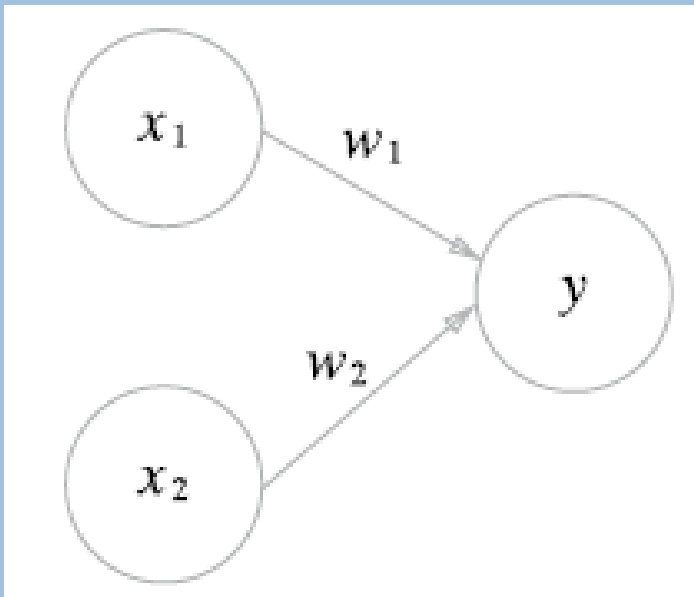


2022-1 겨울방학 딥러닝 분반

분반장: 구은아, 김혜림

퍼셉트론

퍼셉트론



퍼셉트론(Perceptron)

- 다수의 신호를 입력값으로 받아 하나의 신호를 출력하는 알고리즘
- 인간의 뉴런의 신경 신호 처리 방식을 본 따 만든 구조

$$y = \begin{cases} 0 & (w_1x_1 + w_2x_2 \leq \theta) \\ 1 & (w_1x_1 + w_2x_2 > \theta) \end{cases}$$

가중치: 입력값이 결과에 주는 영향력

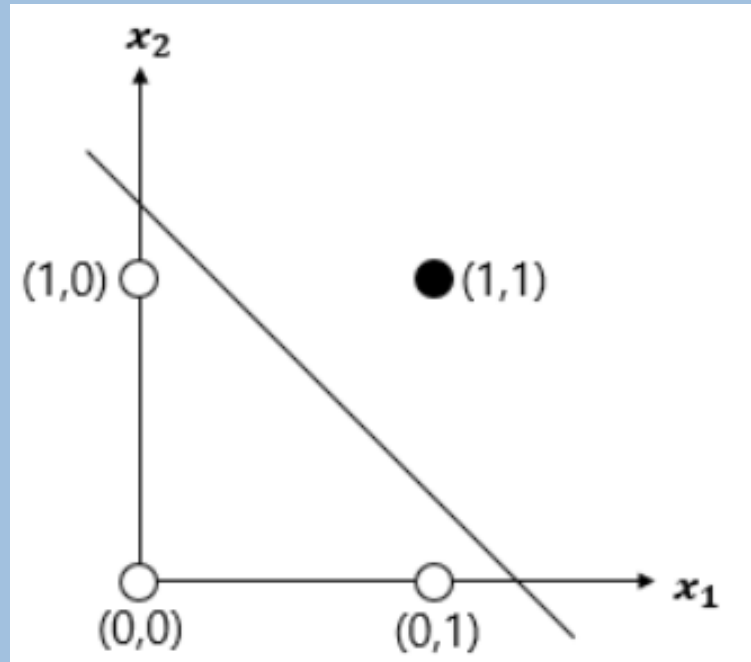
논리회로

AND 게이트

x_1	x_2	y
0	0	0
1	0	0
0	1	0
1	1	1

입력 신호가 모두 참 -> 출력값이 참

ex. $w_1 = 0.5, w_2 = 0.5, \theta = 0.7$



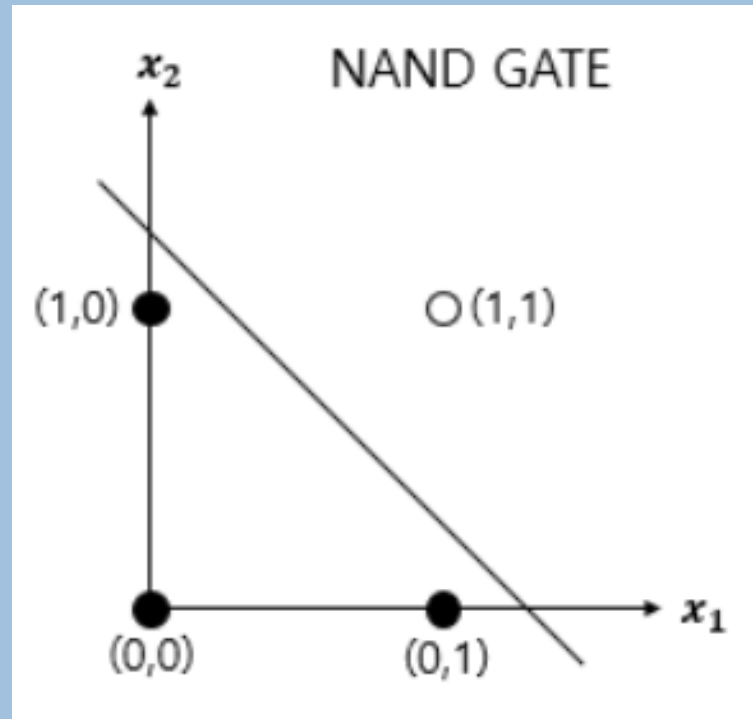
논리회로

NAND 게이트

x_1	x_2	y
0	0	1
1	0	1
0	1	1
1	1	0

입력 신호가 모두 참 -> 출력값이 거짓

ex. $w_1 = -0.5, w_2 = -0.5, \theta = -0.7$



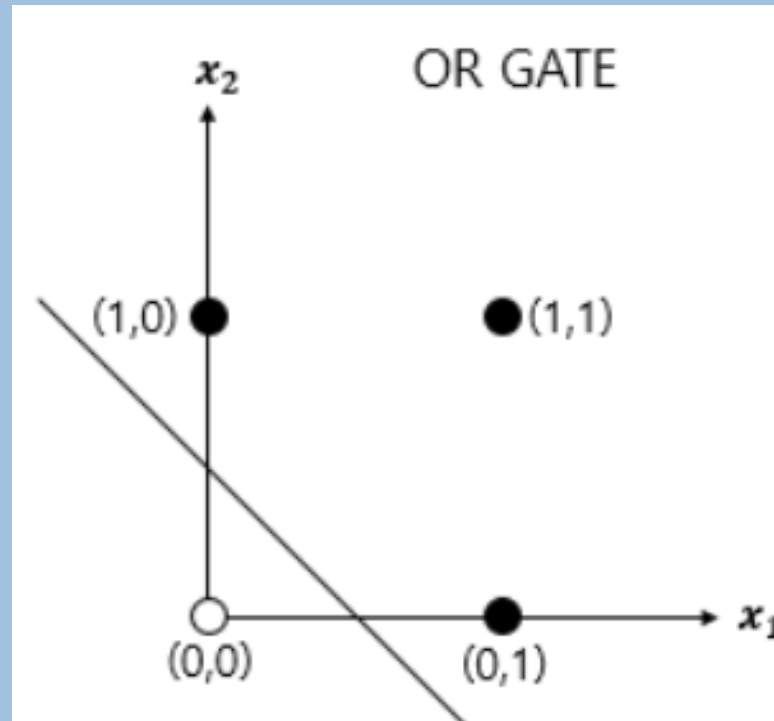
논리회로

OR 게이트

x_1	x_2	y
0	0	0
1	0	1
0	1	1
1	1	1

입력 신호가 모두 거짓 -> 출력값이 거짓

ex. $w_1 = 0.5, w_2 = 0.5, \theta = 0.2$



bias (편향)

