



# 클라우드와 AI

무단배포는 금지(본자료는 학습용으로제작)

출처 : 기계학습, 한빛아카데미  
인공지능, 생능출판사

# 인공지능과 기계 학습의 간략한 역사

- 1843      에이더 “... 해석엔진은 꽤 복잡한 곡을 작곡할 수도 있다.”라는 논문 발표[Ada1843]
- 1950      인공지능 여부를 판별하는 튜링 테스트[Turing1950]
- 1956      최초의 인공지능 학술대회인 다트머스 콘퍼런스 개최. ‘인공지능’ 용어 탄생[McCarthy1955]
- 1958      로젠블랫이 퍼셉트론 제안[Rosenblatt1958]  
            인공지능 언어 Lisp 탄생
- 1959      사무엘이 기계 학습을 이용한 체커 게임 프로그램 개발[Samuel1959]
- 1969      민스키가 퍼셉트론의 과대포장 지적. 신경망 내리막길 시작[Minsky1969]  
            제1회 IJCAI(International Joint Conference on Artificial Intelligence) 개최
- 1972      인공지능 언어 Prolog 탄생
- 1973      Lighthill 보고서로 인해 인공지능 내리막길, 인공지능 겨울<sup>AI winter</sup> 시작
- 1974      웨어보스가 오류 역전파 알고리즘을 기계 학습에 도입[Werbos1974]
- 1975경    의료진단 전문가 시스템 Mycin – 인공지능에 대한 관심 부활
- 1979      『IEEE Transactions on Pattern Analysis and Machine Intelligence』 저널 발간
- 1980      제1회 ICML(International Conference on Machine Learning) 개최  
            후쿠시마가 NeoCognitron 제안[Fukushima1980]
- 1986      『Machine Learning』 저널 발간  
            『Parallel Distributed Processing』 출간  
            다층 퍼셉트론으로 신경망 부활

# 인공지능과 기계 학습의 간략한 역사

- 1987      Lisp 머신의 시장 붕괴로 제2의 인공지능 겨울  
          UCI 리포지토리 서비스 시작  
          NIPSNeural Information Processing Systems 콘퍼런스 시작
- 1989      『Neural Computation』 저널 발간
- 1993      R 언어 탄생
- 1997      IBM 딥블루가 세계 체스 챔피언인 카스파로프 이김  
          LSTMLong short-term memory 개발됨
- 1998경    SVM이 MNIST 인식 성능에서 신경망 추월
- 1998      르쿤이 CNN의 실용적인 학습 알고리즘 제안[LeCun1998]  
          『Neural Networks: Tricks of the Trade』 출간
- 1999      NVIDIA 사에서 GPU 공개
- 2000      『Journal of Machine Learning Research』 저널 발간  
          OpenCV 최초 공개
- 2004      제1회 그랜드 챌린지(자율 주행)
- 2006      층별학습 탄생[Hinton2006a]
- 2007경    딥러닝이 MNIST 인식 성능에서 SVM 추월
- 2007      GPU 프로그래밍 라이브러리인 CUDA 공개

# 인공지능과 기계 학습의 간략한 역사

	어번 챌린지(도심 자율 주행)
	Scikit-learn 라이브러리 최초 공개
2009	Theano 서비스 시작
2010	ImageNet 탄생
	제1회 ILSVRC 대회
2011	IBM 왓슨이 제퍼디 우승자 꺾음
2012	MNIST에 대해 0.23% 오류율 달성
	AlexNet 발표 (3회 ILSVRC 우승)
2013	제1회 ICLR International Conference on Learning Representations 개최
2014	Caffe 서비스 시작
2015	TensorFlow 서비스 시작
	OpenAI 창립
2016	알파고와 이세돌의 바둑 대회에서 알파고 승리[Silver2016]
	『Deep Learning』 출간
2017	알파고 제로[Silver2017]

# 사회적 전망

## ■ 미래의 직업 변화

- 시의적절하고 심사숙고 해야 할 객관적 담론
- 프레이는 702개 직업의 사라질 위기를 확률로 계산 [Frey2017]
- 텔레마케터 99% ..... 오락 치료사 0.28%

## ■ 기계가 사람을 지배할지 모른다는 두려움

- 알파고 이후 매스컴을 통해 여과 없이 전파. 쓸데없는 과장에 불과
- 넷가에 다리 놓는 일과 목포-제주에 대교를 놓는 일은 규모만 다를 뿐 본질적으로 같은 일
- 오목 프로그램이나 바둑 프로그램은 규모만 다를 뿐 본질적으로 같은 일. 오목은 간단한 규칙으로 구현 가능하나 바둑은 미분을 사용한 복잡한 기계 학습 알고리즘 사용
- 현재 기계 학습은 온통 수학과 컴퓨터 알고리즘일 뿐

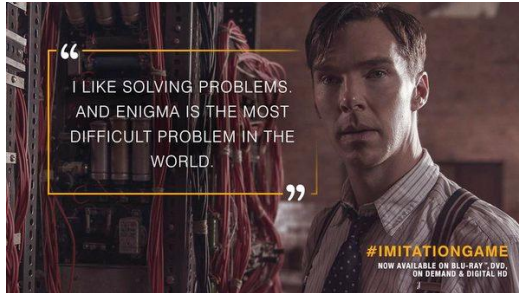
# 인공지능

## ❖ 인공지능(人工知能, Artificial Intelligence)

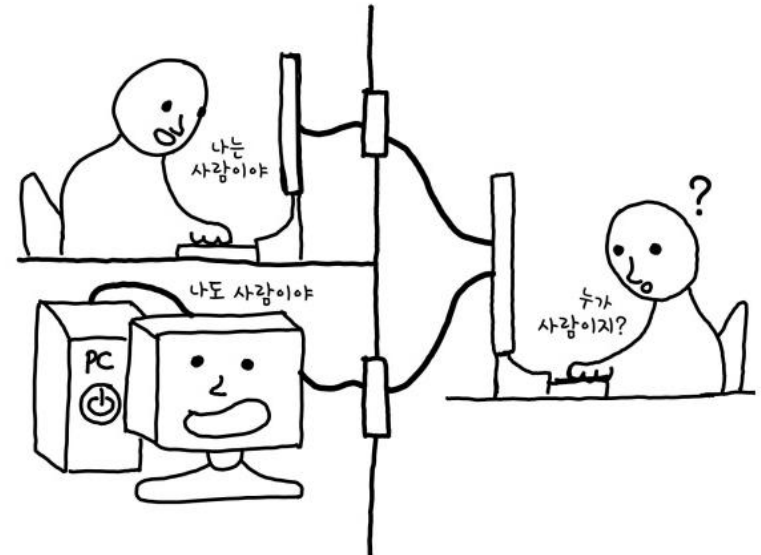
- 인공적으로 만든 지능
- **튜링 테스트**(Turing test)
  - 지능의 조작적 정의 (operational definition)
    - 조작적 정의: 측정할 수 있는 조건으로 어떤 속성을 기술



Alan Mathison Turing  
(1912.6~1954.6)



## 흉내 게임(imitation game)



## 튜링 상(Turing Award)

ACM에서는 매년 컴퓨터 분야의 기여가 큰 연구자에게 시상.  
\$250,000 (Intel, Google 후원)

# 인공지능

## ❖ 인공지능(Artificial Intelligence)

- **다트머스 회의(Dartmouth Conference, 1956)**
  - 존 마카시(John McCarthy)가 AI 용어 제안



(1927-2011)



**AI@50 conference (2005)** : 모어(Trenchard More), 매카시(John McCarthy), 민스키(Marvin Minsky, 1927-2016), 셀프리지 (Oliver Selfridge, 1926-2008), 솔로모노프(Ray Solomonoff, 1926-2009).



