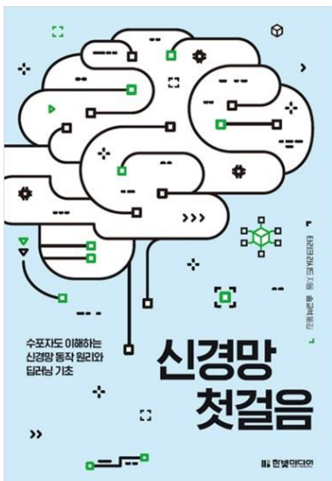
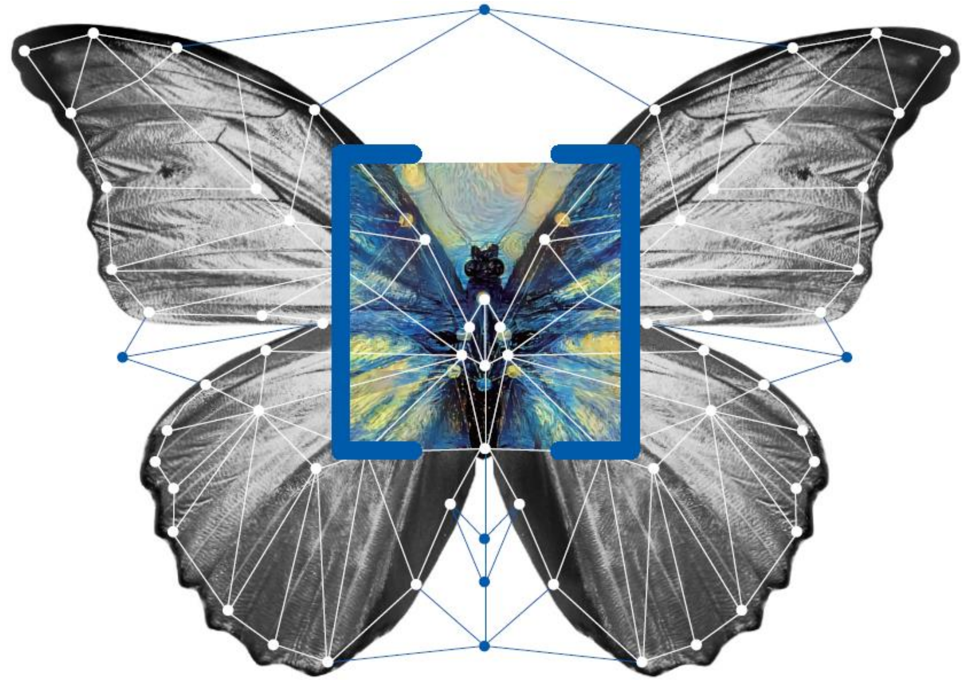




MACHINE LEARNING 기계 학습

신경망 학습

PREVIEW

- 신경망
  - 기계 학습 역사에서 가장 오래된 기계 학습 모델이며, 현재 가장 다양한 형태를 가짐
  - 1950년대 퍼셉트론 → 1980년대 다층 퍼셉트론
  - 신경망은 딥러닝의 기초가 됨

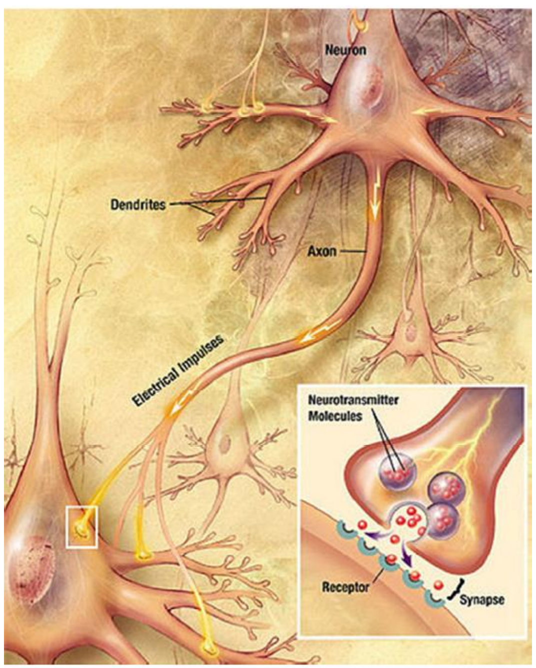
발상과 전개

- 두 줄기 연구의 시너지
  - 컴퓨터 과학
    - 계산 능력의 획기적 발전으로 지능 처리에 대한 욕구
  - 의학
    - 두뇌의 정보처리 방식 연구 → 얼마간의 성과 (뉴런의 동작 이해 등)

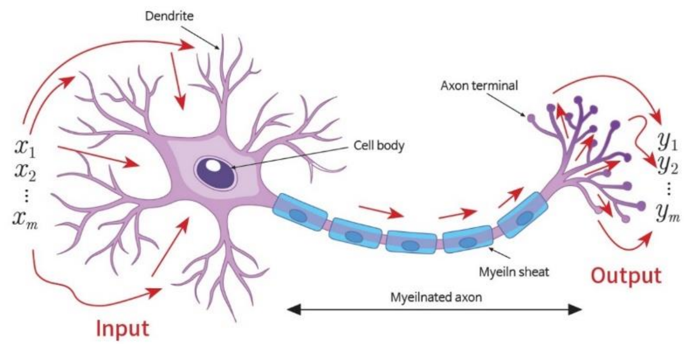
- 뇌의 정보처리 모방하여 인간에 필적하는 지능 컴퓨터에 도전
  - 인공 신경망 (ANN; Artificial Neural Network)이 대표적
  - 인공 신경망은 생물학적인 신경망에서 영감을 받아서 만들어진 컴퓨팅 구조

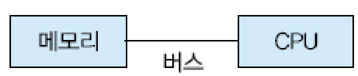
컴퓨터와 두뇌의 비교

- 폰 노이만 컴퓨터
  - 순차 명령어 처리기
- 사람의 뉴런
  - 두뇌의 가장 작은 정보처리 단위
  - 세포체는cell body 간단한 연산, 수상돌기는dendrite 신호 수신, 축삭은axon 처리 결과를 전송
  - 사람은 10^{11}개 정도의 뉴런을 가지며, 뉴런은 1000개 가량 다른 뉴런과 연결되어 있어 10^{14}개 정도의 연결



사람의 뉴런의 구조와 동작





(a) 폰 노이만 컴퓨터의 구조

(b) 사람 뇌의 정보 처리 단위인 뉴런

전통적인 컴퓨터 vs 인공신경망

	기존의 컴퓨터	인간의 두뇌
처리소자의 개수	10^8개의 트랜지스터	10^{10}개의 뉴런
처리소자의 속도	10^{12}Hz	10^2Hz
학습기능	없음	있음
계산 스타일	중앙 집중식, 순차적인 처리	분산 병렬 처리

신경망의 간단한 역사

- 간략한 역사
  - 1943, McCulloch과 Pitts 최초 신경망 제안
  - 1949, Hebb의 학습 알고리즘
  - 1958, Rosenblatt 퍼셉트론
  - Widrow와 Hoff, Adaline과 Madaline
  - 1960대, 신경망의 과대 포장
  - 1969, Minsky와 Papert, Perceptrons라는 저서에서 퍼셉트론 한계 지적
    - 퍼셉트론은 선형 분류기에 불과하고 XOR도 해결 못함
    - 이후 신경망 연구 퇴조
  - 1986, Rumelhart, Hinton, 그리고 Williams, 다층 퍼셉트론과 오류 역전파 학습 알고리즘
    - 필기 숫자 인식같은 복잡하고 실용적인 문제에 높은 성능
    - 신경망 연구 다시 활기 찾음
    - 현재 가장 널리 활용되는 문제 해결 도구
  - 신경망 연구 부활
  - 1990년대 SVM에 밀리는 형국
  - 2000년대 딥러닝이 실현되어 신경망이 기계 학습의 주류 기술로 자리매김

