

인공지능을 위한 수학

CHAP 7 - 이미지 인식



Contents

● 이미지 인식

● 딥러닝으로 손글씨 인식하기

● 데이터 세트 'MNIST'

● 신경망이란? (기초)

● 신경망이란? (심화)

● 심층 신경망이란?

● 순전파

● 손실함수

● 오차역전파법 사용하기

● 완성된 모델 평가하기



01

딥러닝으로 손글씨 인식하기



손 글씨 인식시키기

가장 기본적인 DNN(Deep Neural Network) 알고리즘을 이용

0에서 9까지의 숫자가 손 글씨로 쓰인 이미지를 보고, 해당 숫자가 무엇인지 맞추기

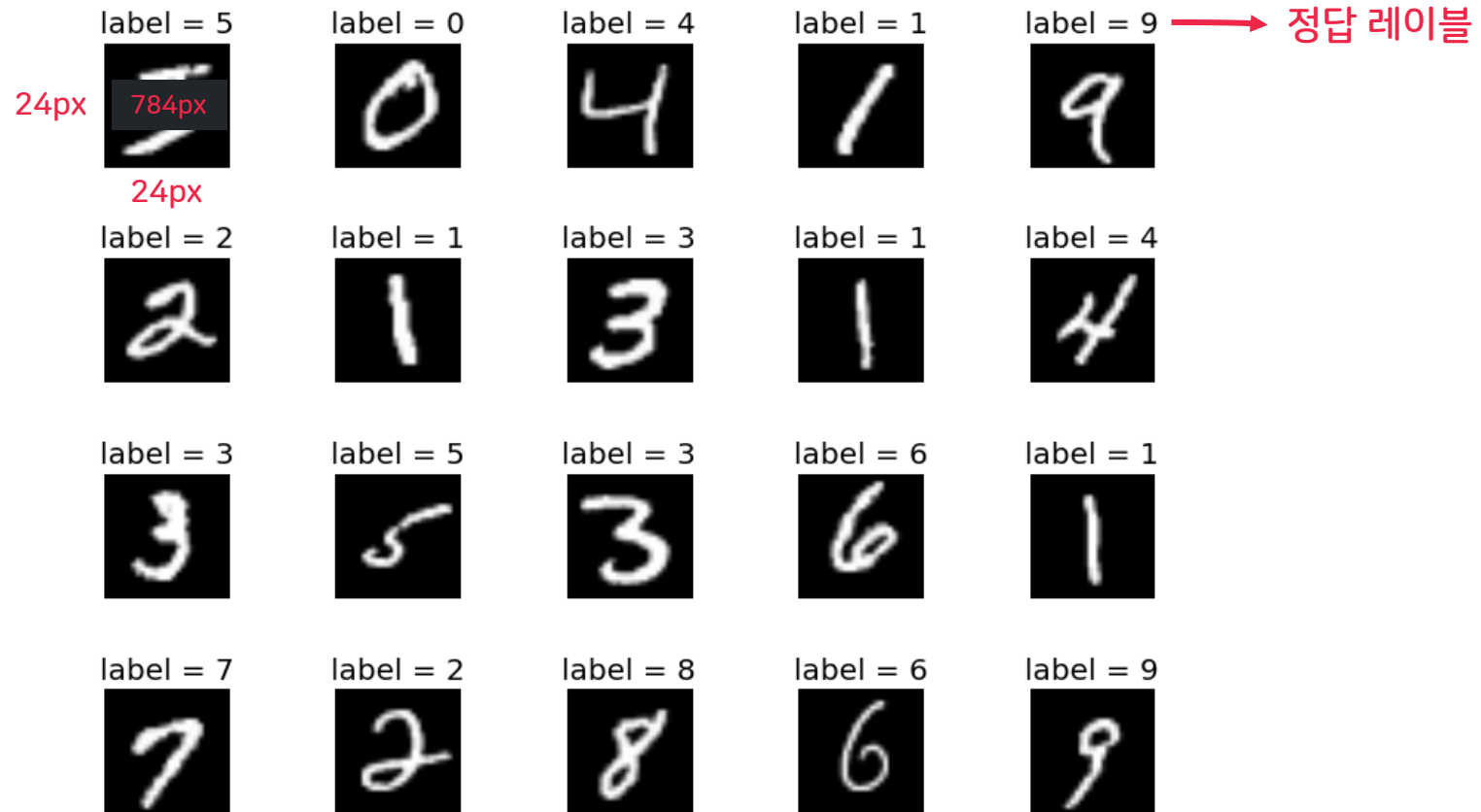


02

데이터 세트 'MNIST'



데이터 세트 'MNIST'



<https://medium.com/fenwicks/tutorial-1-mnist-the-hello-world-of-deep-learning-abd252c47709>



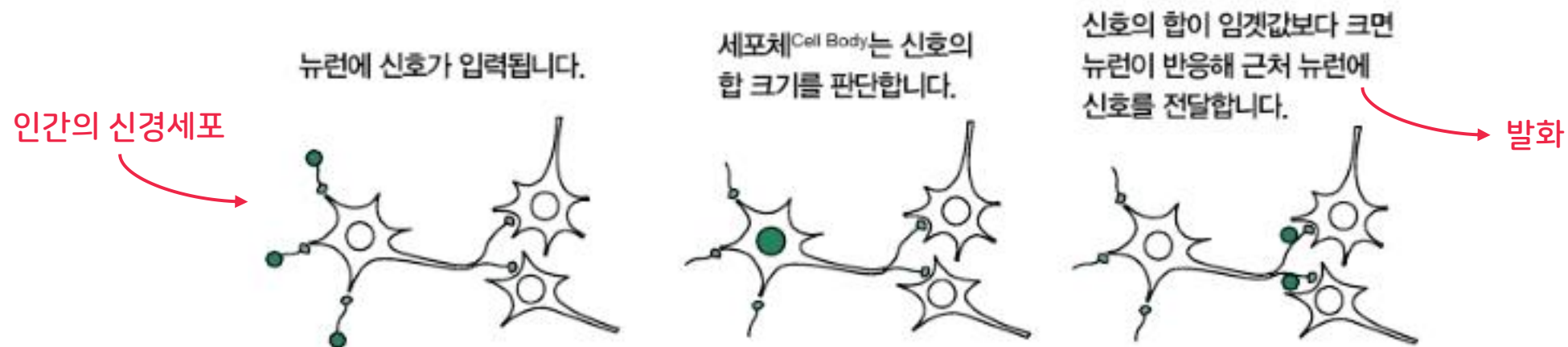
03

신경망이란? (기초)



신경망이란?

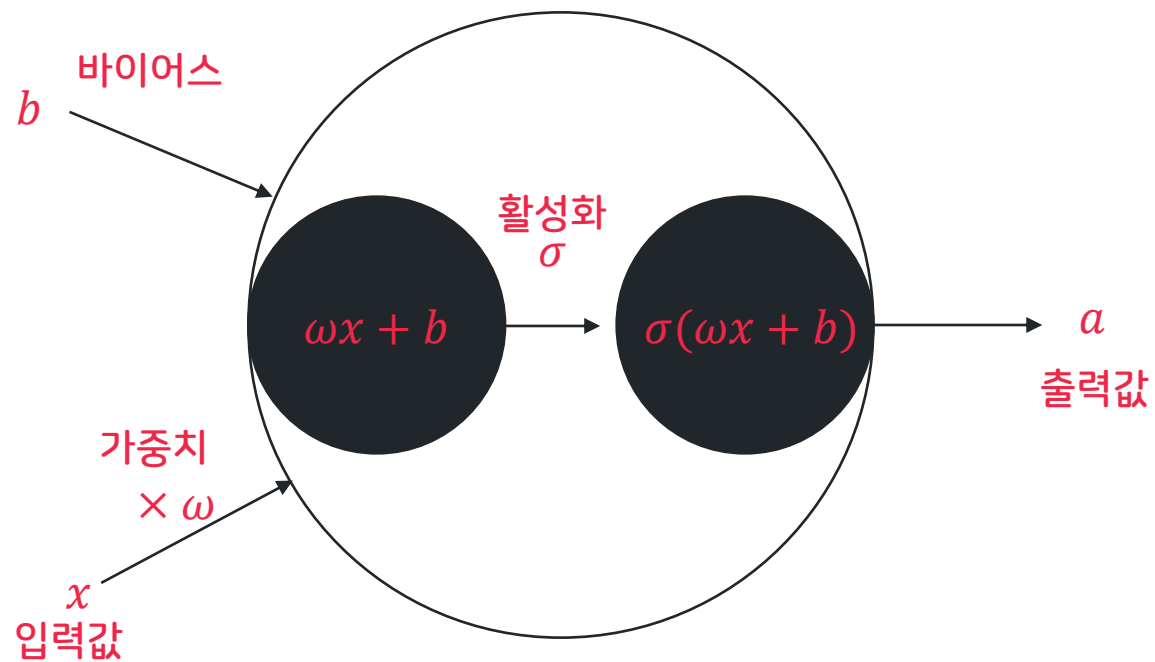
사람의 뇌에 있는 신경세포(뉴런)와 그런 세포들의 연결 관계(네트워크)를 흉내내서 만든 수학적 모델



