ACADENTIAL

6-2. Partial Differentiation (편미분)

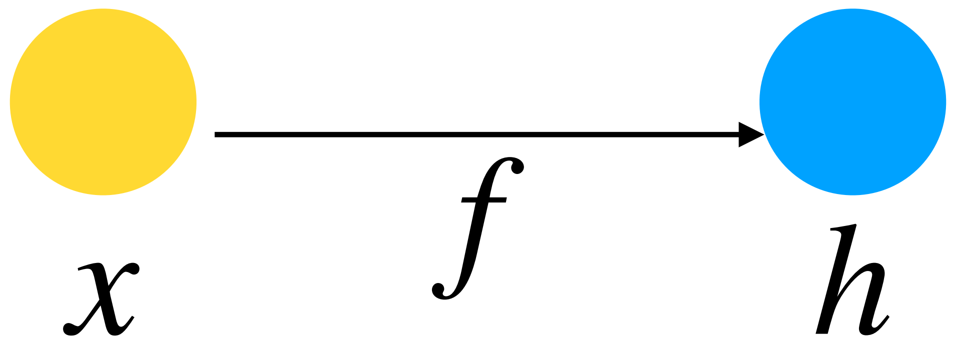
Partial Differentiation

- 본격적으로 Reverse Differentiation에 대해서 살펴보기 전에
- Partial Differentiation에 대해서 이해해야 한다.

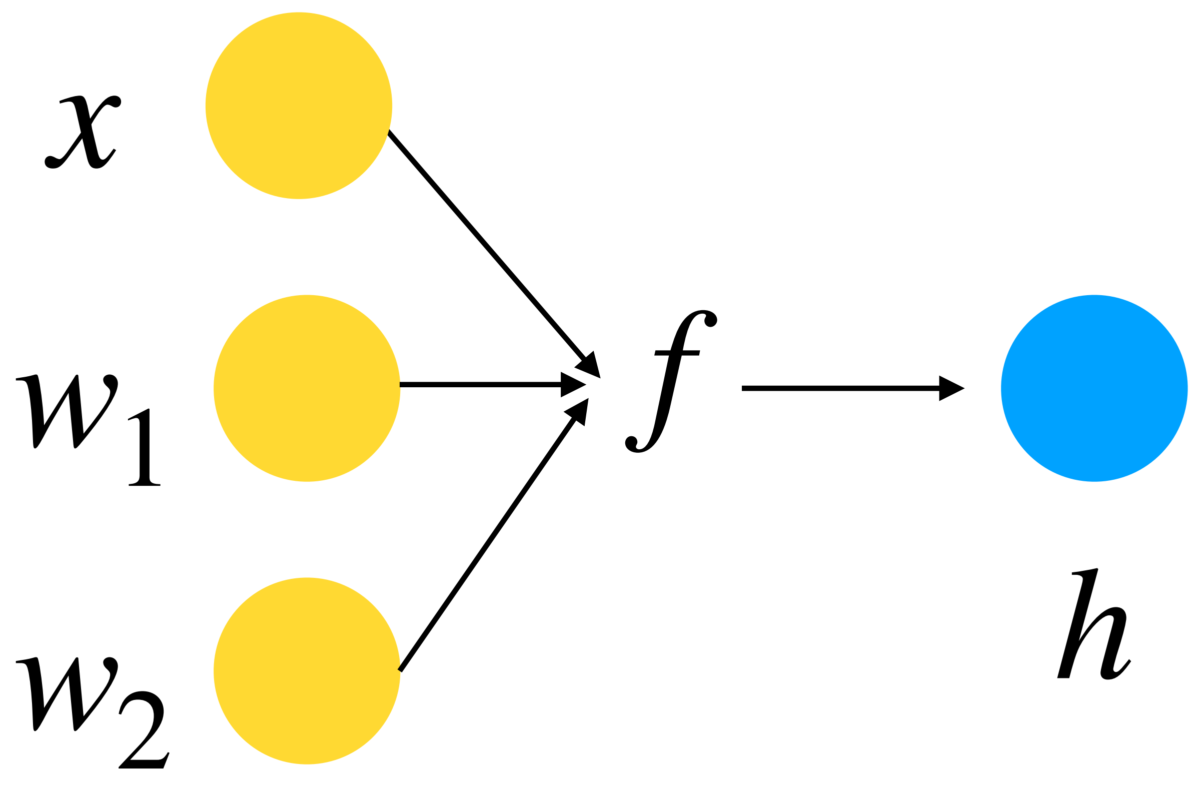
Partial Differentiation

- Partial Differentiation은 $h = f(w_1, w_2, x)$ 와 같이 multi-variate한 경우에 대해서 gradient을 구하는 것이다.

Independent Variable   Dependent Variable

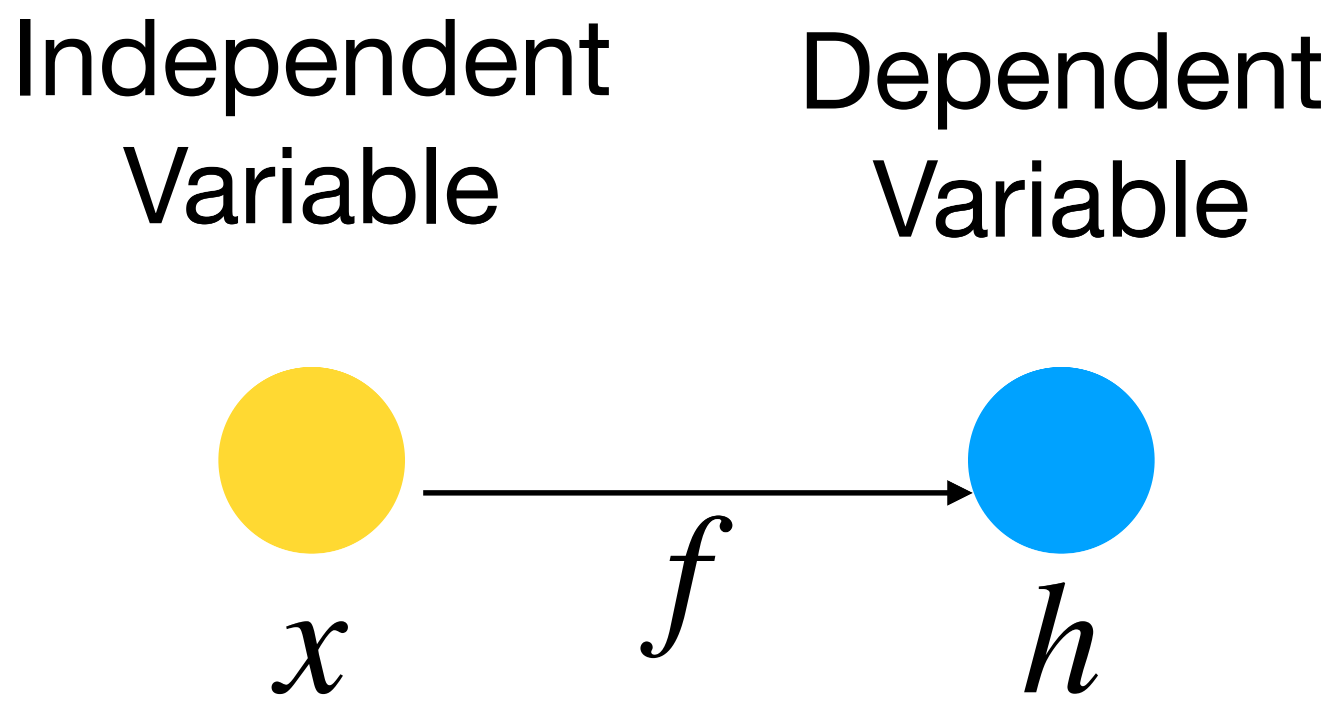


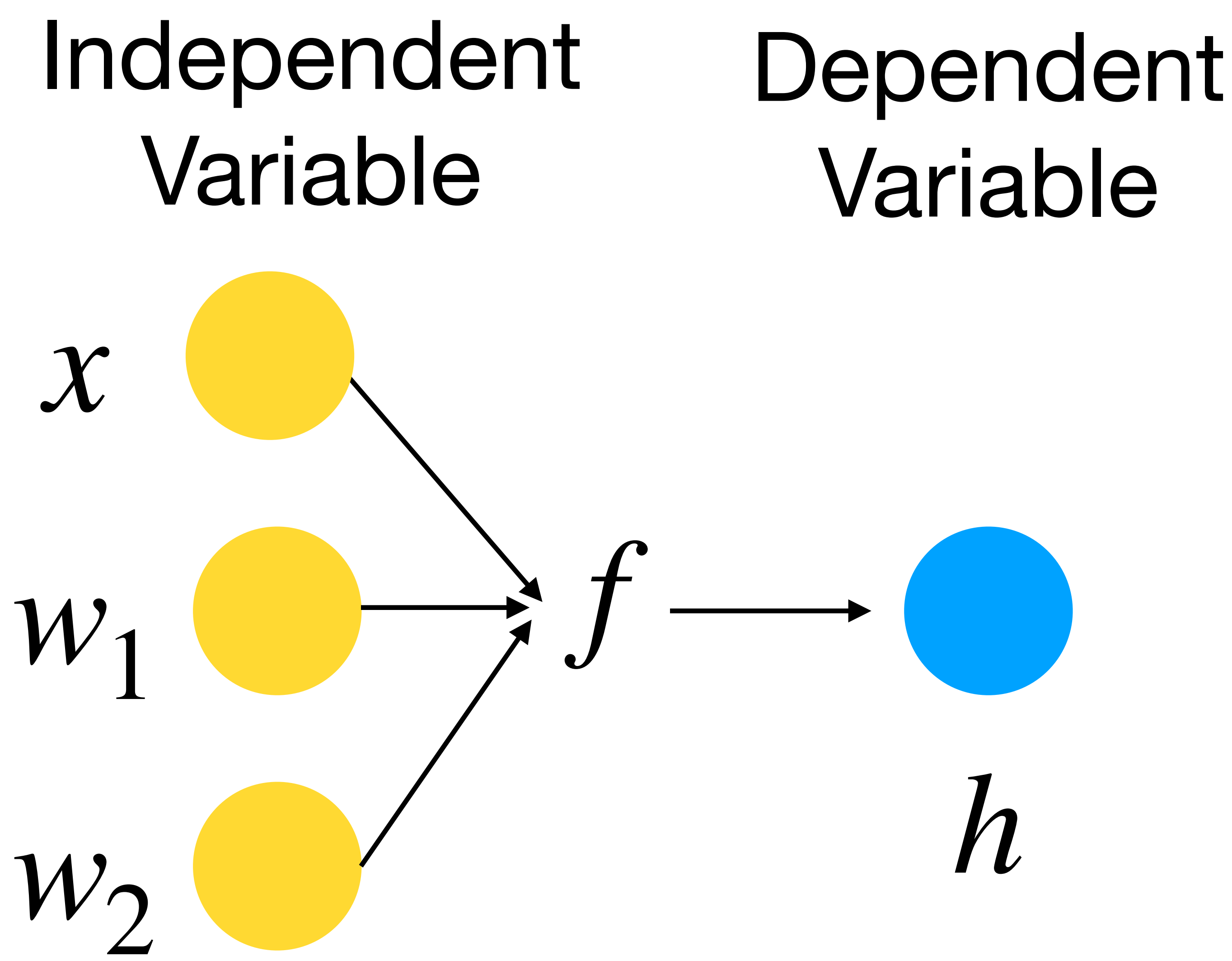
Independent Variable   Dependent Variable



Partial Differentiation

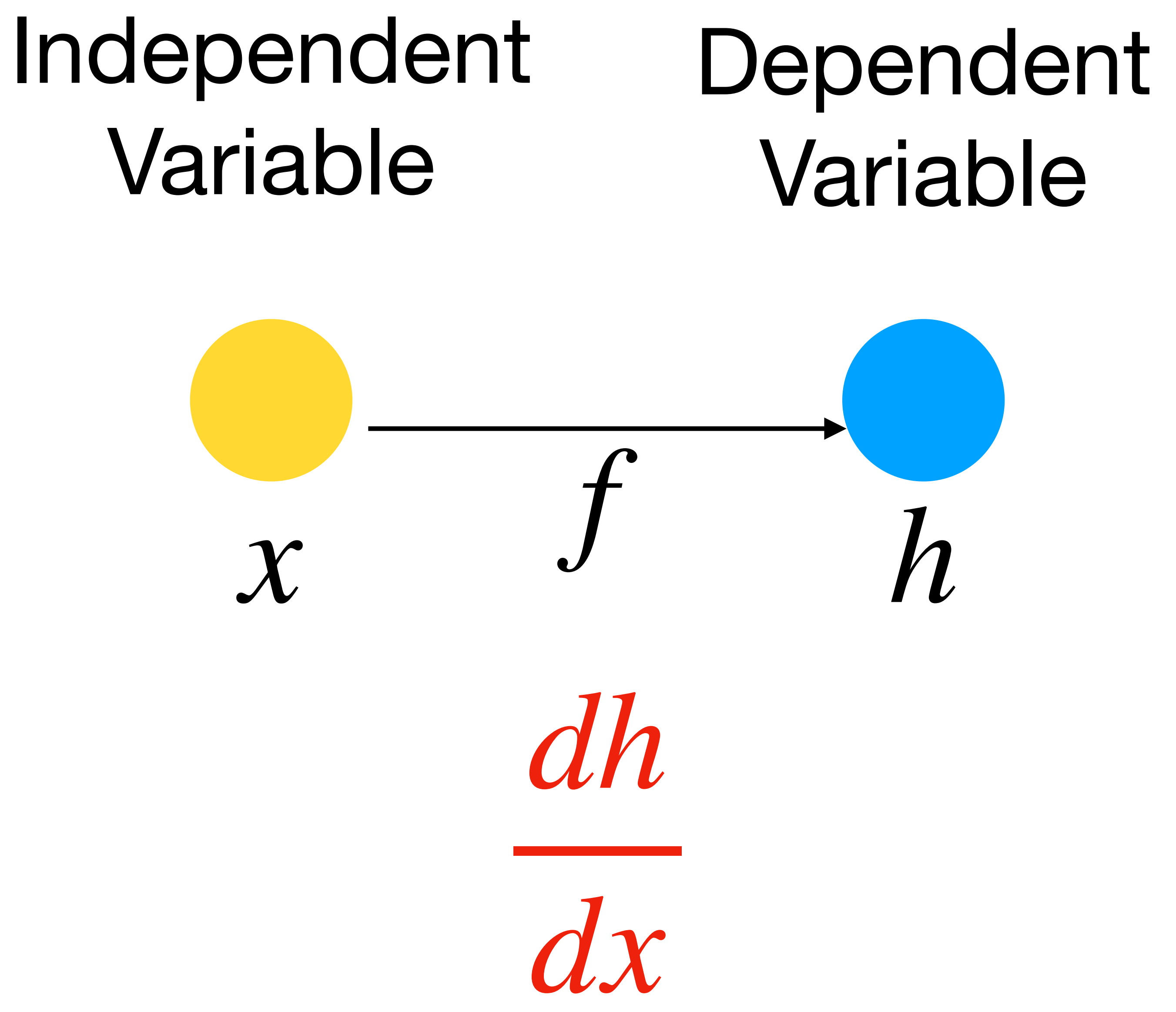
- (약간 더 엄밀히 말하자면) Partial Differentiation은 $h = f(w_1, w_2, x)$에서 다른 변수들 (e.g. x, w_2)을 고정한다고 가정했을때, 변수 (e.g. w_1)에 대한 h의 경사을 구하는 것.

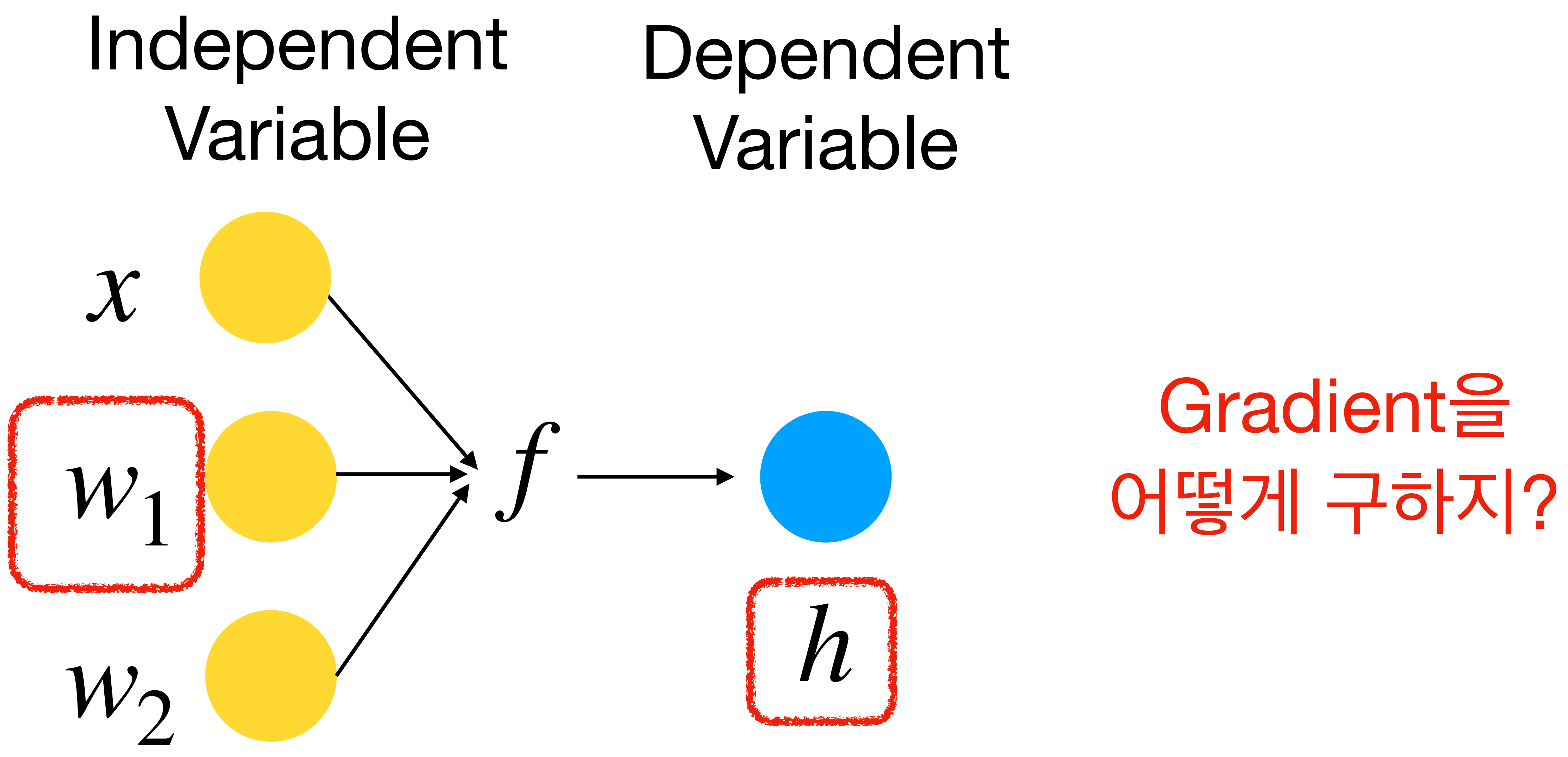




Partial Differentiation

- (약간 더 엄밀히 말하자면) Partial Differentiation은 $h = f(w_1, w_2, x)$에서 다른 변수들 (e.g. x, w_2)을 고정한다고 가정했을때, 변수 (e.g. w_1)에 대한 h의 경사을 구하는 것.





Partial Differentiation

- Partial Differentiation은 변수가 하나 이상인 다변수 함수 $f = f(x_1, x_2, \cdots x_N)$에 대해서 사용한다.
- Partial Differentiation은 변수들 중 하나만을 기준으로 잡고 다른 변수들은 상수로 간주해서 미분하는 것이다.

Partial Differentiation

예시:

- $f(x_1, x_2) = 5x_1 + 3x_2$ 이면,

- $f(x_1, x_2) = x_1^4 x_2 + \sin(x_1) + \log(x_2)$ 이면,

Partial Differentiation

예시:

- $f(x_1, x_2) = 5x_1 + 3x_2$ 이면,
  - $\left.\dfrac{\partial f}{\partial x_1}\right|_{x_2} = 5$

- $f(x_1, x_2) = x_1^4 x_2 + \sin(x_1) + \log(x_2)$ 이면,

Partial Differentiation

예시:

- $f(x_1, x_2) = 5x_1 + 3x_2$ 이면,
  - $\left.\dfrac{\partial f}{\partial x_1}\right|_{x_2} = 5$

- $f(x_1, x_2) = x_1^4 x_2 + \sin(x_1) + \log(x_2)$ 이면,

$$\left.\frac{\partial f}{\partial x_1}\right|_{x_2}$$

짝대기 "|"은 상수로 취해주는 변수를 표시함!

Partial Differentiation

예시:

- $f(x_1, x_2) = 5x_1 + 3x_2$ 이면,
    - $\left.\dfrac{\partial f}{\partial x_1}\right|_{x_2} = 5$
    - $\left.\dfrac{\partial f}{\partial x_2}\right|_{x_1} = ?$
- $f(x_1, x_2) = x_1^4 x_2 + \sin(x_1) + \log(x_2)$ 이면,
    - $\left.\dfrac{\partial f}{\partial x_1}\right|_{x_2} = ?$
    - $\left.\dfrac{\partial f}{\partial x_2}\right|_{x_1} = ?$

$$\left.\frac{\partial f}{\partial x_1}\right|_{x_2}$$

짝대기 “|”은 상수로 취해주는 변수를 표시함!

Partial Differentiation

예시:

- $f(x_1, x_2) = 5x_1 + 3x_2$ 이면,
    - $\left.\dfrac{\partial f}{\partial x_1}\right|_{x_2} = 5$
    - $\left.\dfrac{\partial f}{\partial x_2}\right|_{x_1} = 3$
- $f(x_1, x_2) = x_1^4 x_2 + \sin(x_1) + \log(x_2)$ 이면,
    - $\left.\dfrac{\partial f}{\partial x_1}\right|_{x_2} = 4x_1^3 x_2 + \cos(x_1)$
    - $\left.\dfrac{\partial f}{\partial x_2}\right|_{x_1} = x_1^4 + \dfrac{1}{x_2}$

$$\left.\frac{\partial f}{\partial x_1}\right|_{x_2}$$

짝대기 "|"은 상수로 취해주는 변수를 표시함!

Partial Differentiation

예시:

- $f(x_1, x_2) = 5x_1 + 3x_2$ 이면,
  - $\left.\dfrac{\partial f}{\partial x_1}\right|_{x_2} = 5$
  - $\left.\dfrac{\partial f}{\partial x_2}\right|_{x_1} = 3$
- $f(x_1, x_2) = x_1^4 x_2 + \sin(x_1) + \log(x_2)$ 이면,
  - $\left.\dfrac{\partial f}{\partial x_1}\right|_{x_2} = 4x_1^3 x_2 + \cos(x_1)$
  - $\left.\dfrac{\partial f}{\partial x_2}\right|_{x_1} = x_1^4 + \dfrac{1}{x_2}$

Partial Derivative을 취할때는 다른 변수들은 상수로 취급하고 편미분을 취하는 변수에 대해서만 미분해주면 된다!

$$\left.\frac{\partial f}{\partial x_1}\right|_{x_2}$$

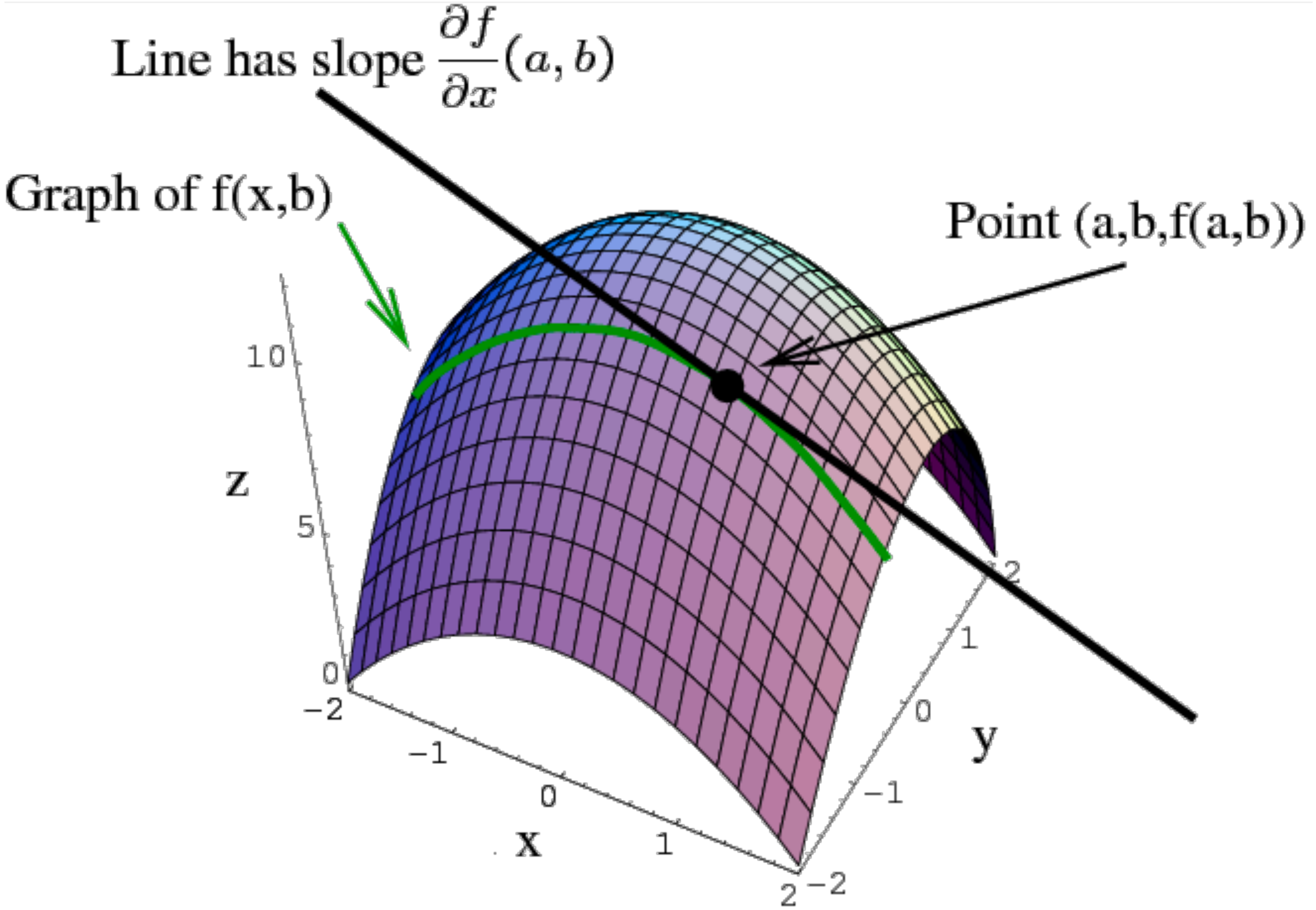
짝대기 "|"은 상수로 취해주는 변수를 표시함!

