



기계학습

4강 선형분류(2), 신경망(1)

장필훈 교수



학습목차

- 1 퍼셉트론 알고리즘
- 2 확률적 생성모델/판별모델
 - 모델링 수식, 로지스틱 회귀
- 3 신경망(1)



01

퍼셉트론



1 퍼셉트론 알고리즘

- 퍼셉트론

$$y(\vec{x}) = f(\vec{w}^T \phi(\vec{x}))$$

$$\text{where } f(a) = \begin{cases} +1, a \geq 0 \\ -1, a < 0 \end{cases}$$

$\phi(\cdot)$: 비선형변환



1 퍼셉트론 알고리즘

- 퍼셉트론 기준

$$C_1 \ni \vec{x}_n \text{ 이면 } \vec{w}^T \phi(\vec{x}_n) > 0,$$

$$C_2 \ni \vec{x}_n \text{ 이면 } \vec{w}^T \phi(\vec{x}_n) < 0$$

t 를 $+1$ 이나 -1 로 가정하면, 모든 패턴에 대해

$\vec{w}^T \phi(\vec{x}_n) t_n > 0$ 을 만족하는 \vec{w} 을 찾고자 함



1 퍼셉트론 알고리즘

- 퍼셉트론 기준(cont.)

올바르게 분류되면 0, 오분류되면 $-\vec{w}^T \phi(\vec{x}_n)t_n$ 을
penalty로 줌. 따라서,

$$\mathbb{E}_p(\vec{w}) = -\sum_{n \in M} \vec{w}^T \phi_n t_n$$

($\phi_n = \phi(\vec{x}_n)$, M 은 오분류된 패턴전체)



