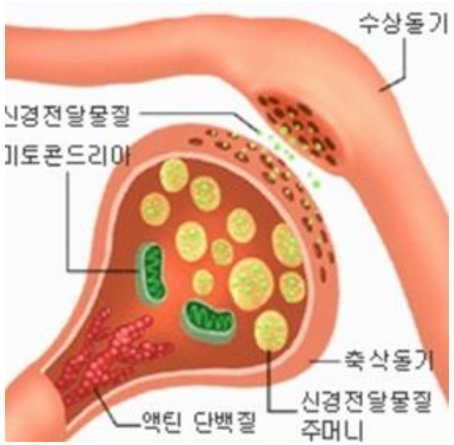
인공 신경망 이해

신경망 구조

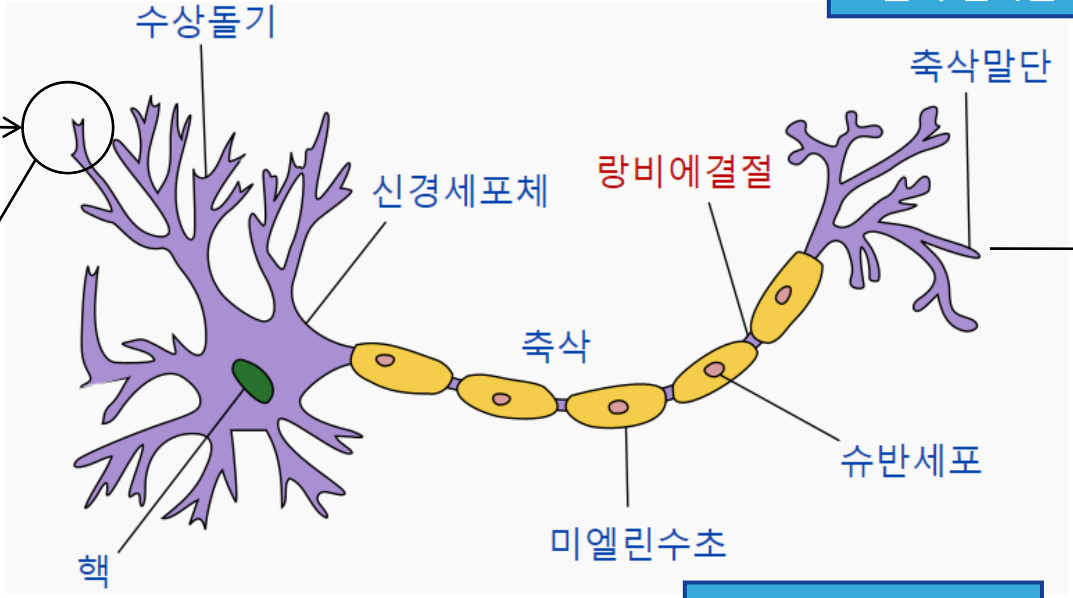
입력신호를 받아들임

외부 자극

(축삭말단)



신경전달물질의 양에 따라
신호 전달 강도 결정



입력된 신호 처리

처리된 신호 전달

출력 결과를 보냄

신호출력

신경망에서의 정보 처리 과정

Input —(시냅스)— 수상돌기 → 세포체 → 축삭말단 → output

Input → Input x 시냅스연결강도 → Input x 시냅스연결강도 + Input x 시냅스연결강도 → → output

신호수용

신호와 연결강도가 곱해지는 부분

신호처리

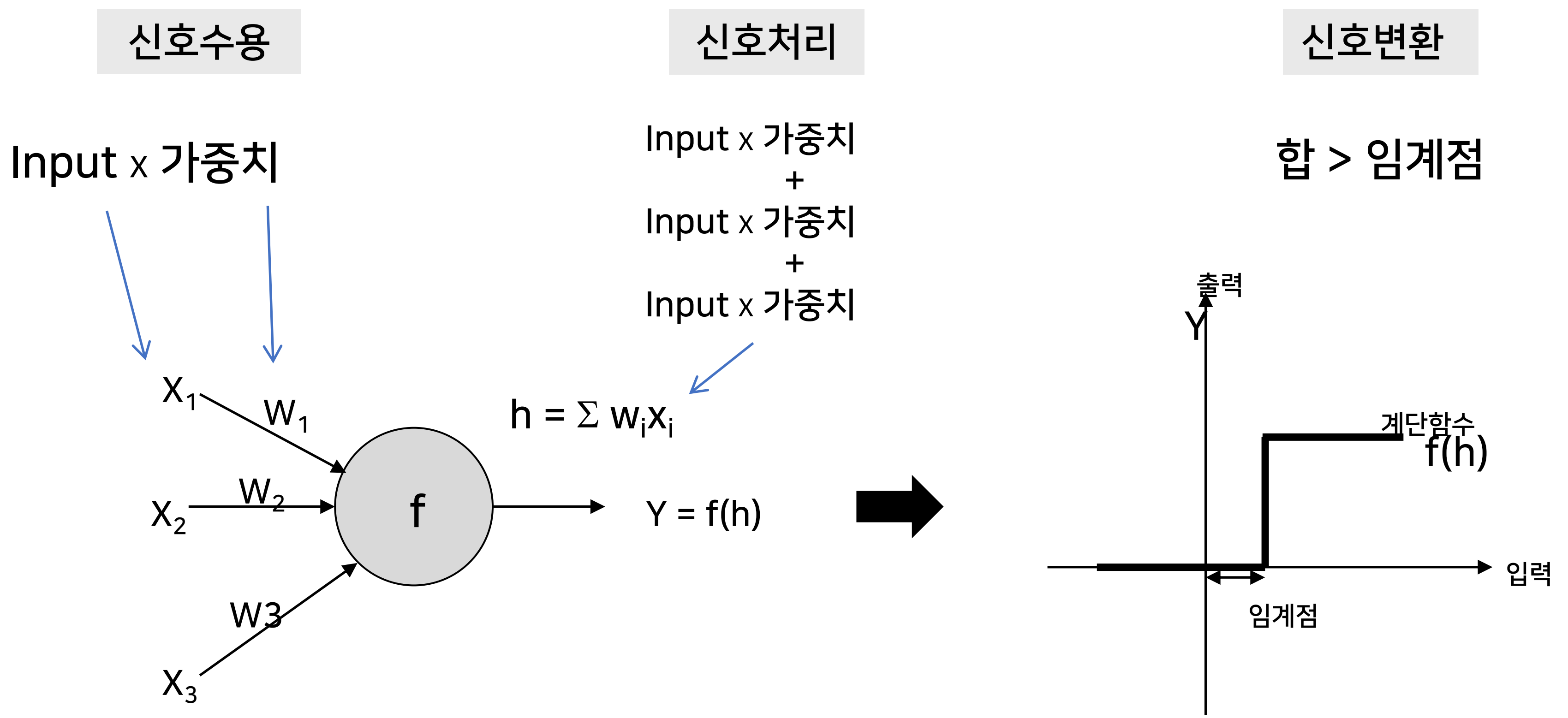
데이터가 모아지는 부분

신호변환

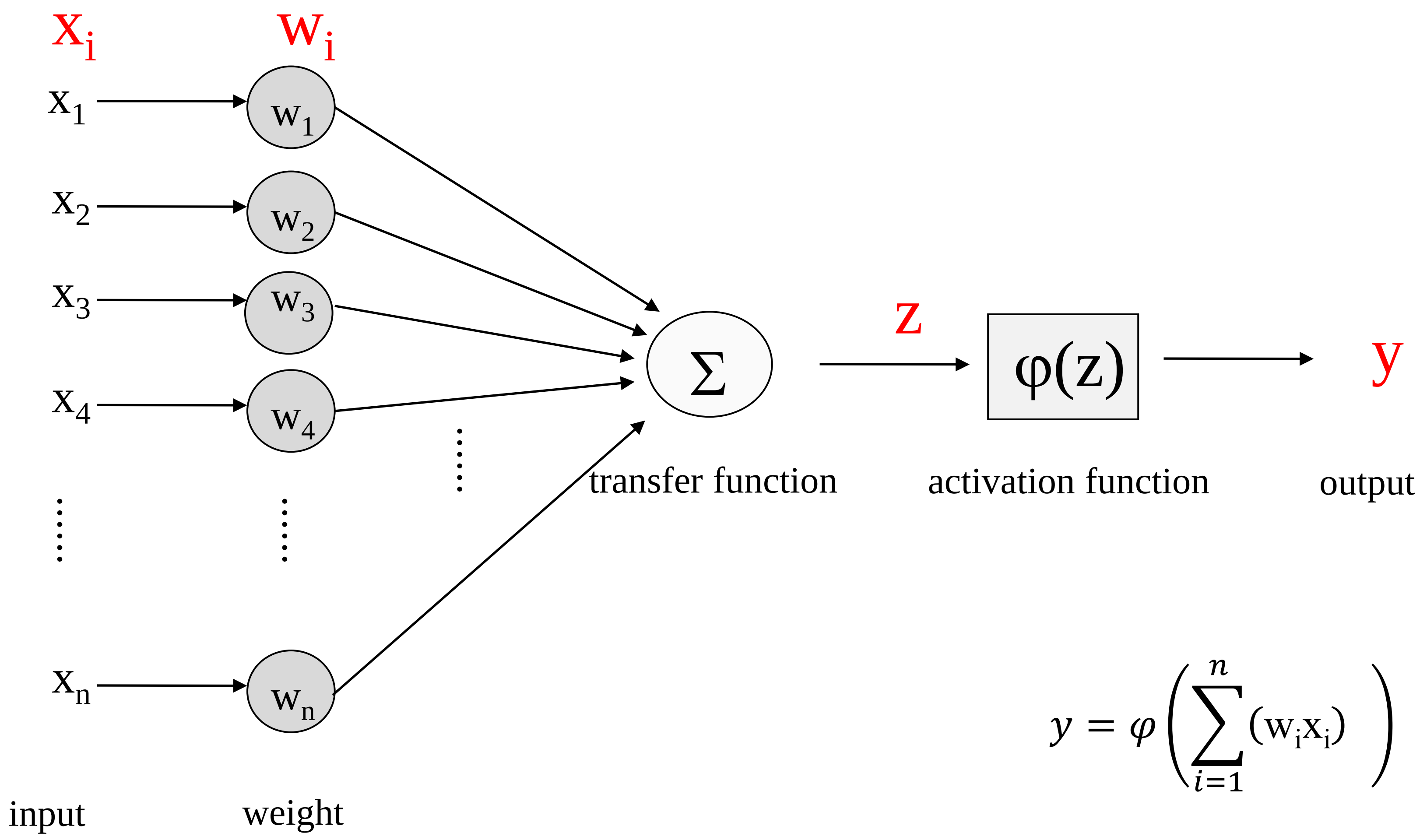
모아진 데이터들의 신호를 0과 1로 변환하는 부분

신경망에 대한 수학적 모델링

- 신경망 모델은 입력 신호와 가중치를 곱하고 곱해진 것을 모두 더한 다음 계단 함수를 통과시켜서 신호를 출력하는 함수



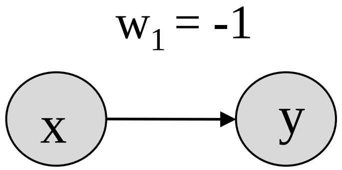
신경망에 대한 수학적 모델링



$$y = \varphi\left(\sum_{i=1}^{n} (\mathrm{w_i x_i})\right)$$

간단한 신경망 예제

NOT 게이트



$\Theta = -0.49$
if $(z <= -0.49)$ $y = 0$
if $(z > -0.49)$ $y = 1$

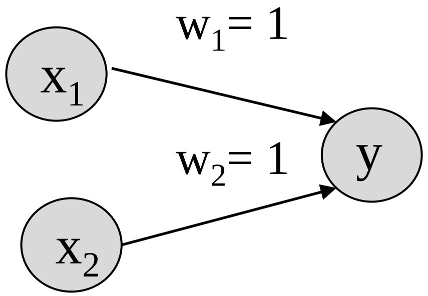
x	z	NOT
0	0	1
1	-1	0

$z = w_1 x$
$y = \varphi(z)$

if $x = 0$
$z = -1 \times 0 = 0$ $(z > -0.49)$
$y = 1$

if $x = 1$
$z = -1 \times 1 = -1$ $(z <= -0.49)$
$y = 0$

AND 게이트



$\Theta = 1.5$
if $(z <= 1.5)$ $y = 0$
if $(z > 1.5)$ $y = 1$

x_1	x_2	z	AND
0	0	0	0
0	1	1	0
1	0	1	0
1	1	2	1

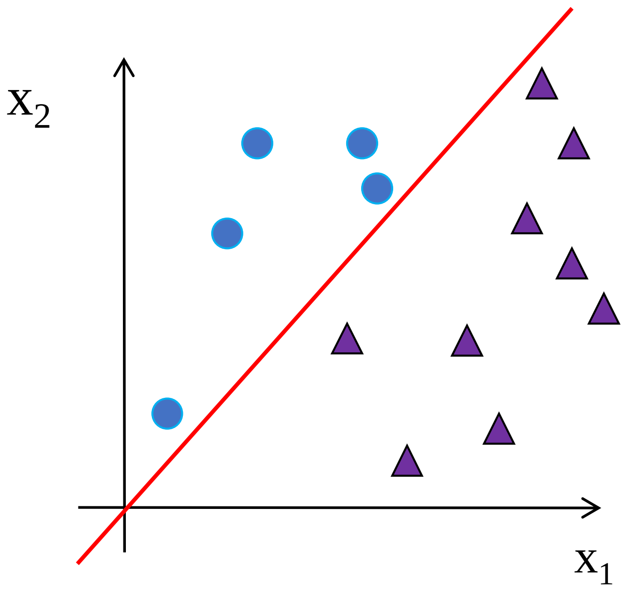
$z = w_1 x_1 + w_2 x_2$
$y = \varphi(z)$

if $x_1 = 1,\ x_2 = 1$
$z = 1 \times 1 + 1 \times 1 = 2$
$(z > 1.5)$
$y = 1$

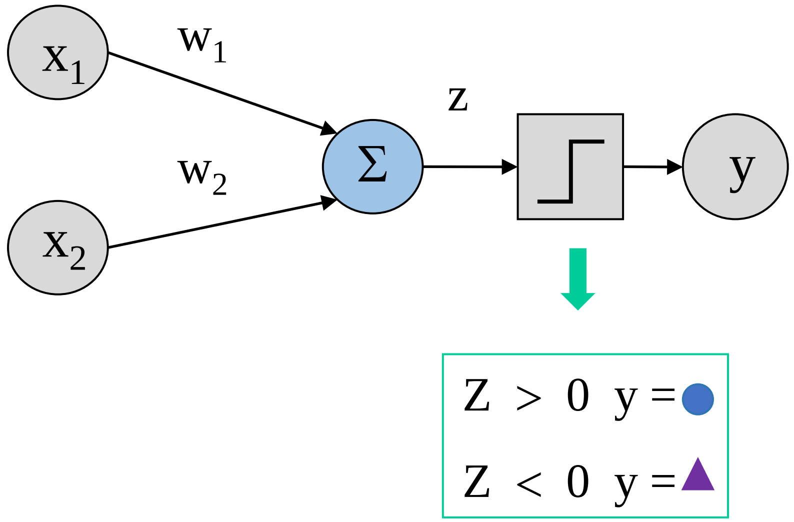
if $x_1 = 0,\ x_2 = 0$
$z = 1 \times 0 + 1 \times 0 = 1$
$(z <= 1.5)$
$y = 0$

