Logistic regression to Feed-forward network

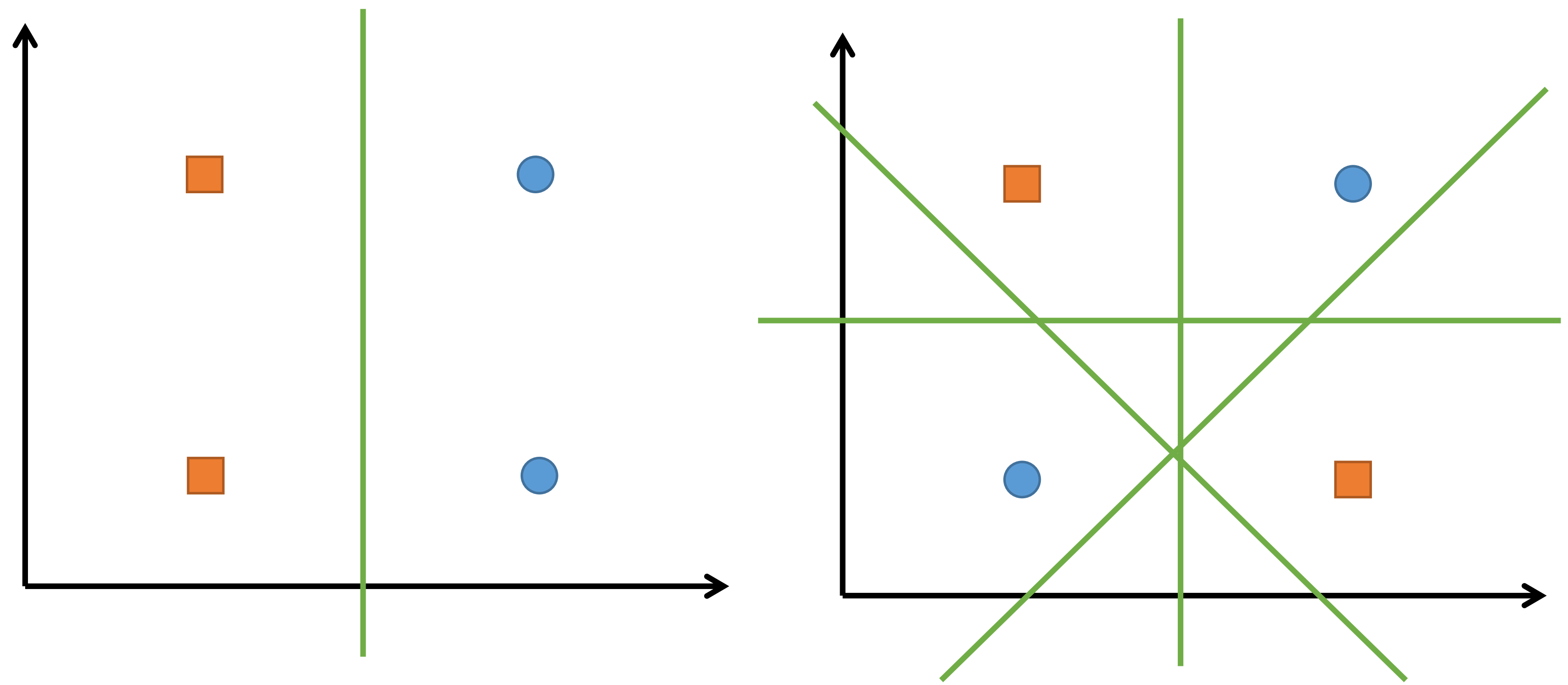
Taehoon Ko (thoon.koh@gmail.com)

목표

- Logistic regression의 분류 경계를 이해한다.
- Feed-forward network가 어떠한 원리로 비선형 분류 경계를 만들어내는지 이해한다.

Limitation of linear models

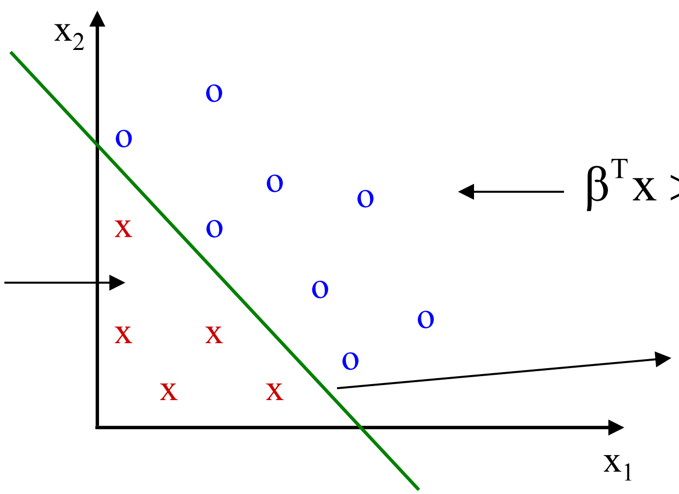
- 단 하나의 직선을 그어서 파란색 원과 붉은색 사각형을 분류할 수 있는가?



Logistic regression (revisited)

- Logistic regression은 학습 알고리즘을 통해 2-class points를 분류하는 초평면(hyper-plane)을 찾는 것
- 선형적인 의사결정경계(linear decision boundary)만 생성 가능

$\beta^T x < 0$



$\beta^T x > 0$

Hyperplane: $\beta^T x = 0$ where $\beta = \begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix}$

Classifier

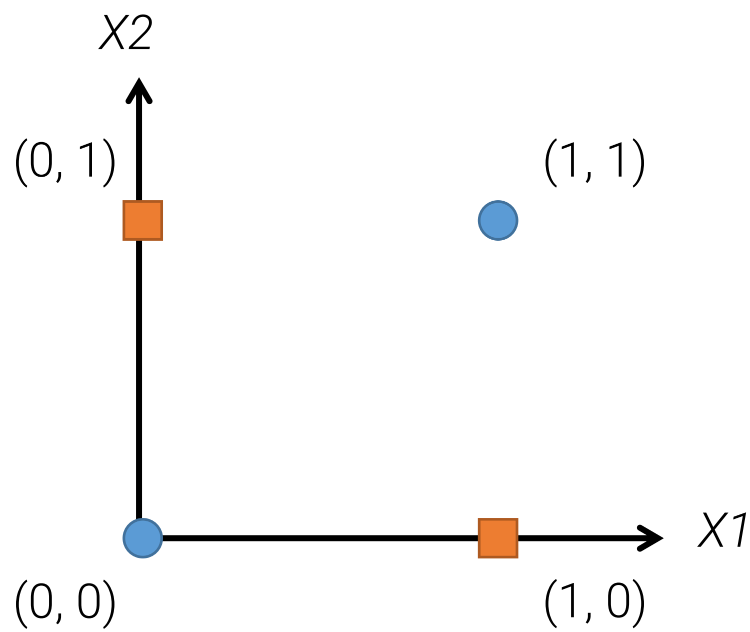
$$y = \frac{1}{\left(1 + \exp(-\beta^T x)\right)} \qquad \begin{pmatrix} y \to 1 & \textit{if} & \beta^T x \to & \infty \\ y = \frac{1}{2} & \textit{if} & \beta^T x = & 0 \\ y \to 0 & \textit{if} & \beta^T x \to & -\infty \end{pmatrix}$$

