



셋째마당

딥러닝의 시작, 신경망

7장 퍼셉트론과 인공지능의 시작

1 인공지능의 시작을 알린 퍼셉트론

2 퍼셉트론의 과제

3 XOR 문제



1 인공지능의 시작을 알린 퍼셉트론

1 인공지능의 시작을 알린 퍼셉트론

● 인공지능의 시작을 알린 퍼셉트론

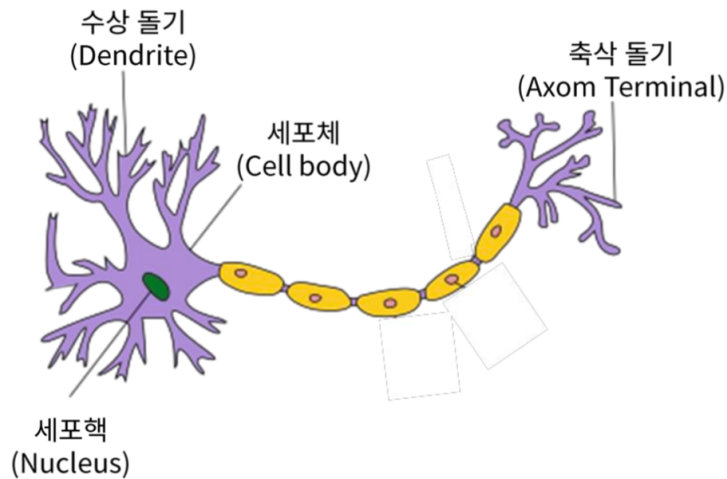
- 인간의 뇌는 치밀하게 연결된 뉴런 약 1,000억 개로 이루어져 있음
- 뉴런과 뉴런 사이에는 시냅스라는 연결 부위가 있는데, 신경 말단에서 자극을 받으면 시냅스에서 화학 물질이 나와 전위 변화를 일으킴
- 전위가 임계 값을 넘으면 다음 뉴런으로 신호를 전달하고, 임계 값에 미치지 못하면 아무것도 하지 않음
- 이 메커니즘은 우리가 앞서 배운 로지스틱 회귀와 많이 닮았음
- 이 간단한 회로는 입력 값을 놓고 활성화 함수에 의해 일정한 수준을 넘으면 참을, 그렇지 않으면 거짓을 내보내는 일을 하는데 뉴런과 유사함

1 인공지능의 시작을 알린 퍼셉트론

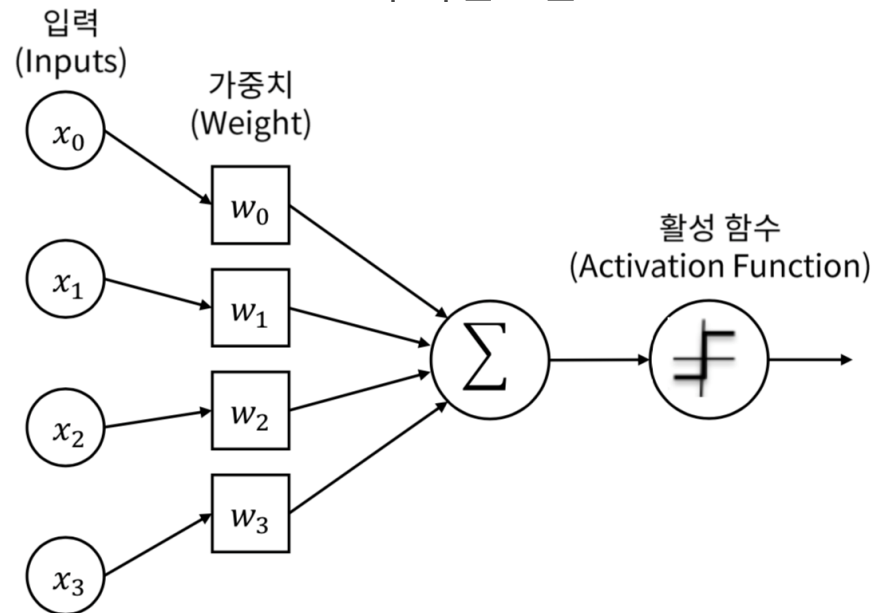
- 퍼셉트론 : 입력층과 출력층으로만 구성된 최초의 인공신경망

▼ 그림 7-1 | 뉴런의 신호 전달

뉴런(신경세포)



