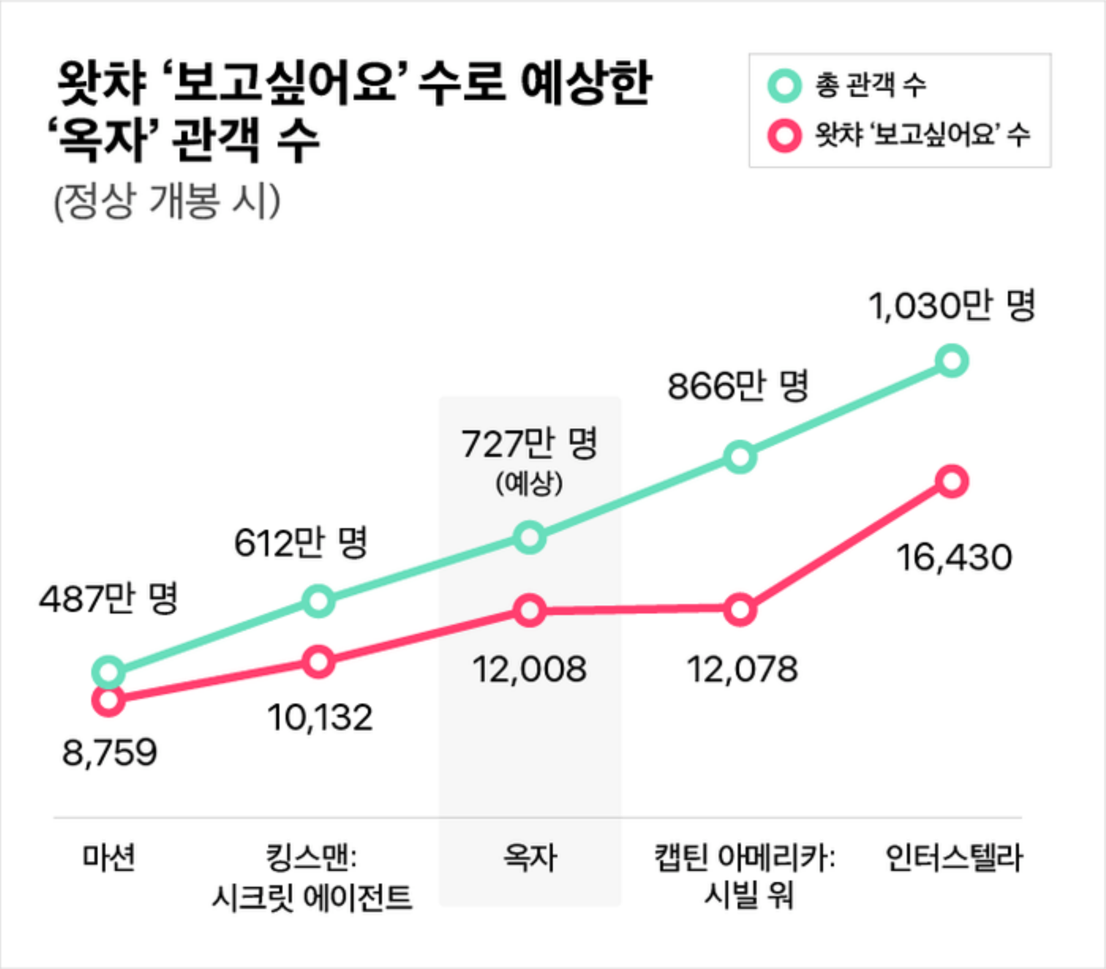
Linear Regression with GD

Linear Regression

Director of TEAMLAB
Sungchul Choi



$$\mathbf{y} = \begin{bmatrix} 487 \\ 612 \\ 866 \\ 1030 \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} 1 & 8,759 \\ 1 & 10,132 \\ 1 & 12,078 \\ 1 & 16,430 \end{bmatrix}$$