# 인공지능

## ❖ 인공지능(人工知能, Artificial Intelligence)

- 사람의 생각과 관련된 활동, 예를 들면 **의사 결정, 문제 해결, 학습** 등의 활동을 자동화하는 것 (벨만<sup>Bellman</sup>, 1978)
- **사람이 하면 더 잘 할 수 있는 일을** 컴퓨터가 하도록 하는 방법을 찾는 학문 (리치<sup>Rich</sup>와 나이트<sup>Knight</sup>, 1991)
- **지능이 요구되는 일을** 할 수 있는 기계를 만드는 예술 (커즈와일<sup>Kurzweil</sup>, 1990)
- **지능적인 에이전트**를 설계하는 학문 (풀<sup>Poole</sup>과 맥워쓰<sup>Mackworth</sup>, 1998)
- **인지하고, 추론하고, 행동**할 수 있도록 하는 컴퓨팅에 관련된 학문 (윌슨<sup>Wilson</sup>, 1992)
- **인공물**이 **지능적인 행위**를 하도록 하는 것 (닐슨<sup>Nissson</sup>, 1990)
- **사람**이 **의식적으로 하는 행동**을 **컴퓨터**가 할 수 있도록 하는 것



# 강한 인공지능과 약한 인공지능

## ❖ 강한 인공지능 (strong AI)

- **사람과 같은 지능**
- 마음을 가지고 사람처럼 느끼면서 지능적으로 행동하는 기계
- 추론, 문제해결, 판단, 계획, 의사소통, 자아 의식(self-awareness), 감정(sentiment), 지혜(sapience), 양심(conscience)
- 튜링 테스트

## ❖ 약한 인공지능 (weak AI, narrow AI)

- **특정 문제를 해결**하는 지능적 행동
- 사람의 지능적 행동을 **흉내** 낼 수 있는 수준
- 대부분의 인공지능 접근 방향
- **중국어 방 사고실험**(Chinese room thought experiment)

# 인공지능

## ■ 사람의 학습

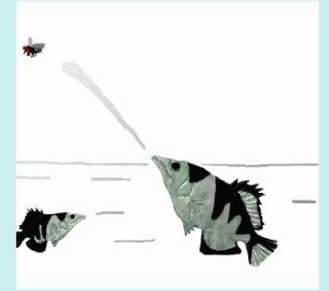
- 수학, 과학, 역사뿐 아니라 수영, 자전거 타기 등

## ■ 동물의 학습

- 예) 물총물고기의 목표물 맞추기 능력 향상



(a) 자전거 타기 학습



(b) 물총물고기의 사냥

그림 1-1 사람과 동물의 학습

## ■ 기계 학습

- 그렇다면 기계도 학습할 수 있을까?
- 경험을 통해 점점 성능이 좋아지는 기계를 만들 수 있을까?
- 이 책은 이 질문에 대한 답을 찾아가는 길

# 기계 학습의 정의

## ■ 학습이란? <표준국어대사전>

“경험의 결과로 나타나는, 비교적 지속적인 행동의 변화나 그 잠재력의 변화. 또는 지식을 습득하는 과정[국립국어원2017]”

## ■ 기계 학습이란?

### ■ 인공지능 초창기 사무엘의 정의

“Programming computers to learn from experience should eventually eliminate the need for much of this detailed programming effort. 컴퓨터가 경험을 통해 학습할 수 있도록 프로그래밍할 수 있다면, 세세하게 프로그래밍해야 하는 번거로움에서 벗어날 수 있다[Samuel1959].”

# 기계 학습의 정의

## ■ 기계 학습이란?

### ■ 현대적 정의

“A computer program is said to learn from experience  $E$  with respect to some class of tasks  $T$  and performance measure  $P$ , if its performance at tasks in  $T$ , as measured by  $P$ , improves with experience  $E$ . 어떤 컴퓨터 프로그램이  $T$ 라는 작업을 수행한다. 이 프로그램의 성능을  $P$ 라는 척도로 평가했을 때 경험  $E$ 를 통해 성능이 개선된다면 이 프로그램은 학습을 한다고 말할 수 있다[Mitchell1997(2쪽)].”

“Programming computers to optimize a performance criterion using example data or past experience 사례 데이터, 즉 과거 경험을 이용하여 성능 기준을 최적화하도록 프로그래밍하는 작업[Alpaydin2010]”

“Computational methods using experience to improve performance or to make accurate predictions 성능을 개선하거나 정확하게 예측하기 위해 경험을 이용하는 계산학 방법들[Mohri2012]”

# 지식기반 방식에서 기계 학습으로의 대전환

## ■ 인공지능의 탄생

- 컴퓨터의 뛰어난 능력
  - 사람이 어려워하는 일을 아주 쉽게 함
  - $80932.46789076 \times 0.39001324$ 와 같은 곱셈을 고속으로 수행(현재는 초당 수십억개)
  - 복잡한 함수의 미분과 적분 척척
- 컴퓨터에 대한 기대감 (컴퓨터의 능력 과신)
  - 사람이 쉽게 하는 일, 예를 들어 고양이/개 구별하는 일도 잘 하지 않을까
  - 1950년대에 인공지능이라는 분야 등장

## ■ 초창기는 지식기반 방식이 주류

- 예) "구멍이 2개이고 중간 부분이 홀쭉하며, 맨 위와 아래가 둥근 모양이라면 8이다"

# 지식기반 방식에서 기계 학습으로의 대전환

## ■ 큰 깨달음

- 지식기반의 한계
- 단추를 "가운데 구멍이 몇 개 있는 물체"라고 규정하면 많은 오류 발생



인식 시스템이 대처해야 하는 심한 변화 양상(8과 단추라는 패턴을 어떻게 기술할 것인가?)

- 사람은 변화가 심한 장면을 아주 쉽게 인식하지만, 왜 그렇게 인식하는지 서술하지는 못함

# 지식기반 방식에서 기계 학습으로의 대전환

## ■ 인공지능의 주도권 전환

- 지식기반 → 기계 학습
- 기계 학습: 데이터 중심 접근방식



\* 기계 학습으로 만든 최첨단 인공지능 제품들



# 연구 지능의 요소 기술

## ❖ 기계 학습(machine learning)

- **경험**을 통해서 나중에 유사하거나 같은 **일(task)**를 더 **효율적**으로 처리할 수 있도록 **시스템의 구조나 파라미터를 바꾸는 것**
- 컴퓨터가 **지식**을 갖게 만드는 작업
- **지도학습**
  - 입력과 대응하는 출력을 데이터로 제공하고 대응관계의 함수 찾기
- **비지도학습**
  - 데이터만 주어진 상태에서 유사한 것들을 서로 묶어 군집을 찾거나 확률분포 표현
- **강화학습**
  - 상황 별 행동에 따른 시스템의 보상 값(reward value)만을 이용하여, 시스템에 대한 바람직한 행동 정책(policy) 찾기



