기계학습응용

-인공신경망 리뷰-

미디어기술콘텐츠학과
강호철

가톨릭대학교
THE CATHOLIC UNIVERSITY OF KOREA

배경

- 뇌의 정보처리 방법 모방
  - 지능 컴퓨터에 도전
  - 인공 신경망(ANN, Artificial Neural Network)
    - 1940대 개발 (디지털 컴퓨터와 탄생 시기가 유사함)
    - 인간 지능에 필적하는 컴퓨터 개발이 목표
- 컴퓨터와 두뇌의 비교
  - 폰 노이만 컴퓨터
    - 순차 명령어 처리기
  - 두뇌
    - 뉴런으로 구성
    - 고도의 병렬 명령어 처리기

(a) 폰 노이만 컴퓨터의 구조 (b) 사람 뇌의 정보 처리 단위인 뉴런

그림 4.1 컴퓨터와 사람

배경

- 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

수학적 모델의 신경망

- 신경망 특성
  - 학습 가능
  - 뛰어난 일반화 능력
  - 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)
  - 딥러닝으로 발전

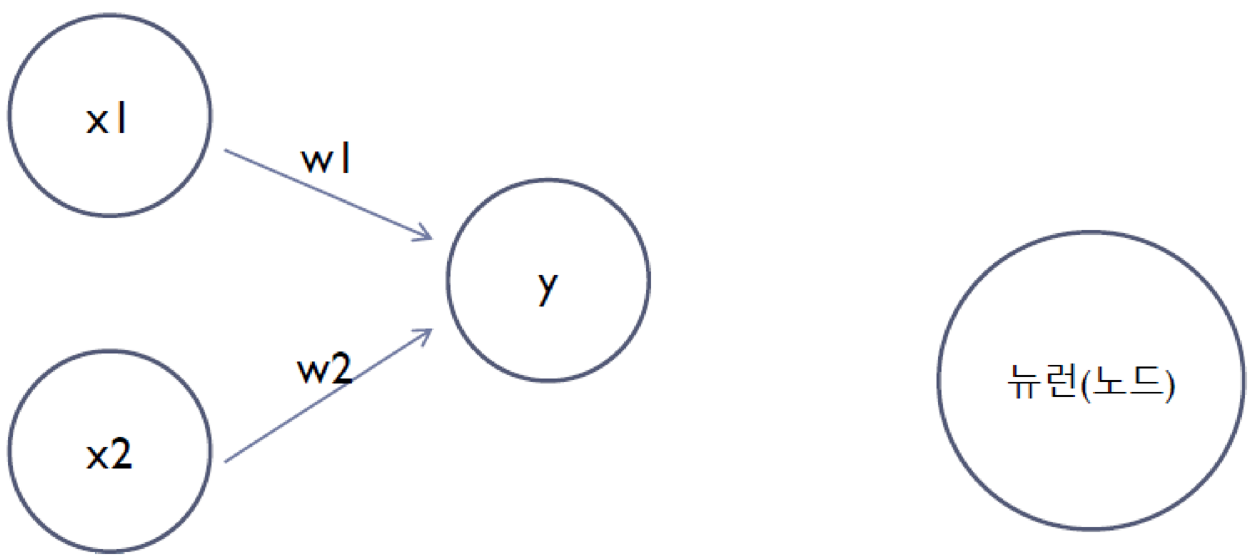
수학적 모델의 신경망

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

- MLP의 초석이 됨

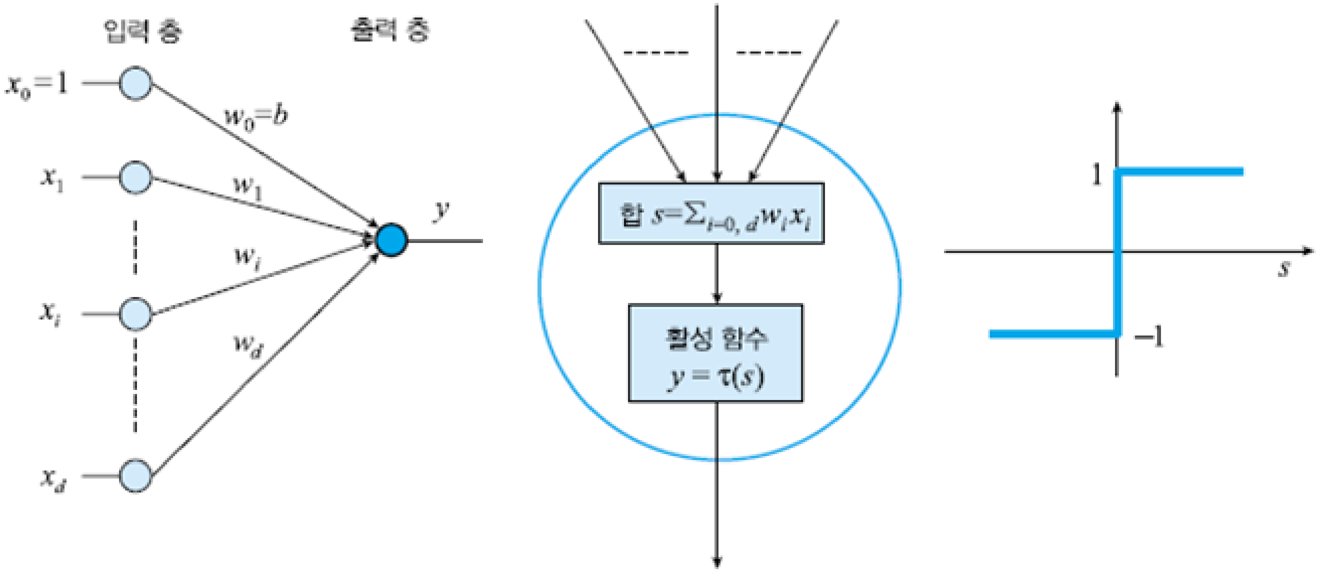
퍼셉트론

- 개념
  - 다수의 신호를 입력으로 받아 하나 혹은 여러 신호 출력
    - 신호는 흐름을 가짐
    - 신호의 총합이 임계치를 넘으면 1, 아니면 0 출력
    - 가중치는 해당 신호의 중요한 정도



퍼셉트론

- 구조
  - 입력층: d+1개의 노드 (특징 벡터 $\mathbf{x}=(x_1,\ldots,x_d)^T$)
  - 출력층: 한 개의 노드 (따라서 2-부류 분류기)
  - 에지와 가중치



(a) 전체 구조 (b) 출력 노드의 연산 (c) 활성 함수

그림 4.2 퍼셉트론의 구조

퍼셉트론

- 예제

$\mathbf{a}=(0,0)^{\mathrm{T}},\ t_a=-1$

$\mathbf{b}=(1,0)^{\mathrm{T}},\ t_b=1$

$\mathbf{c}=(0,1)^{\mathrm{T}},\ t_c=1$

$\mathbf{d}=(1,1)^{\mathrm{T}},\ t_d=1$

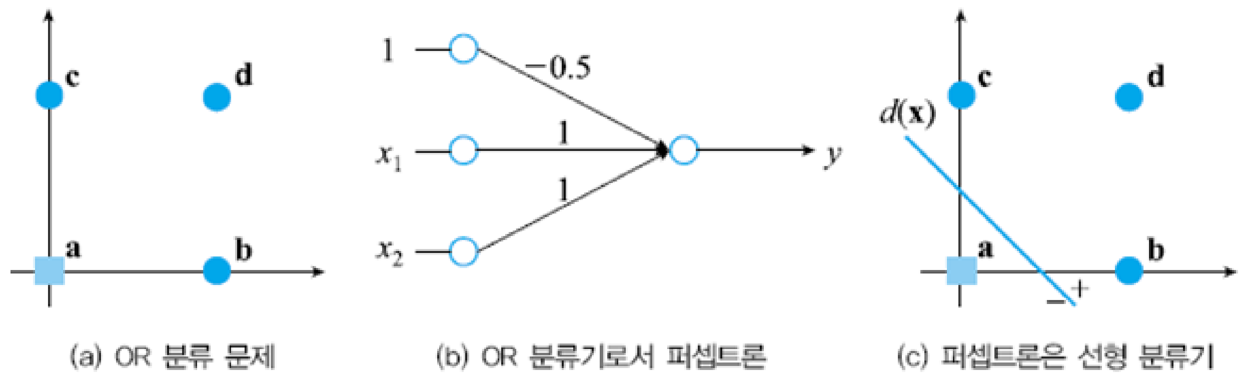


그림 4.3 퍼셉트론의 예

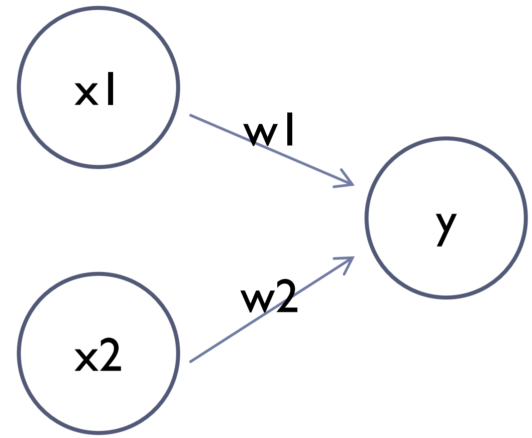
이 퍼셉트론은 $\mathbf{w}=(1,1)^{\mathrm{T}}$, b=-0.5

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

- 샘플 **c** 를 인식해 보자. 맞추나? $y=\tau(\mathbf{w}^{\mathrm{T}}\mathbf{c}+b)=\tau((1,1)\begin{pmatrix}0\\1\end{pmatrix}-0.5)=\tau(0.5)=1$
- 나머지 **a, b, d** 는?

