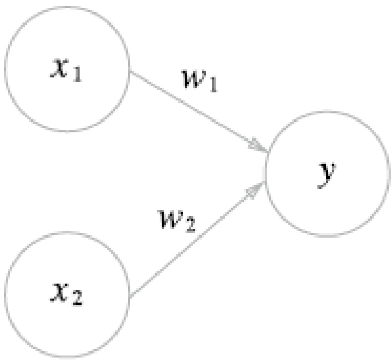
Chapter 2. 퍼셉트론(Perceptron)

- 프랑크 로젠블라트가 1957년 고인한 알고리즘
- 신경망(딥러닝) 기원이 되는 알고리즘

2-1. 퍼셉트론이란?

- 퍼셉트론은 다수의 신호를 입력으로 받아 하나의 신호를 출력함
- 여기서의 신호는 전류나 강물처럼 '흐름'이 있는 것이고, 실제 전류와는 달리 **퍼셉트론은 흐른다(1)/안흐른다(0) 의 두가지 값**을 가짐



x1, x2 : 입력 신호 / w1, w2 : 가중치 / y : 출력신호 / 그림의 원 : 노드, 뉴런

- 입력 신호가 뉴런에 보내질 때 각각 고유한 가중치가 곱해짐(w1*x1, w2*x2)
- 뉴런에서 보낸 신호의 총합이 정해진 한계를 넘어설 때만 1을 출력(뉴런이 활성화 할때만)
- 여기서의 한계를 `임계값` (θ) 로 나타냄

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

퍼셉트론의 동작 원리

- 퍼셉트론은 복수의 입력 신호 각각에 고유한 가중치를 부여함
- 가중치는 각 신호가 결과에 주는 영향력을 조절하는 요소로 작용
- 가중치가 클수록 해당 신호가 그만큼 더 중요함

2-2. 단순한 논리 회로

`AND 게이트`

- 입력이 둘이고 출력은 하나
- 아래는 **AND 게이트의 진리표**(*진리표 : 같은 입력 신호와 출력 신호의 대응표)로, 입력이 모두 1일 때만 1을 출력하고 그 외는 0을 출력
-

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

< AND 게이트의 진리표 >

- 위의 AND 게이트의 진리표를 만족하는 매개변수 조합은 무한히 많은데, (w1, w2, θ)가 (0.5, 0.5, 0.7) 일 때, 혹은 (0.5, 0.5, 0.8)이나 (1.0, 1.0, 1.0) 일 때 모두 AND 게이트의 조건을 만족
- 매개변수를 위와 같이 설정하면 x1, x2가 모두 1일 때만 가중 신호의 총합이 주어진 임계값 웃돎

`NAND 게이트`

- NAND 는 Not AND로, AND 게이트의 출력을 뒤집은 것 입력이 모두 1일 때만 0을 출력, 그 외는 1을 출력

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

<NAND 게이트의 진리표>

- 여러 매개변수가 있으나, (w1, w2, θ) = (-0.5, -0.5, -0.7) 조합
- AND 게이트를 구현하는 매개변수의 부호를 모두 반전하면 NAND 게이트가 됨

`OR 게이트`

- 입력 신호 중 하나 이상이 1이면 출력이 1이 되는 논리 회로

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

<OR 게이트의 진리표>

- 퍼셉트론의 매개변수 값을 정하는 것은 컴퓨터가 아니라 인간임. 인간이 직접 진리표라는 '학습 데이터'를 보면서 매개변수의 값을 생가기함

- 기계학습 문제는 이 매개변수의 값을 정하는 작업을 컴퓨터가 자동으로 하도록 학습함
- 학습은 적절한 매개변수 값을 정하는 작업이며, 사람은 퍼셉트론의 구조(모델)을 고민하고 컴퓨터에 학습할 데이터를 주는 일을 함

- 퍼셉트론으로 AND, NAND, OR 논리 회로를 표현할 수 있음
- 퍼셉트론의 구조는 AND, NAND, OR 게이트에서 모두 같으며, 세 가지 게이트에서 다른 점은 매개변수(가중치와 임계값)임
- 똑같은 구조의 퍼셉트론이 매개변수의 값을 적절히 조정하면 AND, NAND, OR이 됨