<https://m.yes24.com/Goods/ReviewList/58137037?goodsSortNo=001&resourceKeyGb=01&goodsStateGb=02&goodsSetYn=N&GoodsGb=01>



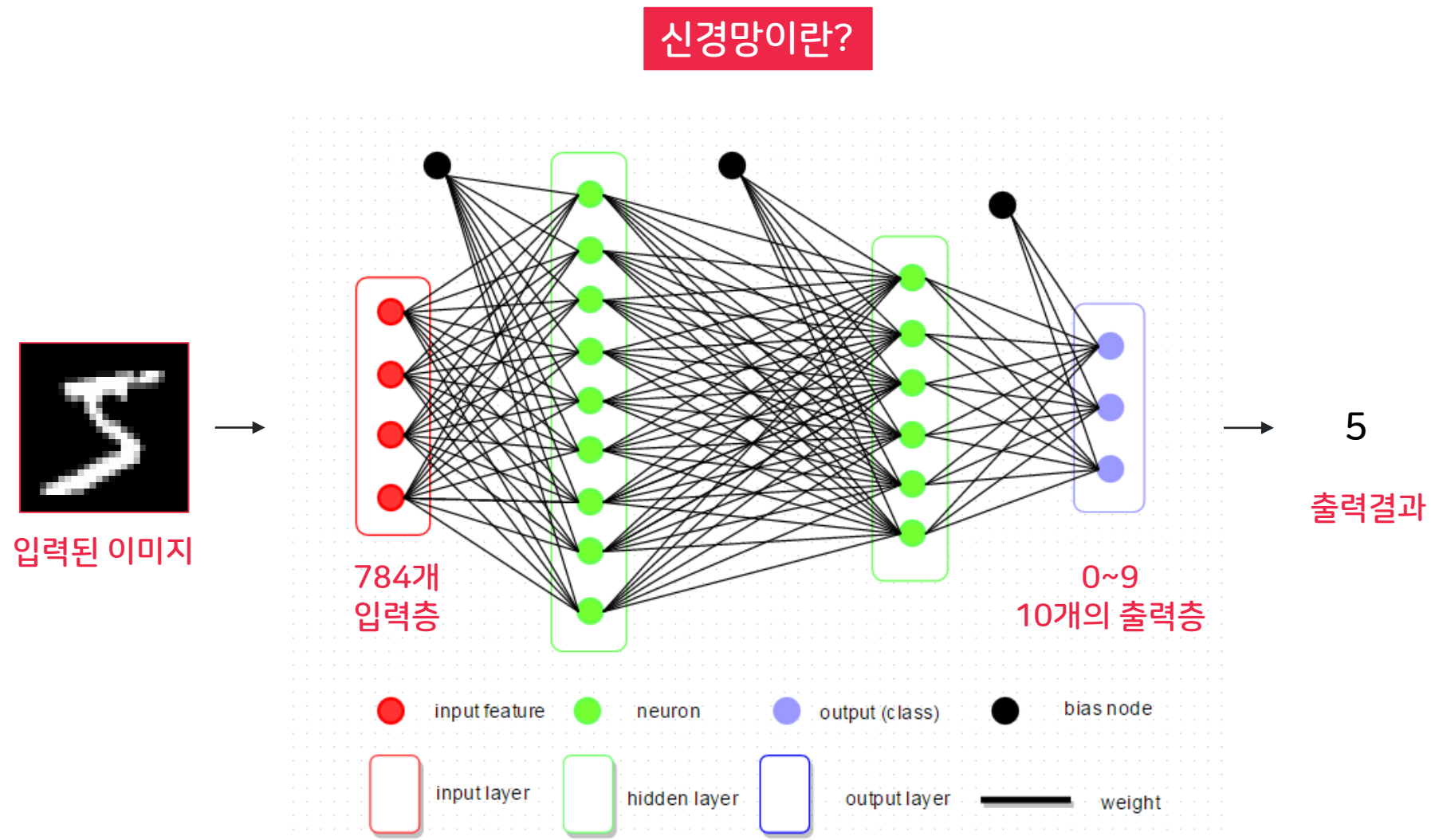
신경망이란?



다음 계층의
입력으로 전달



03. 신경망이란? (기초)



<https://cinema4dr12.tistory.com/1070>



04

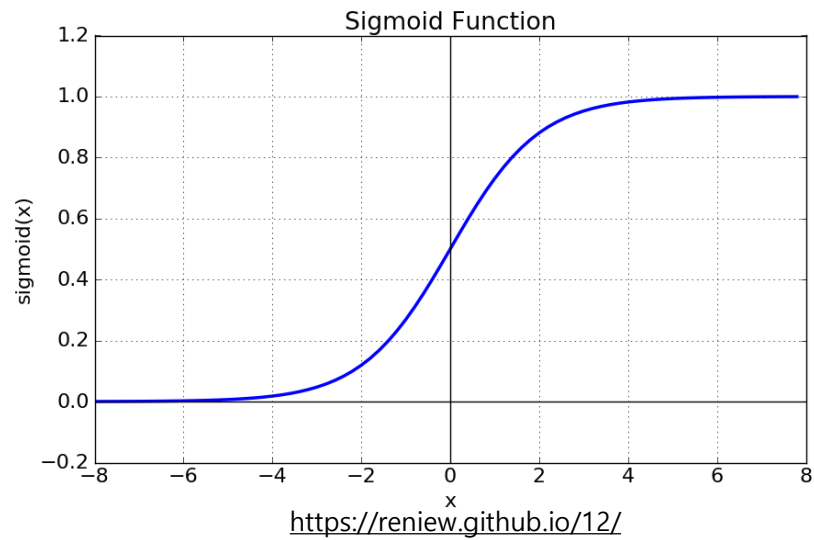
신경망이란? (심화)



04. 신경망이란? (심화)

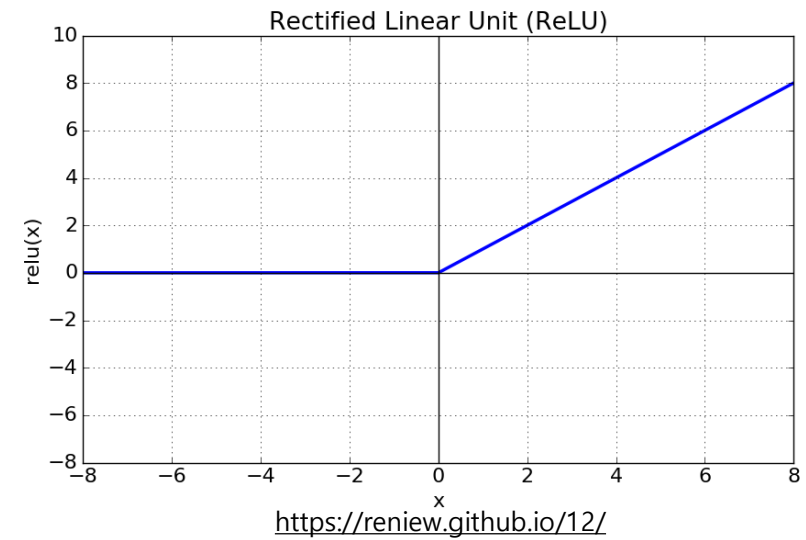
시그모이드 함수

$$\varsigma(x) = \frac{1}{1 + \exp(-x)}$$

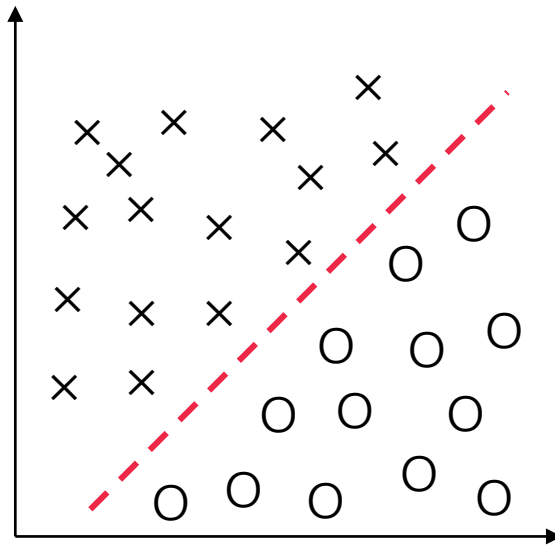


ReLU 함수

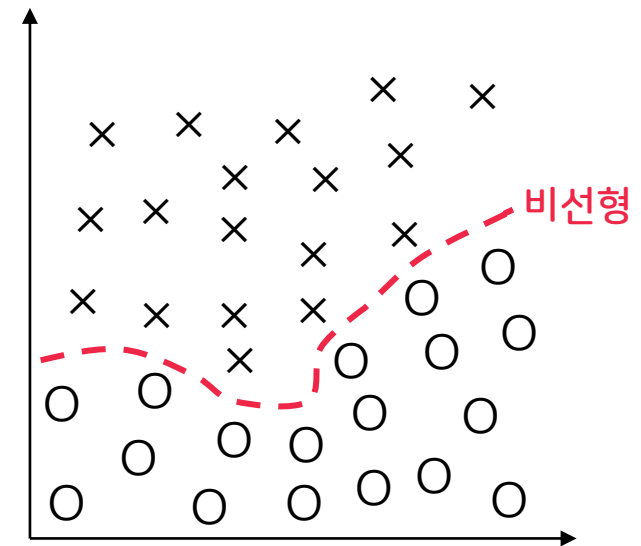
$$\varphi(x) = \max(0, x) = \begin{cases} 0 & (x \leq 0) \\ x & (x > 0) \end{cases}$$