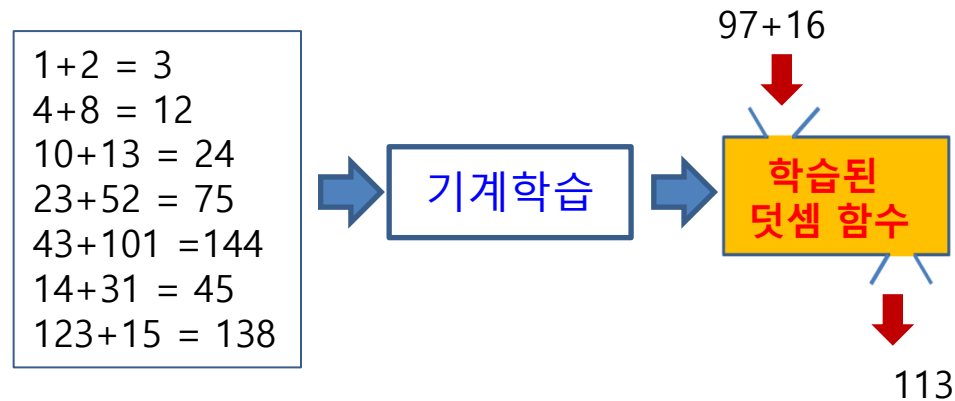
지식

# 연구 지능의 요소 기술

## ❖ 기계 학습(machine learning)

### ■ 지도학습(supervised learning)

- 입력(문제)과 대응하는 출력(답)을 데이터로 제공하고 대응관계의 함수 또는 패턴을 찾는 것



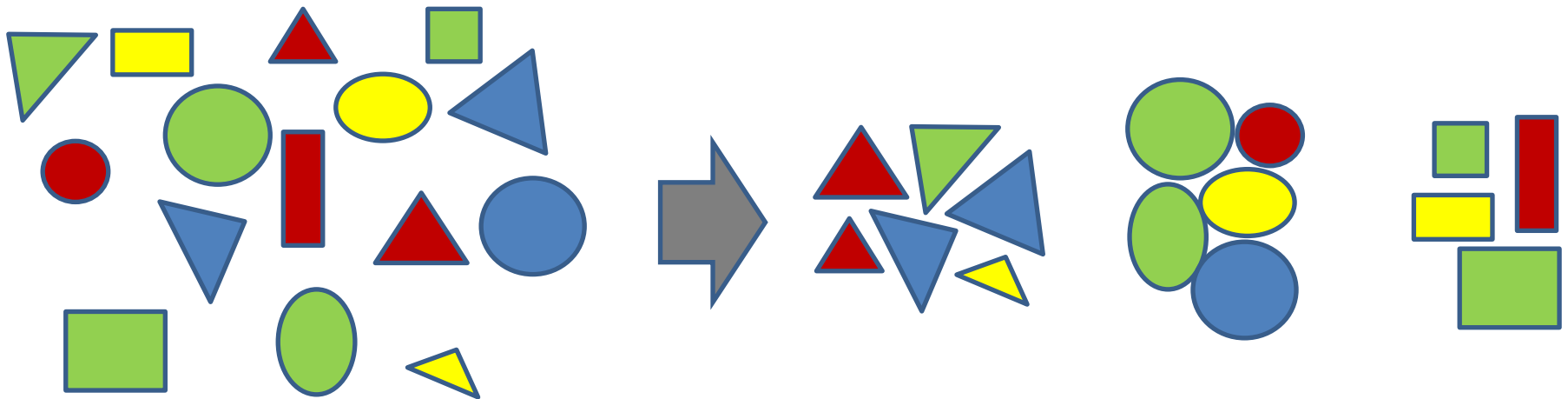
- 분류, 회귀

# 연구 지능의 요소 기술

## ❖ 기계 학습(machine learning) – cont.

### ▪ 비지도학습(unsupervised learning)

- 답이 없는 문제들만 있는 데이터들로부터 패턴을 추출하는 것



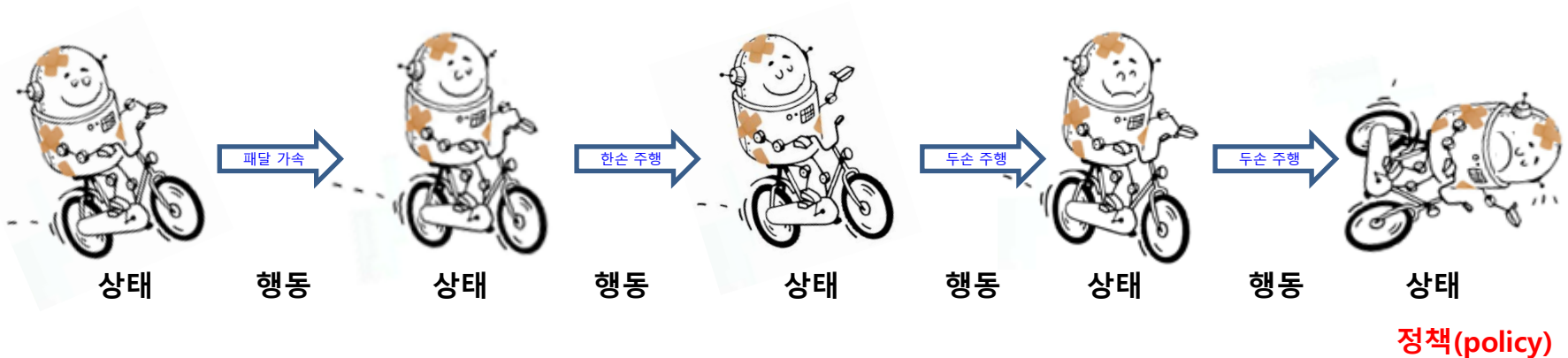
- 군집화, 밀도추정, 토픽 모델링

# 연구 지능의 요소 기술

## ❖ 기계 학습(machine learning) – cont.

### ▪ 강화학습(reinforcement learning)

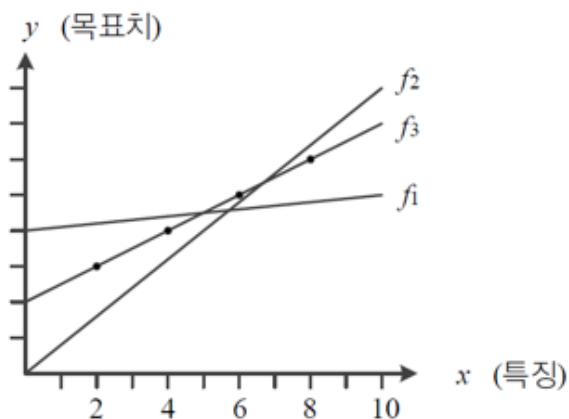
- 문제에 대한 직접적인 답을 주지는 않지만 경험을 통해 **기대 보상(expected reward)**이 최대가 되는 **정책(policy)**을 찾는 학습



# 기계 학습 개념

## ■ 간단한 기계 학습 예제

- 가로축은 시간, 세로축은 이동체의 위치
- 관측한 4개의 점이 데이터



## ■ 예측prediction 문제

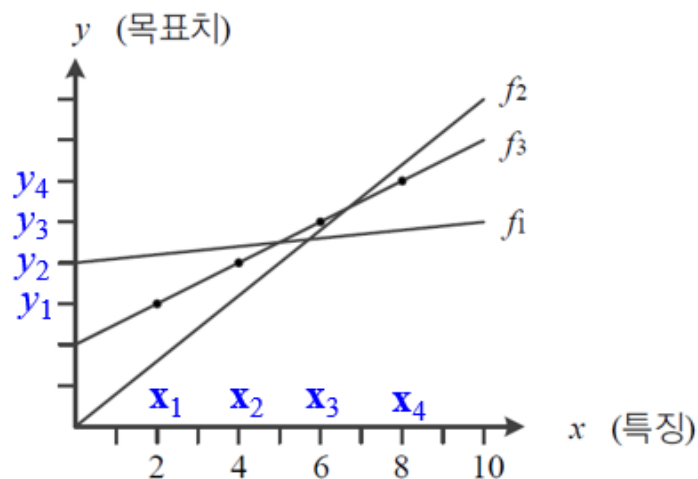
- 임의의 시간이 주어지면 이때 이동체의 위치는?
- 회귀regression 문제와 분류classification 문제로 나뉨
  - 회귀는 목표치가 실수, 분류는 분류값