신경망을 이용한 데이터 분류

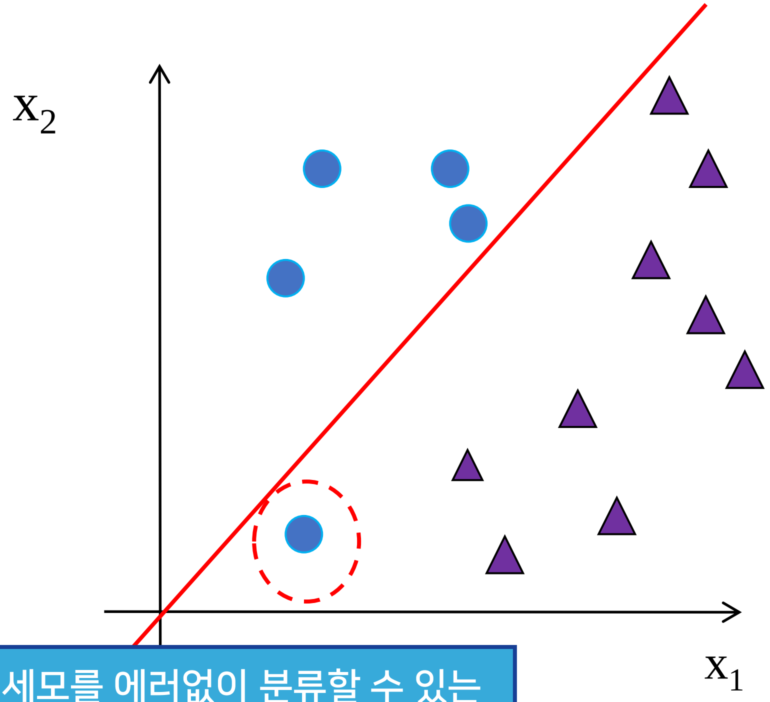
$w_1x_1 + w_2x_2 = 0$



입력 값 x_1 , x_2 가 0일 때 원점을 지남



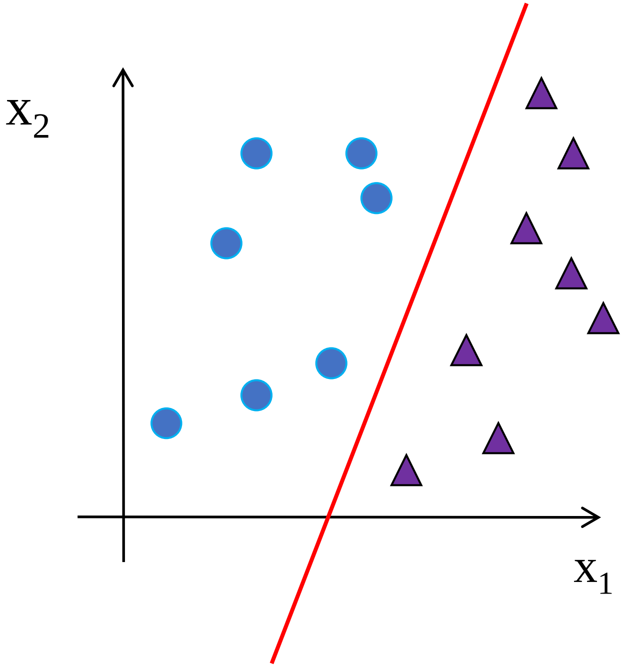
신경망을 이용한 데이터 분류

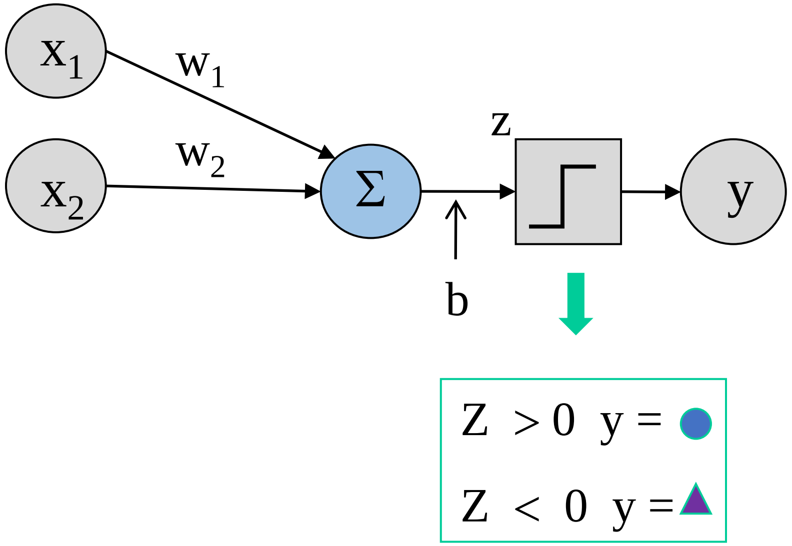


동그라미와 세모를 에러없이 분류할 수 있는
원점을 지나는 직선은 존재하지 않음

바이어스가 있는 신경망

$w_1x_1 + w_2x_2 + b = 0$



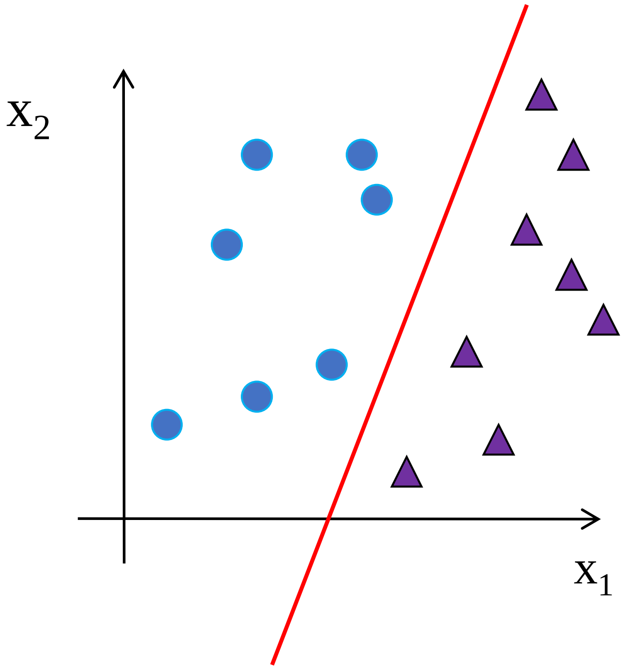


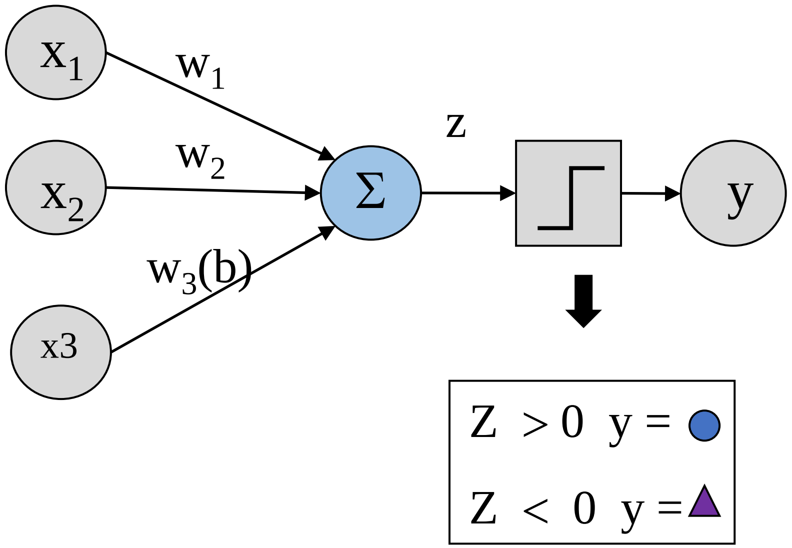
입력 값 x_1 , x_2 가 0일 때 원점을 지나지 않아도 됨

바이어스가 있는 신경망

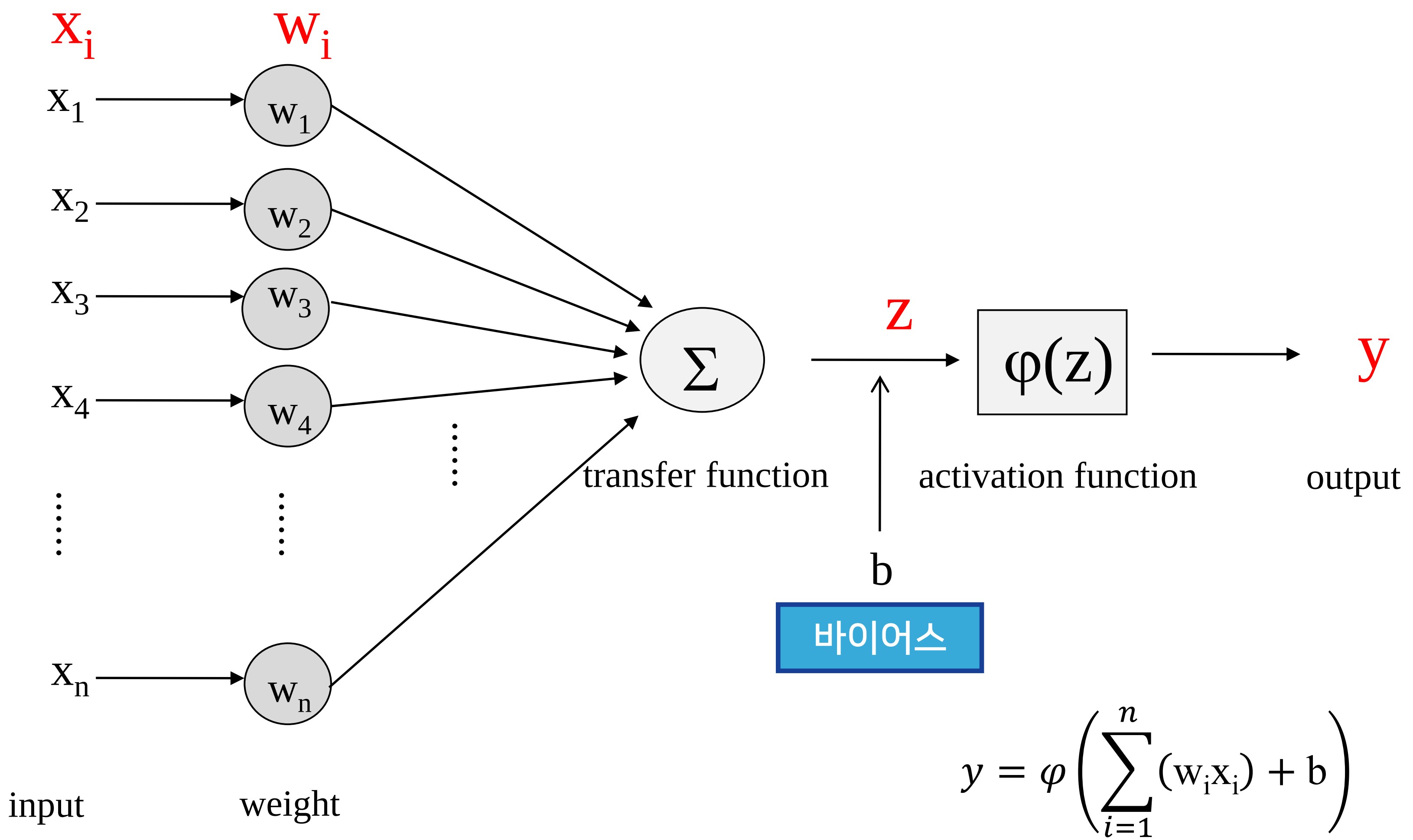
- 입력 신호와 가중치가 세 개인 신경망으로 표현 가능

$$w_1x_1 + w_2x_2 + b \cdot 1 = w_1x_1 + w_2x_2 + w_3 \cdot 1 = 0$$





신경망에 대한 수학적 모델링

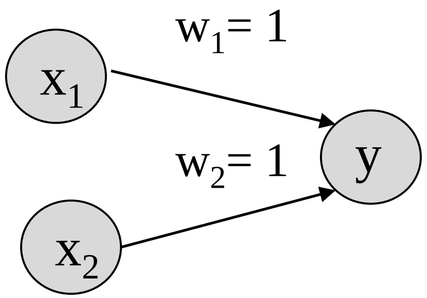


$$y = \varphi\left(\sum_{i=1}^{n}(\mathrm{w_i x_i}) + \mathrm{b}\right)$$

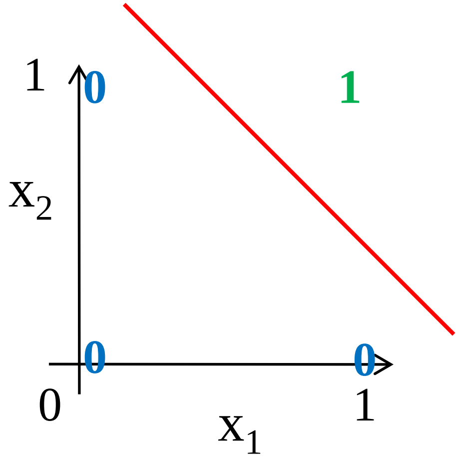
신경망 예측

신경망으로 구현한 논리 소자를 직선의 방정식으로 해석

AND 게이트



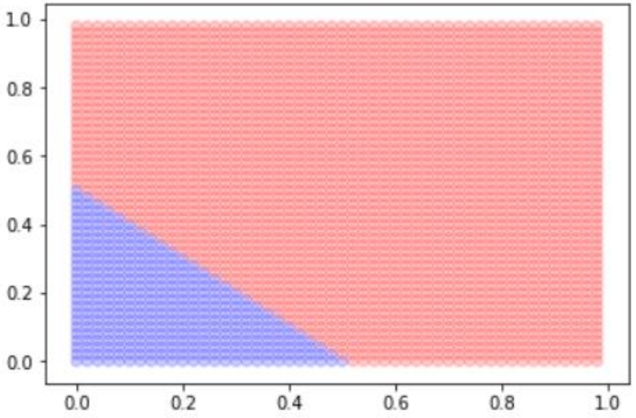
Θ = 1.5