Partial Derivative의 의미를 visualization을 통해서 살펴보자!

ACADENTIAL

Partial Differentiation

Partial Derivative Visualization



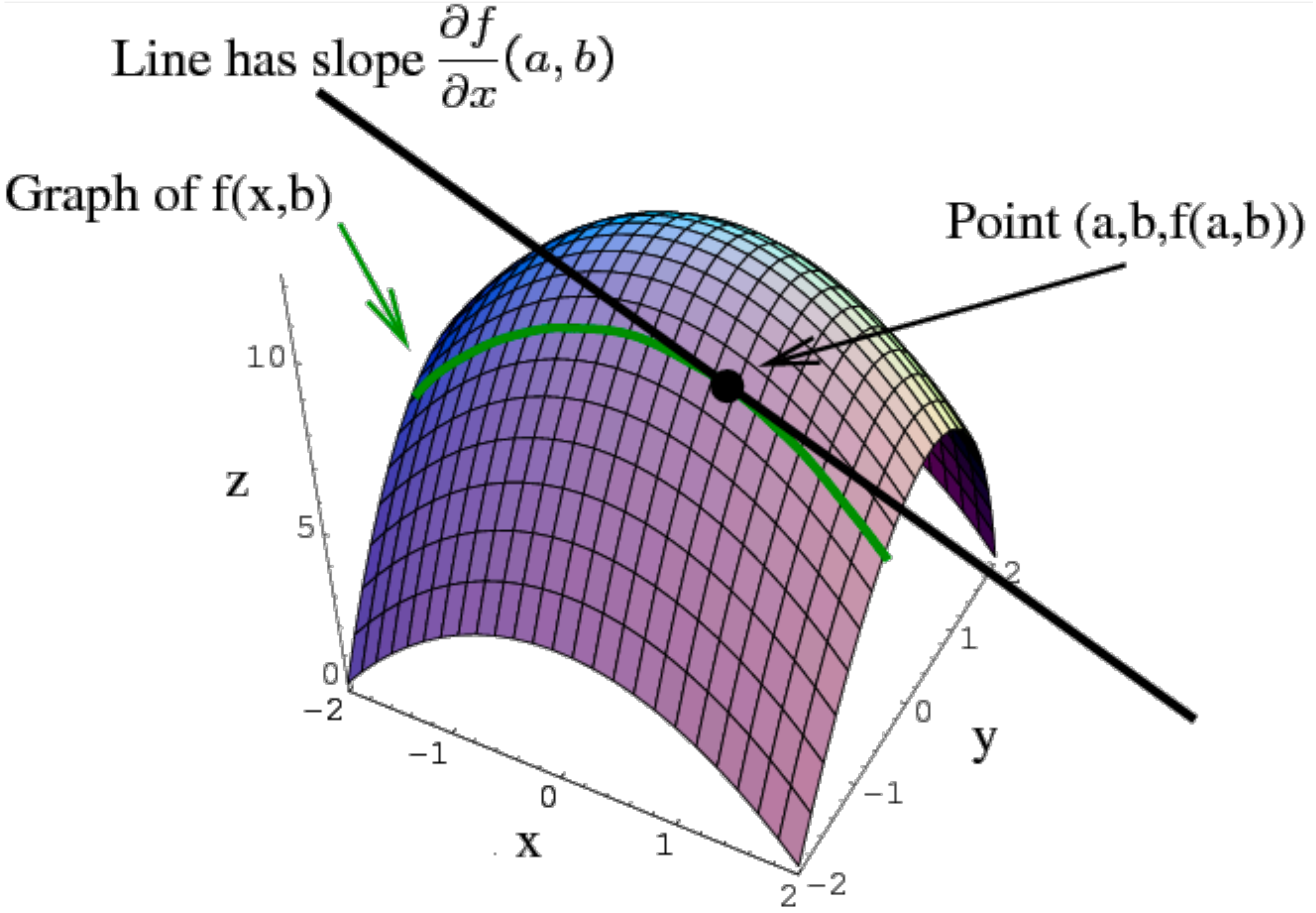
- 독립 변수 x, y (independent variables)
- 종속 변수 z (dependent variable)
- surface $z = f(x, y)$

그렇다면 해당 도표에서 $\left.\frac{\partial z}{\partial x}\right|_{y}$ 의미는 무엇일까?

출처: Math Insight (Partial Derivative Limit Definition)

Partial Differentiation

Partial Derivative Visualization



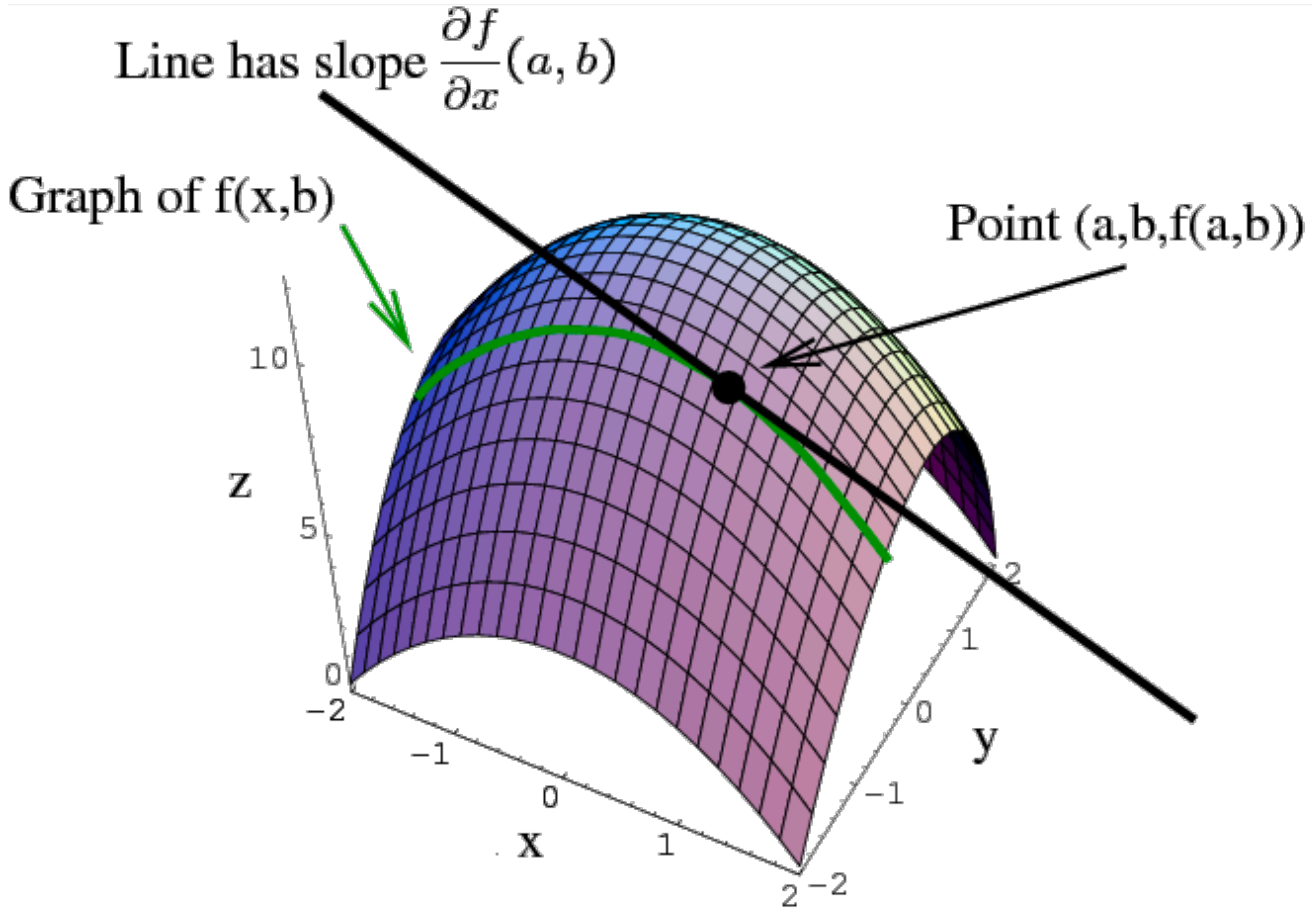
출처: Math Insight (Partial Derivative Limit Definition)

$\left.\frac{\partial z}{\partial x}\right|_{y=b}$ 은 바로 $y = b$로 고정해두고 평면 z에 대해서 접선을 그렸을때, 해당 접선의 기울기이다!

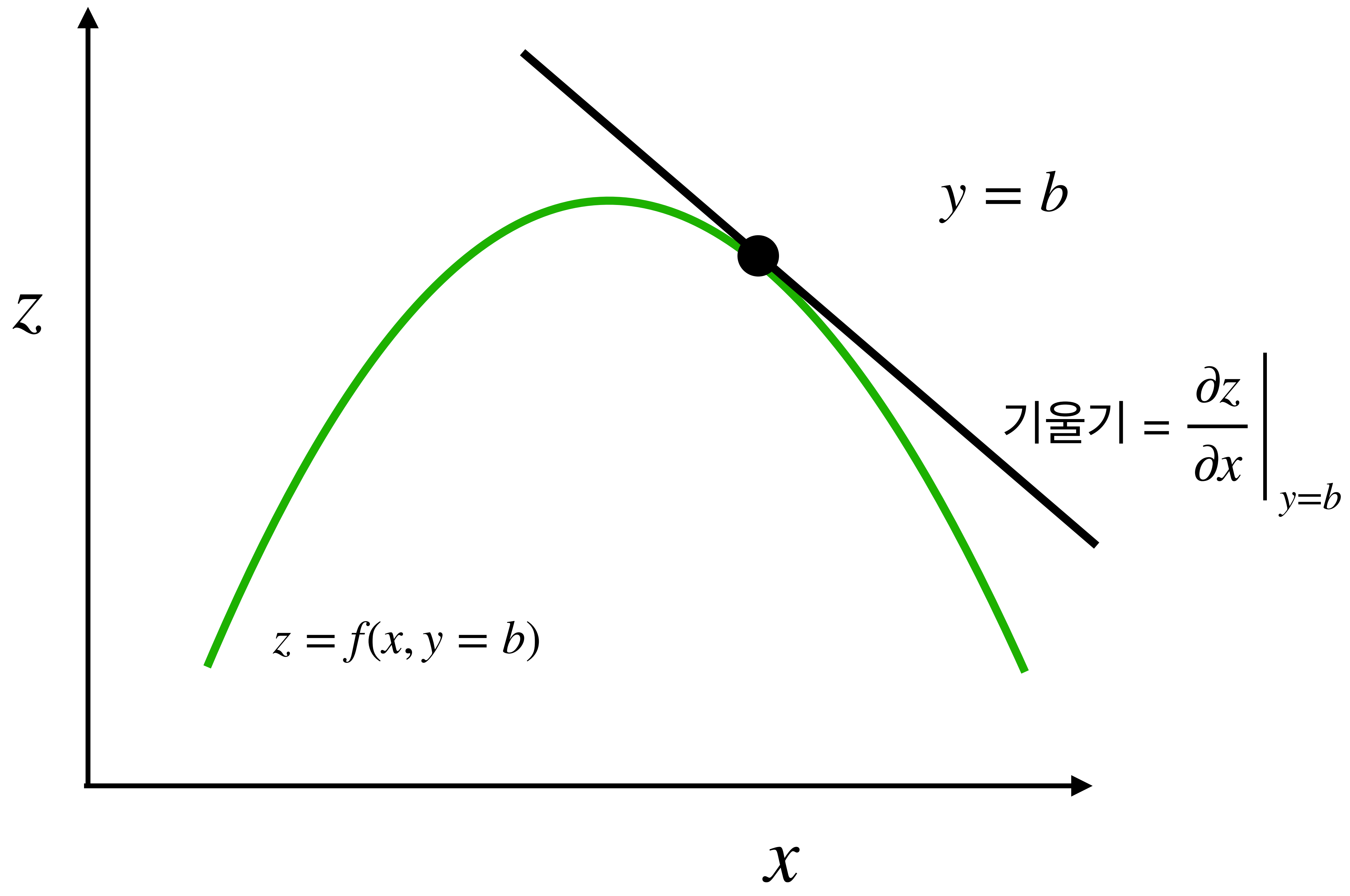
ACADENTIAL

Partial Differentiation

Partial Derivate Visualization



$y = b$의 단면을 따라 2차원 상에서 visualize 했을때



출처: Math Insight (Partial Derivative Limit Definition)

ACADENTIAL

6-3. 미분의 연쇄 법칙 (Chain Rule)

미분의 연쇄 법칙 (Chain Rule)

합성함수 $y = y(f(x))$ 을 예로 살펴보자.

미분의 연쇄법칙 (chain rule)에 따라 합성함수 $y = y(f(x))$을 x에 대해 미분하면:

$$\frac{dy}{dx} = \frac{dy}{df} \cdot \frac{df}{dx}$$

다른 예시도 한번 살펴보자.

미분의 연쇄 법칙 (Chain Rule)

$y = y(f_3(f_2(f_1(x))))$

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

미분의 연쇄 법칙 (Chain Rule)

Computational Graph

ACADENTIAL

$y = y(f_3(f_2(f_1(x))))$

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

한 눈에 보기 어렵다…

합성 함수를 Computational Graph로 표현하고, 미분의 연쇄법칙이 어떻게 Computational Graph 상에서 구현되는지 살펴보자!

참고로, Reverse Differentiation은 Computational Graph을 기반으로 구현됨!

미분의 연쇄 법칙 (Chain Rule)

Computational Graph

Ancestor Node

$$\begin{array}{c} x \\ \downarrow \\ f_1 \\ \downarrow \\ f_2 \\ \downarrow \\ f_3 \\ \downarrow \\ y \end{array}$$

Descendent Node

$$y = y(f_3(f_2(f_1(x))))$$

Computation Graph:

- **Node (노드)**: Operation 혹은 Variable에 해당
- **Edge (에지)**: input variable → target variable로 잇는 edge.
- **Ancestor Node**: Computational Graph의 시작점
- **Descendent Node**: Computational Graph의 끝점

미분의 연쇄 법칙 (Chain Rule)

Computational Graph

Ancestor Node

$$
\begin{array}{c}
x \\
\downarrow \\
f_1 \\
\downarrow \\
f_2 \\
\downarrow \\
f_3 \\
\downarrow \\
y
\end{array}
$$

Descendent Node

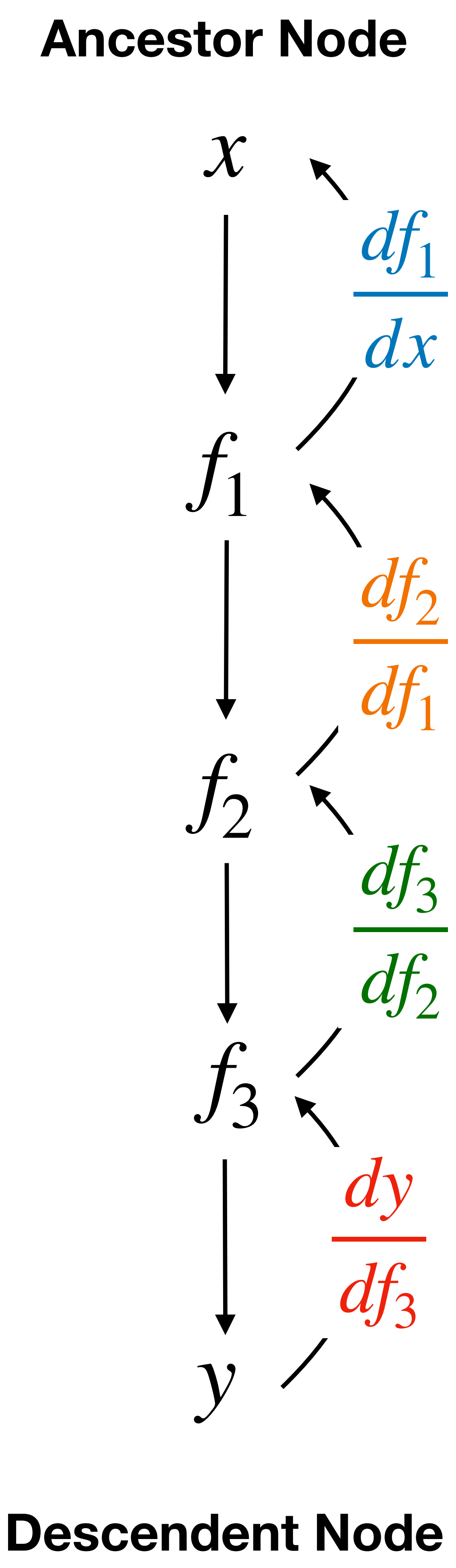
$y = y(f_3(f_2(f_1(x))))$ 의 x 에 대한 미분

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

미분값은 Computational Graph 상에서 어떻게 구할까?

미분의 연쇄 법칙 (Chain Rule)

Computational Graph

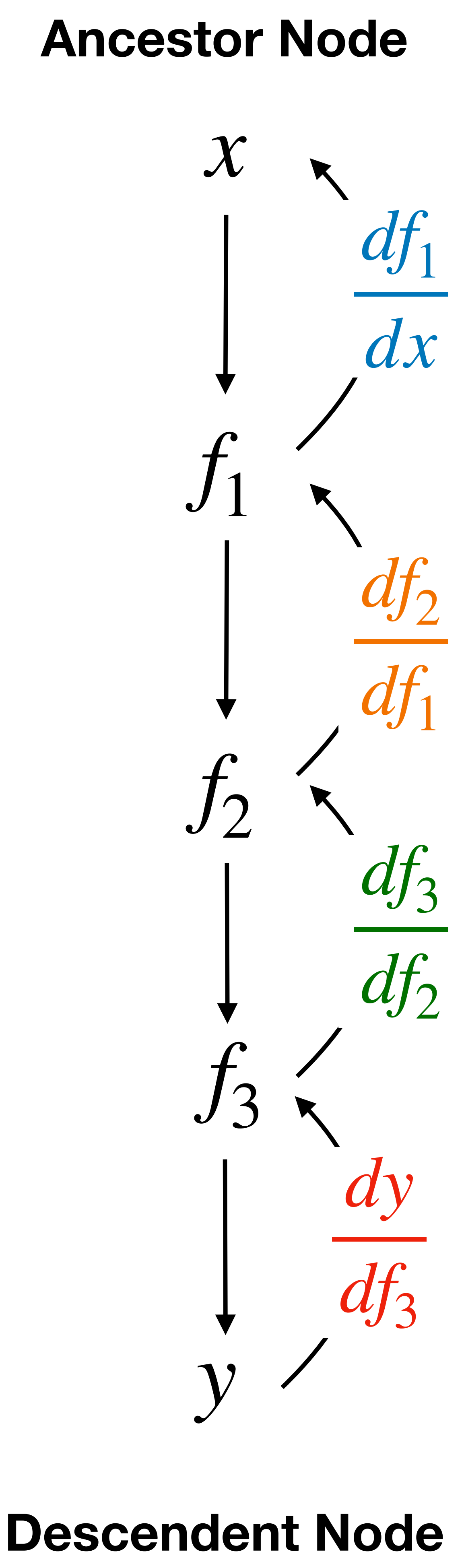


$y = y(f_3(f_2(f_1(x))))$ 의 x 에 대한 미분

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

미분의 연쇄 법칙 (Chain Rule)

Reverse Differentiation



$y = y(f_3(f_2(f_1(x))))$ 의 x 에 대한 미분

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

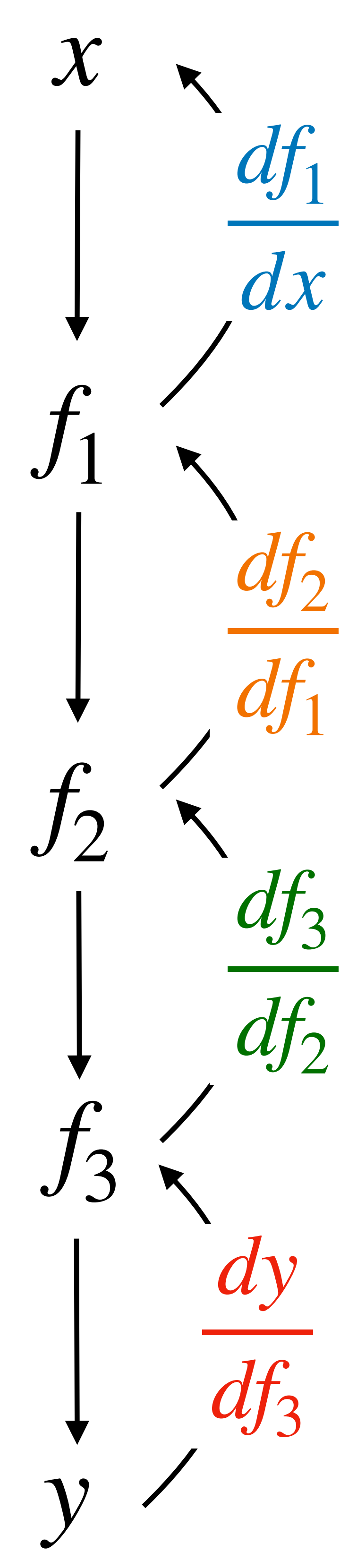
즉, “미분의 연쇄 법칙” 이란:

- 어떤 합성 함수 (Composite Function)의 미분값은 각 구성 함수들의 미분값들의 “연쇄적인 곱”들이다!
- Computational Graph을 거꾸로 타고 올라가면서 구한 미분값들을 곱해주는 형태로 구현
- Reverse Differentiation의 원리!

미분의 연쇄 법칙 (Chain Rule)

Reverse Differentiation

Ancestor Node



Descendent Node

$y = y(f_3(f_2(f_1(x))))$ 의 x 에 대한 미분

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

핵심:

- Computational Graph로 합성 함수 표현 및 미분 가능함.
- Reverse Differentiation은 미분의 연쇄 법칙에 기반됨.

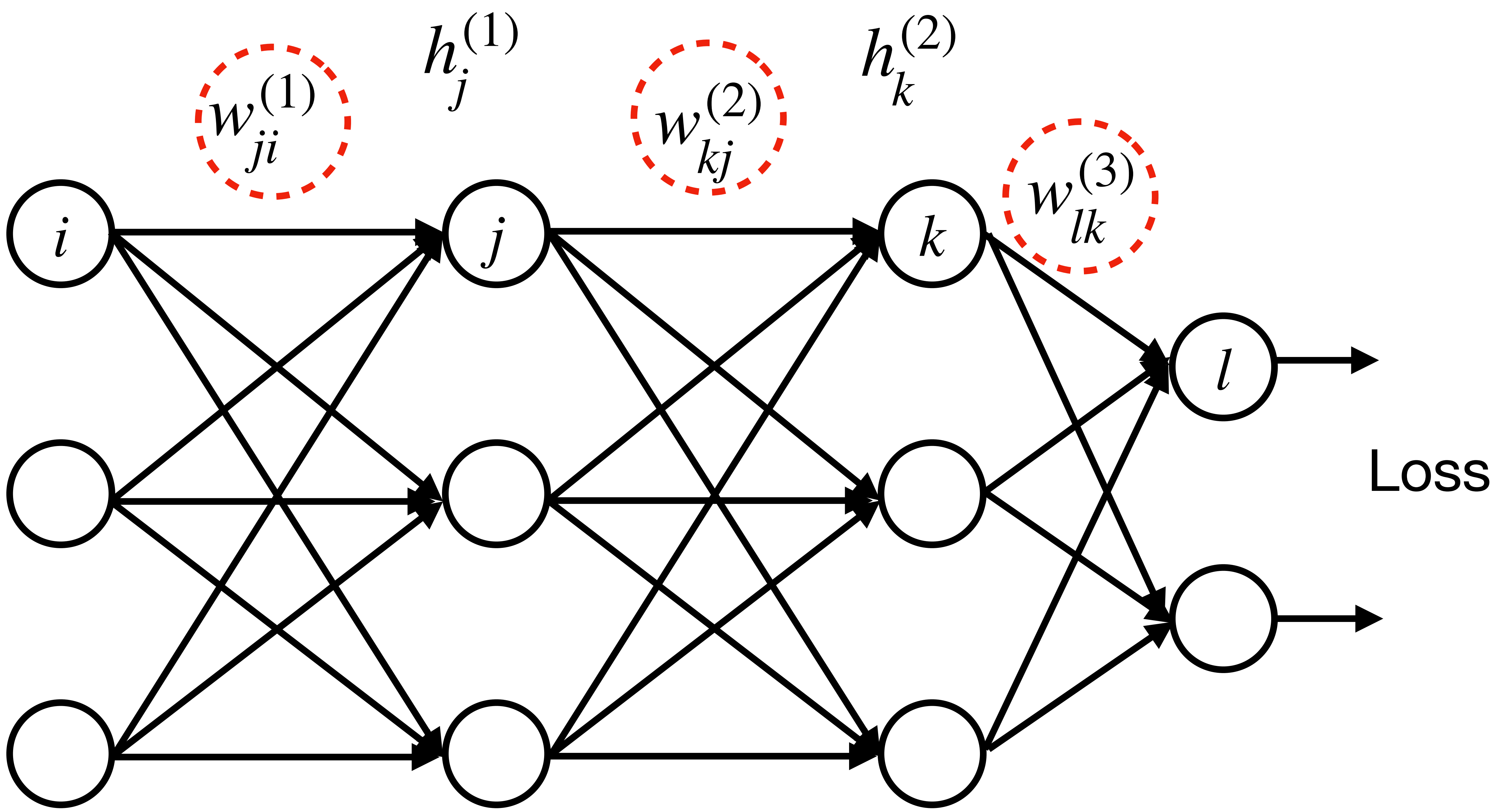
6-4. Auto Differentiation (Reverse Differentiation)

Auto Differentiation의 목적

Auto Differentiation

Auto Differentiation의 목적

- **Gradient Descent**을 수행하기 위해서는
- **Loss**을 뉴럴넷 모델의 **weight parameter**들 **에 대해서 미분한 값**이 필요하다.

