

# 밑바닥부터 시작하는 딥러닝

## Chap2. 퍼셉트론

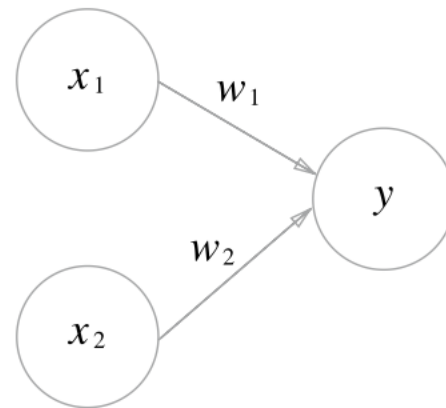
박은우

## 2.1 퍼셉트론이란?

- 신경망(딥러닝)의 기원이 되는 알고리즘
  - 다수의 신호를 입력으로 받아 하나의 신호를 출력
  - 퍼셉트론 신호: 흐른다/안 흐른다(0/1)
  - 신호의 총합이 정해진 한계(임계값,  $\theta$ )를 넘을 때만 1출력
- 뉴런이 활성화 한다

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

그림 2-1 입력이 2개인 퍼셉트론



## 2.2 단순한 논리 회로

### 2.2.1 AND 게이트

그림 2-2 AND 게이트의 진리표

$x_1$	$x_2$	$y$
0	0	0
1	0	0
0	1	0
1	1	1

-> 둘 다 1이어야 1이 된다

## 2.2.2 NAND 게이트와 OR 게이트

그림 2-3 NAND 게이트의 진리표

$x_1$	$x_2$	$y$
0	0	1
1	0	1
0	1	1
1	1	0

그림 2-4 OR 게이트의 진리표

$x_1$	$x_2$	$y$
0	0	0
1	0	1
0	1	1
1	1	1

->AND의 반대(둘 다 1이어야 0)

## 2.3 퍼셉트론 구현하기

### 2.3.1 간단한 구현부터

```
def AND(x1,x2):  
    w1,w2,theta=0.5,0.5,0.7 #초기화  
    tmp=x1*w1+x2*w2  
    if tmp <=theta:  
        return 0  
    elif tmp >theta: #임계값 넘을 시 1반환  
        return 1
```

```
print(AND(0,0)) #0  
print(AND(1,0)) #0  
print(AND(0,1)) #0  
print(AND(1,1)) #1
```

```
0  
0  
0  
1
```

## 2.3.2 가중치와 편향 도입

$\theta \rightarrow -b(\text{bias, 편향})$

$$y = \begin{cases} 0 & (b + w_1x_1 + w_2x_2 \leq 0) \\ 1 & (b + w_1x_1 + w_2x_2 > 0) \end{cases}$$

```
x=np.array([0,1])  
w=np.array([0.5,0.5])  
b=-0.7  
w*x
```

```
array([0. , 0.5])
```

```
np.sum(w*x)
```

```
0.5
```

```
np.sum(w*x)+b
```

```
-0.19999999999999996
```

### 2.3.3 가중치와 편향 구현하기

가중치  $\neq$  편향

가중치: 각 입력 신호가 결과에 주는 영향력(중요도) 조절 매개변수

편향: 뉴런이 얼마나 쉽게 활성화(임계값을 넘느냐)하느냐를 조정하는 매개변수

Ex)  $b = -0.1 \rightarrow w \cdot x$ 의 합이 0.1 초과해야만 뉴런 활성화

```
def AND(x1,x2):
    x=np.array([x1,x2])
    w=np.array([0.5,0.5])
    b=-0.7
    tmp=np.sum(w*x)+b
    if tmp <=0:
        return 0
    else:
        return 1
```

```
: def NAND(x1,x2):
    x=np.array([x1,x2])
    w=np.array([-0.5,-0.5]) #AND와는 w,b만 다르다
    b=0.7
    tmp=np.sum(w*x)+b
    if tmp<=0:
        return 0
    else:
        return 1

def OR(x1,x2):
    x=np.array([x1,x2])
    w=np.array([0.5,0.5]) #AND랑 b만 다르다
    b=-0.2
    tmp=np.sum(w*x)+b
    if tmp<=0:
        return 0
    else: return 1
```