if (z <= 1.5) y = 0
if (z > 1.5) y = 1



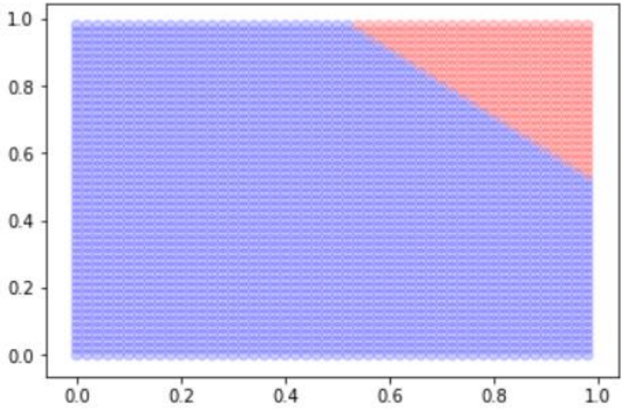
$w_1x_1 + w_2x_2 = 1.5$

$x_1 + x_2 - 1.5 = 0$

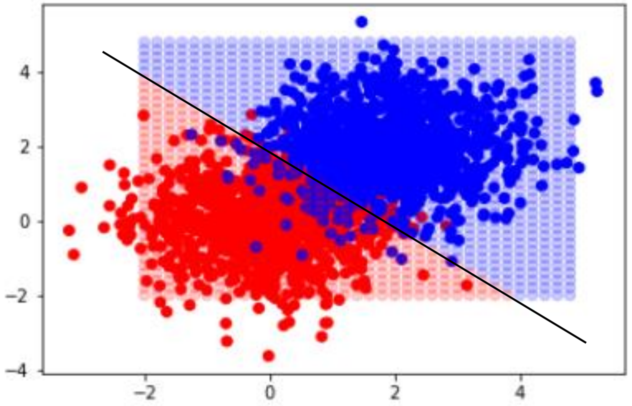
신경망의 결정경계



AND



OR

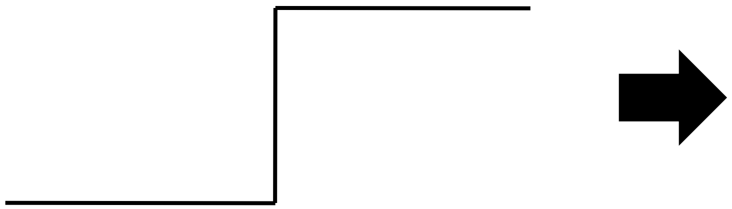


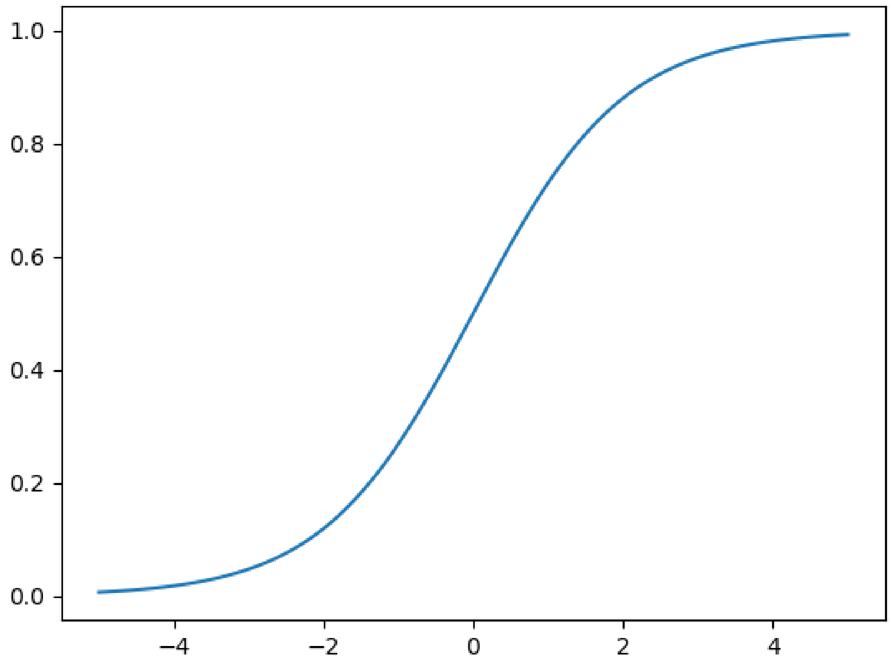
2개의 정규 분포

활성화 함수

- **활성함수는 로지스틱 회귀에 사용한 시그모이드 함수를 사용**

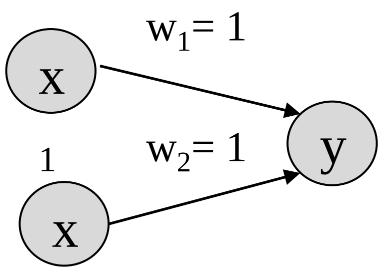
$$y = \frac{1}{1 + e^{-x}}$$





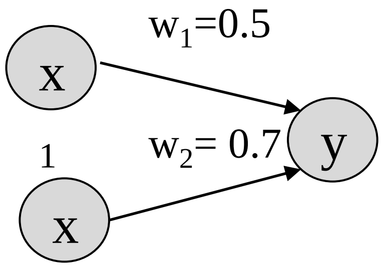
퍼셉트론 학습

- 신경망에서 학습은 가중치와 역치를 학습하는 것



2
AND

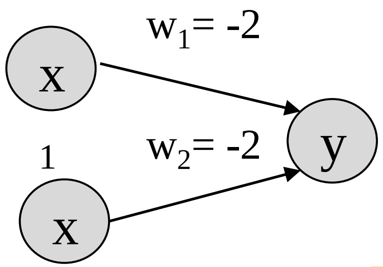
Θ = 1.5
if (z <= 1.5) y = 0
if (z > 1.5) y = 1



2
AND

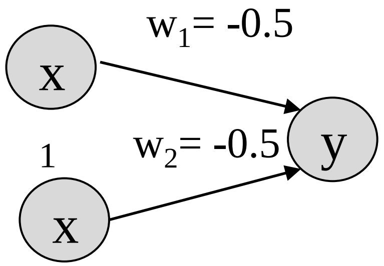
신경망 가중치의 정답이
여러 개 존재할 수 있음

Θ = 1.1
if (z <= 1.1) y = 0
if (z > 1.1) y = 1



2
NAND

Θ = -3
if (z <= -3) y = 0
if (z > -3) y = 1



2
NAND

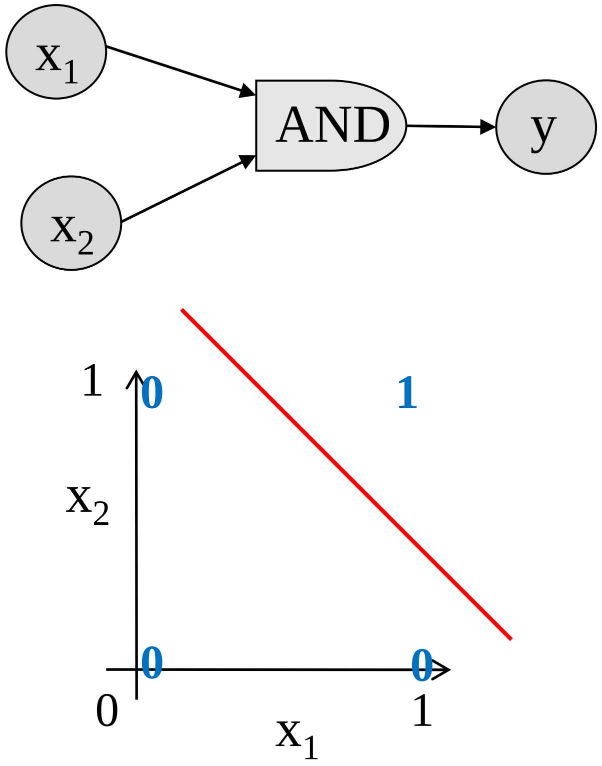
Θ = -0.3
if (z <= -0.3) y = 0
if (z > -0.3) y = 1