$$0.5x_1 + 0.5x_2 \leq 0.7$$

$$0.5x_1 + 0.5x_2 > 0.7$$

$$0.5x_1 + 0.5x_2 \leq 0.2$$

$$0.5x_1 + 0.5x_2 > 0.2$$

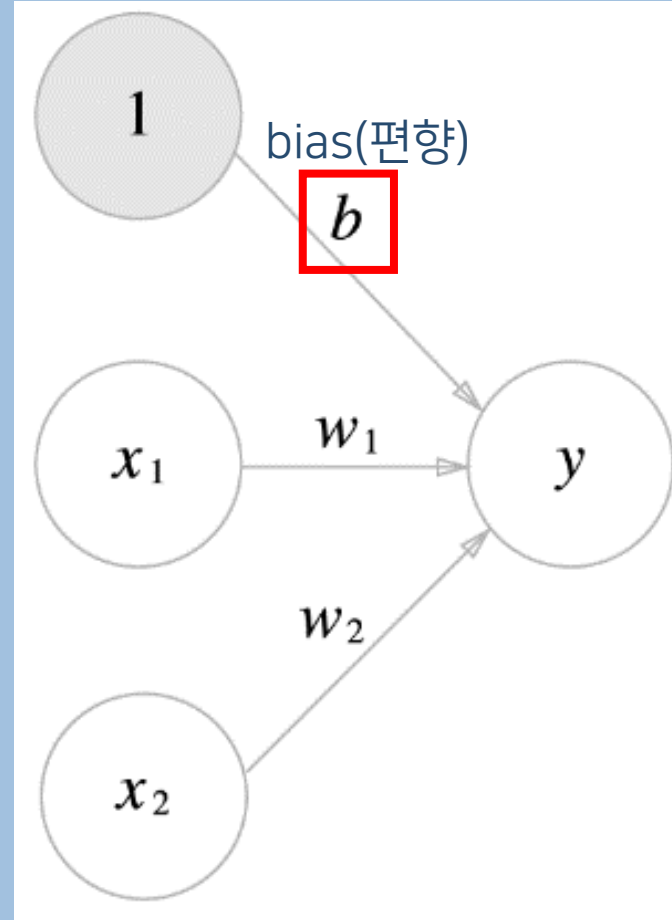


$$-0.7 + 0.5x_1 + 0.5x_2 \leq 0$$

$$-0.7 + 0.5x_1 + 0.5x_2 > 0$$

$$-0.2 + 0.5x_1 + 0.5x_2 \leq 0$$

$$-0.2 + 0.5x_1 + 0.5x_2 > 0$$



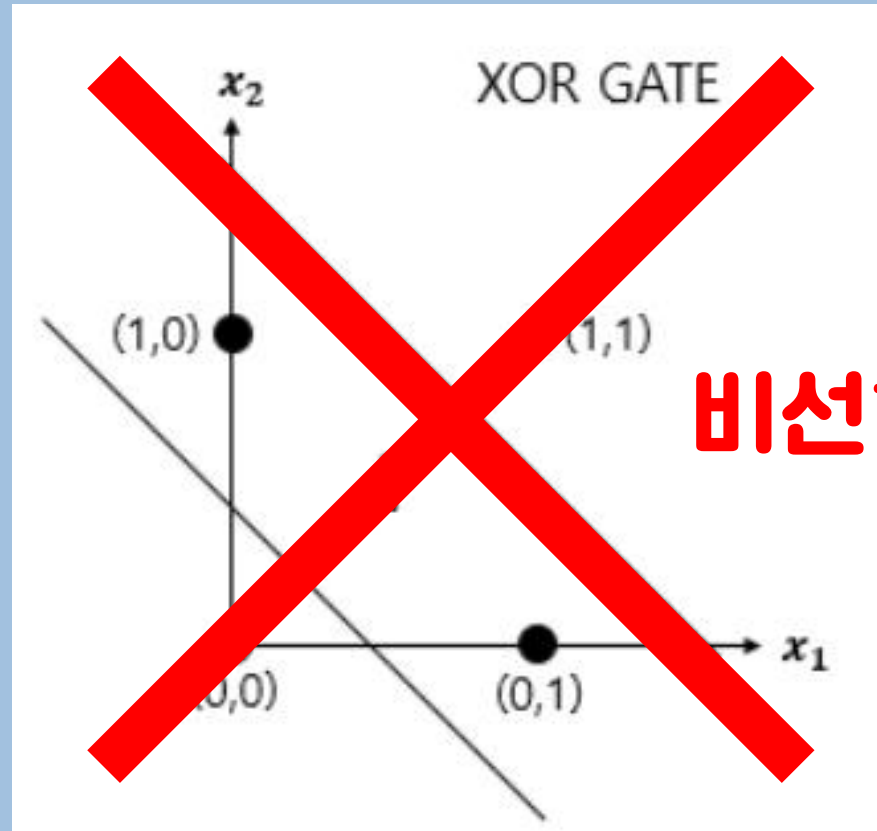
다층 퍼셉트론

다층 퍼셉트론

XOR 게이트

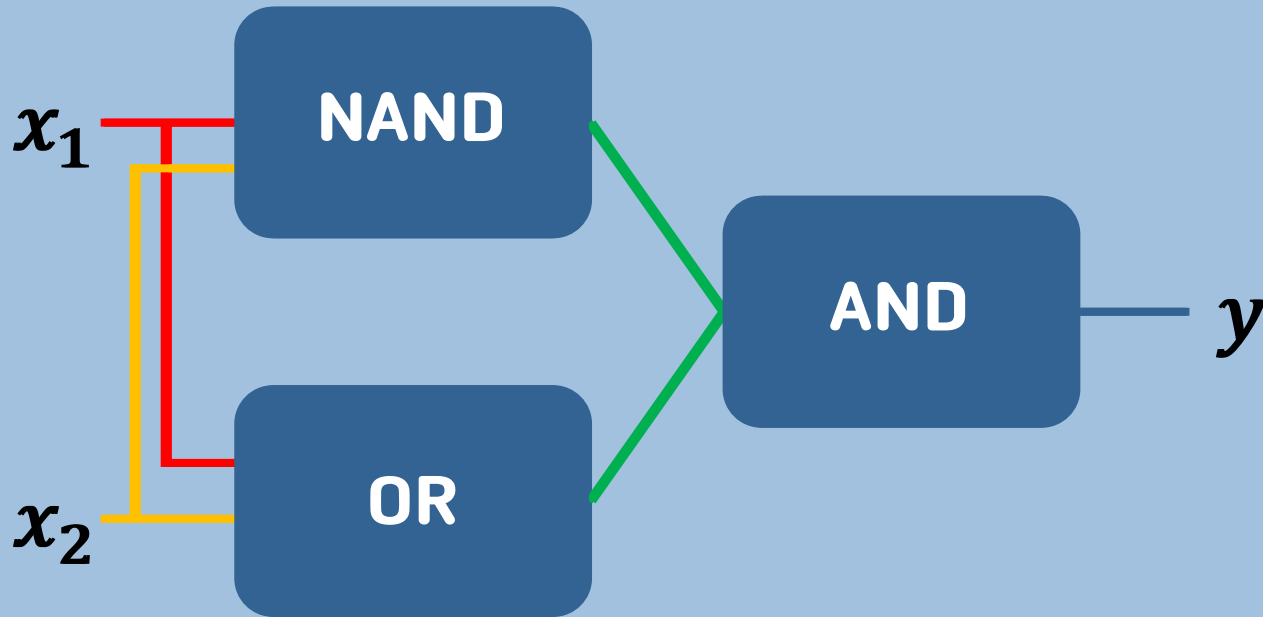
x_1	x_2	y
0	0	0
1	0	1
0	1	1
1	1	0

입력 신호가 다름 -> 출력값이 참
ex. ?



