

인공지능

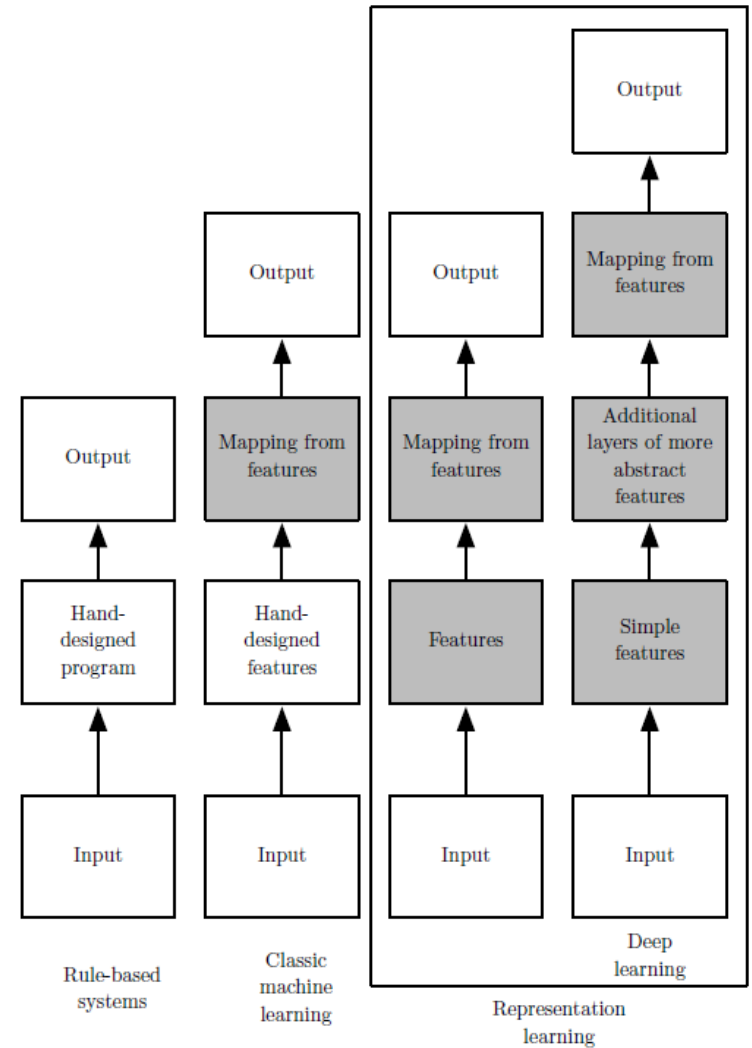
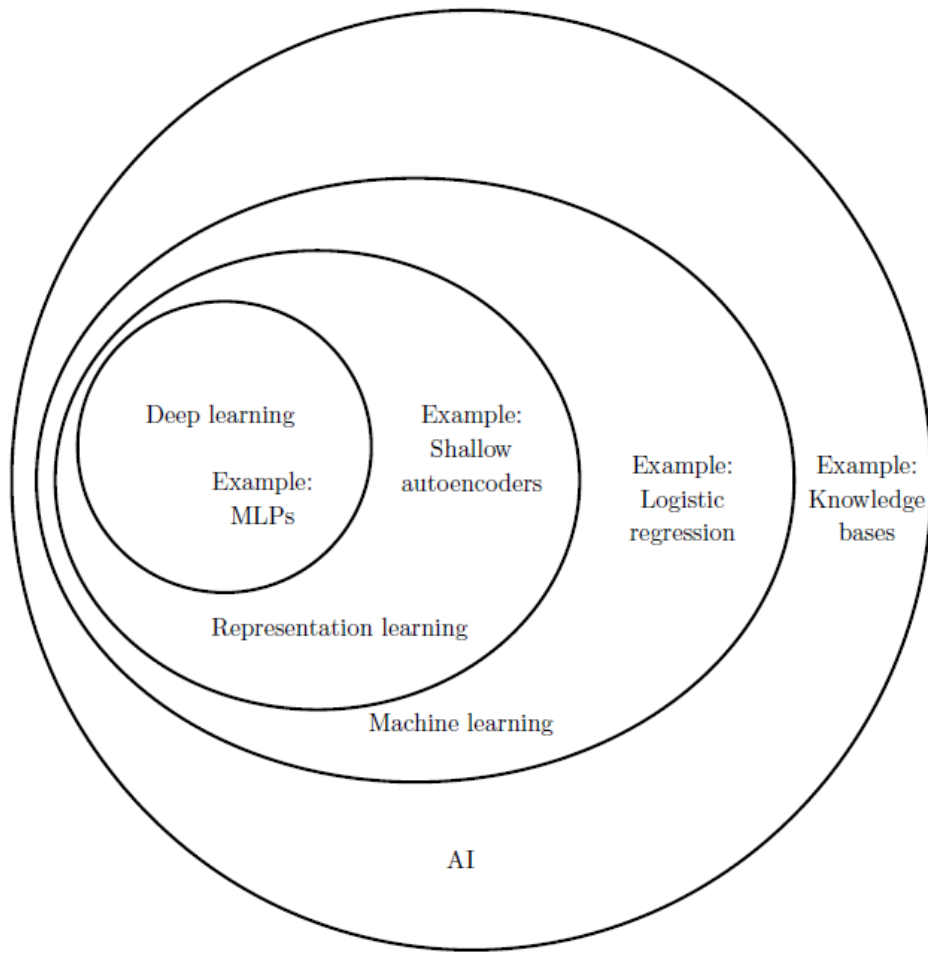
인공신경망

소프트웨어융합대학
소프트웨어학부

본 자료는 해당 수업의 교육 목적으로만 활용될 수 있음.
일부 내용은 다른 교재와 논문으로부터 인용되었으며, 모든 저작권은 원 교재와 논문에 있음.

지난 시간...

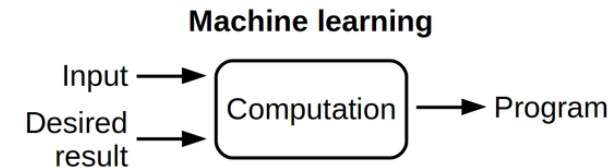
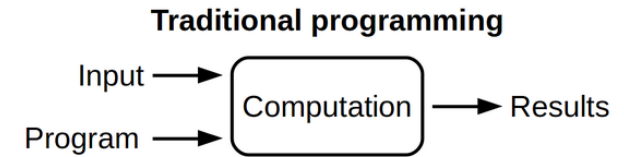
■ 인공지능과 기계학습



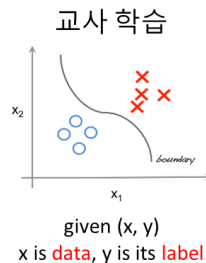
지난 시간...

■ 기계학습

- A computer program
 - improve their **performance P** (accuracy, error rate,...)
 - at some **task T** (classification, regression, detection,...)
 - with **experience E** (data)



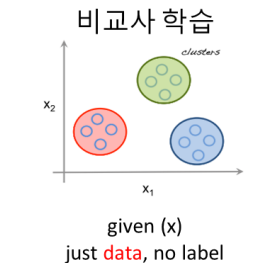
■ 기계학습 문제 분류



Goal:
learn a function
to map $x \rightarrow y$

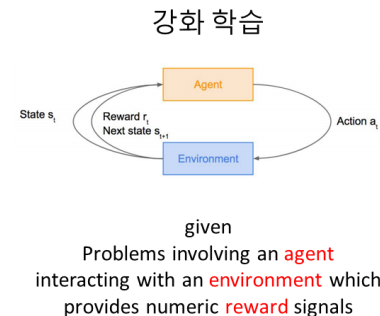
Examples:
Classification
Regression
Object detection
Segmentation
Image captioning

준교사 학습



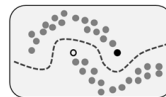
Goal:
learn some underlying hidden
structure of the data

Examples:
Clustering
Dimensionality reduction
Feature learning
Density estimation



Goal:
Learn how to take actions in order to
maximize reward

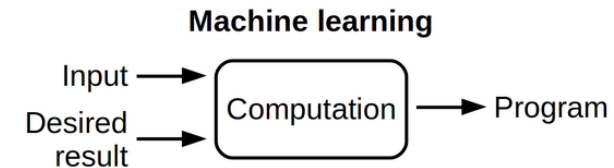
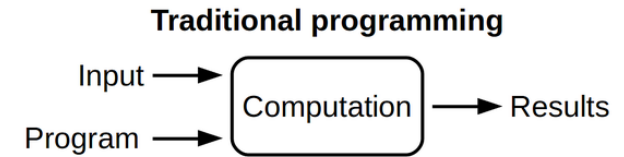
Examples:
Robotics
Self-driving



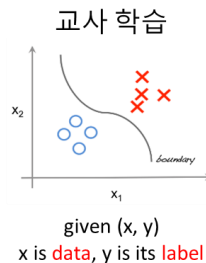
지난 시간...

■ 기계학습

- A computer program
 - improve their **performance P** (accuracy, error rate,...)
 - at some **task T** (classification, regression, detection,...)
 - with **experience E** (data)



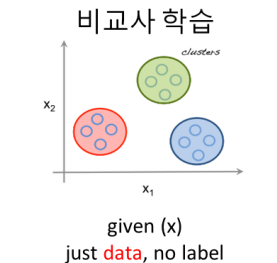
■ 기계학습 문제 분류



Goal:
learn a function
to map $x \rightarrow y$

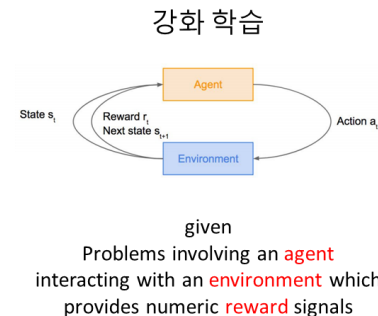
Examples:
Classification
Regression
Object detection
Segmentation
Image captioning

준교사 학습



Goal:
learn some underlying hidden
structure of the data

Examples:
Clustering
Dimensionality reduction
Feature learning
Density estimation



Goal:
Learn how to take actions in order to
maximize reward

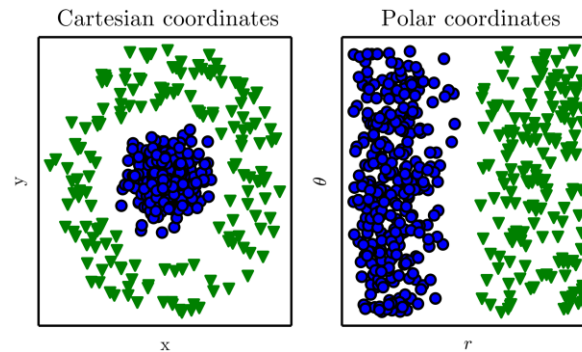
Examples:
Robotics
Self-driving



지난 시간...

■ 표현학습 Representation learning

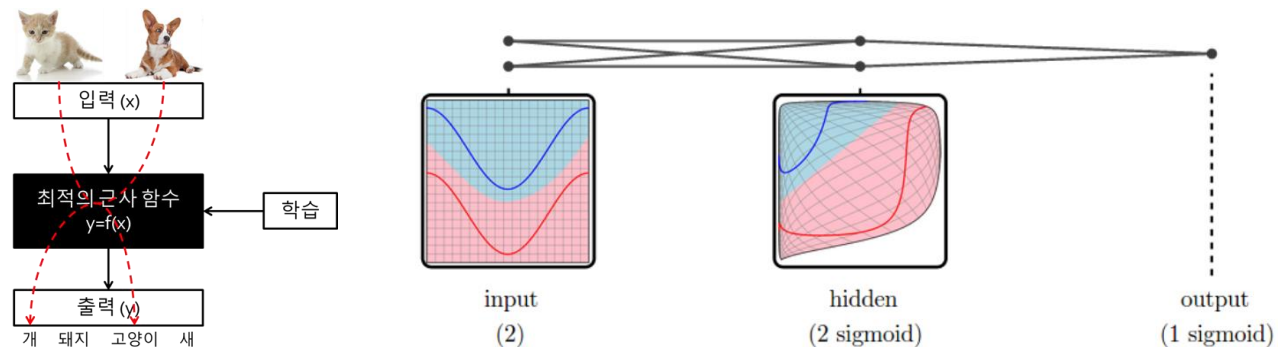
- 기계학습의 파생 방법
- 표현 문제 Representation matter
 - 표현의 차이 비교



- 심층학습 deep learning: 표현학습 representation learning의 주요 방법

: 표현에서 출력으로의 사상mapping뿐만 아니라 표현 자체를 학습하여 보다 좋은 성능을 가짐

- 데이터에서 주어진 작업에 필요한 표현representation을 자동 추출
- 데이터 중심 특징 data-driven feature의 계층적 학습



기계학습 기초

■ 기계학습 문제화

- 사례: 신용 승인 credit approval
 - given: 신청자 정보

feature	value
age	23 years
gender	female
annual salary	\$30,000
years in residence	1 year
years in job	1 year
current debt	\$15,000

- task: 승인? 혹은 거절?