Exclusive OR (XOR) problem

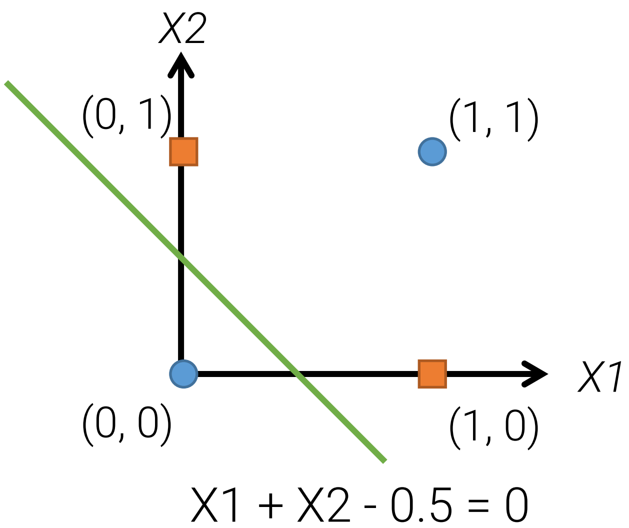
- XOR

Exclusive-OR gate

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0



XOR classification by logistic regression model



$$y = \Pr(class = 1)$$

$$= \frac{1}{1 + e^{-(X_1 + X_2 - 0.5)}}$$

$$= \sigma(X_1 + X_2 - 0.5)$$

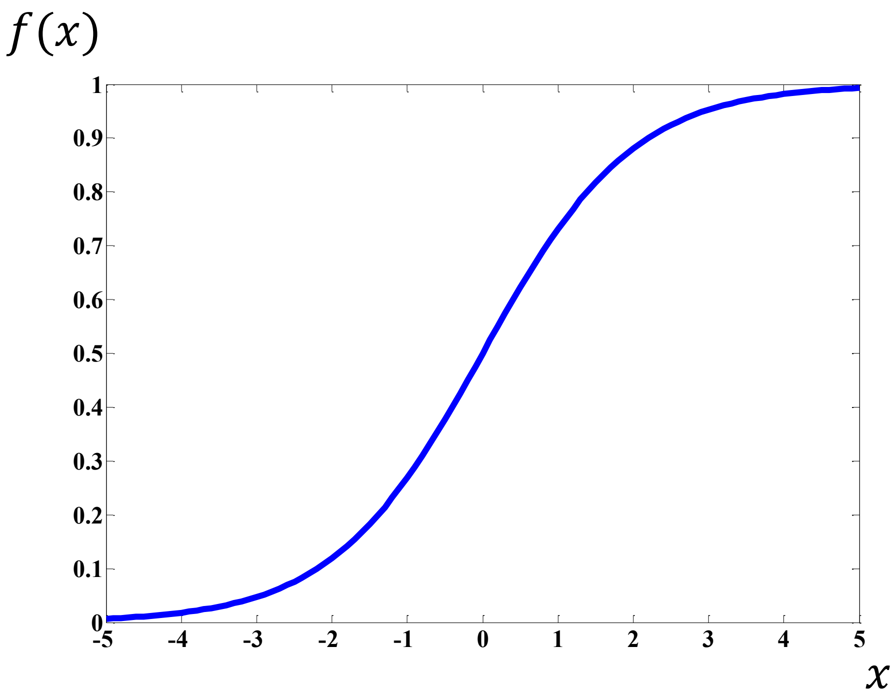
$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Logistic function
(a.k.a. softmax func., sigmoid function)

(Revisited) Logistic function

- Logistic function
  - $-\infty < x < \infty$ 일 때, logistic function $f(x)$ 은 다음과 같이 정의됨.

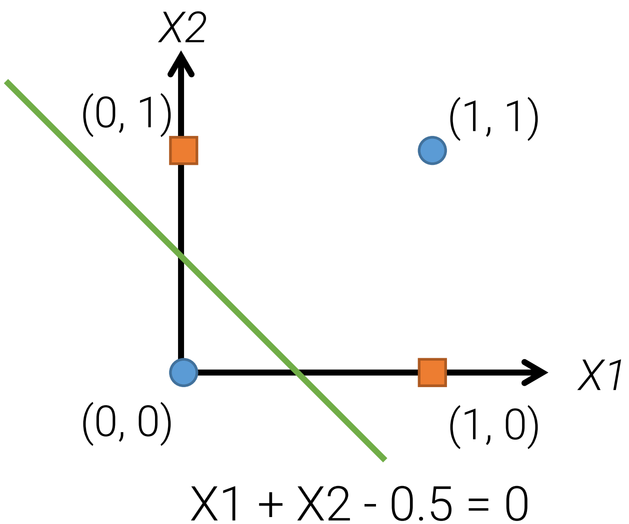
$$f(x) = \frac{1}{1+e^{-x}}$$



- $0 < f(x) < 1$
- $x = 0$일 때, 함수값은 $f(0) = 0.5$
- 위의 점을 기준으로 함수가 대칭 (symmetric)

$f(x)$는 특정사건이 일어날 확률과 매우 유사하므로, 확률을 logistic function으로 모델링하는 것이 가능함.

XOR classification by logistic regression model

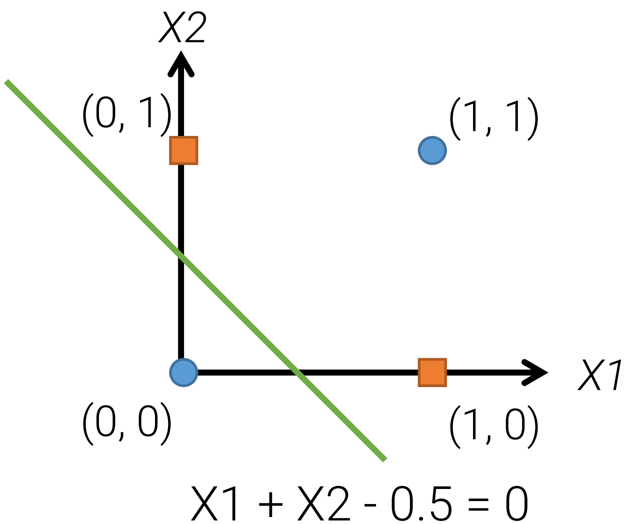


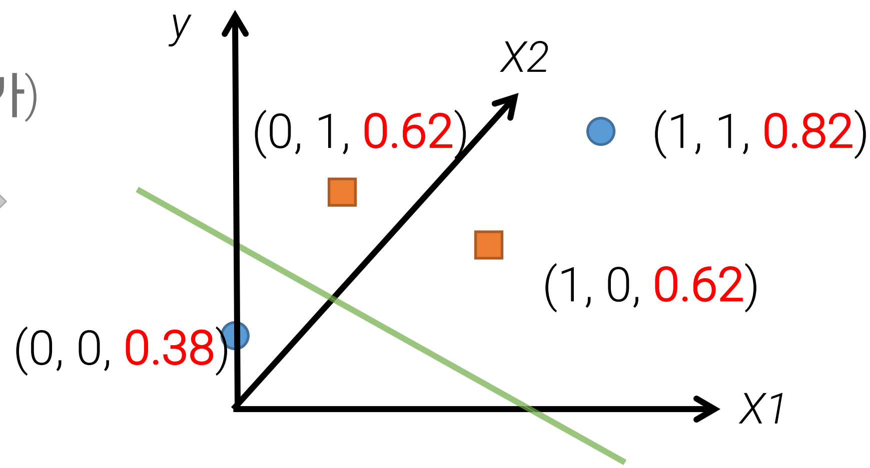
$$y = \frac{1}{1 + e^{-(X_1 + X_2 - 0.5)}}$$

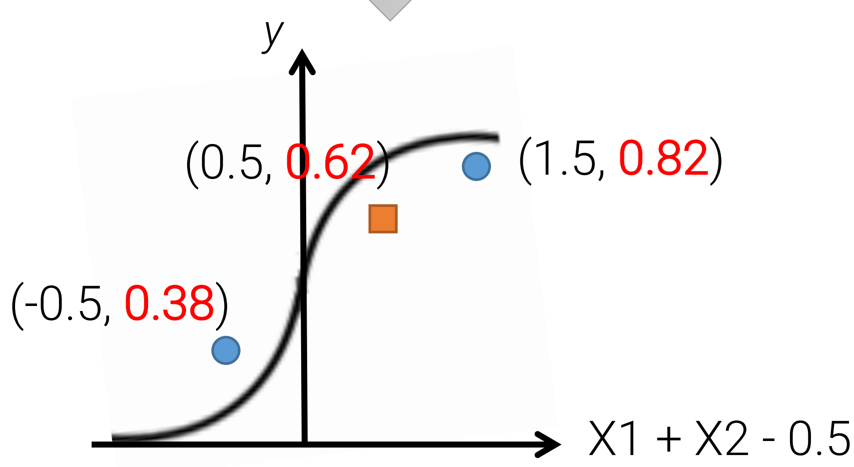
$$= \sigma(X_1 + X_2 - 0.5)$$

```
In [1]: import numpy as np

In [2]: def logistic(x):
            return 1 / (1 + np.exp(-x))

In [3]: logistic(0 + 0 -0.5)
Out[3]: 0.3775406687981454

In [4]: logistic(1 + 1 - 0.5)
Out[4]: 0.8175744761936437

In [5]: logistic(0 + 1 - 0.5)
Out[5]: 0.6224593312018546

In [6]: logistic(1 + 0 - 0.5)
Out[6]: 0.6224593312018546
```

XOR classification by logistic regression model







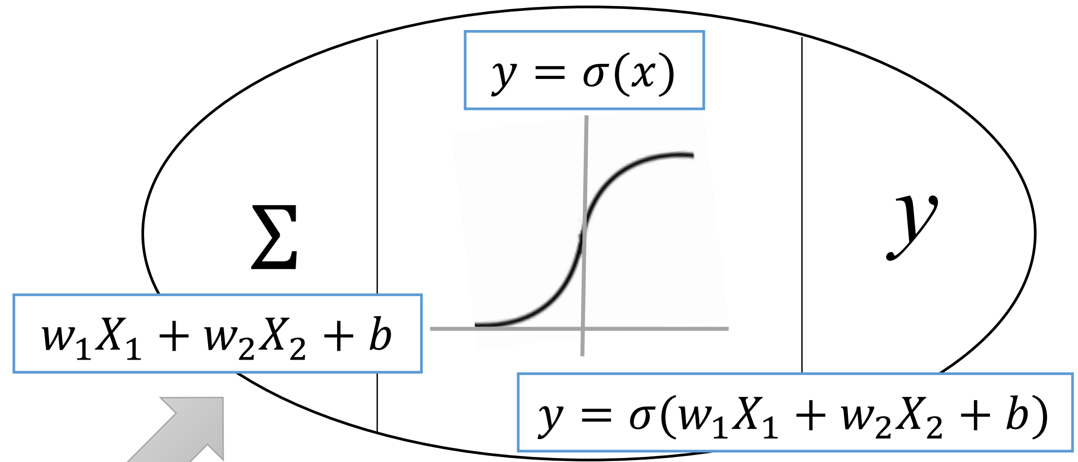
$$y = \frac{1}{1 + e^{-(X_1 + X_2 - 0.5)}}$$

