

# **6-5. Multivariate Gradient Descent**

# Multivariate Gradient Descent

## 앞선 챕터들의 간략한 리뷰

- **Gradient Descent (경사하강법)**

**Loss을 최소화하기 위한 방법이다! Loss의 Gradient 반대방향으로 weight을 update.**

$$w_{i+1} = w_i - \lambda \cdot \frac{\partial L}{\partial w_i}$$

- **Auto Differentiation (Reverse Differentiation)**

미분의 연쇄법칙 특징을 활용해서 Computationally Efficient하게 각 Layer를 구성하는

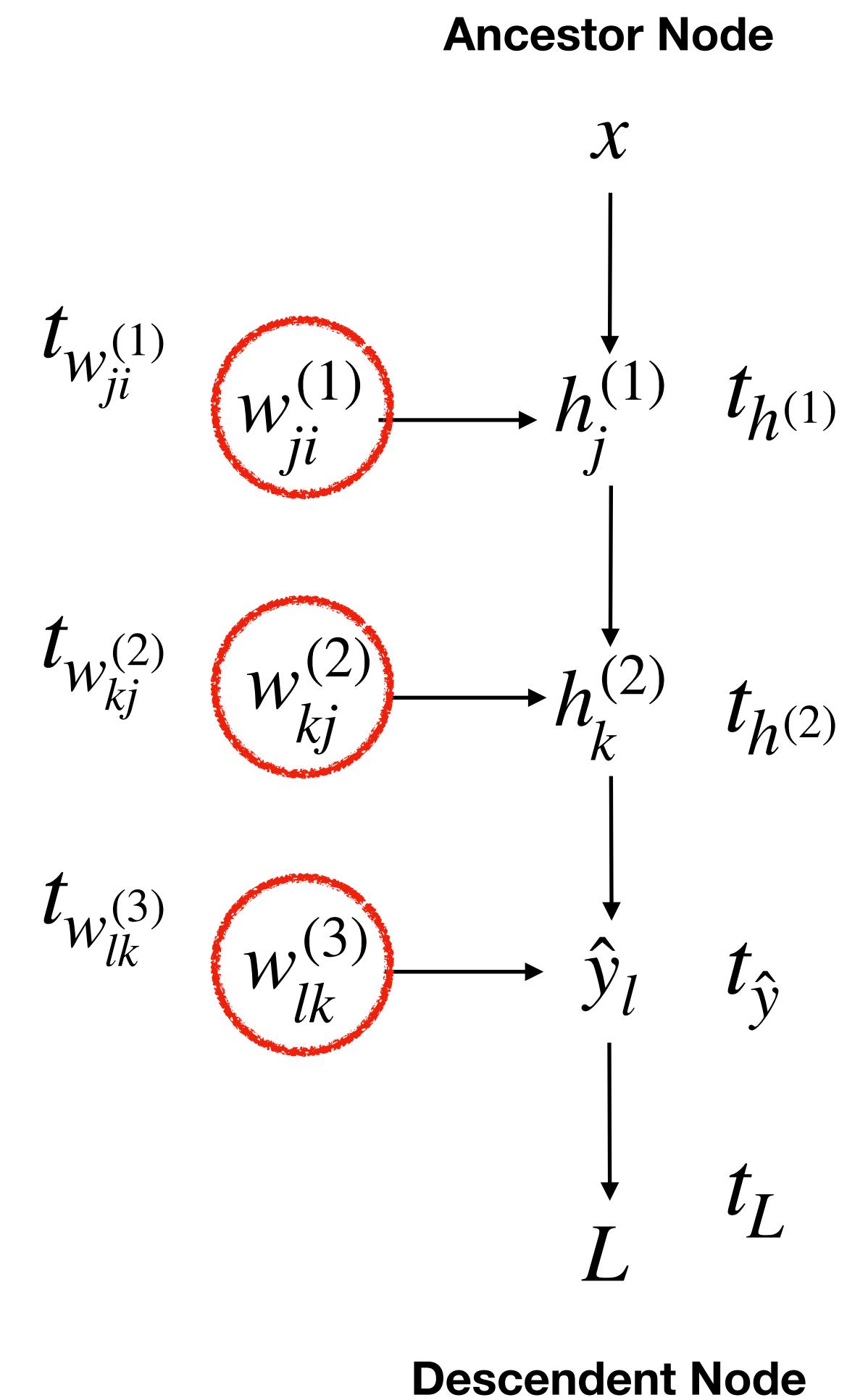
Weight의 Loss Gradient  $\frac{\partial L}{\partial w}$  을 구하는 것.

# Multivariate Gradient Descent

## 앞선 챕터들의 간략한 리뷰

### Auto Differentiation

각 Layer을 구성하는 Weight의 Loss Gradient  $\frac{\partial L}{\partial w}$  을 구하는 것.

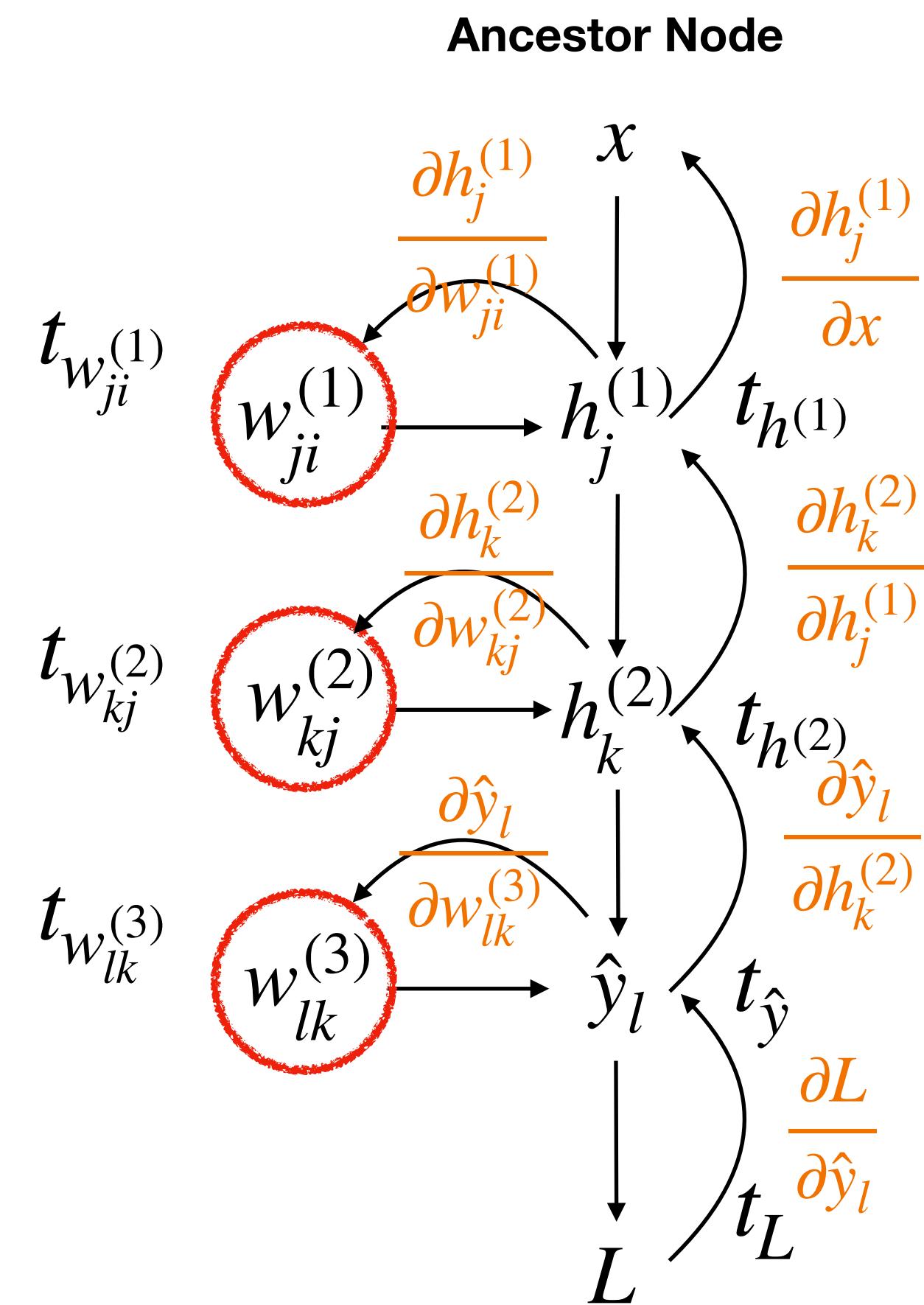


# Multivariate Gradient Descent

앞선 챕터들의 간략한 리뷰

## Auto Differentiation

각 Layer을 구성하는 Weight의 Loss Gradient  $\frac{\partial L}{\partial w}$  을 구하는 것.



# Multivariate Gradient Descent

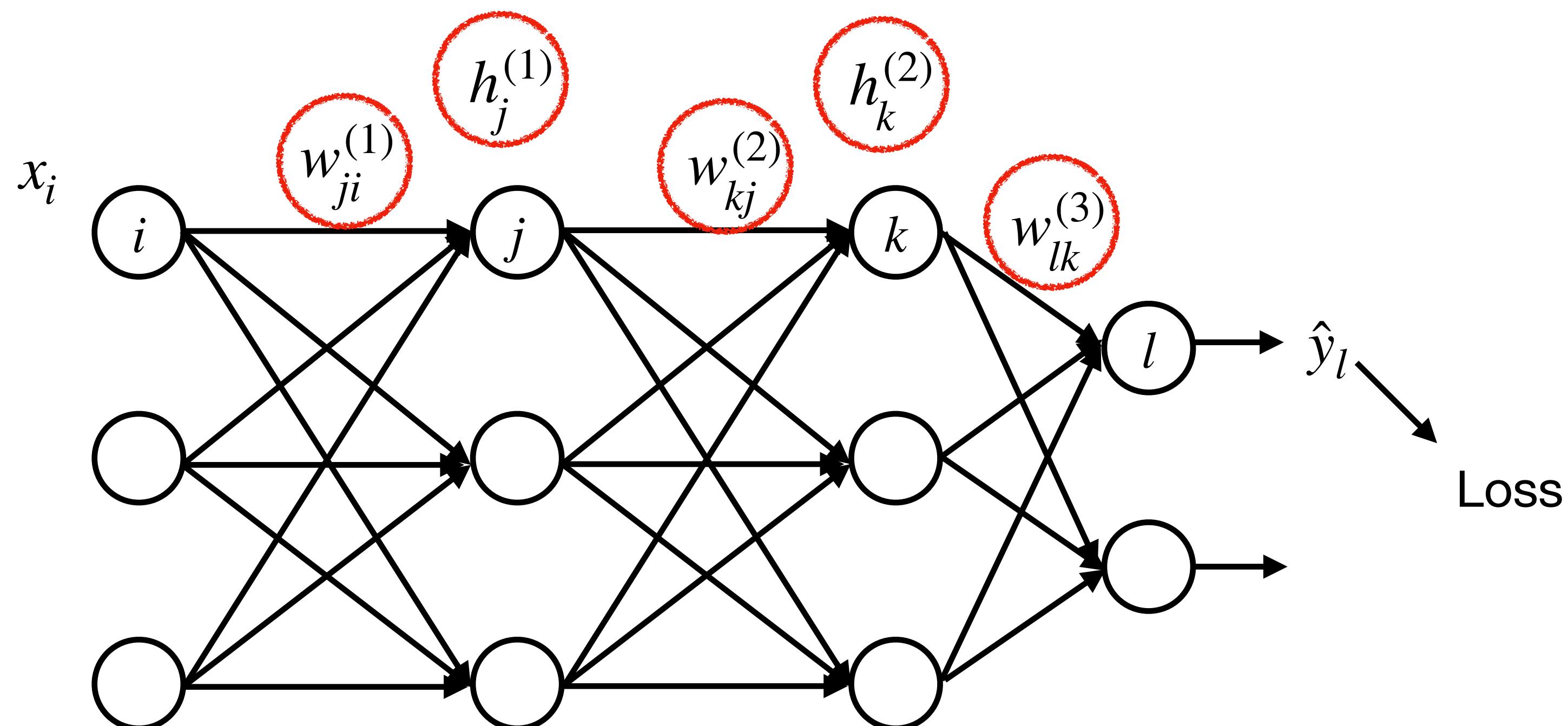
## 앞선 챕터들의 간략한 리뷰

$h_j^{(1)}, h_k^{(2)}, \hat{y}_l :$

- $j, k, l$ 로 인덱스되는 Vector이다.
- $\mathbf{h}^{(1)} \in \mathbb{R}^J, \mathbf{h}^{(2)} \in \mathbb{R}^K, \hat{\mathbf{y}} \in \mathbb{R}^L$

$w_{ji}^{(1)}, w_{kj}^{(2)}, w_{lk}^{(3)} :$

- $i, j, k, l$ 로 인덱스되는 Matrix이다.
- $W^{(1)} \in \mathbb{R}^{J \times I}, W^{(2)} \in \mathbb{R}^{K \times J},$   
 $W^{(3)} \in \mathbb{R}^{L \times K}$



# Multivariate Gradient Descent

## 앞선 챕터들의 간략한 리뷰

$h_j^{(1)}, h_k^{(2)}, \hat{y}_l :$

- $j, k, l$  로 인덱스되는 Vector이다.
- $\mathbf{h}^{(1)} \in \mathbb{R}^J, \mathbf{h}^{(2)} \in \mathbb{R}^K, \hat{\mathbf{y}} \in \mathbb{R}^L$

$$t_{h^{(2)}} = t_{\hat{y}_l} \cdot \frac{\partial \hat{y}_l}{\partial h_k^{(2)}}$$

$w_{ji}^{(1)}, w_{kj}^{(2)}, w_{lk}^{(3)} :$

- $i, j, k, l$  로 인덱스되는 Matrix이다.
- $W^{(1)} \in \mathbb{R}^{J \times I}, W^{(2)} \in \mathbb{R}^{K \times J}, W^{(3)} \in \mathbb{R}^{L \times K}$

$$t_{w_{ji}^{(1)}} = \frac{\partial L}{\partial w_{ji}^{(1)}} = t_{h_j^{(1)}} \cdot \frac{\partial h_j^{(1)}}{\partial w_{ji}^{(1)}}$$

# Multivariate Gradient Descent

## 앞선 챕터들의 간략한 리뷰

$h_j^{(1)}, h_k^{(2)}, \hat{y}_l :$

- $j, k, l$ 로 인덱스되는 Vector이다.
- $\mathbf{h}^{(1)} \in \mathbb{R}^J, \mathbf{h}^{(2)} \in \mathbb{R}^K, \hat{\mathbf{y}} \in \mathbb{R}^L$

$$t_{h^{(2)}} = t_{\hat{y}_l} \cdot \frac{\partial \hat{y}_l}{\partial h_k^{(2)}}$$

$\hat{\mathbf{y}}$ 을  $\mathbf{h}^{(2)}$ 에 대해서 미분하려면...?

$w_{ji}^{(1)}, w_{kj}^{(2)}, w_{lk}^{(3)} :$

- $i, j, k, l$ 로 인덱스되는 Matrix이다.
- $W^{(1)} \in \mathbb{R}^{J \times I}, W^{(2)} \in \mathbb{R}^{K \times J}, W^{(3)} \in \mathbb{R}^{L \times K}$

$$t_{w_{ji}^{(1)}} = \frac{\partial L}{\partial w_{ji}^{(1)}} = t_{h_j^{(1)}} \cdot \frac{\partial h_j^{(1)}}{\partial w_{ji}^{(1)}}$$

$\mathbf{h}^{(1)}$ 을  $W$ 에 대해서 미분하려면...?

# Multivariate Differentiation

# Multivariate Gradient Descent

## Multivariate Differentiation

$w$  가 scalar일때:

$$w_{i+1} = w_i - \lambda \cdot \frac{dL}{dw_i}$$

# Multivariate Gradient Descent

## Multivariate Differentiation

$w \in \mathbb{R}$  가 scalar일 때:

$$w_{i+1} = w_i - \lambda \cdot \frac{dL}{dw_i}$$

$W \in \mathbb{R}^{M \times N}$  가 matrix일 때, index notation으로는:

$$w_{m,n}^{i+1} = w_{m,n}^i - \lambda \cdot \frac{\partial L}{\partial w_{m,n}^i}$$

Matrix notation으로는 어떻게 표현될까?

$W \in \mathbb{R}^{M \times N}$  가 matrix일 때, index notation으로는:

$$w_{m,n}^{i+1} = w_{m,n}^i - \lambda \cdot \frac{\partial L}{\partial w_{m,n}^i}$$

$W \in \mathbb{R}^{M \times N}$  가 matrix일 때, matrix notation으로는:

$$W^{i+1} = W^i - \lambda \cdot ?$$

?

M x N matrix

column = N개

$$\begin{matrix} & \left( \begin{array}{cccc} w_{11}^{i+1} & \dots & w_{1n}^{i+1} & \dots & w_{0N}^{i+1} \\ \dots & & \dots & & \dots \\ w_{m1}^{i+1} & \dots & w_{mn}^{i+1} & \dots & w_{mN}^{i+1} \\ \dots & & \dots & & \dots \\ w_{M1}^{i+1} & \dots & w_{Mn}^{i+1} & \dots & w_{MN}^{i+1} \end{array} \right) \\ \text{row = M개} \end{matrix} = W^{i+1}$$

$$\left( \begin{array}{cccc} w_{11}^i & \dots & w_{1n}^i & \dots & w_{0N}^i \\ \dots & & \dots & & \dots \\ w_{m1}^i & \dots & w_{mn}^i & \dots & w_{mN}^i \\ \dots & & \dots & & \dots \\ w_{M1}^i & \dots & w_{Mn}^i & \dots & w_{MN}^i \end{array} \right) = W^i$$

$$- \lambda \left( \begin{array}{cccc} \frac{\partial L}{\partial w_{11}^i} & \dots & \frac{\partial L}{\partial w_{1n}^i} & \dots & \frac{\partial L}{\partial w_{1N}^i} \\ \dots & & \dots & & \dots \\ \frac{\partial L}{\partial w_{m1}^i} & \dots & \frac{\partial L}{\partial w_{mn}^i} & \dots & \frac{\partial L}{\partial w_{mN}^i} \\ \dots & & \dots & & \dots \\ \frac{\partial L}{\partial w_{M1}^i} & \dots & \frac{\partial L}{\partial w_{Mn}^i} & \dots & \frac{\partial L}{\partial w_{MN}^i} \end{array} \right)$$

# Multivariate Gradient Descent

## Multivariate Differentiation

ACADENTIAL

Copyright©2023. Acadential. All rights reserved.

$$\nabla_{W^i} L = \begin{pmatrix} \frac{\partial L}{\partial w_{11}^i} & \dots & \frac{\partial L}{\partial w_{1n}^i} & \dots & \frac{\partial L}{\partial w_{1N}^i} \\ \frac{\partial L}{\partial w_{m1}^i} & \dots & \frac{\partial L}{\partial w_{mn}^i} & \dots & \frac{\partial L}{\partial w_{mN}^i} \\ \frac{\partial L}{\partial w_{M1}^i} & \dots & \frac{\partial L}{\partial w_{Mn}^i} & \dots & \frac{\partial L}{\partial w_{MN}^i} \end{pmatrix}$$

Partial Derivative의 행렬으로 구성된 것을  
**“Jacobian”**이라고 한다!

Notation은  $\nabla_W L$  으로 표현한다!

# Multivariate Gradient Descent

ACADENTIAL

## Multivariate Differentiation

Copyright©2023. Acadential. All rights reserved.