시그모이드 함수, ReLU함수 - 비선형 함수



선형 분리가 가능한 경우



선형 분리가 불가능한 경우

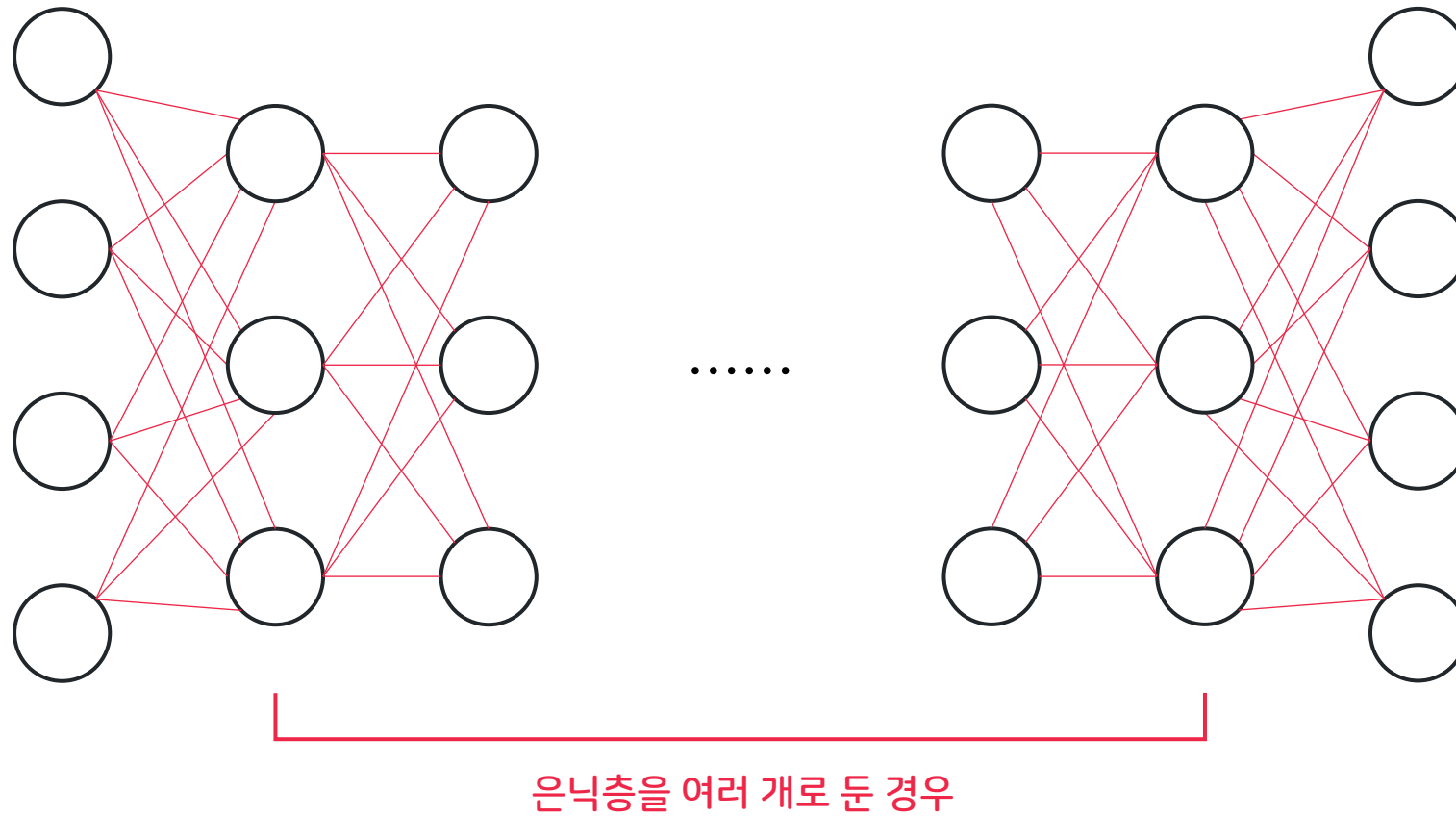


05

심층 신경망이란?



심층 신경망 (Deep Neural Network)