퍼셉트론의 한계

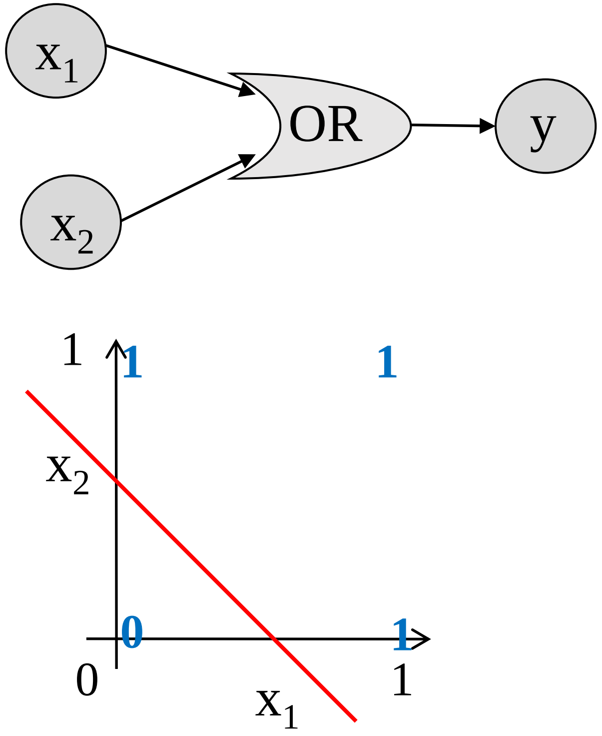
- 선형 분리 가능한 문제만 학습 가능

x_1	x_2	XOR
0	0	0
0	1	1
1	0	1
1	1	0

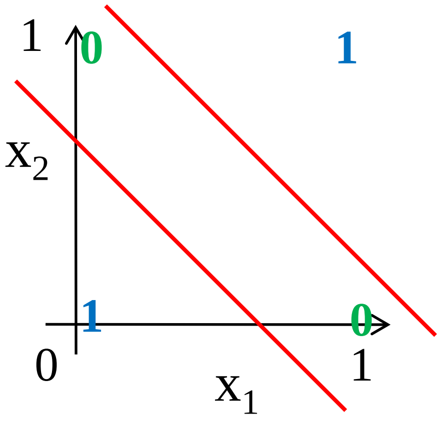
XOR인 경우
퍼셉트론으로 학습 불가능



AND



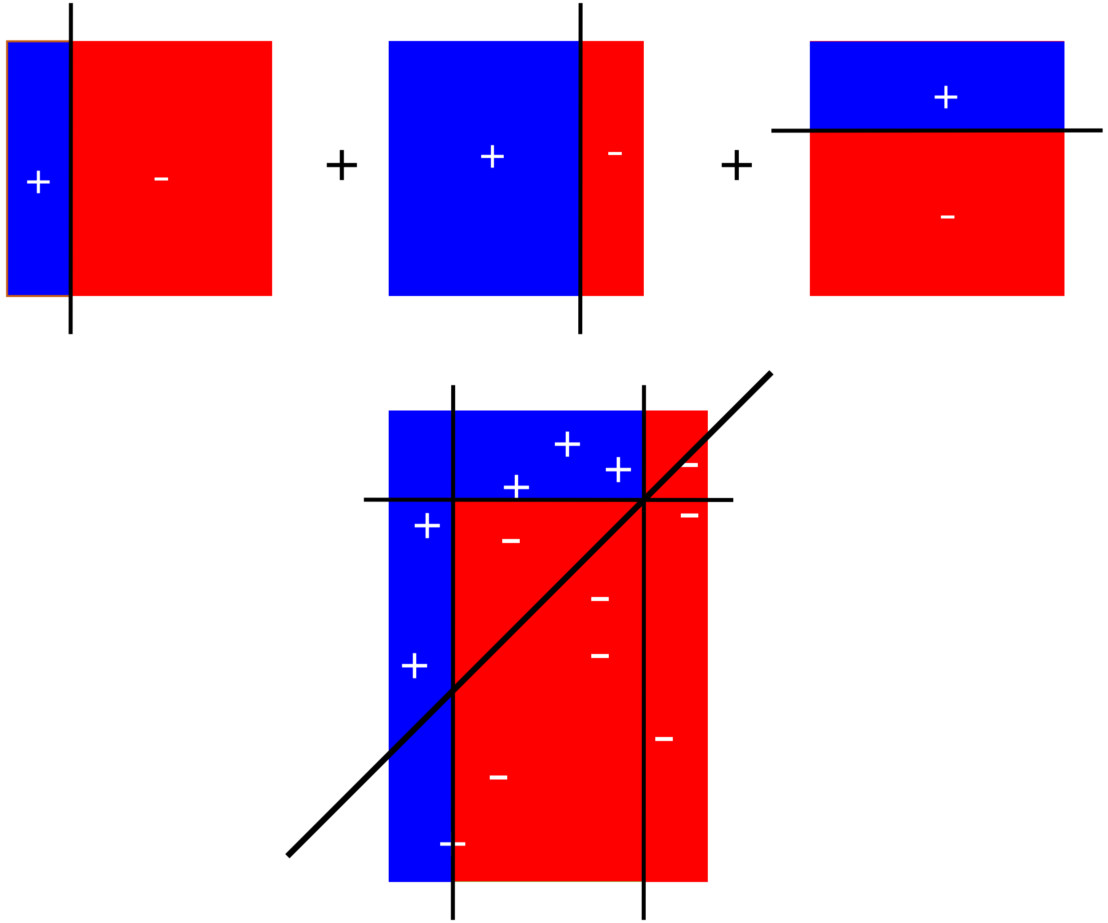
OR



XOR

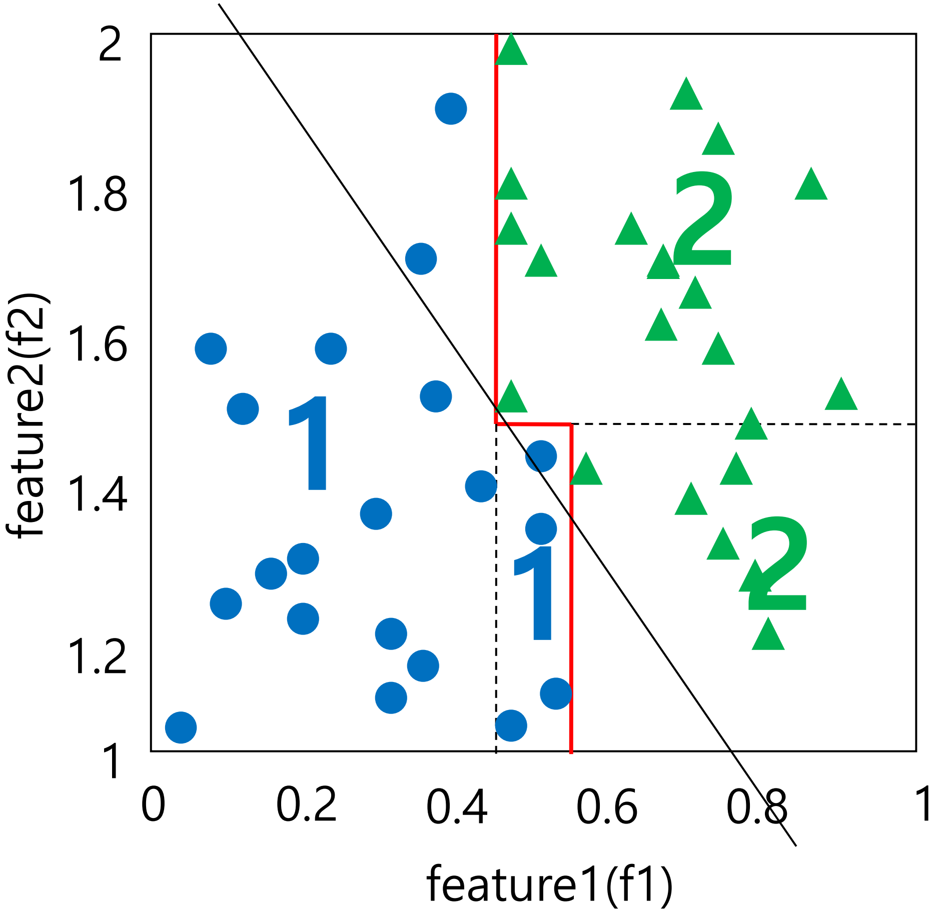
다중 분류기를 이용한 문제 해결

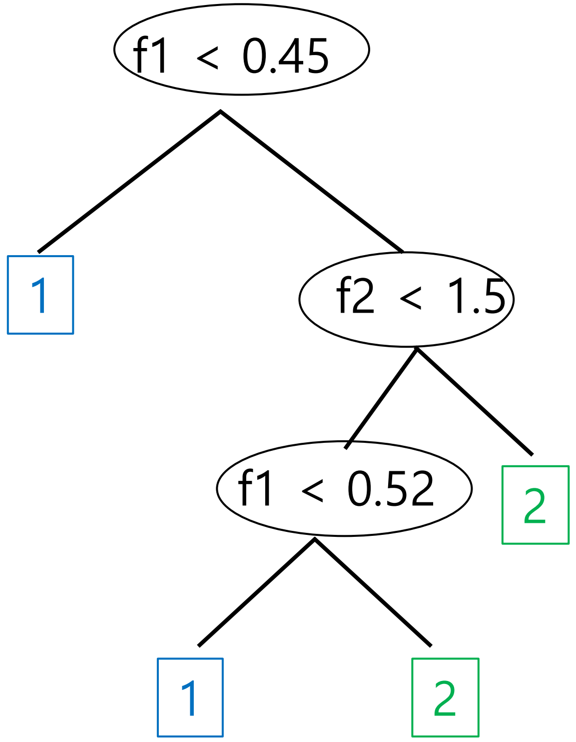
- 3개 분류기 조합으로 계단 형태의 공간 분리 가능



다층 분류기를 이용한 문제 해결

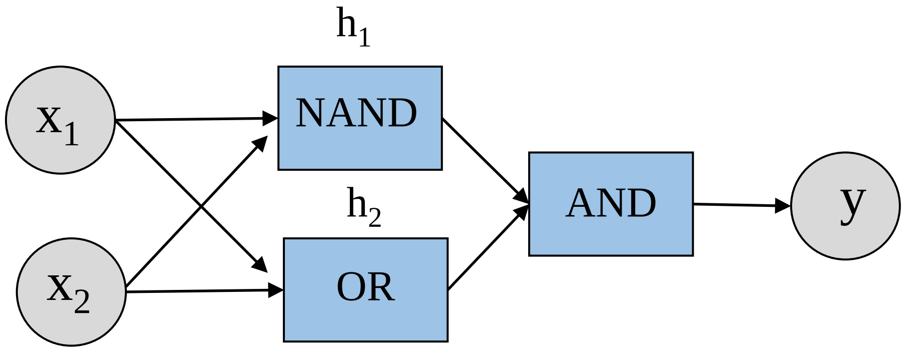
- 비선형 분포를 선형의 계층형 구조로 분류할 수 있음





다층 분류기를 이용한 XOR 문제 해결

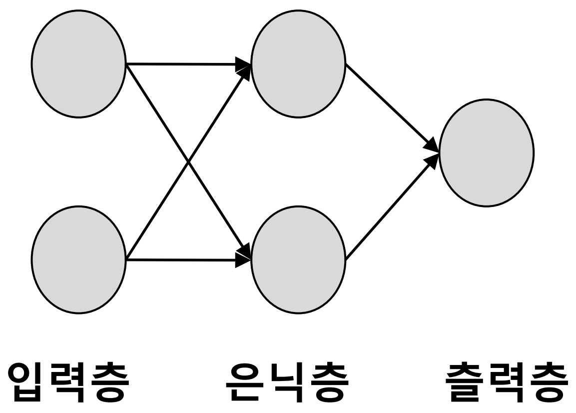
- 선형 분류기를 여러 개 사용하거나 계층적으로 두어 문제 해결

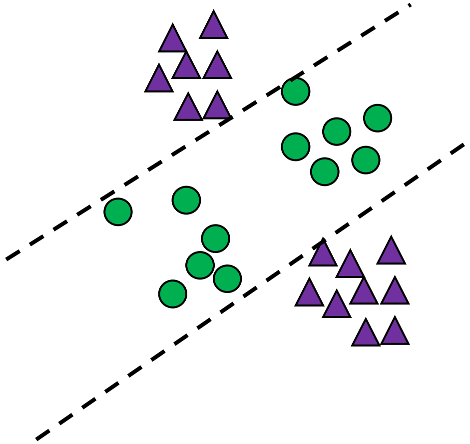


x_1	x_2	h_1	h_2	y	XOR
0	0	1	0	0	0
0	1	1	1	1	1
1	0	1	1	1	1
1	1	0	1	0	0

다층 신경망(Multi-Layer Perceptron)

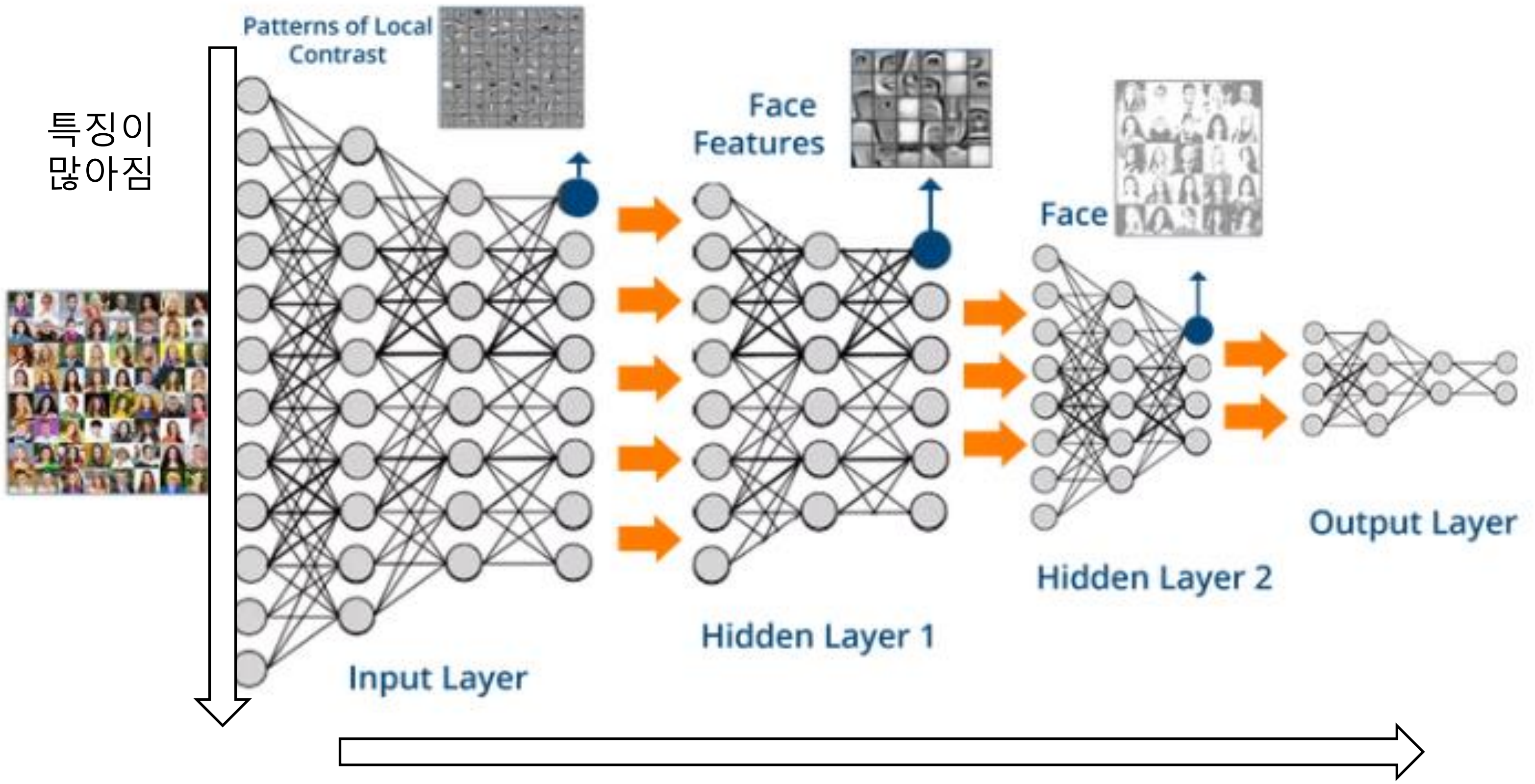
- 다층 퍼셉트론 : 입력층과 출력층 중간에 은닉층을 추가한 것
- 다층 신경망을 사용해 복잡한 비선형 문제 해결





딥러닝(Deep Learning)