기계학습 기초

■ 표기 정리

component	symbol	credit approval metaphor
input	\mathbf{x}	customer application
output	y	approve or deny
target function	$f: \mathcal{X} \rightarrow \mathcal{Y}$	ideal credit approval formula
data	$(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)$	historical records
hypothesis	$g: \mathcal{X} \rightarrow \mathcal{Y}$	formula to be used

- ▶ f : unknown target function
- ▶ \mathcal{X} : input space (set of all possible inputs \mathbf{x})
- ▶ \mathcal{Y} : output space (set of all possible outputs)
- ▶ N : the number of input-output examples (*i.e.* training examples)
- ▶ $\mathcal{D} \triangleq \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$: data set where $y_n = f(\mathbf{x}_n)$



기계학습 기초

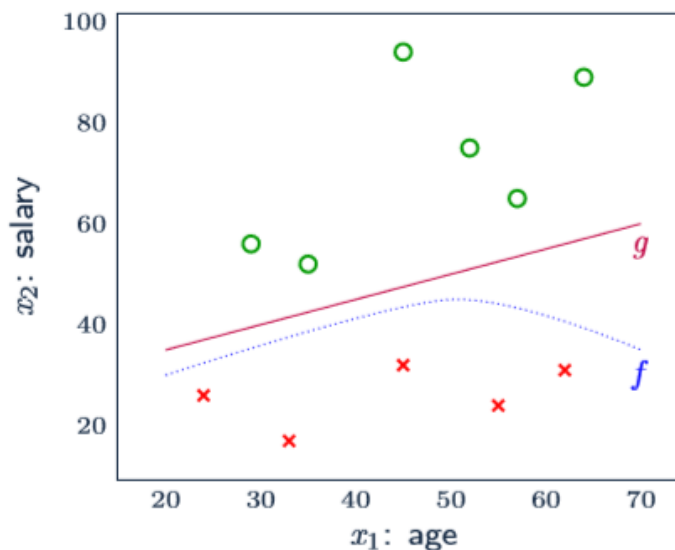
■ 문제 해결

$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ where x_1 : age and x_2 : annual salary in USD

$N = 11$, $d = 2$, $\mathcal{X} = \mathbb{R}^2$, and $\mathcal{Y} = \{\text{approve}, \text{deny}\}$

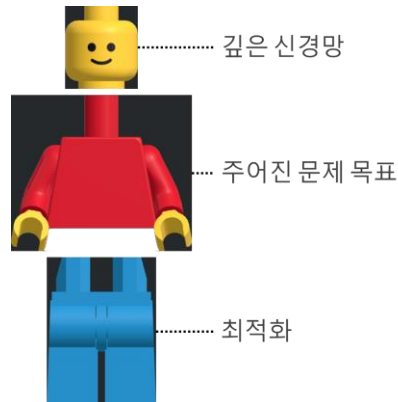
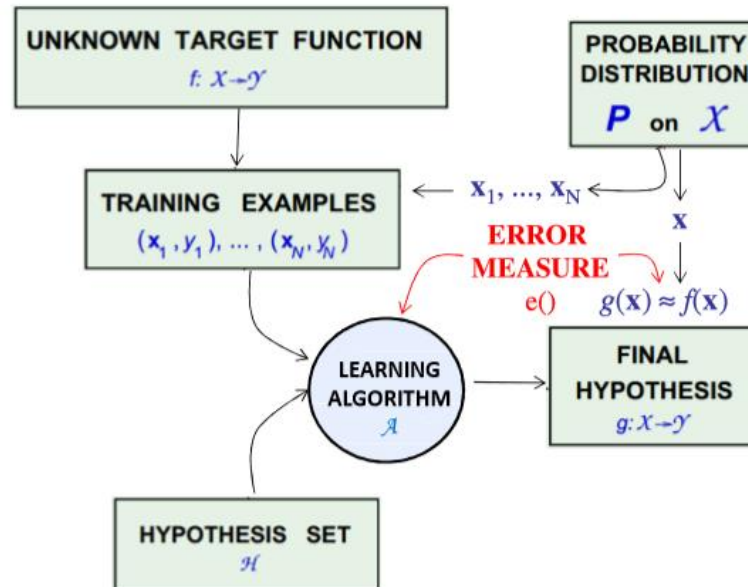
data set \mathcal{D} :

n	x_1	x_2	y
1	29	56k	approve
2	64	89k	approve
3	33	17k	deny
4	45	94k	approve
5	24	26k	deny
6	55	24k	deny
7	35	52k	approve
8	57	65k	approve
9	45	32k	deny
10	52	75k	approve
11	62	31k	deny



