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

Supervised Learning

Logistic Regression

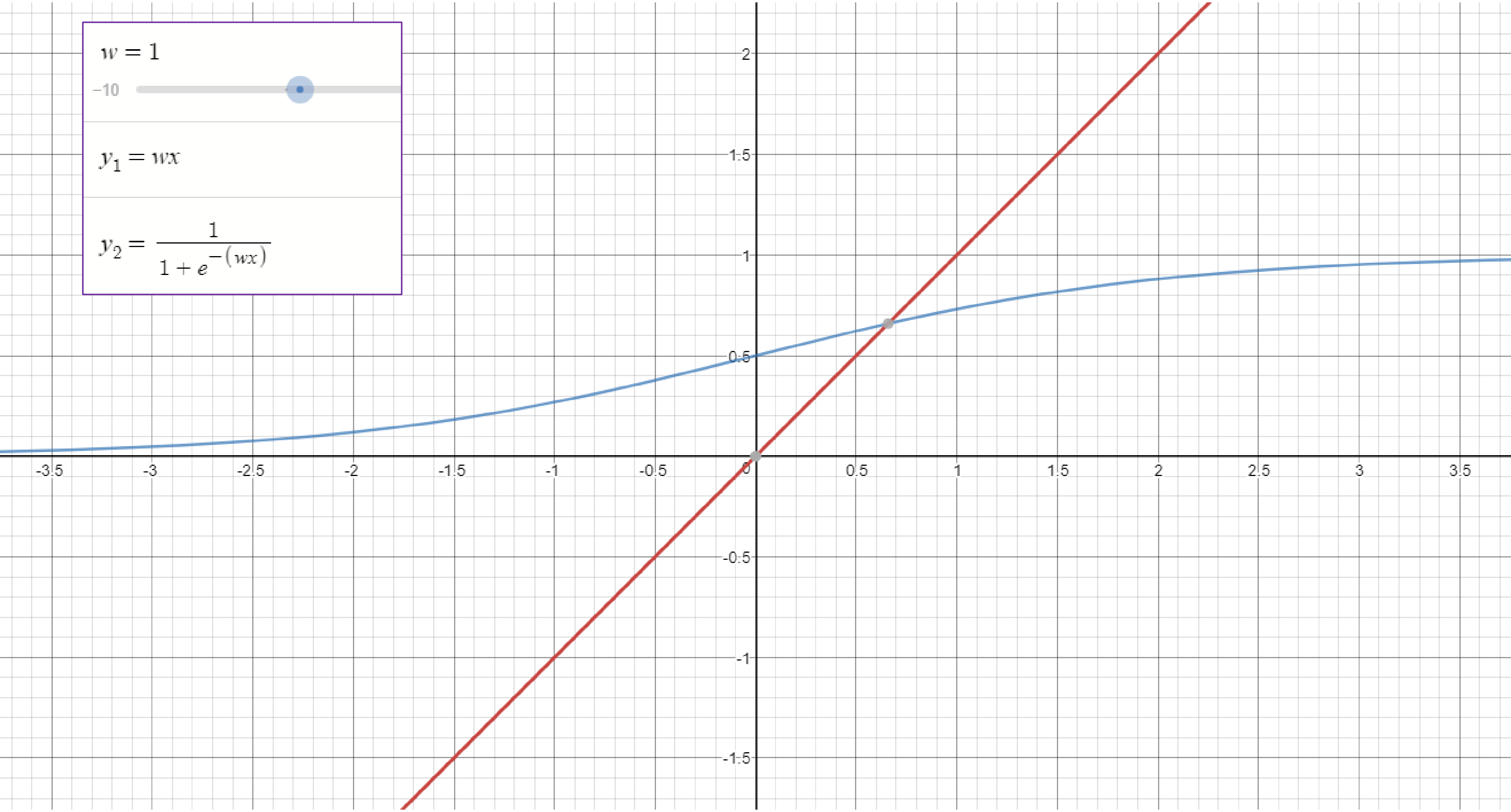
The shape of regression is not linear but logistic form.