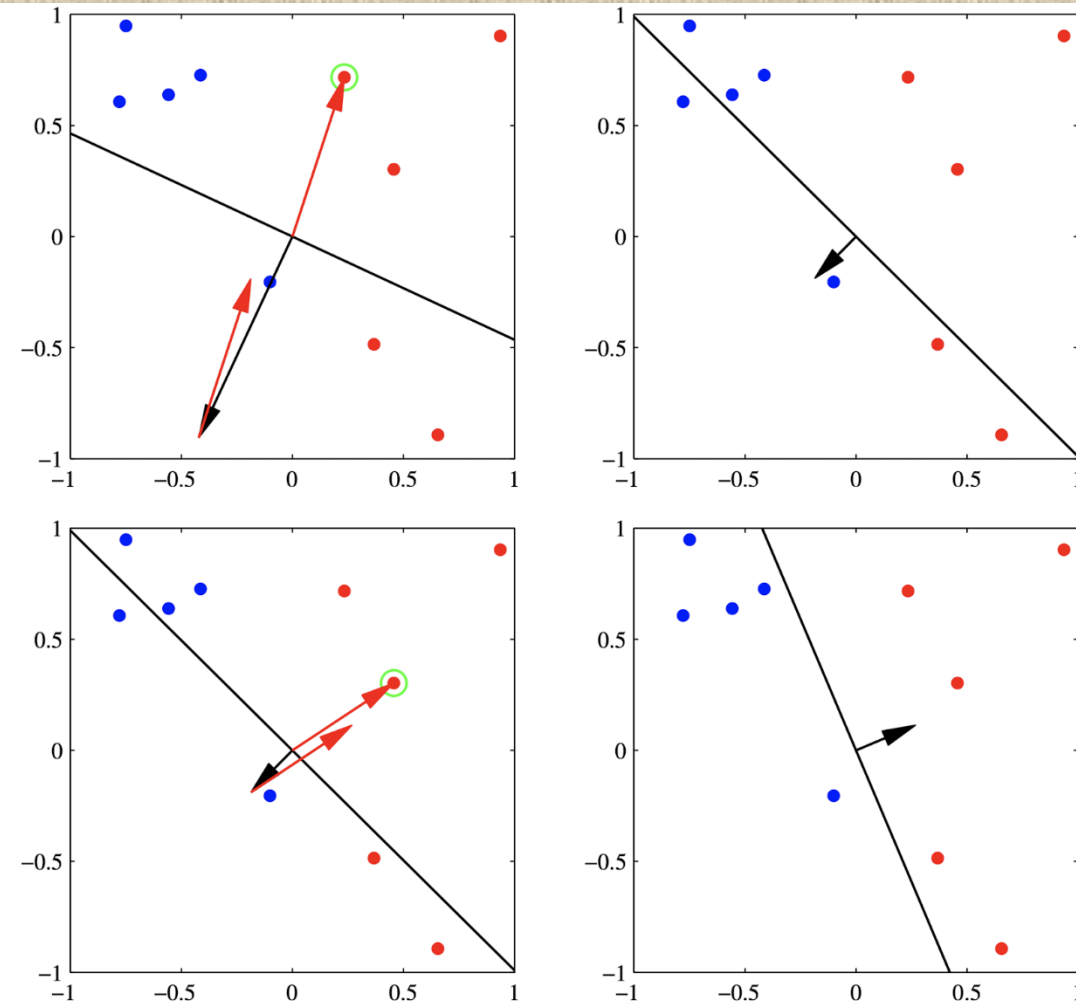
1 퍼셉트론 알고리즘

- 오분류된 패턴에서만 오류함수가 정의되므로,
모든 공간(오분류영역+올바르게 분류된 영역)에서
오류함수는 조각별 선형(미분 불가능하다)
- 퍼셉트론 기준의 계산 : 경사하강법

$$\vec{w}^{(\tau+1)} = \vec{w}^{(\tau)} - \eta \nabla \mathbb{E}_p(\vec{w}) = \vec{w}^{(\tau)} + \eta \phi_n t_n \text{ (2강 참고)}$$

1

퍼셉트론 알고리즘



Bishop, Christopher M. Pattern Recognition and Machine Learning. New York :Springer, 2006. p.195



1 퍼셉트론 알고리즘

- 오분류패턴의 오류함수에 대한 기여도는 점점 감소

$$-\vec{w}^{(\tau+1)} \phi_n t_n = (\vec{w}^{(\tau)} + \eta \phi_n t_n)^T \phi_n t_n$$

$$= -\vec{w}^{(\tau)T} \phi_n t_n - (\phi_n t_n)^T \phi_n t_n < -\vec{w}^{(\tau)T} \phi_n t_n$$

- 단, 오분류된 패턴 때문에 반대로 계산되면
기여도 감소를 보장하지 않는다.



1-1 퍼셉트론 수렴정리

- 훈련집합이 선형분리 가능하면, 퍼셉트론 학습 알고리즘은 **정확한 해를 유한한 단계로 구할 수 있다.**
- 문제점
 1. 유한하다 \neq 짧다
 2. 문제가 선형분리 가능한지 알 수 없다.



1-1 퍼셉트론 수렴정리

- 선형분리 가능하더라도 답이 하나가 아닐 수 있다.
- 데이터가 알고리즘에 입력되는 순서가 해를 결정
- 선형분리 불가능하면,
 - 수렴하지 않는다.



1-2 퍼셉트론 관련문제

1. 확률적인 출력값을 내지 않는다.
2. 다중클래스($K > 2$)문제에 대해 일반화되지 않는다.
3. 고정된 기저함수의 선형결합으로 이루어져 있다.

(지금까지 살펴본 모든 모델이 동일하다)



02

확률적 모델



2

확률적 생성모델, 판별모델

- K=2일때,

$$p(C_1|\vec{x}) = \frac{p(\vec{x}|C_1)p(C_1)}{p(\vec{x}|C_1)p(C_1) + p(\vec{x}|C_2)p(C_2)} = \frac{1}{1 + \exp(-a)} = \sigma(a)$$

where

$$a = \ln \frac{p(\vec{x}|C_1)p(C_1)}{p(\vec{x}|C_2)p(C_2)}$$



2 확률적 생성모델, 판별모델

- 로지스틱 시그모이드
 - squashing function
 - $\sigma(-a) = 1 - \sigma(a)$
 - $a = \ln \frac{\sigma}{1-\sigma}$
 - logit = log odds
 - 두 클래스에 대한 확률비의 로그값 = $\ln \frac{p(C_1|\vec{x})}{p(C_2|\vec{x})}$



2 확률적 생성모델, 판별모델

- $K > 2$ 의 경우

$$p(C_k | \vec{x}) = \frac{p(\vec{x} | C_k) p(C_k)}{\sum_j p(\vec{x} | C_j) p(C_j)} = \frac{\exp(a_k)}{\sum_j \exp(a_j)}$$

(정규화된 지수함수)

where,

$$a_k = \ln(p(\vec{x} | C_k) p(C_k)) : \text{softmax function}$$



2 확률적 생성모델, 판별모델

- 연속입력

클래스별 조건부 밀도 $p(\vec{x}|C_k)$ 가 가우시안이라고 가정.

모든 클래스들이 같은 공분산행렬을 공유한다고 가정.