초기 퍼셉트론





1 인공지능의 시작을 알린 퍼셉트론

● 인공지능의 시작을 알린 퍼셉트론

- 우리 몸 안에 있는 수많은 뉴런은 서로 긴밀히 연결되어 신경 말단부터 뇌까지 곳곳에서 자신의 역할을 수행
- 이처럼 복잡하고 어려운 조합의 결과가 바로 우리의 '생각'
- 뉴런과 비슷한 메커니즘을 사용하면 인공적으로 '생각'하는 그 무언가를 만들 수 있지 않을까?



1 인공지능의 시작을 알린 퍼셉트론

- 인공지능의 시작을 알린 퍼셉트론

- 이러한 상상과 함께 출발한 연구가 바로 인공 신경망(**artificial neural network**, 이하 줄여서 '**신경망**'이라고 함) 연구
- 맨 처음 시작은 '켜고 끄는 기능이 있는 신경'을 그물망 형태로 연결하면 사람의 뇌처럼 동작할 수 있다는 가능성을 처음으로 주장한 맥컬록-윌터 피츠(McCulloch-Walter Pitts)의 1943년 논문
- 그 후 1957년, 미국의 신경 생물학자 프랑크 로젠블랫(Frank Rosenblatt)이 이 개념을 실제 장치로 만들어 선보임
- 이것의 이름이 **퍼셉트론(perceptron)**

1 인공지능의 시작을 알린 퍼셉트론

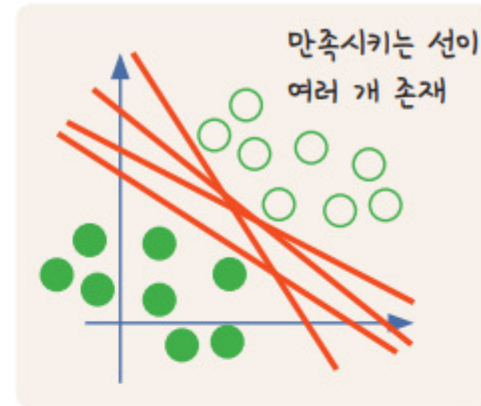
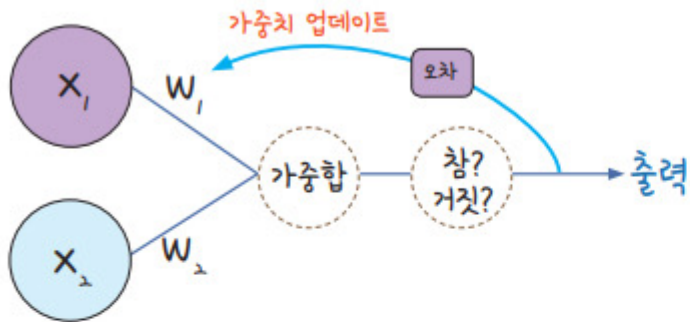
● 인공지능의 시작을 알린 퍼셉트론

- 퍼셉트론은 입력 값을 여러 개 받아 출력을 만드는데, 이때 입력 값에 **가중치**를 조절할 수 있게 만들어 최초로 '**학습**'을 하게 했음(그림 7-2 ① 참조)
- 3년 후, 여기에 앞 장에서 다룬 **경사 하강법을 도입해 최적의 경계선을** 그릴 수 있게 한 **아달라인(Adaline)**이 개발(그림 7-2 ② 참조)
- 특히 아달라인은 이후 서포트 벡터 머신(Support Vector Machine) 등 머신러닝의 중요한 알고리즘들로 발전해 가는데, 이 중 **시그모이드**를 활성화 함수로 사용한 것이 바로 앞서 배웠던 **로지스틱 회귀**

