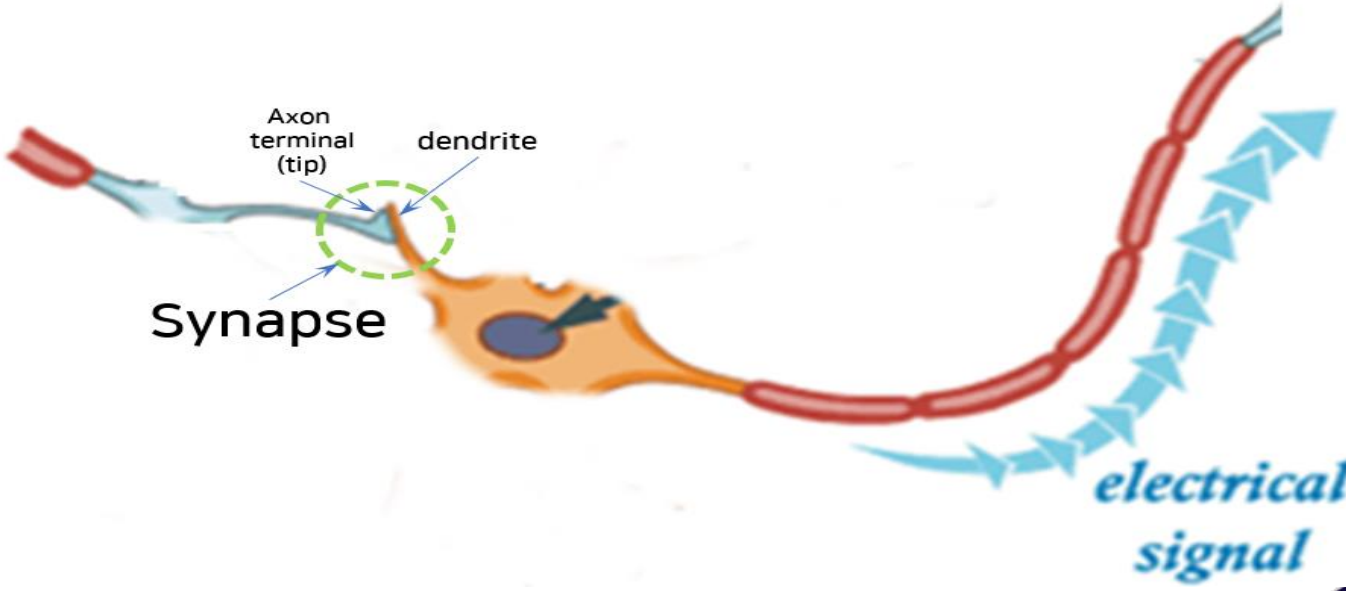
AI and Deep Learning

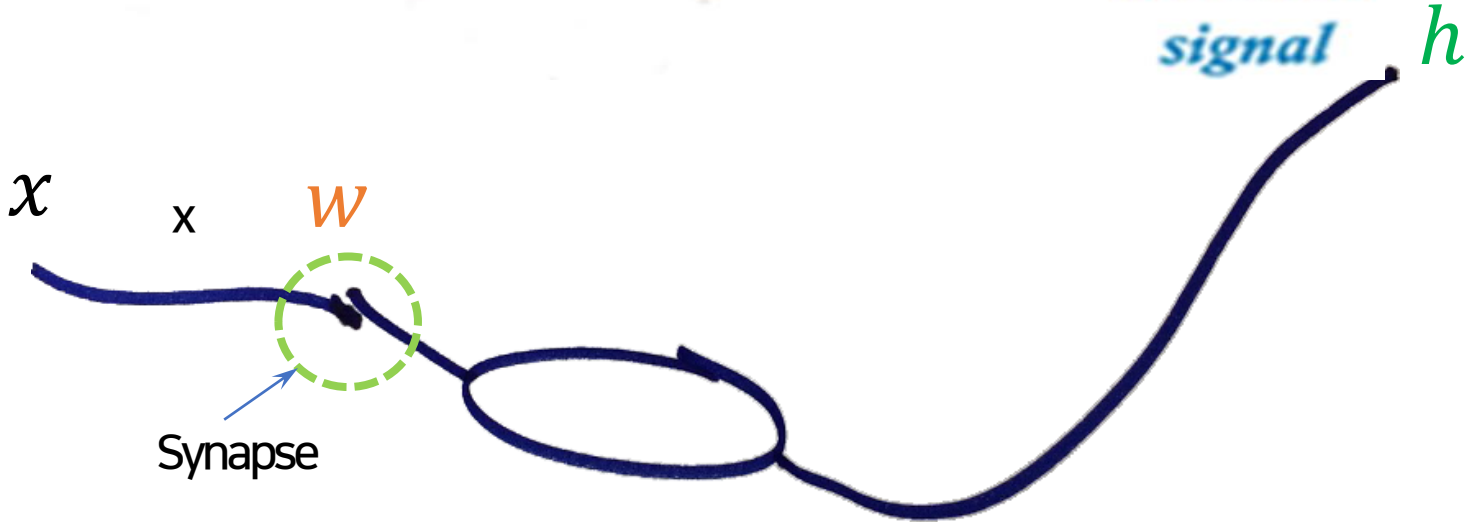
6. Logistic Regression(1)

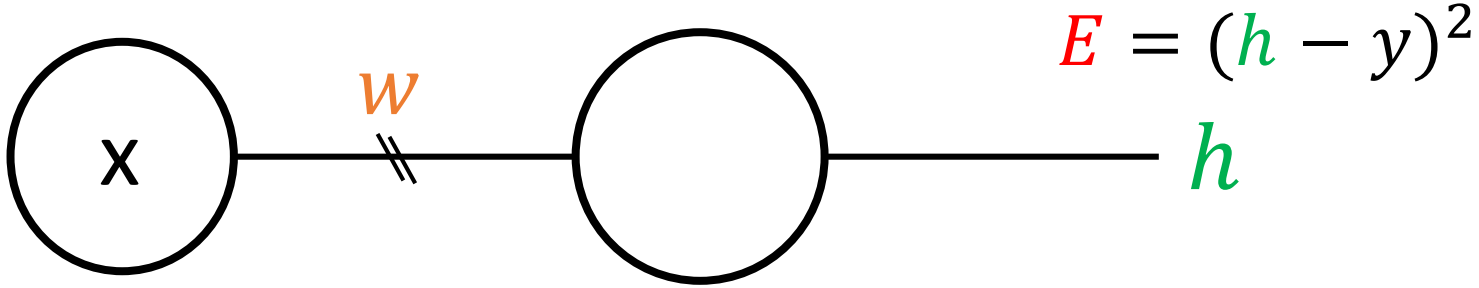
Yung-Cheol BYUN, Jeju National Univ.