# 기계 학습 개념

## ■ 훈련집합

- 가로축은 **특징**, 세로축은 **목표치**
- 관측한 4개의 점이 **훈련집합**을 구성함

훈련집합:  $\mathbb{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ ,  $\mathbb{Y} = \{y_1, y_2, \dots, y_n\}$



## 예제의 훈련집합

$\mathbb{X} = \{\mathbf{x}_1 = (2.0), \mathbf{x}_2 = (4.0), \mathbf{x}_3 = (6.0), \mathbf{x}_4 = (8.0)\}$

$\mathbb{Y} = \{y_1 = 3.0, y_2 = 4.0, y_3 = 5.0, y_4 = 6.0\}$

# 기계 학습 개념

## ■ 데이터를 어떻게 모델링할 것인가

- 눈대중으로 보면 직선을 이루므로 직선을 선택하자 → **모델로 직선을 선택**한 셈
- 직선 모델의 수식
  - 2개의 **매개변수**  $w$ 와  $b$

$$y = \underline{w}x + \underline{b}$$

## ■ 기계 학습은

- 가장 정확하게 예측할 수 있는, 즉 최적의 매개변수를 찾는 작업
- 처음에는 최적값을 모르므로 임의의 값에서 시작하고, 점점 성능을 개선하여 최적에 도달
- $f_1$ 에서 시작하여  $f_1 \rightarrow f_2 \rightarrow f_3$ 
  - 최적인  $f_3$ 은  $w=0.5$ 와  $b=2.0$



# 기계 학습 개념

## ■ 학습을 마치면,

- 예측에 사용
- 예) 10.0 순간의 이동체 위치를 알고자 하면,  $f_3(10.0)=0.5*10.0+2.0=7.0$ 이라 예측함

## ■ 기계 학습의 궁극적인 목표

- 훈련집합에 없는 새로운 샘플에 대한 오류를 최소화 (새로운 샘플 집합: 테스트 집합)
- 테스트 집합에 대한 높은 성능을 **일반화**generalization 능력이라 부름

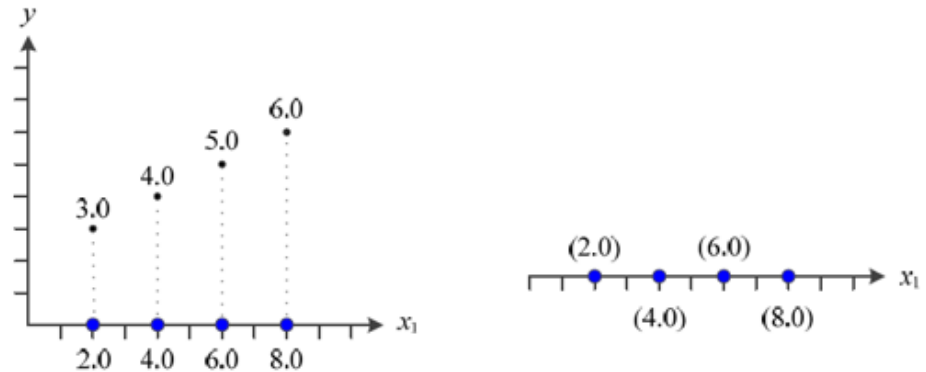
# 사람의 학습과 기계 학습

기준	사람의 학습	기계 학습
학습 과정	능동적	수동적
데이터 형식	자연에 존재하는 그대로	일정한 형식에 맞추어 사람이 준비함
동시에 학습 가능한 과업 수	자연스럽게 여러 과업을 학습	하나의 과업만 가능
학습 원리에 대한 지식	매우 제한적으로 알려져 있음	모든 과정이 밝혀져 있음
수학 의존도	매우 낮음	매우 높음
성능 평가	경우에 따라 객관적이거나 주관적	객관적(수치로 평가, 예를 들어 정확률 99.8%)
역사	수백만 년	60년 가량

\* 사람의 학습과 기계 학습의 비교

# 1차원과 2차원 특징 공간

## ■ 1차원 특징 공간 →



(a) 1차원 특징 공간(왼쪽: 특징과 목표값을 축으로 표시, 오른쪽: 특징만 축으로 표시)

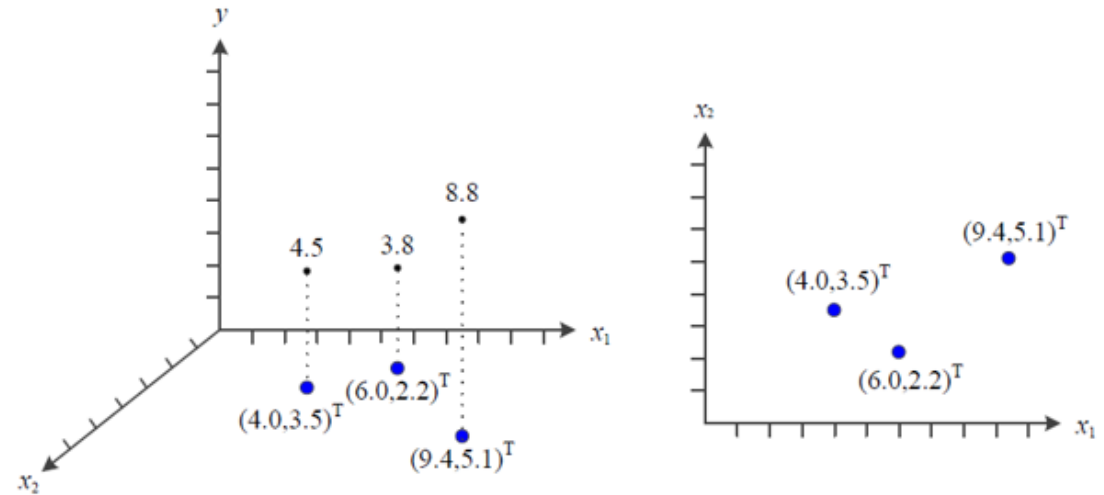
## ■ 2차원 특징 공간 →

### ■ 특징 벡터 표기

- $\mathbf{x}=(x_1, x_2)^T$

### ■ 예시

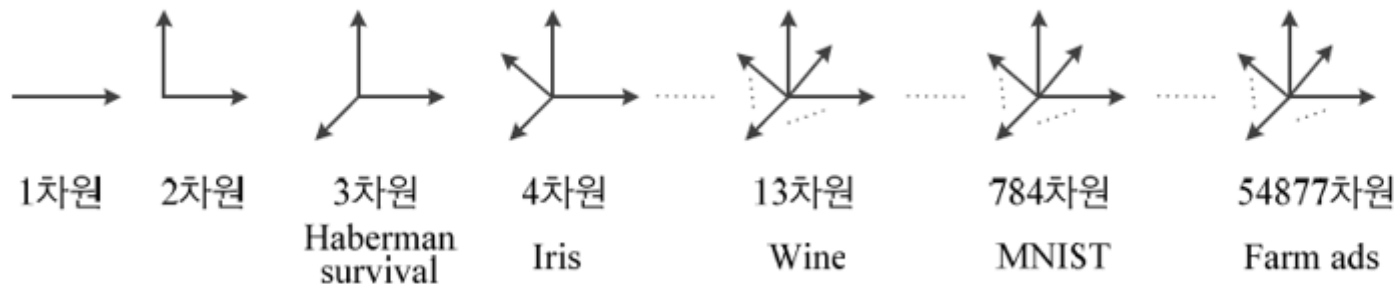
- $\mathbf{x}=(\text{몸무게}, \text{키})^T$ ,  $y=\text{장타율}$
- $\mathbf{x}=(\text{체온}, \text{두통})^T$ ,  $y=\text{감기 여부}$