06

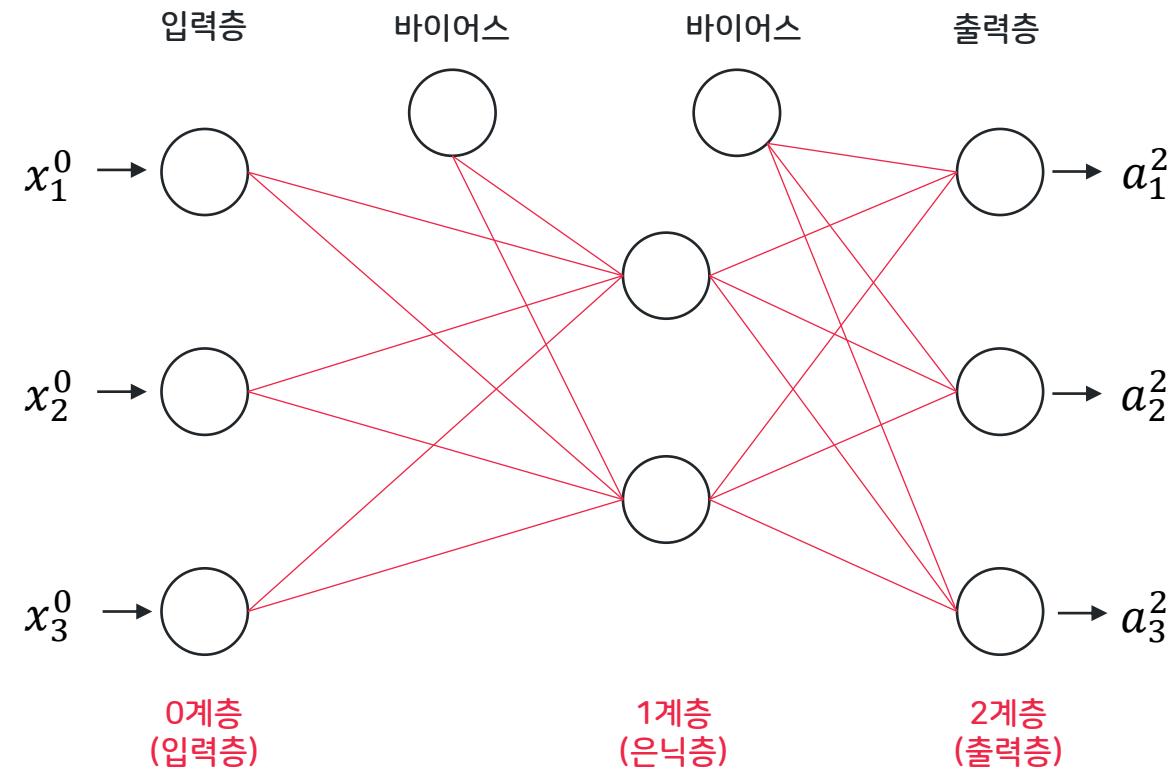
순전파



순전파

입력층에서 출력층까지 정보가 순서대로 전달

Q. 입력층의 노드가 3개, 은닉층의 노드가 2개, 출력층의 노드가 3개인 신경망을 수식으로 표현 하시오.



문자를 표기하는 관례

x_{β}^a
계층 번호
노드 번호
 입력값 x 의 표기법

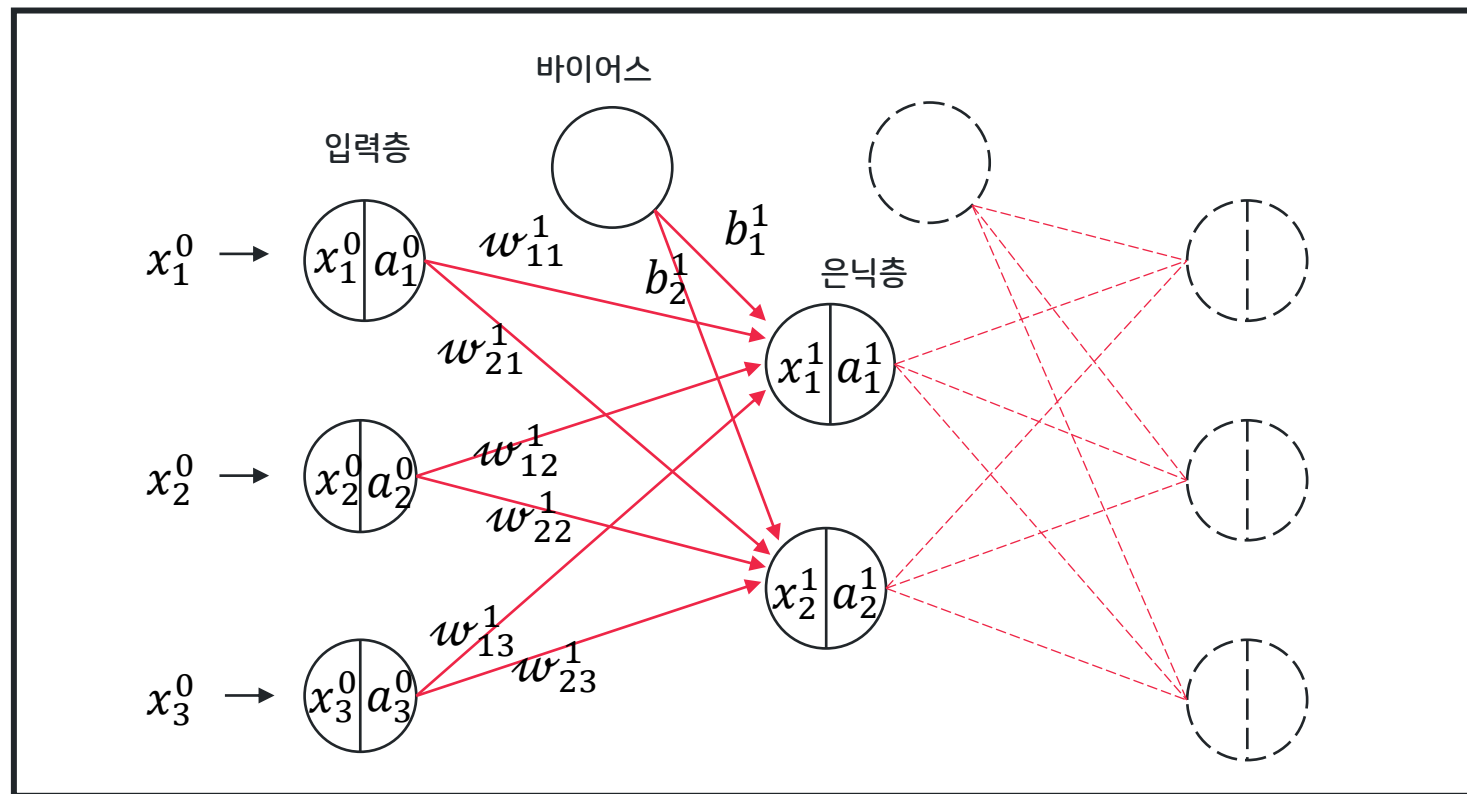
a_{β}
계층 번호
노드 번호
 출력값 a 의 표기법

$w_{\beta r}^a$
다음 계층 번호
이전 계층의 노드 번호
다음 계층의 노드 번호
 가중치 w 의 표기법

b_{β}^a
다음 계층 번호
다음 계층의 노드 번호
 바이어스 b 의 표기법

σ_a
계층 번호
 활성화 함수 σ 의 표기법





[신경망의 입력층에서 은닉층까지의 정보 흐름]

최초의 입력층 : 별다른 처리가 없음

① $x_1^0 = a_1^0, x_2^0 = a_2^0$

은닉층의 입력값

$$x_1^1 = w_{11}^1 a_1^0 + w_{12}^1 a_2^0 + w_{13}^1 a_3^0 + b_1^1$$

$$x_2^1 = w_{21}^1 a_1^0 + w_{22}^1 a_2^0 + w_{23}^1 a_3^0 + b_2^1$$

↓ 행렬의 간소화

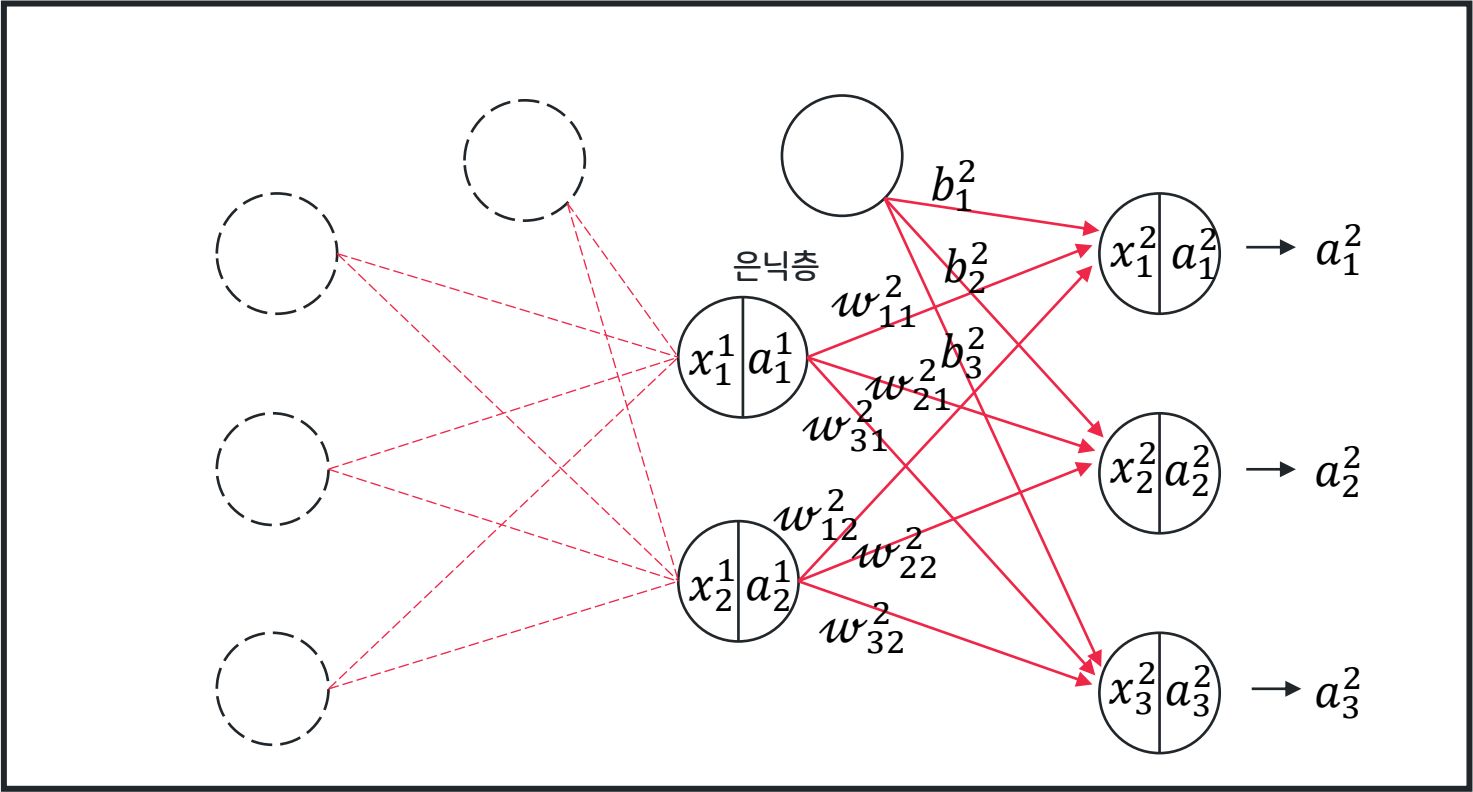
② $x^1 = W^1 a^0 + b^1$

은닉층의 출력값(시그모이드 함수 사용)

③ $a_1^1 = \sigma_1(x_1^1)$

$a_2^1 = \sigma_1(x_2^1)$





[신경망의 은닉층에서 출력층까지의 정보 흐름]

행렬로 표현

$$x^2 = \begin{pmatrix} x_1^2 \\ x_2^2 \\ x_3^2 \end{pmatrix}, W^2 = \begin{pmatrix} w_{11}^2 & w_{12}^2 \\ w_{21}^2 & w_{22}^2 \\ w_{31}^2 & w_{32}^2 \end{pmatrix},$$

$$a^1 = \begin{pmatrix} a_1^1 \\ a_2^1 \end{pmatrix}, b^2 = \begin{pmatrix} b_1^2 \\ b_2^2 \\ b_3^2 \end{pmatrix}$$

↓ 행렬의 간소화

$$x^2 = W^2 a^1 + b^2$$

최종 출력층 softmax함수 사용

$$\boxed{a_1^2} = \sigma_2(x_1^2), \quad \boxed{a_2^2} = \sigma_2(x_2^2), \quad \boxed{a_3^2} = \sigma_2(x_3^2)$$