- 선형으로 결정경계가 나옴을 보일 수 있다.
- 공분산행렬을 공유한다는 가정을 하지 않으면 결정경계로 \vec{x} 의 이차함수를 얻는다.
- 지수족 분포에 모두 적용됨.



2 확률적 생성모델, 판별모델

- 고정된 기저함수
 - 지금까지는 \vec{x} 에 직접 적용되는 형식을 배움
 - 모든 알고리즘은 기저함수들의 벡터 $\vec{\phi}(\vec{x})$ 을 이용하여 비선형변환 후 동일하게 적용 가능
 - 특징공간 $\vec{\phi}$ 에서는 선형, \vec{x} 공간에서는 비선형
 - \vec{x} 에서 선형분리 불가능해도 $\vec{\phi}$ 에서 가능할 수 있음.



2-1 로지스틱 회귀

- 사후확률을 로지스틱 시그모이드 함수로 적을 수 있으므로 최대가능도방법을 이용해서 모델의 매개변수 구하기 가능.

$$p(C_1|\phi) = \eta(\phi) = \sigma(\vec{w}^T \vec{\phi})$$

$$(p(C_2|\phi) = 1 - p(C_1|\phi))$$

$\phi(\cdot)$ 은 로지스틱 시그모이드

- 로지스틱회귀라고 불리지만 **분류**모델

2-1 로지스틱 회귀

- 매개변수 구하기 : 미분 $\frac{d\sigma}{da} = \sigma(1 - \sigma)$ 이용
- 가능도함수

$$p(\vec{t}|w) = \prod_{n=1}^N y_n^{t_n} \{1 - y_n\}^{1-t_n}$$

where $\vec{t} = (t_1, \dots, t_n)^T,$

$$\vec{y}_n = p(C_1|\phi_n) = \sigma(a_n) = \sigma(\vec{w}^T \phi_n),$$

$$\mathbb{E}(w) = -\ln p(\vec{t}|w)$$

2-1 로지스틱 회귀

- 매개변수 구하기(cont.)

$$\nabla \mathbb{E}(w) = \sum_{n=1}^N (y_n - t_n) \phi_n$$

$y_n - t_n$: 표적값과 예측값의 차이

ϕ_n : 기저함수벡터

→ 선형회귀모델의 제곱오차함수의 기울기와 정확히 일치



2-1 로지스틱 회귀

- 매개변수 구하기(cont.)
 - 앞에서는 \vec{w} 에 대해 정리하는 것까지 했었지만,
 σ 함수가 들어가면 해를 닫힌 형태로 구할 수 없다.
 - Newton-Raphson 방법으로 근사한다.
 - 다중 클래스도 가능하다.

(closed form: solution이 analytic.

유한개의 수학적 표현을 사용해 표현 가능)

2-2 Canonical link function

- 타겟변수의 조건부분포가 지수족에 포함되면,
 $\nabla \mathbb{E}(w)$ 의 형태가 아래 형태와 같다.
일반적인 케이스에서 증명 가능하다.

$$\nabla \mathbb{E}(w) = \beta \sum_{n=1}^N (y_n - t_n) \phi_n$$



03

신경망(1)



3-1 개요

- 앞서 살펴본 기저함수들의 선형결합은, **계산할 수 있고** 이해하기도 쉽다는 장점이 있음.
- 실제 사용에는 한계가 있음 : 데이터의 양, 매우 큰 차원
- 큰 스케일의 문제에 적용하려면,
 1. 기저함수를 늘리는 방법
 2. 기저함수를 data에 adaptive하게 만드는 방법



3-2 다층 퍼셉트론

- 매개변수적인 기저함수
 - = 피드포워드 뉴럴 네트워크
- 퍼셉트론 여러층이 아님
 - 로지스틱 회귀모델 여러층
- SVM과 같은 일반화 능력. 모델은 작다



3-2 다층 퍼셉트론

- 가능도함수가 볼록함수가 아니다
 - 시작점에 따라 다른 결과로 수렴할 수 있다.

3-2 다층 퍼셉트론

- 피드포워드 네트워크 함수

$$y(\vec{x}, \vec{w}) = f\left(\sum_{j=1}^M w_j \phi_j(x)\right)$$


f : 비선형 활성화함수(분류) or 항등함수(회귀)

→ $\phi_j(x)$ 가 훈련단계에서 w_j 와 함께 조절된다.

