

인공지능

[1. 소개]

소프트웨어융합대학
소프트웨어학부

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

저번 수업에는 뭐했지?

수업소개

■ 시간 및 장소

- 02분반
 - 월/수 12:00-13:15
 - 7호관 336호

■ 담당: 이재구 교수

- email: jaekoo@kookmin.ac.kr
- 연구실: 생활관 B동 609호

■ 면담시간: 목 15:00-17:00

- 장소, 시간: email 사전 약속 요망

■ 조교: 허성실

- email: seonsil@kookmin.ac.kr
- 연구실: 7호관 7층 KLAB



수업소개

■ 교재: 수업 강의 자료

■ 기본도서

- 기계학습, 오일석

■ 참고도서

- Machine Learning: a Probabilistic Perspective by K. Murphy

- Deep Learning by Goodfellow, Bengio, and Courville

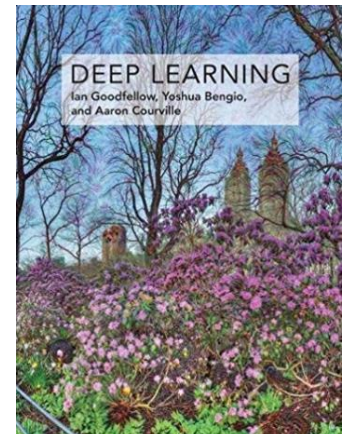
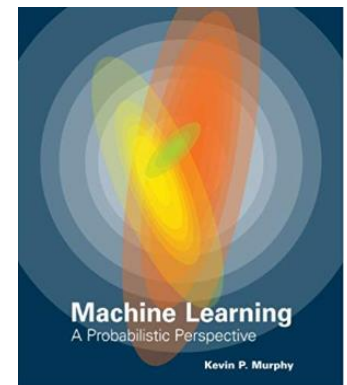
- Stanford CS231N 강의자료

■ 수업자료

- ecampus.kookmin.ac.kr 수업 게시판 공유 예정



MACHINE 기계 학습
LEARNING
오일석 지음
한빛아카데미



수업목표와 선수과목

■ 수업목표

- 기초 기계학습(machine learning)과 인공지능(artificial intelligence)의 이해
- 딥러닝(deep learning)의 포괄적인 이해
- 고급 소프트웨어 응용 실습 (Python with PyTorch) – **라이브코딩 도입 예정**
- 최신 인공지능 기술 흐름 파악
- 인공지능 관련 Kaggle 과제 수행

■ 선수과목

- 프로그래밍 (Python) [필수]
- 선형대수(linear algebra) 및 확률통계(probability and statistics), 정보이론(information theory)

학사일정

■ 총 15주 진행

SUN	MON	TUE	WED	THU	FRI	SAT
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

SUN	MON	TUE	WED	THU	FRI	SAT
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

SUN	MON	TUE	WED	THU	FRI	SAT
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

2019년 9월	
09.01(일)	제 73회 개교일
09.02(월)	2학기 개강일
09.02(월) - 09.06(금)	2학기 수강신청 변경/포기 기간
09.10(화) - 09.17(화)	부전공 신청 및 다전공 변경/포기 기간

2019년 10월	
10.14(월) - 10.17(목)	1전공 신청/변경 및 다전공 신청 기간
10.18(금)	개교 73주년 기념일 휴교일
10.21(월) - 10.25(금)	2학기 중간시험 기간 수업 8주차

2019년 12월	
12.04(수) - 12.06(금)	동계 계절학기 수강신청 기간
12.09(월) - 12.20(금)	전부(과) 신청 기간
12.09(월) - 12.13(금)	2학기 기말시험 기간 수업 15주차
12.10(화) - 12.12(목)	동계 계절학기 등록 기간
12.11(수) - 12.26(목)	2학기 성적 입력 기간

휴일 대체 보강 필수

- 환경 설정
 - PyTorch 기초
- 세부 일정 TBD

수업계획, 평가방법

주차	내용	참고	비고
1	과목 안내 및 환경 소개		Foundations
2	인공지능artificial intelligence 및 기계학습machine learning 소개	[과제 계획서 제출]	
3	기계학습과 수학 (e.g. 선형대수linear algebra)		
4	기계학습과 수학(e.g. 확률probability, 통계statistics, 정보이론information theory)		
5	신경망neural networks 기초 (e.g. 다층 퍼셉트론multilayer perceptron)		
6	딥러닝deep learning 기초	[HW 1 제출]	Deep learning
7	딥러닝 기초		
8	중간시험	[과제 초기보고서 제출]	
9	딥러닝 최적화optimization		
10	딥러닝 최적화		Advanced deep learning
11	비지도학습unsupervised learning	[HW 2 제출]	
12	준지도학습semi-supervised learning과 전이학습transfer learning		
13	순환신경망recurrent neural networks		
14	순환신경망	[HW 3 제출]	
15	기말시험	[과제 최종보고서 제출]	Wrap up

평가방법 (라이브 코딩 적용시)

숙제 3회 (24%[†])

중간/기말 2회 (20%*2=40%[†])

최종 프로젝트 (20%[†])

라이브코딩 (10%)

출석 (6%[†])

[†] 평가 비율

※ No negotiation for your final grade!!

딥러닝을 위한 라이브러리libraries와 프레임워크frameworks

theano

Popular!

PYTORCH



Popular!

TensorFlow

mxnet



Caffe

Software	Creator	Software license	Open source	Platform	Written in	Interface	OpenMP support	OpenCL support	CUDA support	Parallel execution (multi node)	Automatic differentiation	Has pretrained models	Recurrent nets	Convolutional nets	RBM/DBNs
Apache MXNet	Apache Software Foundation	Apache 2.0	Yes	Linux, macOS, Windows, AWS, Android, iOS, JavaScript	Small C++ core library	C++, Python, Julia, Matlab, JavaScript, Go, R, Scala, Perl	Yes	On roadmap	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Caffe	Berkeley Vision and Learning Center	BSD license	Yes	Linux, macOS, Windows	C++	Python, MATLAB, C++	Yes	Under development	Yes	Yes	Yes	Yes	Yes	Yes	
Chainer	Preferred Networks	MIT license	Yes	Linux, macOS, Windows		Python	No	No	Yes	Yes	Yes	Yes	Yes		
Keras	François Chollet	MIT license	Yes	Linux, macOS, Windows	Python	Python, R	Only if using Theano as backend	Under development for the Theano backend (and on roadmap for the TensorFlow backend)	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Microsoft Cognitive Toolkit	Microsoft Research	MIT license	Yes	Windows, Linux, (macOS via Docker on roadmap)	C++	Python (Keras), C++, Command line, BrainScript, (.NET on roadmap)	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
PyTorch	Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan	BSD license	Yes	Linux, macOS, Windows	Python, C, CUDA	Python	Yes	Via separately maintained package	Yes	Yes	Yes	Yes	Yes	Yes	Yes
TensorFlow	Google Brain team	Apache 2.0	Yes	Linux, macOS, Windows, Android	C++, Python, CUDA	Python (Keras), C/C++, Java, Go, R, Julia	No	On roadmap but already with SYCL support	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Theano	Université de Montréal	BSD license	Yes	Cross-platform	Python	Python (Keras)	Yes	Under development	Yes	Yes		Yes	Yes	Yes	Yes
Torch	Ronan Collobert, Koray Kavukcuoglu, Clement Farabet	BSD license	Yes	Linux, macOS, Windows, Android, iOS	C, Lua	Lua, LuaJIT, C, utility library for C++/OpenCL	Yes	Third party implementations	Yes	Through Twitter's Autograd	Yes	Yes	Yes	Yes	Yes

오늘 수업에는 뭐하지?

PREVIEW

■ 사람의 학습

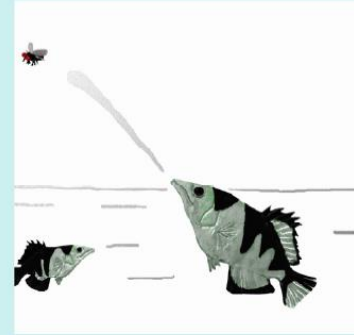
- 예) 수학, 과학, 역사뿐 아니라
- 수영, 자전거 타기 등 행위

■ 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? 혹은 기계도 인간처럼 생각할 수 있을까?
- 경험을 통해 점점 성능이 좋아지는 기계를 만들 수 있을까?
- 본 과목은 이 질문에 대한 답을 찾아가는 길을 함께 고민해보고자 함.

각 절에서 다루는 내용

- 1.1절: 기계 학습의 정의와 개념, 인공지능을 구현하는 도구로서의 역할을 설명한다.
- 1.2절: 특징 공간과 공간 변환을 소개한다.
- 1.3절: 데이터의 중요성과 희소성을 강조한다.
- 1.4절: 선형 회귀를 이용하여 기계 학습을 직관적으로 설명한다.
- 1.5절: 모델 선택의 중요성과 방법, 과소적합과 과잉적합을 설명한다.
- 1.6절: 현대 기계 학습에서 매우 중요한 규제 기법으로 데이터 확대와 가중치 감소를 간략히 기술한다.
- 1.7절: 기계 학습의 유형으로 지도 학습, 비지도 학습, 강화 학습, 준지도 학습을 소개한다.
- 1.8절: 기계 학습의 간략한 역사와 인공지능의 사회적 의미를 살펴본다.

1.1 기계 학습이란

- 1.1.1 기계 학습의 정의
- 1.1.2 지식기반 방식에서 기계 학습으로의 대 전환
- 1.1.3 기계 학습 개념
- 1.1.4 사람의 학습과 기계 학습 비교

1.1.1 기계 학습의 정의

■ 학습이란? <표준국어대사전>

“**경험**의 결과로 나타나는, 비교적 지속적인 행동의 변화나 그 잠재력의 **변화** 또는 **지식**을 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].”

1.1.1 기계 학습의 정의

■ 기계 학습이란?

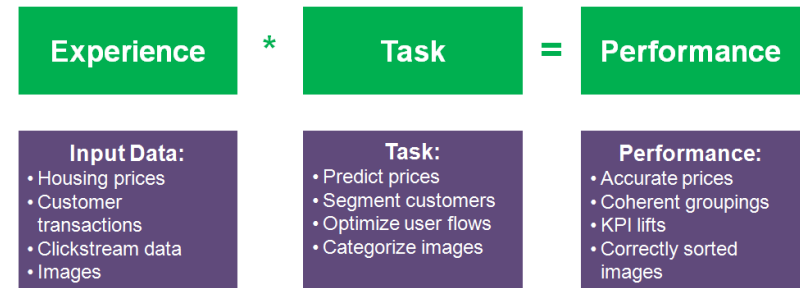
■ 현대적 정의

“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 **작업**을 수행한다. 이 프로그램의 **성능**을 P 라는 척도로 평가했을 때 **경험** E 를 통해 성능이 개선된다면 이 프로그램은 학습을 한다고 말할 수 있다[Mitchell1997(2쪽)].”

$$E * T = P$$

→ **최적의 알고리즘**을 찾는 행위

- **경험** E 를 통해
- 주어진 **작업** T 에 대한
- **성능** P 의 향상



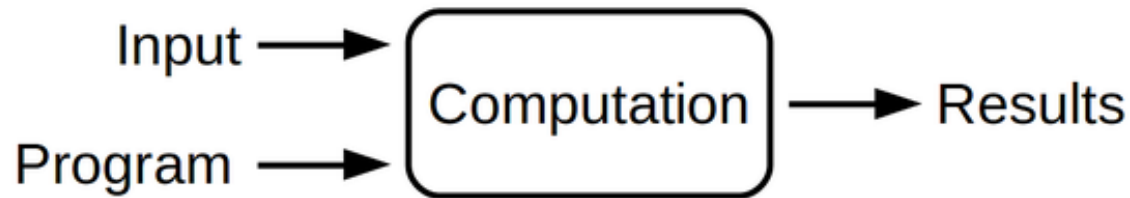
“Programming computers to optimize a performance criterion using example data or past experience 사례 데이터, 즉 과거 경험을 이용하여 성능 기준을 최적화하도록 프로그래밍하는 작업[Alpaydin2010]”

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

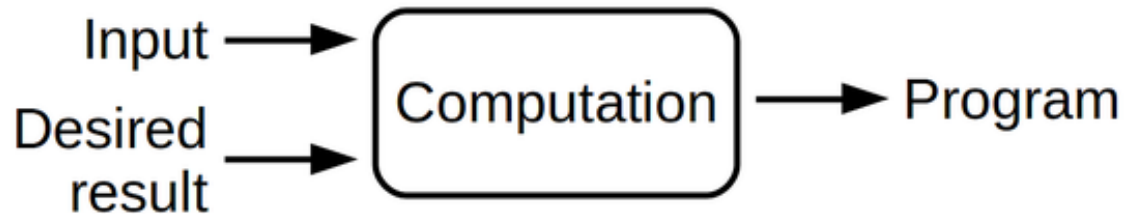
1.1.1 기계 학습의 정의

■ 기계 학습과 전통적인 프로그래밍의 비교

Traditional programming



Machine learning



1.1.2 지식기반 방식에서 기계 학습으로의 대전환

■ 인공지능의 탄생

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

■ 초창기는 **지식기반 방식**이 주류

- 지식기반: 경험적인 지식 혹은 사실을 인위적으로 컴퓨터에 부여하여 학습
- 예) “구멍이 2개이고 중간 부분이 홀쭉하며, 맨 위와 아래가 둥근 모양이라면 8이다”

1.1.2 지식기반 방식에서 기계 학습으로의 대전환

■ 큰 깨달음

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

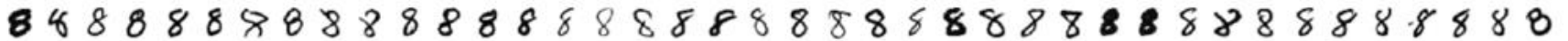


그림 1-2 인식 시스템이 대처해야 하는 심한 변화 양상(8과 단추라는 패턴을 어떻게 기술할 것인가?)