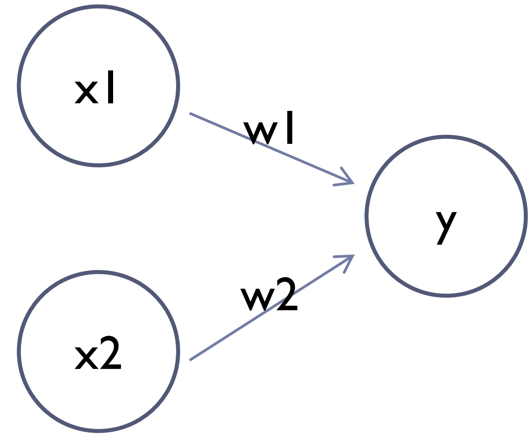
단순 논리 회로

- AND 게이트
  - y가 참인경우?
  - 거짓인 경우?
  - AND를 만족하는 매개변수 쌍
    - (w1=0.5, w2=0.5, theta=0.7)
    - (0.5, 0.5, 0.8)
    - (1.0, 1.0, 1.0)
    - ....



단순 논리 회로

- NAND 게이트
  - y가 참/거짓인경우?
  - NAND를 만족하는 매개변수 쌍
    - (w1=-0.5, w2=-0.5, theta=-0.7)
    - ....



- OR 게이트
  - y가 참/거짓인경우?
  - OR를 만족하는 매개변수 쌍
    - (w1=1.0, w2=1.0, theta=0.5)
    - ....

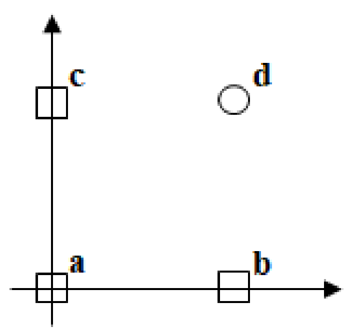
퍼셉트론

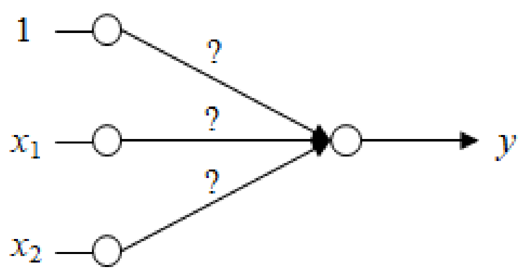
- 퍼셉트론 학습
  - 퍼셉트론 학습이란? 훈련 집합 $X = \{(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), \cdots, (\mathbf{x}_N, t_N)\}$이 주어졌을 때 이들을 모두 옳게 분류하는 퍼셉트론 (즉 $\mathbf{w}$와 b)을 찾아라. 샘플 $(\mathbf{x}_i, t_i)$에서 $\mathbf{x}_i$는 특징 벡터이고 t_i는 부류 표지로서 $\mathbf{x}_i \in \omega_1$이면 $t_i = 1$이고 $\mathbf{x}_i \in \omega_2$이면 $t_i = -1$이다. X는 선형 분리 가능하다고 가정한다.[3]

- 예) AND 분류 문제

$\mathbf{a}=(0,0)^T$ $\mathbf{b}=(1,0)^T$ $\mathbf{c}=(0,1)^T$ $\mathbf{d}=(1,1)^T$

$t_a= -1$ $t_b= -1$ $t_c= -1$ $t_d=1$





퍼셉트론

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

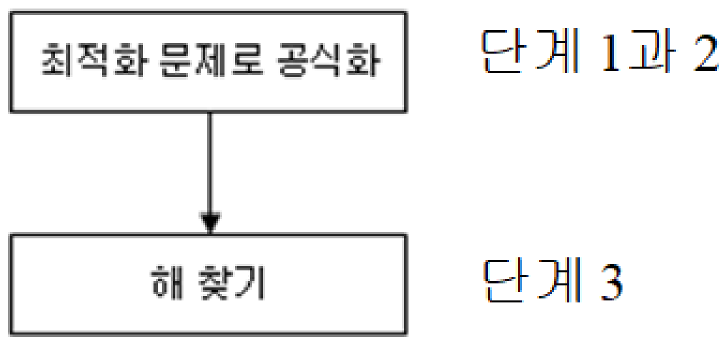
$$\left.\begin{aligned} &y = \tau(s) = \tau\left(\sum_{i=1}^{d} w_i x_i + b\right) = \tau(\mathbf{w}^{\mathrm{T}}\mathbf{x} + b) \\ &\text{이때 } \tau(s) = \begin{cases} +1, & s \ge 0 \\ -1, & s < 0 \end{cases} \end{aligned}\right\} \quad (4.2)$$

- 퍼셉트론은 선형 분류기

$$\left.\begin{aligned} d(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b > 0 \text{ 이면} \quad \mathbf{x} \in \omega_1 \\ d(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b < 0 \text{ 이면} \quad \mathbf{x} \in \omega_2 \end{aligned}\right\} \quad (4.3)$$

퍼셉트론 학습

- (Review) 일반적인 학습 알고리즘 설계 과정
  - 단계 1: 분류기 구조 정의와 분류 과정의 수학식 정의
  - 단계 2: 분류기 품질 측정용 비용함수 $\mathcal{J}(\Theta)$ 정의
  - 단계 3: $\mathcal{J}(\Theta)$를 최적화하는 Θ를 찾는 알고리즘 설계



퍼셉트론 학습

- 단계 1
  - 식 (4.2)
  - 매개변수 집합 $\Theta=\{\mathbf{w}, b\}$

- 단계 2
  - 분류기 품질을 측정하는 $J(\Theta)$를 어떻게 정의할 것인가?

$$J(\Theta) = \sum_{\mathbf{x}_k \in Y} (-t_k)(\mathbf{w}^{\mathrm{T}}\mathbf{x}_k + b) \tag{4.4}$$

  - Y: 오분류된 샘플 집합
  - $J(\Theta)$는 항상 양수
  - Y가 공집합이면 $J(\Theta)=0$
  - $|Y|$가 클수록 $J(\Theta)$ 큼

퍼셉트론 학습

- 단계 3
  - $\mathcal{J}(\Theta)=0$인 Θ를 찾아라.
  - 내리막 경사법 (Gradient descent method)
    - 현재 해를 $-\partial/\partial\Theta$ 방향으로 이동
    - 학습률 ρ를 곱하여 조금씩 이동

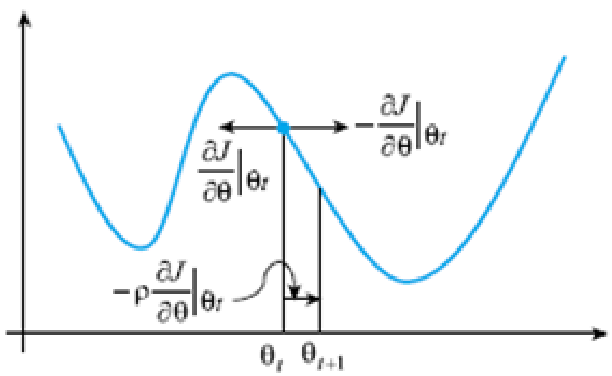
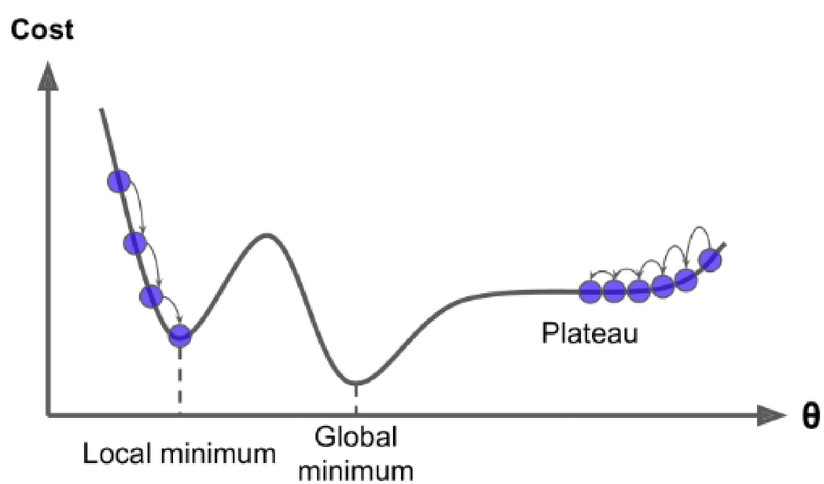