3-2 다층 퍼셉트론

- 피드포워드 네트워크 함수(cont.)

$$a_j = \sum_{i=1}^D w_{ji}^{(1)} x_i + w_{j0}^{(1)}$$

$$z_j = h(a_j)$$


3-2 다층 퍼셉트론

- 피드포워드 네트워크 함수(cont.)

$$a_j = \sum_{i=1}^D w_{kj}^{(2)} x_i + w_{k0}^{(2)} \quad \text{반복}$$

합하면?

$$y_k(\vec{x}, \vec{w}) = \sigma \left(\sum_{j=1}^M w_{kj}^{(2)} h \left(\sum_{i=1}^D w_{ji}^{(1)} x_i + w_{j0}^{(1)} \right) + w_{k0}^{(2)} \right)$$

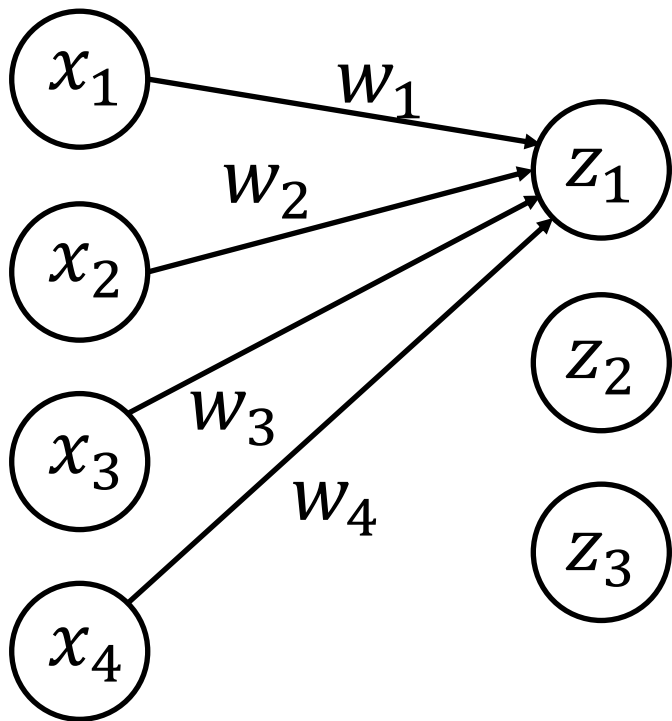
3-2 다층 퍼셉트론

- 피드포워드 네트워크 함수(cont.)

bias는 없앨 수 있다.

$$y_k(\vec{x}, \vec{w}) = \sigma \left(\sum_{j=1}^M w_{kj}^{(2)} h \left(\sum_{i=1}^D w_{ji}^{(1)} x_i \right) \right)$$

3-2 다층 퍼셉트론



$$\begin{pmatrix} w_1 & w_2 & w_3 & w_4 \\ w_5 & w_6 & w_7 & w_8 \\ w_9 & w_{10} & w_{11} & w_{12} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix}$$

fully connected network



3-2 다층 퍼셉트론

- 피드포워드 네트워크 함수(cont.)
 1. 선형변환이 아니다(activation때문)
 2. 점점 유닛수를 줄인다.(차원감소)
 - cf) 선형으로 차원을 감소시킬 수 있다(PCA)
- 퍼셉트론과 차이점
 - 연속적 시그모이드 비선형함수 vs 불연속적 비선형 계단함수

3-2 다층 퍼셉트론

- 그 외 가능한 구조
 - skip layer
 - sparse connection
- universal approximator

『모든』연속함수를 원하는 만큼의 정확도로 근사할 수 있다.



