기계학습 개론

- 신경망 및 퍼셉트론

정보통신공학과

Prof. Jinkyu Kang

MYONGJI UNIVERSITY

이번주 수업의 목차

- 신경망 기초
- 퍼셉트론
- 다층 퍼셉트론
- 오류 역전파 알고리즘
- 다층 퍼셉트론에 의한 인식
- 다층 퍼셉트론의 특성

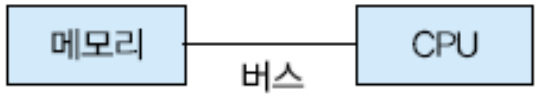
MYONGJI
UNIVERSITY

신경망 기초

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

신경망 기초

- 컴퓨터와 두뇌의 비교
  - 폰 노이만 컴퓨터
    - 순차 명령어 처리기
  - 두뇌
    - 뉴런으로 구성 (약 10^{11}개, 약 10^{14} 연결 (시냅스))
    - 고도의 병렬 명령어 처리기



(a) 폰 노이만 컴퓨터의 구조

(b) 사람 뇌의 정보 처리 단위인 뉴런

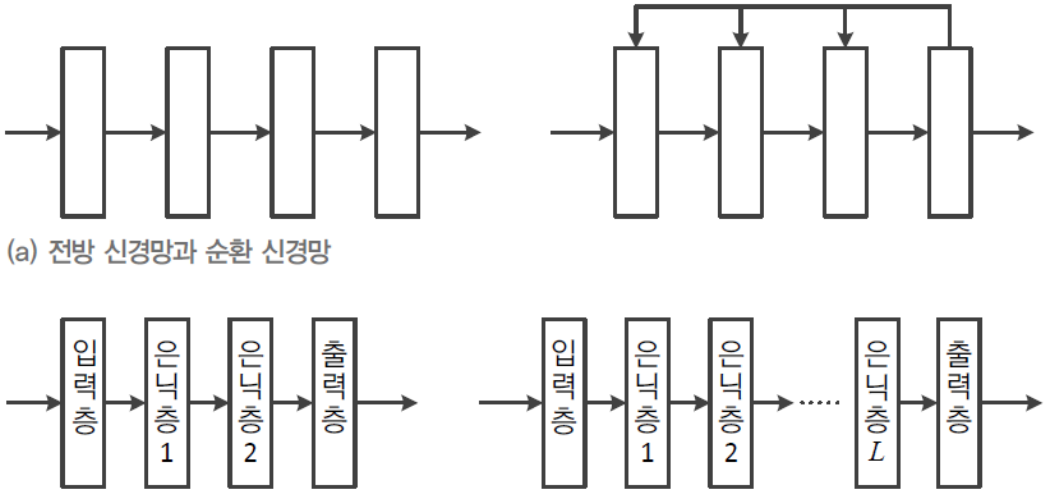
신경망 기초 | 수학적 모델로서의 신경망

- 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)

신경망 기초 | 신경망의 종류

- 신경망에는 아주 다양한 모델이 존재함
  - 전방 신경망과 순환 신경망
  - 얕은 신경망과 깊은 신경망
  - 결정론 신경망과 스토캐스틱 신경망



(a) 전방 신경망과 순환 신경망

(b) 얕은 신경망과 깊은 신경망

그림 3-2 신경망의 종류

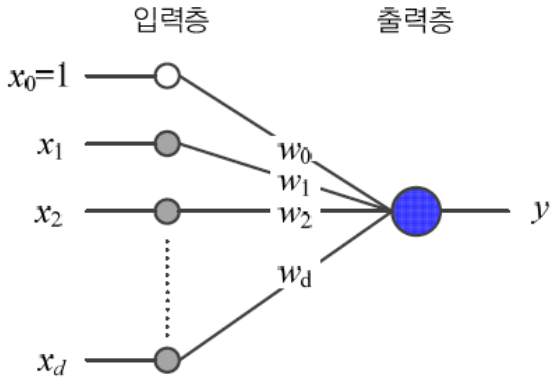
퍼셉트론

- 퍼셉트론은 노드, 가중치, 층과 같은 새로운 개념을 도입하고 학습 알고리즘을 창안함

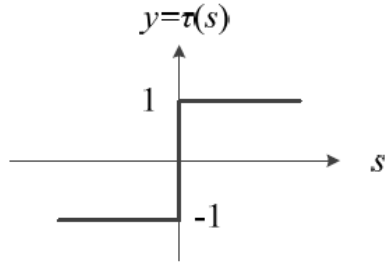
- 퍼셉트론은 원시적 신경망이지만, 딥러닝을 포함한 현대 신경망은 퍼셉트론을 병렬과 순차 구조로 결합하여 만듬 → 현대 신경망의 중요한 구성 요소

퍼셉트론 | 구조

- 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 입력 노드와 출력 노드를 연결하는 에지는 가중치 w_i를 가짐



(a) 퍼셉트론의 구조



(b) 계단함수를 활성함수 $\tau(s)$로 이용함

그림 3-3 퍼셉트론의 구조와 동작

MYONGJI UNIVERSITY

퍼셉트론 | 동작

- 퍼셉트론의 동작
  - 해당하는 특징값과 가중치를 곱한 결과를 모두 더하여 s를 구하고, 활성함수 τ를 적용함
  - 활성함수 τ로 계단함수를 사용하므로 최종 출력 y는 +1 또는 -1

$$y = \tau(s)$$

$$\text{이때 } s = w_0 + \sum_{i=1}^{d} w_i x_i, \qquad \tau(s) = \begin{cases} 1 & s \geq 0 \\ -1 & s < 0 \end{cases} \tag{3.1}$$

MYONGJI UNIVERSITY

퍼셉트론 | 동작

- 행렬 표기

$$s = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0, \qquad \text{여기서 } \mathbf{x} = (x_1, x_2, \cdots, x_d)^{\mathrm{T}},\ \mathbf{w} = (w_1, w_2, \cdots, w_d)^{\mathrm{T}} \tag{3.2}$$

  - 바이어스 항을 벡터에 추가하면,

$$s = \mathbf{w}^{\mathrm{T}}\mathbf{x}, \qquad \text{여기서 } \mathbf{x} = (1, x_1, x_2, \cdots, x_d)^{\mathrm{T}},\ \mathbf{w} = (w_0, w_1, w_2, \cdots, w_d)^{\mathrm{T}} \tag{3.3}$$