그림 11.9 내리막 경사법

경사하강법 (Gradient Descent)

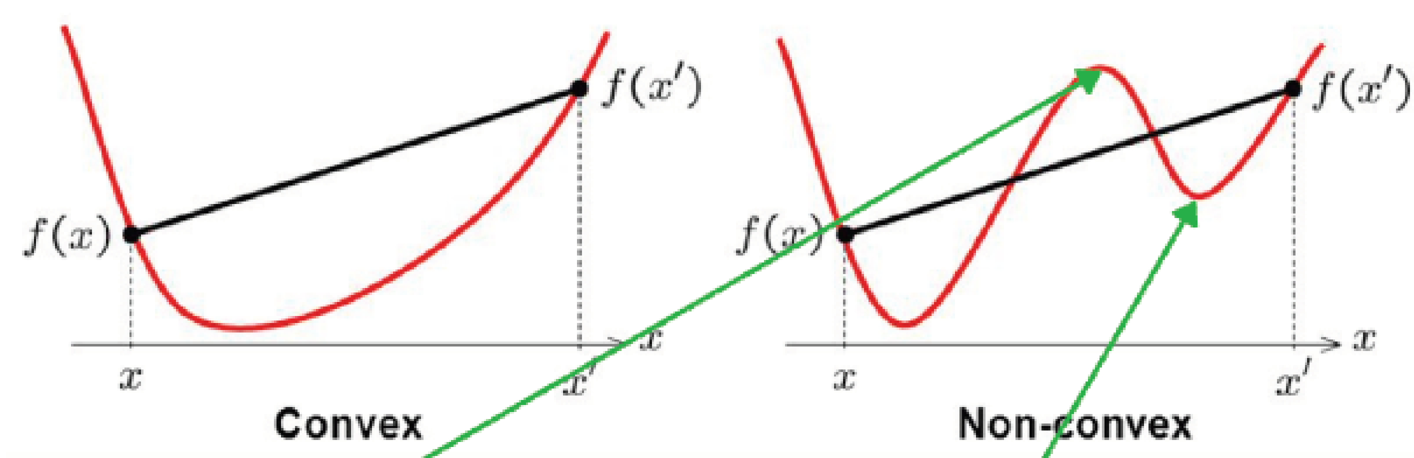
- 경사도(기울기)
  - 모든 변수의 편미분을 벡터로 정리한 것
  - 경사도가 가리키는 쪽은 각 장소에서 함수의 출력 값을 가장 줄이는 방향임

- 경사하강법
  - 경사도를 이용하여 손실함수의 최소값을 찾는 방법



경사하강법 (Gradient Descent)

- Convex Function



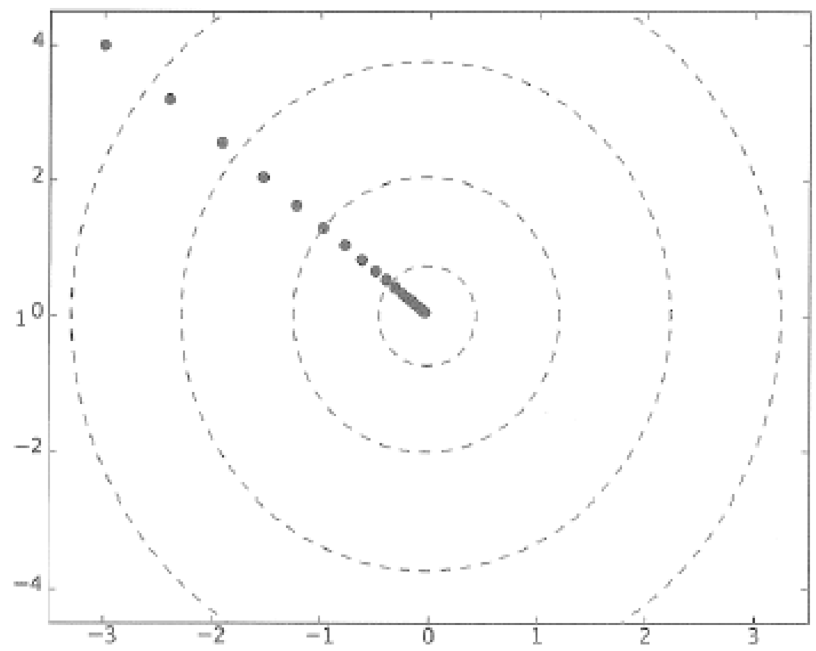
- 미분값이 0이라고 무조건 최솟값은 아닙니다.
- 최댓값일 수도 있습니다.
- 최솟값은 아니지만 미분 값이 0인 곳을 Local Minimum이라고 합니다.

경사하강법 (Gradient Descent)

- 개념
  - 현 위치에서 기울어진 방향으로 일정 거리만큼 이동
  - 이동한 거리에서 다시 경사도 구하고 이동
  - 위의 과정을 반복

$$x_0 = x_0 - \eta \frac{\partial f}{\partial x_0}$$

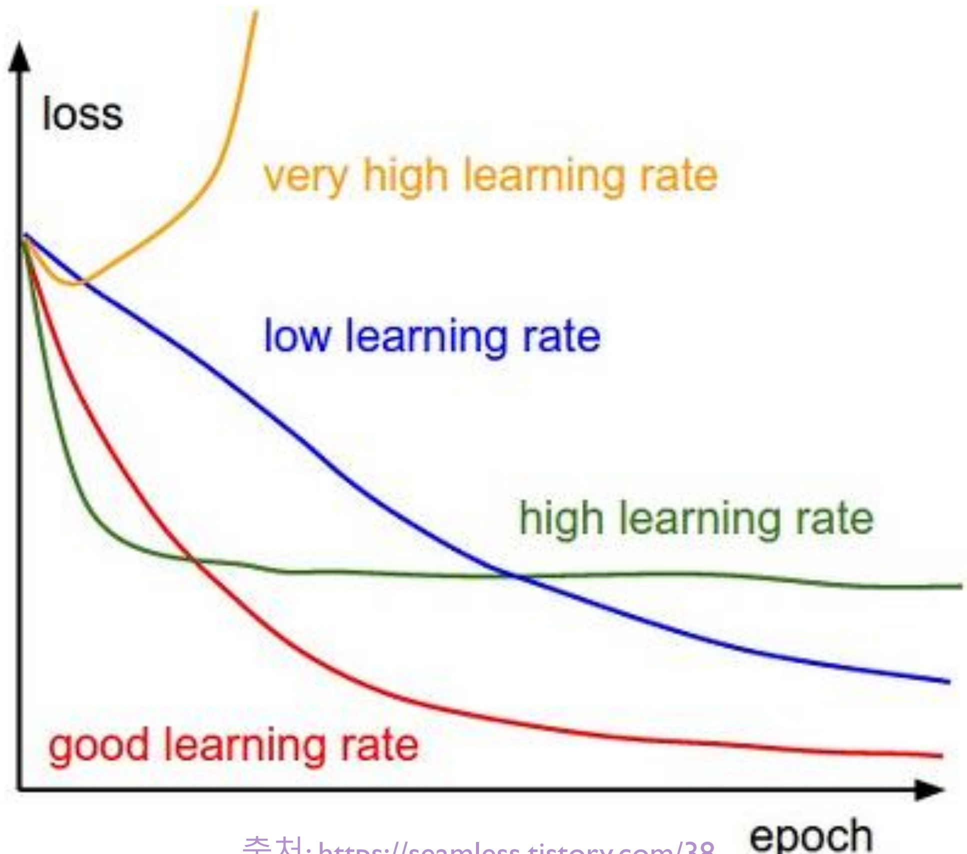
$$x_1 = x1 - \eta \frac{\partial f}{\partial x_1}$$



  - 에타: 학습률(learning rate)
  - hyper-parameter

경사하강법 (Gradient Descent)

