Deep Learning from Scratch
chapter 2

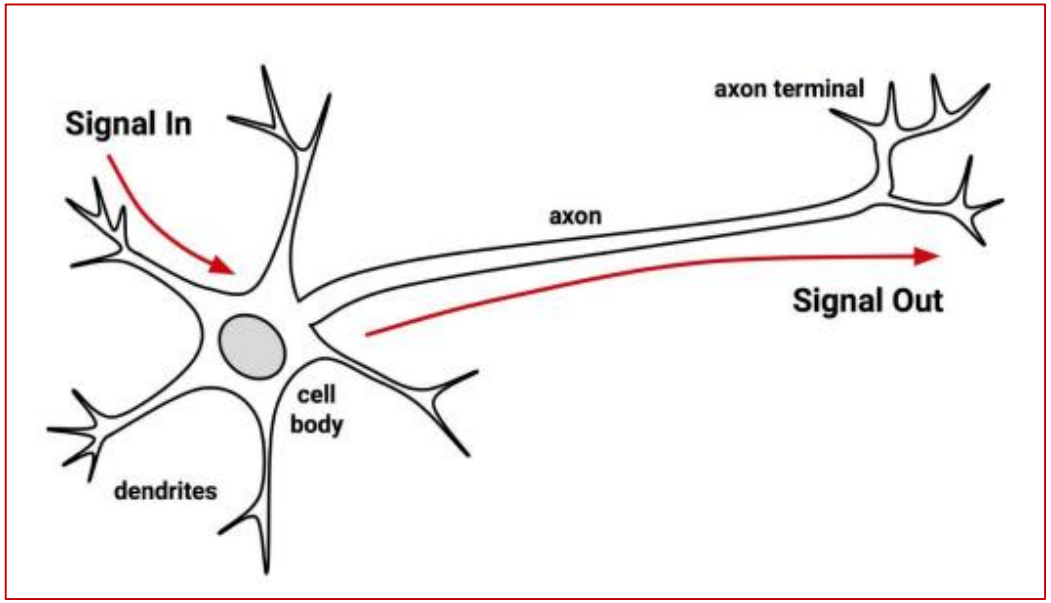
이새봄

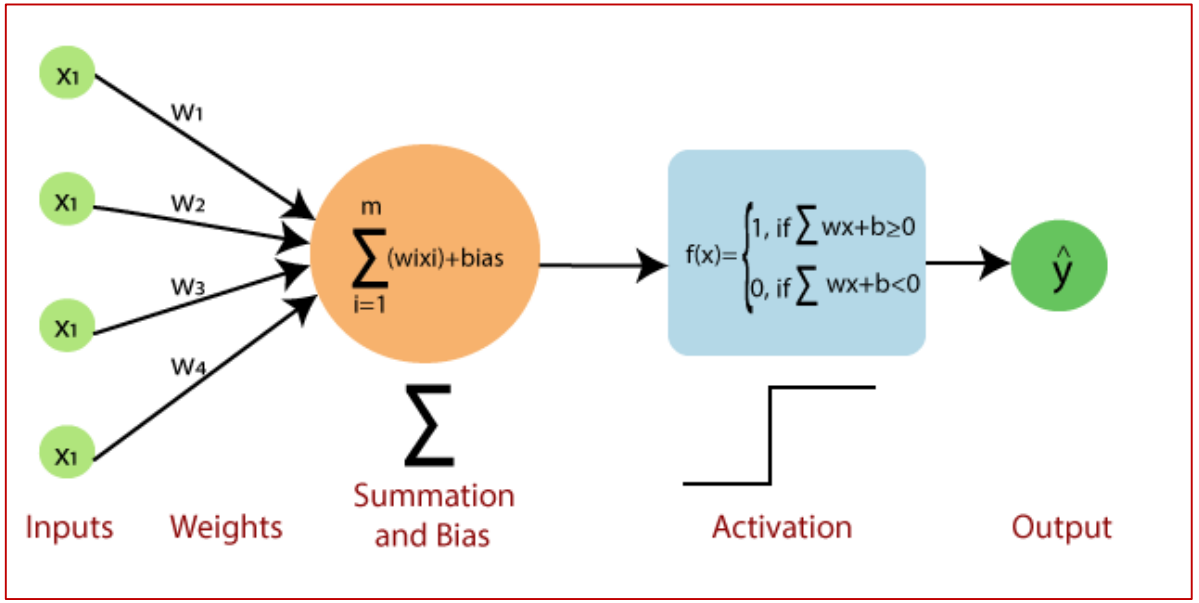
Ⅰ. Why Python?