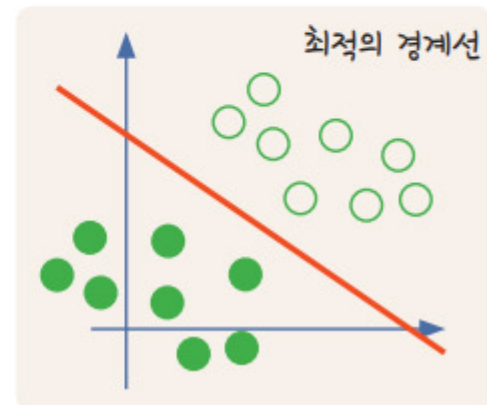
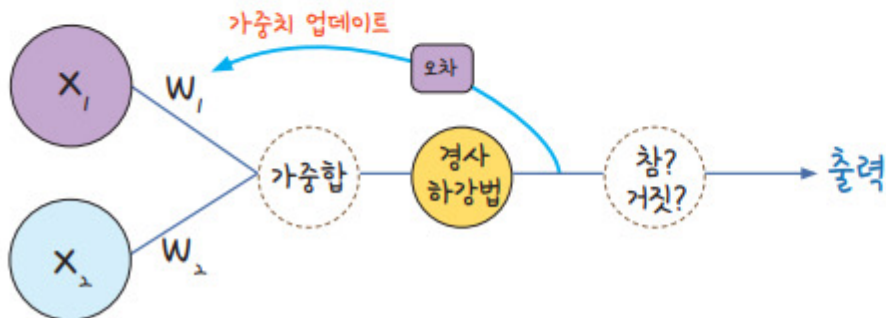
1 인공지능의 시작을 알린 퍼셉트론