- 개념
  - 학습률(learning rate)



출처: https://seamless.tistory.com/38

퍼셉트론 학습

- 전체 알고리즘
  - 초기해 설정
  - 멈춤조건이 만족될때까지 현재 해를 조금씩 이동

$$\Theta(h+1) = \Theta(h) - \rho(h)\frac{\partial J(\Theta)}{\partial \Theta} \tag{4.5}$$

$$\left.\begin{aligned} \frac{\partial J(\Theta)}{\partial \mathbf{w}} &= \sum_{\mathbf{x}_k \in Y} (-t_k)\mathbf{x}_k \\ \frac{\partial J(\Theta)}{\partial b} &= \sum_{\mathbf{x}_k \in Y} (-t_k) \end{aligned}\right\} \tag{4.6}$$

$$\left.\begin{aligned} \mathbf{w}(h+1) &= \mathbf{w}(h) + \rho(h)\sum_{\mathbf{x}_k \in Y} t_k \mathbf{x}_k \\ b(h+1) &= b(h) + \rho(h)\sum_{\mathbf{x}_k \in Y} t_k \end{aligned}\right\} \tag{4.7}$$

퍼셉트론

알고리즘 [4.1] 퍼셉트론 학습 (배치 모드 batch mode)

입력: 훈련 집합 $X = \{(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), \cdots, (\mathbf{x}_N, t_N)\}$, 학습률 ρ

출력: 퍼셉트론 가중치 $\mathbf{w}$, b

알고리즘:

1. $\mathbf{w}$와 b를 초기화한다.
2. **repeat** {
3. $\quad Y = \varnothing$;
4. $\quad$ **for** ($i = 1$ **to** N) {
5. $\quad\quad y = \tau(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b)$; // (4.2)로 분류를 수행함
6. $\quad\quad$ **if** ($y \neq t_i$) $Y = Y \cup \mathbf{x}_i$; // 오분류된 샘플 수집
7. $\quad$ }
8. $\quad \mathbf{w} = \mathbf{w} + \rho \sum_{\mathbf{x}_k \in Y} t_k \mathbf{x}_k$; // (4.7)로 가중치 갱신
9. $\quad b = b + \rho \sum_{\mathbf{x}_k \in Y} t_k$;
10. } **until** ($Y = \varnothing$);
11. $\mathbf{w}$와 b를 저장한다.

퍼셉트론

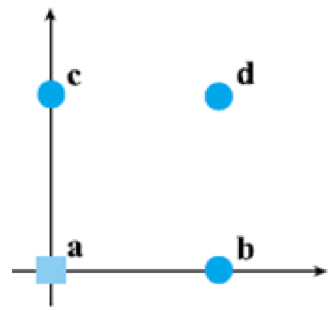
- 예제 - 학습

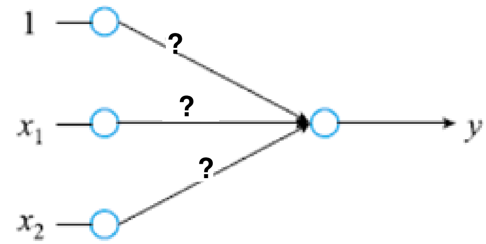
$\mathbf{a} = (0,0)^T,\ t_a = -1$

$\mathbf{b} = (1,0)^T,\ t_b = 1$

$\mathbf{c} = (0,1)^T,\ t_c = 1$

$\mathbf{d} = (1,1)^T,\ t_d = 1$





- ρ = 0.4로 설정
- 초기값은 $w(0) = (-0.5, 0.75)^T$, $b(0) = 0.375$
- $d(x) = -0.5x_1 + 0.75x_2 + 0.375$
- Y(오분류 샘플) = {a, b}
- 가중치 갱신

반복

8. $\mathbf{w} = \mathbf{w} + \rho \sum_{\mathbf{x}_k \in Y} t_k \mathbf{x}_k$;

9. $b = b + \rho \sum_{\mathbf{x}_k \in Y} t_k$;

퍼셉트론

- 예제 - 학습
  - w(1) = ?, b(1) = ?, Y = ?
  - w(2) = ?, b(2) = ?, Y = ?
  - w(3) = ?, b(3) = ?, Y = ?
  - w(4) = ?, b(4) = ?, Y = ?
  - ....
  - 최종 w = ?, b = ?

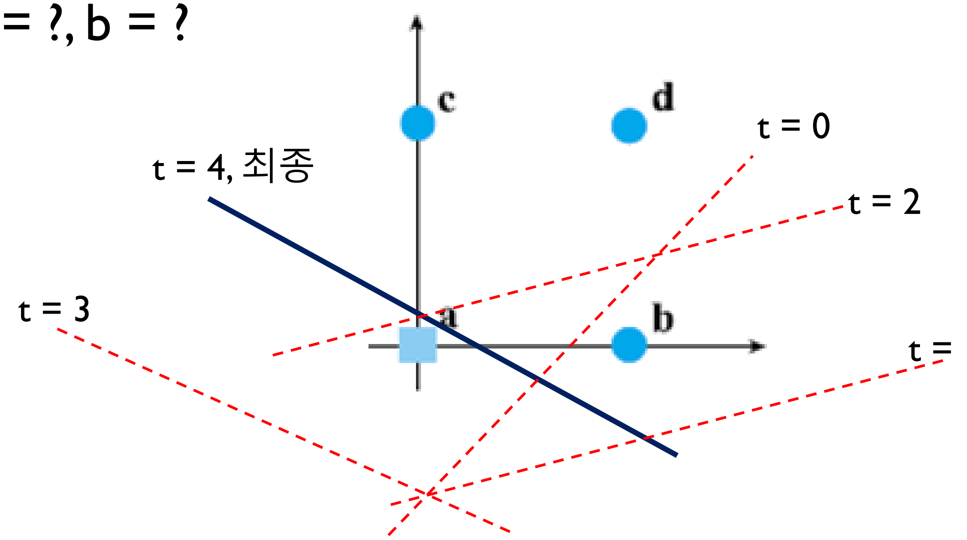
알고리즘 [4.1] 퍼셉트론 학습 (배치 모드 batch mode)

입력: 훈련 집합 $X=\{(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), \cdots, (\mathbf{x}_N, t_N)\}$, 학습률 ρ

출력: 퍼셉트론 가중치 $\mathbf{w}$, b

알고리즘:

```
w와 b를 초기화한다.
repeat {
    Y = ∅;
    for (i = 1 to N) {
        y = τ(wᵀxᵢ+b);            // (4.2)로 분류를 수행함
        if (y ≠ tᵢ) Y = Y ∪ xᵢ;   // 오분류된 샘플 수집
    }
    w = w + ρ Σ_{x_k∈Y} t_k x_k ;  // (4.7)로 가중치 갱신
    b = b + ρ Σ_{x_k∈Y} t_k ;
} until (Y = ∅);
w와 b를 저장한다.
```