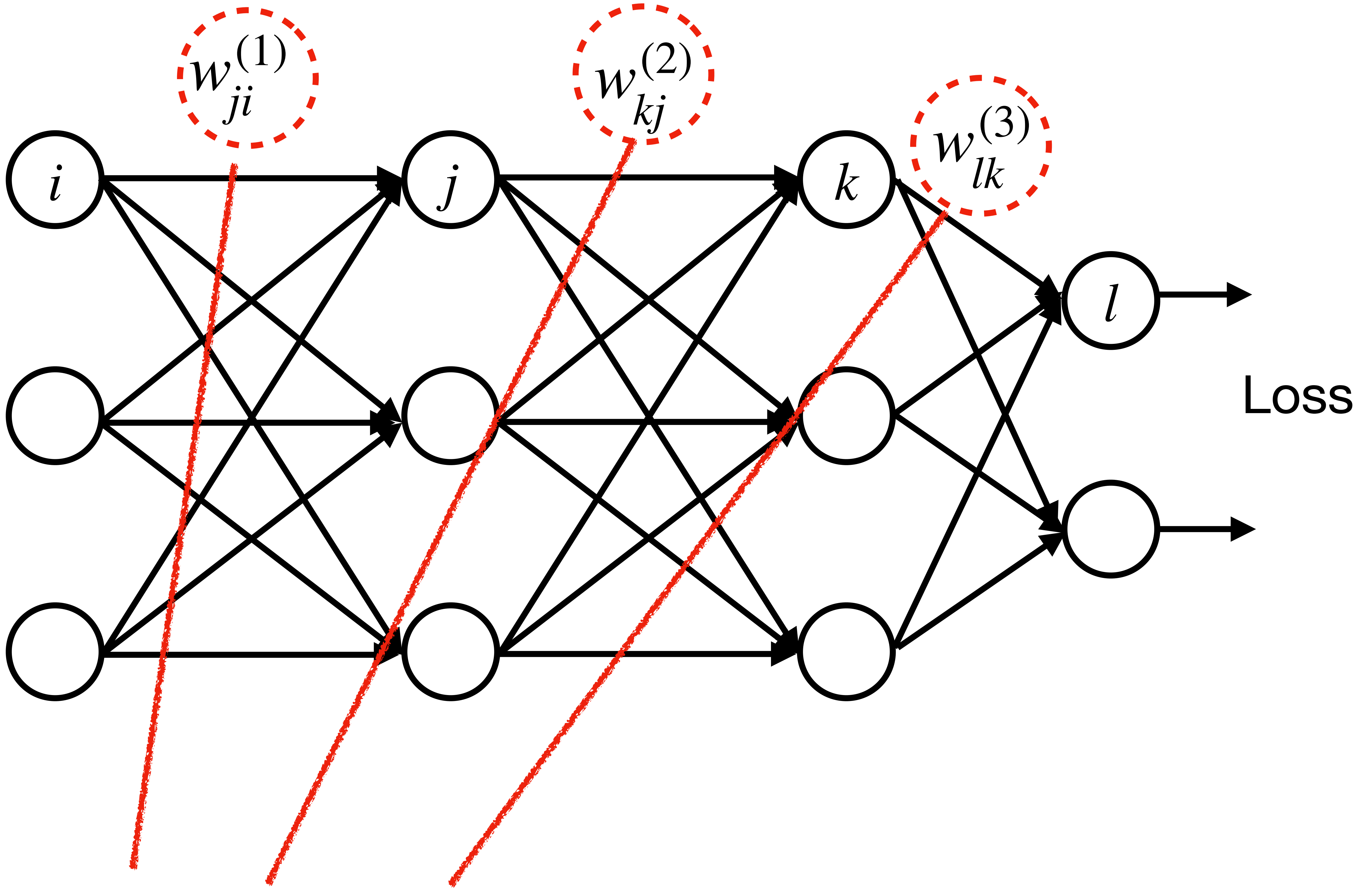


ACADENTIAL

Auto Differentiation

Auto Differentiation의 목적

- **Gradient Descent**을 수행하기 위해서는
- Loss을 뉴럴넷 모델의 weight parameter들에 대해서 미분한 값이 필요하다.
- 즉, $\frac{\partial L}{\partial w}$을
    - 뉴럴넷을 구성하는 각 weight parameter들에 대해서 구해야한다.

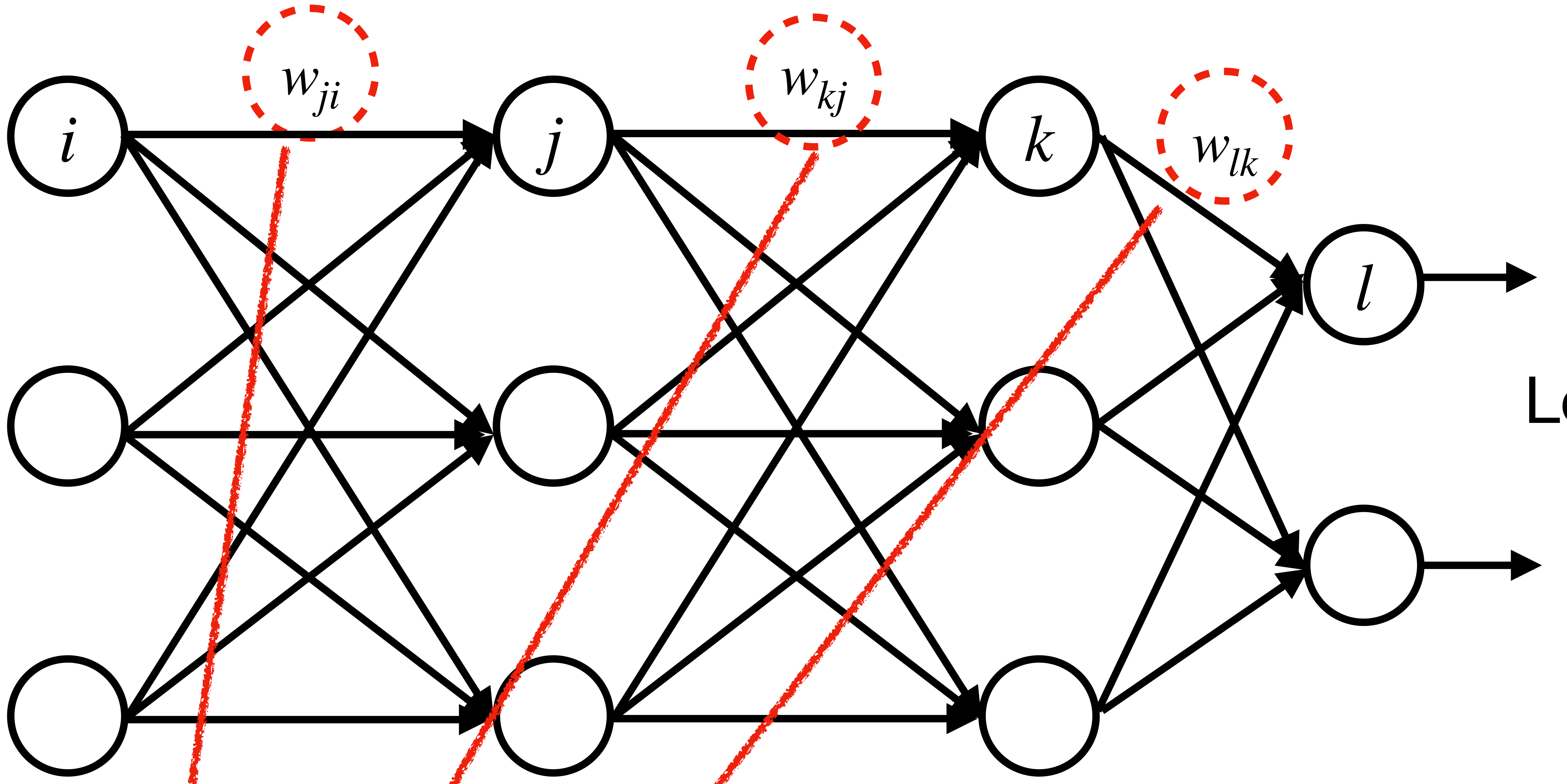


$\frac{\partial L}{\partial w_{ji}^{(1)}}, \frac{\partial L}{\partial w_{kj}^{(2)}}, \frac{\partial L}{\partial w_{lk}^{(3)}}$ 각각에 대해서 모두 구해야한다.

Auto Differentiation

Auto Differentiation의 목적

- **Gradient Descent**을 수행하기 위해서는
- Loss을 뉴럴넷 모델의 weight parameter들에 대해서 미분한 값이 필요하다.
- 즉, $\frac{\partial L}{\partial w}$을
    - 뉴럴넷을 구성하는 각 weight parameter $w_i \in \{w_1, \cdots, w_N\}$들에 대해서 구해야한다.
- **Auto Differentiation**은 **미분의 Chain Rule 특징**을 활용해서 **Computationally Efficient 하게** 각 $\frac{\partial L}{\partial w}$ 을 **구하는** 방법이다



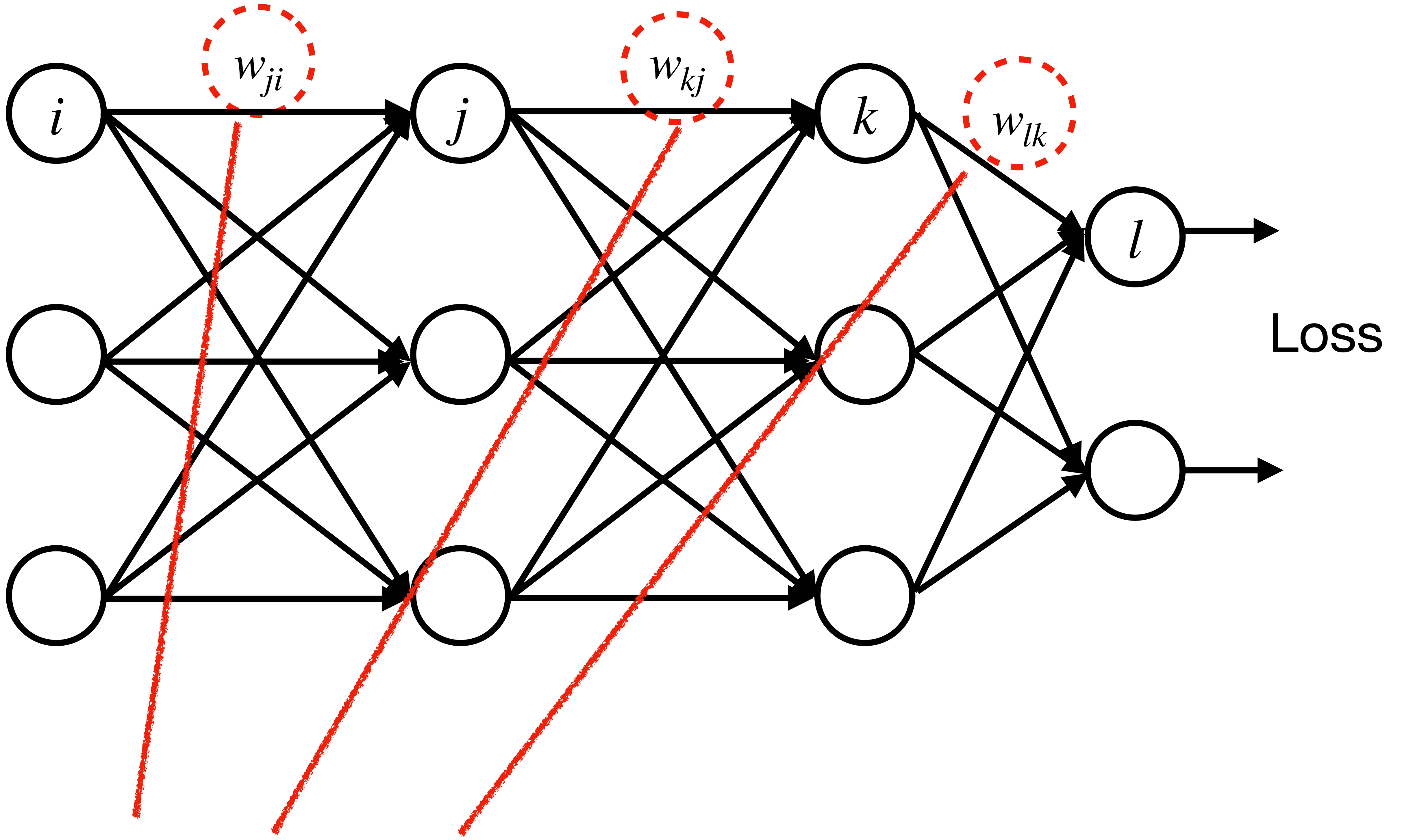
$\frac{\partial L}{\partial w_{ji}^{(1)}}, \frac{\partial L}{\partial w_{kj}^{(2)}}, \frac{\partial L}{\partial w_{lk}^{(3)}}$ 각각에 대해서 모두 구해야한다.

ACADENTIAL

Auto Differentiation

Auto Differentiation

- **2가지 종류:**
    - Forward Differentiation
    - Reverse Differentiation
- **Reverse Differentiation:**
    - PyTorch, Tensorflow 등등 DL Framework 의 기반이다



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

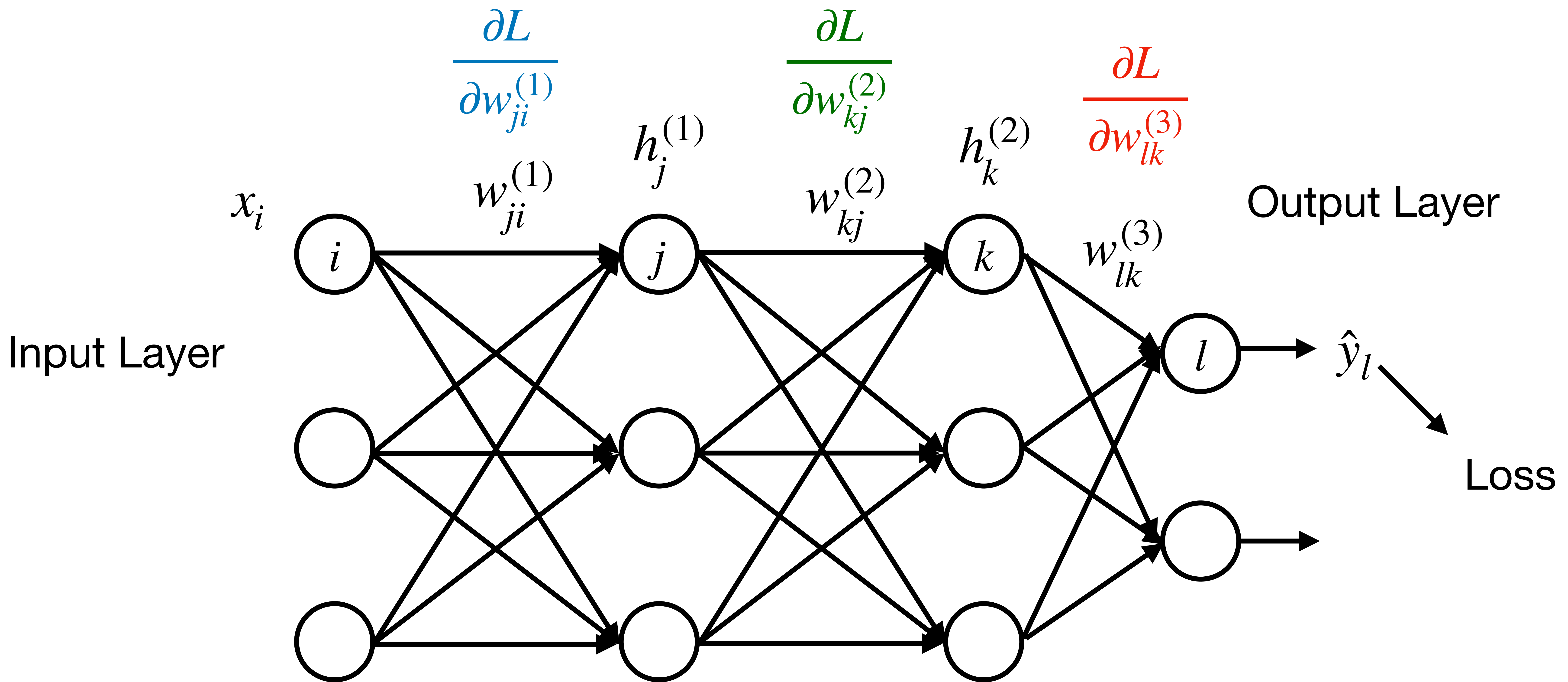
Reverse Differentiation 원리

ACADENTIAL

Auto Differentiation

Reverse Differentiation

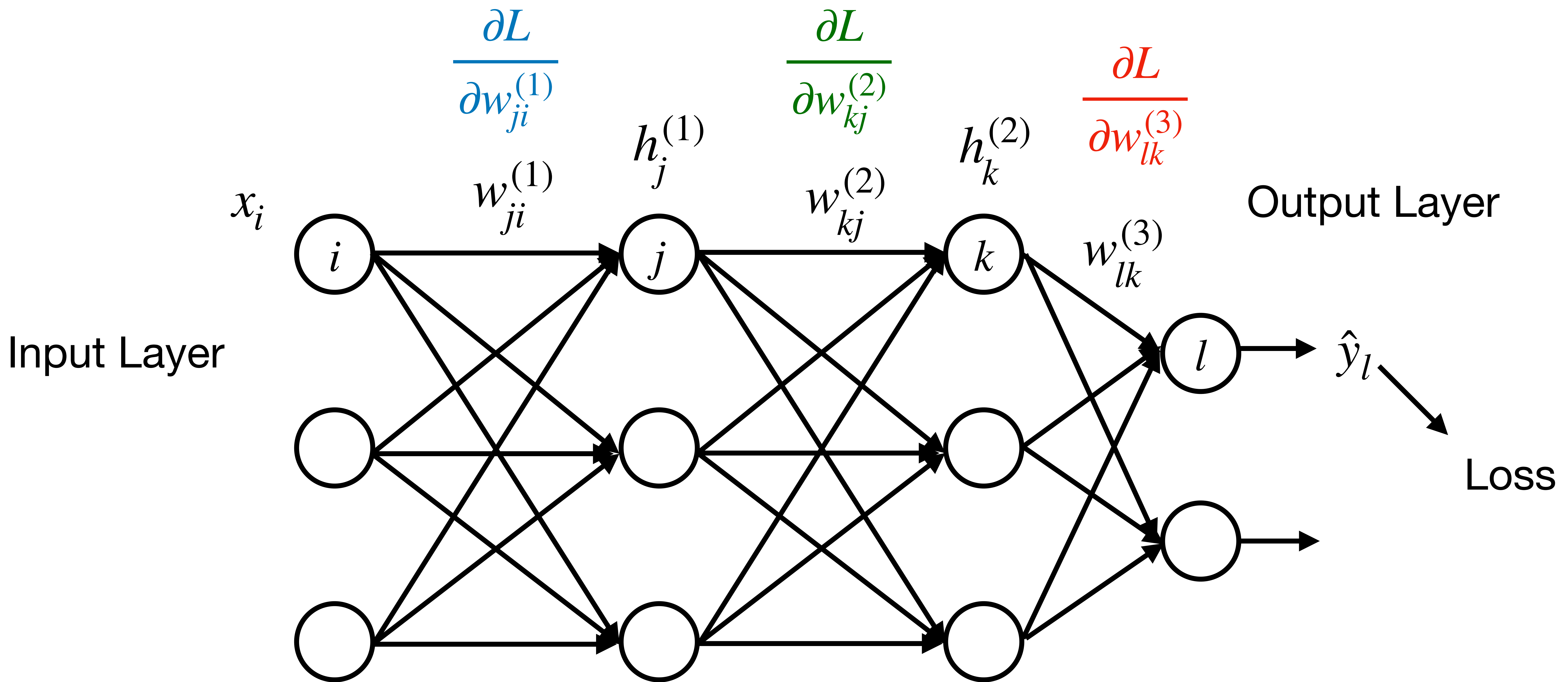
$\frac{\partial L}{\partial w_{lk}^{(3)}}, \frac{\partial L}{\partial w_{kj}^{(2)}}, \frac{\partial L}{\partial w_{ji}^{(1)}}$ 을 구하는 것이 목표



Auto Differentiation

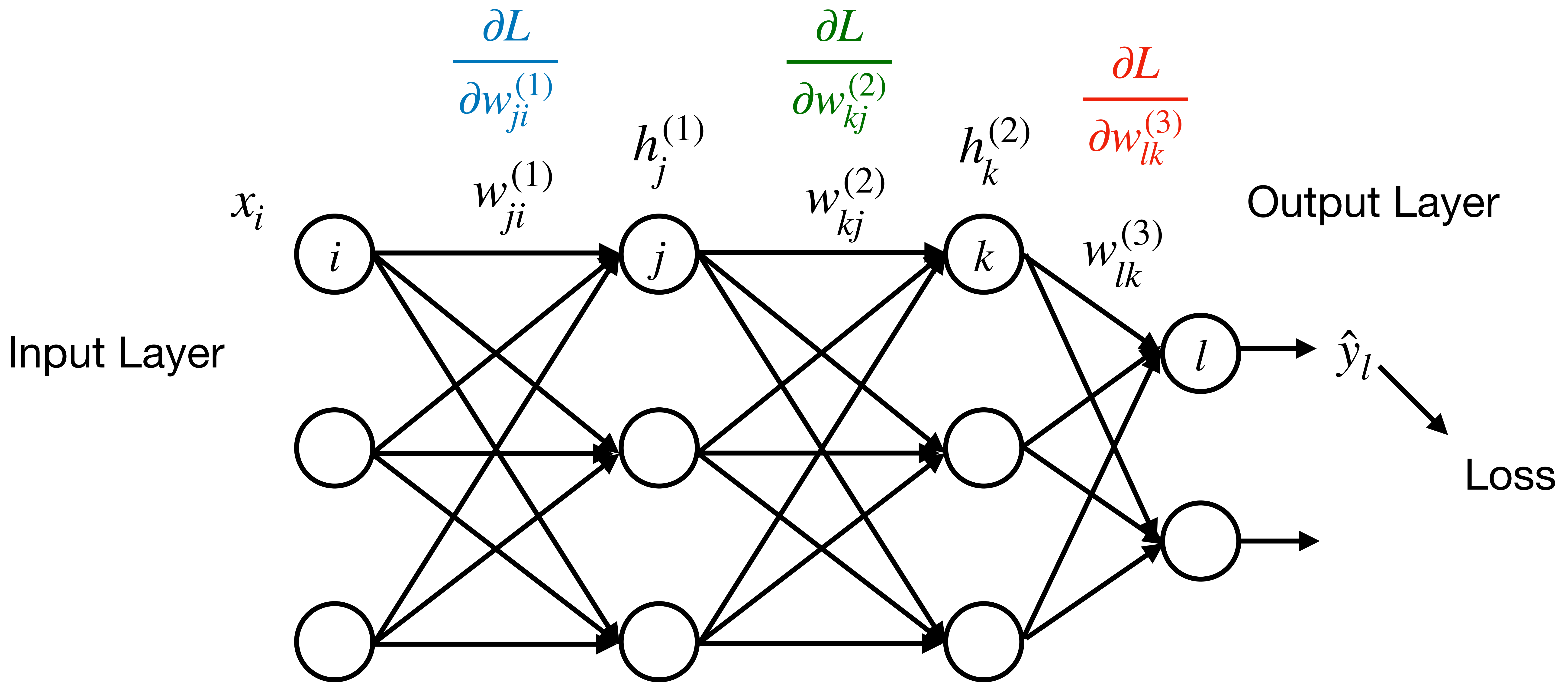
(Reverse differentiation을 살펴보기에 앞서서)

각 $\dfrac{\partial L}{\partial w^{(n)}}$을 직접 계산해보자.



