

# 파이썬으로 배우는 딥러닝(Deep Learning)

1회차 수업  
논리 회로에서 신경망으로, 퍼셉트론 첫걸음

# 목 차

---

퍼셉트론

신경망

신경망학습

오차역전파법

학습관련기술들

합성곱신경망

전이학습과 ResNet

암석식별머신실습

퍼셉트론

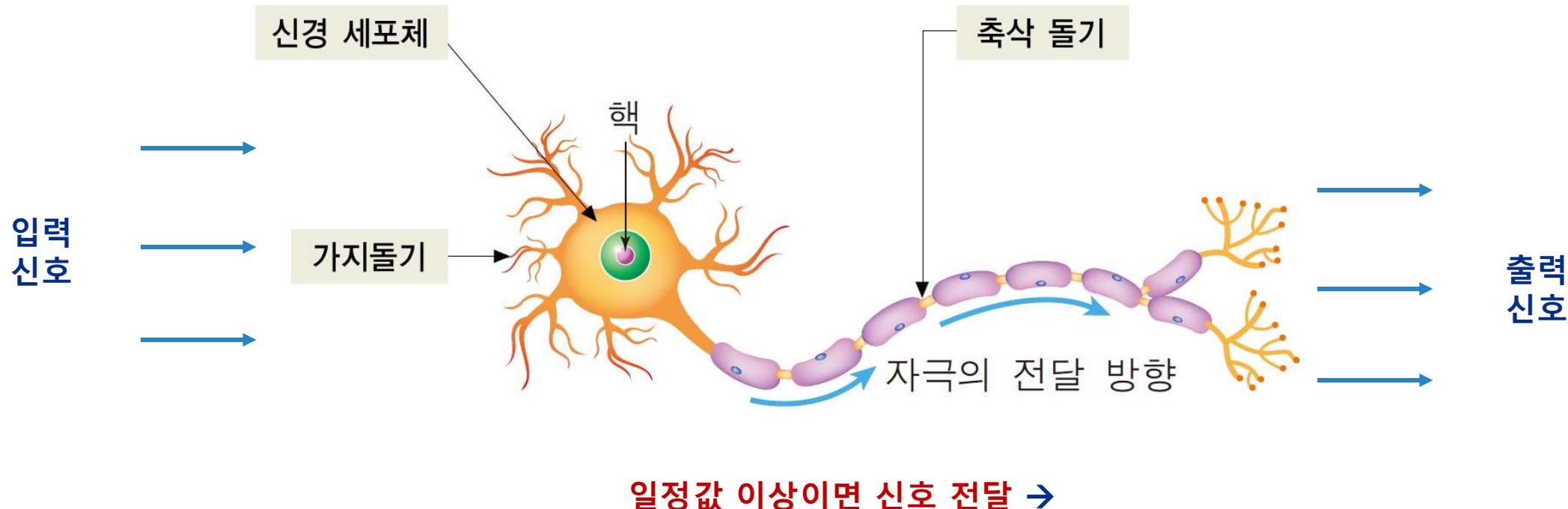
AND/OR/NAND Gate

XOR Gate

# 인간의 뉴런(Neuron)

인간의 뉴런(신경세포)은 감각을 통해 얻은 신호를 처리함.

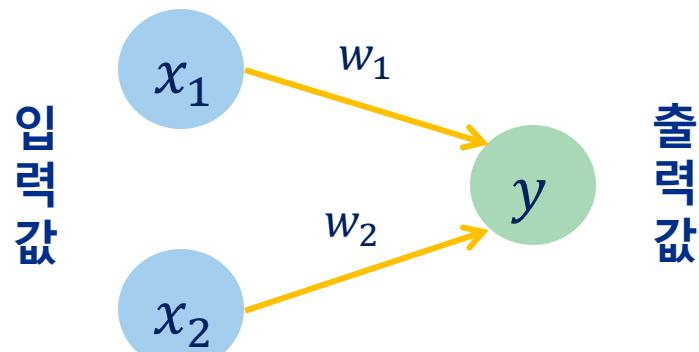
뉴런은 신호를 취합하여 일정 값 이상이면 다음 뉴런으로 신호를 전달함.