다중 퍼셉트론

- 선형 분리 불가능한 상황
  - 퍼셉트론의 한계

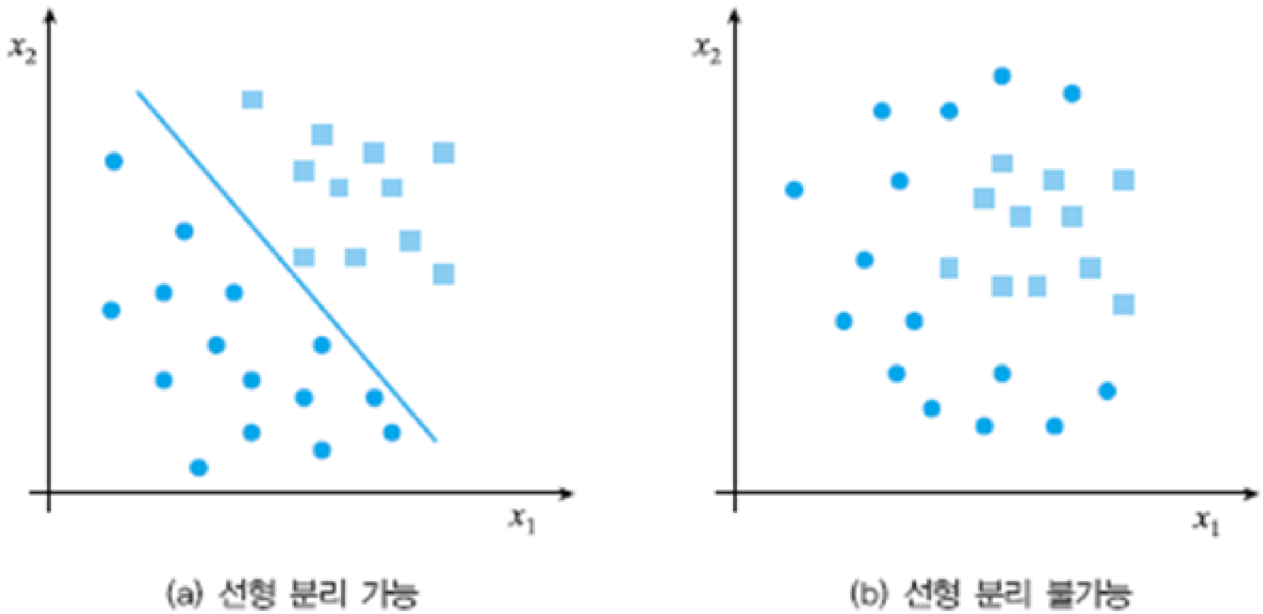


그림 4.5 선형 분리 가능과 불가능

다중 퍼셉트론

- XOR 문제
  - 퍼셉트론은 75% 정인식률이 한계
  - 이 한계를 어떻게 극복?
    - 두 개의 퍼셉트론 (결정 직선) 사용

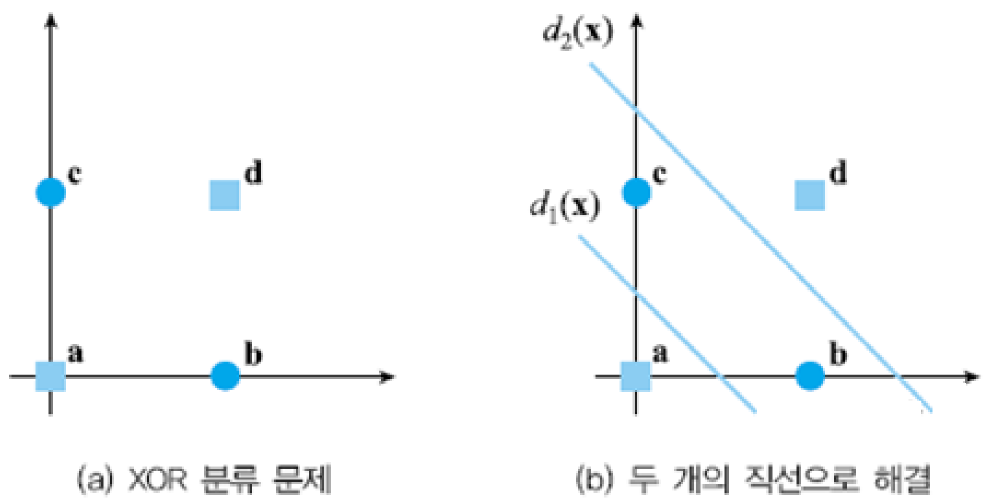


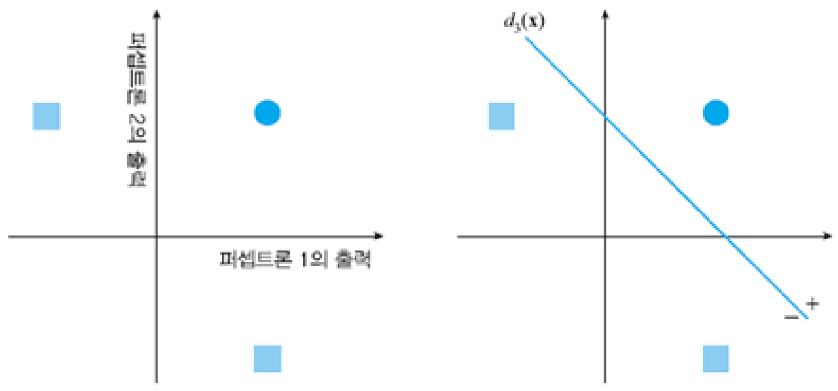
그림 4.6 XOR 분류 문제의 해결

다중 퍼셉트론

- 두 단계에 걸쳐 문제 해결
  - 단계 1: 원래 특징 공간을 새로운 공간으로 매핑
  - 단계 2: 새로운 공간에서 분류

표 4.1 두 단계로 XOR 문제 해결

샘플	특징 벡터 (x)		첫 번째 단계		두 번째 단계
	x_1	x_2	퍼셉트론1	퍼셉트론2	퍼셉트론3
a	0	0	−1	+1	−1
b	1	0	+1	+1	+1
c	0	1	+1	+1	+1
d	1	1	+1	−1	−1

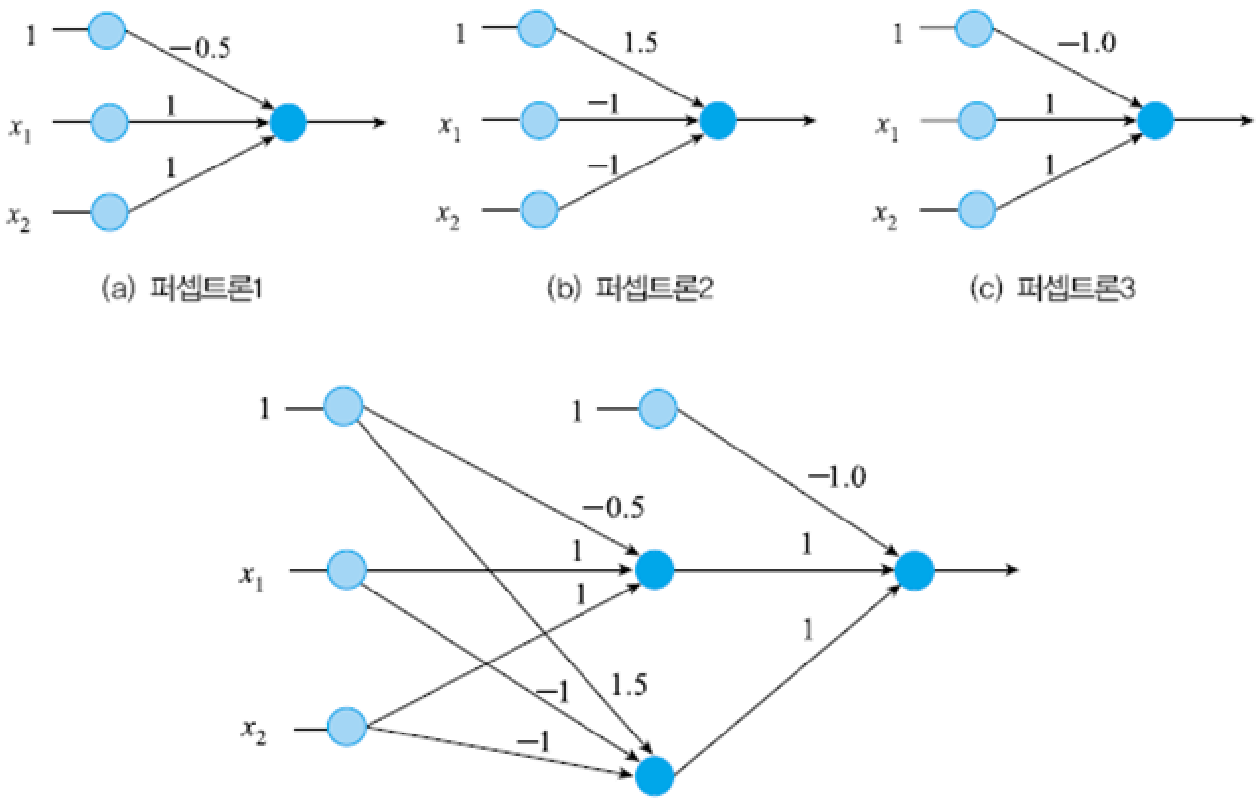


(a) 매핑 공간에서의 샘플 분포 (b) 퍼셉트론3에 의한 영역 분할

그림 4.7 새로운 공간에서의 샘플 분포와 영역 분할

다중 퍼셉트론

- 다층 퍼셉트론 (MLP; Multi-layer perceptron)