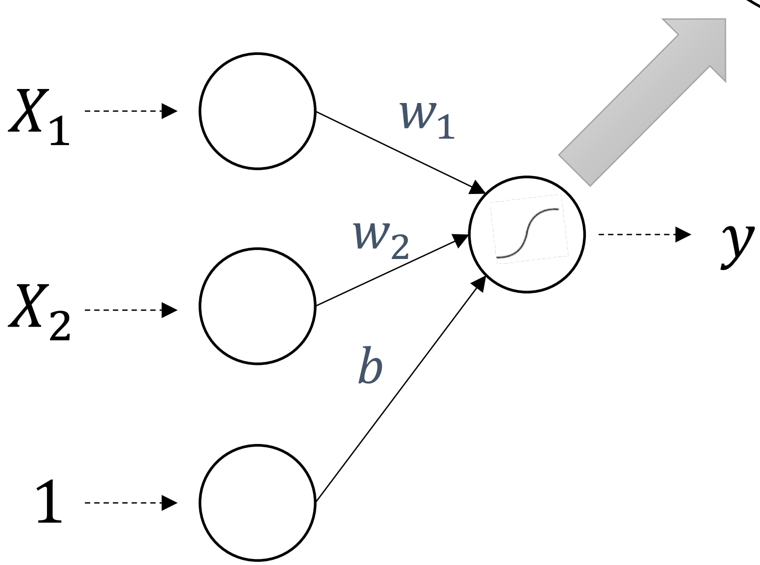
$$= \sigma(X_1 + X_2 - 0.5)$$

Logistic regression as a network

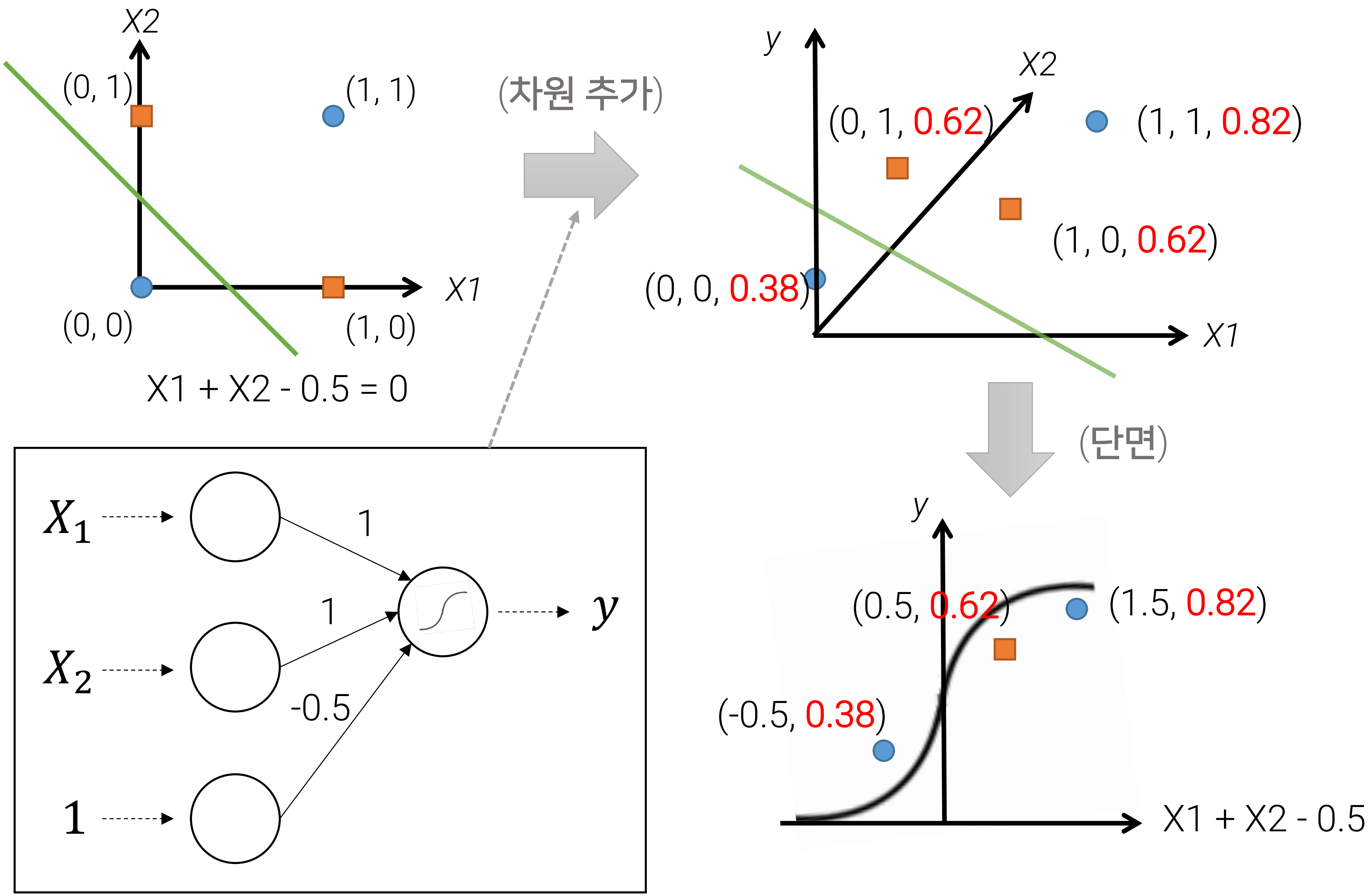
- Logistic regression을 network로 표현하면 다음과 같다.

$$y = \frac{1}{1 + e^{-(w_1X_1 + w_2X_2 + b)}}$$
$$= \sigma(w_1X_1 + w_2X_2 + b)$$