수학적 모델로서의 신경망

- 신경망 특성
  - 학습 가능 : 데이터만 주어지면 신경망은 예제로부터 배울 수 있음
  - 뛰어난 일반화 능력 : 몇 개의 소자가 오동작하더라도 전체적으로는 큰 문제가 없음
  - 병렬 처리 가능
  - 현실적 문제에서 우수한 성능
  - 다양한 문제 해결 도구 (분류, 예측, 함수 근사화, 합성, 평가, …)

- 절반의 성공
  - 인간 지능에 필적하는 컴퓨터 만들지 못함
  - 제한된 환경에서 실용적인 시스템 만드는데 크게 기여 (실용적인 수학적 모델로서 자리매김)

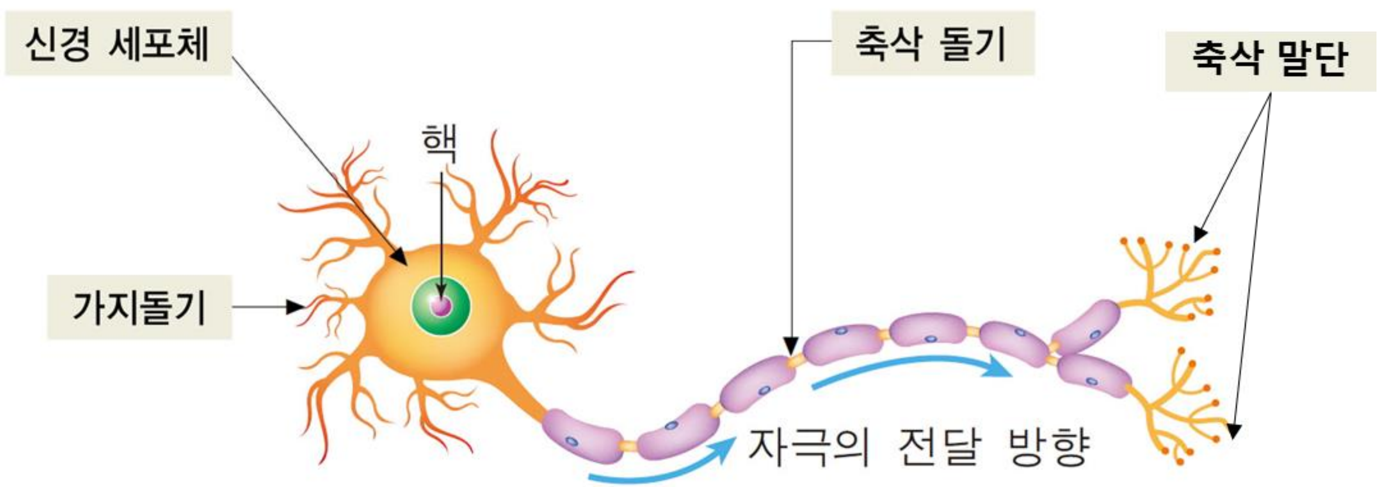
퍼셉트론

- 새로운 개념들 등장
  - 층
  - 노드와 가중치
  - 학습
  - 활성 함수

- 비록 분명한 한계를 가지지만 MLP의 초석이 됨

뉴런의 구성과 동작원리

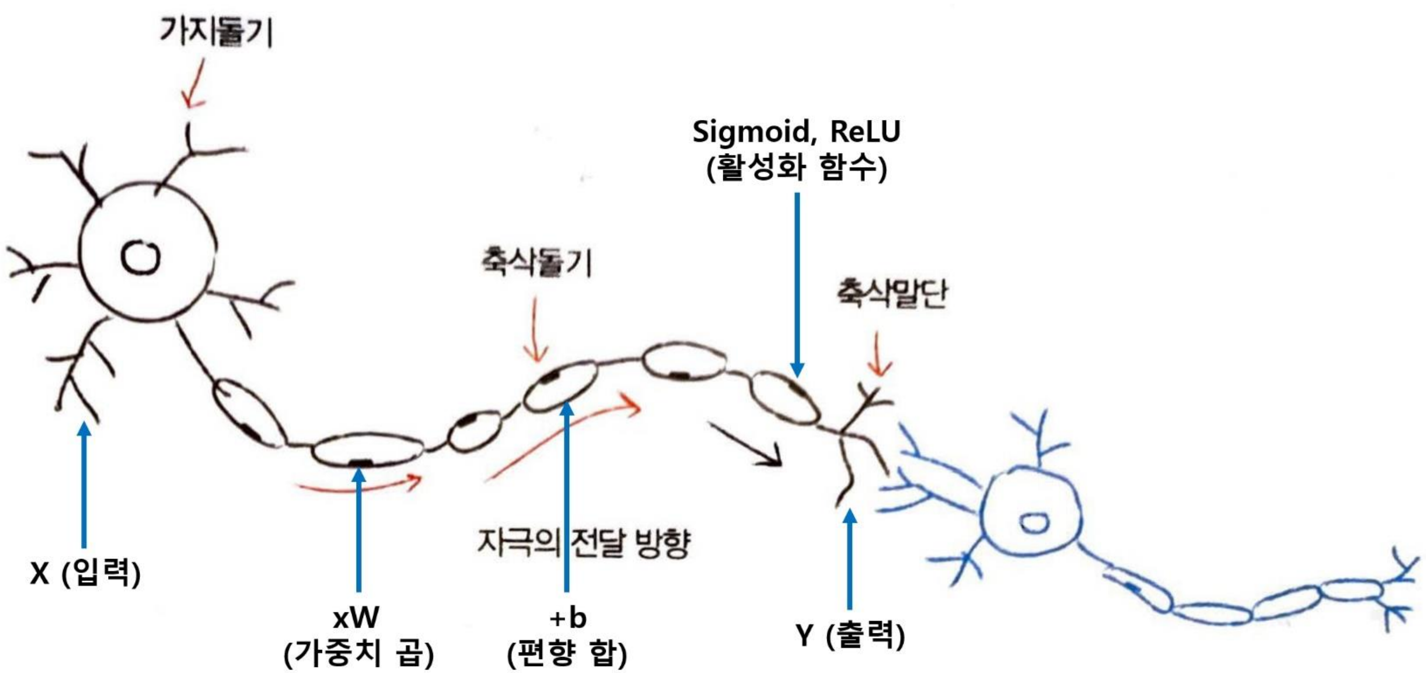
- 뉴런의 기본 동작
  - 신호를 받아들이고, 이 신호가 축삭 돌기를 지나 축삭 말단으로 전달
  - 축삭 돌기를 지나는 동안 신호가 약해지거나, 너무 약해서 축삭 말단까지 전달되지 않거나 혹은 강하게 전달되기도 함
  - 축삭 말단까지 전달된 신호는 연결된 다음 뉴런의 가지 돌기로 전달



