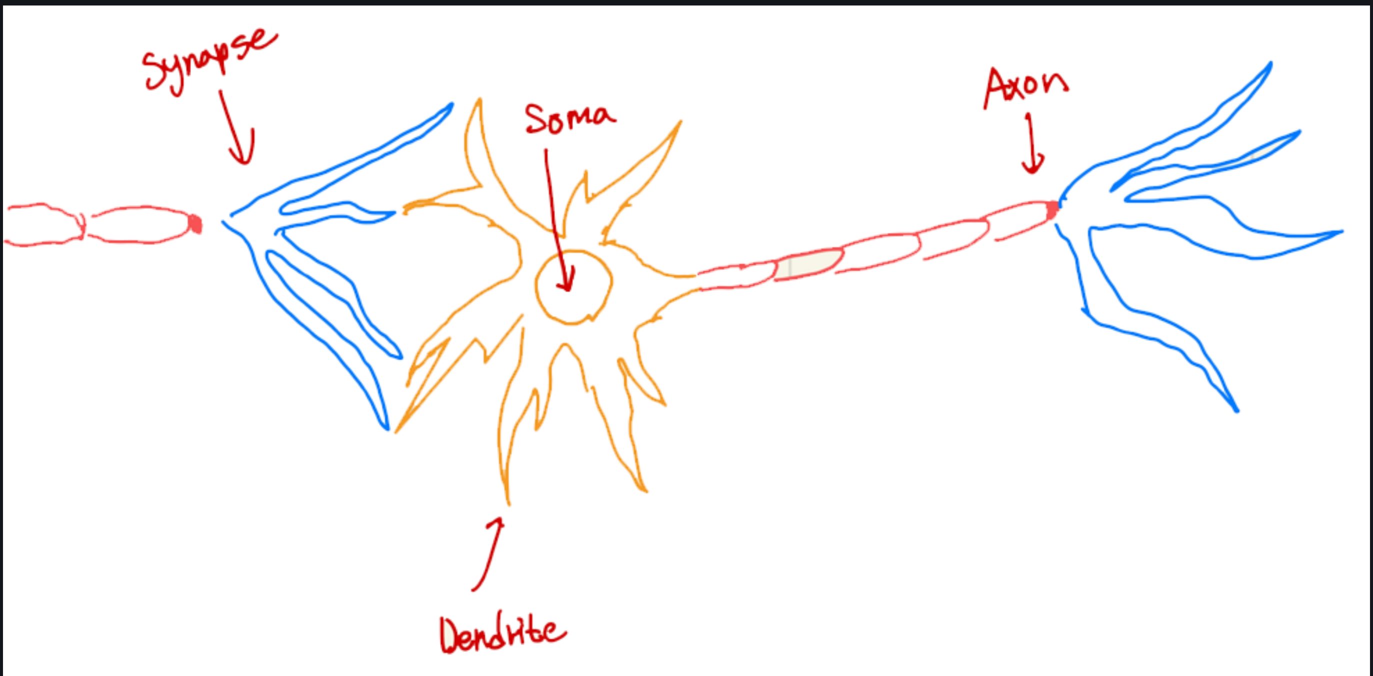
인공신경망

뉴런(Neuron)



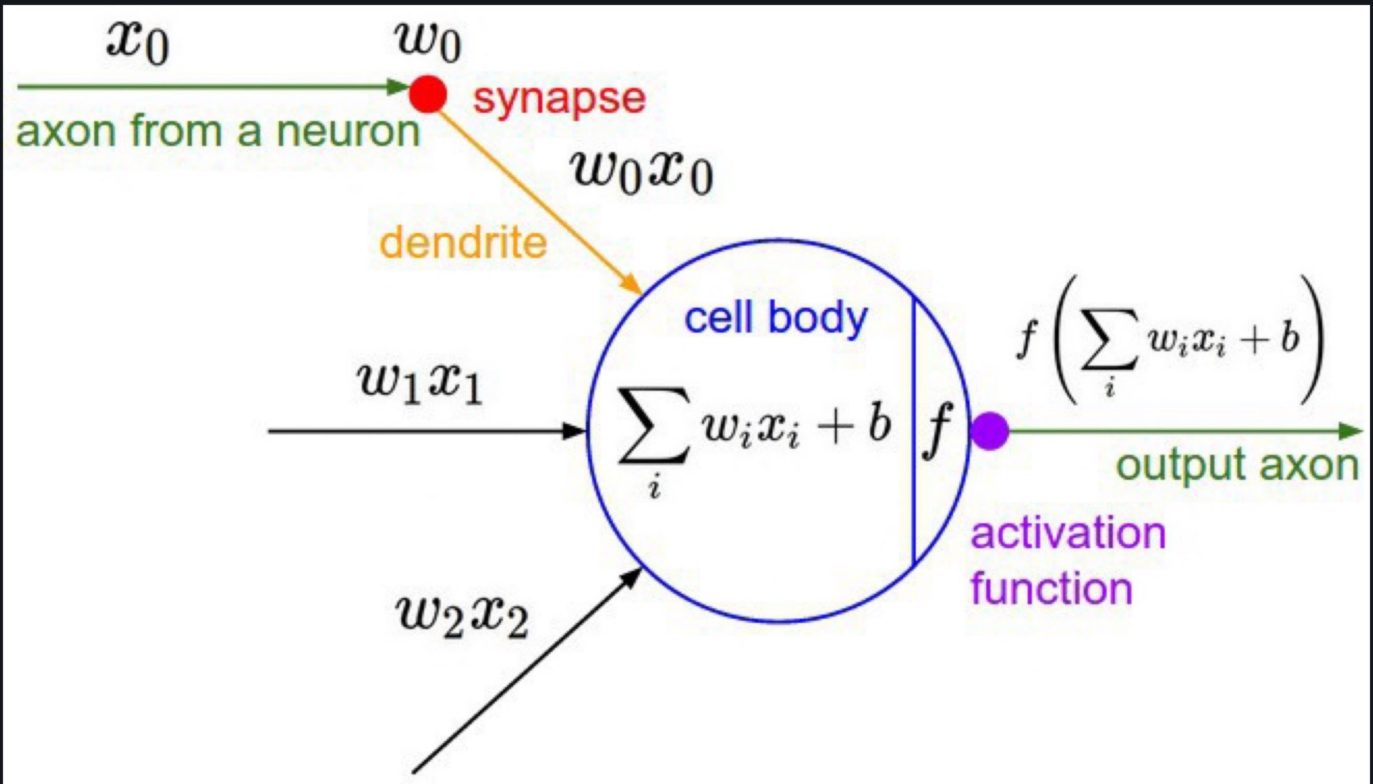
<Neuron>

Synapse : 시냅스
다른 뉴런과 Dendrite의 연결 부위에 있으며 전기 신호의 세기를 재조정 한다. (가중치와 비슷한 역할)

Dendrite : 수상돌기
이웃 뉴런에서 전기 신호를 받는다.

SOMA : 세포체
Dendrite로 부터 받은 여러 전기 신호들을 모두 합친다.

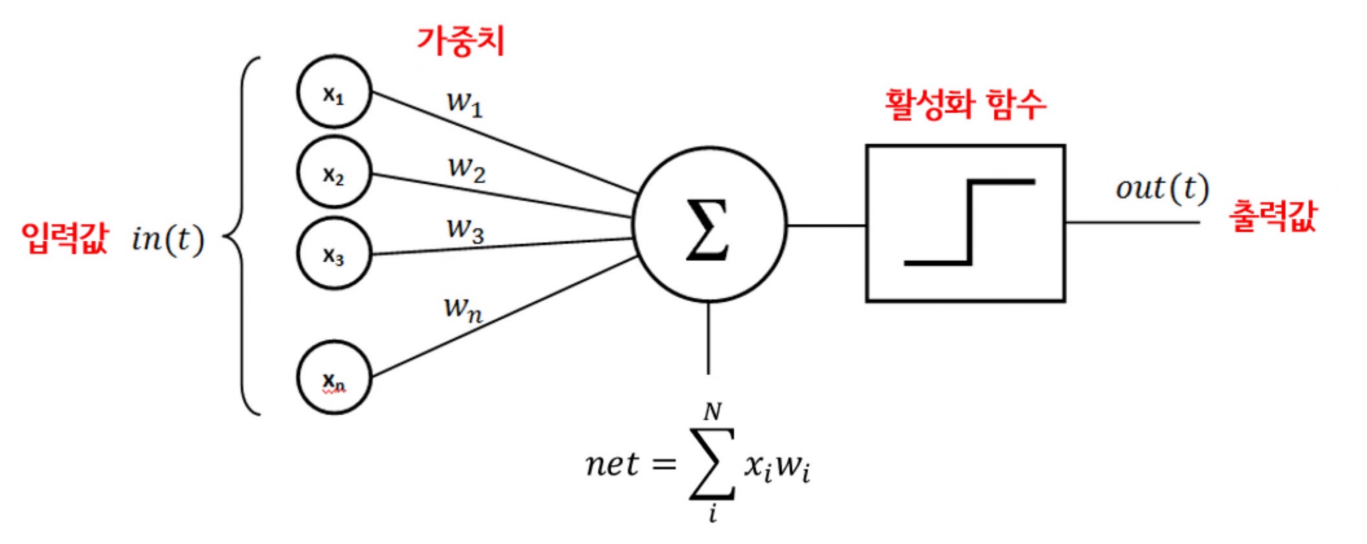
Axon : 축삭
Soma의 전위가 일정 이상이 되면 이웃 뉴런으로 전기 신호를 보낸다.