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

1.1.2 지식기반 방식에서 기계 학습으로의 대전환

■ 인공지능의 주도권 전환

- 지식기반 → 기계 학습
- 기계 학습: 데이터 중심 접근방식



그림 1-3 기계 학습으로 만든 최첨단 인공지능 제품들

1.1.3 기계 학습 개념

■ 간단한 기계 학습 예제

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

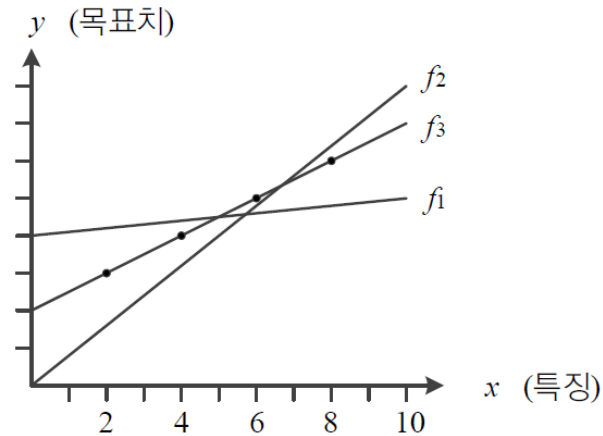


그림 1-4 간단한 기계 학습 예제

■ 예측prediction 문제

- 임의의 시간이 주어지면 이때 이동체의 위치는?
- 회귀regression 문제와 분류classification 문제로 나뉨
 - 회귀는 목표치가 실수, 분류는 부류 혹은 종류의 값 ([그림 1-4]는 회귀 문제)

1.1.3 기계 학습 개념

■ 훈련집합(training set)

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

$$\text{훈련집합: } \mathbb{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}, \quad \mathbb{Y} = \{y_1, y_2, \dots, y_n\} \quad (1.1)$$

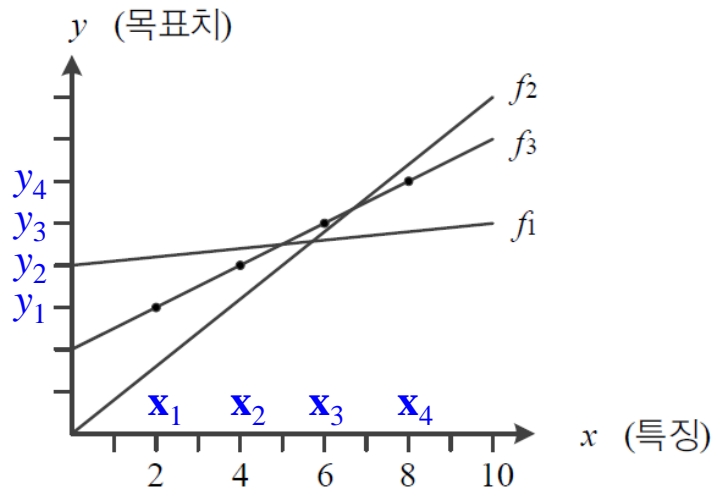


그림 1-4 간단한 기계 학습 예제

[그림 1-4] 예제의 훈련집합

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

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

1.1.3 기계 학습 개념

■ 관찰된 데이터들을 어떻게 설명할 것인가?

- 눈대중으로 보면 직선을 이루므로 직선을 선택하자 → 모델로 직선을 선택을 가정
- 직선 모델의 수식
 - 2개의 매개변수 w 와 b

$$y = \underline{w}x + \underline{b} \quad (1.2)$$

■ 기계 학습은

- 가장 정확하게 예측할 수 있는, 즉 최적의 매개변수를 찾는 작업
- 처음에는 최적값을 모르므로 임의의 값에서 시작하고, 점점 성능을 개선하여 최적에 도달
- [그림 1-4]의 예에서는 f_1 에서 시작하여 $f_1 \rightarrow f_2 \rightarrow f_3$
 - 최적인 f_3 은 $w=0.5$ 와 $b=2.0$

1.1.3 기계 학습 개념

■ 학습을 마치면,

- 새로운^{unknown} 특징에 대응되는 목표치의 예측에 사용
- 예) 10.0 순간의 이동체 위치를 알고자 하면, $f_3(10.0)=0.5*10.0+2.0=7.0$ 이라 예측함

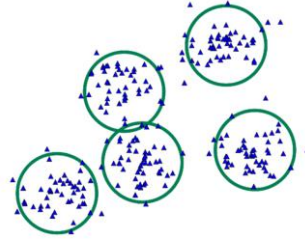
■ 기계 학습의 궁극적인 목표

- 훈련집합에 없는 새로운 샘플에 대한 오류를 최소화 (새로운 샘플 집합: 테스트 집합^{test set})
- 테스트 집합에 대한 높은 성능을 일반화^{generalization} 능력이라 부름

1.1.3 기계 학습 개념

■ 기계학습의 필수요소

- 내, 외부적 규칙 존재



- 수학적으로 설명 불가능



- 데이터가 있어야 함



1.1.4 사람의 학습과 기계 학습

표 1-1 사람의 학습과 기계 학습의 비교

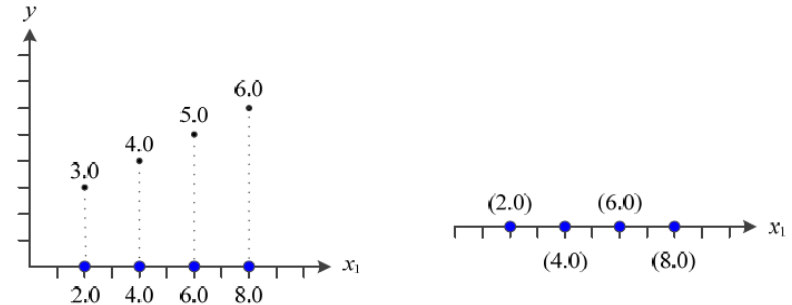
기준	사람의 학습	기계 학습
학습 과정	능동적	수동적
데이터 형식	자연에 존재하는 그대로	일정한 형식에 맞추어 사람이 준비함
동시에 학습 가능한 과업 수	자연스럽게 여러 과업을 학습	하나의 과업만 가능
학습 원리에 대한 지식	매우 제한적으로 알려져 있음	모든 과정이 밝혀져 있음
수학 의존도	매우 낮음	매우 높음
성능 평가	경우에 따라 객관적이거나 주관적	객관적(수치로 평가, 예를 들어 정확률 99.8%)
역사	수백만 년	60년 가량

1.2 특징 공간에 대한 이해

- 1.2.1 1차원과 2차원 특징 공간
- 1.2.2 다차원 특징 공간
- 1.2.3 특징 공간 변환과 표현 학습

1.2.1 1차원과 2차원 특징 공간

■ 1차원 특징 공간



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

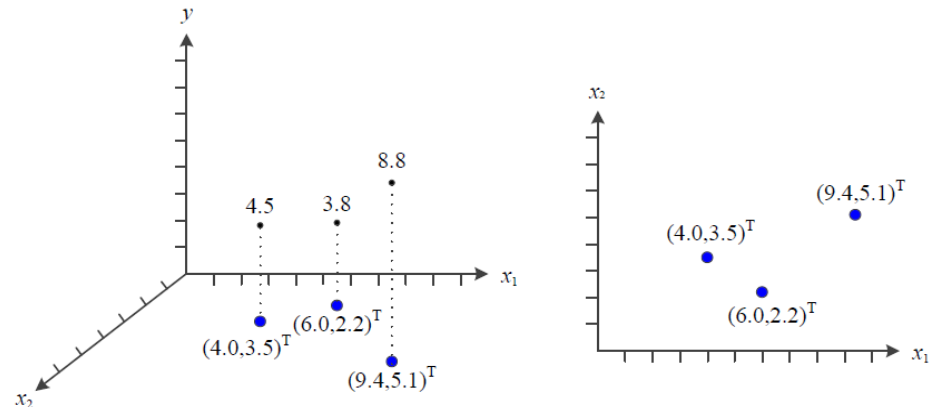
■ 2차원 특징 공간

■ 특징 벡터 표기

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

■ 예시

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

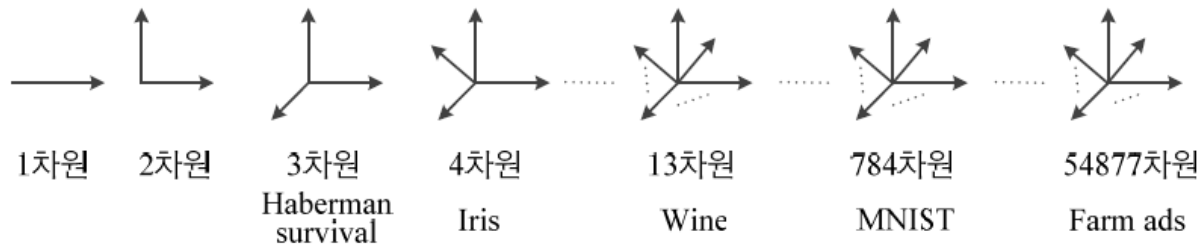


(b) 2차원 특징 공간(왼쪽: 특징 벡터와 목표값을 축으로 표시, 오른쪽: 특징 벡터만 축으로 표시)

그림 1-5 특징 공간과 데이터의 표현

1.2.2 다차원 특징 공간

■ 다차원 특징 공간 예제



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

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

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

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

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

그림 1-6 다차원 특징 공간

1.2.2 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 = $d+1$

$$y = \underline{w_1}x_1 + \underline{w_2}x_2 + \dots + \underline{w_d}x_d + \underline{b} \quad (1.3)$$

- 2차 곡선 모델을 사용하면 매개변수 수가 크게 증가

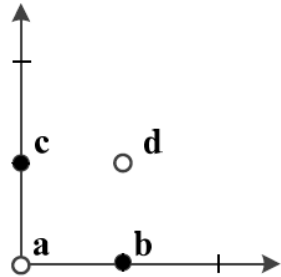
- 매개변수 수 = d^2+d+1
- 예) Iris 데이터: $d=4$ 이므로 21개의 매개변수
- 예) MNIST 데이터: $d=784$ 이므로 615,441개의 매개변수

$$y = \underline{w_1}x_1^2 + \underline{w_2}x_2^2 + \dots + \underline{w_d}x_d^2 + \underline{w_{d+1}}x_1x_2 + \dots + \underline{w_{d^2}}x_{d-1}x_d + \underline{w_{d^2+1}}x_1 \\ + \dots + \underline{w_{d^2+d}}x_d + \underline{b} \quad (1.5)$$

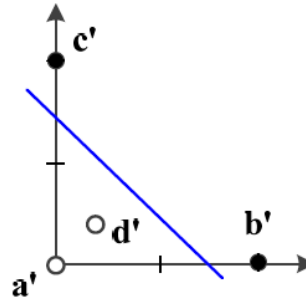
1.2.3 특징 공간 변환과 표현 문제

■ 선형 분리 불가능 linearly non-separable 한 원래 특징 공간 ([그림 1-7(a)])

- 직선 모델을 적용하면 75% 정확도 한계



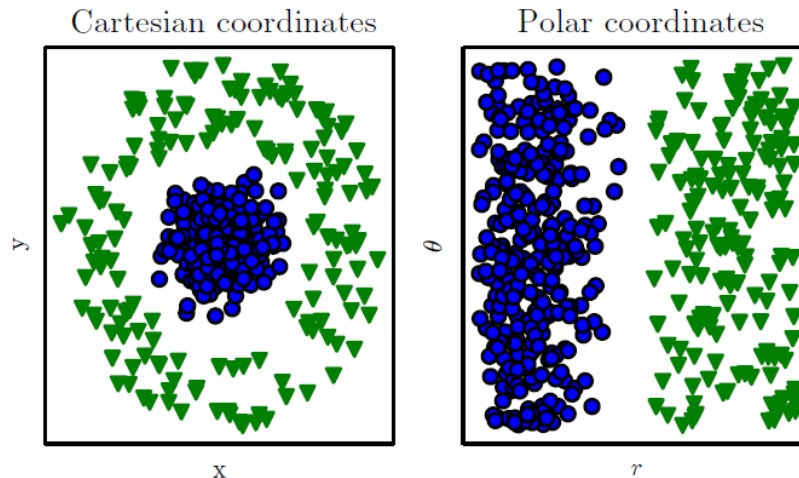
(a) 원래 특징 공간



(b) 분류에 더 유리하도록 변환된 새로운 특징 공간

그림 1-7 특징 공간 변환

■ 표현 문제 representations matter의 예



1.2.3 특징 공간 변환과 표현 문제

■ 식 (1.6)으로 변환된 새로운 특징 공간 ([그림 1-7(b)])

- 공간 변환을 통해 직선 모델로 100% 정확도

$$\text{원래 특징 벡터 } \mathbf{x} = (x_1, x_2)^T \rightarrow \text{변환된 특징 벡터 } \mathbf{x}' = \left(\frac{x_1}{2x_1x_2 + 0.5}, \frac{x_2}{2x_1x_2 + 0.5} \right)^T \quad (1.6)$$

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

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

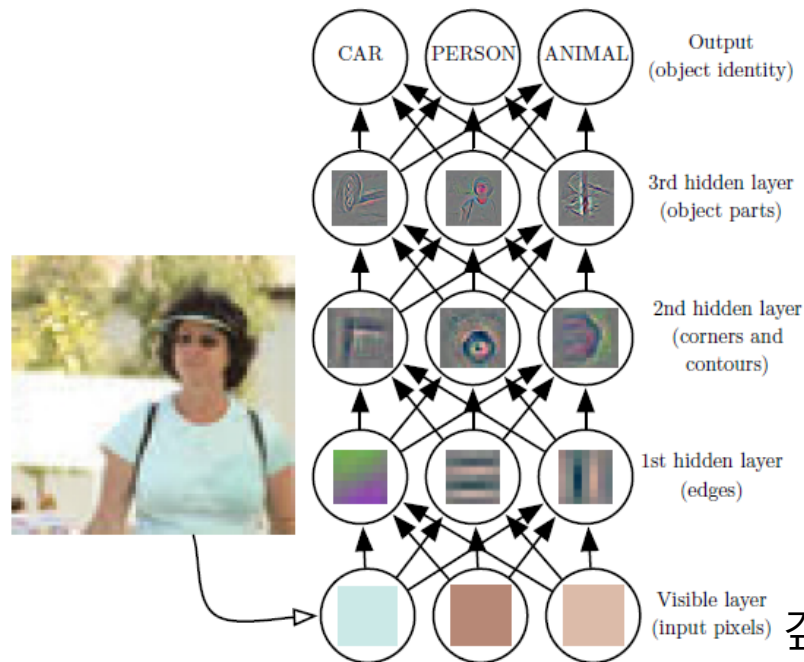
$$\mathbf{c} = (0,1)^T \rightarrow \mathbf{c}' = (0,2)^T$$

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

1.2.3 표현 학습

■ 표현 학습 representation learning

- 좋은 특징 공간을 **자동**으로 찾는 작업
- [그림 1-7]은 표현 학습을 사람이 **직관**으로 수행한 셈
- **딥러닝** deep learning
 - 다수의 은닉층을 가진 신경망을 이용하여 **최적의 계층적인 특징 공간을 찾아냄**
 - 아래쪽 은닉층은 저급 특징(선, 구석점 등), 위쪽 고급 특징(얼굴, 바퀴 등) 추출