▼ 그림 7-2 | 퍼셉트론, 아달라인 그리고 로지스틱 회귀 모델

① 퍼셉트론



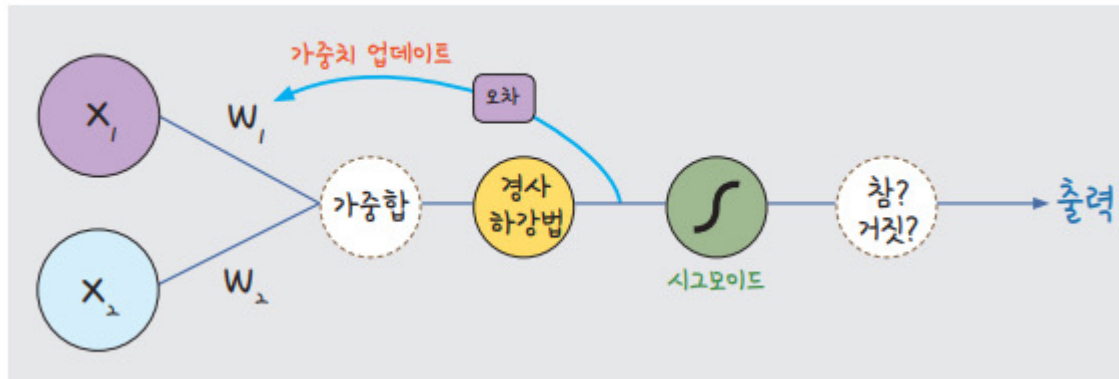
② 아달라인





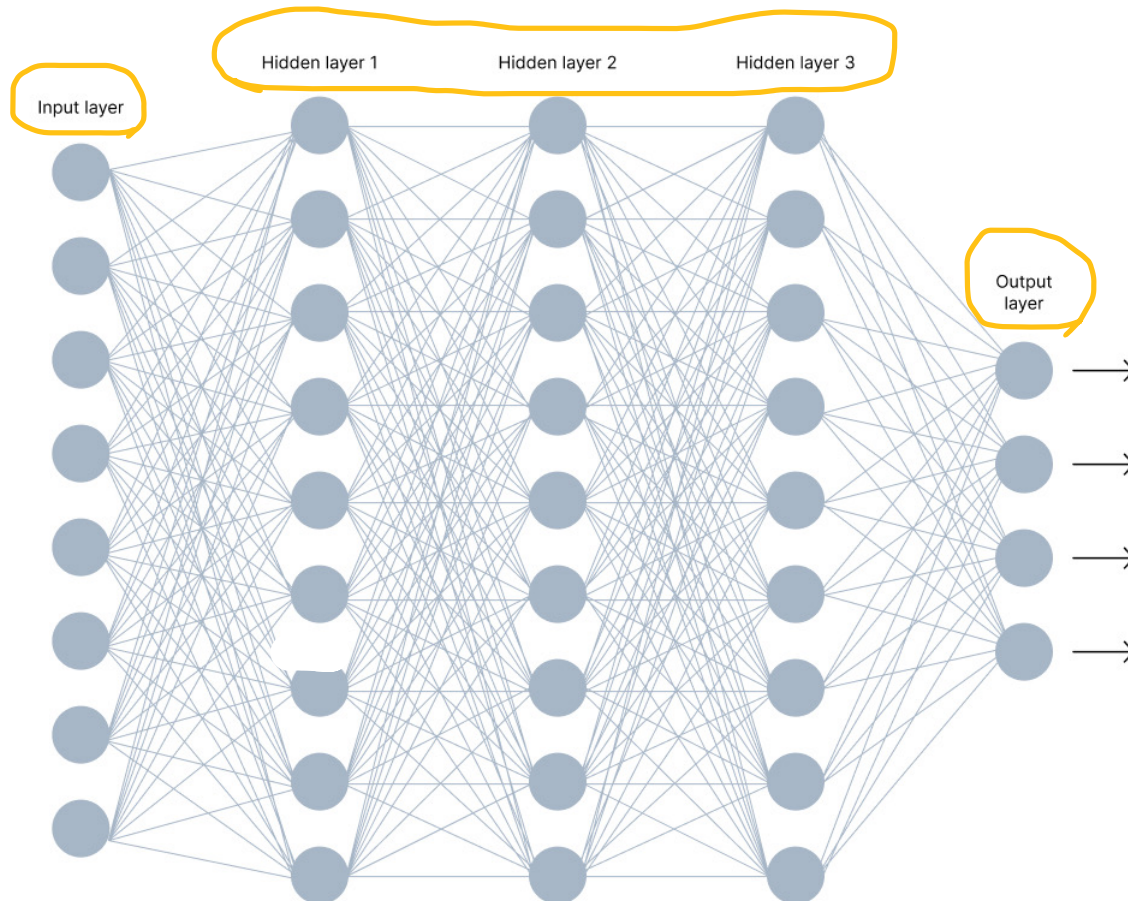
1 인공지능의 시작을 알린 퍼셉트론

(비교) 로지스틱 회귀 모델



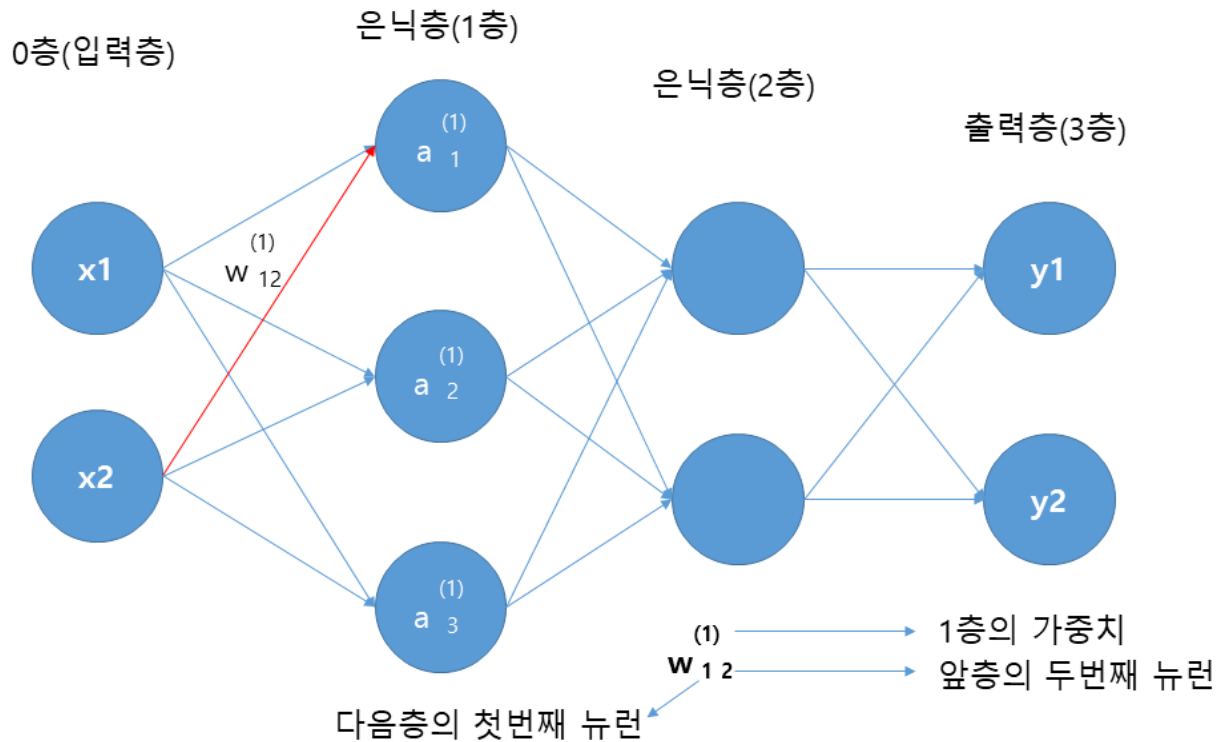
1 인공지능의 시작을 알린 퍼셉트론

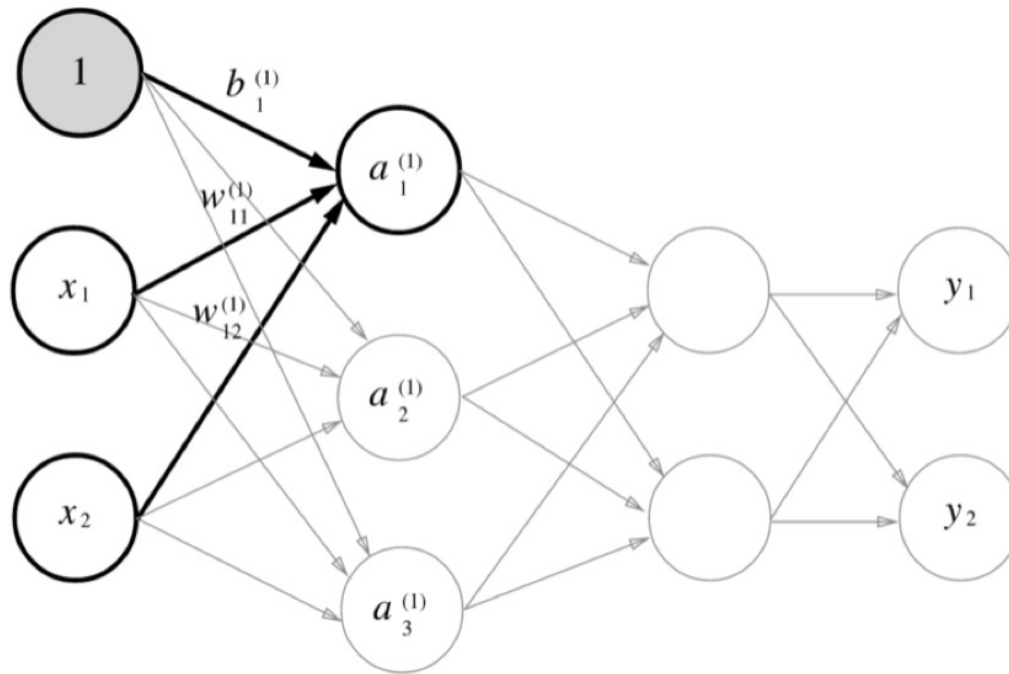
- 심층 신경망 : Deep Neural Network





3층 신경망





$$a_1 = w_{11}x_1 + w_{12}x_2 + b_1$$

$$a_2 = w_{21}x_1 + w_{22}x_2 + b_2$$

$$a_3 = w_{31}x_1 + w_{32}x_2 + b_3$$

$$\begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} w_{11} & w_{21} & w_{31} \\ w_{12} & w_{22} & w_{32} \end{bmatrix} + \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix}$$