www.desmos.com

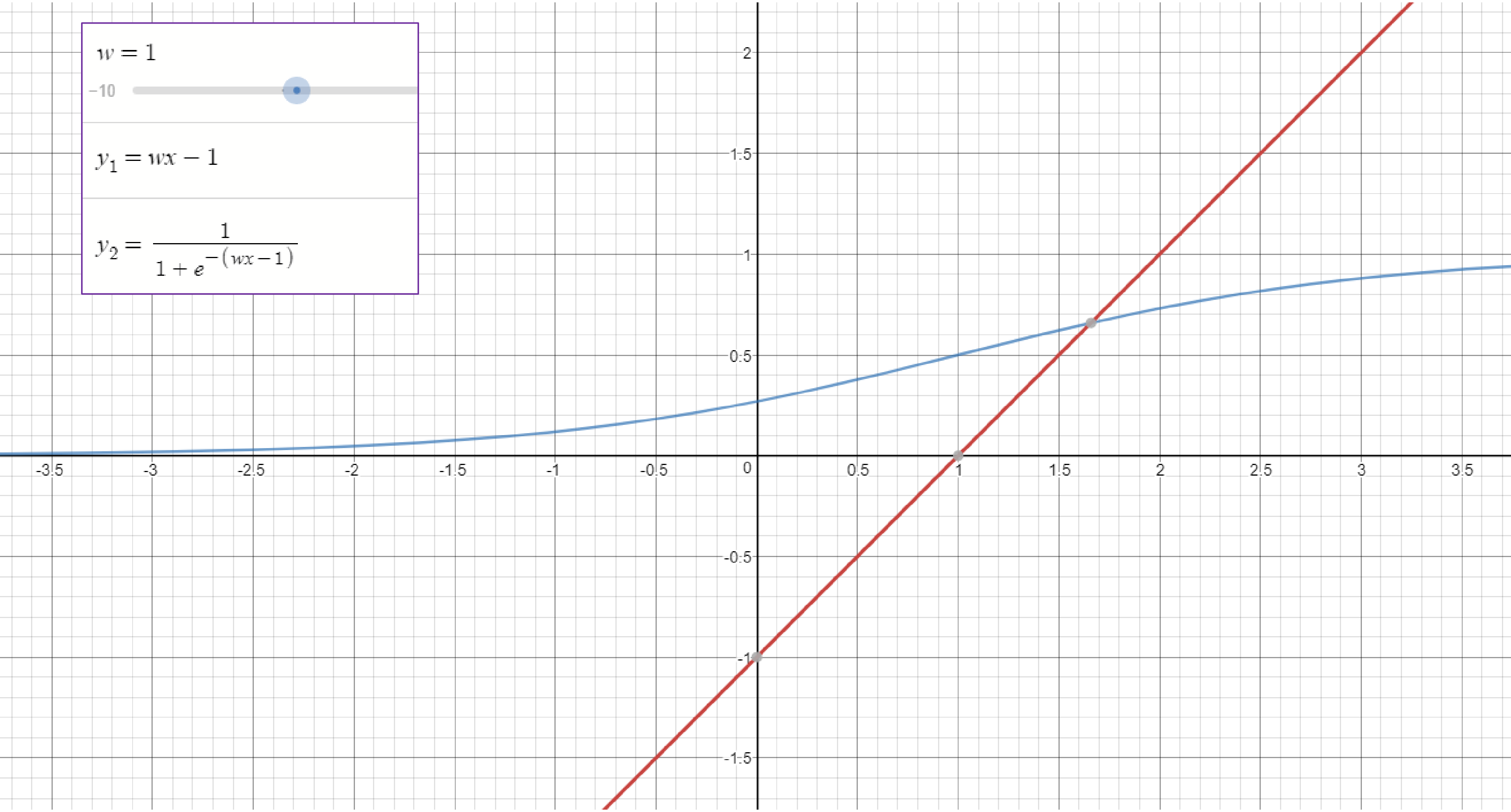
0, 1을 결정(decision)하는
경계(boundary)

