

AI and Deep Learning

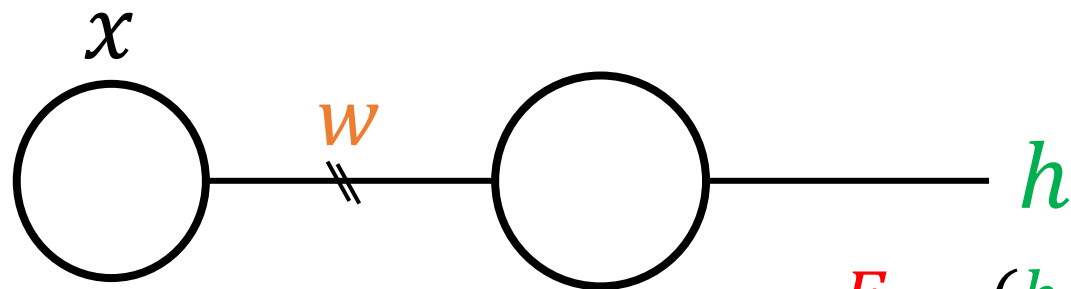
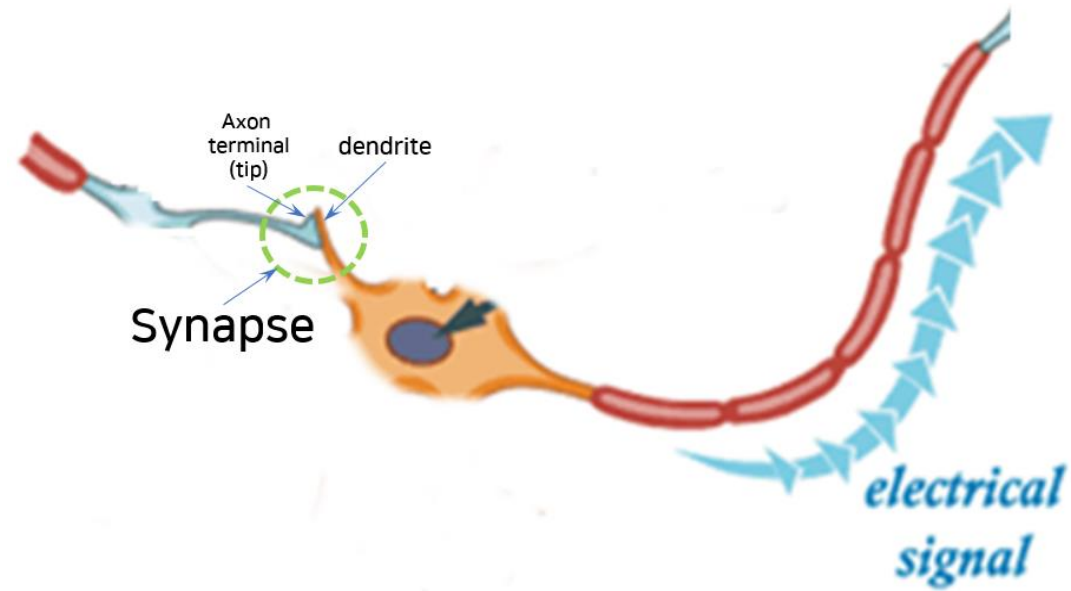
로지스틱 회귀와 분류(1)

- 결정 경계 -

제주대학교

변영철

<http://github.com/yungbyun/ml>



$$E = (h - y)^2$$

로지스틱 회귀

Logistic Regression

회귀 모양이 로지스틱

→ 무슨 의미인가?

"1시간 공부하면 1시간 게임하게 해줄게"

a.csv 파일

study, game

-2, 0

-1, 0

1, 1

2, 1



desmos

$$\begin{pmatrix} -2 \\ -1 \\ 1 \\ 2 \end{pmatrix} (w) \longrightarrow h$$

다음 점을 표시 $(-2, 0), (-1, 0), (1, 1), (2, 1)$

$$h = wx$$

$$h = \frac{1}{1 + e^{-wx}}$$

시냅스 연결강도 w 의 의미는?

종별	접수일자	응시일자	과목	상태
1종보통	2017/03/	2017/03/	도로주행	합격
1종보통	2017/03/	2017/03/	기능	합격
1종보통	2017/03/	2017/03/	기능	불합격
1종보통	2017/02/	2017/03/	기능	불합격
1종보통	2017/02/	2017/02/	기능	불합격
1종보통	2017/02/	2017/02/	학과	합격
1종보통	2017/02/	2017/02/	학과	불합격

80점 이상 합격!

합격 (1) / 불합격 (0)

양성(1) / 음성(0)

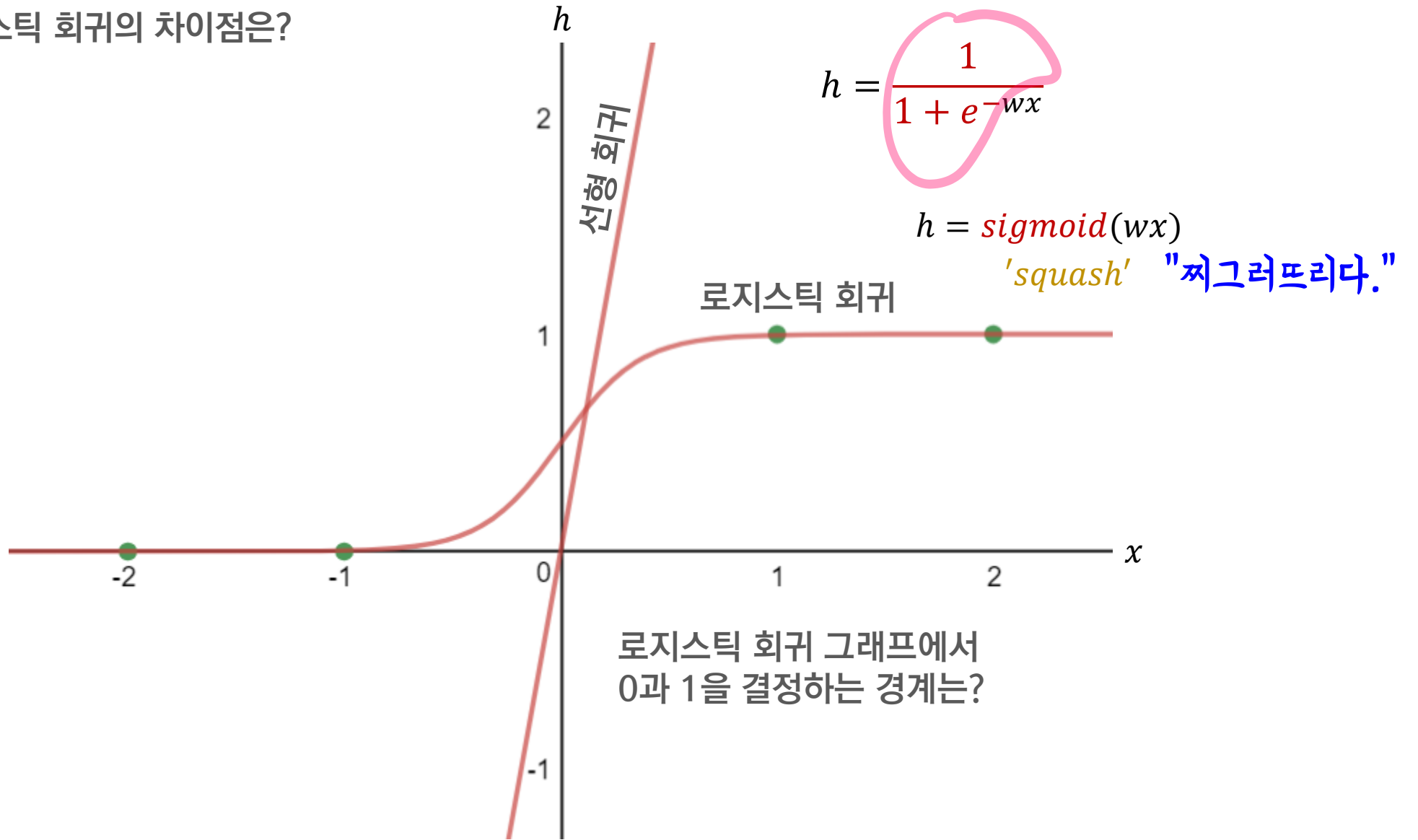
위험 (1) / 안전 (0)

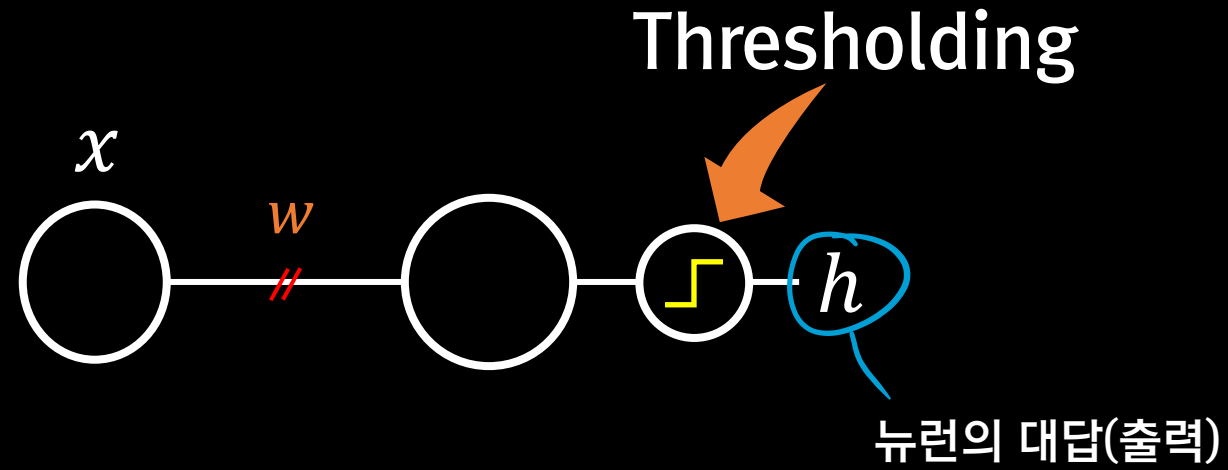
당선 (1) / 탈락 (0)

"둘 중 하나로 분류하는 문제"

뉴런(신경세포)의 **대답**(회귀)은 달라져야 한다.
잘 분류하여 정답을 맞출 수 있도록...