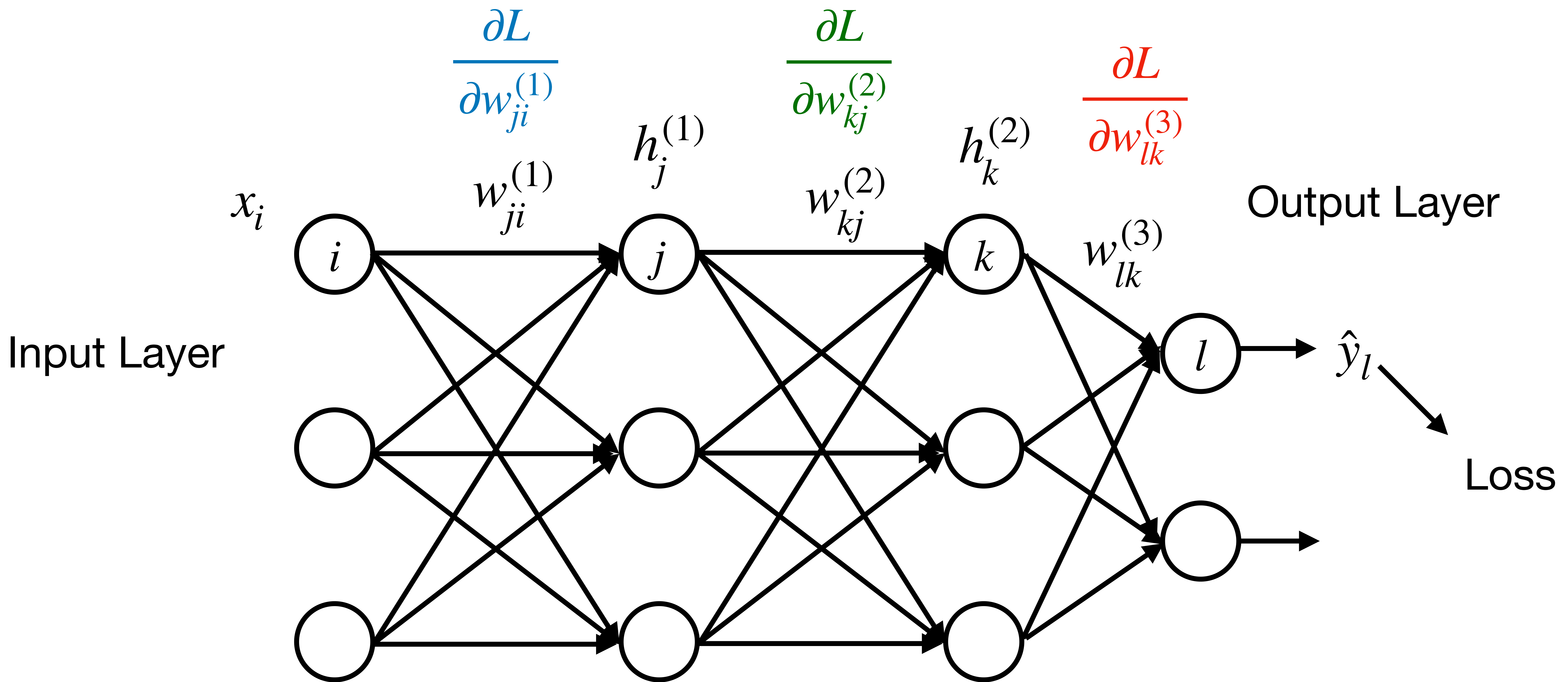
Auto Differentiation

각 $\dfrac{\partial L}{\partial w^{(n)}}$을 직접 계산해보자. → 더 효율적인 계산에 대한 **Hint** → **Reverse Differentiation**의 핵심



Auto Differentiation

$\frac{\partial L}{\partial w^{(n)}}$을 계산하려면 L이 $w^{(n)}$ 으로 어떻게 표현되는지 알아야함.

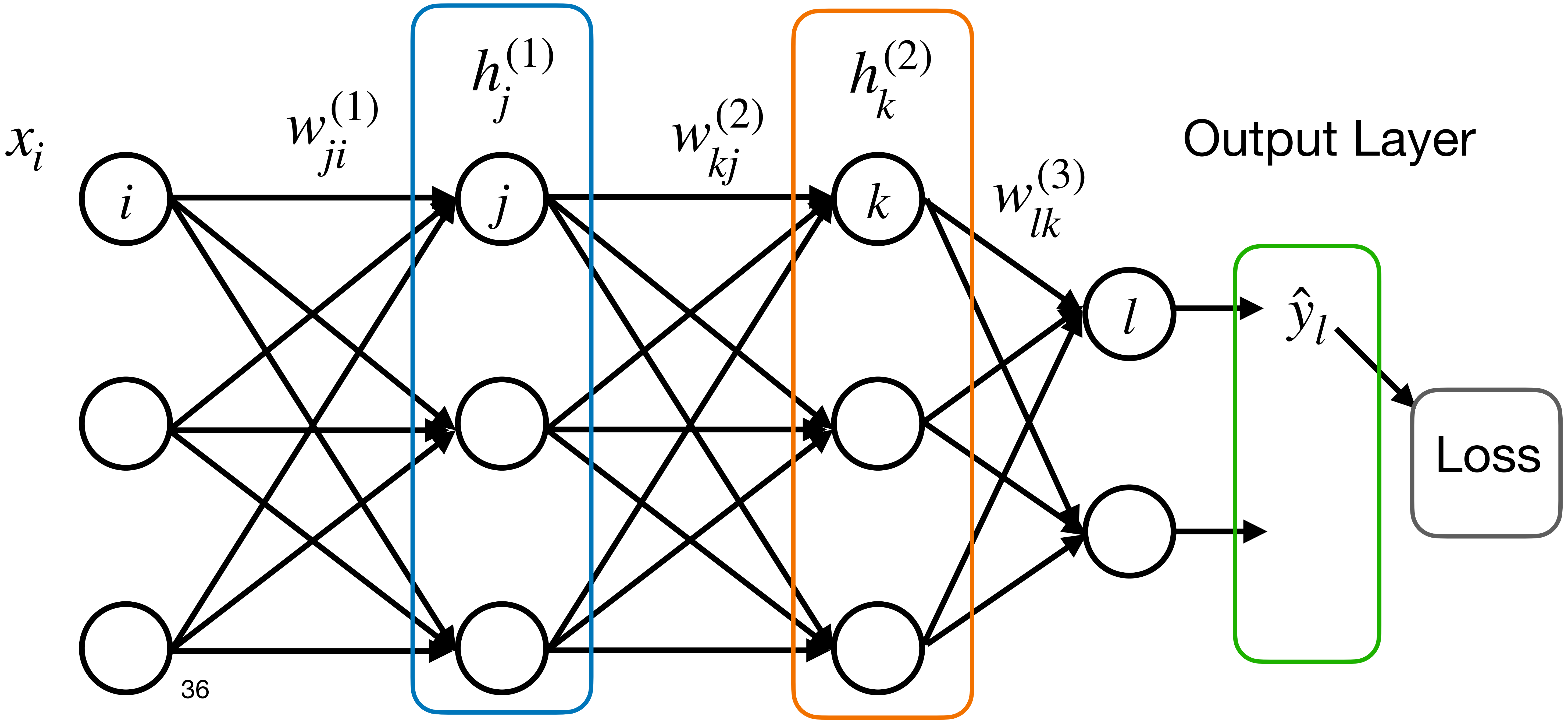


Auto Differentiation

Forward Pass

$$L = L(y, \hat{y}(h^{(2)}(h^{(1)}(x, w^{(1)}), w^{(2)}), w^{(3)}))$$

- $L = L(y_l, \hat{y}_l)$
- $\hat{y}_l = \hat{y}_l(h_k^{(2)}, w_{lk}^{(3)})$
- $h_k^{(2)} = h_k^{(2)}(h_j^{(1)}, w_{kj}^{(2)})$
- $h_j^{(1)} = h_j^{(1)}(x_i, w_{ji}^{(1)})$



ACADENTIAL

Auto Differentiation

Forward Pass

$$L = L(y_l, \hat{y}_l) \longrightarrow L = \sum_l L(y_l, \hat{y}_l)$$

$$\hat{y}_l = \hat{y}_l(h_k^{(2)}, w_{lk}^{(3)}) \longrightarrow \hat{y}_l = \sigma\Big(\sum_k w_{lk}^{(3)} h_k^{(2)}\Big)$$

3번째 NN Layer

$$h_k^{(2)} = h_k^{(2)}(h_j^{(1)}, w_{kj}^{(2)}) \longrightarrow h_k^{(2)} = \sigma\Big(\sum_j w_{kj}^{(2)} h_j^{(1)}\Big)$$

2번째 NN Layer

$$h_j^{(1)} = h_j^{(1)}(x_i, w_{ji}^{(1)}) \longrightarrow h_j^{(1)} = \sigma\Big(\sum_i w_{ji}^{(3)} x_i\Big)$$

1번째 NN Layer

Weight Multiplication → Aggregation → Activation Function

Auto Differentiation

각 weight에 대한 경사

L에 대한 식을
대입해서 전개해보자!

Forward pass:

$$L = \boxed{\sum_l L(y_l, \hat{y}_l)}$$

Backward pass:

$$\frac{\partial \boxed{L}}{\partial w_{lk}^{(3)}}$$

ACADENTIAL

Auto Differentiation

각 weight에 대한 경사

Forward pass:

$$L = \sum_{l} L(y_l, \hat{y}_l)$$

Backward pass:

$$\frac{\partial L}{\partial w_{lk}^{(3)}} = \sum_{l'} \frac{\partial L(y_{l'}, \hat{y}_{l'})}{\partial w_{lk}^{(3)}}$$

ACADENTIAL

Auto Differentiation

각 weight에 대한 경사

Forward pass:

$$L = \sum_{l} L(y_l, \hat{y}_l)$$

$$\hat{y}_{l'} = \sigma\Big(\sum_{k} w^{(3)}_{l'k} h^{(2)}_{k}\Big)$$

Backward pass:

$\boxed{l' \neq l \text{ 의 항들은 0 이다!}}$

$$\frac{\partial L}{\partial w^{(3)}_{lk}} = \cancel{\sum_{l'} \frac{\partial L(y_{l'}, \hat{y}_{l'})}{\partial w^{(3)}_{lk}}}$$

$$= \frac{\partial L(y_l, \hat{y}_l)}{\partial w^{(3)}_{lk}}$$

ACADENTIAL

Auto Differentiation

각 weight에 대한 경사

Forward pass:

$$L = \sum_{l} L(y_l, \hat{y}_l)$$

$$\hat{y}_l = \sigma\Big(\sum_{k} w_{lk}^{(3)} h_k^{(2)}\Big)$$

Backward pass:

$l' \neq l$ 의 항들은 0 이다!

$$\frac{\partial L}{\partial w_{lk}^{(3)}} = \cancel{\sum_{l'} \frac{\partial L(y_{l'}, \hat{y}_{l'})}{\partial w_{lk}^{(3)}}}$$

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

$\hat{y}$ 은 $w_{lk}^{(3)}$의 함수
미분의 연쇄법칙 적용

ACADENTIAL

Auto Differentiation

각 weight에 대한 경사

Forward pass:

$$L = \sum_{l} L(y_l, \hat{y}_l)$$

$$\hat{y}_l = \sigma\Big(\sum_{k} w_{lk}^{(3)} h_k^{(2)}\Big)$$

Backward pass:

$l' \neq l$ 의 항들은 0 이다!

$$\frac{\partial L}{\partial w_{lk}^{(3)}} = \cancel{\sum_{l'} \frac{\partial L(y_{l'}, \hat{y}_{l'})}{\partial w_{lk}^{(3)}}}$$

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

$\hat{y}$ 은 $w_{lk}^{(3)}$의 함수
미분의 연쇄법칙 적용

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

ACADENTIAL

Auto Differentiation

각 weight에 대한 경사

Forward pass:

$$L = \sum_l L(y_l, \hat{y}_l)$$

$$\hat{y}_l = \sigma\Big(\sum_k w_{lk}^{(3)} h_k^{(2)}\Big)$$

$$h_k^{(2)} = \sigma\Big(\sum_j w_{kj}^{(2)} h_j^{(1)}\Big)$$

Backward pass:

$$\frac{\partial L}{\partial w_{kj}^{(2)}}$$

ACADENTIAL

Auto Differentiation

각 weight에 대한 경사

Forward pass:

$$L = \boxed{\sum_l L(y_l, \hat{y}_l)}$$

$$\hat{y}_l = \sigma\Big(\sum_k w_{lk}^{(3)} h_k^{(2)}\Big)$$

$$h_k^{(2)} = \sigma\Big(\sum_j w_{kj}^{(2)} h_j^{(1)}\Big)$$

L에 대한 식을 대입해서 전개

Backward pass:

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

ACADENTIAL

Auto Differentiation

각 weight에 대한 경사

Forward pass:

$$L = \sum_l L(y_l, \hat{y}_l)$$

$$\boxed{\hat{y}_l} = \sigma\Big(\sum_k w_{lk}^{(3)} \boxed{h_k^{(2)}}\Big)$$

$$\boxed{h_k^{(2)}} = \sigma\Big(\sum_j \boxed{w_{kj}^{(2)}} h_j^{(1)}\Big)$$

Backward pass:

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

- $\hat{y}_l$ 은 $h_k^{(2)}$의 함수이고
- $h_k^{(2)}$은 $w_{kj}^{(2)}$의 함수이다
- 따라서 미분의 연쇄법칙으로
- $\hat{y}_l \rightarrow h_k^{(2)} \rightarrow w_{kj}^{(2)}$ 순으로 미분한다!

Auto Differentiation

각 weight에 대한 경사

Forward pass:

$$L = \sum_l L(y_l, \hat{y}_l)$$

$$\boxed{\hat{y}_l = \sigma\Big(\sum_{k'} w_{lk'}^{(3)} h_{k'}^{(2)}\Big)}$$

$$h_k^{(2)} = \sigma\Big(\sum_j w_{kj}^{(2)} h_j^{(1)}\Big)$$

Backward pass:

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

- $\hat{y}_l$ 은 k'들로부터 계산됨
- 연쇄법칙에 따라 k' 의 편미분들의 합으로 구성됨

$$\boxed{\sum_{k'} \frac{\partial \hat{y}_l}{\partial h_{k'}^{(2)}} \cdot \frac{\partial h_{k'}^{(2)}}{\partial w_{kj}^{(2)}}}$$

ACADENTIAL

Auto Differentiation

각 weight에 대한 경사

Forward pass:

$$L = \sum_l L(y_l, \hat{y}_l)$$

$$\boxed{\hat{y}_l = \sigma\Big(\sum_{k'} w_{lk'}^{(3)} h_{k'}^{(2)}\Big)}$$

$$h_k^{(2)} = \sigma\Big(\sum_j w_{kj}^{(2)} h_j^{(1)}\Big)$$

Backward pass:

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

$$\boxed{\sum_{k'} \frac{\partial \hat{y}_l}{\partial h_{k'}^{(2)}} \cdot \frac{\partial h_{k'}^{(2)}}{\partial w_{kj}^{(2)}}}$$

$k' \neq k$ 의 항들은 0 이다!
최종적으로 $k' = k$의 항만 남는다

Auto Differentiation

각 weight에 대한 경사

Forward pass:

$$L = \sum_l L(y_l, \hat{y}_l)$$

$$\hat{y}_l = \sigma\Big(\sum_k w_{lk}^{(3)} h_k^{(2)}\Big)$$

$$h_k^{(2)} = \sigma\Big(\sum_j w_{kj}^{(2)} h_j^{(1)}\Big)$$

Backward pass:

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

최종적으로 $k' = k$의 항만 남는다

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

ACADENTIAL

Auto Differentiation

각 weight에 대한 경사

Forward pass:

$$L = L(y_l, \hat{y}_l)$$

$$\hat{y}_l = \hat{y}_l(h_k^{(2)}, w_{lk}^{(3)})$$

$$h_k^{(2)} = h_k^{(2)}(h_j^{(1)}, w_{kj}^{(2)})$$

$$h_j^{(1)} = h_j^{(1)}(x, w_{ji}^{(1)})$$

Backward pass:

$$\frac{\partial L}{\partial w_{ji}^{(1)}}$$

Auto Differentiation

각 weight에 대한 경사

Forward pass:

$$L = L(y_l, \hat{y}_l)$$

$$\hat{y}_l = \hat{y}_l(h_k^{(2)}, w_{lk}^{(3)})$$

$$h_k^{(2)} = h_k^{(2)}(h_j^{(1)}, w_{kj}^{(2)})$$

$$h_j^{(1)} = h_j^{(1)}(x, w_{ji}^{(1)})$$

Backward pass:

앞의 예시들을 참고해서 계산하면...

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

Auto Differentiation

각 weight에 대한 경사

Forward pass:

$$L = L(y_l, \hat{y}_l)$$

$$\hat{y}_l = \hat{y}_l(h_k^{(2)}, w_{lk}^{(3)})$$

$$h_k^{(2)} = h_k^{(2)}(h_j^{(1)}, w_{kj}^{(2)})$$

$$h_j^{(1)} = h_j^{(1)}(x, w_{ji}^{(1)})$$

Backward pass:

정리해보자면...

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

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

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

ACADENTIAL

Auto Differentiation

Reverse Differentiation의 핵심

중간에 계산된 편미분 값들을 저장 및 재사용

→ Computational Cost 절약

→ Reverse Differentiation의 핵심

Backward pass:

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

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

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

ACADENTIAL

Auto Differentiation

Reverse Differentiation의 핵심

중간에 계산된 편미분 값들을 저장 및 재사용

→ Computational Cost 절약

→ Reverse Differentiation의 핵심

Backward pass:

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

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

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

Reverse Differentiation의 과정

Auto Differentiation

Reverse Differentiation의 과정

Rule 1.

- Descendent Node (Loss)를 현재 Node n_m 에 대해서 미분한 값 $t_{n_m} = \frac{\partial L}{\partial n_m}$이 저장됨.

ACADENTIAL

Auto Differentiation

Reverse Differentiation의 과정

Rule 1.

- Descendent Node (Loss)를 현재 Node n_m 에 대해서 미분한 값 $t_{n_m} = \frac{\partial L}{\partial n_m}$이 저장됨.

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

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

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

ACADENTIAL

Auto Differentiation

Reverse Differentiation의 과정

Rule 1.

- Descendent Node (Loss)를 현재 Node n_m 에 대해서 미분한 값 $t_{n_m} = \frac{\partial L}{\partial n_m}$이 저장됨.

$$t_{\hat{y}_l}$$

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

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

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

Auto Differentiation

Reverse Differentiation의 과정

Rule 1.

- Descendent Node (Loss)를 현재 Node n_m 에 대해서 미분한 값 $t_{n_m} = \dfrac{\partial L}{\partial n_m}$이 저장됨.

$$t_{\hat{y}_l}$$

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

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

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

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

ACADENTIAL

Auto Differentiation

Reverse Differentiation의 과정

Rule 1.

- Descendent Node (Loss)를 현재 Node n_m 에 대해서 미분한 값 $t_{n_m} = \dfrac{\partial L}{\partial n_m}$이 저장됨.

$$t_{\hat{y}_l}$$

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

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

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

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

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

Auto Differentiation

Reverse Differentiation의 과정

Rule 2.

- Reverse Differentiation은 Computational Graph 상에서 Descendent Node로부터 시작해서 거꾸로 수행됨.

$$t_{\hat{y}_l}$$

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

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

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

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

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

Auto Differentiation

Reverse Differentiation의 과정

Forward Pass

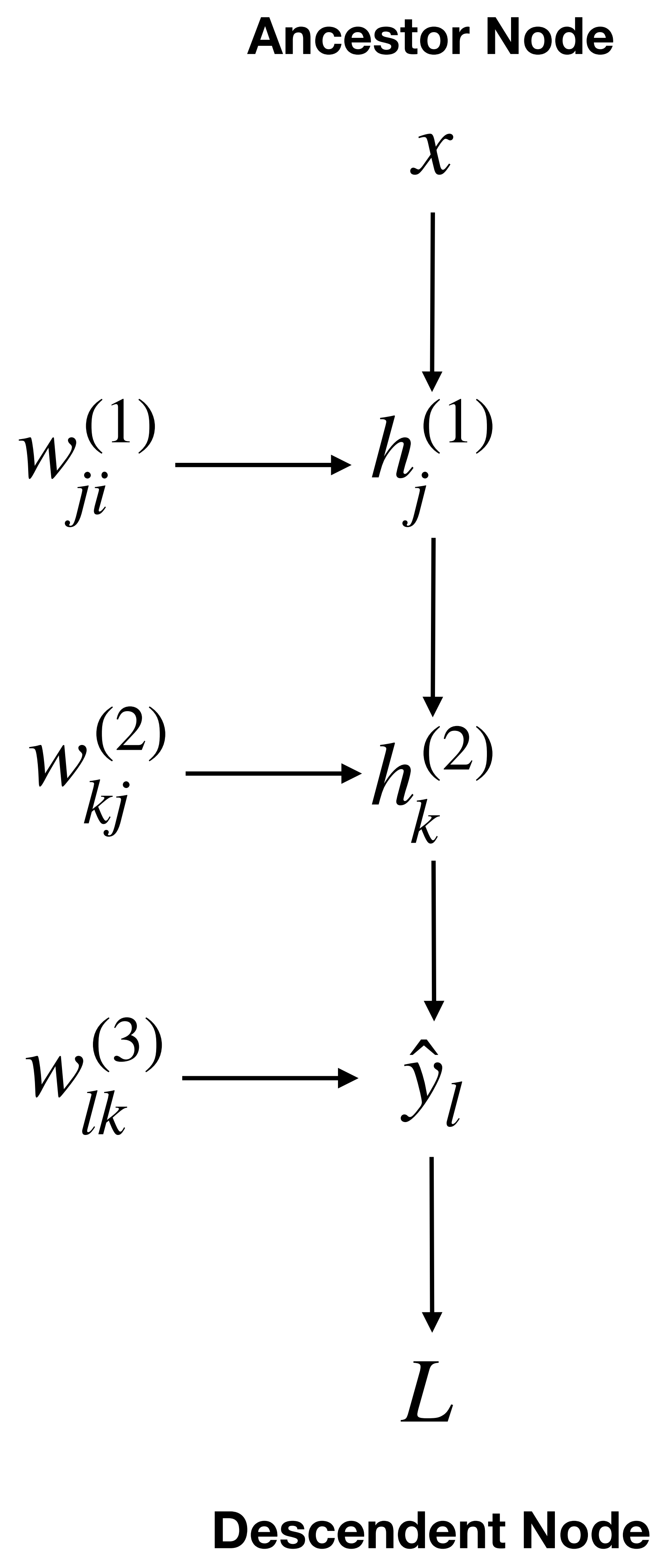
$$L = L(y_l, \hat{y}_l)$$

$$\hat{y}_l = \hat{y}_l(h_k^{(2)}, w_{lk}^{(3)})$$

$$h_k^{(2)} = h_k^{(2)}(h_j^{(1)}, w_{kj}^{(2)})$$

$$h_j^{(1)} = h_j^{(1)}(x, w_{ji}^{(1)})$$

Computational Graph



Auto Differentiation

Reverse Differentiation의 과정

Forward Pass

$$L = L(y_l, \hat{y}_l)$$

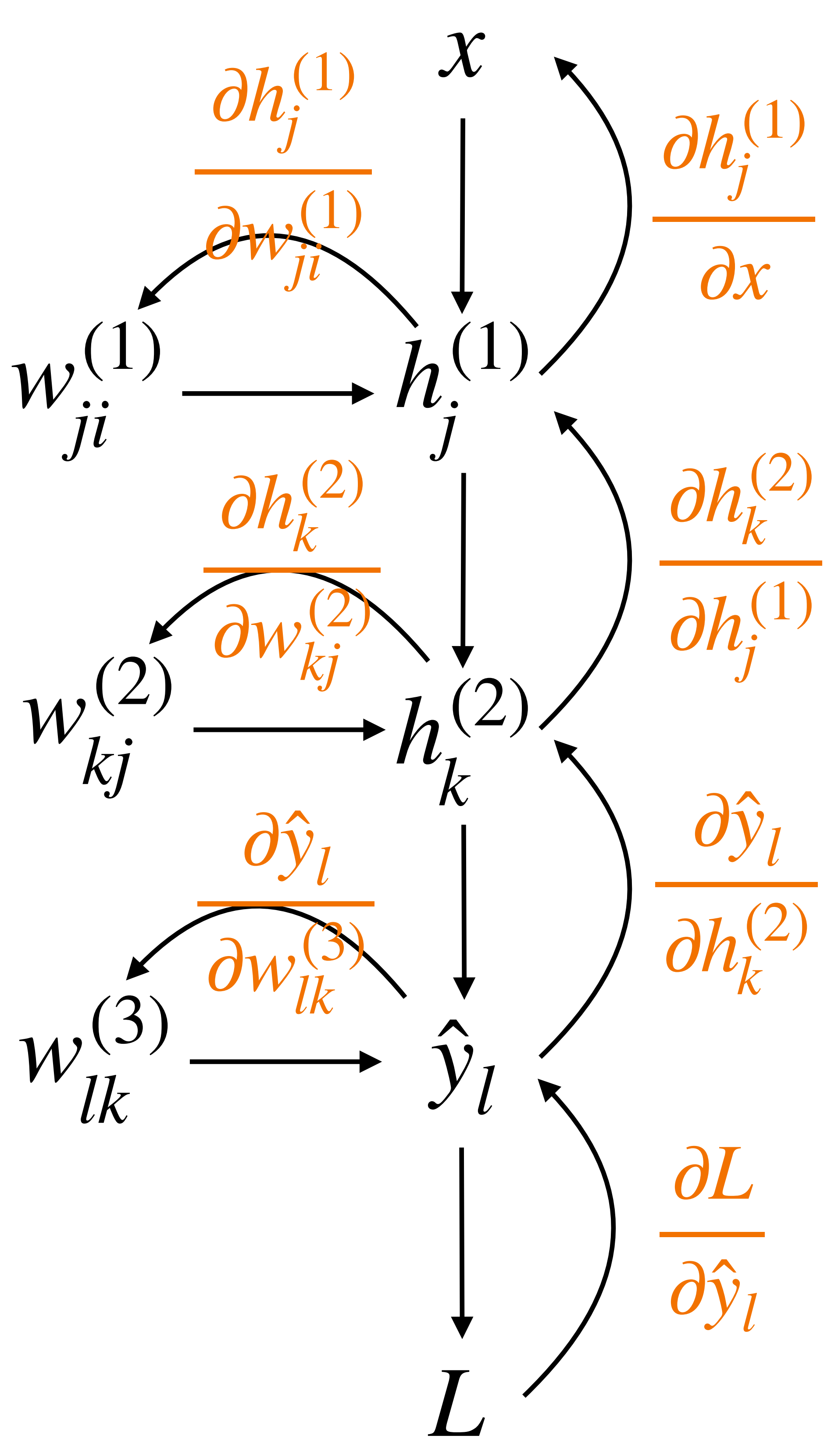
$$\hat{y}_l = \hat{y}_l(h_k^{(2)}, w_{lk}^{(3)})$$

