

인공지능개론

기계학습

Gradient descent

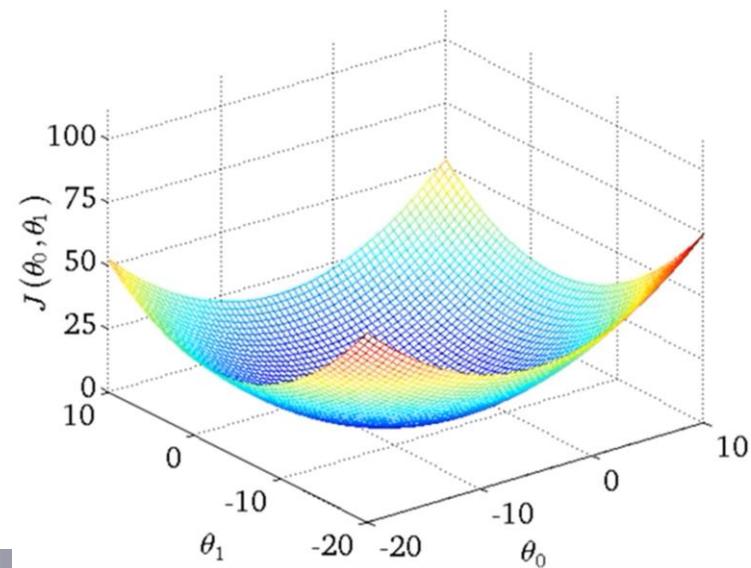
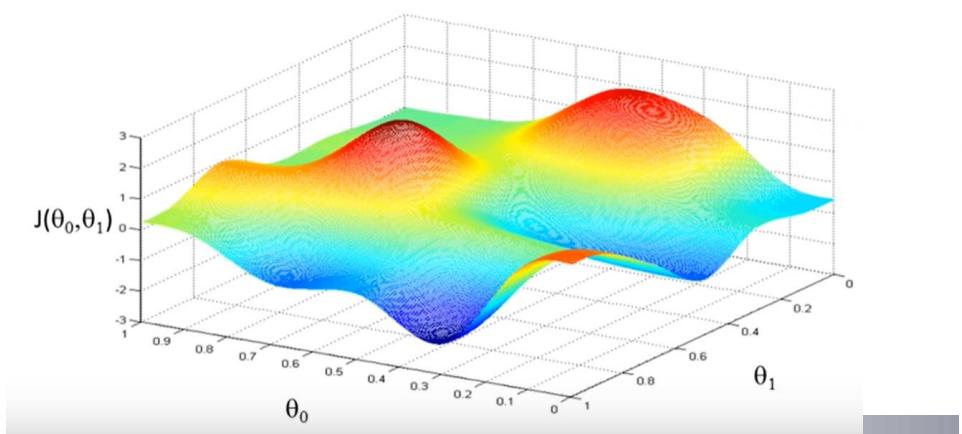
For the cost function $J(\beta_0, \beta_1, \dots, \beta_n)$ \rightarrow ex) $J(\beta_0, \beta_1) = \frac{1}{2m} \sum_{i=1}^m (h(x^i) - y^i)^2$

We want minimize $J(\beta_0, \beta_1, \dots, \beta_n)$ $\rightarrow \min_{\beta_0, \beta_1} J(\beta_0, \beta_1)$

Then, how to minimize?