딥러닝 모델의 예시 그림
깊이 (층): 반복되는 구성 단위

1.2.3 특징 공간 변환과 표현 학습

■ 차원에 대한 몇 가지 설명

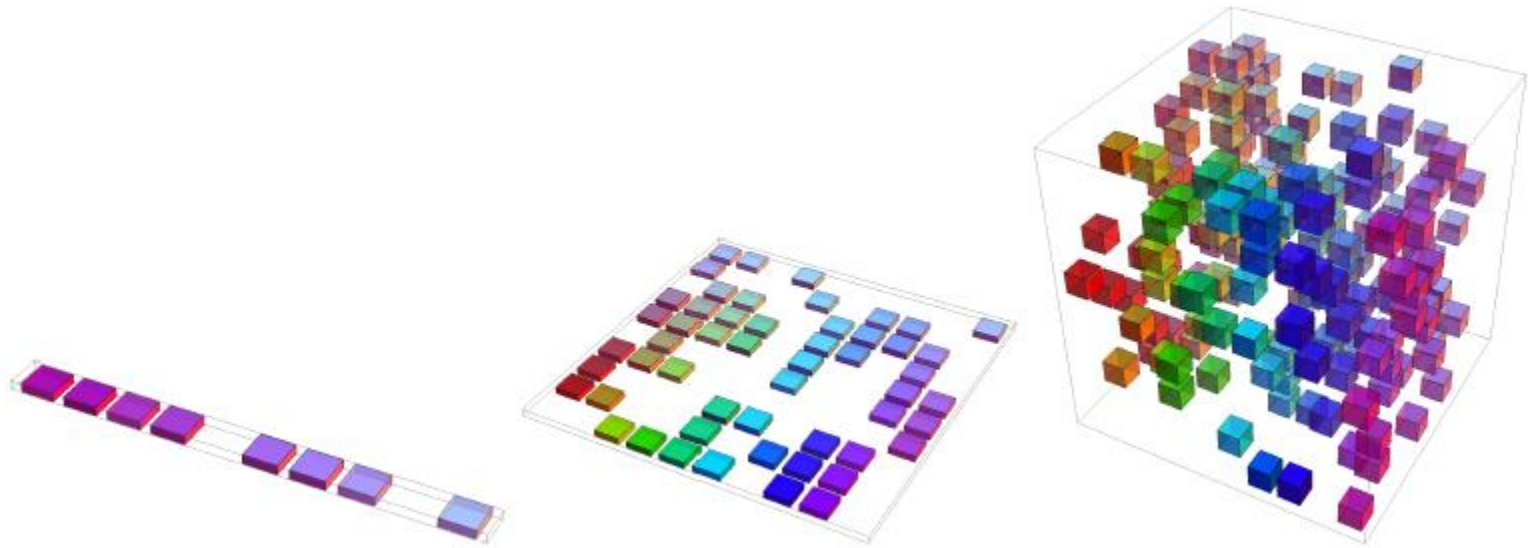
- 거리: 차원에 무관하게 수식 적용 가능함
 - 예) 두 점 $\mathbf{a}=(a_1, a_2, \dots, a_d)^T$ 와 $\mathbf{b}=(b_1, b_2, \dots, b_d)^T$ 사이의 **거리**는 모든 d 에 대해 성립

$$\text{dist}(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{i=1}^d (a_i - b_i)^2} \quad (1.7)$$

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

1.2.3 특징 공간 변환과 표현 학습

- 차원의 저주 curse of dimensionality (=number of features)
 - 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
 - 1차, 2차, 3차원에서의 차원의 저주 예시



1.3 데이터에 대한 이해

- 1.3.1 데이터 생성 과정
- 1.3.2 데이터의 중요성
- 1.3.3 데이터베이스 크기와 기계 학습 성능
- 1.3.4 데이터 가시화

1.3 데이터에 대한 이해

■ 과학 기술의 정립 과정

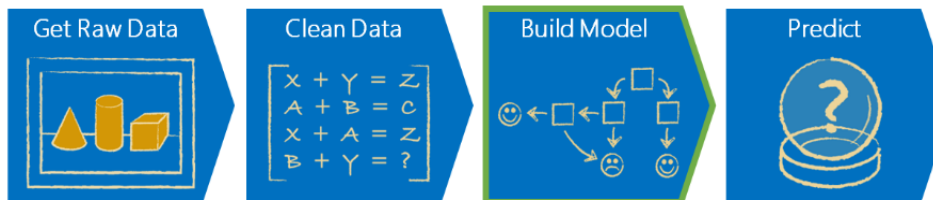


그림 1-8 과학기술의 발전 과정

- 예) 튀코 브라헤는 천동설이라는 틀린 모델을 선택함하여 수집한 데이터를 설명하지 못함.
하지만, 케플러는 지동설 모델을 도입하여 제1, 제2, 제 3법칙을 완성함

■ 기계 학습

- 기계 학습이 푸는 문제는 훨씬 복잡함
 - 예) [그림 1-2]의 '8' 숫자 패턴과 '단추' 패턴의 다양한 변화 양상
- 단순한 수학 공식으로 표현 불가능함
- 데이터로부터 자동으로 학습 모델을 찾아내는 과정이 필수적임 → 데이터 수집 및 전처리 중요



1.3.1 데이터 생성 과정

■ 데이터 생성 과정을 완전히 아는 인위적 상황의 예제 (가상)

- 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

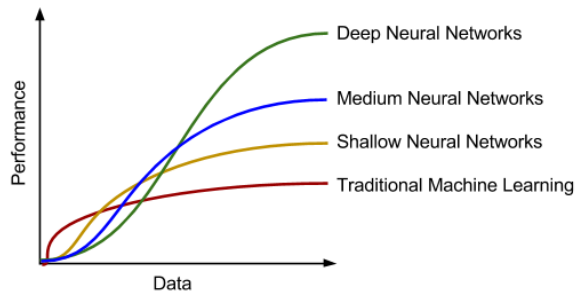
■ [그림 1-6]과 같은 실제 기계 학습 문제 (현실)

- 데이터 생성 과정을 알 수 없음
- 단지 주어진 **훈련집합** \mathbb{X}, \mathbb{Y} 로 예측 모델 또는 생성 모델을 **근사 추정**만 가능

1.3.2 데이터의 중요성

■ 데이터의 양과 질

- 주어진 응용에 맞는 충분히 다양한 데이터를 **충분한 양**만큼 수집 → **추정 정확도 높아짐**
 - 예) 정면 얼굴만 가진 데이터로 학습하고 나면, 기운 얼굴은 매우 낮은 성능
→ 주어진 응용 환경을 자세히 살핀 다음 그에 맞는 데이터 확보는 아주 중요함
- 데이터의 양과 학습 모델의 성능 경향성 비교



■ 공개 데이터

- 기계 학습의 초파리로 여겨지는 3가지 데이터베이스: Iris, MNIST, ImageNet
- 위키피디아(wikipedia)에서 'list of datasets for machine learning research'로 검색
- UCI 리퍼지토리(repository) (2017년11월 기준으로 394개 데이터베이스 제공)

1.3.2 데이터의 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



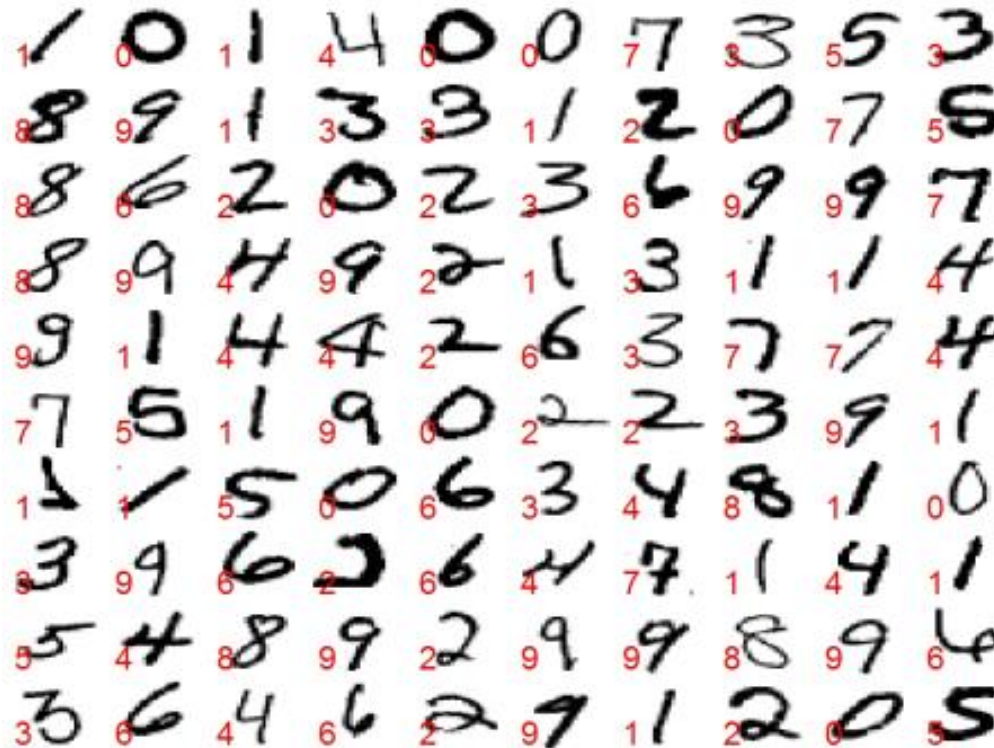
Versicolor



Virginica

1.3.2 데이터의 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.



1.3.2 데이터의 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

1.3.3 데이터베이스 크기와 기계 학습 성능

■ 데이터베이스의 왜소한 크기 → 차원의 저주와 관련

- 예) MNIST: 28*28 흑백 비트맵이라면 서로 다른 총 샘플 수는 2^{784} 가지이지만, MNIST는 고작 6만 개 샘플

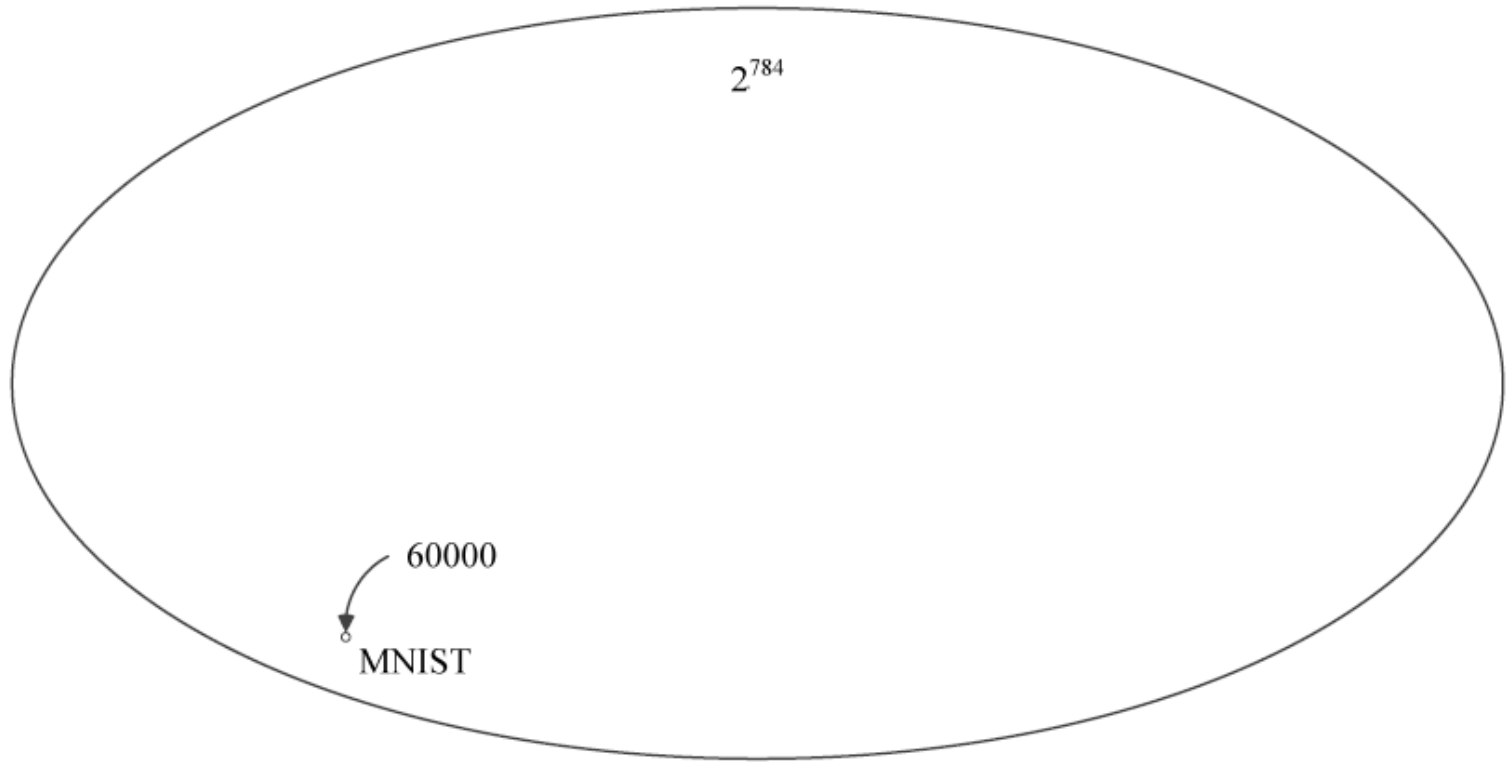


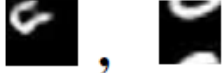
그림 1-9 방대한 특징 공간과 희소한 데이터베이스

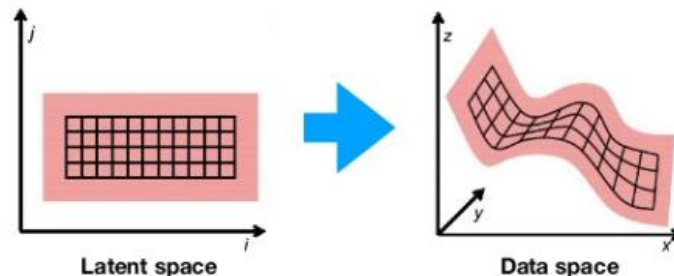
1.3.3 데이터베이스 크기와 기계 학습 성능


■ 왜소한 양의 데이터베이스로 어떻게 높은 성능을 달성하는가?

