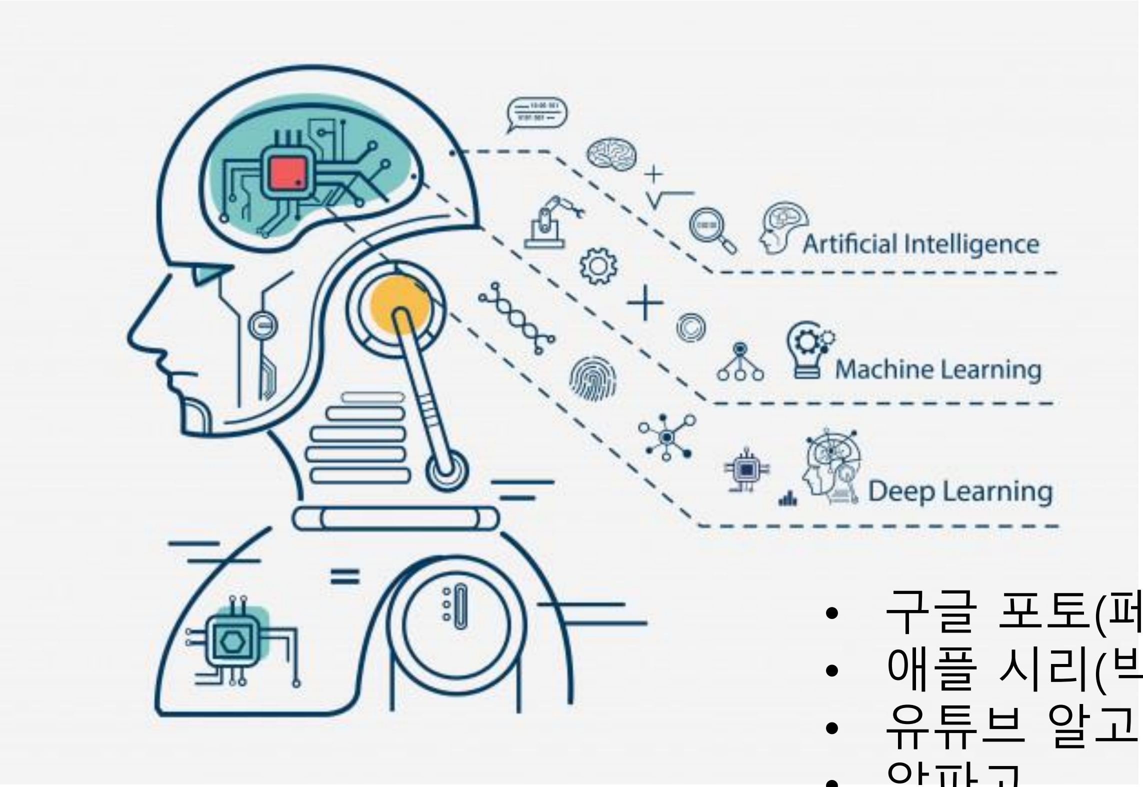
10.1.1-3: 딥러닝 (퍼셉트론)