실제 뉴런과 인공 뉴런

인공 뉴런의 기본

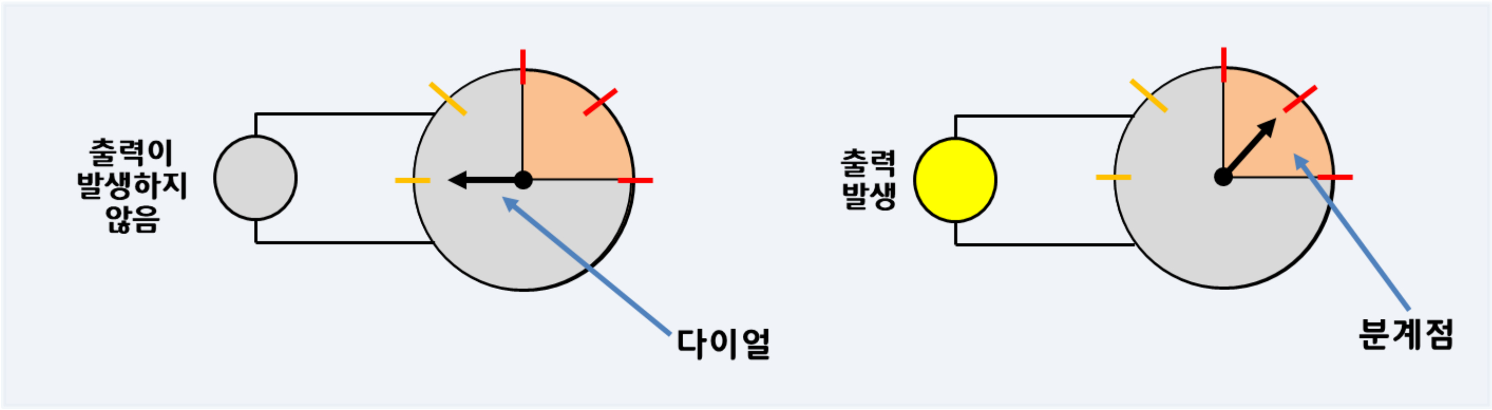
- 입력신호, 즉 입력값 X에 가중치(W)를 곱하고 편향(b)을 더한 뒤 활성화 함수(Sigmoid(시그모이드), ReLU(렐루) 등)를 거쳐 결과값 y를 만들어 냄
- 원하는 y값을 만들어내기 위해 W와 b의 값을 변경해가면서 적절한 값을 찾아냄(이러한 최적화 과정을 학습 또는 훈련)
- Y= Sigmoid(x * W + b)



뉴런의 동작원리

뉴런의 출력

- 입력 값이 어떤 **분계점(threshold)**에 도달해야 출력 발생
- Ex) 컵에 물을 채울 때 컵을 가득 채워야만 물이 넘친다.

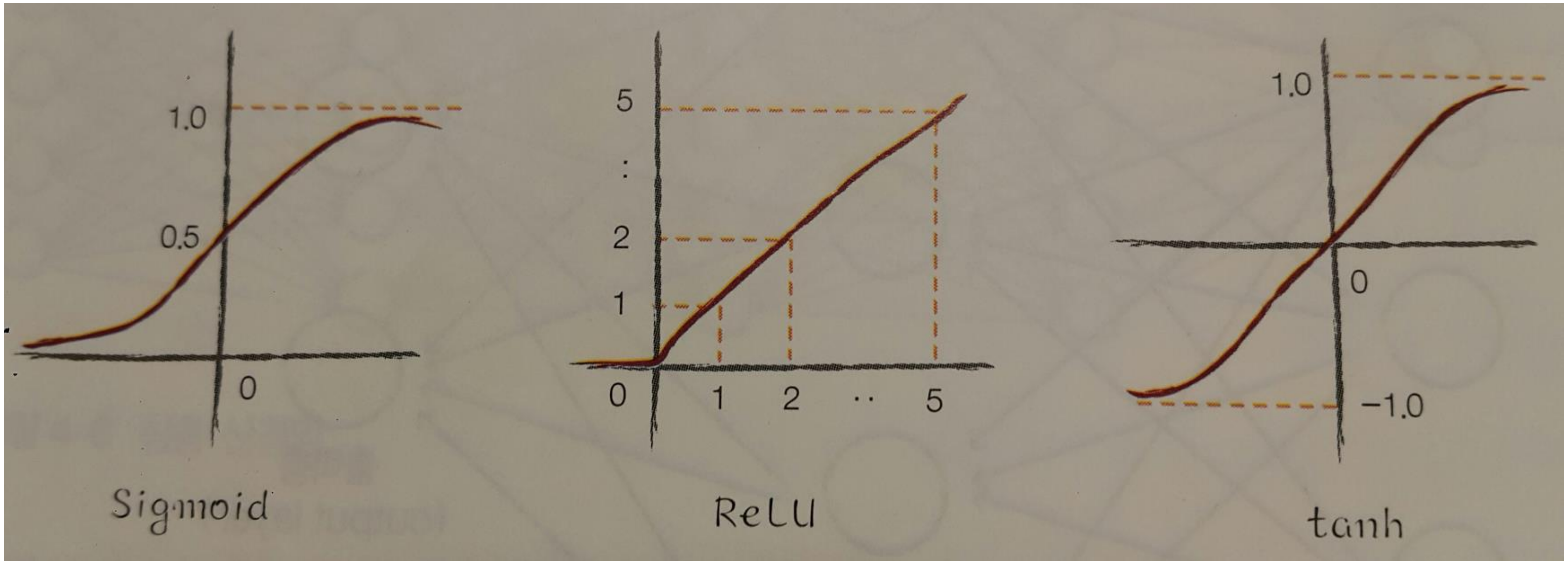


- 입력 신호를 받아 특정 분계점을 넘어서는 경우에 출력 신호를 생성해주는 함수를 활성화 함수라고 함

활성화 함수

활성화 함수

- 인공신경망을 통과해온 값을 최종적으로 어떤 값으로 만들지를 결정
- 이 함수가 인공 뉴런의 핵심 중에서도 가장 중요한 요소
- 활성화 함수
  - Sigmoid(시그모이드), ReLU(렐루), tanh(쌍곡탄젠트)



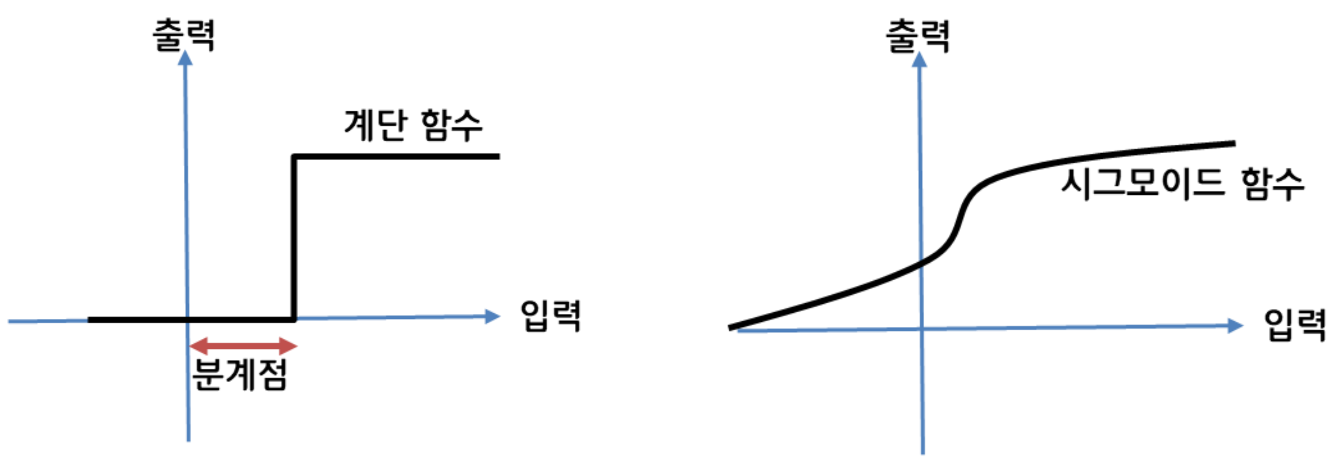
활성화 함수 예

계단 함수(step function)