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

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



$w = 1$
-10
$y_1 = wx$
$y_2 = \frac{1}{1 + e^{-(wx)}}$


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

결정 경계

$$y = \frac{1}{1 + e^{-(\underbrace{wx}_{=0})}}$$

결정 경계

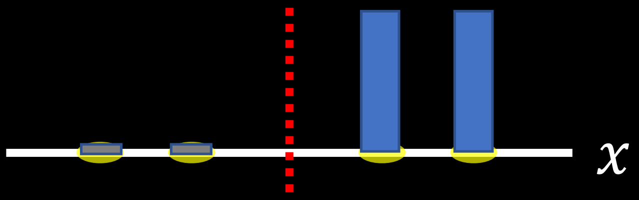
$or,$

$$y = \text{sigmoid}\,(wx)$$

결정 경계

$$wx = 0$$

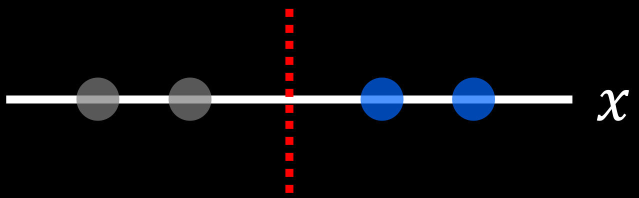
$$x = 0$$



{옆에서 본 모습}

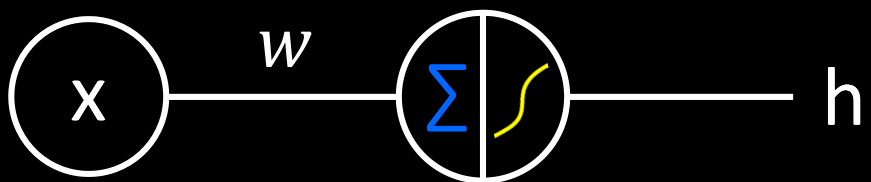
$$wx + b = 0$$

$$x + 1 = 0$$



{위에서 본 모습}

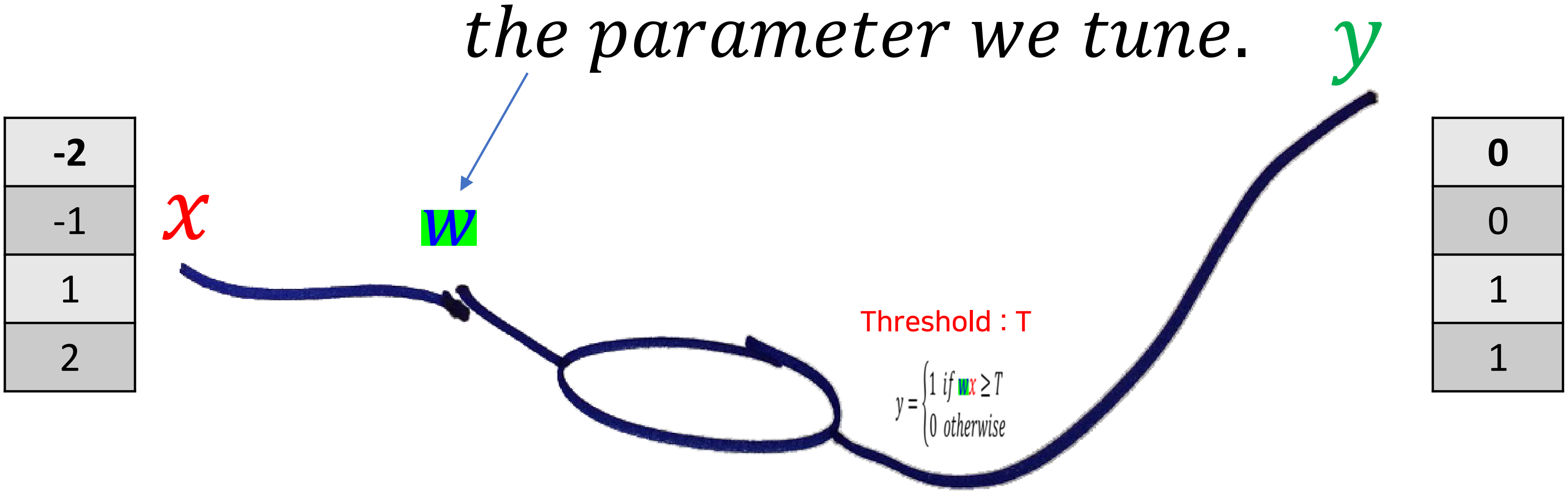
신경 세포 기능