- 딥러닝을 사용한 다층 신경망은 은닉층 수가 기존 신경망보다 훨씬 많음

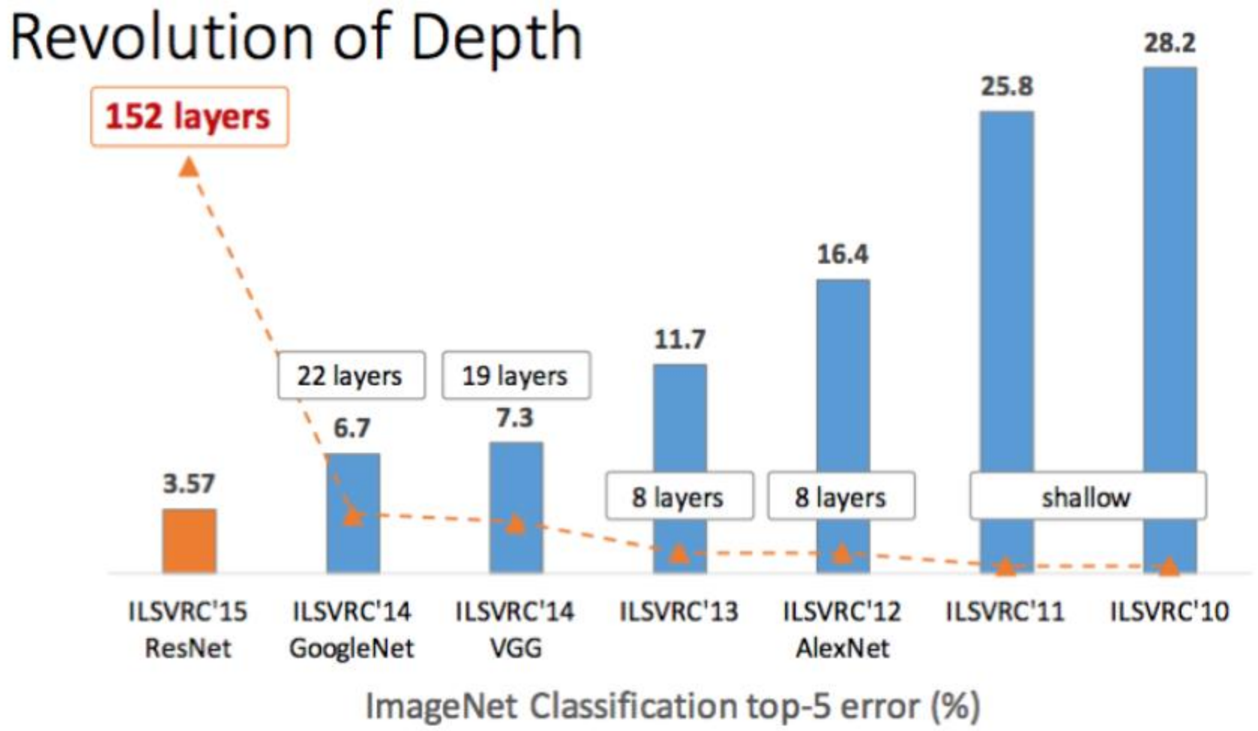


은닉층이 깊어짐(deep)

Image from: https://cdn.edureka.co/blog/wp-content/uploads/2017/05/Deep-Neural-Network-What-is-Deep-Learning-Edureka.png

딥러닝(Deep Learning)

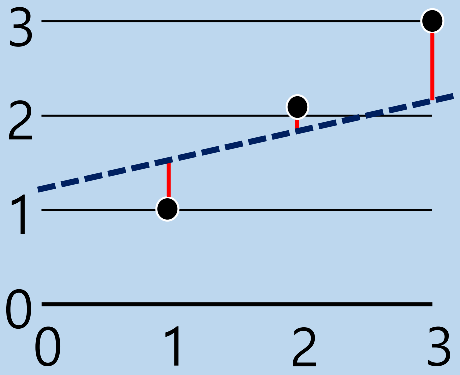
- 알고리즘이 사용하는 은닉층의 개수와 인식 오류를 나타낸 그래프
  > 은닉층이 깊어질수록 학습시키기 매우 어려워짐



https://medium.com/@Lidinwise/the-revolution-of-depth-facf174924f5

학습 모델 비용 정의

$H(x) = wx + b$



오차

오차의 합

$$\frac{\left(H\left(x^{(1)}\right) - y^{(1)}\right)^2 + \left(H\left(x^{(2)}\right) - y^{(2)}\right)^2 + \left(H\left(x^{(3)}\right) - y^{(3)}\right)^2}{3}$$

$$\downarrow$$

$$cost = \frac{1}{m}\sum_{i=1}^{m}\left(H\left(x^{(i)}\right) - y^{(i)}\right)^2$$

$$\downarrow$$

$$cost(w, b) = \frac{1}{m}\sum_{i=1}^{m}\left(H\left(x^{(i)}\right) - y^{(i)}\right)^2$$

최적의 솔루션에 대한 정의

학습모델

추정값 $\longrightarrow$ $H(x) = wx + b$

학습데이터 학습데이터레이블

$$cost(w,b) = \frac{1}{m}\sum_{i=1}^{m}(H(x^{(i)}) - y^{(i)})^2$$

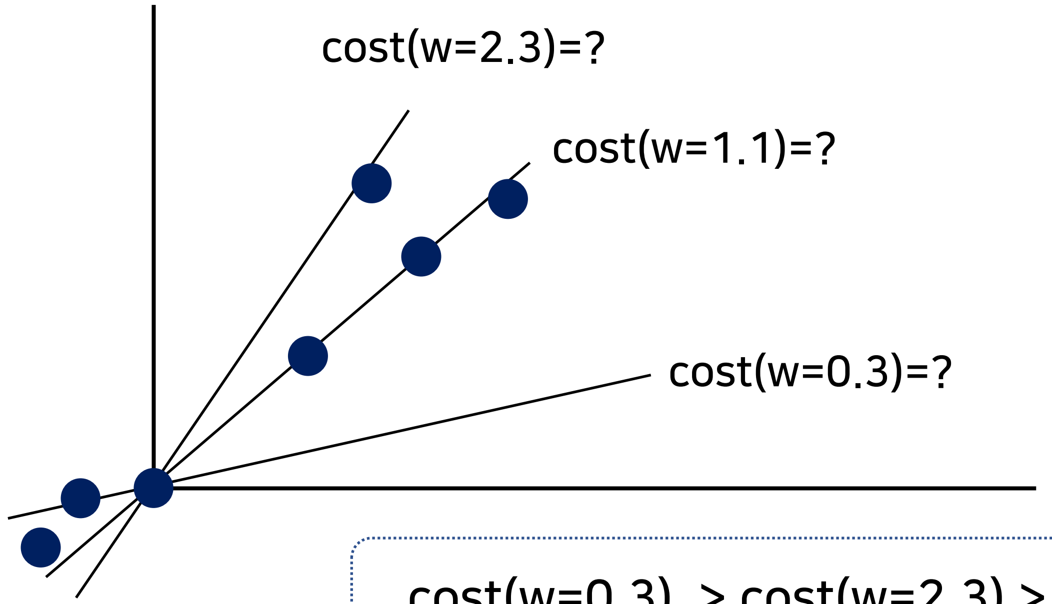
$$\underset{w,b}{minimize}\ cost(w,b)$$

- **오차의 합이 최소가 되는 w, b 추정**

최적의 솔루션 찾기

$$\underset{W,b}{minimize}\ cost(w,b) \xrightarrow{\text{b=0}} H(x) = wx$$

$$cost(w) = \frac{1}{m}\sum_{i=1}^{m}(wx^{(i)} - y^{(i)})^2$$



cost(w=0.3) > cost(w=2.3) > cost(w=1.1)

최적의 솔루션 찾기

$$cost(w) = \frac{1}{m}\sum_{i=1}^{m}\left(wx^{(i)} - y^{(i)}\right)^2$$

학습데이터

x	y
1	1
2	2
3	3

$w = -3, cost(w) = 74.67$
$\frac{1}{3}(((-3) * 1 - 1)^2 + ((-3) * 2 - 2)^2 + ((-3) * 3 - 3)^2)$

$w = -2, cost(w) = 42$
$\frac{1}{3}(((-2) * 1 - 1)^2 + ((-2) * 2 - 2)^2 + ((-2) * 3 - 3)^2$

$w = -1, cost(w) = 18.67$
$\frac{1}{3}(((-1) * 1 - 1)^2 + ((-1) * 2 - 2)^2 + ((-1) * 3 - 3)^2$

$w = 0, cost(w) = 4.67$
$\frac{1}{3}((0 * 1 - 1)^2 + (0 * 2 - 2)^2 + (0 * 3 - 3)^2$

$w = 1, cost(w) = 0$
$\frac{1}{3}((1 * 1 - 1)^2 + (1 * 2 - 2)^2 + (1 * 3 - 3)^2$

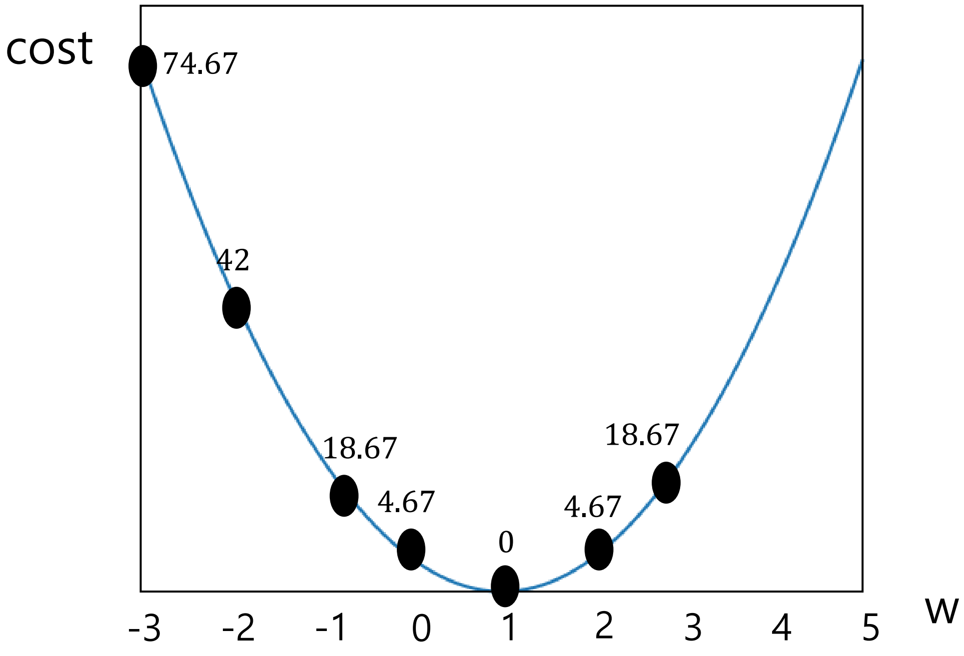
$w = 2, cost(w) = 4.67$
$\frac{1}{3}((2 * 1 - 1)^2 + (2 * 2 - 2)^2 + (2 * 3 - 3)^2$

$w = 3, cost(w) = 18.67$
$\frac{1}{3}((3 * 1 - 1)^2 + (3 * 2 - 2)^2 + (3 * 3 - 3)^2$

최적의 솔루션 찾기

- 위에서부터 보았을 때 기울기의 값이 점점 줄어듦(cost가 최소일 때 기울기 =0)

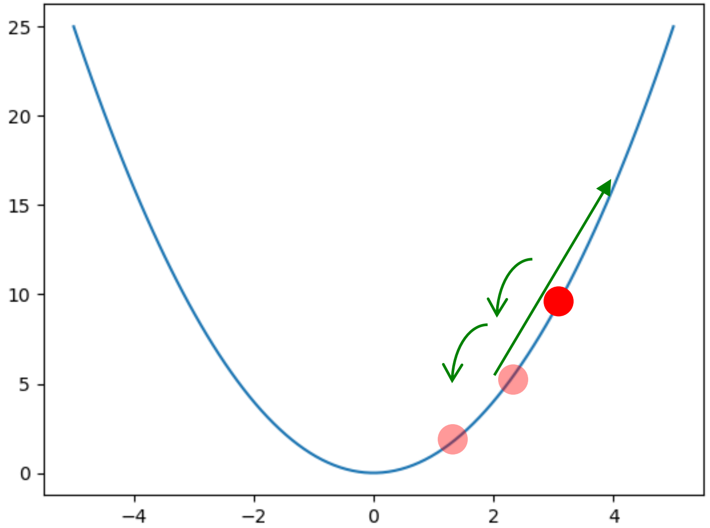
$$cost(w) = \frac{1}{m}\sum_{i=1}^{m}\left(wx^{(i)} - y^{(i)}\right)^2$$



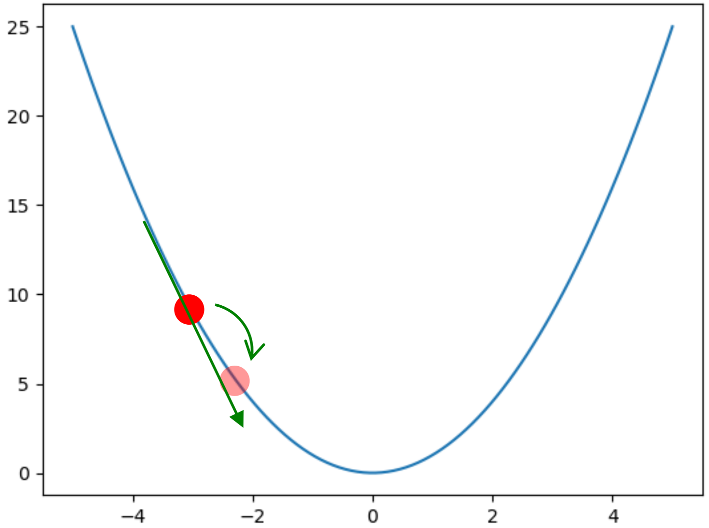
경사하강법

- 현재 위치에서 기울기의 반대 방향으로 일정 비율 이동하는 방법
- 기울기가 거의 0으로 w가 거의 변하지 않는 지점 → cost가 가장 낮은 지점

w = w - 반영비율 x 기울기



양의 기울기



음의 기울기

경사하강법에 대한 수학적 정의

$$cost(w) = \frac{1}{m}\sum_{i=1}^{m}\left(wx^{(i)} - y^{(i)}\right)^2$$

⬇

$$cost(w) = \frac{1}{2m}\sum_{i=1}^{m}\left(wx^{(i)} - y^{(i)}\right)^2$$

w = w – 반영비율 x 기울기

⬇

$$w = w - \alpha \frac{\partial}{\partial w} cost(w)$$

⬇

$$w = w - \alpha \frac{1}{m}\sum_{i=1}^{m}\left(wx^{(i)} - y^{(i)}\right)x^{(i)}$$