- 입력 값이 작은 경우 출력 값은 0
- 입력 값이 분계점 이상이면 출력 값은 임의의 값이 출력

시그모이드 함수

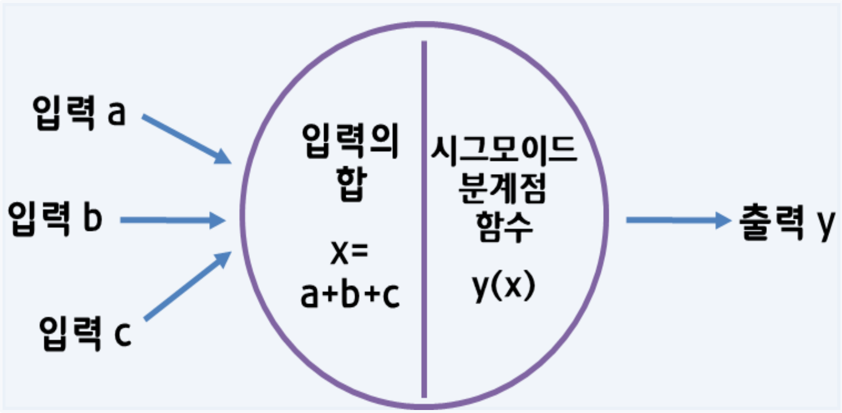
- 로지스틱 함수(logistic function)
- $y = \frac{1}{1+e^{-x}}$
- 예를 들어 x=0 일때 e^{-x} = 1 이므로 y= 1/(1+1) = ½

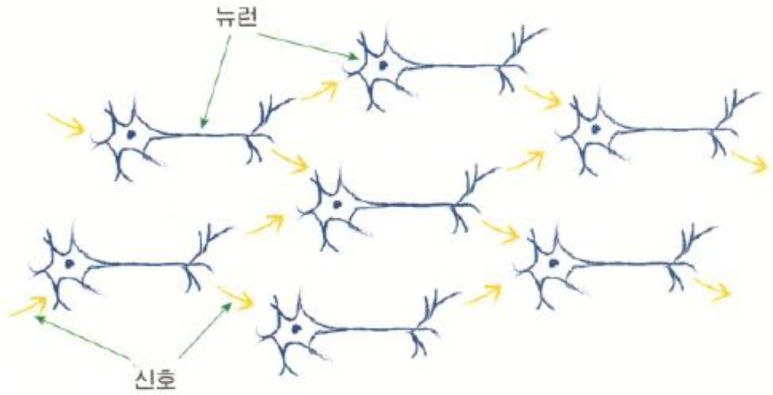


뉴런의 동작 과정

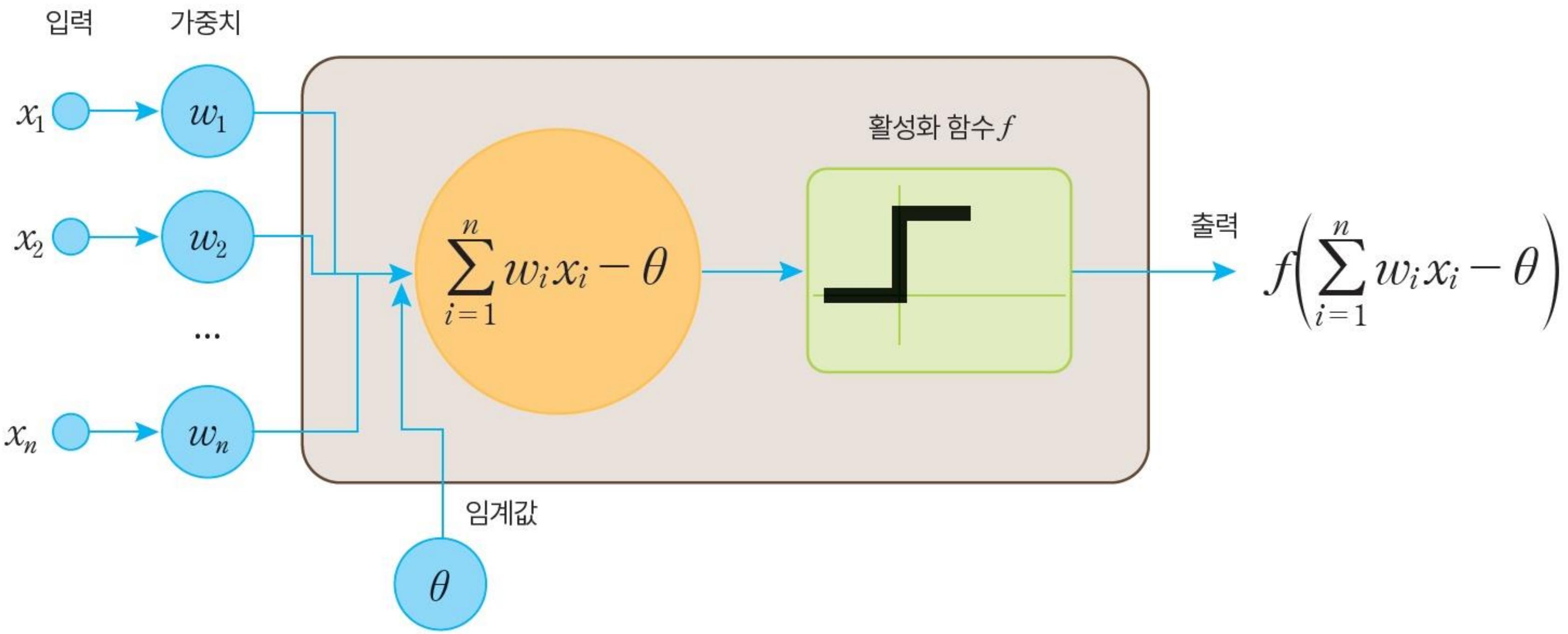
여러 개의 입력이 있을 경우

- 입력 a,b,c의 합 x > 분계점 ➔ 시그모이드 함수에서 출력X
- 입력 a,b,c의 합 x < 분계점 ➔ 시그모이드 함수에서 출력O
- 3개의 입력 값 중 1개의 입력 값만 커도 뉴런 작동 가능