<Perceptron>

퍼셉트론의 파생은 이러한 생물학적 뉴런의 동작 방식을 인공적으로 만들기 위해서 수학적으로 모델링한 개념(인공뉴런)

퍼셉트론(Perceptron)

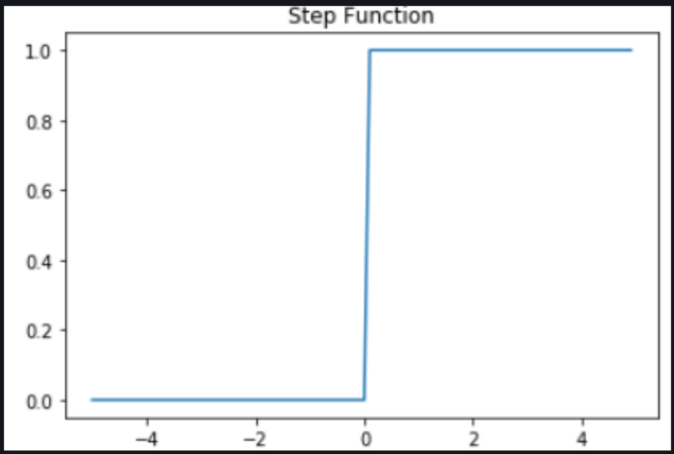


<Perceptron>

퍼셉트론(단층 퍼셉트론) 신호는 Activation Function이 Step function이기 때문에 흐른다 or 안 흐른다(1 or 0) 두 가지 값만 갖는다.

퍼셉트론의 활성화 함수를 먼저 알아보자.

Step function



<Step function>

$$
\begin{gathered}
if \sum_{i}^{n} w_i x_i > \theta \rightarrow 1 \\
if \sum_{i}^{n} w_i x_i \leq \theta \rightarrow 0 \\
\theta = threshold
\end{gathered} \tag{1}
$$

θ 값을 기준으로 보다 클 경우 전기 신호가 흐르며 작을 경우 전기 신호를 안 흐른다고 볼 수 있다.

퍼셉트론은 직선 하나로 나눈 영역으로, 선형분류만 표현할 수 있어 선형분류기이다.

이 때, 선형 분류기는 하나의 직선으로 두 종류의 데이터를 이진 분류할 수 있는 모델을 말한다.

이제 퍼셉트론의 선형 분류의 예시를 위해 AND Gate를 사용해서 아래 좌표평면에 표기하겠다.

AND GATE

X_1	X_2	Y
1	1	1
1	0	0
0	1	0
0	0	0

이때 가독성을 위해 bias를 θ로 치환하고 이를 임계값으로 사용하겠다.

$$
\begin{gathered}
y = \begin{cases} 0, & x_1w_1 + x_2w_2 + b \leq 0 \\ 1, & x_1w_1 + x_2w_2 + b > 0 \end{cases}, \quad (b를\ -\theta로치환) \\
y = \begin{cases} 0, & x_1w_1 + x_2w_2 \leq \theta \\ 1, & x_1w_1 + x_2w_2 > \theta \end{cases}
\end{gathered} \tag{2}
$$

AND GATE 를 분류하기 위해 $w_1 = 0.5, w_2 = 0.5\ \theta = 0.7$ 를 대입하면 아래와 같은 수식이 만들어진다.

그리고 AND GATE의 Input을 x_1과 x_2에 넣어본다.

$$
y = \begin{cases} 0, & 0.5x_1 + 0.5x_2 \leq 0.7 \\ 1, & 0.5x_1 + 0.5x_2 > 0.7 \end{cases}
$$

$$
\begin{array}{ll}
i)\ (1,\ 1) \rightarrow 1 & ii)\ (1,\ 0) \rightarrow 0 & (3) \\
0.5 \times 1 + 0.5 \times 1 = 1 > 0.7 \rightarrow 1 & 0.5 \times 1 + 0.5 \times 0 = 0.5 \leq 0.7 \rightarrow 0 & (4) \\
 & & (5) \\
iii)\ (0,\ 1) \rightarrow 0 & iv)\ (0,\ 0) \rightarrow 0 & (6) \\
0.5 \times 0 + 0.5 \times 1 = 0.5 \leq 0.7 \rightarrow 0 & 0.5 \times 0 + 0.5 \times 0 = 0 \leq 0.7 \rightarrow 0 & (7)
\end{array}
$$

직선의 방정식을 구하기 위해 일차 부등식에서 일차 방정식으로 정리하겠다.

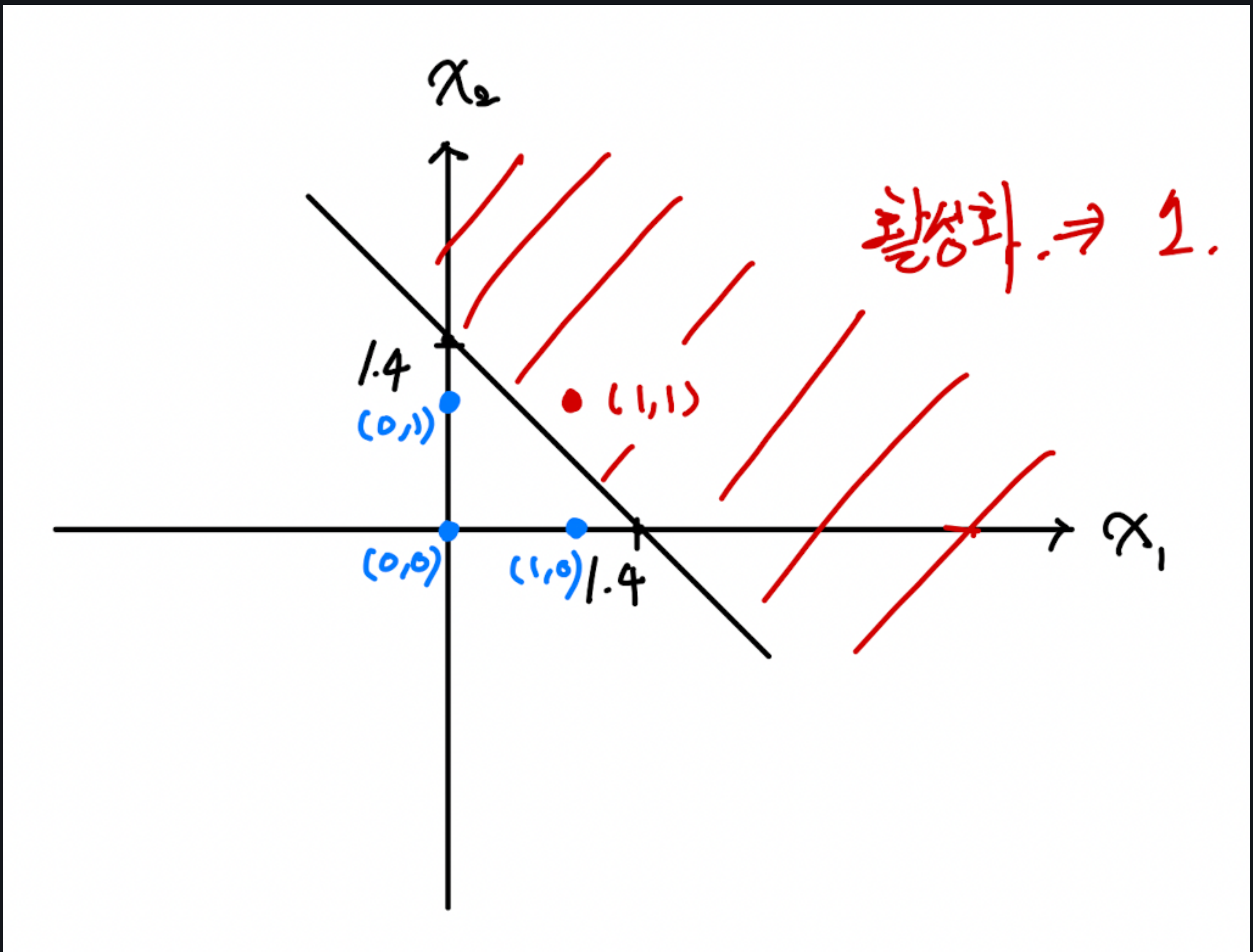
$$
\begin{gathered}
w_1x_1 + w_2x_2 = \theta \\
0.5x_1 + 0.5x_2 = 0.7 \\
\frac{1}{1.4}x_1 + \frac{1}{1.4}x_2 = 1
\end{gathered} \tag{8}
$$

위 수식에서 각각의 절편을 구해보자.

$$
\begin{gathered}
x_1의\ 절편 : \frac{\theta}{w_1} = \frac{0.7}{0.5} = 1.4 \\
x_2의\ 절편 : \frac{\theta}{w_2} = \frac{0.7}{0.5} = 1.4
\end{gathered} \tag{9}
$$

