

인공지능개론 정리노트 202104295 이인서

-퍼셉트론 손으로 구하기

AND게이트

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

$w_1=20, w_2=15, \Theta=25$

-> $20*0+15*0=0<25$ 흐르지X

-> $20*1+15*0=20<25$ 흐르지X

-> $20*0+15*1=15<25$ 흐르지X

-> $20*1+15*1=35>25$ 흐름

```
def AND(x1, x2):  
    w1, w2, theta = 20, 15, 25  
    sum = x1*w1 + x2*w2  
    if sum <= theta:  
        return 0  
    elif sum > theta:  
        return 1
```

```
print('(0,0)=', AND(0,0))  
print('(1,0)=', AND(1,0))  
print('(0,1)=', AND(0,1))  
print('(1,1)=', AND(1,1))
```

(0,0)= 0
(1,0)= 0
(0,1)= 0
(1,1)= 1

NAND 게이트

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

$w_1=-6, w_2=-10, \Theta=-15$

-> $-6*0+-10*0=0>-15$ 흐름

-> $-6*1+-10*0=-6>-15$ 흐름

-> $-6*0+-10*1=-10>-15$ 흐름

-> $-6*1+-10*1=-16>-15$ 흐르지X

```
import numpy as np  
def NAND(x1, x2):  
    w1, w2, theta = -6, -10, -15  
    sum = (x1*w1 + x2*w2) + b  
    if sum <= theta:  
        return 0  
    elif sum > theta:  
        return 1
```

```
print('(0,0)=', NAND(0,0))  
print('(1,0)=', NAND(1,0))  
print('(0,1)=', NAND(0,1))  
print('(1,1)=', NAND(1,1))
```

(0,0)= 1
(1,0)= 1
(0,1)= 1
(1,1)= 0

OR게이트

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

$w_1=3, w_2=3, \Theta=2$

$\rightarrow 3*0+3*0=0 < 2$ 흐르지X

$\rightarrow 3*1+3*0=3 > 2$ 흐름

$\rightarrow 3*0+3*1=3 > 2$ 흐름

$\rightarrow 3*1+3*1=6 > 2$ 흐름

```

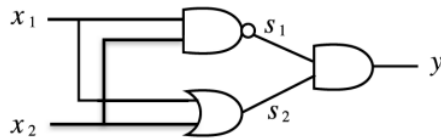
import numpy as np
def OR(x1, x2):
    w1, w2, c = 3, 3, 2
    sum = (x1*w1 + x2*w2) + c
    if sum <= c:
        return 0
    elif sum > c:
        return 1

print('(0,0)=', OR(0,0))
print('(1,0)=', OR(1,0))
print('(0,1)=', OR(0,1))
print('(1,1)=', OR(1,1))

```

(0,0)= 0
 (1,0)= 1
 (0,1)= 1
 (1,1)= 1

AND, NAND, OR 게이트를 조합하여 XOR 게이트를 구현하기



질문

- 단층 퍼셉트론으로 xor게이트를 왜 표현하지 못하는가?

: 단층 퍼셉트론은 선형 분리만 가능하고, xor게이트는 비선형에 해당한다.

단층 퍼셉트론은 입력 공간을 직선으로 나누어 두 개의 영역으로만 분리할 수 있다.

반면 xor게이트의 출력은 입력값에 따라 다르게 결정되어 입출력을 직선으로 나타낼 수 없다.

따라서 xor게이트를 모델링하기 위해서는 다층 퍼셉트론이나 비선형 모델을 사용해야만 한다.

-퍼셉트론으로 컴퓨터를 실제로 구현할 수 없는 이유?

: 퍼셉트론은 선형 분리할 때 사용되는 모델로 복잡한 비선형 문제를 해결할 때에는 적합하지 않다. 다양한 연산과 복잡한 프로그램을 실행하기 위해서는 복잡한 신경망이나 다른 컴퓨터 모델이 필요하다.

-퍼셉트론에서 활성화 함수로 계단 함수를 사용하는 이유는?

: 퍼셉트론은 주로 이진 분류 작업에 쓰이는데, 계단 함수 같은 경우에는 0과 1 두가지 값을 출력하기 때문에 적합하다. 또한 계산이 빠르고, 구현하기가 쉬워 퍼셉트론 모델의 기초를 이루는데 좋다.

-계단 함수와 시그모이드 함수의 공통점은?

두 함수 모두 비선형성 활성화 함수이다. 이는 복잡한 비서형 관계를 모델링할 수 있도록 해준다. 또한 출력 범위가 한정적이다. 차이점은 계단 함수는 0 또는 1의 이진 값을 출력하고, 시그모이드 함수는 0과 1 사이의 연속적인 실수를 출력한다. 또한 두 함수는 입력 값의 변화가 출력 값에 끼치는 영향을 부드럽게 만들어 준다.