α : 학습률, 움직이는 양을 조절하는 상수값
(α가 크면 w가 크게, 작으면 조금씩 업데이트 됨)

경사하강법 직접 계산하기

- y = wx로 가정
- w = 2에서 시작
- α = 0.1

$$w = w - \alpha \frac{1}{m} \sum_{i=1}^{m} (wx^{(i)} - y^{(i)}) x^{(i)}$$

x	y
1	1
2	2.2
3	3.4

w = 2 - 0.1×([(2×1-1) ×1 + (2×2-2.2) ×2 + (2×3-3.4) ×3] /3)
w = 2 - 0.1× 4.13 = 1.58

w = 1.58 - 0.1×([1.58×1-1)×1 + (1.58×2-2.2)×2 + (1.58×3-3.4) ×3] / 3)
w = 1.58 - 0.1 × 2.17 = 1.36

w = 1.36 - 0.1 × 0.62 = 1.18
w = 1.18 - 0.1 × 0.33 = 1.15
…

w = 1.114 - 0.1 × 0.00062 = 1.114

y = 1.114x 최종 학습모델

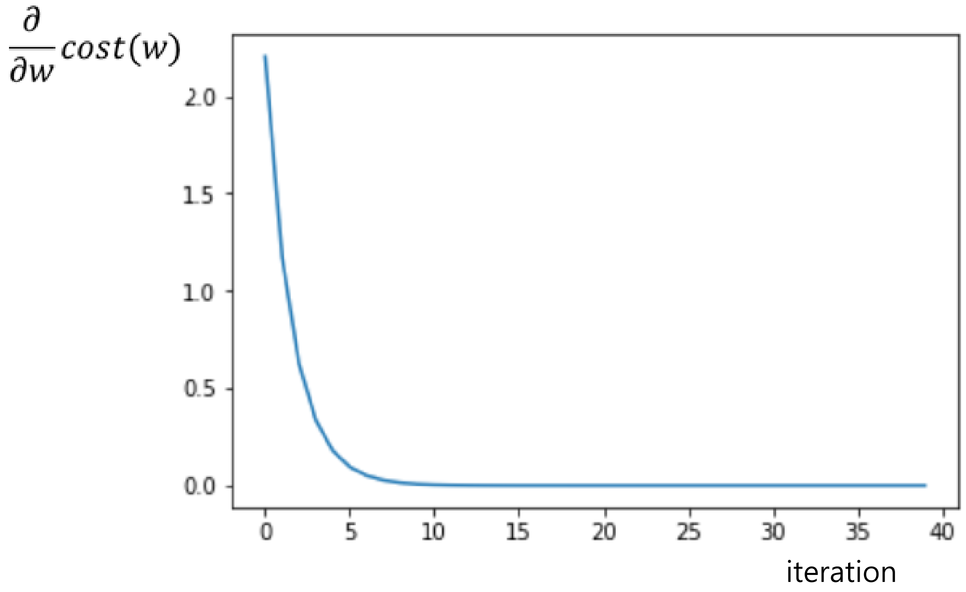
경사하강법 직접 계산하기

- 정답에서 멀 경우 학습이 빠르게 이루어지고,
  정답에 가까울수록 학습이 천천이 이루어짐

w = 2 - 0.1 × 4.13 = 1.58
w= 1.58 - 0.1 × 1.175 = 1.36
w = 1.36 - 0.1 × 0.62 = 1.18
w = 1.18 - 0.1 × 0.33 = 1.15
w = 1.15 - 0.1 × 0.17 = 1.13
w = 1.13 - 0.1 × 0.09 = 1.12..

…

w = 1.114 - 0.1 × 0.00062 = 1.114



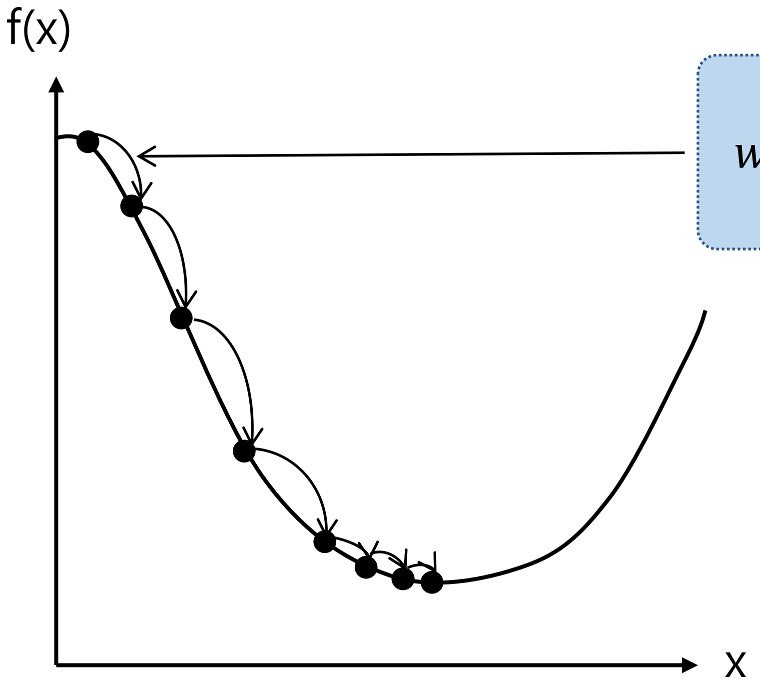
학습률 α

α가 작은 경우

조금씩 이동하면 최저 지점을 찾아 감
이동횟수가 많아져 학습 속도가 느림

α가 큰 경우

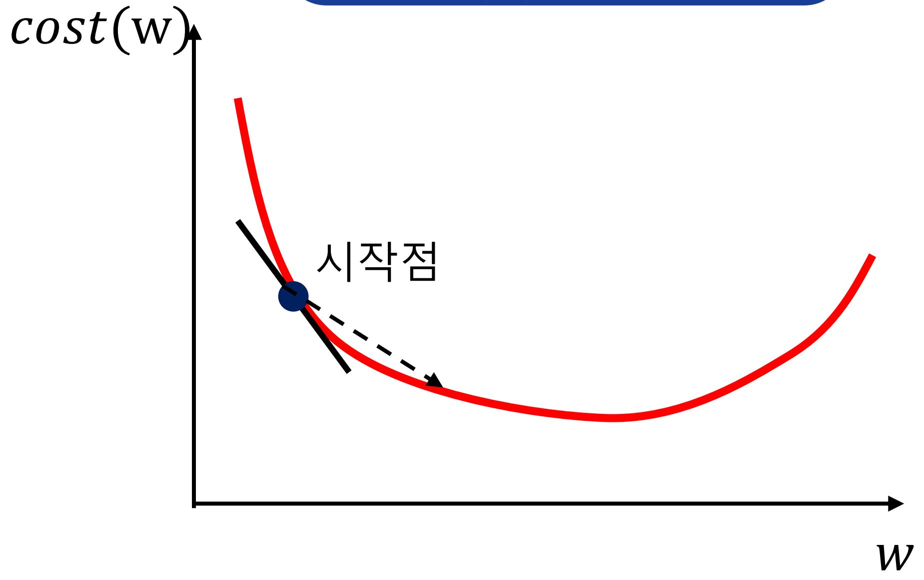
최저 지점을 빠르게 찾아 감
최저 지점을 지나칠 수 있음



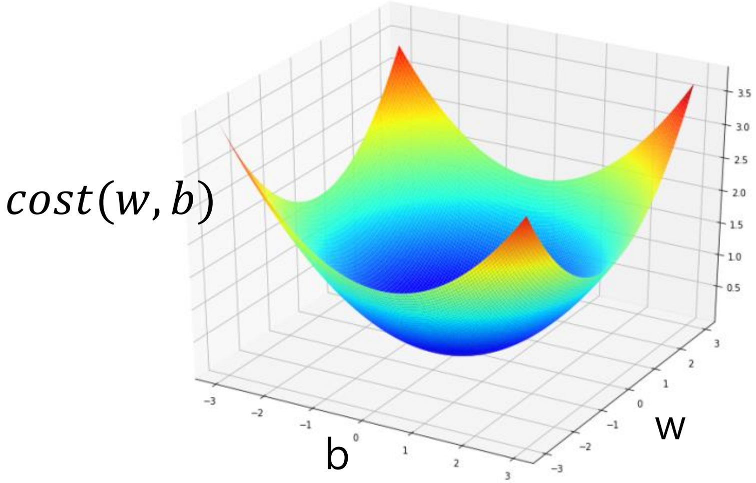
$$w = w - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(wx^{(i)} - y^{(i)}\right) x^{(i)}$$

w, b에 대한 경사하강법

$H(x) = wx$



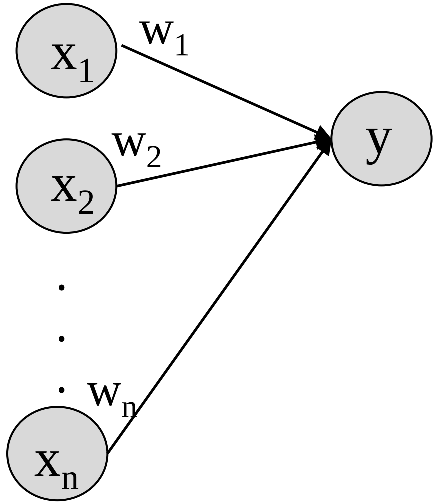
$H(x) = wx + \mathrm{b}$



$$\underset{w}{minimize}\ cost(w)$$

$$\underset{w,b}{minimize}\ cost(w,b)$$

경사하강법을 이용한 퍼셉트론 학습



단층 퍼셉트론

$$y = \varphi\left(\sum_{i=1}^{n}(\mathrm{w_i}\ \mathrm{xi} + b\ )\right)$$

$$cost = \frac{1}{2}\sum_{k=1}^{m}(d_k - y_k)^2$$

$$= \frac{1}{2}\sum_{k=1}^{m}(d_k - f\ (z_k))^2$$

$$w_i = w_i - \alpha\frac{\partial\ cost}{\partial\ w_i}$$

$$w_i = w_i + \alpha\sum_{k=1}^{m}(d_k - y_k)\ x_i$$

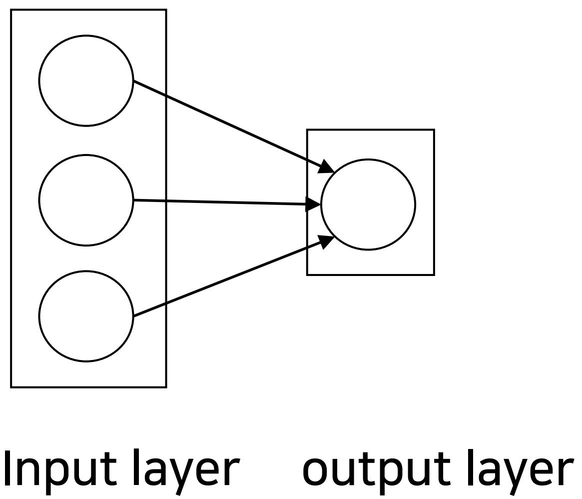
신경망 구조

인간의 신경망 : 1000억 개 수준의 대규모 네트웍 / 구조적으로 복잡함

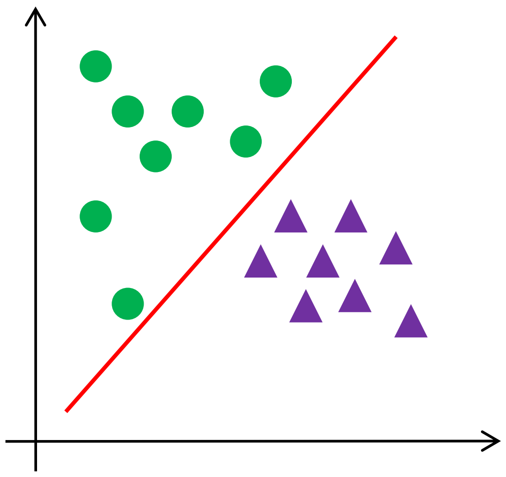
https://radiostudent.si/znanost/znanstveni-britoff/%C5%BEi%C4%8Dke-v-celicah-in-zdravilo-nusinersen

단층신경망과 다층신경망

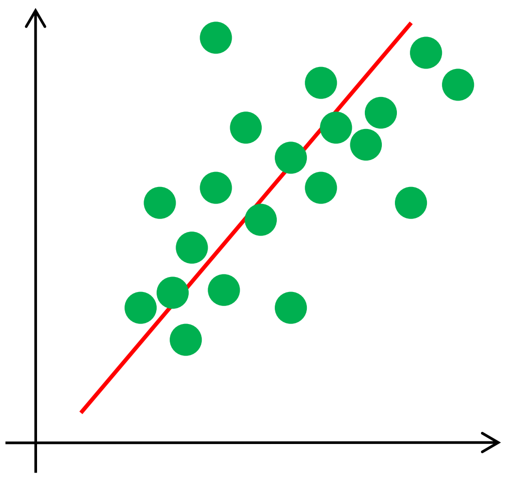
- 단층신경망
  > 복잡한 데이터의 경우 오류 가능성이 큰 한계를 지님