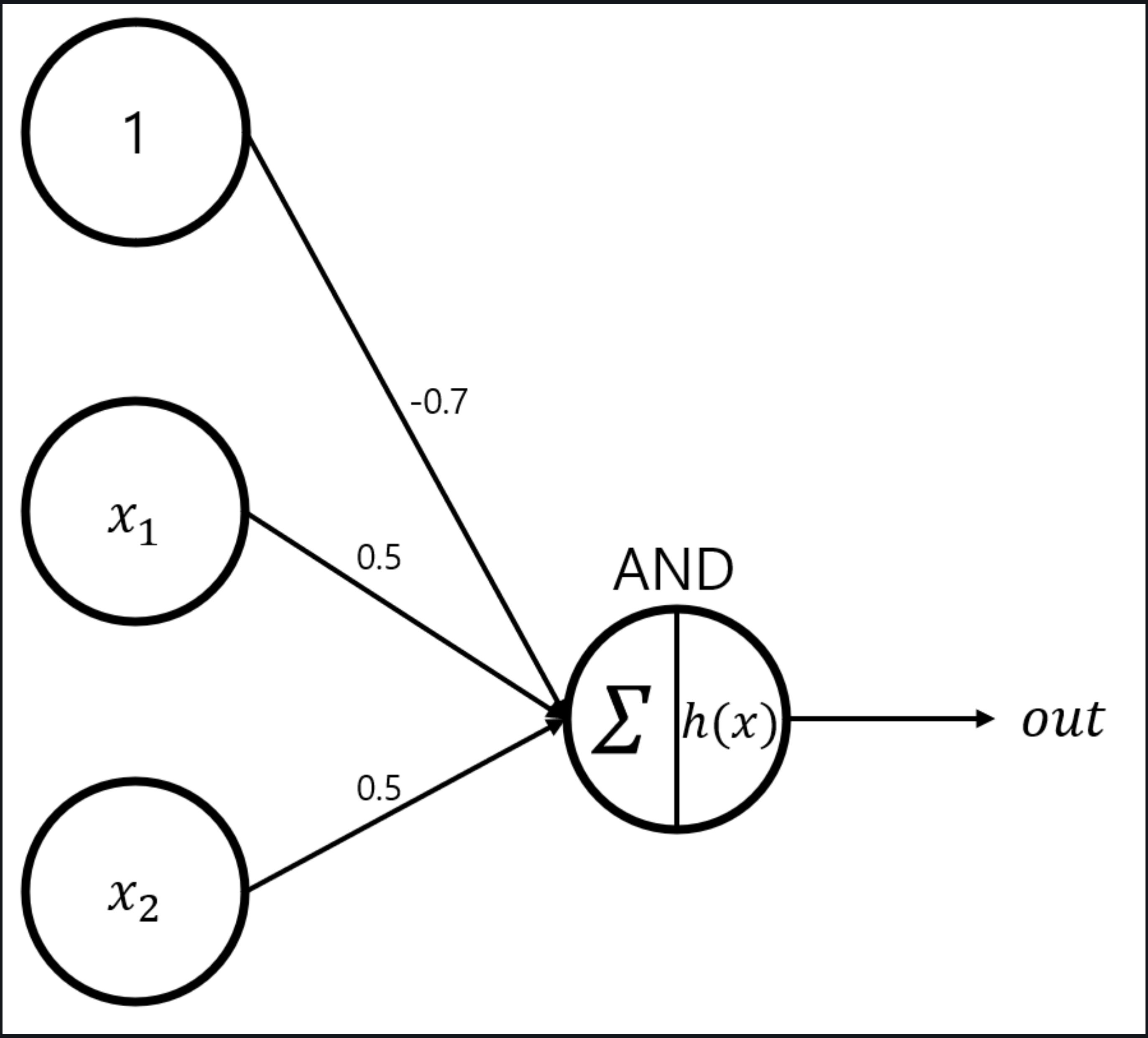
따라서 다음과 같은 부등식은 AND Gate에서 True에 해당하는 점에 대해 활성화 시켜준다.

$$
\begin{aligned}
0.5x_1 + 0.5x_2 &> 0.7 \\
\frac{1}{1.4}x_1 + \frac{1}{1.4}x_2 &> 1
\end{aligned} \tag{10}
$$



<AND Gate>

퍼셉트론으로 표현하면 아래 그림과 같다.



<AND Gate>

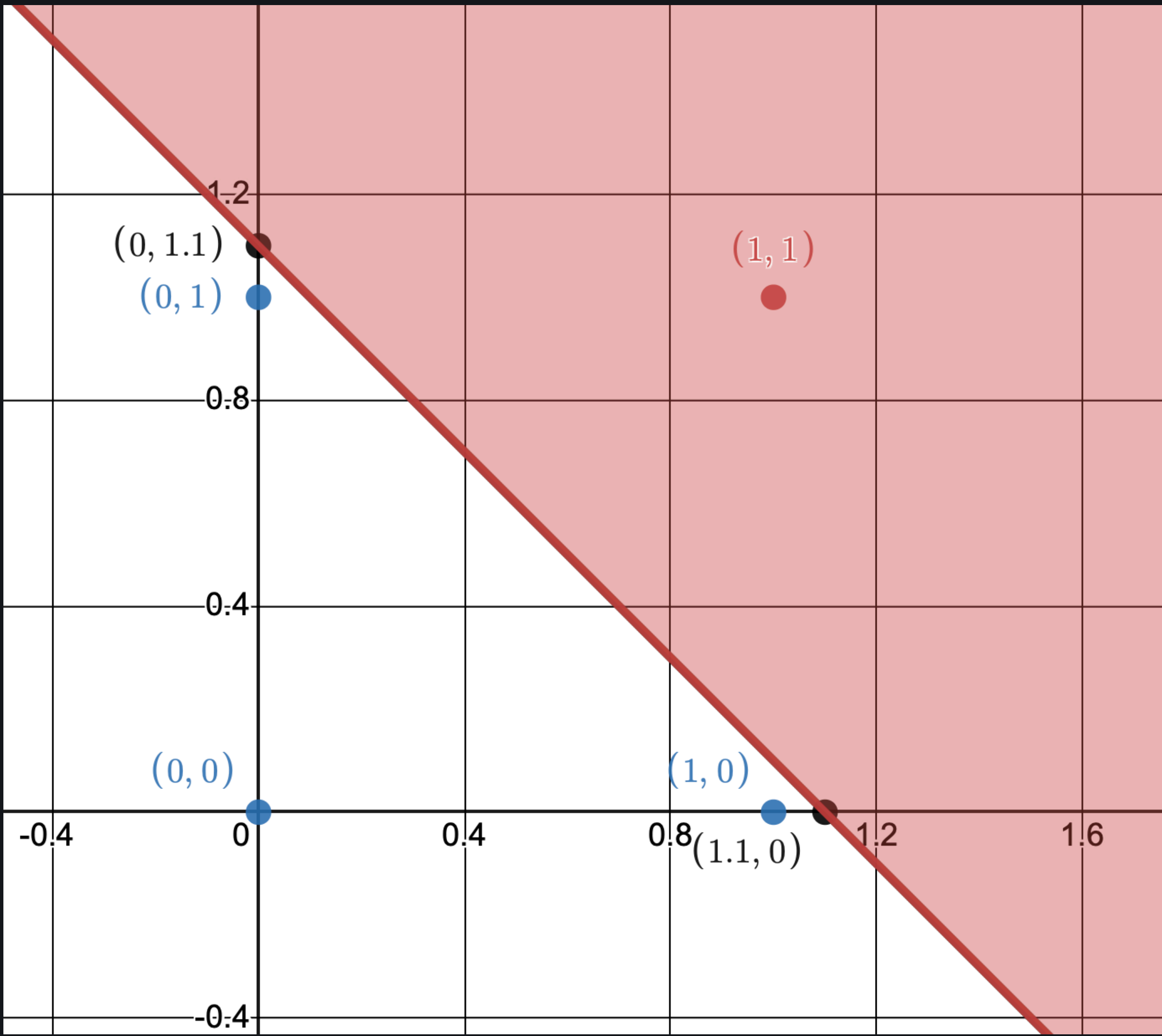
그렇다면 꼭 매개변수($w_1,\ w_2,\ \theta$) 의 값이 0.5, 0.5, 0.7만 가능한가? 정답은 아니다.

매개변수($w,\ \theta$)의 조합에 따라서 다양한 그래프가 나올 수 있다.

예를들어 x_1과 x_2의 각각의 절편을 1.1로 둔다면 다음과 같은 식이 성립된다. ($w_1 = 1,\ w_2 = 1,\ \theta = 1.1$)

$$\begin{gathered} \frac{1}{1.1}x_1 + \frac{1}{1.1}x_2 = 1 \\ \frac{1}{1.1}x_1 + \frac{1}{1.1}x_2 > 1 \end{gathered} \tag{11}$$

위 부등식의 영역은 다음과 같은 영역을 가진다.



<AND Gate>

다음은 OR GATE에 대해서도 알아보겠다.

OR GATE

X_1	X_2	Y
1	1	1
1	0	1
0	1	1
0	0	0

$$y = \begin{cases} 0, & x_1w_1 + x_2w_2 + b \le 0 \\ 1, & x_1w_1 + x_2w_2 + b > 0 \end{cases}, \quad (b\text{를 } -\theta\text{로치환})$$
$$y = \begin{cases} 0, & x_1w_1 + x_2w_2 \le \theta \\ 1, & x_1w_1 + x_2w_2 > \theta \end{cases} \tag{12}$$

위 수식에서 다음과 같이 쓸 수 있다.

OR GATE 를 분류하기 위해 $w_1 = 0.5, w_2 = 0.5\ \theta = 0.2$ 를 대입하면 아래와 같은 수식이 만들어진다.

그리고 OR GATE의 Input을 x_1과 x_2에 넣어본다.

$$y = \begin{cases} 0, & 0.5x_1 + 0.5x_2 \le 0.2 \\ 1, & 0.5x_1 + 0.5x_2 > 0.2 \end{cases}$$

$$i)\ (1,\ 1) \rightarrow 1 \qquad ii)\ (1,\ 0) \rightarrow 1 \tag{13}$$
$$0.5 \times 1 + 0.5 \times 1 = 1 > 0.2 \rightarrow 1 \qquad 0.5 \times 1 + 0.5 \times 0 = 0.5 > 0.2 \rightarrow 1 \tag{14}$$
$$\tag{15}$$
$$iii)\ (0,\ 1) \rightarrow 1 \qquad iv)\ (0,\ 0) \rightarrow 0 \tag{16}$$
$$0.5 \times 0 + 0.5 \times 1 = 0.5 > 0.2 \rightarrow 1 \qquad 0.5 \times 0 + 0.5 \times 0 = 0 \le 0.2 \rightarrow 0 \tag{17}$$

다음과 같이 잘 분류해낸 것을 볼 수 있으며 직선의 방정식을 구하기 위해 일차 부등식에서 일차 방정식으로 정리하겠다.