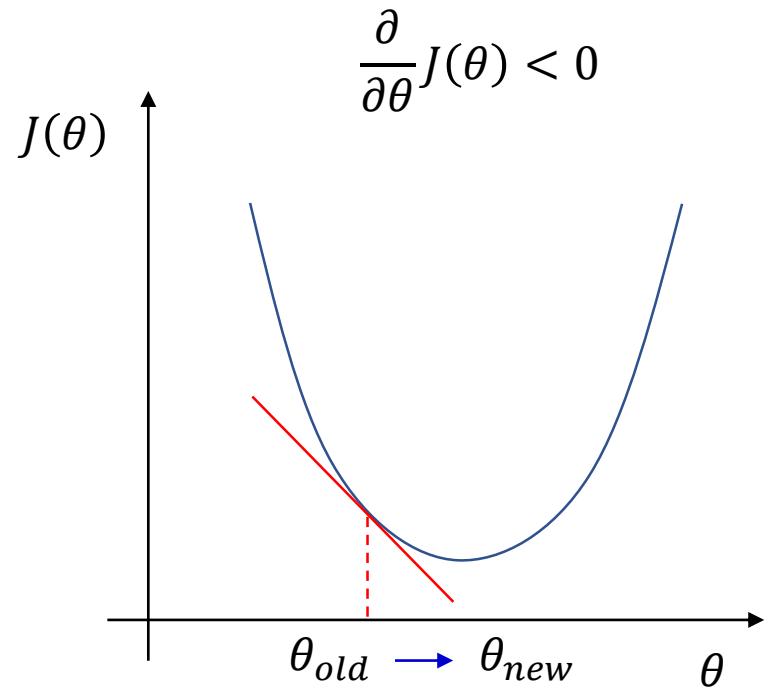
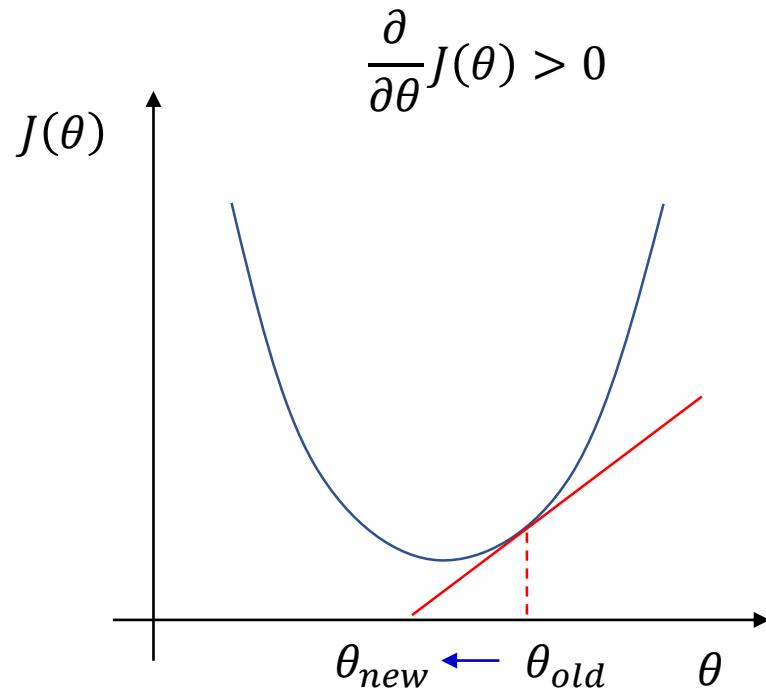
Start with initial β_0, β_1 ex) $(\beta_0, \beta_1) = (0,0)$

Change β_0, β_1 to reduce $J(\beta_0, \beta_1, \dots, \beta_n)$ until it reaches at a minimum