- 간단하고 쉬움
- 오픈소스-> 무료
- 컴파일 과정 X
- 뛰어난 성능

패키지 설치 및 관리
부분 용이

Ⅱ. Perceptron



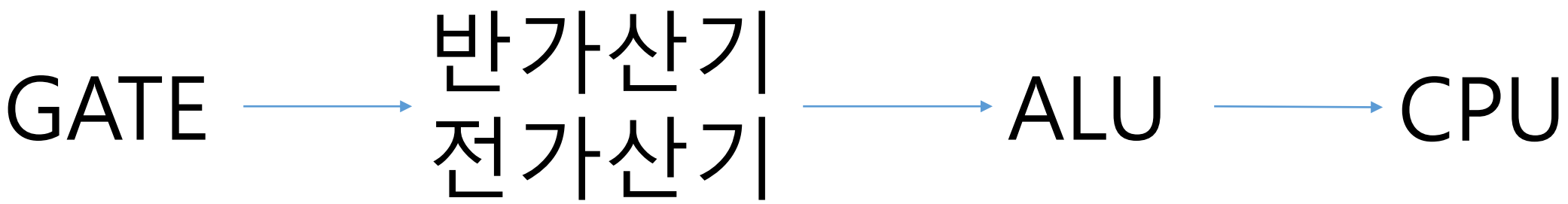


$$y=\begin{cases}0\ (w_1x_1+w_2x_2\leq(\text{임계값}))\\1\ (w_1x_1+w_2x_2>(\text{임계값}))\end{cases}$$

Biological Neuron

Artifical Nueral Network

Ⅱ. Perceptron



Ⅱ. Perceptron

▼ AND 진리표

입력		연산자	출력
x_1	x_2		y
0	0	AND	0
0	1		0
1	0		0
1	1		1

▼ OR 진리표

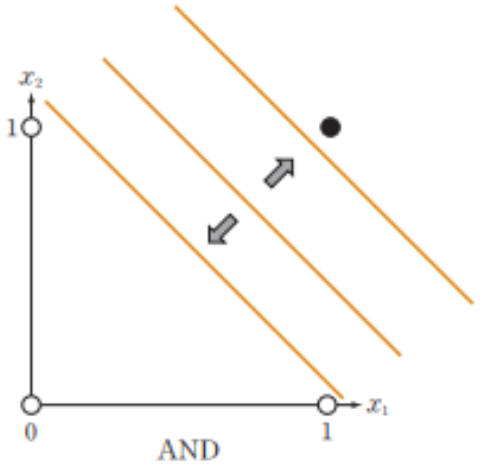
입력		연산자	출력
x_1	x_2		y
0	0	OR	0
0	1		1
1	0		1
1	1		1

▼ NAND 진리표

입력		연산자	출력
x_1	x_2		y
0	0	NAND	1
0	1		1
1	0		1
1	1		0

▼ 퍼셉트론식을 적용한 AND 진리표

입력		퍼셉트론 $y=\begin{cases}0\ (w_1x_1+w_2x_2\leq(\text{임계값}))\\1\ (w_1x_1+w_2x_2>(\text{임계값}))\end{cases}$ $(w_1, w_2, \text{임계값})=(0.2, 0.2, 0.3)$	출력
x_1	x_2		y
0	0	$0\times0.2+0\times0.2\leq0.3$	0
0	1	$0\times0.2+1\times0.2\leq0.3$	0
1	0	$1\times0.2+0\times0.2\leq0.3$	0
1	1	$1\times0.2+1\times0.2>0.3$	1



▲ 퍼셉트론의 범위

Ⅱ. Perceptron

```
In [5]: def AND(x1,x2):
            w1,w2,theta=0.5,0.5,0.7
            tmp = x1*w1 + x2*w2
            if tmp<=theta:
                return 0
            elif tmp > theta:
                return 1

In [6]: AND(0,0)
Out[6]: 0

In [7]: AND(1,0)
Out[7]: 0

In [8]: AND(0,1)
Out[8]: 0

In [9]: AND(1,1)
Out[9]: 1
```

```
In [23]: def AND(x1,x2):
             x=np.array([x1,x2])
             y=np.array([0.5,0.5])
             b=-0.7
             tmp = np.sum(w*x)+b
             if tmp<=0:
                 return 0
             else:
                 return 1

In [24]: AND(0,0)
Out[24]: 0

In [25]: AND(1,0)
Out[25]: 0

In [26]: AND(0,1)
Out[26]: 0

In [27]: AND(1,1)
Out[27]: 1
```

기존 퍼셉트론의 식

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

⇨

변형된 퍼셉트론의 식

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

가중치:입력신호가 결과에 주는 영향력을 조절

편향: 뉴런이 얼마나 쉽게 활성화하느냐를 조정

Ⅱ. Perceptron

In [28]:
```
def NAND(x1,x2):
    x=np.array([x1,x2])
    y=np.array([-0.5,-0.5])
    b= 0.7
    tmp = np.sum(w*x)+b
    if tmp<=0:
        return 0
    else:
        return 1
```