  - 퍼셉트론의 동작을 식 (3.4)로 표현할 수 있음

$$y = \tau(\mathbf{w}^{\mathrm{T}}\mathbf{x}) \tag{3.4}$$

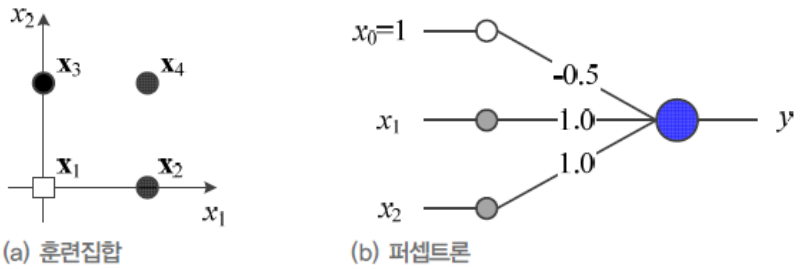
MYONGJI UNIVERSITY

퍼셉트론 | 동작

예제 3-1 **퍼셉트론의 동작**

2차원 특징 벡터로 표현되는 샘플을 4개 가진 훈련집합 $\mathbb{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4\}$, $\mathbb{Y} = \{y_1, y_2, y_3, y_4\}$를 생각하자. [그림 3-4(a)]는 이 데이터를 보여준다.

$$\mathbf{x}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, y_1 = -1, \quad \mathbf{x}_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, y_2 = 1, \quad \mathbf{x}_3 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, y_3 = 1, \quad \mathbf{x}_4 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, y_4 = 1$$



(a) 훈련집합 (b) 퍼셉트론

그림 3-4 OR 논리 게이트를 이용한 퍼셉트론의 동작 예시

샘플 4개를 하나씩 입력하여 제대로 분류하는지 확인해 보자.

$$\begin{aligned}
&\mathbf{x}_1\text{: } s = -0.5 + 0 * 1.0 + 0 * 1.0 = -0.5, && \tau(-0.5) = -1 \\
&\mathbf{x}_2\text{: } s = -0.5 + 1 * 1.0 + 0 * 1.0 = 0.5, && \tau(0.5) = 1 \\
&\mathbf{x}_3\text{: } s = -0.5 + 0 * 1.0 + 1 * 1.0 = 0.5, && \tau(0.5) = 1 \\
&\mathbf{x}_4\text{: } s = -0.5 + 1 * 1.0 + 1 * 1.0 = 1.5, && \tau(1.5) = 1
\end{aligned}$$

결국 [그림 3-4(b)]의 퍼셉트론은 샘플 4개를 모두 맞추었다. 이 퍼셉트론은 훈련집합을 100% 성능으로 분류한다고 말할 수 있다.

퍼셉트론 | 동작

- [그림 3-4(b)]를 기하학적으로 설명하면,
  - 결정 직선 $d(\mathbf{x}) = d(x_1, x_2) = w_1 x_1 + w_2 x_2 + w_0 = 0$ → $x_1 + x_2 - 0.5 = 0$
    - w_1과 w_2는 직선의 방향, w_0은 절편을 결정
    - 결정 직선은 전체 공간을 +1과 -1의 두 부분공간으로 분할하는 분류기 역할

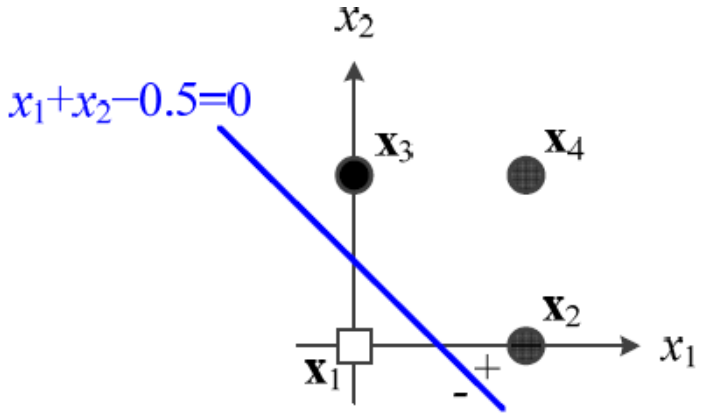
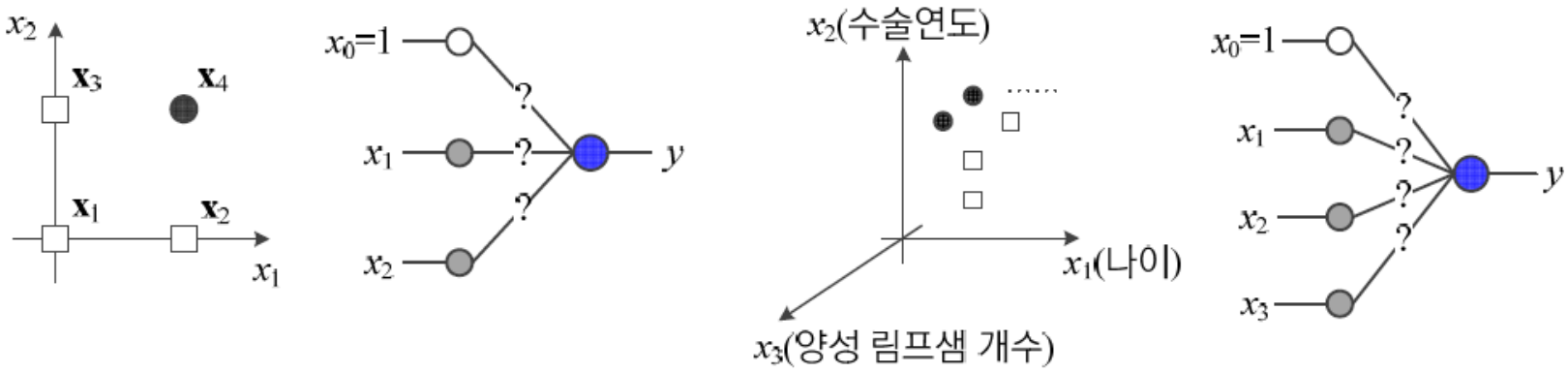


그림 3-5 [그림 3-4(b)]의 퍼셉트론에 해당하는 결정 직선

  - d차원 공간에서는 $d(\mathbf{x}) = w_1 x_1 + w_2 x_2 + \cdots + w_d x_d + w_0 = 0$
    - 2차원은 결정 직선decision line, 3차원은 결정 평면decision plane, 4차원 이상은 결정 초평면decision hyperplane

퍼셉트론 | 학습

- 학습 문제
  - 지금까지는 학습을 마친 퍼셉트론을 가지고 동작을 설명한 셈
  - [그림 3-6]은 학습 문제: w_1과 w_2, w_0이 어떤 값을 가져야 100% 옳게 분류할까?
  - [그림 3-6]은 2차원 공간에 4개 샘플이 있는 훈련집합이지만, 현실 세계는 d차원 공간에 수백~수만 개의 샘플이 존재 (예, MNIST는 784차원에 6만개 샘플)



(a) AND 분류 문제

(b) Haberman survival 분류 문제

그림 3-6 어떻게 학습시킬 것인가?