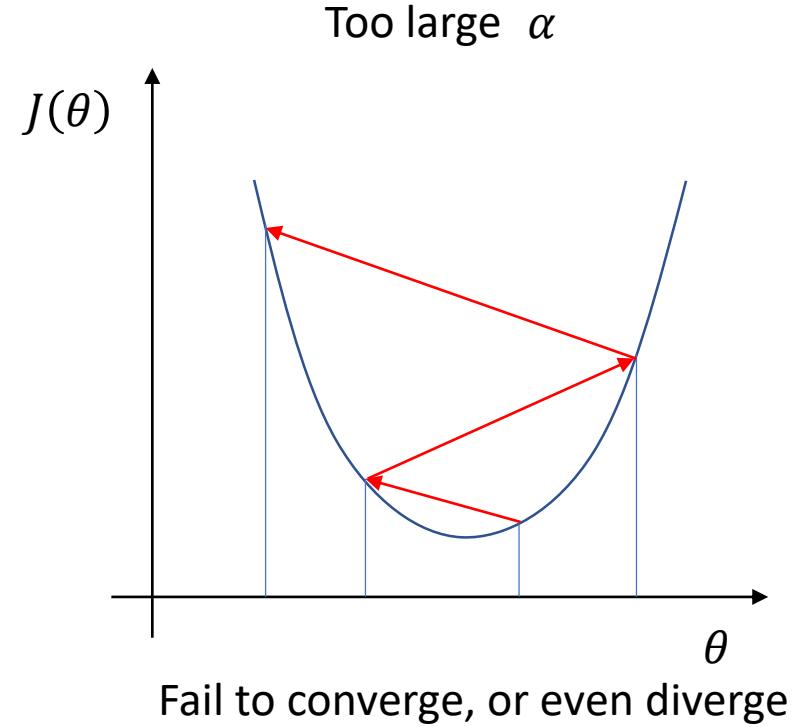
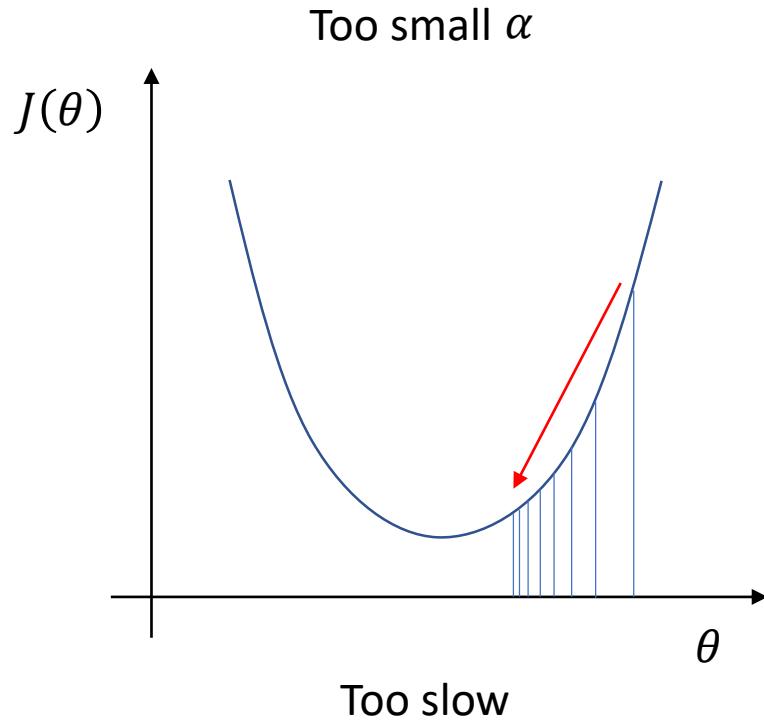
Gradient descent: convergence

$$\theta_{new} = \theta_{old} - \alpha \frac{\partial}{\partial \theta} J(\theta) \Big|_{\theta=\theta_{old}}$$



