min_{W, b} \sum_{i=0}^{m} (Wx_i + b - y_i)^2$

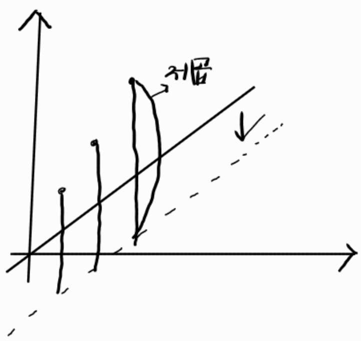
i) b가 고정

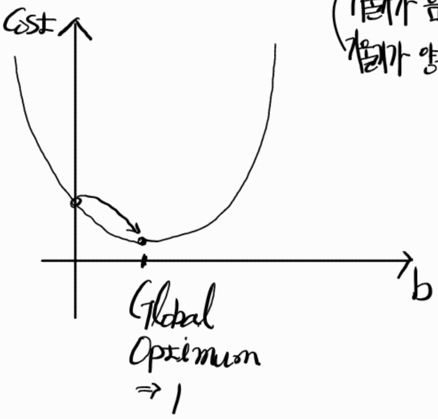


Convex graph



ii) W가 고정





기울기가 음수라면 오른쪽으로 (W/b 증가)
기울기가 양수라면 왼쪽으로 (W/b 감소)

∴ Cost 함수의 W, b의 각 기울기를 구하여 업데이트 최적의 W, b 구함

경사하강법 : Cost를 줄이기 위해 반복적으로 기울기를 계산하여 변수의 값을 변경해 나가는 과정.
=기울기 = ↓아래로

Cost(W, b)를 각각 W와 b에 대해 미분 (편미분)

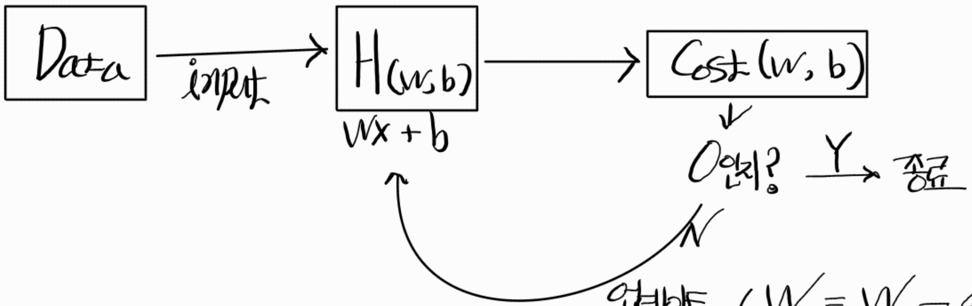
컴퓨터가 푸는 방법

by 경사하강법

$$\frac{1}{m} \times \sum_{i=1}^{m} (Wx_i + b - y_i)^2$$

$$\left(\begin{array}{l} W \text{ 기울기} = \frac{\partial Cost(W, b)}{\partial W} = \frac{1}{m} \cdot \sum_{i=1}^{m} 2(Wx_i + b - y_i) \cdot x_i \\ b \text{ 기울기} = \frac{\partial Cost(W, b)}{\partial b} = \frac{1}{m} \cdot \sum_{i=1}^{m} 2(Wx_i + b - y_i) \cdot 1 \end{array} \right.$$

[머신러닝 (경사하강법)]



업데이트

$$\left(\begin{array}{l} W = W - \alpha \cdot \frac{\partial Cost(W,b)}{\partial W} \Rightarrow 2 \\ b = b - \alpha \cdot \frac{\partial Cost(W,b)}{\partial b} \Rightarrow 1 \end{array} \right.$$

1000↑

epoch

1000번 이상 많이

⇒ 적절한 W, b 찾아 가며 학습

사람이 푸는 방법

by 연립, 수학적 수식

$$W\text{의 기울기} = \frac{2}{m}\cdot\Sigma(WX_i+b-y_i)\cdot X_i \Rightarrow 0$$

$$b\text{의 기울기} = \frac{2}{m}\cdot\Sigma(WX_i+b-y_i) \Rightarrow 0$$

$$\left.\begin{aligned}\Sigma X_iY_i &= \Sigma X_i^2 W + \Sigma X_i\cdot b\\ \Sigma y_i &= \Sigma X_i\cdot W + \Sigma b\end{aligned}\right\} \rightarrow \text{연립방정식}$$

by 행렬식

$$\begin{pmatrix}\Sigma X_i^2 & \Sigma X_i\\ \Sigma X_i & m\end{pmatrix}\begin{pmatrix}W\\ b\end{pmatrix} = \begin{pmatrix}\Sigma X_iY_i\\ \Sigma Y_i\end{pmatrix}$$

$AB = C$

$B = A^{-1}C$ → 빠르게 역행렬 계산할 수 있는 방법 "크레머 룰"

※ 분산 공식, $E(x)$ 기댓값 (in C.S) $= \frac{1}{m}\sum_{i=1}^{m} X_i$ (평균값)

$$W = \frac{m\cdot\Sigma x_iy_i - \Sigma x_i\Sigma y_i}{m\cdot\Sigma x_i^2 - \Sigma x_i\Sigma x_i} \quad \begin{matrix}\div m^2\\ \div m^2\end{matrix}$$

$$= \frac{E(XY)-E(X)E(Y)}{E(X^2)-E(X)E(Y)} = \frac{Cov(X,Y)}{Var(X)}\ \begin{matrix}\text{공분산}\\ \text{분산}\end{matrix}$$

$$\Sigma b = \Sigma y_i - \Sigma X_i\cdot W$$

$$m\cdot b = \Sigma y_i - \Sigma X_i\cdot W$$

$$b = E(Y) - E(X)\cdot W$$

$X: [1\ 2\ 3]$ $E(X) = \frac{1+2+3}{3} = 2$

$Y\ [3\ 5\ 7]$ $E(Y) = \frac{3+5+7}{3} = 5$

$$Cov(X,Y) = \frac{1}{n}\cdot\sum_{i=1}^{n}(X_i - avg_x)\cdot(Y_i - avg_Y)$$

$$= \frac{1}{3}\{(1-2)(3-5)+(2-2)(5-5)+(3-2)(7-5)\}$$

$$= \frac{1}{3}\cdot(2+0+2) = \frac{4}{3}$$

$$Var(x) = \frac{1}{n}\sum_{i=1}^{n} X_i^2 - avg_x^2$$

$$= \frac{14}{3} - 4 = \frac{2}{3}$$

$$\therefore W = 2,\ b = 1$$

현실세계의 머신러닝 사례에서는

cost 함수가  형태

따라서 경사하강법을 사용