$W \in \mathbb{R}^{M \times N}$  가 matrix일 때, matrix notation으로는:

$$W^{i+1} = W^i - \lambda \cdot \nabla_{W^i} L$$

$$\nabla_{W^i} L$$

$M \times N$  matrix

column = N개

$$\text{row} = M\text{개} \quad \begin{pmatrix} w_{11}^{i+1} & \dots & w_{1n}^{i+1} & \dots & w_{0N}^{i+1} \\ w_{m1}^{i+1} & \dots & w_{mn}^{i+1} & \dots & w_{mN}^{i+1} \\ w_{M1}^{i+1} & \dots & w_{Mn}^{i+1} & \dots & w_{MN}^{i+1} \end{pmatrix} =$$

$W^{i+1}$

$$\begin{pmatrix} w_{11}^i & \dots & w_{1n}^i & \dots & w_{0N}^i \\ w_{m1}^i & \dots & w_{mn}^i & \dots & w_{mN}^i \\ w_{M1}^i & \dots & w_{Mn}^i & \dots & w_{MN}^i \end{pmatrix} \quad \boxed{\begin{pmatrix} w_{11}^i & \dots & w_{1n}^i & \dots & w_{0N}^i \\ w_{m1}^i & \dots & w_{mn}^i & \dots & w_{mN}^i \\ w_{M1}^i & \dots & w_{Mn}^i & \dots & w_{MN}^i \end{pmatrix}} - \lambda \cdot \boxed{\begin{pmatrix} \frac{\partial L}{\partial w_{11}^i} & \dots & \frac{\partial L}{\partial w_{1n}^i} & \dots & \frac{\partial L}{\partial w_{1N}^i} \\ \frac{\partial L}{\partial w_{m1}^i} & \dots & \frac{\partial L}{\partial w_{mn}^i} & \dots & \frac{\partial L}{\partial w_{mN}^i} \\ \frac{\partial L}{\partial w_{M1}^i} & \dots & \frac{\partial L}{\partial w_{Mn}^i} & \dots & \frac{\partial L}{\partial w_{MN}^i} \end{pmatrix}}$$

$W^i$

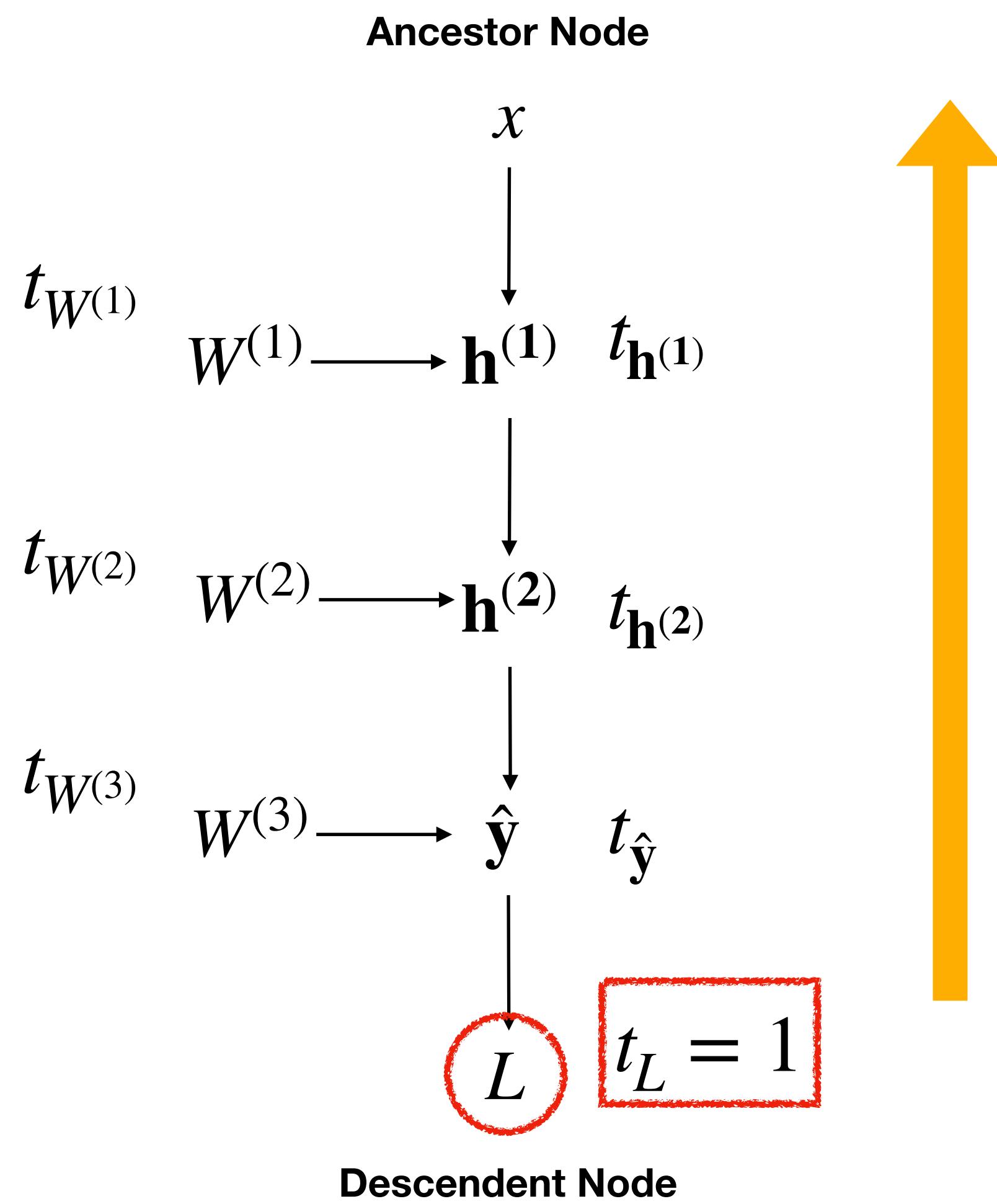
# Back to Auto Differentiation

# Multivariate Gradient Descent

ACADENTIAL

Copyright©2023. Acadential. All rights reserved.

## Computational Graph



$$W^{i+1} = W^i - \lambda \cdot \nabla_{W^i} L$$

1. Descendent Node로 부터 시작 ( $t_L = 1$ )

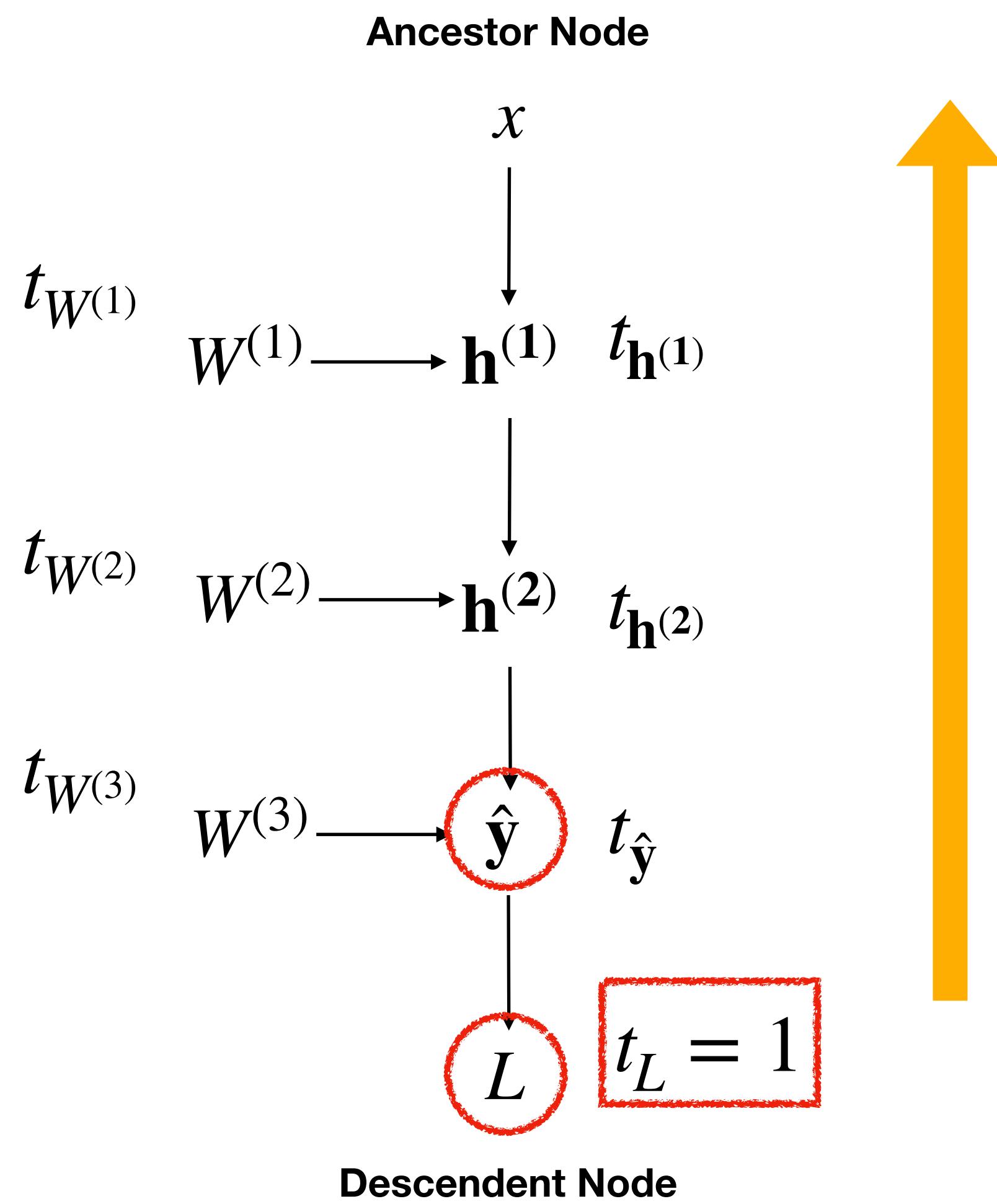
$$t_L = 1$$

# Multivariate Gradient Descent

ACADENTIAL

Copyright©2023. Acadential. All rights reserved.

## Computational Graph



$$W^{i+1} = W^i - \lambda \cdot \nabla_{W^i} L$$

1. Descendent Node로 부터 시작 ( $t_L = 1$ )
2.  $\hat{\mathbf{y}}$  노드의  $t_{\hat{\mathbf{y}}}$  계산:

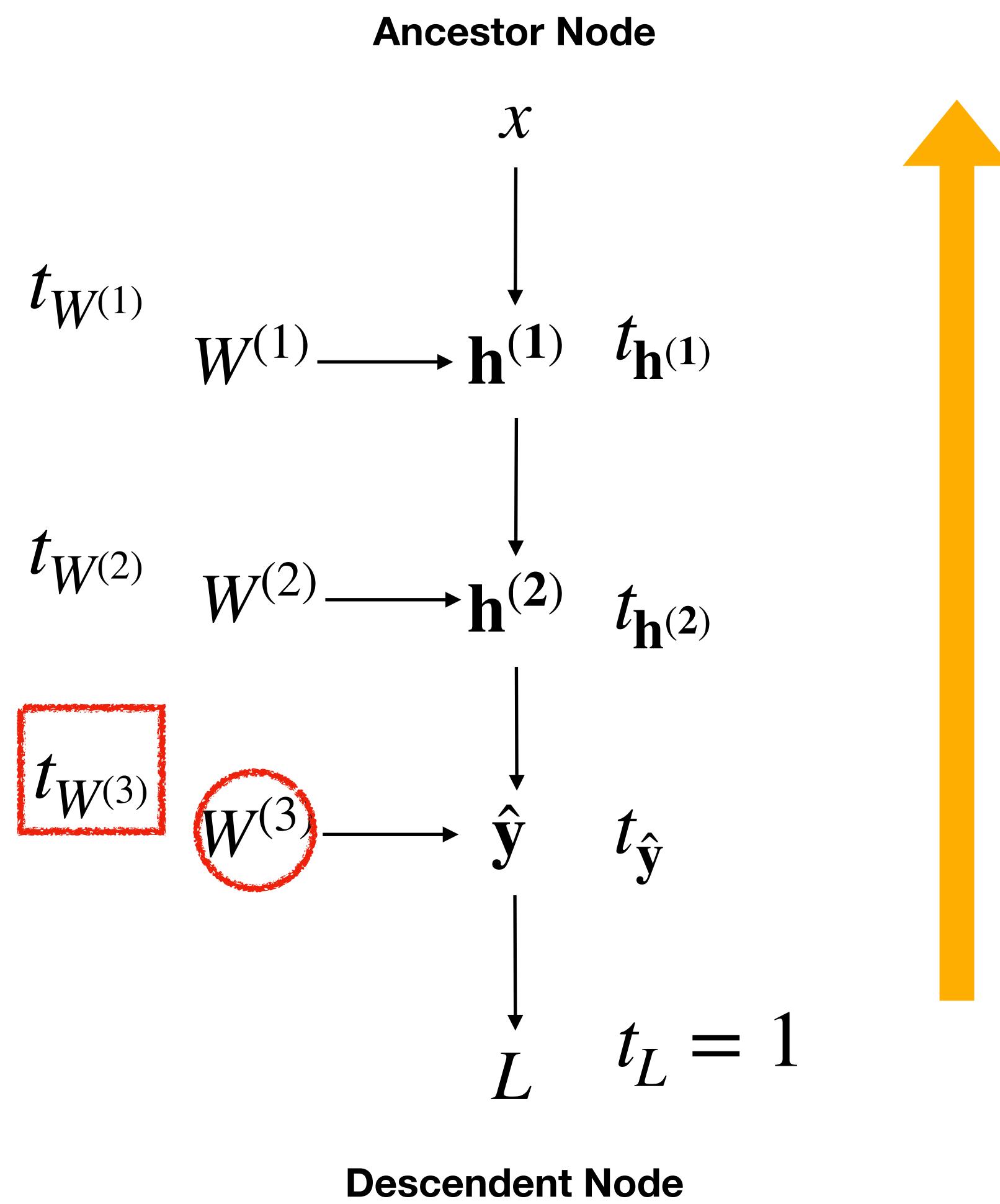
$$t_{\hat{\mathbf{y}}} = t_L \nabla_{\hat{\mathbf{y}}} L$$

# Multivariate Gradient Descent

ACADENTIAL

Copyright©2023. Acadential. All rights reserved.

## Computational Graph



$$W^{i+1} = W^i - \lambda \cdot \nabla_{W^i} L$$

1. Descendent Node로 부터 시작 ( $t_L = 1$ )
2.  $\hat{y}$  노드의  $t_{\hat{y}}$  계산
3.  $W^{(3)}$  노드의  $t_{W^{(3)}}$  계산:

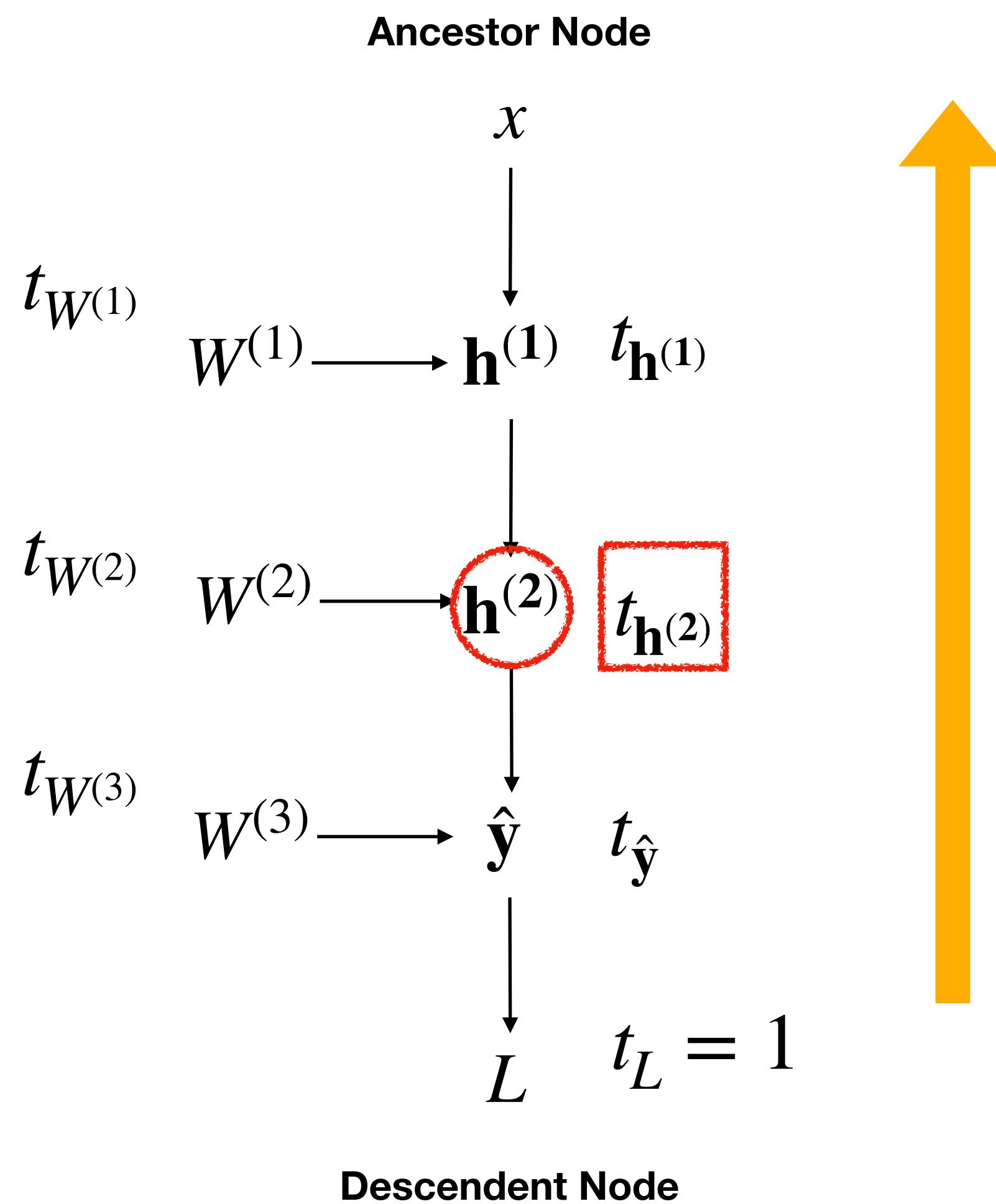
$$t_{W^{(3)}} = t_{\hat{y}} \nabla_{W^{(3)}} \hat{y}$$

# Multivariate Gradient Descent

ACADENTIAL

Copyright©2023. Acadential. All rights reserved.

## Computational Graph



$$W^{i+1} = W^i - \lambda \cdot \nabla_{W^i} L$$

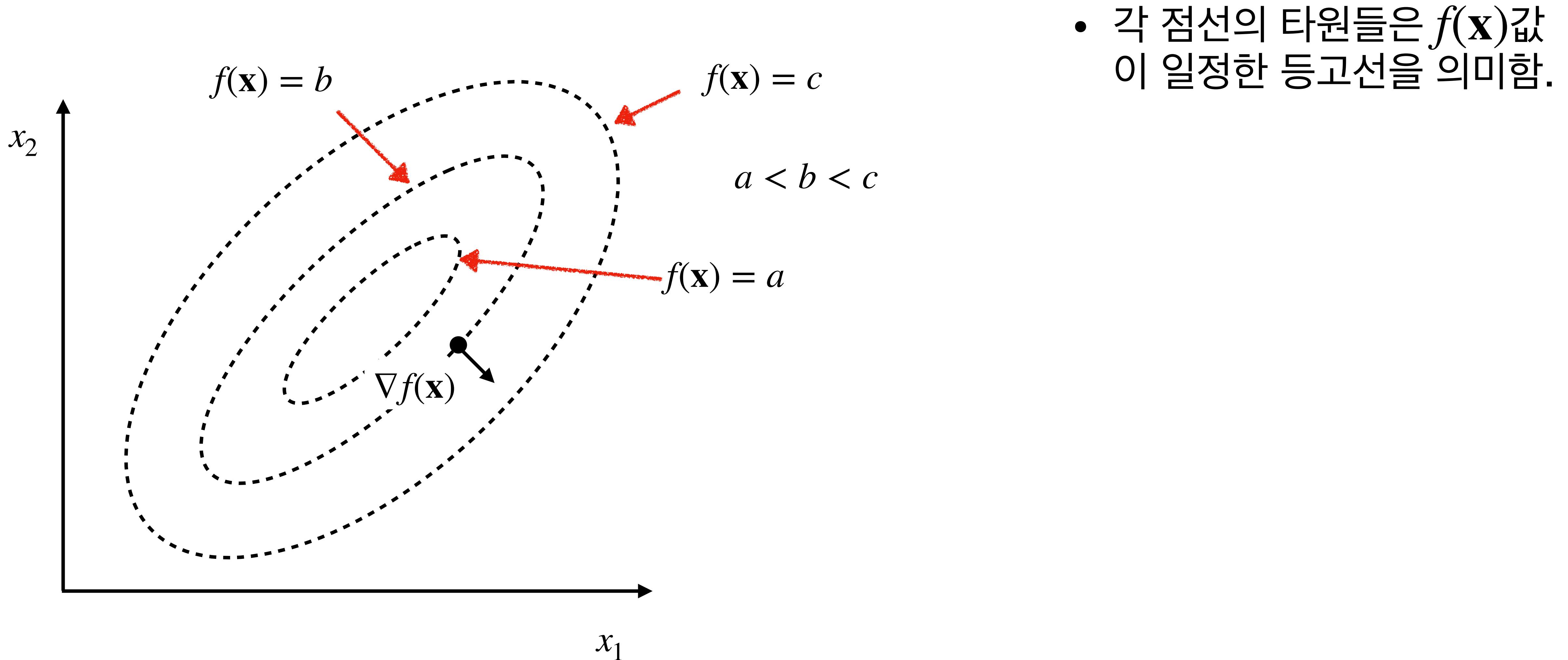
1. Descendent Node로 부터 시작 ( $t_L = 1$ )
2.  $\hat{y}$  노드의  $t_{\hat{y}}$  계산
3.  $W^{(3)}$  노드의  $t_{W^{(3)}}$  계산
4.  $h^{(2)}$  노드의  $t_{h^{(2)}}$  계산