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

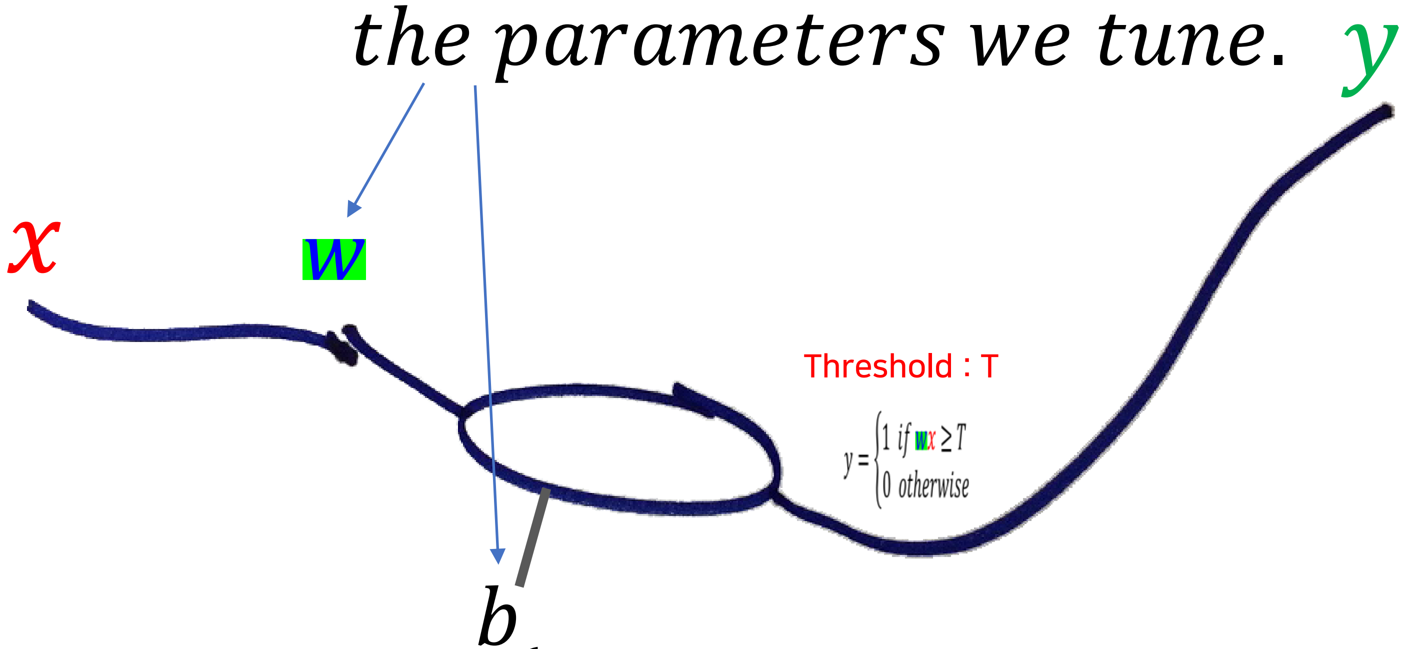
Classification

- 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



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

-2
-1
1
2



0
0
1
1

$$y = \begin{cases} 1 \ \ if \ wx + b \geq T \\ 0 \ \ otherwise \end{cases}$$

가설

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

오류 함수

Prediction by computer

Correct answer

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

“로지스틱 리그레션에도 동작할까?”

오류 함수

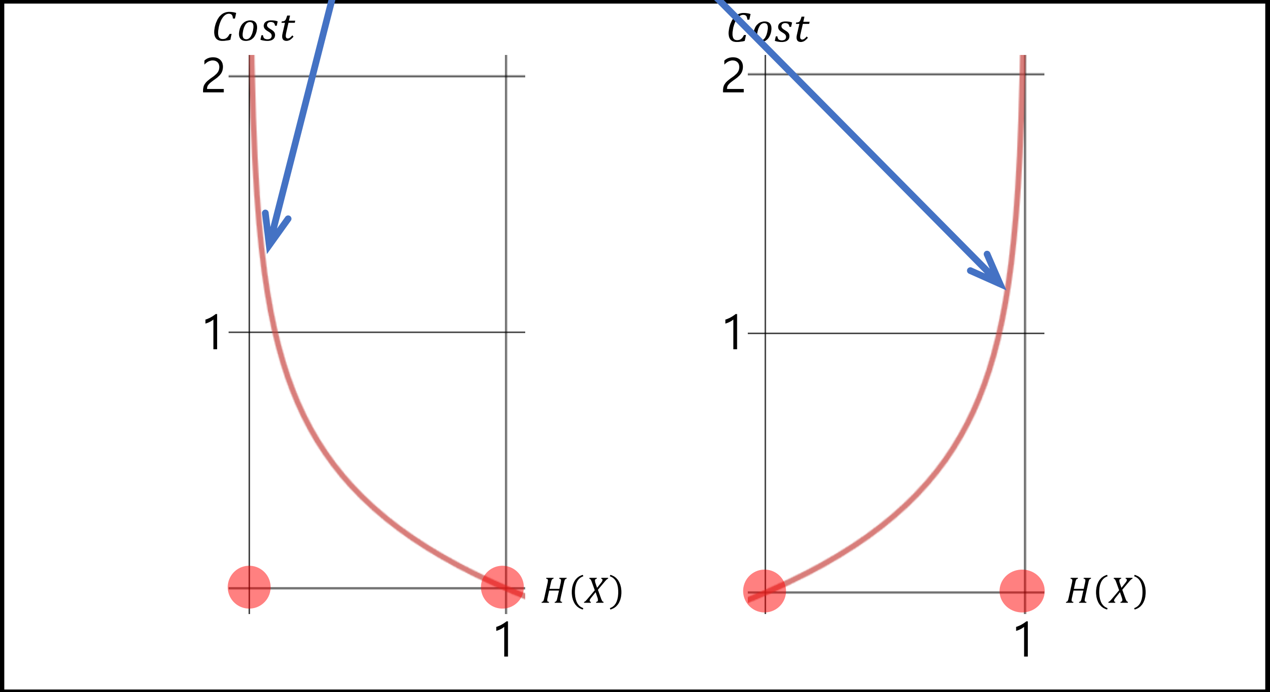
- 로지스틱 리그레션의 오류 함수로 MSE를 사용할 경우 어떤 문제가 발생할까?