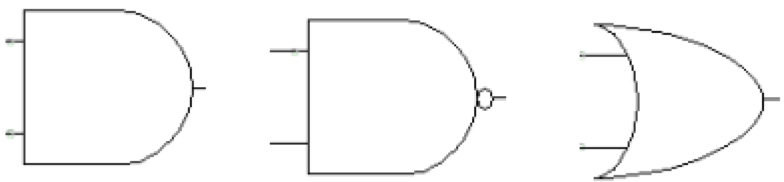
(a) 퍼셉트론1 (b) 퍼셉트론2 (c) 퍼셉트론3

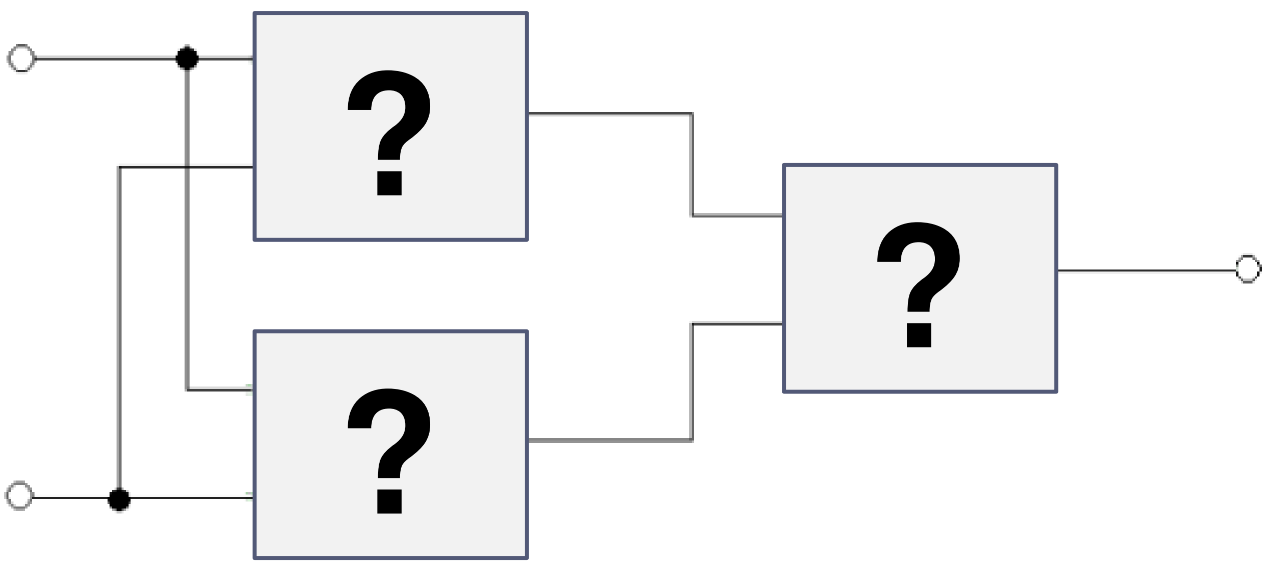
(d) 다층 퍼셉트론

그림 4.8 세 개의 퍼셉트론과 이들을 연결하여 만든 다층 퍼셉트론

다중 퍼셉트론

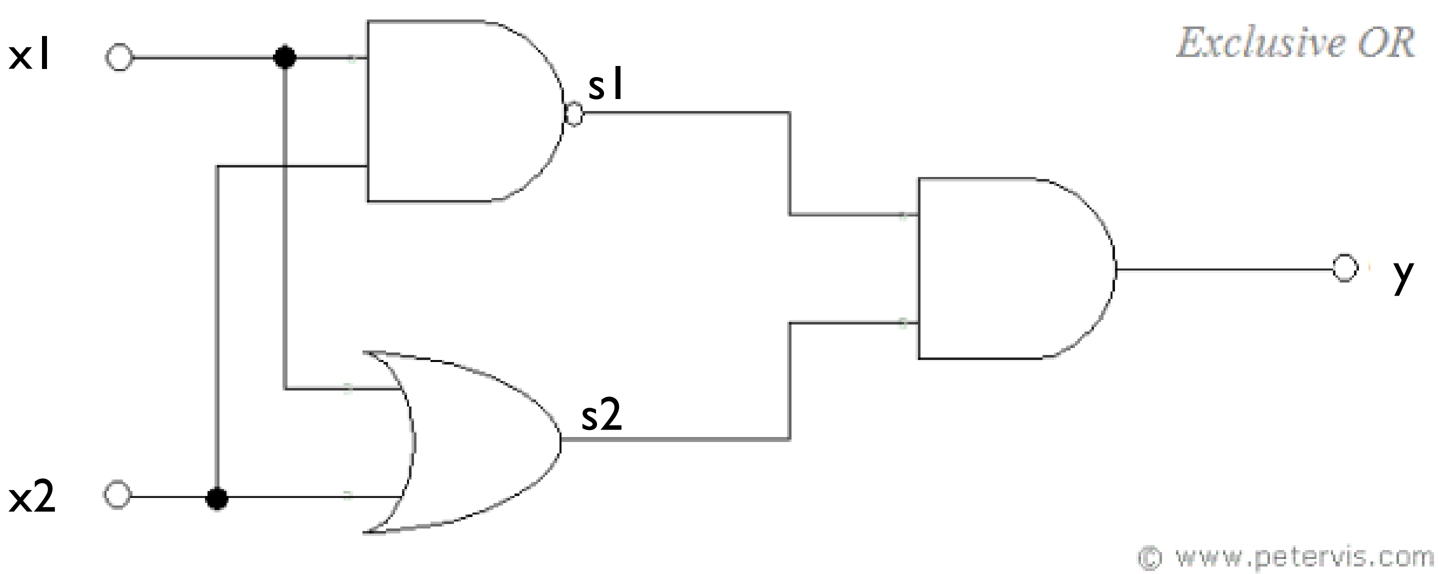
- XOR
  - AND, NAND, OR 이용





다중 퍼셉트론

- XOR
  - AND, NAND, OR 이용



다중 퍼셉트론

- 다층 퍼셉트론의 아키텍처
  - 입력층, 은닉층, 출력층
  - 가중치: **u**와 **v**

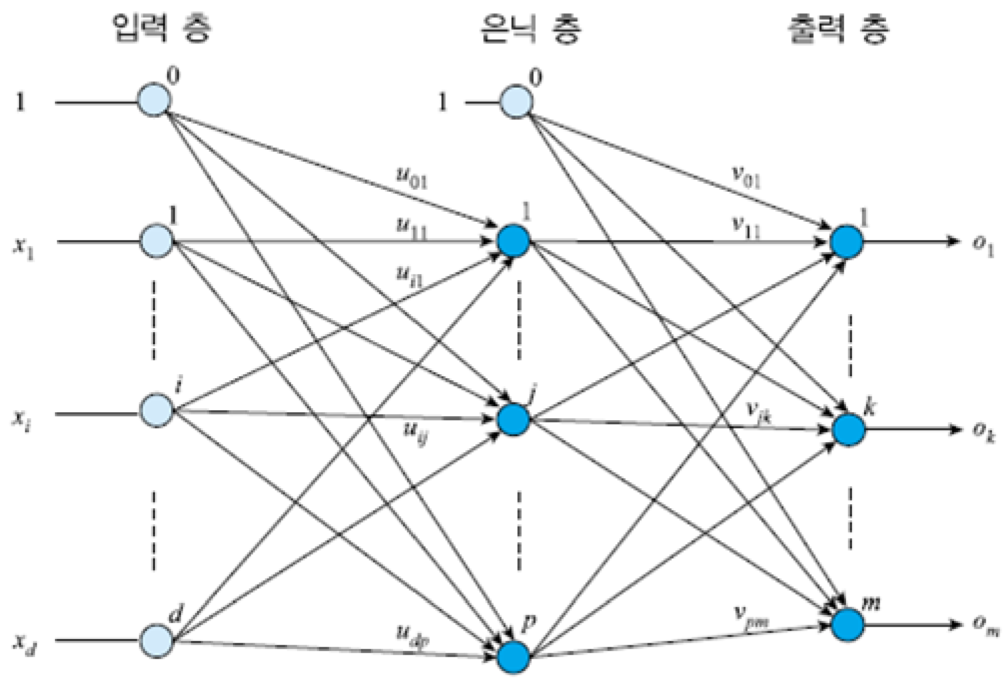


그림 4.9 다층 퍼셉트론의 구조와 표기

- 신경망은 일종의 함수

$$\mathbf{o} = f(\mathbf{x})$$

$$\left.\begin{array}{l}\mathbf{z} = p(\mathbf{x}) \\ \mathbf{o} = q(\mathbf{z}) \\ \\ \text{또는} \\ \\ \mathbf{o} = q(p(\mathbf{x}))\end{array}\right\}$$