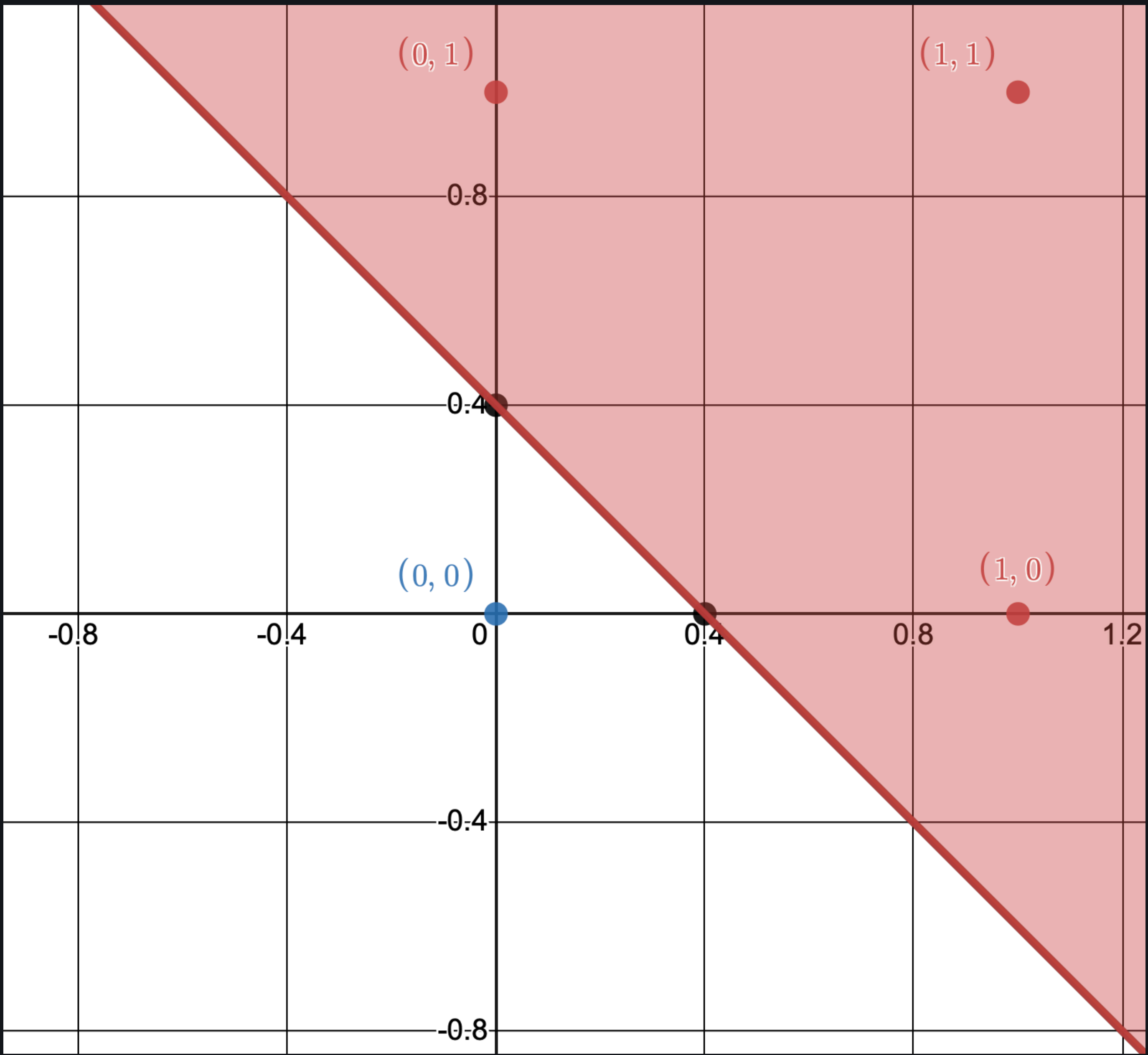
$$\begin{gathered} w_1x_1 + w_2x_2 = \theta \\ 0.5x_1 + 0.5x_2 = 0.2 \\ \frac{1}{0.4}x_1 + \frac{1}{0.4}x_2 = 1 \end{gathered} \tag{18}$$

위 수식에서 각각의 절편을 구해보자

$$\begin{gathered} x_1\text{의 절편} : \frac{\theta}{w_1} = \frac{0.2}{0.5} = 0.4 \\ x_2\text{의 절편} : \frac{\theta}{w_2} = \frac{0.2}{0.5} = 0.4 \end{gathered} \tag{19}$$

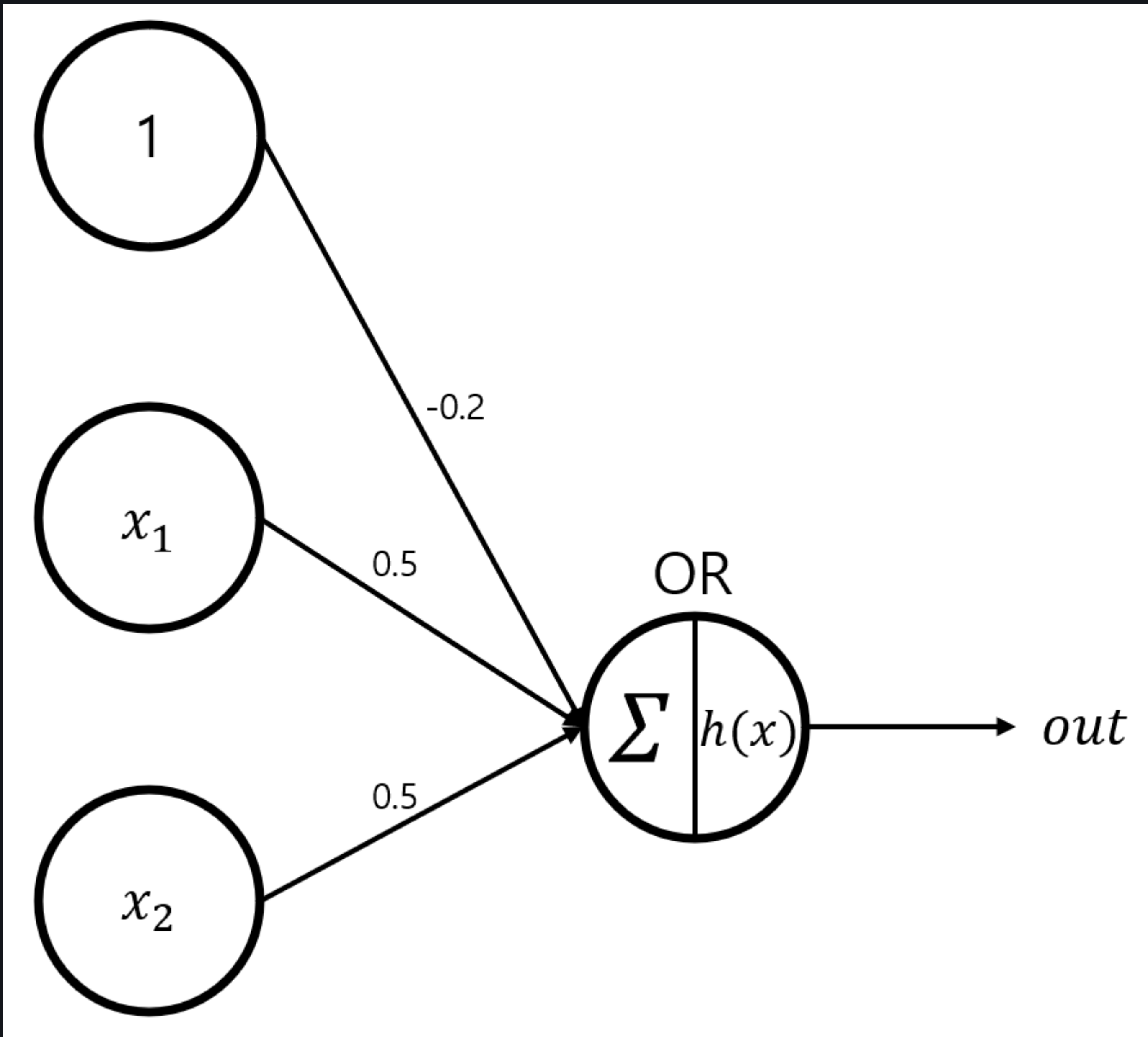
따라서 다음과 같은 부등식은 OR Gate에서 True에 해당하는 점에 대해 활성화 시켜준다.

$$
\begin{aligned}
0.5x_1 + 0.5x_2 &> 0.2 \\
\frac{1}{0.4}x_1 + \frac{1}{0.4}x_2 &> 1
\end{aligned} \tag{20}
$$



<OR Gate>

퍼셉트론으로 표현하면 아래 그림과 같다.