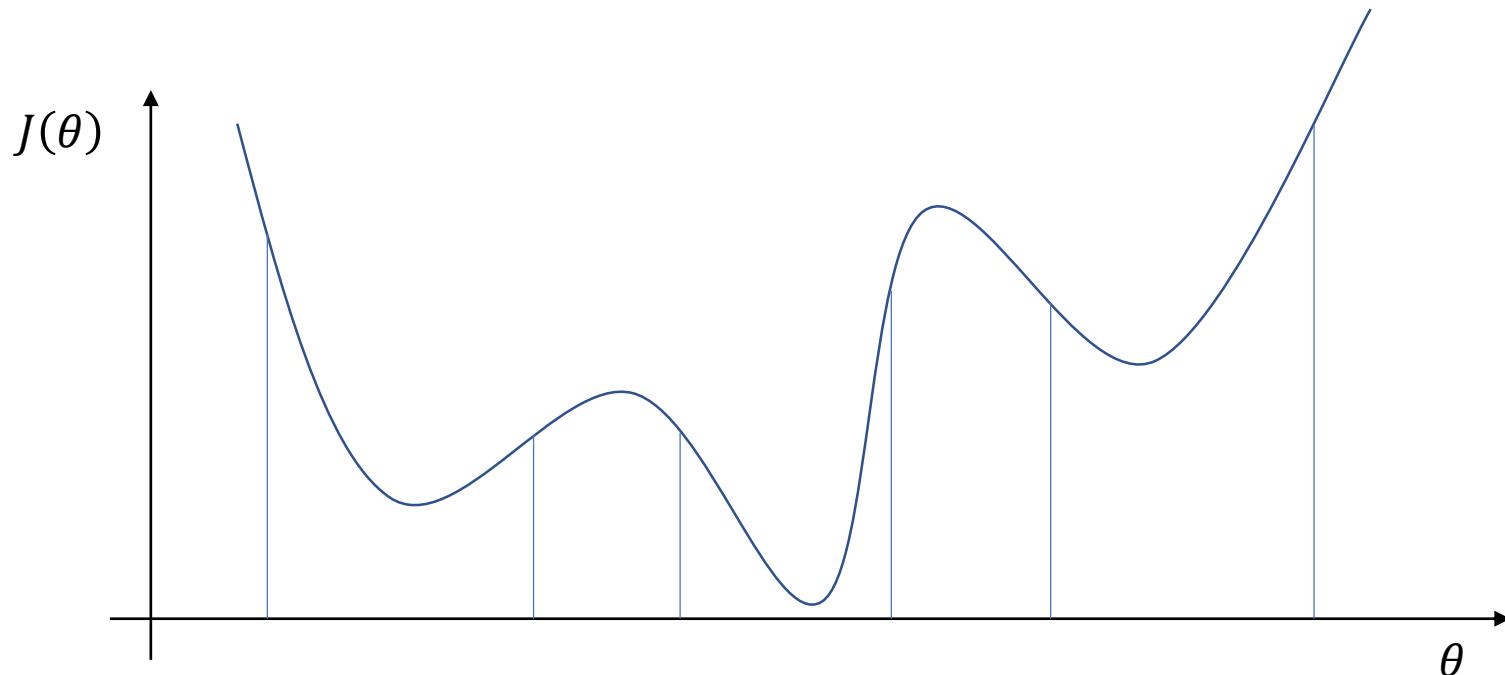
Gradient descent: learning rate

$$\theta_{new} = \theta_{old} - \alpha \frac{\partial}{\partial \theta} J(\theta) \Big|_{\theta=\theta_{old}}$$
$$\alpha > 0$$



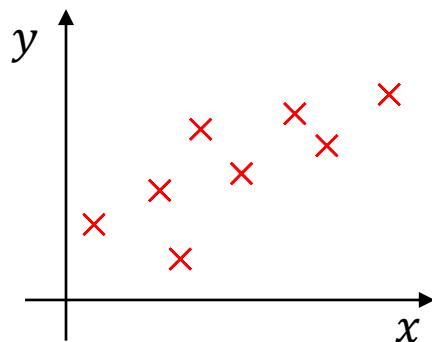
Gradient descent: local optima

$$\theta_{new} = \theta_{old} - \alpha \frac{\partial}{\partial \theta} J(\theta) \Big|_{\theta=\theta_{old}} \quad \alpha > 0$$



Gradient descent: procedure

$$\theta_{new} = \theta_{old} - \alpha \frac{\partial}{\partial \theta} J(\theta) \Big|_{\theta=\theta_{old}}$$



① Assume $y = h(x) = \theta x$

② Define $J(\theta)$

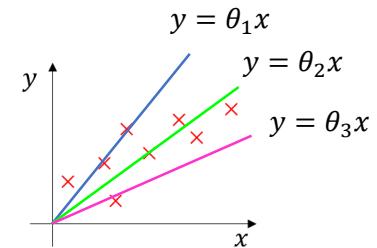
$$J(\theta) = \frac{1}{2m} \sum_{i=1}^m (h(x_i) - y_i)^2 = \frac{1}{2m} \sum_{i=1}^m (\theta x_i - y_i)^2$$

③ Choose $\theta_{initial}$

$$\theta_{initial} = 0$$

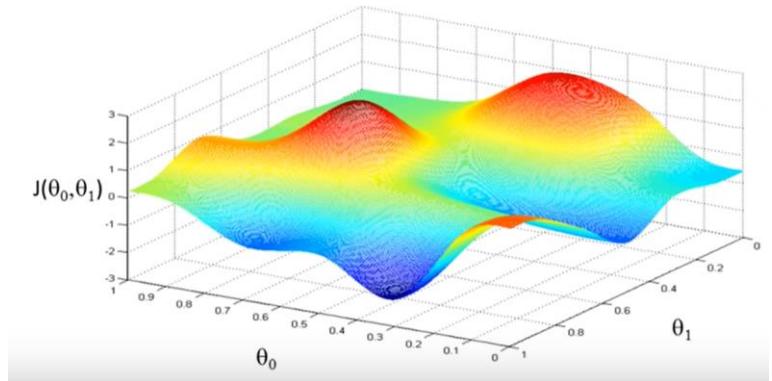
④ Update θ

$$\theta_{new} = \theta_{old} - \alpha \frac{\partial}{\partial \theta} J(\theta) \Big|_{\theta=\theta_{old}} \quad \text{where} \quad \frac{\partial}{\partial \theta} J(\theta) = \frac{1}{m} \sum_{i=1}^m \theta x_i^2 - \frac{1}{m} \sum_{i=1}^m x_i y_i$$