다중 퍼셉트론

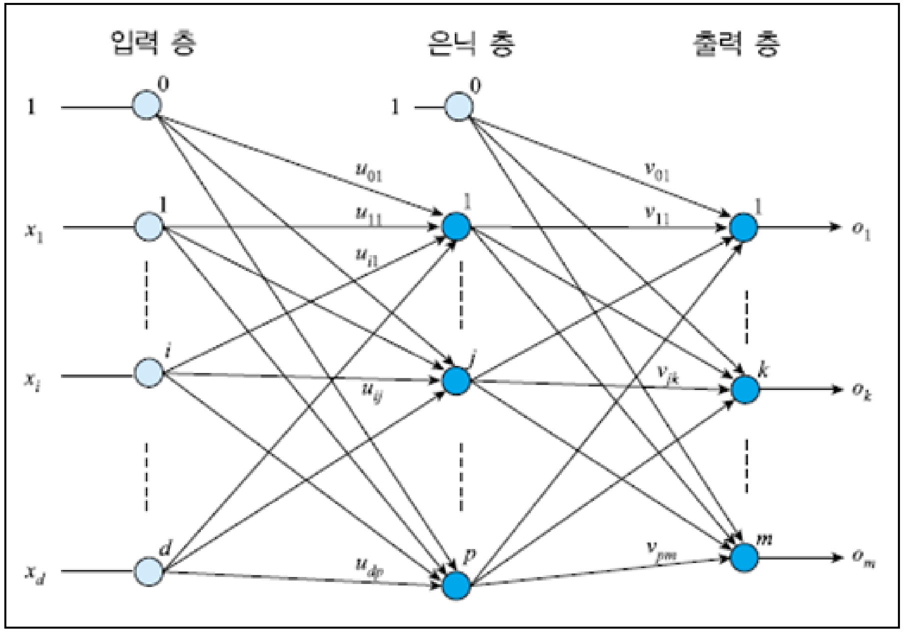
- 전방 계산 (forward computation)

은닉 층의 j번째 노드, $1 \le j \le p$:

$$\left.\begin{aligned} z\_sum_j &= \sum_{i=1}^{n} x_i u_{ij} + u_{0j} \\ z_j &= \tau(z\_sum_j) \end{aligned}\right\} \qquad (4.12)$$

출력 층의 k번째 노드, $1 \le k \le m$:

$$\left.\begin{aligned} o\_sum_k &= \sum_{j=1}^{p} z_j v_{jk} + v_{0k} \\ o_k &= \tau(o\_sum_k) \end{aligned}\right\}$$



다중 퍼셉트론

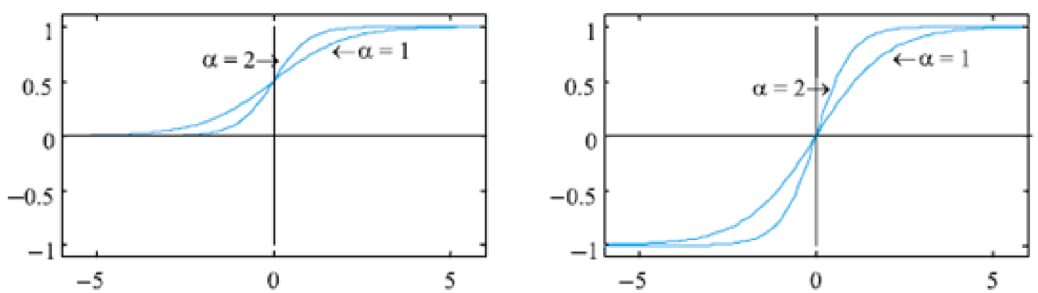
- 활성 함수 (activation function)
  - 시그모이드 사용

이진 시그모이드 함수:

$$\left.\begin{aligned}\tau_1(x) &= \frac{1}{1+e^{-\alpha x}}\\ \tau_1{}'(x) &= \alpha\tau_1(x)(1-\tau_1(x))\end{aligned}\right\} \quad (4.14)$$

양극 시그모이드 함수:

$$\left.\begin{aligned}\tau_2(x) &= \frac{2}{1+e^{-\alpha x}}-1\\ \tau_2{}'(x) &= \frac{\alpha}{2}(1+\tau_2(x))(1-\tau_2(x))\end{aligned}\right\} \quad (4.15)$$

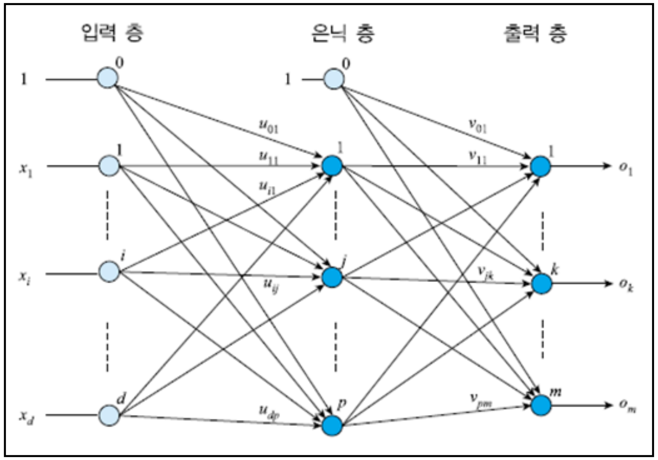


(a) 이진 시그모이드 (b) 양극 시그모이드

그림 4.10 활성 함수로 널리 사용되는 두 가지 시그모이드 함수

다중 퍼셉트론

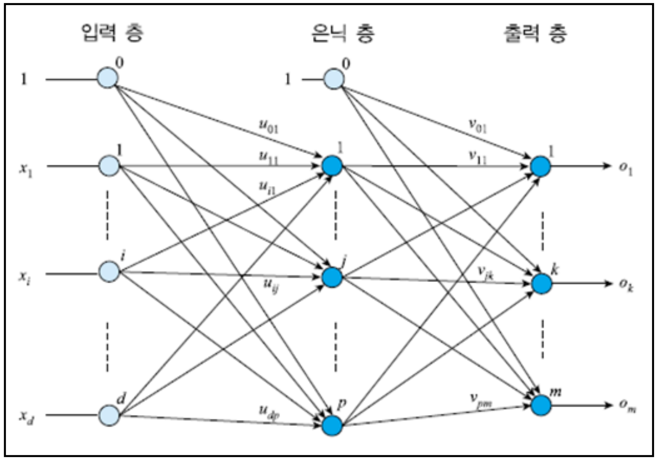
- 신경망 구조 설계 시 고려사항
  - 몇 개의 층을 둘 것 인가?
  - 층간의 연결은 어떻게 할 것인가?
  - 각 층에 있는 노드를 몇 개로 할 것인가?
  - 어떤 활성화 함수를 사용할 것인가?



다중 퍼셉트론

- MLP의 학습이란?

  - MLP 학습이란? 훈련 집합 $X = \{(\mathbf{x}_1, \mathbf{t}_1), (\mathbf{x}_2, \mathbf{t}_2), \cdots, (\mathbf{x}_N, \mathbf{t}_N)\}$이 주어졌을 때 이들을 분류하는 다층 퍼셉트론 (즉 **u**와 **v**)을 찾아라. $(\mathbf{x}_i, \mathbf{t}_i)$에서 $\mathbf{x}_i$는 특징 벡터이고 $\mathbf{t}_i$는 부류 표지 벡터로서 class label vector (또는 목적 벡터라고도 target vector 함) $\mathbf{x}_i \in \omega_j$이면 $\mathbf{t}_i = (0, \cdots, 1, \cdots, 0)^T$이다. 즉 j 번째 요소만 1이고 나머지 요소는 모두 0을 갖는다. 이것은 이진 모드를 사용할 때의 값이고 만일 양극 모드를 사용한다면 $\mathbf{t}_i = (-1, \cdots, 1, \cdots, -1)^T$로 하면 된다.



다중 퍼셉트론

- 단계 1
  - (4.12)와 (4.13)의 전방 계산이 분류기의 식
  - 매개변수 집합 Θ={**u**, **v**}

- 단계 2 (비용 함수 정의)

$$E = \frac{1}{2}\sum_{k=1}^{m}(t_k - o_k)^2 \qquad (4.16)$$

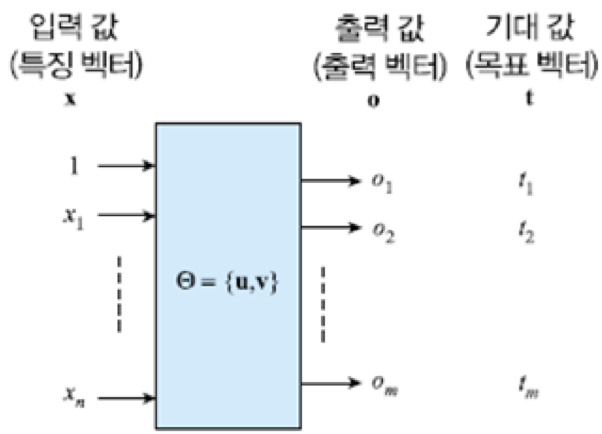


그림 4.12 다층 퍼셉트론의 입력, 출력, 그리고 기대값

다중 퍼셉트론

- 단계 3 (최적 해 찾음)
  - (4.16)의 오류를 줄이는 방향으로 Θ를 수정해 나감

$$\left.\begin{aligned}\mathbf{v}(h+1) &= \mathbf{v}(h) + \Delta\mathbf{v} = \mathbf{v}(h) - \rho\frac{\partial E}{\partial \mathbf{v}} \\ \mathbf{u}(h+1) &= \mathbf{u}(h) + \Delta\mathbf{u} = \mathbf{u}(h) - \rho\frac{\partial E}{\partial \mathbf{u}}\end{aligned}\right\} \qquad (4.17)$$