Agenda

- Artificial Intelligence
- Brain, Neurons
- Learning
- Regression
- Deep Neural Networks
- CNN


Axon terminal (tip)
dendrite
Synapse
electrical signal


x
x
w
Synapse
h


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

Logistic Regression

The shape of regression is not linear but logistic.

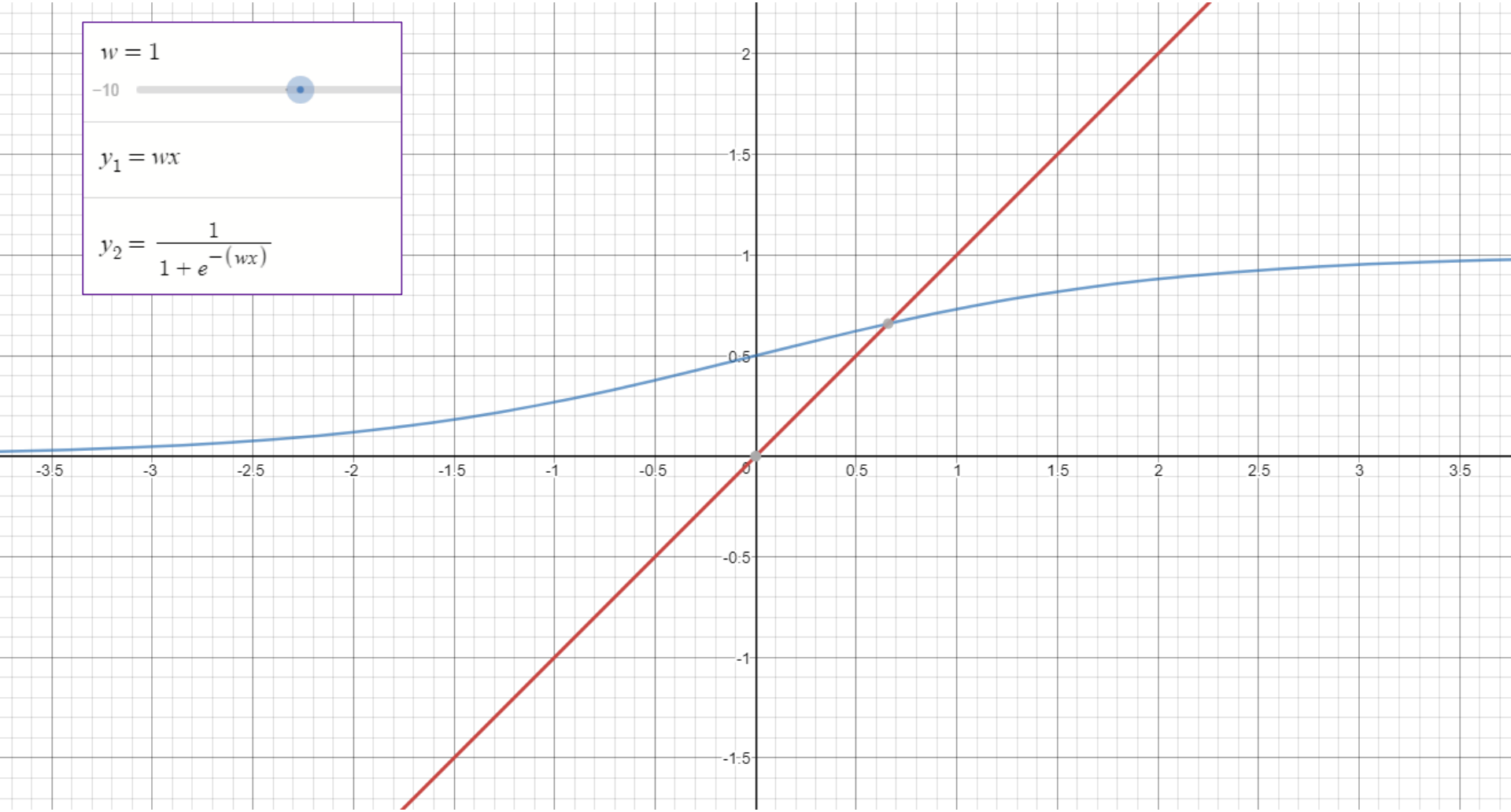
What does that mean?

