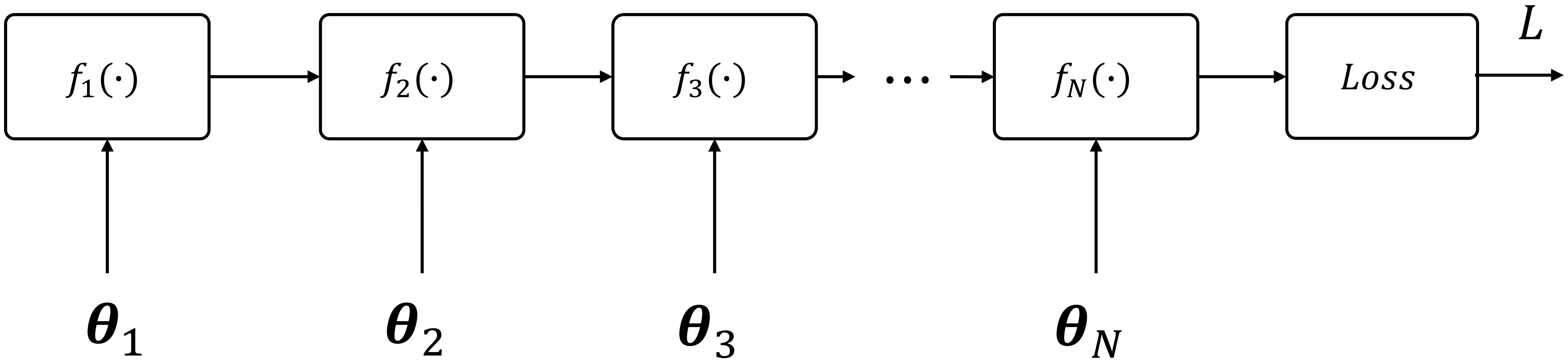
심층신경망의 연쇄 법칙



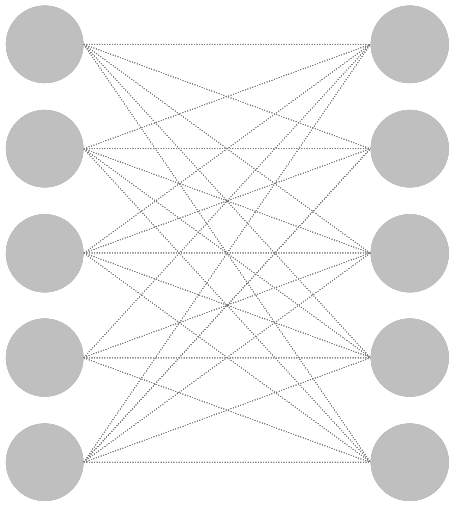
$$f_n(\boldsymbol{\theta}_n) = a(W_n \boldsymbol{h}_{n-1} + \boldsymbol{b}_n)$$

$$\boldsymbol{\theta}_n = [vec(W_n)^T | \boldsymbol{b}_n^T]^T$$

$$\boxed{\frac{\partial L}{\partial \boldsymbol{\theta}_n} = \frac{\partial L}{\partial \boldsymbol{h}_N} \prod_{i=n+1}^{N} \frac{\partial f_i(\boldsymbol{h}_{i-1})}{\partial \boldsymbol{h}_{i-1}} \frac{\partial f_n(\boldsymbol{\theta}_n)}{\partial \boldsymbol{\theta}_n}}$$

미분하고자 하는 경로 사이에 있는 모든 미분 값을 곱하면 원하는 미분을 구할 수 있다.