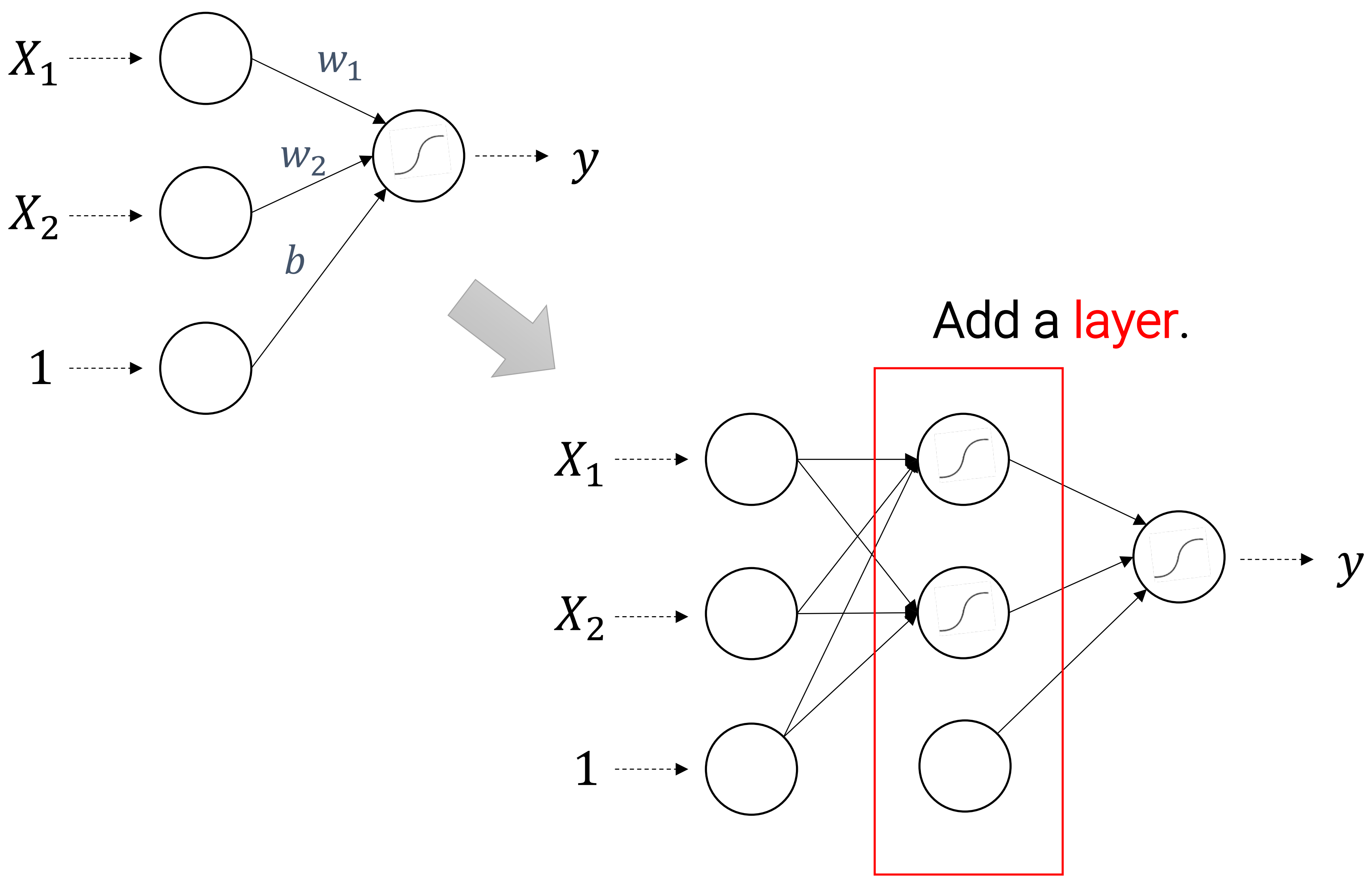




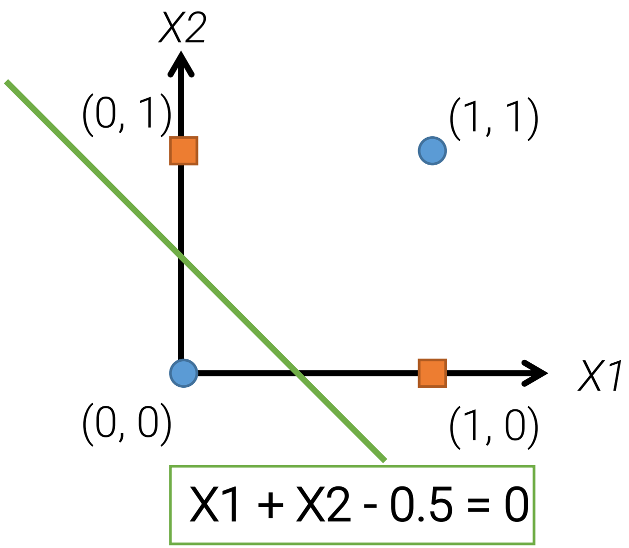
XOR classification by logistic regression model