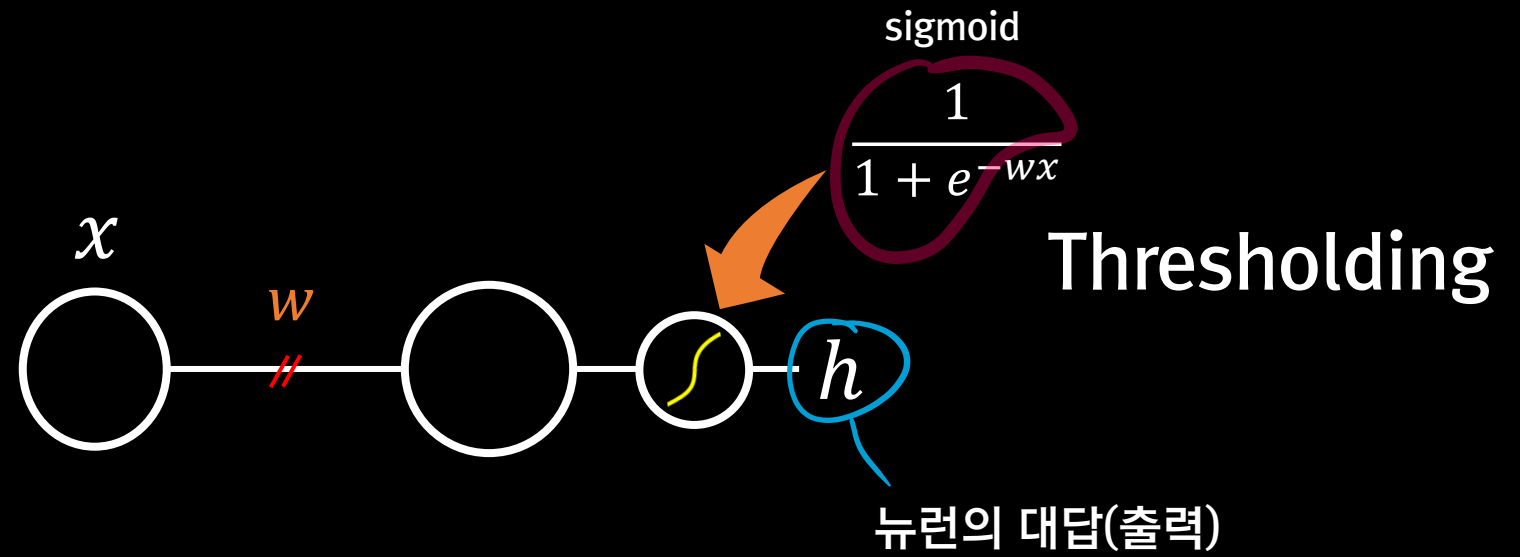
선형회귀와 로지스틱 회귀의 차이점은?





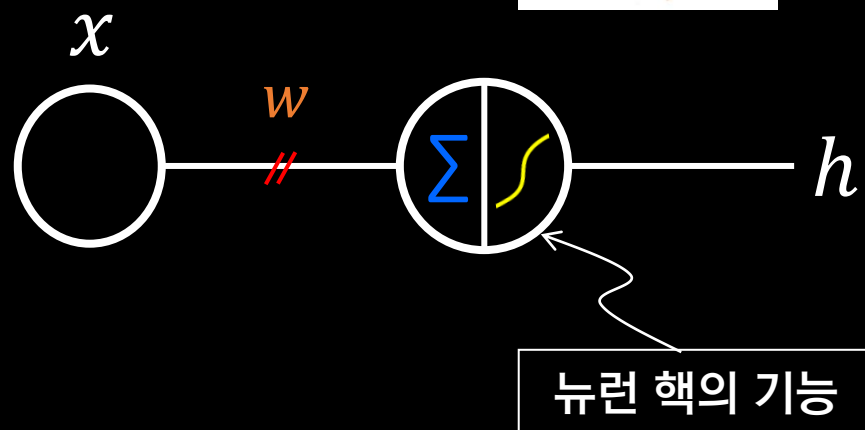
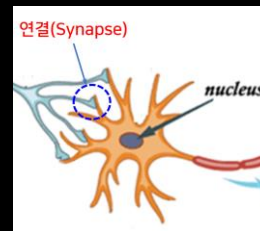
뉴런(신경세포)의 대답

Weighted sum이 특정 값 T 보다 크면 전기신호 발생(ON)
그렇지 않으면 OFF



뉴런(신경세포)의 대답

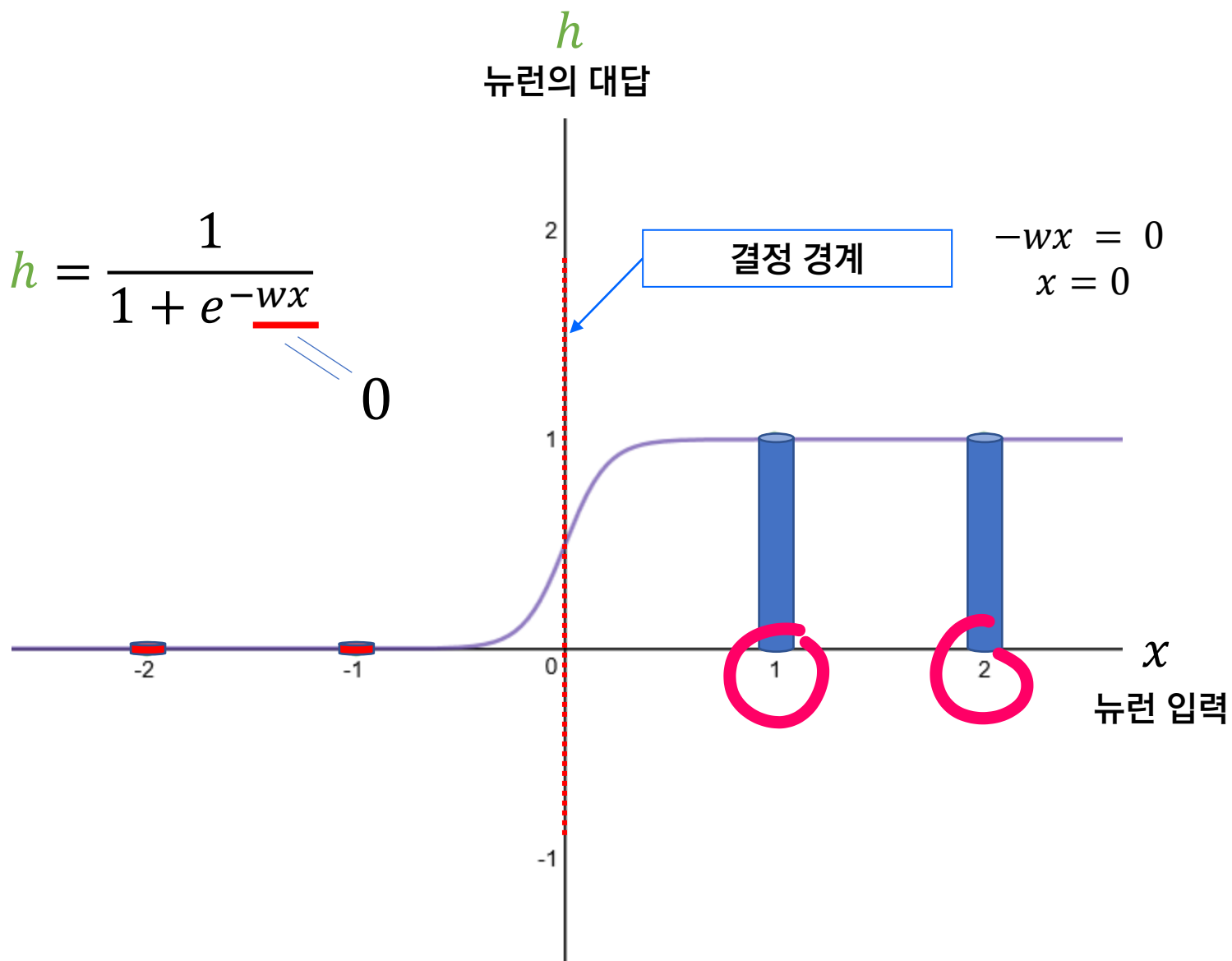
Weighted sum이 특정 값 T 보다 크면 전기신호 발생(ON)
그렇지 않으면 OFF



- 입력 값 x 에 따라 활성화(1) 혹은 비활성화(0)
- 결정 경계는?