# 퍼셉트론

인간의 뉴런을 모방함

입력값(input)과 가중치(weight)가 선형 결합 되어 있음



$$y = \begin{cases} 0, & w_1x_1 + w_2x_2 \leq \theta \\ 1, & w_1x_1 + w_2x_2 > \theta \end{cases}$$

선형 결합의 결괏값과 주어진 임계값( $\theta$ )에 따라 출력값의 활성화(신호 전달) 여부를 정하는 모델

$\theta$  : 임계값  
 $b$  : bias, 편향

→ 간단한 분류 문제를 해결함

# 목 차

---

퍼셉트론

신경망

신경망학습

오차역전파법

학습관련기술들

합성곱신경망

전이학습과 ResNet

암석식별머신실습

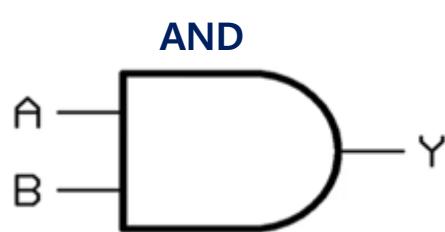
퍼셉트론

AND/OR/NAND Gate

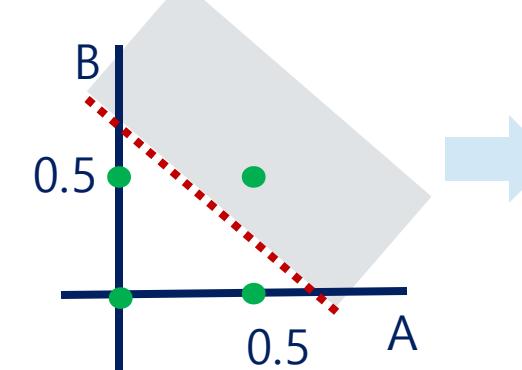
XOR Gate

# AND 게이트(1)

AND 게이트의 진리표 : 두입력이 모두 1 일 때만 1 을 출력하고, 그 외에는 0 을 출력



A	B	AND
0	0	0
0	1	0
1	0	0
1	1	1



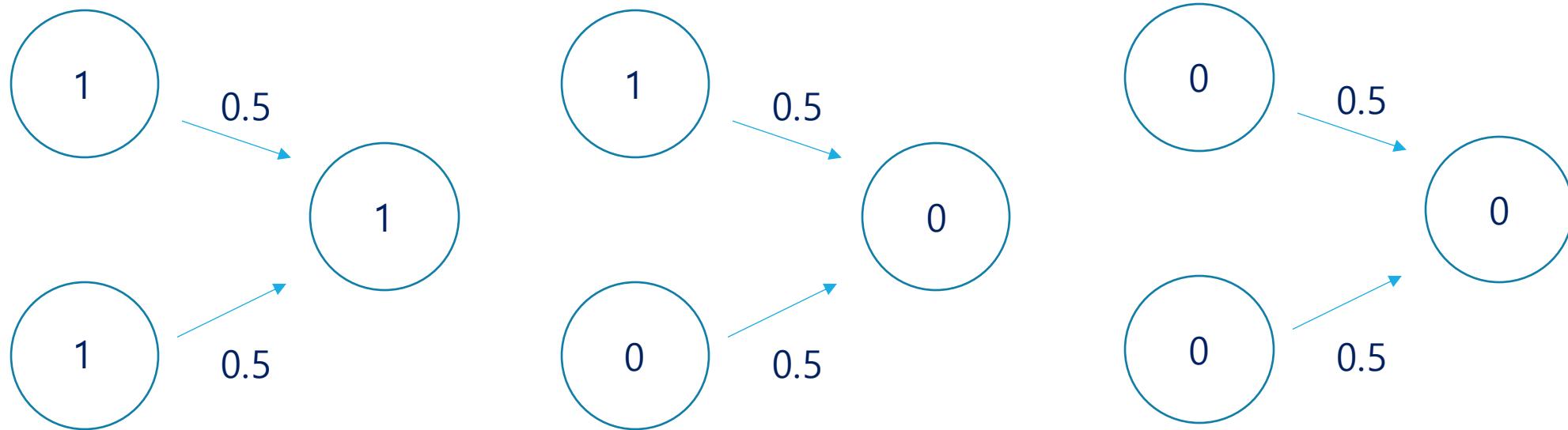
$$y = \begin{cases} 0, & w_1x_1 + w_2x_2 \leq \theta \\ 1, & w_1x_1 + w_2x_2 > \theta \end{cases}$$