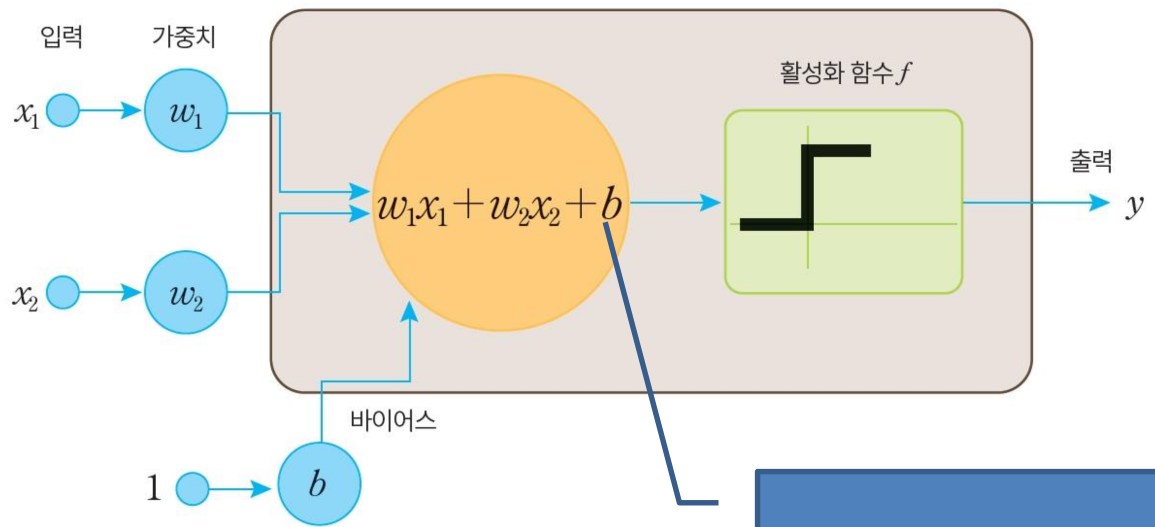
뉴런의 수학적인 모델



뉴런의 수학적인 구조

퍼셉트론

- 퍼셉트론(perceptron)은 1957년에 로젠블라트(Frank Rosenblatt)가 고안한 인공 신경망이다.



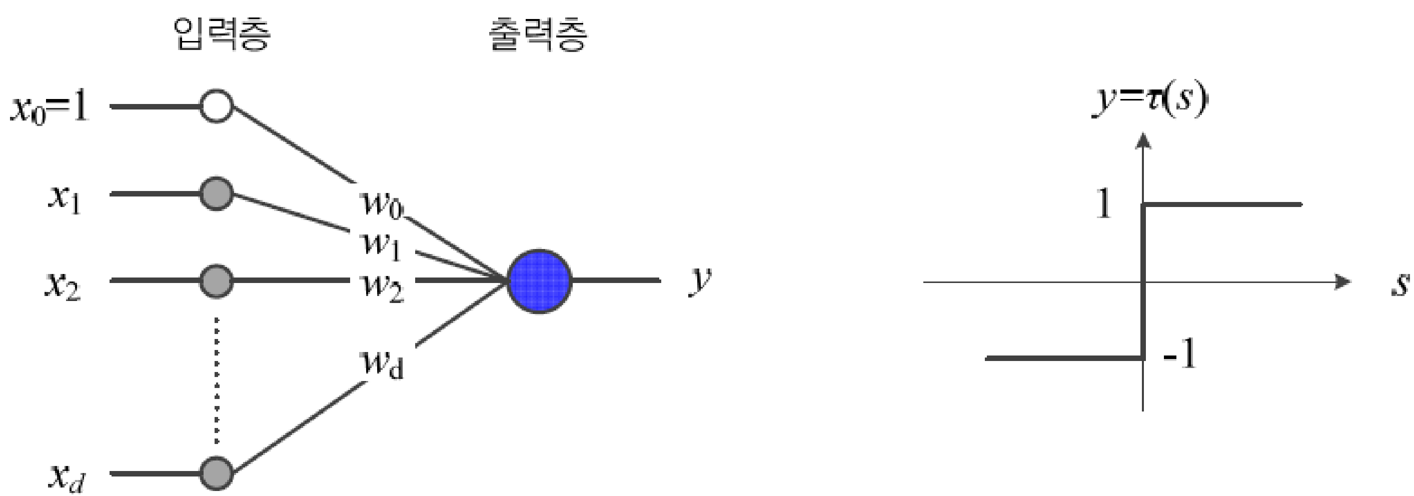
퍼셉트론에서의 뉴런의 모델

b = -θ

구조와 원리

퍼셉트론의 구조

- 입력층과 출력층을 가짐
  - 입력층은 연산을 하지 않으므로 퍼셉트론은 단일 층 구조라고 간주
- 입력층의 i번째 노드는 특징 벡터 $\mathbf{x} = (x_1, x_2, \cdots, x_d)^{\mathrm{T}}$의 요소 x_i를 담당
- 항상 1이 입력되는 바이어스 노드
- 출력층은 한 개의 노드
- i번째 입력층 노드와 출력층을 연결하는 에지는 가중치 w_i를 가짐



(a) 퍼셉트론의 구조

(b) 계단함수를 활성함수 $\tau(s)$로 이용함

퍼셉트론의 구조와 동작

구조와 원리

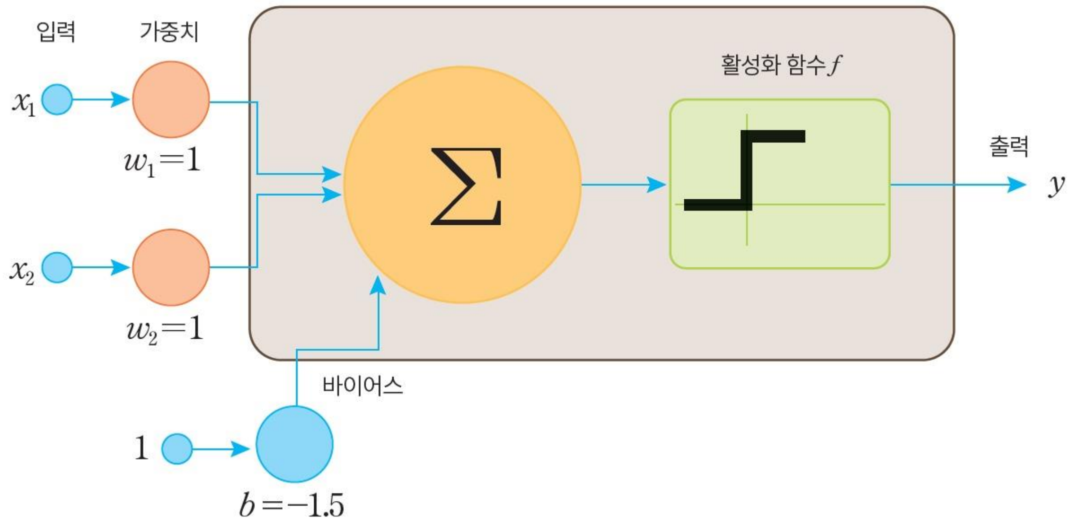
- 노드의 연산
  - 입력 노드: 받은 신호를 단순히 전달
  - 출력 노드: 합 계산과 활성 함수 계산

$$\left.\begin{aligned} & y = \tau(s) = \tau\left(\sum_{i=1}^{d} w_i x_i + b\right) = \tau(\mathbf{w}^{\mathrm{T}}\mathbf{x} + b) \\ & \text{이때} \quad \tau(s) = \begin{cases} +1, & s \ge 0 \\ -1, & s < 0 \end{cases} \end{aligned}\right\}$$

- 퍼셉트론은 선형 분류기

$$\left.\begin{aligned} & d(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b > 0 \text{ 이면} \quad \mathbf{x} \in \omega_1 \\ & d(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b < 0 \text{ 이면} \quad \mathbf{x} \in \omega_2 \end{aligned}\right\}$$

퍼셉트론은 논리 연산을 학습할 수 있을까?