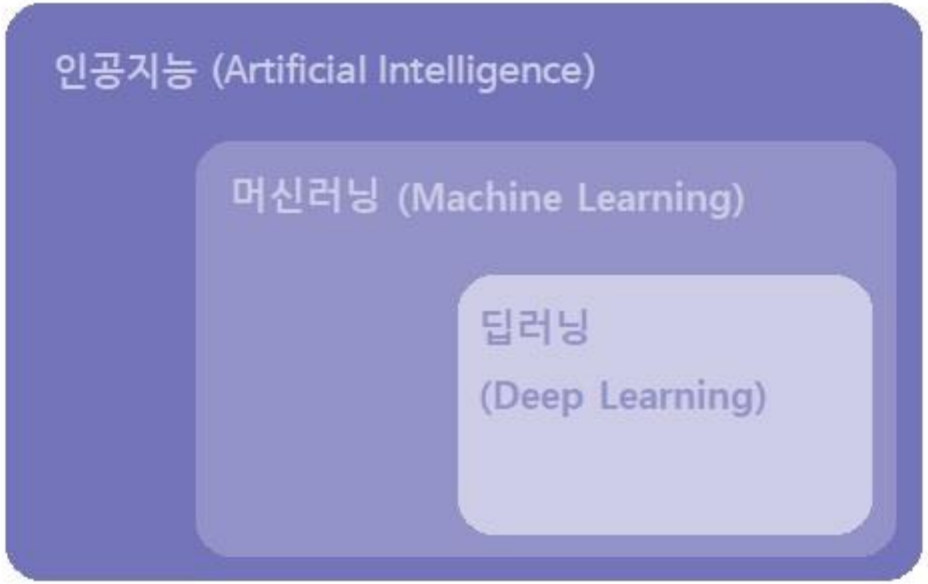
핸즈온 머신러닝 2판
- 오렐리앙 제롱

화학소재솔루션센터
김민근

KRICT 한국화학연구원

인공지능 > 머신러닝 > 딥러닝

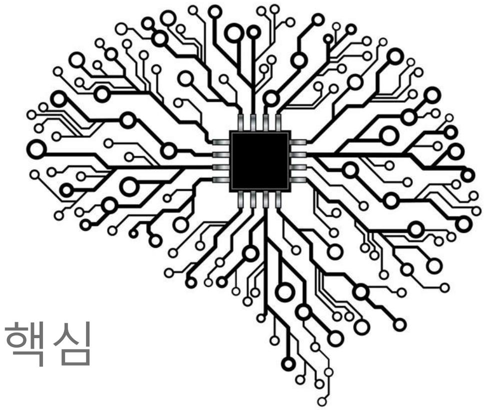




- 구글 포토(페이스북)
- 애플 시리(빅스비)
- 유튜브 알고리즘(Spotify)
- 알파고
- 무인주행 자동차

- 데이터 多
- 컴퓨터 하드웨어 발전
- 훈련 알고리즘 향상
- 이론상 제한 but 실전 X (지역 최저점)
- 투자&진보

인공 신경망artificial neural network



- 뇌에 있는 생물학적 뉴런의 네트워크에서 영감을 받은 머신러닝 모델 > 딥러닝의 핵심
- 1943년 신경생리학자 워런 매컬러와 수학자 월터 피츠에 의해 처음 소개
- 명제 논리를 사용해 동물 뇌의 생물학적 뉴런이 복잡한 계산을 위해 어떻게 상호작용하는지 에 대한 간간한 계산 모델을 제시

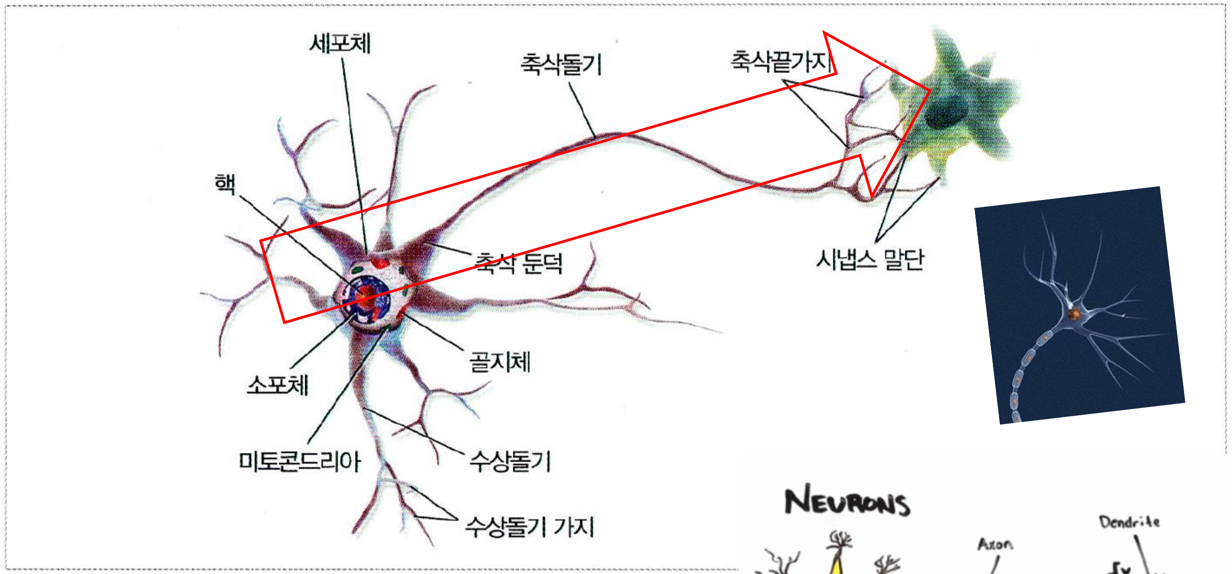
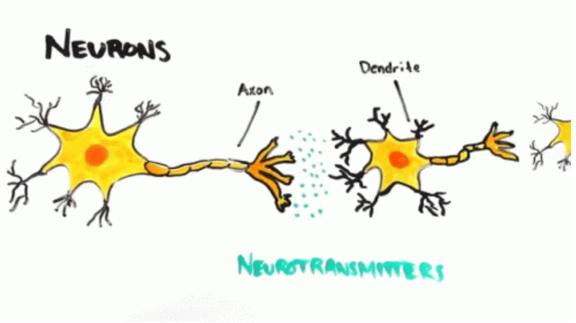


그림 10-1 생물학적 뉴런[8]



생물학적 뉴런 > *퍼셉트론*

- 동물의 뇌에서 발견되는 세포
- 핵을 포함하는 세포체와 복잡한 요소들로 구성
- 시냅스(말단) ~ 다른 뉴런의 수상돌기나 세포체에 연결
- 신호(짧은 전기 자극): 축삭돌기 > 시냅스: 신경전달물질(화학적 신호)
- 수십억 개로 구성된 거대한 네트워크 형성 (수천개의 다른 뉴런들과 연결)

퍼셉트론Perceptron

$$z = w_1x_1 + w_2x_2 + \cdots + w_nx_n = \mathbf{w}^T\mathbf{x}$$

식 10-1 퍼셉트론에서 일반적으로 사용하는 계단 함수(임곗값을 0으로 가정)

$$\text{heaviside}(z) = \begin{cases} 0 & z < 0\text{일 때} \\ 1 & z \geq 0\text{일 때} \end{cases} \qquad \text{sgn}(z) = \begin{cases} -1 & z < 0\text{일 때} \\ 0 & z = 0\text{일 때} \\ +1 & z > 0\text{일 때} \end{cases}$$