AND 연산에 대해  
선형분류를 수행.

예시 :  $w_1, w_2 = 0.5, \theta = 0.7$

## AND 게이트(2)

Perceptron으로 AND 게이트를 표현해보자



$$0.5 \times 1 + 0.5 \times 1 = 1 > 0.7$$

$$0.5 \times 1 + 0.5 \times 0 = 0.5 \leq 0.7$$

$$0.5 \times 0 + 0.5 \times 0 = 0 \leq 0.7$$

$w_1 = 0.5, w_2 = 0.5, \theta = 0.7$ 로 잡아보자

AND 게이트를 Perceptron으로 구현한 Python 코드

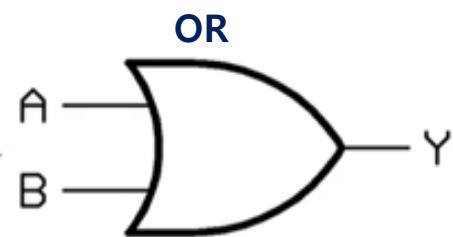
```
1 # coding: utf-8
2 import numpy as np
3
4
5 def AND(x1, x2):
6     x = np.array([x1, x2]) ← 초기 가중치 값 설정
7     w = np.array([0.5, 0.5]) ← 초기 가중치 값 설정
8     b = -0.7 ← 한계 값 설정
9     tmp = np.sum(w*x) + b
10    if tmp <= 0:
11        return 0 ← 한계값이하이면 FALSE
12    else:
13        return 1 ← 한계값이상이면 TRUE
14
15 if __name__ == '__main__':
16     for xs in [(0, 0), (1, 0), (0, 1), (1, 1)]:
17         y = AND(xs[0], xs[1])
18         print(str(xs) + " -> " + str(y))
19
```

<실습 과제>

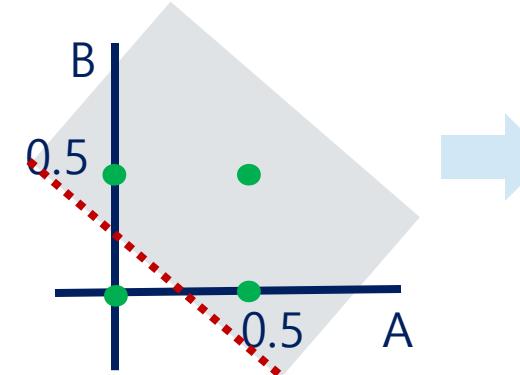
ch01/and\_gate.ipynb를  
실행해보자

# OR 게이트(1)

OR 게이트의 진리표 : 두 입력중 하나라도 1 일 때 1 을 출력하고, 그 외에는 0 을 출력



A	B	OR
0	0	0
0	1	1
1	0	1
1	1	1

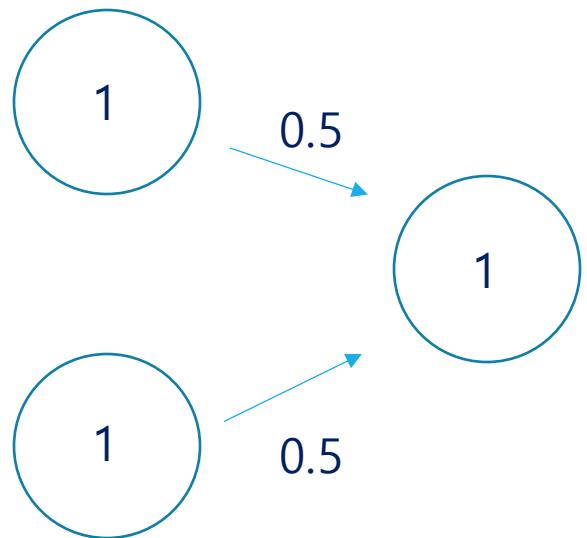


OR 연산에 대해  
선형분류를 수행.