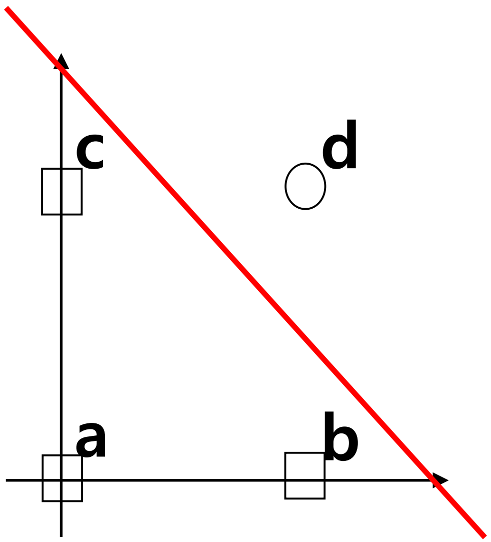


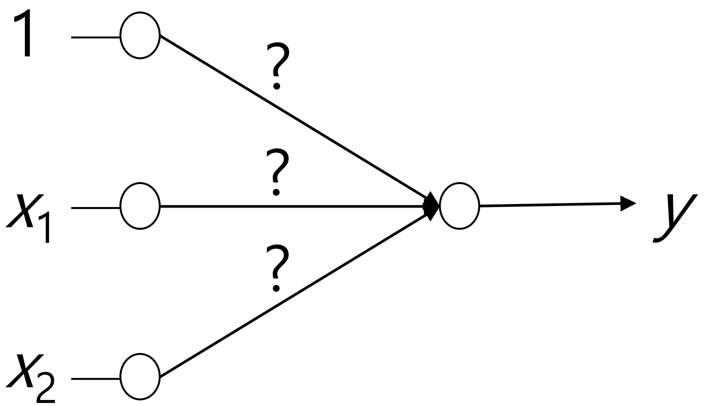
논리 연산을 하는 퍼셉트론

x1	x2	y
0	0	0
1	0	0
0	1	0
1	1	1

퍼셉트론은 논리 연산을 학습할 수 있을까?

a=(0,0)^T **b**=(1,0)^T **c**=(0,1)^T **d**=(1,1)^T





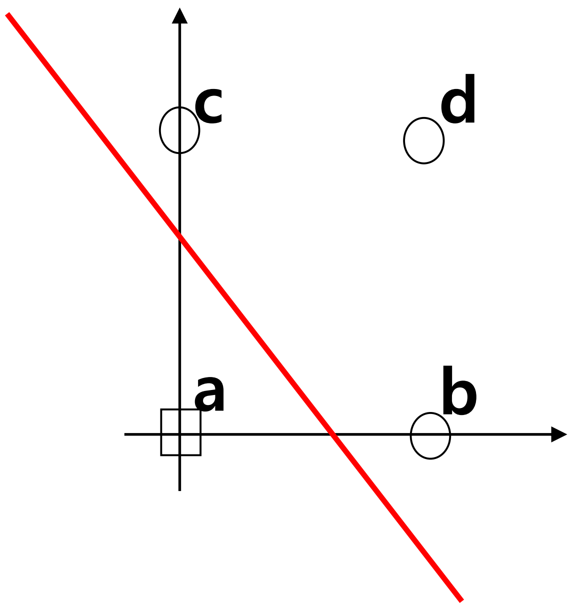
결정 직선 방정식 $d(x) = x_1 + x_2 - 1.5$

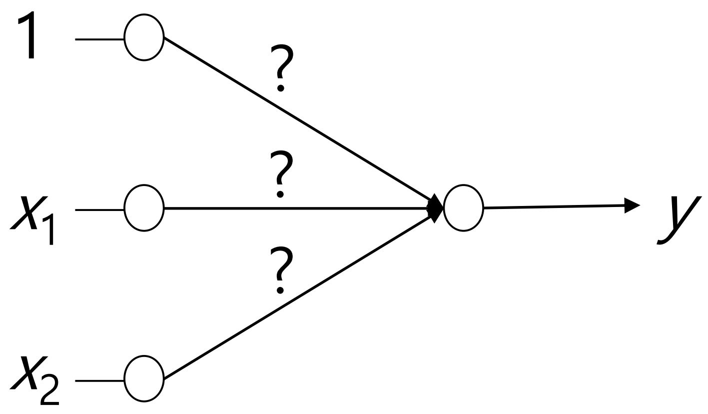
퍼셉트론 출력 계산

x1	x2	$w_1x_1+w_2x_2$	b	$w_1x_1+w_2x_2+\mathrm{b}$	출력
0	0	1*0+1*0=0	−1.5	−1.5	0
1	0	1*1+1*0=1	−1.5	−0.5	0
0	1	1*0+1*1=1	−1.5	−0.5	0
1	1	1*1+1*1=2	−1.5	+0.5	1

퍼셉트론의 OR 연산 학습

a=$(0,0)^T$ **b**=$(1,0)^T$ **c**=$(0,1)^T$ **d**=$(1,1)^T$





이 퍼셉트론은 **w**=$(1,1)^T$, b=-0.5, 활성 함수는 계단 함수

따라서 결정 직선은 $d(\mathbf{x}) = x_1 + x_2 - 0.5$

퍼셉트론 학습 알고리즘

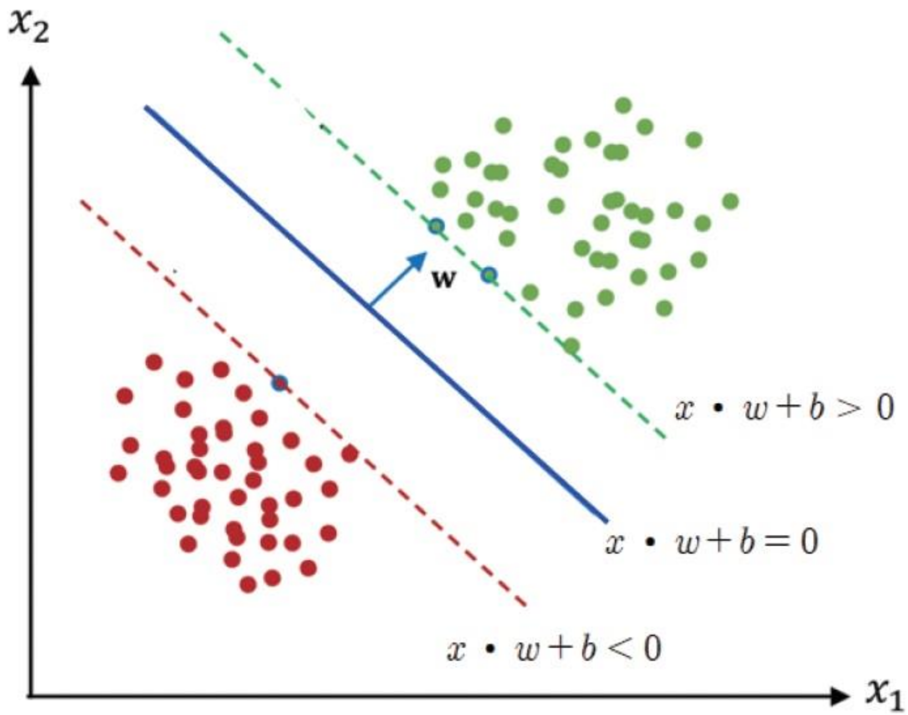
- 학습 세트 : m개의 샘플
- $w \cdot x = w_0(t)x_0 + w_1(t)x_1 + \ldots + w_n(t)x_n$ 벡터의 내적
- $W_i(t)$는 시간 t에서의 가중치 i의 값

```
input: 학습 데이터 (x^1, d^1), ..., (x^m, d^m)

① 모든 w들과 바이어스 b를 0 또는 작은 난수로 초기화한다.
② while (가중치가 변경되지 않을 때까지 반복)
③     각 학습 데이터 x^k와 정답 d^k에 대하여
④             y^k(t) = f(w • x)
⑤             if d^k == y^k(t)
⑥                     continue
⑦             else
⑧                     모든 가중치 w_i에 대하여 w_i(t+1) = w_i(t) + η • (d^k − y^k(t)) • x_i^k
```