Gradient descent: questions

$$\theta_{i,new} = \theta_{i,old} - \alpha \frac{\partial}{\partial \theta_i} J(\theta_i) \Big|_{\theta_i=\theta_{i,old}}$$



Q1. What should human (engineer) do? $y = \theta_1 x?$ $y = \theta_1 x + \theta_2 x^2 + \theta_3 x^3?$ $y = \theta_2 e^{\theta_1 x}?$

Which one is a design parameter?

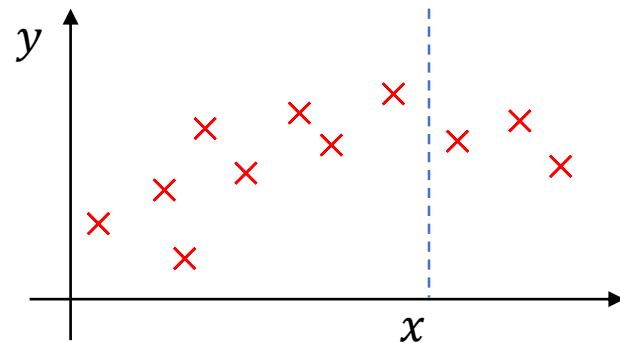
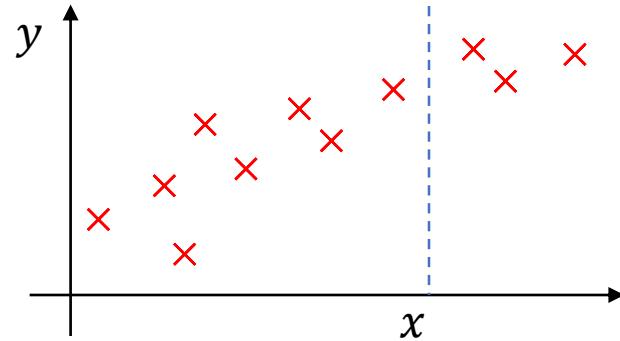
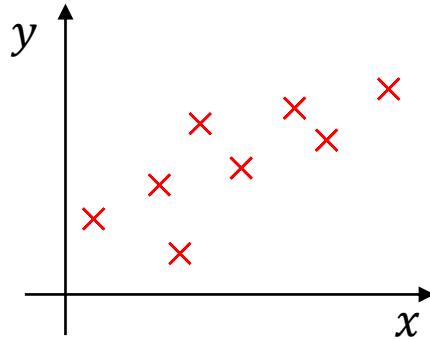
Q2. Fixed learning rate?

Q3. How to determine the cost function? $J(\theta) = \frac{1}{2m} \sum_{i=1}^m (h(x_i) - y_i)^3 ??$

Gradient descent: problem statement

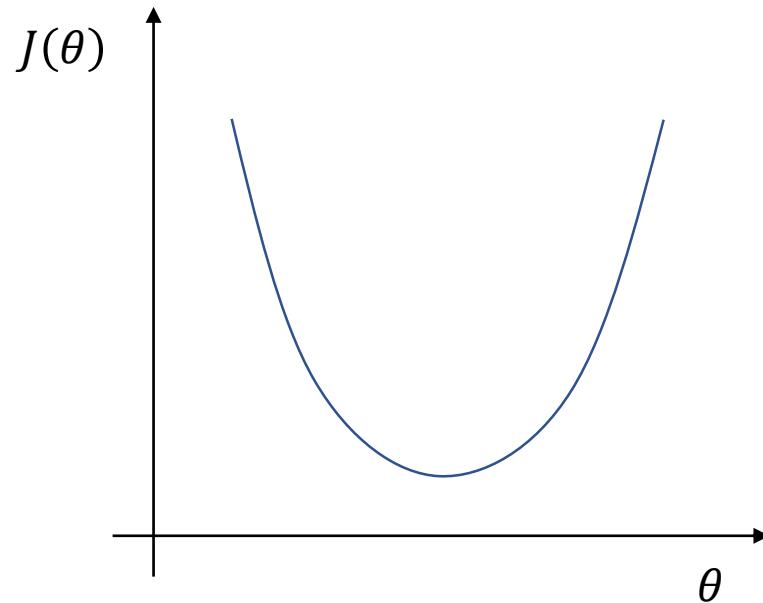
Q1. What should human (engineer) do? Which one is a design parameter?