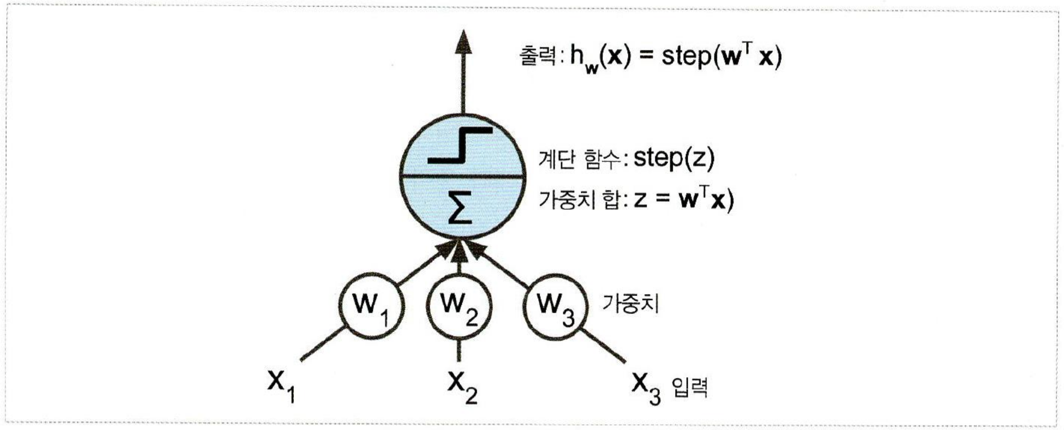
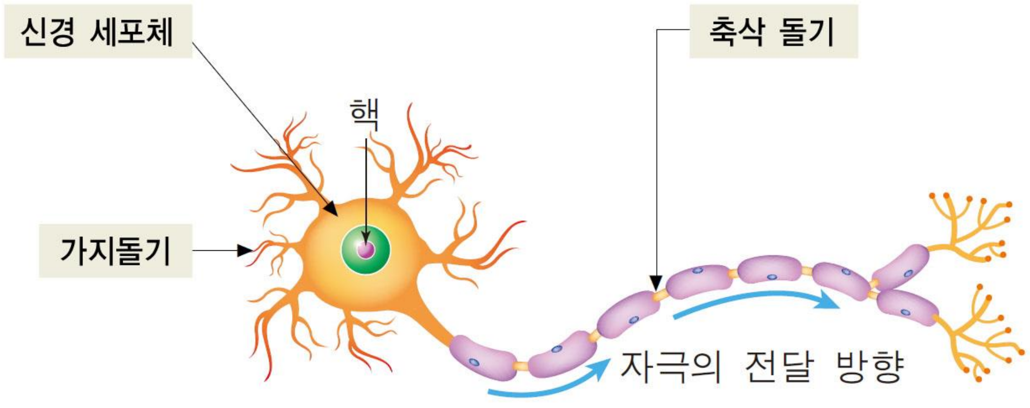
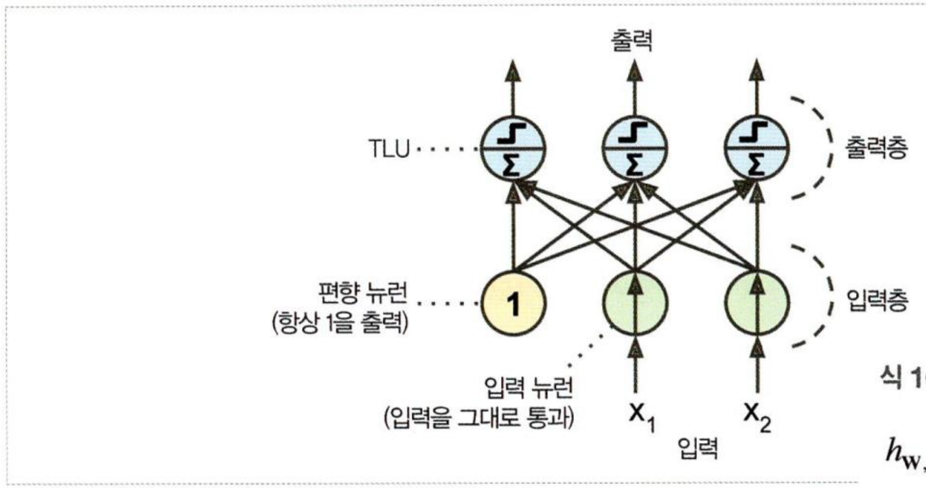


그림 10-4 TLU: 입력의 가중치 합을 계산한 다음 계단 함수를 적용하는 인공 뉴런

- 프랑크 로젠블라트가 1957년에 고안한 알고리즘
- 신경망(딥러닝)의 기원이 되는 알고리즘
- TLU (threshold logic unit) / LTU (linear threshold unit)
- 다수의 신호를 입력으로 받아 하나의 신호를 출력
- 하나의 TLU >> 로지스틱 회귀/선형 SVM 분류기