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

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

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

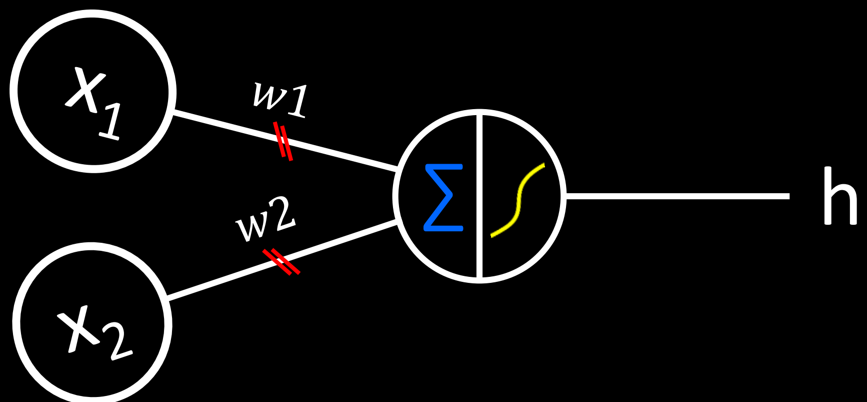
$$W = W - \alpha \frac{\partial}{\partial W} cost(W)$$

(실습) 11.py

음수는 0으로 양수는 1로 분류

```python
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)
```

신경 세포 (2 입력)



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

신경 세포 (2 입력)

- 결정 경계는?

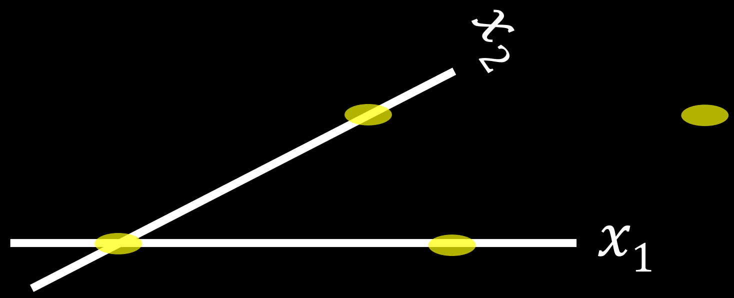
$$w_1 x_1 + w_2 x_2 = 0$$

$$x_1 + x_2 = 0$$

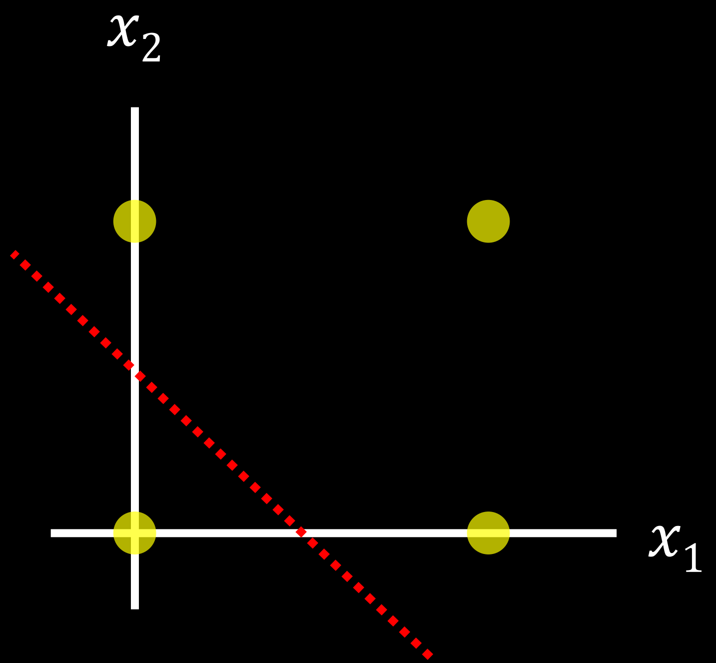
$$w_1 x_1 + w_2 x_2 + b = 0$$

$$x_1 + x_2 + 1 = 0$$

신경 세포 (2 입력)