선형 분류 가능 문제

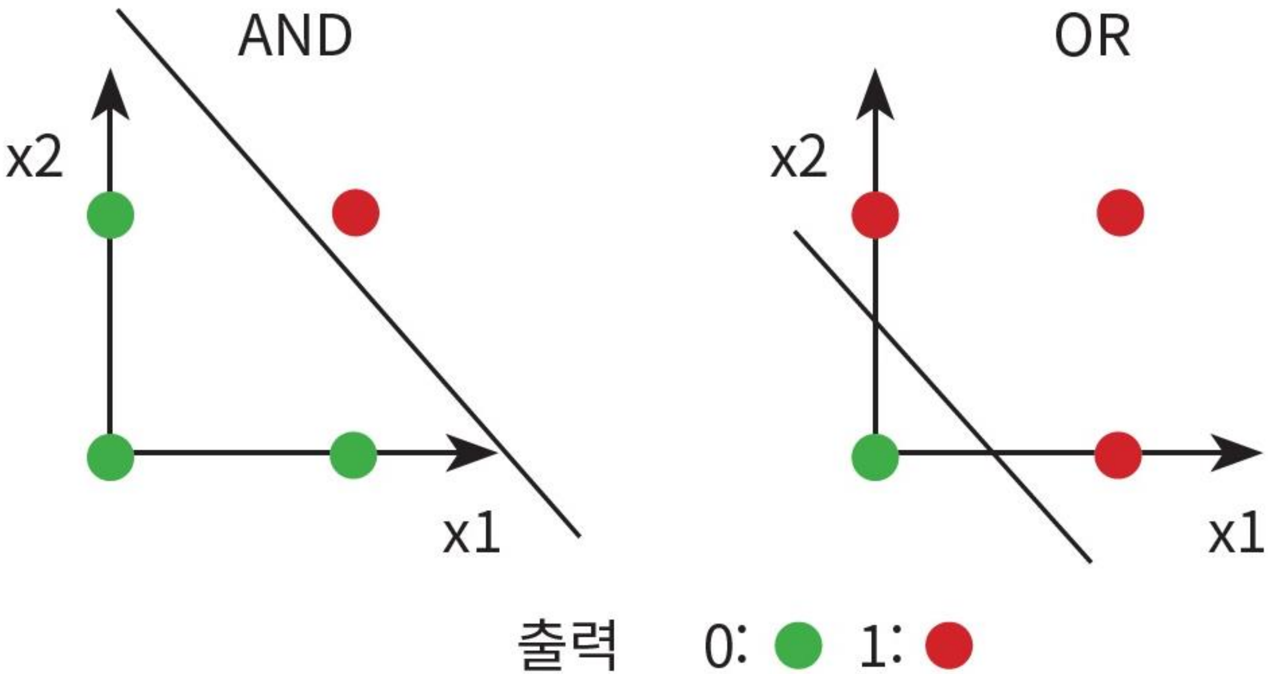
- 패턴 인식 측면에서 보면 퍼셉트론은 직선을 이용하여 입력 패턴을 분류하는 선형 분류자(linear classifier)의 일종이라고 말할 수 있다.



선형 분류자

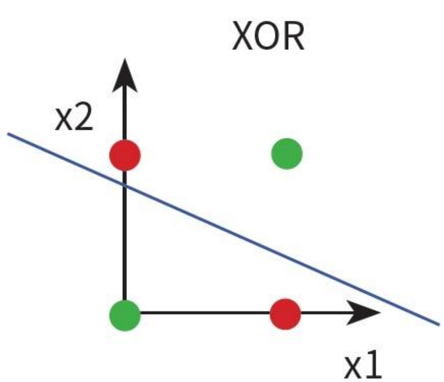
선형 분류 가능 문제

- AND나 OR 연산은 모두 선형 분리 가능한 문제에 속한다

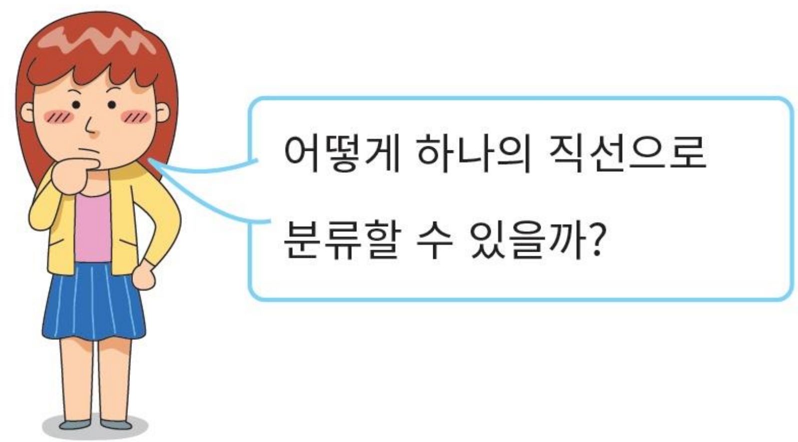


XOR 학습 문제

- XOR 연산 : 2개의 입력이 같으면 0, 2개의 입력이 서로 다르면 1이 되는 논리적인 연산
- Minsky와 Papert는 1969년에 발간된 책 “Perceptrons”에서 1개의 레이어(layer, 계층)으로 구성된 퍼셉트론은 XOR 문제를 학습할 수 없다는 것을 수학적으로 증명
  - 당시에 진행 중이던 모든 신경망 연구가 중단 (신경망의 암흑기)