비선형구조

다층 퍼셉트론

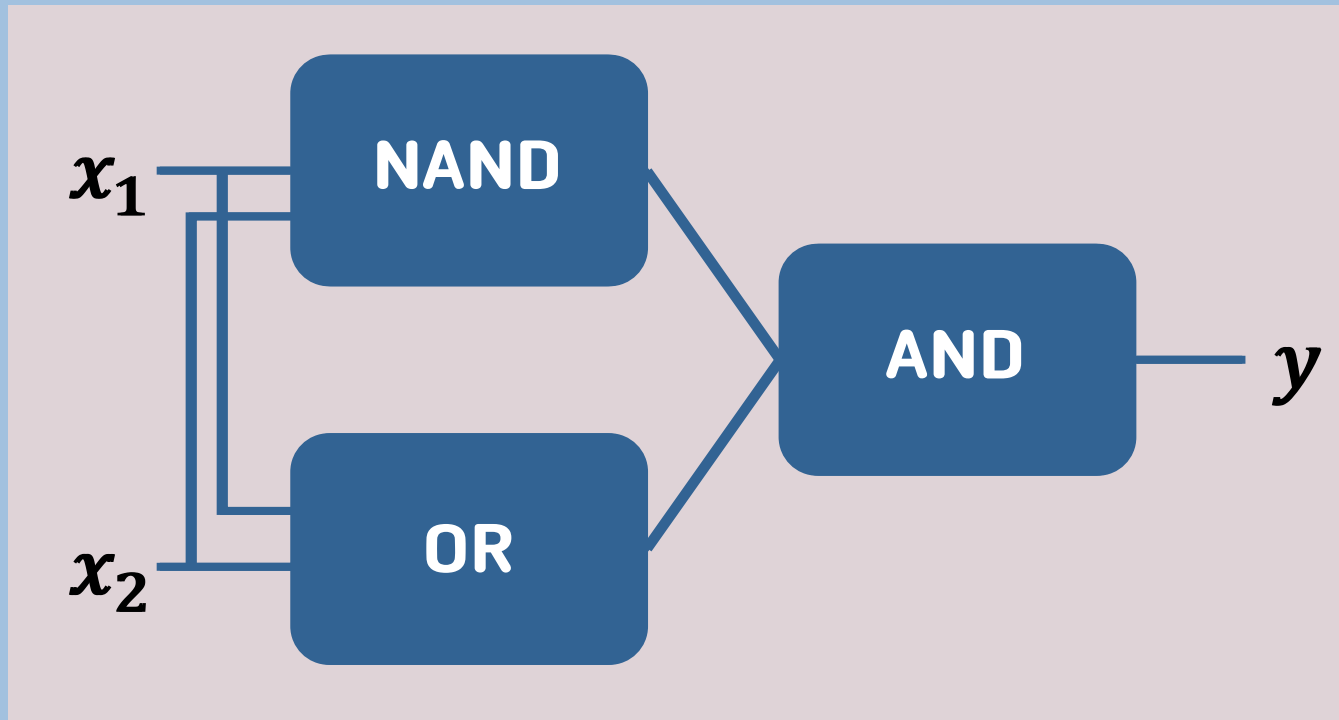


XOR 게이트

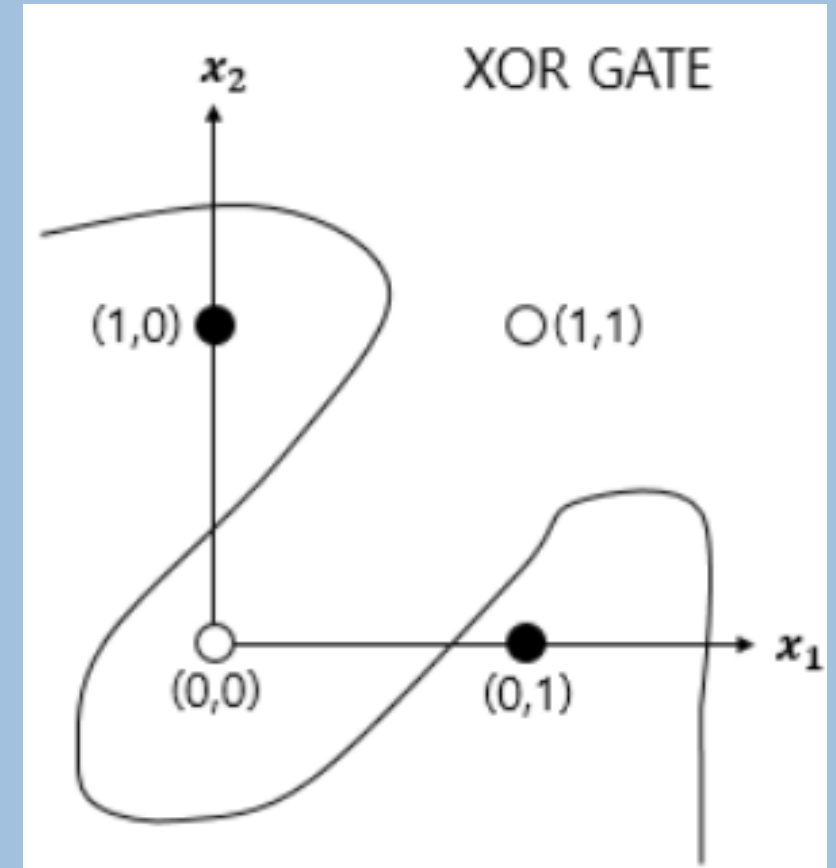
x_1	x_2	s_1	s_2	y
0	0	1	0	0
1	0	1	1	1
0	1	1	1	1
1	1	0	1	0

입력 신호가 다름 -> 출력값이 참

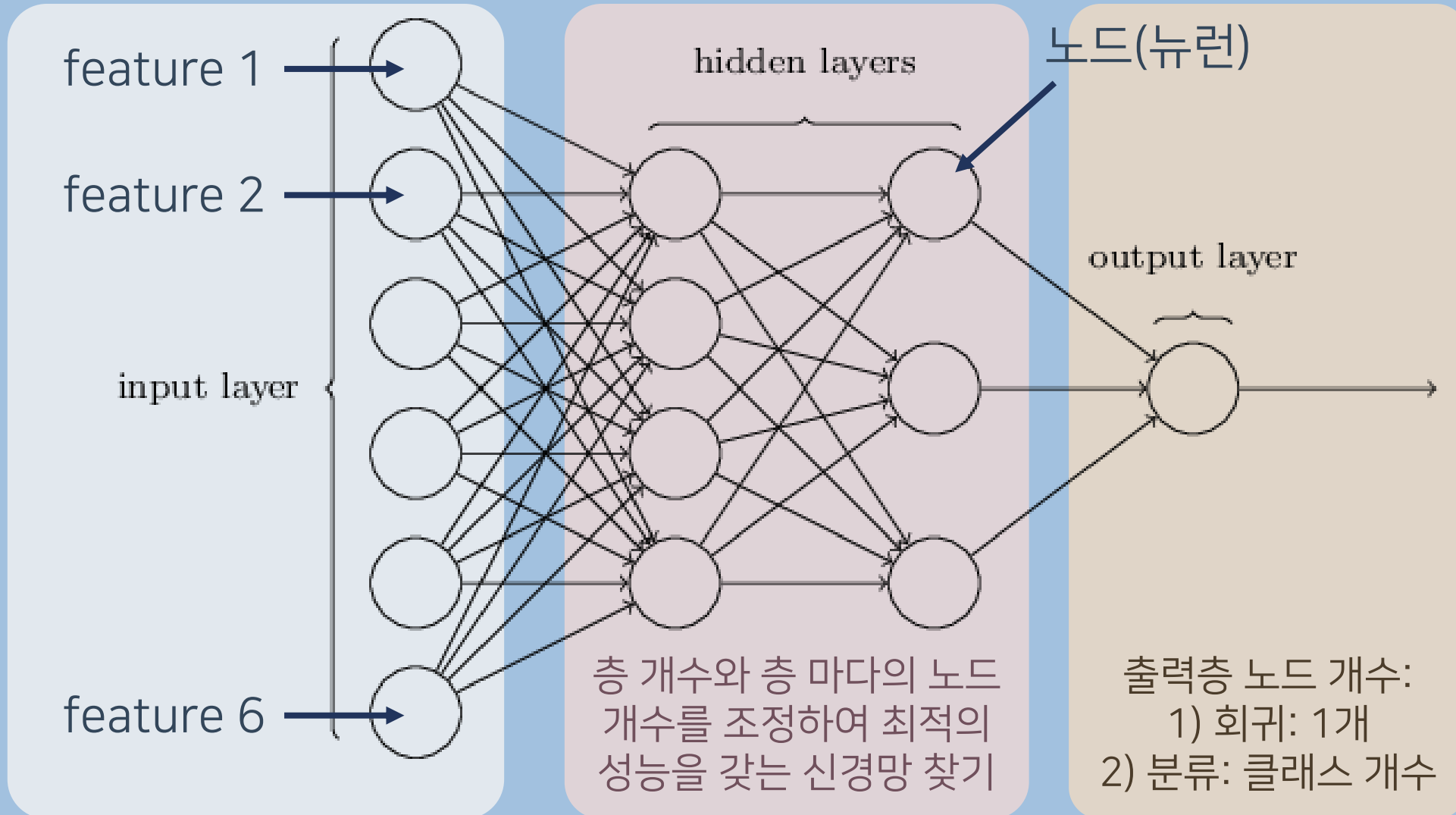
다층 퍼셉트론



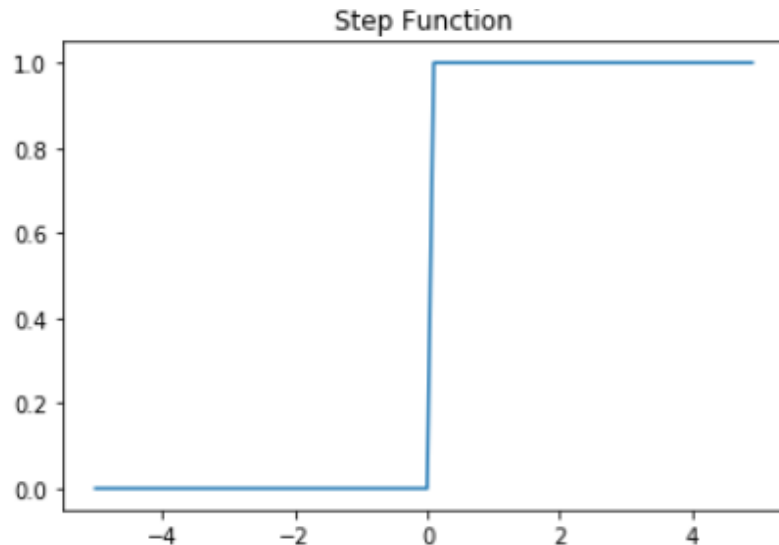
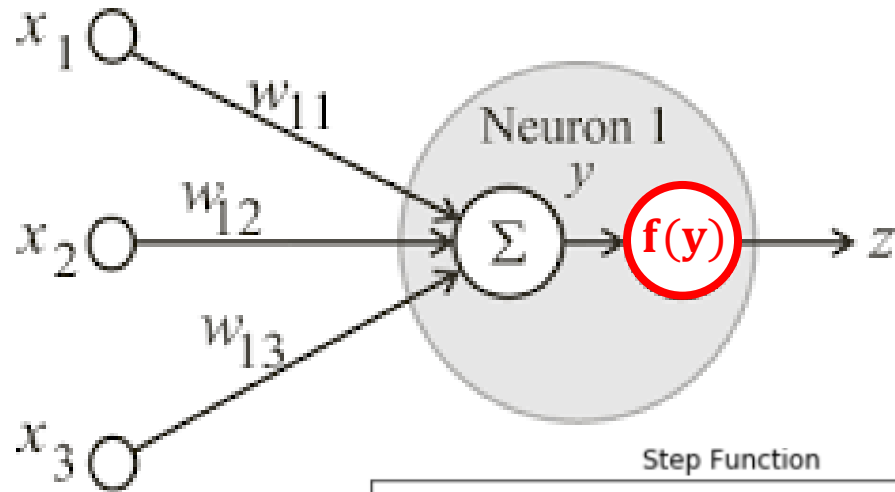
다층 퍼셉트론
Multi layer perceptron