(b) 2차원 특징 공간(왼쪽: 특징 벡터와 목표값을 축으로 표시, 오른쪽: 특징 벡터만 축으로 표시)

## 다차원 특징 공간

## ■ 다차원 특징 공간 예제



Haberman survival:  $\mathbf{x} = (\text{나이}, \text{수술년도}, \text{양성 림프샘 개수})^T$

$$\text{Inis: } \mathbf{x} = (\text{꽃받침 길이, 꽃받침 너비, 꽃잎 길이, 꽃잎 너비})^T$$

Wine:  $\mathbf{x} = (\text{Alcohol, Malic acid, Ash, Alkalinity of ash, Magnesium, Total phenols, Flavanoids, Nonflavanoid phenols, Proanthocyanins, Color intensity, Hue, OD280 / OD315 of diluted wines, Proline})^T$

$$\text{MNIST: } \mathbf{x} = (\text{화소1}, \text{화소2}, \dots, \text{화소784})^T$$

Farm ads:  $\mathbf{x} = (\text{단어1}, \text{단어2}, \dots, \text{단어54877})^T$

# 다차원 특징 공간

## ■ $d$ -차원 데이터

- 특징 벡터 표기:  $\mathbf{x}=(x_1, x_2, \dots, x_d)^T$

## ■ $d$ -차원 데이터를 위한 학습 모델

- 직선 모델을 사용하는 경우 매개변수 수= $d+1$

$$y = \underline{w_1}x_1 + \underline{w_2}x_2 + \dots + \underline{w_d}x_d + \underline{b}$$

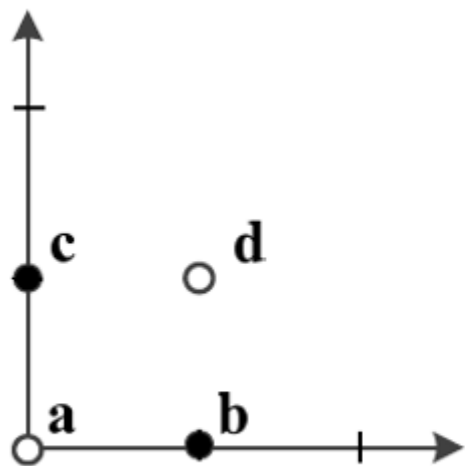
- 2차 곡선 모델을 사용하면 매개변수 수가 크게 증가
  - 매개변수 수= $d^2+d+1$
  - 예) Iris 데이터:  $d=4$ 이므로 21개의 매개변수
  - 예) MNIST 데이터:  $d=784$ 이므로 615,441개의 매개변수

$$y = \underline{w_1}x_1^2 + \underline{w_2}x_2^2 + \dots + \underline{w_d}x_d^2 + \underline{w_{d+1}}x_1x_2 + \dots + \underline{w_{d^2}}x_{d-1}x_d + \underline{w_{d^2+1}}x_1 \\ + \dots + \underline{w_{d^2+d}}x_d + \underline{b}$$

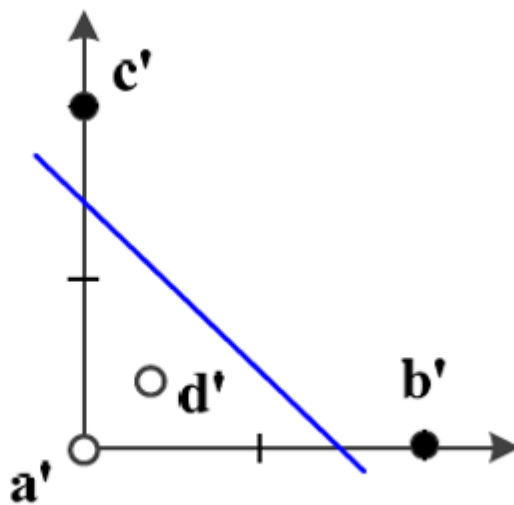
# 특징 공간 변환과 표현 학습

## ■ 선형 분리 불가능(Linearly non-separable)한 원래 특징 공간

- 직선 모델을 적용하면 75% 정확률이 한계



(a) 원래 특징 공간



(b) 분류에 더 유리하도록 변환된 새로운 특징 공간

# 특징 공간 변환과 표현 학습

## ■ 새로운 특징 공간

- 직선 모델로 100% 정확률

$$\text{원래 특징 벡터 } \mathbf{x} = (x_1, x_2)^T \rightarrow \text{변환된 특징 벡터 } \mathbf{x}' = \left( \frac{x_1}{2x_1x_2 + 0.5}, \frac{x_2}{2x_1x_2 + 0.5} \right)^T$$

$$\mathbf{a} = (0,0)^T \rightarrow \mathbf{a}' = (0,0)^T$$

$$\mathbf{b} = (1,0)^T \rightarrow \mathbf{b}' = (2,0)^T$$

$$\mathbf{c} = (0,1)^T \rightarrow \mathbf{c}' = (0,2)^T$$

$$\mathbf{d} = (1,1)^T \rightarrow \mathbf{d}' = (0.4,0.4)^T$$

## ■ 표현 학습representation learning

- 좋은 특징 공간을 자동으로 찾는 작업
- 딥러닝은 다수의 은닉층을 가진 신경망을 이용하여 계층적인 특징 공간을 찾아냄
  - 왼쪽 은닉층은 저급 특징(에지, 구석점 등), 오른쪽은 고급 특징(얼굴, 바퀴 등) 추출



# 특징 공간 변환과 표현 학습

## ■ 차원에 대한 몇 가지 설명

### ■ 차원에 무관하게 수식 적용 가능함

- 예) 두 점  $\mathbf{a}=(a_1, a_2, \dots, a_d)^T$ 와  $\mathbf{b}=(b_1, b_2, \dots, b_d)^T$  사이의 거리는 모든  $d$ 에 대해 성립

$$\text{dist}(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{i=1}^d (a_i - b_i)^2}$$

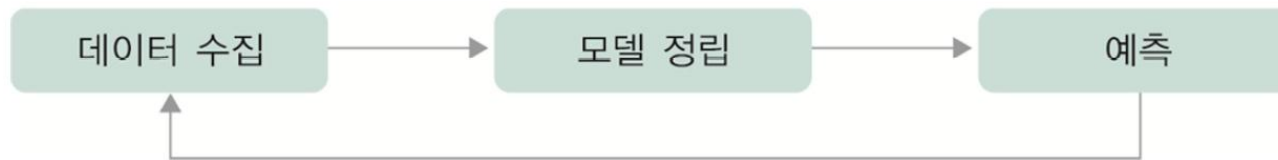
- 보통 2~3차원의 저차원에서 식을 고안한 다음 고차원으로 확장 적용

### ■ 차원의 저주

- 차원이 높아짐에 따라 발생하는 현실적인 문제들
- 예)  $d=4$ 인 Iris 데이터에서 축마다 100개 구간으로 나누면 총  $100^4=1$ 억 개의 칸
- 예)  $d=784$ 인 MNIST 샘플의 화소가 0과 1값을 가진다면  $2^{784}$ 개의 칸. 이 거대한 공간에 고작 6만 개의 샘플을 흩뿌린 매우 희소한 분포

# 데이터에 대한 이해

## ■ 과학 기술의 발전 과정



- 예) 튀코 브라헤는 천동설이라는 틀린 모델을 선택함으로써 자신이 수집한 데이터를 설명하지 못함. 케플러는 지동설 모델을 도입하여 제1, 제2, 제3법칙을 완성함