$$t_{h^{(2)}} = t_{\hat{y}} \nabla_{h^{(2)}} \hat{y}$$

## 6-6. Gradient의 또다른 의미

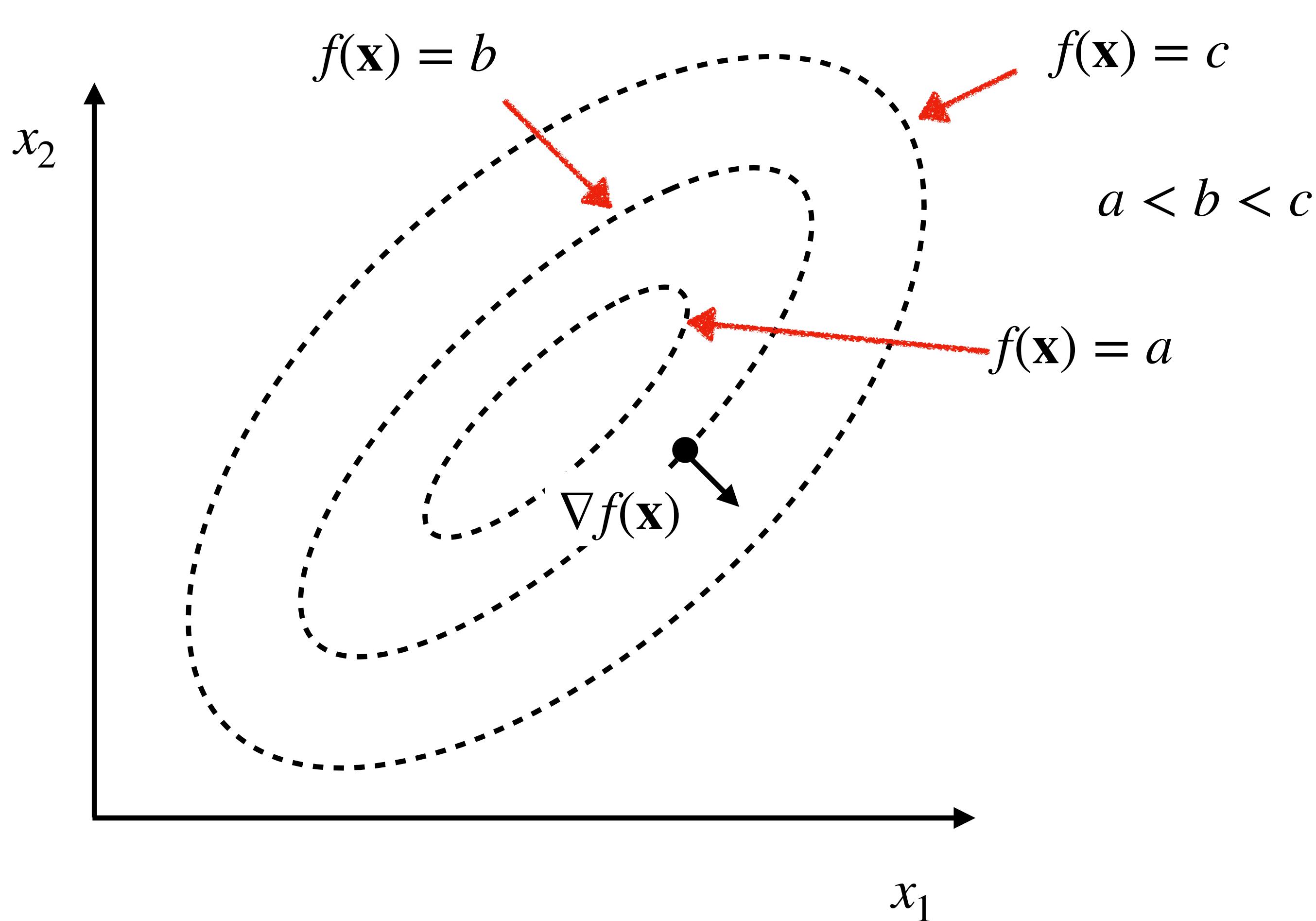
# Multivariate Gradient Descent

## Gradient의 또다른 의미



# Multivariate Gradient Descent

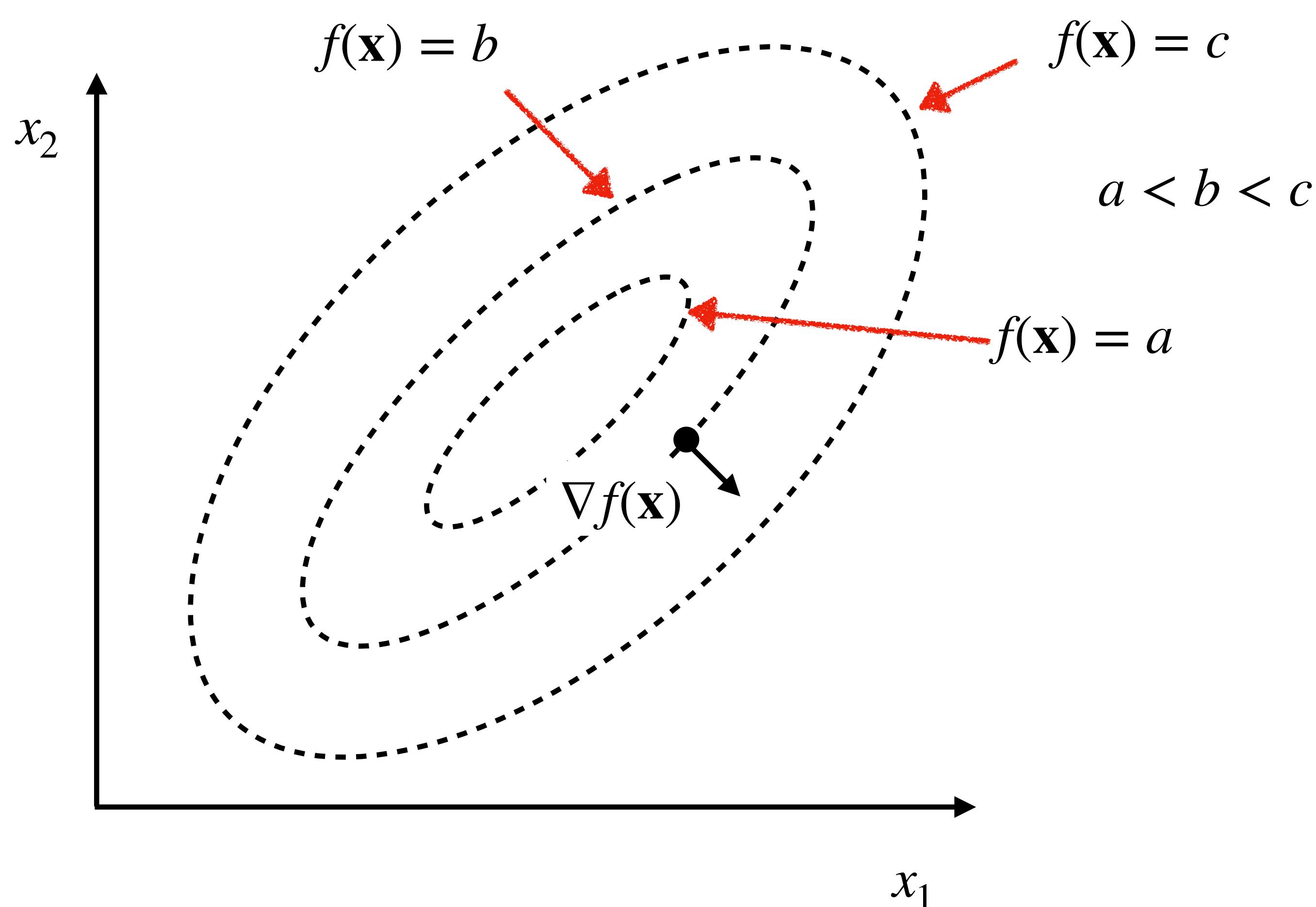
## Gradient의 또다른 의미



- 각 점선의 타원들은  $f(\mathbf{x})$ 값이 일정한 등고선을 의미함.
- 좌표  $\mathbf{x}$ 에서  $\nabla_{\mathbf{x}} f(\mathbf{x})$ 은 함수의 최대 증가폭의 방향을 향해 있다.

# Multivariate Gradient Descent

## Gradient의 또다른 의미



- 각 점선의 타원들은  $f(\mathbf{x})$ 값이 일정한 등고선을 의미함.
- 좌표  $\mathbf{x}$ 에서  $\nabla_{\mathbf{x}} f(\mathbf{x})$ 은 함수의 최대 증가폭의 방향을 향해 있다.
- 즉,  $\nabla_{\mathbf{x}} f(\mathbf{x})$ 가 가르키는 방향이 함수가 가장 가파르게 상승하는 방향이다.
- 왜 그런 것인가?

# Gradient의 또다른 의미

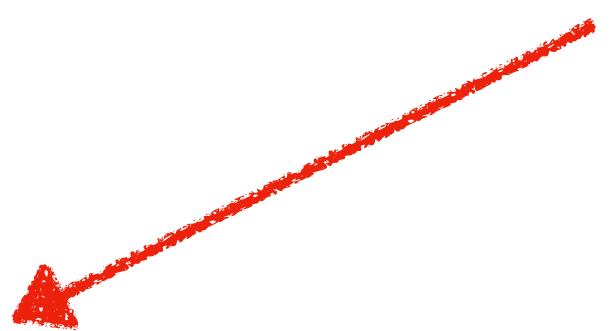
- 먼저 함수  $f(\mathbf{x})$ 가 있다고 가정해보자.
- vector  $\mathbf{x}$ 가 아주 작은 크기의 displacement vector  $\delta$  만큼 이동했을때, 즉  $\mathbf{x} \rightarrow \mathbf{x} + \delta$

# Gradient의 또다른 의미

- 먼저 함수  $f(\mathbf{x})$ 가 있다고 가정해보자.
- vector  $\mathbf{x}$ 가 아주 작은 크기의 displacement vector  $\boldsymbol{\delta}$  만큼 이동했을때, 즉  $\mathbf{x} \rightarrow \mathbf{x} + \boldsymbol{\delta}$
- 함수  $f(\mathbf{x} + \boldsymbol{\delta})$ 의 값은 다음과 같다:

$$f(\mathbf{x} + \boldsymbol{\delta}) = f(\mathbf{x}) + \sum_i \delta_i \frac{\partial f}{\partial x_i} + O(\boldsymbol{\delta}^2)$$

Taylor Expansion  
에 따라 성립!



# Gradient의 또다른 의미

## Taylor Expansion이란?

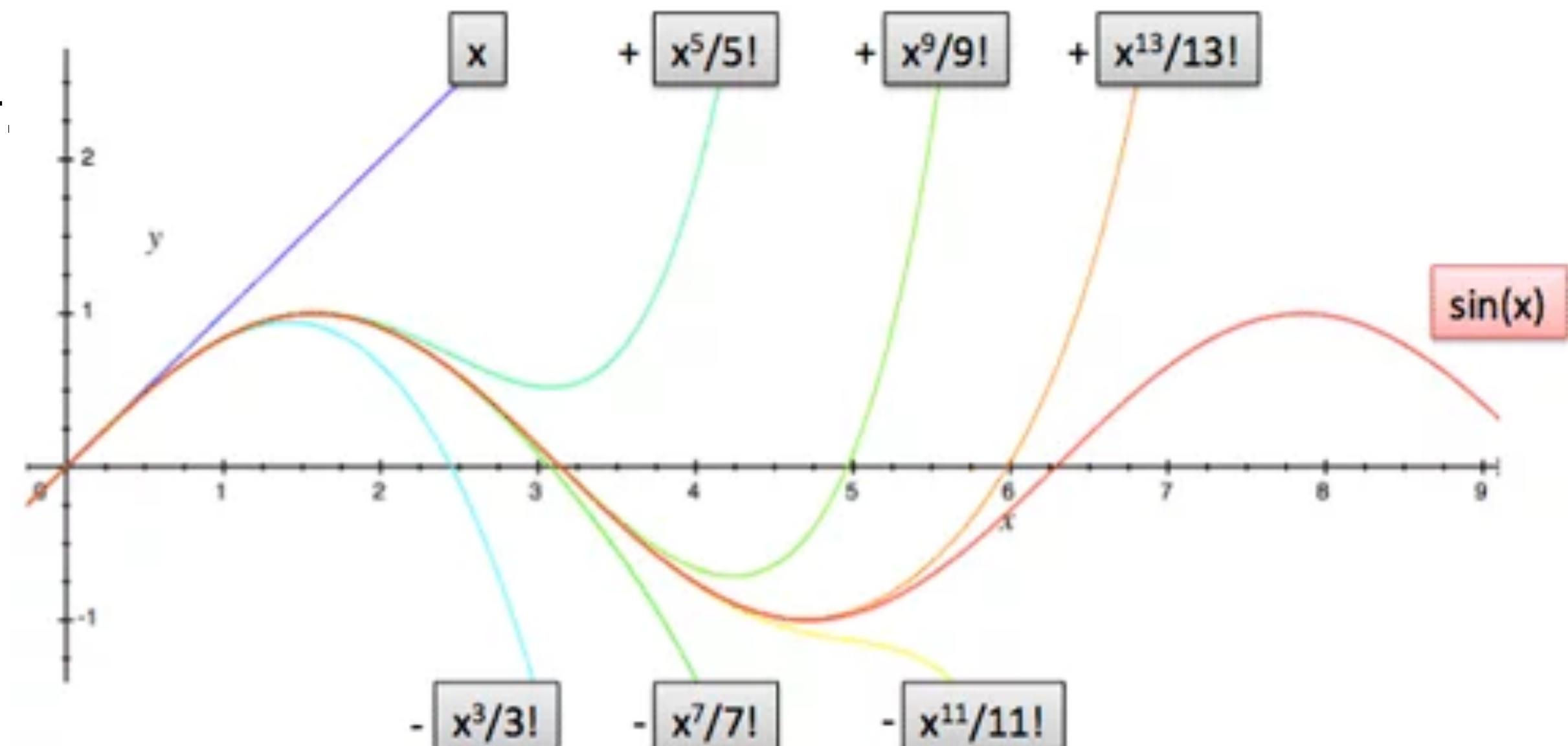
- Taylor Expansion에 대해서 간략히 설명하자면 다음과 같다.
- 어떤 임의의 함수  $f(x)$ 가 있고 아주 작은  $\delta$ 에 대해서  $f(\delta)$ 을

$$f(x + \delta) = f(x) + \frac{df}{dx}\delta + \frac{d^2f}{dx^2}\delta^2 + \frac{d^3f}{dx^3}\delta^3 + \dots$$

으로 표현할 수 있다는 것이다!

예를 들어  $x = 0$ 에서  $\sin(x)$ 을 Taylor expand하게 되었을 때:

$$\sin(\delta) = \delta - \frac{\delta^3}{3!} + \frac{\delta^5}{5!} - \frac{\delta^7}{7!}$$



출처: <https://betterexplained.com/articles/taylor-series/>

# Gradient의 또다른 의미

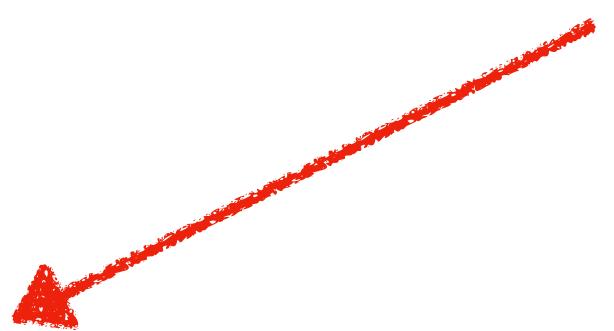
- 먼저 함수  $f(\mathbf{x})$ 가 있다고 가정해보자.
- vector  $\mathbf{x}$ 가 아주 작은 크기의 displacement vector  $\delta$  만큼 이동했을 때, 즉  $\mathbf{x} \rightarrow \mathbf{x} + \delta$
- 함수  $f(\mathbf{x} + \delta)$ 의 값은 다음과 같다:

$$f(\mathbf{x} + \delta) = f(\mathbf{x}) + \sum_i \delta_i \frac{\partial f}{\partial x_i} + O(\delta^2)$$

- 참고로, 함수  $f$ 를 vector  $\mathbf{x}$ 에 대해서 미분한 것은  $\nabla_{\mathbf{x}} f$ 이다.  $\nabla_{\mathbf{x}} f$ 의  $i$ 번째 요소는  $\frac{\partial f}{\partial x_i}$ 이다. 즉,

$$[\nabla_{\mathbf{x}} f]_i = \frac{\partial f}{\partial x_i}$$

Taylor Expansion  
에 따라 성립!



# Gradient의 또다른 의미

- 먼저 함수  $f(\mathbf{x})$ 가 있다고 가정해보자.
- vector  $\mathbf{x}$ 가 아주 작은 크기의 displacement vector  $\boldsymbol{\delta}$  만큼 이동했을때, 즉  $\mathbf{x} \rightarrow \mathbf{x} + \boldsymbol{\delta}$
- 함수  $f(\mathbf{x} + \boldsymbol{\delta})$ 의 값은 다음과 같다:

$$f(\mathbf{x} + \boldsymbol{\delta}) = f(\mathbf{x}) + \sum_i \delta_i \frac{\partial f}{\partial x_i} + O(\boldsymbol{\delta}^2)$$

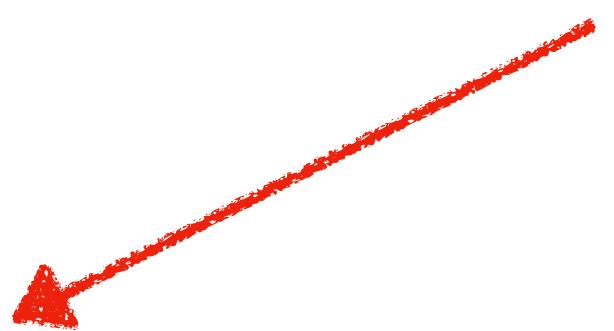
- 참고로, 함수  $f$ 를 vector  $\mathbf{x}$ 에 대해서 미분한 것은  $\nabla_{\mathbf{x}} f$ 이다.  $\nabla_{\mathbf{x}} f$ 의  $i$ 번째 요소는  $\frac{\partial f}{\partial x_i}$ 이다. 즉,

$$[\nabla_{\mathbf{x}} f]_i = \frac{\partial f}{\partial x_i}$$

- 따라서,  $\sum_i$  항은  $\nabla f$ 와 displacement vector  $\boldsymbol{\delta}$  간의 내적으로 볼 수 있다!

$$f(\mathbf{x} + \boldsymbol{\delta}) = f(\mathbf{x}) + (\nabla_{\mathbf{x}} f)^T \boldsymbol{\delta} + O(\boldsymbol{\delta}^2)$$

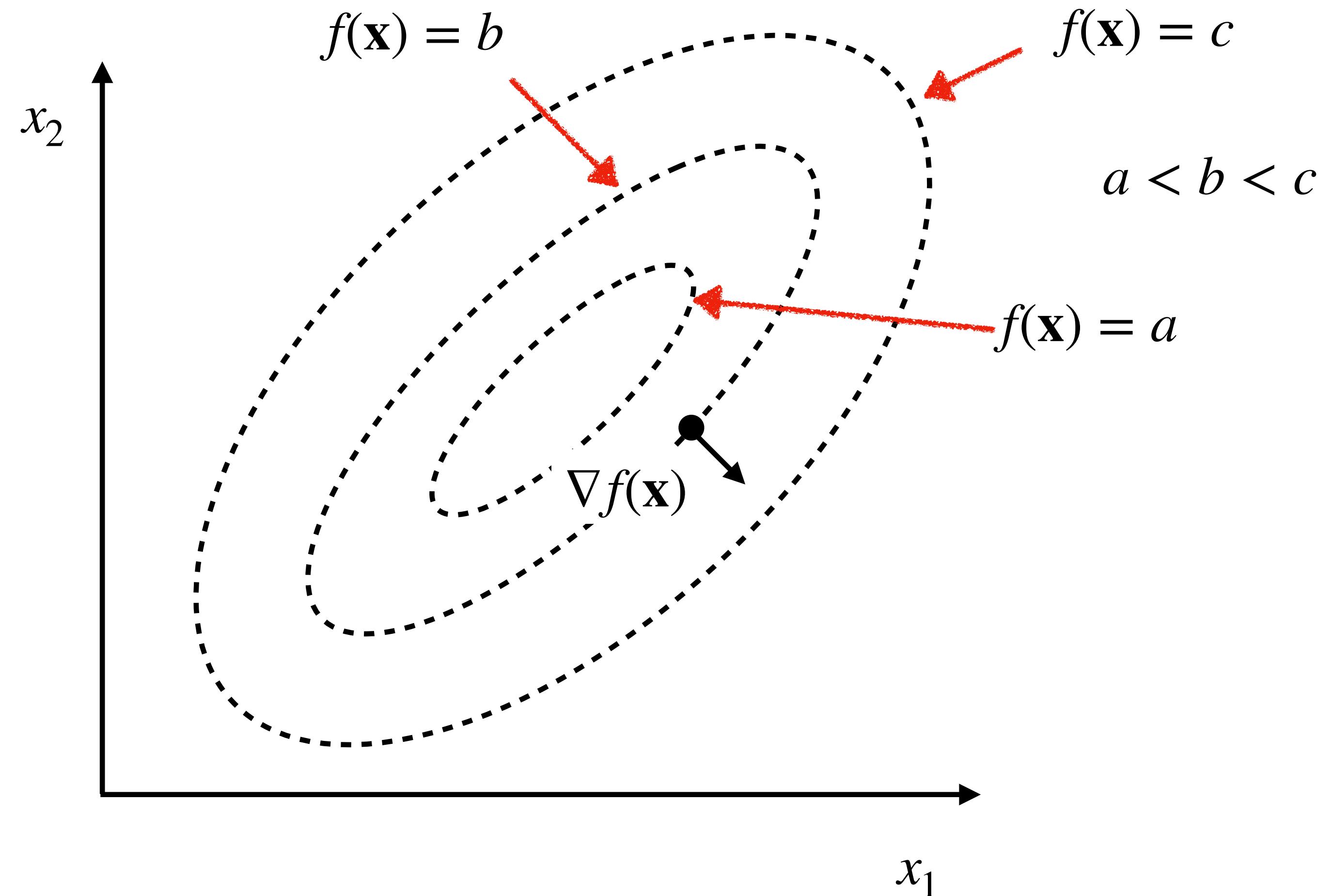
Taylor Expansion  
에 따라 성립!