식 10-2 완전 연결 층의 출력 계산

$$h_{\mathbf{W},\mathbf{b}}(\mathbf{X}) = \phi(\mathbf{X}\mathbf{W} + \mathbf{b})$$

그림 10-5 입력 뉴런 두 개, 편향 뉴런 한 개, 출력 뉴런 세 개로 구성된 퍼셉트론의 구조

KRICT 한국화학연구원

퍼셉트론Perceptron

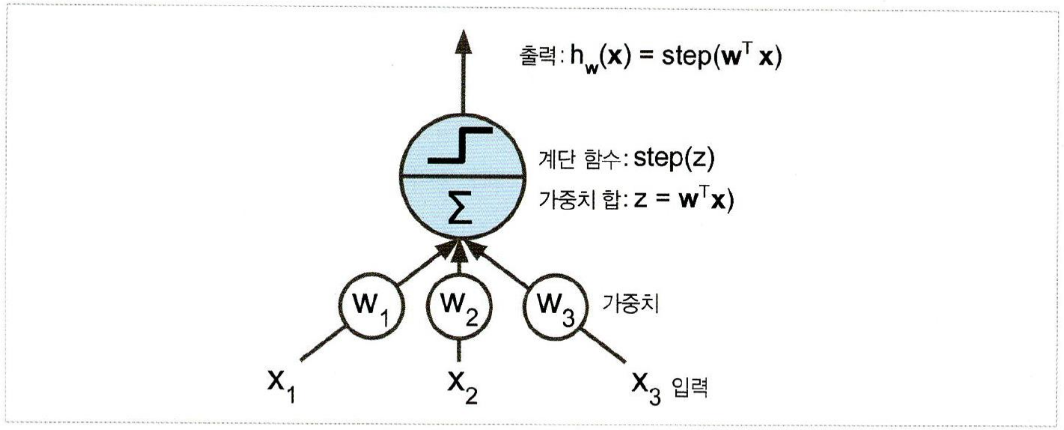
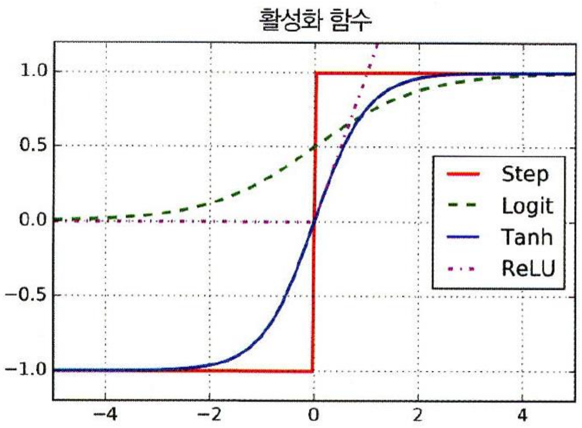


그림 10-4 TLU: 입력의 가중치 합을 계산한 다음 계단 함수를 적용하는 인공 뉴런

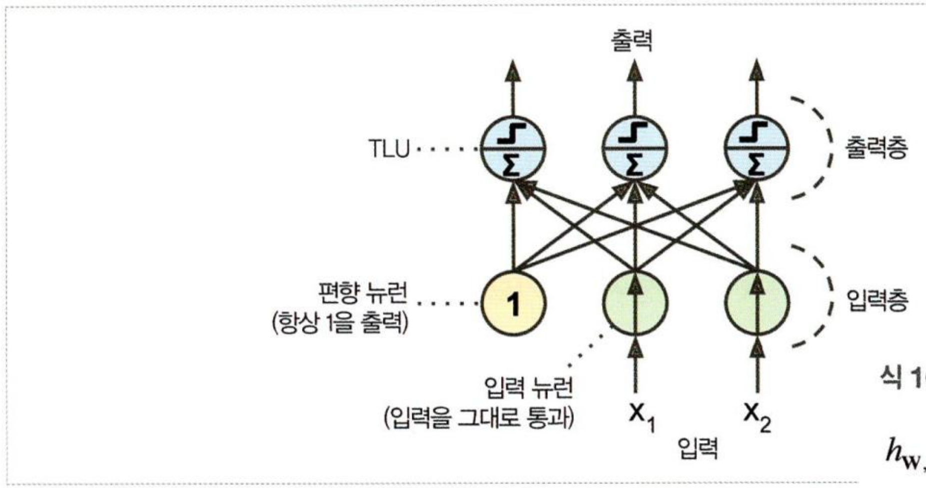


$$z = w_1x_1 + w_2x_2 + \cdots + w_nx_n = \mathbf{w}^T\mathbf{x}$$

식 10-1 퍼셉트론에서 일반적으로 사용하는 계단 함수(임곗값을 0으로 가정)

$$\text{heaviside}(z) = \begin{cases} 0 & z < 0\text{일 때} \\ 1 & z \geq 0\text{일 때} \end{cases} \qquad \text{sgn}(z) = \begin{cases} -1 & z < 0\text{일 때} \\ 0 & z = 0\text{일 때} \\ +1 & z > 0\text{일 때} \end{cases}$$