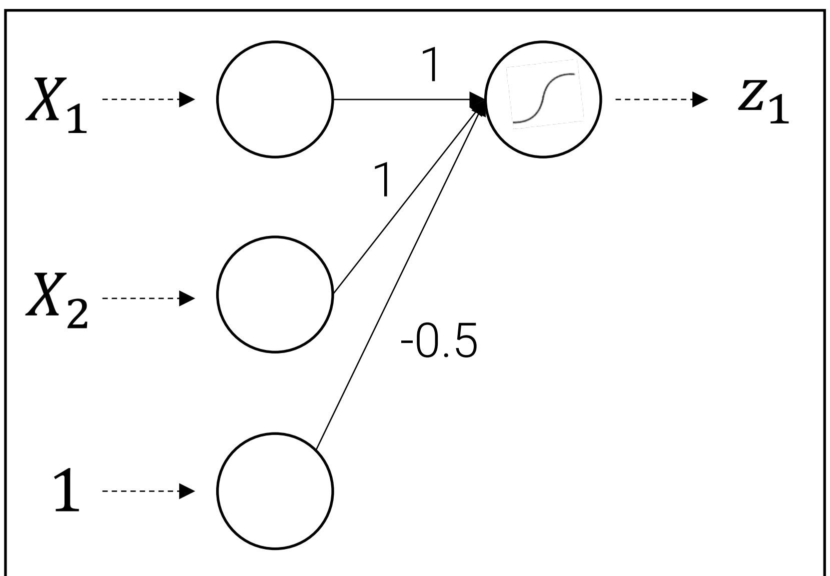


How about this model? ➔ Feed-forward network



XOR classification by feed-forward network model





In [1]:
```
import numpy as np
```

In [2]:
```
def logistic(x):
    return 1 / (1 + np.exp(-x))
```

In [3]:
```
logistic(0 + 0 -0.5)
```

Out [3]: 0.3775406687981454

In [4]:
```
logistic(1 + 1 - 0.5)
```

Out [4]: 0.8175744761936437

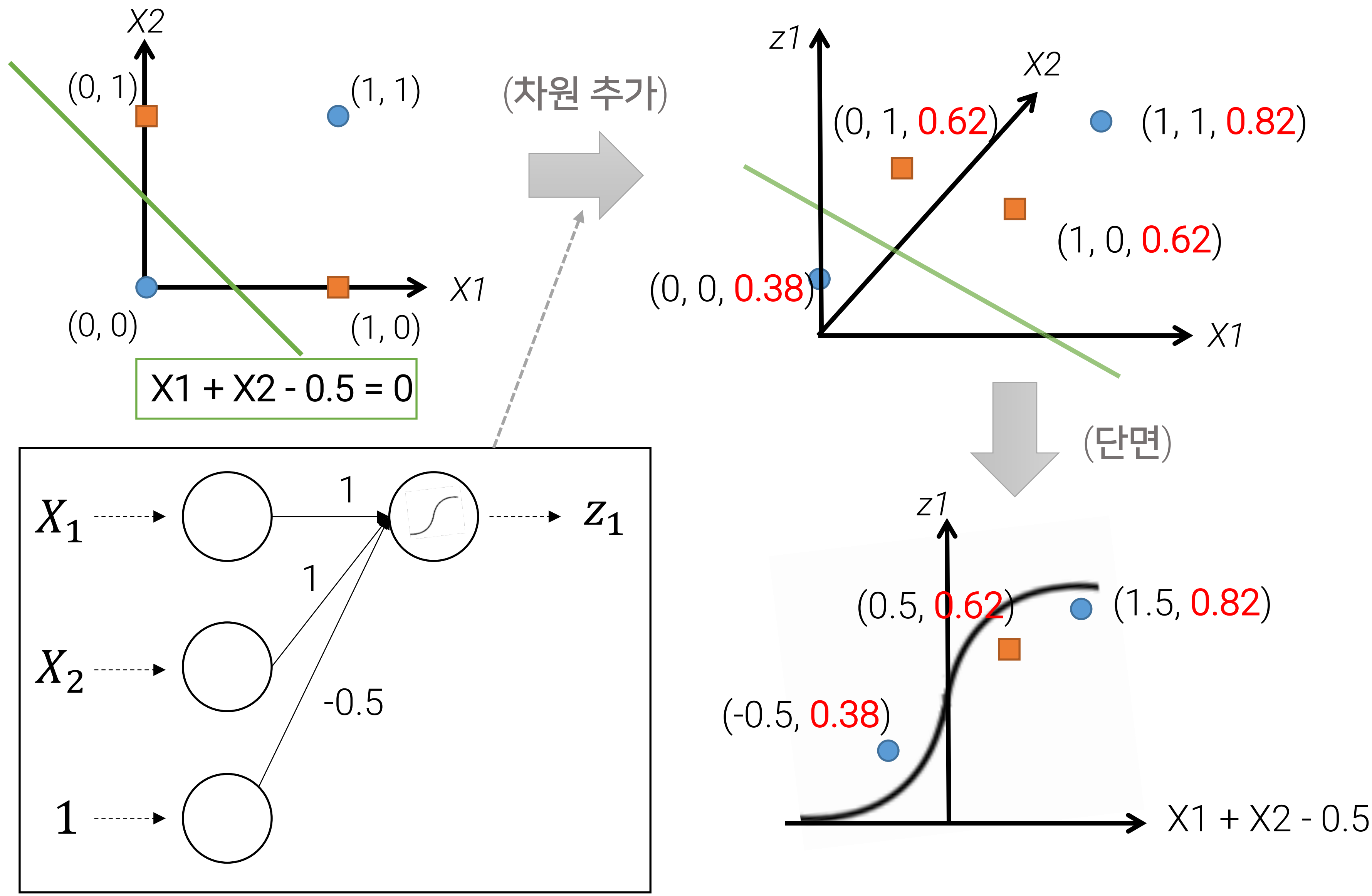
In [5]:
```
logistic(0 + 1 - 0.5)
```

Out [5]: 0.6224593312018546

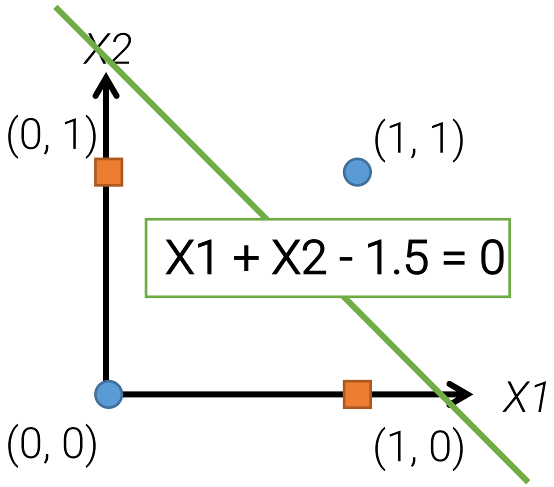
In [6]:
```
logistic(1 + 0 - 0.5)
```

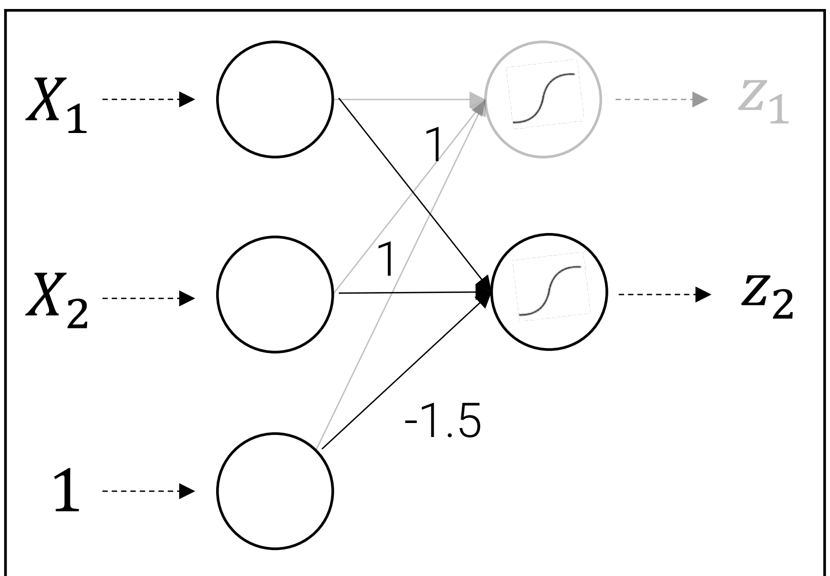
Out [6]: 0.6224593312018546

XOR classification by feed-forward network model



XOR classification by feed-forward network model





In [1]:
```python
import numpy as np
```

In [2]:
```python
def logistic(x):
    return 1 / (1 + np.exp(-x))
```

In [3]:
```python
logistic(0 + 0 - 1.5)
```

Out[3]: 0.18242552380635635

In [4]:
```python
logistic(1 + 1 - 1.5)
```

Out[4]: 0.6224593312018546

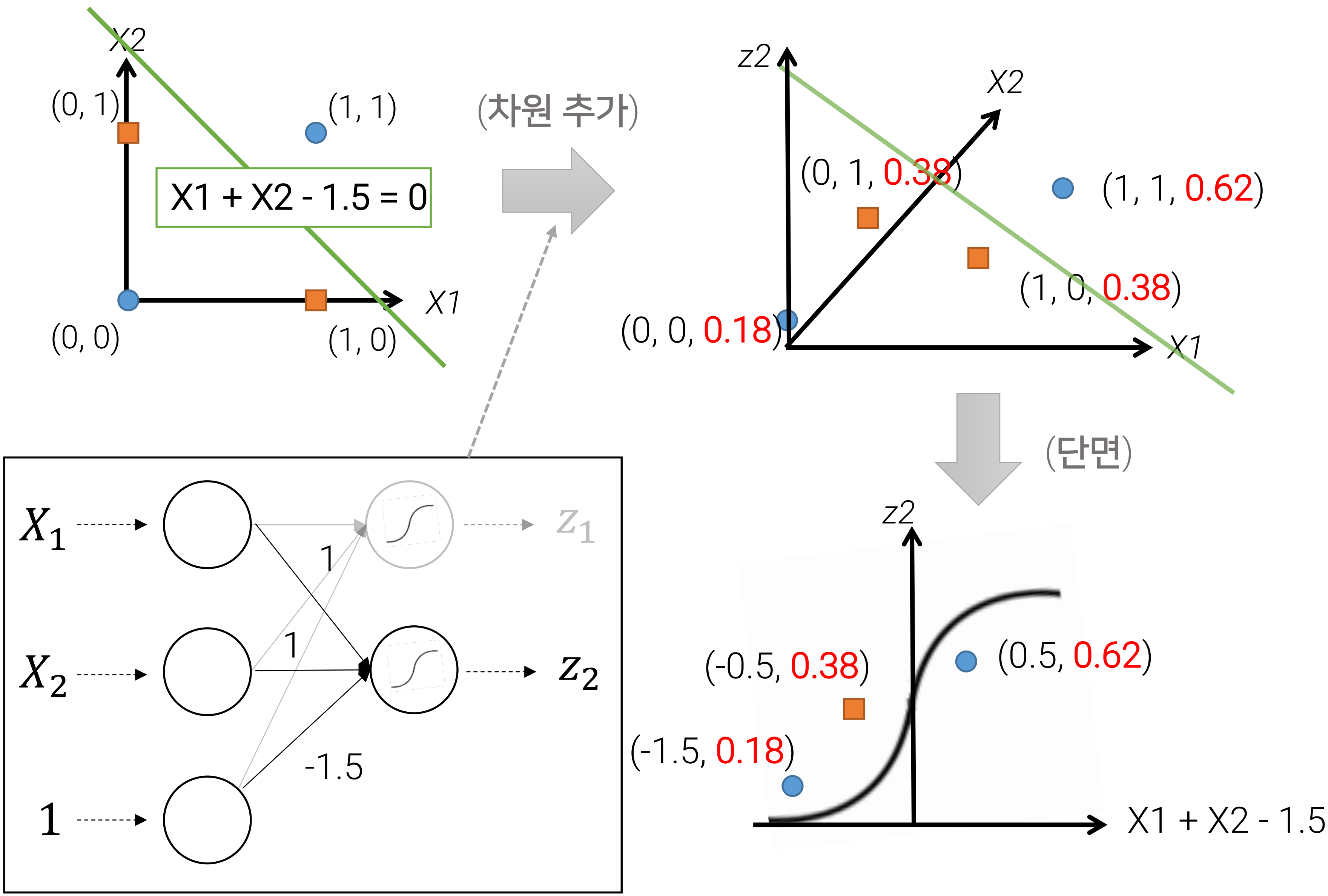
In [5]:
```python
logistic(0 + 1 - 1.5)
```

Out[5]: 0.3775406687981454

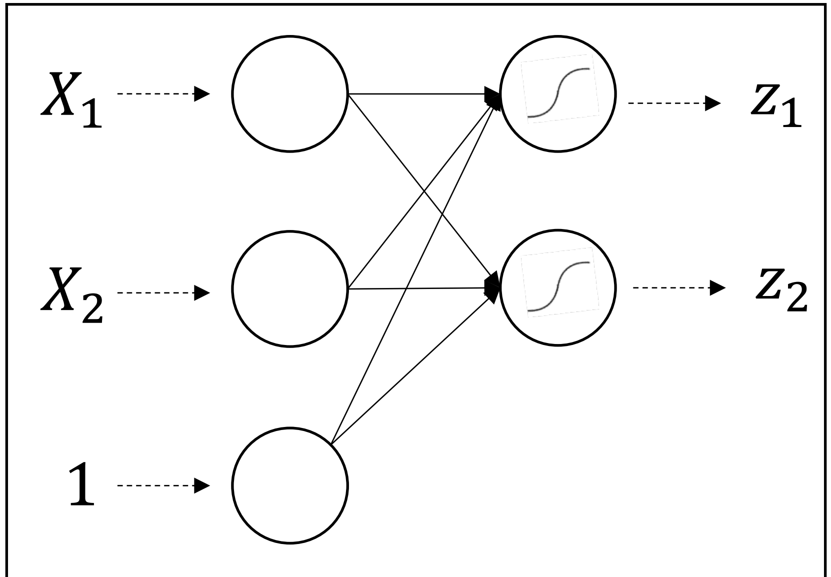
In [6]:
```python
logistic(1 + 0 - 1.5)
```