분류

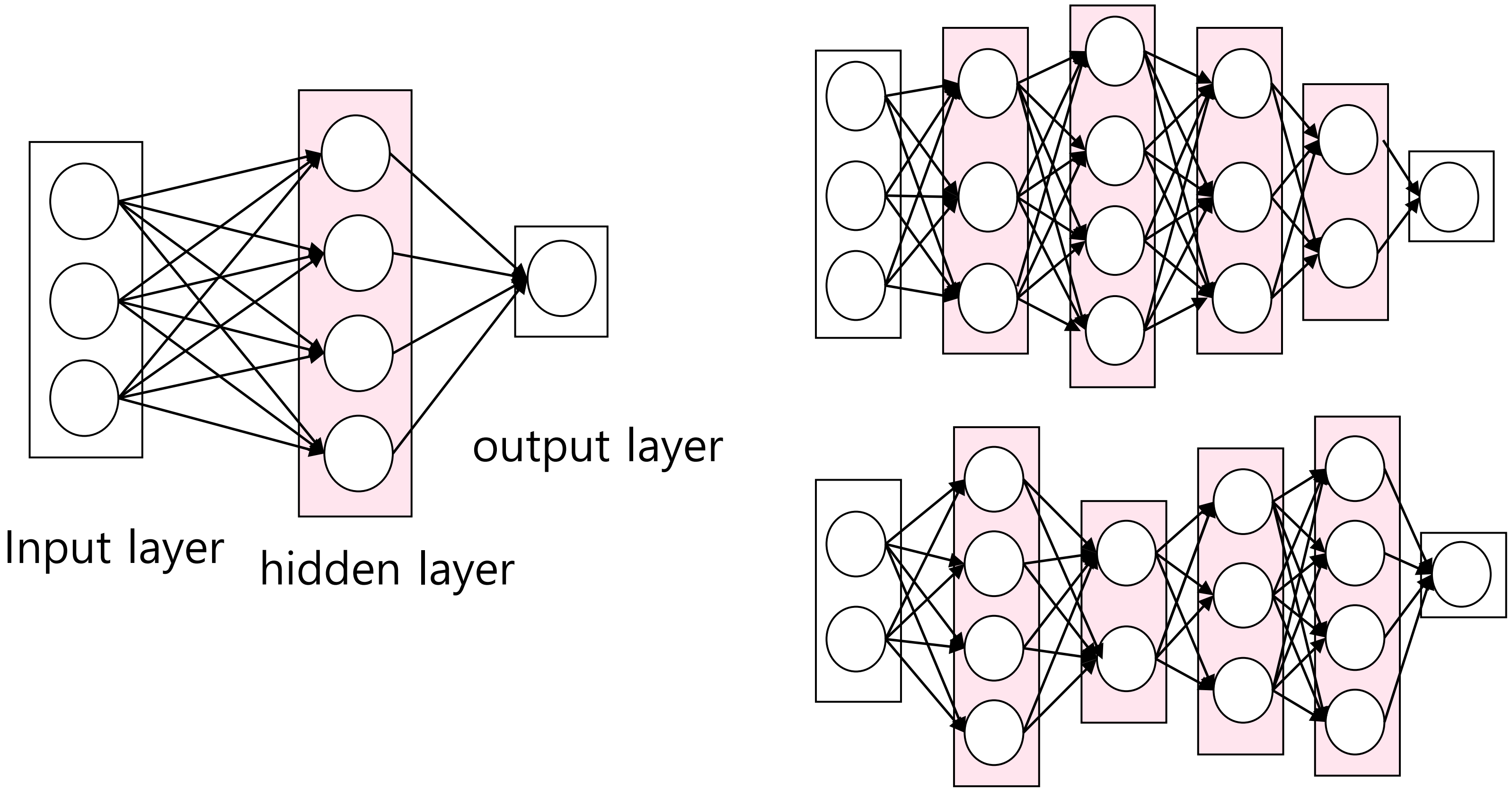


회귀



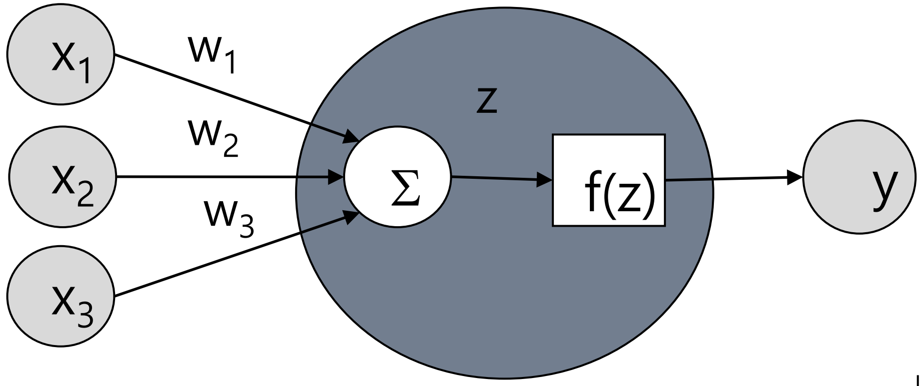
다층신경망과 다층신경망

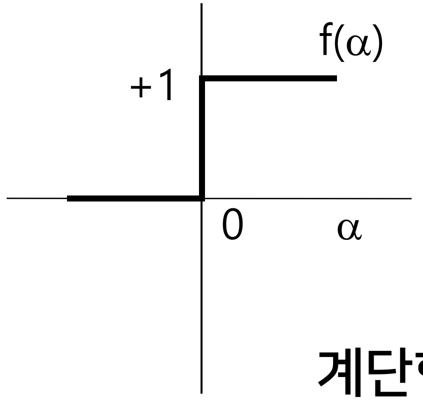
- 다층신경망은 은닉층을 추가하는 구조변경을 통해 복잡한 문제 해결
  > 다층 신경망의 은닉층은 입력층 노드 수보다 클 수도, 작을 수도 있음



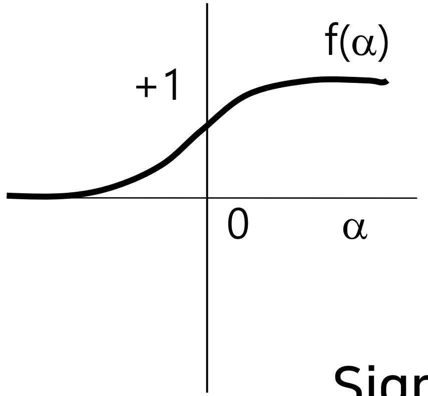
비선형 활성화 함수

- 선형함수만 사용해서 깊은 네트워크 구조를 만들어낼 수 없기 때문에 비선형 함수를 사용
- 신경망에서는 계단함수보다는 시그모이드 함수를 주로 사용





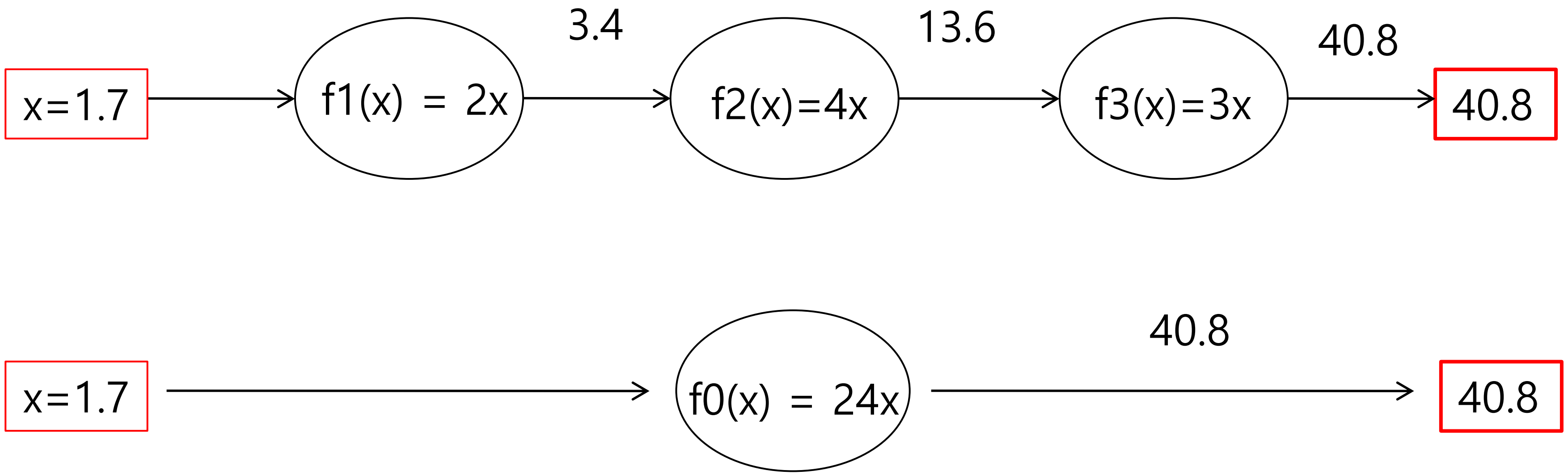
계단함수



Sigmoid 함수

선형 함수 조합의 한계

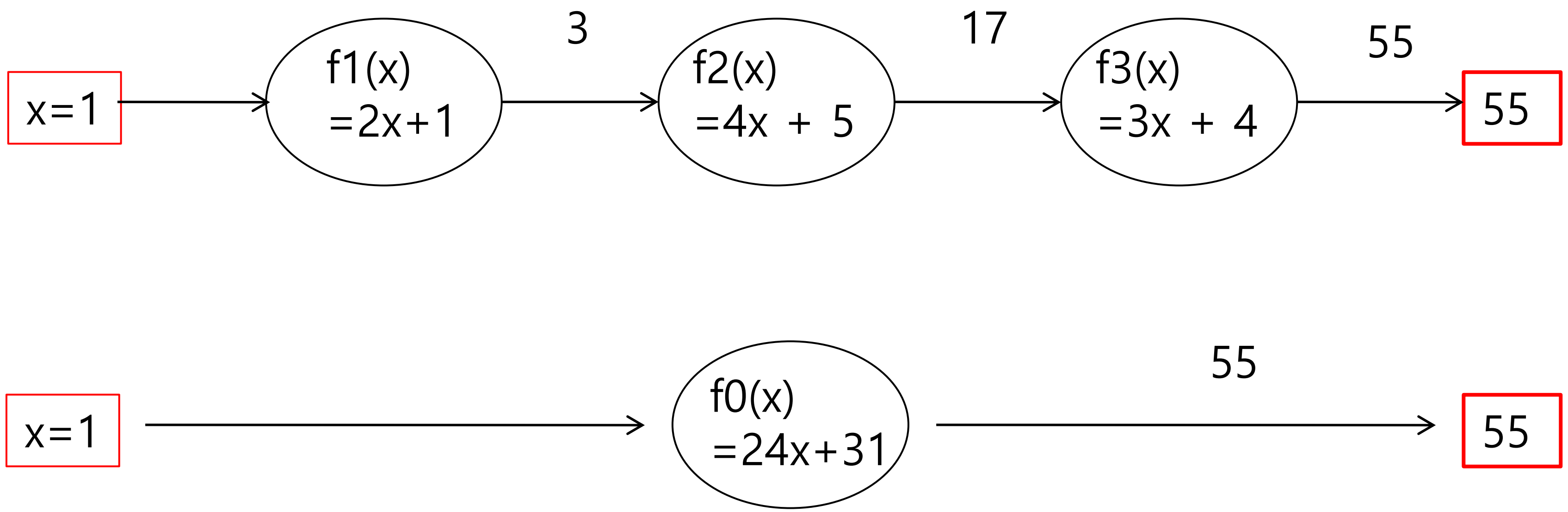
- 세 개의 선형함수를 통과하나, 하나의 선형 함수를 통과하나 같은 값



$$f3(\ f2(\ f1(1.7)\ )\ ) = f0(1.7)$$

선형 함수 조합의 한계

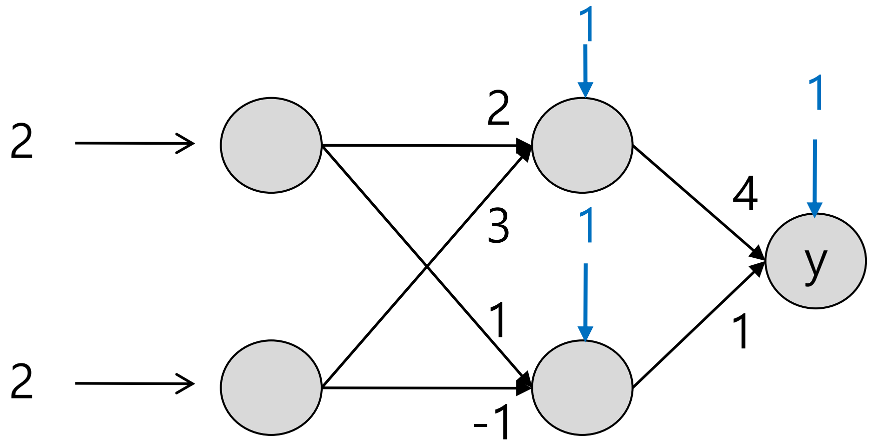
- 세 개의 선형함수를 통과하나, 하나의 선형 함수를 통과하나 같은 값



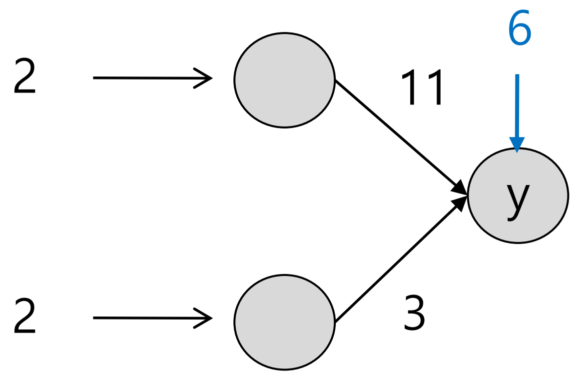
$$f3(\ f2(\ f1(1)\ )\ ) = f0(1)$$

다층신경망에서 선형 활성화 함수의 한계

- 선형 함수로만 이루어진 경우에 다층 신경망은 단층으로 만들 수 있음



=



$$y = \left(\begin{bmatrix} 2 & 3 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right) \begin{bmatrix} 4 \\ 1 \end{bmatrix} + 1$$

$$= \underline{\begin{bmatrix} 2 & 3 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 4 \\ 1 \end{bmatrix}} \begin{bmatrix} 2 \\ 2 \end{bmatrix} + \underline{\begin{bmatrix} 4 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix}} + 1$$