- 방대한 공간에서 실제 데이터가 발생하는 곳은 매우 작은 부분 공간임

→ 데이터 희소(data sparsity) 특성 가정

-  와 같은 샘플의 발생 확률은 거의 0 가까움
- 매니폴드 가정(manifold assumption (or manifold hypothesis))
 - 대부분의 실제 고차원 데이터집합이 더 낮은 저차원 매니폴드에 가깝게 놓여 있음



-  와 같이 일정한 규칙에 따라 매끄럽게 변화

1.3.4 데이터 가시화

- 4차원 이상의 초공간^{hyperplane}은 한꺼번에 가시화^{visualization} 불가능
- 여러 가지 가시화 기법
 - 2개씩 조합하여 여러 개의 그래프 그림

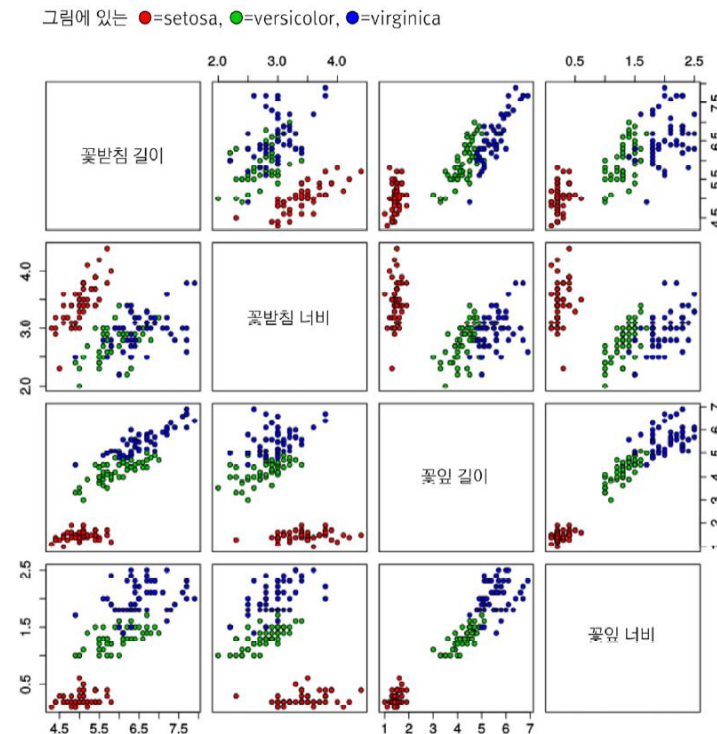


그림 1-10 고차원 특징 공간의 가시화

- 고차원 공간을 저차원으로 변환하는 기법들 (6.6.1절)

1.4 간단한 기계 학습의 예

■ 선형 회귀 linear regression 문제

- [그림 1-4]: 식 (1.2)의 직선 모델을 사용하므로 두 개의 매개변수 $\theta = (w, b)^T$

$$y = wx + b \quad (1.2)$$

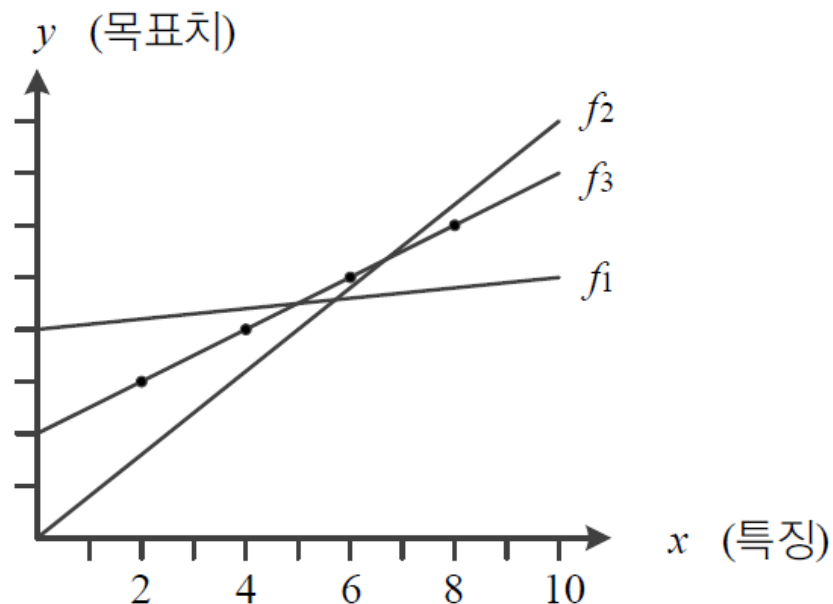


그림 1-4 간단한 기계 학습 예제

1.4 간단한 기계 학습의 예

■ 목적 함수objective function (또는 비용 함수cost function)

- 식 (1.8)은 선형 회귀를 위한 목적 함수
 - $f_{\theta}(\mathbf{x}_i)$ 는 예측함수의 **예측 출력**, y_i 는 예측함수가 맞추어야 하는 **실제 목표치**
 - $f_{\theta}(\mathbf{x}_i) - y_i$ 는 **오차error** 혹은 **손실loss**
 - 식 (1.8)을 **평균제곱오차MSE(mean squared error)**라 부름

$$J(\Theta) = \frac{1}{n} \sum_{i=1}^n (f_{\theta}(\mathbf{x}_i) - y_i)^2 \quad (1.8)$$

- 처음에는 최적 매개변수 값을 알 수 없으므로 임의의 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 $J(\theta_1) > J(\theta_2) > J(\theta_3)$

1.4 간단한 기계 학습의 예

■ [예제 1-1]

■ 훈련집합

$$\mathbb{X} = \{x_1 = (2.0), x_2 = (4.0), x_3 = (6.0), x_4 = (8.0)\},$$

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

■ 초기 직선의 매개변수 $\theta_1 = (0.1, 4.0)^T$ 라 가정

$$\mathbf{x}_1, y_1 \rightarrow (f_{\theta_1}(2.0) - 3.0)^2 = ((0.1 * 2.0 + 4.0) - 3.0)^2 = 1.44$$

$$\mathbf{x}_2, y_2 \rightarrow (f_{\theta_1}(4.0) - 4.0)^2 = ((0.1 * 4.0 + 4.0) - 4.0)^2 = 0.16$$

$$\mathbf{x}_3, y_3 \rightarrow (f_{\theta_1}(6.0) - 5.0)^2 = ((0.1 * 6.0 + 4.0) - 5.0)^2 = 0.16$$

$$\mathbf{x}_4, y_4 \rightarrow (f_{\theta_1}(8.0) - 6.0)^2 = ((0.1 * 8.0 + 4.0) - 6.0)^2 = 1.44$$

$$\longrightarrow J(\theta_1) = 0.8$$

1.4 간단한 기계 학습의 예

■ [예제 1-1] (계속)

- θ_1 을 개선하여 $\theta_2 = (0.8, 0.0)^T$ 가 되었다고 가정

$$\mathbf{x}_1, y_1 \rightarrow (f_{\theta_2}(2.0) - 3.0)^2 = ((0.8 * 2.0 + 0.0) - 3.0)^2 = 1.96$$

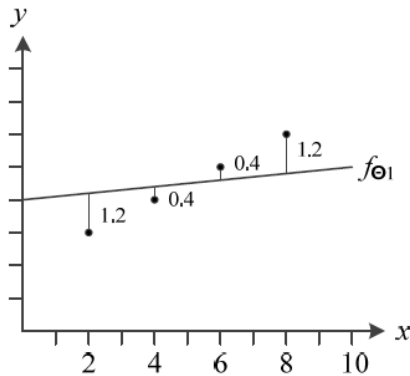
$$\mathbf{x}_2, y_2 \rightarrow (f_{\theta_2}(4.0) - 4.0)^2 = ((0.8 * 4.0 + 0.0) - 4.0)^2 = 0.64$$

$$\mathbf{x}_3, y_3 \rightarrow (f_{\theta_2}(6.0) - 5.0)^2 = ((0.8 * 6.0 + 0.0) - 5.0)^2 = 0.04$$

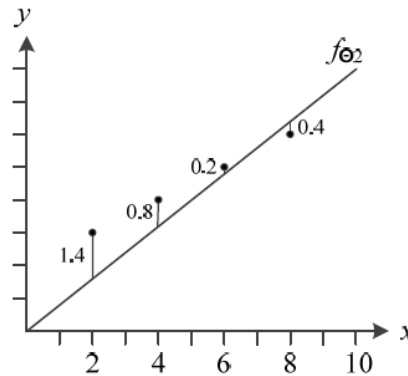
$$\mathbf{x}_4, y_4 \rightarrow (f_{\theta_2}(8.0) - 6.0)^2 = ((0.8 * 8.0 + 0.0) - 6.0)^2 = 0.16$$

$$\longrightarrow J(\theta_2) = 0.7$$

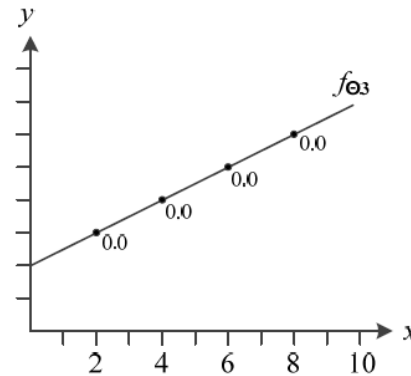
- 다음으로 θ_2 를 개선하여 $\theta_3 = (0.5, 2.0)^T$ 가 되었다고 가정
- 이때 $J(\theta_3) = 0.0$ 이 되어 θ_3 은 최적값 $\hat{\theta}$ 이 됨



(a) 초기 매개변수 θ_1



(b) θ_1 을 개선하여 θ_2 가 됨

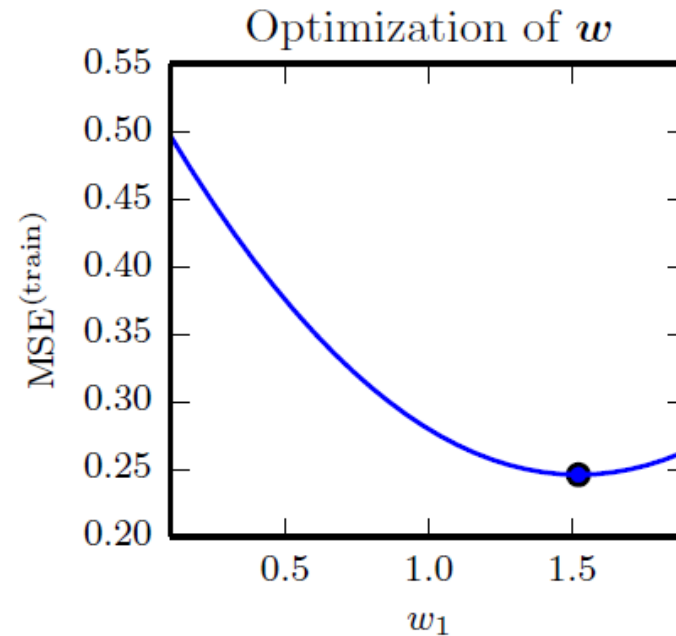
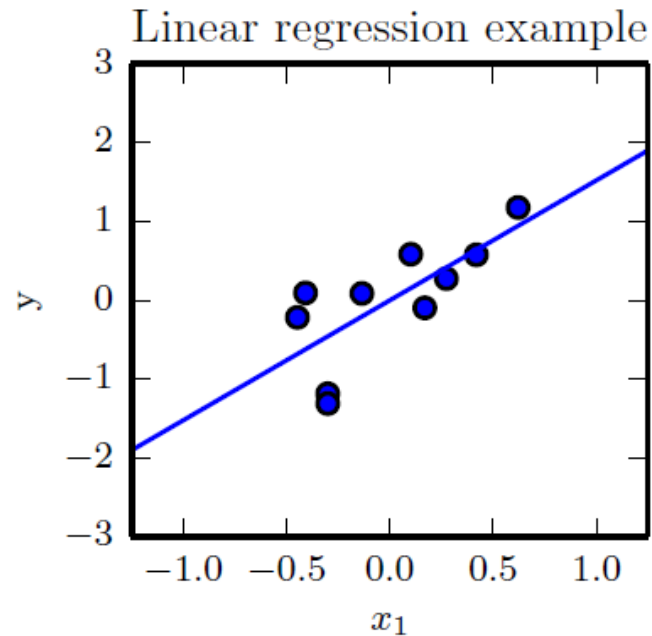


(c) θ_2 를 개선하여 최적의 θ_3 을 찾음

그림 1-11 기계 학습에서 목적함수의 역할

1.4 간단한 기계 학습의 예

■ 선형 회귀 문제와 매개변수 최적화 관계의 예



1.4 간단한 기계 학습의 예

- 기계 학습이 할 일을 공식화하면,

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmin}} J(\Theta) \quad (1.9)$$