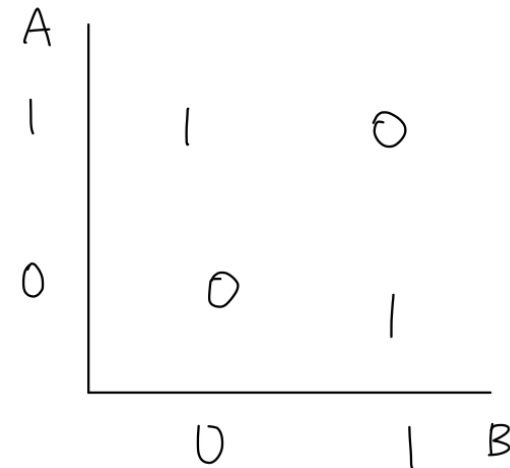
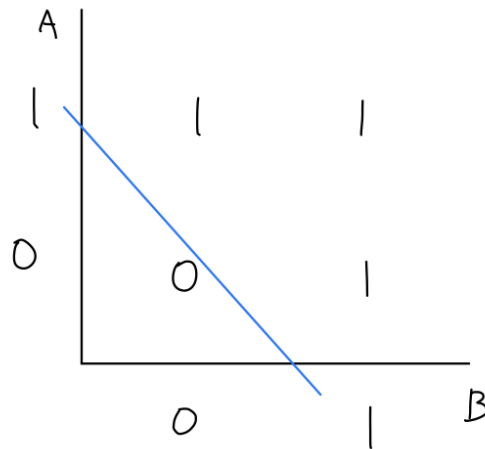
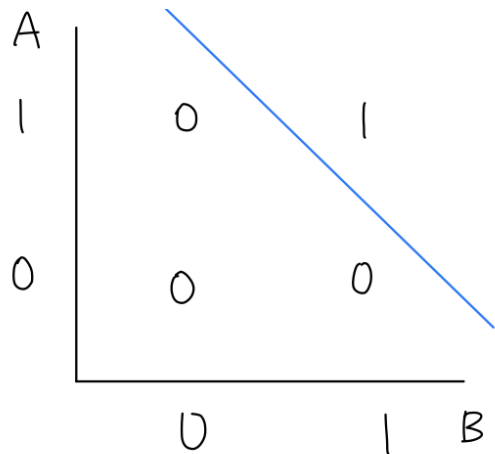
## 2.4 퍼셉트론의 한계

### 2.4.1 도전! XOR 게이트

XOR 게이트 -> 배타적 논리합. 서로 다른 값->1, 같은 값->0

**\*문제점\***

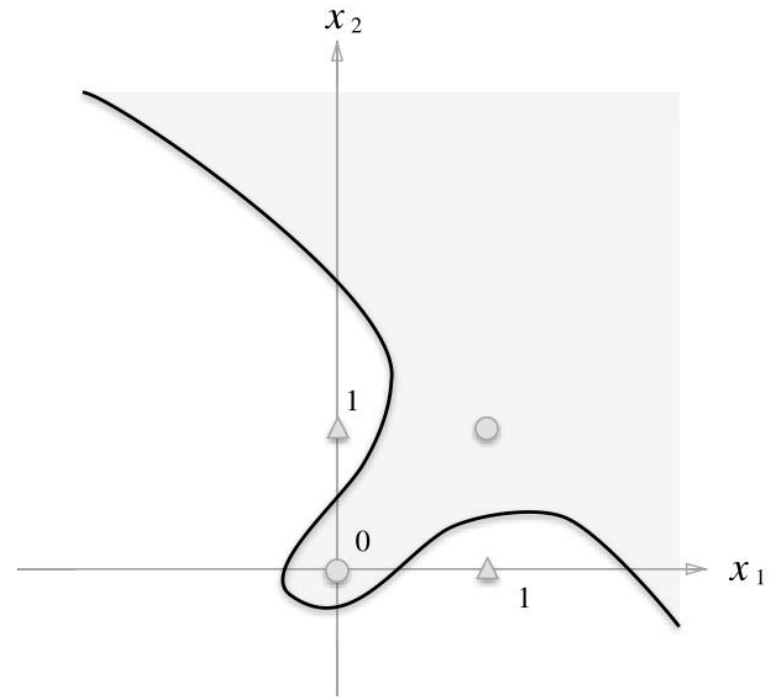
지금까지 본 퍼셉트론으로는 XOR 게이트 구현 불가!