$$y = \theta_1 x? \quad y = \theta_1 x + \theta_2 x^2 + \theta_3 x^3? \quad y = \theta_2 e^{\theta_1 x}?$$



Gradient descent: convergence

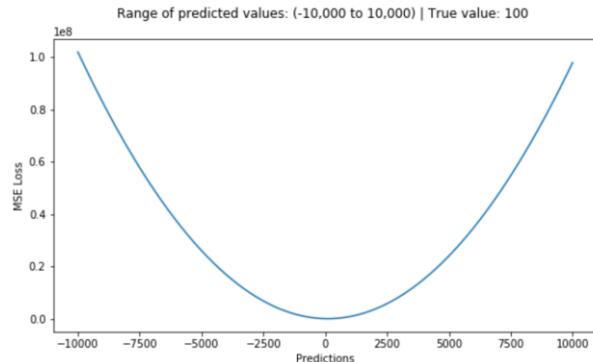
Q2. Fixed learning rate?



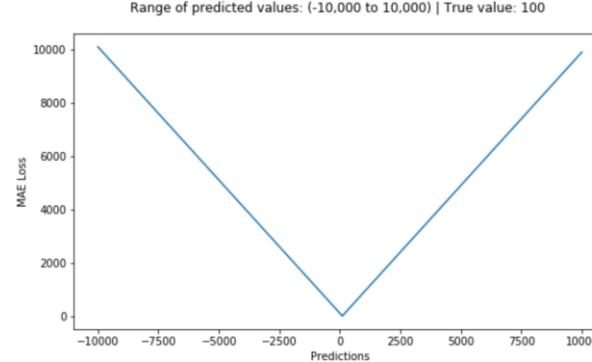
Gradient descent: cost function

Q3. How to determine the cost function?

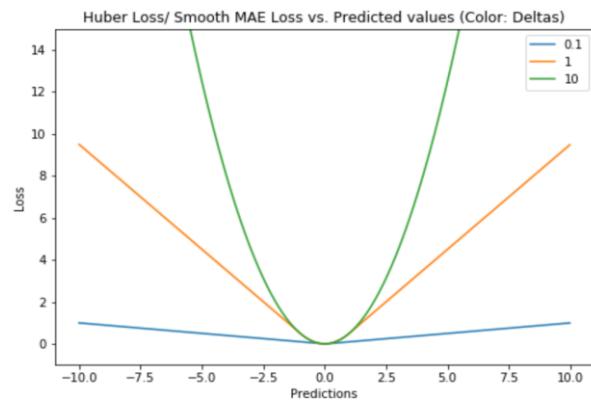
$$J(\theta) = \frac{1}{2m} \sum_{i=1}^m (h(x_i) - y_i)^2 ??$$



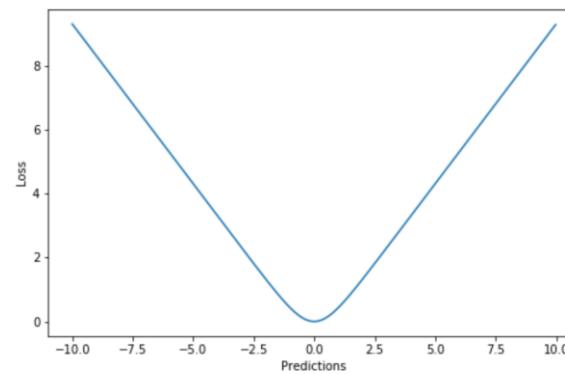
Plot of MSE Loss (Y-axis) vs. Predictions (X-axis)



Plot of MAE Loss (Y-axis) vs. Predictions (X-axis)