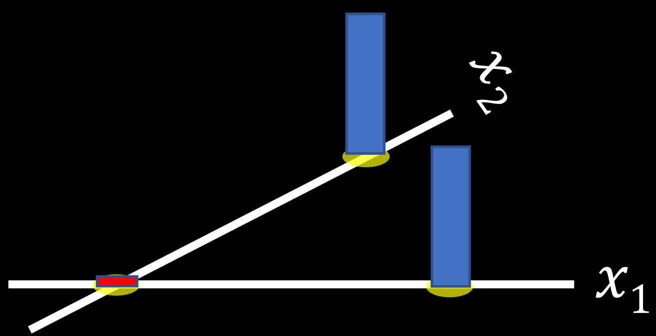


{옆에서 본 모습}

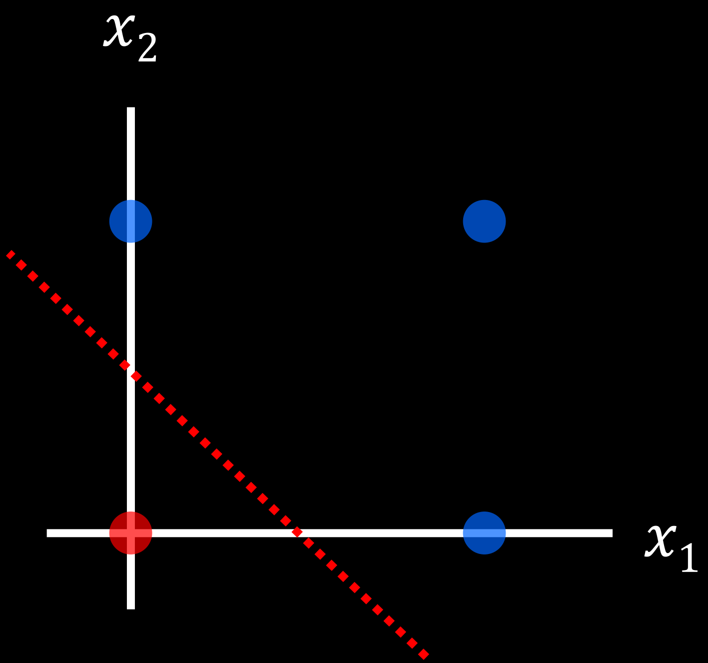


{위에서 본 모습}

신경 세포 (2 입력)