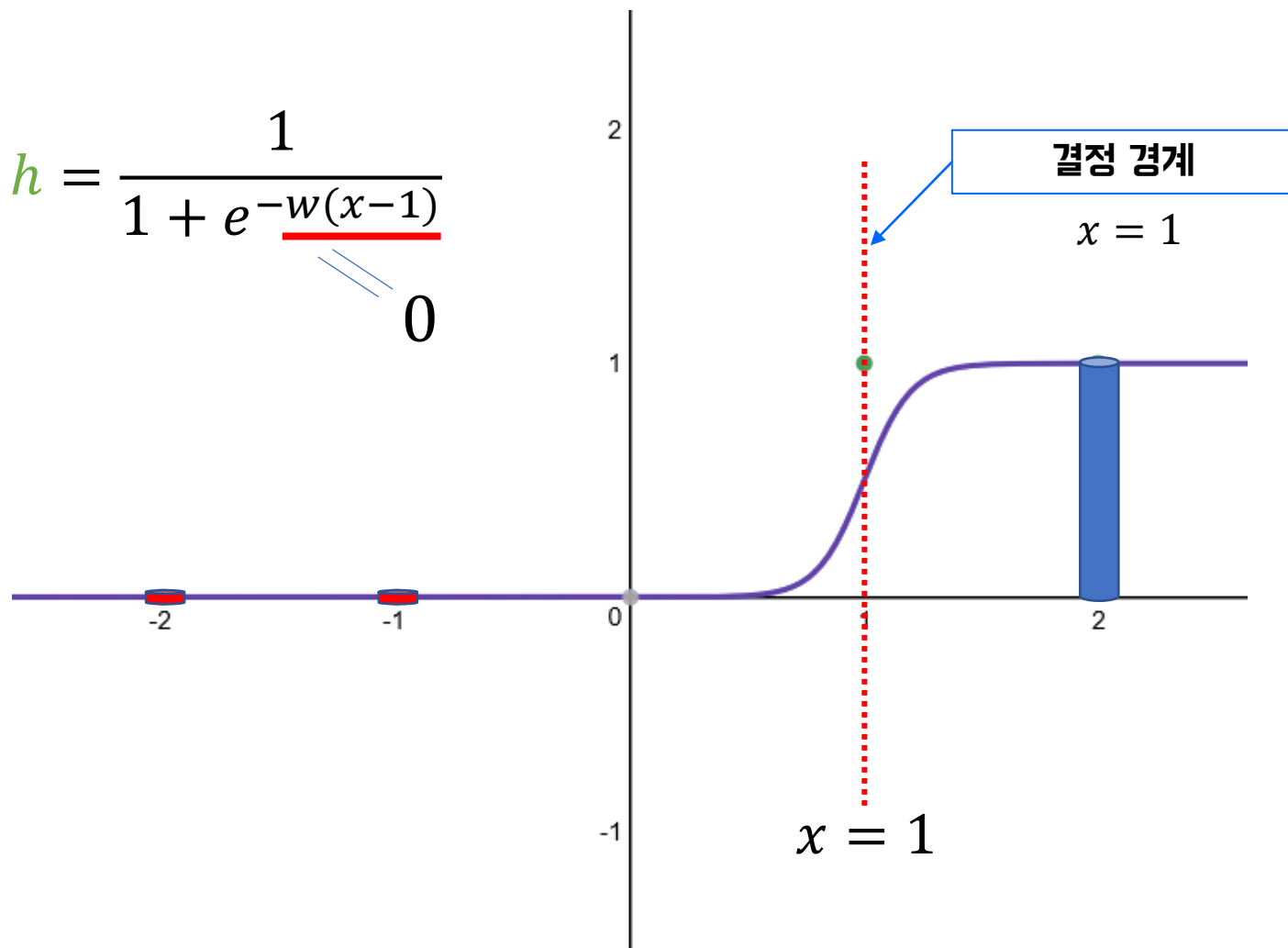
$$h = \frac{1}{1 + e^{-wx}}$$

0

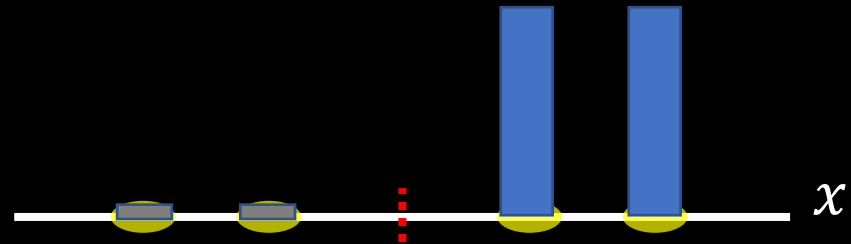


$$h = \frac{1}{1 + e^{-w(x-1)}}$$

$\underbrace{\hspace{1.5cm}}_{=0}$



위에서 본 결정 경계



{옆에서 본 모습}

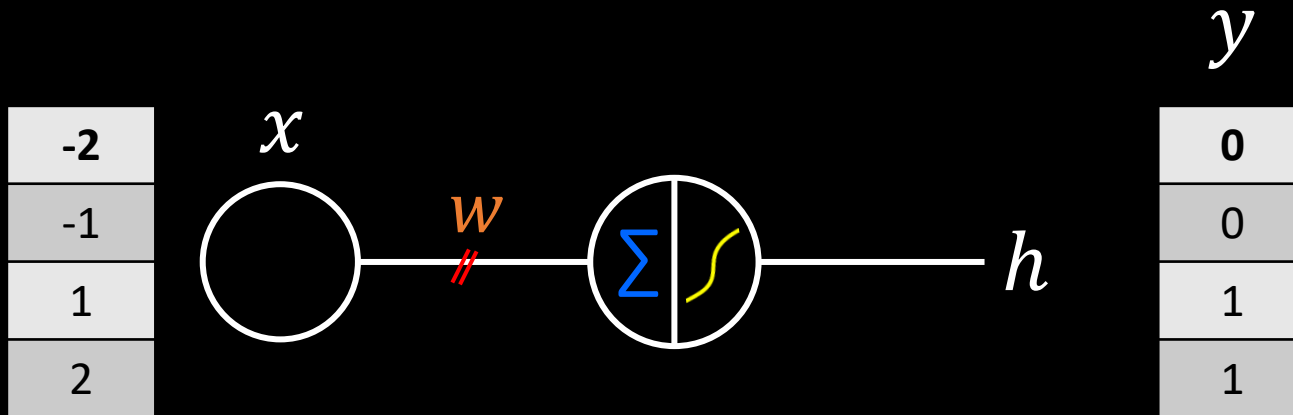


{위에서 본 모습}

분류(Classification)

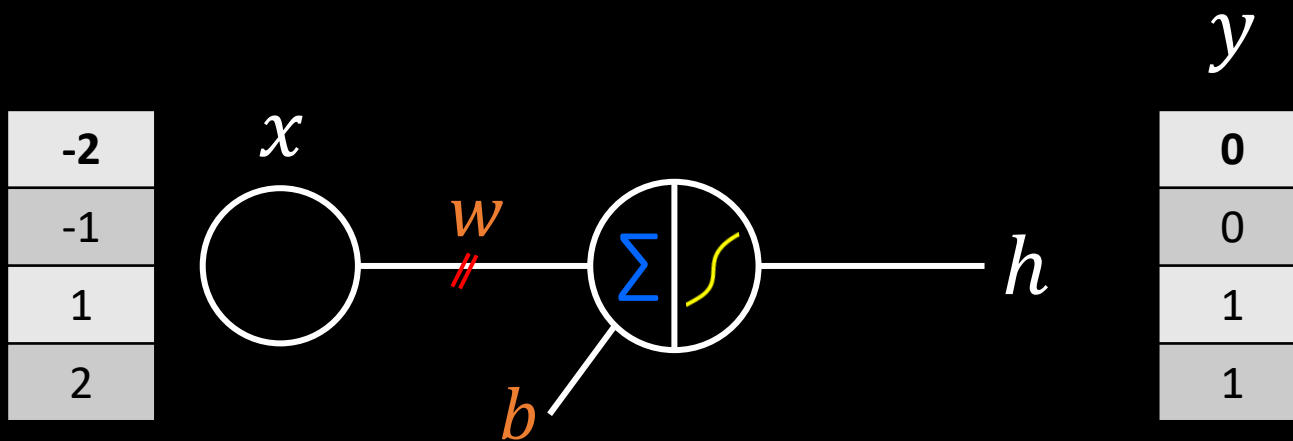
- 합격(1)이나 불합격(0)이나
 - 스팸(1)이나 아니냐(0)
 - 사기(scam)냐 아니냐 (0)
 - 안전(1)한가 위험(0)한가
 - 침입(1)이나 아니냐 (0)
 - 암(1)이나 아니냐(0)
- 이진 분류, 다중 분류로 확장가능

뉴런 그림만 보고 결정 경계를 아는 방법



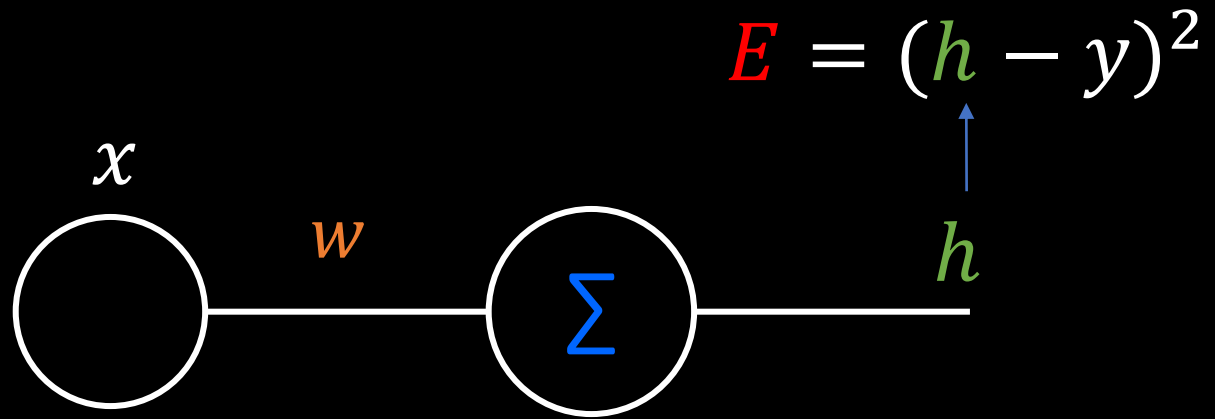
$$h = \begin{cases} 1 & \text{if } wx \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

뉴런 그림만 보고 결정 경계를 아는 방법

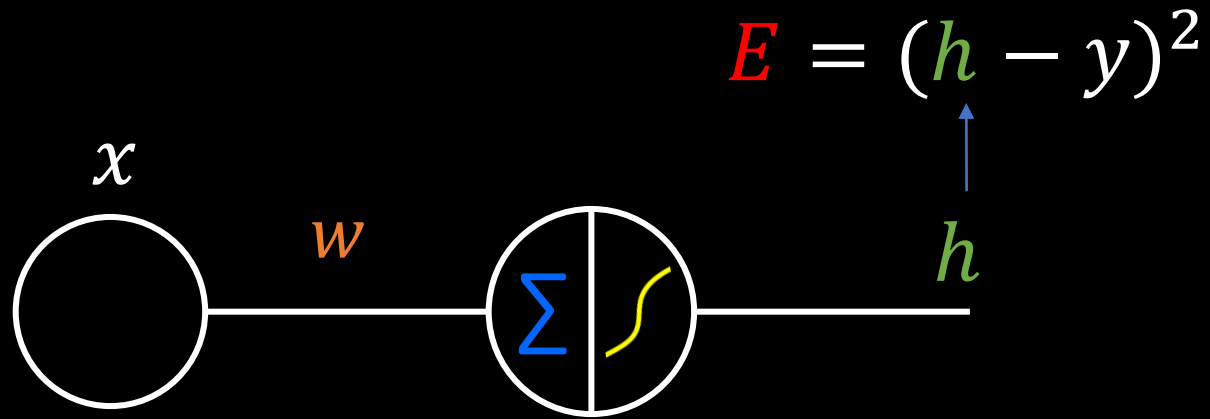


$$h = \begin{cases} 1 & \text{if } wx + b \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

선형 회귀 오류 함수



선형 회귀 오류 함수



MSE가 동작할까?



desmos

다음 점을 표시 $(-2, 0), (-1, 0), (1, 1), (2, 1)$.

$$h = wx$$

$$h = \frac{1}{1 + e^{-wx}}$$

$(1, 1)$ 점만 있을 경우

$$E = \left(\frac{1}{1 + e^{-w \cdot 1}} - 1 \right)^2$$

$$(w, E)$$



세 점을 더 추가: $(-1, 0)$, $(1, 1)$, $(-3, 0)$, $(3, 1)$

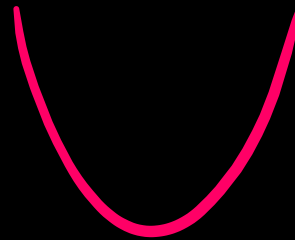
$$E = \left(\frac{1}{1 + e^{-w(-1)}} - 0 \right)^2 + \left(\frac{1}{1 + e^{-w(1)}} - 1 \right)^2 + \left(\frac{1}{1 + e^{-w(-3)}} - 0 \right)^2 + \left(\frac{1}{1 + e^{-w(3)}} - 1 \right)^2$$

뉴런에 바이어스 b 추가

w 값에 따른 오류표시

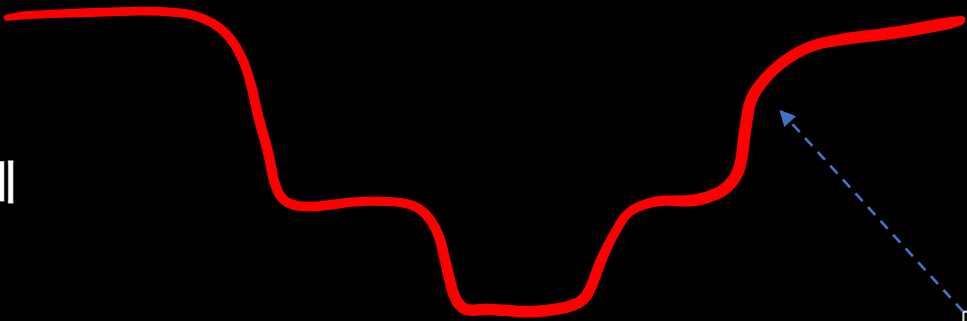
$$\left(w, \frac{E}{2} \right)$$

오류 그래프, 어떤 문제?

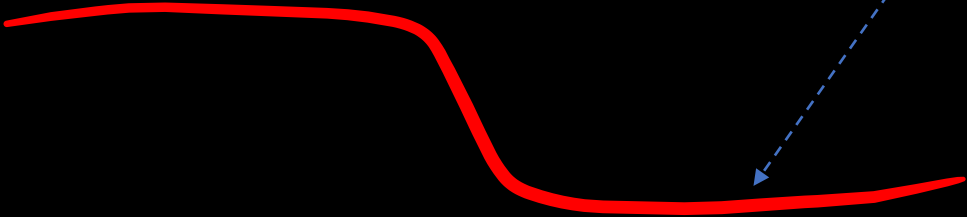


ex1)

경사하강 하는데
문제없을까?