2 퍼셉트론의 과제



2 퍼셉트론의 과제

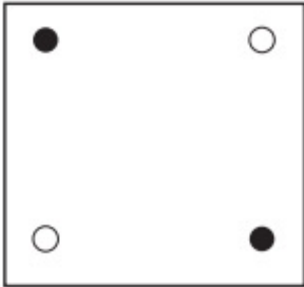
● 퍼셉트론의 과제

- 퍼셉트론이 완성되고 아달라인에 의해 보완되며 드디어 현실 세계의 다양한 문제를 해결하는 인공지능이 개발될 것으로 기대했음
- 곧 퍼셉트론의 한계가 보고
- 퍼셉트론의 한계가 무엇이었는지 알고 이를 극복하는 과정을 이해하는 것은 우리에게도 매우 중요함
- 이것을 해결한 것이 바로 딥러닝이기 때문임
- 지금부터는 퍼셉트론의 한계와 이를 해결하는 과정을 보며 신경망의 기본 개념을 확립해 보자



2 퍼셉트론의 과제

▼ 그림 7-3 | 사각형 종이에 놓인 검은색 점 두 개와 흰색 점 두 개

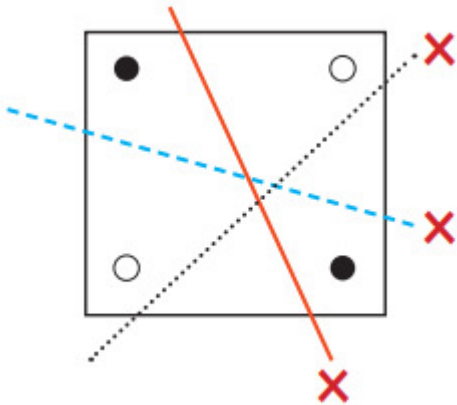


2 퍼셉트론의 과제

● 퍼셉트론의 과제

- 사각형 종이에 검은색 점 두 개와 흰색 점 두 개가 놓여 있음
- 이 네 점 사이에 직선을 하나 긋는다고 하자
- 이때 직선의 한쪽 편에는 검은색 점만 있고, 다른 한쪽에는 흰색 점만 있게끔 선을 그을 수 있을까?