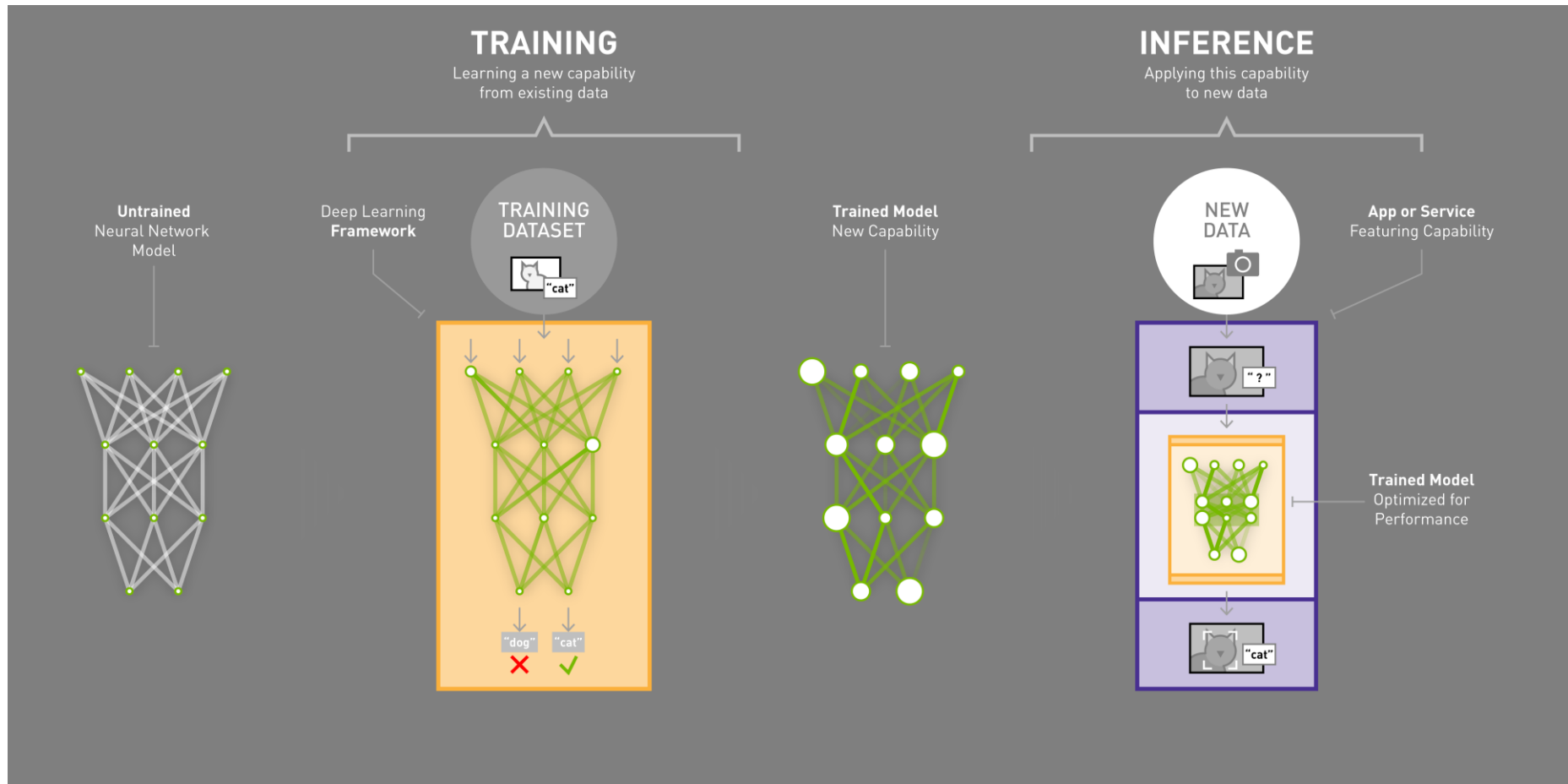
기계학습 기초

■ 기계학습 개요



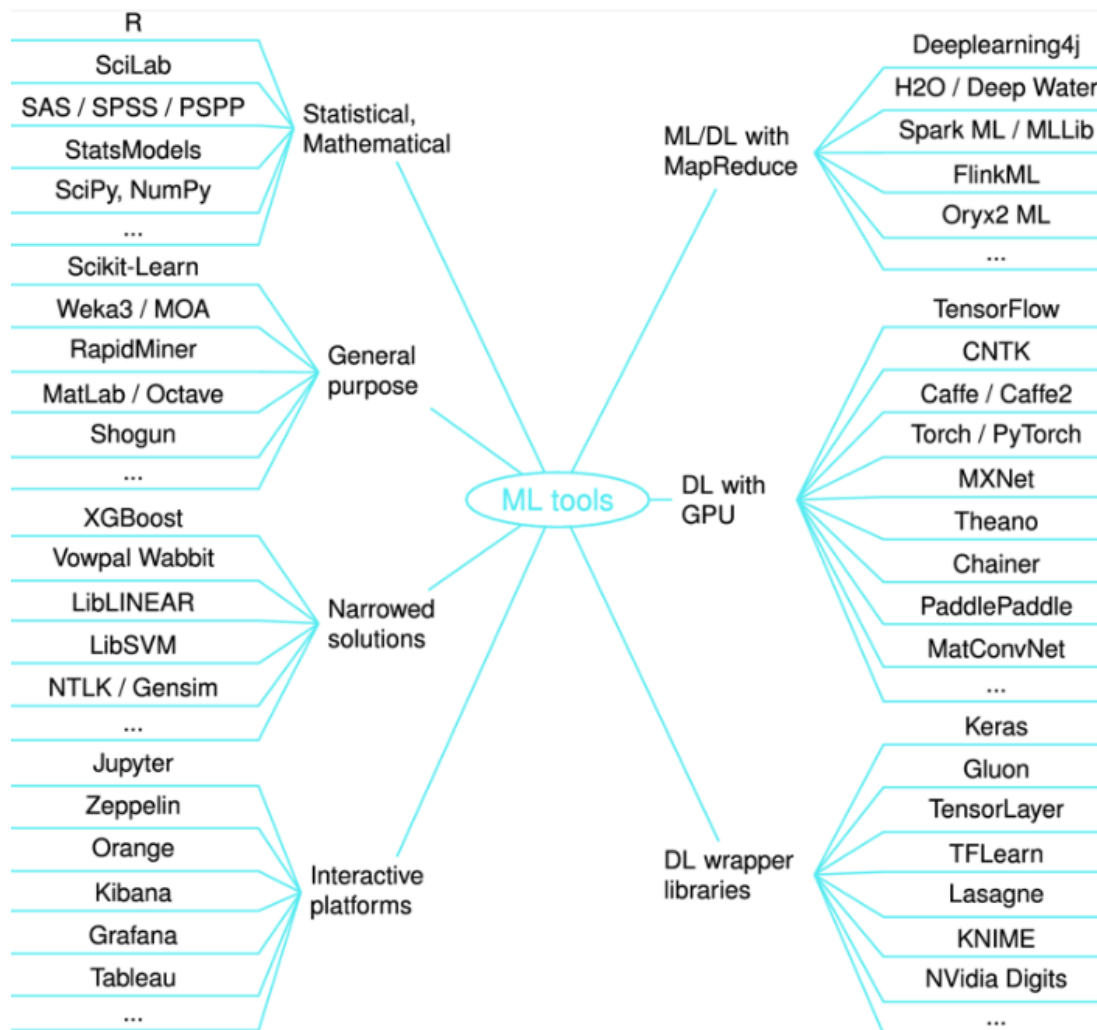
기계학습 기초

■ 기계학습 절차: 훈련training 과 추론inference



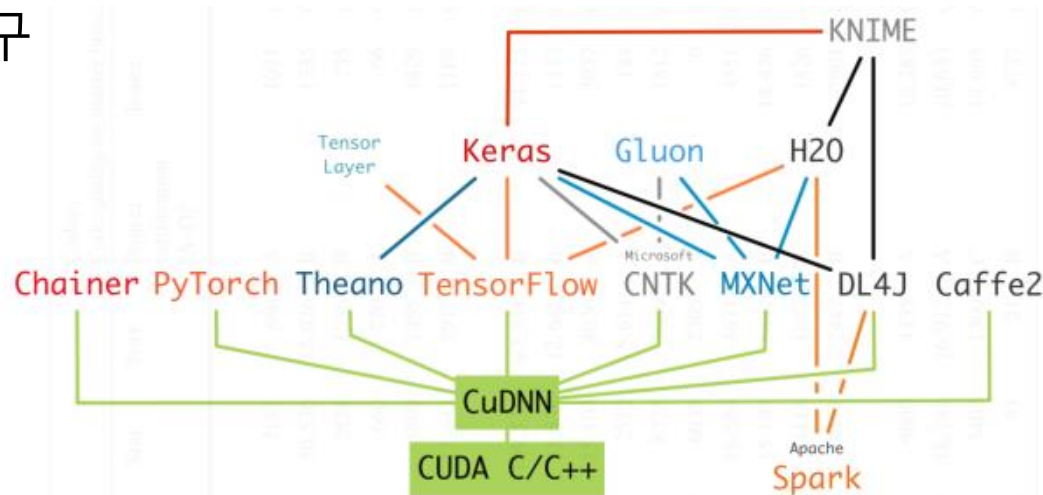
기계학습 기초

■ 기계학습 도구



기계학습 기초

■ 심층학습 도구



Tool	Licence	Written in	Computation graph	Interface	Popularity	Usage	Creator (notes)
TensorFlow (Numerical framework)	Open source, Apache 2.0	C++, Python	Static with small support for dynamic graph	Python, C++ ^a , Java ^a , Go ^a	Very High Growing very fast	Academic Industrial	– Google
Keras (Library)	Open source, MIT	Python	Static	Python Wrapper for TensorFlow, CNTK, DL4J, MXNet, Theano	High Growing very fast	Academic Industrial	F. Chollet
CNTK (Framework)	Open source, Microsoft permissive license	C++	Static	Python, C++, BrainScript, ONNX	Medium Growing fast	Academic Industrial Limited mobile solution	– Microsoft
Caffe (Framework)	Open source, BSD 2-clause	C++	Static	C++, Python, MatLab	High Growing fast	Academic Industrial	Y. Jia BAIR
Caffe2 (Framework)	Open source, Apache 2.0	C++	Static	C++, Python, ONNX	Medium-low Growing fast	Academic Industrial Mobile solution	Y. Jia Facebook
Torch (Framework)	Open source, BSD	C++, Lua	Static	C, C++, LuaJIT, Lua, OpenCL	Medium-low Growing low	Academic Industrial	R. Collobert, K. Kavukcuoglu, C. Farabet
PyTorch (Library)	Open source, BSD	Python, C	Dynamic	Python, ONNX	Medium Growing very fast	Academic Industrial	A. Paszke, S. Gross, S. Chintala, G. Chanan
MXNet (Framework)	Open source, Apache 2.0	C++	Dynamic dependency scheduler	C++, Python, Julia, MatLab, Go, R, Scala, Perl, ONNX	Medium Growing fast	Academic Industrial	– Apache
Chainer (Framework)	Open source, Owners permissive license	Python	Dynamic	Python	Low Growing low	Academic Industrial	– Preferred Networks
Theano (Numerical framework)	Open source, BSD	Python	Static	Python	Medium-low Growing low	Academic Industrial	Y. Bengio University of Montreal

이 장에서는...

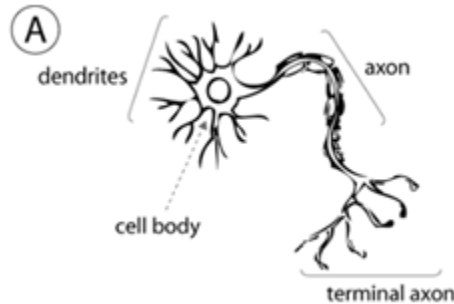
■ 깊은 인공신경망의 해석을 위한 배경

- 인공신경망artificial neural network의 변천사와 종류
- 인공신경망의 핵심 요소인 퍼셉트론perceptron 구조와 동작
- 퍼셉트론의 한계와 다층 퍼셉트론multilayer perceptron
- 다층 퍼셉트론의 구조와 동작, 학습
- 깊은 인공신경망과 심층학습deep learning

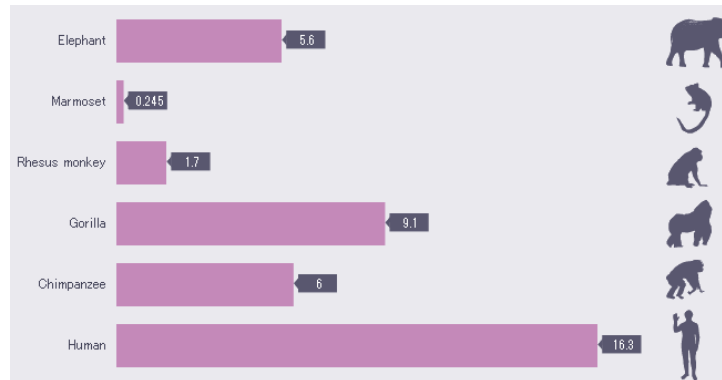
인공신경망의 변천사와 종류

■ 사람의 뉴런neuron

- 두뇌의 가장 작은 정보처리 단위
- 세포체cell body는 간단한 연산, 수상돌기dendrite는 신호 수신, 축삭axon은 처리 결과를 전송



- 사람은 10^{11} 개 정도의 뉴런을 가지며,
각 뉴런은 1000여개의 다른 뉴런과 연결되어 10^{14} 개 정도 복잡한 연결 구조를 가짐



뉴런의 개수 비교

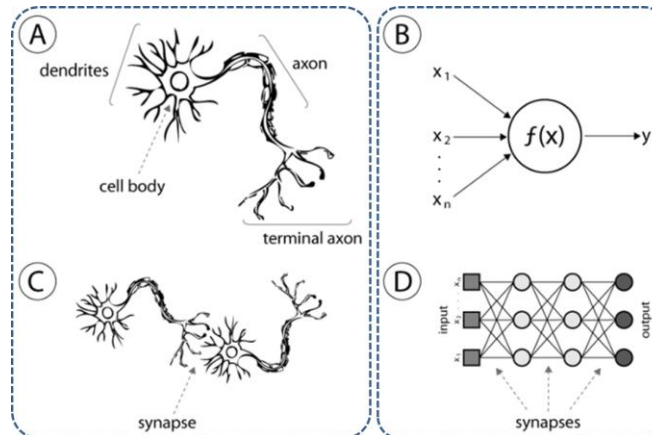
인공신경망의 변천사와 종류

■ 두 줄기 연구의 시너지|synergy 효과

- 컴퓨터 과학
 - 계산(연산) 능력의 획기적 발전으로 지능 처리에 대한 욕구 확대
- 뇌(의학) 과학
 - 뇌의 정보처리 방식 연구

→ 뇌의 정보처리 모방하여 사람처럼 지능적 행위를 할 수 있는 인공지능 도전

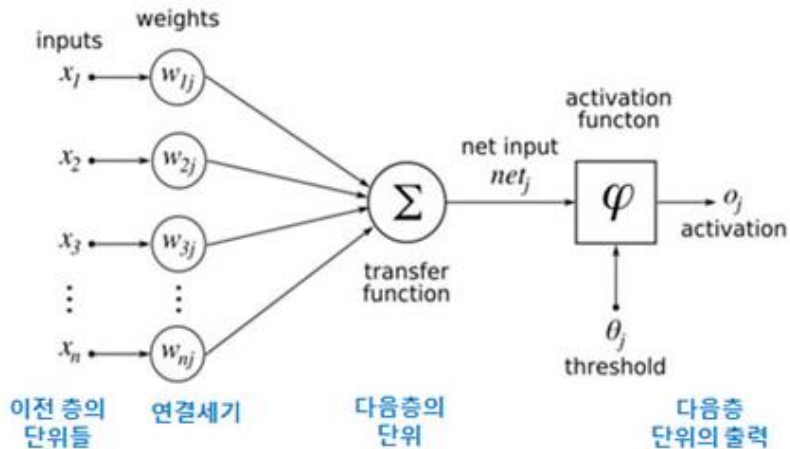
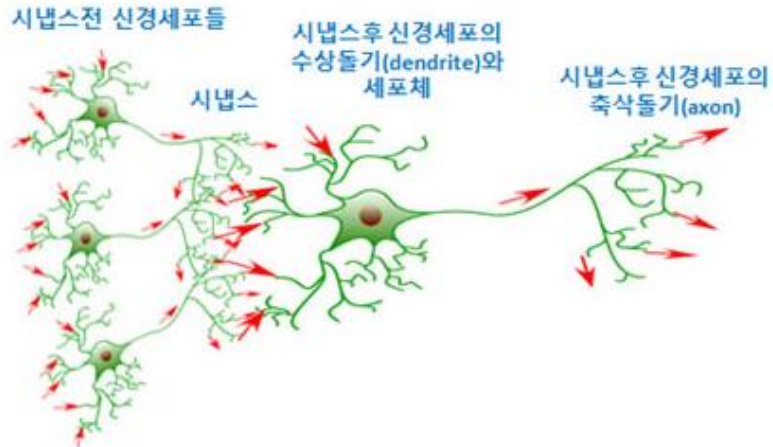
- 뉴런의 동작 이해를 모방한 인공 신경망|artificial neural networks (ANN) 연구 수행됨
- 페셉트론 고안



<http://www.intechopen.com/source/html/39067/media/image1.png>

인공신경망의 변천사와 종류

■ 사람의 신경망과 인공신경망 비교

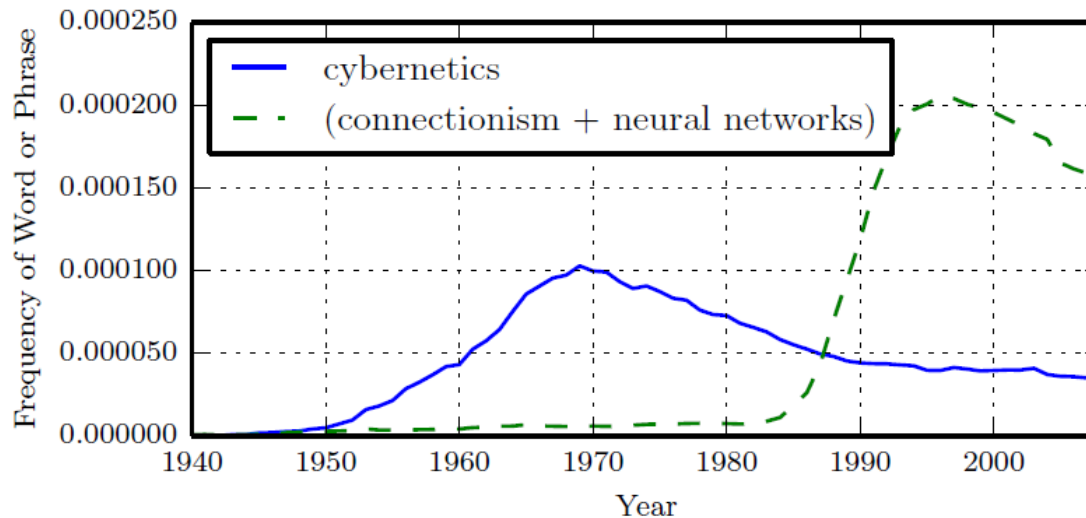


사람의 신경망	인공신경망
세포체	노드
수상돌기	입력
축삭	출력
시냅스	가중치

인공신경망의 변천사와 종류

■ 인공신경망의 역사

- 1940-1960: cybernetics
- 1980-1990: connectionism (or parallel distributed processing)
- 2006-today: deep learning