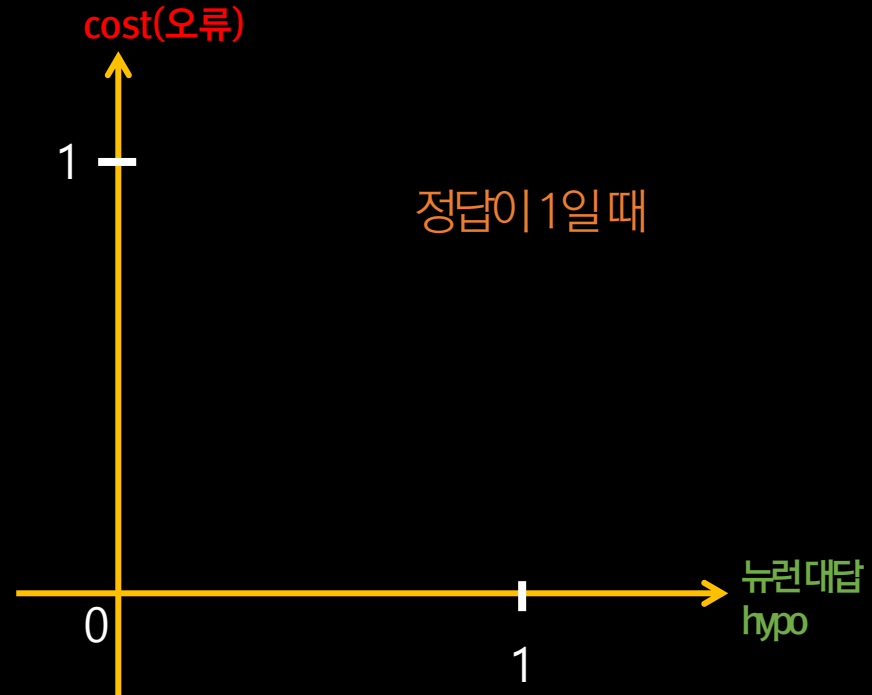
ex2)



MSE 오류 그래프

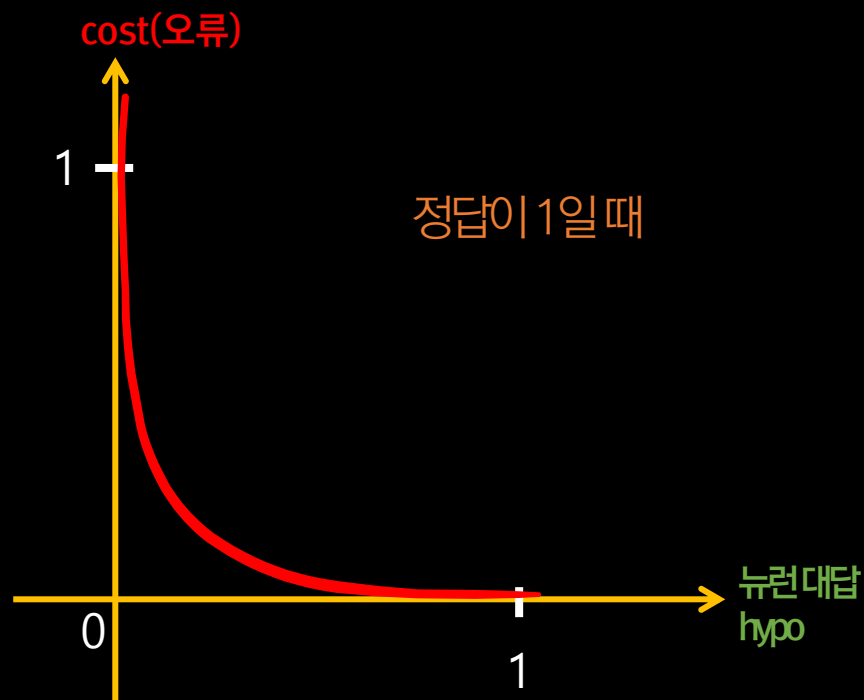
새로운 오류 함수

- 정답이 1일 때
 - 뉴런이 1이라고 대답하면 $\text{good}(\text{오류 } 0)$!
 - 뉴런이 0이라고 정반대로 대답하면 $\text{bad}(\text{오류 } \infty)$!



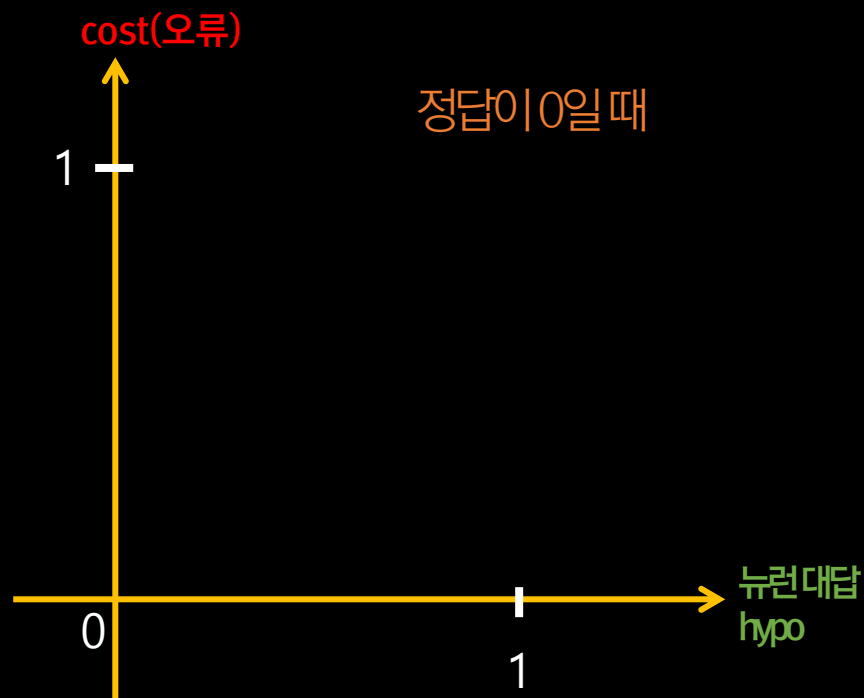
새로운 오류 함수

- 정답이 1일 때
 - 뉴런이 1이라고 대답하면 good(오류 0)!
 - 뉴런이 0이라고 정반대로 대답하면 bad(오류 ∞)!



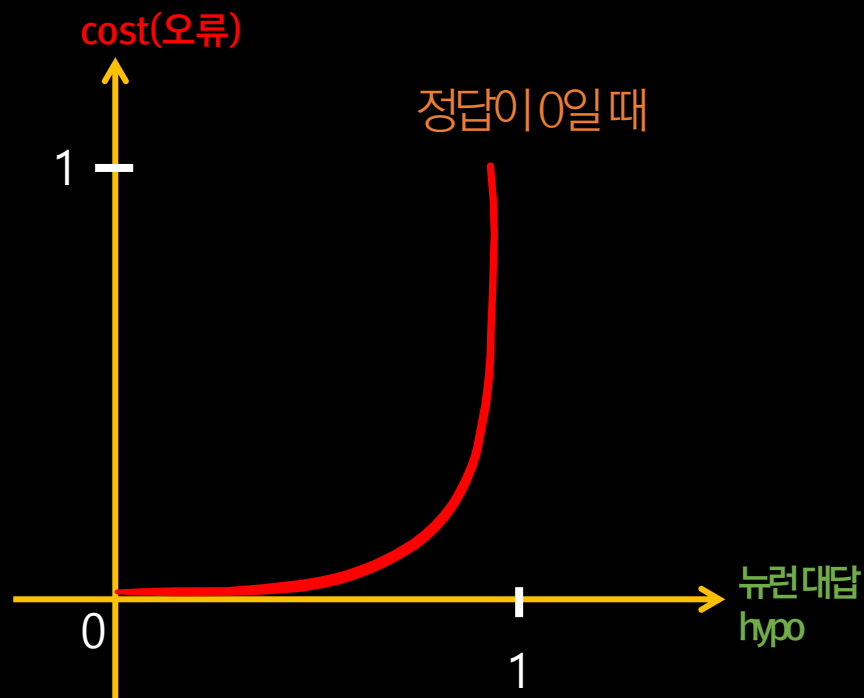
새로운 오류 함수

- 정답이 0인데
 - 뉴런이 1이라고
정반대로 대답하면
 $\text{bad}(\text{오류 } \infty)$!
 - 뉴런이 0이라고
대답하면 $\text{good}(\text{오류 } 0)$!



새로운 오류 함수

- 정답이 0인데
 - 뉴런이 1이라고
정반대로 대답하면
bad(오류 ∞)!
 - 뉴런이 0이라고
대답하면 good(오류 0)!





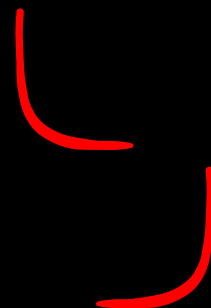
$$\begin{aligned} E &= -\log(h) && y = 1 \text{ 일 때} \\ E &= -\log(1 - h) && y = 0 \text{ 일 때} \end{aligned}$$

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

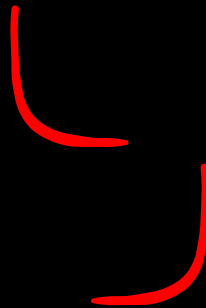
$$E = -\log\left(1 - \frac{1}{1 + e^{-wx}}\right)$$

새로운 오류 함수

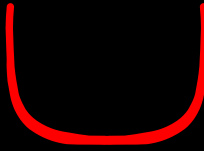
$$E = \begin{cases} -\log(h) & : y = 1 \\ -\log(1 - h) & : y = 0 \end{cases}$$