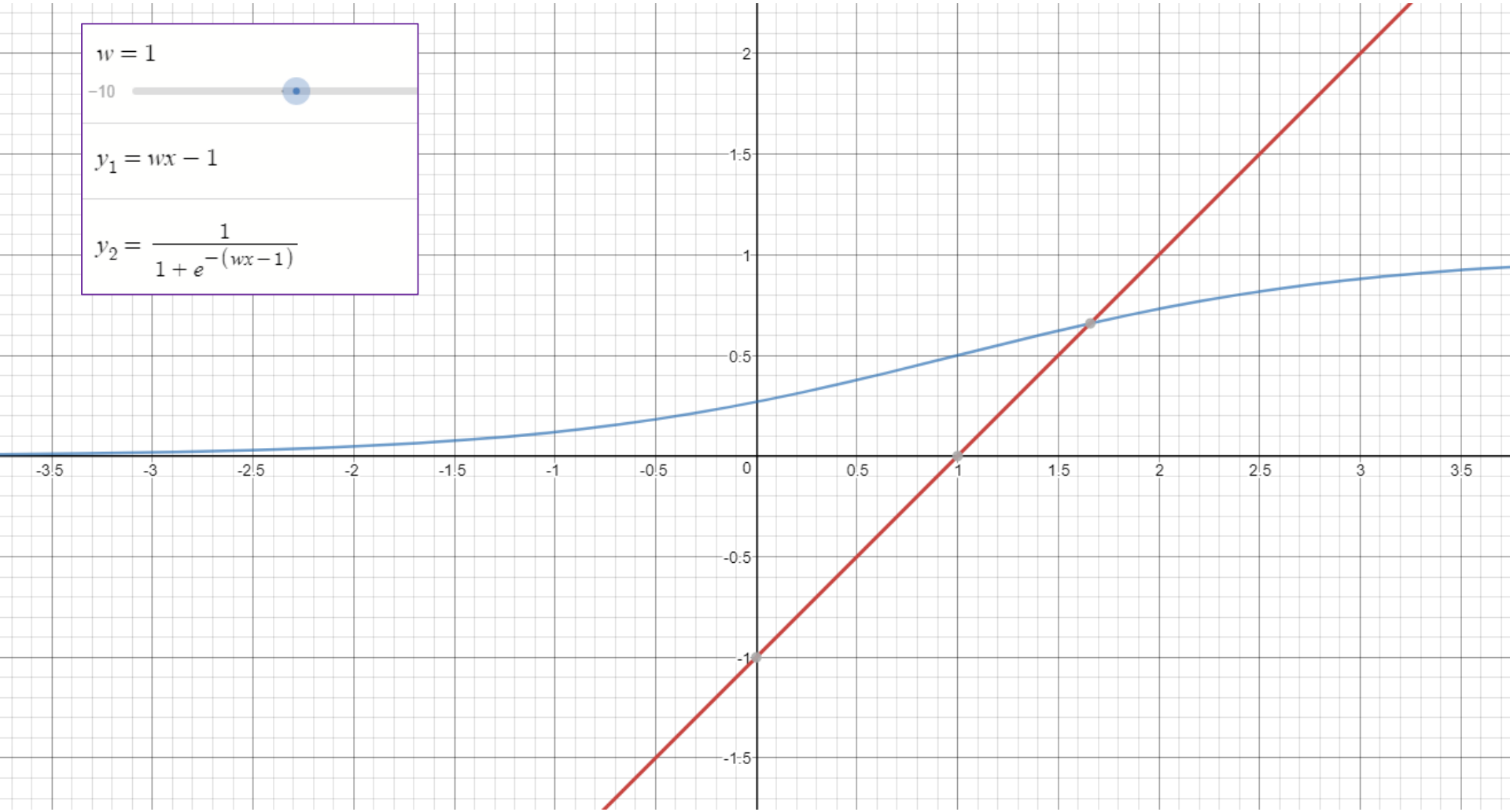
www.desmos.com

x	y
-2	0
-1	0
1	1
2	1

y	0	0	1	1
x	-2	-1	1	2

0, 1을 결정(decision)하는
경계(boundary)


$w = 1$
−10
$y_1 = wx$
$y_2 = \frac{1}{1 + e^{-(wx)}}$
2
1.5
1
0.5
-0.5
-1
-1.5
-3.5
-3
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
3
3.5


$w = 1$
−10
$y_1 = wx - 1$
$y_2 = \frac{1}{1 + e^{-(wx-1)}}$
2
1.5
1
0.5
-0.5
-1
-1.5
-3.5
-3
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
3
3.5