다층 퍼셉트론



※활성화 함수

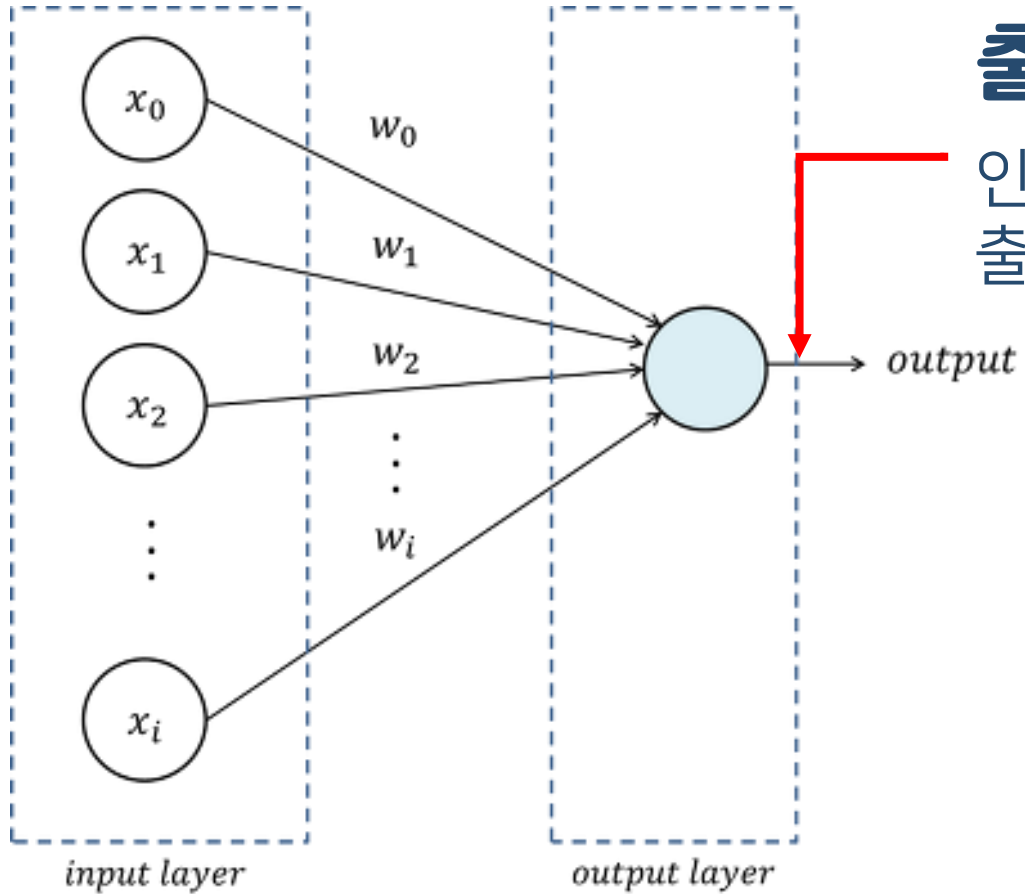


활성화 함수 activation function

인공신경망에서 계산된 가중합을 출력값으로 변환하는 함수

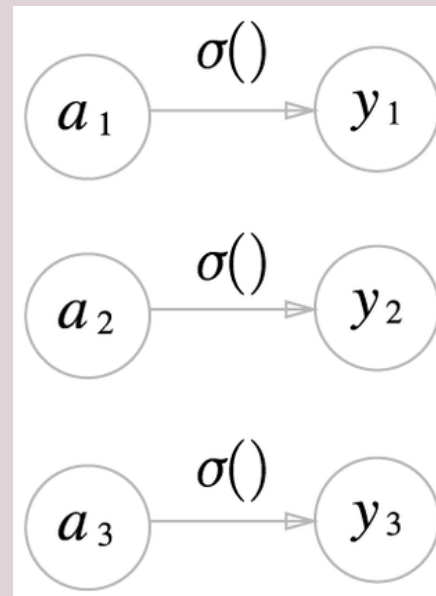
출력층 함수

출력층 함수



출력층 함수

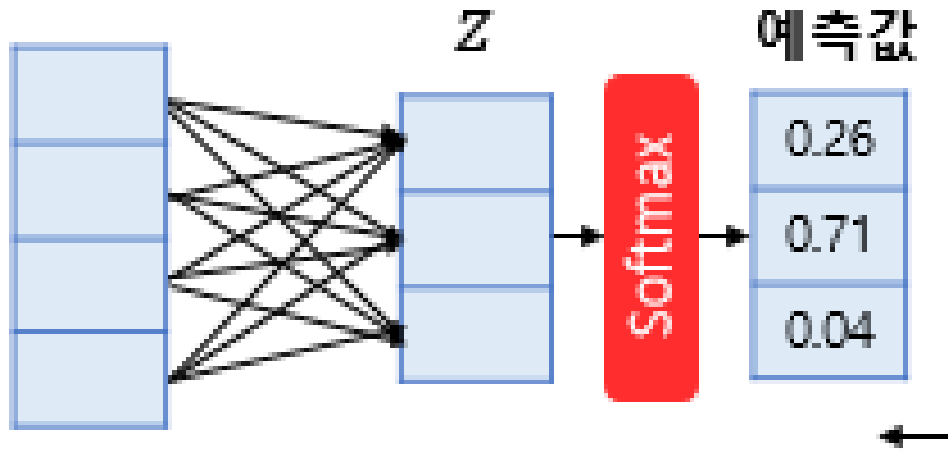
인공 신경망에서 계산된 최종값을 출력값으로 변환하는 함수



ex. 항등함수

- 입력값을 그대로 출력하는 함수
- 보통 회귀 문제에서 사용

소프트맥스 함수



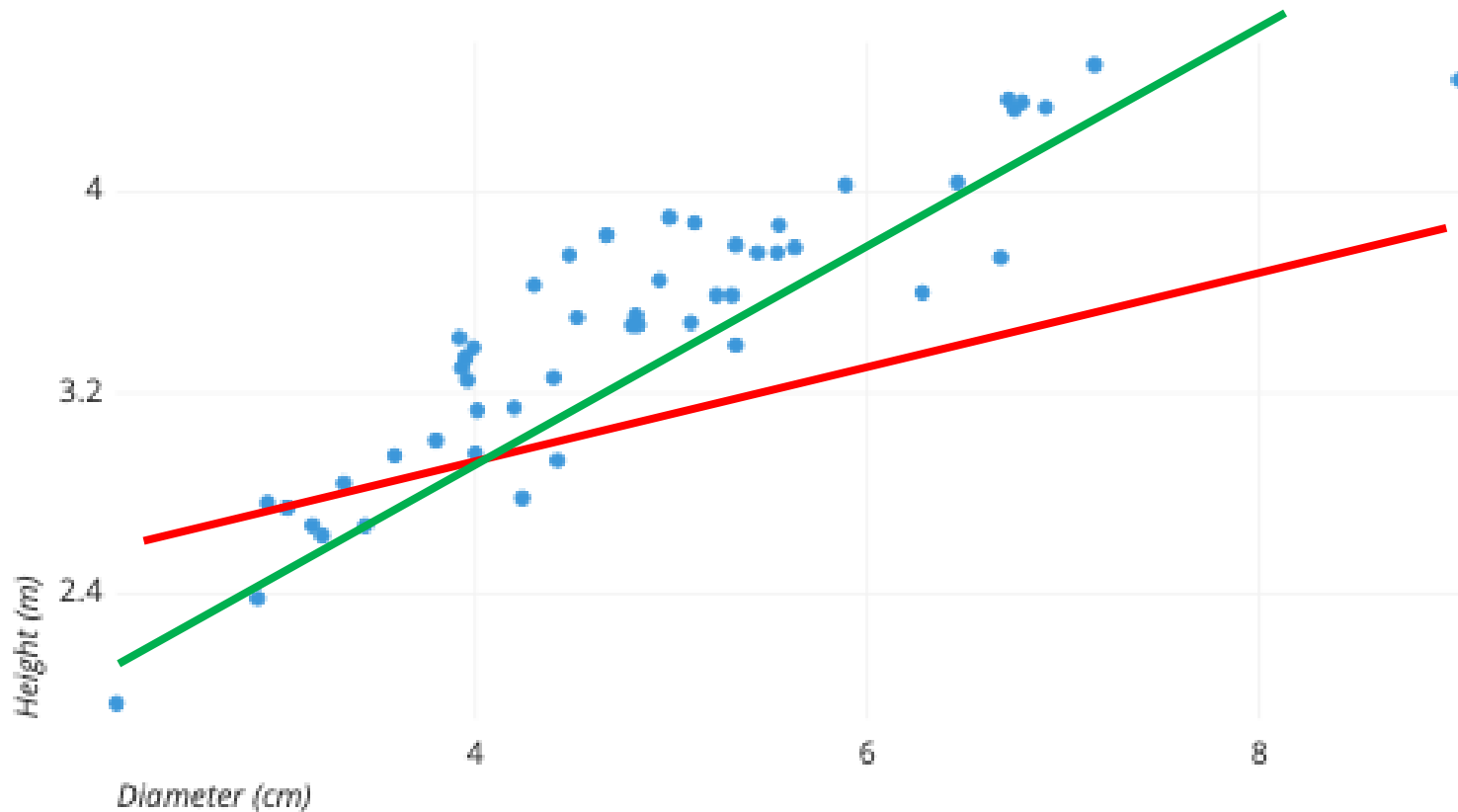
소프트맥스 함수 softmax function

- 전체에서 입력값에 해당하는 확률을 구하는 함수
- 보통 다중 분류 문제에서 사용(multi classification)
- 소프트맥스의 출력은 모든 입력값의 영향을 받음

$$y_k = \frac{\exp(a_k)}{\sum_{i=1}^n \exp(a_i)}$$

loss function

loss function



Q. 어떻게 학습해야 할까요?

A. 실제값과 예측값이 비슷해지도록 학습해야 합니다.

loss function

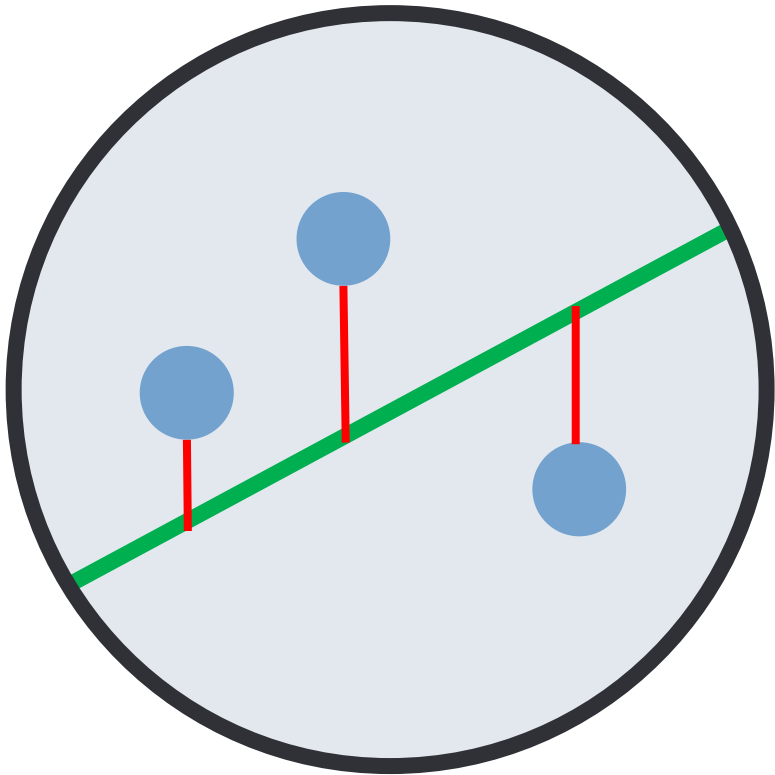
실제값과 예측값을 비슷하게 만들어야 함

실제값과 예측값의 차이(cost)를 측정하는 함수 필요

loss function 필요!