가장 큰 값이 나오는 카테고리가 신경망의 판별결과



07

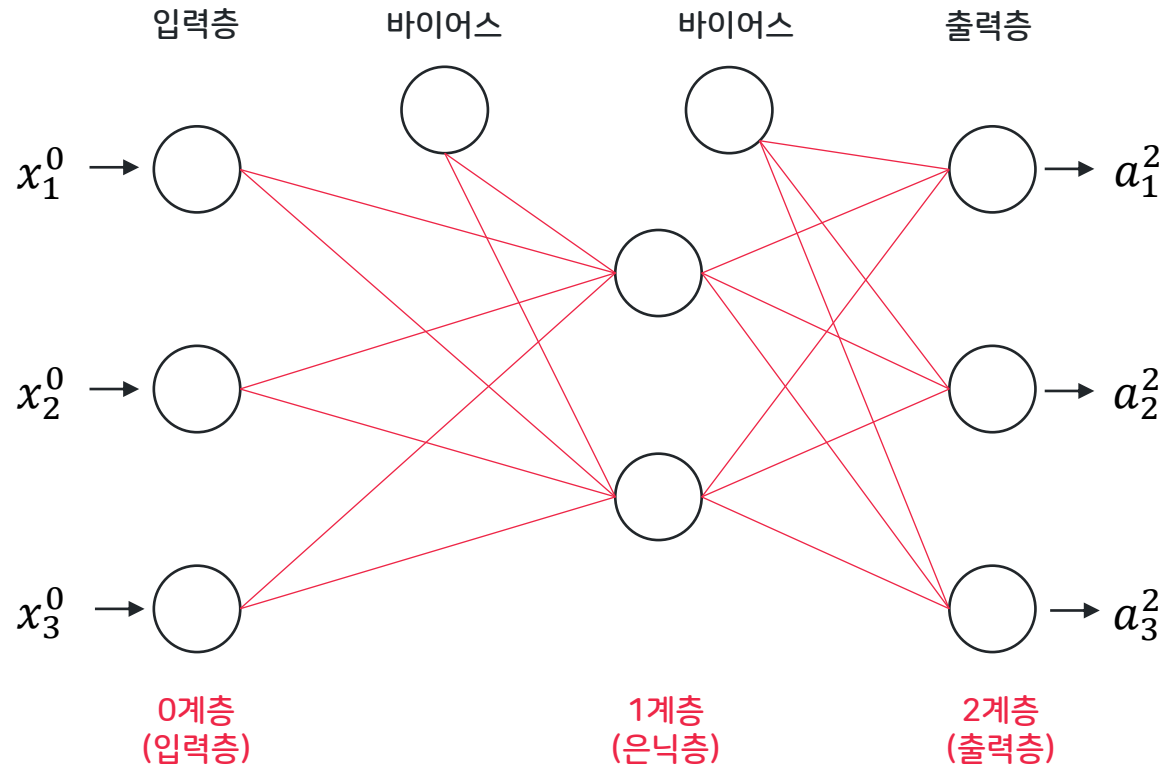
손실 함수



손실함수 - 평균제곱오차(MSE:Mean Squared Error)

신경망이 출력한 값과 실제 값과의 오차에 대한 함수

Q. 7.6절의 예제에서 다루었던 3계층 신경망에서 2계층의 출력 y^2 가 $y^2 = (0.1w, 0.5w, 1 - 0.6w)$ 이고
정답 t 는 $t = (0, 1, 0)$ 이라고 할 때, 평균 제곱오차 E 를 구하고 이 값을 최소로 만드는 w 를 구하시오.



$$E = \frac{1}{2} \|t - y\|^2$$

t : 정답 레이블, y : 신경망의 출력

평균제곱오차 E 구하기



Q. 7.6절의 예제에서 다루었던 3계층 신경망에서 2계층의 출력 y^2 가 $y^2 = (0.1w, 0.5w, 1 - 0.6w)$ 이고 정답 t 는 $t = (0, 1, 0)$ 이라고 할 때, 평균 제곱오차 E 를 구하고 이 값을 최소로 만드는 w 를 구하시오.

$$E = \frac{1}{2} \|t - y\|^2$$

↓ 수식 사용

$$E = \frac{1}{2} \{ \overset{t}{(0 - 0.1w)}^2 + \overset{t}{(1 - 0.5w)}^2 + \overset{t}{(0 - (1 - 0.6w))}^2 \}$$

$$E = 0.31w^2 - 1.1w + 1$$

↓ w 를 미분했을 때 값이 0이 되도록

$$\frac{dE}{dw} = 0.62w - 1.1 = 0$$











↓

$$w \doteq 1.774$$



07. 손실 함수

softmax

	→	0.01	0	—	0
	→	0.02	0	—	1
	→	0.05	0	—	2
	→	0.02	0	—	3
	→	0.67	1	—	4
	→	0.13	0	—	5
	→	0.05	0	—	6
	→	0.01	0	—	7
	→	0.01	0	—	8
	→	0.03	0	—	9

$$\begin{aligned} E &= \frac{1}{2} \{ (0 - 0.01)^2 + (0 - 0.02)^2 + (0 - 0.05)^2 + (0 - 0.02)^2 + (1 - 0.67)^2 \\ &\quad + (0 - 0.13)^2 + (0 - 0.05)^2 + (0 - 0.01)^2 + (0 - 0.01)^2 + (0 - 0.03)^2 \} \\ &= 0.0664 \end{aligned}$$

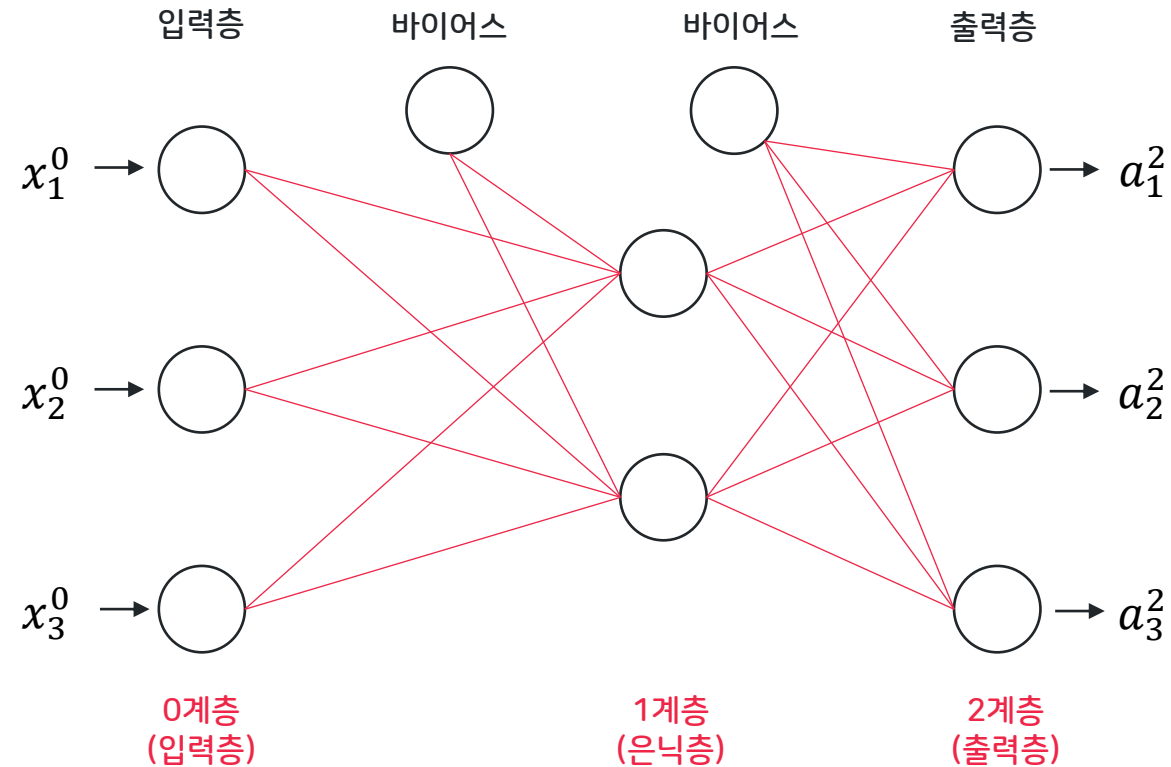


08

경사하강법 사용하기



- Q. 입력층에 노드 3개, 은닉층에 노드 2개, 출력층에 노드 3개가 있는 신경망의 관계를 수식으로 표현 하시오.
이 때, 은닉층의 활성화 함수는 시그모이드 함수를, 출력층의 활성화 함수는 softmax함수를 사용 하시오.