퍼셉트론 | 학습

- 목적함수 설계
  - 퍼셉트론의 매개변수를 $\mathbf{w} = (w_0, w_1, w_2, \cdots, w_d)^{\mathrm{T}}$라 표기하면, 매개변수 집합은 $\Theta = \{\mathbf{w}\}$
  - 목적함수를 $J(\Theta)$ 또는 $J(\mathbf{w})$로 표기함
  - 목적함수의 조건
    - $J(\mathbf{w}) \geq 0$이다.
    - $\mathbf{w}$가 최적이면, 즉 모든 샘플을 맞히면 $J(\mathbf{w}) = 0$이다.
    - 틀리는 샘플이 많은 $\mathbf{w}$일수록 $J(\mathbf{w})$는 큰 값을 가진다.
  - 식 (3.7)은 세 가지 조건을 만족하므로, 퍼셉트론의 목적함수로 적합
    - Y는 $\mathbf{w}$가 틀리는 샘플의 집합

$$J(\mathbf{w}) = \sum_{\mathbf{x}_k \in Y} -y_k \left(\mathbf{w}^{\mathrm{T}} \mathbf{x}_k\right) \qquad (3.7)$$

MYONGJI UNIVERSITY

퍼셉트론 | 학습

- 내리막 경사법을 위한 그레이디언트 계산
  - 식 (2.58)의 가중치 갱신 규칙 $\boldsymbol{\Theta} = \boldsymbol{\Theta} - \rho\mathbf{g}$를 적용하려면 그레이디언트 $\mathbf{g}$가 필요
  - 식 (3.7)을 편미분하면,

$$\frac{\partial J(\mathbf{w})}{\partial w_i} = \sum_{\mathbf{x}_k \in Y} \frac{\partial(-y_k(w_0 x_{k0} + w_1 x_{k1} + \cdots + w_i x_{ki} + \cdots + w_d x_{kd}))}{\partial w_i} = \sum_{\mathbf{x}_k \in Y} -y_k x_{ki}$$

$$\frac{\partial J(\mathbf{w})}{\partial w_i} = \sum_{\mathbf{x}_k \in Y} -y_k x_{ki}, \qquad i = 0,1,\cdots,d \tag{3.8}$$

  - 편미분 결과인 식 (3.8)을 식 (2.58)에 대입하면,

$$\text{델타 규칙:} \quad w_i = w_i + \rho \sum_{\mathbf{x}_k \in Y} y_k x_{ki}, \qquad i = 0,1,\cdots,d \tag{3.9}$$

퍼셉트론 | 학습

- 퍼셉트론 학습 알고리즘
  - 식 (3.9)를 이용하여 학습 알고리즘을 쓰면,
    - 훈련집합의 샘플을 모두 맞출(즉 $Y = \emptyset$) 때까지 세대epoch(라인 3~9)를 반복함

알고리즘 3-1 퍼셉트론 학습(배치 버전)

입력: 훈련집합 $\mathbb{X}$와 $\mathbb{Y}$, 학습률 ρ
출력: 최적 가중치 $\hat{\mathbf{w}}$

```
1  난수를 생성하여 초기해 w를 설정한다.
2  repeat
3     Y = ∅    // 틀린 샘플 집합
4     for j=1 to n
5        y = τ(w^T x_j)                  // 식 (3.4)
6        if(y ≠ y_j) Y = Y ∪ x_j         // 틀린 샘플을 집합에 추가한다.
7     if(Y ≠ ∅)
8        for i=0 to d                    // 식 (3.9)
9           w_i = w_i + ρ Σ_{x_k∈Y} y_k x_ki
10 until (Y = ∅)
11 ŵ = w
```

MYONGJI UNIVERSITY

퍼셉트론 | 학습

- 행렬 표기
  - 행렬을 사용하여 간결하게 표기: 델타 규칙: $\mathbf{w} = \mathbf{w} + \rho \sum_{\mathbf{x}_k \in Y} y_k \mathbf{x}_k$
  - 행렬 표기로 [알고리즘 3-1]을 수정하면,

$$\left.\begin{array}{l} 8.\ \text{for } i = 0 \text{ to } d \\ 9.\quad w_i = w_i + \rho \sum_{\mathbf{x}_k \in Y} y_k x_{ki} \end{array}\right\} \rightarrow\ 8.\ \mathbf{w} = \mathbf{w} + \rho \sum_{\mathbf{x}_k \in Y} y_k \mathbf{x}_k$$

- 선형분리 불가능한 경우에는 무한 반복
  - until($Y = \emptyset$) 또는 until(quit)를 until(더 이상 개선이 없다면)으로 수정해야 함

MYONGJI UNIVERSITY

퍼셉트론 | 학습과 인식

- 예제 3-1) 초기 가중치 $\boldsymbol{w}(0) = (0.375, -0.5, 0.75)^T$, $\rho = 0.4$

① 결정직선: $\boldsymbol{d}(\boldsymbol{x}) = 0.375 - 0.5x_1 + 0.75x_2$

오분류된 샘플: $Y = \{a, b\}$

새로운 가중치: $\boldsymbol{w}(1) = \boldsymbol{w}(0) + 0.4(y_a\boldsymbol{a} + y_b\boldsymbol{b}) = \begin{bmatrix} 0.375 \\ -0.5 \\ 0.75 \end{bmatrix} + 0.4\left(-1 * \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + 1 * \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}\right)$

$= (0.375, -0.1, 0.75)^T$

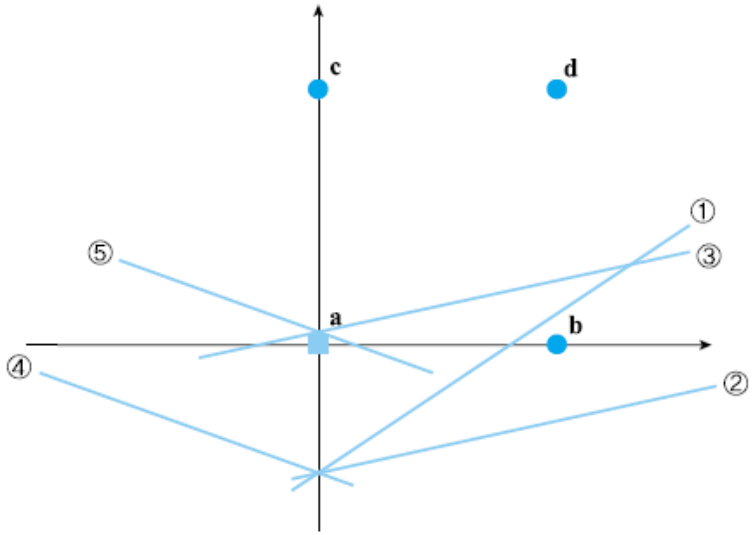
② 결정직선: $\boldsymbol{d}(\boldsymbol{x}) = 0.375 - 0.1x_1 + 0.75x_2$