**딥러닝 학습의 목적:
loss function의 최소화**

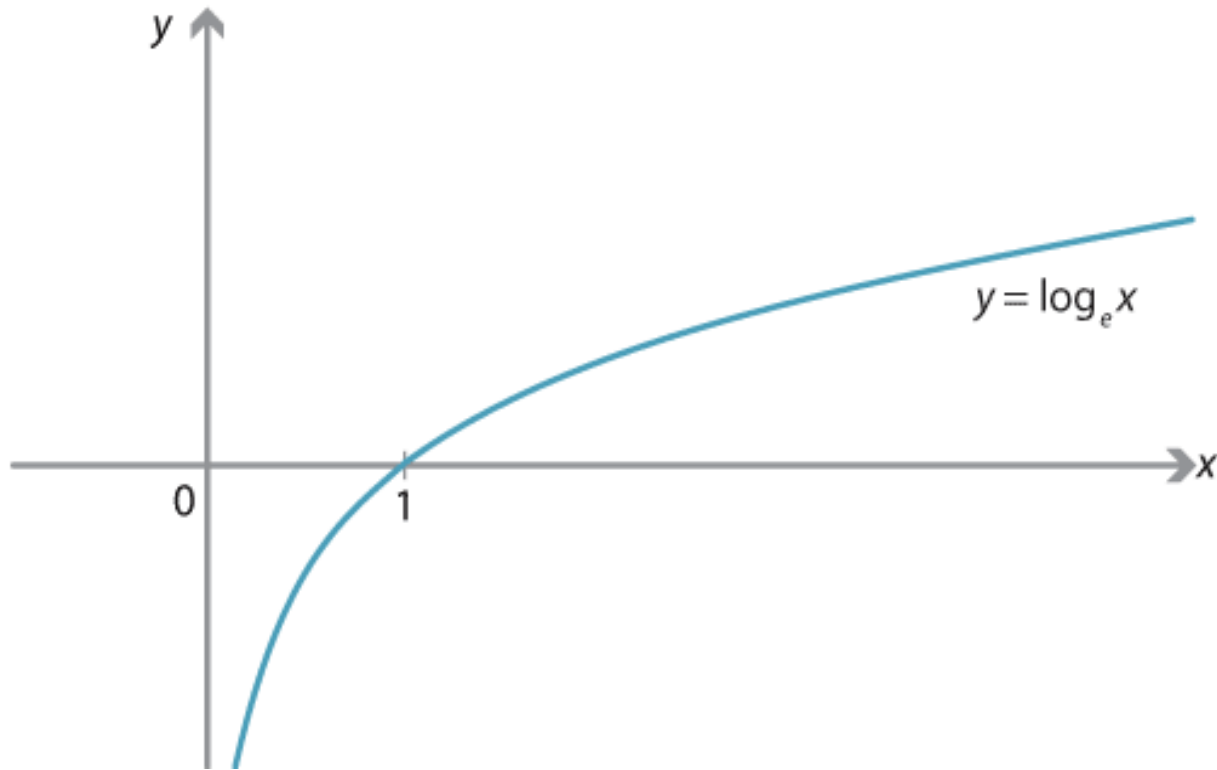
평균제곱오차 Mean Square Error



$$MSE = \frac{1}{n} \sum_{k=1}^n (y_k - t_k)^2$$

- 실제값과 예측값의 차이의 제곱합의 평균
- MSE가 작을수록 성능이 좋음

교차 엔트로피 오차 Cross Entropy Error

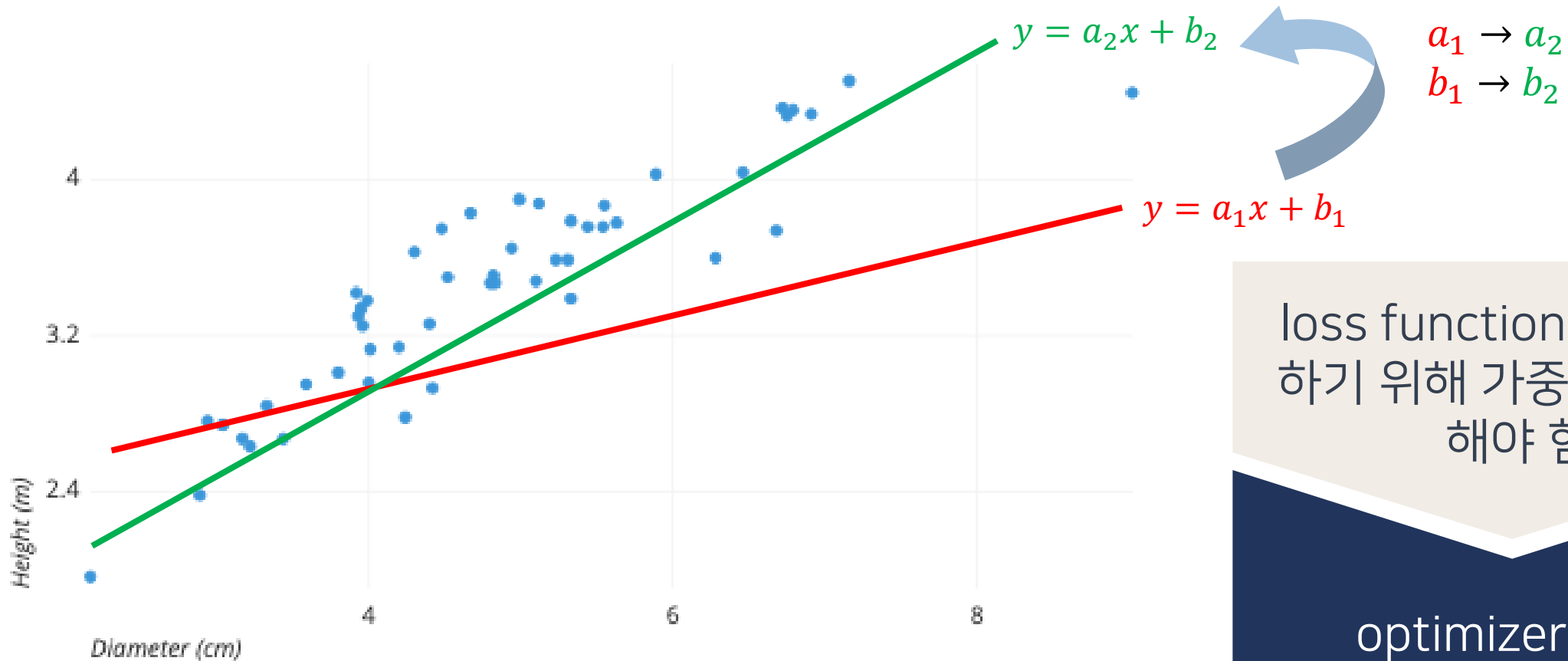


$$CEE = - \sum_k t_k \log y_k$$

- 예측값의 자연로그와 실제값을 곱한 것의 총합