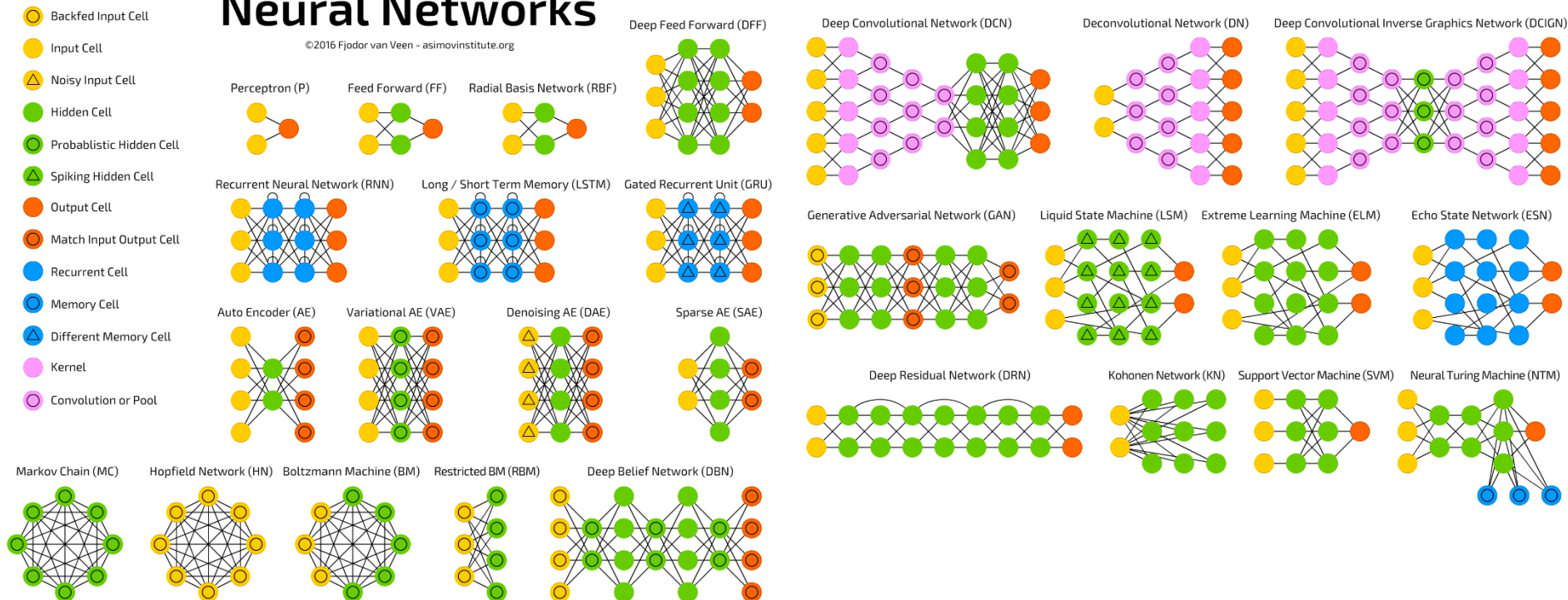


인공신경망의 변천사와 종류

■ 현대 인공신경망은 다양한 파생 모델들이 존재

A mostly complete chart of Neural Networks

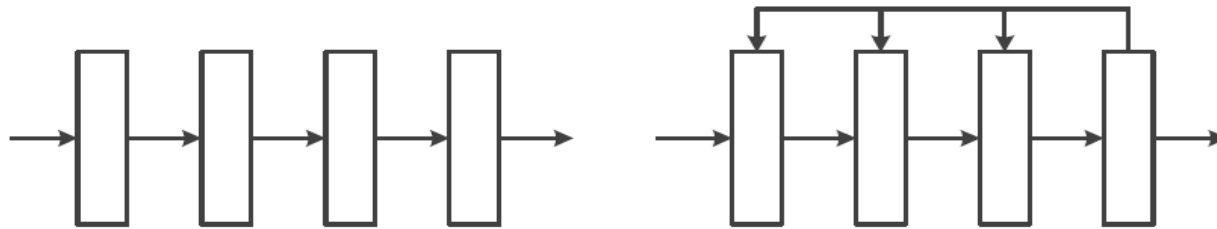
©2016 Fjodor van Veen - asimovinstitute.org



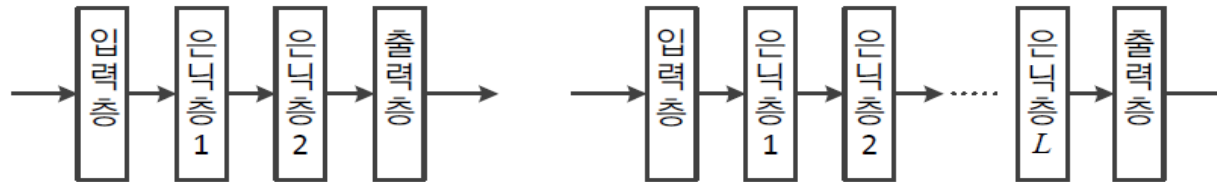
인공신경망의 변천사와 종류

■ 대표적인 인공신경망 종류

- 전방^{forward} 인공신경망과 순환^{recurrent} 인공신경망
- 얇은^{shallow} 인공신경망과 깊은^{deep} 인공신경망



(a) 전방 신경망과 순환 신경망



(b) 얇은 신경망과 깊은 신경망

퍼셉트론 구조와 동작

■ 퍼셉트론은

- 연산 **노드** node, **가중치** weight, **층** layer과 같은 새로운 개념을 도입
- 선형+비선형 연산으로 구성됨
- **학습 알고리즘**을 제안

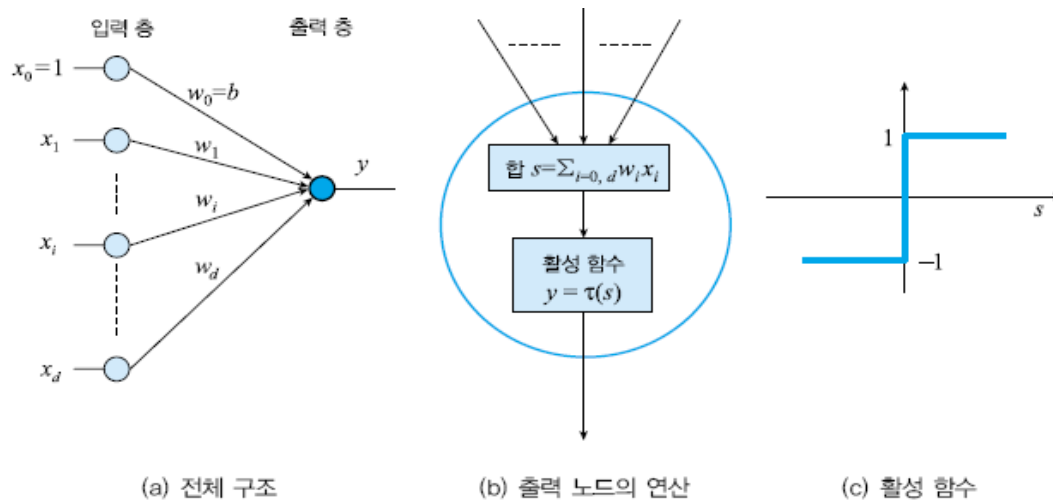
■ 퍼셉트론은

- 원시적 인공신경망이지만,
 - 현대 인공신경망은 퍼셉트론을 토대로 다양한 다층 구조로 결합함
- 깊은 인공신경망의 중요한 구성 요소: 심층 학습 deep learning

퍼셉트론 구조와 동작

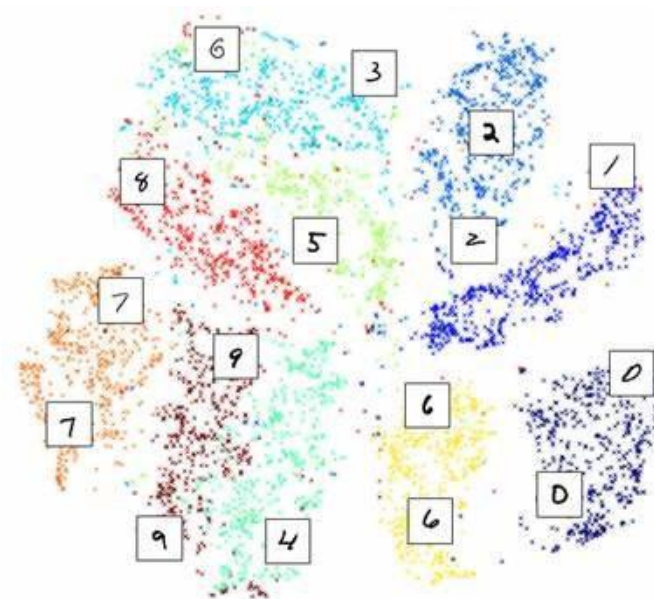
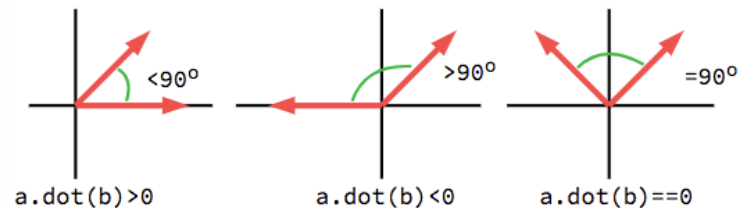
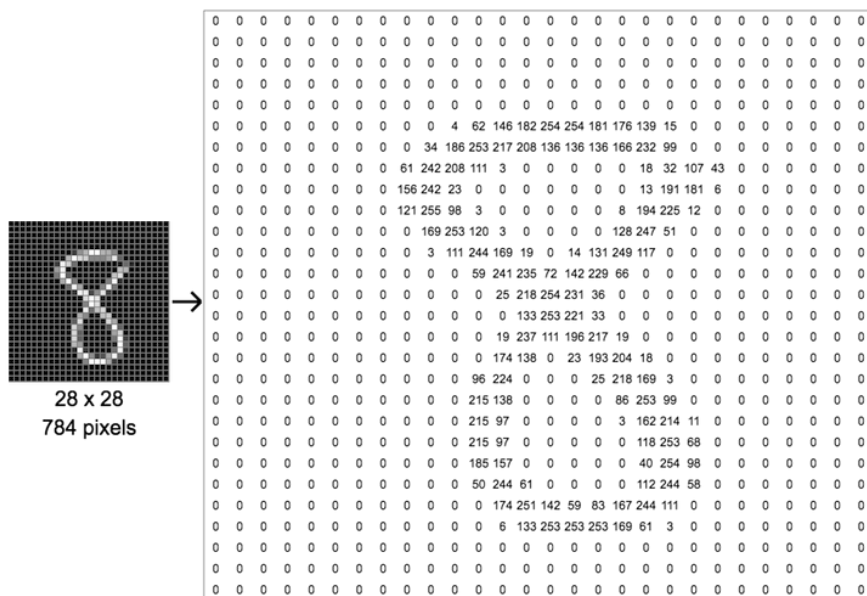
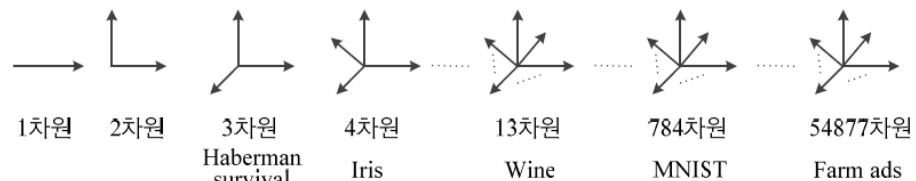
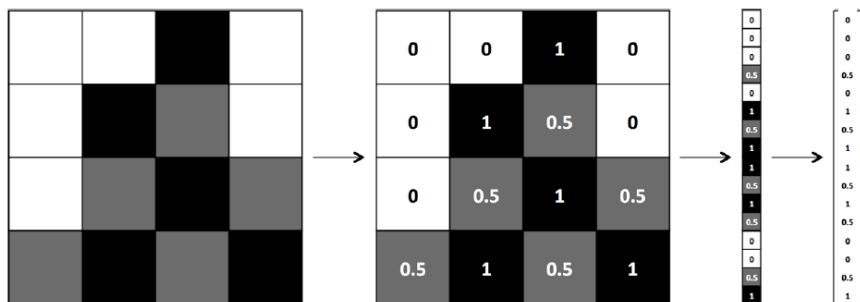
■ 퍼셉트론의 구조

- 입력층과 출력층을 가짐
 - 입력층은 연산을 하지 않으므로 퍼셉트론은 **단일층 구조**라고 간주
- **입력층**의 i 번째 노드는 (입력) 특징 벡터 $\mathbf{x} = (x_1, x_2, \dots, x_d)^T$ 의 요소 x_i 를 담당
 - 항상 1이 입력되는 편향^{bias} 노드 포함
- **출력층**은 한 개의 노드
- i 번째 입력 노드와 출력 노드를 연결하는 변^{edge}은 가중치 w_i 를 가짐



선형대수

데이터와 특징공간



퍼셉트론 구조와 동작

■ 퍼셉트론의 동작

1. [내적^{inner product}; 선형연산] 입력(특징)값과 가중치를 곱하고, 합한 s 를 구함
2. [비선형연산] 비선형 활성화함수 τ 를 적용

→ 활성화함수 τ 로 계단함수^{step function}를 사용하여 최종 출력 y 는 +1 또는 -1 (이진 분류)

$$\left. \begin{aligned} y &= \tau(s) \\ \text{이때 } s &= w_0 + \sum_{i=1}^d w_i x_i, \quad \tau(s) = \begin{cases} 1 & s \geq 0 \\ -1 & s < 0 \end{cases} \end{aligned} \right\}$$

퍼셉트론 구조와 동작

■ 행렬 표기 | matrix vector notation

$$s = \mathbf{w}^T \mathbf{x} + w_0, \quad \text{여기서 } \mathbf{x} = (x_1, x_2, \dots, x_d)^T, \mathbf{w} = (w_1, w_2, \dots, w_d)^T$$

- 바이어스 항을 벡터에 추가하면,

$$s = \mathbf{w}^T \mathbf{x}, \quad \text{여기서 } \mathbf{x} = (1, x_1, x_2, \dots, x_d)^T, \mathbf{w} = (w_0, w_1, w_2, \dots, w_d)^T$$

- 퍼셉트론의 동작을 식 (3.4)로 표현할 수 있음

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

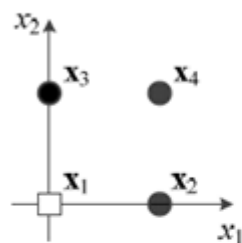
퍼셉트론 구조와 동작

■ 퍼셉트론의 동작 예

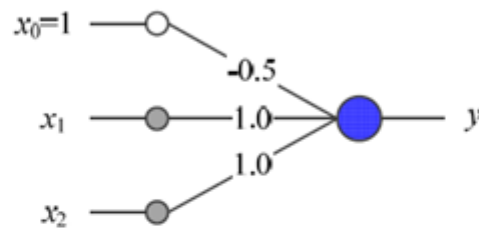
2차원 특징 벡터로 표현되는 샘플을 4개 가진 훈련집합 $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4\}$, $\mathcal{Y} = \{y_1, y_2, y_3, y_4\}$ 를 생각하자.