$$h_k^{(2)} = h_k^{(2)}(h_j^{(1)}, w_{kj}^{(2)})$$

$$h_j^{(1)} = h_j^{(1)}(x, w_{ji}^{(1)})$$

Computational Graph

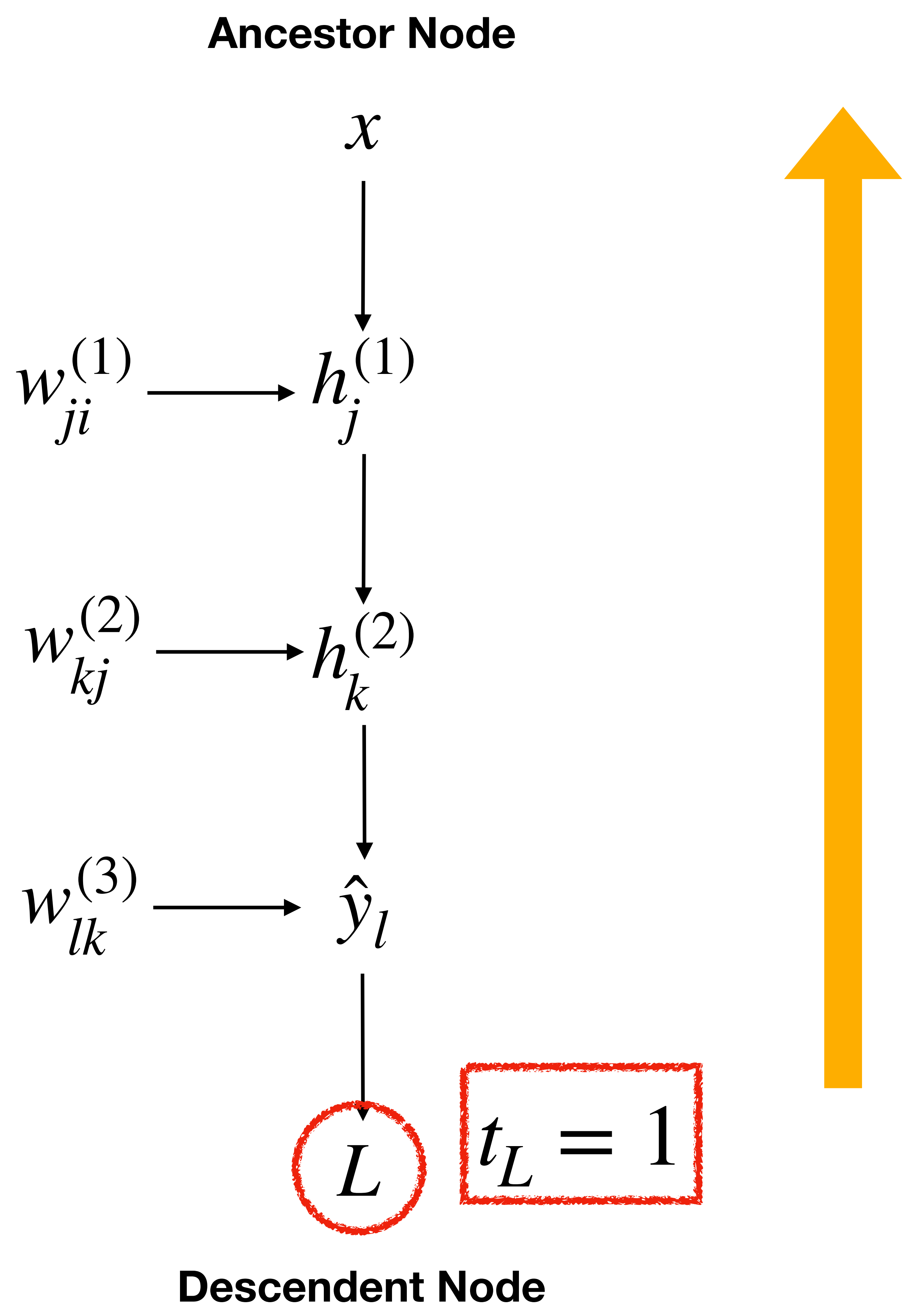
Ancestor Node



Descendent Node

Auto Differentiation

Computational Graph

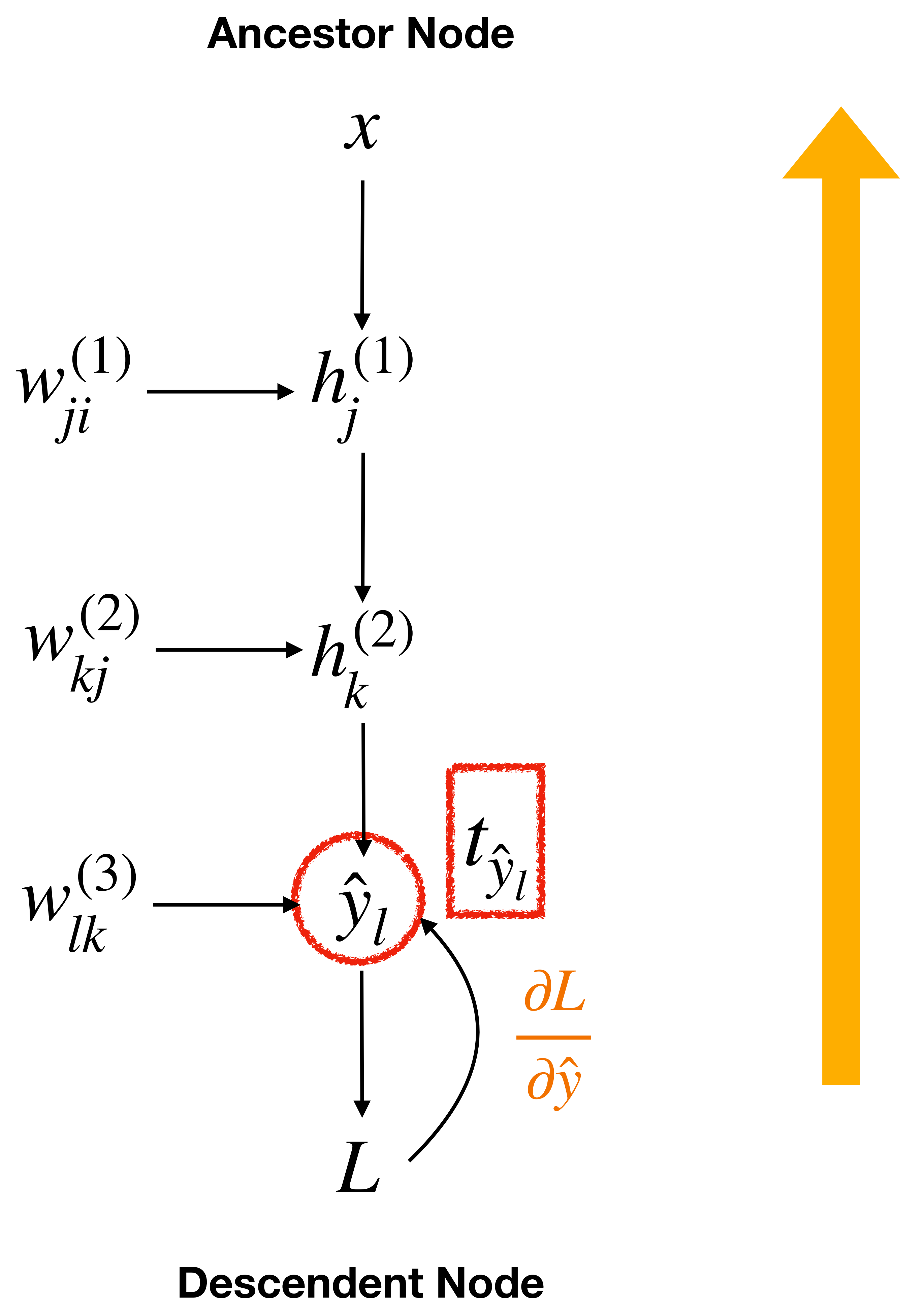


Steps:

1. Descendent Node로 부터 시작 ($t_L = 1$)

Auto Differentiation

Computational Graph



Steps:

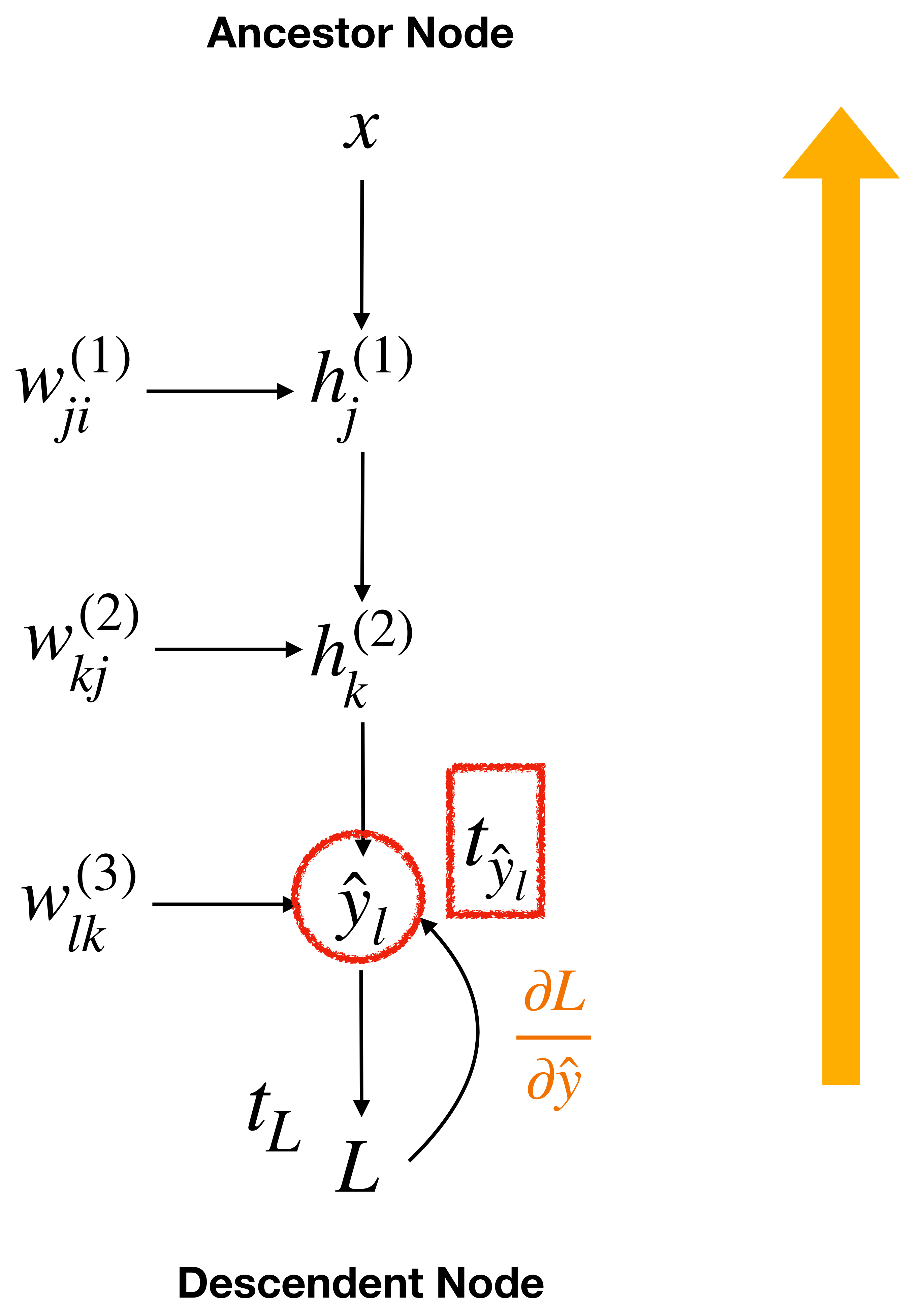
1. Descendent Node로 부터 시작 ($t_L = 1$)
2. $\hat{y}_l$ 노드의 $t_{\hat{y}_l}$ 계산:

$$t_{\hat{y}} = t_L \cdot \frac{\partial L}{\partial \hat{y}_l}$$

ACADENTIAL

Auto Differentiation

Computational Graph



Steps:

1. Descendent Node로 부터 시작 ($t_L = 1$)

2. $\hat{y}_l$ 노드의 $t_{\hat{y}_l}$ 계산:

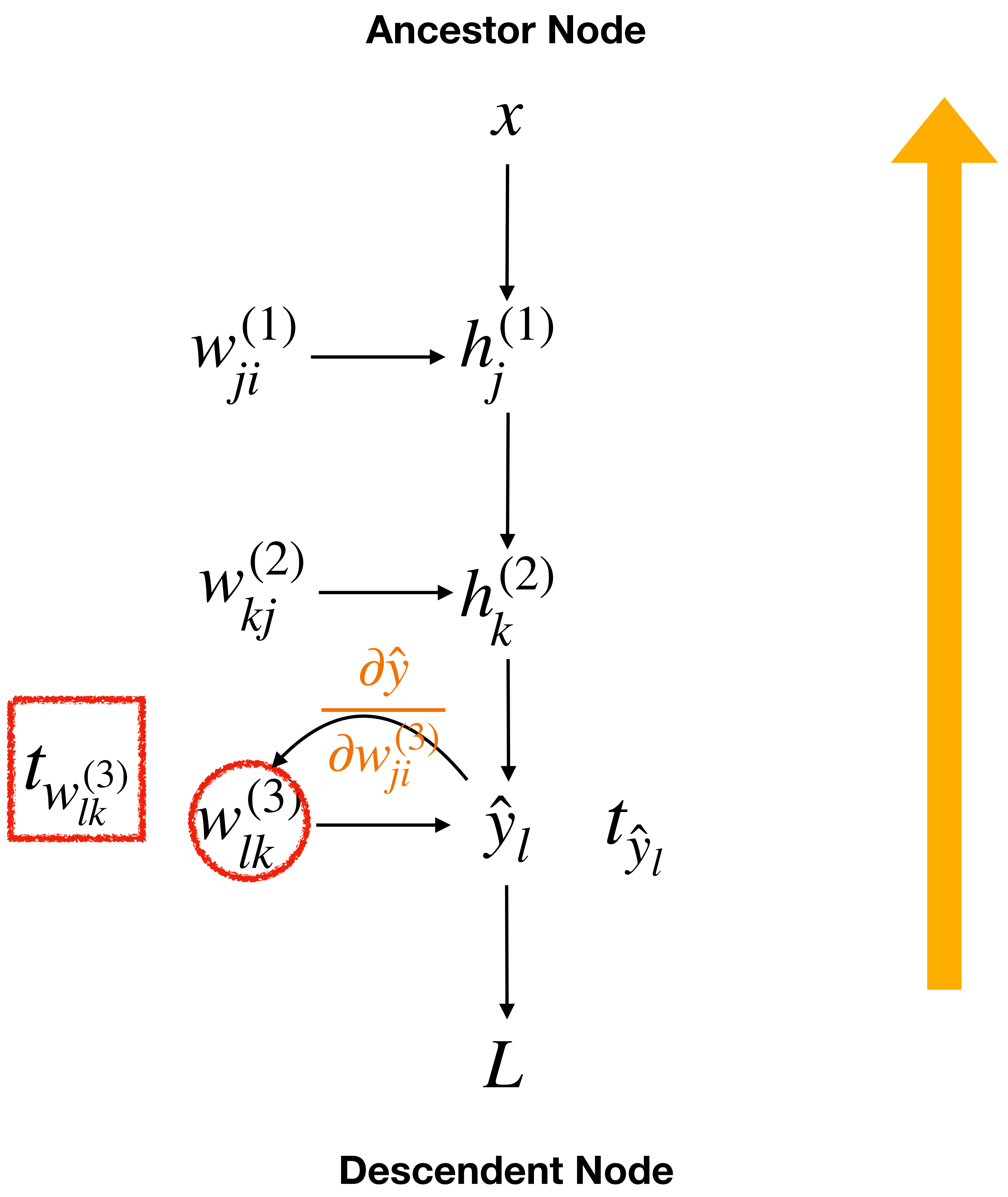
$$t_{\hat{y}} = t_L \cdot \frac{\partial L}{\partial \hat{y}_l}$$

Update Rule:

$$\rightarrow t_{n_m} = t_{n_{m+1}} \cdot \frac{\partial n_{m+1}}{\partial n_m}$$

Auto Differentiation

Computational Graph



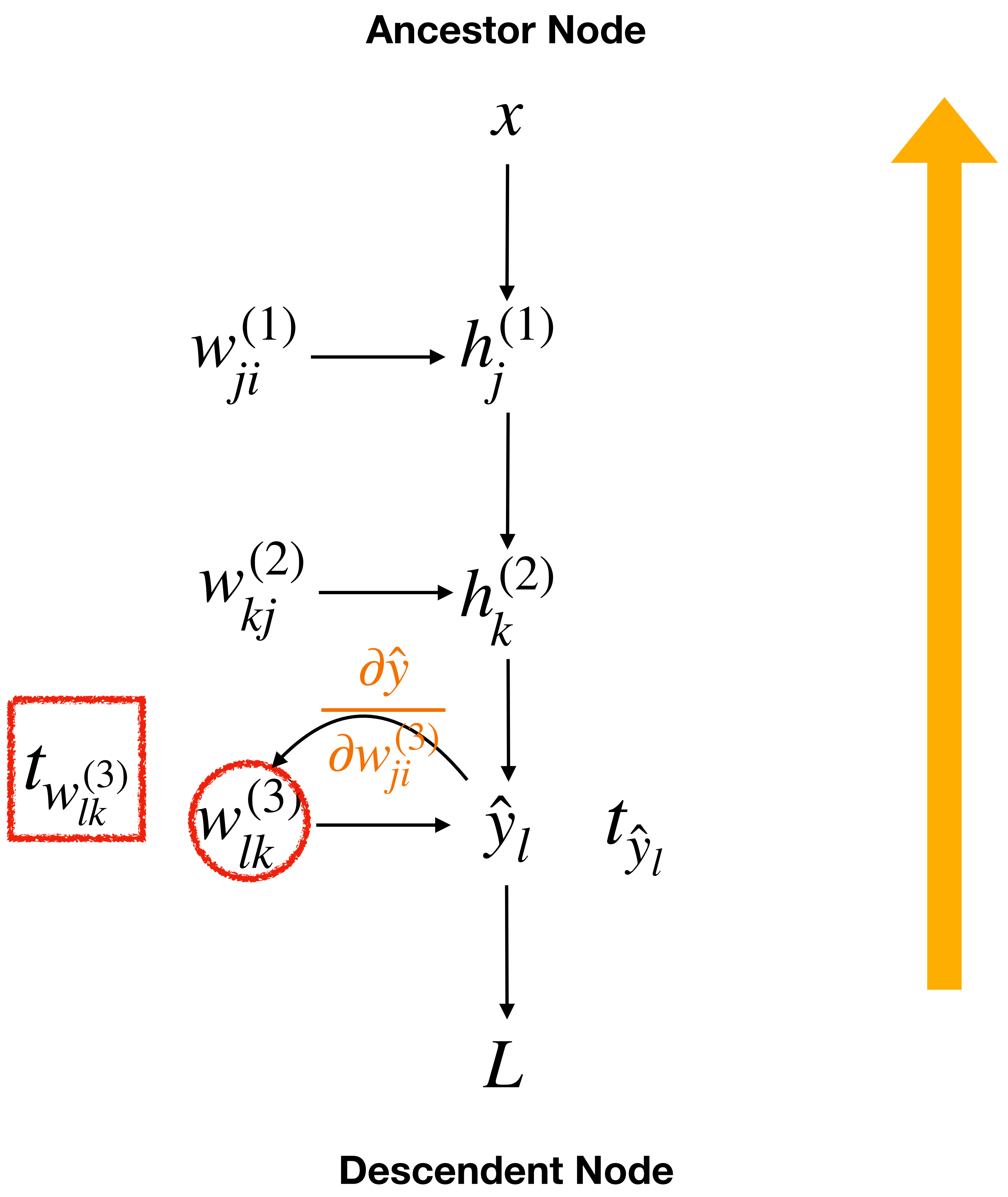
Steps:

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:

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

Auto Differentiation

Computational Graph



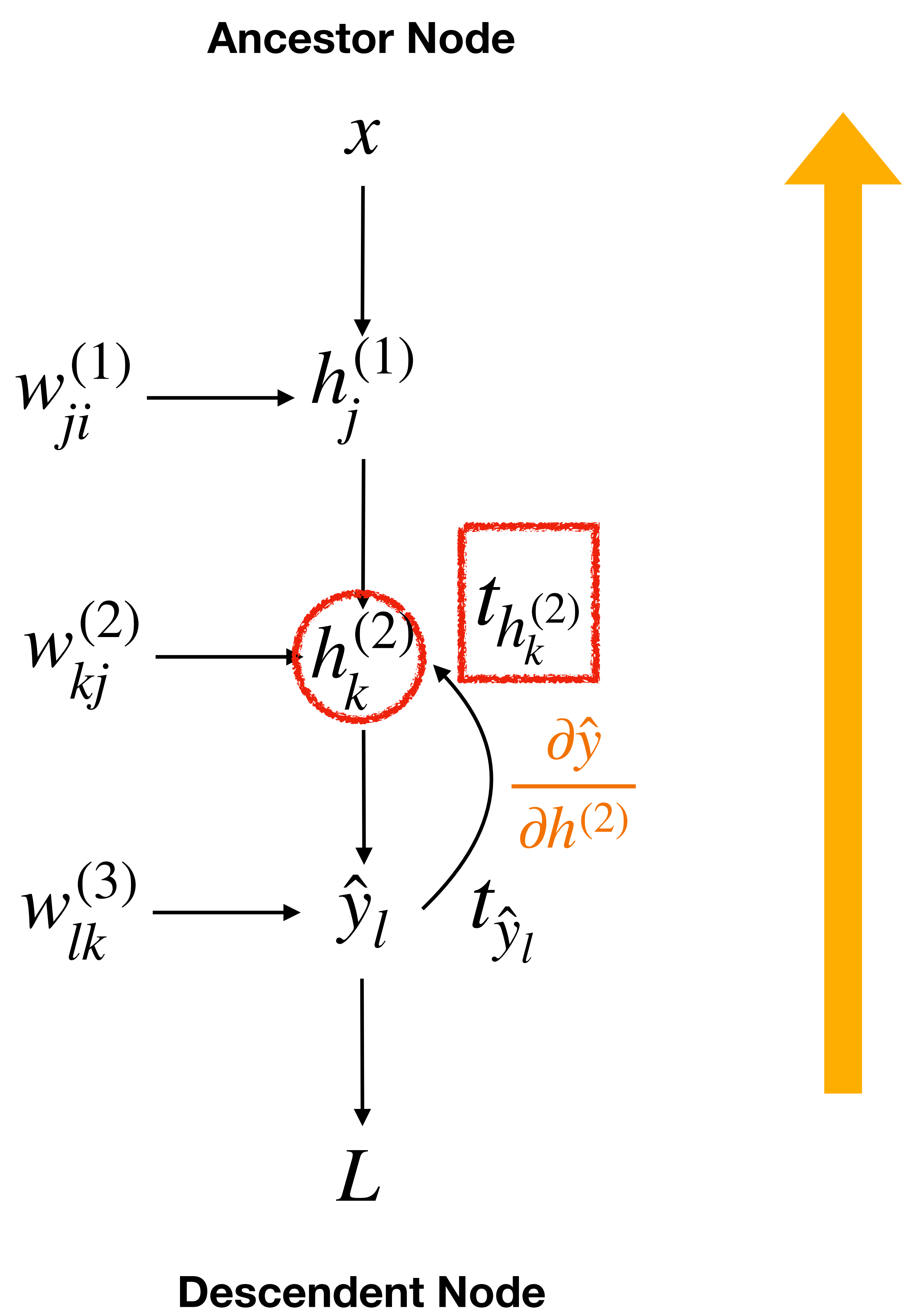
Steps:

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:

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

Auto Differentiation

Computational Graph



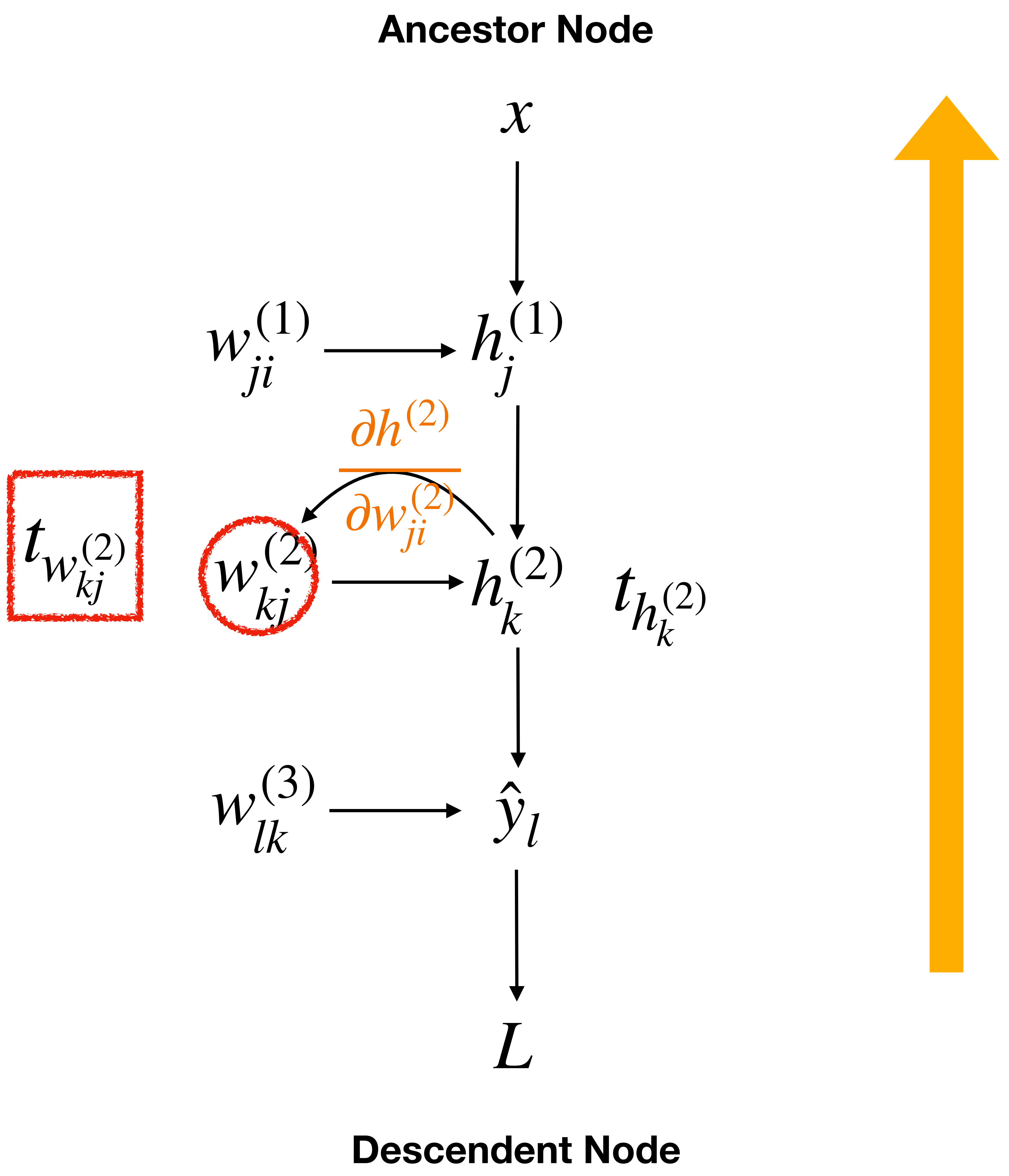
Steps:

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
4. $h_k^{(2)}$ 노드의 $t_{h_k^{(2)}}$ 계산

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

Auto Differentiation

Computational Graph



Steps:

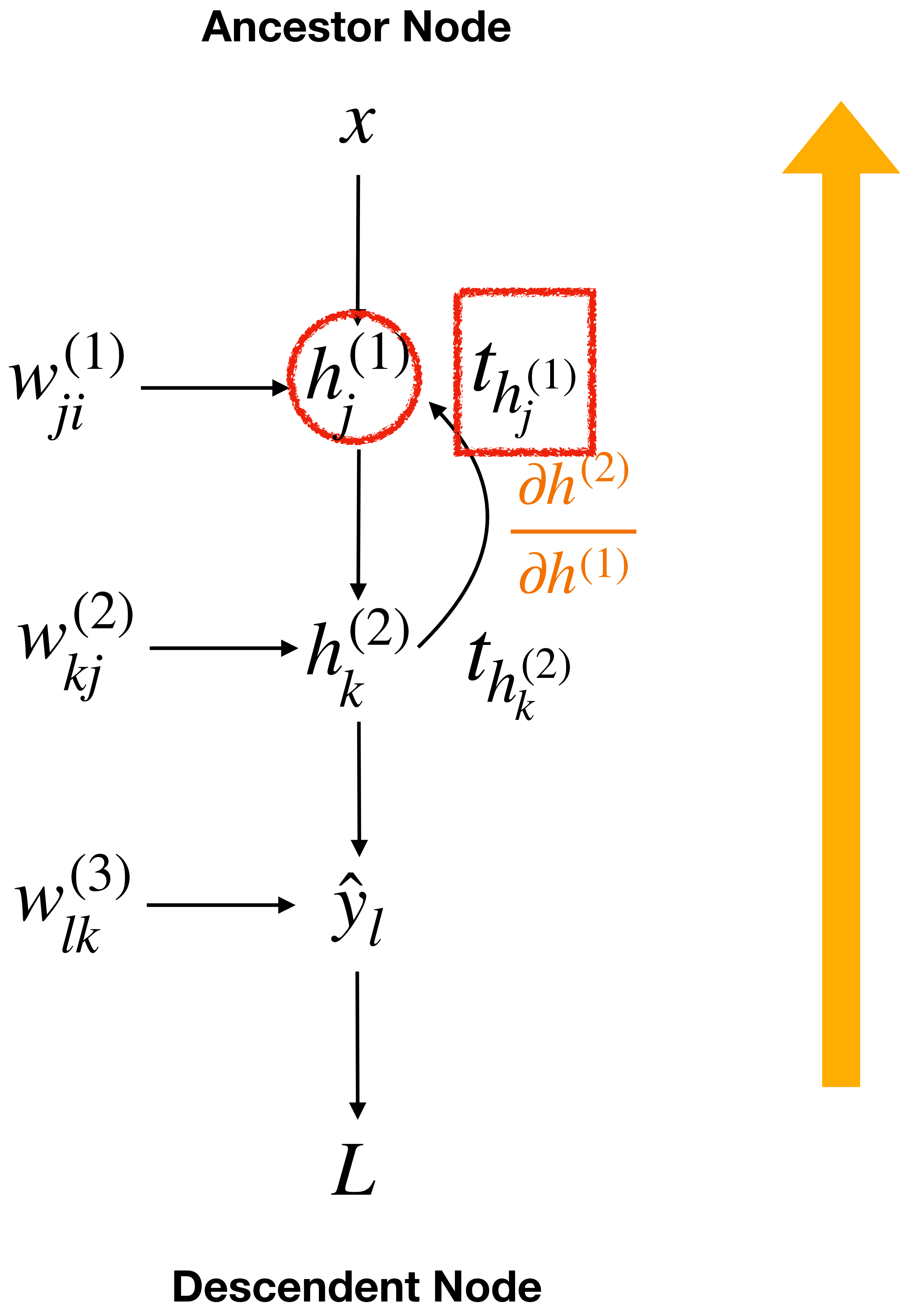
1. Descendent Node로 부터 시작 ($t_L = 1$)
2. $\hat{y}_l$ 노드의 $t_{\hat{y}_l}$ 계산
3. $w_{lk}^{(3)}$ 노드의 $t_{w_{lk}^{(3)}}$ 계산
4. $h_k^{(2)}$ 노드의 $t_{h_k^{(2)}}$ 계산
5. $w_{kj}^{(2)}$ 노드의 $t_{w_{kj}^{(2)}}$ 계산

$$t_{w_{kj}^{(2)}} = \boxed{\frac{\partial L}{\partial w_{kj}^{(2)}}} = t_{h_k^{(2)}} \cdot \frac{\partial h_k^{(2)}}{\partial w_{kj}^{(2)}}$$

ACADENTIAL

Auto Differentiation

Computational Graph



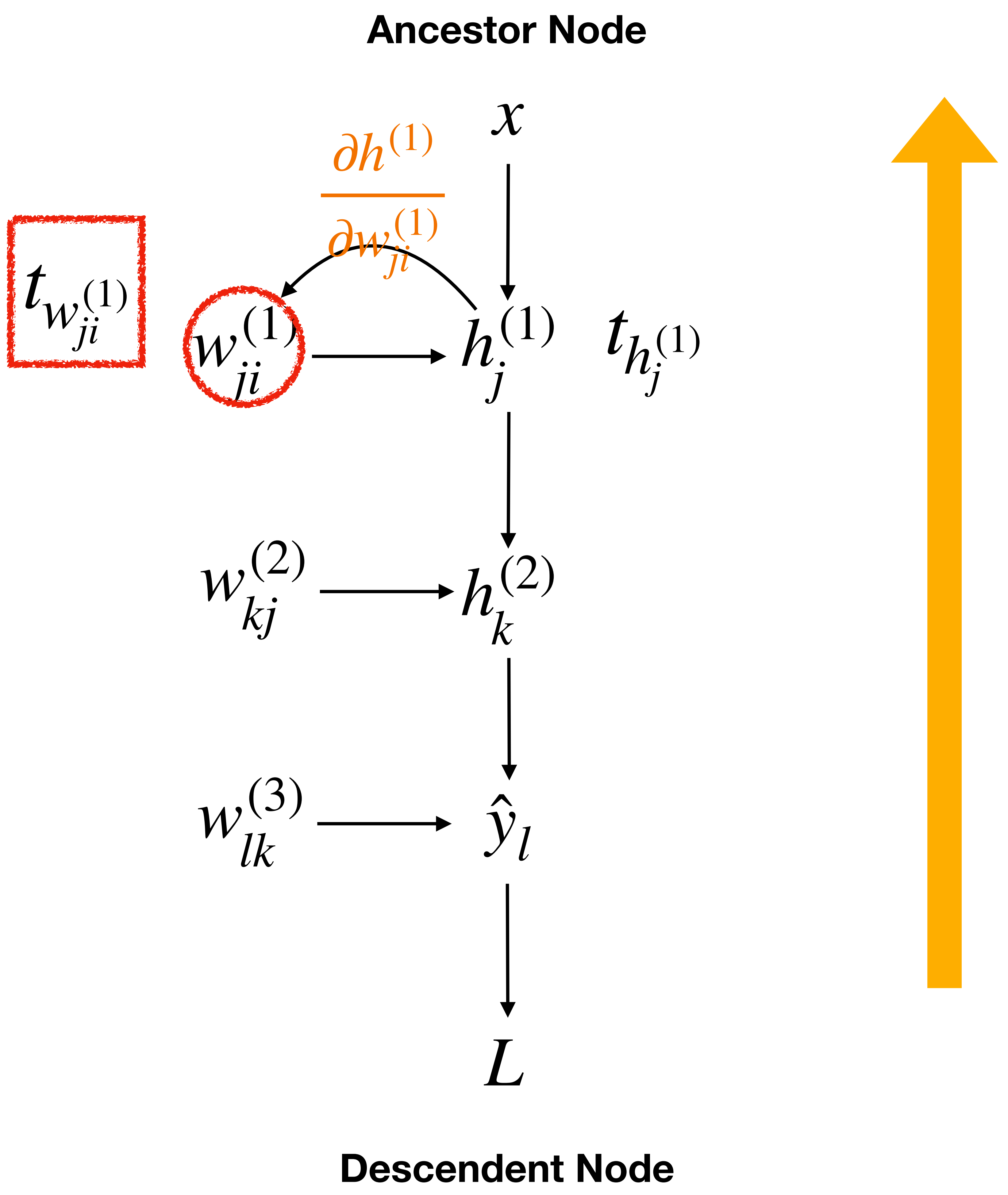
Steps:

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
4. $h_k^{(2)}$ 노드의 $t_{h_k^{(2)}}$ 계산
5. $w_{kj}^{(2)}$ 노드의 $t_{w_{kj}^{(2)}}$ 계산
6. $h_j^{(1)}$ 노드의 $t_{h_j^{(1)}}$ 계산

$$t_{h_j^{(1)}} = \sum_k t_{h_k^{(2)}} \cdot \frac{\partial h_k^{(2)}}{\partial h_j^{(1)}}$$

Auto Differentiation

Computational Graph



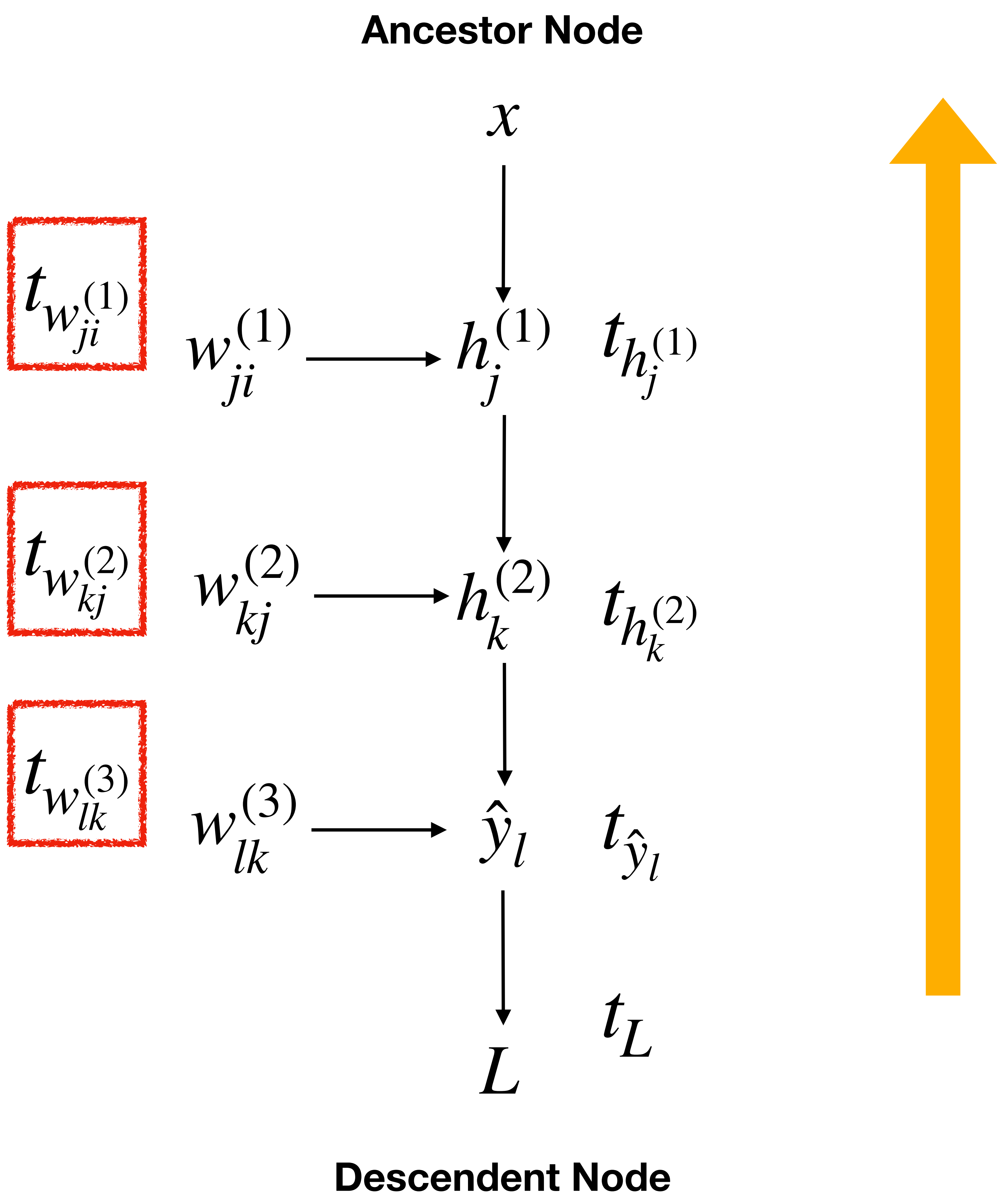
Steps:

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
4. $h_k^{(2)}$ 노드의 $t_{h_k^{(2)}}$ 계산
5. $w_{kj}^{(2)}$ 노드의 $t_{w_{kj}^{(2)}}$ 계산
6. $h_j^{(1)}$ 노드의 $t_{h_j^{(1)}}$ 계산
7. $w_{ji}^{(1)}$ 노드의 $t_{w_{ji}^{(1)}}$ 계산

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

Auto Differentiation

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
4. $h_k^{(2)}$ 노드의 $t_{h_k^{(2)}}$ 계산
5. $w_{kj}^{(2)}$ 노드의 $t_{w_{kj}^{(2)}}$ 계산
6. $h_j^{(1)}$ 노드의 $t_{h_j^{(1)}}$ 계산
7. $w_{ji}^{(1)}$ 노드의 $t_{w_{ji}^{(1)}}$ 계산

Update Rule:

$$\rightarrow t_{n_m} = t_{n_{m+1}} \cdot \frac{\partial n_{m+1}}{\partial n_m}$$

ACADENTIAL

Supplementary:

Auto Differentiation의 예시

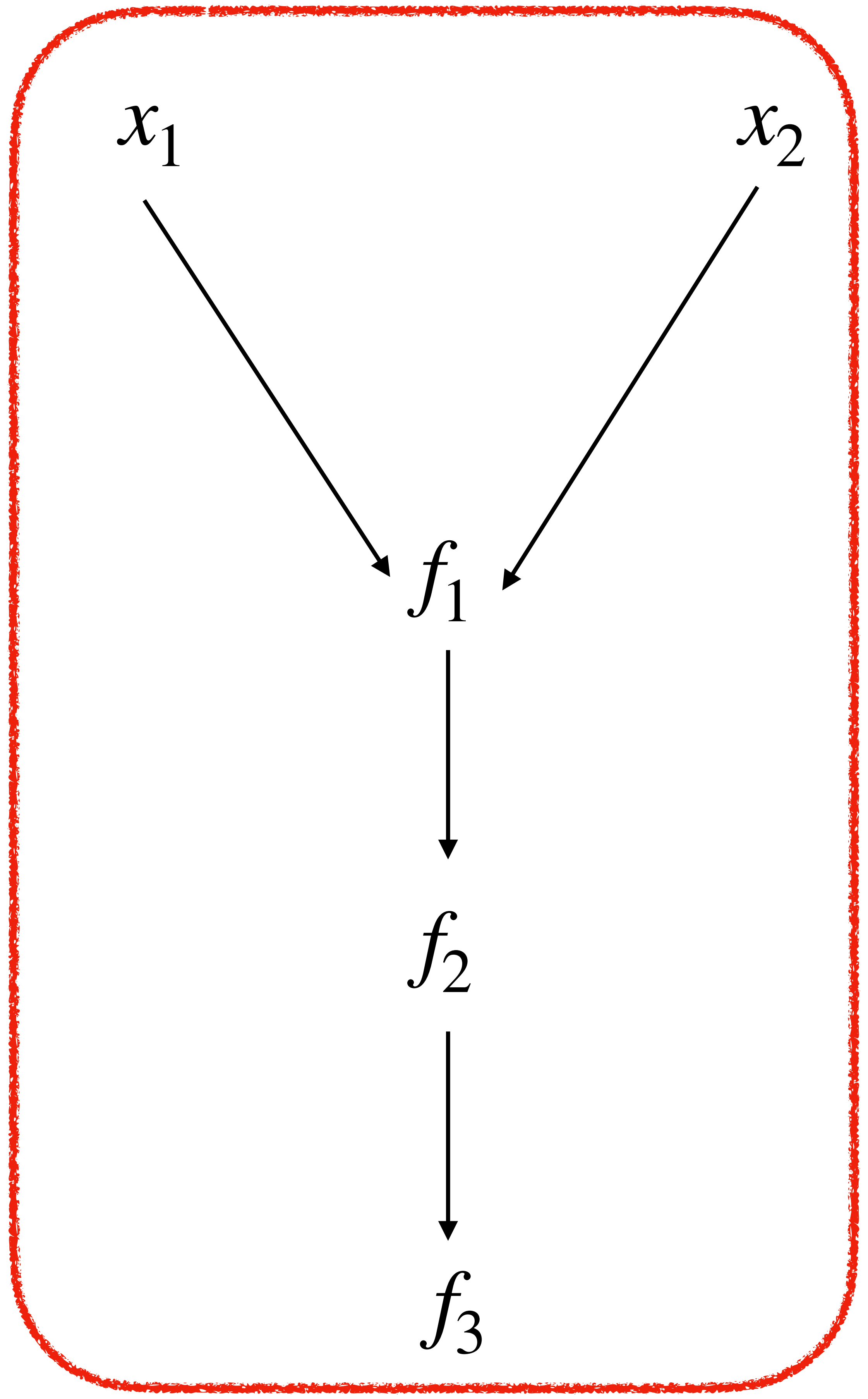
Auto Differentiation

Auto Differentiation 예제

- Auto Differentiation의 개념이 조금 헷갈릴 수 있다.
- 이해하기 쉽게 simple, concrete한 예제를 한 번 살펴보자!

Auto Differentiation

Auto Differentiation 예제



다음과 같은 함성함수의 예를들어 보자.

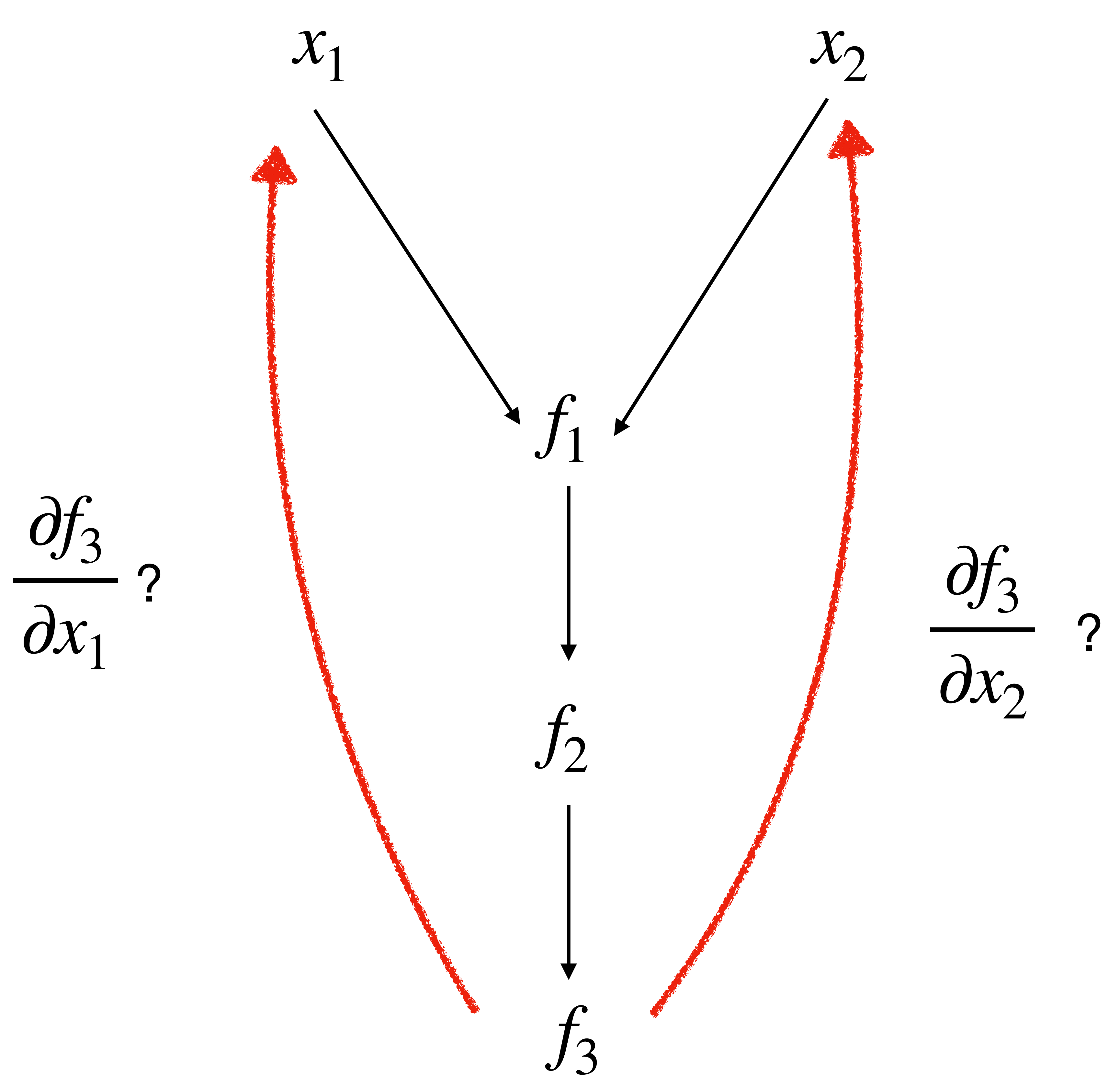
$$f = f_3(f_2(f_1(x_1, x_2)))$$

출처: David Barber's Introduction to Deep Learning

ACADENTIAL

Auto Differentiation

Auto Differentiation 예제



다음과 같은 합성함수의 예를들어 보자.

$$f = f_3(f_2(f_1(x_1, x_2)))$$

이는 Computational Graph로 왼쪽과 같이 표현할 수 있다.