오분류된 샘플: $Y = \{a\}$

새로운 가중치: $\boldsymbol{w}(2) = \boldsymbol{w}(1) + 0.4(y_a\boldsymbol{a})$

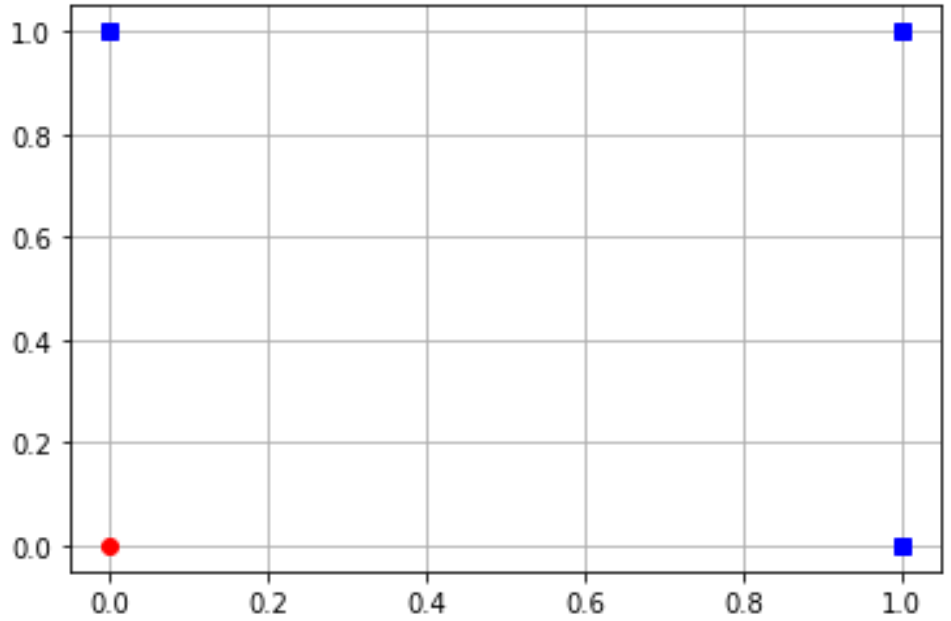
$= \begin{bmatrix} 0.375 \\ -0.1 \\ 0.75 \end{bmatrix} + 0.4\left(-1 * \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}\right)$

$= (-0.025, -0.1, 0.75)^T$



퍼셉트론 | Python을 이용한 예제

- 퍼셉트론을 이용한 OR 연산 예제
  - ●: 0, ■: 1
  - Test sample: [0.5, 0.5], [0.2, 0.2]



MYONGJI UNIVERSITY

다층 퍼셉트론

- 퍼셉트론은 선형 분류기라는 한계
  - [그림 3-7(b)]의 선형 분리 불가능한 상황에서는 일정한 양의 오류
  - 예) XOR 문제에서는 75%가 정확률 한계

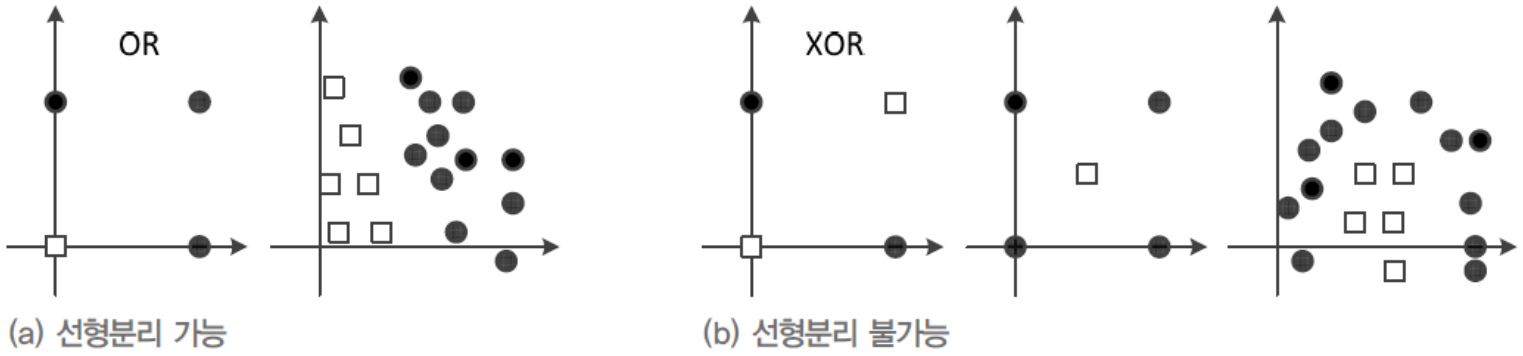


그림 3-7 선형분리가 가능한 상황과 불가능한 상황

다층 퍼셉트론

- 다층 퍼셉트론의 핵심 아이디어

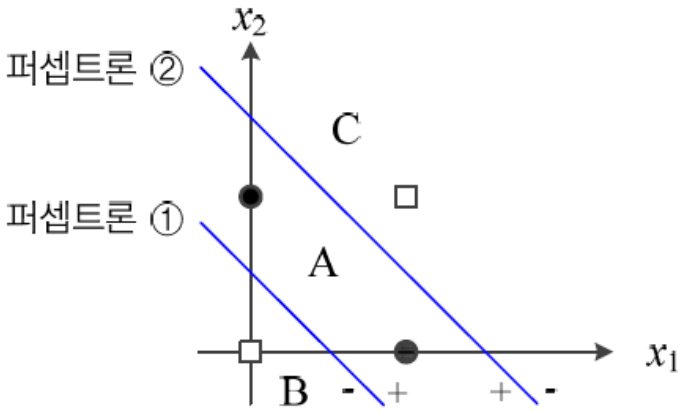
  - 은닉층을 둔다. 은닉층은 원래 특징 공간을 분류하는 데 훨씬 유리한 새로운 특징 공간으로 변환한다. 3.3.1절에서 다층 퍼셉트론의 공간 변환 능력을 설명한다.

  - 시그모이드 활성함수를 도입한다. 퍼셉트론은 [그림 3-3(b)]의 계단함수를 활성함수로 사용하였다. 이 함수는 경성hard 의사결정에 해당한다. 반면, 다층 퍼셉트론은 연성soft 의사결정이 가능한 [그림 3-12]의 시그모이드함수를 활성함수로 사용한다. 연성에서는 출력이 연속값인데, 출력을 신뢰도로 간주함으로써 더 융통성 있게 의사결정을 할 수 있다. 3.3.2절에서 시그모이드 활성함수를 자세히 설명한다.

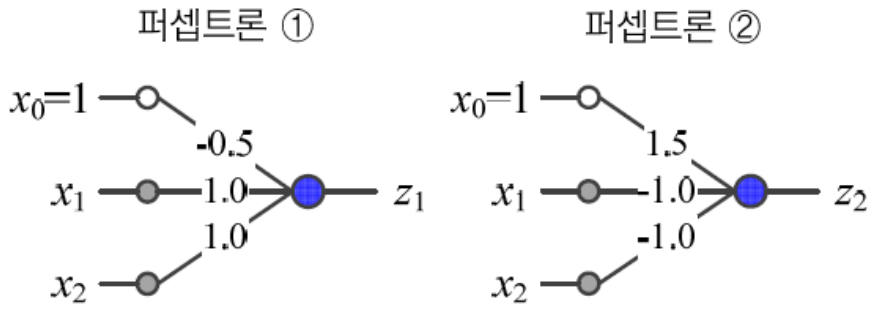
  - 오류 역전파 알고리즘을 사용한다. 다층 퍼셉트론은 여러 층이 순차적으로 이어진 구조이므로, 역방향으로 진행하면서 한 번에 한 층씩 그레이디언트를 계산하고 가중치를 갱신하는 방식의 오류 역전파 알고리즘을 사용한다. 이 학습 알고리즘에 대해서는 3.4절에서 다룬다.