- 기계 학습은 작은 개선을 반복하여 최적의 해를 찾아가는 수치적 방법으로 식 (1.9)를 풀

- 알고리즘 형식으로 쓰면,

알고리즘 1-1 기계 학습 알고리즘

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   $\hat{\Theta} = \theta_t$ 
```

1.4 간단한 기계 학습의 예

■ 좀더 현실적인 상황

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

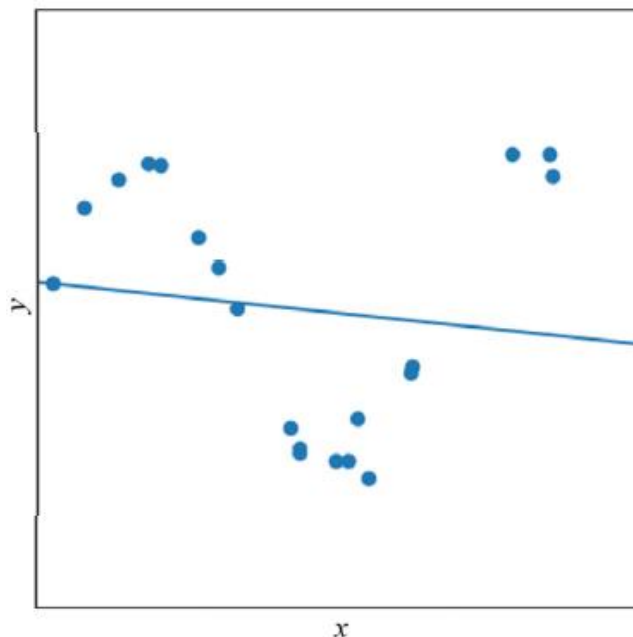


그림 1-12 선형 모델의 한계

1.4 간단한 기계 학습의 예

■ 기계학습 요소

■ 카드 승인 예제

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
...	...



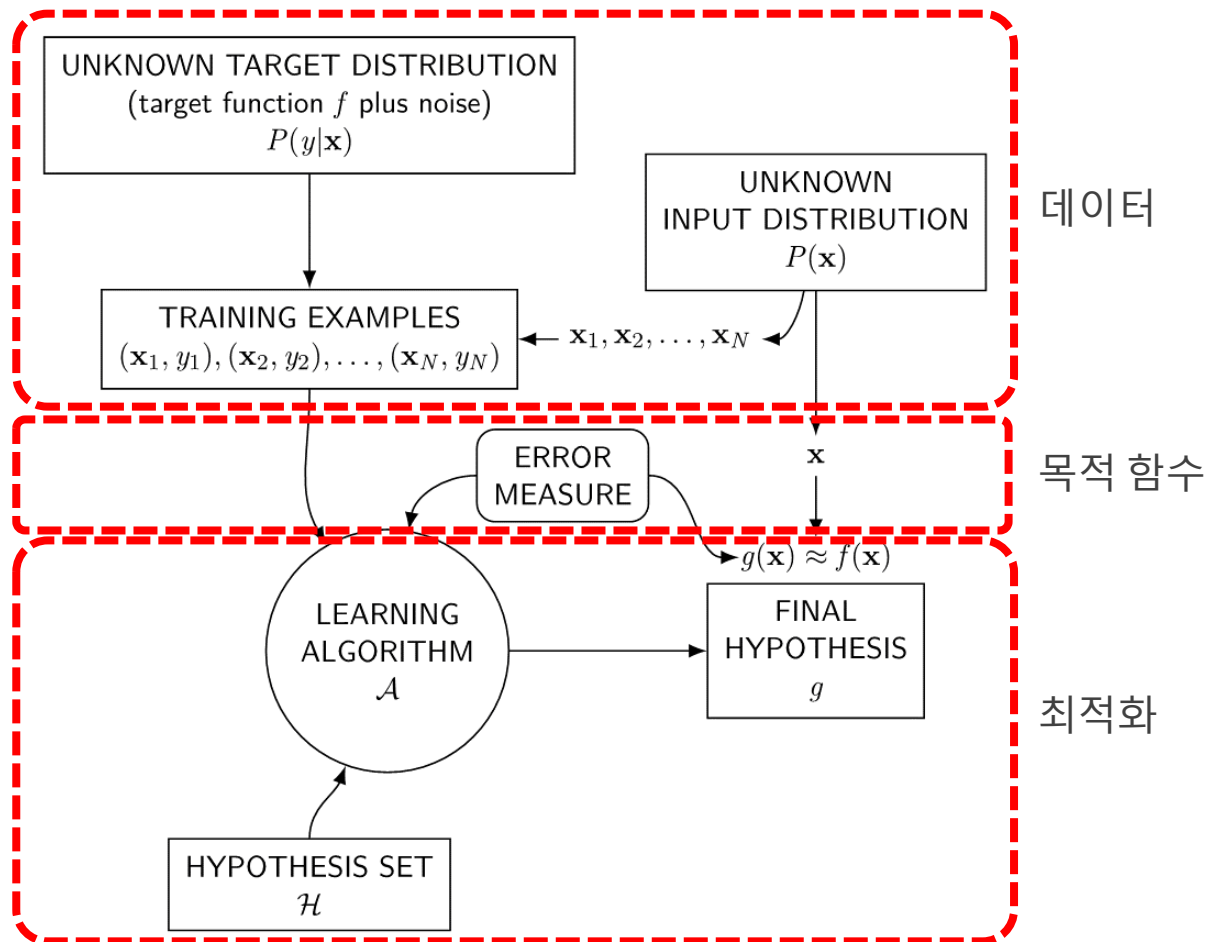
■ 요소

요소	기호	카드 승인 예제
input	x	customer application
output	y	approve or deny
target distribution	$f = P(y x)$	ideal credit approval formula
data	$(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$	historical records
hypothesis	$g: \mathcal{X} \rightarrow \mathcal{Y}$	formula to be used

1.4 간단한 기계 학습의 예

■ 기계학습 설정

- 교사학습 경우,



1.5 모델 선택

- 1.5.1 과소적합과 과잉적합
- 1.5.2 바이어스와 분산
- 1.5.3 검증집합과 교차검증을 이용한 모델 선택 알고리즘
- 1.5.4 모델 선택의 한계와 현실적인 해결책

1.5.1 과소적합과 과잉적합

■ [그림 1.13]의 1차 모델은 과소적합underfitting

- 모델의 '용량이 작아' 오차가 클 수밖에 없는 현상

■ 비선형 모델을 사용하는 대안

- [그림 1-13]의 2차, 3차, 4차, 12차는 다항식 곡선을 선택한 예
- 1차(선형)에 비해 오차가 크게 감소함

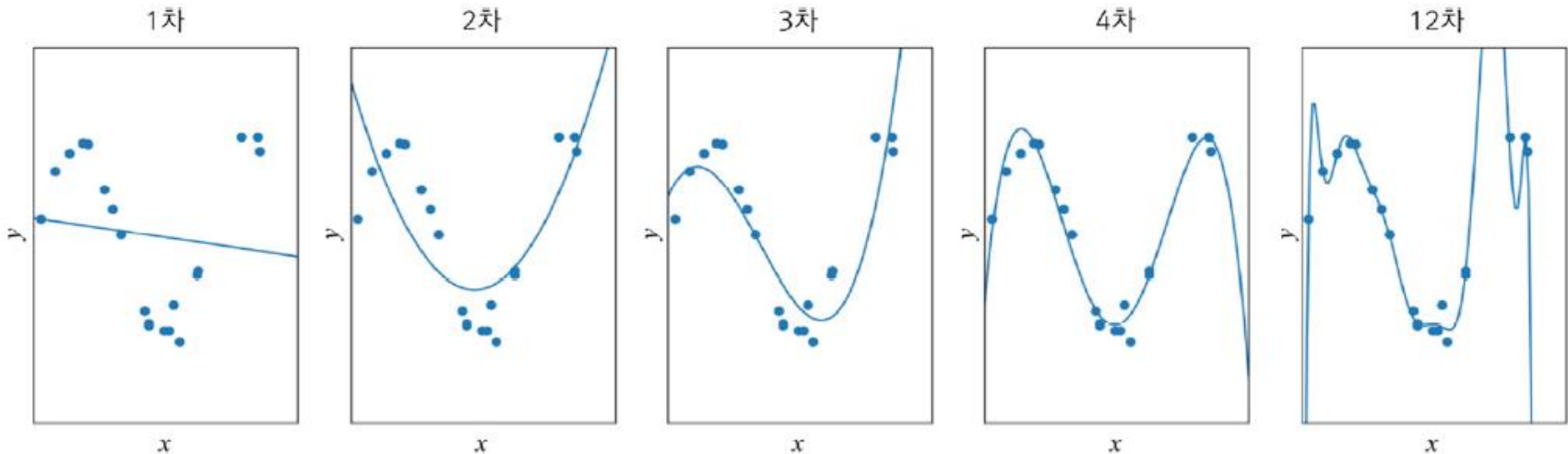


그림 1-13 과소적합과 과잉적합 현상

1.5.1 과소적합과 과잉적합

■ 과잉적합overfitting

- 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 '모델의 용량capacity이 크기' 때문. 학습 과정에서 잡음까지 수용 → 과잉적합 현상
- 훈련집합에 과몰입해서 단순 암기했기 때문임
- 적절한 용량의 모델을 선택하는 모델 선택 작업이 필요함