# Gradient의 또다른 의미

앞에서 보다시피, 다음과 같이 근사할 수 있다:

$$f(\mathbf{x} + \boldsymbol{\delta}) \approx f(\mathbf{x}) + \nabla f(\mathbf{x}) \cdot \boldsymbol{\delta}$$



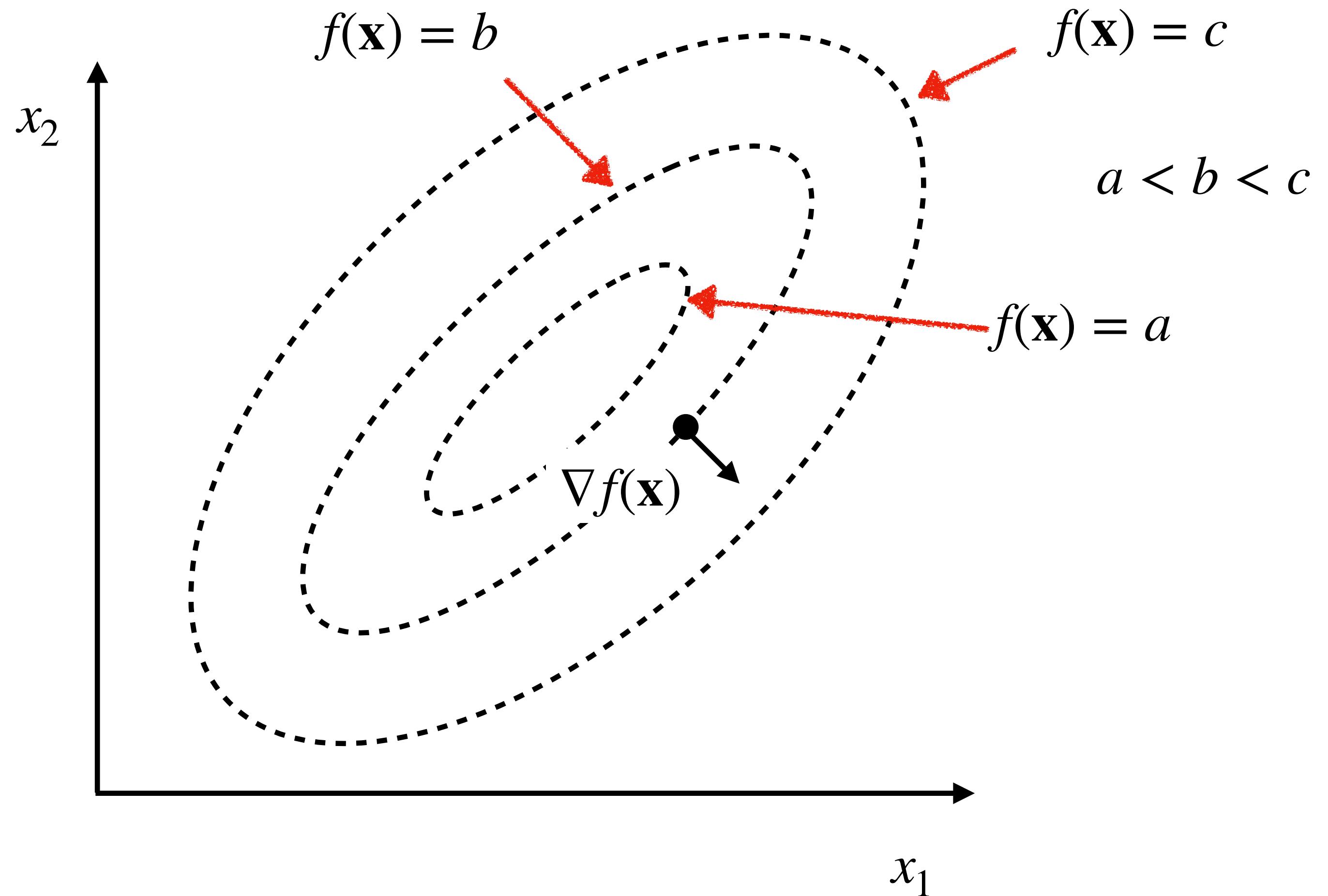
# Gradient의 또다른 의미

앞에서 보다시피, 다음과 같이 근사할 수 있다:

$$f(\mathbf{x} + \boldsymbol{\delta}) \approx f(\mathbf{x}) + \nabla f(\mathbf{x}) \cdot \boldsymbol{\delta}$$

$\hat{\mathbf{p}}$ 을  $\boldsymbol{\delta}$ 의 방향을 향하는 unit vector

$\delta$ 을  $\boldsymbol{\delta}$ 의 크기를 갖는 scalar 값



# Gradient의 또다른 의미

앞에서 보다시피, 다음과 같이 근사할 수 있다:

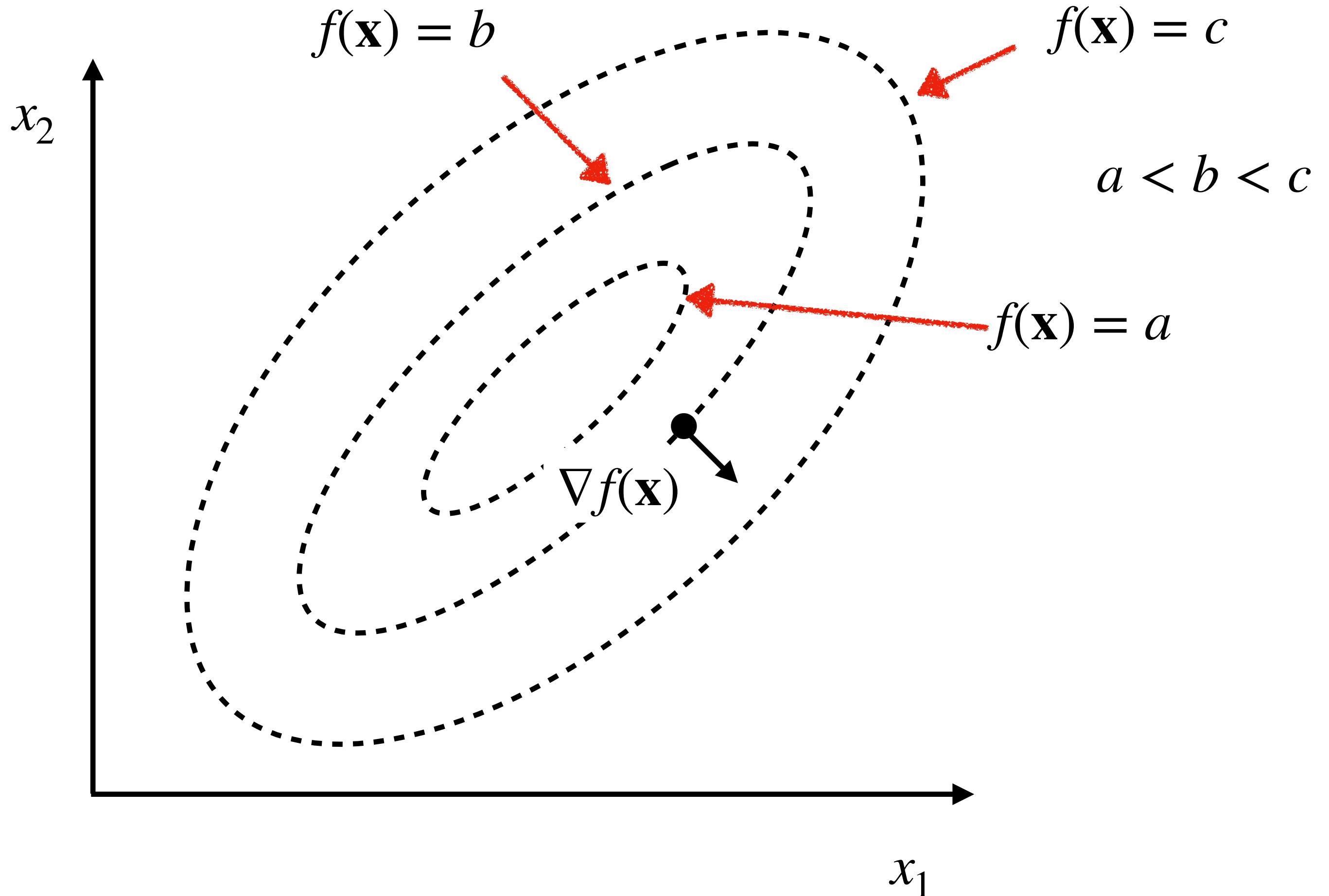
$$f(\mathbf{x} + \boldsymbol{\delta}) \approx f(\mathbf{x}) + \nabla f(\mathbf{x}) \cdot \boldsymbol{\delta}$$

$\hat{\mathbf{p}}$ 을  $\boldsymbol{\delta}$ 의 방향을 향하는 unit vector

$\delta$ 을  $\boldsymbol{\delta}$ 의 크기를 갖는 scalar 값

으로 두었을 때:

$$f(\mathbf{x} + \delta \hat{\mathbf{p}}) \approx f(\mathbf{x}) + \delta \nabla f(\mathbf{x}) \cdot \hat{\mathbf{p}}$$



# Gradient의 또 다른 의미

$$f(\mathbf{x} + \delta \hat{\mathbf{p}}) \approx f(\mathbf{x}) + \delta \boxed{\nabla f(\mathbf{x}) \cdot \hat{\mathbf{p}}}$$

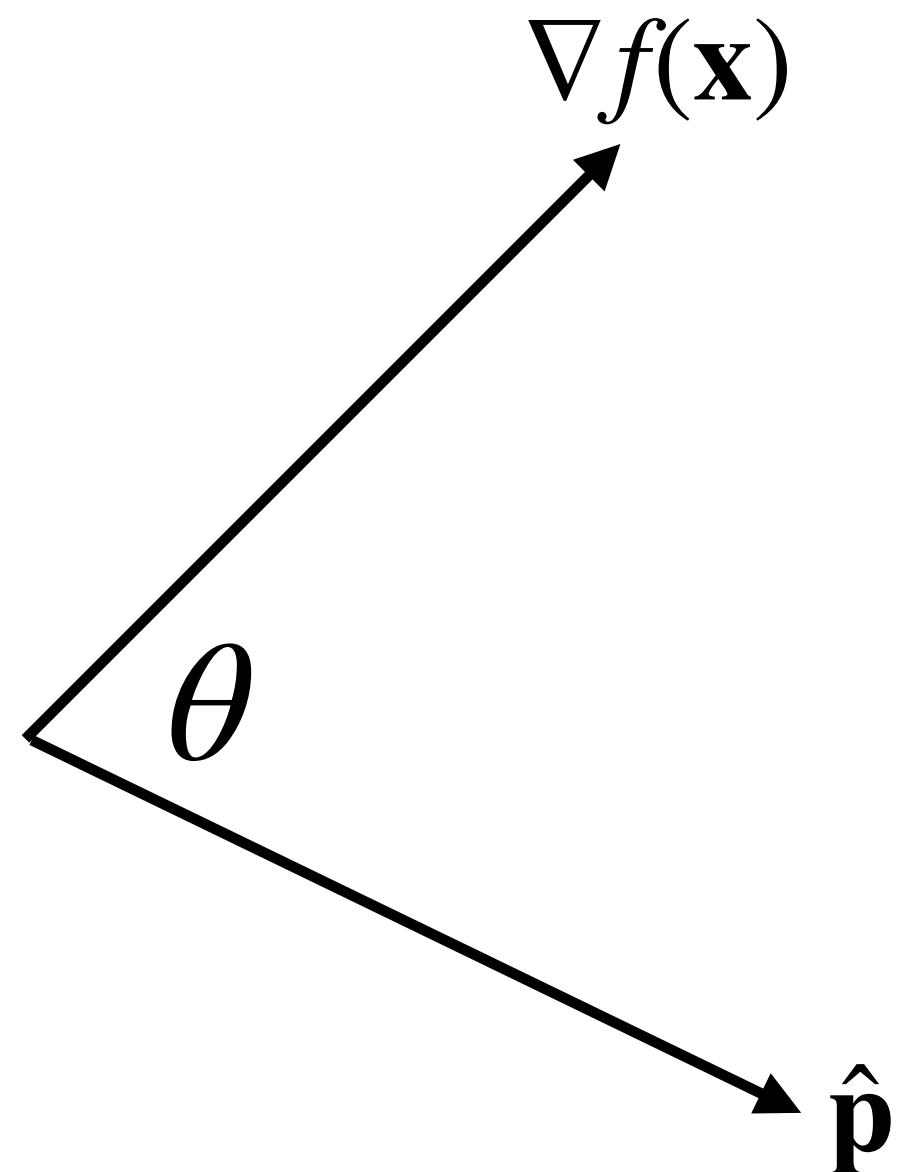
# Gradient의 또다른 의미

$$f(\mathbf{x} + \delta \hat{\mathbf{p}}) \approx f(\mathbf{x}) + \delta \boxed{\nabla f(\mathbf{x}) \cdot \hat{\mathbf{p}}}$$

여기서

$$\begin{aligned}\nabla f(\mathbf{x}) \cdot \hat{\mathbf{p}} &= |\nabla f(\mathbf{x})| |\hat{\mathbf{p}}| \cos \theta \\ &= |\nabla f(\mathbf{x})| \cos \theta\end{aligned}$$

여기서  $\theta$ 는  $\hat{\mathbf{p}}$ 와  $\nabla f(\mathbf{x})$  vector 사이의 각도이다.



# Gradient의 또다른 의미

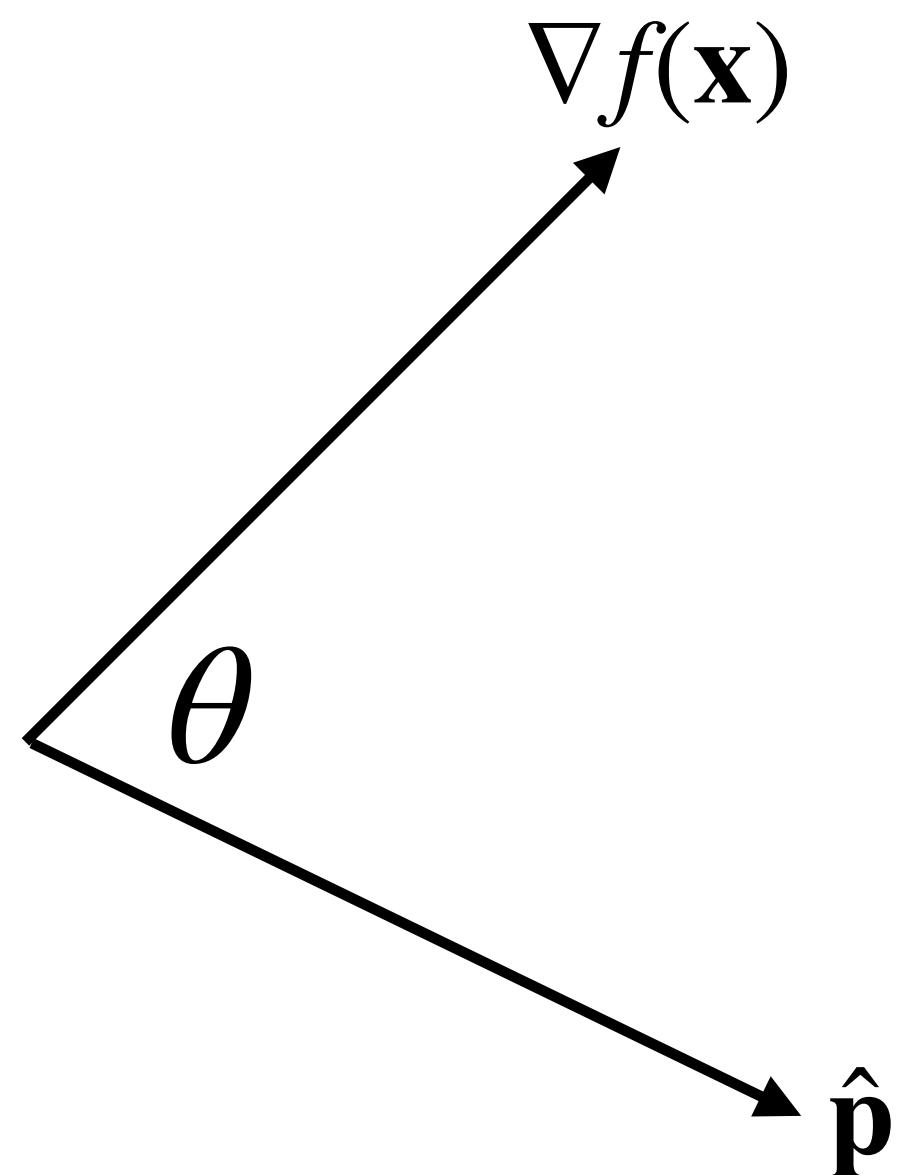
$$f(\mathbf{x} + \delta \hat{\mathbf{p}}) \approx f(\mathbf{x}) + \delta \boxed{\nabla f(\mathbf{x}) \cdot \hat{\mathbf{p}}}$$

여기서

$$\begin{aligned}\nabla f(\mathbf{x}) \cdot \hat{\mathbf{p}} &= |\nabla f(\mathbf{x})| |\hat{\mathbf{p}}| \cos \theta \\ &= |\nabla f(\mathbf{x})| \cos \theta\end{aligned}$$

여기서  $\theta$  는  $\hat{\mathbf{p}}$ 와  $\nabla f(\mathbf{x})$  vector 사이의 각도이다.

$\theta = 0$  일때  $\cos \theta$  가 가장 크고, 이때  $\nabla f(\mathbf{x}) \cdot \hat{\mathbf{p}}$ 가 **최대값 (maximal change)** 을 가진다.