$$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

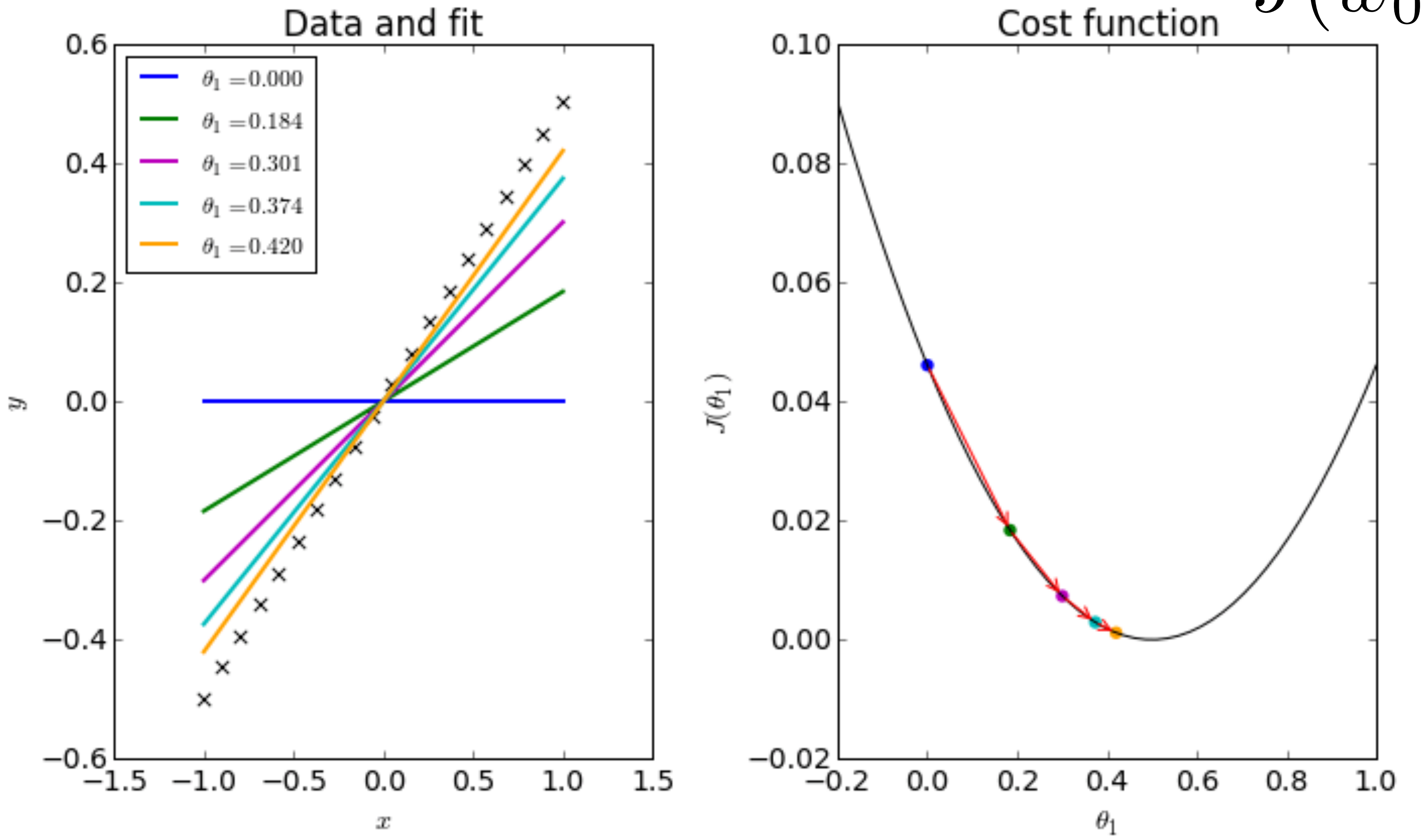
$$J(w_0, w_1) = \frac{1}{2m} \sum_{i=1}^{m} (w_1 x^{(i)} + w_0 - y^{(i)})^2$$