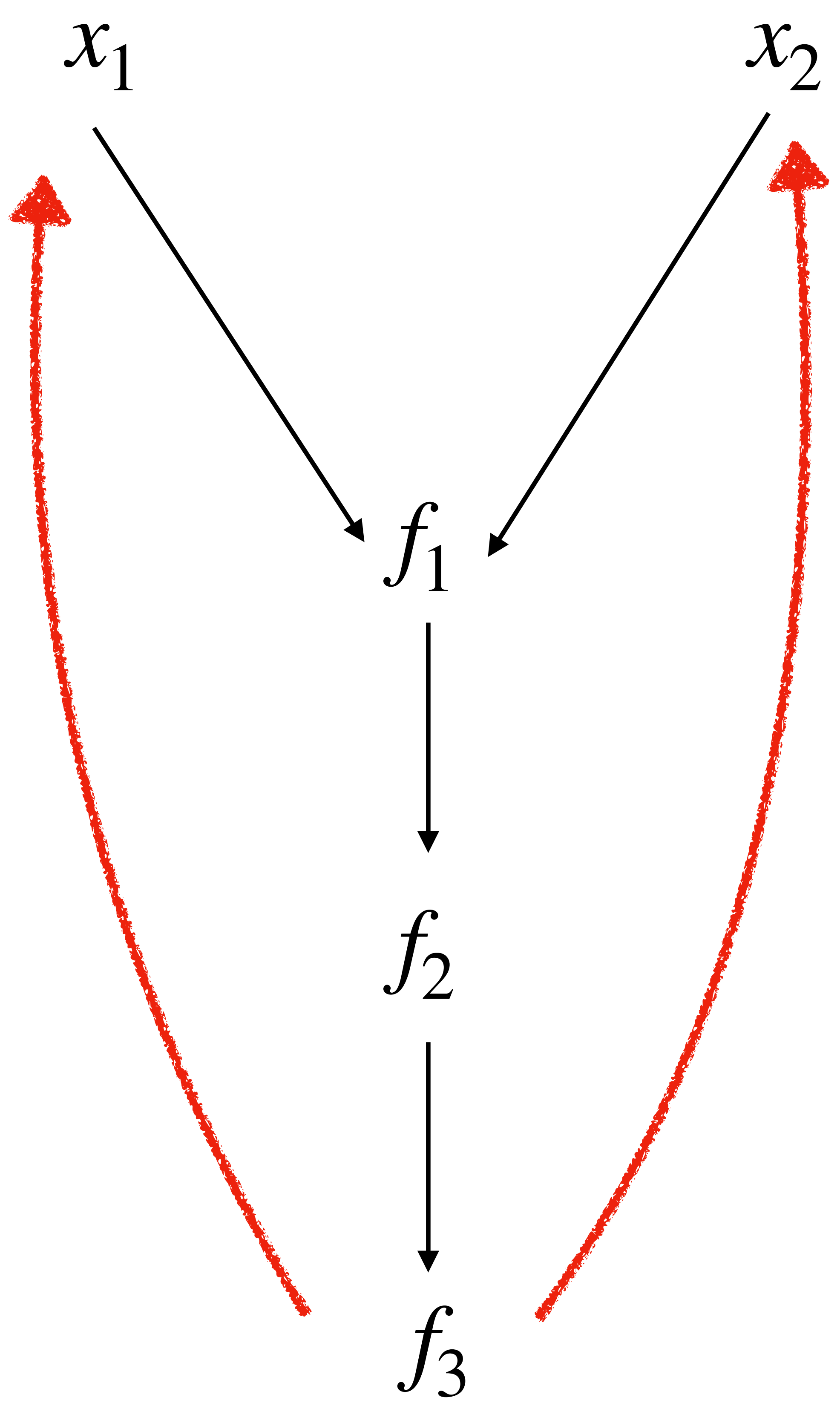
여기서 $\dfrac{\partial f}{\partial x_1}$와 $\dfrac{\partial f}{\partial x_2}$을 구하고 싶으면?

참고, $\dfrac{\partial f}{\partial x_1} = \dfrac{\partial f_3}{\partial x_1}$

출처: David Barber's Introduction to Deep Learning

Auto Differentiation

Auto Differentiation 예제



$\frac{\partial f_3}{\partial x_1}$?

$\frac{\partial f_3}{\partial x_2}$?

$$\frac{\partial f_3}{\partial x_1} = \frac{df_3}{df_2}\frac{\partial f_2}{\partial x_1} = \boxed{\frac{df_3}{df_2}\frac{df_2}{df_1}}\frac{\partial f_1}{\partial x_1}$$

$$= \frac{df_3}{df_1}$$

$$\frac{\partial f_3}{\partial x_2} = \frac{df_3}{df_2}\frac{\partial f_2}{\partial x_2} = \boxed{\frac{df_3}{df_2}\frac{df_2}{df_1}}\frac{\partial f_1}{\partial x_2}$$

$$= \frac{df_3}{df_1}$$

계산하는데 있어 공통적으로 $\frac{df_3}{df_1}$이 필요하다!

이 특징을 어떻게 활용할 수 있을까?

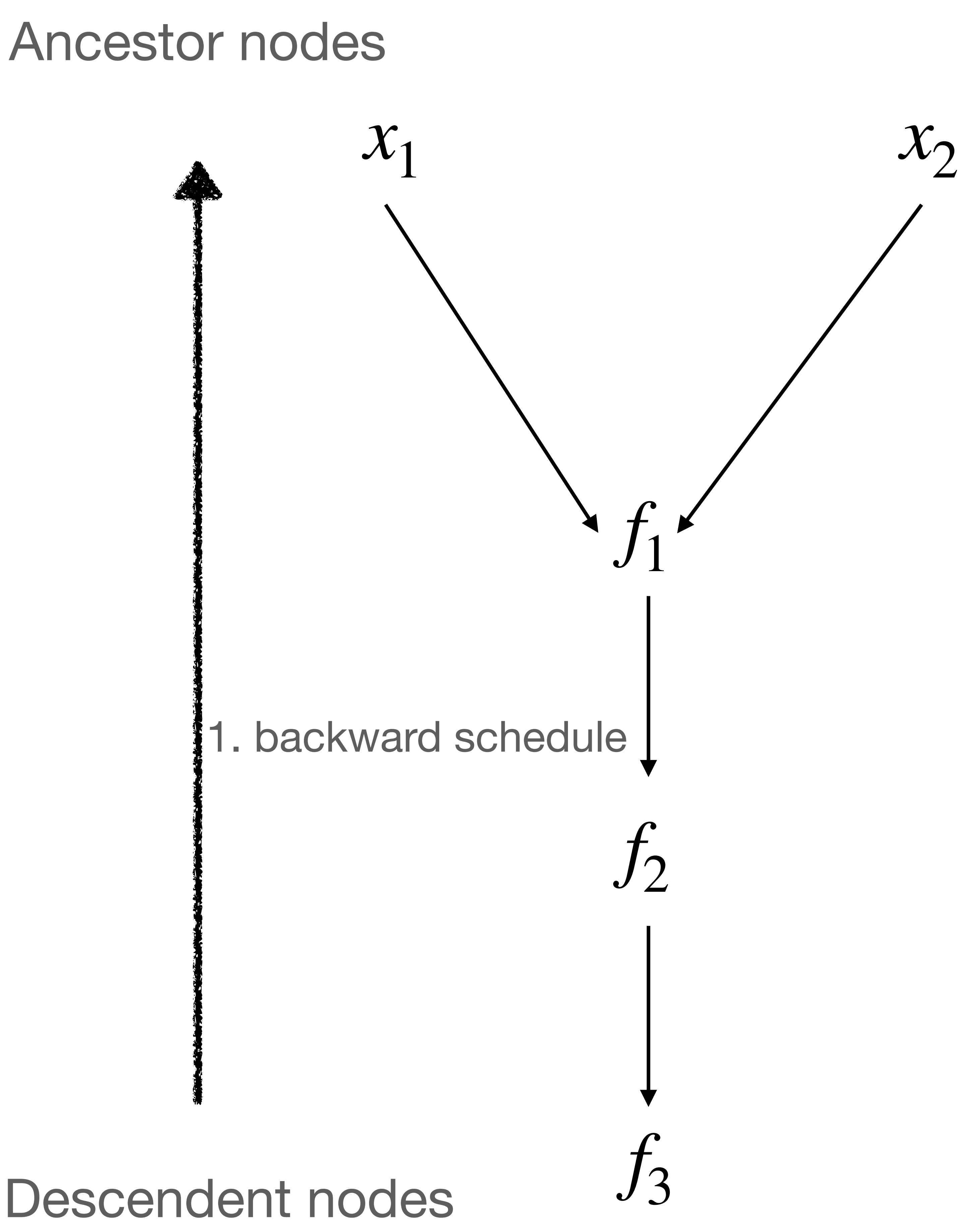
공통적으로 계산에 필요한 derivative을 **불필요하게 중복**되게 계산하는 것을 어떻게 **피할 수 있을까?**

→ 이것이 바로 Auto Differentiation의 핵심

출처: David Barber's Introduction to Deep Learning

Auto Differentiation

Auto Differentiation 예제



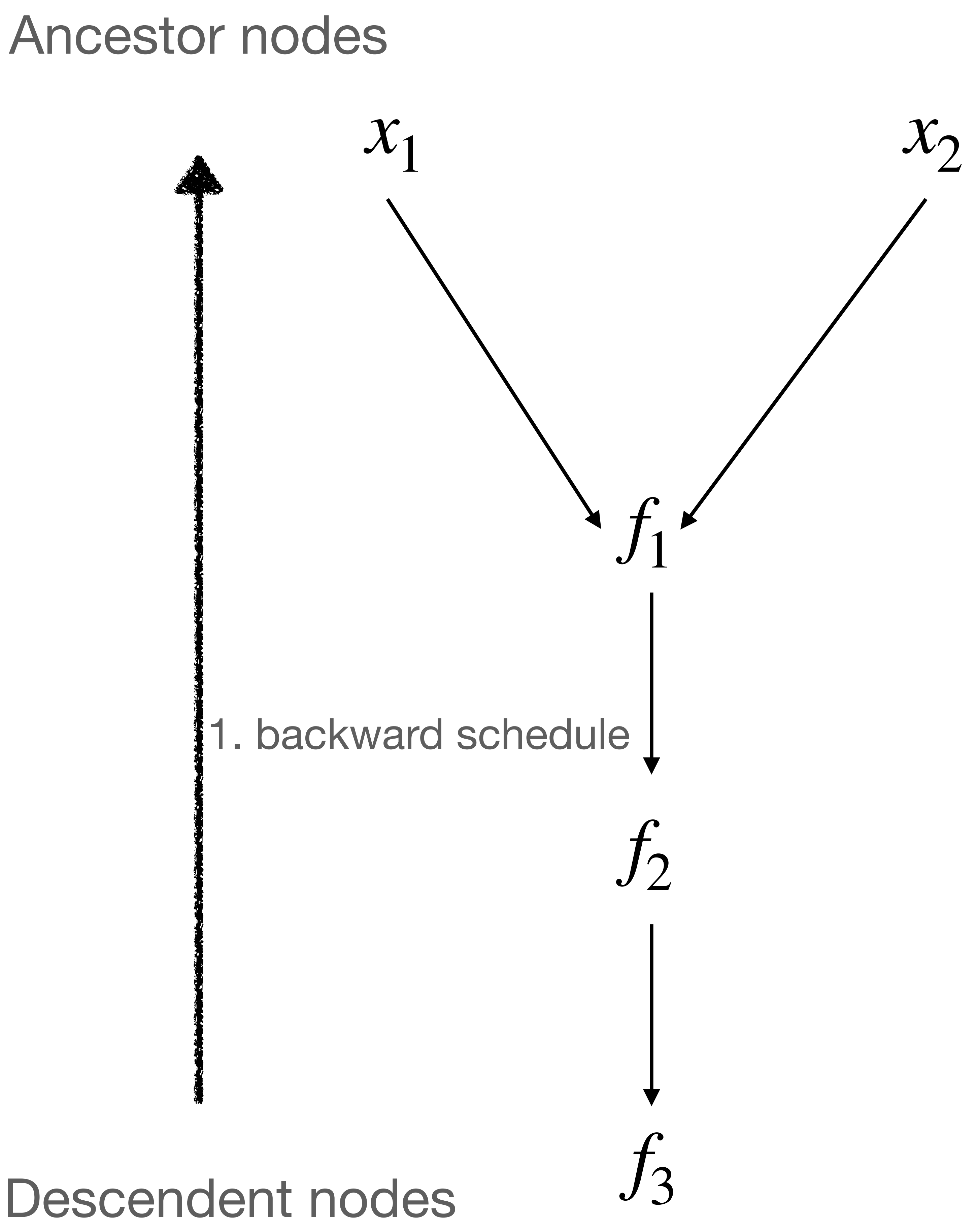
1. **Descendent (후손) node**으로부터 시작해서 **ancestor (선조) node**로 향하는 순서로 **reverse schedule**을 설정한다. 즉, "$f_3 \to f_2 \to f_1 \to x_1$" 와 "$f_3 \to f_2 \to f_1 \to x_1$"

후손 node은 뉴럴넷의 Output layer에 해당되고, 선조 node은 input layer라고 볼 수 있다.

출처: David Barber's Introduction to Deep Learning

Auto Differentiation

Auto Differentiation 예제



1. **Descendent (후손) node**으로부터 시작해서 **ancestor (선조) node**로 향하는 순서로 **reverse schedule**을 설정한다. 즉, “$f_3 \rightarrow f_2 \rightarrow f_1 \rightarrow x_1$” 와 “$f_3 \rightarrow f_2 \rightarrow f_1 \rightarrow x_1$”

2. Backward schedule의 첫번째 node n_1으로부터 시작하고 $t_{n_1} = 1$로 정의한다.

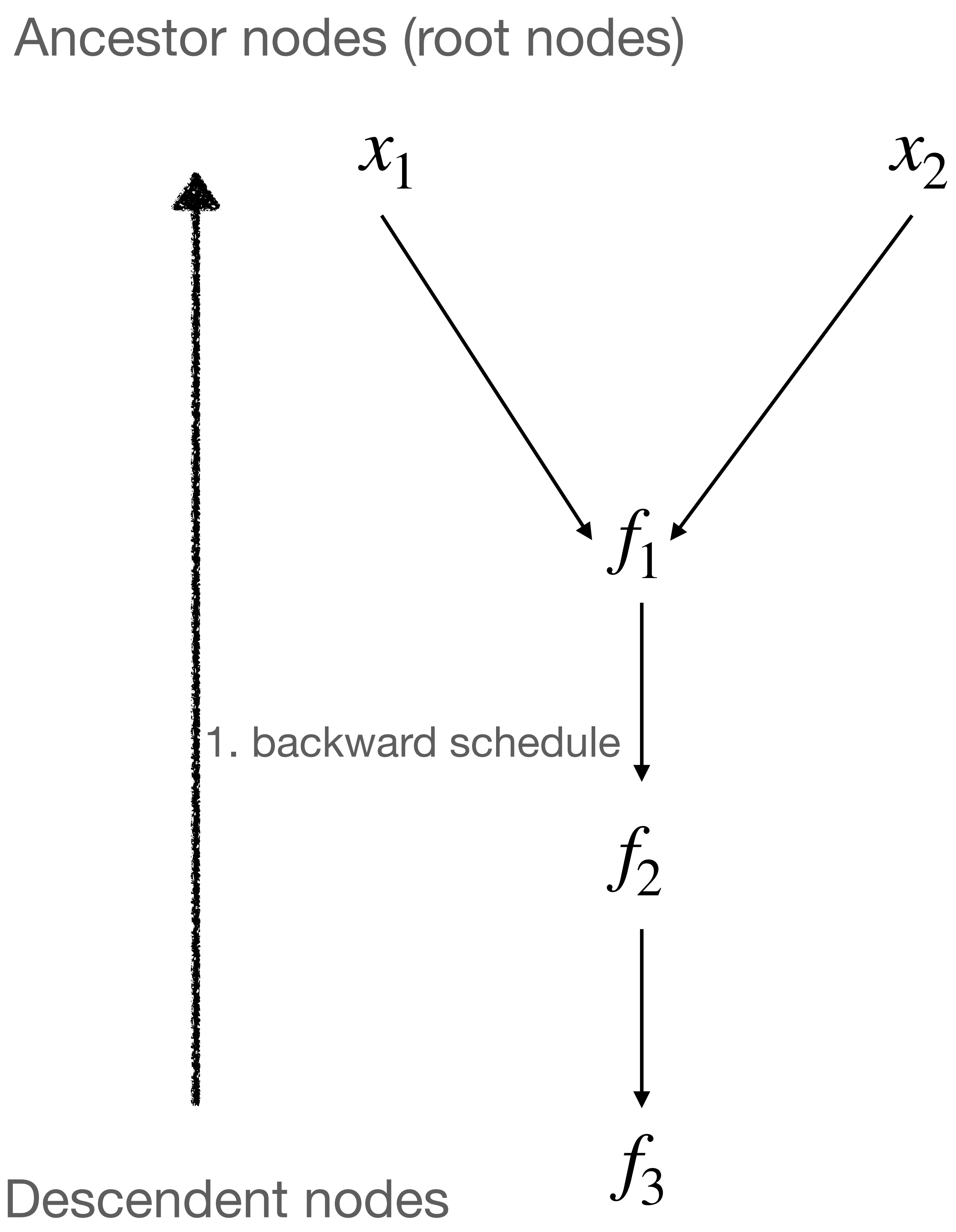
t_n은 descendent node을 node n에 대해서 미분한 값이다

출처: David Barber’s Introduction to Deep Learning

ACADENTIAL

Reverse Differentiation

Computational Graph



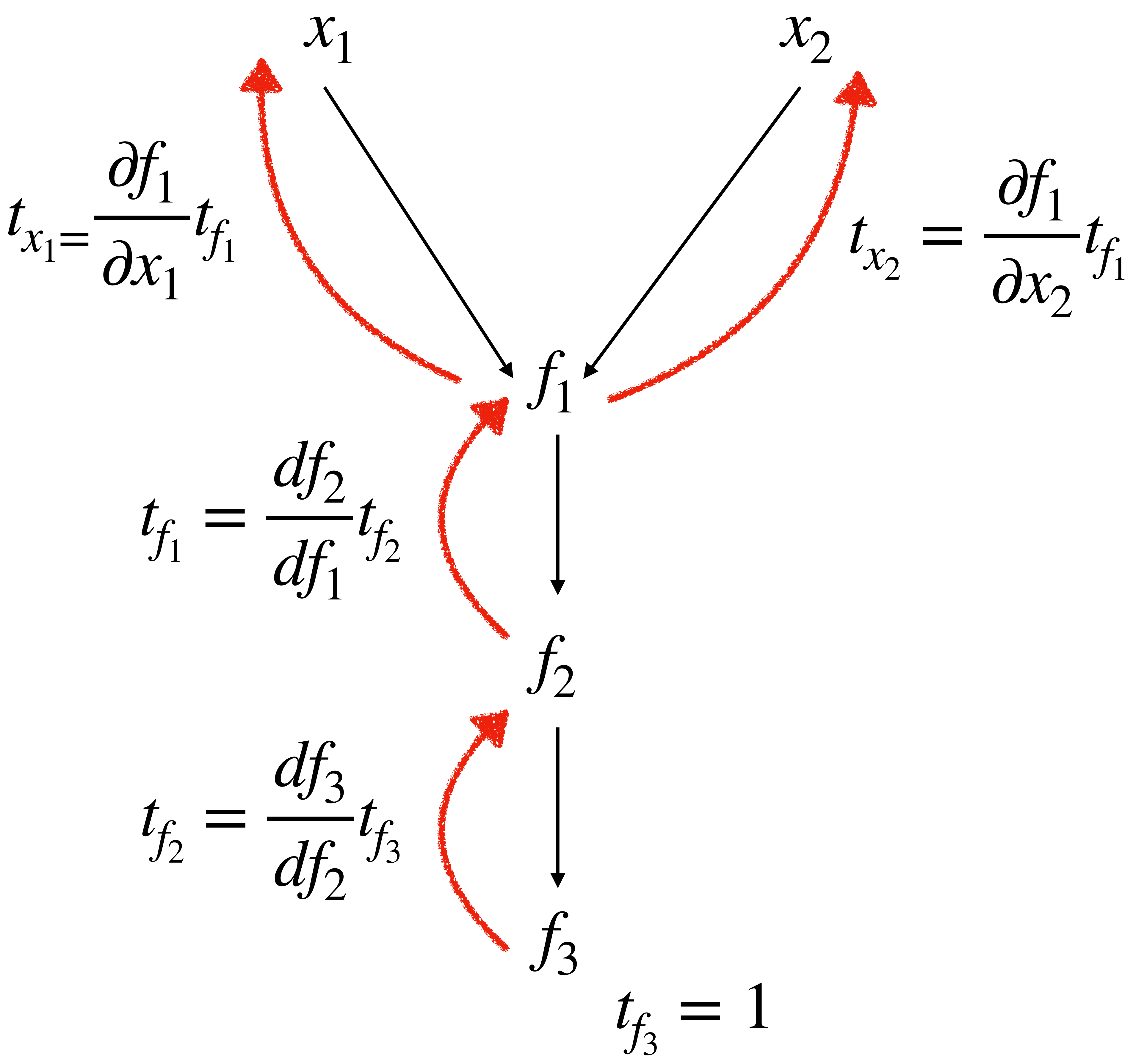
1. **Descendent (후손) node**으로부터 시작해서 **ancestor (선조) node**로 향하는 순서로 **reverse schedule**을 설정한다. 즉, "$f_3 \rightarrow f_2 \rightarrow f_1 \rightarrow x_1$" 와 "$f_3 \rightarrow f_2 \rightarrow f_1 \rightarrow x_1$"
2. Backward schedule의 첫번째 node n_1으로부터 시작하고 $t_{n_1} = 1$로 정의한다.
3. Reverse schedule에서 다음 node n에 대한 child node $ch(n)$을 찾아 다음 노드의 t_n을 다음과 같이 계산함:

$$t_n = \sum_{c \in ch(n)} \frac{\partial f_c}{\partial f_n} t_c$$

출처: David Barber's Introduction to Deep Learning

Reverse Differentiation

Computational Graph



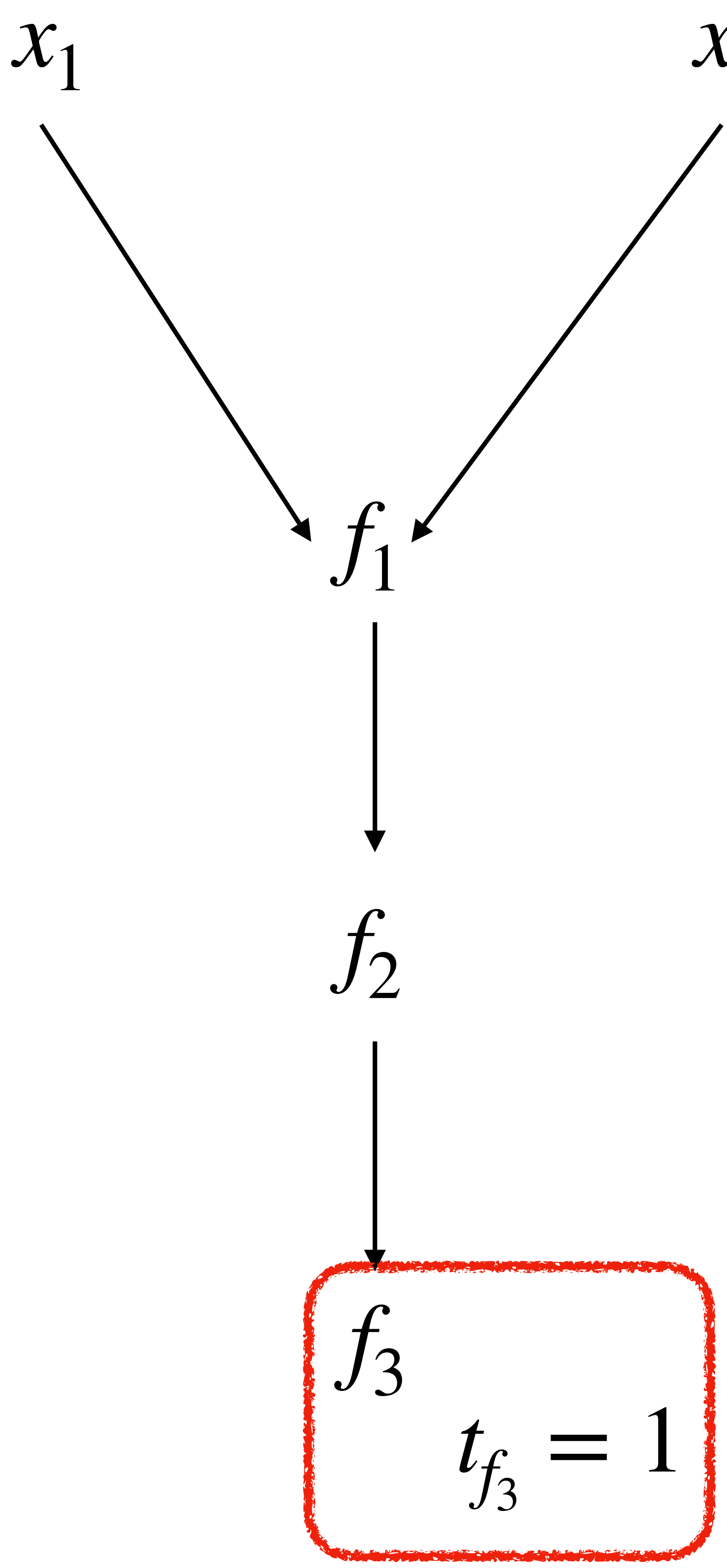
1. **Descendent (후손) node**으로부터 시작해서 **ancestor (선조) node**로 향하는 순서로 **reverse schedule**을 설정한다. 즉, “$f_3 \rightarrow f_2 \rightarrow f_1 \rightarrow x_1$” 와 “$f_3 \rightarrow f_2 \rightarrow f_1 \rightarrow x_1$”
2. Backward schedule의 첫번째 node n_1으로부터 시작하고 $t_{n_1} = 1$로 정의한다.
3. Reverse schedule에서 다음 node n에 대한 child node $ch(n)$을 찾아 다음 노드의 t_n을 다음과 같이 계산함:

$$t_n = \sum_{c \in ch(n)} \frac{\partial f_c}{\partial f_n} t_c$$

출처: David Barber’s Introduction to Deep Learning

Auto Differentiation

Auto Differentiation 예제



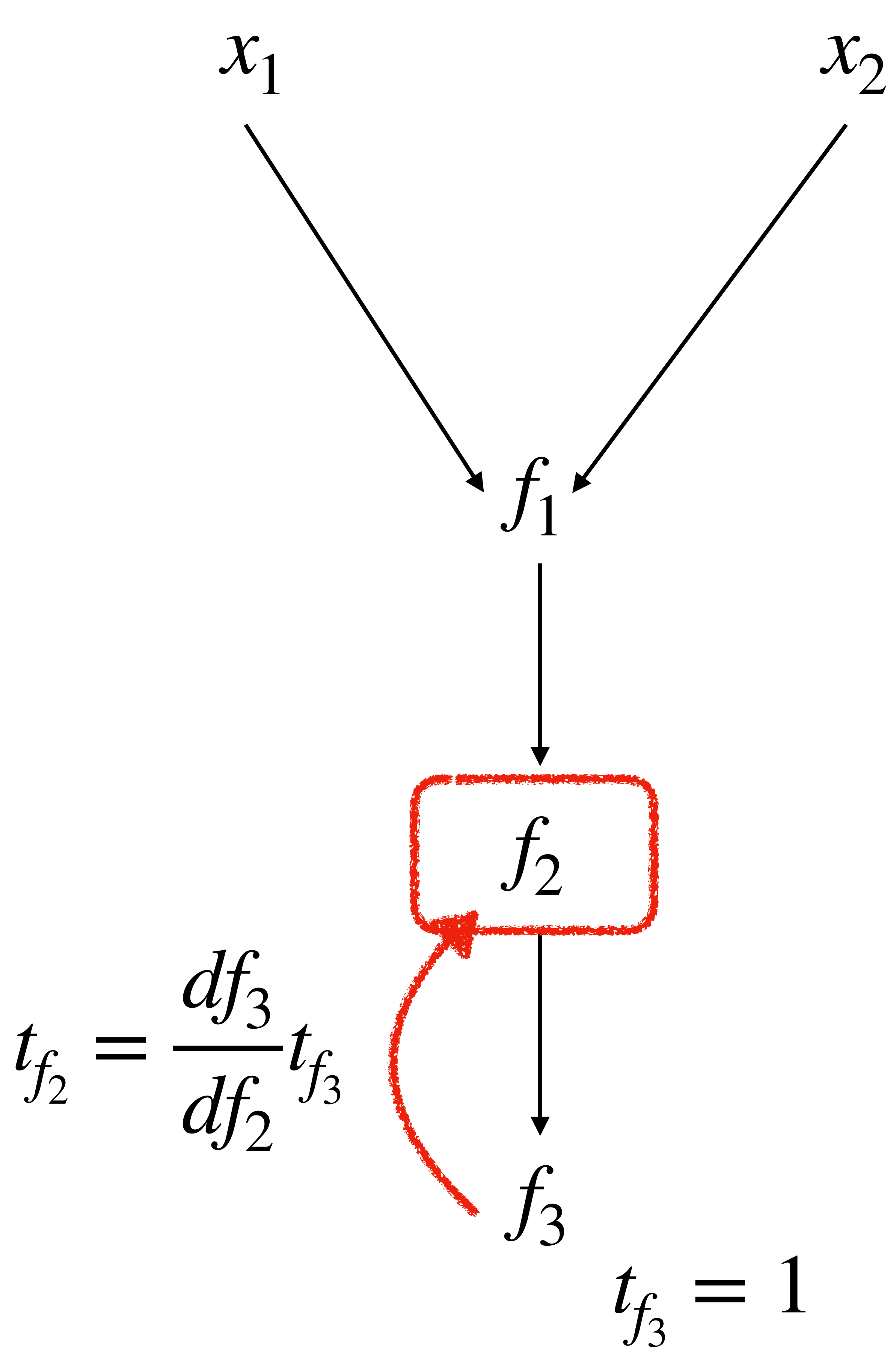
Reverse schedule: $\boxed{f_3} \rightarrow f_2 \rightarrow f_1 \rightarrow x_1$