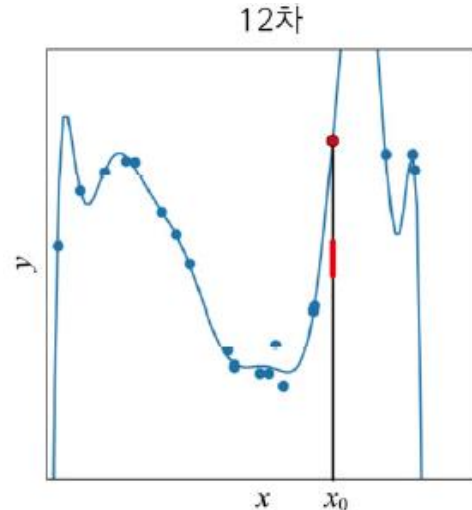


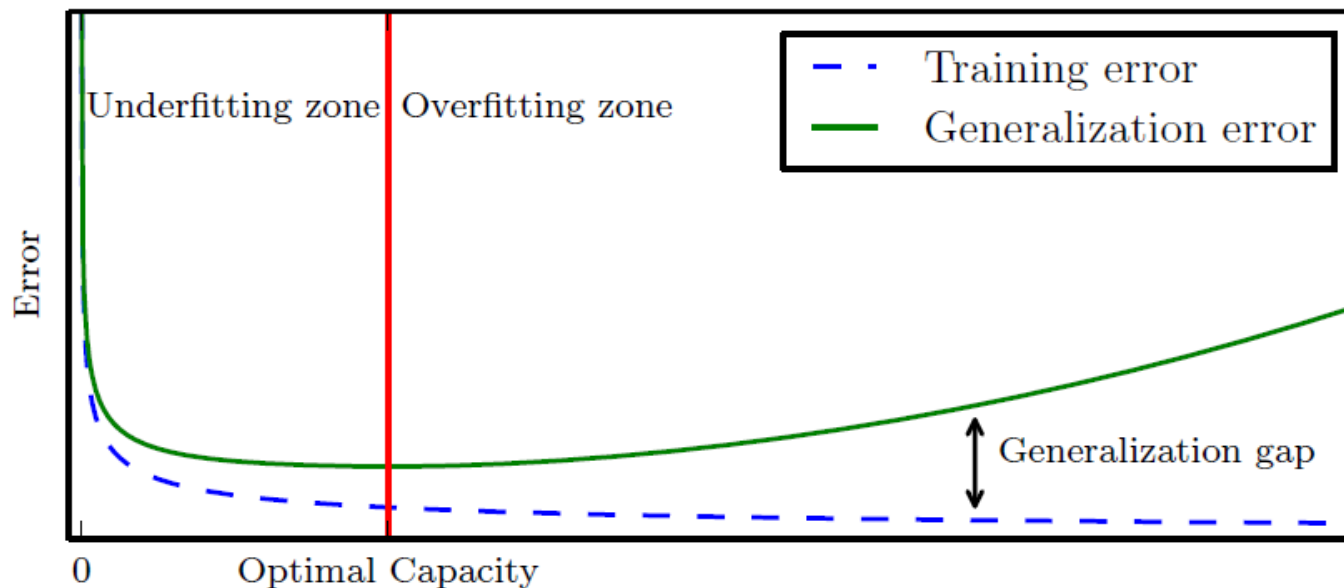
그림 1-14 과잉적합되었을 때 부정확한 예측 현상

1.5.1 과소적합과 과잉적합

■ 1차~12차 다항식 모델의 비교 관찰

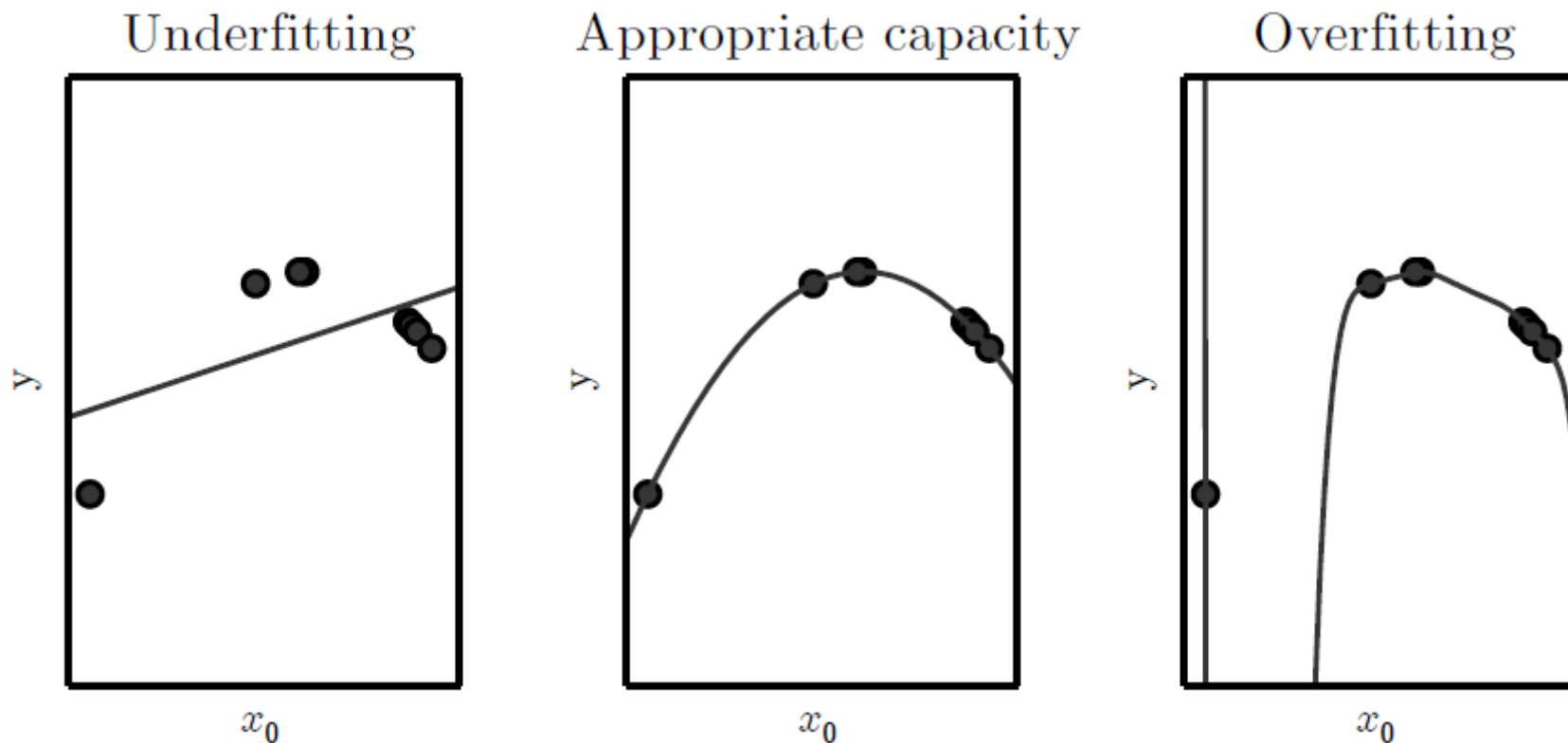
- 1~2차는 훈련집합과 테스트집합 모두 낮은 성능: 과소적합
- 12차는 훈련집합에 높은 성능을 보이나 테스트집합에서는 낮은 성능 → 낮은 일반화 능력: 과잉적합
- 3~4차는 훈련집합에 대해 12차보다 낮겠지만 테스트집합에는 높은 성능 → 높은 일반화 능력: 적합 모델 선택

■ 모델의 일반화 능력과 용량 관계



1.5.1 과소적합과 과잉적합

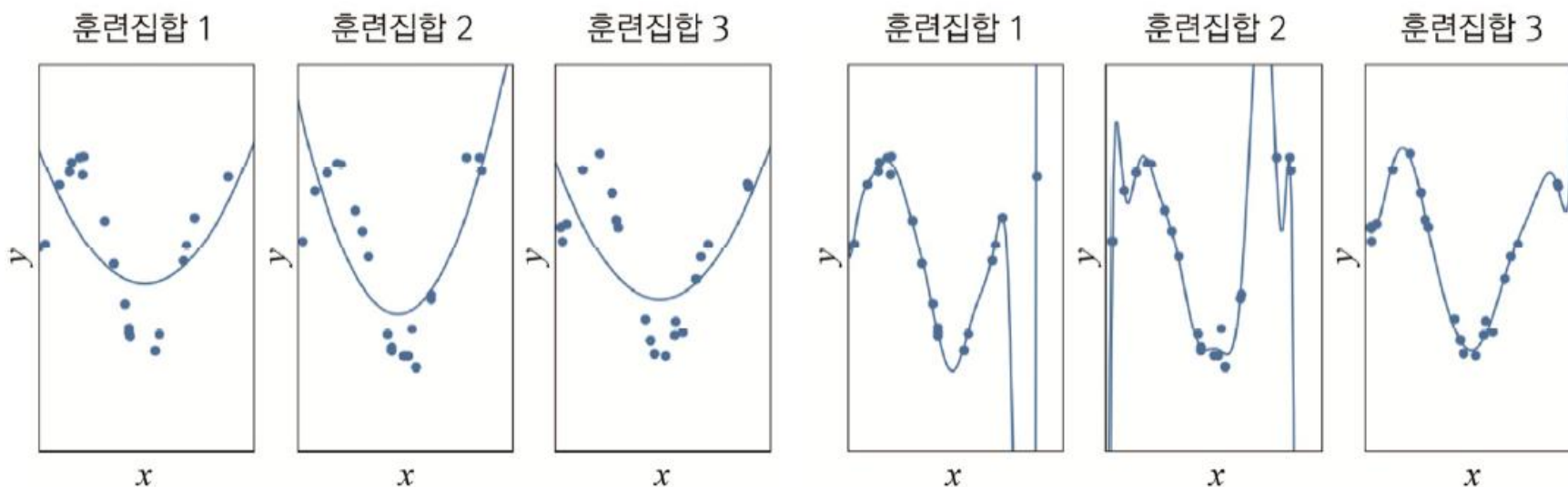
- 훈련집합에 대한 세가지 모델 적합도 예



1.5.2 바이어스^{bias}와 분산^{variance}

■ 훈련집합을 여러 번 수집하여 1차~12차에 적용하는 실험

- 2차는 매번 큰 오차 → 바이어스가 큼. 하지만 비슷한 모델을 얻음 → 낮은 분산
- 12차는 매번 작은 오차 → 바이어스가 작음. 하지만 크게 다른 모델을 얻음 → 높은 분산
- 일반적으로 용량이 작은 모델은 바이어스는 크고 분산은 작음
복잡한 모델은 바이어스는 작고 분산은 큼
- 바이어스와 분산은 트레이드오프^{trade-off} 관계



(a) 2차 모델(바이어스는 크고, 분산은 작음)

(b) 12차 모델(바이어스는 작고, 분산은 큼)

그림 1-15 모델의 바이어스와 분산 특성

1.5.2 바이어스와 분산

■ 기계 학습의 목표

- 낮은 바이어스와 낮은 분산을 가진 예측 모델을 만드는 것이 목표. 즉 왼쪽 아래 상황

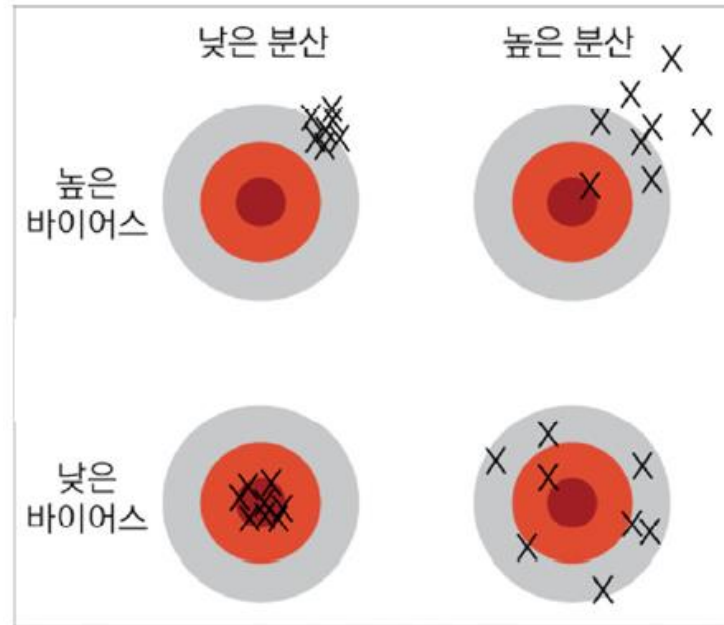


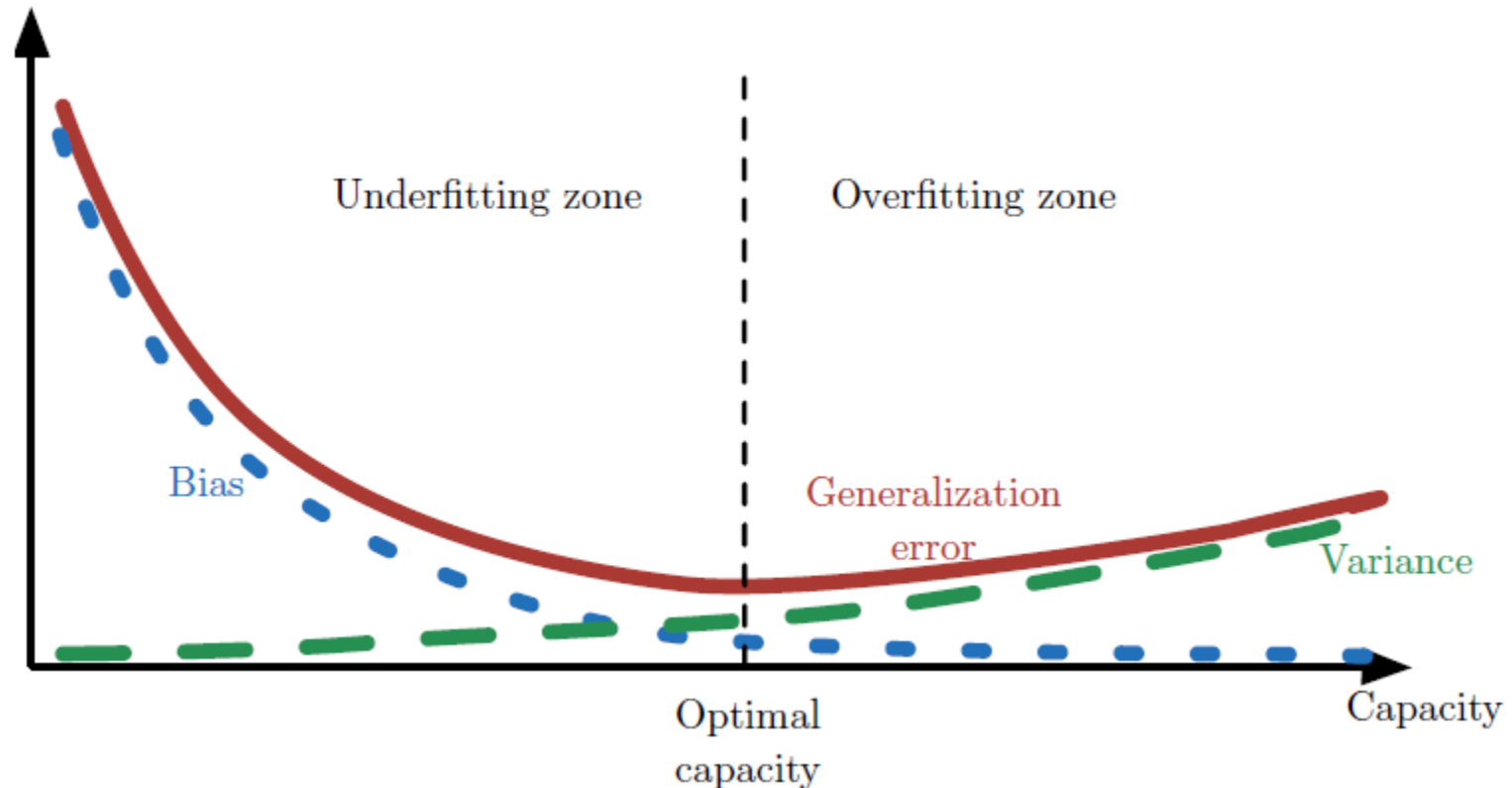
그림 1-16 바이어스와 분산

- 하지만 바이어스와 분산은 트레이드오프 관계
- 따라서 바이어스 희생을 최소로 유지하며 분산을 최대한 낮추는 전략 필요

1.5.2 바이어스와 분산

■ 바이어스와 분산의 관계

- 용량 증가 → 바이어스 감소, 분산 증가 경향
- 일반화 오차 성능은 U형의 곡선을 가짐



1.5.3 검증집합과 교차검증을 이용한 모델 선택 알고리즘

■ 검증집합을 이용한 모델 선택

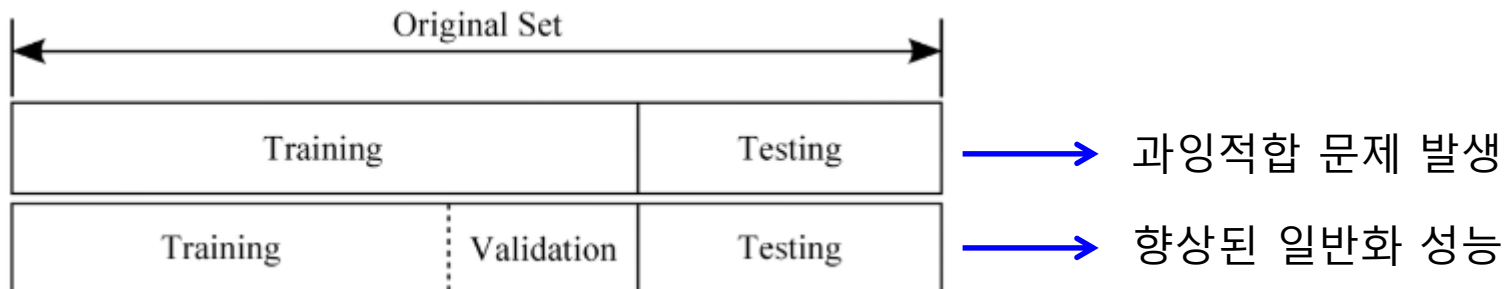
- 훈련집합과 테스트집합과 다른 별도의 검증집합(validation set)을 가진 상황

알고리즘 1-2 검증집합을 이용한 모델 선택

입력: 모델집합 Ω , 훈련집합, 검증집합, 테스트집합

출력: 최적 모델과 성능