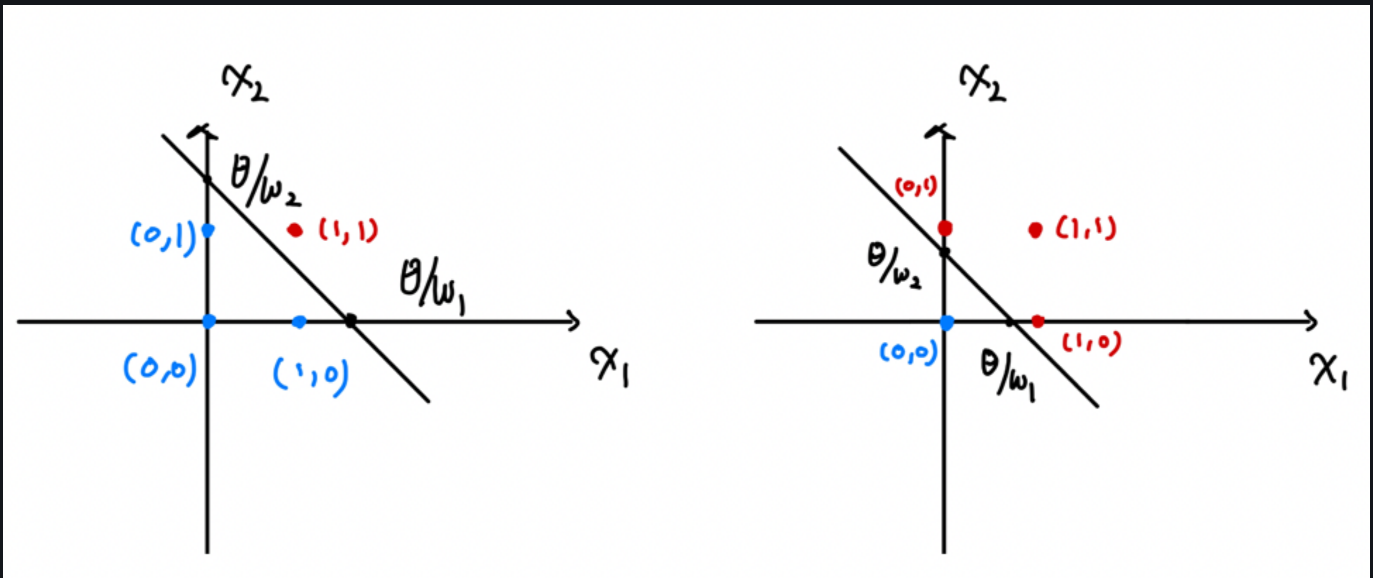


<OR Gate>

OR GATE도 마찬가지로 매개변수($w,\ \theta$)의 조합에 따라서 다양한 그래프가 나올 수 있다.

방법은 AND GATE에서 설명했으니 생략하겠다.

아래 AND, OR GATE의 일반적인 그래프 개형이다.



<AND, OR Gate>

마지막으로 NAND GATE에 대해서도 생각해 보자.

NAND GATE

X_1	X_2	Y
1	1	0
1	0	1
0	1	1
0	0	1

$$y = \begin{cases} 0, & x_1w_1 + x_2w_2 + b \leq 0 \\ 1, & x_1w_1 + x_2w_2 + b > 0 \end{cases}, \quad (b\text{를 } -\theta\text{로치환})$$
$$y = \begin{cases} 0, & x_1w_1 + x_2w_2 \leq \theta \\ 1, & x_1w_1 + x_2w_2 > \theta \end{cases} \tag{21}$$

NAND GATE 를 분류하기 위해 $w_1 = -0.5, w_2 = -0.5$ $\theta = -0.7$ 를 대입하면 아래와 같은 수식이 만들어진다.

그리고 NAND GATE의 Input을 x_1과 x_2에 넣어본다.

$$y = \begin{cases} 0, & -0.5x_1 - 0.5x_2 \le -0.7 \\ 1, & -0.5x_1 - 0.5x_2 > -0.7 \end{cases}$$

$$i)\ (1,\ 1) \to 0 \qquad ii)\ (1,\ 0) \to 1 \tag{22}$$
$$-0.5 \times 1 - 0.5 \times -1 = -1 \le -0.7 \to 0 \qquad -0.5 \times 1 - 0.5 \times 0 = -0.5 \ge -0.7 \to 1 \tag{23}$$
$$\tag{24}$$
$$iii)\ (0,\ 1) \to 1 \qquad iv)\ (0,\ 0) \to 1 \tag{25}$$
$$-0.5 \times 0 - 0.5 \times 1 = -0.5 \ge -0.7 \to 1 \qquad -0.5 \times 0 - 0.5 \times 0 = 0 \ge -0.7 \to 1 \tag{26}$$

직선의 방정식을 구하기 위해 일차 부등식에서 일차 방정식으로 정리하겠다.

$$\begin{gathered} w_1x_1 + w_2x_2 = \theta \\ -0.5x_1 - 0.5x_2 = -0.7 \\ \frac{1}{1.4}x_1 + \frac{1}{1.4}x_2 = 1 \end{gathered} \tag{27}$$

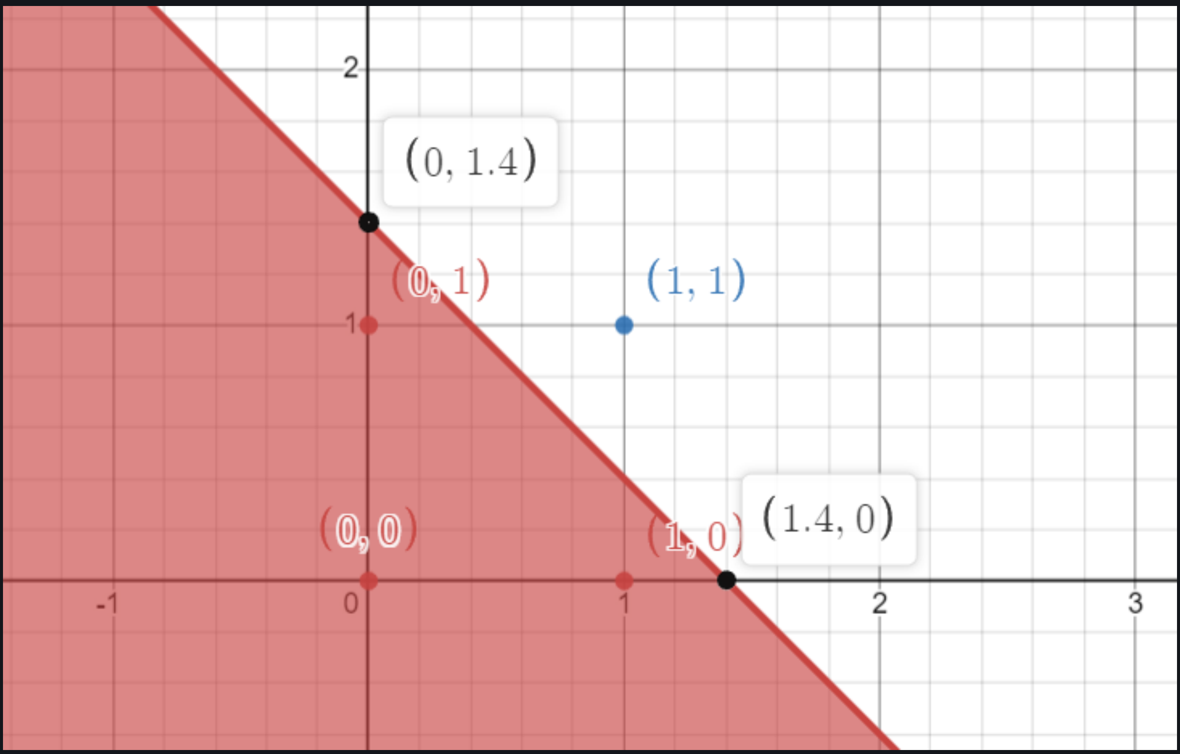
위 수식에서 각각의 절편을 구해보자.

$$\begin{gathered} x_1 의\ 절편 : \frac{\theta}{w_1} = \frac{0.7}{0.5} = 1.4 \\ x_2 의\ 절편 : \frac{\theta}{w_2} = \frac{0.7}{0.5} = 1.4 \end{gathered} \tag{28}$$

따라서 다음과 같은 부등식은 NAND Gate에서 True에 해당하는 점에 대해 활성화 시켜준다.

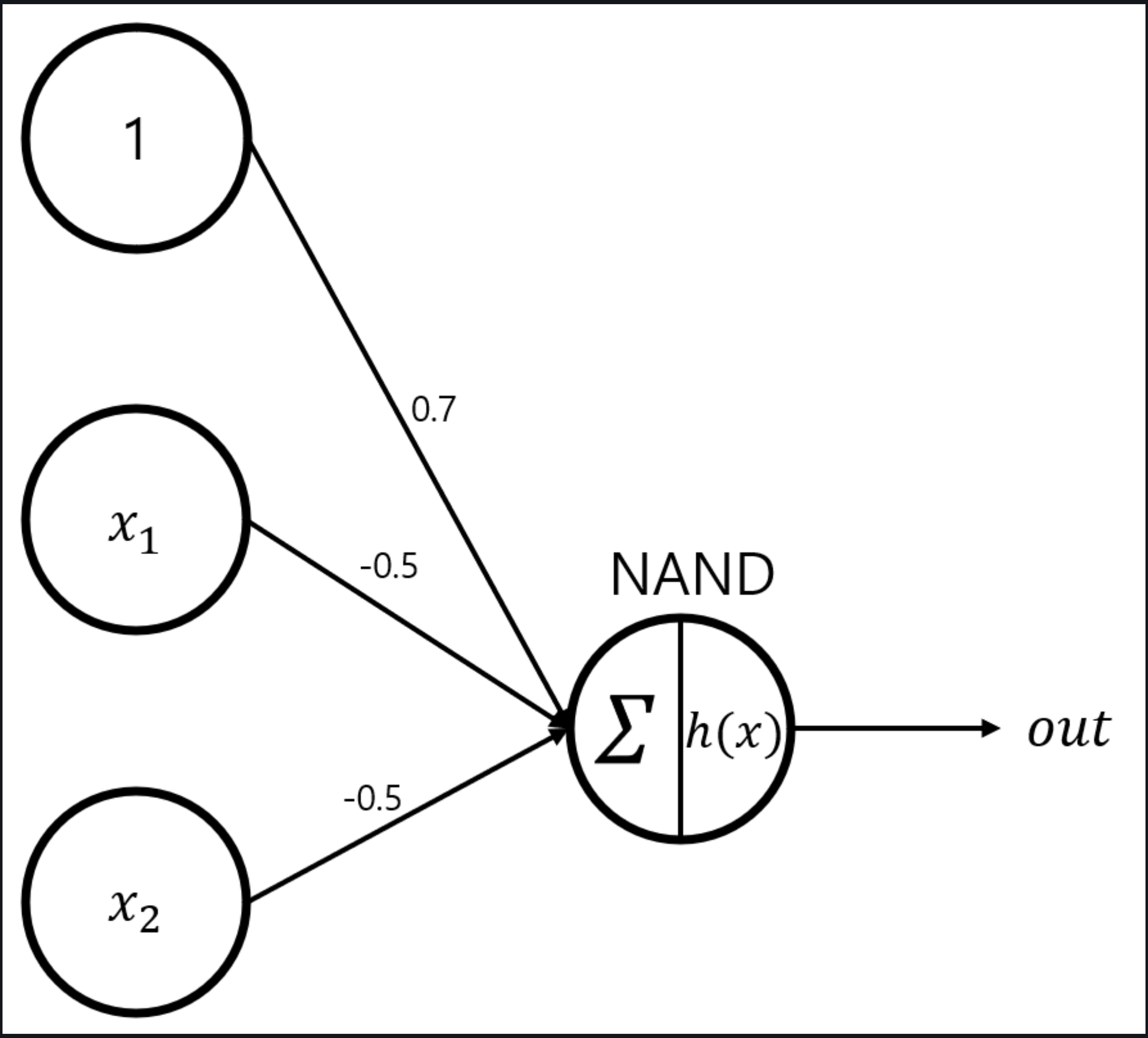
$$\begin{gathered} -0.5x_1 - 0.5x_2 > -0.7 \\ \frac{1}{1.4}x_1 + \frac{1}{1.4}x_2 < 1 \end{gathered} \tag{29}$$