- 각 TLU은 모든 입력에 연결
- 한 층에 있는 모든 뉴런이 이전 층의 모든 뉴런과 연결 >> 완전 연결 층 또는 밀집 층



식 10-2 완전 연결 층의 출력 계산

$$h_{\mathbf{W},\mathbf{b}}(\mathbf{X}) = \phi(\mathbf{X}\mathbf{W} + \mathbf{b})$$

그림 10-5 입력 뉴런 두 개, 편향 뉴런 한 개, 출력 뉴런 세 개로 구성된 퍼셉트론의 구조

KRICT 한국화학연구원

퍼셉트론Perceptron

식 10-3 퍼셉트론 학습 규칙(가중치 업데이트)

$$w_{i,j}^{(\text{next step})} = w_{i,j} + \eta\left(y_j - \hat{y}_j\right)x_i$$

- $w_{i,j}$는 i번째 입력 뉴런과 j번째 출력 뉴런 사이를 연결하는 가중치입니다.
- x_i는 현재 훈련 샘플의 i번째 뉴런의 입력값입니다.
- $\hat{y}_j$는 현재 훈련 샘플의 j번째 출력 뉴런의 출력값입니다.
- y_j는 현재 훈련 샘플의 j번째 출력 뉴런의 타깃값입니다.
- η는 학습률입니다.

식 10-2 완전 연결 층의 출력 계산

$$h_{\mathbf{W},\mathbf{b}}(\mathbf{X}) = \phi(\mathbf{X}\mathbf{W} + \mathbf{b})$$

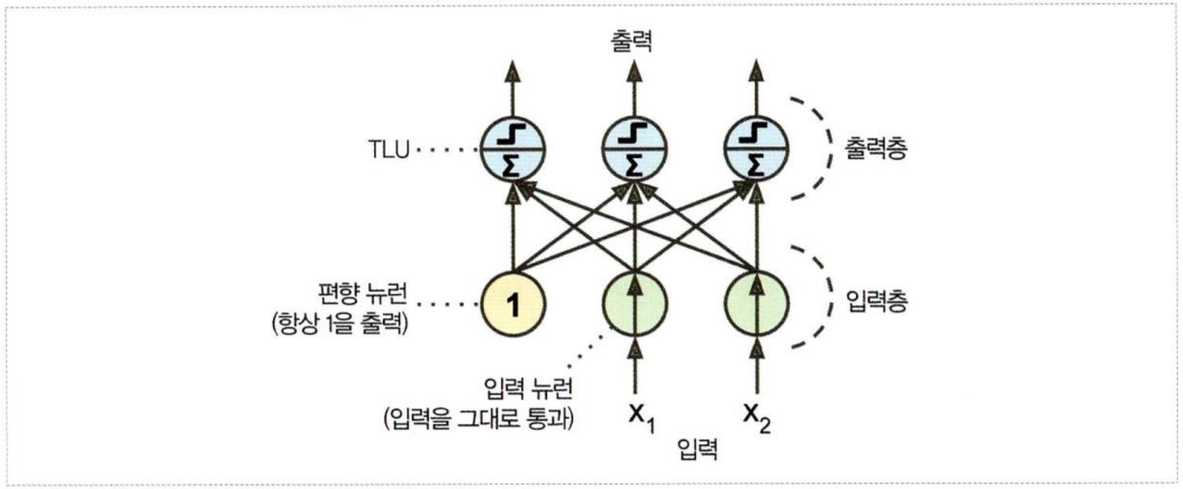


그림 10-5 입력 뉴런 두 개, 편향 뉴런 한 개, 출력 뉴런 세 개로 구성된 퍼셉트론의 구조

인공 뉴런 (on/off > 1/0)

- 매컬러와 피츠 착안한 신경망 모델 > 인공 뉴런
- 논리 명제 계산: **AND, OR, NAND, XOR**

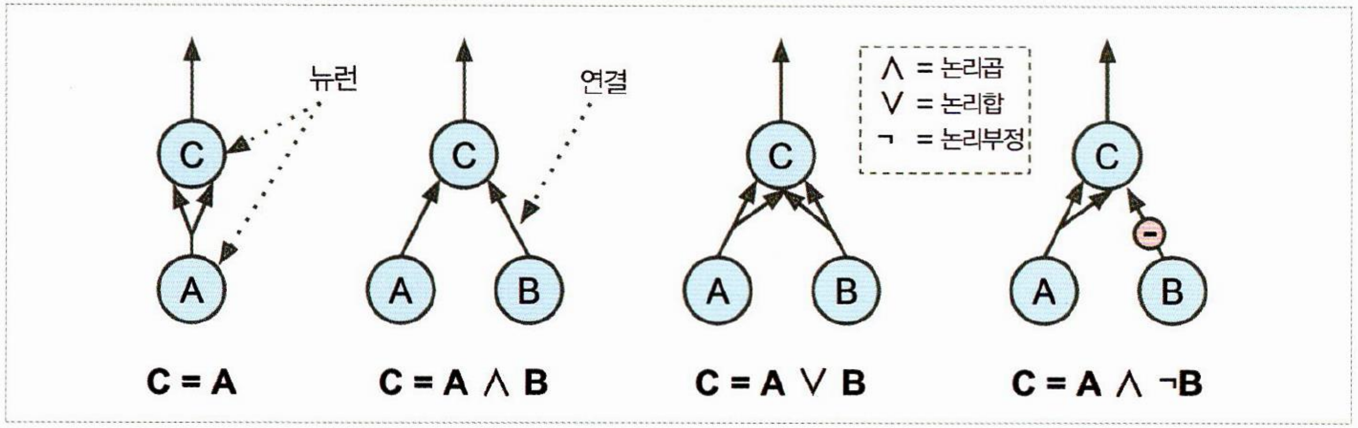


그림 10-3 간단한 논리 연산을 수행하는 인공 신경망

$$\begin{cases} A?B = 0 \rightarrow C = 0 \\ A?B \neq 0 \rightarrow C = 1 \end{cases}$$