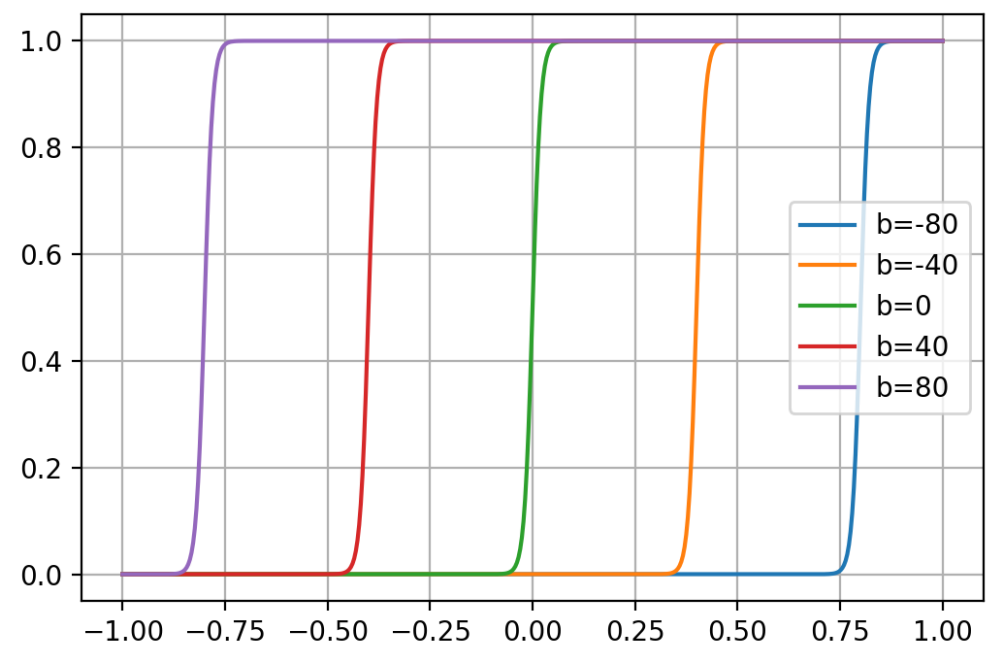
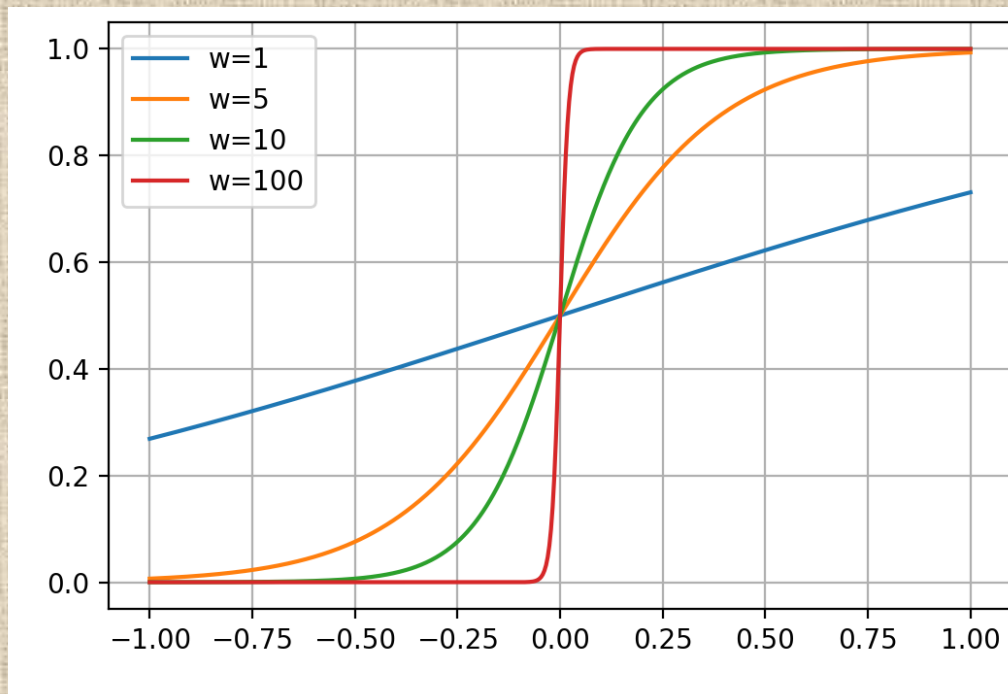
3-2 다층 퍼셉트론

- universal approximator 개념적인 증명
 1. step function을 근사할 수 있다.
 2. step function 둘을 합해서 특정구역에서만 출력을 만들 수 있다(piecewise constant)
 3. 2를 확장하여 임의의 공간(차원)에서 함수를 근사할 수 있다.