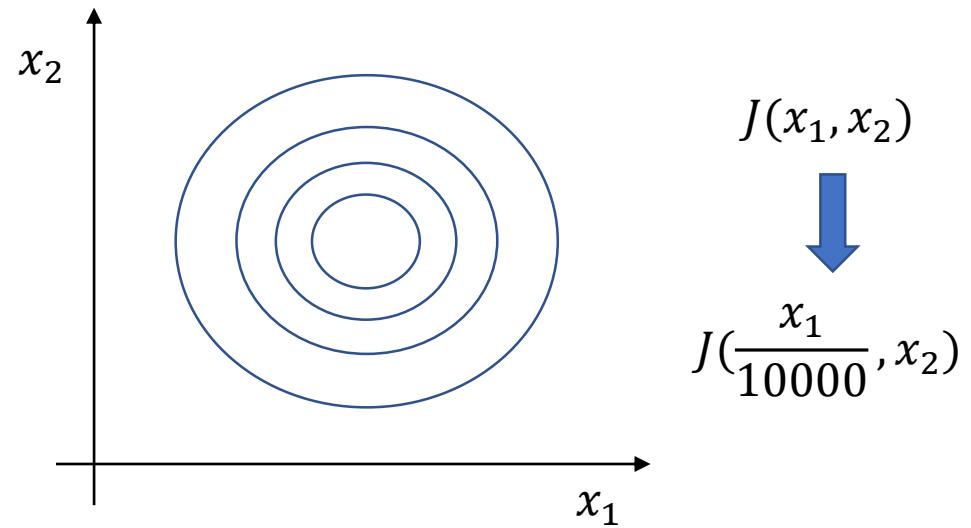
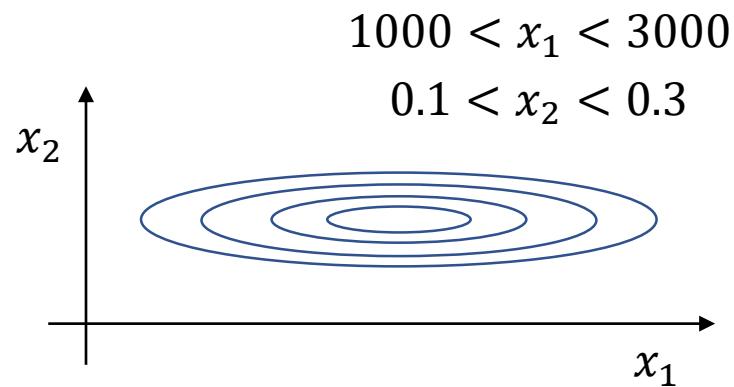
Plot of Hoss Loss (Y-axis) vs. Predictions (X-axis). True value = 0



Plot of Log-cosh Loss (Y-axis) vs. Predictions (X-axis). True value = 0

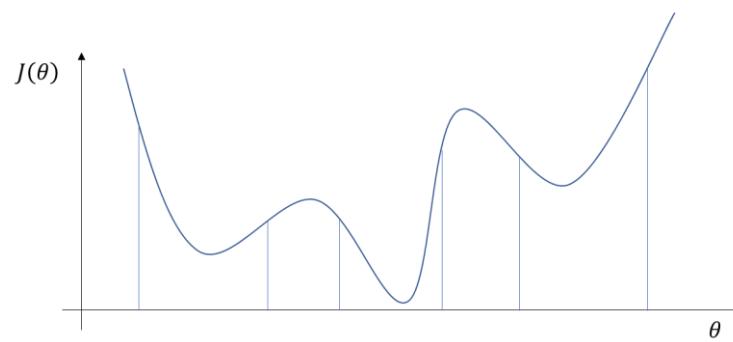
Gradient descent: practical issues

- Feature scaling



- Learning rate

$0.001 \rightarrow 0.01 \rightarrow 0.1 \rightarrow 1 \rightarrow 10$



Gradient descent: normal equation

x_0	x_1	x_2	x_3	x_4	y
1	2104	5	1	45	460
1	1416	3	2	40	232
1	1534	3	2	30	315
1	852	2	1	36	178

- Gradient descent

$$J(\theta_j) = \frac{1}{2m} \sum_{i=1}^m (h(x) - y)^2$$

$$\frac{\partial}{\partial \theta_j} J(\theta_j) = 0 \quad (\text{for every } j)$$

- Normal equation

$$y = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4$$

$$X = \begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix} \quad y = \begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \end{bmatrix}$$

$$\theta = (X^T X)^{-1} X^T y$$

- Need to choose α
- Many iterations
- Works well even large features

- No need to choose α
- No iterations & feature scaling
- Large computational cost
- Invertible