▼ 그림 7-4 | 선으로는 같은 색끼리 나눌 수 없다: 퍼셉트론의 한계





2 퍼셉트론의 과제

● 퍼셉트론의 과제

- 선을 여러 개 아무리 그어 보아도 하나의 직선으로는 흰색 점과 검은색 점을 구분할 수 없음
- 퍼셉트론이나 아달라인은 모두 2차원 평면상에 직선을 긋는 것만 가능함
- 이 예시는 경우에 따라 선을 아무리 그어도 해결되지 않는 상황이 있다는 것을 말해 줌



3 XOR 문제



3 XOR 문제

● XOR 문제

- 이것이 퍼셉트론의 한계를 설명할 때 등장하는 XOR(exclusive OR) 문제
- XOR 문제는 논리 회로에 등장하는 개념
- 컴퓨터는 두 가지의 디지털 값, 즉 0과 1을 입력해 하나의 값을 출력하는 회로가 모여 만들어지는데, 이 회로를 '게이트(gate)'라고 함
- 그림 7-5는 AND 게이트, OR 게이트, XOR 게이트에 대한 값을 정리한 것
- AND게이트는 x_1 과 x_2 둘 다 1일 때 결괏값이 1로 출력
- OR 게이트는 둘 중 하나라도 1이면 결괏값이 1로 출력
- XOR 게이트는 둘 중 하나만 1일 때 1이 출력