결정 경계

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

$$wx = 0$$

결정 경계

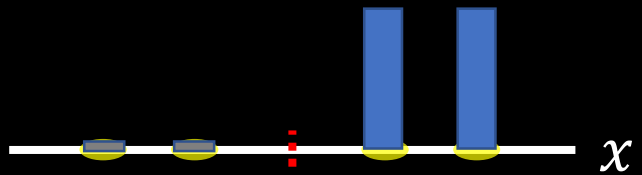
$or,$

$h = \text{sigmoid}\ (wx)$

결정 경계

$wx = 0$

$x = 0$

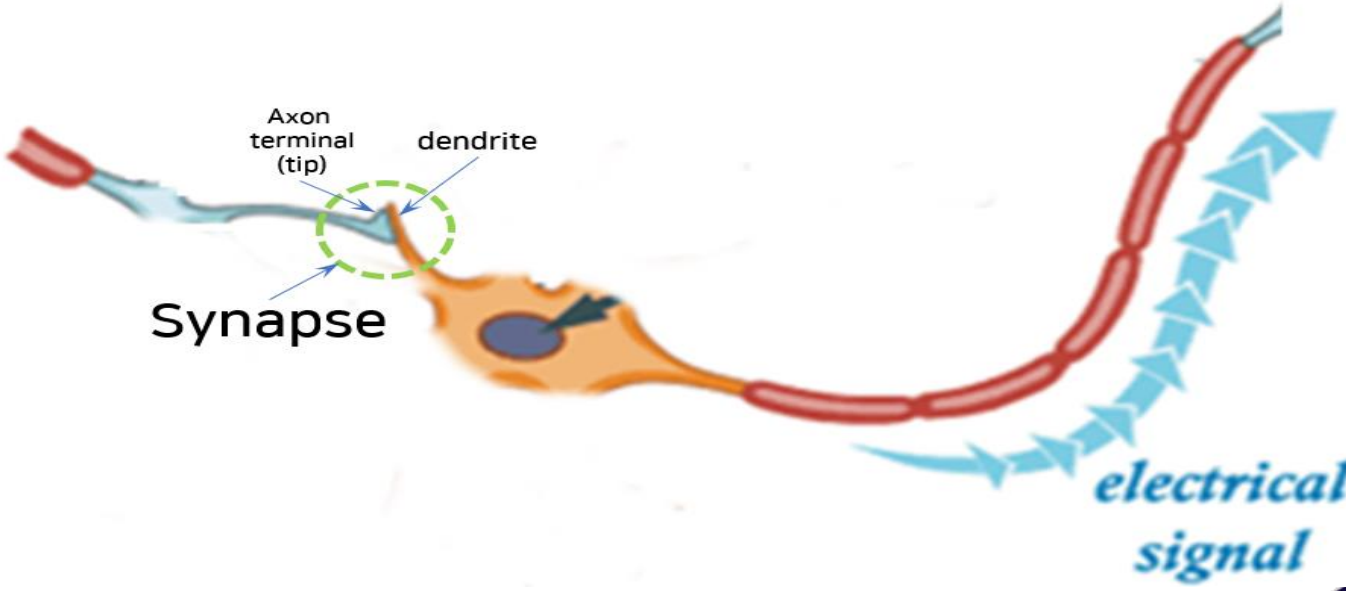


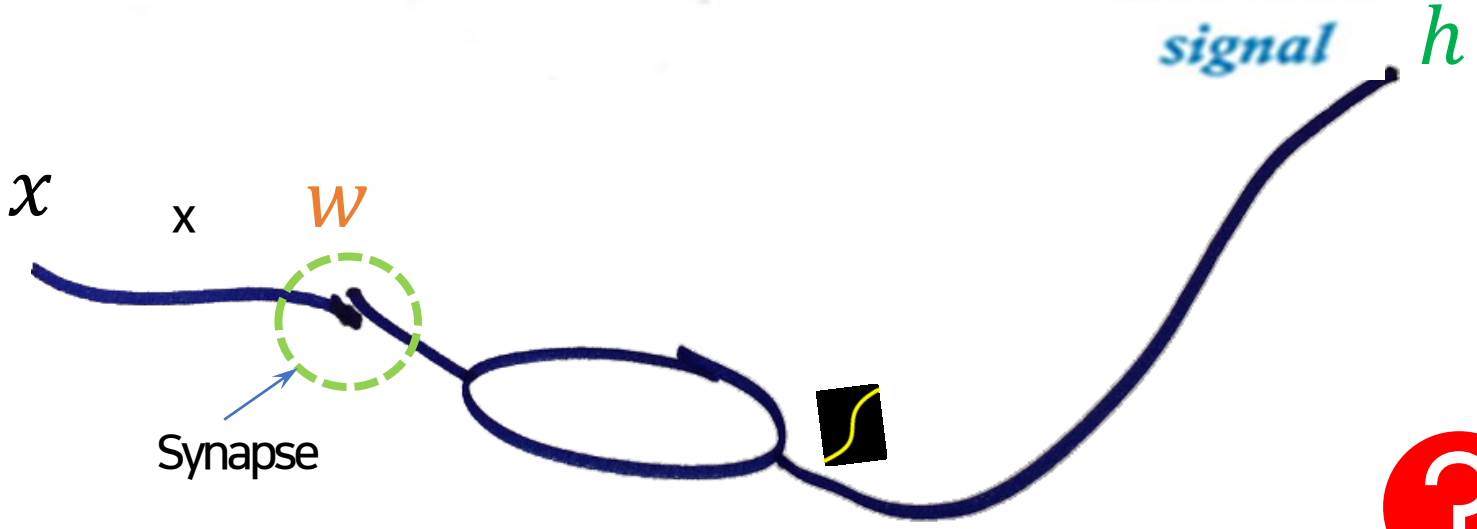
{옆에서 본 모습}

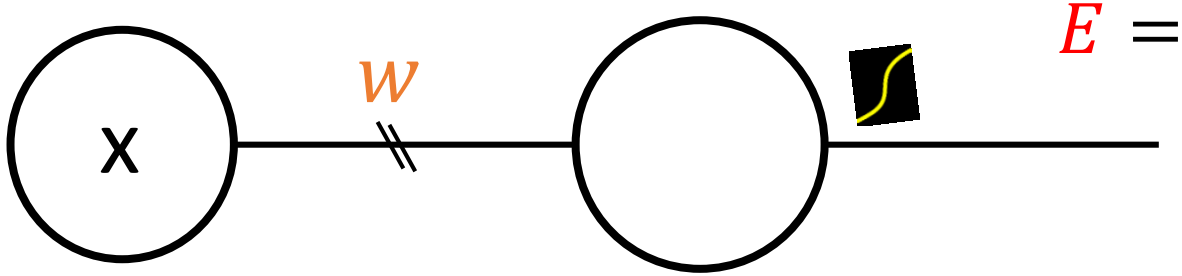
$wx + b = 0$

$x + 1 = 0$

{위에서 본 모습}

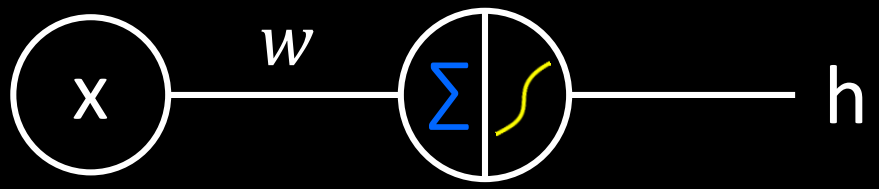

Axon terminal (tip)
dendrite
Synapse
electrical signal


x
x
w
Synapse
h


x
w
h

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

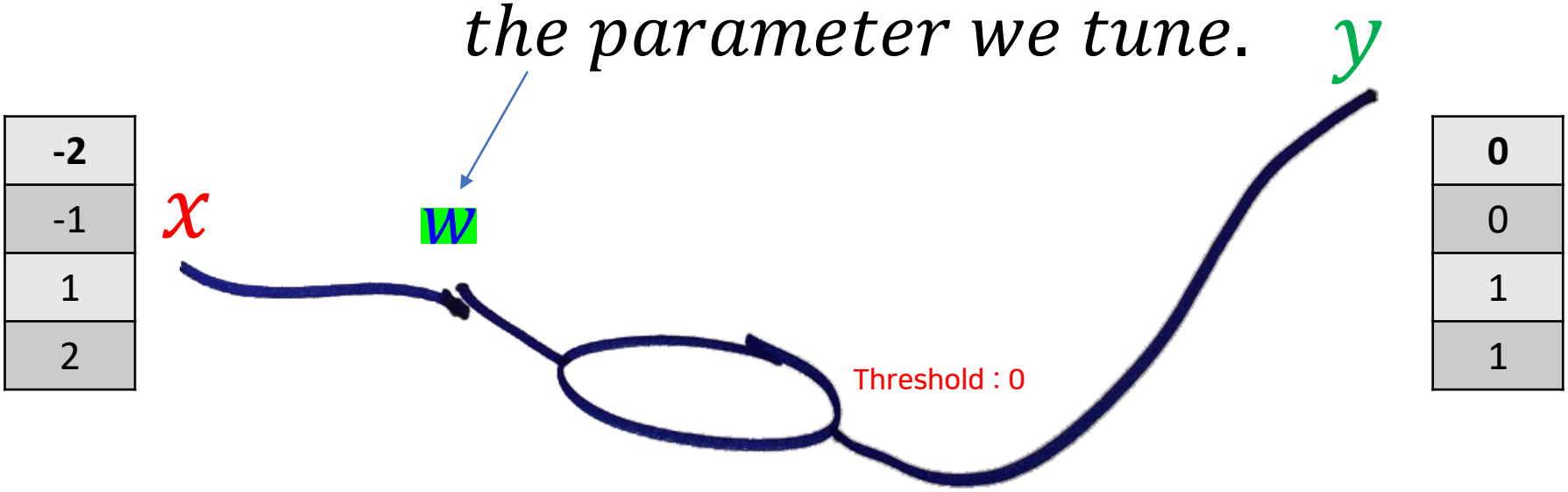
신경 세포 기능



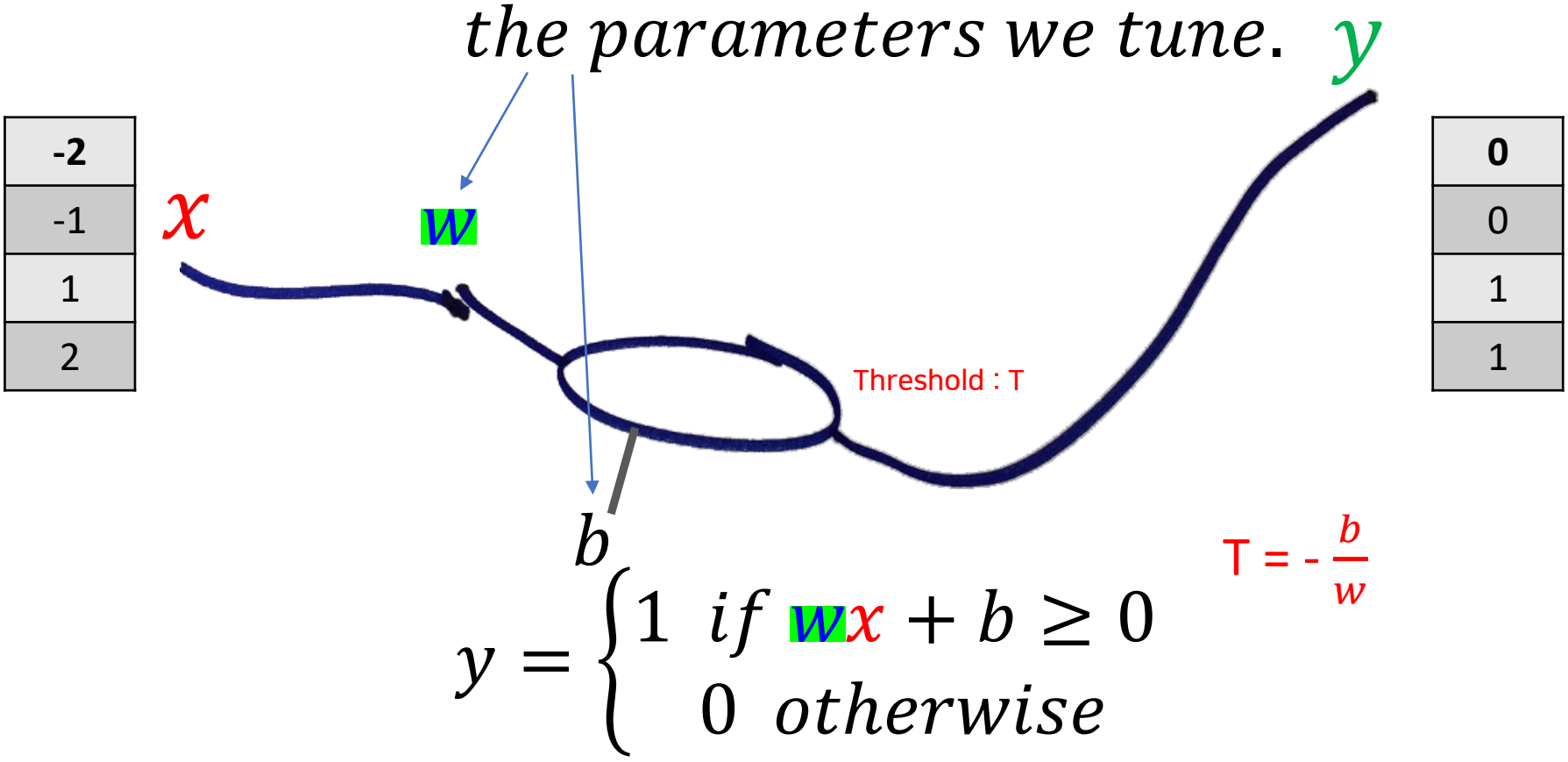
- 신경세포 1개가 할 수 있는 것은?