Minimize $J(w_0, w_1)$

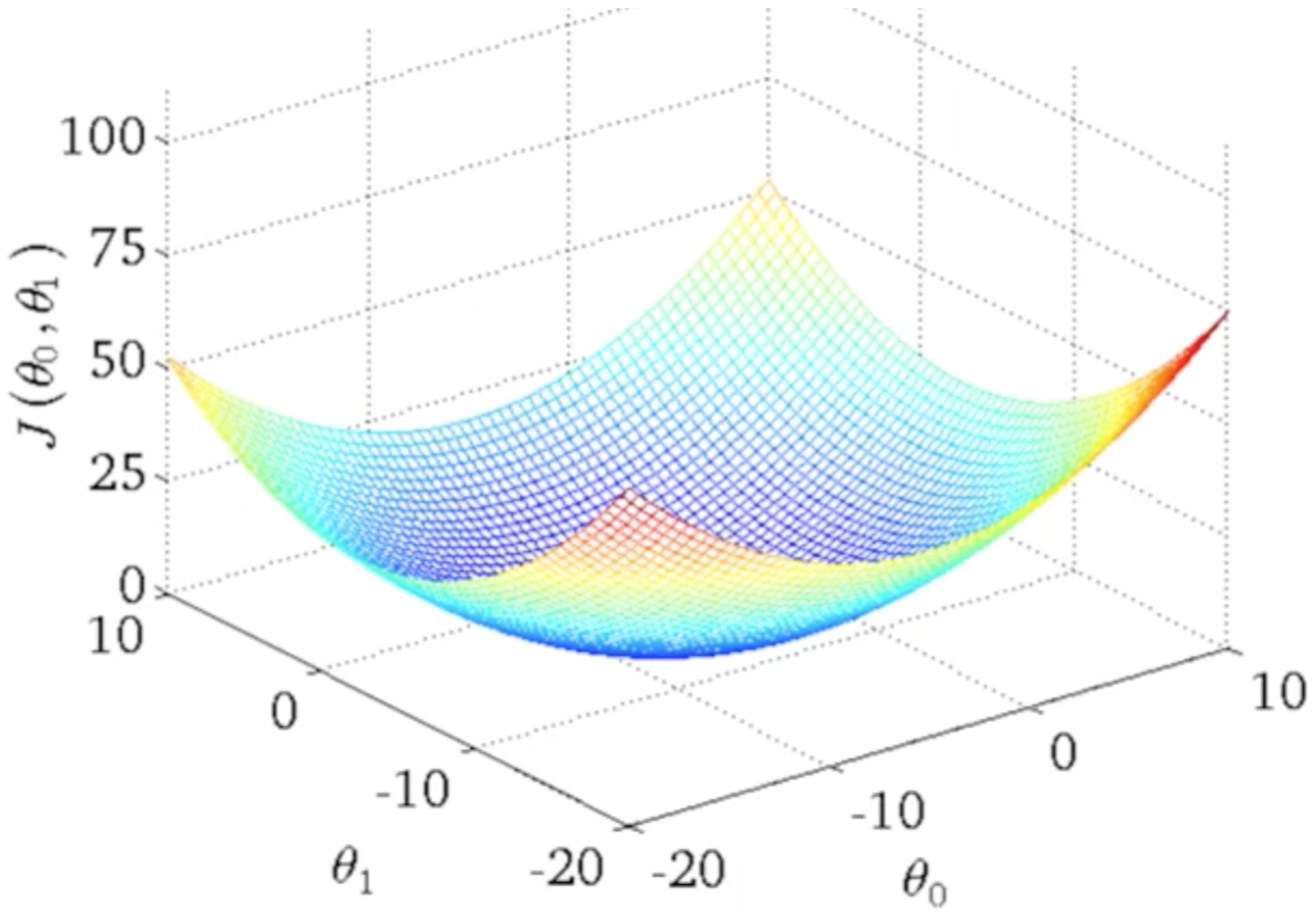
$$J(w_0, w_1) = \frac{1}{2m} \sum_{i=1}^{m} (w_1 x^{(i)} + w_0 - y^{(i)})^2$$

Minimize $J(w_0, w_1)$

$$J(w_0, w_1)$$



https://scipython.com/blog/visualizing-the-gradient-descent-method/



100
75
50
25
0
$J(\theta_0, \theta_1)$
10
0
-10
-20
θ_1
-20
-10
0
10
θ_0

결국 parameter의 업데이트

Linear regression with GD

- 임의의 θ_0, θ_1 값으로 초기화
- Cost function $J(\theta_0, \theta_1)$ 이 최소화 될 때까지 학습
- 더 이상 cost function이 줄어들지 않거나 학습 횟수를 초과할 때 종료

Linear regression with GD

$$x_{new} = x_{old} - \alpha \times (2x_{old})$$

$$\frac{\partial J}{\partial w_0} = \frac{1}{m}\sum_{i=1}^{m}(w_1 x^{(i)} + w_0 - y^{(i)})$$

$$\frac{\partial J}{\partial w_1} = \frac{1}{m}\sum_{i=1}^{m}(w_1 x^{(i)} + w_0 - y^{(i)})x^{(i)}$$

$$\text{loop until convergence}\{$$

$$\text{do } \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

$$\}$$

simultaneously

Linear regression with GD

- Learning rate, Iteration 횟수 등 Parameter 지정
- Feature가 많으면 Normal equation에 비해 상대적으로 빠름
- 최적값에 수렴하지 않을 수도 있음

TEAMLAB

Human knowledge belongs to the world.