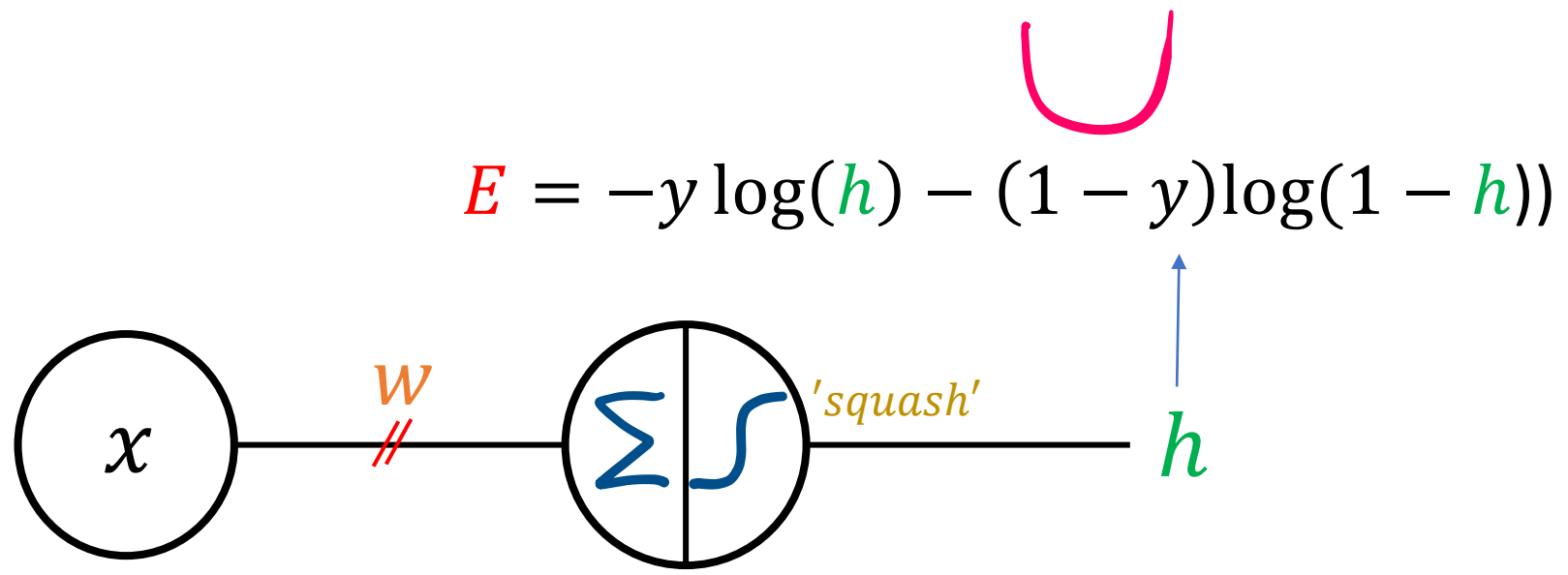
새로운 오류 함수

$$E = \begin{cases} -\log(h) & : y = 1 \\ -\log(1 - h) & : y = 0 \end{cases}$$


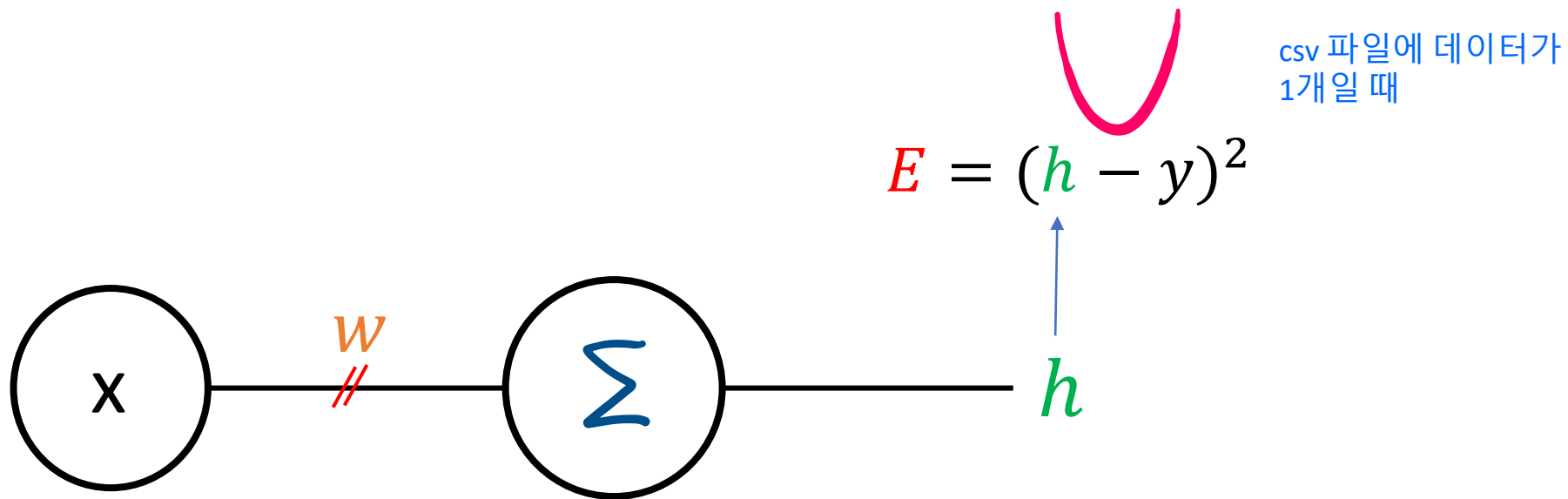


$$E = -y \log(h) - (1 - y) \log(1 - h)$$


$$w = w - \alpha \cdot \frac{\partial E}{\partial w}$$



$$E = -\frac{1}{N} \sum y \log(h) + (1 - y) \log(1 - h)$$

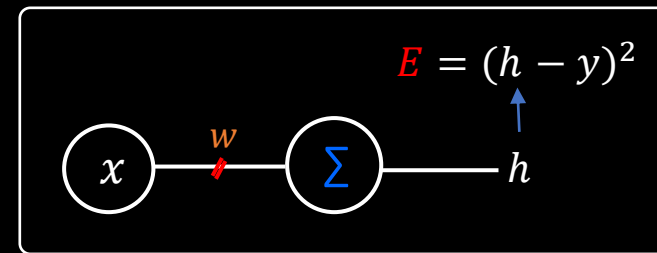


$$E = \frac{1}{N} \sum (h - y)^2$$

csv 파일에 데이터가 여러 개일 때

새로운 오류 함수를
계산 그래프로 표현하면?

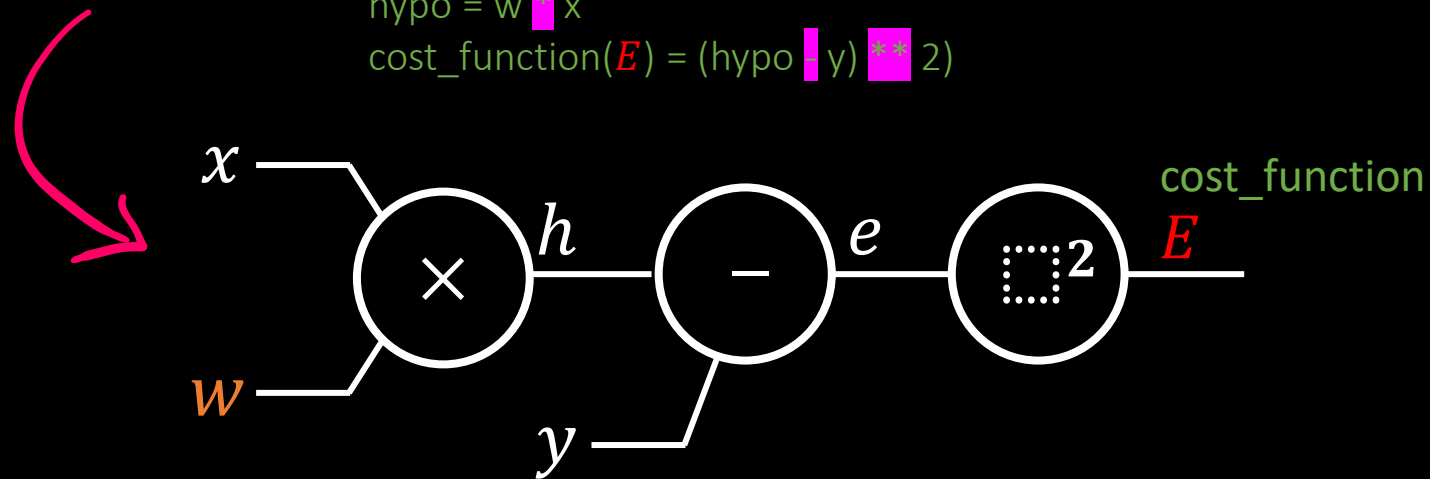
계산 그래프



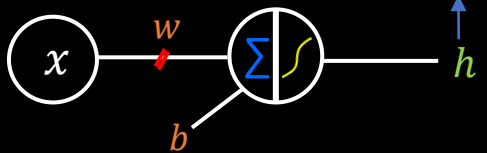
$$E = (wx - y)^2$$

hypo = w * x

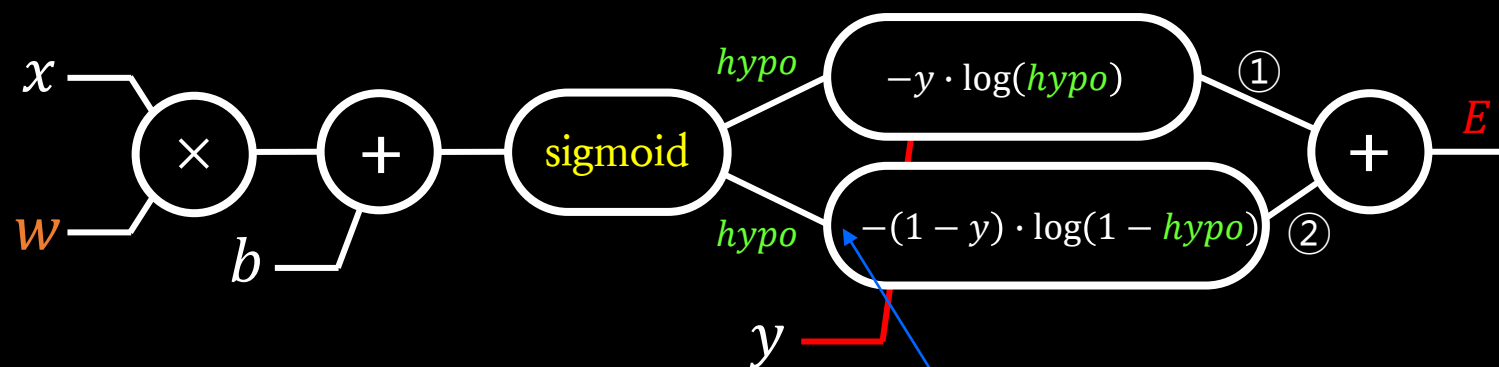
cost_function(E) = (hypo - y) ** 2



계산 그래프

$$E = -(y \log(h) + (1 - y) \log(1 - h))$$


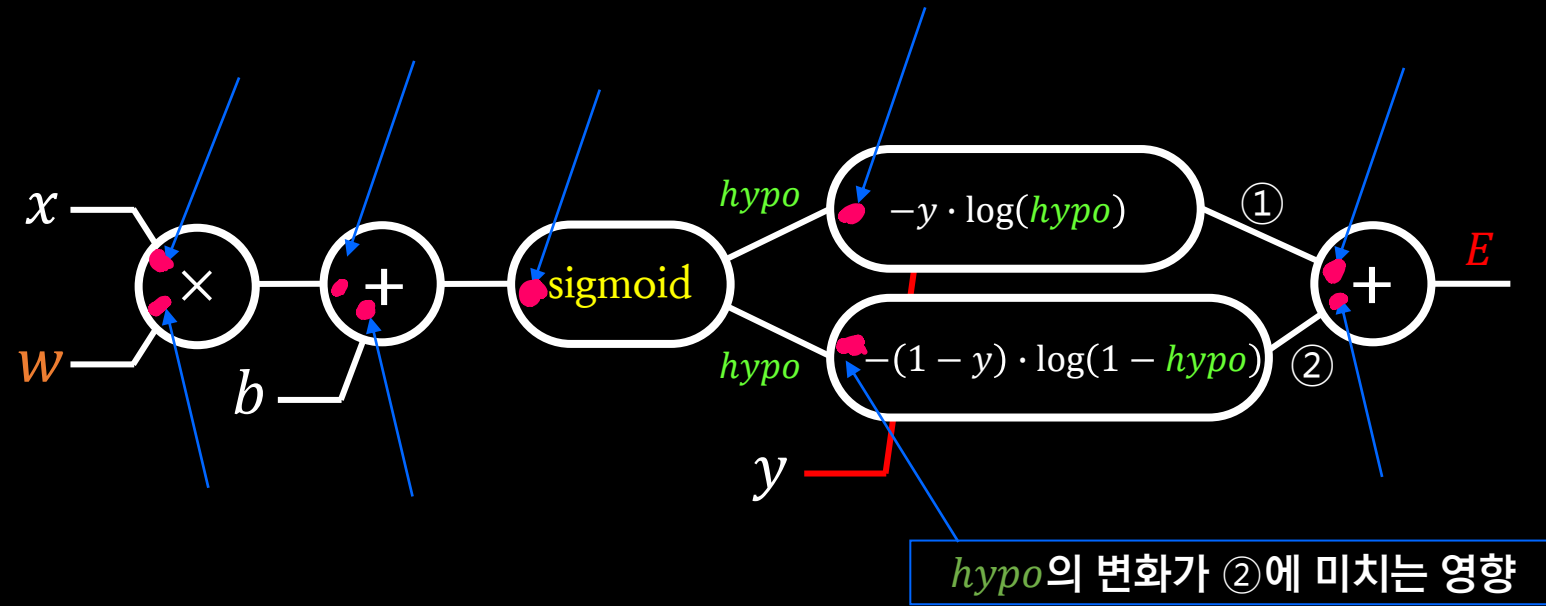
$$E = \overset{\textcircled{1}}{-y \cdot \log(hypo)} - \overset{\textcircled{2}}{(1 - y) \cdot \log(1 - hypo)}$$



$hypo$ 의 변화가 ②에 미치는 영향

$$\frac{\partial \textcircled{2}}{\partial h}$$

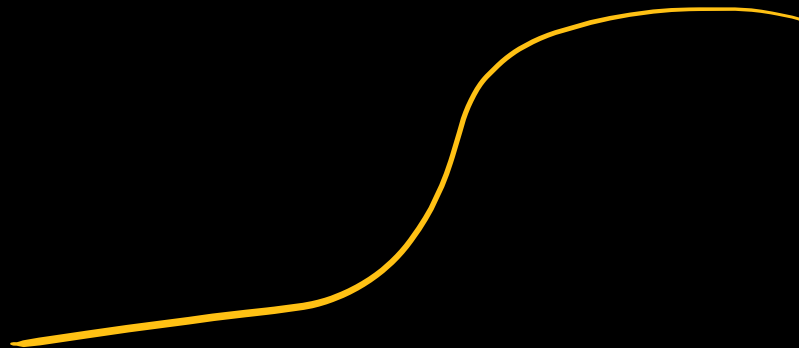
지역(게이트)별 미치는 영향력



$$\frac{\partial ②}{\partial h}$$

sigmoid 함수를 그려보라.

sigmoid 함수 기울기(미치는 영향)를 구해보라.