- 한 개의 입력 x에 따라 0, 혹은 1을 출력(h)함. 예를 들어, 입력이 -1이면 0을, 3이면 1을 출력함.

Classification

- Pass(1) or Fail(0)
- Spam(1) or Ham(0)
- Scam(fraud, 1) or not(0)
- Safe(1) or Dangerous(0)
- Intrusion/virus(1) or not(0)
- Cancer(1) or not(0)
- Binary classification -> Multiple classification



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


the parameters we tune.
y
x
w
b
Threshold : T
-2
-1
1
2
0
0
1
1
T = - b/w
y = 1 if wx + b ≥ 0
0 otherwise

가설

$$H(X) = \frac{1}{1 + e^{-WX}}$$

오류 함수

Prediction by computer

Correct answer

$$cost = \frac{1}{m}\sum_{i=1}^{m}(H(x_i) - y_i)^2$$

"로지스틱 리그레션에도 동작할까?"

오류 함수

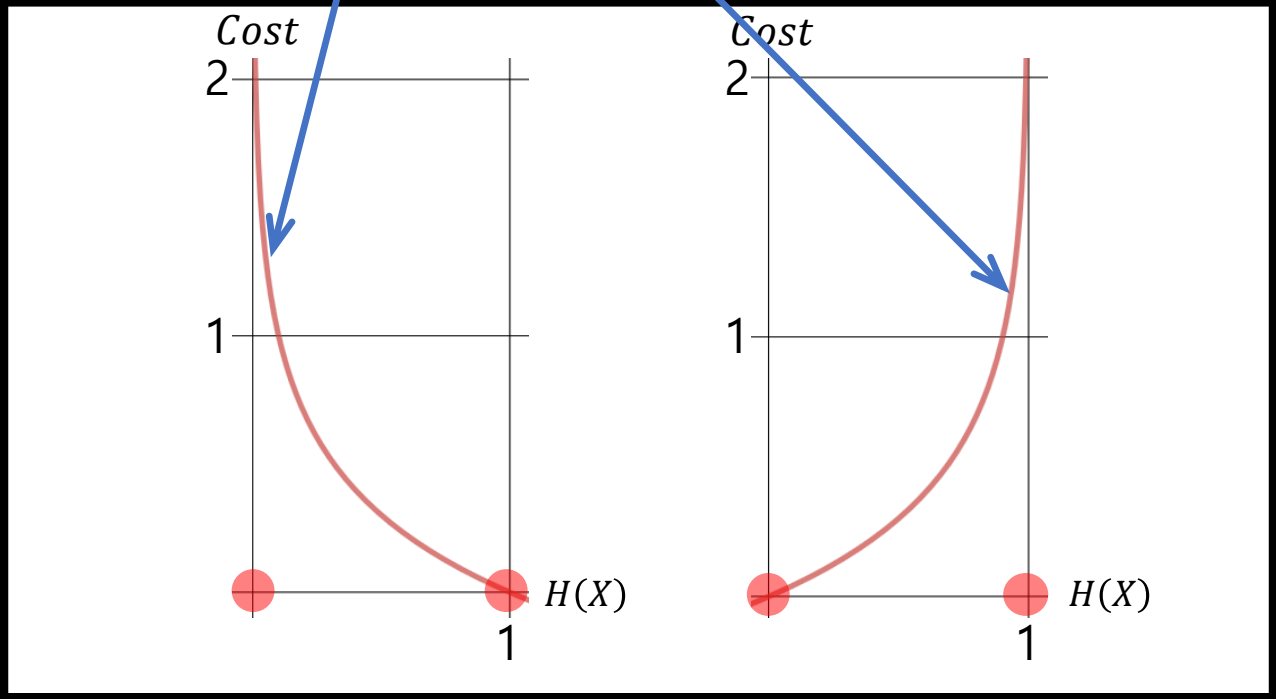
- 로지스틱 리그레션의 오류 함수로 MSE를 사용할 경우 어떤 문제가 발생할까?