# Gradient의 또다른 의미

$$f(\mathbf{x} + \delta \hat{\mathbf{p}}) \approx f(\mathbf{x}) + \delta \boxed{\nabla f(\mathbf{x}) \cdot \hat{\mathbf{p}}}$$

여기서

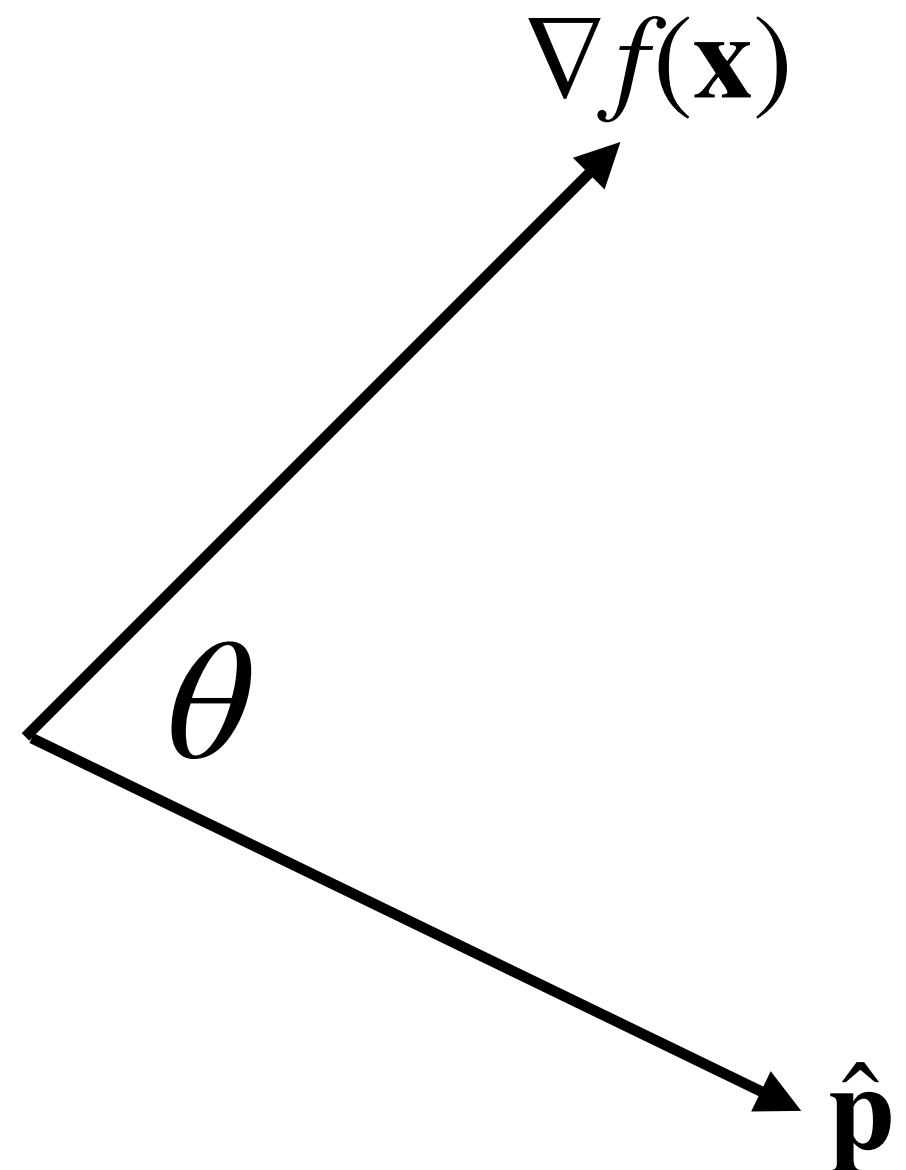
$$\begin{aligned}\nabla f(\mathbf{x}) \cdot \hat{\mathbf{p}} &= |\nabla f(\mathbf{x})| |\hat{\mathbf{p}}| \cos \theta \\ &= |\nabla f(\mathbf{x})| \cos \theta\end{aligned}$$

여기서  $\theta$ 는  $\hat{\mathbf{p}}$ 와  $\nabla f(\mathbf{x})$  vector 사이의 각도이다.

$\theta = 0$  일때  $\cos \theta$ 가 가장 크고, 이때  $\nabla f(\mathbf{x}) \cdot \hat{\mathbf{p}}$ 가 최대값 (**maximal change**) 을 가진다.

즉 **displacement**  $\hat{\mathbf{p}}$ 의 방향이  $\nabla f(\mathbf{x})$ 의 방향과 일치할때 증가량이 가장 크다.

$\nabla f(\mathbf{x})$ 가 가르키는 방향이 함수가 가장 가파르게 상승하는 방향이다!

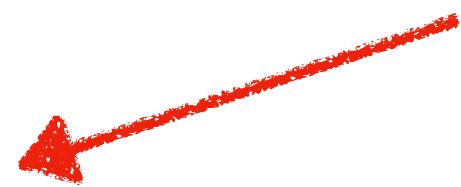


## **6-7. 경사 하강이 잘 작동하기 위한 전제 조건**

# 경사하강이 가능한 경우, 전제 조건

- 경사하강을 통해서 Loss를 줄여나갈 수 있다.
- 하지만 경사하강을 통해서 수렴된 지점이 과연 **Global Minimum**일까?

Weight space 상에서 Loss가 가장 낮은 지점

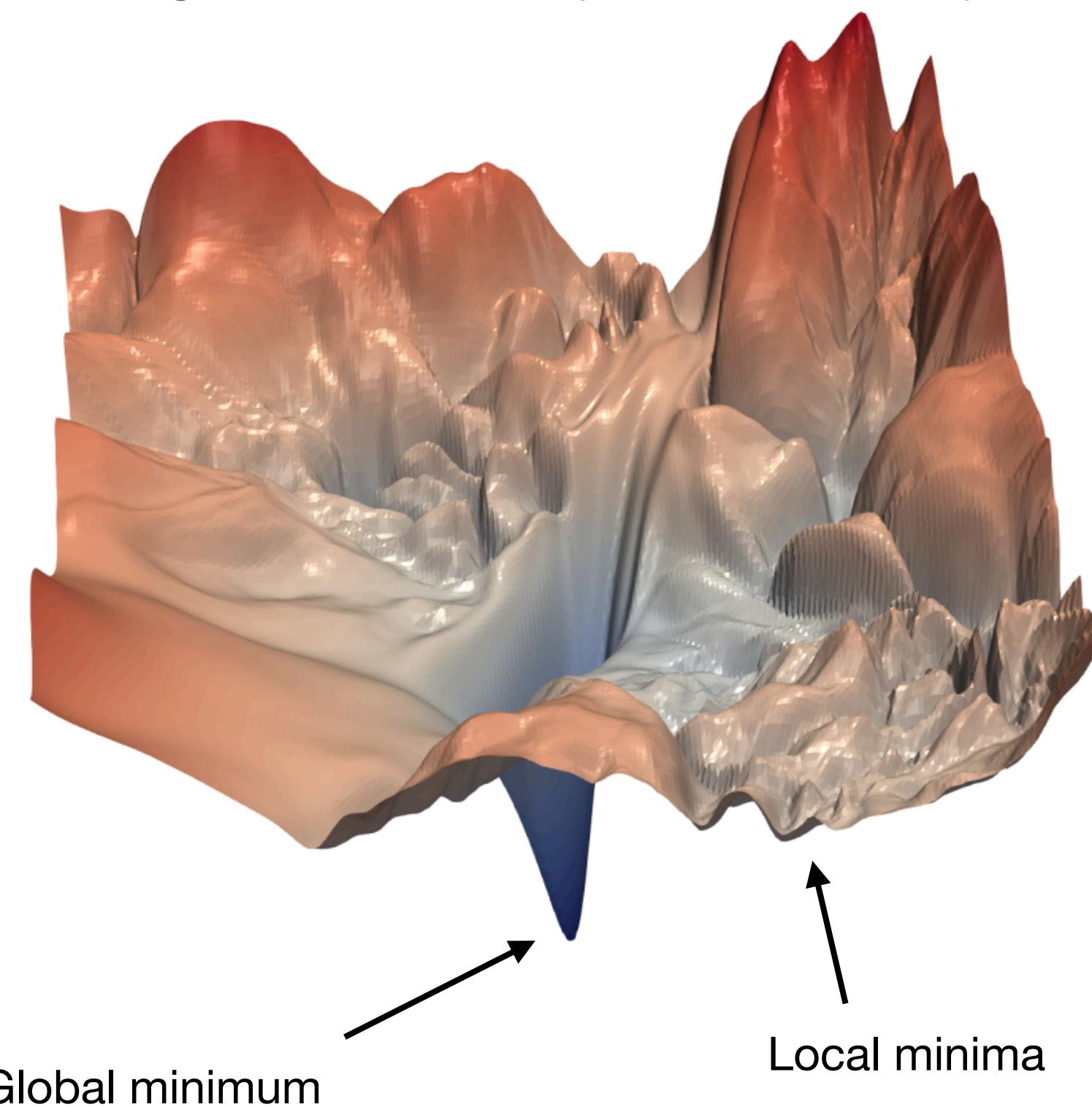


# 경사하강이 가능한 경우, 전제 조건

- 경사하강을 통해서 Loss를 줄여나갈 수 있다.
- 하지만 경사하강을 통해서 수렴된 지점이 과연 **Global Minimum**일까?

Weight space 상에서 Loss가 가장 낮은 지점

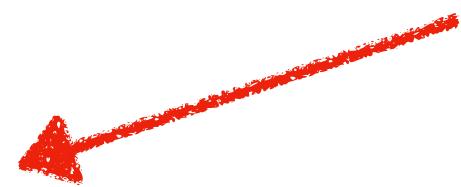
출처: Visualizing the Loss Landscape of Neural Nets (NeurIPS 2018)



# 경사하강이 가능한 경우, 전제 조건

- 경사하강을 통해서 Loss를 줄여나갈 수 있다.
- 하지만 경사하강을 통해서 수렴된 지점이 과연 **Global Minimum**일까?
- Global Minimum을 찾기 위한 전제조건은?

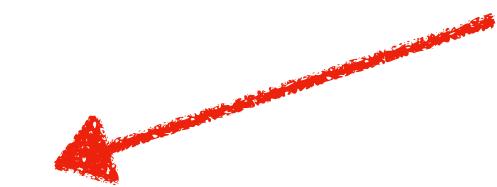
Weight space 상에서 Loss가 가장 낮은 지점



# 경사하강이 가능한 경우, 전제 조건

- 경사하강을 통해서 Loss를 줄여나갈 수 있다.
- 하지만 경사하강을 통해서 수렴된 지점이 과연 **Global Minimum**일까?
- Global Minimum을 찾기 위한 전제조건은?

Weight space 상에서 Loss가 가장 낮은 지점



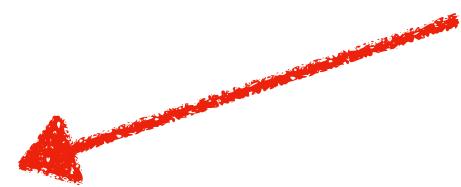
**바로 Convexity!**

즉, 아래로 볼록한 조건을 만족시켜야 한다!

# 경사하강이 가능한 경우, 전제 조건

- 경사하강을 통해서 Loss를 줄여나갈 수 있다.
- 하지만 경사하강을 통해서 수렴된 지점이 과연 **Global Minimum**일까?
- Global Minimum을 찾기 위한 전제조건은?

Weight space 상에서 Loss가 가장 낮은 지점



**바로 Convexity!**

즉, 아래로 볼록한 조건을 만족시켜야 한다!

그렇다면

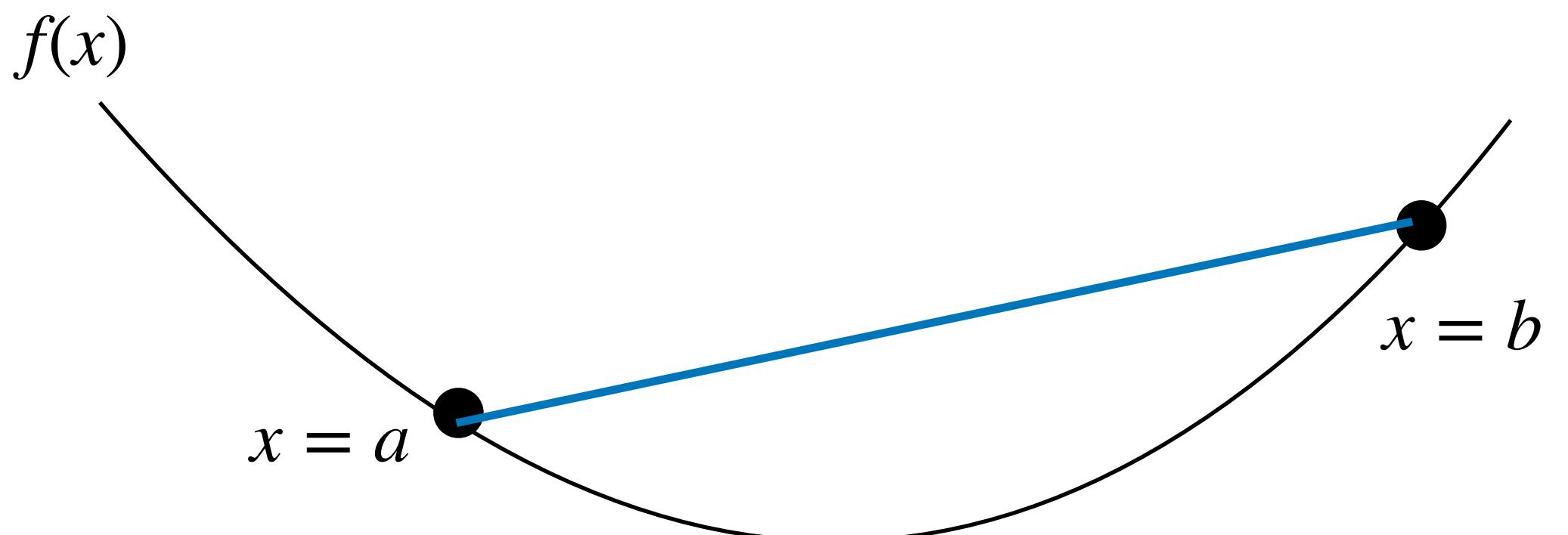
“아래로 볼록하다”의 조건은?

# 경사하강이 가능한 경우, 전제 조건

- Global Minimum을 찾기 위한 전제조건은?

**바로 Convexity!**

**Convexity의 조건은**  $= \frac{d^2L}{dw^2} \geq 0$  (모든 w에 대해서)

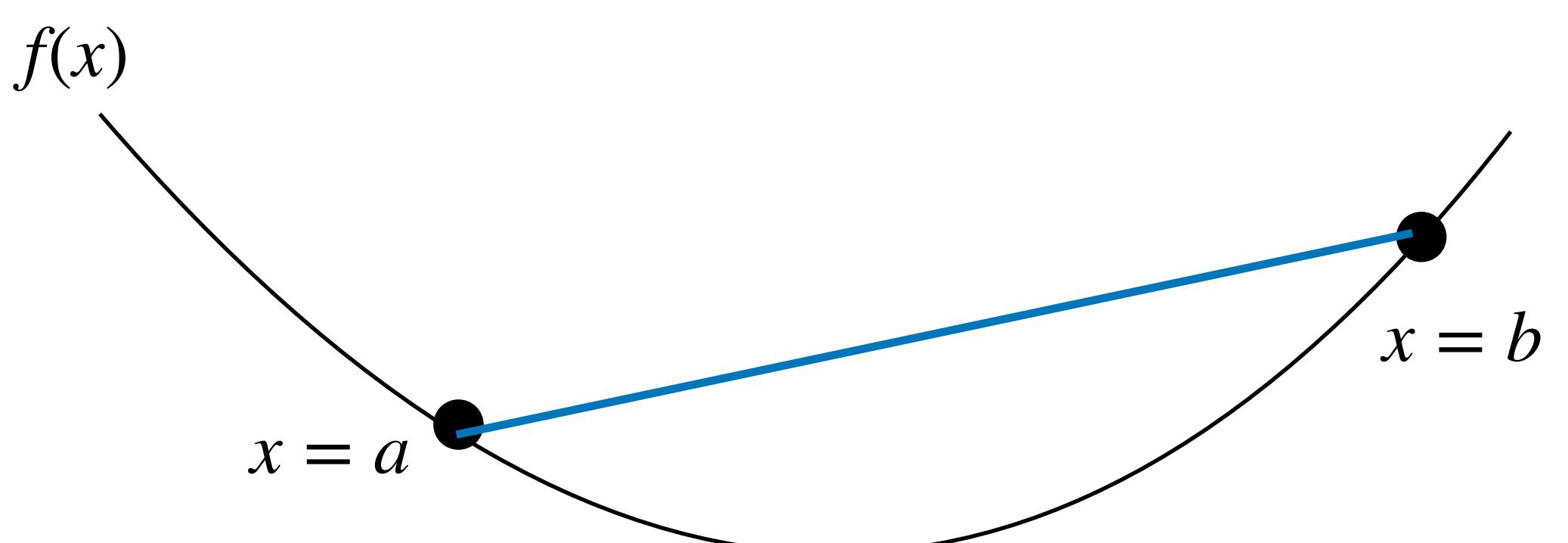


# 경사하강이 가능한 경우, 전제 조건

- Global Minimum을 찾기 위한 전제조건은?

바로 Convexity!

Convexity의 조건은  $\frac{d^2L}{dw^2} \geq 0$  (모든 w에 대해서)



또 다른 조건으로는 다음 식을 만족시키는 것이다!

$$f(\theta \cdot a + (1 - \theta) \cdot b) \leq \theta \cdot f(a) + (1 - \theta) \cdot f(b)$$

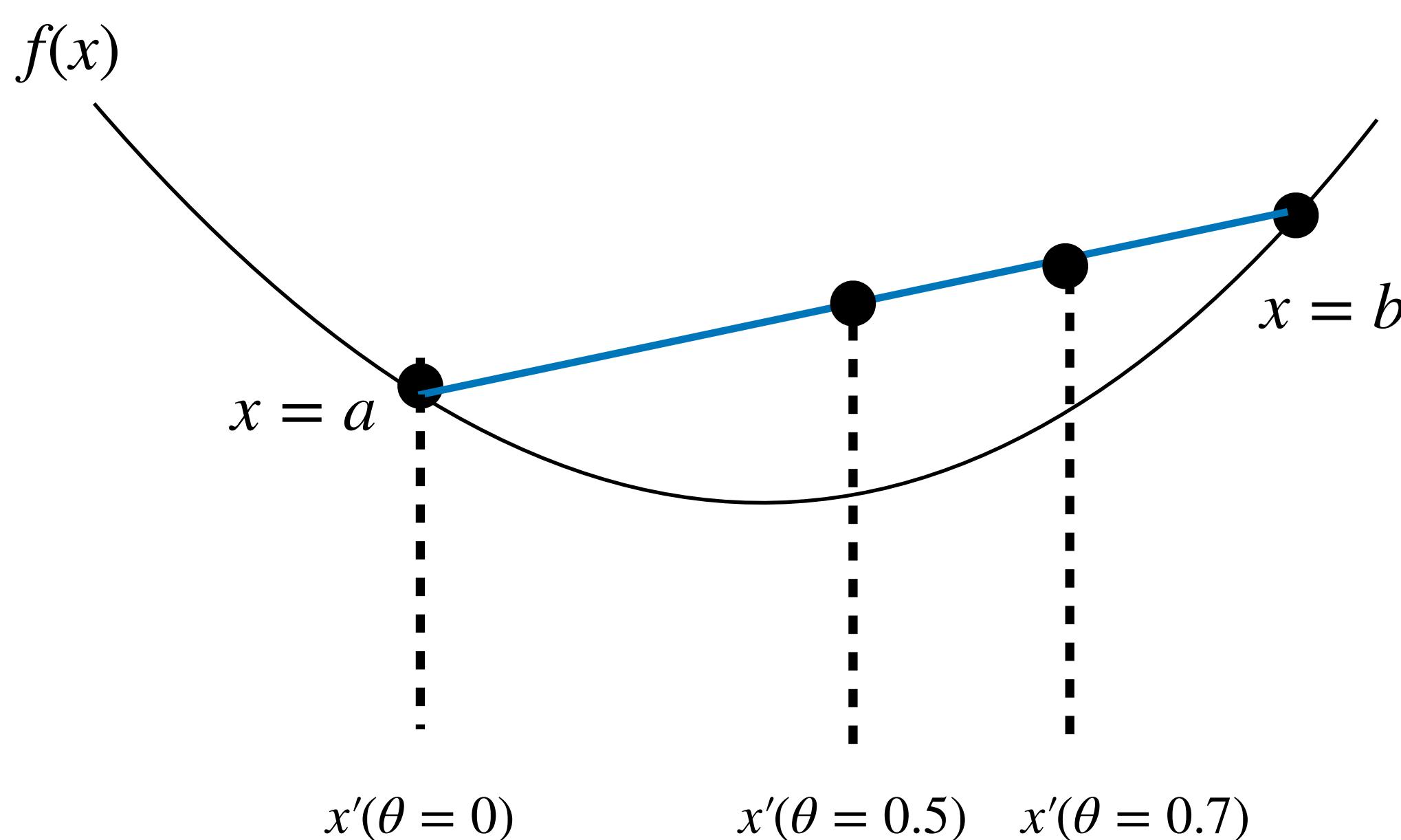
이것은 어떤 의미일까?

# 경사하강이 가능한 경우, 전제 조건

- Global Minimum을 찾기 위한 전제조건은?

**바로 Convexity!**

**Convexity의 조건은**  $= \frac{d^2L}{dw^2} \geq 0$  (모든 w에 대해서)



또 다른 조건으로는 다음 식을 만족시키는 것이다!