평균제곱오차 사용

$$E = \frac{1}{2} \|a^2 - y\|^2$$

$$E = \frac{1}{2} \|\sigma_2(a^1 W^2 + b^2) - y\|^2$$

$$E = \frac{1}{2} \|\sigma_2(\sigma_1(a^0 W^1 + b^1) W^2 + b^2) - y\|^2$$

$$E = \frac{1}{2} \|\sigma_2(\sigma_1(x^0 W^1 + b^1) W^2 + b^2) - y\|^2$$



$$W^1 = \begin{pmatrix} w_{11}^1 & w_{12}^1 & w_{13}^1 \\ w_{21}^1 & w_{21}^1 & w_{21}^1 \end{pmatrix} \quad W^2 = \begin{pmatrix} w_{11}^2 & w_{12}^2 \\ w_{21}^2 & w_{12}^2 \\ w_{31}^2 & w_{32}^2 \end{pmatrix}, \quad b^1 = \begin{pmatrix} b_1^1 \\ b_2^1 \end{pmatrix}, \quad b^2 = \begin{pmatrix} b_1^2 \\ b_2^2 \\ b_3^2 \end{pmatrix}$$

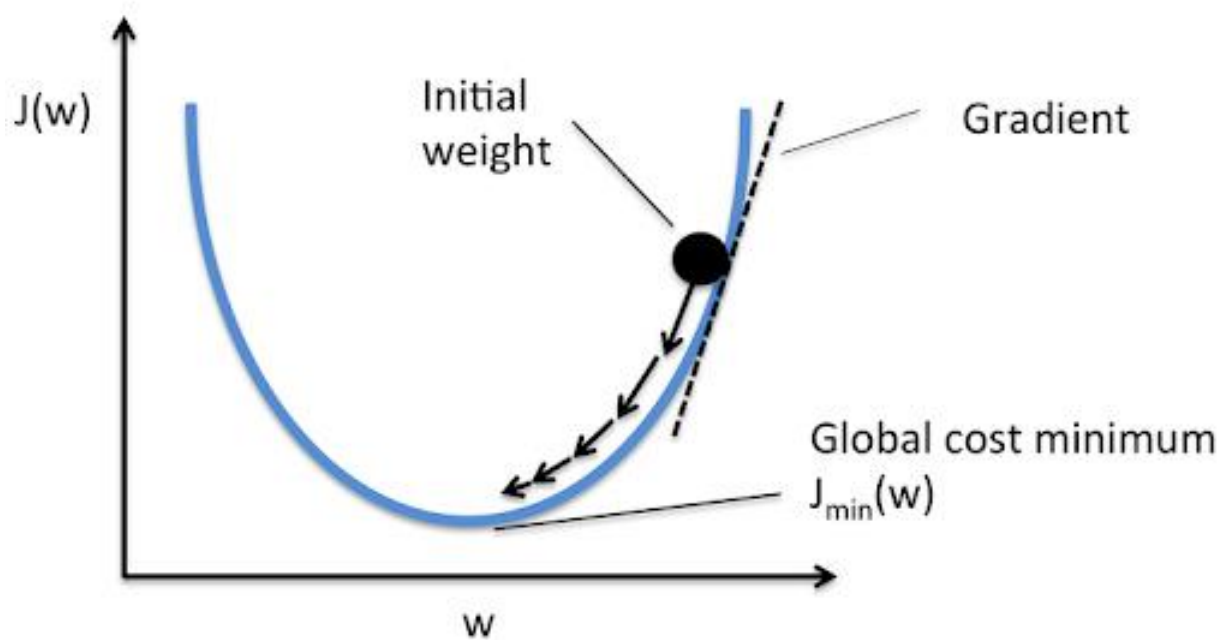


간단한 예에서조차 17개의 변수가 나옴



경사하강법

함수의 그래프를 따라 움직이면서 기울기를 조사하고, 이 때 구한 기울기의 값이 작아지는 방향으로 조금씩 내려가는 방법



<http://blog.naver.com/PostView.nhn?blogId=qbxlvnf11&logNo=221386939202&parentCategoryNo=&categoryNo=52&viewDate=&isShowPopularPosts=false&from=postView>

$$\Delta f(x) = \frac{df(x)}{dx} \Delta x$$



08. 경사하강법 사용하기

$$\textcircled{1} \quad \frac{df(x)}{dx} = \lim_{\Delta x \rightarrow 0} \frac{\Delta f(x)}{\Delta x} = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

↓ $h = \Delta x$ 라고 가정

$$\textcircled{2} \quad \frac{df(x)}{dx} = \lim_{\Delta x \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

↓ Δx 를 최대한 0에 가깝게

$$\textcircled{3} \quad \frac{df(x)}{dx} \approx \frac{f(x+h) - f(x)}{h}$$

↓ Δx 를 양변에 곱하기

$$\textcircled{4} \quad \frac{df(x)}{dx} \Delta x \approx f(x + \Delta x) - f(x)$$

↓ 함수 $f(x)$ 에 대해 x 에 대해 Δx 변화를 주었을 때,
 $f(x)$ 의 변화량을 $\Delta f(x)$ 라고 한다면

$$\textcircled{5} \quad \Delta f(x) = f(x + \Delta x) - f(x)$$

↓ ⑤번식에 ④번식 대입

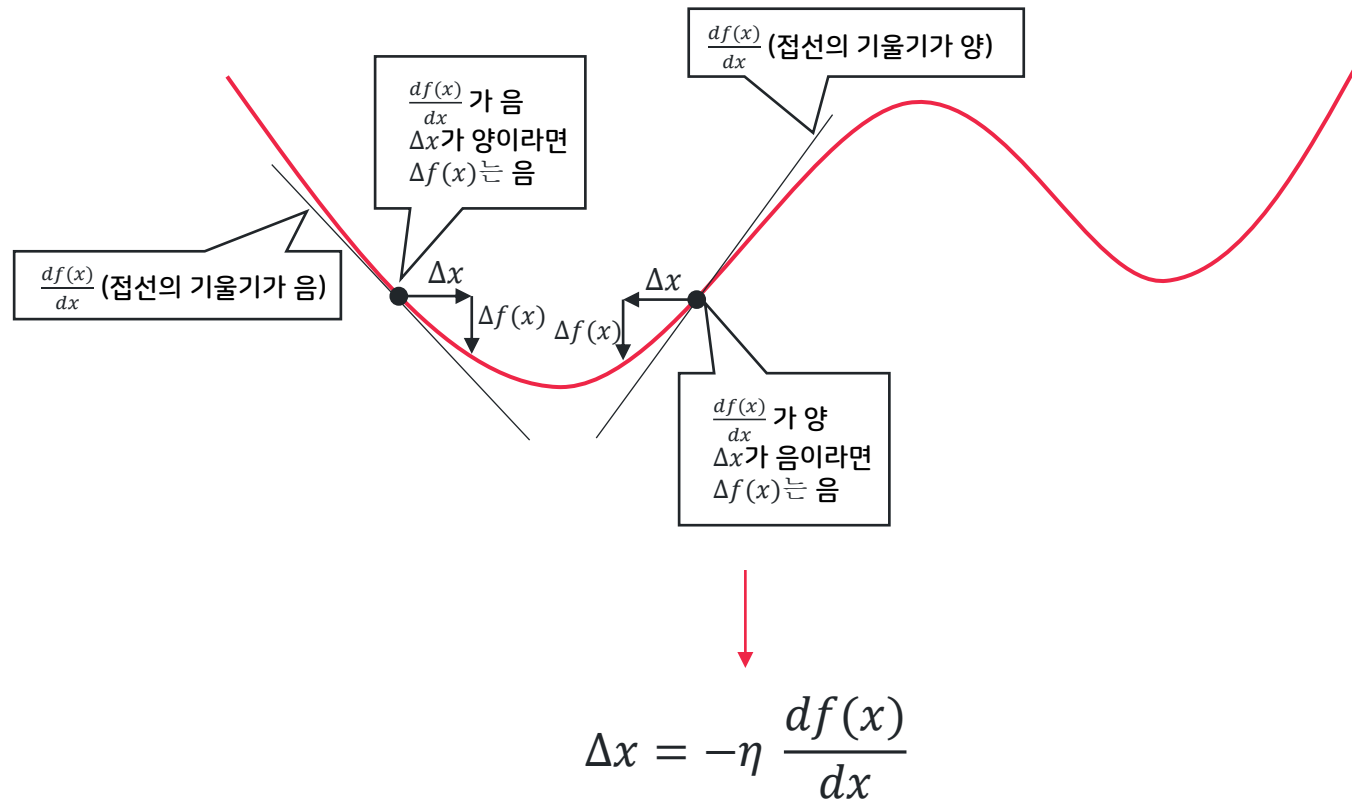
$$\Delta f(x) \approx \frac{df(x)}{dx} \Delta x$$



08. 경사하강법 사용하기

$$\Delta f(x) \approx \frac{df(x)}{dx} \Delta x$$

음이 되는 방향으로 조금씩 이동하면서 최소값 찾기



08. 경사하강법 사용하기

$$\textcircled{1} \quad \Delta x = -\eta \frac{df(x)}{dx}$$

학습률

(공이 어느 정도의 폭으로 움직이는가)

$$\textcircled{2} \quad \Delta x = x_{new} - x_{old}$$



$$\textcircled{3} \quad x_{new} = x_{old} - \frac{df(x)}{dx}$$



다변수 함수로 확장

$$E = -\eta \left(\frac{\partial E}{\partial w_{11}^1}, \frac{\partial E}{\partial w_{21}^1}, \frac{\partial E}{\partial w_{31}^1} \right)$$