오류 함수

Prediction by computer

Correct answer

$$cost = \begin{cases} -\log(H(X)) & : y = 1 \\ -\log(1 - H(X)) & : y = 0 \end{cases}$$



오류 함수

$$cost = \begin{cases} -\log\big(H(X)\big) & : y = 1 \\ -\log(1 - H(X)) & : y = 0 \end{cases}$$

$$cost = -y\log\big(H(X)\big) - (1 - y)\log(1 - H(X))$$

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

(실습) 11.py

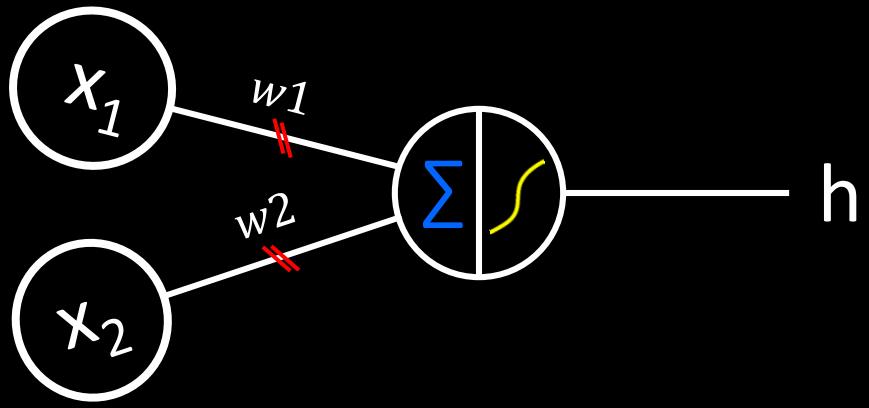
음수는 0으로 양수는 1로 분류

```
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)
```

(실습) 12.py

바이어스를 갖는 뉴런

신경 세포 (2 입력)



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

신경 세포 (2 입력)

- 결정 경계는?

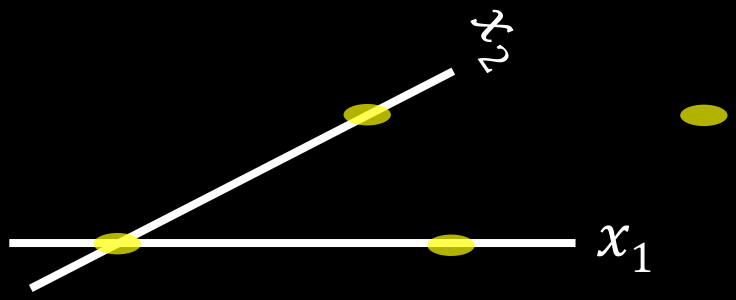
$w_1 x_1 + w_2 x_2 = 0$

$x_1 + x_2 = 0$

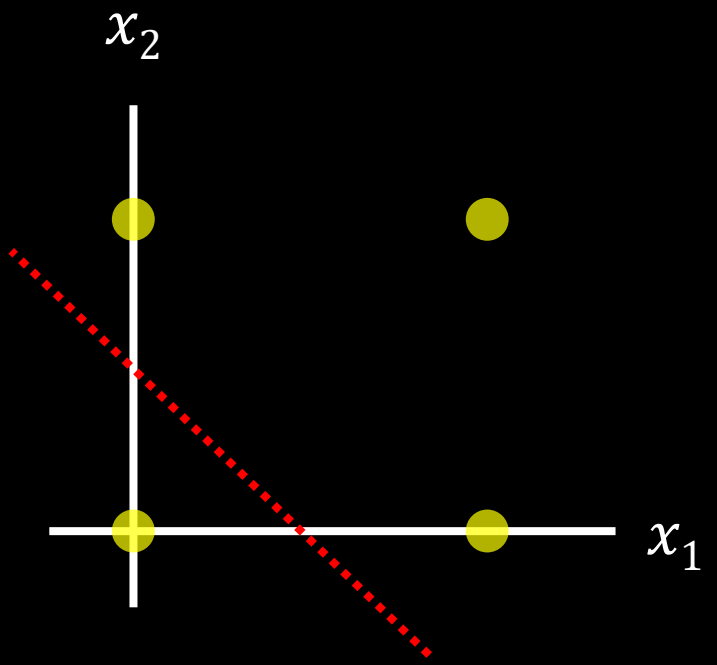
$w_1 x_1 + w_2 x_2 + b = 0$

$x_1 + x_2 - 0.5 = 0$

신경 세포 (2 입력)

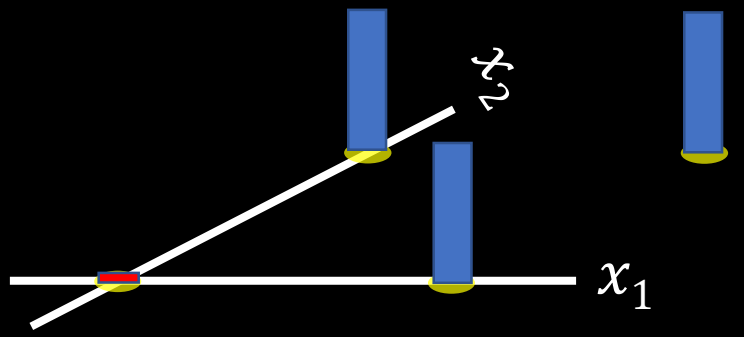


{옆에서 본 모습}

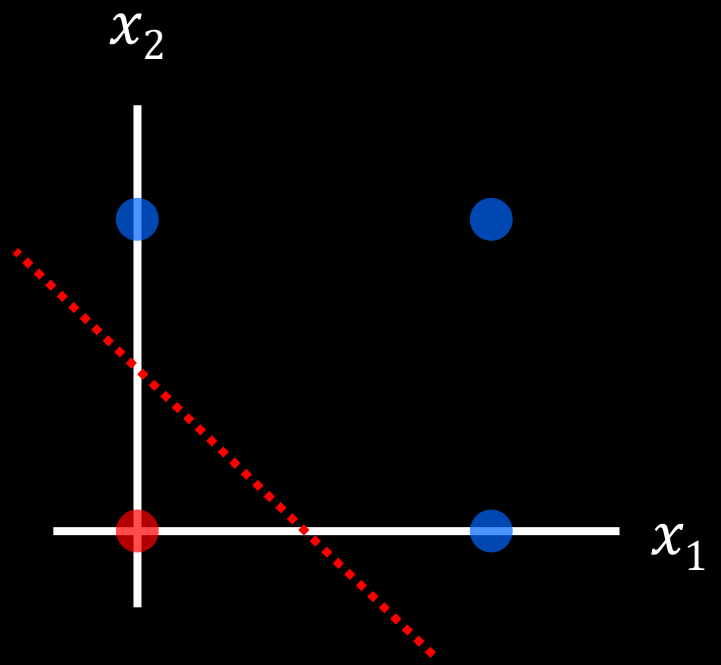


{위에서 본 모습}

신경 세포 (2 입력)