## ■ 기계 학습

- 기계 학습이 푸는 문제는 훨씬 복잡함
  - 예) '8' 숫자 패턴과 '단추' 패턴의 다양한 변화 양상
- 단순한 수학 공식으로 표현 불가능함
- 자동으로 모델을 찾아내는 과정이 필수

# 데이터 생성 과정

## ■ 데이터 생성 과정을 완전히 아는 인위적 상황의 예제

- 예) 두 개 주사위를 던져 나온 눈의 합을  $x$ 라 할 때,  $y=(x-7)^2+1$  점을 받는 게임
- 이런 상황을 '데이터 생성 과정을 완전히 알고 있다'고 말함
  - $x$ 를 알면 정확히  $y$ 를 예측할 수 있음
    - 실제 주사위를 던져  $\mathbb{X} = \{3,10,8,5\}$ 를 얻었다면,  $\mathbb{Y} = \{17,10,2,5\}$
  - $x$ 의 발생 확률  $P(x)$ 를 정확히 알 수 있음
  - $P(x)$ 를 알고 있으므로, 새로운 데이터 생성 가능

## ■ 실제 기계 학습 문제

- 데이터 생성 과정을 알 수 없음
- 단지 주어진 훈련집합  $\mathbb{X}, \mathbb{Y}$ 로 예측 모델 또는 생성 모델을 근사 추정할 수 있을 뿐

# 데이터베이스의 중요성

## ■ 데이터베이스의 품질

- 주어진 응용에 맞는 충분히 다양한 데이터를 충분한 양만큼 수집 → 추정 정확도 높아짐
- 예) 정면 얼굴만 가진 데이터베이스로 학습하고 나면, 기운 얼굴은 매우 낮은 성능  
→ 주어진 응용 환경을 자세히 살핀 다음 그에 맞는 데이터베이스 확보는 아주 중요함

## ■ 아주 많은 공개 데이터베이스

- 기계 학습의 초파리로 여겨지는 3가지 데이터베이스: Iris, MNIST, ImageNet
- 위키피디아에서 'list of datasets for machine learning research'로 검색
- UCI 리퍼지토리 (2017년 11월 기준으로 394개 데이터베이스 제공)

# 데이터베이스의 중요성

- Iris 데이터베이스는 통계학자인 피셔 교수가 1936년에 캐나다 동부 해안의 가스페 반도에 서식하는 3종의 붓꽃(*setosa*, *versicolor*, *virginica*)을 50송이씩 채취하여 만들었다[Fisher1936]. 150개 샘플 각각에 대해 꽃받침 길이, 꽃받침 너비, 꽃잎 길이, 꽃잎 너비를 측정하여 기록하였다. 따라서 4차원 특징 공간이 형성되며 목꽃값은 3종을 숫자로 표시함으로써 1, 2, 3 값 중의 하나이다. <http://archive.ics.uci.edu/ml/datasets/Iris>에 접속하여 내려받을 수 있다.

Sepal length ◆	Sepal width ◆	Petal length ◆	Petal width ◆	Species ◆
5.2	3.5	1.4	0.2	<i>I. setosa</i>
4.9	3.0	1.4	0.2	<i>I. setosa</i>
4.7	3.2	1.3	0.2	<i>I. setosa</i>
4.6	3.1	1.5	0.2	<i>I. setosa</i>
7.0	3.2	4.7	1.4	<i>I. versicolor</i>
6.4	3.2	4.5	1.5	<i>I. versicolor</i>
6.9	3.1	4.9	1.5	<i>I. versicolor</i>
5.5	2.3	4.0	1.3	<i>I. versicolor</i>
6.3	3.3	6.0	2.5	<i>I. virginica</i>
5.8	2.7	5.1	1.9	<i>I. virginica</i>
7.1	3.0	5.9	2.1	<i>I. virginica</i>
6.3	2.9	5.6	1.8	<i>I. virginica</i>



Setosa



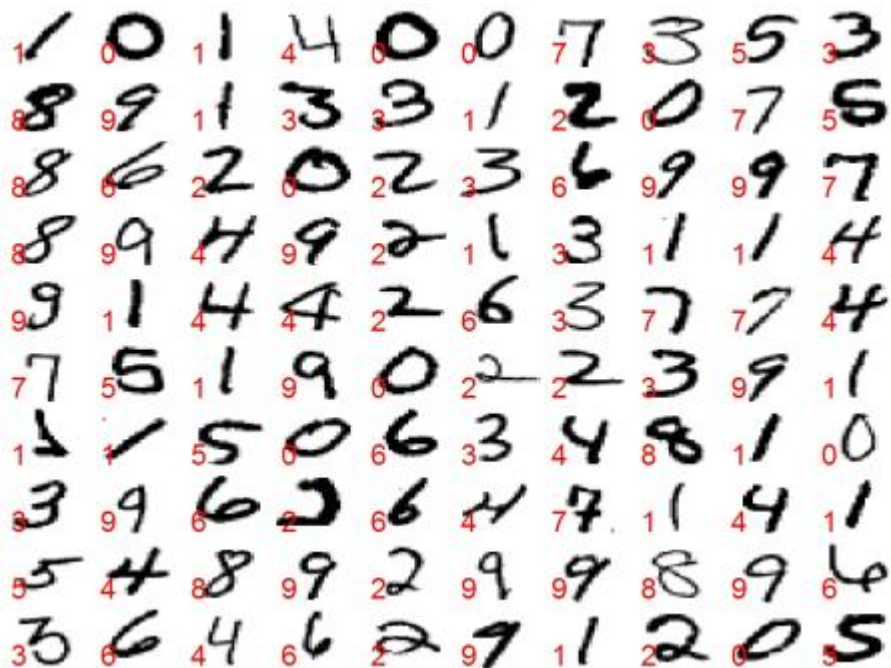
Versicolor



Virginica

# 데이터베이스의 중요성

- MNIST 데이터베이스는 미국표준국(NIST)에서 수집한 필기 숫자 데이터베이스로, 훈련집합 60,000자, 테스트집합 10,000자를 제공한다. <http://yann.lecun.com/exdb/mnist>에 접속하면 무료로 내려받을 수 있으며, 1988년부터 시작한 인식률 경쟁 기록도 볼 수 있다. 2017년 8월 기준으로는 [Ciresan2012] 논문이 0.23%의 오류율로 최고 자리를 차지하고 있다. 테스트집합에 있는 10,000개 샘플에서 단지 23개만 틀린 것이다.



# 데이터베이스의 중요성

- ImageNet 데이터베이스는 정보검색 분야에서 만든 WordNet의 단어 계층 분류를 그대로 따랐고, 부류마다 수백에서 수천 개의 영상을 수집하였다[Deng2009]. 총 21,841개 부류에 대해 총 14,197,122개의 영상을 보유하고 있다. 그중에서 1,000개 부류를 뽑아 ILSVRC(ImageNet Large Scale Visual Recognition Challenge)라는 영상인식 경진대회를 2010년부터 매년 개최하고 있다. 대회 결과에 대한 자세한 내용은 4.4절을 참조하라. <http://image-net.org>에서 내려받을 수 있다.