다음은 영역을 표시한 그림이다.



<NAND Gate>

퍼셉트론으로 표현하면 아래 그림과 같다.

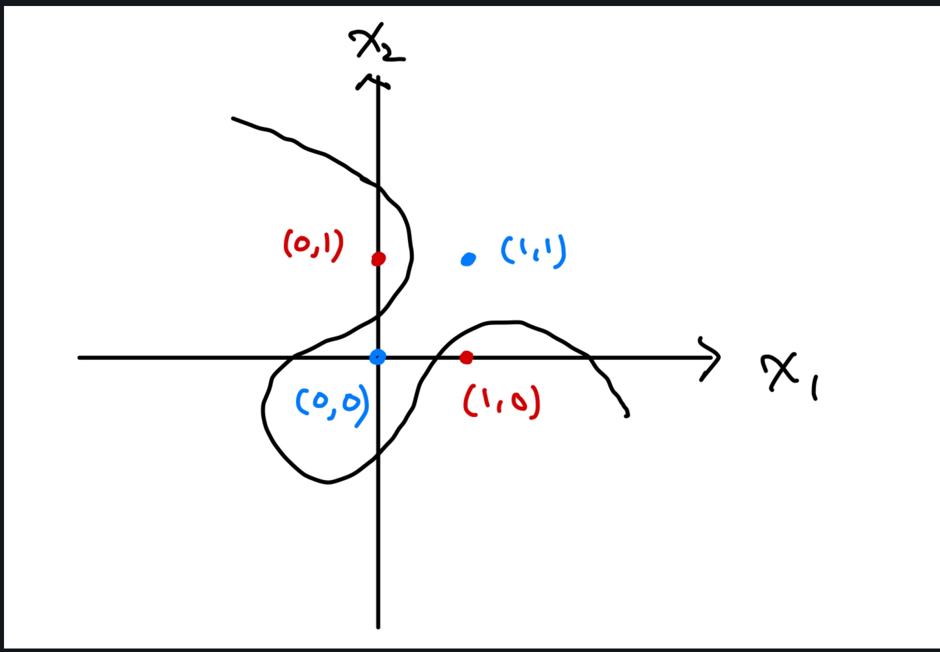


<NAND Gate>

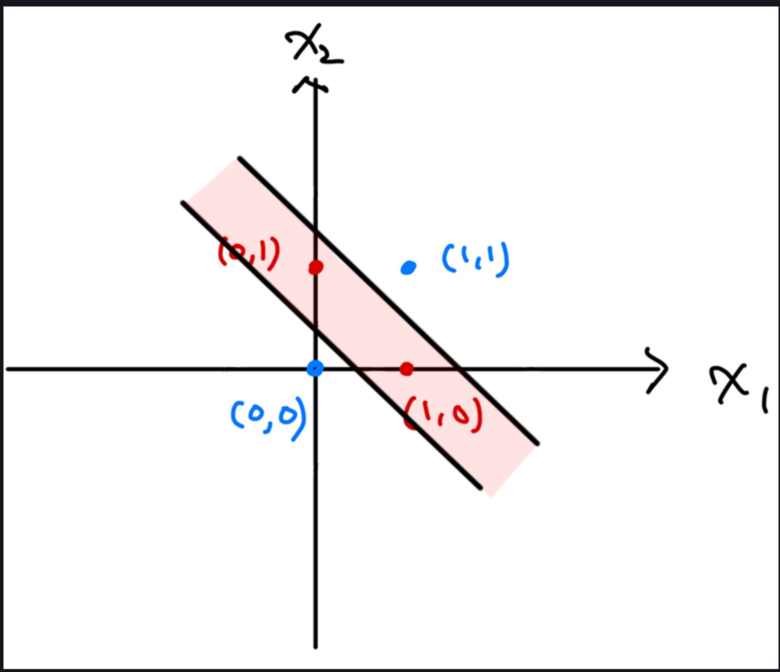
하지만 아래 그림은 퍼셉트론의 한계인 XOR 게이트와 같은 비선형 데이터는 분류하지 못한다.

XOR GATE

X_1	X_2	Y
1	1	0
1	0	1
0	1	1
0	0	0



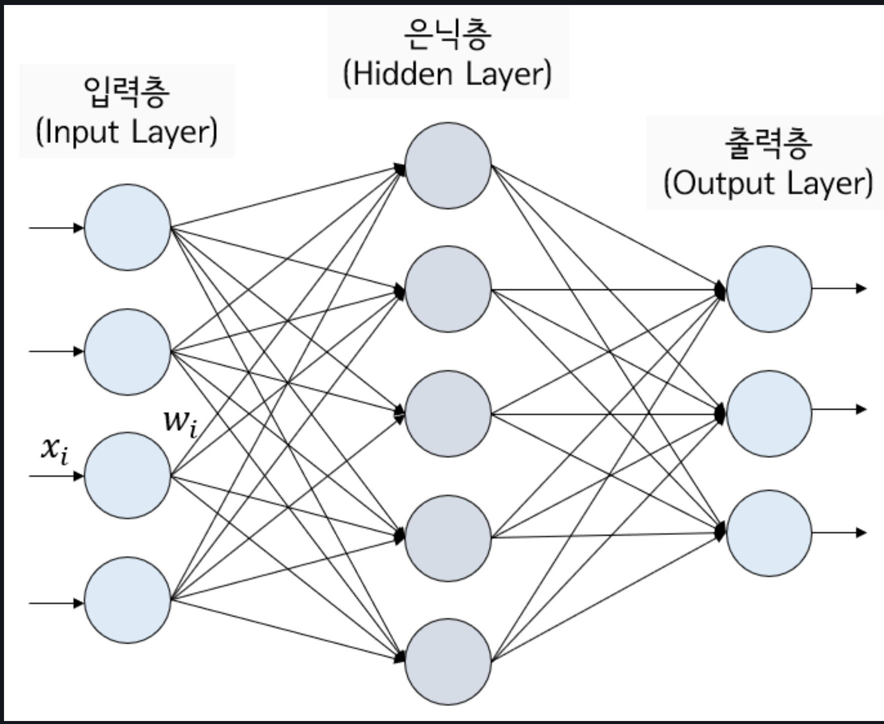
<XOR Gate>



<XOR Gate>

XOR와 같은 비선형 데이터를 분류하기 위해 층을 여러개 쌓은 MLP가 등장한다.

MLP(다층 퍼셉트론)