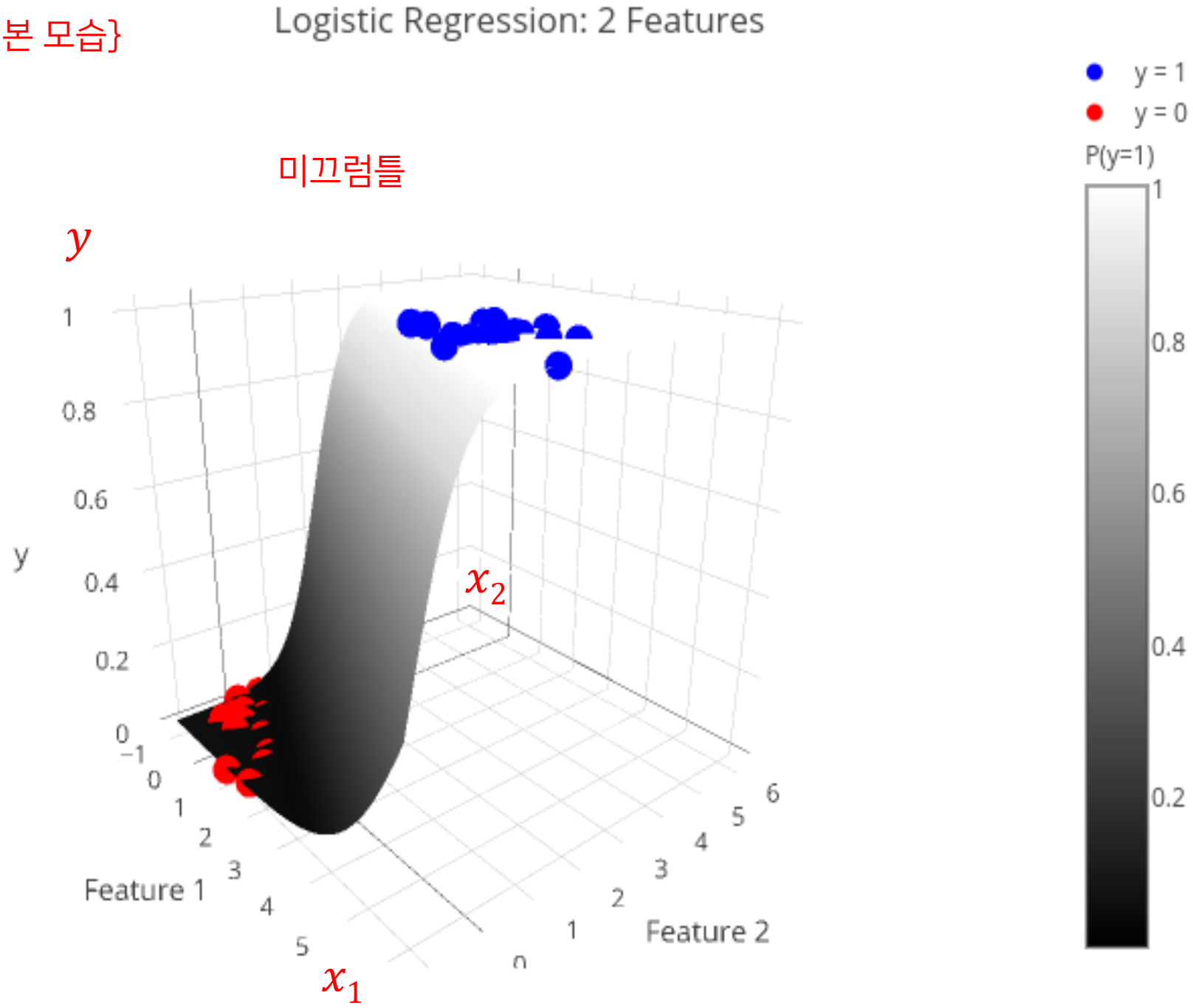


{옆에서 본 모습}



{위에서 본 모습}

{옆에서 본 모습}



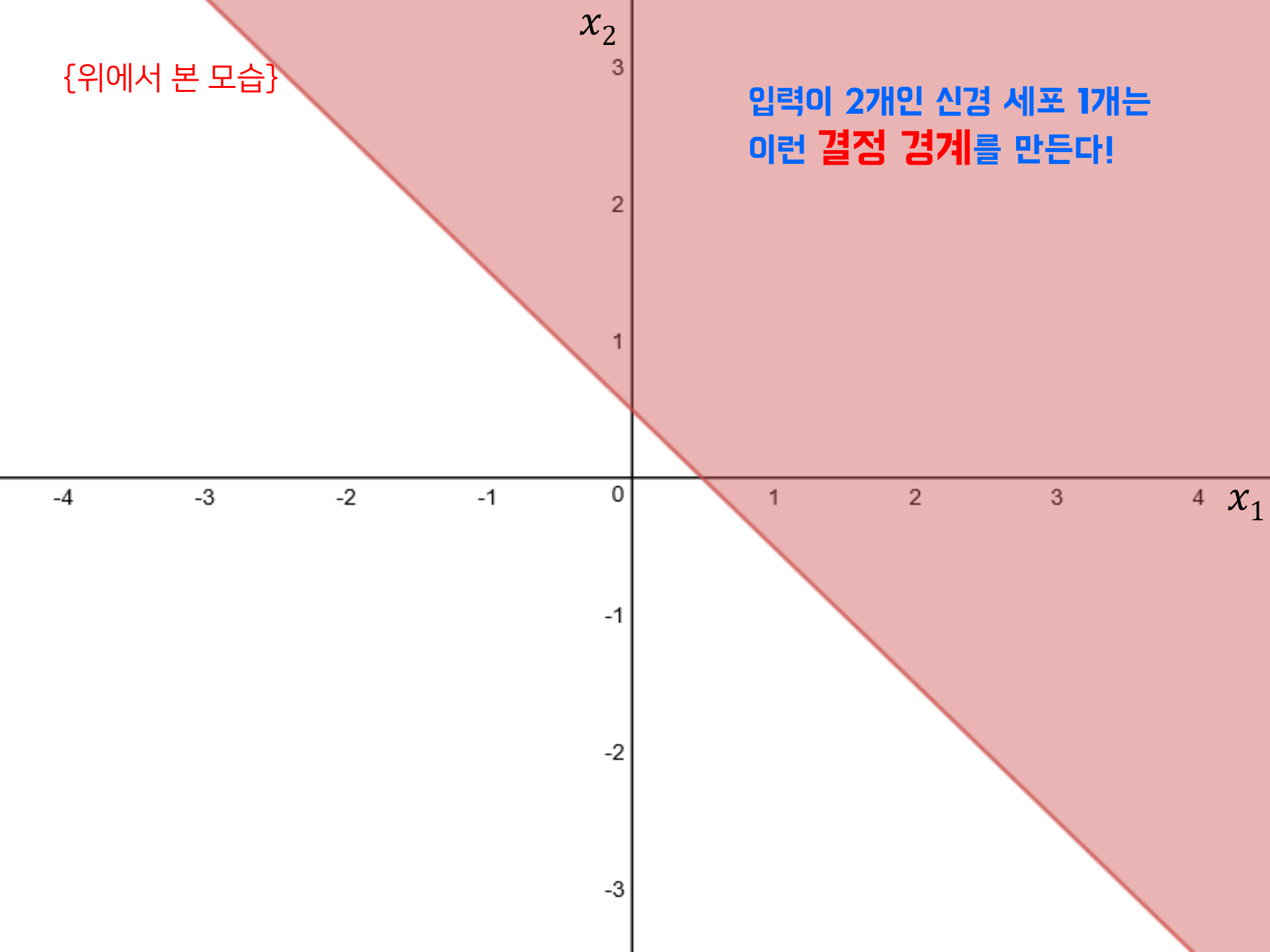
미끄럼틀

y

x_2

x_1

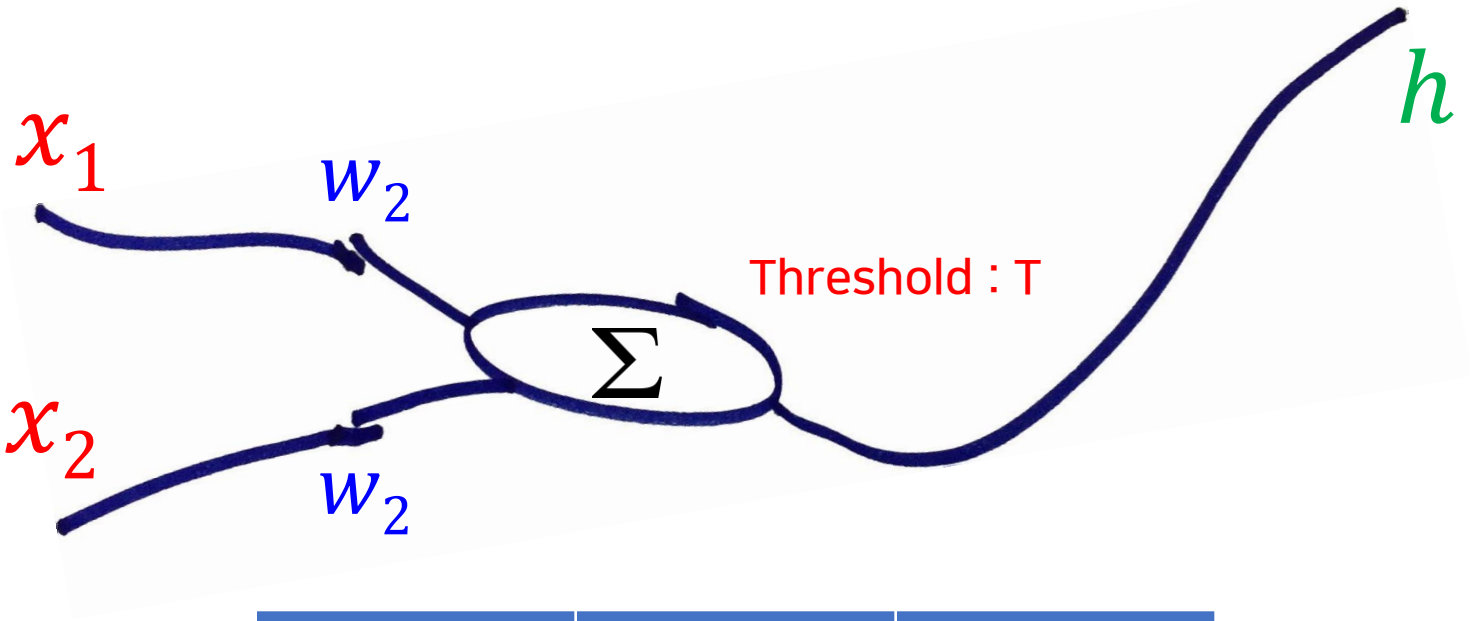
{위에서 본 모습}



입력이 2개인 신경 세포 1개는
이런 **결정 경계**를 만든다!

(실습) 13.py

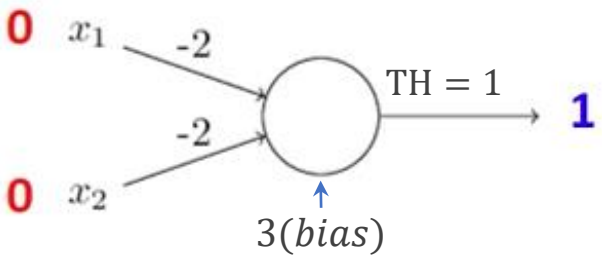
- 입력 두개(x1, x2)를 갖는 뉴런을 이용하여 OR 게이트를 구현함
- 한 개의 결정 경계
- (0, 0) -> 0, 그외 -> 1



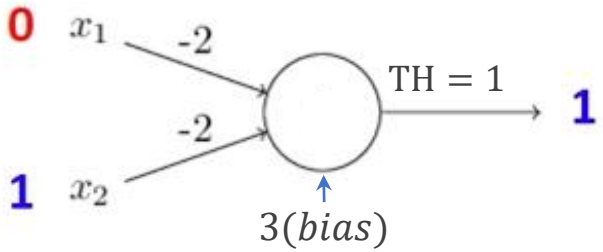
x_1	x_2	$AND(h)$
0	0	0
0	1	0
1	0	0
1	1	1