(a) 'swing' 부류



(b) 'Great white shark' 부류

그림 4-20 ImageNet의 예제 영상



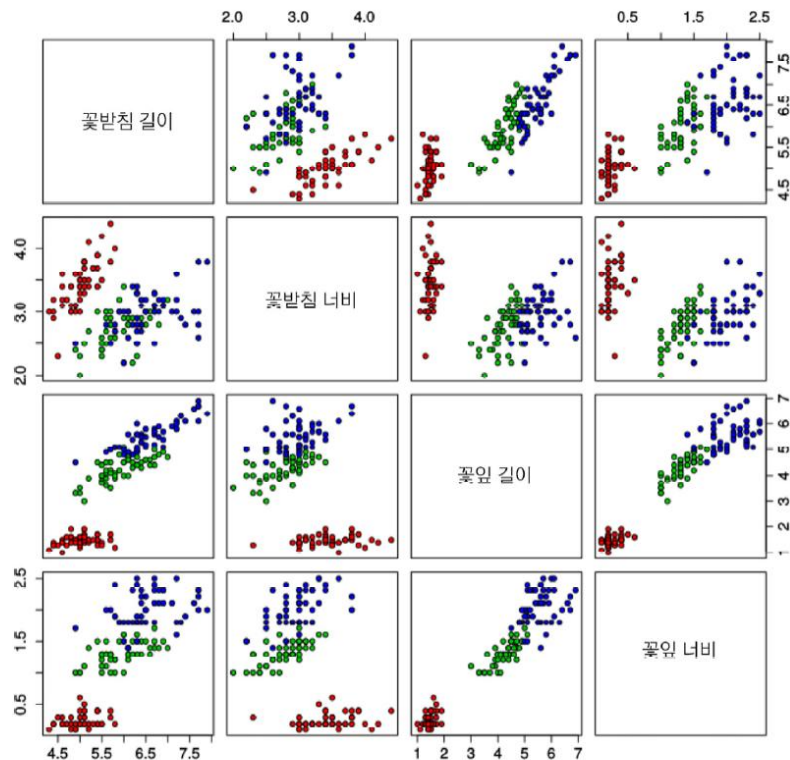
# 데이터 가시화

- 4차원 이상의 초공간은 한꺼번에 가시화 불가능

- 여러 가지 가시화 기법

- 2개씩 조합하여 여러 개의 그래프 그림

그림에 있는 ●=setosa, ●=versicolor, ●=virginica



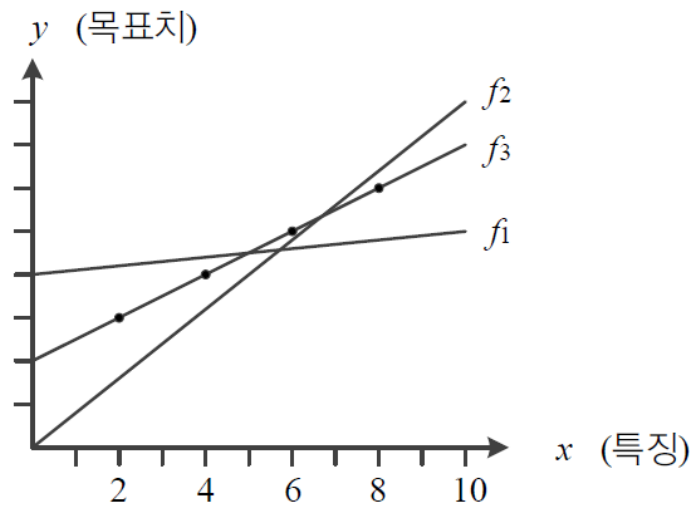
- 고차원 공간을 저차원으로 변환하는 기법들

# 간단한 기계 학습의 예

## ■ 선형 회귀 문제

- 직선 모델을 사용하므로 두 개의 매개변수  $\theta = (w, b)^T$

$$y = wx + b$$



간단한 기계 학습 예제

# 간단한 기계 학습의 예

## ■ 목적 함수objective function (또는 비용 함수cost function)

### ■ 선형 회귀를 위한 목적 함수

- $f_{\theta}(\mathbf{x}_i)$ 는 예측함수의 출력,  $y_i$ 는 예측함수가 맞추어야 하는 목표값이므로  $f_{\theta}(\mathbf{x}_i) - y_i$ 는 오차
- 아래 식을 **평균제곱오차**MSE(mean squared error)라 부름

$$J(\theta) = \frac{1}{n} \sum_{i=1}^n (f_{\theta}(\mathbf{x}_i) - y_i)^2$$

- 처음에는 최적 매개변수 값을 알 수 없으므로 난수로  $\theta_1 = (w_1, b_1)^T$  설정  $\rightarrow \theta_2 = (w_2, b_2)^T$ 로 개선  $\rightarrow \theta_3 = (w_3, b_3)^T$ 로 개선  $\rightarrow \theta_3$ 는 최적해  $\hat{\theta}$ 
  - 이때  $J(\theta_1) > J(\theta_2) > J(\theta_3)$

# 간단한 기계 학습의 예

- 기계 학습이 할 일을 공식화하면,

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} J(\theta)$$

- 기계 학습은 작은 개선을 반복하여 최적해를 찾아가는 수치적 방법

- 알고리즘 형식으로 쓰면,

입력: 훈련집합  $\mathbb{X}$ 와  $\mathbb{Y}$

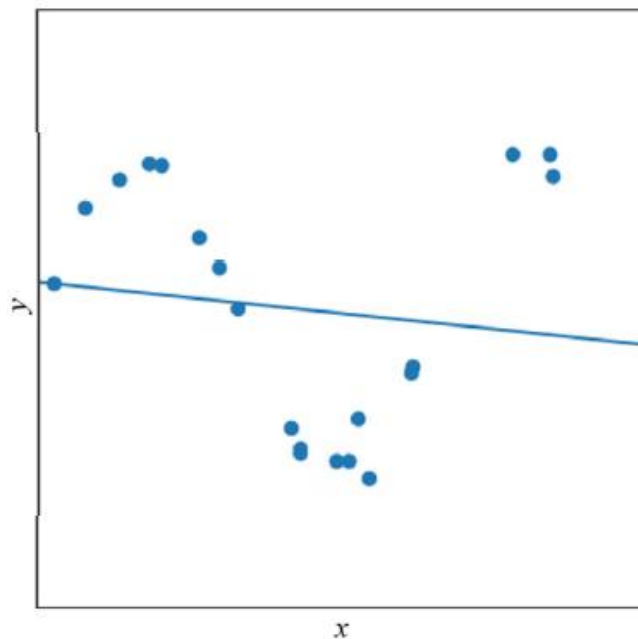
출력: 최적의 매개변수  $\hat{\theta}$

```
1  난수를 생성하여 초기 해  $\theta_1$ 을 설정한다.
2   $t=1$ 
3  while ( $J(\theta_t)$ 가 0.0에 충분히 가깝지 않음)    // 수렴 여부 검사
4       $J(\theta_t)$ 가 작아지는 방향  $\Delta\theta_t$ 를 구한다.    //  $\Delta\theta_t$ 는 주로 미분을 사용하여 구함
5       $\theta_{t+1} = \theta_t + \Delta\theta_t$ 
6       $t=t+1$ 
7   $\hat{\theta} = \theta_t$ 
```

# 간단한 기계 학습의 예

## ■ 좀더 현실적인 상황

- 지금까지는 데이터가 선형을 이루는 아주 단순한 상황을 고려함
- 실제 세계는 선형이 아니며 잡음이 섞임 → 비선형 모델이 필요



선형 모델의 한계

# 분류

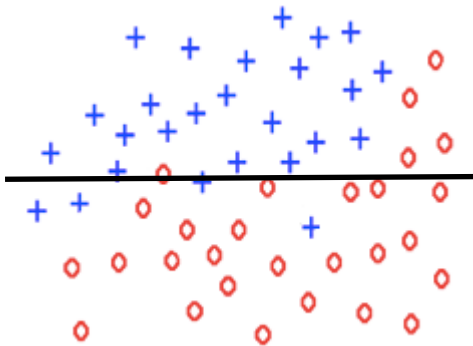
## ❖ 과적합(overfitting)과 부적합(underfitting)

