

# 6장: 퍼셉트론

• 퍼셉트론! 신경망을 이루는 가장 중요한 기본 단위

가중치  $y = w \cdot x + b$  ( $w$ 는 가중치,  $b$ 는 바이어스)



XOR 문제

# 7장: 다층 퍼셉트론

• 은닉층 설계



$$n_1 = \sigma(x_1 w_{11} + x_2 w_{21} + b_1)$$

$$n_2 = \sigma(x_1 w_{12} + x_2 w_{22} + b_2)$$

⇓

$$y = \sigma(n_1 w_{31} + n_2 w_{32} + b_3)$$

$$W(1) = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} \quad B(1) = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

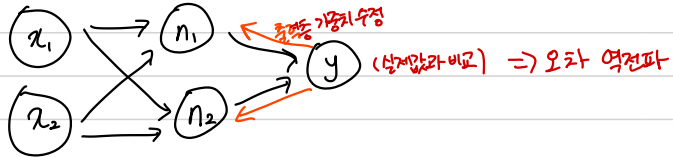
$$W(2) = \begin{bmatrix} w_{31} \\ w_{32} \end{bmatrix} \quad B(2) = \begin{bmatrix} b_3 \end{bmatrix}$$

(XOR 실험)

```
3 import numpy as np
4
5 # 가중치와 바이어스
6 w11 = np.array([-2, -2])
7 w12 = np.array([2, 2])
8 w2 = np.array([1, 1])
9 b1 = 3
10 b2 = -1
11 b3 = -1
12
13 # 퍼셉트론
14 def MLP(x, w, b):
15     y = np.sum(w * x) + b
16     if y <= 0:
17         return 0
18     else:
19         return 1
20
21 # NAND 게이트
22 def NAND(x1, x2):
23     return MLP(np.array([x1, x2]), w11, b1)
24
25 # OR 게이트
26 def OR(x1, x2):
27     return MLP(np.array([x1, x2]), w12, b2)
28
29 # AND 게이트
30 def AND(x1, x2):
31     return MLP(np.array([x1, x2]), w2, b3)
```

```
33 # XOR 게이트
34 def XOR(x1, x2):
35     return AND(NAND(x1, x2), OR(x1, x2))
36
37
38 # x1, x2 값을 번갈아 대입해 가며 최종값 출력
39 if __name__ == '__main__':
40     for x in [(0, 0), (1, 0), (0, 1), (1, 1)]:
41         y = XOR(x[0], x[1])
42         print("입력 값: " + str(x) + " 출력 값: " + str(y))
43
44 입력 값: (0, 0) 출력 값: 0
45 입력 값: (1, 0) 출력 값: 1
46 입력 값: (0, 1) 출력 값: 1
47 입력 값: (1, 1) 출력 값: 0
```

## 8장: 오차 역전파



(과정)

- ① 입력의 초기 가중치( $w$ )를 줬 뒤 결과( $y_{out}$ )를 계산
- ② 결과와 원하는 값 사이의 오차를 구함
- ③ 경사 하강법을 이용하여 바로 앞 가중치를 오차가 작아지는 방향으로 업데이트 (가중치가 0이되는 방향)
- ④ 위 과정 반복

$$w(t+1) = w_t - \frac{\partial \text{오차}}{\partial w}$$

가중치에 대한 기울기

## 9장: 신경망에서 딥러닝으로

- 기울기 소실  $\rightarrow$   $\Rightarrow$  여러 계층 도입
- 효율적 경사 하강법 : 경사하강법을 보완 속도 개선
- 모멘텀 : 효율적 경사하강법을 보완 정확도 개선
- 네스테로프 모멘텀
- 아다그라드
- 알엠 에스프롭