로그(\ln) 함수를 그려보라.

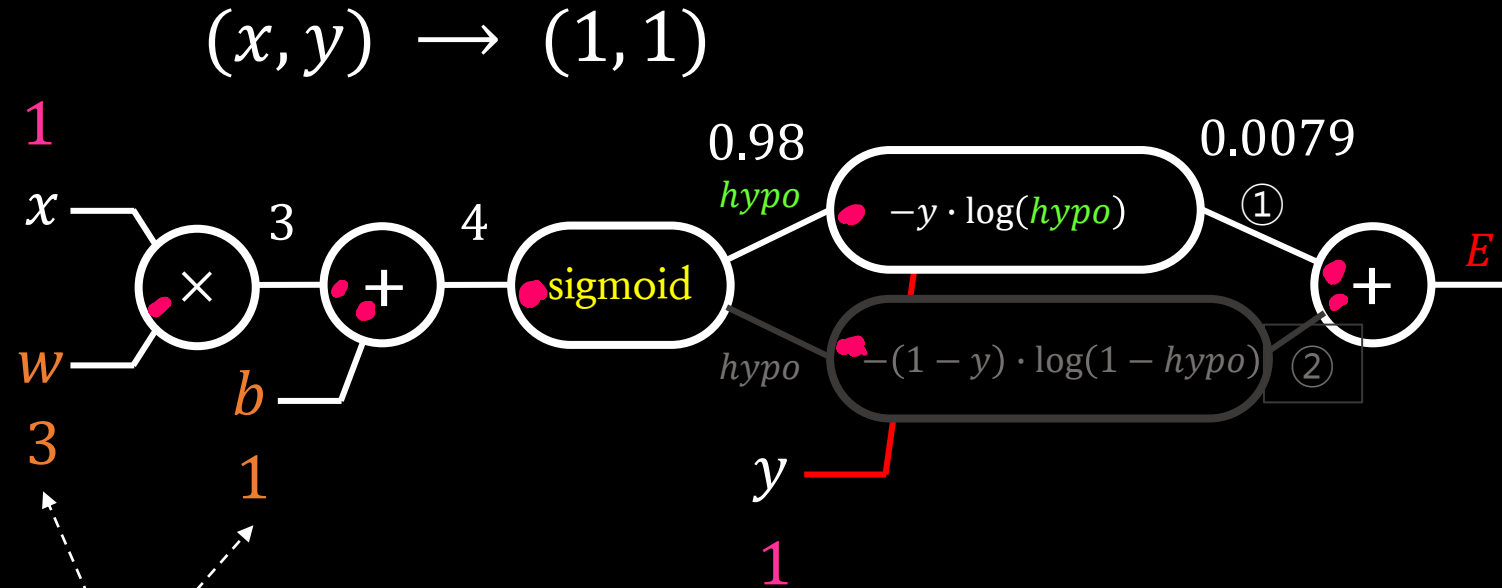
로그 함수 기울기(미치는 영향)를 구해보라.

앞으로 전파

a.csv

x, y

1, 1



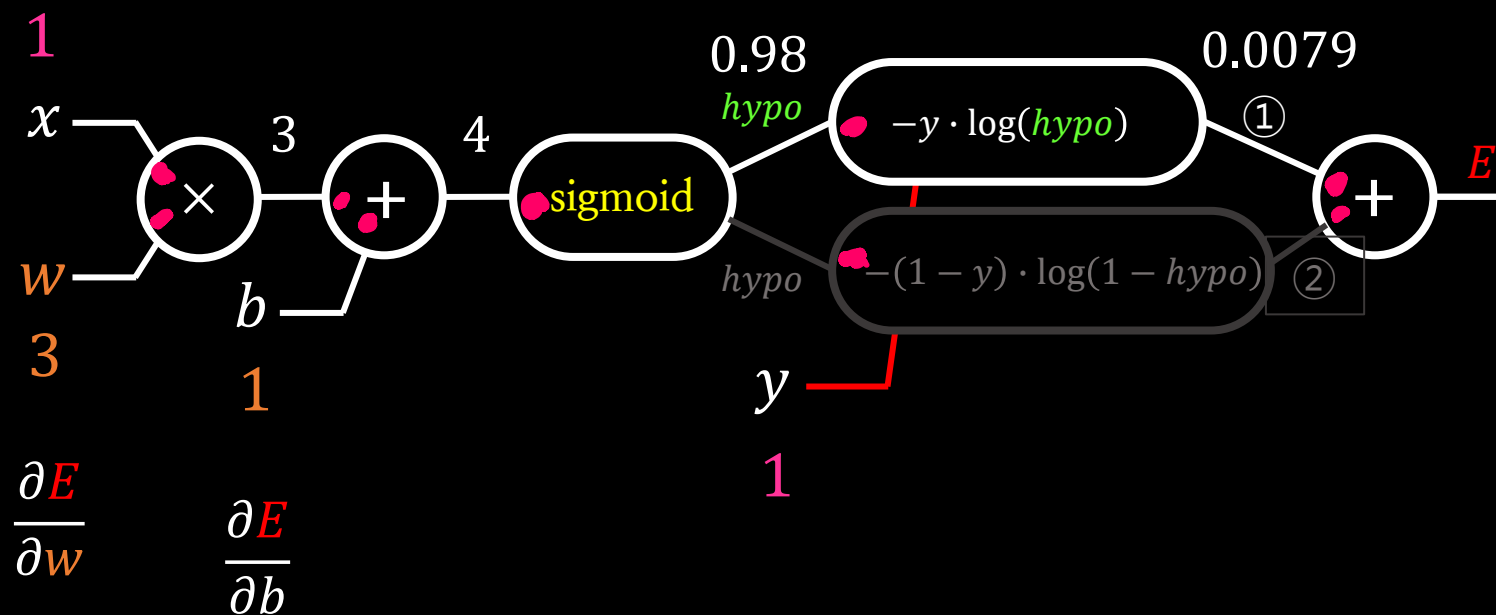
처음에 난수로 초기화 → 경사하강하여 최적화됨 (이를 학습이라 함.)

역전파

a.csv

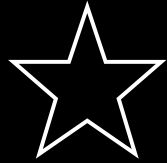
x, y

1, 1



$$w = w - \alpha \cdot \frac{\partial E}{\partial w}$$

$$b = b - \alpha \cdot \frac{\partial E}{\partial b}$$



w, b 를

업데이트(튜닝, 학습)한다는
의미는?

결정경계

$$wx + b = 0$$

더 나은 결정 경계를
찾는다는 의미!

Lab 11.py

모든 입력 값을
0과 1로 분류하기

$$\text{cost} = -(y \log(H(X)) + (1 - y) \log(1 - H(X)))$$

```
x_data = [-2., -1, 1, 2]
y_data = [0., 0, 1, 1]
```

```
#----- a neuron
w = tf.Variable(tf.random_normal([1]))
hypo = tf.sigmoid(x_data * w)
```

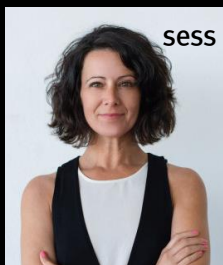
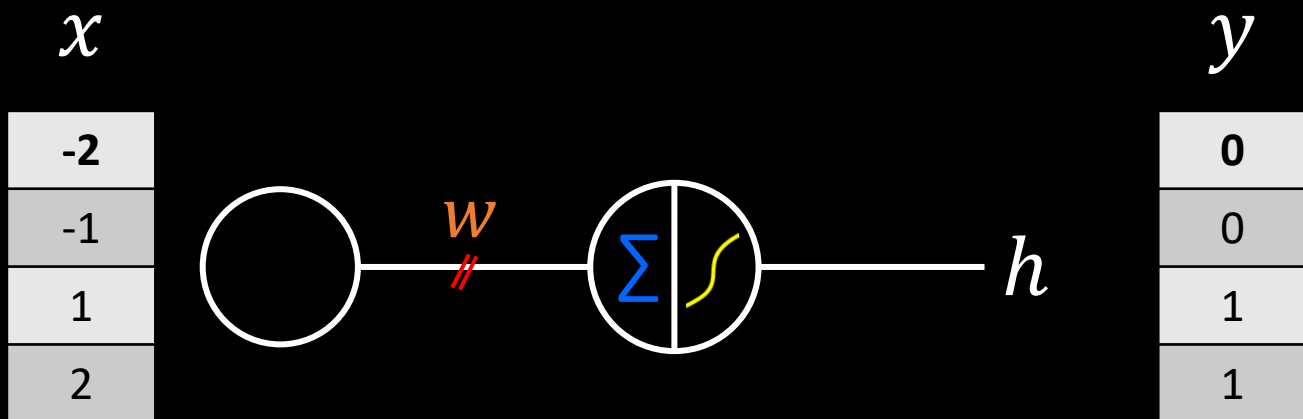
```
#----- learning
cost = -tf.reduce_mean(y_data * tf.log(hypo) +
                       tf.subtract(1., y_data) * tf.log(tf.subtract(1., hypo)))

train = tf.train.GradientDescentOptimizer(learning_rate=0.01).minimize(cost)

sess = tf.Session()
sess.run(tf.global_variables_initializer())

for step in range(5001):
    sess.run(train)
```

```
#----- testing(classification)
predicted = tf.cast(hypo > 0.5, dtype=tf.float32)
p = sess.run(predicted)
print("Predicted: ", p)
```



`sess.run(train)`

$-\log(h)$

$-\log(1 - h)$

sess.run()

텐서 w

텐서 $w x$

텐서 $h = \text{sigmod}(w x)$

텐서 $-y \log(h), -(1-y) \log(1-h)$

텐서 $-y \log(h) - (1-y) \log(1-h)$

텐서 cost

텐서 $h > 0.5$

텐서 predicted

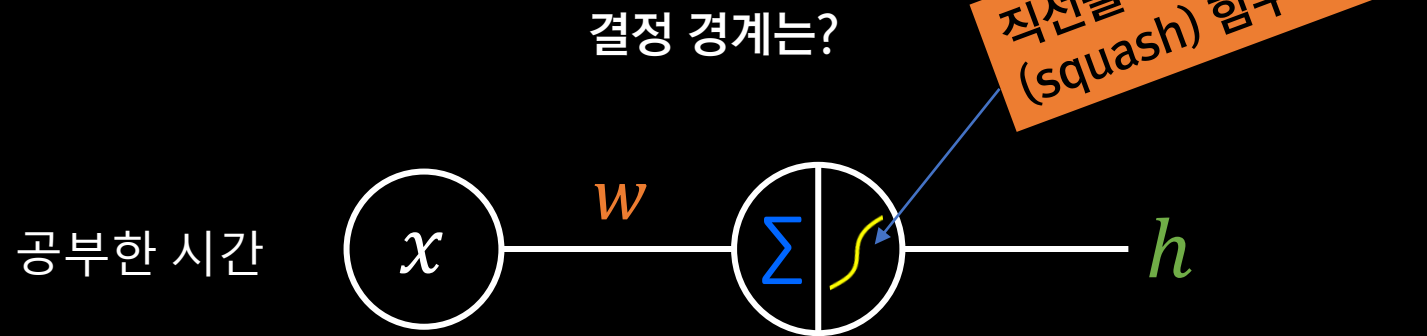
행렬 연산
tf.reduce_mean()

