Softmax Classifier 및 Logistic Regression의 Backpropagation

수업 목표

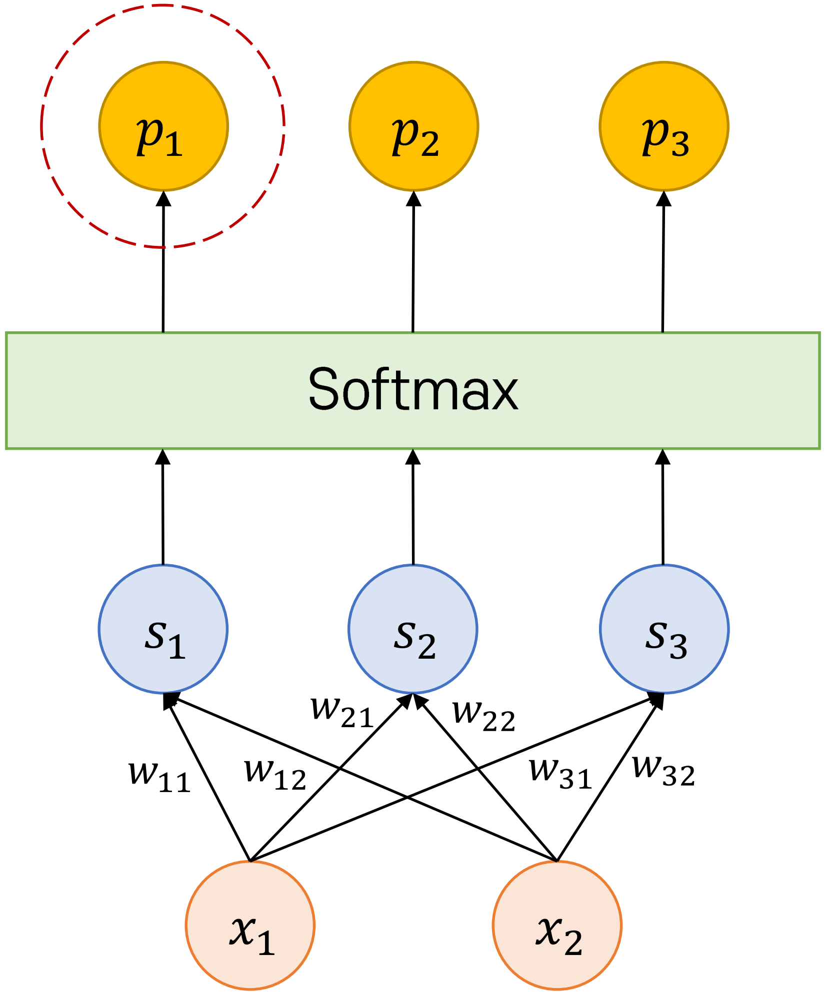
이번 수업의 핵심:

- Softmax classifier의 Backpropagation
- Logistic regression의 Backpropagation
- Backpropagation을 통한 Neural network 학습

핵심 개념

- Softmax classifier 및 Logistic regression의 Backpropagation
- Neural network 학습

Softmax Classifier의 Backpropagation



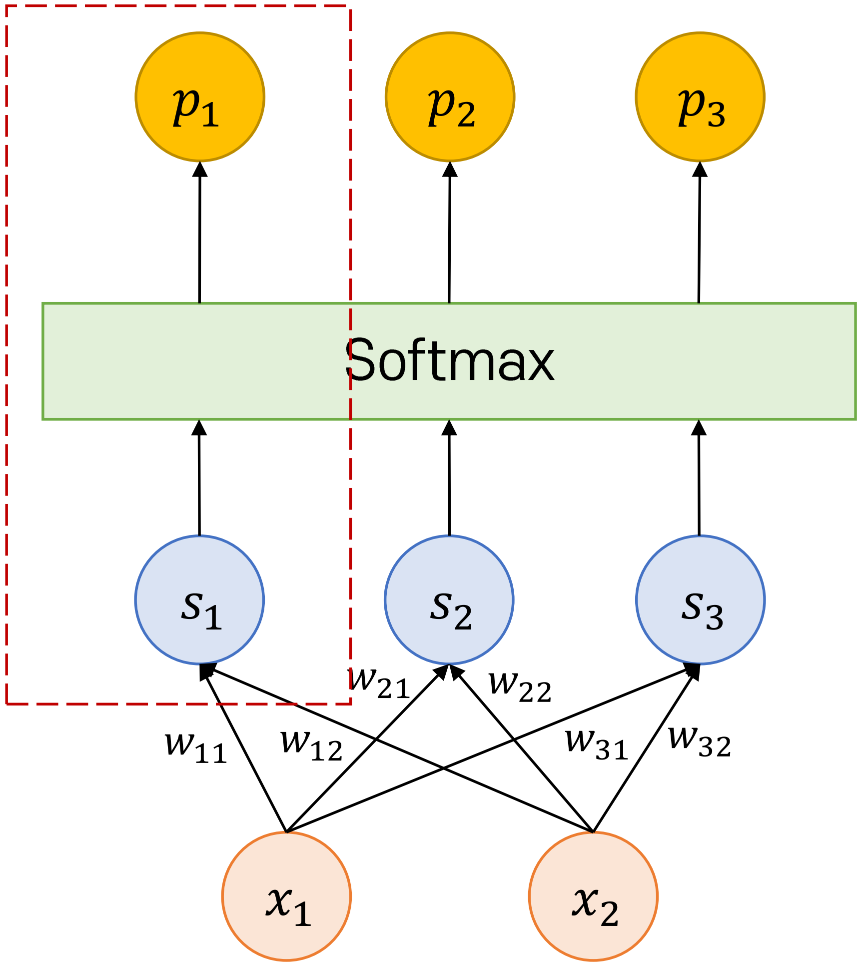
Cross Entropy Loss:

