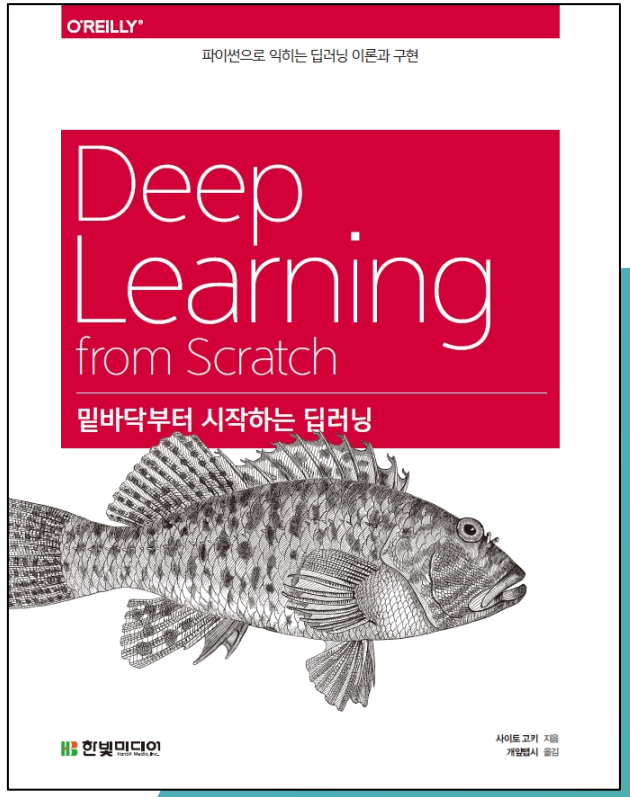
CHAPTER 2 퍼셉트론

밑바닥부터 시작하는 딥러닝



홍익대학교 컴퓨터공학과
박 준

이 책의 학습 목표

- CHAPTER 1 파이썬에 대해 간략하게 살펴보고 사용법 익히기
- CHAPTER 2 퍼셉트론에 대해 알아보고 퍼셉트론을 써서 간단한 문제를 풀어보기
- CHAPTER 3 신경망의 개요, 입력 데이터가 무엇인지 신경망이 식별하는 처리 과정 알아보기
- CHAPTER 4 손실 함수의 값을 가급적 작게 만드는 경사법에 대해 알아보기
- CHAPTER 5 가중치 매개변수의 기울기를 효율적으로 계산하는 오차역전파법 배우기
- CHAPTER 6 신경망(딥러닝) 학습의 효율과 정확도를 높이기
- CHAPTER 7 CNN의 메커니즘을 자세히 설명하고 파이썬으로 구현하기
- CHAPTER 8 딥러닝의 특징과 과제, 가능성, 오늘날의 첨단 딥러닝에 대해 알아보기

Contents

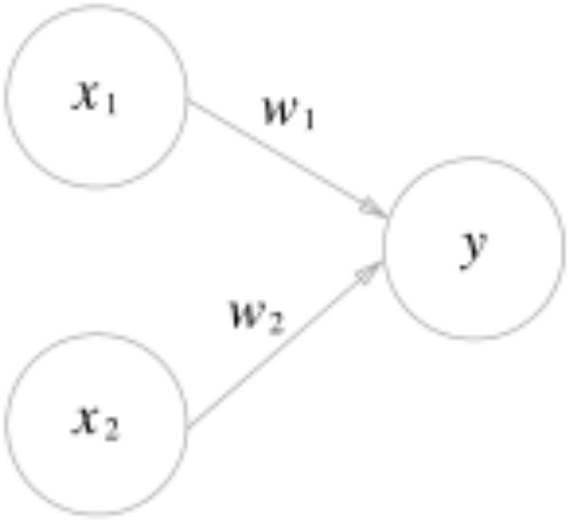
CHAPTER 2 퍼셉트론

퍼셉트론에 대해 알아보고 퍼셉트론을 써서 간단한 문제를 풀어보기

2.1 퍼셉트론이란?

퍼셉트론 * 은 다수의 신호를 입력으로 받아 하나의 신호를 출력.
퍼셉트론 신호도 흐름을 만들고 정보를 앞으로 전달.
다만, 실제 전류와 달리 퍼셉트론 신호는 '흐른다/안 흐른다(1 이나 0)'의 두 가지 값을 가진다.
이 책에서는 1 을 '신호가 흐른다', 0 을 '신호가 흐르지 않는다'라는 의미쓰인다.