전결합 계층의 미분 (1)



$$\boldsymbol{h}_n = a(W_n \boldsymbol{h}_{n-1} + \boldsymbol{b}_n)$$

$$\boxed{\hat{\boldsymbol{h}}_n = W_n \boldsymbol{h}_{n-1} + \boldsymbol{b}_n}$$

$$\hat{h}_n^j = \sum_i w_{n-1}^i h_{n-1}^i + b_{n-1}^j$$

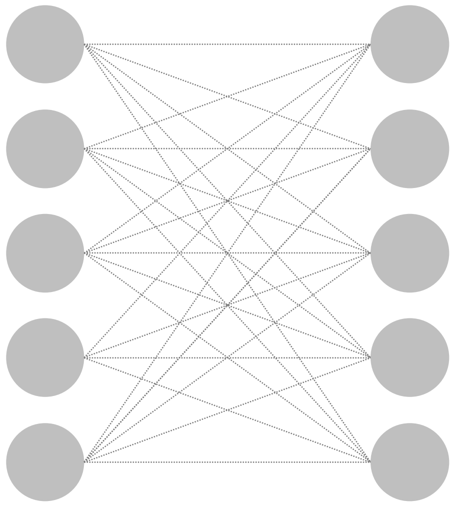
$k = i$인 항만 살아남는다.

$$\left\{\begin{aligned}
\frac{\partial \hat{h}_n^j}{\partial h_{n-1}^i} &= \frac{\partial\left[\sum_k w_{n-1}^{k,j} h_{n-1}^k + \cancel{b_{n-1}^j}\right]}{\partial h_{n-1}^i} = \boxed{w_{n-1}^{i,j}} \\
\frac{\partial \hat{h}_n^j}{\partial w_{n-1}^{i,j}} &= \frac{\partial\left[\sum_k w_{n-1}^{k,j} h_{n-1}^k + \cancel{b_{n-1}^j}\right]}{\partial w_{n-1}^{i,j}} = \boxed{h_{n-1}^i} \\
\frac{\partial \hat{h}_n^j}{\partial b_{n-1}^j} &= \frac{\partial\left[\sum_k \cancel{w_{n-1}^{k,j} h_{n-1}^k} + b_{n-1}^j\right]}{\partial b_{n-1}^j} = \boxed{1}
\end{aligned}\right.$$

전결합 계층의 미분 (2)

Activation function의 미분

$$
\begin{cases}
\dfrac{\partial h_n^j}{\partial h_{n-1}^i} = \dfrac{\partial a^j}{\partial \hat{h}_n^j}\dfrac{\partial \hat{h}_n^j}{\partial h_{n-1}^i} = \boxed{\dfrac{\partial a^j}{\partial \hat{h}_n^j}}\boxed{w_n^{i,j}} \\[2ex]
\dfrac{\partial h_n^j}{\partial w_n^{i,j}} = \dfrac{\partial a^j}{\partial \hat{h}_n^j}\dfrac{\partial \hat{h}_n^j}{\partial w_n^{i,j}} = \boxed{\dfrac{\partial a^j}{\partial \hat{h}_n^j}}\boxed{h_{n-1}^i} \\[2ex]
\dfrac{\partial h_N^j}{\partial b_N^i} = \dfrac{\partial a^j}{\partial \hat{h}_N^j}\dfrac{\partial \hat{h}_N^j}{\partial b_N^i} = \boxed{\dfrac{\partial a^j}{\partial \hat{h}_N^j}}
\end{cases}
$$

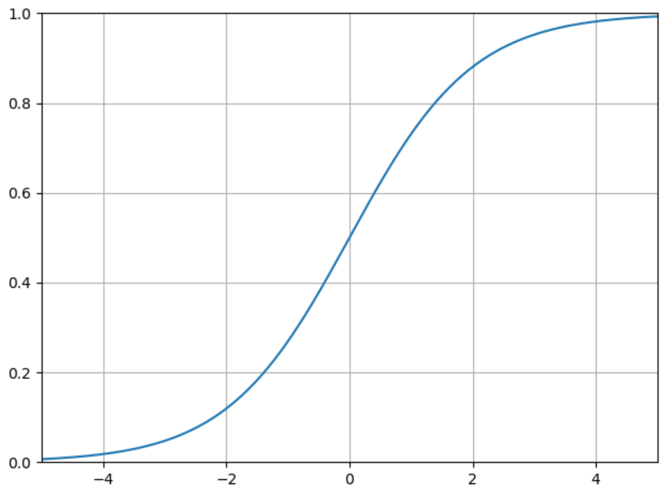


$$\boldsymbol{h}_n = a(W_n \boldsymbol{h}_{n-1} + \boldsymbol{b}_n)$$

$$\widehat{\boldsymbol{h}}_n = W_n \boldsymbol{h}_{n-1} + \boldsymbol{b}_n$$

Activation function의 미분을 알고 있다면, 쉽게 Fully-connected layer를 미분할 수 있다.