Out[6]: 0.3775406687981454

XOR classification by feed-forward network model



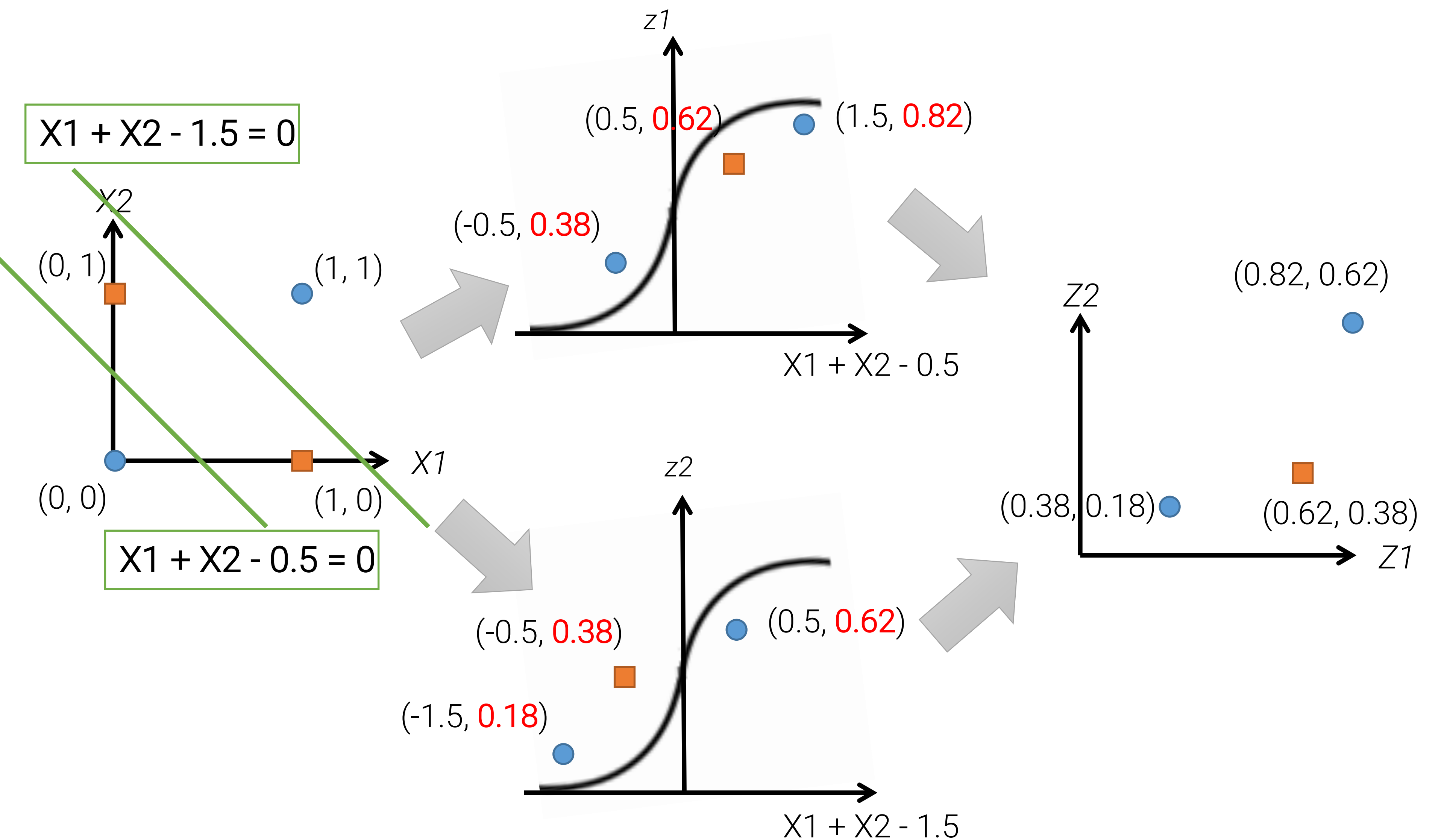
FFN changes data representation



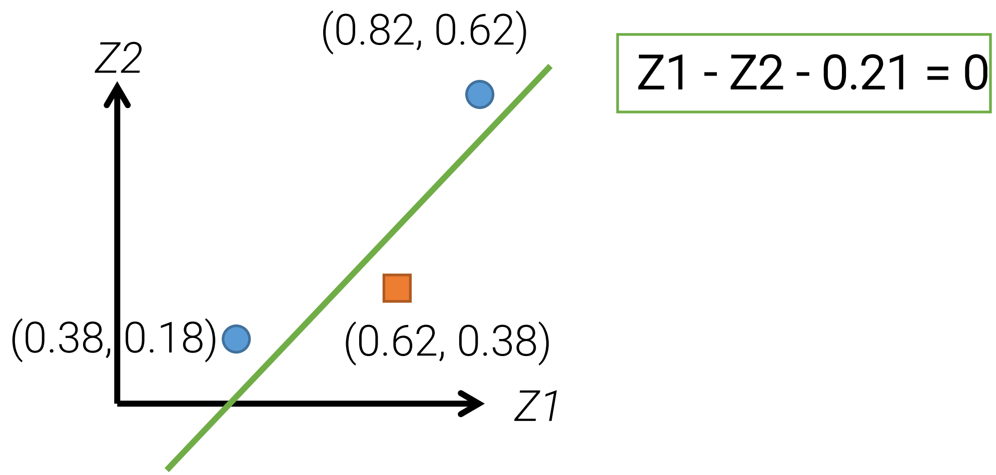
ID	class		X1	X2
1	0	●	0	0
2	1	■	0	1
3	1	■	1	0
4	0	●	1	1

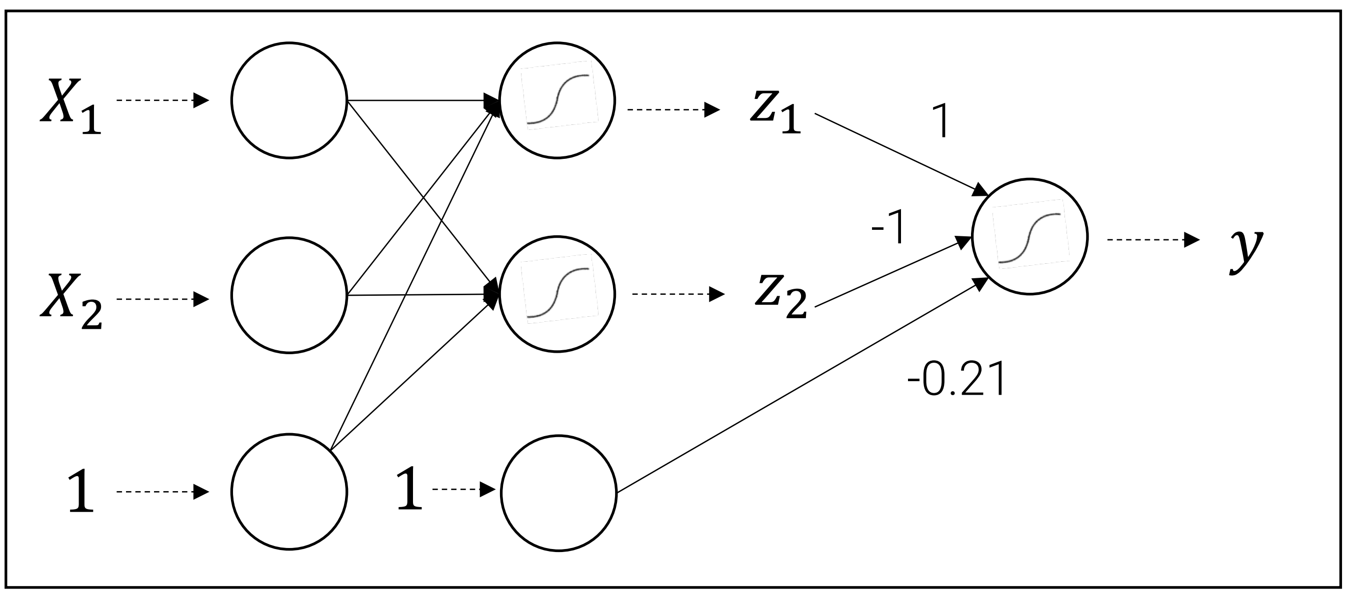
ID	class		Z1	Z2
1	0	●	0.38	0.18
2	1	■	0.62	0.38
3	1	■	0.62	0.38
4	0	●	0.82	0.62