[그림 (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

OR 논리 게이트를 이용한 퍼셉트론의 동작 예시

샘플 4개를 하나씩 입력하여 제대로 분류하는지 확인해 보자.

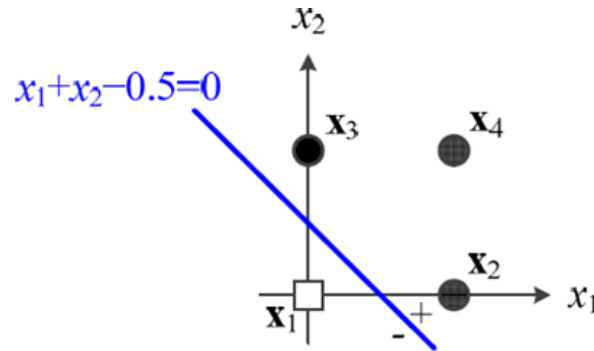
$$\begin{aligned} \mathbf{x}_1: s &= -0.5 + 0 * 1.0 + 0 * 1.0 = -0.5, & \tau(-0.5) &= -1 \\ \mathbf{x}_2: s &= -0.5 + 1 * 1.0 + 0 * 1.0 = 0.5, & \tau(0.5) &= 1 \\ \mathbf{x}_3: s &= -0.5 + 0 * 1.0 + 1 * 1.0 = 0.5, & \tau(0.5) &= 1 \\ \mathbf{x}_4: s &= -0.5 + 1 * 1.0 + 1 * 1.0 = 1.5, & \tau(1.5) &= 1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

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

퍼셉트론 구조와 동작

■ 기하학적 관점에서 살펴본 퍼셉트론의 동작

- 결정 직선 $d(\mathbf{x}) = d(x_1, x_2) = w_1x_1 + w_2x_2 + w_0 = 0 \rightarrow x_1 + x_2 - 0.5 = 0$
 - w_1 과 w_2 는 직선의 방향, w_0 은 절편^{intercept}을 결정
 - 결정 직선은 전체 공간을 +1과 -1의 두 부분공간으로 분할하는 이진 분류기 역할



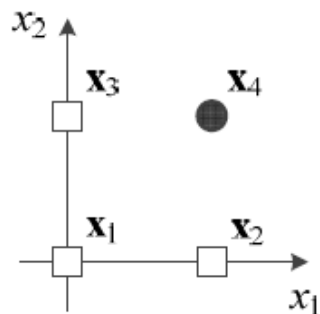
퍼셉트론에 해당하는 결정 직선

- d 차원 공간의 경우, $d(\mathbf{x}) = w_1x_1 + w_2x_2 + \dots + w_dx_d + w_0 = 0$
 - 2차원은 결정 직선^{decision line}, 3차원은 결정 평면^{decision plane},
 - 4차원 이상은 결정 초평면^{decision hyperplane}

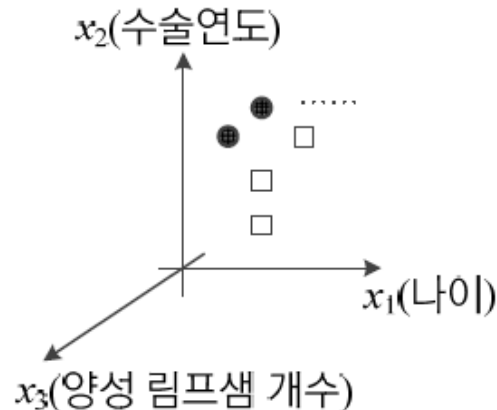
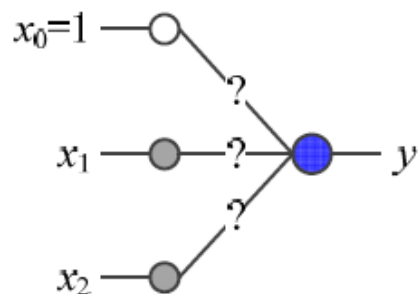
퍼셉트론 구조와 동작

■ 퍼셉트론의 학습 문제 (훈련)

- 지금까지는 **학습된 퍼셉트론의 동작**을 설명
- **학습 문제**: w_1 과 w_2 , w_0 이 어떤 값을 가져야 올바르게 분류할까?
 - 예시에서는 2차원 공간에 4개 샘플 데이터가 있는 훈련집합
 - 현실 세계는 d 차원 공간에 수백~수만 개의 샘플 데이터가 존재
 - MNIST는 784차원에 6만개 데이터 샘플 존재

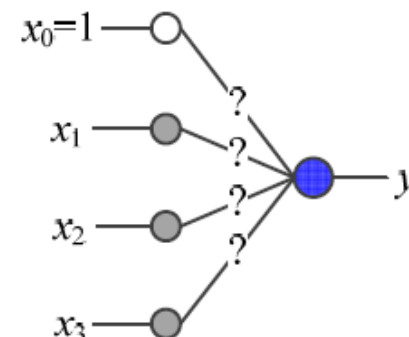


(a) AND 분류 문제



(b) Haberman survival 분류 문제

UCI 데이터 (유방암 수술 생존 관련 데이터)



퍼셉트론 구조와 동작

■ 학습 문제

■ 일반적인 분류기의 학습 과정

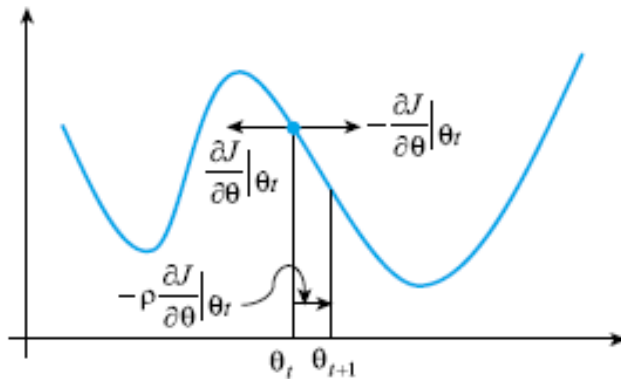
- 단계1: 문제^{task} 정의 (예, 분류, 회기)
- 단계2: 목적함수 $J(\theta)$ 정의 (학습이 잘 진행되고 있는지를 정량적으로 판단하는 기준)
 - 퍼셉트론의 매개변수를 $\mathbf{w} = (w_0, w_1, w_2, \dots, w_d)^T$ 라 표기하면, 매개변수 집합은 $\theta = \{\mathbf{w}\}$
- 단계3: $J(\theta)$ 를 최소화하는 θ 를 찾기 위한 반복 최적화 방법 수행
 - 일반적으로 반복적인 최적화 접근^{iterative optimization}인 경사 하강법^{gradient descent} 활용

퍼셉트론 구조와 동작

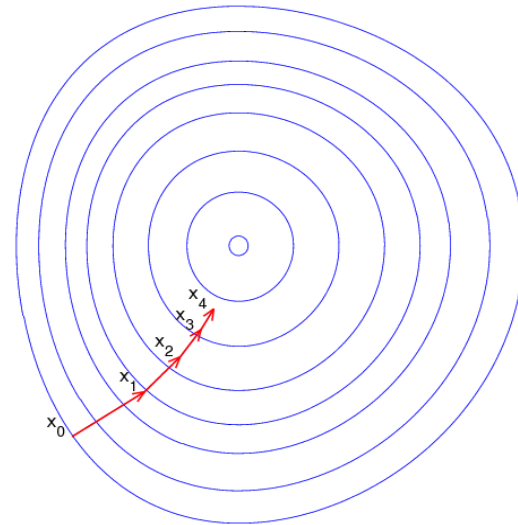
■ (3단계) 경사 하강법

- 최소 $J(\theta)$ 값을 얻기 위해
학습 모델의 현재 매개변수 **기울기 (경사)**를 구하고
반대 방향으로 이동(가중치 갱신)하는 **반복 탐색**을 통해 **극값**을 찾음