1. **Ancestor node f_3으로 부터 시작**

출처: David Barber's Introduction to Deep Learning

Auto Differentiation

Auto Differentiation 예제



Reverse schedule: $f_3 \rightarrow \boxed{f_2} \rightarrow f_1 \rightarrow x_1$

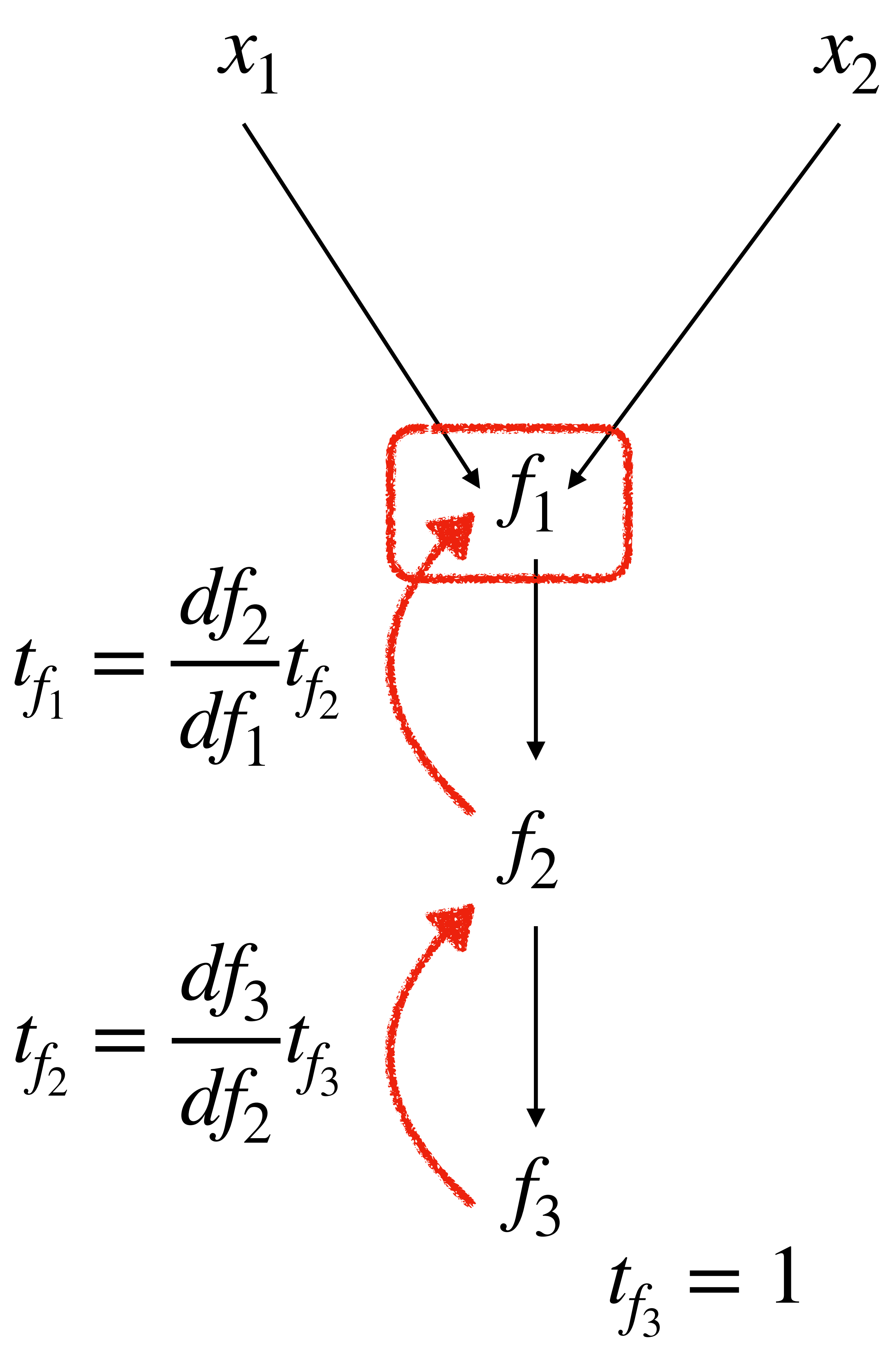
1. Ancestor node f_3으로 부터 시작
2. **다음 노드인 f_2에 대한 t_{f_2}을 구한다.**

출처: David Barber's Introduction to Deep Learning

ACADENTIAL

Auto Differentiation

Auto Differentiation 예제



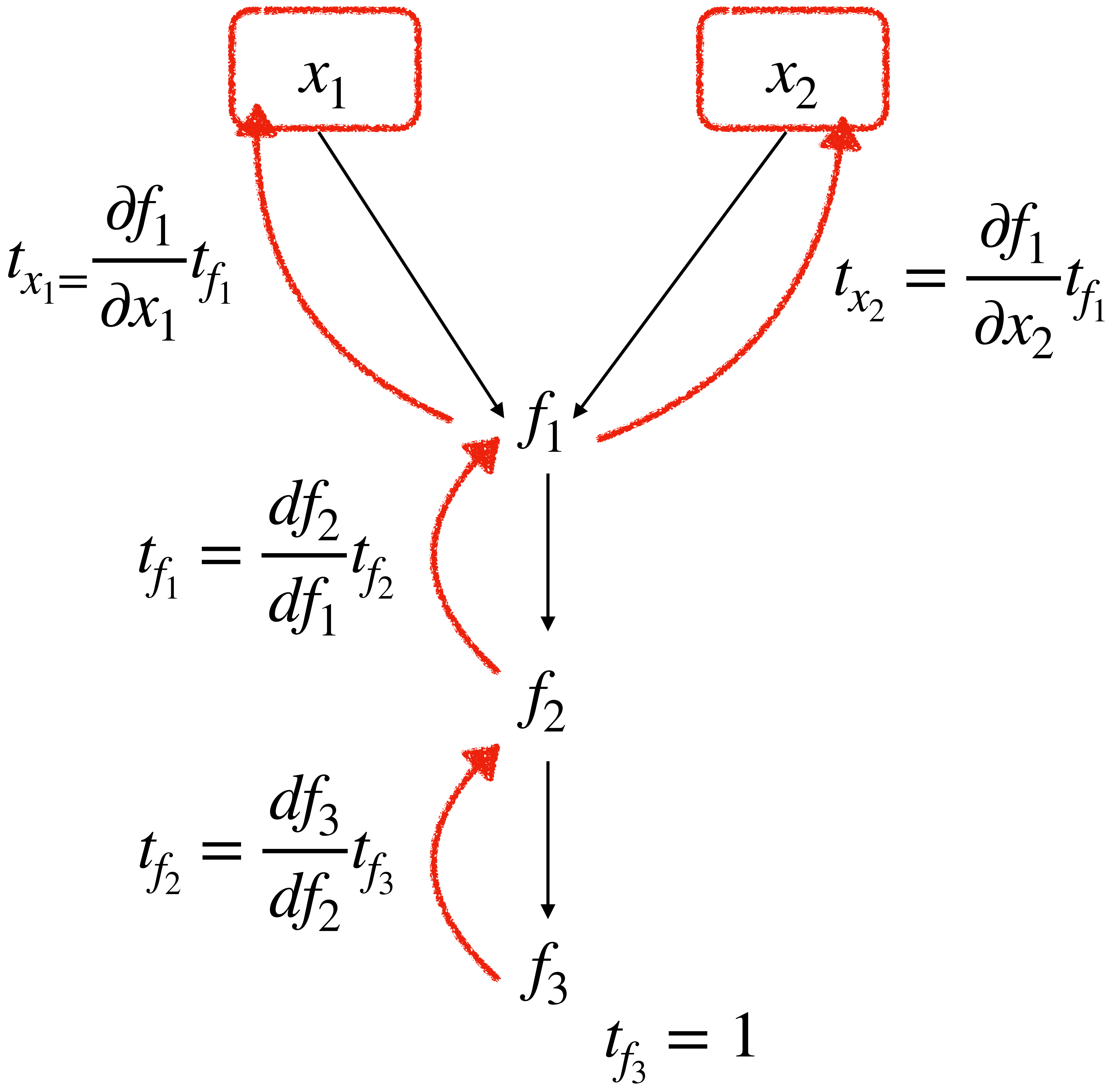
Reverse schedule: $f_3 \rightarrow f_2 \rightarrow \boxed{f_1} \rightarrow x_1$

1. Ancestor node f_3으로 부터 시작
2. f_2에 대한 t_{f_2}을 구한다.
3. **다음 노드인 f_1에 대한 t_{f_1}을 구한다.**

출처: David Barber's Introduction to Deep Learning

Auto Differentiation

Auto Differentiation 예제



Reverse schedule: $f_3 \rightarrow f_2 \rightarrow f_1 \rightarrow \boxed{x_1, x_2}$

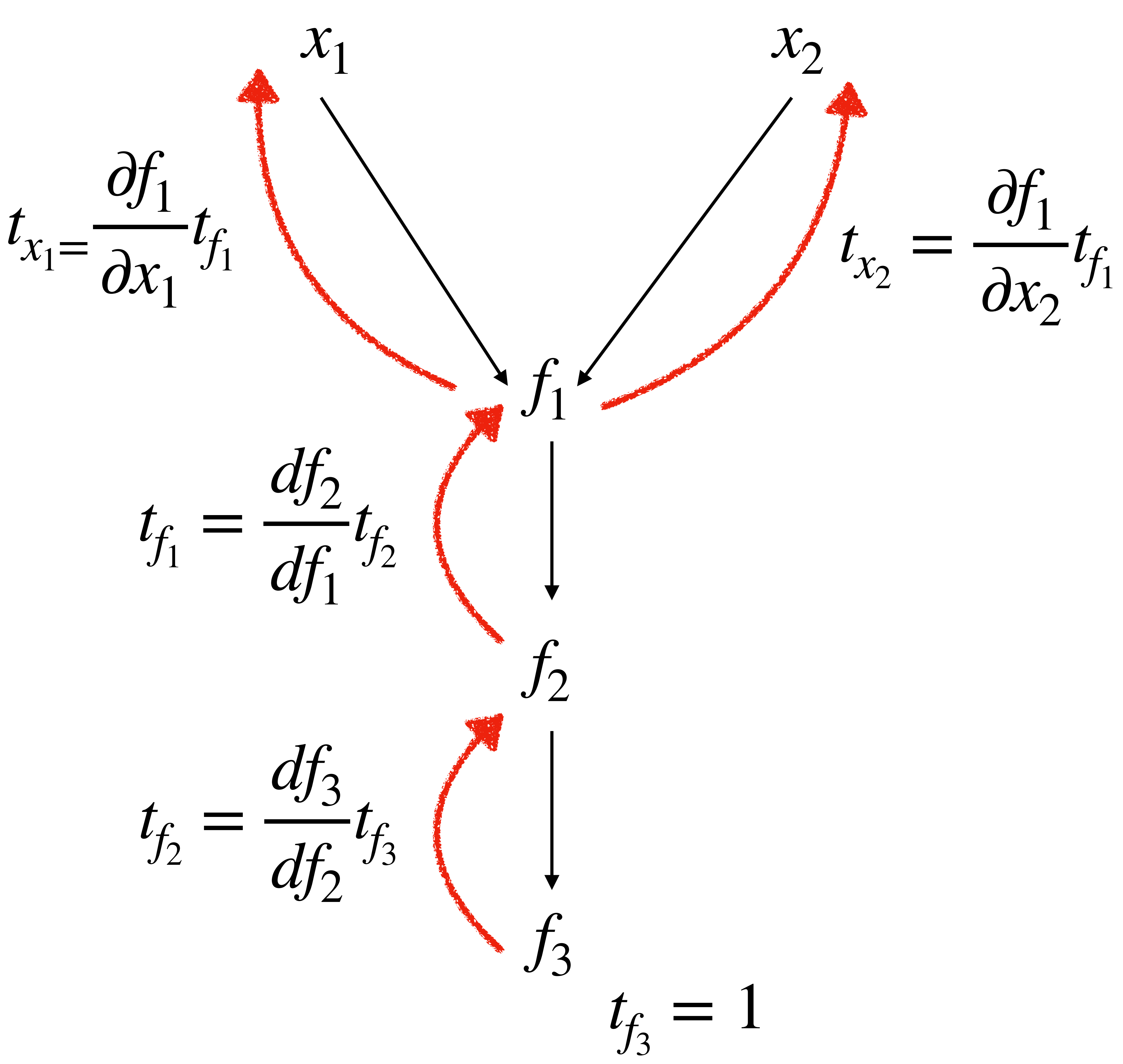
1. Ancestor node f_3으로 부터 시작
2. f_2에 대한 t_{f_2}을 구한다.
3. f_1에 대한 t_{f_1}을 구한다.
4. Root node x_1과 x_2각각에 대해서 t_{x_1}, t_{x_2}을 구한다.

그러면 최종적으로 구하고 싶은 $\frac{\partial f}{\partial x_1} = t_{x_1}$ 와 $\frac{\partial f}{\partial x_2} = t_{x_2}$이다!

출처: David Barber's Introduction to Deep Learning

Auto Differentiation

Auto Differentiation 예제



3. Reverse schedule에서 다음 node n에 대한 child node $ch(n)$을 찾아 다음 노드의 t_n을 다음과 같이 계산함:

$$t_n = \sum_{c \in ch(n)} \frac{\partial f_c}{\partial f_n} t_c$$

4. 최종적으로 구하고 싶은 $\frac{\partial f}{\partial x_1} = t_{x_1}$ 와 $\frac{\partial f}{\partial x_2} = t_{x_2}$이다!

출처: David Barber's Introduction to Deep Learning

Auto Differentiation

Auto Differentiation 예제

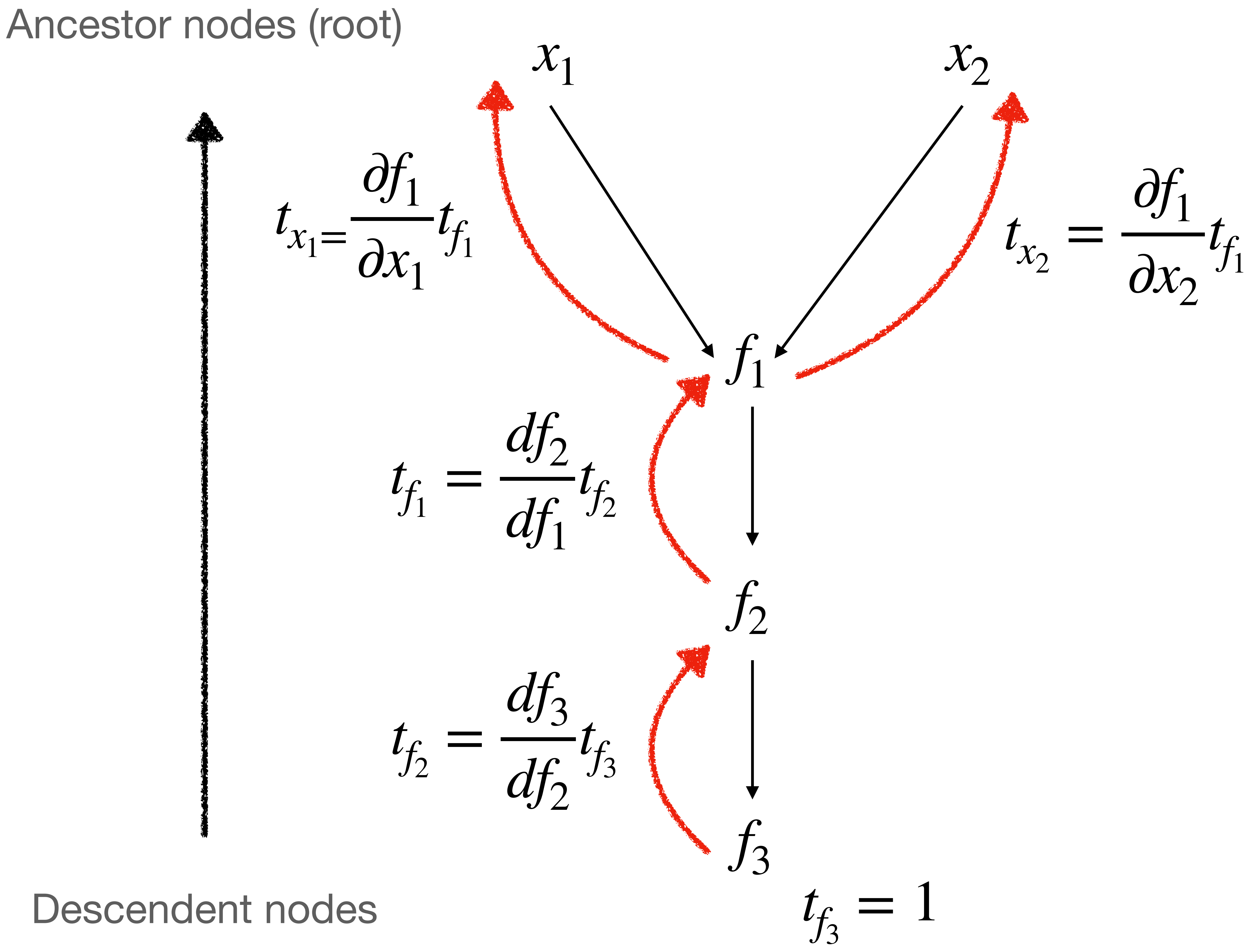
1. **Descendent (후손) node**으로부터 시작해서 **ancestor (선조) node**로 향하는 순서로 **reverse schedule**을 설정한다. 즉, "$f_3 \to f_2 \to f_1 \to x_1, x_2$" 와

2. Backward schedule의 첫번째 node n_1 (즉, f_3) 으로부터 시작하고 $t_{n_1} = 1$로 정의한다.

3. Reverse schedule에서 다음 node n에 대한 child node $ch(n)$을 찾아 다음 노드의 t_n을 다음과 같이 계산함:

$$t_n = \sum_{c \in ch(n)} \frac{\partial f_c}{\partial f_n} t_c$$

4. 최종적으로 구하고 싶은 $\dfrac{\partial f}{\partial x_1} = t_{x_1}$ 와 $\dfrac{\partial f}{\partial x_2} = t_{x_2}$이다!



출처: David Barber's Introduction to Deep Learning

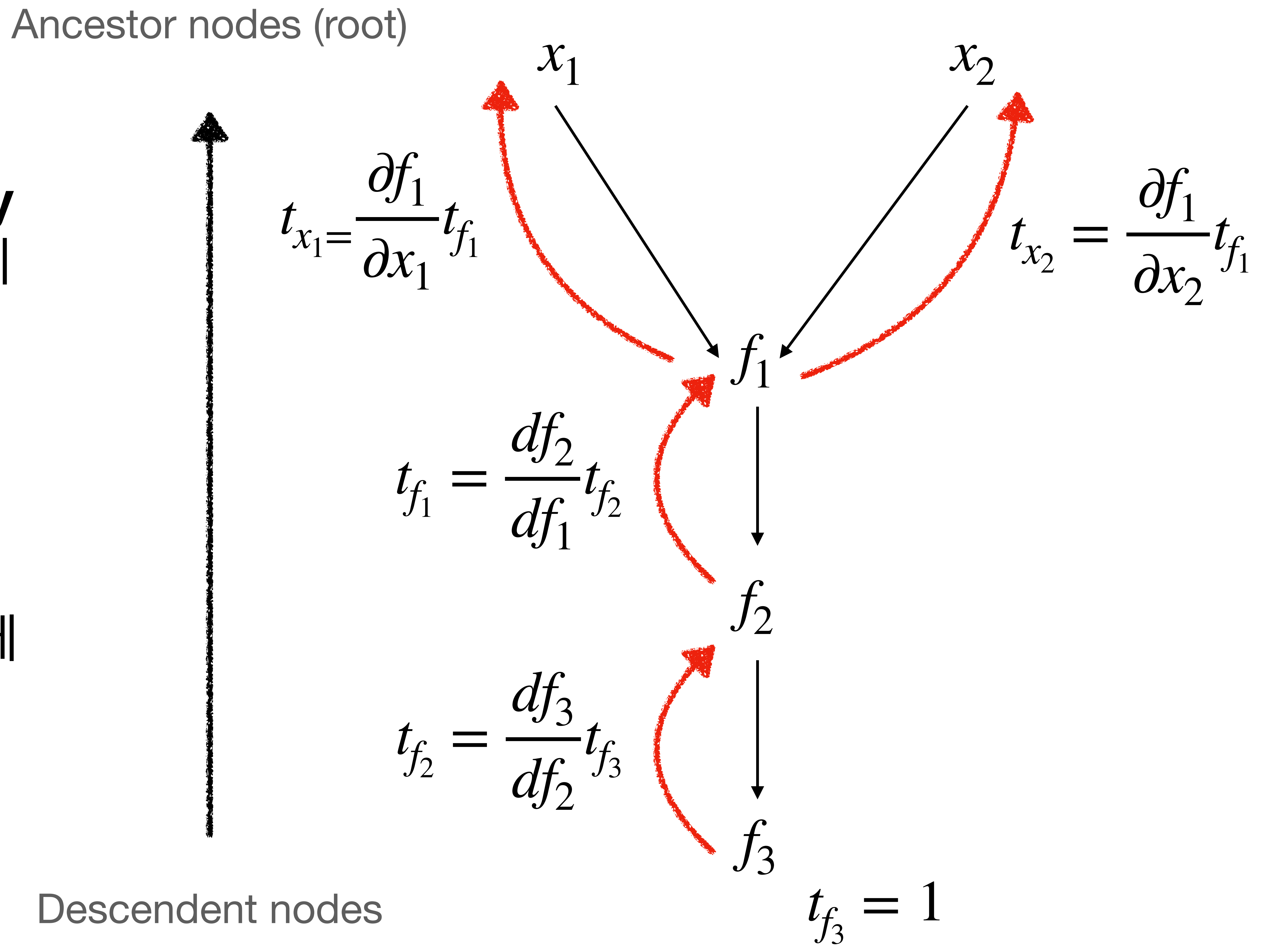
Auto Differentiation

Auto Differentiation 예제

이것이 바로 **PyTorch, Tensorflow** 등등의 **딥러닝 Library 들의 Backward Propagation**의 **핵심개념**이자 **작동원리**이다!

모두 **Reverse Differentiation을 활용**해서 Gradient Descent에 필요한 Gradient을 계산하는 것이다!

즉, $\mathbf{w} = (w_0, w_1, \cdots, w_N)$인 어떤 임의의 합성함수 $L(\mathbf{w})$에 대해서 우리는 $\dfrac{dL}{dw_i}$을 구할 수 있는 것이다!



출처: David Barber's Introduction to Deep Learning