3-2 다층 퍼셉트론

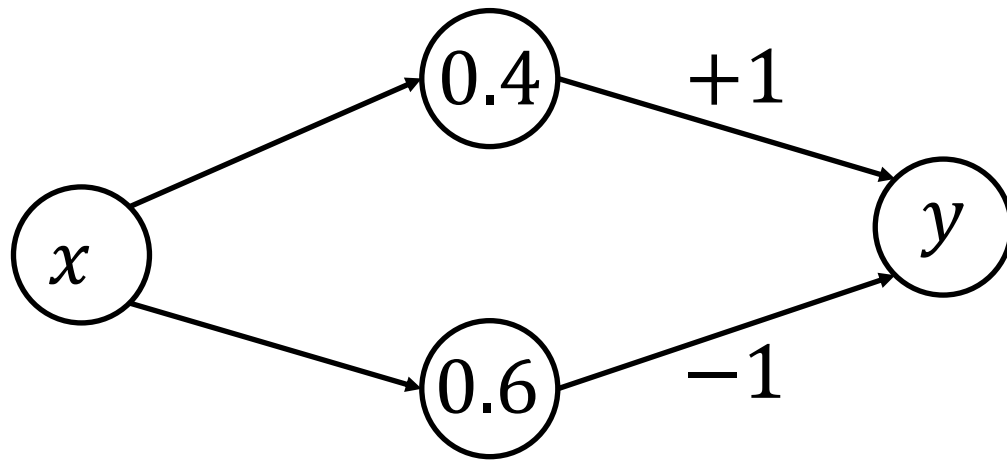
- universal approximator 개념적인 증명(cont.)

$$y = \sigma(wx + b)$$



3-2 다층 퍼셉트론

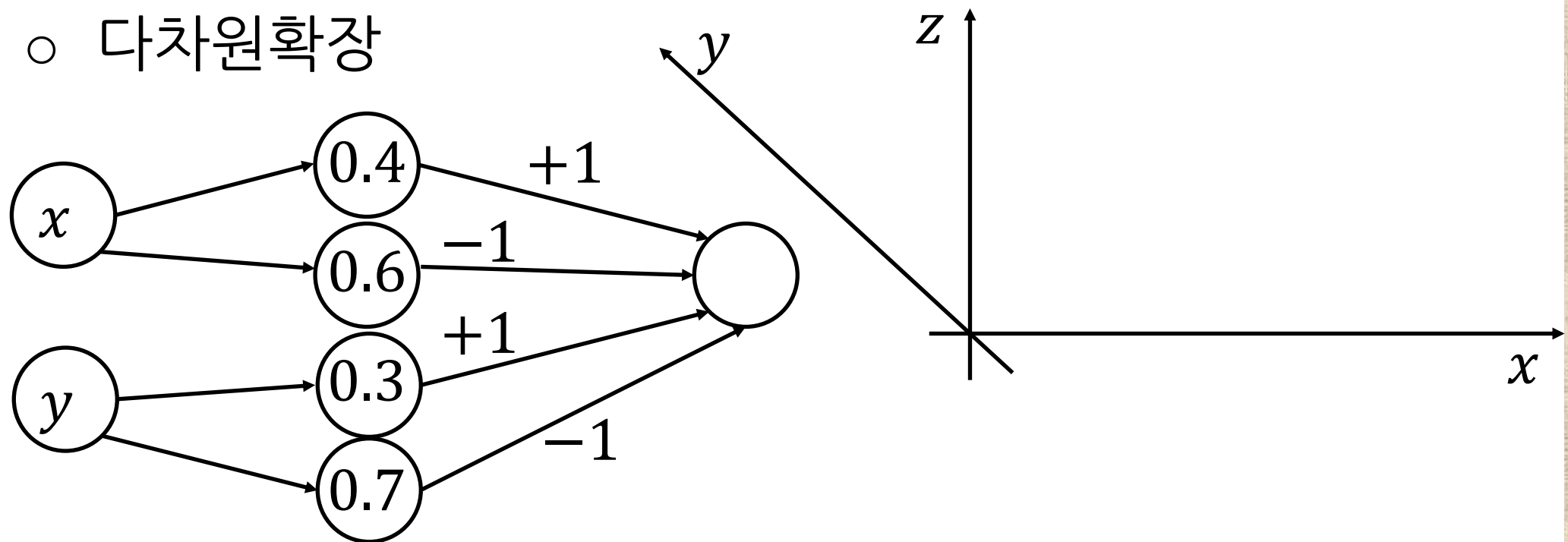
- universal approximator 개념적인 증명(cont.)



3-2 다층 퍼셉트론

- universal approximator 개념적인 증명(cont.)

○ 다차원확장





3-2 다층 퍼셉트론

- 임의의 차원에서 임의의 형태를 근사할 수 있다.
- 매우 많은(무한한) hidden unit 수를 가정해야 한다
- 실제로는 유한한 hidden unit을 사용하지만, 상당히 정확하게 데이터를 모델링 해낸다.
→ 여러 분야에서 기존방법들을 압도하는 성능을 보여줌.



다음시간

5강

- 신경망(2)
 - 네트워크 훈련
 - 뉴럴넷의 정규화