NAND

- NAND gates are functionally complete.
- We can build any logical function out of them.



(-2)*0 + (-2)*0+3=3

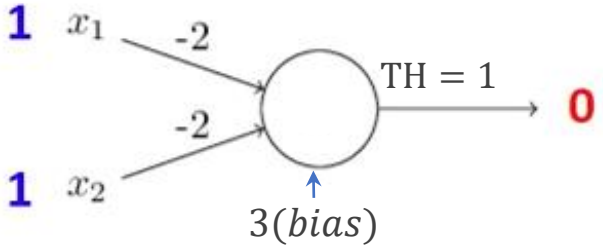


(-2)*0 + (-2)*1+3=1

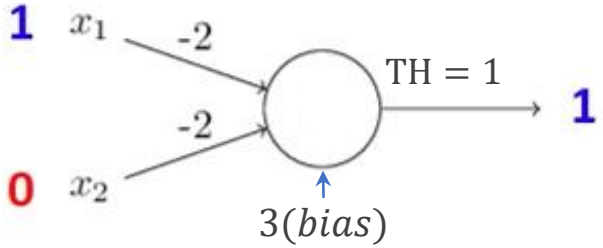
NAND

Truth Table

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



(-2)*1 + (-2)*1+3=-1



(-2)*1 + (-2)*0+3=1

요 약

- 로지스틱 리그레션과 분류 (classification)
- 로지스틱 리그레션을 위한 cost 함수
- 한 개의 뉴런이 만들어 내는 결정 경계
- 텐서 플로우를 이용한 ML 프로그래밍