$on:1, \quad off:0$

$\wedge : *$

$\vee : +$

$\neg : 0 \leftrightarrow 1$

AND

A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

둘 다 on

OR

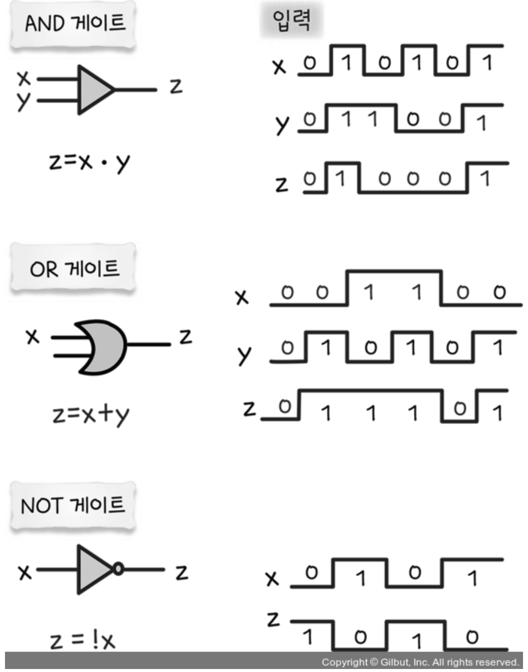
A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

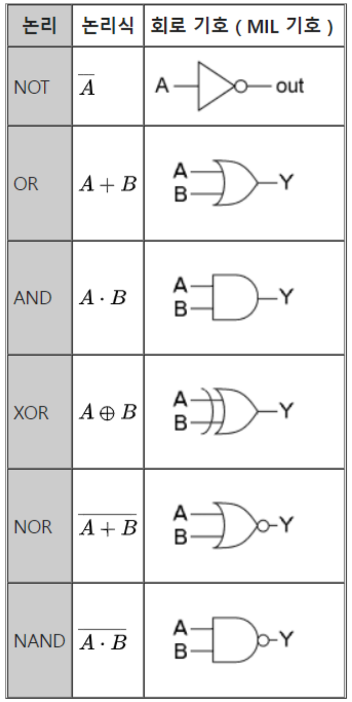
하나 이상

?

A	B	┐B	C
0	0	**1**	**0**
0	1	**0**	**0**
1	0	**1**	**1**
1	1	**0**	**0**

A-on
B-off





XOR

A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

서로 다를 때

NAND

A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

하나 이상

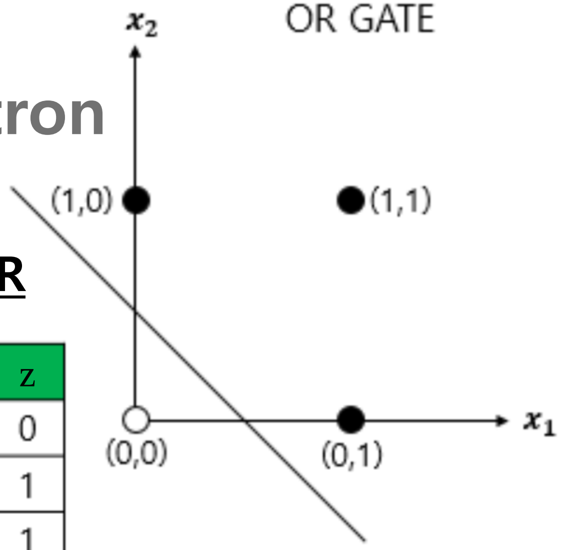
$$C = \neg A \vee \neg B$$

A	B	┐A	┐B	C
0	0	**1**	**1**	**1**
0	1	**1**	**0**	**1**
1	0	**0**	**1**	**1**
1	1	**0**	**0**	**0**

KRICT 한국화학연구원

퍼셉트론 Perceptron

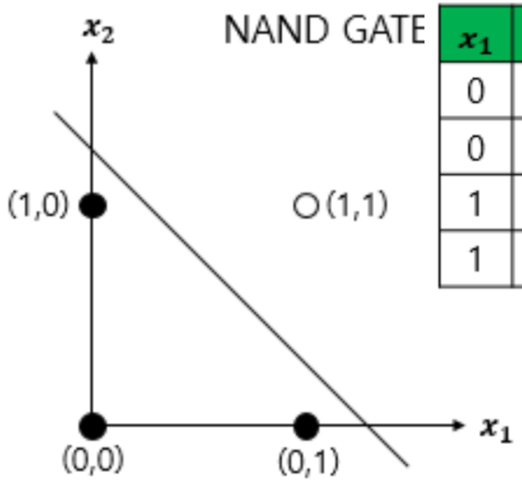
OR



x_1	x_2	z
0	0	0
0	1	1
1	0	1
1	1	1

```python
def OR_gate(x1, x2):
    w1 = 0.6
    w2 = 0.6
    b = -0.5
    result = x1*w1 + x2*w2 + b
    if result <= 0:
        return 0
    else:
        return 1
```