- 정답일 때의 출력이 전체 값을 결정
- CEE가 작을수록 성능이 좋음

경사하강법

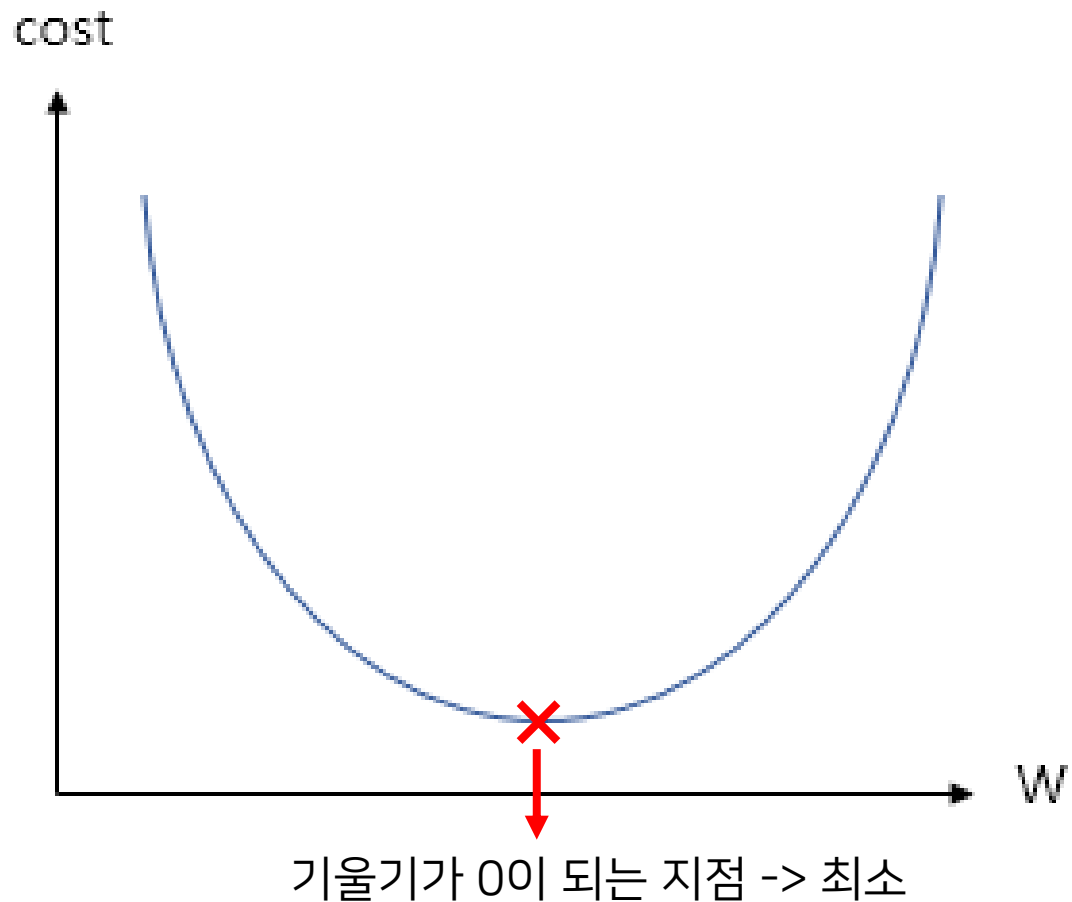
Optimizer



loss function을 최소화
하기 위해 가중치를 갱신
해야 함

optimizer 필요!

경사하강법

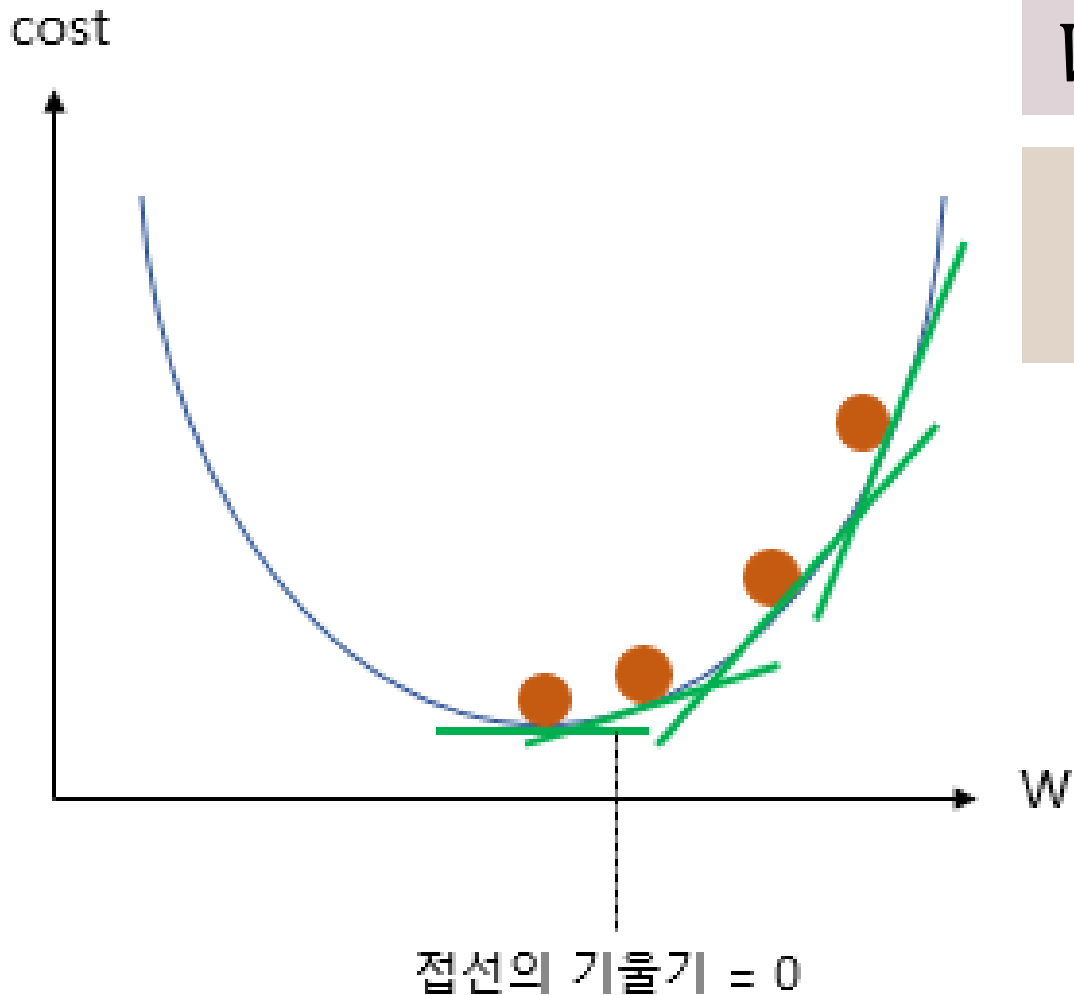


경사하강법 프로세스

1. loss function을 가중치에 대해서 미분
2. 미분한 함수가 0이 되는 지점(접선의 기울기=0)을 찾기

※ 딥러닝의 loss function은 가중치에 대해서 미분 가능한 함수이어야 함!