1차원 공간에서의 예

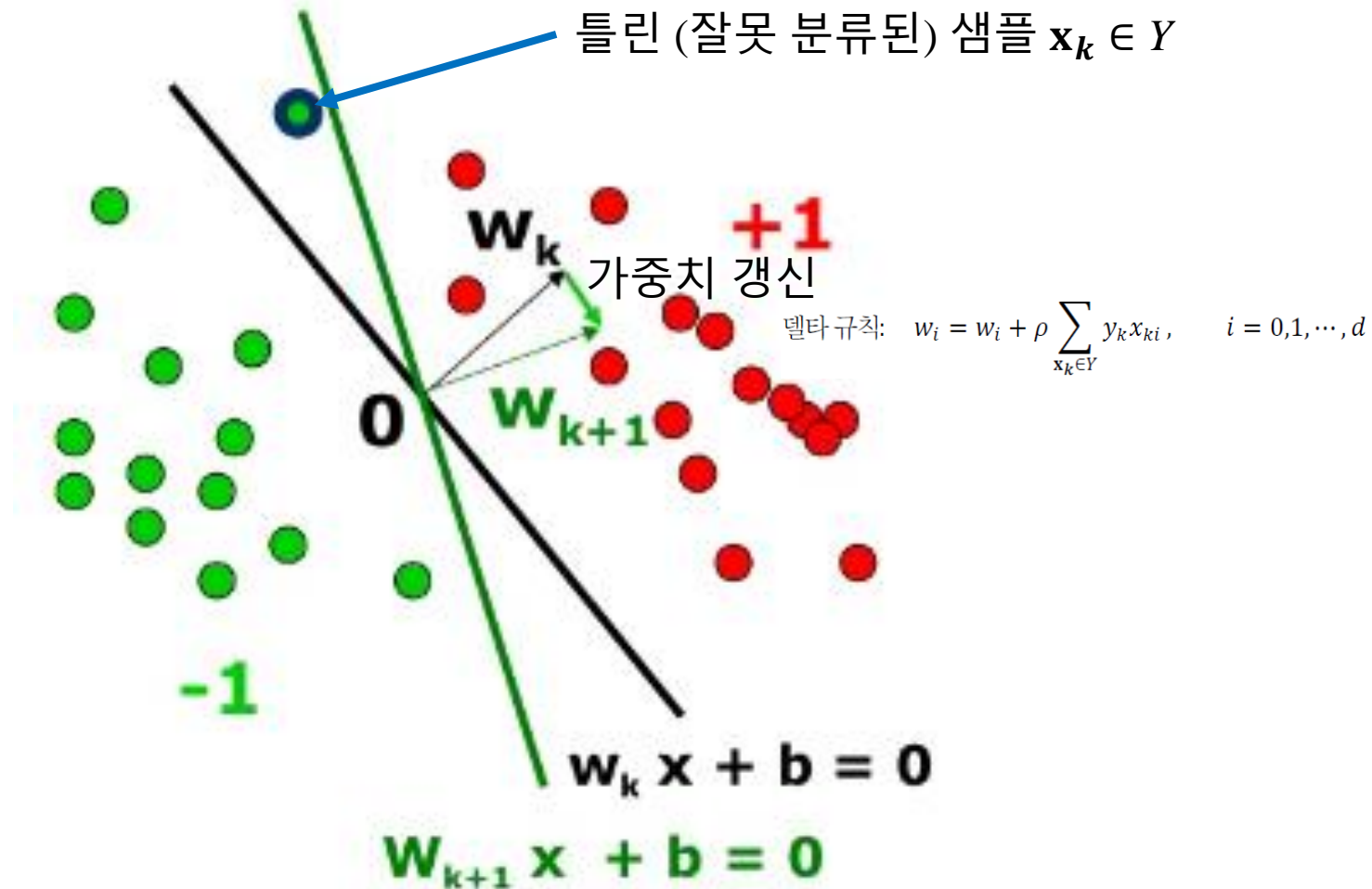


2차원 공간에서의 예



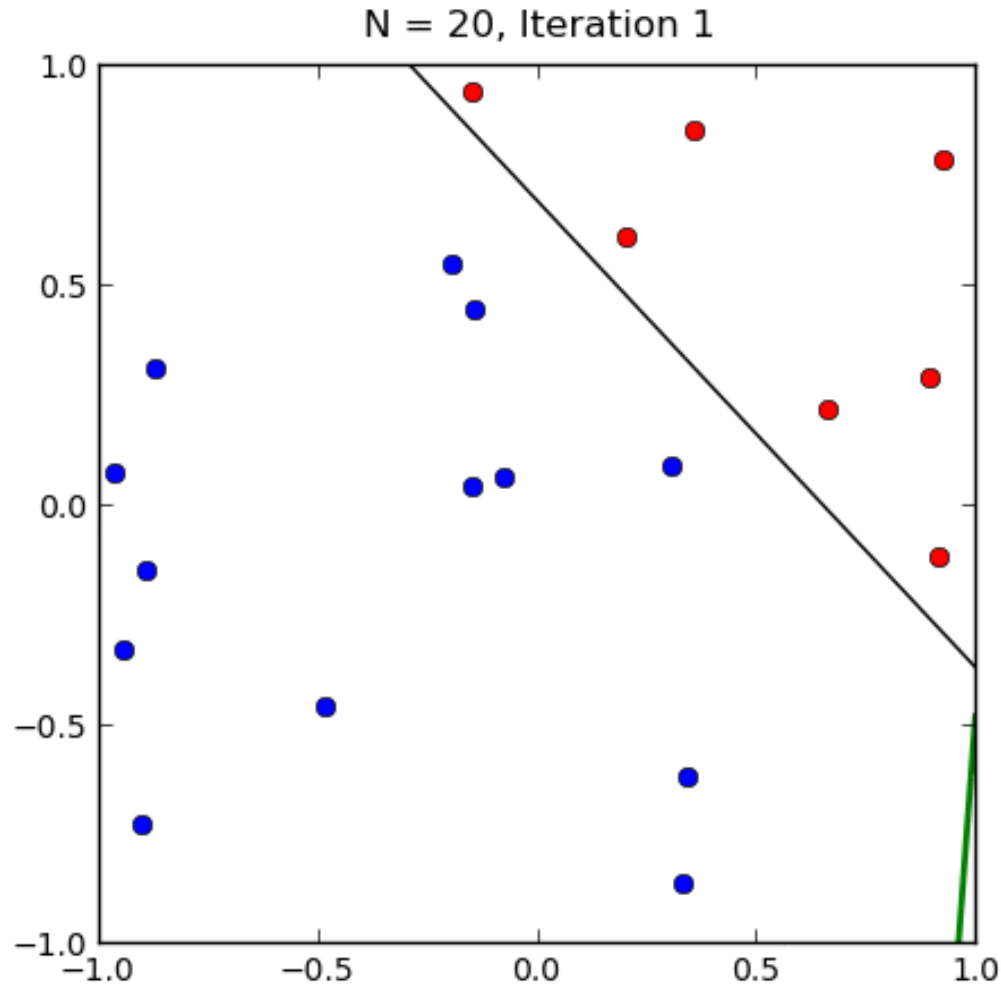
퍼셉트론 구조와 동작

■ 퍼셉트론 학습 알고리즘 **동작** (델타^{delta} 규칙)



퍼셉트론 구조와 동작

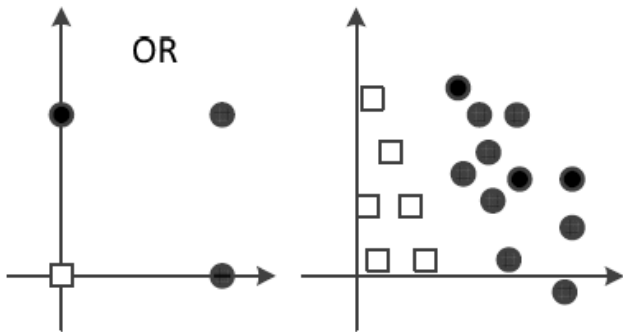
■ 퍼셉트론 학습 알고리즘 동작 예



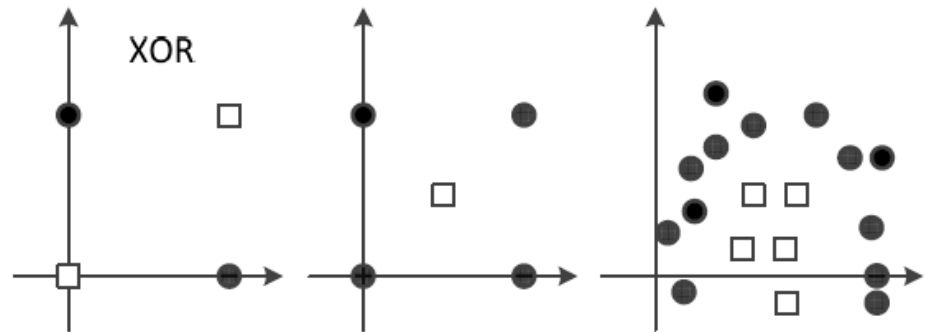
다층 퍼셉트론

■ 퍼셉트론은 선형 분류기(linear classifier)라는 **한계**

- 선형 분리 불가능한 상황에서는 일정한 양의 오류
- 예) XOR 문제에서는 75%가 정확도 한계



(a) 선형분리 가능



(b) 선형분리 불가능

- 1969년 퍼셉트론의 한계를 지적
- 1986년 **다층 퍼셉트론** 이론 정립 인공지능망 부활

다층 퍼셉트론

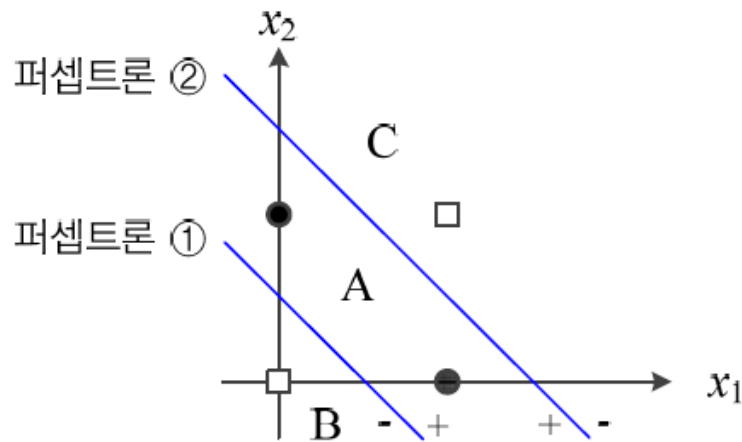
■ 다층 퍼셉트론의 핵심

- 여러 개의 퍼셉트론들을 병렬 구조의 층으로 구성하여 은닉층을 둠
→ 은닉층은 원래 특징 공간을 문제를 해결에 유리한 새로운 공간으로 변환함 (표현학습)
== 데이터로부터 자동 특징 추출 extracting data-driven feature
- 시그모이드 sigmoid 활성화함수 도입 (출력 연속적임)
퍼셉트론: 계단함수 (경성^{hard} 혹은 이산적인 의사결정) → 다층 퍼셉트론 (연성^{soft} 의사결정)
- 오류 역전파 backpropagation 알고리즘 사용
 - 다층 퍼셉트론 여러 개의 층이 순차적으로 연결된 구조
 - 학습을 위해 출력부터 입력까지 역방향으로 기울기 미분을 계산하여 해당 가중치 갱신 (경사 하강 최적화 gradient descent optimization)

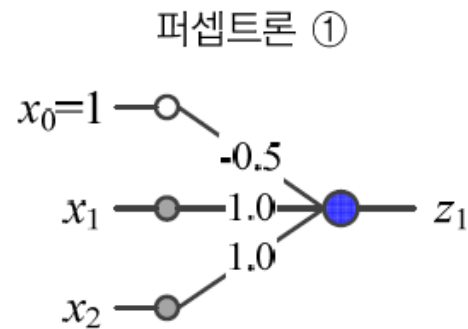
다층 퍼셉트론

■ 특징 공간 변환: 퍼셉트론 2개를 사용한 XOR 문제의 해결

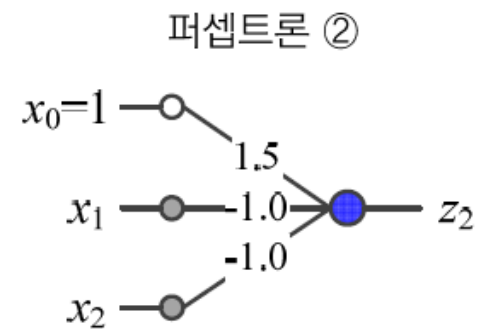
- 퍼셉트론①과 퍼셉트론②가 모두 +1이면 ● 부류이고 그렇지 않으면 □ 부류임



(a) 퍼셉트론 2개를 이용한 공간분할



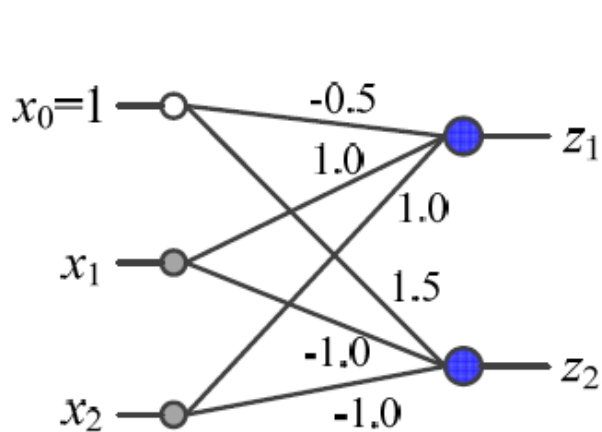
(b) 퍼셉트론 2개



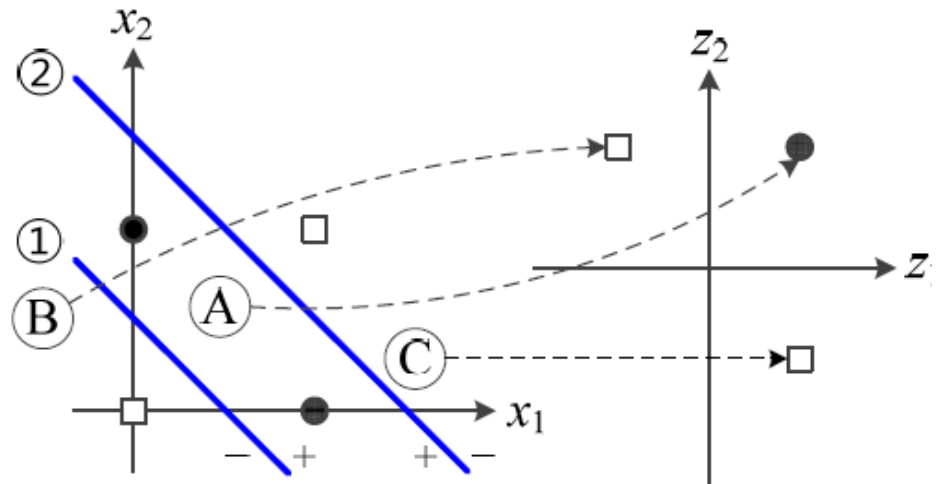
다층 퍼셉트론

■ 퍼셉트론 2개를 병렬 결합하면,

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



(a) 두 퍼셉트론을 병렬로 결합



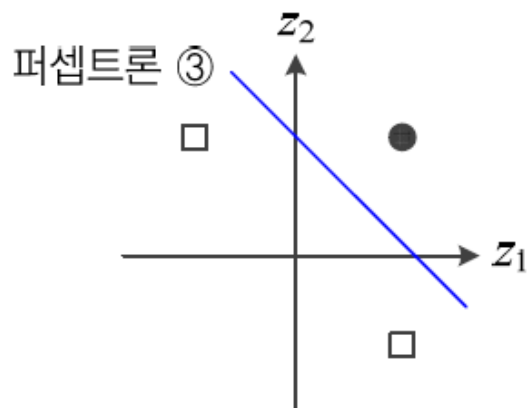
(b) 원래 특징 공간 \mathbf{x} 를 새로운 특징 공간 \mathbf{z} 로 변환