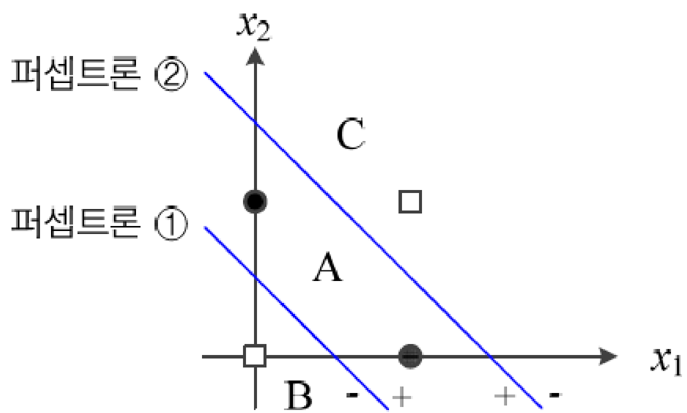
선형 분리 불가능한 문제



다층 퍼셉트론

퍼셉트론 2개를 사용한 XOR 문제의 해결

- 퍼셉트론①과 퍼셉트론②가 모두 +1이면 ● 부류이고 그렇지 않으면 □ 부류임



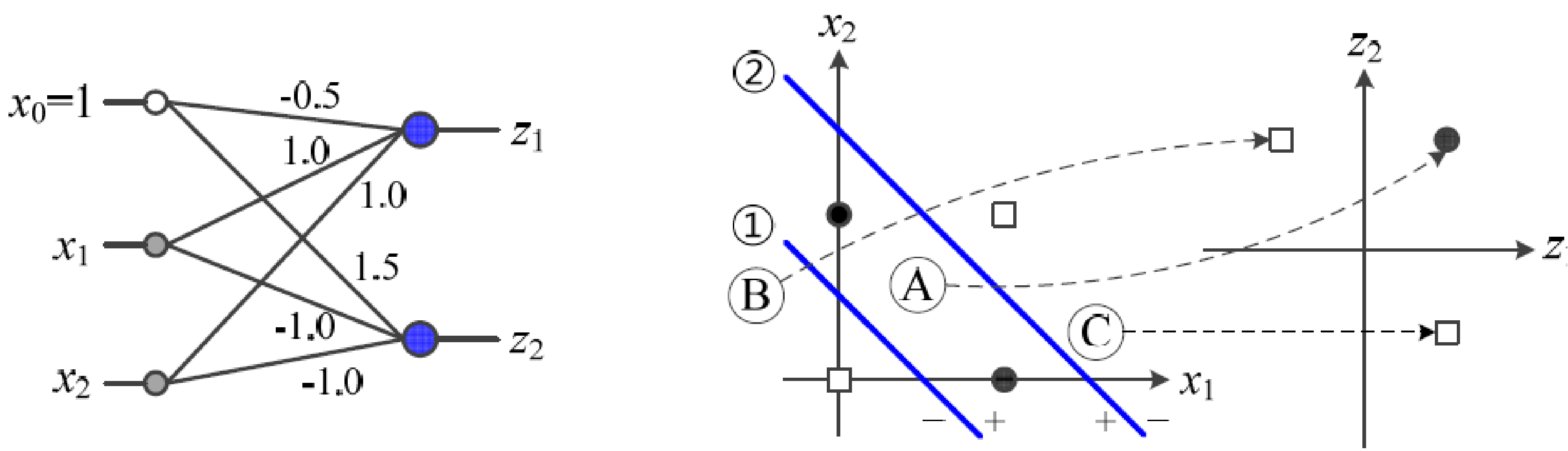
(a) 퍼셉트론 2개를 이용한 공간분할

(b) 퍼셉트론 2개

XOR 문제의 해결

다층 퍼셉트론

- 퍼셉트론 2개를 병렬로 결합하면
  - 원래 공간 $\mathbf{x} = (x_1, x_2)^{\mathrm{T}}$를 새로운 특징 공간 $\mathbf{z} = (z_1, z_2)^{\mathrm{T}}$로 변환
  - 새로운 특징 공간 $\mathbf{z}$에서는 선형 분리 가능함

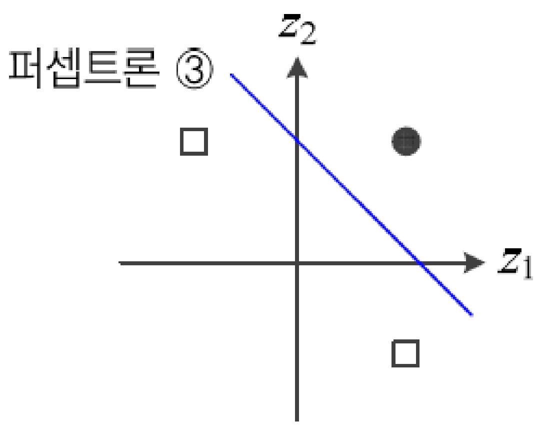


(a) 두 퍼셉트론을 병렬로 결합　　(b) 원래 특징 공간 **x**를 새로운 특징 공간 **z**로 변환

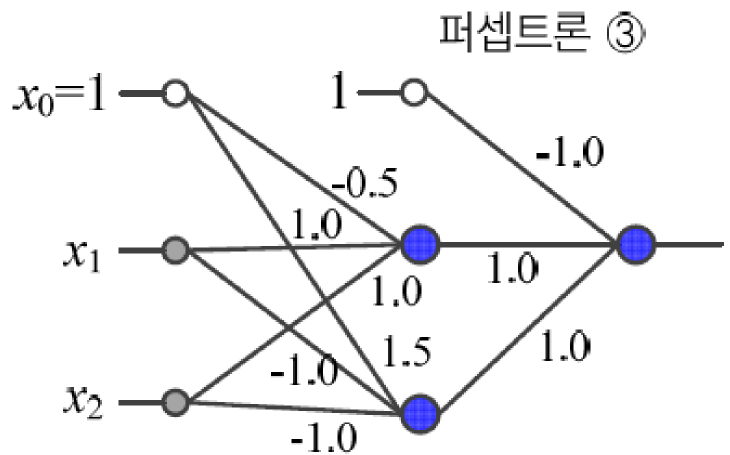
특징 공간의 변환

다층 퍼셉트론

- 퍼셉트론 1개를 순차 결합하면,
  - 새로운 특징 공간 **z**에서 선형 분리를 수행하는 퍼셉트론③을 순차 결합하면, (b)의 다층 퍼셉트론이 됨



(a) 새로운 특징 공간에서 분할



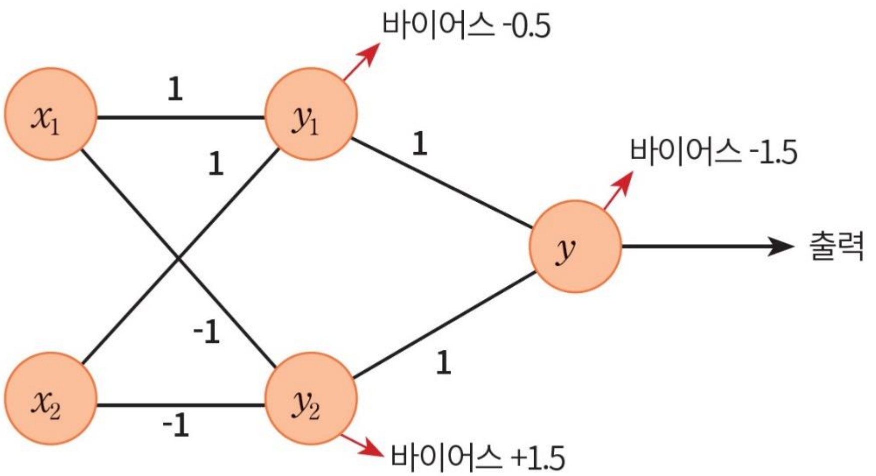
(b) 퍼셉트론 3개를 결합한 다층 퍼셉트론

다층 퍼셉트론

  - 이 다층 퍼셉트론은 훈련집합에 있는 4개 샘플 $\begin{pmatrix}0\\0\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix}\begin{pmatrix}1\\1\end{pmatrix}$을 제대로 분류하나?

다층 퍼셉트론(은닉층)

- XOR 문제는 입력층과 출력층 사이에 은닉층을 두면 쉽게 풀 수 있다



다층 퍼셉트론에서 XOR 문제 해결

x1	x2	y1	y2	y	출력
0	0	0	1	0	0
1	0	1	1	1	1
0	1	1	1	1	1
1	1	1	0	0	0

다층 퍼셉트론 학습 알고리즘

- Minsky와 Papert는 다층 퍼셉트론을 학습시키는 알고리즘을 찾기가 아주 어려울 것이라고 예언하였다. 예언에 맞았을까? 그렇지 않았다. 1980년대 중반에 Rumelhart와 Hinton 등은 다층 퍼셉트론을 위한 학습 알고리즘을 재발견하게 된다.