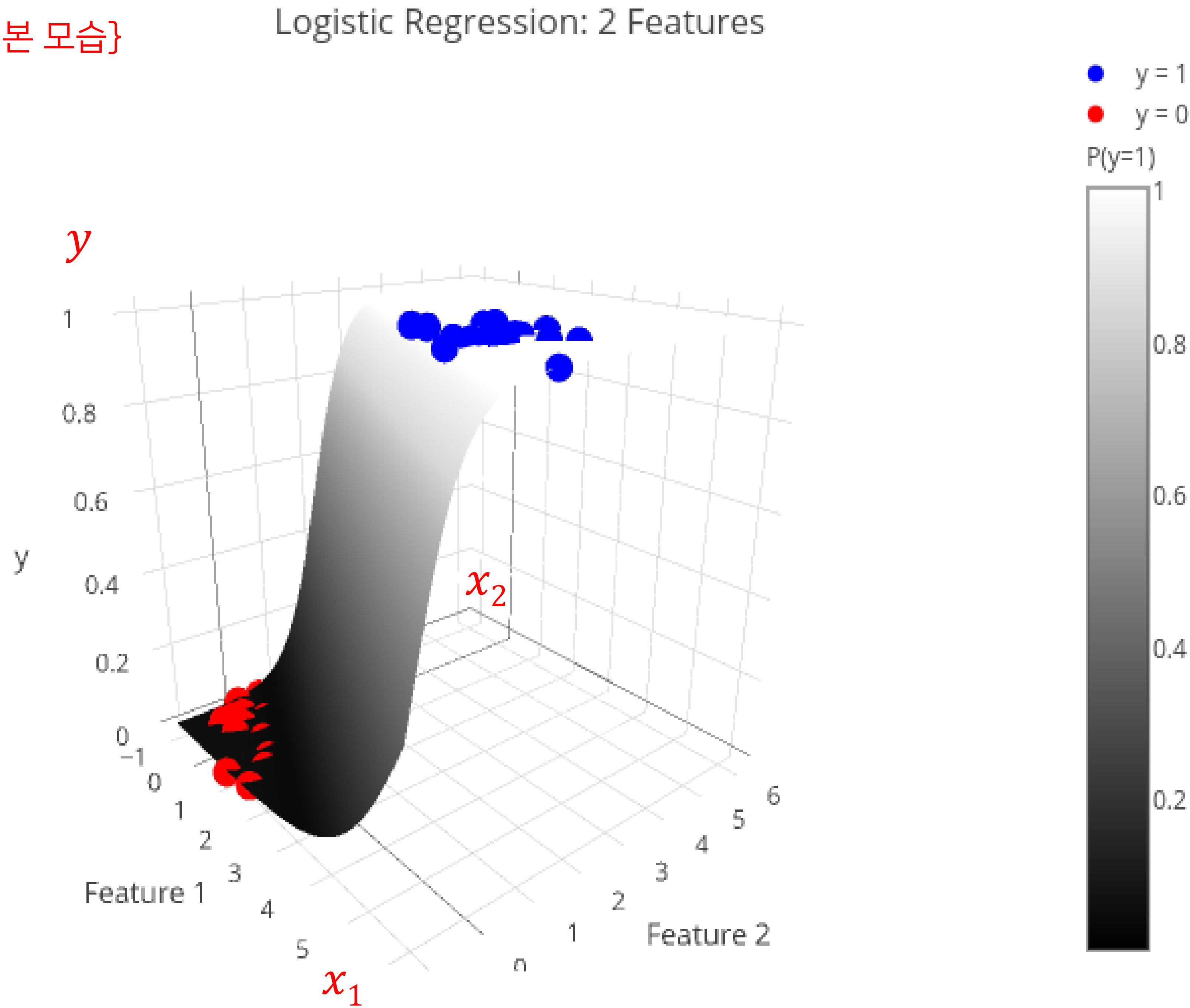


{옆에서 본 모습}

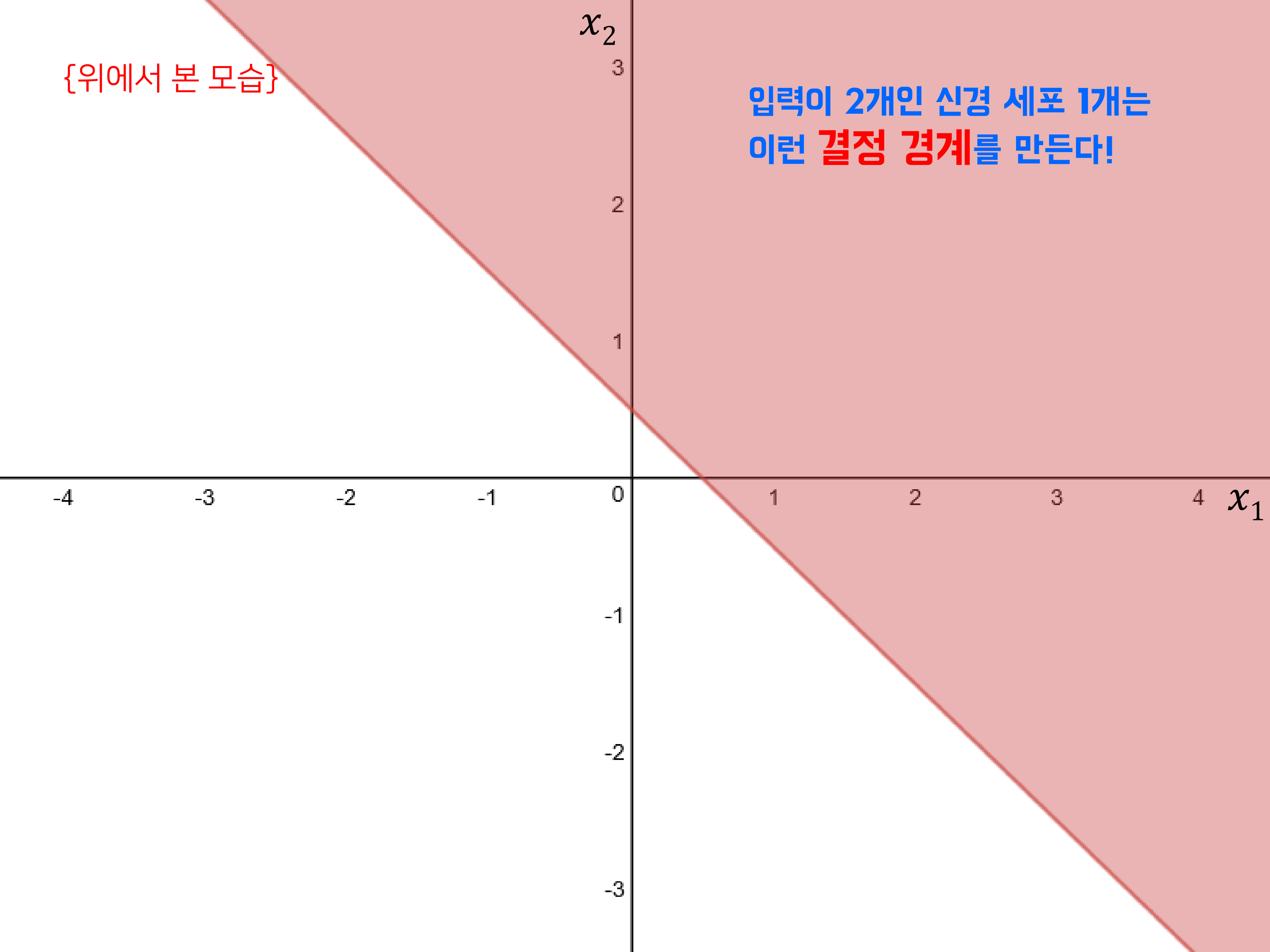


{위에서 본 모습}

{옆에서 본 모습}

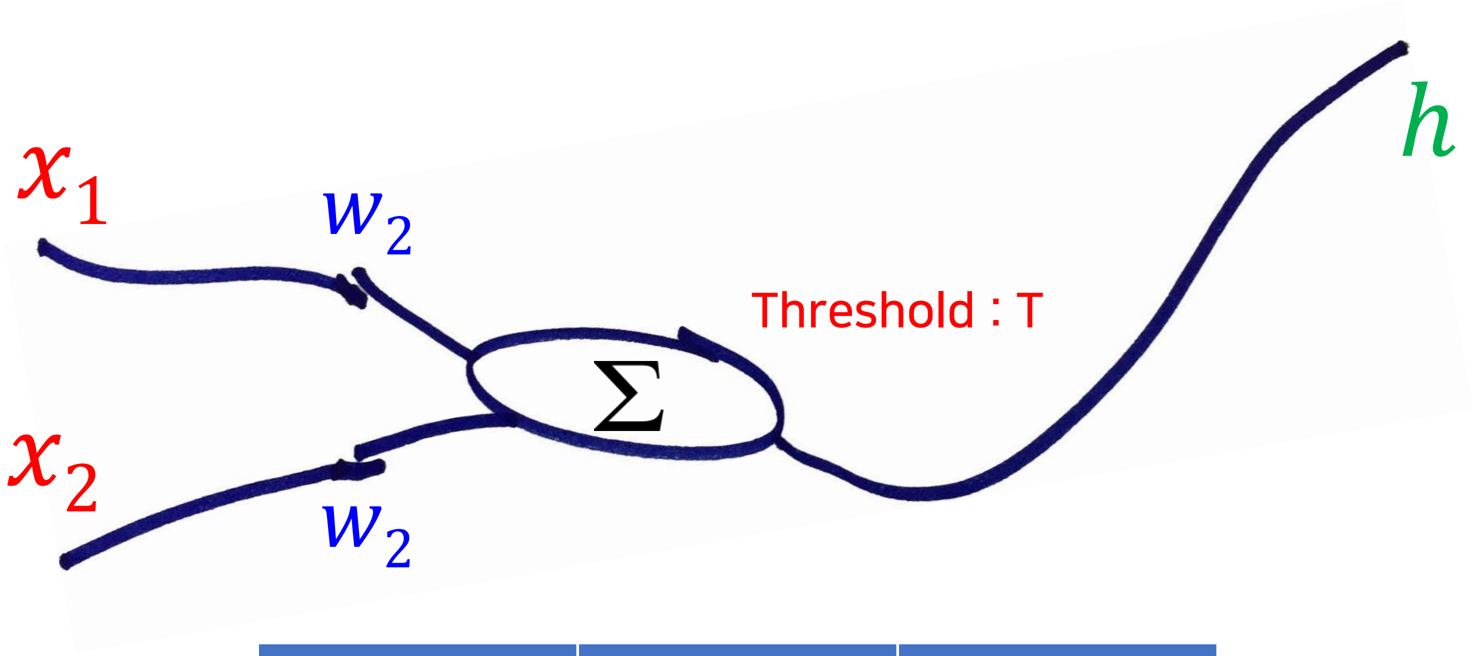


Logistic Regression: 2 Features
y = 1
y = 0
P(y=1)
1
0.8
0.6
0.4
0.2
y
0
−1
1
2
3
4
5
6
Feature 1
Feature 2
y
x_1
x_2


{위에서 본 모습}
입력이 2개인 신경 세포 1개는
이런 결정 경계를 만든다!
x_1
x_2
-4
-3
-2
-1
0
1
2
3
4
-1
-2
-3
1
2
3

(실습) 13.py