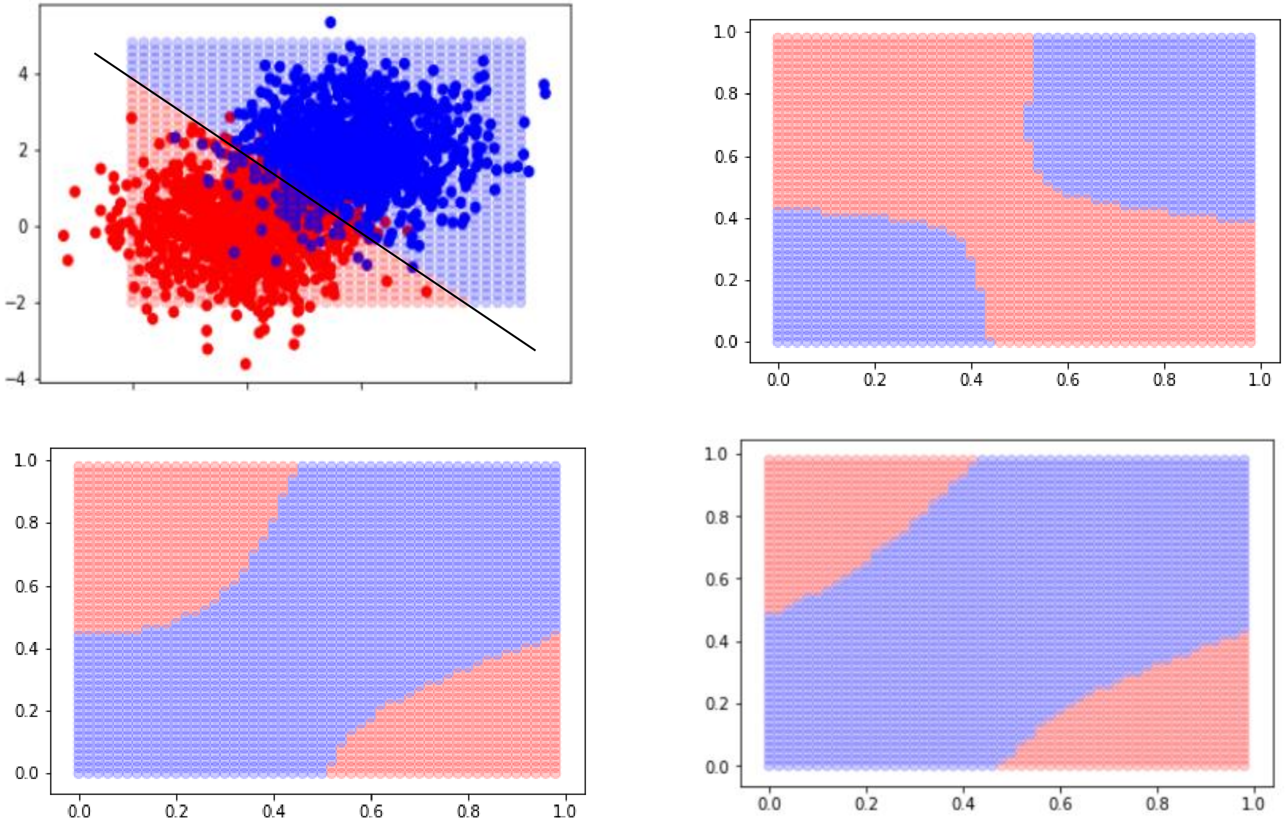
$$= \begin{bmatrix} 11 \\ 3 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix} + 5 + 1$$

벡터의내적 $(4 \times 1) + (1 \times 1)$

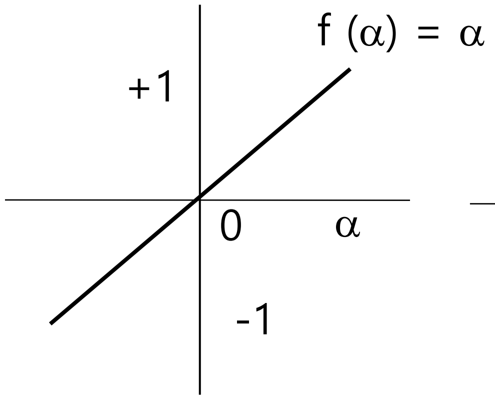
$$= \begin{bmatrix} 11 \\ 3 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \end{bmatrix} + 6$$

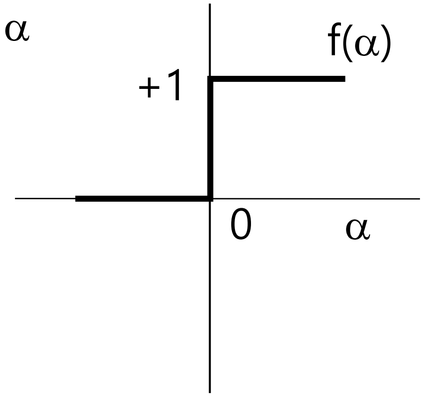
비선형 분포 분류

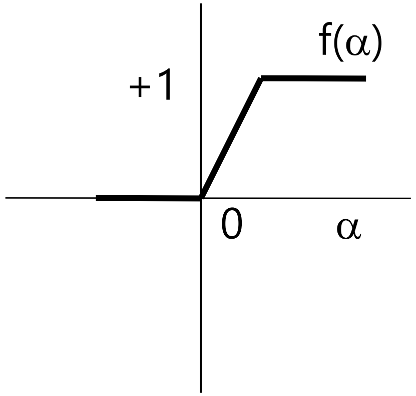
- 비선형 함수, 다수의 은닉층을 사용하면 비선형 결정 경계면을 갖게되어 복잡한 분포를 좀 더 정교하게 분류할 수 있음

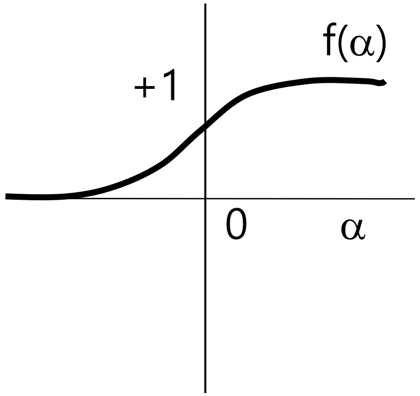


활성화 함수의 특성









항등 함수

- 출력 범위 제한이 없음
- 모든 구간에서 미분 값 존재

계단 함수

- 0 이나 1 출력 값이 단순
- 출력 범위 제한
- 불연속적이다.
- 극점에서 급격히 변한다.
- 미분 값이 0이 아닌 구간
- 거의 한 점으로 좁다
- 미분 값이 너무 단순

임계논리 함수

- 입력 값이 커질 수록 출력 값은 1에 가까워지고 출력 값이 0과 1 사이의 값
- 다양한 미분 값
- 극점에서 급격히 변한

시그모이드 함수

- 부드럽게 연속적이다.
- 극점에서 천천히 함

=> 학습에 적합하다.

세 함수의 공통점 : +무한대에서 1로 수렴 – 무한대에서 0으로 수렴

활성화 함수에 따른 경사하강법

- 새로운 가중치는 이전 가중치의 cost함수를 w에 대해서 미분한 값을 반대방향으로 일정 부분 반영

1) 계단함수를 사용할 때 가중치 업데이트

$$y = \left(\sum_{i=1}^{n} (w_i x_i + b) \right) > 0$$

$$w_i = w_i - \alpha \sum_{k=1}^{m} (d_k - y_k) x_i$$

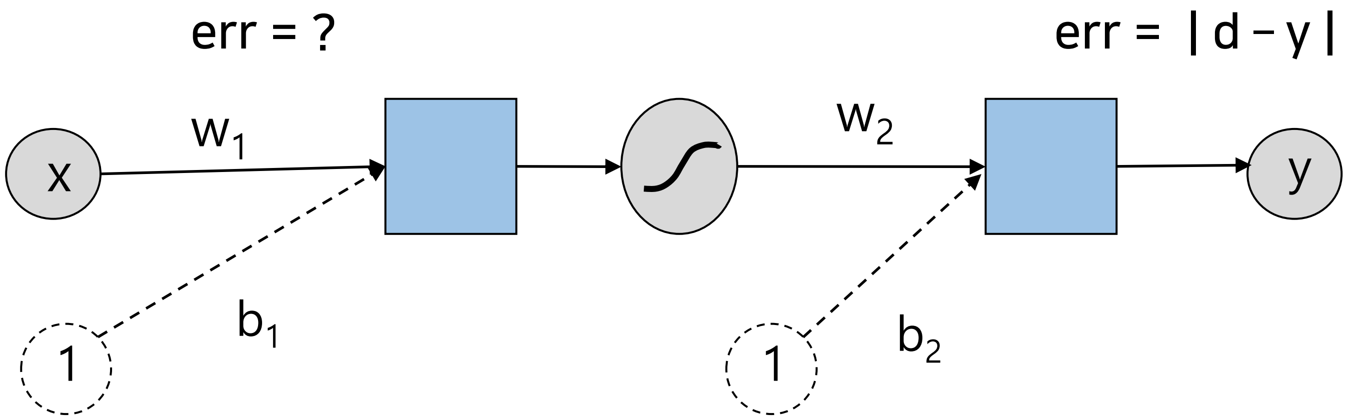
2) 시그모이드 함수를 사용할 때 가중치 업데이트

$$y = \text{sigmoid}\left(\sum_{i=1}^{n} (w_i x_i + b) \right)$$

$$w_i = w_i - \alpha \sum_{k=1}^{m} (d_k - y_k)\, y_k (1 - y_k) x_i$$

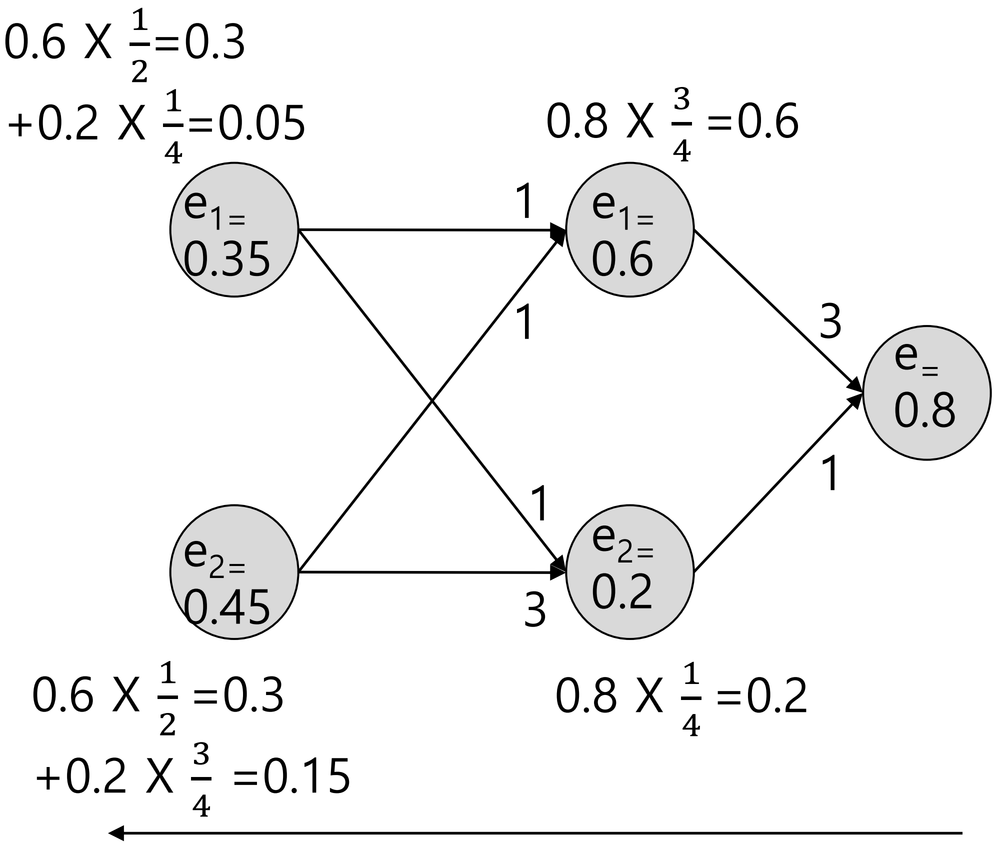
다층 신경망 학습

- 오차를 계산할 수 없기 때문에 w1, b1은 학습시킬 수 없음



다층 신경망 학습

- 역전파 오류법 : 출력층에서 입력층까지 거꾸로 오류를 계산해가며 모든 계층을 학습



역전파 오류법 (Back Propagation)