다층 퍼셉트론 | 특징 공간 변환

- 퍼셉트론 2개를 사용한 XOR 문제의 해결
  - 퍼셉트론①과 퍼셉트론②가 모두 +1이면 ● 부류이고 그렇지 않으면 □ 부류임



(a) 퍼셉트론 2개를 이용한 공간분할



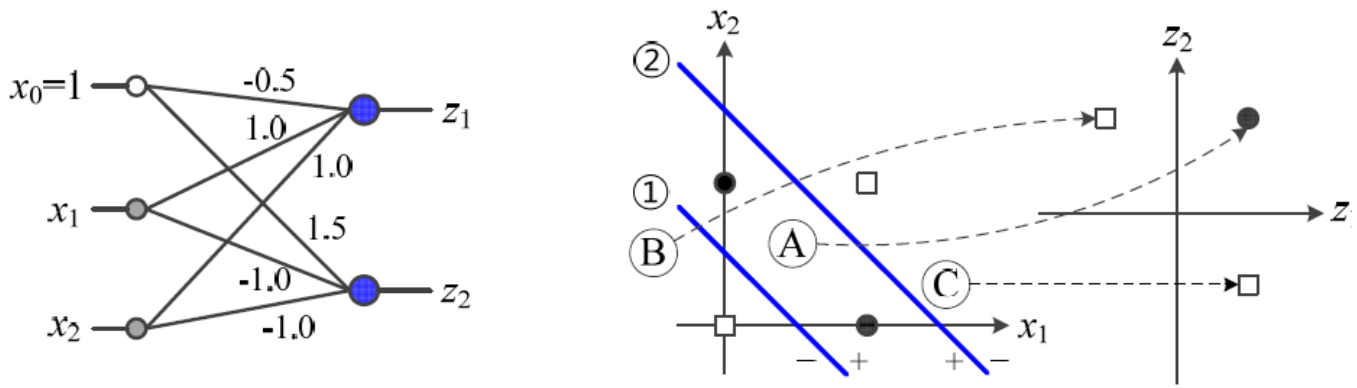
(b) 퍼셉트론 2개

그림 3-8 XOR 문제의 해결

MYONGJI UNIVERSITY

다층 퍼셉트론 | 특징 공간 변환

- 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 x를 새로운 특징 공간 z로 변환

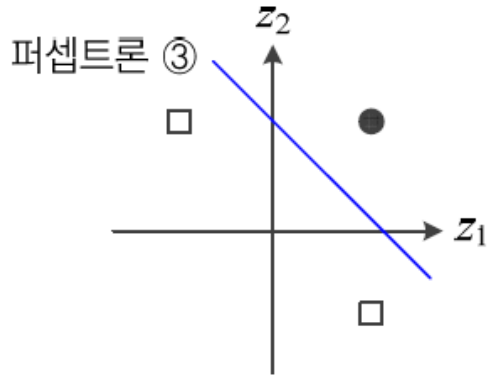
그림 3-9 특징 공간의 변환

  - 사람이 수작업으로 특징 학습을 수행한 셈

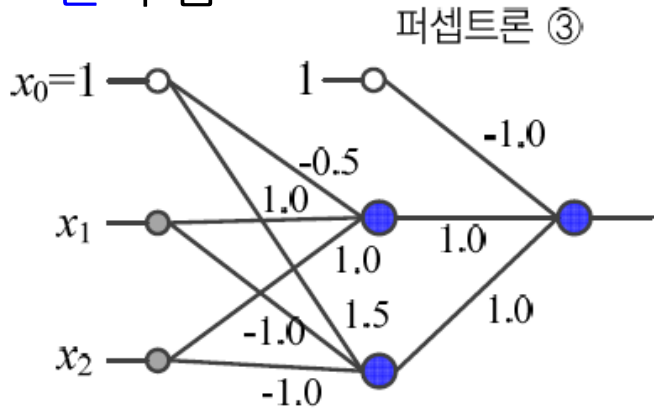
MYONGJI UNIVERSITY

다층 퍼셉트론 | 특징 공간 변환

- 퍼셉트론 1개를 순차 결합하면,
  - 새로운 특징 공간 **z**에서 선형 분리를 수행하는 퍼셉트론③을 순차 결합하면, [그림 3-10(b)]의 다층 퍼셉트론이 됨



(a) 새로운 특징 공간에서 분할



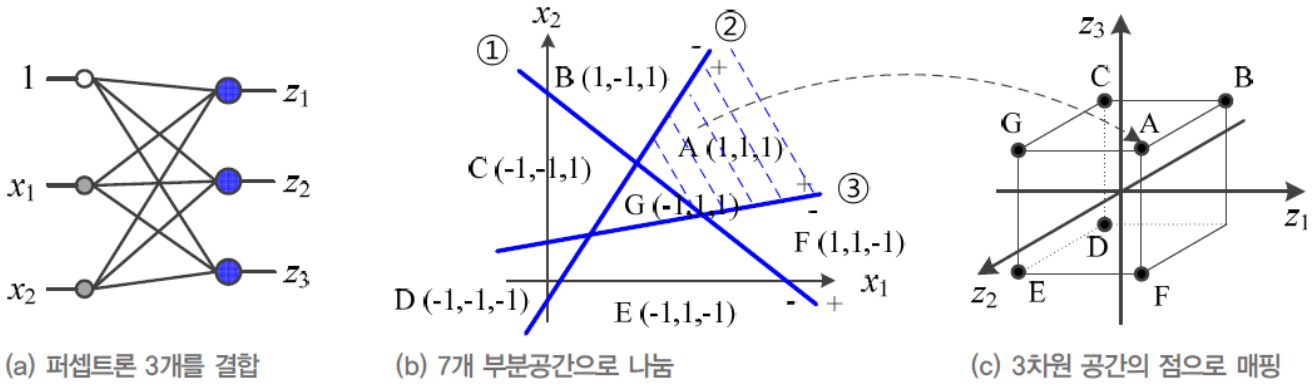
(b) 퍼셉트론 3개를 결합한 다층 퍼셉트론

그림 3-10 다층 퍼셉트론

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

다층 퍼셉트론 | 특징 공간 변환

- 다층 퍼셉트론의 용량
  - [그림 3-11]처럼 3개 퍼셉트론을 결합하면, 2차원 공간을 7개 영역으로 나누고 각 영역을 3차원 점으로 변환
  - 활성함수 τ로 계단함수를 사용하므로 영역을 점으로 변환