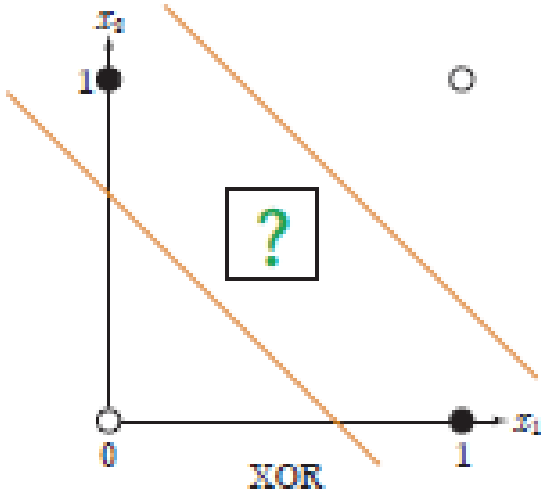
In [29]:
```
def OR(x1,x2):
    x=np.array([x1,x2])
    y=np.array([0.5,0.5])
    b= -0.2
    tmp = np.sum(w*x)+b
    if tmp<=0:
        return 0
    else:
        return 1
```

가중치와 편향만 조절

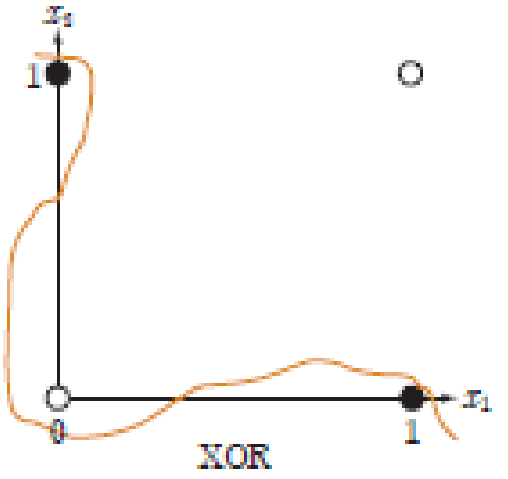
Ⅱ. Perceptron

▼ XOR 진리표

입력		연산자	출력
x_1	x_2		y
0	0	XOR	0
0	1		1
1	0		1
1	1		0



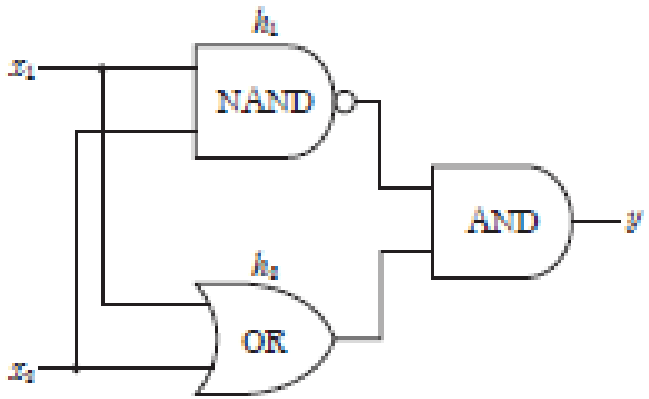
(a) XOR 연산 결과 분류에 실패한 퍼셉트론



(b) XOR 연산 결과를 분류하는 곡선

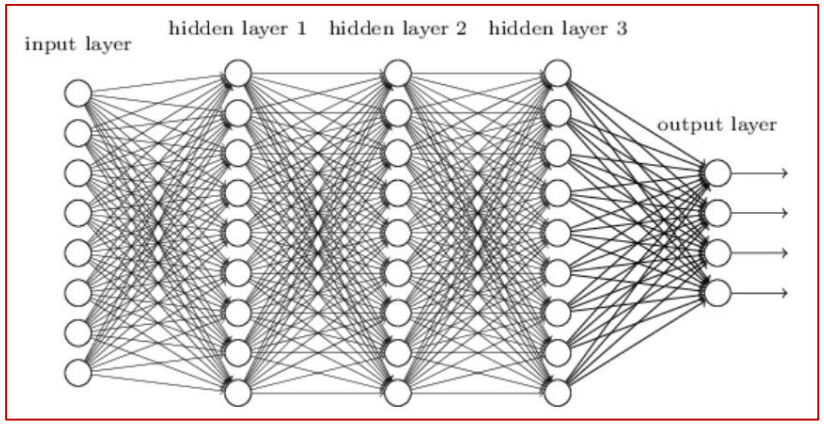
Ⅱ. Perceptron

다층 퍼셉트론



▼ XOR 진리표

입력		연산자		연산자	출력
		NAND	OR		
x_1	x_2	h_1	h_2		y
0	0	1	0	AND	0
0	1	1	1		1
1	0	1	1		1
1	0	0	1		0




```
In [71]: def XOR(x1,x2):
             s1=NAND(x1,x2)
             s2=OR(x1,x2)
             y=AND(s1,s2)
             return y

In [72]: XOR(0,0)
Out[72]: 0

In [73]: XOR(1,0)
Out[73]: 1

In [74]: XOR(0,1)
Out[74]: 1

In [75]: XOR(1,1)
Out[75]: 0
```