■ 각 퍼셉트론은

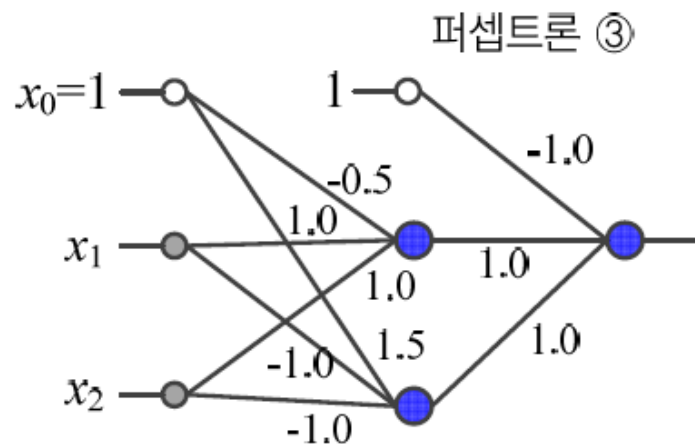
입력과 가중치의 내적을 통해 유사성을 정량화하고 분할하여 특징 분별하는 필터 역할
→ 여러 개의 퍼셉트론을 층으로 구성하여 입력의 다양한 특징들을 자동 추출함

다층 퍼셉트론

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



(a) 새로운 특징 공간에서 분할

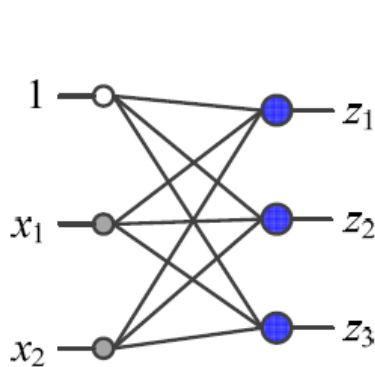


(b) 퍼셉트론 3개를 결합한 다층 퍼셉트론

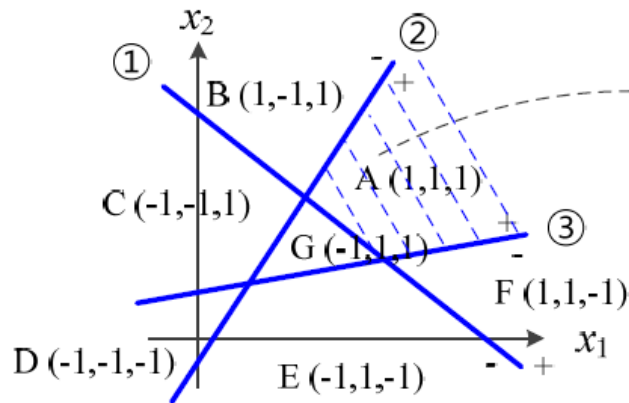
다층 퍼셉트론

■ 다층 퍼셉트론의 (은닉)층 의미

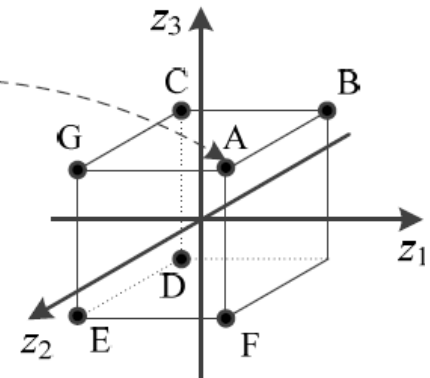
- [그림]처럼 3개 퍼셉트론을 결합하면,
2차원 공간을 7개 영역으로 나누고 각 영역을 3차원 점으로 변환
- ← 활성화함수 τ 로 계단함수를 사용하므로 영역을 점으로 변환
(시그모이드 함수의 경우, 연속적인 새로운 공간으로 변환)



(a) 퍼셉트론 3개를 결합



(b) 7개 부분공간으로 나눔


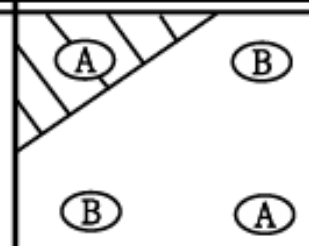
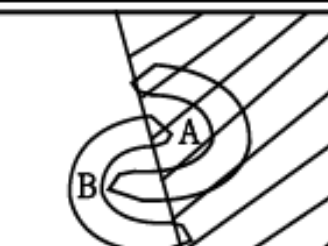
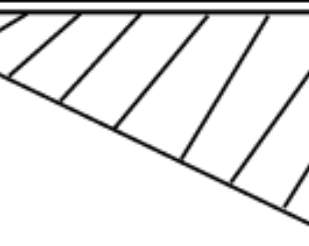

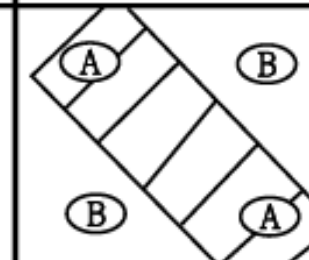
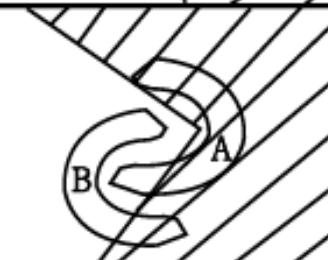
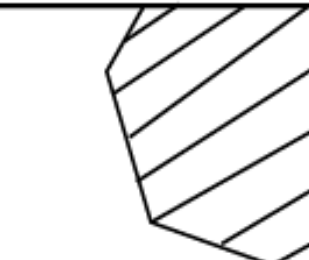
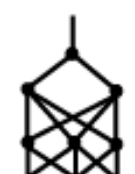
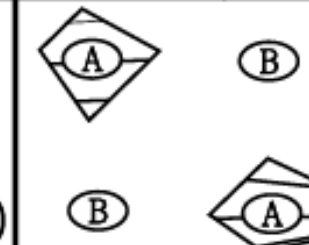

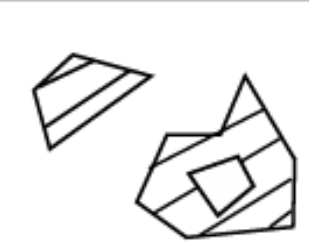


(c) 3차원 공간의 점으로 매핑

- 일반화하여, p 개 퍼셉트론을 결합하면 p 차원 공간으로 변환

다층 퍼셉트론

■ 다층 퍼셉트론의 용량capacity

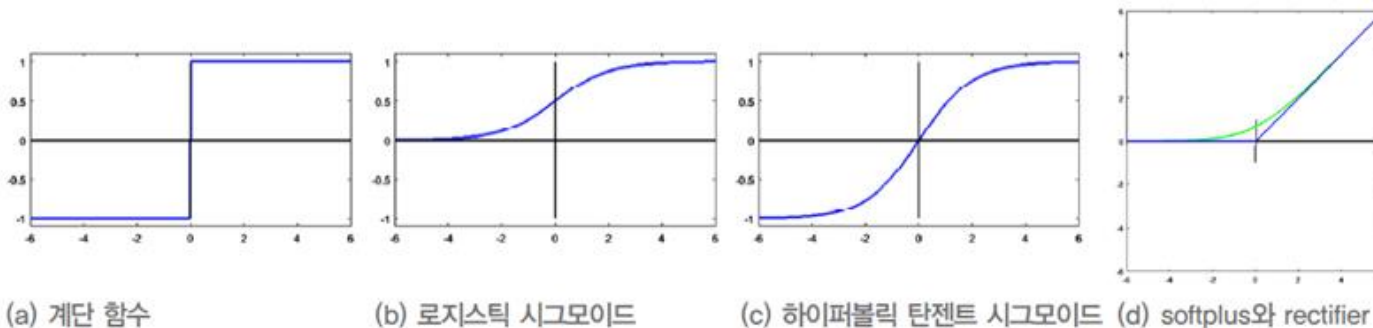
구 조	결정 구역	Exclusive -or	Classes with Meshed Regions	Most General Region Shapes
Single-layer j i 	Half Plane Bounded by Hyperplane			
Two-layer k j i 	Convex Open or Closed Regions			
Three-layer l k j i 	Arbitrary (Complexity limited by Number of Units)			

다층 퍼셉트론

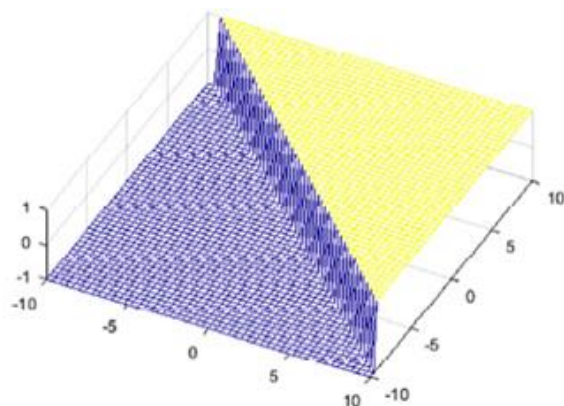
■ 활성화함수: 비선형 함수 변화

- 계단함수는 딱딱한 공간 분할 (영역을 점으로 변환)

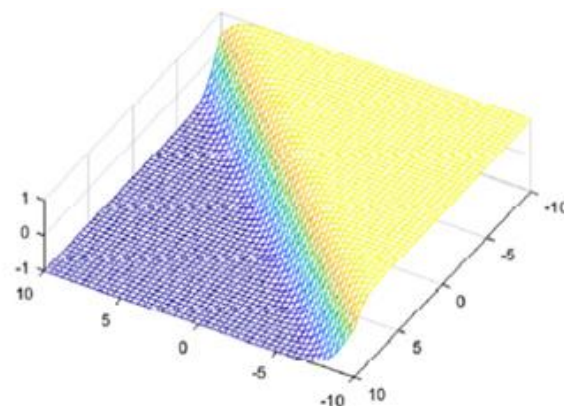
나머지 활성화함수는 부드러운 공간 분할 (영역을 영역으로 변환)



신경망이 사용하는 활성화함수



(a) 계단함수의 딱딱한 공간 분할



(b) 로지스틱 시그모이드의 부드러운 공간 분할

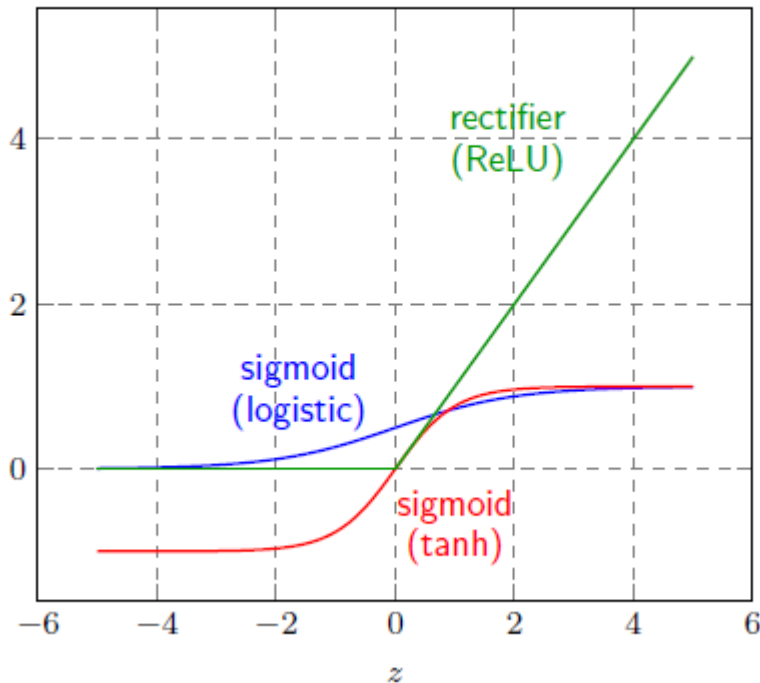
퍼셉트론의 공간 분할 유형

다층 퍼셉트론

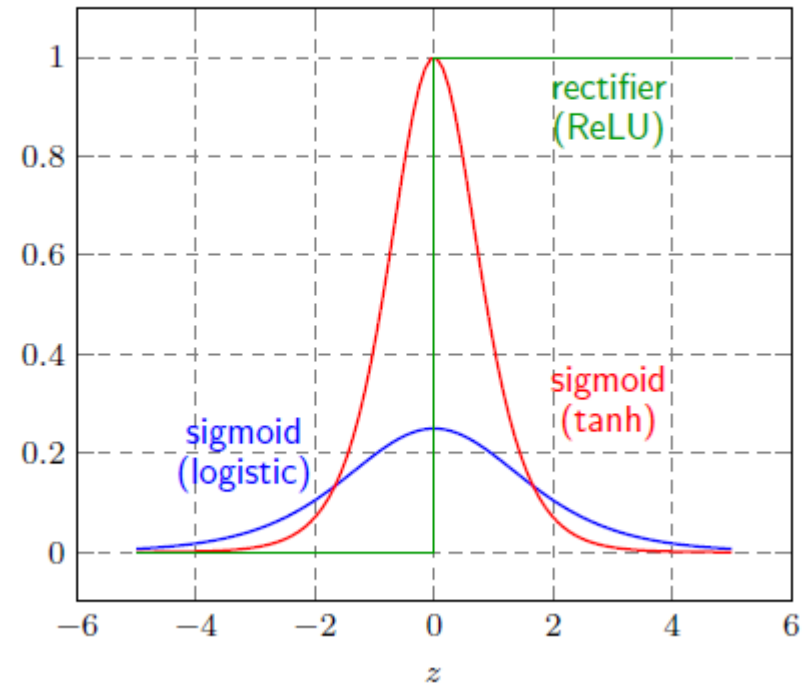
■ 신경망이 사용하는 다양한 활성화 함수

- 일반적으로 은닉층에서 로지스틱 시그모이드를 활성 함수로 많이 사용했음
 - 하지만, 시그모이드의 넓은 포화곡선은 경사 (기울기 미분) 기반한 학습을 어렵게 함
- 현재, 깊은 신경망은 ReLU를 사용