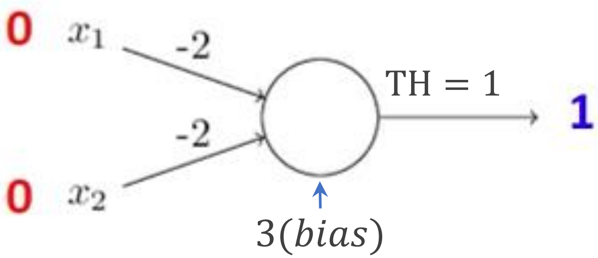
- OR



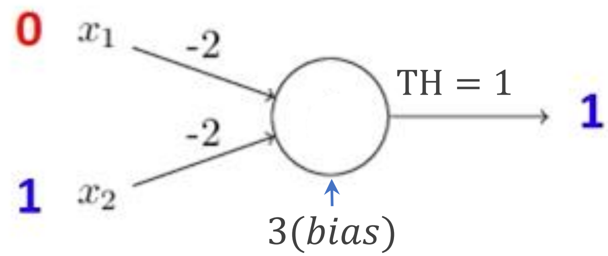
x_1	x_2	$AND(h)$
0	0	0
0	1	0
1	0	0
1	1	1

NAND

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



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

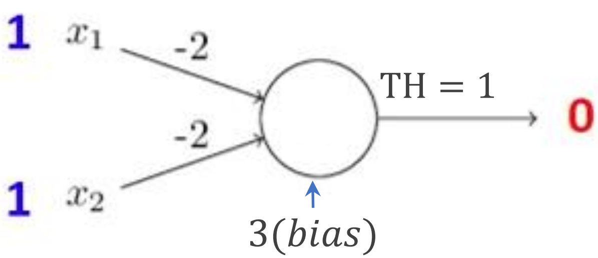


(-2)*0 + (-2)*1+3=1

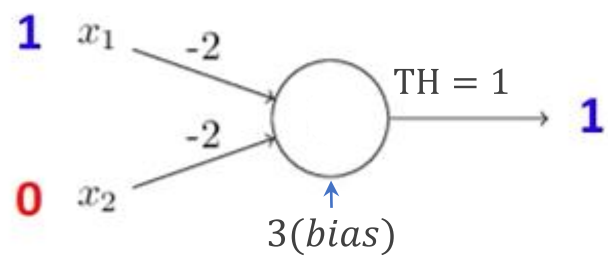
NAND

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