예시 :  $w_1, w_2 = 0.5, \theta = 0.2$

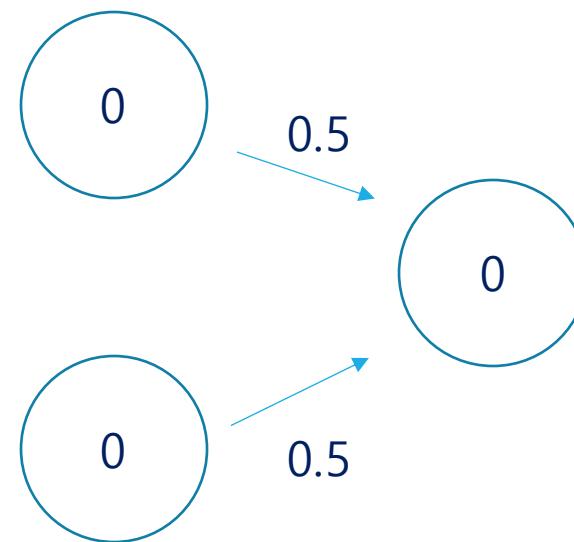
## OR 게이트(2)

Perceptron으로 OR 게이트를 표현해보자



$$0.5 \times 1 + 0.5 \times 1 = 1 > 0.2$$

$$0.5 \times 1 + 0.5 \times 0 = 0.5 > 0.2$$



$$0.5 \times 0 + 0.5 \times 0 = 0 \leq 0.2$$

$w_1 = 0.5, w_2 = 0.5, \theta = 0.2$ 로 잡아보자

<실습 과제>

Perceptron으로

OR 게이트를 구현해보자

참고> ch01/or\_gate.ipynb 에 작성

OR 게이트를 Perceptron으로 구현한 Python 코드

<예제> or\_gate.ipynb

```
1 # coding: utf-8
2 import numpy as np
3
4
5 def OR(x1, x2):
6     x = np.array([x1, x2])
7     w = np.array([0.5, 0.5])
8     b = -0.2
9     tmp = np.sum(w*x) + b
10    if tmp <= 0:
11        return 0
12    else:
13        return 1
14
15 if __name__ == '__main__':
16     for xs in [(0, 0), (1, 0), (0, 1), (1, 1)]:
17         y = OR(xs[0], xs[1])
18         print(str(xs) + " -> " + str(y))
```