(a) 퍼셉트론 3개를 결합 (b) 7개 부분공간으로 나눔 (c) 3차원 공간의 점으로 매핑

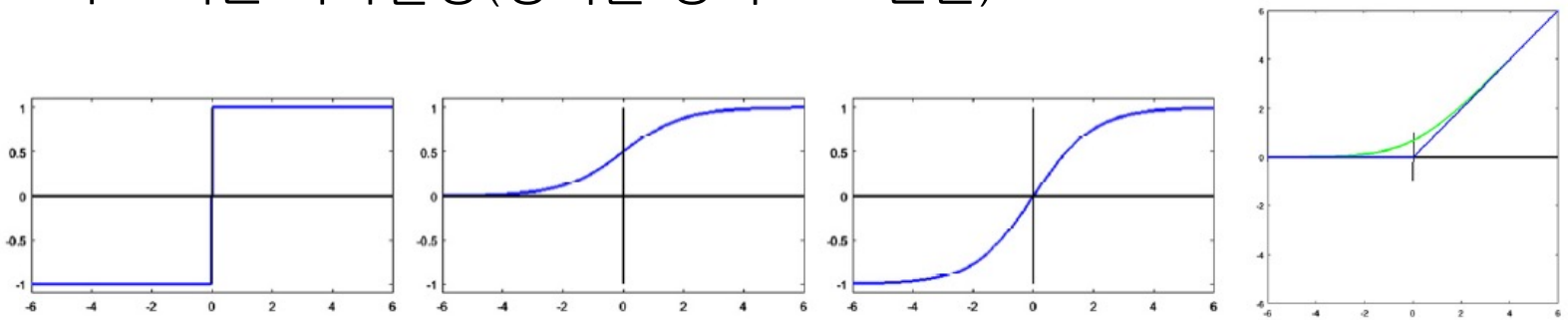
그림 3-11 퍼셉트론을 3개 결합했을 때 공간 변환

  - 일반화하여, p개 퍼셉트론을 결합하면 p차원 공간으로 변환
    - $1 + \sum_{i=1}^{p} i$개의 영역으로 분할

MYONGJI UNIVERSITY

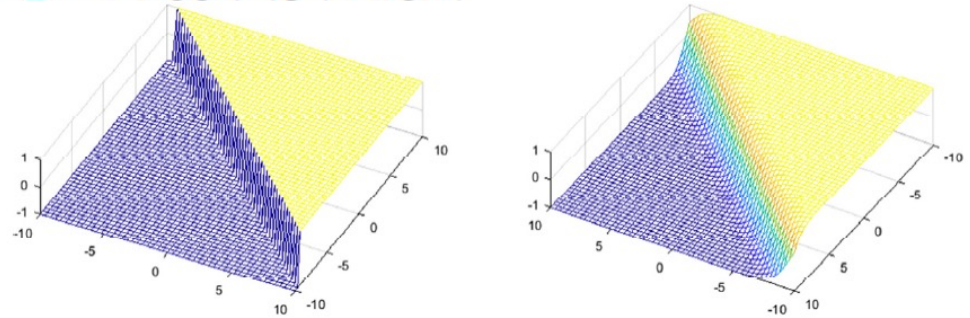
다층 퍼셉트론 | 활성 함수

- 딱딱한 공간 분할과 부드러운 공간 분할
  - 계단함수는 딱딱한 의사결정(영역을 점으로 변환). 나머지 활성함수는 부드러운 의사결정(영역을 영역으로 변환)



(a) 계단 함수 (b) 로지스틱 시그모이드 (c) 하이퍼볼릭 탄젠트 시그모이드 (d) softplus와 rectifier

그림 3-12 신경망이 사용하는 활성함수



(a) 계단함수의 딱딱한 공간 분할 (b) 로지스틱 시그모이드의 부드러운 공간 분할

그림 3-13 퍼셉트론의 공간 분할 유형

MYONGJI UNIVERSITY

다층 퍼셉트론 | 활성 함수

- 신경망이 사용하는 다양한 활성함수
  - 로지스틱 시그모이드와 하이퍼볼릭 탄젠트는 *a*가 커질수록 계단함수에 가까워짐
  - 모두 1차 도함수 계산이 빠름 (특히 ReLU는 비교 연산 한 번)

표 3-1 활성함수로 사용되는 여러 함수

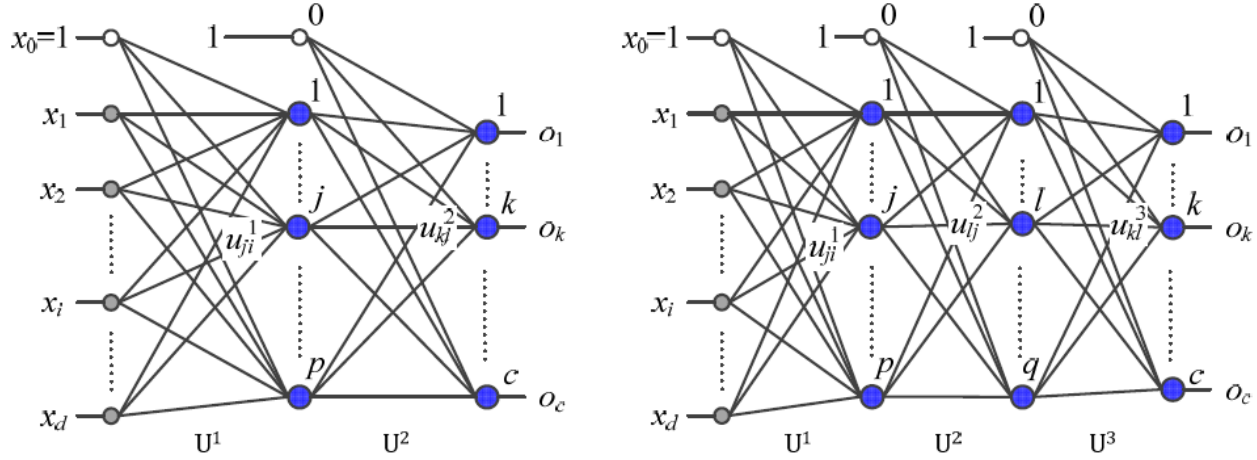
함수 이름	함수	1차 도함수	범위
계단	$\tau(s)=\begin{cases}1 & s\geq 0\\ -1 & s<0\end{cases}$	$\tau'(s)=\begin{cases}0 & s\neq 0\\ \text{불가} & s=0\end{cases}$	-1과 1
로지스틱 시그모이드	$\tau(s)=\frac{1}{1+e^{-as}}$	$\tau'(s)=a\tau(s)(1-\tau(s))$	(0,1)
하이퍼볼릭 탄젠트	$\tau(s)=\frac{2}{1+e^{-as}}-1$	$\tau'(s)=\frac{a}{2}(1-\tau(s)^2)$	(-1,1)
소프트플러스	$\tau(s)=\log_e(1+e^s)$	$\tau'(s)=\frac{1}{1+e^{-s}}$	$(0,\infty)$
렉티파이어(ReLU)	$\tau(s)=\max(0,s)$	$\tau'(s)=\begin{cases}0 & s<0\\ 1 & s>0\\ \text{불가} & s=0\end{cases}$	$[0,\infty)$