$$\mathcal{L} = -\sum_{c=1}^{3} y_c \log p_c$$

$$\text{정답 } \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \text{이라 가정}$$

$$\mathcal{L} = -\log p_1 = -\log\left(\frac{e^{s_1}}{e^{s_1}+e^{s_2}+e^{s_3}}\right)$$

Softmax Classifier의 Backpropagation



s_1의 Gradient:

$$\mathcal{L} = -\log\left(\frac{e^{s_1}}{e^{s_1}+e^{s_2}+e^{s_3}}\right)$$

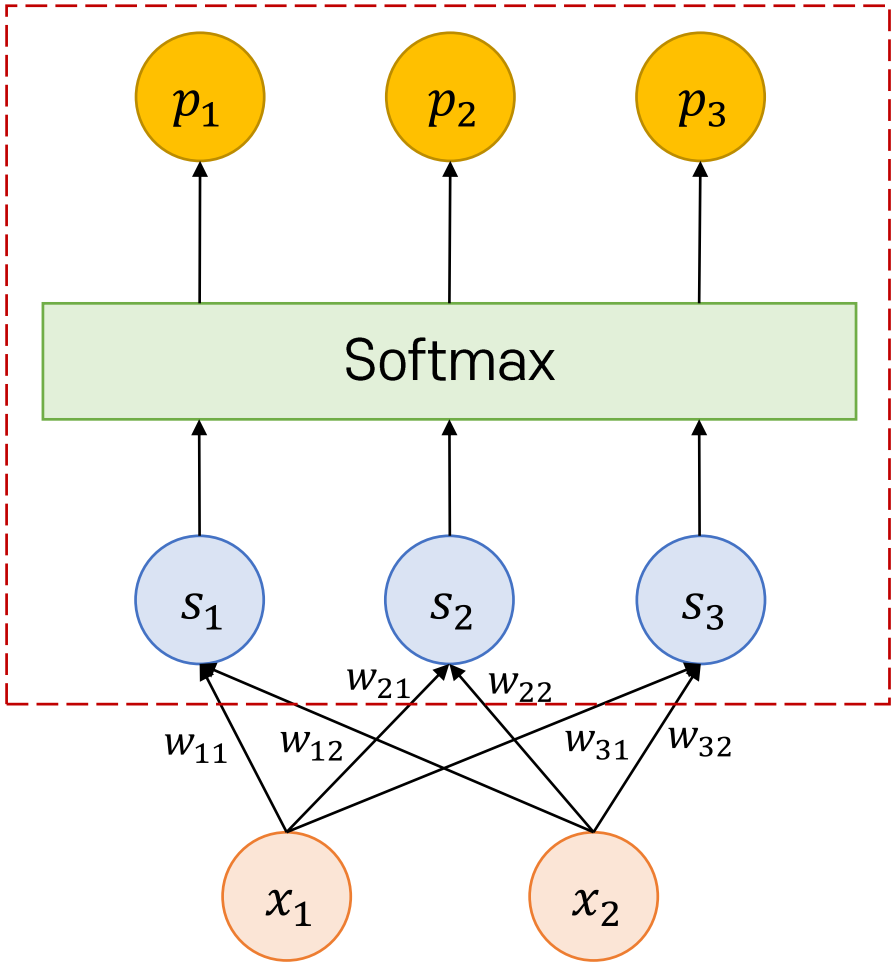
$$\frac{\partial \mathcal{L}}{\partial s_1} = -\frac{\frac{\partial}{\partial s_1}\left(\frac{e^{s_1}}{e^{s_1}+e^{s_2}+e^{s_3}}\right)}{\frac{e^{s_1}}{e^{s_1}+e^{s_2}+e^{s_3}}}$$

$$= -\frac{\frac{e^{s_1}(e^{s_1}+e^{s_2}+e^{s_3})-e^{s_1}e^{s_1}}{(e^{s_1}+e^{s_2}+e^{s_3})^2}}{\frac{e^{s_1}}{e^{s_1}+e^{s_2}+e^{s_3}}}$$

$$= -\frac{e^{s_2}+e^{s_3}}{e^{s_1}+e^{s_2}+e^{s_3}}$$

$$= p_1 - 1$$

Softmax Classifier의 Backpropagation



$\mathbf{s}$의 Gradient:

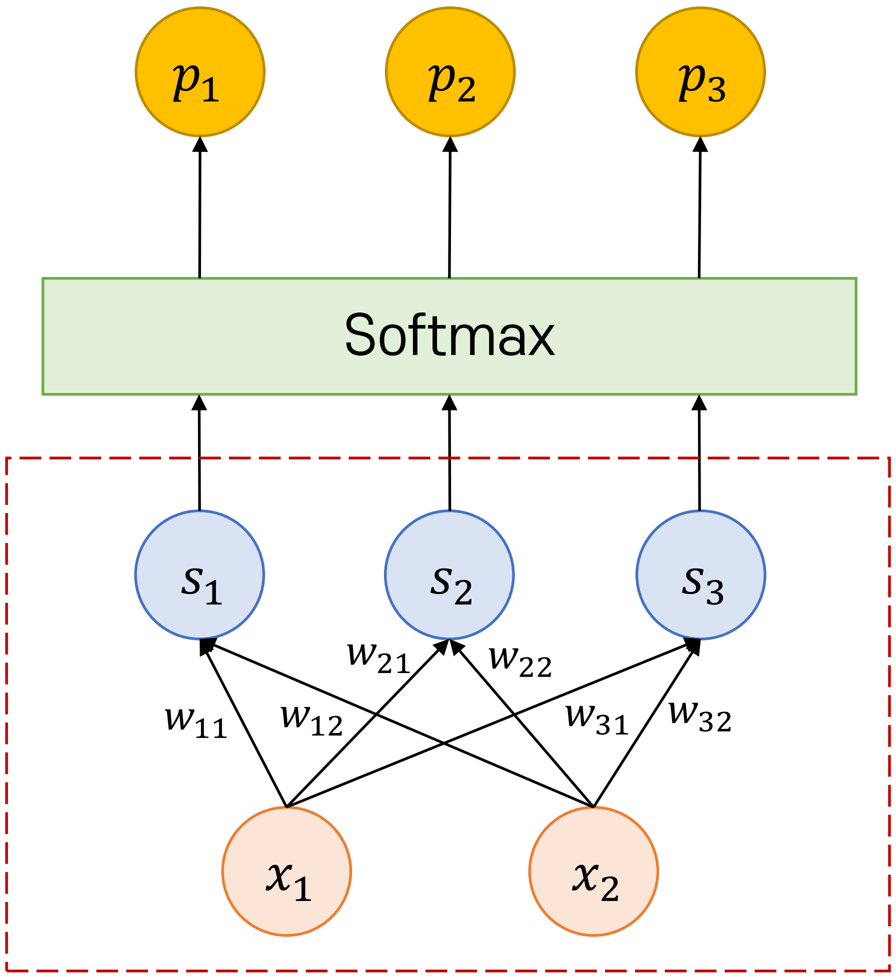
$$\frac{\partial \mathcal{L}}{\partial s_2} = \frac{e^{s_2}}{e^{s_1}+e^{s_2}+e^{s_3}} = p_2 - 0$$

$$\frac{\partial \mathcal{L}}{\partial s_3} = \frac{e^{s_3}}{e^{s_1}+e^{s_2}+e^{s_3}} = p_3 - 0$$

$$\frac{\partial \mathcal{L}}{\partial s_1} = p_1 - 1, \qquad \frac{\partial \mathcal{L}}{\partial s_2} = p_2, \qquad \frac{\partial \mathcal{L}}{\partial s_3} = p_3$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{s}} = \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial s_1} \\ \frac{\partial \mathcal{L}}{\partial s_2} \\ \frac{\partial \mathcal{L}}{\partial s_3} \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \mathbf{p} - \mathbf{y}$$

Softmax Classifier의 Backpropagation



W의 Gradient:

$$\frac{\partial \mathcal{L}}{\partial s_1} = p_1 - 1, \qquad \frac{\partial \mathcal{L}}{\partial s_2} = p_2, \qquad \frac{\partial \mathcal{L}}{\partial s_3} = p_3$$