FFN changes data representation



XOR classification by feed-forward network model



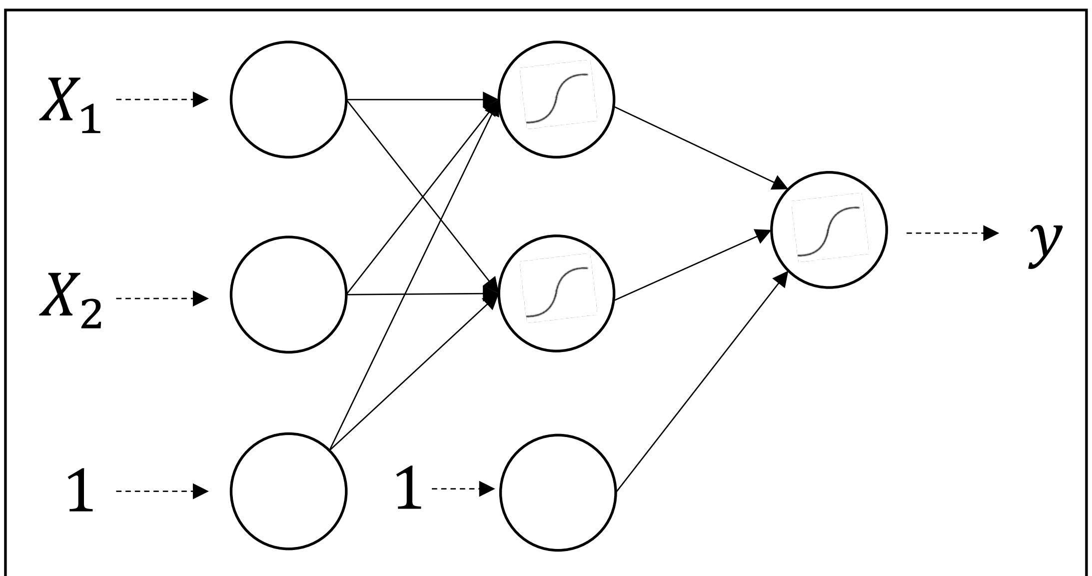


XOR classification by feed-forward network model

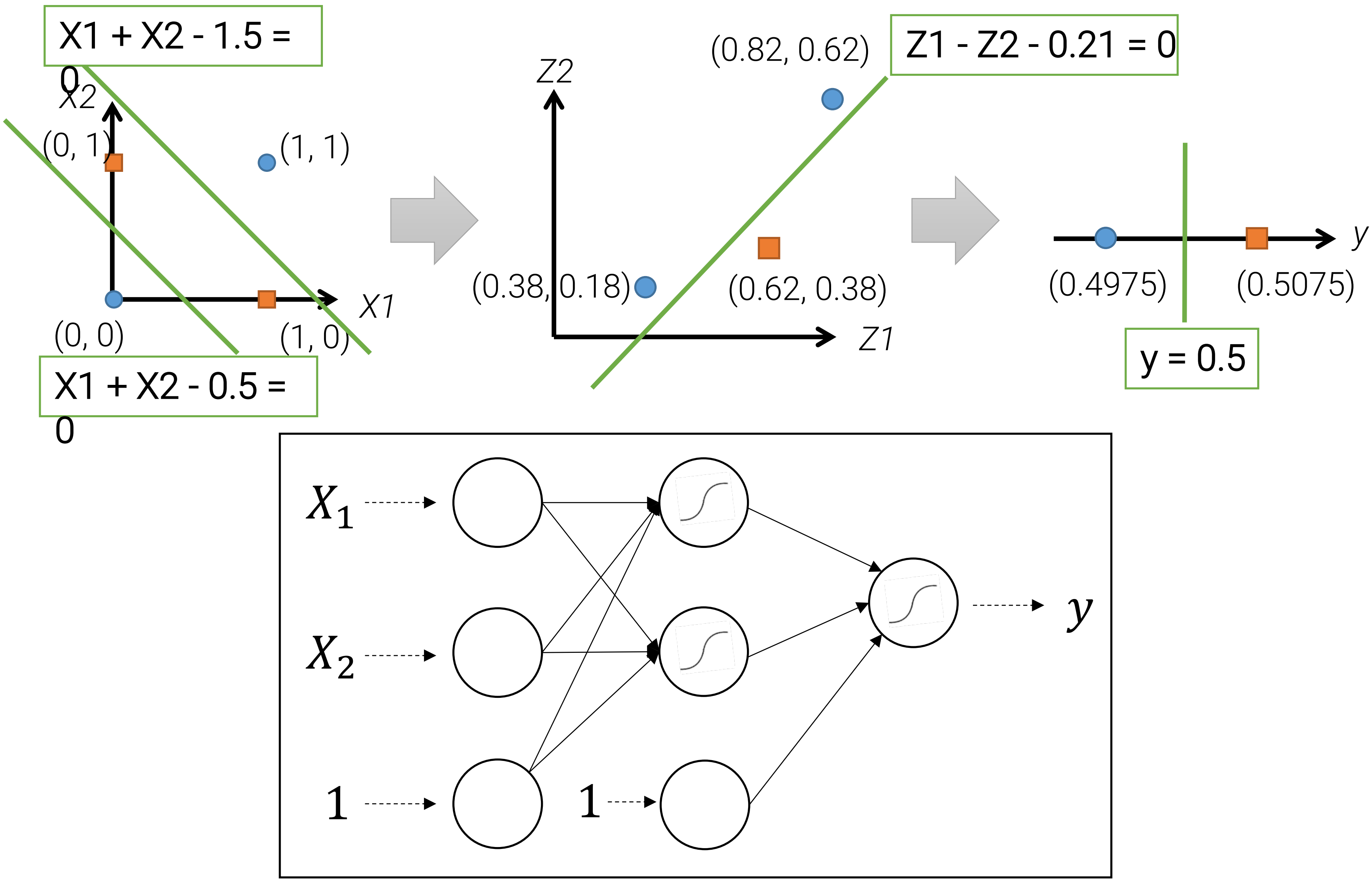
ID	class	X1	X2
1	0	0	0
2	1	0	1
3	1	1	0
4	0	1	1

ID	class	Z1	Z2
1	0	0.38	0.18
2	1	0.62	0.38
3	1	0.62	0.38
4	0	0.82	0.62

ID	class	y
1	0	0.4975
2	1	0.5075
3	1	0.5075
4	0	0.4975

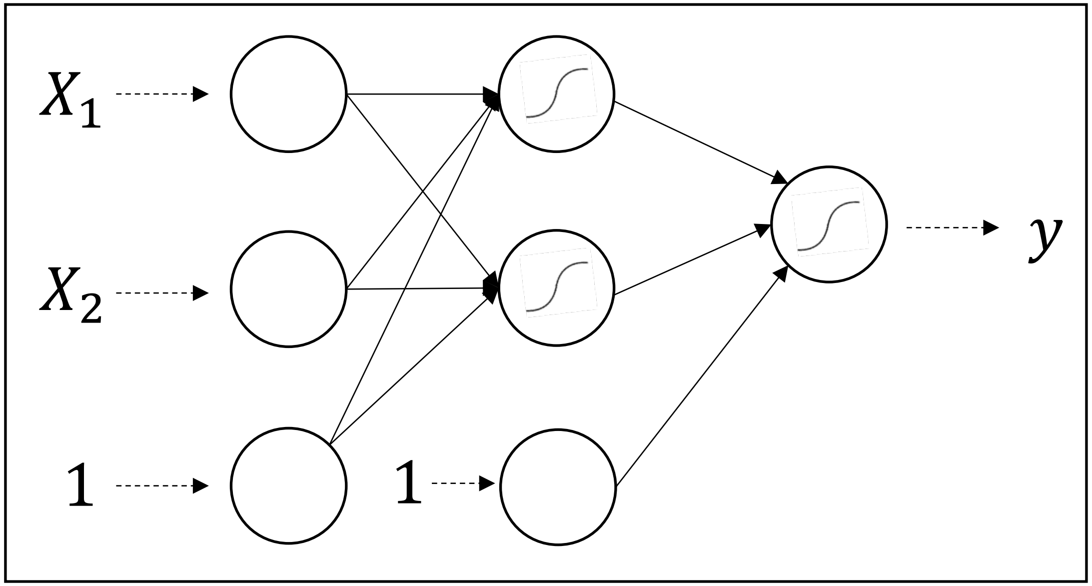


XOR classification by feed-forward network model



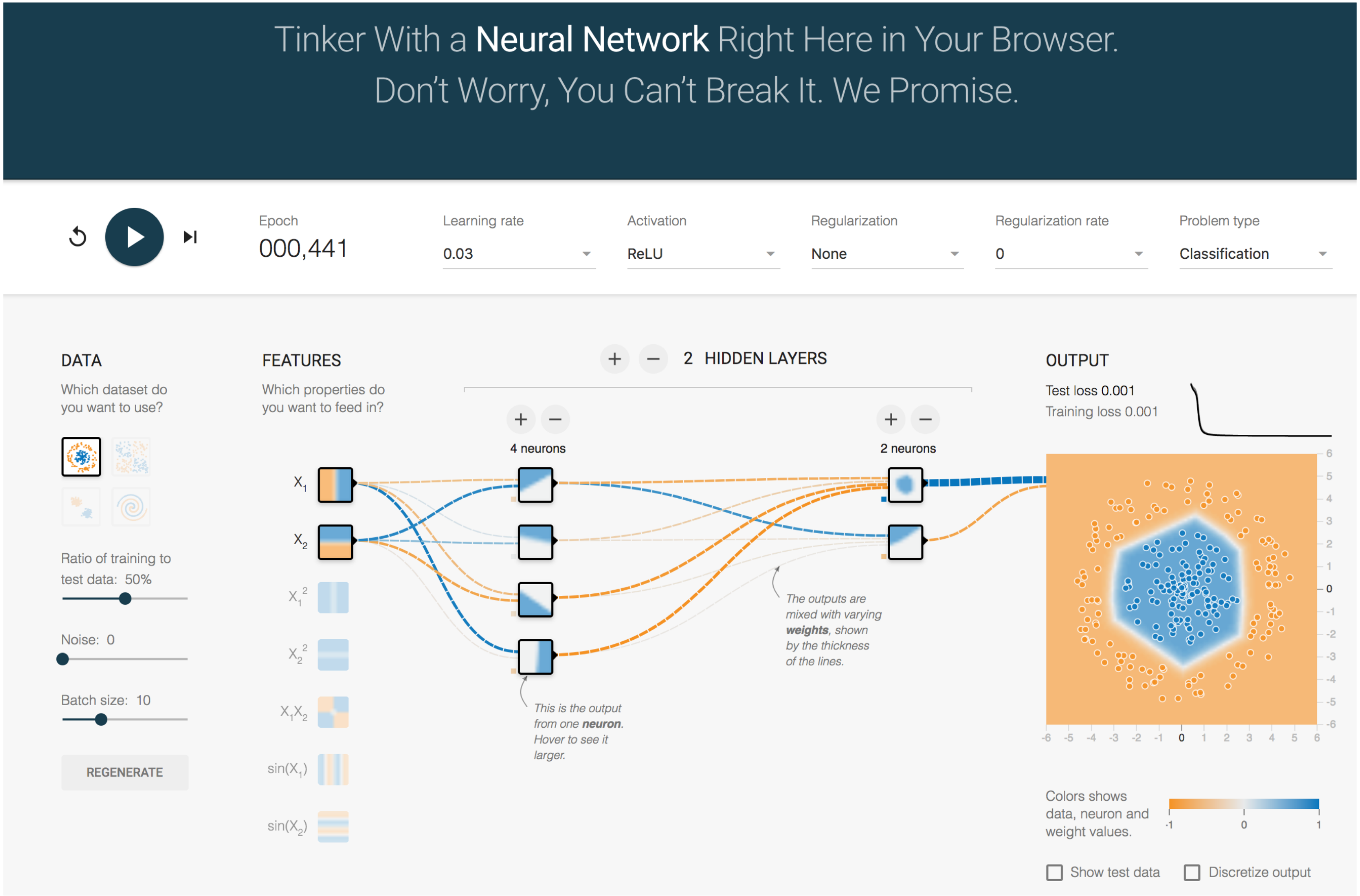
XOR classification by feed-forward network model

- **쉽게 말로 풀어서 쓰자면,**
  - 관측된 (X1, X2) 라는 포인트들이
  - 네트워크의 다음 층(layer)에서 (Z1, Z2)의 형식으로 표현이 바뀐 후,
  - 마지막 층에서 클래스가 분류되었다.