NAND



x_1	x_2	z
0	0	1
0	1	1
1	0	1
1	1	0

```python
def NAND_gate(x1, x2):
    w1 = -0.5
    w2 = -0.5
    b = 0.7
    result = x1*w1 + x2*w2 + b
    if result <= 0:
        return 0
    else:
        return 1
```

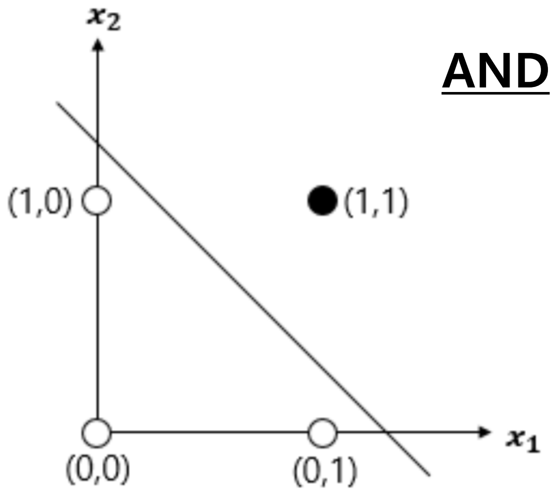
$$z = w_1x_1 + w_2x_2 + \cdots + w_nx_n = \mathbf{w}^T\mathbf{x}$$

식 10-1 퍼셉트론에서 일반적으로 사용하는 계단 함수(임곗값을 0으로 가정)

$$\text{heaviside}(z) = \begin{cases} 0 & z < 0\text{일 때} \\ 1 & z \ge 0\text{일 때} \end{cases} \qquad \text{sgn}(z) = \begin{cases} -1 & z < 0\text{일 때} \\ 0 & z = 0\text{일 때} \\ +1 & z > 0\text{일 때} \end{cases}$$

```python
def AND_gate(x1, x2):
    w1 = 0.5
    w2 = 0.5
    b = -0.7
    result = x1*w1 + x2*w2 + b
    if result <= 0:
        return 0
    else:
        return 1
```