2-3. 퍼셉트론 구현하기

`AND 게이트 구현`

```
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
```

- 매개변수 w1, w2, theta는 함수 안에서 초기화하고,
가중치를 곱한 입력의 총합이 임계값을 넘으면 1을 반환, 그 외는 0을 반환함

`가중치와 편향 도입`

- AND 게이트는 직관적이고 알기 쉽지만 θ를 -b로 치환하면, 퍼셉트론의 동작은

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

b : 편향(bias) / w1, w2 : 가중치

- 퍼셉트론은 입력 신호에 가중치를 곱한 값과 편향을 합하여 그 값이 0이 넘으면 1을, 그렇지 않으면 0을 출력함

가중치와 편향 구하기

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

```

- θ가 편향 b로 치환됨
- 편향은 가중치 w1, w2와 기능이 다름
- w1, w2는 각 입력 신호가 결과에 영향을 주는 영향력(중요도)를 조절하는 매개변수
- bias(편향)은 뉴런이 얼마나 쉽게 활성화(결과로 1을 출력)하느냐를 조정하는 매개변수

- 예를 들어, b가 -0.1이면 각 입력 신호에 가중치를 곱한 값들의 합이 -0.1을 초과할 때만 뉴런이 활성화
반면 b가 -20.0이면 각 입력 신호에 가중치를 곱한 값들이 20.0가 넘지 않으면 뉴런이 활성화 하지 않음
- 편향의 값은 뉴런이 얼마나 쉽게 활성화되는지를 결정함
- w1, w2를 '가중치', b를 '편향'으로 서로 구별하기도 하지만, 문맥에 따라 셋 모두를 '가중치'라고 할 때도 있음

- 편향 '한쪽으로 치우쳐 균형을 깬다', 입력이모두 0이어도 결과로 (0이 아닌) 편향 값을 출력함

NAND 게이트 & OR 게이트