```
1 for (Ω에 있는 각각의 모델)
2     모델을 훈련집합으로 학습시킨다.
3     검증집합으로 학습된 모델의 성능을 측정한다. // 검증 성능 측정
4 가장 높은 성능을 보인 모델을 선택한다.
5 테스트집합으로 선택된 모델의 성능을 측정한다.
```



1.5.3 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

알고리즘 1-3 교차검증에 의한 모델 선택

입력: 모델집합 Ω , 훈련집합, 테스트집합, 그룹 개수 k

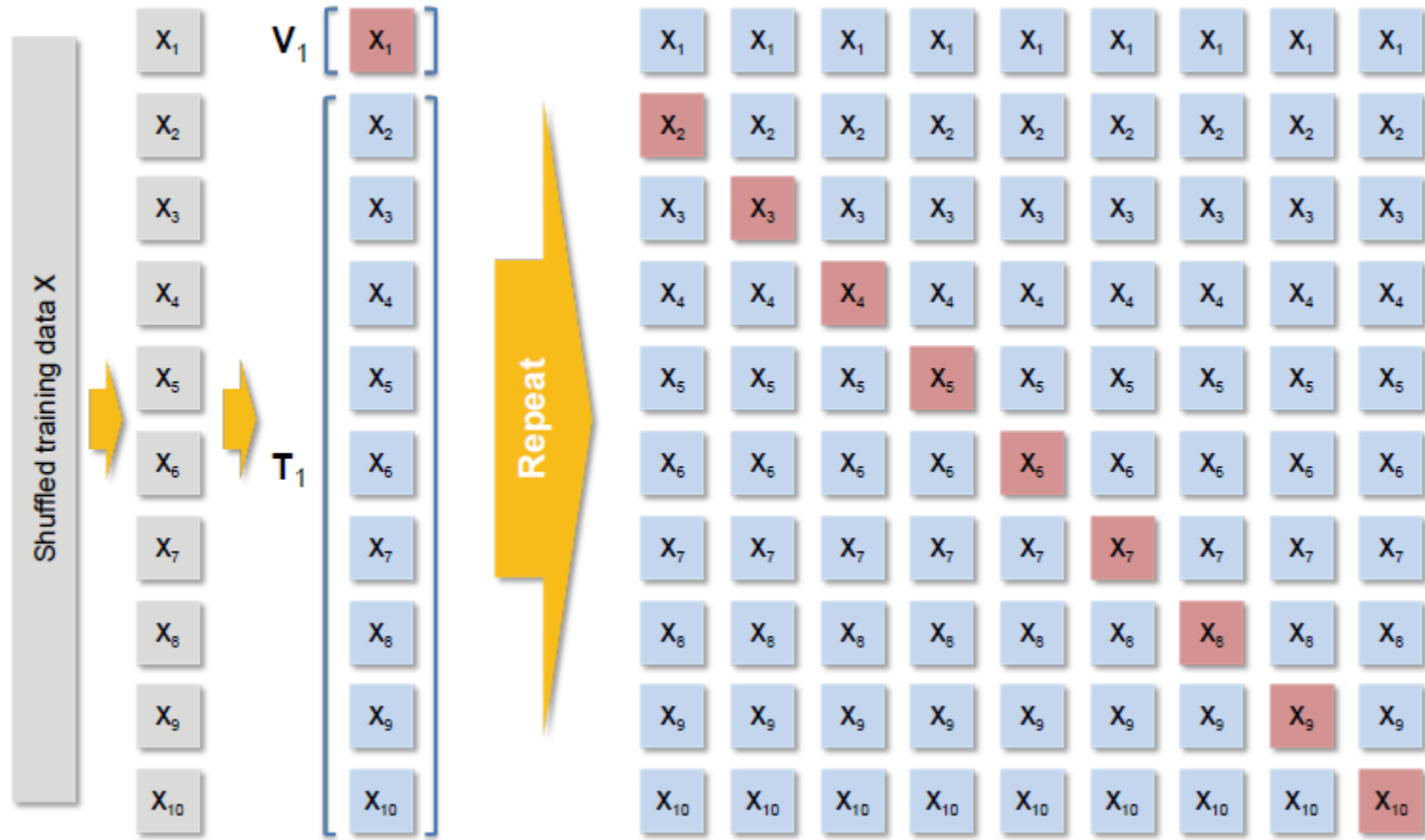
출력: 최적 모델과 성능

- 1 훈련집합을 k 개의 그룹으로 등분한다.
- 2 for (Ω 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.



1.5.3 검증집합과 교차검증을 이용한 모델 선택 알고리즘

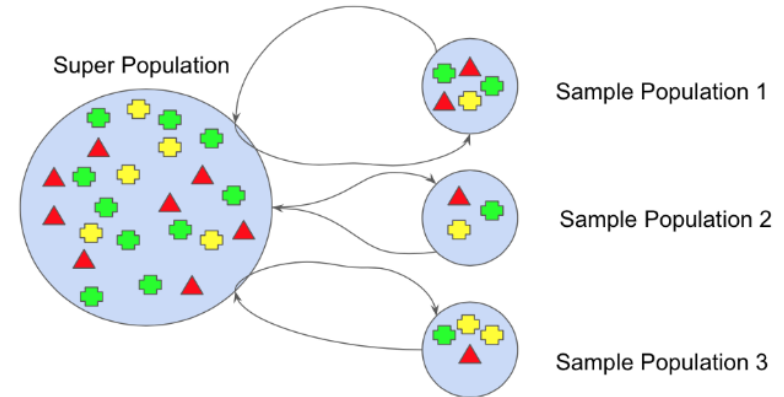
■ 10겹 교차검증 10-fold cross validation의 예



1.5.3 검증집합과 교차검증을 이용한 모델 선택 알고리즘

■ 부트스트랩bootstrap

- 임의의 복원 추출 샘플링sampling with replacement 반복
 - 데이터 분포가 불균형일 때 적용



알고리즘 1-4 부트스트랩을 이용한 모델 선택

입력: 모델집합 Ω , 훈련집합, 테스트집합, 샘플링 비율 $p(0 < p \leq 1)$, 반복횟수 T

출력: 최적 모델과 성능

- 1 for (Ω 에 있는 각각의 모델)
- 2 for ($i=1$ to T)
- 3 훈련집합 \mathbb{X} 에서 pn 개 샘플을 뽑아 새로운 훈련집합 \mathbb{X}' 를 구성한다. 이때 대치를 허용한다.
- 4 \mathbb{X}' 로 모델을 학습시킨다.
- 5 $\mathbb{X} - \mathbb{X}'$ 를 이용하여 학습된 모델의 성능을 측정한다.
- 6 T 개 성능을 평균하여 해당 모델의 성능으로 취한다.
- 7 가장 높은 성능을 보인 모델을 선택한다.
- 8 테스트집합으로 선택된 모델의 성능을 측정한다.

1.5.4 모델 선택의 한계와 현실적인 해결책

■ [알고리즘 1-2, 1-3, 1-4]에서 모델 집합 Ω

- [그림 1-13]에서는 서로 다른 차수의 다항식이 Ω 인 셈
- 현실에서는 학습 모델들이 아주 다양함
 - 신경망neural networks(3,4,8장), 강화 학습reinforcement learning(9장), 확률 그래피컬 모델probabilistic graphical models(10장), 서포트 벡터 머신support vector machine (SVM)(11장), 트리 분류기decision trees(12장) 등이 선택 대상
 - 신경망을 채택하더라도 MLP(3장), 깊은 MLP(4장), CNN(4장), RNN(8장) 등 아주 많음

■ 현실에서는 경험으로 큰 틀 선택한 후

- 모델 선택 알고리즘으로 세부 모델 선택하는 전략 사용
 - 예) CNN을 사용하기로 정한 후, 은닉층 개수, 활성화함수, 모멘텀 계수 등을 정하는데 모델 선택 알고리즘을 적용함

1.5.4 모델 선택의 한계와 현실적인 해결책

■ 이런 경험적인 접근방법에 대한 『Deep Learning』 책의 비유

“To some extent, we are always trying to fit a square peg(the data generating process) into a round hole(our model family). 어느 정도 우리가 하는 일은 항상 둥근 홈(우리가 선택한 모델)에 네모 막대기(데이터 생성 과정)를 끼워 넣는 것이라고 말할 수 있다[Goodfellow2016(222쪽)].”



■ 현대 기계 학습의 전략

- 용량이 충분히 큰 모델을 선택 한 후,
선택한 모델이 정상을 벗어나지 않도록 여러 가지 규제(regularization) 기법을 적용함
- 예) [그림 1-13]의 경우 12차 다항식을 선택하고, 다음 절에 설명될 적절히 규제를 적용

1.6 규제

■ 1.6.1 데이터 확대 dataset augmentation

■ 1.6.2 가중치 감소 weight decay

→ 규제를 통해서 일반화 성능 향상

■ 규제를 중요하게 다룬 책 [Goodfellow2016(7장)] [Haykin2009(7장)]

■ 5.3~5.4절에서 자세히 다룰 예정

- 가중치 벌칙 parameter norm penalty, 조기 멈춤 early stopping, 데이터 확대 bagging (bootstrap aggregation), 드롭아웃 dropout, 앙상블 ensemble methods 등

1.6.1 데이터 확대

- 데이터를 더 많이 수집하면 일반화 능력이 향상됨

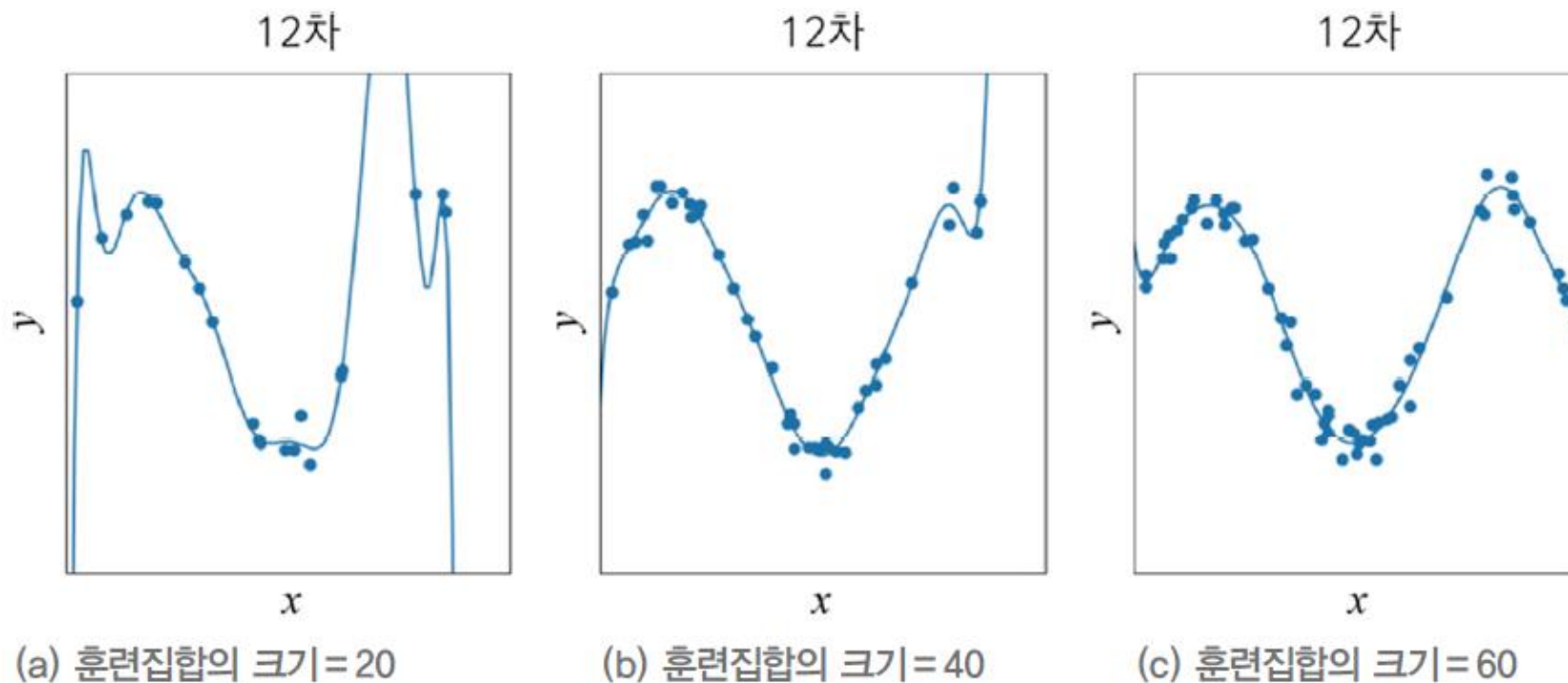
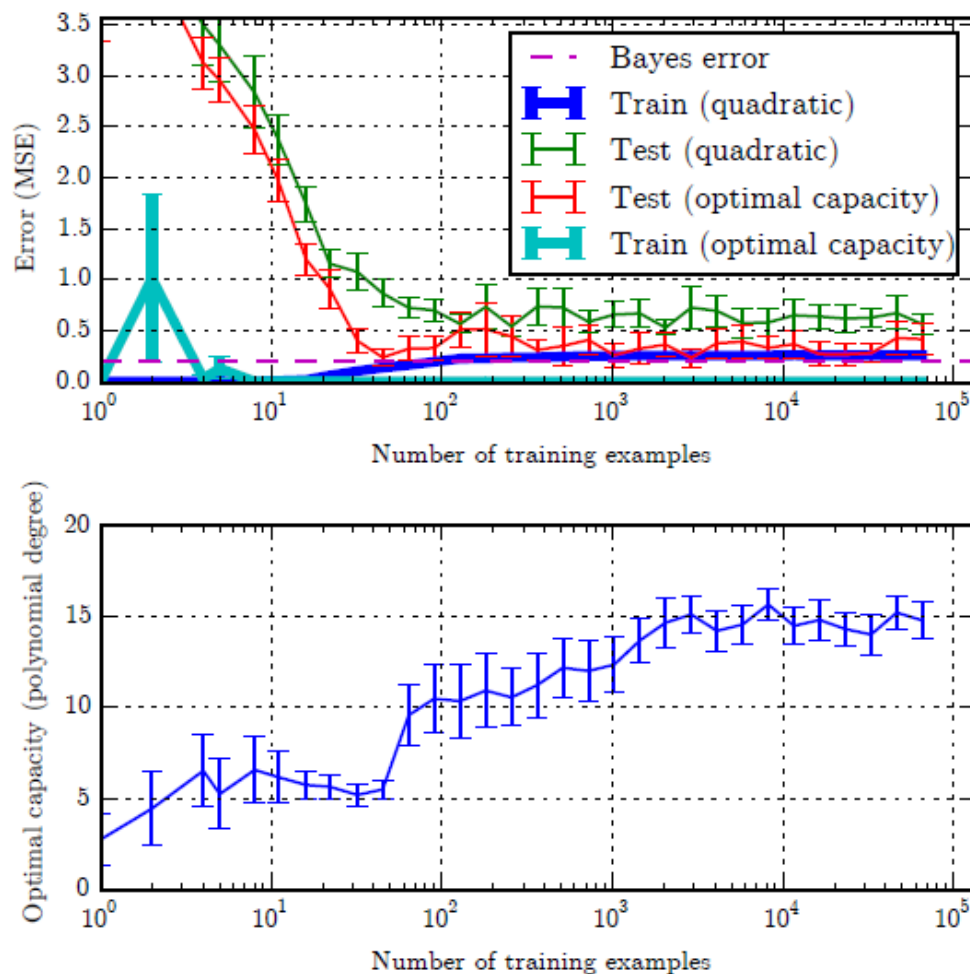


그림 1-17 데이터를 확대하여 일반화 능력을 향상함

1.6.1 데이터 확대

■ 훈련집합의 크기가 오차 성능에 미치는 영향 예



1.6.1 데이터 확대

■ 데이터 수집은 많은 비용이 듭

- 검증자료ground truth를 사람이 일일이 표식labelling을 해야 함

■ 인위적으로 데이터 확대data augmentation

- 훈련집합에 있는 샘플을 변형transform함
- 예) 약간 회전rotation 또는 왜곡warping (원 데이터의 부류 소속 등의 고유 특성 변하지 않게 주의)

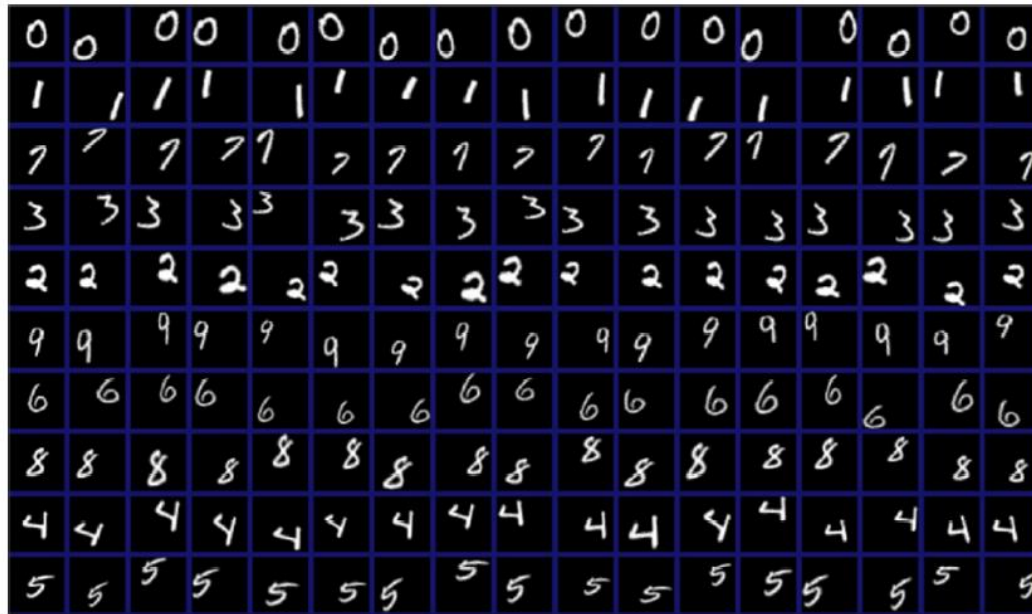


그림 5-24 필기 숫자 데이터의 다양한 변형⁸⁾

1.6.2 가중치 감쇠

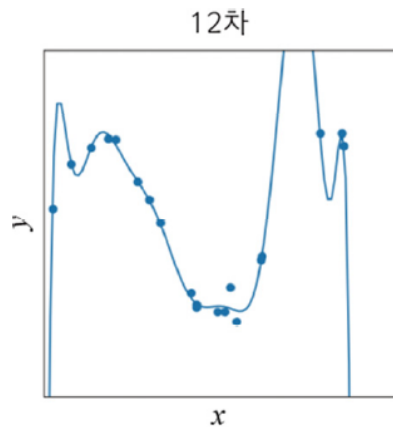
■ 가중치를 작게 조절하는 기법

- [그림 1-18(a)]의 12차 곡선은 가중치가 매우 큼

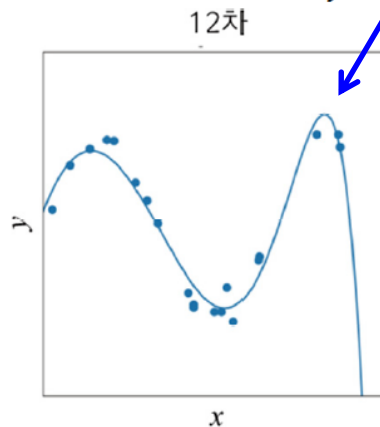
$$y = 1005.7x^{12} - 27774.4x^{11} + \dots - 22852612.5x^1 - 12.8$$

- 가중치 감쇠는 개선된 목적함수를 이용하여 가중치를 작게 조절하는 규제 기법
 - 식 (1.11)의 두 번째 항은 규제 항으로서 가중치 크기를 작게 유지해줌

$$J(\Theta) = \frac{1}{n} \sum_{i=1}^n (f_{\Theta}(\mathbf{x}_i) - y_i)^2 + \lambda \|\Theta\|_2^2 \quad (1.11)$$