경사하강법



$$W_{new} \leftarrow W - \alpha \times \Delta$$

$$\Delta = \frac{\partial}{\partial W} cost(W)$$

$cost(W)$: loss function
 α : learning rate(학습률)

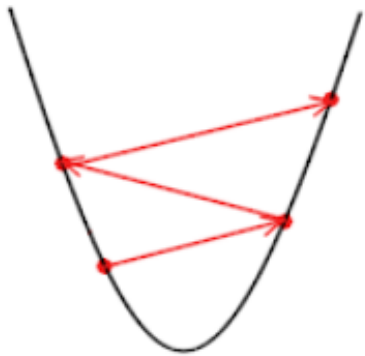
경사하강법

최적의 가중치를 찾기 위해 접선의 기울기가 음인 방향으로 이동하는 것을 반복하는 optimizer

learning rate

learning rate(학습률): 가중치를 얼마나 크게 변경할 지 결정하는 파라미터
경험적으로 0.1~0.0001 범위에 적당한 학습률이 존재

Big Learning Rate

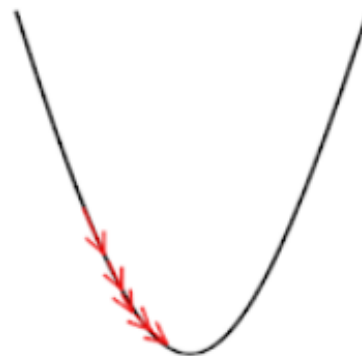


가중치가
오히려 발산

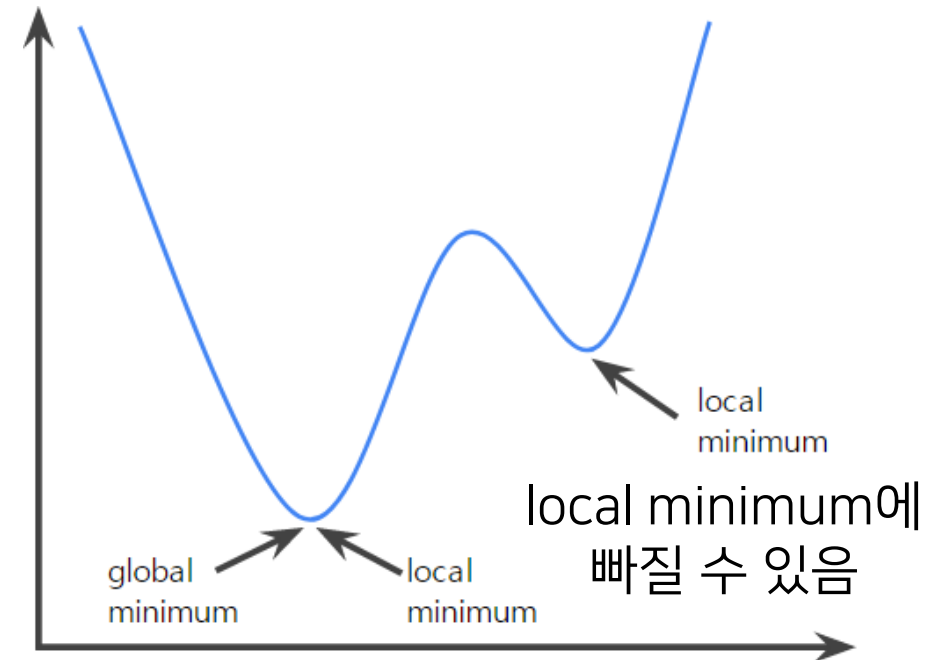
Just right



Too small

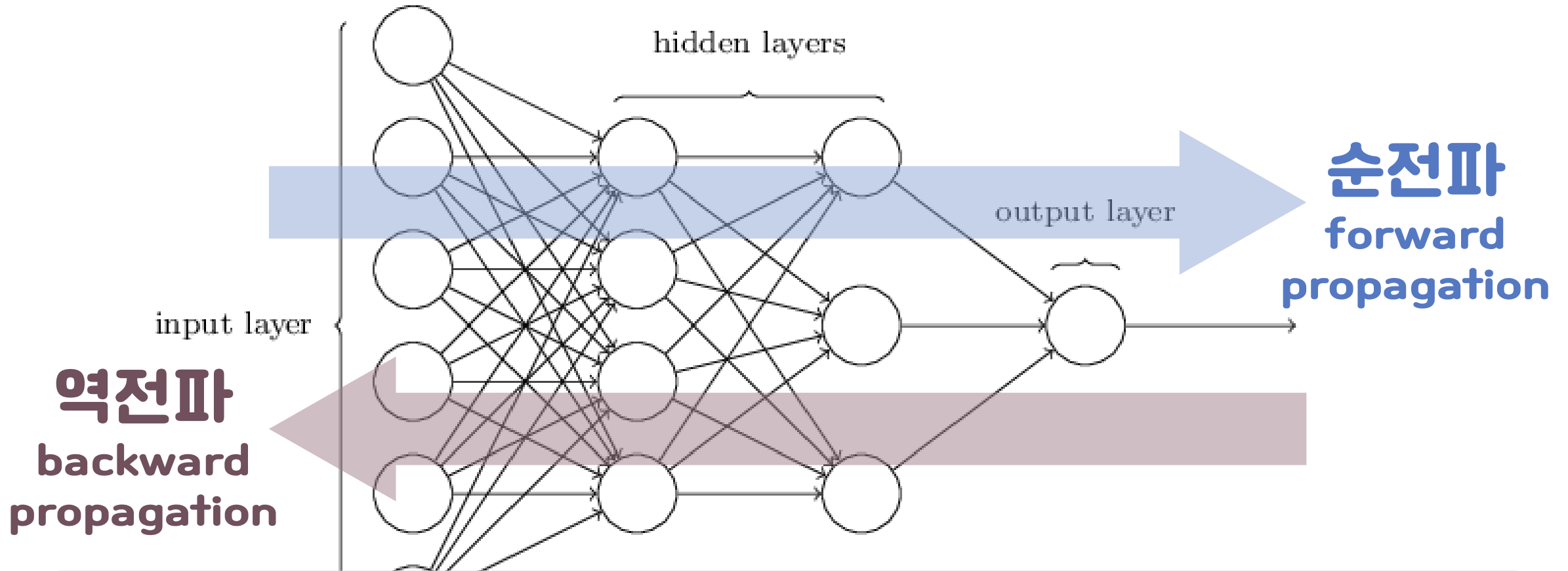


학습이
너무 느림



역전파

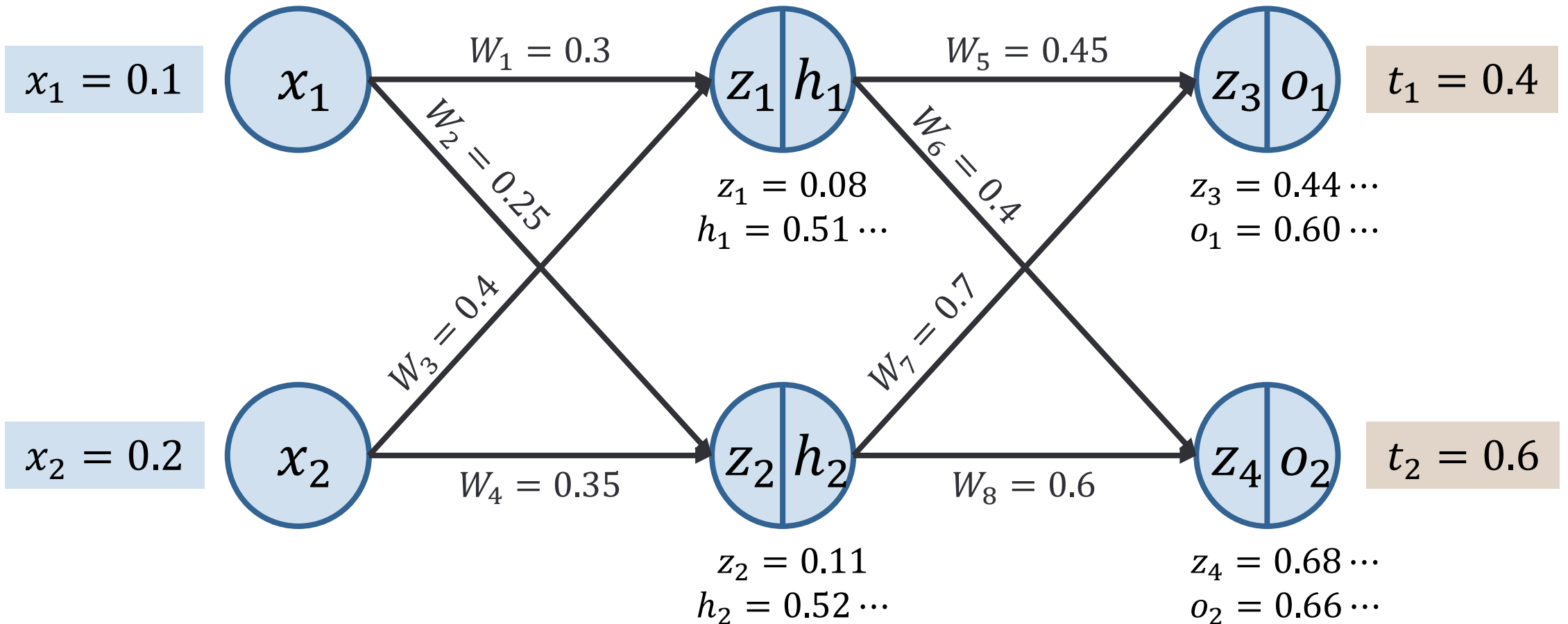
역전파



인공신경망은 역전파 과정을 통해 기울기를 계산하고 가중치를 업데이트한다.