activation function: $g(z)$



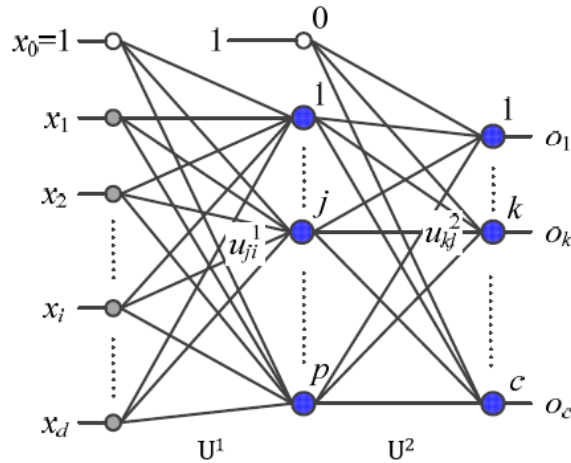
gradient: $g'(z)$



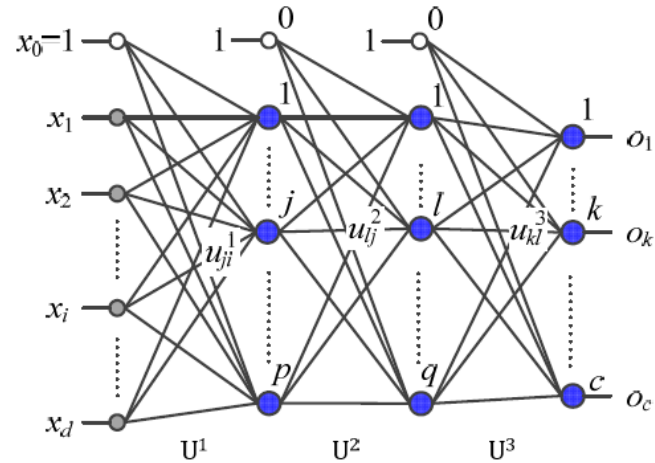
다층 퍼셉트론

■ 다층 퍼셉트론 구조

- [그림 (a)]는 입력층-은닉층-출력층의 2층 구조
- [그림 (b)]는 입력층-은닉층-은닉층-출력층의 3층 구조



(a) 2층 퍼셉트론



(b) 3층 퍼셉트론

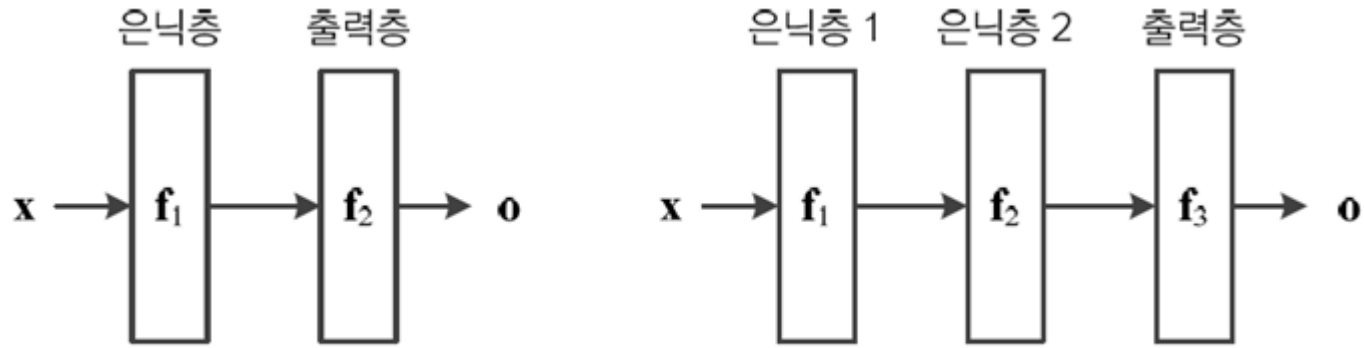
2층 퍼셉트론의 기준치 행렬:

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

다층 퍼셉트론

- 특징 벡터 \mathbf{x} 를 출력 벡터 \mathbf{o} 로 연결(mapping)하는 비선형함수로 간주할 수 있음

$$\left. \begin{array}{l} \text{2층 퍼셉트론: } \mathbf{o} = \mathbf{f}(\mathbf{x}) = \mathbf{f}_2(\mathbf{f}_1(\mathbf{x})) \\ \text{3층 퍼셉트론: } \mathbf{o} = \mathbf{f}(\mathbf{x}) = \mathbf{f}_3(\mathbf{f}_2(\mathbf{f}_1(\mathbf{x}))) \end{array} \right\}$$



다층 퍼셉트론을 간략화한 구조

- 깊은 인공신경망(deep neural networks)은 $\mathbf{o} = \mathbf{f}_L(\cdots \mathbf{f}_2(\mathbf{f}_1(\mathbf{x})))$, $L \geq 4$

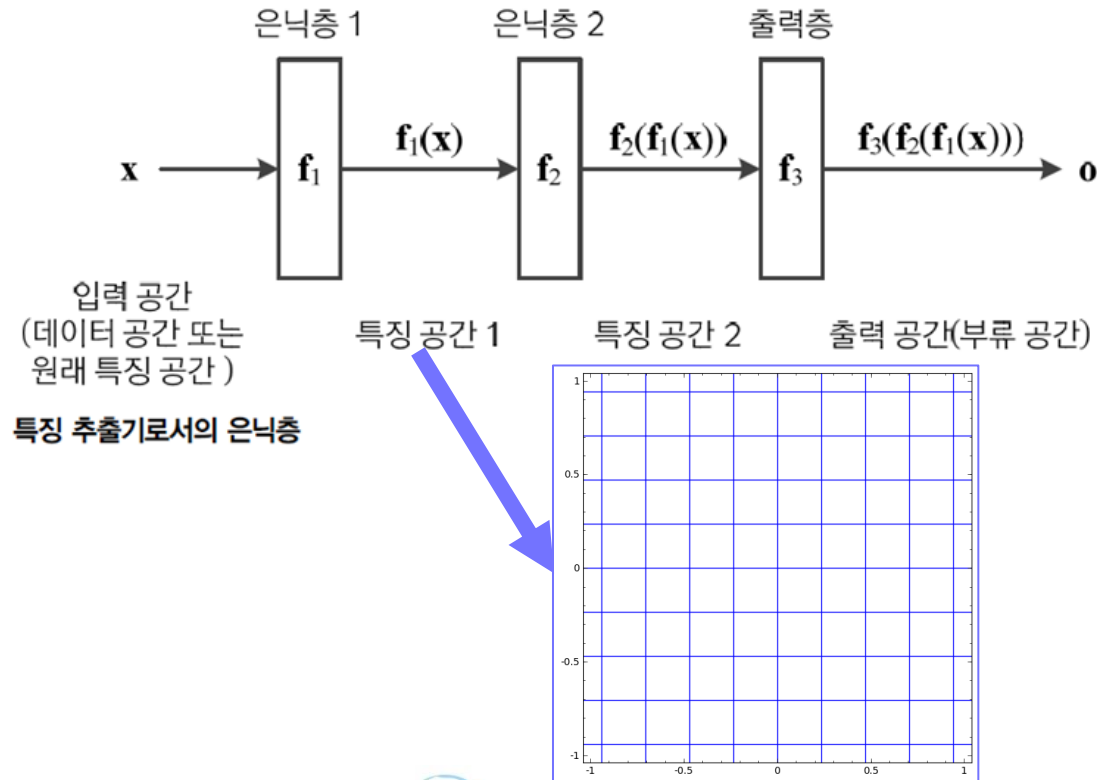
← 깊은 인공신경망의 학습 방법: 심층학습(deep learning)

다층 퍼셉트론

■ 은닉층은 특징 자동 추출기 (여러 개의 필터들을 적용함과 같음)

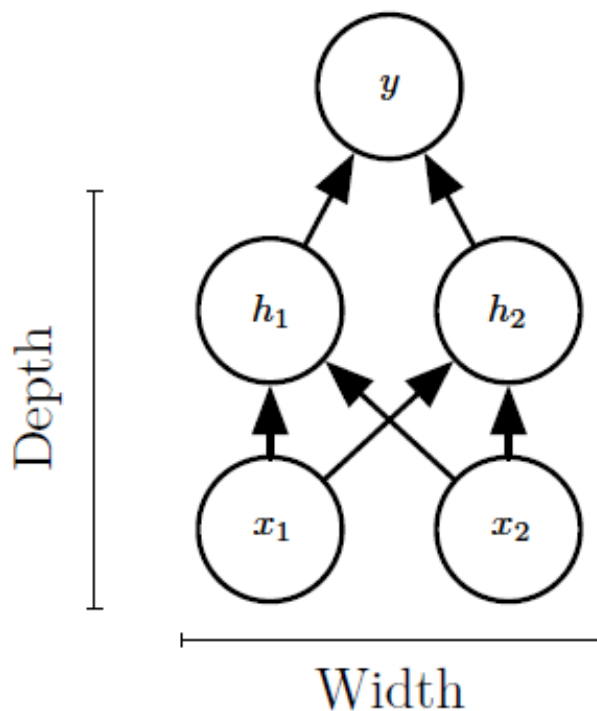
- 은닉층은 특징 벡터를 분류에 더 유리한 새로운 특징 공간으로 변환
- 표현 학습 (representation learning) 을 통해 데이터로부터 자동으로 특징 (data-driven feature) 추출함

(깊은 인공신경망은 더 많은 층을 거쳐 계층적인 표현학습을 함)



다층 퍼셉트론

■ 기본 구조

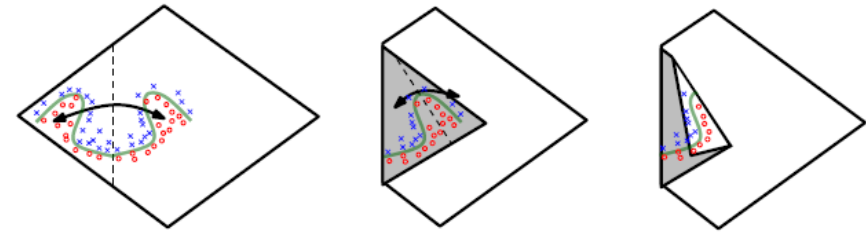
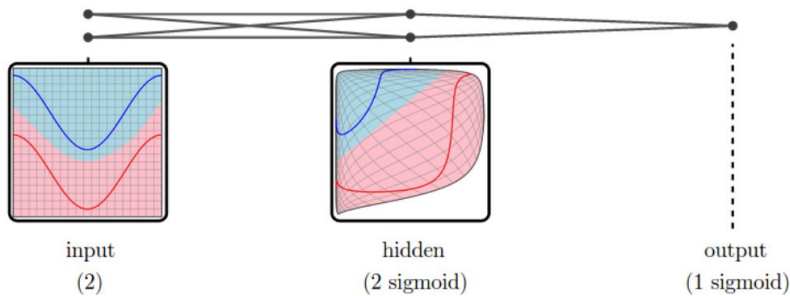


- 범용적 근사 이론 universal approximation theorem
 - 하나의 은닉층은 함수의 근사를 표현
- 다층 퍼셉트론도 공간을 변환하는 함수로 근사됨
- 얇은 은닉층의 구조
 - 더 과잉적합 되기 쉬움
- 일반적으로 깊은 은닉층의 구조가 좋은 성능을 가짐

다층 퍼셉트론

■ 은닉층의 깊이에 따른 이점

- 공간 변환과 지수의 표현 exponential representation



- 은닉층은 특징 공간 변환 → 지수적으로 많은 선형적인 영역 조각들로 표현
- 여러 은닉층을 통해 계층적으로 특징을 추출, 특징은 층이 깊어짐에 따라 점차 추상화됨
- 좋은 성능