  - 퍼셉트론은 계단함수, 다층 퍼셉트론은 로지스틱 시그모이드와 하이퍼볼릭 탄젠트, 딥러닝은 ReLU를 주로 사용

MYONGJI UNIVERSITY

다층 퍼셉트론 | 구조

- [그림 3-14(a)]는 입력층-은닉층-출력층의 2층 구조
  - d+1개의 입력 노드 (d는 특징의 개수). c개의 출력 노드 (c는 부류 개수)
  - p개의 은닉 노드: p는 하이퍼 매개변수(사용자가 정해주는 매개변수)
    - p가 너무 크면 과잉적합, 너무 작으면 과소적합 → 5.5절의 하이퍼 매개변수 최적화
- [그림 3-14(b)]는 입력층-은닉층-은닉층-출력층의 3층 구조



(a) 2층 퍼셉트론 (b) 3층 퍼셉트론

그림 3-14 다층 퍼셉트론의 구조

다층 퍼셉트론 | 구조

- 다층 퍼셉트론의 매개변수(가중치)
  - 입력층-은닉층을 연결하는 $\mathbf{U}^1$ (u_{ji}^1은 입력층의 i번째 노드를 은닉층의 j번째 노드와 연결)
  - 은닉층-출력층을 연결하는 $\mathbf{U}^2$ (u_{kj}^2는 은닉층의 j번째 노드를 출력층의 k번째 노드와 연결)

2층 퍼셉트론의 가중치 행렬:

$$\mathbf{U}^1 = \begin{pmatrix} u_{10}^1 & u_{11}^1 & \cdots & u_{1d}^1 \\ u_{20}^1 & u_{21}^1 & \cdots & u_{2d}^1 \\ \vdots & \vdots & \ddots & \vdots \\ u_{p0}^1 & u_{p1}^1 & \cdots & u_{pd}^1 \end{pmatrix}, \quad \mathbf{U}^2 = \begin{pmatrix} u_{10}^2 & u_{11}^2 & \cdots & u_{1p}^2 \\ u_{20}^2 & u_{21}^2 & \cdots & u_{2p}^2 \\ \vdots & \vdots & \ddots & \vdots \\ u_{c0}^2 & u_{c1}^2 & \cdots & u_{cp}^2 \end{pmatrix} \tag{3.11}$$

  - 일반화하면 u_{ji}^l은 $l-1$번째 은닉층의 i번째 노드를 l번째 은닉층의 j번째 노드와 연결하는 가중치
    - 입력층을 0번째 은닉층 , 출력층을 마지막 은닉층으로 간주

MYONGJI UNIVERSITY

다층 퍼셉트론 | 동작

- 특징 벡터 **x**를 출력 벡터 **o**로 매핑하는 함수로 간주할 수 있음

2층 퍼셉트론: $\mathbf{o} = \mathbf{f}(\mathbf{x}) = \mathbf{f}_2\left(\mathbf{f}_1(\mathbf{x})\right)$

3층 퍼셉트론: $\mathbf{o} = \mathbf{f}(\mathbf{x}) = \mathbf{f}_3\left(\mathbf{f}_2\left(\mathbf{f}_1(\mathbf{x})\right)\right)$ (3.12)

$$\mathbf{o} = \boldsymbol{\tau}\left(\mathbf{U}^2\boldsymbol{\tau}_h\left(\mathbf{U}^1\mathbf{x}\right)\right) \quad (3.15)$$

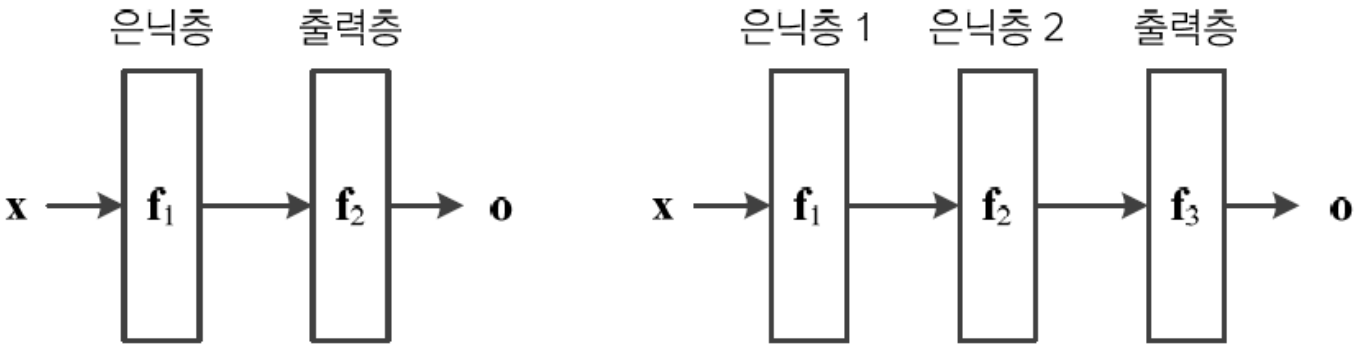


그림 3-15 다층 퍼셉트론을 간략화한 구조

- 깊은 신경망은 $\mathbf{o} = \mathbf{f}_L\left(\cdots \mathbf{f}_2\left(\mathbf{f}_1(\mathbf{x})\right)\right), L \geq 4$ ← 딥러닝

MYONGJI UNIVERSITY

다층 퍼셉트론 | 동작

- 은닉층은 특징 추출기
  - 은닉층은 특징 벡터를 분류에 더 유리한 새로운 특징 공간으로 변환
  - 현대 기계 학습에서는 특징 학습이라feature learning 부름 (딥러닝은 더 많은 단계를 거쳐 특징학습을 함)