초기 가중치 값 설정

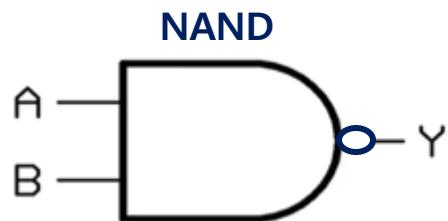
한계 값 설정

한계값이하이면 FALSE

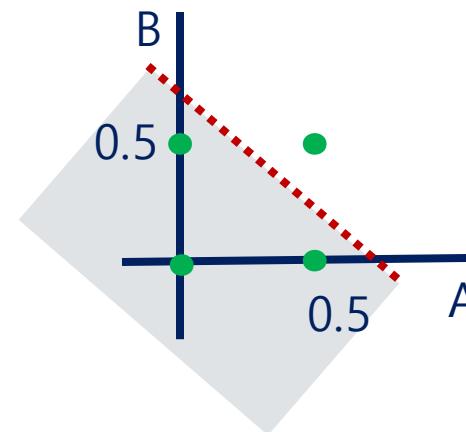
한계값이상이면 TRUE

# NAND 게이트(1)

NAND 게이트의 진리표 : AND게이트의 정 반대 결과를 출력



A	B	NAND
0	0	1
0	1	1
1	0	1
1	1	0



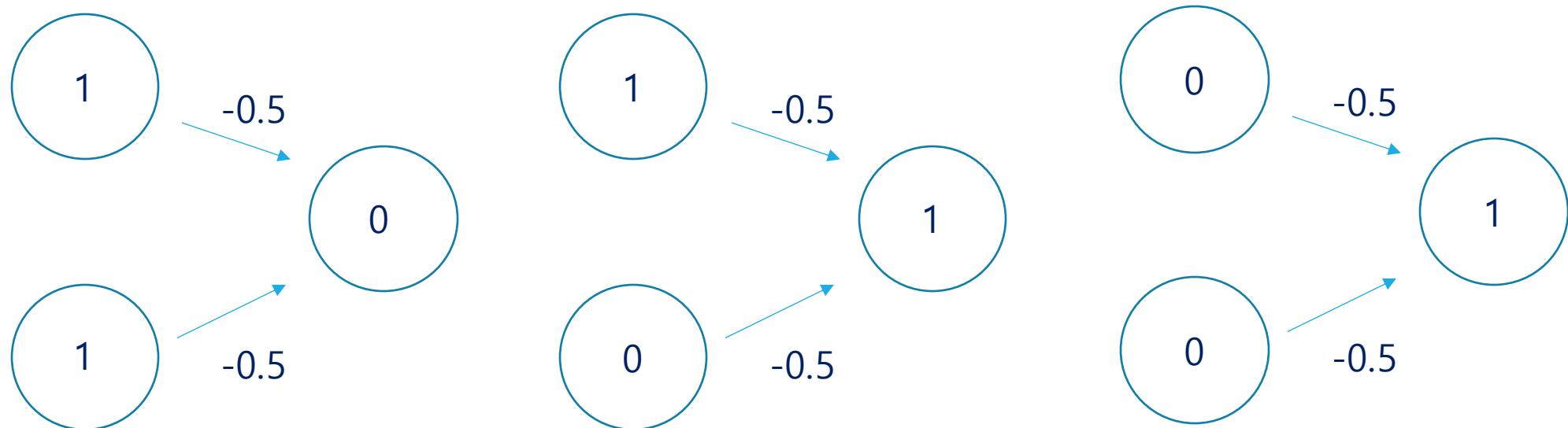
$$y = \begin{cases} 0, & w_1x_1 + w_2x_2 \leq \theta \\ 1, & w_1x_1 + w_2x_2 > \theta \end{cases}$$

NAND 연산에 대해  
선형분류를 수행.

예시 :  $w_1, w_2 = -0.5, \theta = -0.7$

## NAND 게이트(2)

Perceptron으로 NAND 게이트를 표현해보자



$$-0.5 \times 1 - 0.5 \times 1 = -1 \leq -0.7 \quad -0.5 \times 1 - 0.5 \times 0 = -0.5 > -0.7 \quad -0.5 \times 0 - 0.5 \times 0 = 0 > -0.7$$

$w_1 = -0.5, w_2 = -0.5, \theta = -0.7$ 로 잡아보자

<실습 과제>

Perceptron으로

NAND 게이트를 구현해보자

참고> ch01/and\_gate.ipynb 코드에 작성

NAND 게이트를 Perceptron으로 구현한 Python 코드

## <예제> nand\_gate.ipynb

```
1 # coding: utf-8
2 import numpy as np
3
4
5 def NAND(x1, x2):
6     x = np.array([x1, x2])
7     w = np.array([-0.5, -0.5])           ← 초기 가중치 값 설정
8     b = 0.7                            ← 한계 값 설정
9     tmp = np.sum(w*x) + b
10    if tmp <= 0:
11        return 0                      ← 한계값이하이면 FALSE
12    else:
13        return 1                      ← 한계값이상이면 TRUE
14
15 if __name__ == '__main__':
16     for xs in [(0, 0), (1, 0), (0, 1), (1, 1)]:
17         y = NAND(xs[0], xs[1])
18         print(str(xs) + " -> " + str(y))
19
```