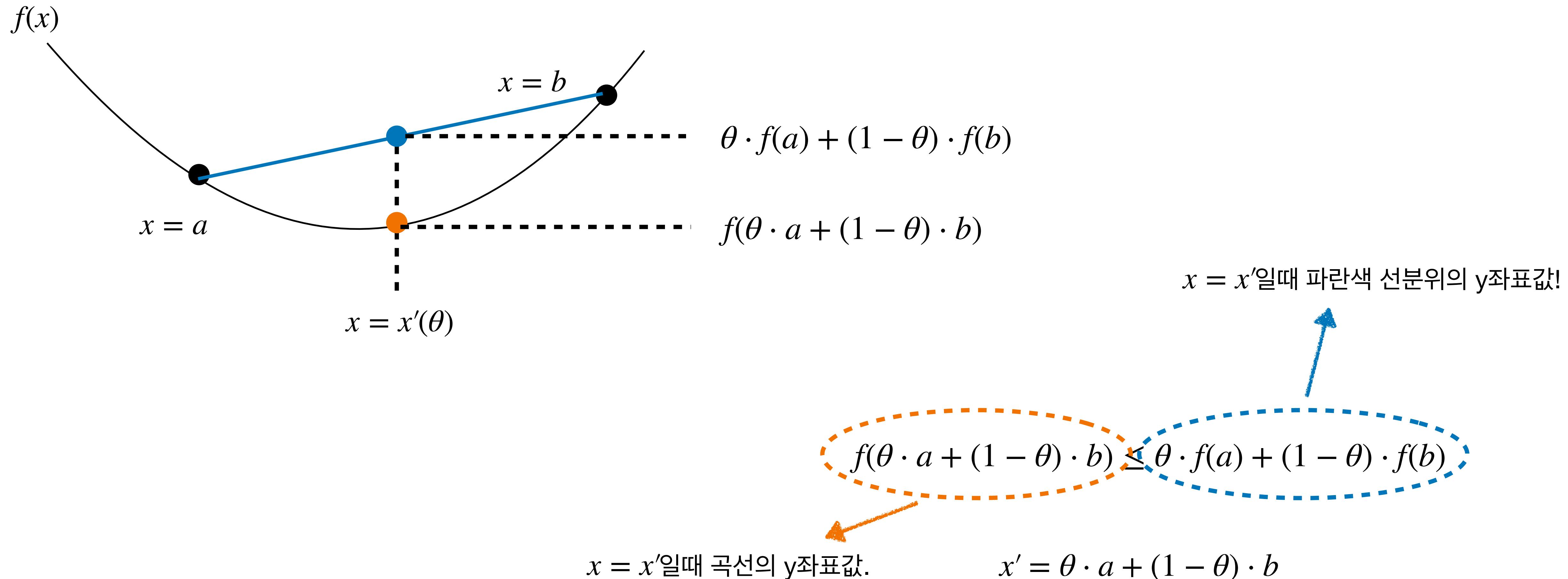
$$f(\theta \cdot a + (1 - \theta) \cdot b) \leq \theta \cdot f(a) + (1 - \theta) \cdot f(b)$$

이것은 어떤 의미일까?

$$x' = \theta \cdot a + (1 - \theta) \cdot b$$

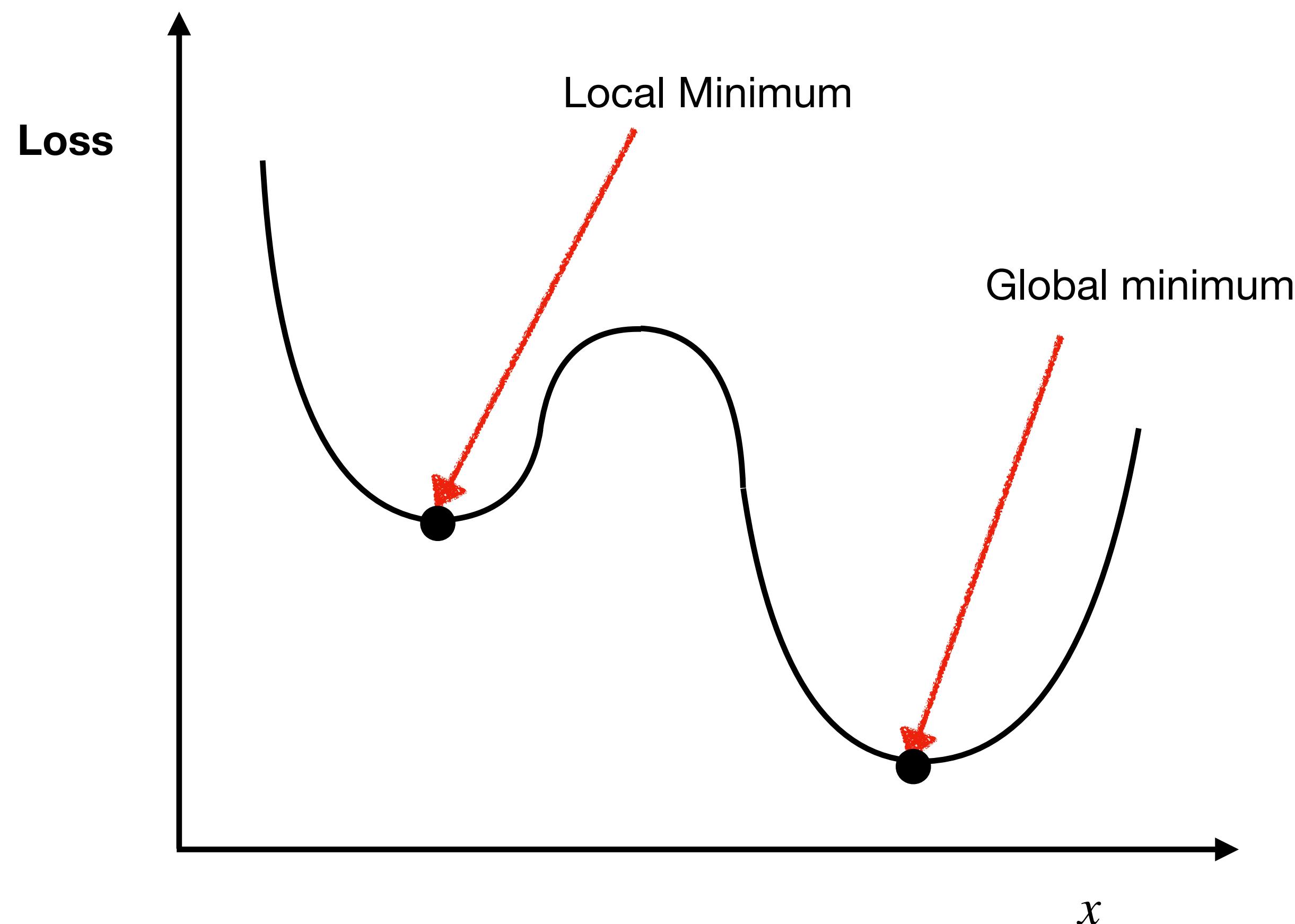
- 참고로  $\theta$ 은 0~1사이의 값이다.
- $\theta$ 가 0에 가까울수록  $x'$ 은  $a$ 에 가깝고,
- $\theta$ 가 1에 가까울수록  $x'$ 은  $b$ 에 가깝다!

# 경사하강이 가능한 경우, 전제 조건



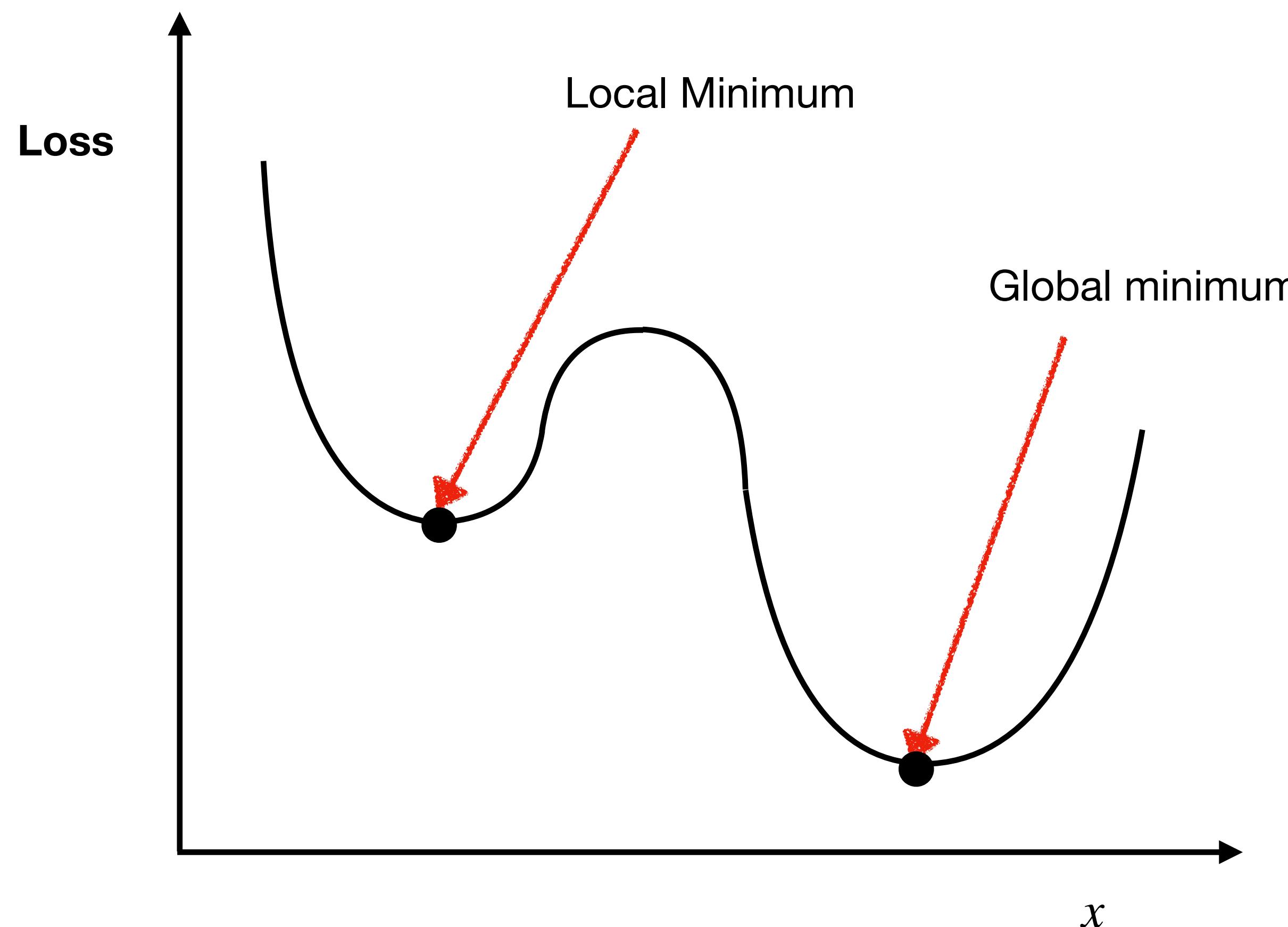
# 경사하강이 가능한 경우, 전제 조건

## Non-convex한 경우에 대해서 경사하강하면?



# 경사하강이 가능한 경우, 전제 조건

## Non-convex한 경우에 대해서 경사하강하면?



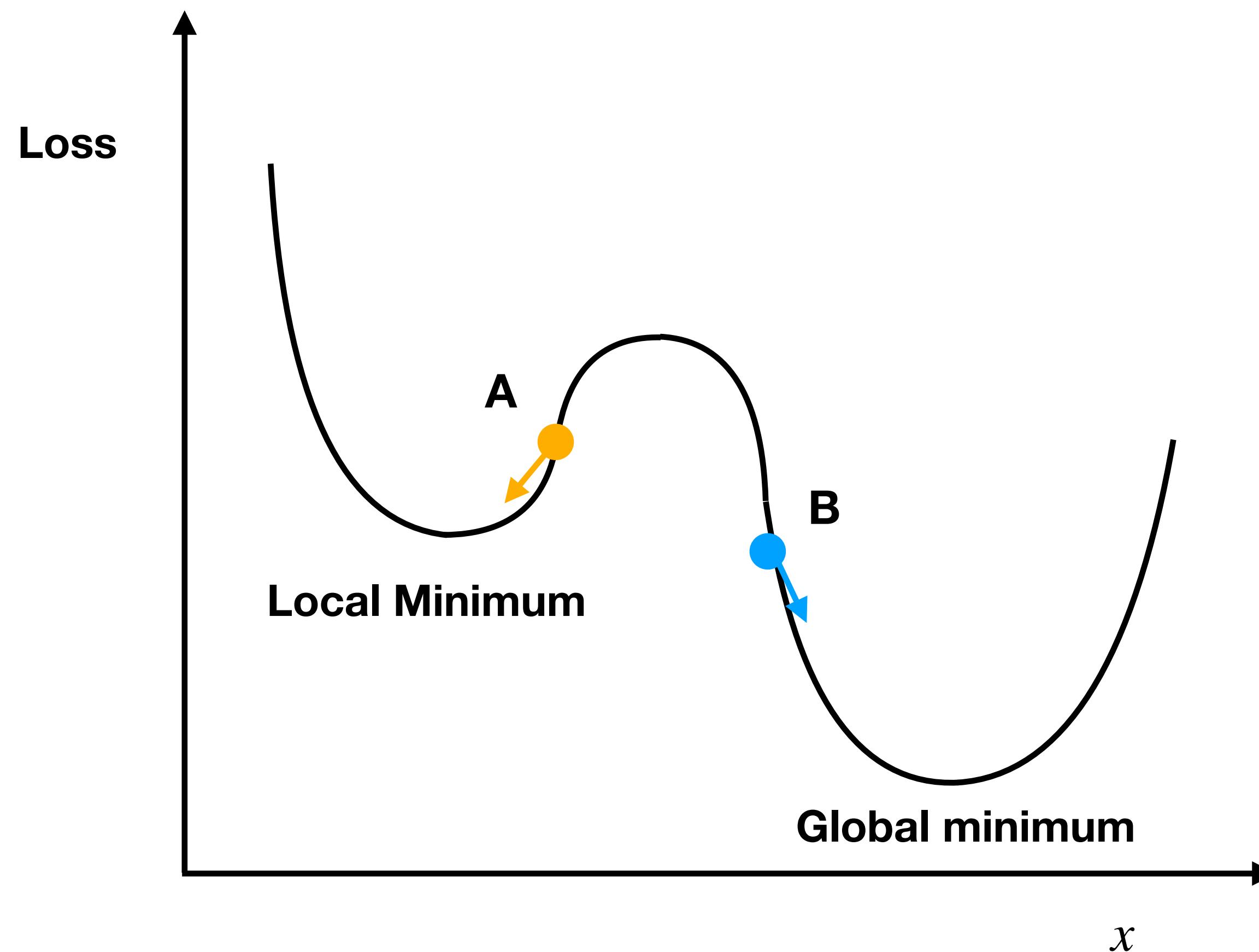
### Non Convex의 경우

Non-convex해도 경사하강은 가능하나

경사하강으로 모델의 weight값들이 수렴하여도  
수렴한 지점이 Global minimum이 아니고  
**Local minimum**일 수 있다!

# 경사하강이 가능한 경우, 전제 조건

## Non-convex한 경우에 대해서 경사하강하면?

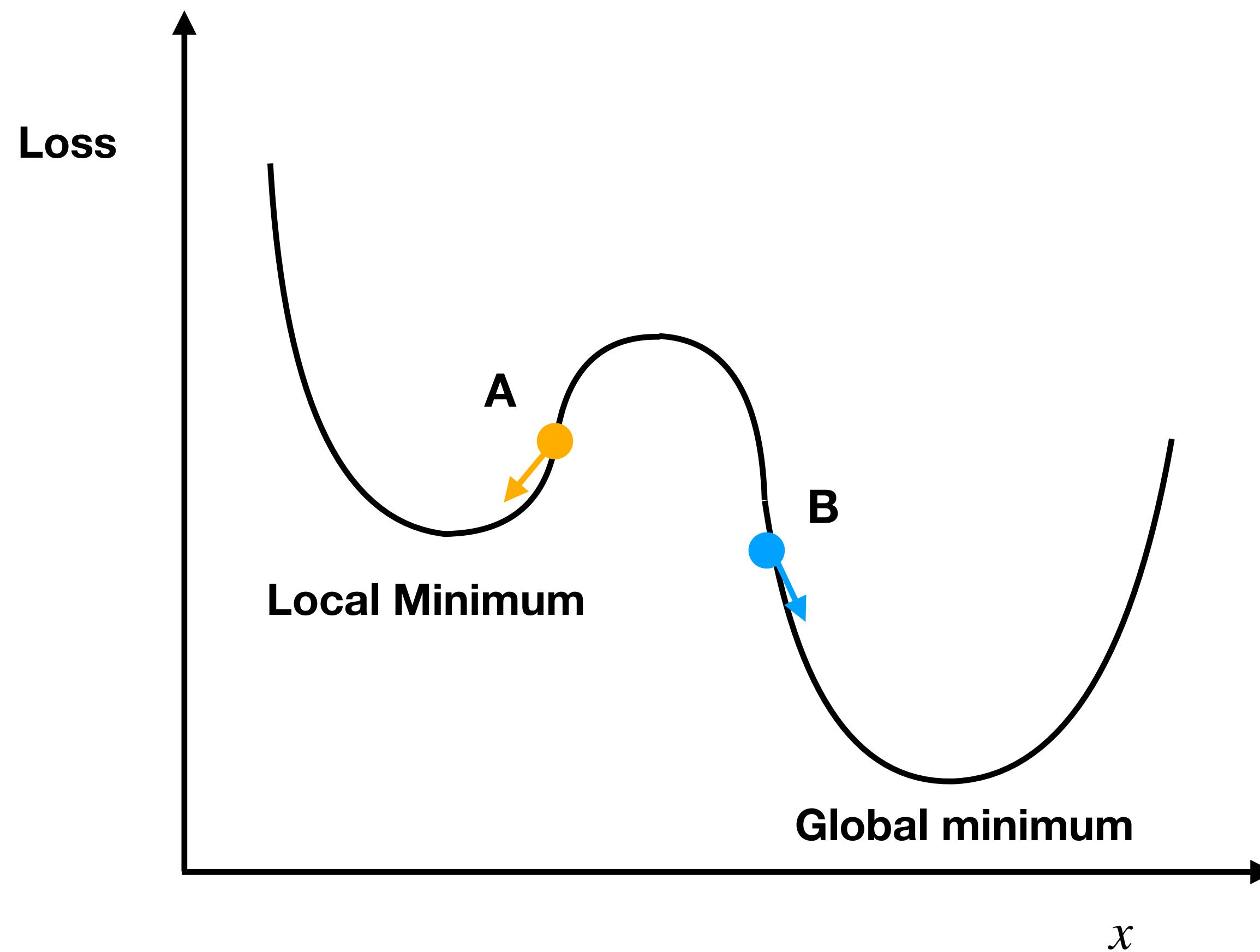


만약에 모델이 A 지점에서 시작했을시, Local minimum에 도달하고,

만약에 모델이 B 지점에서 시작했을시, Global minimum에 도달한다!

# 경사하강이 가능한 경우, 전제 조건

Non-convex한 경우에 대해서 경사하강하면?



만약에 모델이 A 지점에서 시작했을시, Local minimum에 도달하고,

만약에 모델이 B 지점에서 시작했을시, Global minimum에 도달한다!

즉 모델의 시작 지점 (initialization)에 따라 경사하강으로 최종 수렴된 모델의 성능이 다를 수 있다!

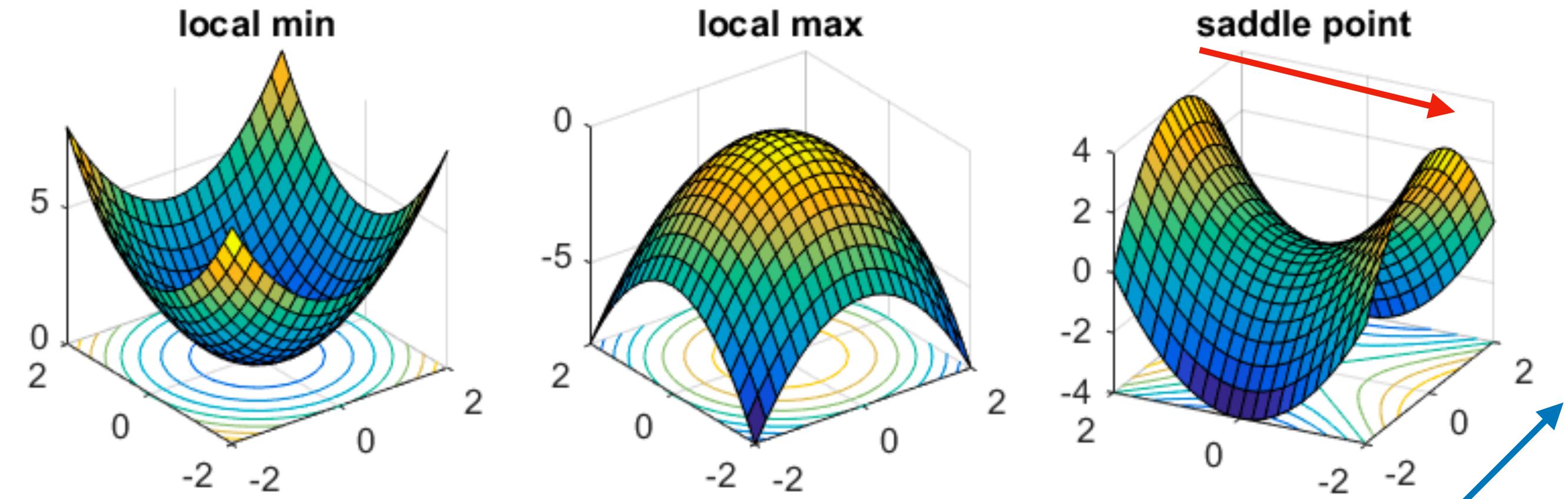
→ Weight Initialization의 중요성! (나중에 다룰 예정)

# 경사하강이 가능한 경우, 전제 조건

## Saddle Point이란?

Saddle Point란

- $\nabla f = \mathbf{0}$  이지만
- Minimum도 Maximum도 아닌 경우
- **빨간색 방향**에 대해서는 **Minimum**이다.
- **파란색 방향**에 대해서는 **Maximum**이다.