역전파

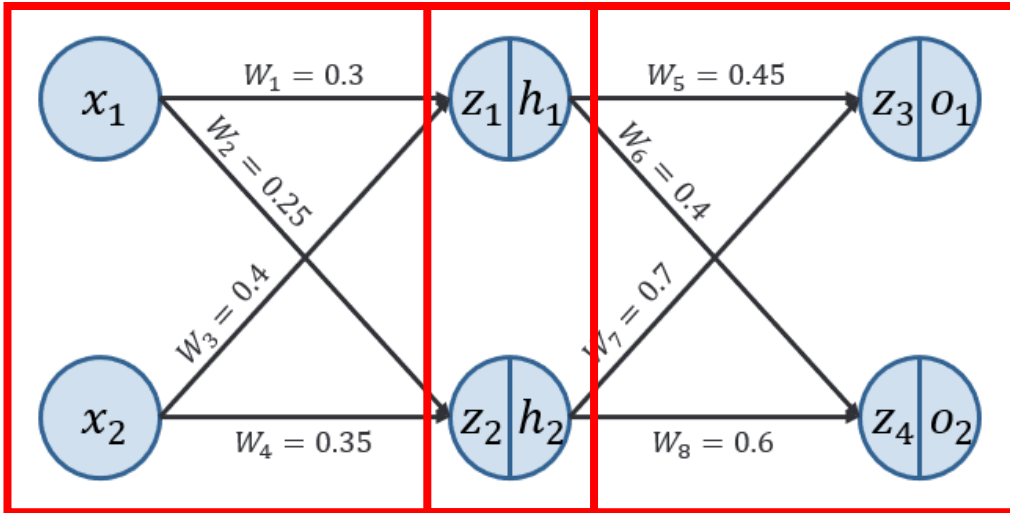
활성화 함수: 시그모이드 함수



Chain Rule

$$W_{new} \leftarrow W - \alpha \times \frac{\partial cost(W)}{\partial W}$$

$$cost(W) = \frac{1}{2} (0.4 - 0.60 \dots)^2 + \frac{1}{2} (0.6 - 0.66 \dots)^2$$

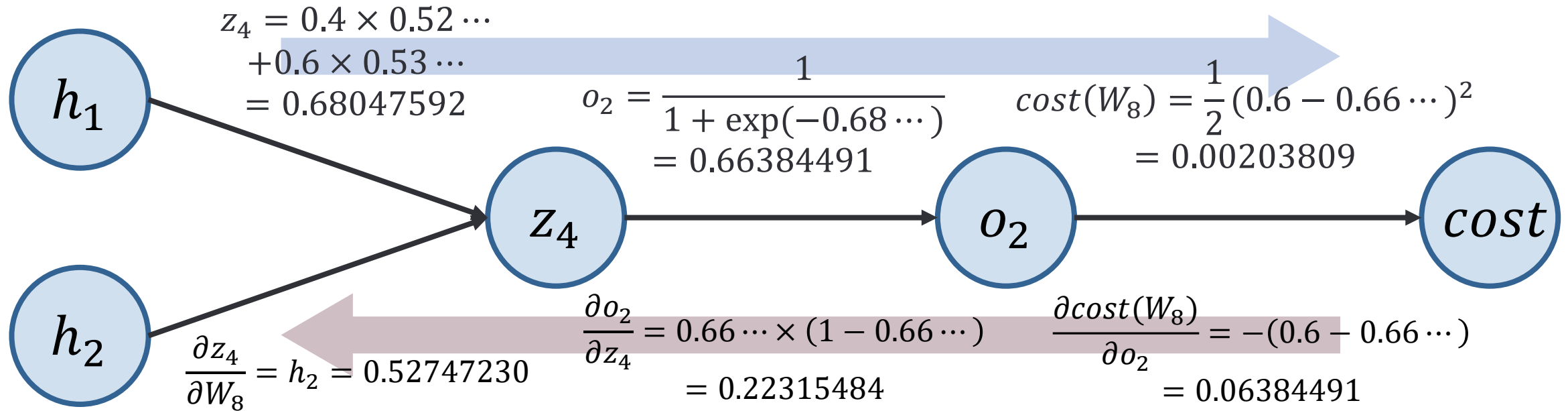


$$\frac{\partial cost(W)}{\partial W} = \frac{\partial cost(W)}{\partial o} \times \frac{\partial o}{\partial z} \times \frac{\partial z}{\partial W}$$

$$\frac{\partial cost(W)}{\partial W} = \frac{\partial cost(W)}{\partial h} \times \frac{\partial h}{\partial z} \times \frac{\partial z}{\partial W}$$

chain rule을 이용하여 쉽게 기울기를 구할 수 있습니다.

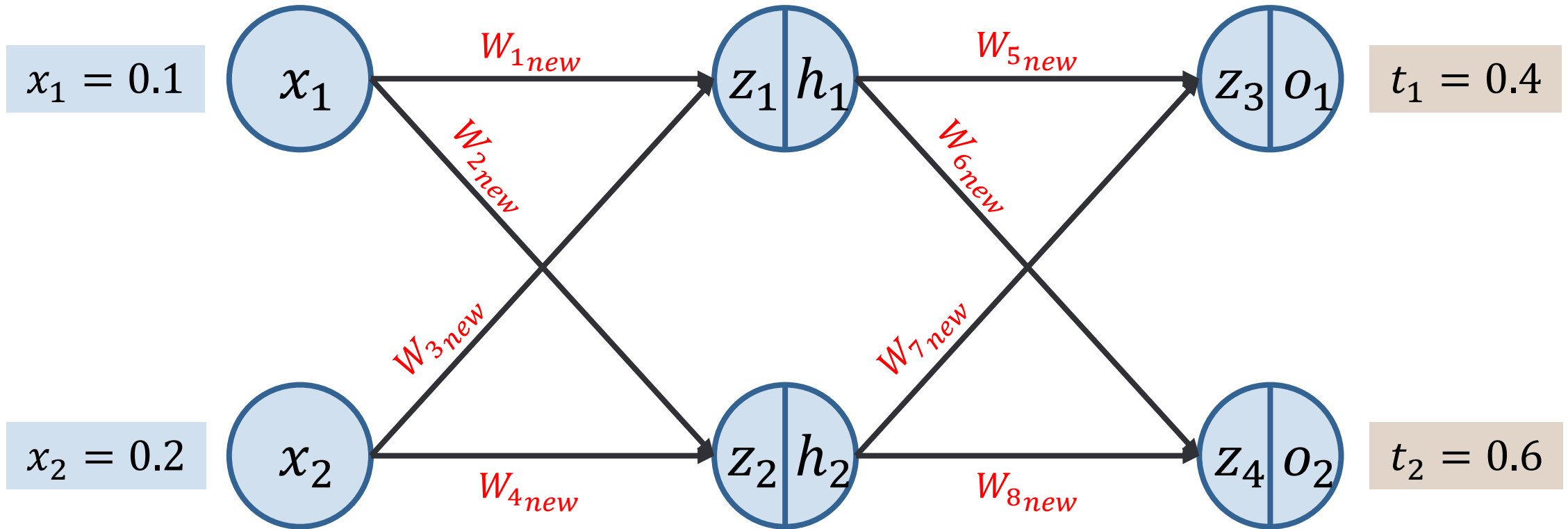
역전파



$$\therefore \frac{\partial cost(W_8)}{\partial W_8} = 0.06384491 \times 0.22315484 \times 0.52747230 = 0.00751505$$

$$0.6 - 0.5 \times 0.00751505 \rightarrow W_{8_{new}} = 0.59624247$$

역전파



업데이트 된 가중치로 다시 순전파를 진행하면 오차가 줄어드는 것을 확인할 수 있습니다.
인공 신경망의 학습은 이와 같이 순전파와 역전파를 반복하며 진행됩니다.

수고하셨습니다!
다음 주에 새로운 내용으로 만나요~