그림 2-1 입력이 2개인 퍼셉트론



$$y = \begin{cases} 0\ (w_1x_1 + w_2x_2 \leq \theta) \\ 1\ (w_1x_1 + w_2x_2 > \theta) \end{cases} \quad [식\ 2.1]$$

2.2 단순한 논리 회로

[그림 2 - 2]와 같은 입력 신호와 출력 신호의 대응 표를 진리표 라고 한다.
이 그림은 AND 게이트의 진리표로, 두입력이 모두 1 일 때만 1 을 출력하고, 그 외에는 0 을 출력

그림 2-2 AND 게이트의 진리표

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

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

NAND 게이트를 표현하려면 예를 들어 (w 1 , w 2 , θ) = (- 0 . 5 , - 0 . 5 , - 0 . 7) 조합 이 있다.
사실 AND 게이트를 구현하는 매개변수의 부호를 모두 반전하기만 하면 NAND 게이트가 된다

2.3 퍼셉트론 구현하기

x1 과 x2 를 인수로 받는 AND 라는 함수.

```
def AND(x1, x2):
    w1, w2, theta = 0.5, 0.5, 0.7
```

매개변수 w1 , w2 , theta 는 함수 안에서 초기화하고, 가중치를 곱한 입력의 총합이 임계값을 넘으면 1 을 반환하고 그 외에는 0 을 반환.
이 함수의 출력이 [그림 2 - 2]와 같은지 확인

```
AND(0, 0) # 0을 출력
AND(1, 0) # 0을 출력
AND(0, 1) # 0을 출력
AND(1, 1) # 1을 출력
```

$$y = \begin{cases} 0\ (w_1x_1 + w_2x_2 \le \theta) \\ 1\ (w_1x_1 + w_2x_2 > \theta) \end{cases} \quad [식\ 2.1]$$

2.3.2 가중치와 편향 도입

$$y = \begin{cases} 0\ (b + w_1x_1 + w_2x_2 \le 0) \\ 1\ (b + w_1x_1 + w_2x_2 > 0) \end{cases} \quad [식\ 2.2]$$

[식 2 . 1]과 [식 2 . 2]는 기호 표기만 바꿨을 뿐, 그 의미는 같다.
여기에서 b 를 편향 bias 이라하며 w 1 과 w 2 는 그대로 가중치 weight .
[식 2 . 2] 관점에서 해석해보자면, 퍼셉트론은 입력신호에 가중치를 곱한 값과 편향을 합하여, 그 값이 0 을 넘으면 1 을 출력하고 그렇지 않으면 0을 출력.
그럼 넘파이를 사용해서 [식 2 . 2] 방식으로 구현해보자.

```
>>> import numpy as np
>>> x = np.array([0, 1])      # 입력
>>> w = np.array([0.5, 0.5]) # 가중치
>>> b = -0.7                  # 편향
>>> w*x
array([ 0. ,  0.5])
>>> np.sum(w*x)
0.5
>>> np.sum(w*x) + b
-0.19999999999999996  # 대략 -0.2 ( 부동소수점 수에 의한 연산 오차 )
```

2.3.3 가중치와 편향 구현하기

```
def AND(x1, x2):
```

여기에서 - θ 가 편향 b 로 치환(2 . 3 . 1 절에서 구현한 AND 의 theta 가 - b 가 되었다).
그리고 편향은 가중치 w 1 , w 2 와 기능이 다르다는 사실에 주의.
w 1 과 w 2 는 각 입력 신호가 결과에 주는 영향력(중요도)을 조절하는 매개변수고, 편향은 뉴런이 얼마나 쉽게 활성화(결과로 1 을 출력)하느냐를 조정하는 매개변수.

2.3.3 가중치와 편향 구현하기

실습

OR와 NAND 함수 작성하기

2.4 퍼셉트론의 한계

그림 2-5 XOR 게이트의 진리표

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

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

[식 2.3]

우선 OR 게이트의 동작을 시각적으로 생각해보자.
OR 게이트는, 예를 들어 가중치 매개변수가 (b , w 1 , w 2) = (- 0 . 5 , 1 . 0 , 1 . 0)일 때 [그림 2 - 4]의 진리표를 만족합니다. 이때의 퍼셉트론은 [식 2 . 3]으로 표현.

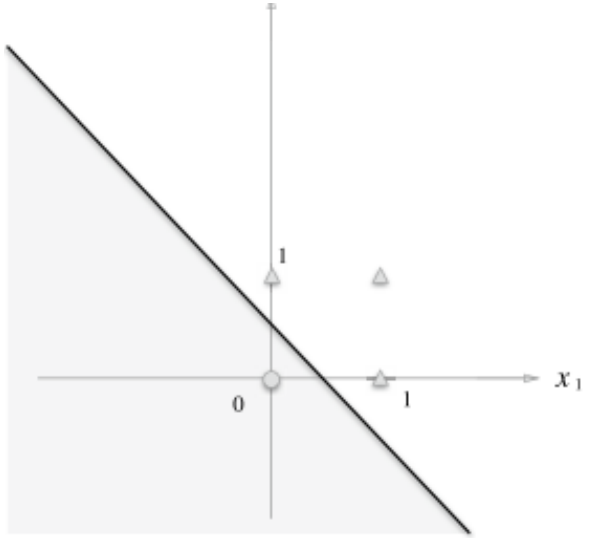


그림 2-6 퍼셉트론의 시각화 : 회색 영역은 0 을 출력하는 영역이며, 전체 영역은 OR 게이트의 성질을 만족한다.