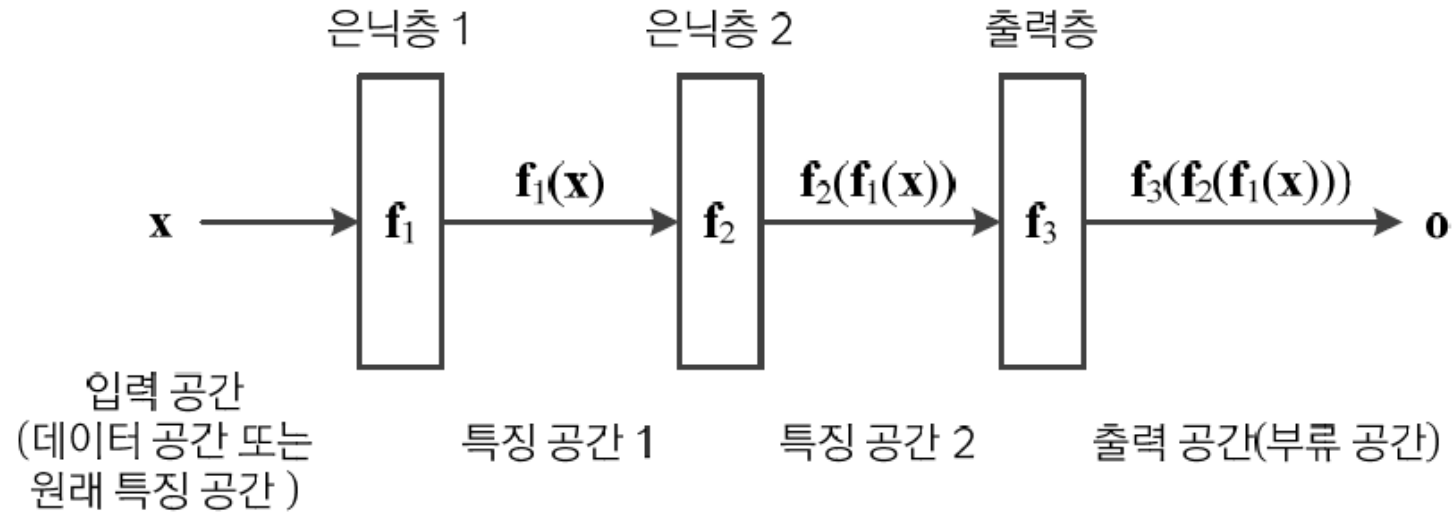


그림 3-16 특징 추출기로서의 은닉층

MYONGJI UNIVERSITY

다층 퍼셉트론의 특성

- 실용적인 성능
  - 1980~1990년대에 다층 퍼셉트론은 실용 시스템 제작에 크게 기여
    - 인쇄/필기 문자 인식으로 우편물 자동 분류기, 전표 인식기, 자동차 번호판 인식기 등
    - 음성 인식, 게임, 주가 예측, 정보 검색, 의료 진단, 유전자 검색, 반도체 결함 검사 등
- 하지만 한계 노출
  - 잡음이 섞인 상황에서 음성인식 성능 저하
  - 필기 주소 인식 능력 저하
  - 바둑에서의 한계
- 딥러닝은 이들 한계를 극복함

MYONGJI UNIVERSITY

다층 퍼셉트론의 특성

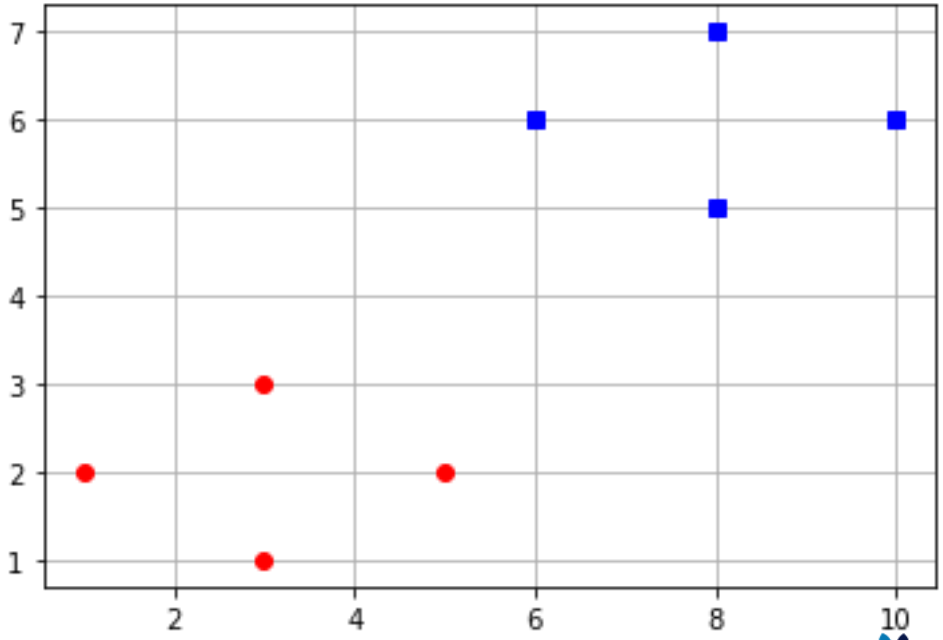
- 매개변수 설정
  - 일반적인 경우에 적용되는 보편 규칙은 없다.
  - 경험과 실험을 통해 설정해야 한다.
  - 신경망 성능이 매개변수에 아주 민감하지는 않기 때문에 어느 정도의 실험과 경험을 통해 설정 가능

6주차 예제

1. 아래와 같이 부류 1(ω_1)과 부류 2 (ω_2)가 샘플을 가졌다. Python을 통해 퍼셉트론을 학습하고 $(6,4)^T$ 테스트 샘플이 어느 부류인지 찾아라.

ω_1(●): $(1,2)^T, (3,1)^T, (5,2)^T, (3,3)^T$

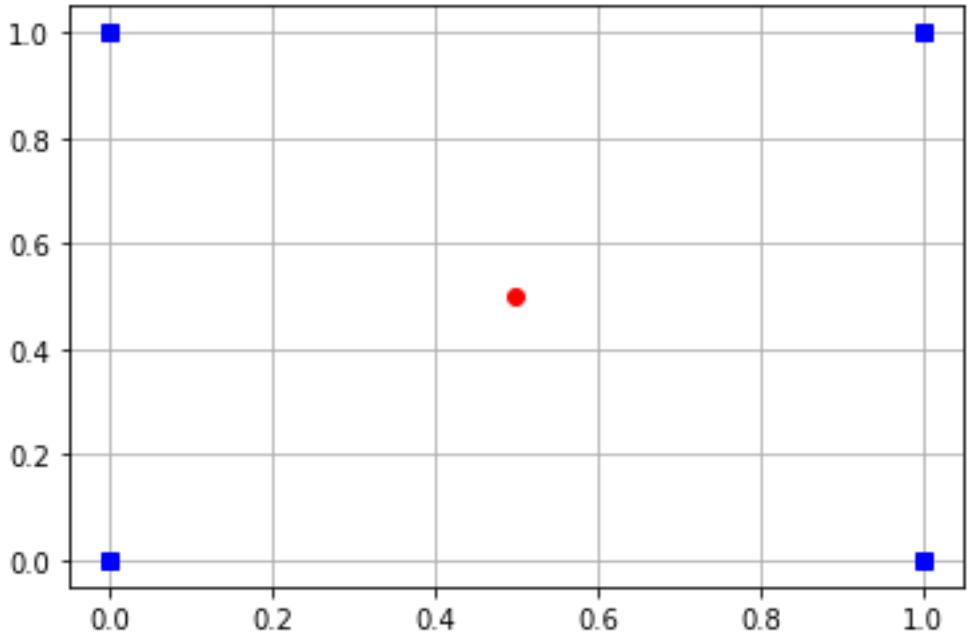
ω_2(■): $(6,6)^T, (8,5)^T, (10,6)^T, (8,7)^T$



MYONGJI UNIVERSITY

6주차 예제

2. 아래 그림에 해당하는 다층 퍼셉트론을 제시하고 옳게 분류되는지 계산하시오.



MYONGJI UNIVERSITY

Thank you

MYONGJI
UNIVERSITY