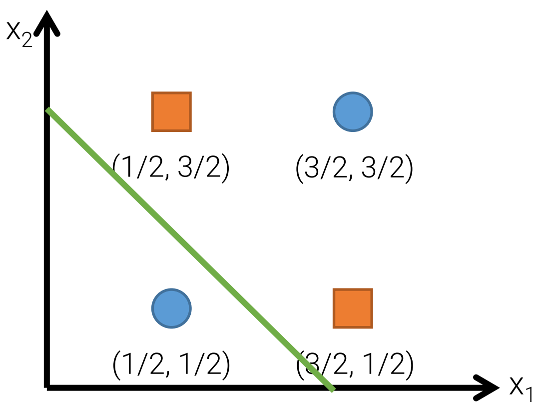
A Neural network playground (by TensorFlow)

- http://playground.tensorflow.org/

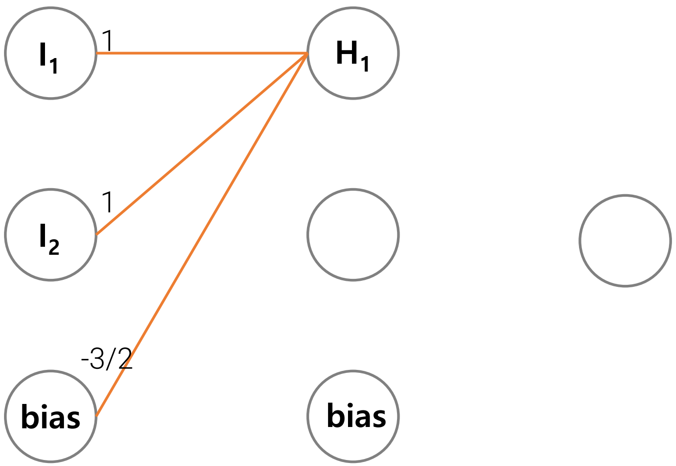


XOR classification by feed-forward network model

- XOR Problem: Revisited



$$h_1 = x_1 + x_2 - \frac{3}{2}$$



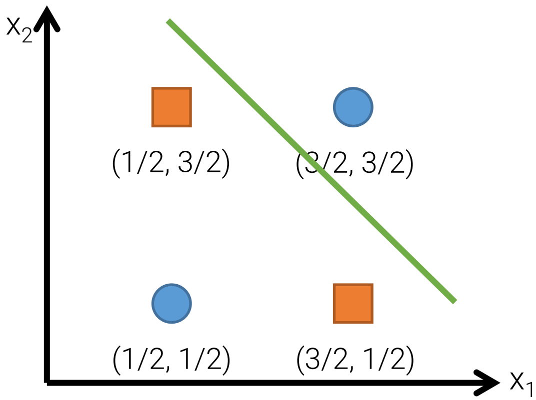
$$z_1 = g(h_1) = \begin{Bmatrix} 1 & if\ h_1 \geq 0 \\ -1 & if\ h_1 < 0 \end{Bmatrix}$$

	x_1	x_2	h_1	z_1
●	1/2	1/2	-1/2	-1
■	3/2	1/2	1/2	1
■	1/2	3/2	1/2	1
●	3/2	3/2	3/2	1

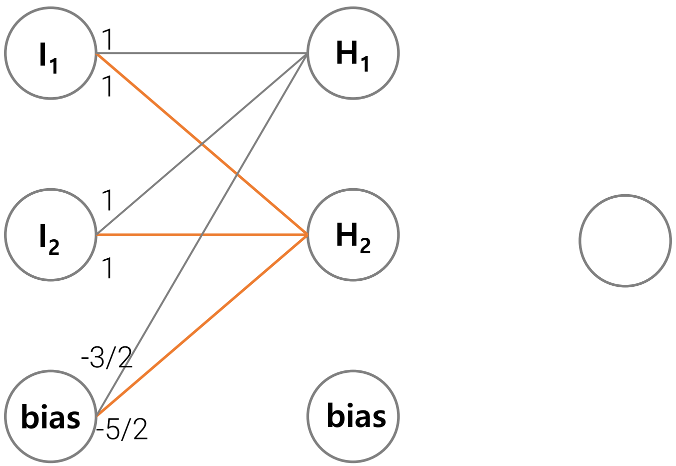
(출처: 고려대 강필성 교수님 강의노트)

XOR classification by feed-forward network model

- XOR Problem: Revisited



$$h_2 = x_1 + x_2 - \frac{5}{2}$$



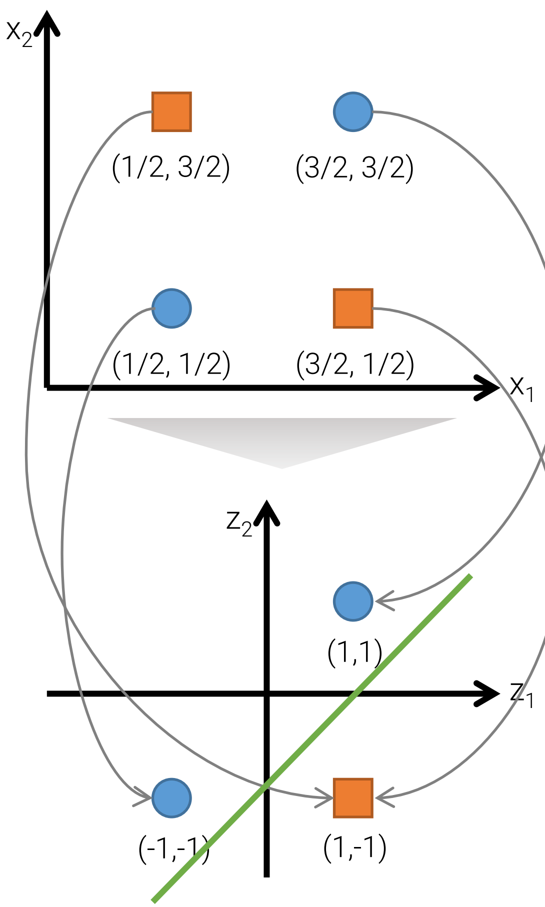
$$z_2 = g(h_2) = \begin{cases} 1 & if\ h_2 \geq 0 \\ -1 & if\ h_2 < 0 \end{cases}$$

x_1	x_2	h_2	z_2
1/2	1/2	-3/2	-1
3/2	1/2	-1/2	-1
1/2	3/2	-1/2	-1
3/2	3/2	1/2	1

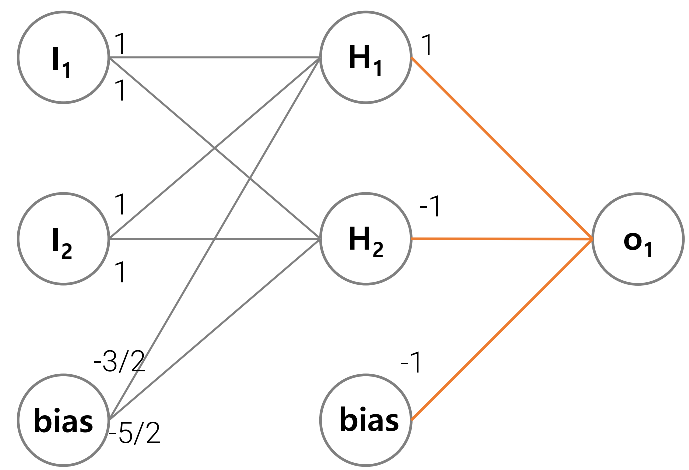
(출처: 고려대 강필성 교수님 강의노트)

XOR classification by feed-forward network model

- XOR Problem: Revisited



$$o_1 = z_1 - z_2 - 1$$



	x_1	x_2	h_1	z_1	h_2	z_2	o_1	z
	1/2	1/2	-1/2	-1	-3/2	-1	-1	-1
	3/2	1/2	1/2	1	-1/2	-1	1	1
	1/2	3/2	1/2	1	-1/2	-1	1	1
	3/2	3/2	3/2	1	1/2	1	-1	-1

$$z = g(o_1) = \begin{cases} 1 & if\ o_1 \geq 0 \\ -1 & if\ o_1 < 0 \end{cases}$$

(출처: 고려대 강필성 교수님 강의노트)