출처: <https://www.offconvex.org/2016/03/22/saddlepoints/>

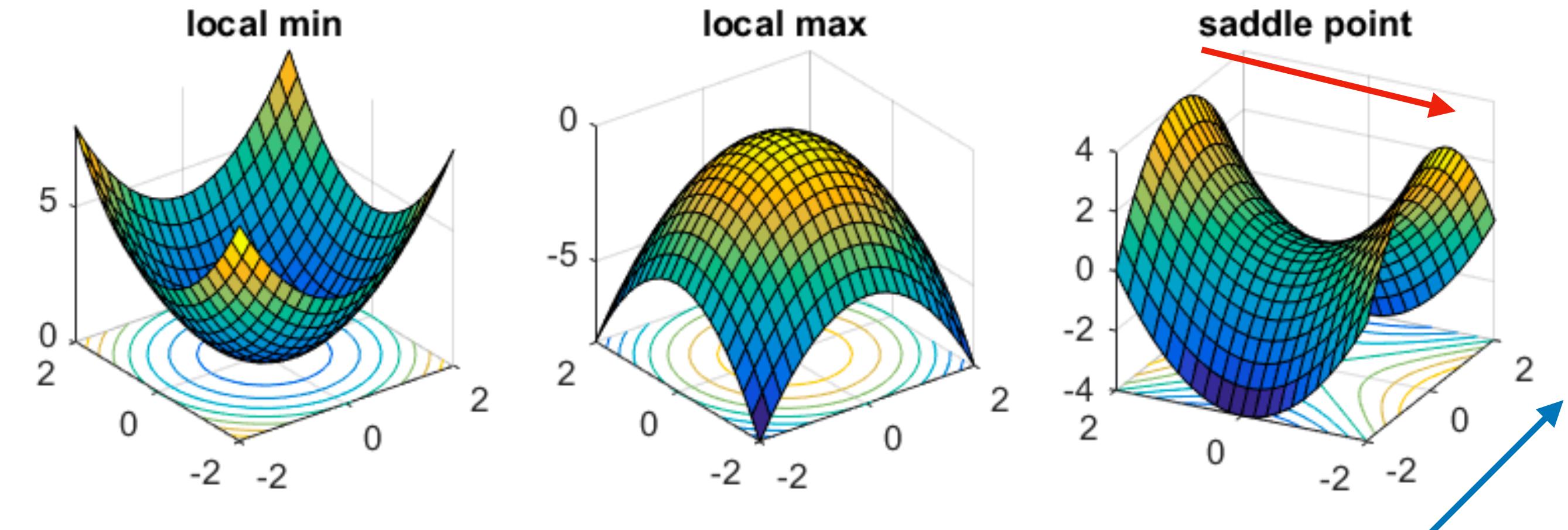
# 경사하강이 가능한 경우, 전제 조건

## Saddle Point이란?

왜 문제가 되는가?

$\nabla f = \mathbf{0}$  (즉 경사가 0)이므로 경사하강  
으로 더 이상 최적화하지 못한다.

즉, saddle point을 지나게 되는 경우 해  
당 saddle point에 수렴해 버릴 수 있다.

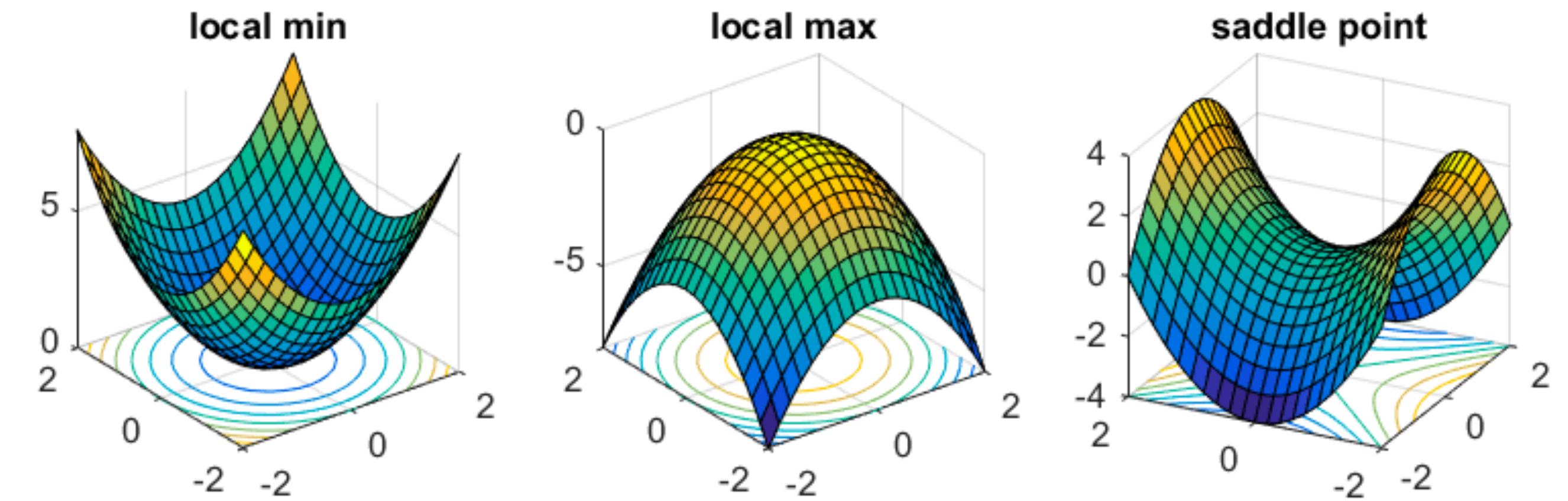
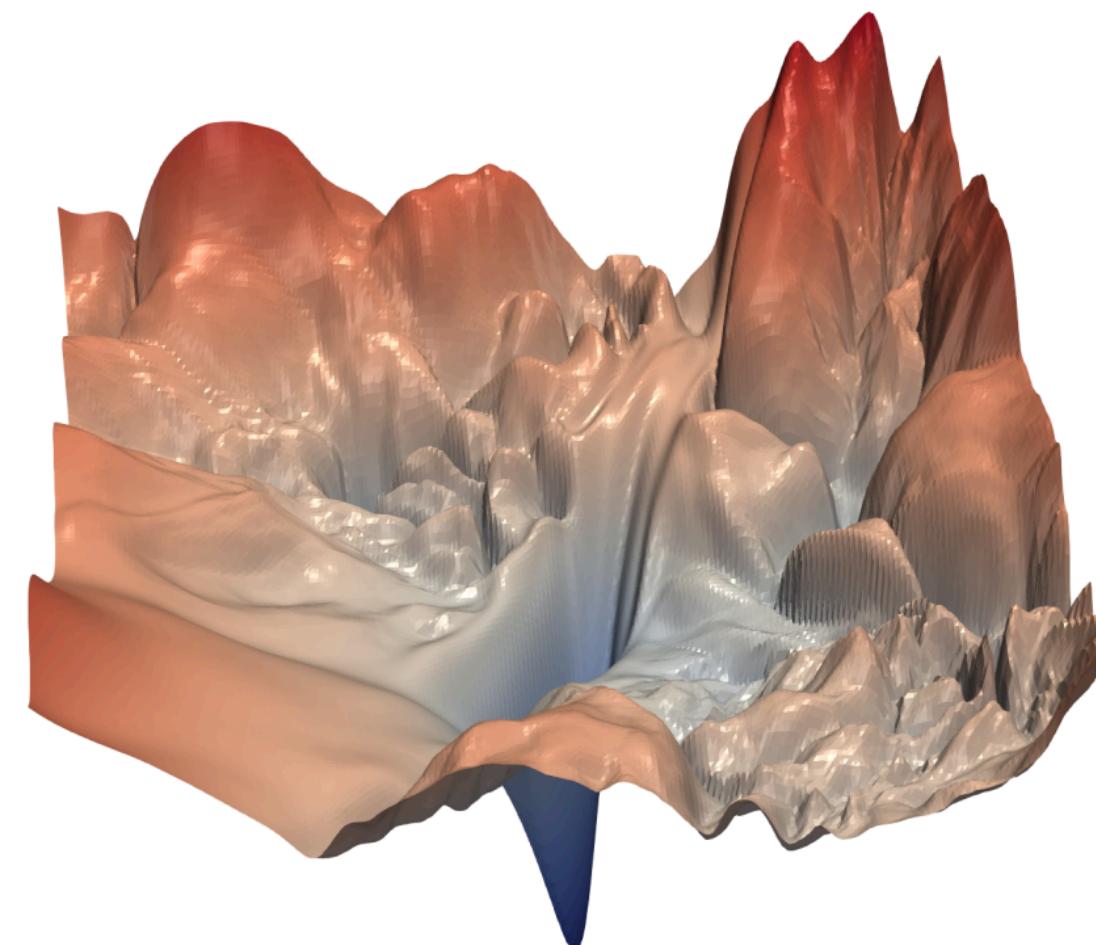


출처: <https://www.offconvex.org/2016/03/22/saddlepoints/>

# 경사하강이 가능한 경우, 전제 조건 해결책

Non-convex한 경우 경사하강은 **Local minimum** 혹은 **saddle point**에 수렴해버릴 수 있다.

이것을 어떻게 해결할 수 있을까? → **Momentum (관성)**, **Mini-batch Stochastic Gradient Descent**을 사용하는 것이 도움이 된다. (추후에 다룰 예정!)



# 6-8. PyTorch로 구현해보는 Gradient Descent

# 6-9. Section 6 요약

# Section 6 요약

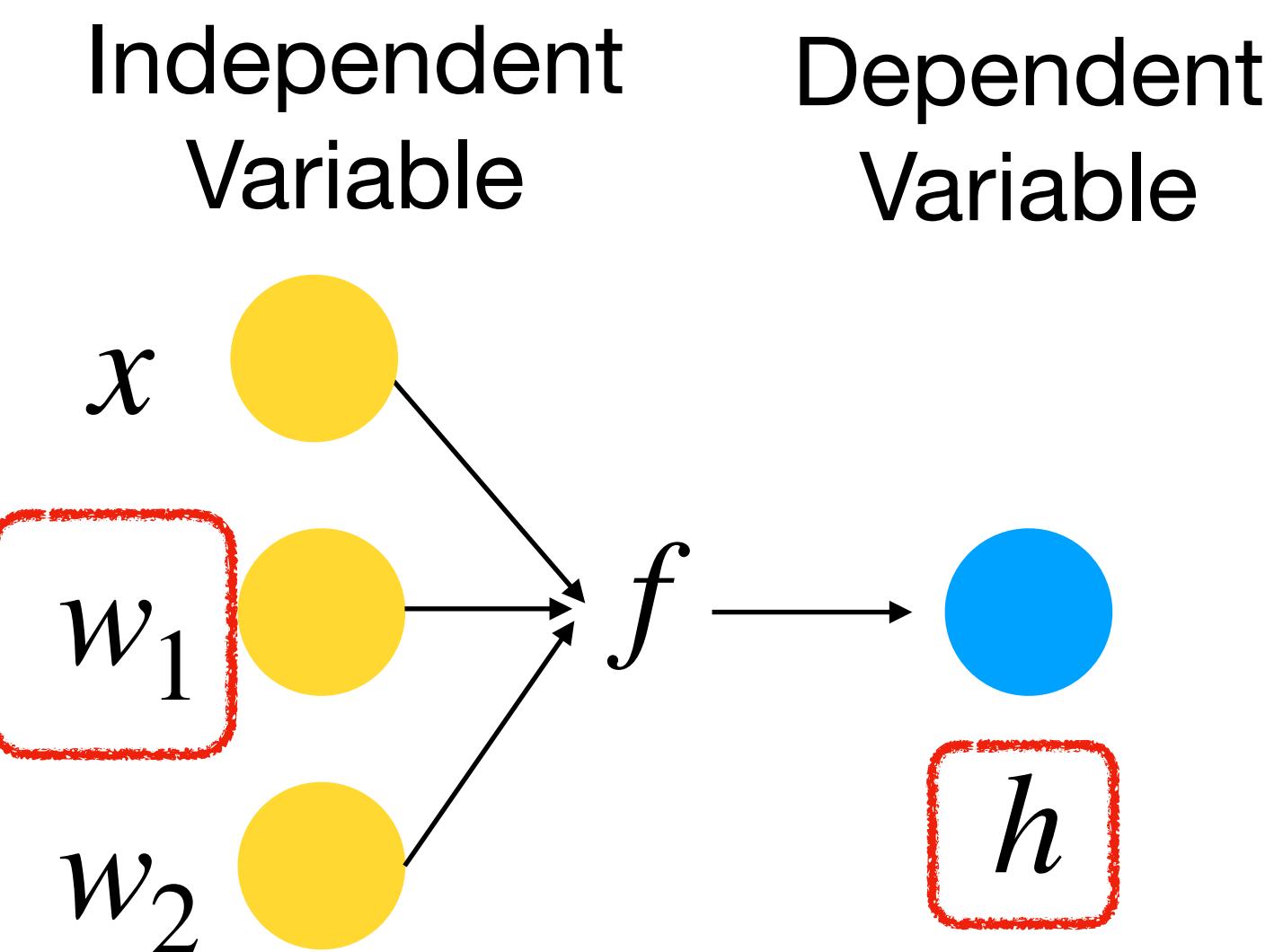
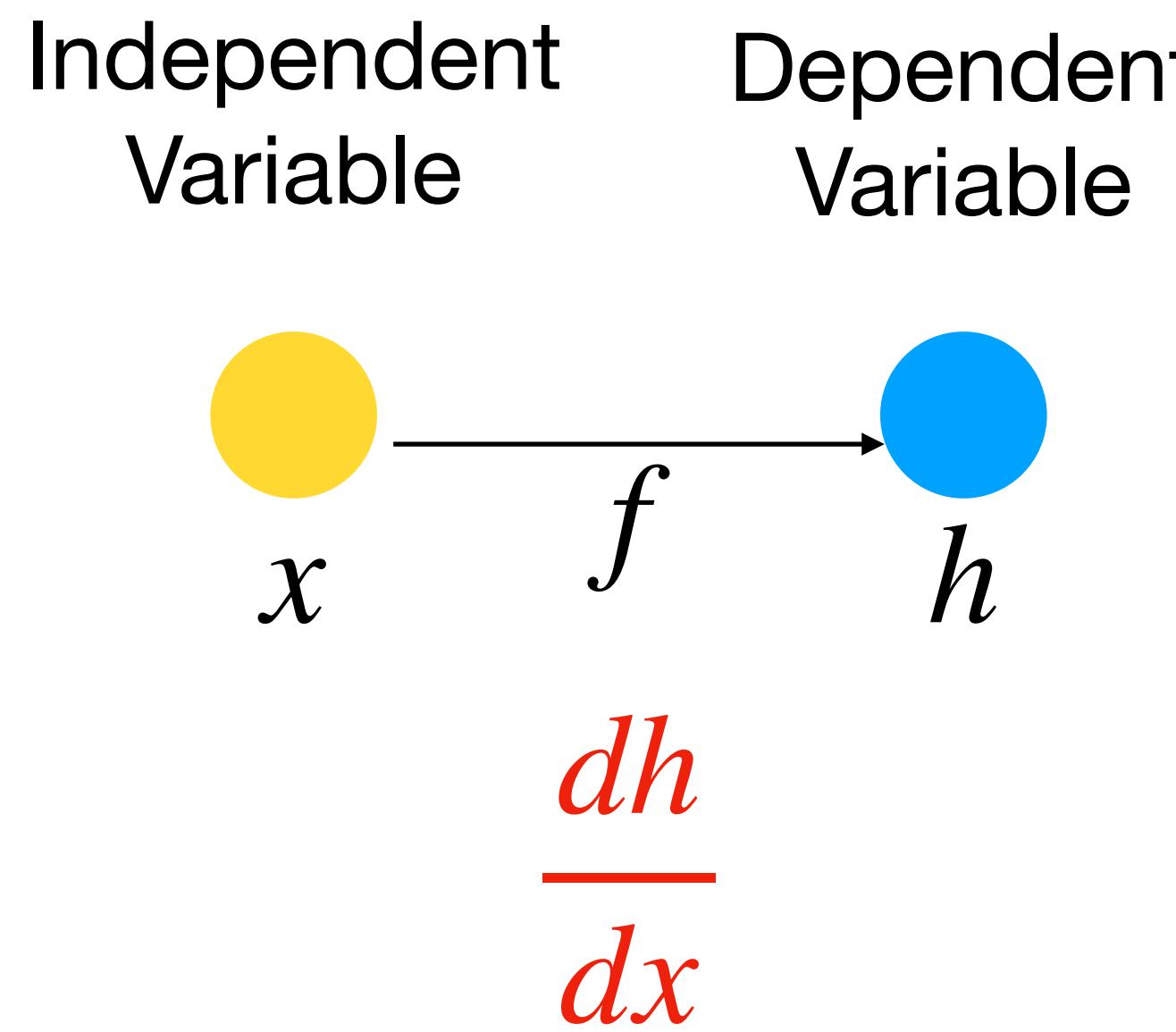
## 학습한 내용

- 편미분 (Partial Differentiation)
- 미분의 연쇄 법칙 (Chain Rule)
- Auto Differentiation과 Jacobian
- Gradient의 또다른 의미
- 경사하강이 잘 작동하기 위한 전제조건

# Section 6 요약

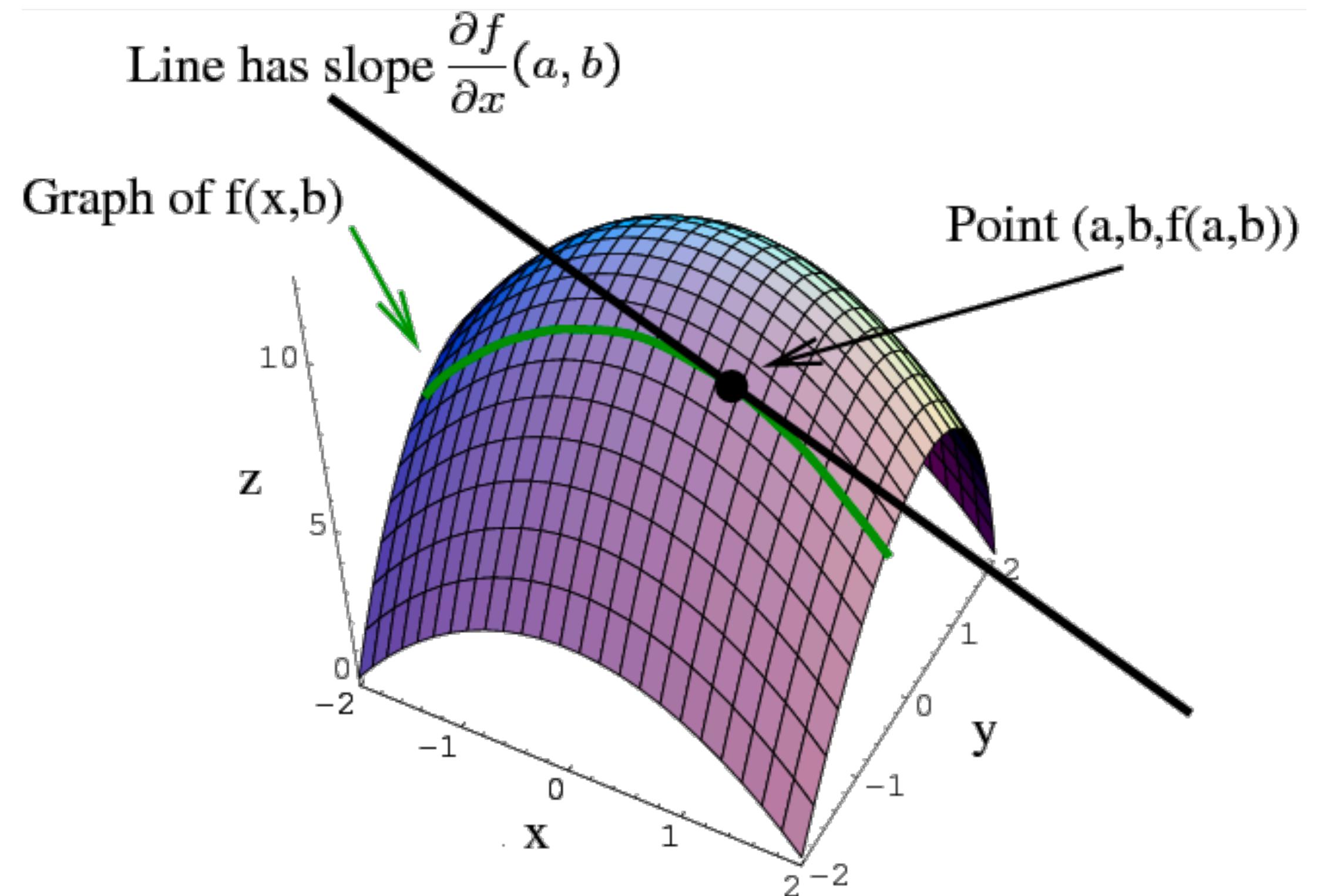
## Partial Differentiation

- Partial Differentiation은  $h = f(w_1, w_2, x)$ 에서 다른 변수들 (e.g.  $x, w_2$ )을 고정한다고 가정했을 때, 변수 (e.g.  $w_1$ )에 대한  $h$ 의 경사를 구하는 것.