$$s_1 = w_{11}x_1 + w_{12}x_2$$

$$s_2 = w_{21}x_1 + w_{22}x_2$$

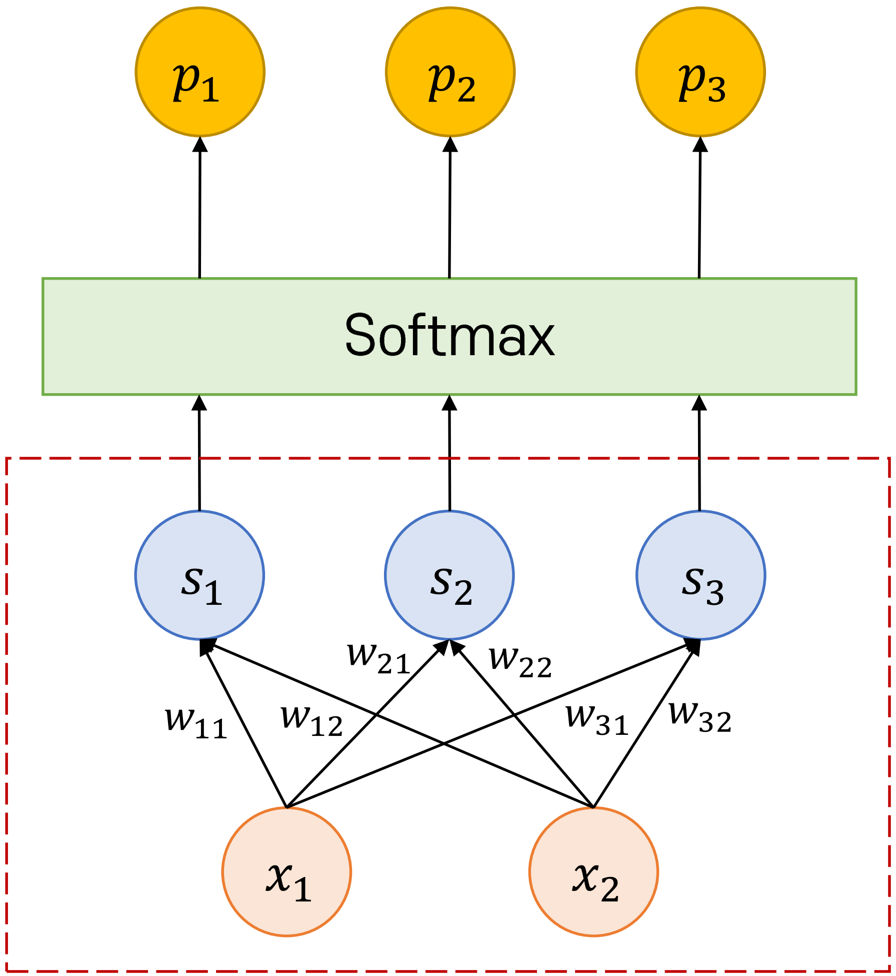
$$s_3 = w_{31}x_1 + w_{32}x_2$$

$$\frac{\partial \mathcal{L}}{\partial w_{11}} = \frac{\partial \mathcal{L}}{\partial s_1}\frac{\partial s_1}{\partial w_{11}} = (p_1 - 1)x_1$$

$$\frac{\partial \mathcal{L}}{\partial w_{21}} = \frac{\partial \mathcal{L}}{\partial s_2}\frac{\partial s_2}{\partial w_{21}} = p_2 x_1$$

$$\vdots$$

Softmax Classifier의 Backpropagation



W의 Gradient:

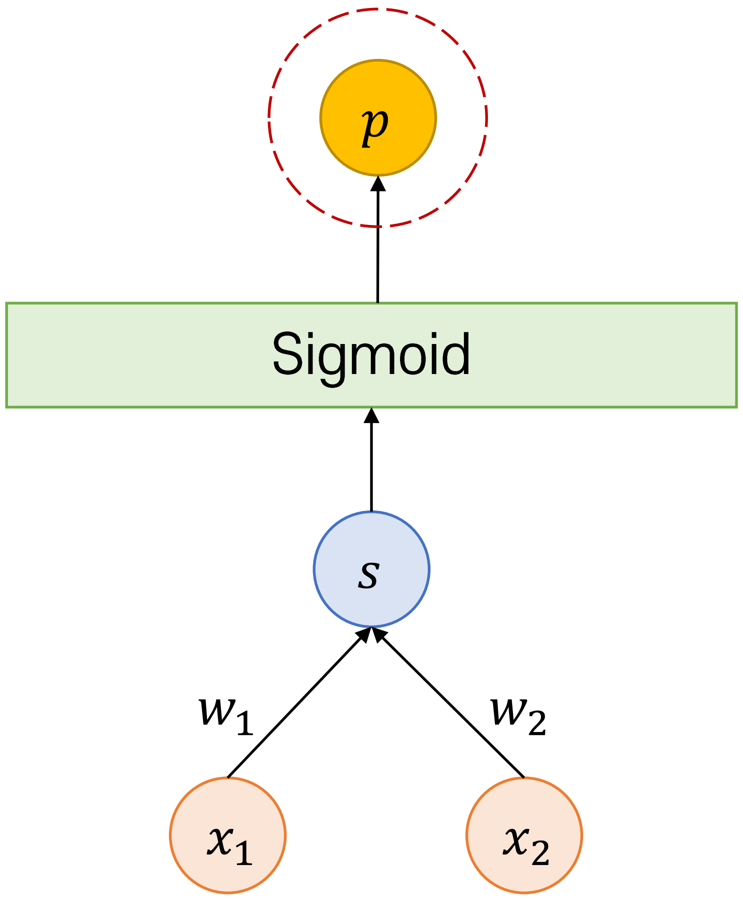
$$\frac{\partial \mathcal{L}}{\partial W} = \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial w_{11}} & \frac{\partial \mathcal{L}}{\partial w_{12}} \\ \frac{\partial \mathcal{L}}{\partial w_{21}} & \frac{\partial \mathcal{L}}{\partial w_{22}} \\ \frac{\partial \mathcal{L}}{\partial w_{31}} & \frac{\partial \mathcal{L}}{\partial w_{32}} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial s_1}\frac{\partial s_1}{\partial w_{11}} & \frac{\partial \mathcal{L}}{\partial s_1}\frac{\partial s_1}{\partial w_{12}} \\ \frac{\partial \mathcal{L}}{\partial s_2}\frac{\partial s_2}{\partial w_{21}} & \frac{\partial \mathcal{L}}{\partial s_2}\frac{\partial s_2}{\partial w_{22}} \\ \frac{\partial \mathcal{L}}{\partial s_3}\frac{\partial s_3}{\partial w_{31}} & \frac{\partial \mathcal{L}}{\partial s_3}\frac{\partial s_3}{\partial w_{32}} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial s_1} x_1 & \frac{\partial \mathcal{L}}{\partial s_1} x_2 \\ \frac{\partial \mathcal{L}}{\partial s_2} x_1 & \frac{\partial \mathcal{L}}{\partial s_2} x_2 \\ \frac{\partial \mathcal{L}}{\partial s_3} x_1 & \frac{\partial \mathcal{L}}{\partial s_3} x_2 \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial s_1} \\ \frac{\partial \mathcal{L}}{\partial s_2} \\ \frac{\partial \mathcal{L}}{\partial s_3} \end{bmatrix} \begin{bmatrix} x_1 & x_2 \end{bmatrix}$$