# 목 차

---

퍼셉트론

신경망

신경망학습

오차역전파법

학습관련기술들

합성곱신경망

전이학습과 ResNet

암석식별머신실습

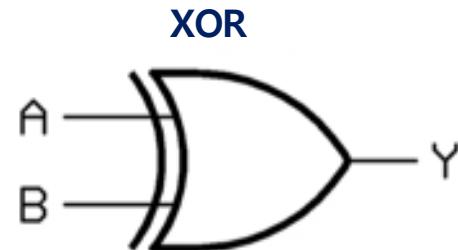
퍼셉트론

AND/OR/NAND Gate

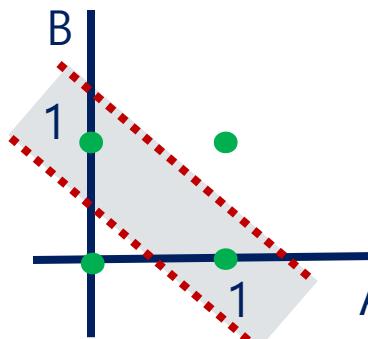
XOR Gate

# XOR 게이트(1) : 문제의 시작

NAND 게이트의 진리표 : 둘 중 하나만 1일 경우 1을 출력하고, 그 외에는 0을 출력한다.



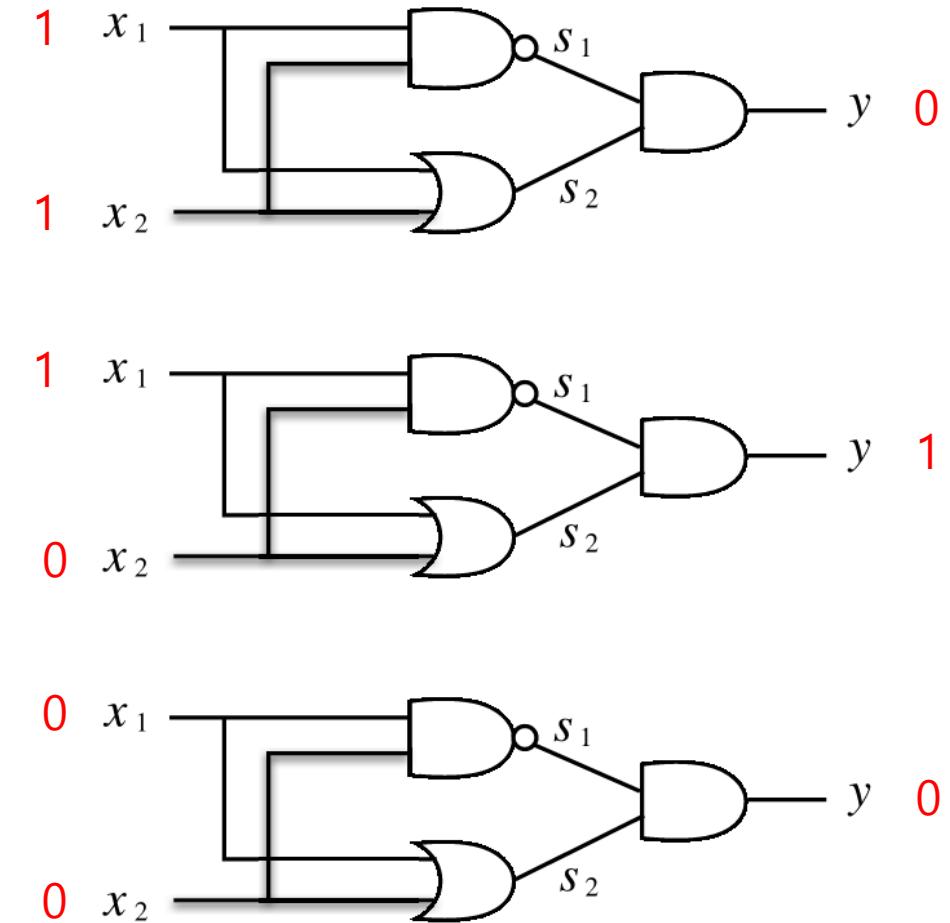
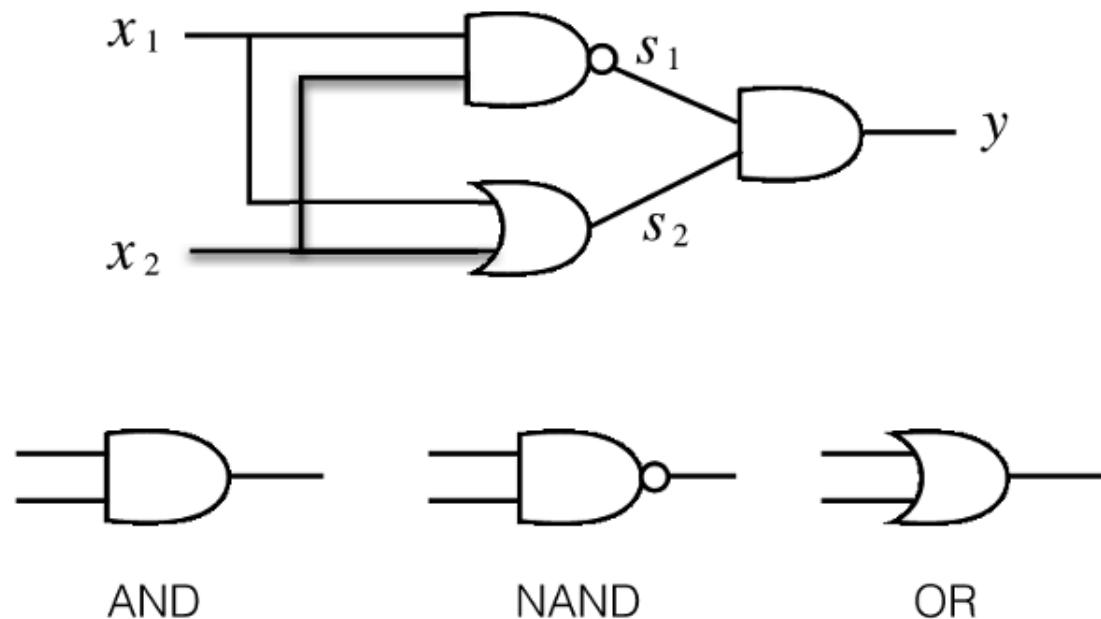
A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0