Sigmoid 함수의 미분

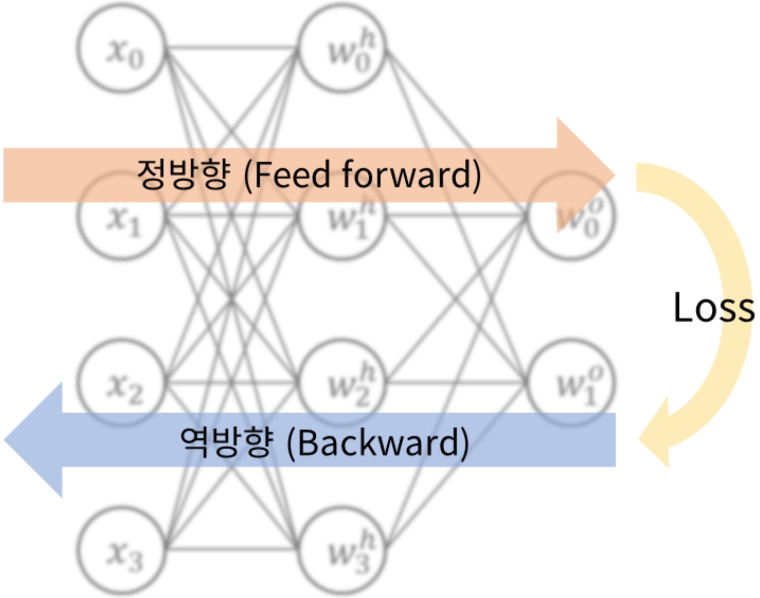


$$\mathrm{sigmoid}(x) = \frac{1}{1+e^{-x}}$$

$$\begin{aligned}
\frac{d}{dx} sigmoid(x) &= \frac{d}{dx}(1+e^{-x})^{-1} \\
&= (-1)\frac{1}{(1+e^{-x})^2}\frac{d}{dx}(1+e^{-x}) \\
&= (-1)\frac{1}{(1+e^{-x})^2}e^{-x}\frac{d}{dx}(-x) \\
&= (-1)\frac{1}{(1+e^{-x})^2}e^{-x}(-1) \\
&= \frac{e^{-x}}{(1+e^{-x})^2} \\
&= \frac{1+e^{-x}-1}{(1+e^{-x})^2} \\
&= \frac{1}{1+e^{-x}} - \frac{1}{(1+e^{-x})^2} \\
&= sigmoid(x) - sigmoid(x)^2 \\
&= \underline{sigmoid(x)\big(1 - sigmoid(x)\big)}
\end{aligned}$$

초창기 신경망에 가장 많이 쓰인 Sigmoid 활성 함수의 미분. 정리된 결과를 이용하면 매우 간단하게 미분할 수 있다.

역전파 알고리즘



오류 역전파 알고리즘
(Back-Propagation Algorithm; BP)

- 학습 데이터로 정방향 연산을 하여 Loss를 구한다.
- 정방향 연산 시, 계층별로 BP에 필요한 중간 결과를 저장한다.
- Loss를 각 파라미터로 미분한다. 연쇄 법칙(역방향 연산)을 이용한다.
  - 마지막 계층부터 하나씩 이전 계층으로 연쇄적으로 계산한다.
  - 역방향 연산 시, 정방향 연산에서 저장한 중간 결과를 사용한다.

미분의 연쇄 법칙과 각 함수의 수식적 미분을 이용하면, 단 한번의 손실함수 평가로 미분을 구할 수 있다.
단, 중간 결과를 저장해야 하므로 메모리를 추가로 사용한다.