```
def NAND(x1, x2):
    x = np.array([x1, x2])
    w = np.array([-0.5, -0.5])
    b = 0.7
    tmp = np.sum(x*w) +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])
    b = -0.2
    tmp = np.sum(x*w)
    if tmp <=0:
        return 0
    else :
        return 1
```

2-4. 퍼셉트론의 한계

XOR 게이트

- XOR 게이트는 **배타적 논리합**의 논리 회로
- x1, x2 중 한 쪽이 1일 때만 1을 출력함('배타적' 자기 외에 거부)

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

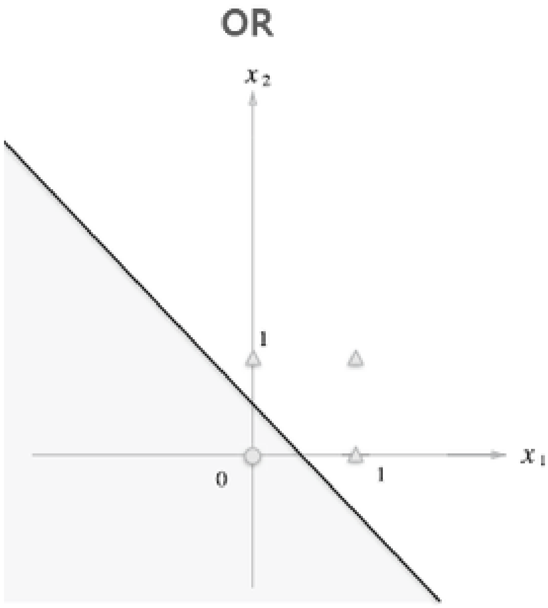
- 위의 퍼셉트론 구조로는 XOR 게이트를 구현할 수 없음

  * OR 게이트의 동작 *

  - OR 게이트는 가중치 매개변수가 (b, w1, w2) → (-0.5, 1.0, 1.0) 일 때 OR 진리표를 만족하고, 이때의 퍼셉트론 식은

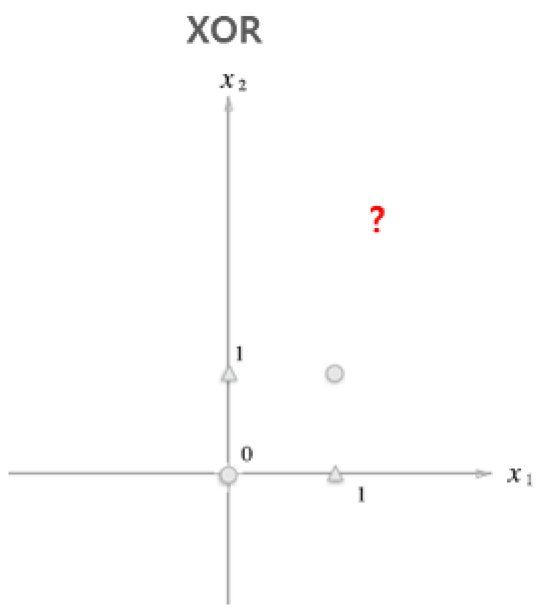
$$y = \begin{cases} 0 \; (-0.5 + x_1 + x_2 \leq 0) \\ 1 \; (-0.5 + x_1 + x_2 > 0) \end{cases}$$

  - 퍼셉트론을 시각화 하면(회색 영역 → 0을 출력, 전체 영역 OR 게이트의 성질 만족)



- OR 게이트는 (x1, x2) = (0,0) 일 때 0 출력 (0,1), (1,0), (1,1) 일 때는 1을 출력

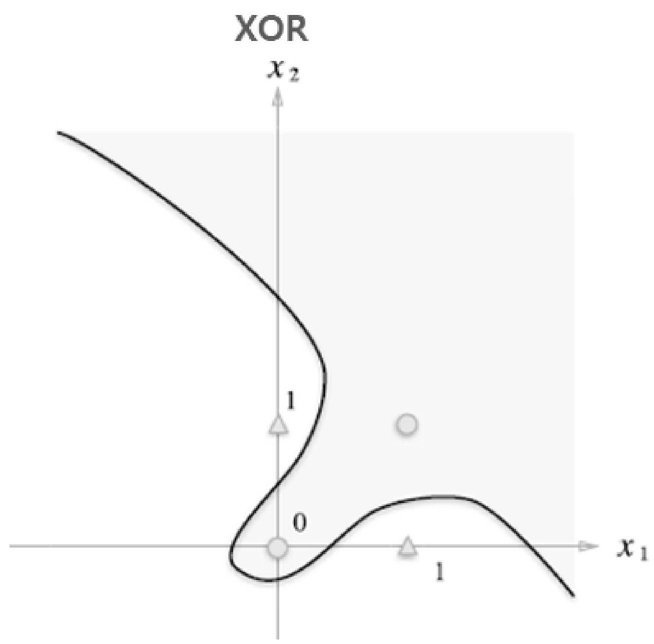
- OR 게이트를 만드려면 그림의 원과 세모를 직선 하나로 나눌 수 있음



- XOR 게이트의 경우에는 그림의 원과 세모를 직선 하나로 나눌 수 없음

선형과 비선형

- 직선 하나라는 조건을 빼면 XOR 게이트를 곡선으로 나눌 수 있음



- 퍼셉트론은 직선 하나로 나눈 영역만 표현할 수 있다는 한계가 있음

- 곡선의 영역인 ‘비선형 영역’, 직선의 영역인 ‘선형’ 영역

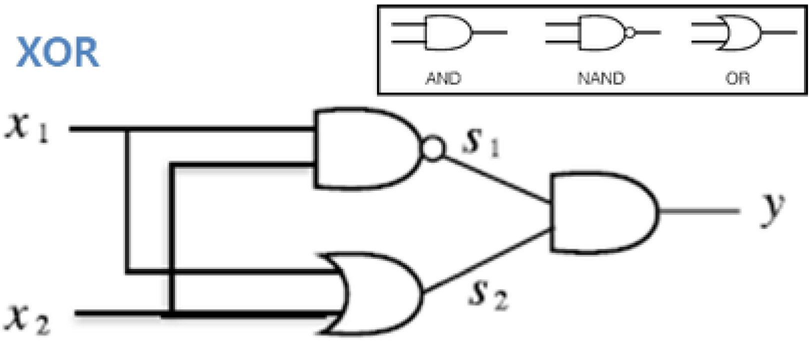
2-5. 다층 퍼셉트론

- 퍼셉트론으로는 XOR 게이트를 표현할 수 없어, 퍼셉트론의 ‘층을 쌓아’ 다층 퍼셉트론(multi-layer perceptron)을 만들 수 있음
- 퍼셉트론의 한계는 ‘단층 퍼셉트론(single perceptron)’ 으로 XOR 게이트를 표현할 수 없다. 또는 단층 퍼셉트론으로는 비선형 영역을 분리할 수 없다임

`기존 게이트 조합하기`

- XOR 게이트를 만드는 방법은 AND, NAND, OR 게이트를 조합하는 방법이 있음

XOR



AND

NAND OR XOR

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

XOR 게이트 구현하기

XOR 게이트

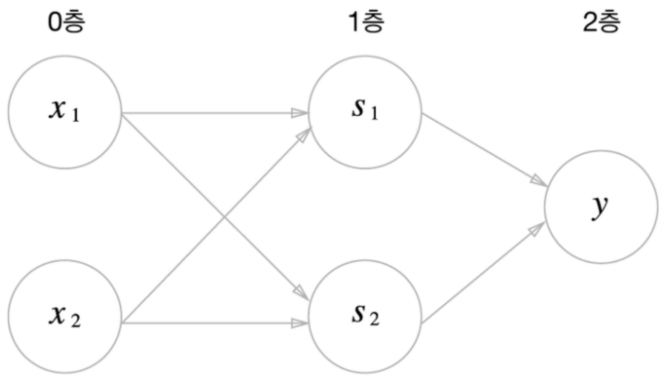
```python
def XOR(x1, x2):
    s1 = NAND(x1, x2)
    s2 = OR(x1, x2)
    y = AND(x1, x2)
    return y