<MLP>

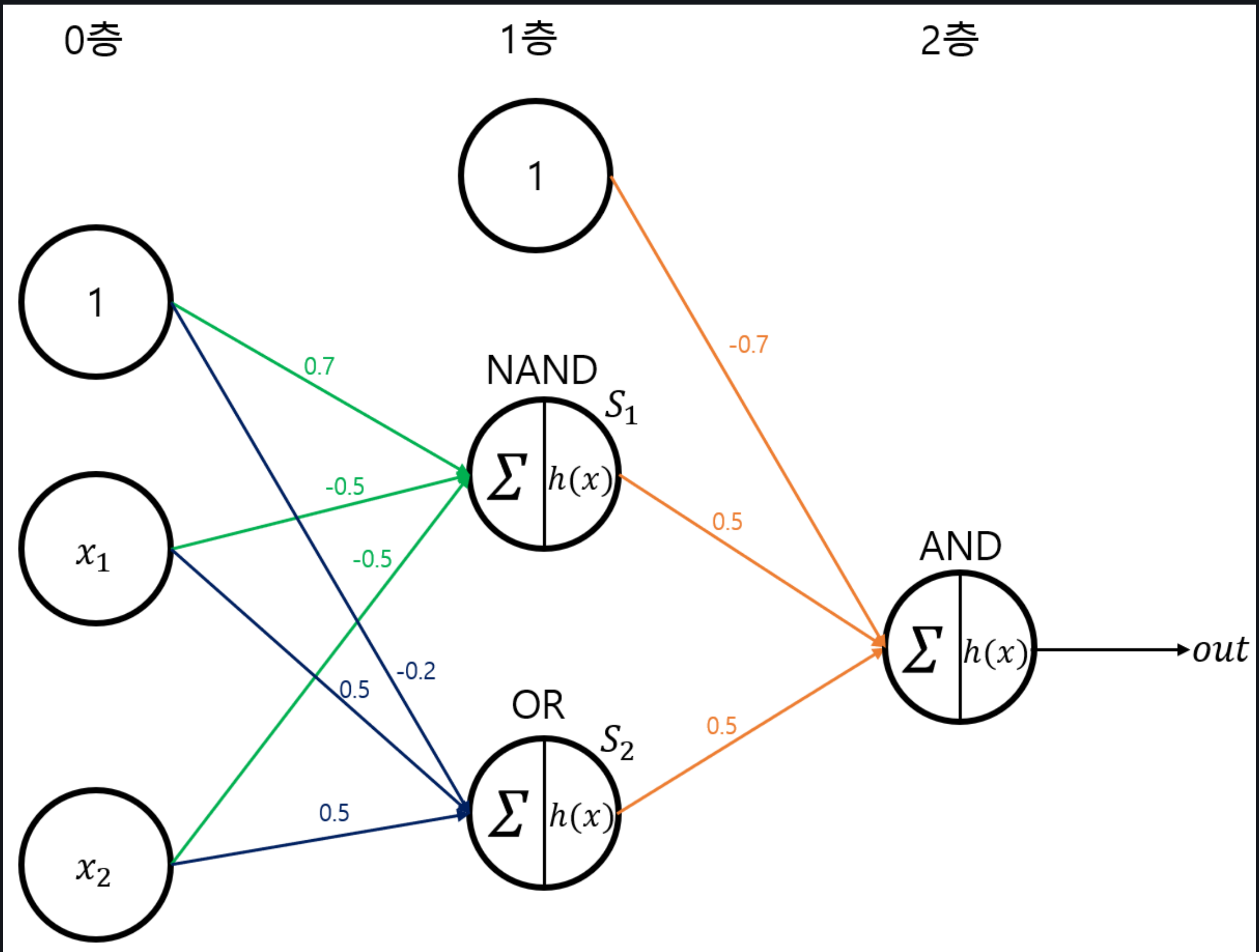
비선형 분류를 위해 **다층 퍼셉트론(Multi-Layer Perceptron)**이 등장한다. 다층 퍼셉트론은 비선형 분류기이다.

비선형 분류기란 두 개 이상의 직선 혹은 곡선의 영역으로 데이터들을 분류할 수 있는 모델을 말한다.

그럼 XOR GATE를 분류해보자.

방법은 앞에서 구한 NAND GATE와 OR GATE 그리고 AND GATE로 2층 퍼셉트론으로 구현하면 된다.

이해를 돕기 위해 아래 그림을 추가하겠다.



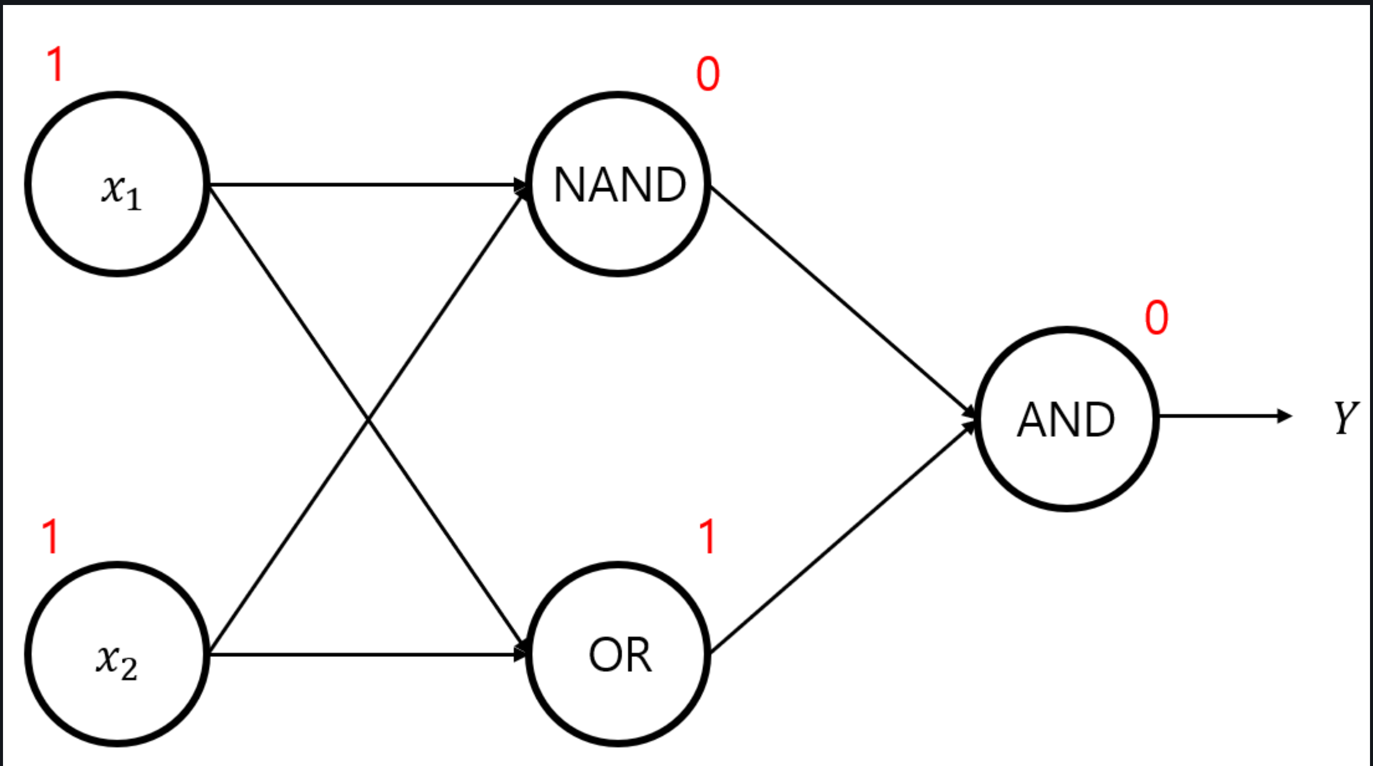
<XOR Gate>

이제 XOR의 input을 각각 넣어 보겠다.

$$\begin{gathered}
input : (1, 1) \\
s_1 = -0.5x - 0.5x + 0.7 \quad (NAND) \\
-0.5 - 0.5 + 0.7 = -0.3 \leq 0 \rightarrow 0 \\
s_1 = 0 \\
s_2 = 0.5x_1 + 0.5x_2 - 0.2 \quad (OR) \\
0.5 + 0.5 - 0.2 = 0.8 > 0 \rightarrow 1 \\
s_2 = 1 \\
input : (0, 1) \\
y = 0.5x_1 + 0.5x_2 - 0.7 \quad (AND) \\
0 + 0.5 - 0.7 = -0.2 \leq 0 \rightarrow 0
\end{gathered} \tag{30}$$

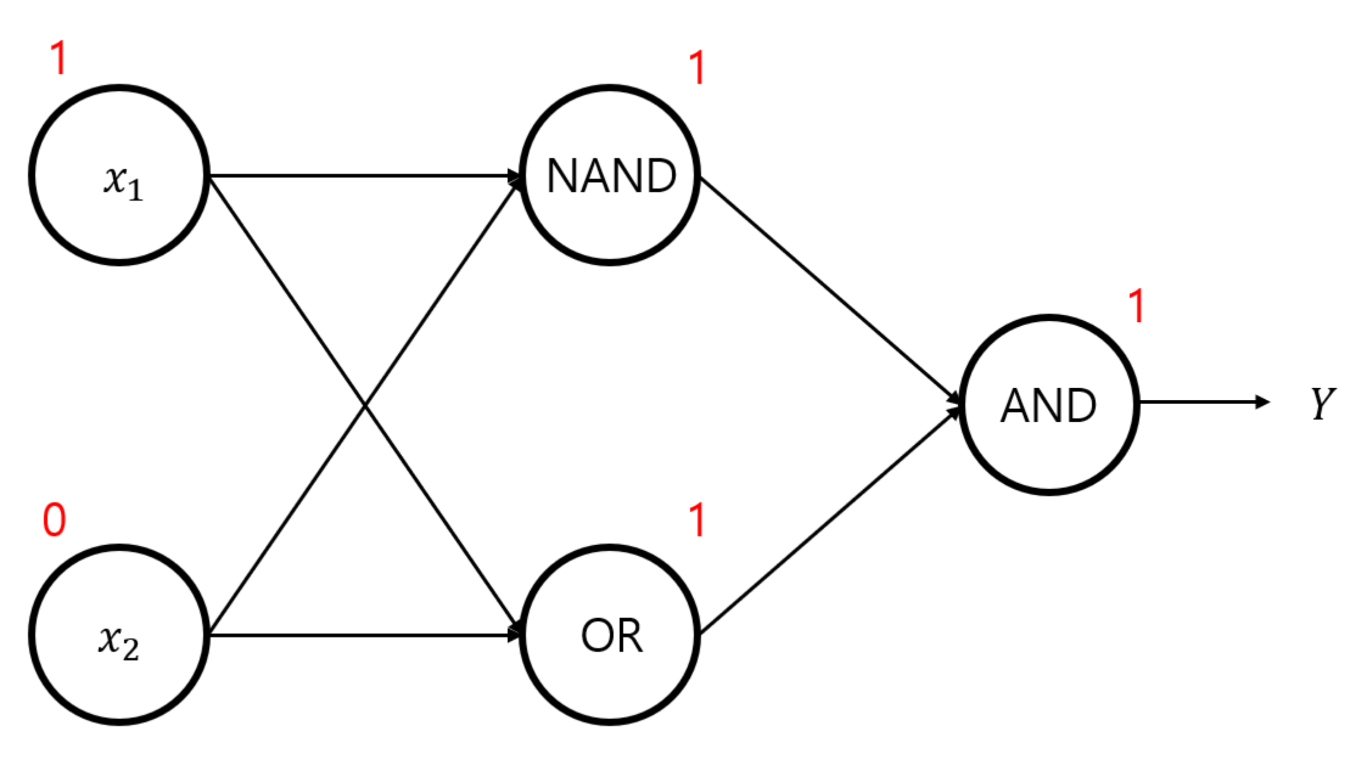
나머지 input도 위와 같이 연산하면 되니 앞으로 수식은 생략 후 아래 그림으로만 표현하겠다.