2.4.2 선형과 비선형

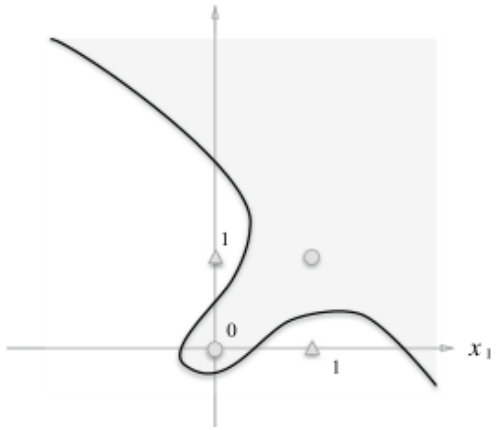


그림 2-8 곡선이라면 ○과 △을 나눌 수 있다.

퍼셉트론은 직선 하나로 나눈 영역만 표현할 수 있다는 한계가있다.
[그림 2 – 8] 같은 곡선은 표현할 수 없다는 것이다.
덧붙여서 [그림 2 – 8]과 같은 곡선의 영역을 비선형 영역, 직선의 영역을 선형 영역이라고 한다.
선형, 비선형이라는 말은 기계학습 분야에서 자주 쓰이는용어로, [그림 2 - 6]과 [그림 2 - 8] 같은 이미지를 떠올리시면 된다

2.5.1 기존 게이트 조합하기

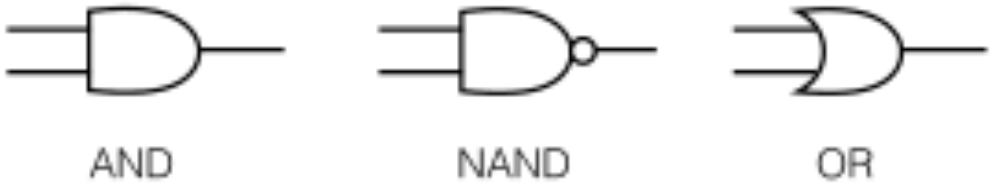


그림 2-9 AND , NAND , OR 게이트 기호

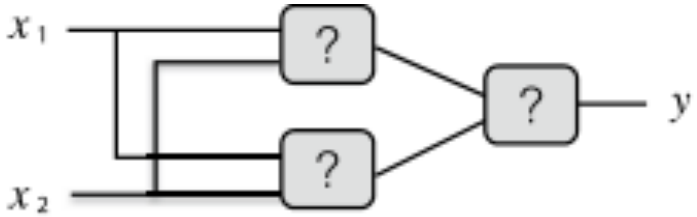


그림 2-10 AND , NAND , OR 게이트 하나씩을 '?'에 대입해 XOR 를 완성하자

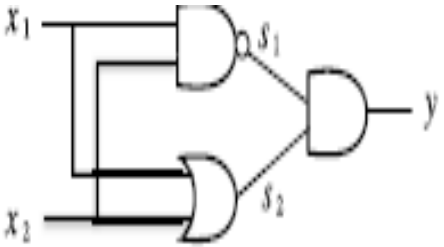


그림 **2-11**

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

그림 **2-12**

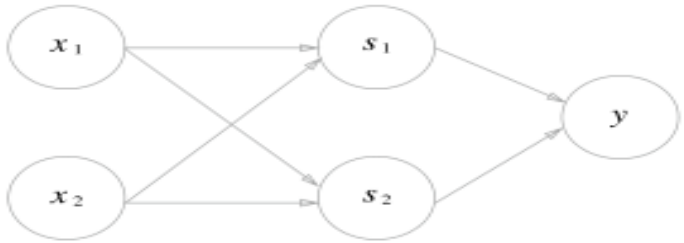
NAND 의 출력을 s 1 , OR의 출력을 s 2 로 해서 진리표를 만들면 [그림 2 - 12]처럼 된다.
X 1 , x 2 , y 에 주목하면 분명히턖 의 출력과 같다.

2.5.2 XOR 게이트 구현하기

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

이 XOR 함수는 기대한 대로의 결과를 출력.

```
XOR(0, 0) # 0을 출력
XOR(1, 0) # 1을 출력
XOR(0, 1) # 1을 출력
XOR(1, 1) # 0을 출력
```



이상으로 2 층 구조를 사용해 퍼셉트론으로 XOR 게이트를 구현할 수 있게 되었다. 이처럼 퍼셉트론은 층을 쌓아(깊게 하여) 더 다양한 것을 표현할 수 있다.

2.6 NAND에서 컴퓨터까지

NAND 게이트의 조합만으로 컴퓨터가 수행하는 일을 재현할 수 있다. NAND 게이트만으로 컴퓨터를만들 수 있다? 이 말은 곧 퍼셉트론으로도 컴퓨터를 표현할 수 있다는 놀라운 사실로 이어진다.
지금까지 살펴본 것처럼 NAND 게이트는 퍼셉트론으로 만들 수 있기 때문이다

산술 논리 연산 장치(ALU), 그다음에는 CPU라는 식.
그래서 퍼셉트론으로 표현하는 컴퓨터도 여러 층을 다시 층층이 겹친 구조로 만드는 방향이 자연스러운 흐름.