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

```
0
0
0
1
```

```python
def NAND(x1, x2):
    x = np.array([x1, x2])
    w = np.array([-0.5, -0.5])
    b = 0.7
    tmp = np.sum(x*w) +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])
    b = -0.2
    tmp = np.sum(x*w)
    if tmp <=0:
        return 0
    else :
        return 1
```



[XOR 의 퍼셉트론]

- XOR은 다층 구조의 네트워크임
- AND, OR가 단층 프셉트론인 데 반해, XOR은 2층 퍼셉트론임

  ⇒ 층이 여러 개인 퍼셉트론을 **다층 퍼셉트론**
- 위의 XOR 퍼셉트론은 모두 3층으로 구성되나, 가중치를 갖는 층은 2개(0층과 1층 사이, 1층과 2층 사이) 뿐이라 ‘2층 퍼셉트론’ 이라고 함

  (문헌에 따라 구성 층의 수를 기준으로 ‘3층 퍼셉트론’ 이라고 하는 경우가 있음)

[1] 0층의 두 뉴런이 입력 신호를 받아 1층의 뉴런으로 신호를 보냄

[2] 1층의 뉴런이 2층의 뉴런으로 신호를 보내고, 2층의 뉴런은 y를 출력함

- 단층 퍼셉트론으로는 표현하지 못한 것을 층을 하나 늘려 구현할 수 있음

2-7. 정리

- 퍼셉트론은 입출력을 갖춘 알고리즘이며, 입력을 주면 정해진 규칙에 따른 값을 출력함
- 퍼셉트론에서는 '가중치'와 '편향'을 매개변수로 설정
- 퍼셉트론으로 AND, OR 게이트 등의 논리 회로를 표현할 수 있음
- XOR 게이트는 단층 퍼셉트론으로는 표현할 수 없음
- 2층 퍼셉트론을 이용하면 XOR 게이트를 표현할 수 있음
- 단층 퍼셉트론은 직선형 영역만 표현할 수 있고, 다층 퍼셉트론은 비선형 영역도 표현 가능
- 다층 퍼셉트론은 (이론상) 컴퓨터를 표현 할 수 있음