위 수식을 아래 그림으로 간략화해서 보면 다음과 같다.

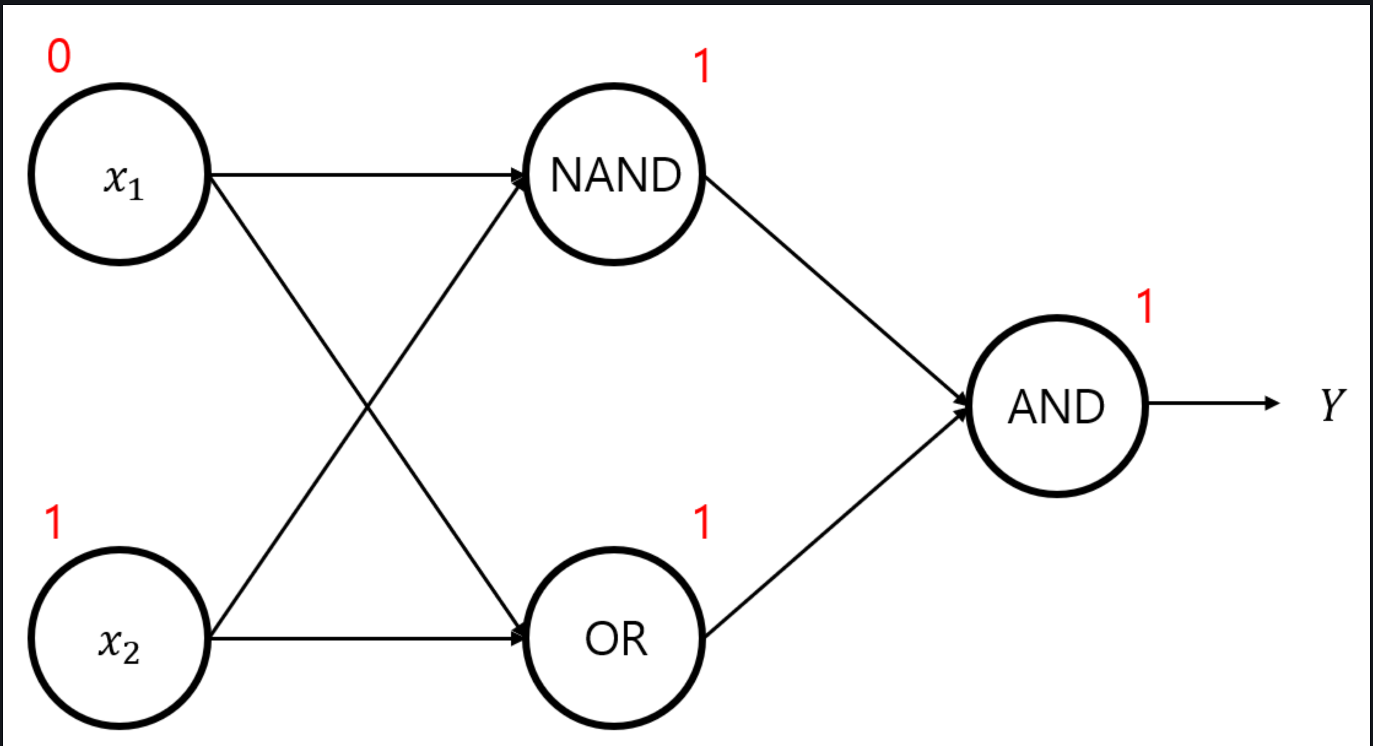


<XOR Gate>

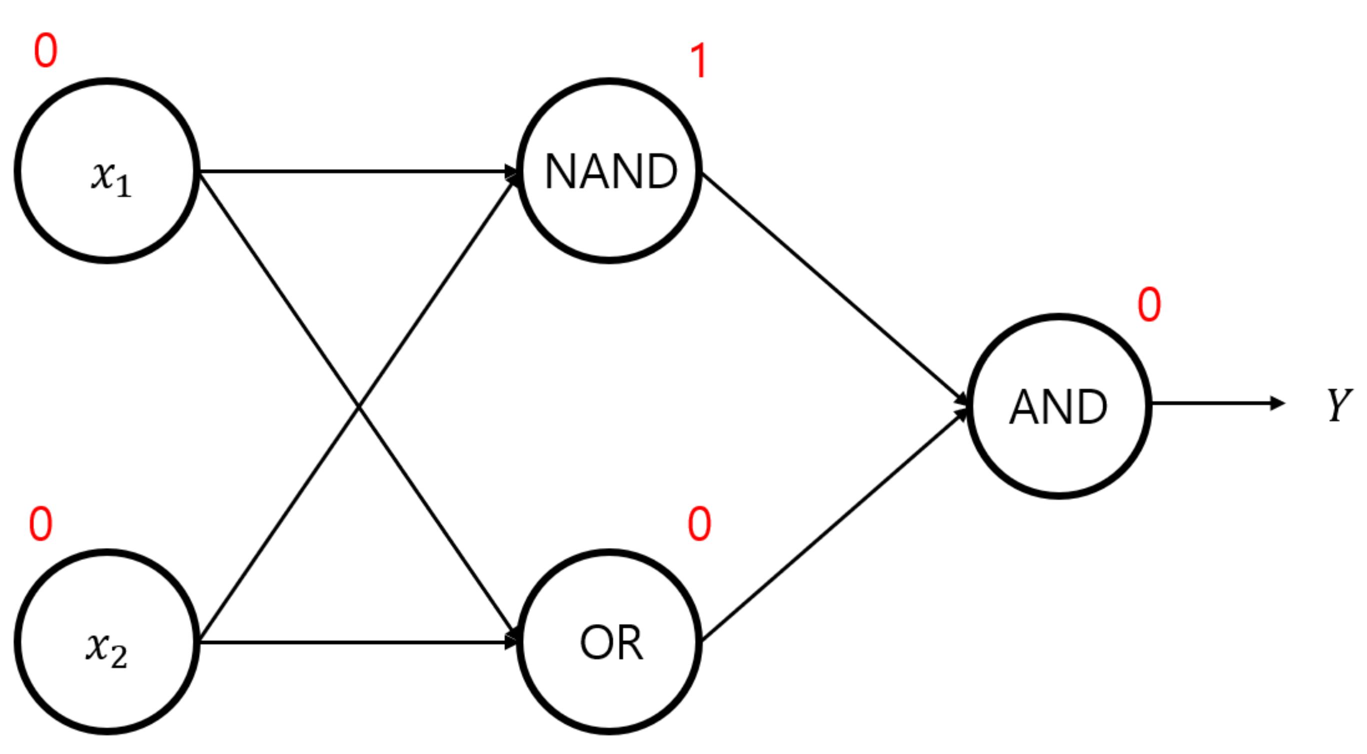
나머지 Input도 그림으로 보면,



<XOR Gate>

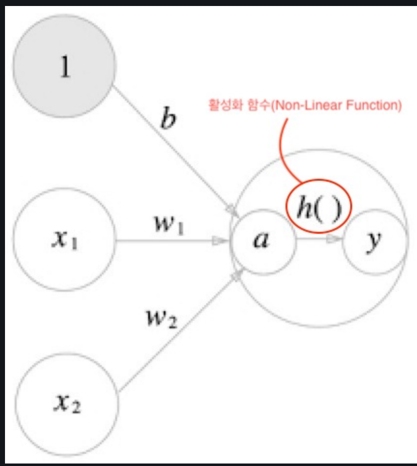


<XOR Gate>



<XOR Gate>

활성화 함수는 비선형 함수



<Activation Function>

활성화 함수는 이전 레이어에서의 신호의 총합을 다음 레이어의 뉴런에 전달할 때 해당 값을 비선형 형태로 변환해 주는 역할을 한다.

활성화 함수를 사용하지 않거나 선형함수를 사용하는 경우 은닉층을 쌓는 의미가 없어지기 때문에 결국 단층 퍼셉트론과 다층퍼셉트론의 차이가 없어진다.

예를들어 다층 퍼셉트론의 활성화 함수를 $h(x) = cx$라 하자. 활성화 함수의 input으로 어떤 y라는 값이 들어온다면 $h(y) = c \times y$가 되고 두 개의 층을 쌓게되면 $h(h(y)) = c(cy)$가 된다.

세개의 층인 경우 $h(h(h(y))) = c(c(cy)) = c^3y$

$a = c^3$이라 하면 결국 층이 하나만 있는 $h(y) = ay$인 퍼셉트론으로 표현될 수 있기 때문이다.

따라서 선형 분류만 가능하게 되므로 비선형 분류를 위한 다층 퍼셉트론은 활성화 함수가 반드시 비선형 함수여야 한다.

신경망

신경망을 알아보기전에 퍼셉트론에 대해 잠깐 이야기 하겠다.

퍼셉트론의 장점은 층을 쌓음으로써 복잡한 함수도 표현이 가능하다.

퍼셉트론의 단점은 매개변수(가중치와 편향)을 사람이 수동으로 설정해야 한다.

퍼셉트론의 활성화 함수로 계단함수를 사용하는데, 계단함수는 미분이 불가능함으로 미분을 통해 학습이 불가능하다.

하지만 신경망은 활성화 함수로 계단함수가 아닌 다른 활성화 함수(ex. sigmoid, Relu)를 사용하는 MLP이기 때문에
미분이 가능하며 매개변수(가중치와 편향)의 적절한 값을 데이터로부터 자동으로 학습한다.