Lab 12.py

With a bias

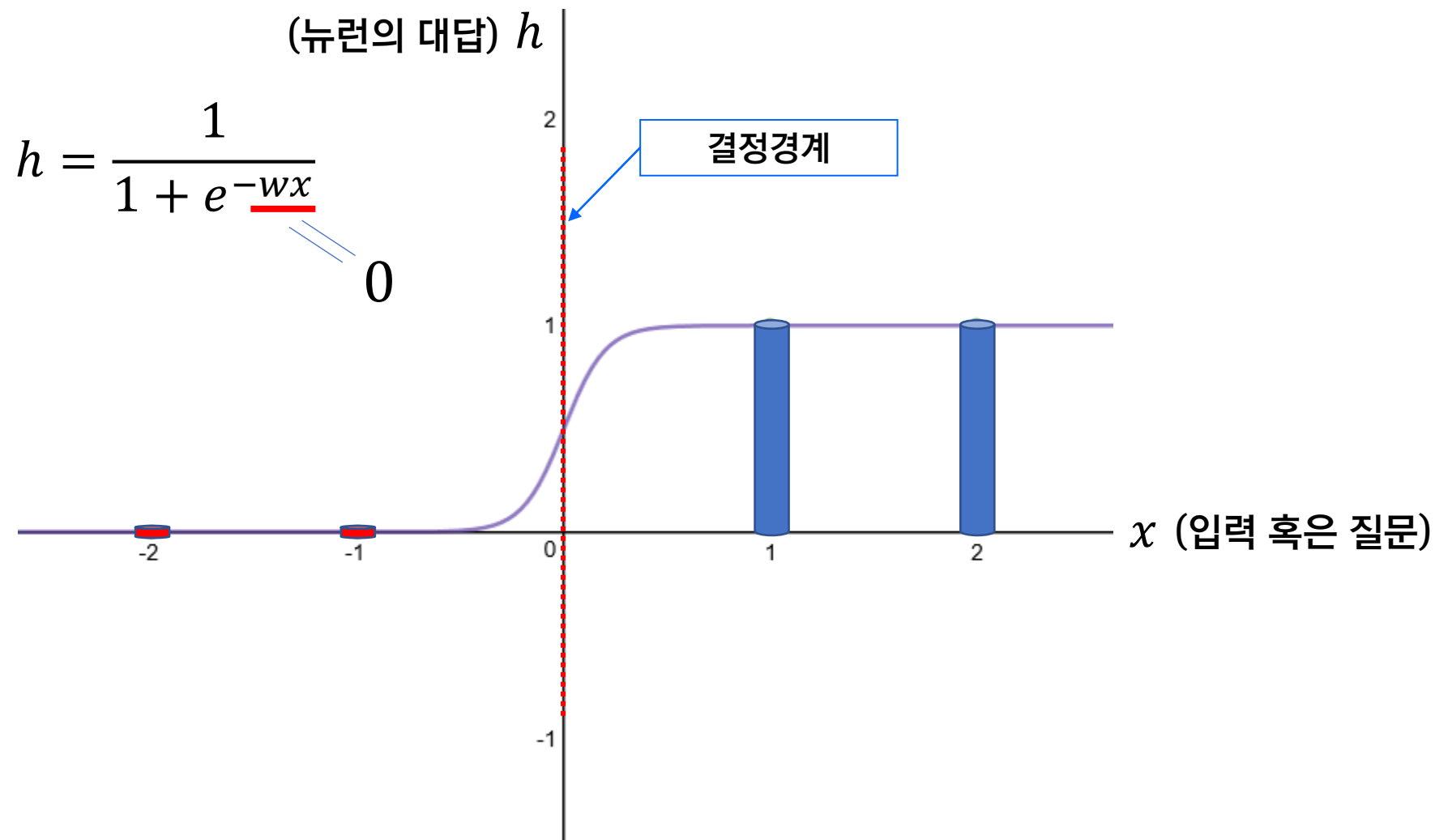
결국, 학습 후 최종 결과는?
그리고 결정경계에 대하여

신경 세포 (1 입력)

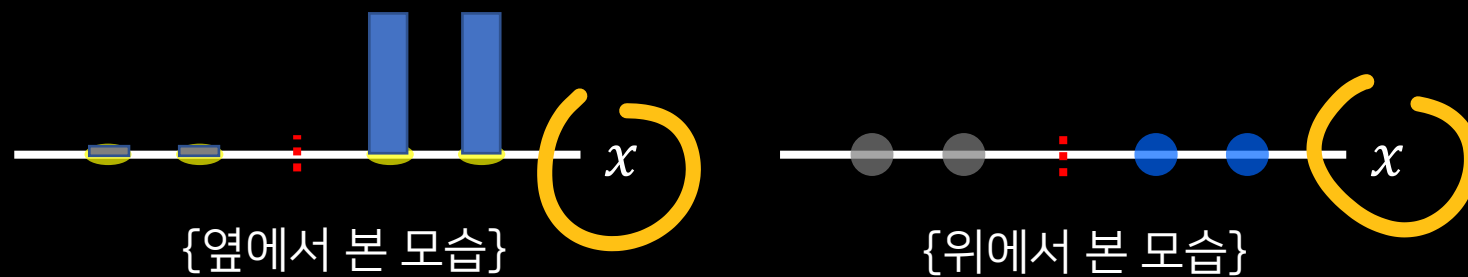


$$h = \frac{1}{1 + e^{-(wx)}}$$

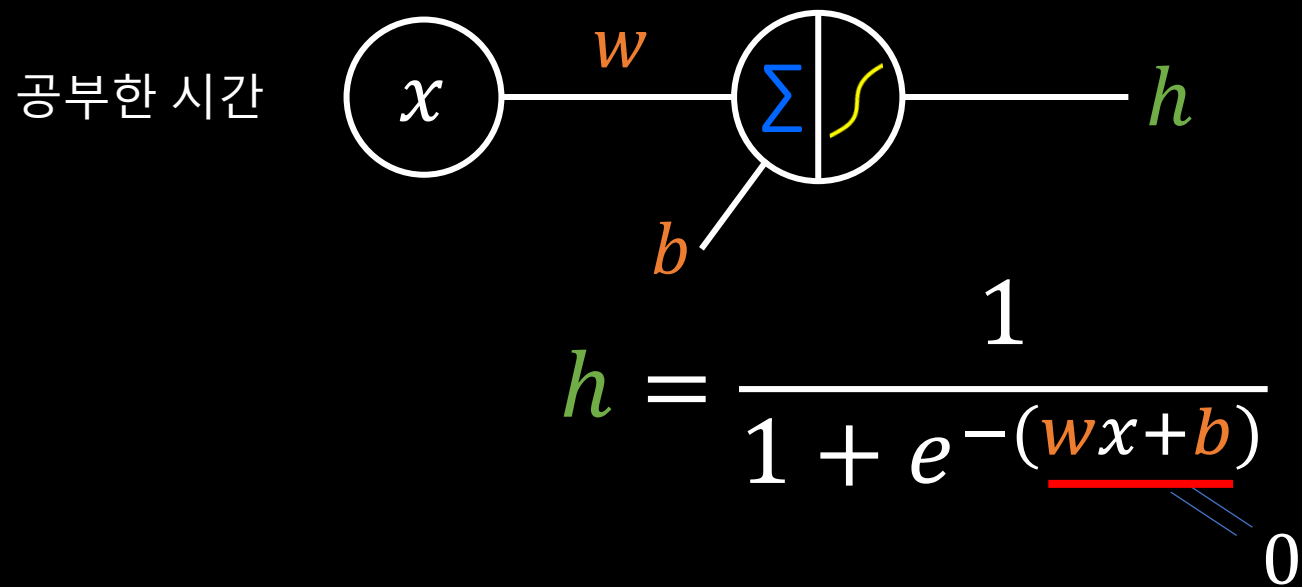
신경 세포 (1 입력)



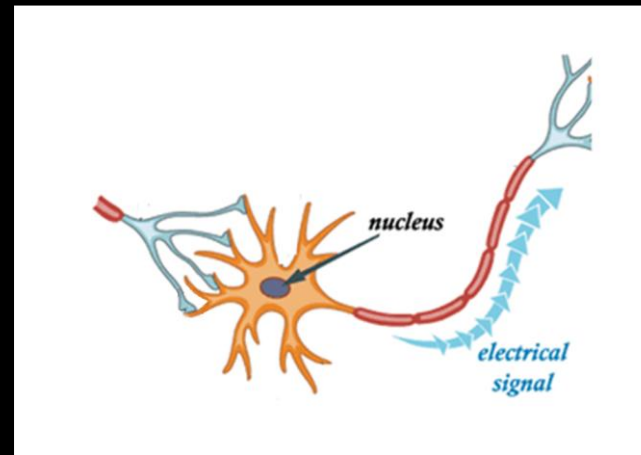
신경 세포 (1 입력)



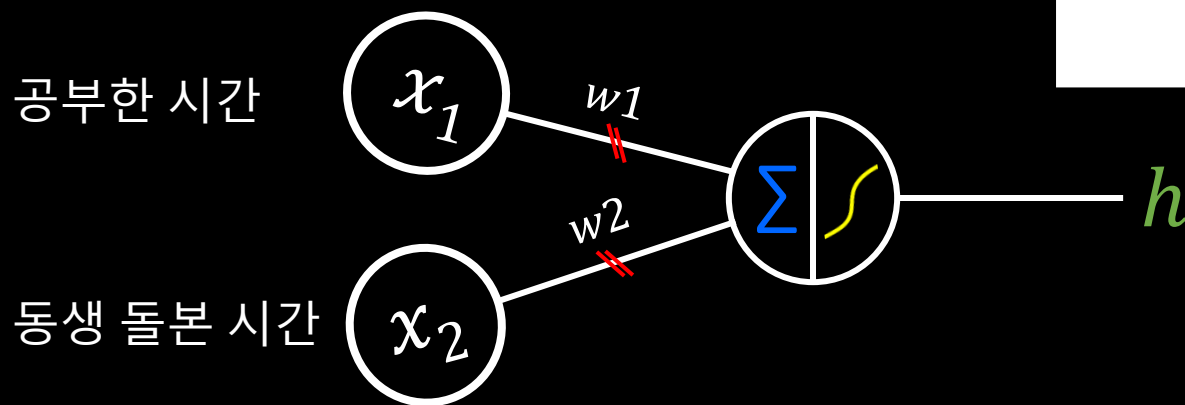
신경 세포 (1 입력)



신경 세포 (2 입력)