$$= \frac{\partial \mathcal{L}}{\partial \mathbf{s}} \mathbf{x}^\top$$

$$= (\mathbf{p} - \mathbf{y})\mathbf{x}^\top$$

Logistic Regression의 Backpropagation



BCE Loss:

$$\mathcal{L} = -y\log p - (1-y)\log(1-p)$$

정답 $y = 1$을 가정:

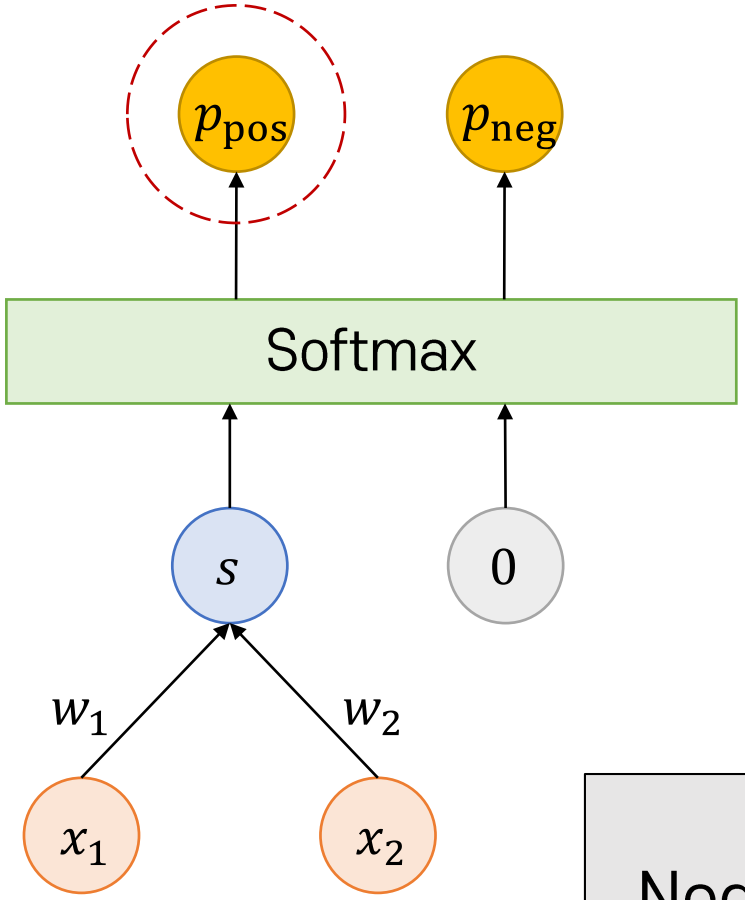
$$\mathcal{L} = -\log p = -\log\left(\frac{1}{1+e^{-s}}\right)$$

s의 Gradient:

$$\frac{\partial \mathcal{L}}{\partial s} = -\frac{\frac{\partial}{\partial s}\left(\frac{1}{1+e^{-s}}\right)}{\frac{1}{1+e^{-s}}} = -\frac{\frac{e^{-s}}{(1+e^{-s})^2}}{\frac{1}{1+e^{-s}}}$$

$$= -\frac{e^{-s}}{1+e^{-s}} = p - 1$$

Logistic Regression과 Softmax Classification



Softmax Loss:

$$\mathcal{L} = -\sum_{c \in \{\text{pos},\text{neg}\}} y_c \log p_c$$

정답 $\boldsymbol{y} = \begin{bmatrix}1\\0\end{bmatrix}$이라 가정:

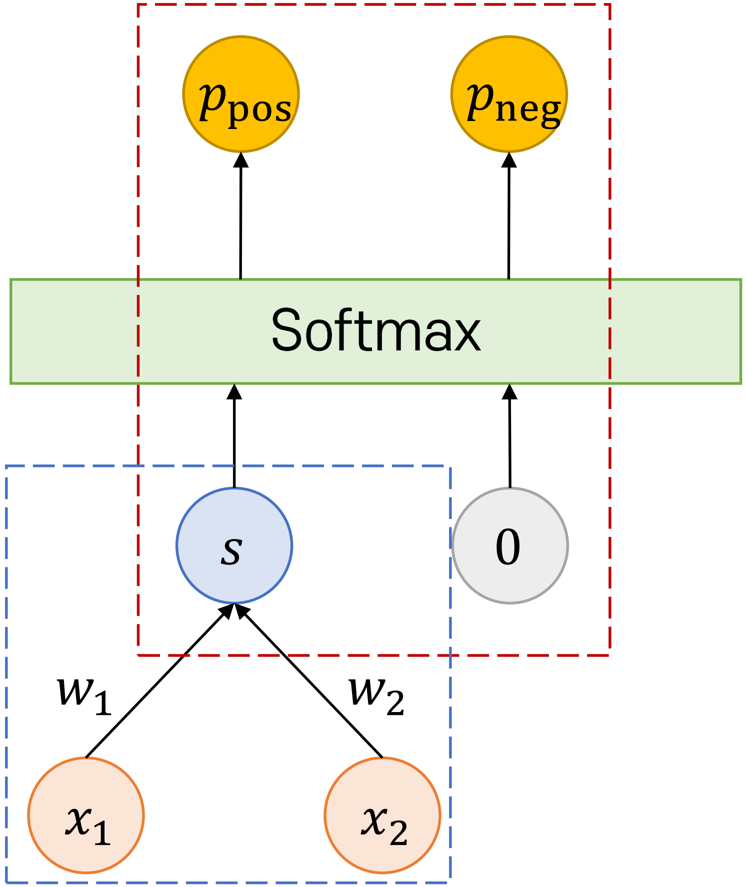
$$\mathcal{L} = -\log p_{\text{pos}}$$

$$= -\log\left(\frac{e^s}{e^s+e^0}\right) = -\log\left(\frac{1}{1+e^{-s}}\right)$$

→ BCE Loss

Logistic regression은
Negative class logit을 0으로 계산한
Softmax classification과 같음

Logistic Regression의 Backpropagation



$\mathbf{s} = \begin{bmatrix} s \\ 0 \end{bmatrix}$의 Gradient:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{s}} = \mathbf{p} - \mathbf{y} = \begin{bmatrix} p_{\text{pos}} - 1 \\ p_{\text{neg}} \end{bmatrix}$$

따라서, $\frac{\partial \mathcal{L}}{\partial s} = p_{\text{pos}} - 1$

w_1, w_2의 Gradient:

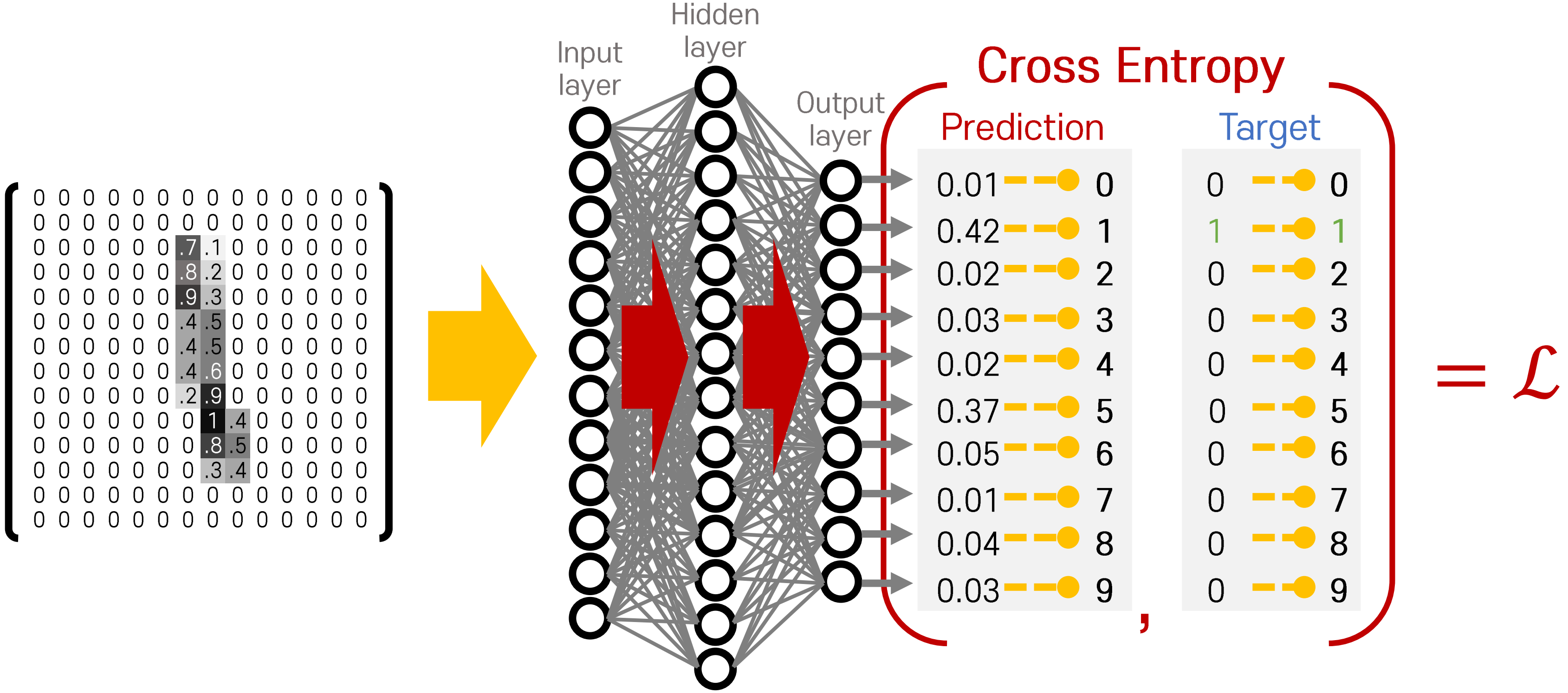
$$s = w_1 x_1 + w_2 x_2, \qquad \frac{\partial \mathcal{L}}{\partial s} = p_{\text{pos}} - 1$$