### ■ 과적합

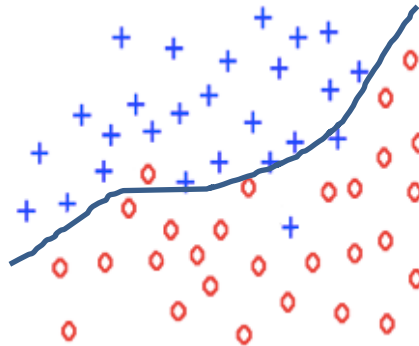
- 학습 데이터에 대해서 지나치게 잘 학습된 상태
- 데이터는 오류나 잡음을 포함할 개연성이 크기 때문에, 학습 데이터에 대해 매우 높은 성능을 보이더라도 학습되지 않은 데이터에 대해 좋지 않은 성능을 보일 수 있음

### ■ 부적합

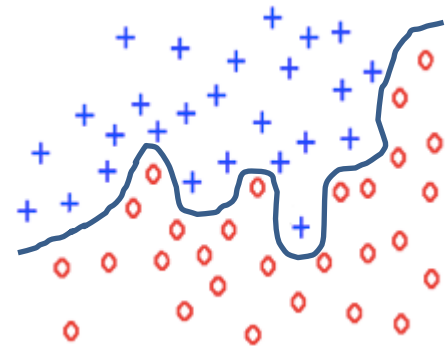
- 학습 데이터를 충분히 학습하지 않은 상태



부적합(underfitting)



정적합(good fitting)



과적합(overfitting)

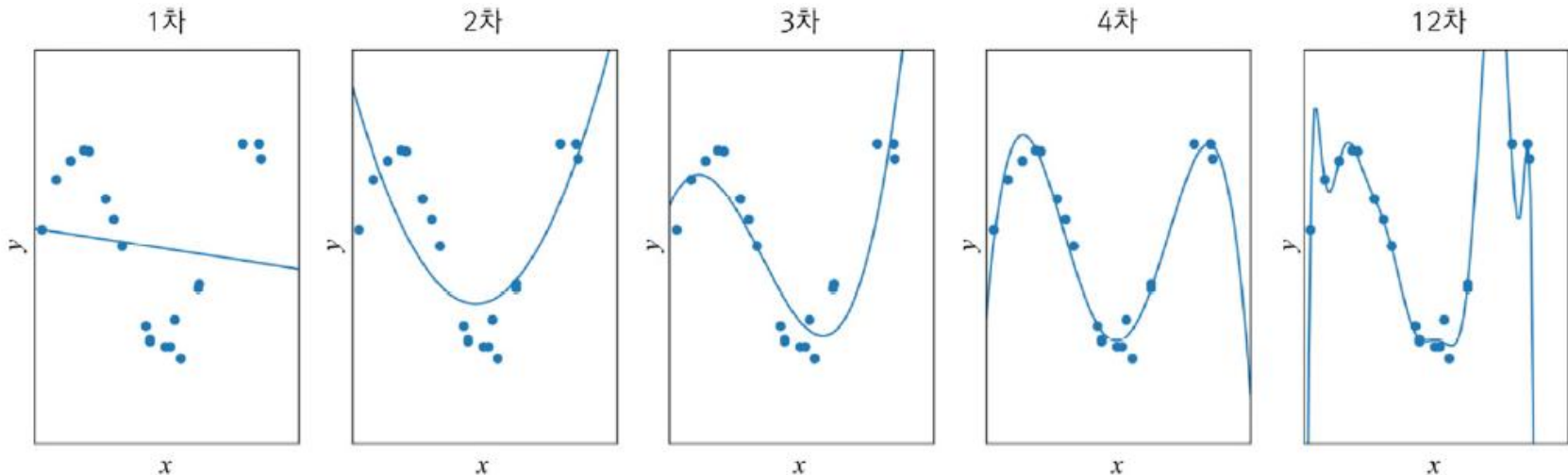
# 과소적합과 과잉적합

## ■ 1차 모델은 과소적합

- 모델의 '용량이 작아' 오차가 클 수밖에 없는 현상

## ■ 비선형 모델을 사용하는 대안

- 2차, 3차, 4차, 12차는 다항식 곡선을 선택한 예
- 1차(선형)에 비해 오차가 크게 감소함

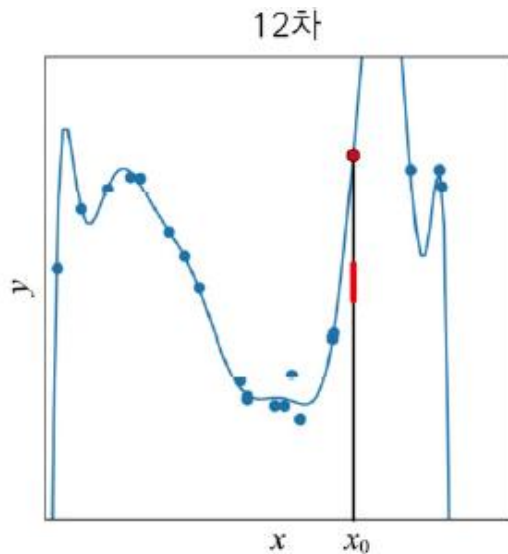


과소적합과 과잉적합 현상

# 과소적합과 과잉적합

## ■ 과잉적합

- 12차 다항식 곡선을 채택한다면 훈련집합에 대해 거의 완벽하게 근사화함
- 하지만 '새로운' 데이터를 예측한다면 큰 문제 발생
  - $x_0$ 에서 빨간 막대 근방을 예측해야 하지만 빨간 점을 예측
- 이유는 '용량이 크기' 때문. 학습 과정에서 잡음까지 수용 → 과잉적합 현상
- 적절한 용량의 모델을 선택하는 모델 선택 작업이 필요함



**과잉적합되었을 때 부정확한 예측 현상**



# 바이어스와 분산

## ■ 1차~12차 다항식 모델의 비교 관찰

- 1~2차는 훈련집합과 테스트집합 모두 낮은 성능
- 12차는 훈련집합에 높은 성능을 보이나 테스트집합에서는 낮은 성능 → 낮은 일반화 능력
- 3~4차는 훈련집합에 대해 12차보다 낮겠지만 테스트집합에는 높은 성능 → 높은 일반화 능력

# 검증집합과 교차검증을 이용한 모델 선택 알고리즘

## ■ 교차검증cross validation

- 비용 문제로 별도의 검증집합이 없는 상황에 유용한 모델 선택 기법
- 훈련집합을 등분하여, 학습과 평가 과정을 여러 번 반복한 후 평균 사용

**입력:** 모델집합  $\Omega$ , 훈련집합, 테스트집합, 그룹 개수  $k$

**출력:** 최적 모델과 성능

- 1 훈련집합을  $k$ 개의 그룹으로 등분한다.
- 2 for ( $\Omega$ 에 있는 각각의 모델)
- 3     for ( $i=1$  to  $k$ )
- 4          $i$ 번째 그룹을 제외한  $k-1$ 개 그룹으로 모델을 학습시킨다.
- 5         학습된 모델의 성능을  $i$ 번째 그룹으로 측정한다.
- 6      $k$ 개 성능을 평균하여 해당 모델의 성능으로 취한다.
- 7 가장 높은 성능을 보인 모델을 선택한다.
- 8 테스트집합으로 선택된 모델의 성능을 측정한다.



THANK  
YOU