손실함수 E 의 기울기



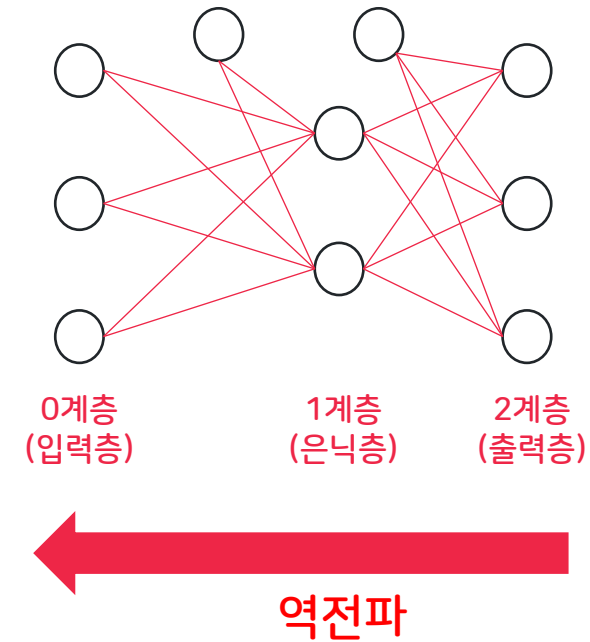
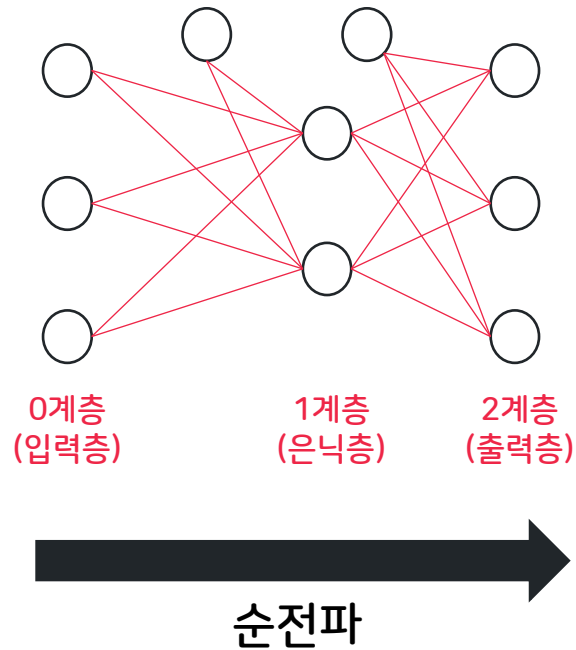
09

오차역전파법 사용하기



오차역전파법(Backpropagation)

출력값의 오차를 기반으로 출력층에서 입력층 방향으로 가중치와 바이어스를 거꾸로 갱신해 나가는 방법



09. 오차역전파법 사용하기

$$E = -\eta \left(\frac{\partial E}{\partial w_{11}^1}, \frac{\partial E}{\partial w_{21}^1}, \frac{\partial E}{\partial w_{31}^1} \right) \xrightarrow{\text{일반화}} \frac{\partial E}{\partial w_{kj}^l}$$

미분의 연쇄법칙 적용

$$\textcircled{1} \quad \frac{\partial E}{\partial w_{kj}^l} = \frac{\partial E}{\partial x_k^l} \frac{\partial x_k^l}{\partial w_{kj}^l}$$

풀어 쓰기

$$\textcircled{2} \quad x_k^l = w_{k1}^l a_1^{l-1} + w_{k2}^l a_2^{l-1} + \dots + w_{kj}^l a_j^{l-1} + b_k^l$$

x_{kj}^l 로 미분

$$\textcircled{3} \quad \frac{\partial x_k^l}{\partial w_{kj}^l} = a_j^{l-1}$$

①에 ③ 대입

$$\textcircled{4} \quad \frac{\partial E}{\partial w_{kj}^l} = \frac{\partial E}{\partial x_k^l} a_j^{l-1}$$

$$\textcircled{5} \quad \frac{\partial E}{\partial w_{kj}^l} = \delta_k^l a_j^{l-1}$$

δ_k^l 로 변환



Case 1 : 마지막 계층일 때 δ_k^L 구하기

$$\frac{\partial E}{\partial w_{kj}^L} = \delta_k^L a_j^{l-1}$$

연쇄법칙

$$\textcircled{1} \quad \delta_k^L = \frac{\partial E}{\partial x_k^L} = \frac{\partial E}{\partial a_k^L} \frac{\partial a_k^L}{\partial x_k^L}$$

$$\textcircled{2} \quad \frac{\partial a_k^L}{\partial x_k^L} = \frac{\partial \zeta(x_k^L)}{\partial x_k^L} = \zeta'(x_k^L)$$

$$\textcircled{3} \quad \frac{\partial E}{\partial a_k^L} = \frac{\partial \frac{1}{2}(a_k^L - y_k)^2}{\partial a_k^L} = (a_k^L - y_k)$$

$$\textcircled{4} \quad \delta_k^L = \frac{\partial E}{\partial x_k^L} = \frac{\partial E}{\partial a_k^L} \frac{\partial a_k^L}{\partial x_k^L} = \frac{\partial \frac{1}{2}(a_k^L - y_k)^2}{\partial a_k^L} \zeta'(x_k^L) = (a_k^L - y_k) \zeta'(x_k^L)$$

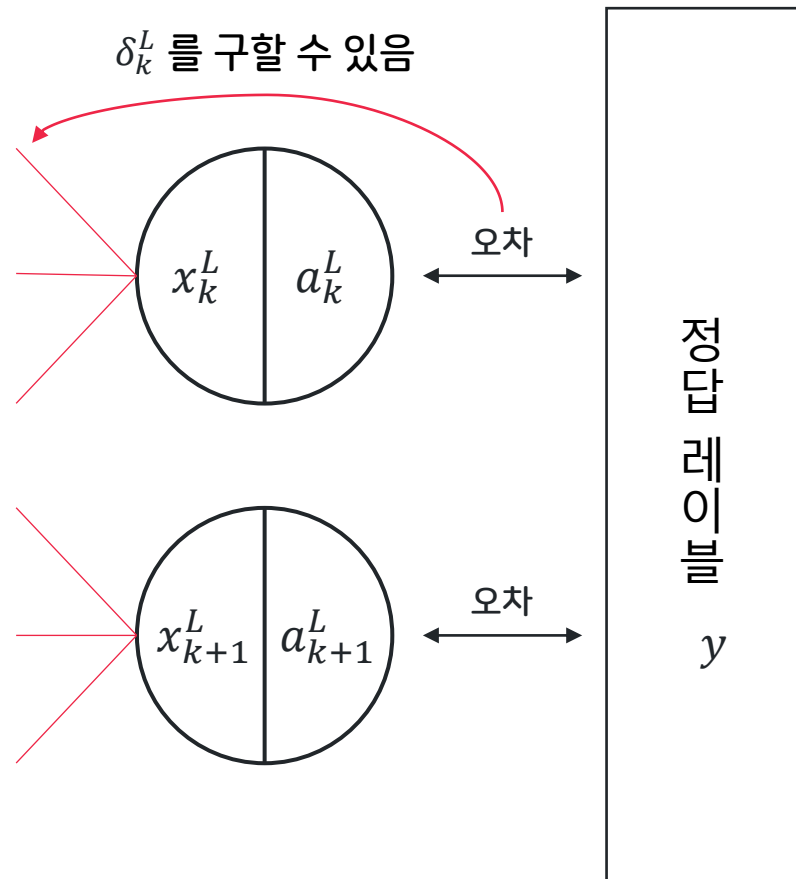
마지막 계층일 때의 입력을
활성화함수에 대입한 후 미분

마지막 계층일 때 출력

마지막 계층일 때의
정답 레이블



Case 1 : 마지막 계층일 때 δ_k^L 구하기



Case 2 : 마지막 계층이 아닐 때 δ_k^l 구하기

합성 함수의 미분공식 사용

$$\textcircled{1} \delta_1^2 = \frac{\partial E}{\partial x_1^3} \frac{\partial x_1^3}{\partial a_1^2} \frac{\partial a_1^2}{\partial x_1^2} + \frac{\partial E}{\partial x_2^3} \frac{\partial x_2^3}{\partial a_1^2} \frac{\partial a_1^2}{\partial x_1^2} + \frac{\partial E}{\partial x_3^3} \frac{\partial x_3^3}{\partial a_1^2} \frac{\partial a_1^2}{\partial x_1^2}$$

각 항의 첫번째 부분(초록색) 표현

$$\frac{\partial E}{\partial x_1^3} = \delta_1^3, \frac{\partial E}{\partial x_2^3} = \delta_2^3, \frac{\partial E}{\partial x_3^3} = \delta_3^3$$