3 XOR 문제

▼ 그림 7-5 | AND, OR, XOR 게이트에 대한 진리표

AND
(논리곱)
두 개 모두 1일 때 1

x_1	x_2	결괏값
0	0	0
0	1	0
1	0	0
1	1	1

OR
(논리합)
두 개 중 한 개라도 1이면 1

x_1	x_2	결괏값
0	0	0
0	1	1
1	0	1
1	1	1

XOR
(배타적 논리합)
하나만 1이어야 1

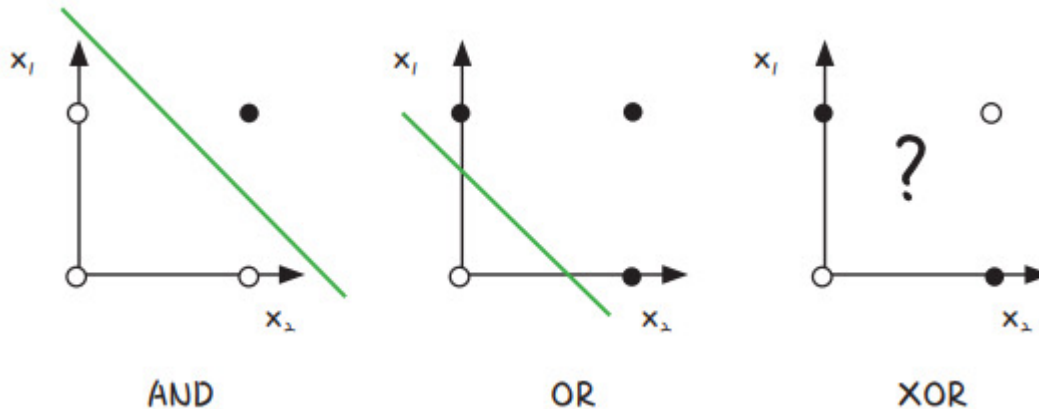
x_1	x_2	결괏값
0	0	0
0	1	1
1	0	1
1	1	0

3 XOR 문제

● XOR 문제

- 그림 7-5를 각각 그래프로 좌표 평면에 나타내 보자
- 결괏값이 0이면 흰색 점으로, 1이면 검은색 점으로 나타낸 후 조금 전처럼 직선을 그어 위 조건을 만족할 수 있는지 보자

▼ 그림 7-6 | AND, OR, XOR 진리표대로 좌표 평면에 표현한 후 선을 그어 색이 같은 점끼리 나누기(XOR는 불가능)





3 XOR 문제

● XOR 문제

- AND와 OR 게이트는 직선을 그어 결괏값이 1인 값(검은색 점)을 구별할 수 있음
- XOR의 경우 선을 그어 구분할 수 없음
- 이는 인공지능 분야의 선구자였던 MIT의 마빈 민스키(Marvin Minsky) 교수가 1969년에 발표한 "퍼셉트론즈(Perceptrons)"라는 논문에 나오는 내용
- '뉴런 → 신경망 → 지능'이라는 도식에 따라 '퍼셉트론 → 인공 신경망 → 인공지능'이 가능하리라 꿈꾸었던 당시 사람들은 이것이 생각처럼 쉽지 않다는 사실을 깨닫게 됨
- 알고 보니 간단한 XOR 문제조차 해결할 수 없었던 것
- 이 논문 이후 인공지능 연구가 한동안 침체기를 겪게 됨
- 이 문제는 두 가지 방법이 순차적으로 개발되면서 해결
- 하나는 다층 퍼셉트론(multilayer perceptron), 그리고 또 하나는 오차 역전파(back propagation)