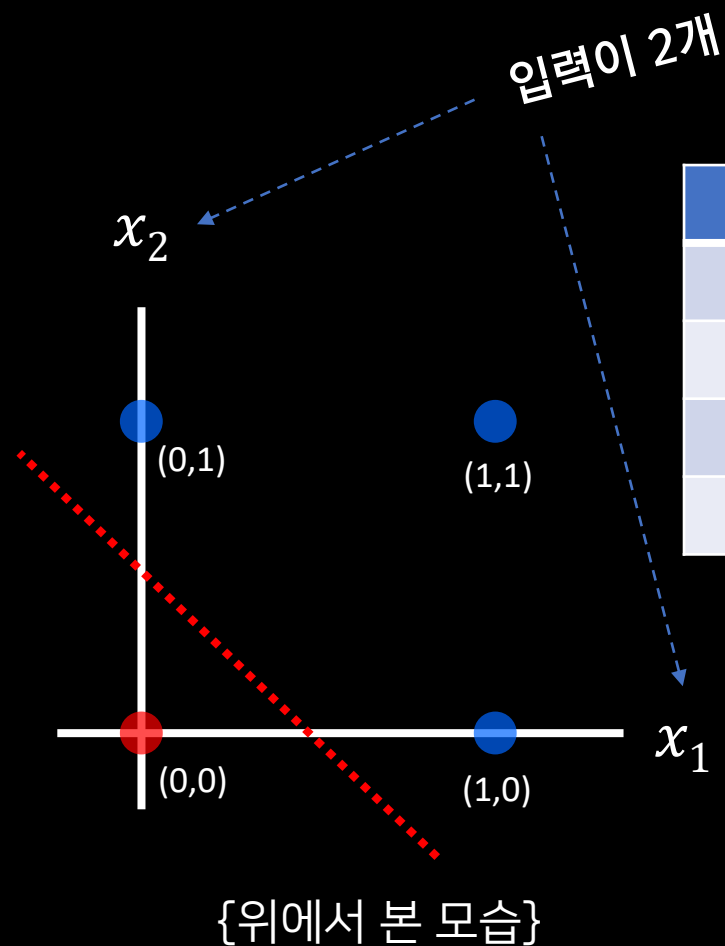
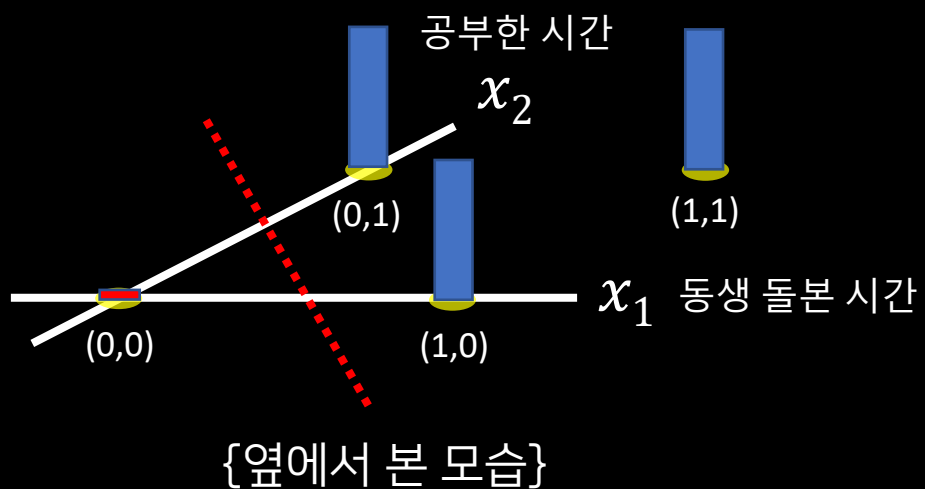


결정경계는?



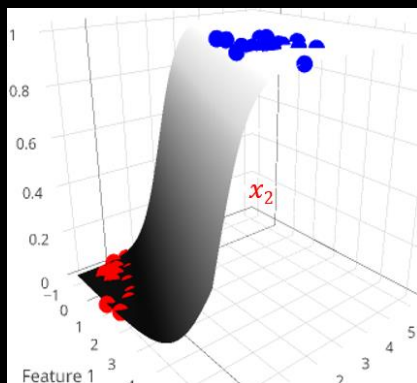
$$h = \frac{1}{1 + e^{-(w_1 x_1 + w_2 x_2)}}$$

신경 세포 (2 입력)



배열 혹은 csv 파일

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



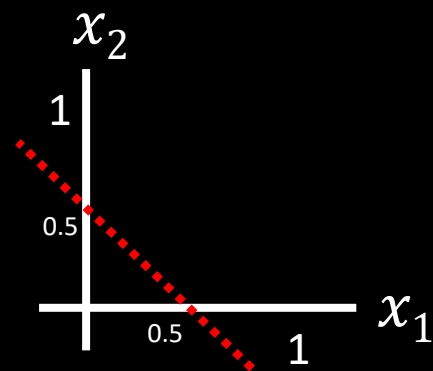
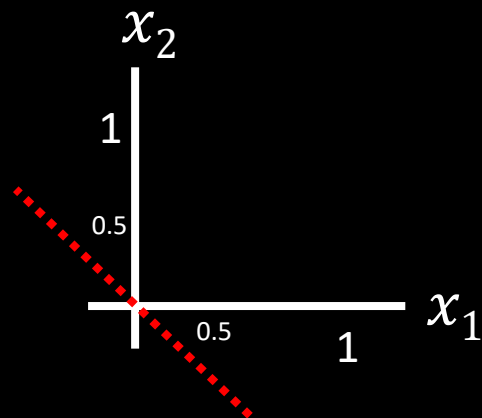
로지스틱 회귀: 특징 2개일때

신경 세포 (2 입력)

$$x_1 + x_2 = 0$$

$$x_1 + x_2 - 0.5 = 0$$

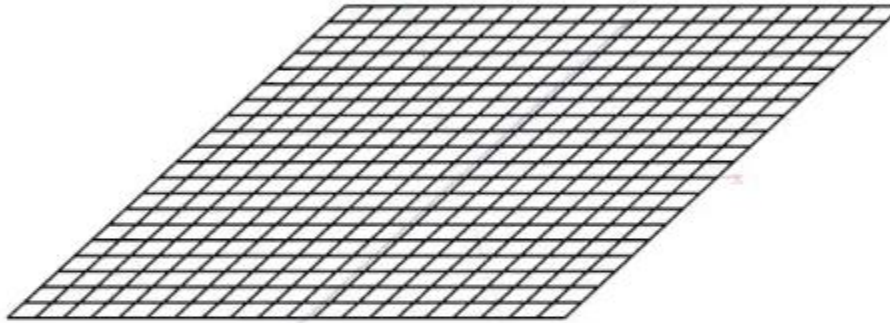
위에서 본 모습



2-Input 뉴런이 만드는 결정경계

`sigmoid(w1·length + w2·width + b)`

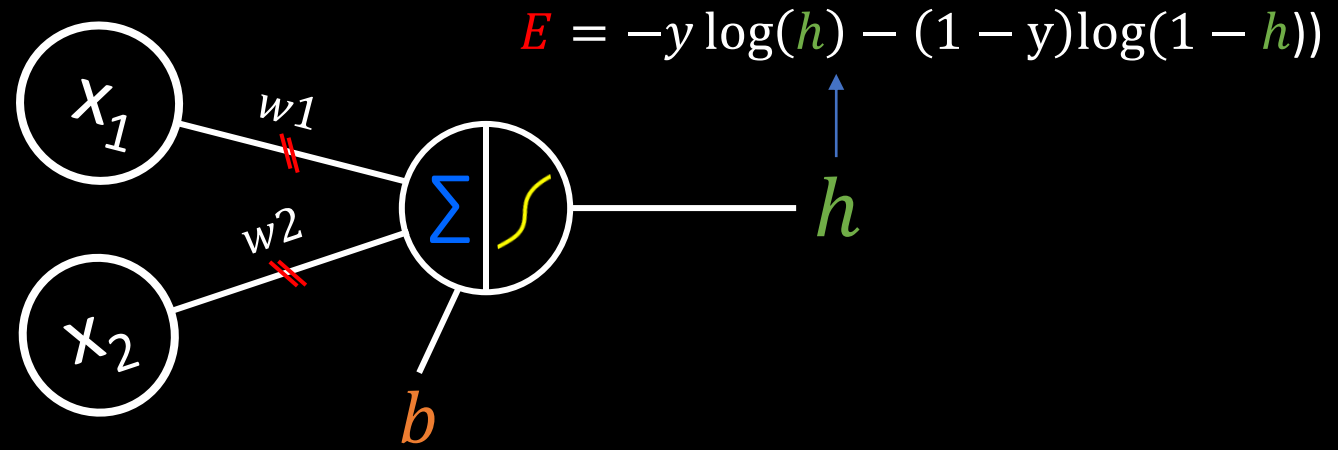
$$w_1x_1 + w_2x_2 + b = 0$$



```
surface(f(x,z)=sig(w1·x+w2·z+b))  
w1 = 0.00  
w2 = 0.00  
b = 0.00
```

Lab 13.py

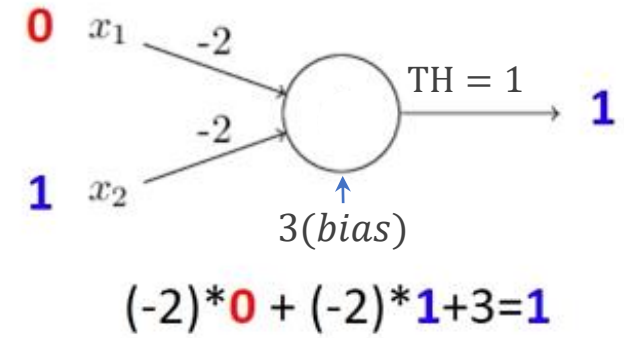
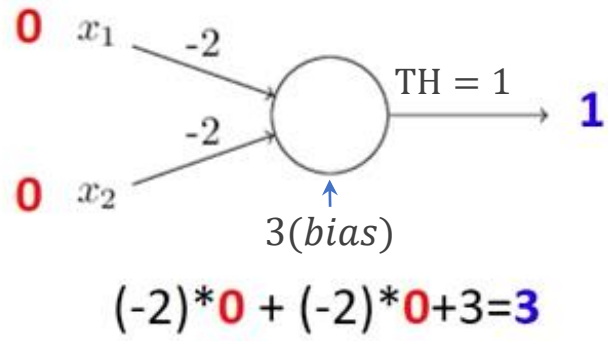
OR 연산을 수행하는
2입력 뉴런 구현하기
(1개의 결정경계)



x_1	x_2	$AND(h)$
0	0	0
0	1	0
1	0	0
1	1	1

NAND 게이트

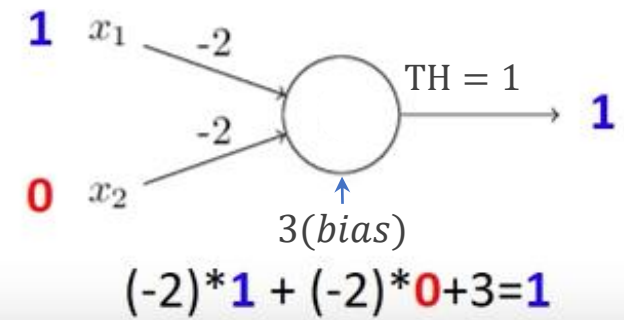
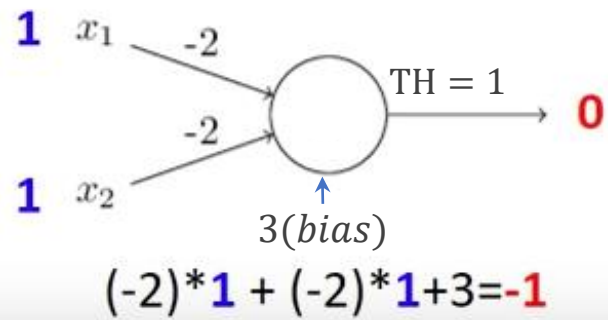
- NAND 게이트 구현하기
- 기능적으로 완전한 게이트,
모든 디바이스는 이 게이트로
구현가능



NAND

Truth Table

Input A	Input B	Output Q
0	0	1
0	1	1
1	0	1
1	1	0



프로그래밍 vs. 학습

이번 학습에서는

- 로지스틱 리그레션을 위한 오류함수를 어떻게 디자인 하는지 이해할 수 있다.
- 한 개의 뉴런이 만드는 결정 경계를 이해할 수 있다.
- 결정경계를 옆에서 본 모습, 위에서 본 모습을 이해할 수 있다.
- 신경세포의 입력 수에 따른 결정경계 모양을 상상할 수 있다.
- 학습 = 제대로 된 결정경계 만들기