알고리즘 [4.4] 다층 퍼셉트론 (MLP) 학습

입력: 훈련 집합 $X=\{(\mathbf{x}_1, \mathbf{t}_1), (\mathbf{x}_2, \mathbf{t}_2), \cdots, (\mathbf{x}_N, \mathbf{t}_N)\}$, 학습률 ρ

출력: 가중치 **u**와 **v**

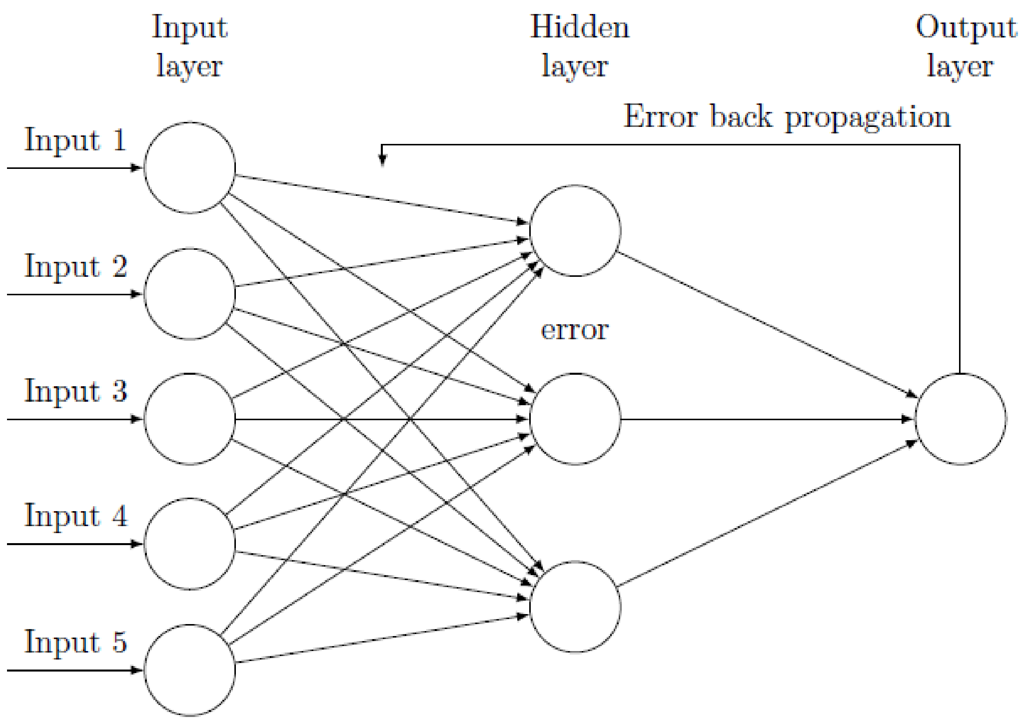
알고리즘:

1. **u**와 **v**를 초기화한다.
2. **repeat** {
3.   **for** (X의 샘플 각각에 대해) {
4.   (4.12)와 (4.13)으로 전방 계산을 한다.
5. $\frac{\partial E}{\partial \mathbf{v}}$와 $\frac{\partial E}{\partial \mathbf{u}}$를 계산한다.
6.   (4.17)로 새로운 **u**와 **v**를 계산한다.
7.   }
8. } **until** (stop-condition);

라인 5를 어떻게?

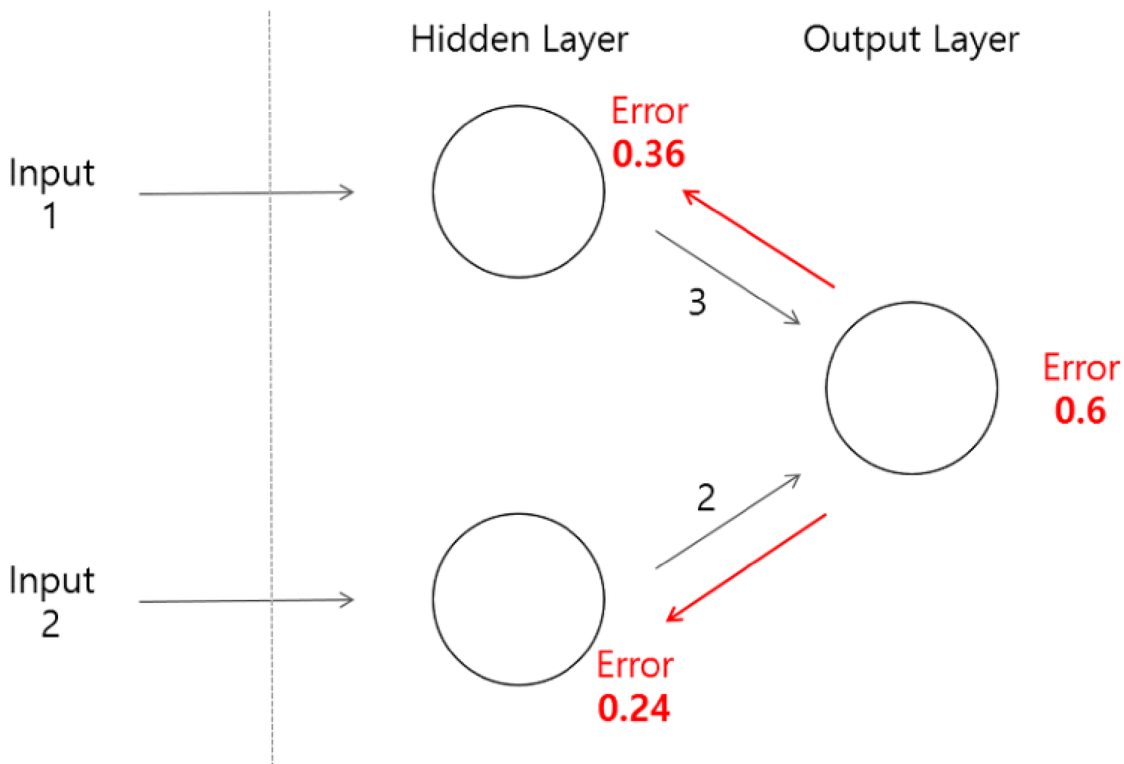
오류 역전파

- 개요
  - 결과에서 입력 방향(역방향)으로 오차를 보내어 가중치를 재업데이트 하는 방법



오류 역전파

- 개요
  - 결과에서 입력 방향(역방향)으로 오차를 보내어 가중치를 재업데이트 하는 방법



그밖에..

- 부연 설명
  - 네트워크 아키텍처
    - 은닉 층 개수, 은닉 노드 개수, …
  - 가중치 초기화
  - 종료 시점
  - 활성함수
  - 학습률
  - …..

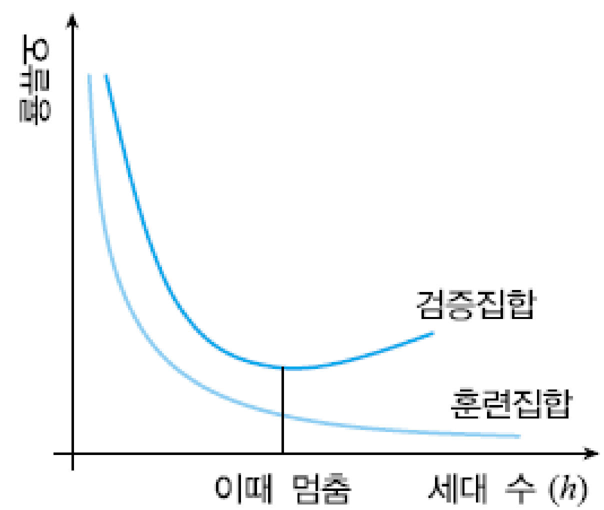


그림 4.15 일반화 기준에 따른 멈춤 조건

참고자료

- Introduction to Machine Leanring with Python (파이썬 라이브러리를 활용한 머신러닝)
  - 안드레아스 뮐러, 세라 가이도 지음 / 박해선 옮김
  - 한빛미디어, 2019
- 패턴인식
  - 오일석 지음
  - 교보문고, 2008
- 밑바닥 부터 시작하는 딥러닝
  - 사이토 고키 지음, 개앞맵시 옮김
  - 한빛미디어, 2017