<스터디 모임 인증>

<스터디 내용 인증>

가중치와 편향 구현하기

In [50]:
```
#AND 게이트 구현
def AND(x1,x2):
    x = np.array([x1, x2])
    w = np.array([0.5, 0.5])
    b = -0.7  # -theta가 편향 b로 치환되었음

    #가중치와 입력 신호를 곱한 값들의 합이 -b를 넘겨야 1이 출력된다
    tmp = np.sum(w*x) + b
    if tmp <= 0:
        return 0
    else:
        return 1
```

In [51]:
```
#NAND 게이트 구현
#AND와는 가중치 (w와 b)만 다르다! -> ANDdptj 가중치 값만 음수로 변환

def NAND(x1,x2):
    x = np.array([x1, x2])
    w = np.array([-0.5, -0.5])
    b = 0.7  # -theta가 편향 b로 치환되었음

    #가중치와 입력 신호를 곱한 값들의 합이 -b를 넘겨야 1이 출력된다
    tmp = np.sum(w*x) + b
    if tmp <= 0:
        return 0
    else:
        return 1
```

In [52]:
```
#OR 게이트 구현

def OR(x1,x2):
    x = np.array([x1, x2])
    w = np.array([0.5, 0.5])
    b = -0.2  # -theta가 편향 b로 치환되었음

    #가중치와 입력 신호를 곱한 값들의 합이 -b를 넘겨야 1이 출력된다
    tmp = np.sum(w*x) + b
    if tmp <= 0:
        return 0
    else:
        return 1
```

2.4 퍼셉트론의 한계

윤빈

퍼셉트론

- x1과 x2는 입력 신호, y는 출력 신호, w1과 w2는 가중치(weight)를 의미한다.
- 원을 **뉴런** 또는 **노드**라고 부른다.
- 입력 신호가 뉴런에 보내질 때는 각각 고유한 **가중치**가 곱해진다(w1x1, w2x2).
- 뉴런에서 전달 받은 신호의 총합이 **임계값** θ 를 넘을 때만 1을 출력한다.

단순한 논리 회로

AND게이트

두 입력이 **모두 1일 때만 1을 출력**

NAND게이트

== Not And 게이트

And 게이트의 출력을 뒤집은 것

OR게이트

두 입력 중 하나라도 1이면 1을 출력

퍼셉트론 구현

가중치와 편향 도입

- theta를 왼쪽 항으로 이항한 것(theta → -b)
- 기호 표기만 바뀐 채 똑같은 구조

퍼셉트론은 뇌의 뉴런을 모방한 것으로 인공신경망의 기본 단위입니다.

구조에 대해 간단히 살펴보면 입력값이 들어가는 input node와 가중치가 곱해지는 edge, hidden node, output node가 있습니다.

또 이 노드들이 여러 개 모인 것에는 layer라는 이름이 붙습니다.

차례대로 input layer , hidden layer, 지금은 출력값이 하나지만 여러 개가 될 경우 outpur layer라는 이름이 붙게 됩니다.

다음으로 퍼셉트론의 동작에 대해 살펴보겠습니다.

데이터를 통해 인풋이 주어지면 가중치가 곱해진 후 히든 레이어에서 더해지게 됩니다. 히든 레이어는 총 두 가지 역할을 하는 것으로 볼 수 있습니다. 가중치가 곱해져 더해지는 weighted sum과 activation function으로서의 역할을 합니다.

Activation function 이란 weighted sum을 출력 값으로 변환시키는 함수를 의미합니다. 책에서 나왔던 and gate와 or gate 등과 같은 역할을 한다고 생각하시면 될 거 같습니다.(필요시 and gate로 설명)

Activation function은 non linear activation function과 inear activation function으로 나뉩니다.

Non-linear activation function에는 시그모이드 함수, relu 함수, tanh 등이 있는데 범위를 제한하거나 선형성을 제거하는 용도로 사용됩니다.

유의해야할 점은 linear activation이 들어가면 출력 값이 변하지 않는다는 점입니다..

왜 그런지 보면

(식 적기)

괄호를 하나 더 묶어 다르게 표현한 것일 뿐 사실 weight에서 다 학습될 수 있는 내용이었습니다.

즉 linear activation function은 출력 값에 영향을 주지 못한다고 볼 수 있습니다.

변화를 주지 못하므로 linear activation function은 대부분 사용하지 않습니다.

이렇게 hidden layer를 통과하면 출력 값이 완성됩니다.

출력 값을 통과한 후에는 퍼셉트론을 통해 출력된 결괏값과 y와 실제 값의 차이를 줄여가며 퍼셉트론을 개선해 나가는 것입니다.

이번에는 여기까지 하겠습니다. 감사합니다.