$$\frac{\partial \mathcal{L}}{\partial w_1} = \frac{\partial \mathcal{L}}{\partial s}\frac{\partial s}{\partial w_1} = (p_{\text{pos}} - 1)x_1$$

$$\frac{\partial \mathcal{L}}{\partial w_2} = \frac{\partial \mathcal{L}}{\partial s}\frac{\partial s}{\partial w_2} = (p_{\text{pos}} - 1)x_2$$

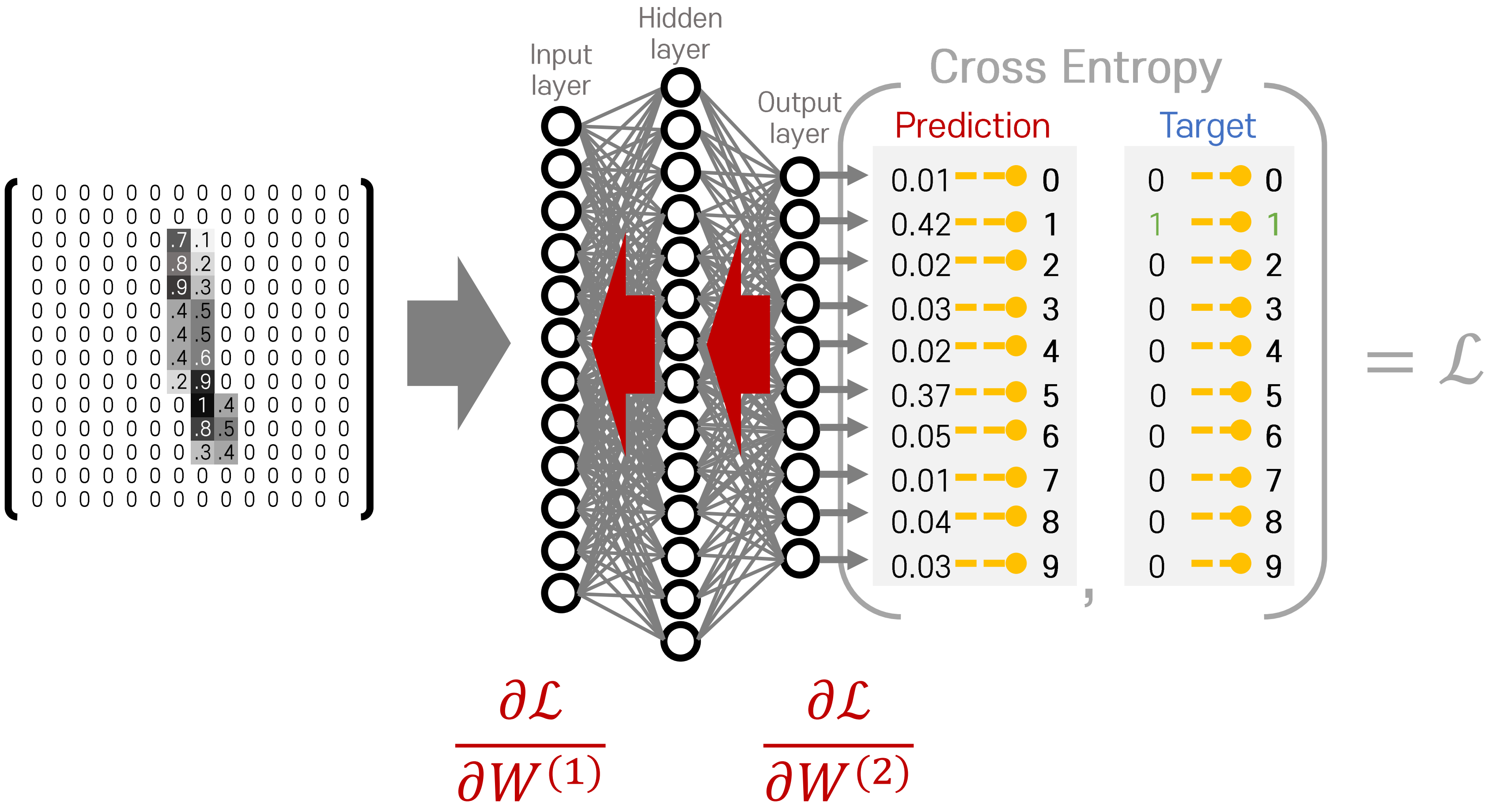
일반적인 Neural Network의 학습 과정

Forward Propagation를 통해서 주어진 입력에 대해 Loss 계산



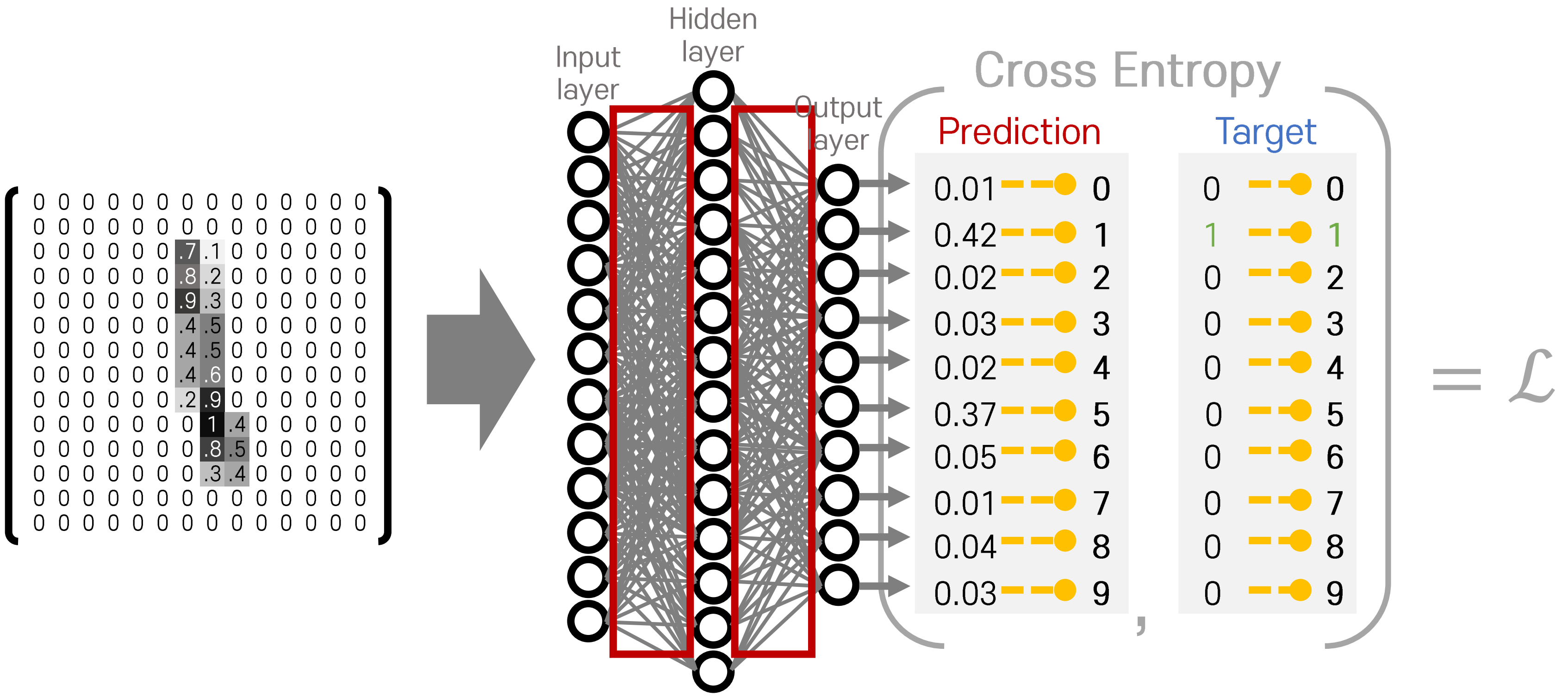
Neural Network 학습

Backpropagation를 통하여 각 Weight들에 대한 Gradient 계산



Neural Network 학습

Gradient Descent로 각 Weight를 Gradient를 사용해 갱신



$$W^{(1)} := W^{(1)} - \alpha \frac{\partial \mathcal{L}}{\partial W^{(1)}} \qquad W^{(2)} := W^{(2)} - \alpha \frac{\partial \mathcal{L}}{\partial W^{(2)}}$$

요약

- Softmax classifier에서의 backpropagation
- Logistic regression의 backpropagation은
  Softmax classifier backpropagation의 특수한 예시임을 확인
- Backpropagation을 통한 Weight gradient 계산 및 Neural Network 학습