AND

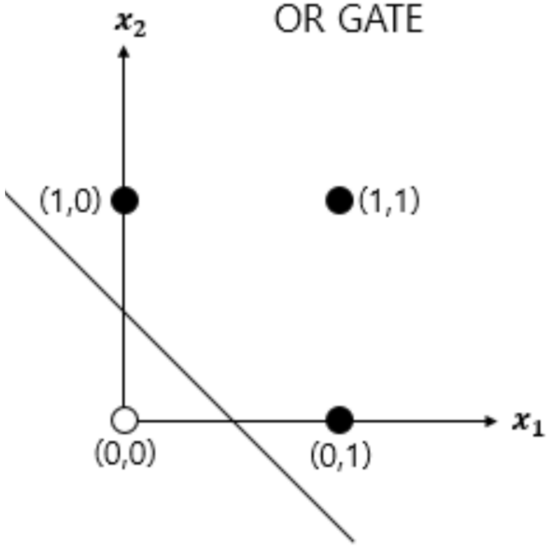


x_1	x_2	z
0	0	0
0	1	0
1	0	0
1	1	1

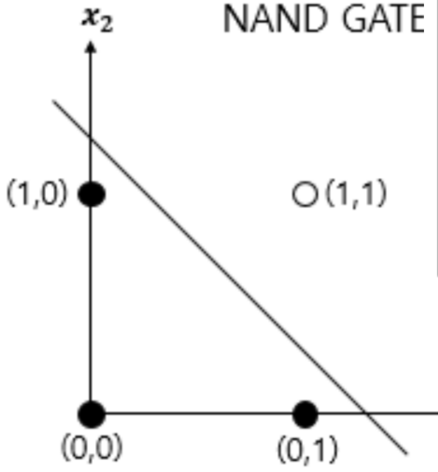
퍼셉트론Perceptron

x_1	x_2	z
0	0	0
0	1	1
1	0	1
1	1	1

OR



NAND



x_1	x_2	z
0	0	1
0	1	1
1	0	1
1	1	0

XOR

x_1	x_2	z
0	0	0
0	1	1
1	0	1
1	1	0

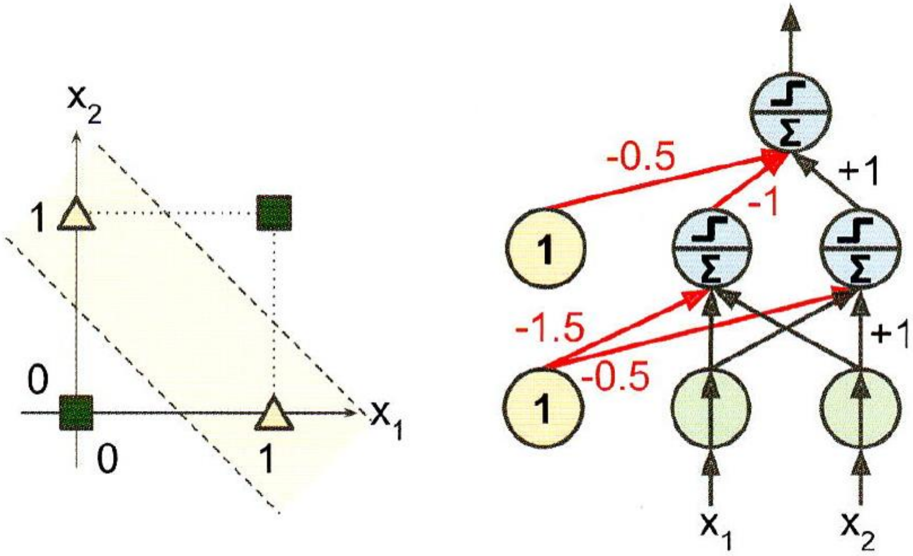
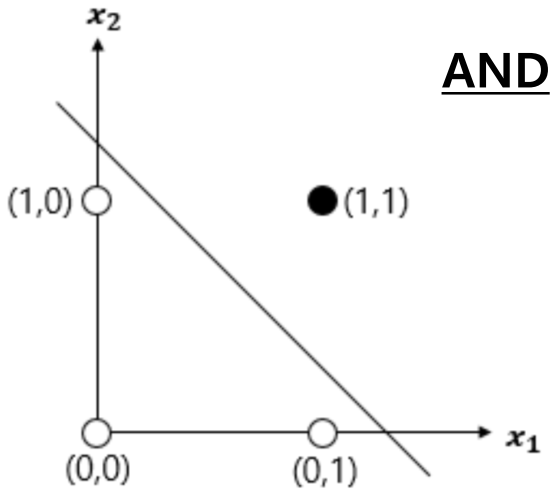


그림 10-6 XOR 분류 문제와 이를 푸는 다층 퍼셉트론



AND

x_1	x_2	z
0	0	0
0	1	0
1	0	0
1	1	1

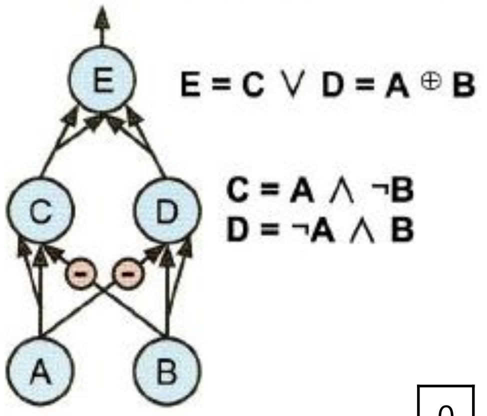
2. ([그림 10-3]에 있는 것과 같은) 초창기 인공 뉴런을 사용해 A⊕B (⊕는 XOR 연산입니다)를 계산하는 인공 신경망을 그려보세요. 힌트: $A \oplus B = (A \wedge \neg B) \vee (\neg A \wedge B)$

2. $A \oplus B = (A \wedge \neg B) \vee (\neg A \wedge B)$일 때 $A \oplus B$를 계산하는 초창기 인공 뉴런 기반의 신경망은 아래 그림과 같습니다. 다른 방법도 있습니다. 예를 들어 $A \oplus B = (A \vee B) \wedge \neg(A \wedge B)$ 또는 $A \oplus B = (A \vee B) \wedge (\neg A \vee \wedge B)$ 등을 사용합니다.[5]

[5] 옮긴이_ 문제에서 주어진 식을 논리 연산의 분배 법칙과 드모르간의 법칙을 적용하여 바꾼 것입니다. 논리합∨, 논리곱∧, 부정¬의 논리 기호를 사용하여 표시하면, 아래와 같다.

$\neg(P \vee Q) = \neg P \wedge \neg Q$

$\neg(P \wedge Q) = \neg P \vee \neg Q$



∧ = AND ¬ = NOT
∨ = OR ⊕ = XOR

$$\begin{bmatrix}0\\1\\1\\1\end{bmatrix} \quad \begin{bmatrix}1\\1\\1\\0\end{bmatrix}$$

$$\begin{bmatrix}0\\1\\1\\1\end{bmatrix} \quad \begin{bmatrix}1\\1\\1\\0\end{bmatrix}$$

$$\begin{cases} A?B = 0 \rightarrow C = 0 \\ A?B \neq 0 \rightarrow C = 1 \end{cases}$$

$on: 1, \quad off: 0$

$\wedge : *$

$\vee : +$

$\neg : 0 \leftrightarrow 1$

A	B	┐A	┐B	C	D	E
0	0	1	1	**0**	**0**	**<u>0</u>**
0	1	1	0	**0**	**1**	**<u>1</u>**
1	0	0	1	**1**	**0**	**<u>1</u>**
1	1	0	0	**0**	**0**	**<u>0</u>**

KRICT 한국화학연구원