# Section 6 요약

## Partial Differentiation



출처: Math Insight (Partial Derivative Limit Definition)

$$\left. \frac{\partial z}{\partial x} \right|_{y=b}$$

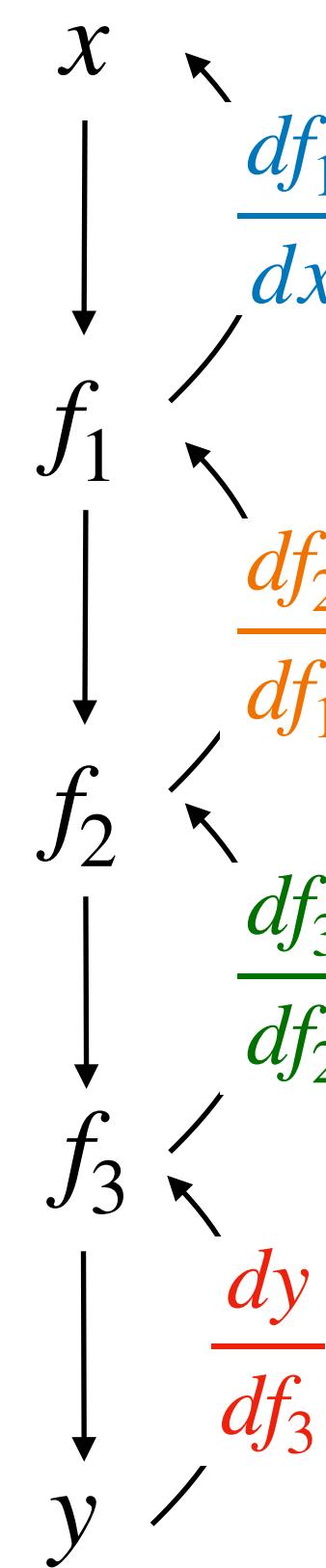
은 바로  $y = b$ 로 고정해두고 평면

$z$ 에 대해서 접선을 그렸을때, 해당 접선의 기울기이다!

# Section 6 요약

## 미분의 연쇄법칙 (Chain Rule)

Ancestor Node



$y = y(f_3(f_2(f_1(x))))$  의  $x$  에 대한 미분

$$\frac{dy}{dx} = \frac{dy}{df_3} \cdot \frac{df_3}{df_2} \cdot \frac{df_2}{df_1} \cdot \frac{df_1}{dx}$$

즉, “미분의 연쇄 법칙” 이란:

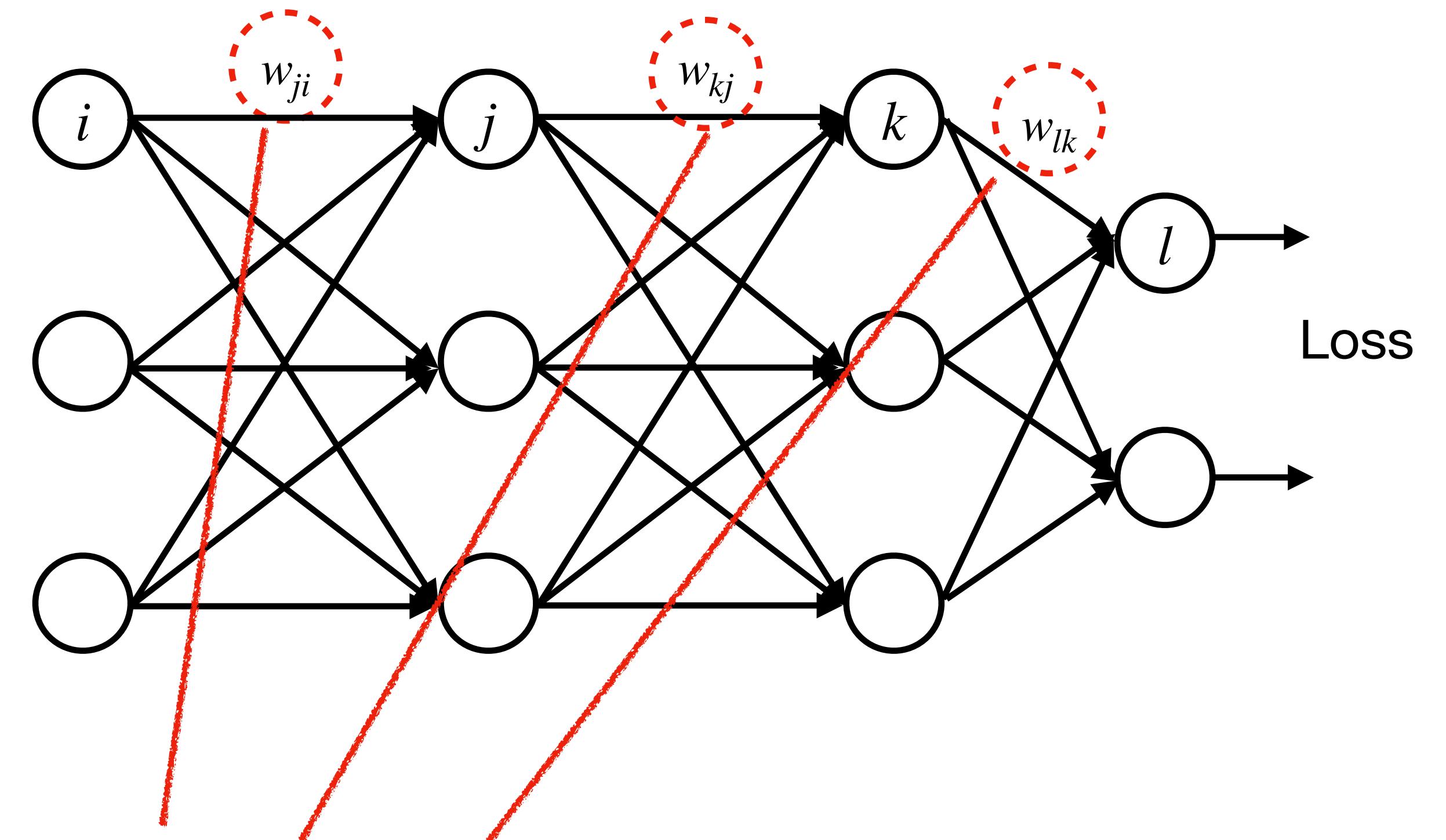
- 어떤 합성 함수 (Composite Function)의 미분값은 각 구성 함수들의 미분값들의 “연쇄적인 곱”들이다!

Descendent Node

# Section 6 요약

## Auto Differentiation

Auto Differentiation은 미분의 Chain Rule 특징을 활용해서 Computationally Efficient하게 각  $\frac{\partial L}{\partial w}$  을 구하는 방법.



$$\frac{\partial L}{\partial w_{ji}^{(1)}}, \frac{\partial L}{\partial w_{kj}^{(2)}}, \frac{\partial L}{\partial w_{lk}^{(3)}} \text{ 각각에 대해서 모두 구해야한다.}$$

# Section 6 요약

## Auto Differentiation

중간에 계산된 편미분 값들을 저장 및 재사용

→ Computational Cost 절약

→ Auto Differentiation의 핵심

Backward pass:

$$\frac{\partial L}{\partial w_{lk}^{(3)}} = \frac{\partial L}{\partial \hat{y}_l} \cdot \frac{\partial \hat{y}_l}{\partial w_{lk}^{(3)}}$$

$$\frac{\partial L}{\partial w_{kj}^{(2)}} = \sum_l \frac{\partial L}{\partial \hat{y}_l} \cdot \frac{\partial \hat{y}_l}{\partial h_k^{(2)}} \cdot \frac{\partial h_k^{(2)}}{\partial w_{kj}^{(2)}}$$

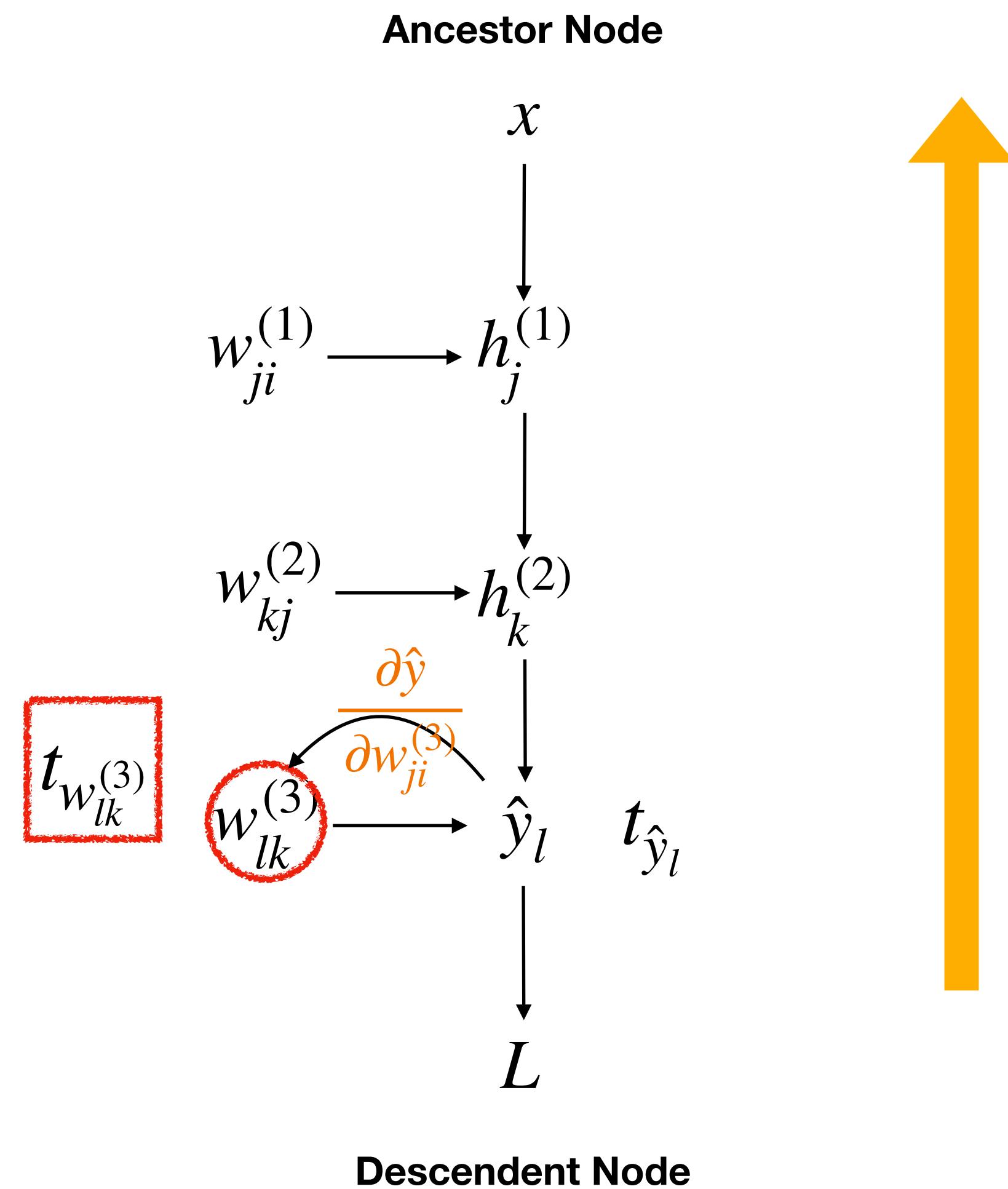
$$\frac{\partial L}{\partial w_{ji}^{(1)}} = \sum_{l,k} \frac{\partial L}{\partial \hat{y}_l} \cdot \frac{\partial \hat{y}_l}{\partial h_k^{(2)}} \cdot \frac{\partial h_k^{(2)}}{\partial h_j^{(1)}} \cdot \frac{\partial h_j^{(1)}}{\partial w_{ji}^{(1)}}$$

$$\frac{\partial L}{\partial h_k^{(2)}}$$

$$\frac{\partial L}{\partial h_j^{(1)}}$$

# Section 6 요약

## Computational Graph



Steps:

1. Descendent Node로 부터 시작 ( $t_L = 1$ )
2.  $\hat{y}_l$  노드의  $t_{\hat{y}_l}$  계산
3.  $w_{lk}^{(3)}$  노드의  $t_{w_{lk}^{(3)}}$  계산:

$$t_{w_{lk}^{(3)}} = \frac{\partial L}{\partial w_{lk}^{(3)}} = t_{\hat{y}_l} \cdot \frac{\partial \hat{y}_l}{\partial w_{lk}^{(3)}}$$

# Section 6 요약

## Multivariate Differentiation and Jacobian

Partial Derivative의 벡터 혹은 행렬으로 구성된 것을  
**“Jacobian”**이라고 한다!

$$\nabla_{W^i} L = \begin{pmatrix} \frac{\partial L}{\partial w_{11}^i} & \dots & \frac{\partial L}{\partial w_{1n}^i} & \dots & \frac{\partial L}{\partial w_{1N}^i} \\ \frac{\partial L}{\partial w_{m1}^i} & \dots & \frac{\partial L}{\partial w_{mn}^i} & \dots & \frac{\partial L}{\partial w_{mN}^i} \\ \frac{\partial L}{\partial w_{M1}^i} & \dots & \frac{\partial L}{\partial w_{Mn}^i} & \dots & \frac{\partial L}{\partial w_{MN}^i} \end{pmatrix}$$

Notation은  $\nabla_W L$  으로 표현한다!

# Section 6 요약

## Multivariate Differentiation and Jacobian

$W \in \mathbb{R}^{M \times N}$  가 matrix일 때, matrix notation으로는:

$$W^{i+1} = W^i - \lambda \cdot \nabla_{W^i} L$$

$$\nabla_{W^i} L$$

$M \times N$  matrix

column = N개

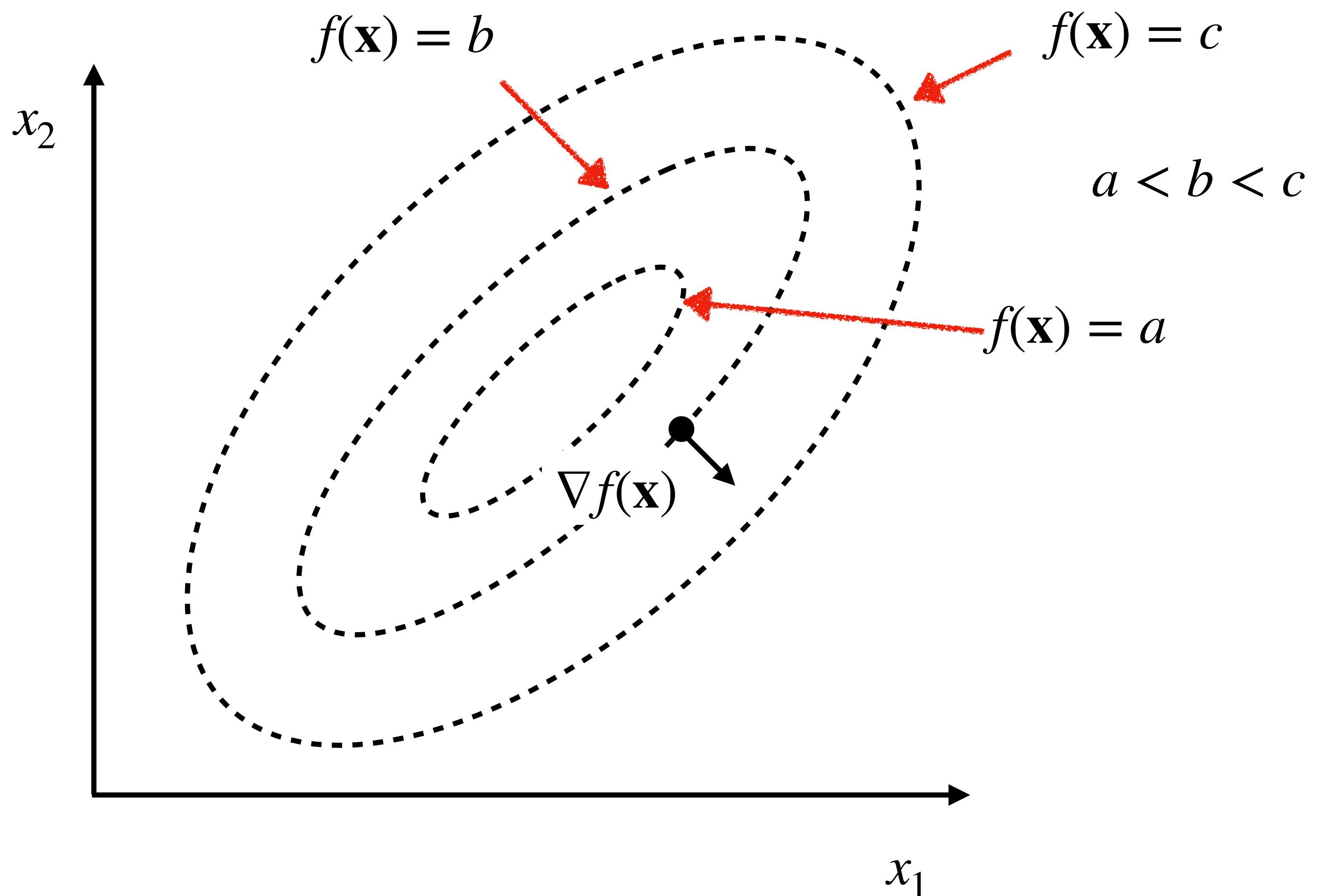
$$\text{row } M\text{개} \quad \begin{pmatrix} w_{11}^{i+1} & \dots & w_{1n}^{i+1} & \dots & w_{0N}^{i+1} \\ w_{m1}^{i+1} & \dots & w_{mn}^{i+1} & \dots & w_{mN}^{i+1} \\ w_{M1}^{i+1} & \dots & w_{Mn}^{i+1} & \dots & w_{MN}^{i+1} \end{pmatrix} = W^{i+1}$$

$$= \begin{pmatrix} w_{11}^i & \dots & w_{1n}^i & \dots & w_{0N}^i \\ w_{m1}^i & \dots & w_{mn}^i & \dots & w_{mN}^i \\ w_{M1}^i & \dots & w_{Mn}^i & \dots & w_{MN}^i \end{pmatrix} = W^i$$

$$- \lambda \begin{pmatrix} \frac{\partial L}{\partial w_{11}^i} & \dots & \frac{\partial L}{\partial w_{1n}^i} & \dots & \frac{\partial L}{\partial w_{1N}^i} \\ \frac{\partial L}{\partial w_{m1}^i} & \dots & \frac{\partial L}{\partial w_{mn}^i} & \dots & \frac{\partial L}{\partial w_{mN}^i} \\ \frac{\partial L}{\partial w_{M1}^i} & \dots & \frac{\partial L}{\partial w_{Mn}^i} & \dots & \frac{\partial L}{\partial w_{MN}^i} \end{pmatrix}$$

# Section 6 요약

## Gradient의 또다른 의미



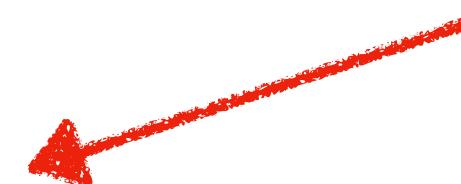
- 좌표  $\mathbf{x}$ 에서  $\nabla_{\mathbf{x}} f(\mathbf{x})$ 은 함수의 최대 증가폭의 방향을 향해 있다.
- 즉,  $\nabla_{\mathbf{x}} f(\mathbf{x})$ 가 가르키는 방향이 함수가 가장 가파르게 상승하는 방향이다.

# Section 6 요약

## 경사 하강이 잘 작동하기 위한 전제 조건

- 경사하강을 통해서 Loss를 줄여나갈 수 있다.
- 하지만 경사하강을 통해서 수렴된 지점이 과연 Global Minimum일까?
- Global Minimum을 찾기 위한 전제조건은?

Weight space 상에서 Loss가 가장 낮은 지점



바로 Convexity!

즉, 아래로 볼록한 조건을 만족시켜야 한다!

# Section 6 요약

## 경사 하강이 잘 작동하기 위한 전제 조건

Copyright©2023. Acadential. All rights reserved.

Non-convex한 경우 경사하강은 **Local minimum** 혹은 **saddle point**에 수렴해버릴 수 있다.

이것을 어떻게 해결할 수 있을까? → **Momentum (관성)**, **Mini-batch Stochastic Gradient Descent**을 사용하는 것이 도움이 된다. (추후에 다룰 예정!)