XOR 연산에 대해  
선형분류가 동작하지 못함

## XOR 게이트(2)

퍼셉트론의 문제인 XOR 연산은 AND, OR, NOT 연산자의 조합으로 만들 수 있음.



<실습 과제>

Perceptron으로  
XOR 게이트를 구현해보자

참고> ch01/xor\_gate.ipynb 코드에 작성

XOR 게이트를 Perceptron으로 구현한 Python 코드

```

1 # coding: utf-8
2 import numpy as np
3
4 def AND(x1, x2):
5     x = np.array([x1, x2])
6     w = np.array([0.5, 0.5])
7     b = -0.7
8     tmp = np.sum(w*x) + b
9     if tmp <= 0:
10        return 0
11    else:
12        return 1
13
14 def OR(x1, x2):
15     x = np.array([x1, x2])
16     w = np.array([0.5, 0.5])
17     b = -0.2
18     tmp = np.sum(w*x) + b
19     if tmp <= 0:
20        return 0
21    else:
22        return 1
23
24 def XOR(x1, x2):
25     s1 = NAND(x1, x2)
26     s2 = OR(x1, x2)
27     y = AND(s1, s2)
28     return y
29
30
31 def NAND(x1, x2):
32     x = np.array([x1, x2])
33     w = np.array([-0.5, -0.5])
34     b = 0.7
35     tmp = np.sum(w*x) + b
36     if tmp <= 0:
37        return 0
38    else:
39        return 1
40
41 if __name__ == '__main__':
42     for xs in [(0, 0), (1, 0), (0, 1), (1, 1)]:
43         y = XOR(xs[0], xs[1])
44         print(str(xs) + " -> " + str(y))
45
46

```

3개 논리회로의 조합  
으로 XOR 구현