각 항의 두번째 부분(파란색) 표현

$$\frac{\partial x_1^3}{\partial a_1^2} = w_{11}^3, \frac{\partial x_2^3}{\partial a_1^2} = w_{21}^3, \frac{\partial x_3^3}{\partial a_1^2} = w_{31}^3$$

각 항의 세번째 부분(노란색) 표현

$$\frac{\partial a_1^2}{\partial x_1^2} = \frac{\partial \zeta(x_1^2)}{\partial x_1^2} = \zeta'(x_1^2)$$

결국 δ_1^2 는 다음과 같이 표현 가능

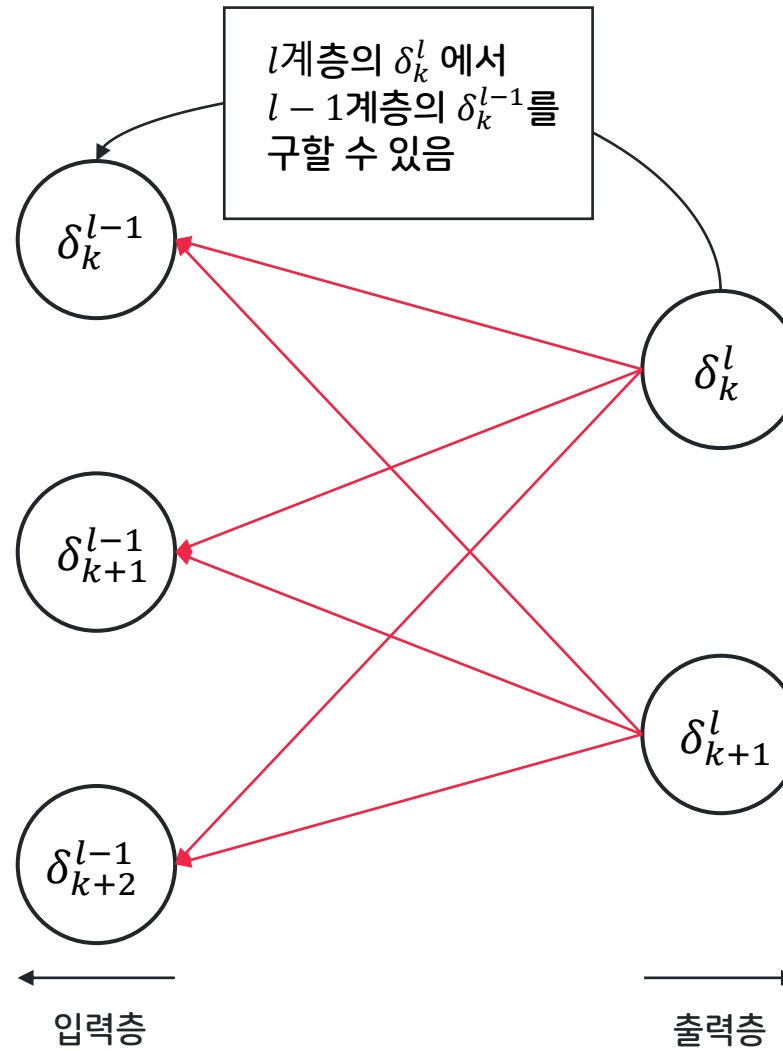
$$\begin{aligned} \textcircled{2} \delta_1^2 &= \delta_1^3 w_{11}^3 \zeta'(x_1^2) + \delta_2^3 w_{21}^3 \zeta'(x_1^2) + \delta_3^3 w_{31}^3 \zeta'(x_1^2) \\ &= (\delta_1^3 w_{11}^3 + \delta_2^3 w_{21}^3 + \delta_3^3 w_{31}^3) \zeta'(x_1^2) \end{aligned}$$

일반화(l 계층, $l+1$ 계층에도 적용 가능)

$$\begin{aligned} \textcircled{3} \delta_k^l &= (\delta_1^{l+k} w_{1k}^{l+k} + \delta_2^{l+1} w_{2k}^{l+1} + \dots + \delta_m^{l+1} w_{mk}^{l+1}) \zeta'(x_k^l) \\ \delta_k^l &= \sum_{i=1}^m (\delta_i^{l+1} w_{ik}^{l+1}) \zeta'(x_k^l) \end{aligned}$$



Case 2 : 마지막 계층일 때 δ_k^l 구하기



Case 1 & 2

$$\delta_k^l \begin{cases} (a_k^L - y_k) \varsigma'(x_k^L) \longrightarrow (l \text{이 마지막 계층일 때}) \\ \sum_{i=1}^m (\delta_i^{l+1} w_{ik}^{l+1}) \varsigma'(x_k^l) \longrightarrow (l \text{이 마지막 계층이 아닐 때}) \end{cases}$$

바이어스 구하기

$l - 1$ 계층의 출력이 1일 때, $a_j^{l-1} = 1$ 이 됨

대입

$$\textcircled{1} \quad \frac{\partial E}{\partial w_{kj}^l} = \delta_k^l a_j^{l-1}$$

$$\textcircled{2} \quad \frac{\partial E}{\partial b_k^l} = \delta_k^l$$



오차역전파법 총정리

$$\frac{\partial E}{\partial w_{kj}^l} = \delta_k^l a_j^{l-1}$$

$$\frac{\partial E}{\partial b_k^l} = \delta_k^l$$

$$\delta_k^l \begin{cases} (a_k^l - y_k) \zeta'(x_k^l) \longrightarrow & (l \text{이 마지막 계층일 때}) \\ \sum_{i=1}^m (\delta_i^{l+1} w_{ik}^{l+1}) \zeta'(x_k^l) \longrightarrow & (l \text{이 마지막 계층이 아닐 때}) \end{cases}$$

E : 손실 함수

w_{kj}^l : l 계층 k 번째 노드의 $l-1$ 계층 j 번째 노드로부터의 가중치

δ_k^l : l 계층 k 번째 노드의 오차

a_j^{l-1} : $l-1$ 계층 j 번째 노드의 출력

b_k^l : l 계층 k 번째 노드의 바이어스

y_k : k 번째 노드의 정답 레이블

$\zeta(x_k^l)$: 활성화 함수 (표준 시그모이드 함수)

m : $l+1$ 계층의 노드 개수



① 손실함수를 구한 후, 그 값을 최소화 하기 위한 w 와 b 를 구한다

$$E = \frac{1}{2} \|t - y\|^2, y = Wx + b$$

t : 정답 레이블

y : 신경망의 출력

W : 가중치

x : 출력

b : 바이어스

문제점

최소화 하고 싶은 변수 w 와 b 의 개수가 너무 많아, 미분할 때 0이 되는 연립방정식을 푸는 것이 사실상 어려움



② 경사하강법을 사용하여 손실 함수의 값이 작아지는 방향을 확인한다

$$x_{new} = x_{old} - \eta \frac{\partial E}{\partial w_{old}}, b_{new} = b_{old} - \eta \frac{\partial E}{\partial b_{old}}$$

w_{new} : 이동 후의 가중치

w_{old} : 이동 전의 가중치

η : 학습률

b_{new} : 이동 후의 바이어스

b_{old} : 이동 전의 바이어스

문제점

값을 움직일 양을 구하기 위해 $\frac{\partial E}{\partial w}$ 와 $\frac{\partial E}{\partial b}$ 를 계산하고 싶지만, 미분할 값이 너무 많아 계산하는 것이 사실상 어렵다.



③ 오차 역전파법을 사용하여 가중치를 결정한다.

$$\frac{\partial E}{\partial w_{kj}^l} = \delta_k^l a_j^{l-1}, \quad \frac{\partial E}{\partial b_k^l} = \delta_k^l$$

$$\delta_k^l \begin{cases} (a_k^l - y_k) \zeta'(x_k^l) \longrightarrow (l \text{이 마지막 계층일 때}) \\ \sum_{i=1}^m (\delta_i^{l+1} w_{ik}^{l+1}) \zeta'(x_k^l) \longrightarrow (l \text{이 마지막 계층이 아닐 때}) \end{cases}$$

E : 손실 함수

w_{kj}^l : l 계층 k 번째 노드의 $l-1$ 계층 j 번째 노드로부터의 가중치

δ_k^l : l 계층 k 번째 노드의 오차

a_j^{l-1} : $l-1$ 계층 j 번째 노드의 출력

b_k^l : l 계층 k 번째 노드의 바이어스

y_k : k 번째 노드의 정답 레이블

$\zeta(x_k^l)$: 활성화 함수 (표준 시그모이드 함수)

m : $l+1$ 계층의 노드 개수

배치 사이즈의 개수만큼 순전파를 하고, 경사하강법과 오차역전파법을 사용해서 가중치와 바이어스를 갱신한다.
②와 ③의 처리를 반복하면 가중치 w 와 바이어스 b 의 근삿값을 찾을 수 있다.



완성된 모델이 90%의 정답률로 손글씨를 판별해 냄



THANK YOU