## 2.4.2 선형과 비선형

XOR -> 직선 하나로 분류 제대로 할 수 x  
=> **'직선'이 아니라면?**

곡선의 영역 = 비선형 영역  
직선의 영역 = 선형 영역



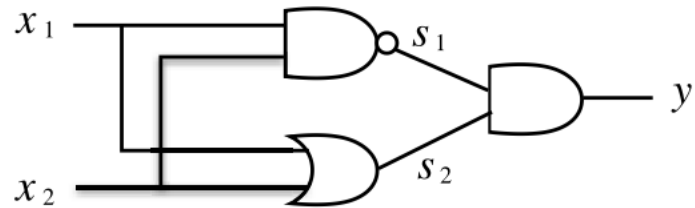
## 2.5 다층 퍼셉트론이 출동한다면

### 2.5.1 기존 게이트 조합하기

XOR 게이트 만드는 방법

-> **AND, NAND, OR 게이트 조합**

그림 2-11 AND, NAND, OR 게이트를 조합해 구현한 XOR 게이트



$x_1$	$x_2$	$s_1$	$s_2$	$Y$
0	0	1	0	0
1	0	1	1	1
0	1	1	1	1
1	1	0	1	0

## 2.5.2 XOR 게이트 구현하기

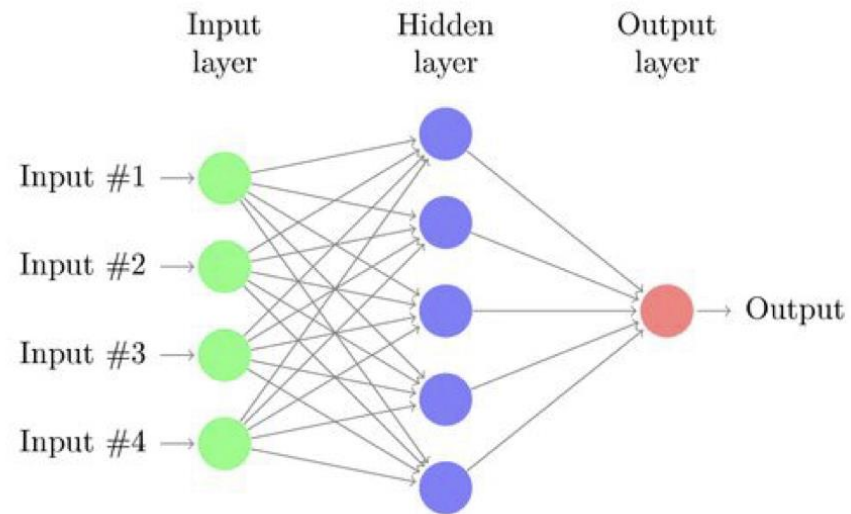
```
def XOR(x1,x2):  
    s1=NAND(x1,x2)  
    s2=OR(x1,x2)  
    y=AND(s1,s2)  
    return y
```

```
print(XOR(0,0)) #0  
print(XOR(0,1)) #1  
print(XOR(1,0)) #1  
print(XOR(1,1)) #0
```

0  
1  
1  
0

층이 여러 개인 퍼셉트론 = **다층 퍼셉트론**

-> 단층 퍼셉트론으로는 표현 $x$ 를 층을 하나 늘려 구현



## 2.6 NAND에서 컴퓨터까지

NAND 게이트의 조합만으로 컴퓨터가 수행하는 일 재현 가능

-> '퍼셉트론으로 컴퓨터를 표현할 수 있다'

-> 어떻게?

⇒ 비선형인 **시그모이드 함수** 사용-> 임의의 함수 표현 가능

⇒ 점점 단계적으로 크게 만들어가면 가능!