(a) 가중치 감쇠 적용 안 함[식 (1.8)의 목적함수]



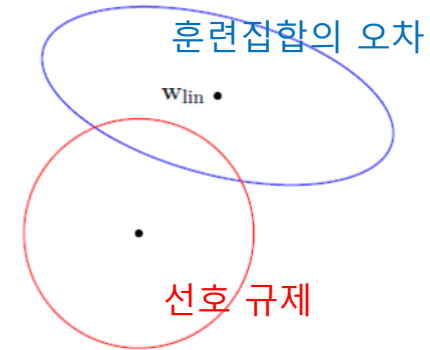
(b) 가중치 감쇠 적용함[식 (1.11)의 목적함수]

그림 1-18 가중치 감쇠에 의한 규제 효과

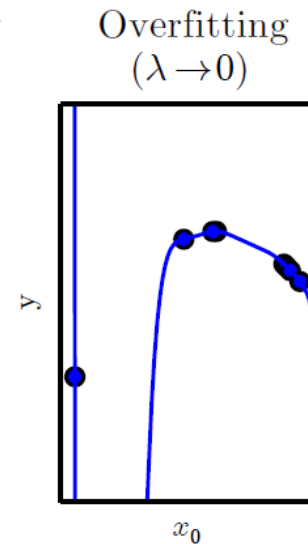
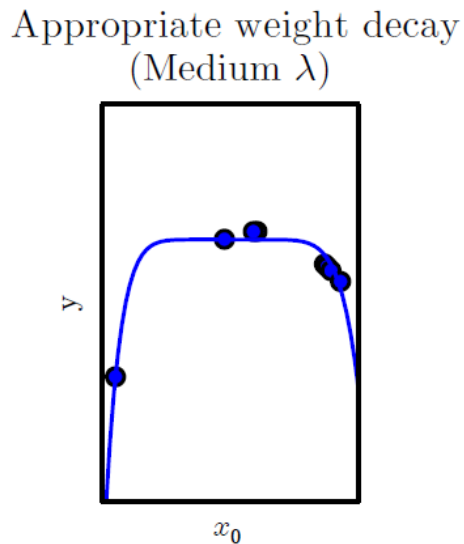
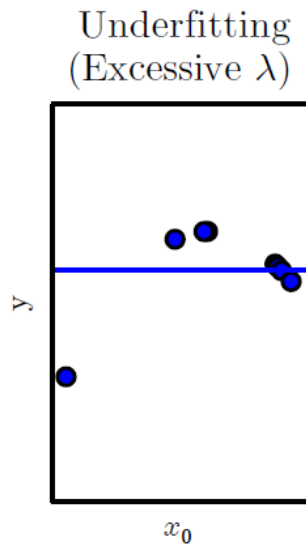
1.6.2 가중치 감쇠

■ 가중치 감쇠를 가진 선형 회귀 예

$$J(\theta) = \underbrace{\frac{1}{n} \sum_{i=1}^n (f_{\theta}(\mathbf{x}_i) - y_i)^2}_{\text{훈련집합의 오차}} + \underbrace{\lambda \|\theta\|_2^2}_{\text{선호 preference}} \quad (1.11)$$



- λ 는 주어진 **가중치의 우선** 혹은 **선호 정도**를 제어
 - $\lambda=0$: 선호도 없음
 - 큰 λ 는 가중치가 더 작아지도록 함



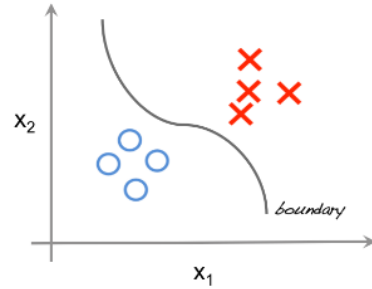
1.7 기계 학습 유형

- 1.7.1 지도 방식에 따른 유형
- 1.7.2 다양한 기준에 따른 유형

1.7.1 지도 방식에 따른 유형

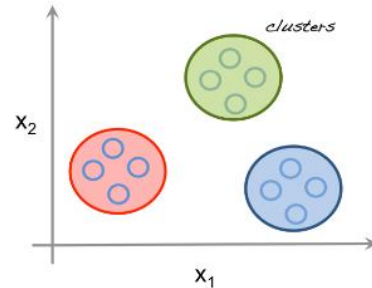
■ 지도 학습 supervised learning

- 특징 벡터 \mathbf{x} 와 목표치 \mathbf{y} (정답 있음)가 모두 주어진 상황
- 회귀 regression와 분류 classification 문제로 구분



■ 비지도 학습 unsupervised learning

- 특징 벡터 \mathbf{x} 는 주어지는데 목표치 \mathbf{y} 가 주어지지 않는 상황 (정답 없음)
- 군집화 clustering 과업 (고객 성향에 따른 맞춤 홍보 응용 등)
- 밀도 추정 density estimation, 특징 공간 변환 과업 (예. PCA)
- 6장의 주제



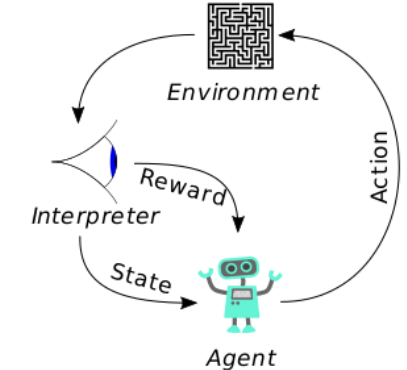
1.7.1 지도 방식에 따른 유형

■ 강화 학습 reinforcement learning

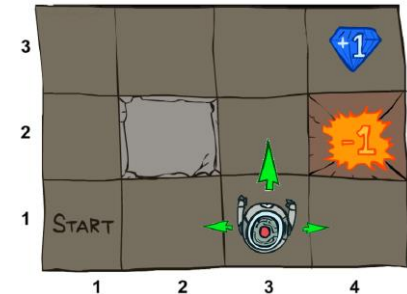
- 목표치가 주어지는데, 지도 학습과 다른 형태임 (== 보상 reward)
- 예) 바둑
 - 수를 두는 행위가 샘플인데, 게임이 끝나면 목표치 하나가 부여됨
 - 이기면 1, 패하면 -1을 부여
 - 게임을 구성한 샘플들 각각에 목표치를 나누어 주어야 함
- 9장의 주제

■ 준지도 학습 semi-supervised learning

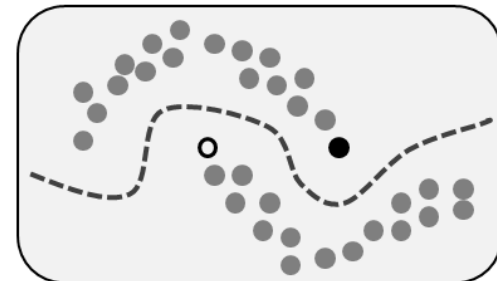
- 일부는 \mathbb{X} 와 \mathbb{Y} 를 모두 가지지만, 나머지는 \mathbb{X} 만 가진 상황
- 최근, 대부분의 데이터가 \mathbb{X} 의 수집은 쉽지만, \mathbb{Y} 는 수작업이 필요하여 최근 중요성 부각
- 7장의 주제



예) 강화 학습 모델의 동작



예) 그리드 월드 grid world 강화 학습 적용



1.7.2 다양한 기준에 따른 유형

- 오프라인 학습offline learning과 온라인 학습online learning
 - 보통은 오프라인 학습을 다룸
 - 온라인 학습은 IoT 등에서 추가로 발생하는 데이터 샘플을 가지고 점증적 학습 수행
- 결정론적 학습deterministic learning과 확률적 학습stochastic learning
 - 결정론적에서는 같은 데이터를 가지고 다시 학습하면 같은 예측 모델이 만들어짐
 - 확률적 학습은 학습 과정에서 확률 분포를 사용하므로 같은 데이터로 다시 학습하면 다른 예측 모델이 만들어짐
 - 10.4절의 RBM과 DBN이 확률적 학습
- 분별 모델과 생성 모델generative models
 - 분별 모델은 부류 예측에만 관심. 즉 $P(y|x)$ 의 추정에 관심
 - 생성 모델은 $P(x)$ 또는 $P(x|y)$ 를 추정함
 - 따라서 새로운 샘플을 '생성'할 수 있음
 - 4.5절의 GAN, 10.4절의 RBM은 생성 모델
 - 8.5절의 순환신경망(RNN)을 생성 모델로 활용하는 응용 예제

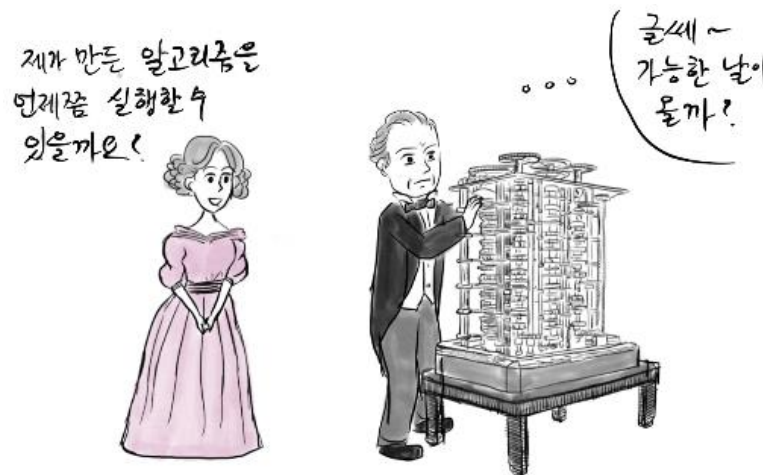
1.8 기계 학습의 과거와 현재, 미래

- 1.8.1 인공지능과 기계 학습의 간략한 역사
- 1.8.2 기술 추세
- 1.8.3 사회적 전망

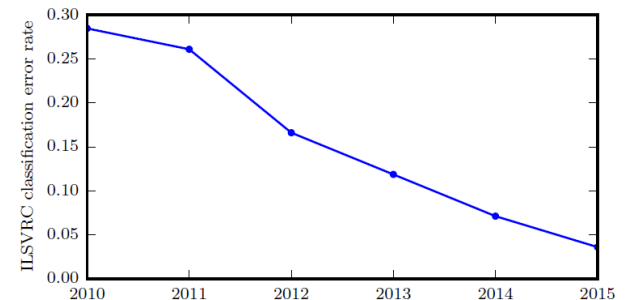
1.8.1 인공지능과 기계 학습의 간략한 역사

- 찰스 배비지의 제자인 에이다 러브레이스 여사의 통찰력 (19세기 중반)
 - "... 해석기관^{analytic engine}은 숫자 이외의 것도 처리할 수 있을 것이다. ... 예를 들어 화음과 음조를 해석기관의 표기에 맞출 수 있다면, 해석기관은 꽤 복잡한 곡을 작곡할 수도 있다."

[Ada1843]



- 200여 년이 지난 지금,
 - 인간 수준의 사진 인식 능력
 - 알파고^{AlphaGo}는 바둑으로 이세돌을 이김
 - 구글사^{Google}의 듀플렉스^{Duplex}는 인간과 대화



1.8.1 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 겨울AI winter 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



1.8.1 인공지능과 기계 학습의 간략한 역사

- 1987 Lisp 머신의 시장 붕괴로 제2의 인공지능 겨울
 UCI 리포지토리 서비스 시작
 NIPSNeural Information Processing Systems 컨퍼런스 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

1.8.1 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

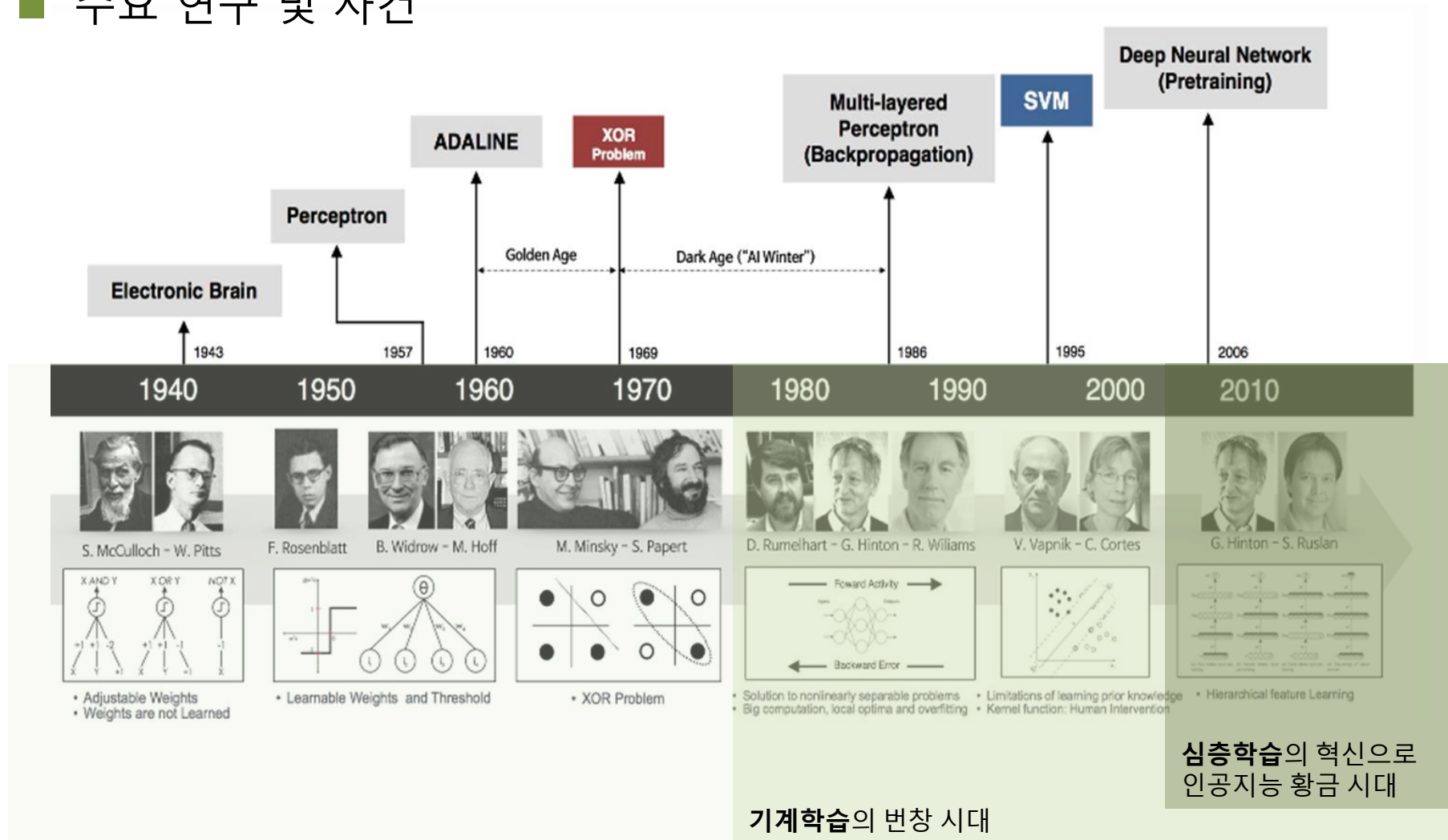
2016 알파고와 이세돌의 바둑 대회에서 알파고 승리[Silver2016]

『Deep Learning』 출간

2017 알파고 제로[Silver2017]

1.8.1 인공지능과 기계 학습의 간략한 역사

■ 주요 연구 및 사건



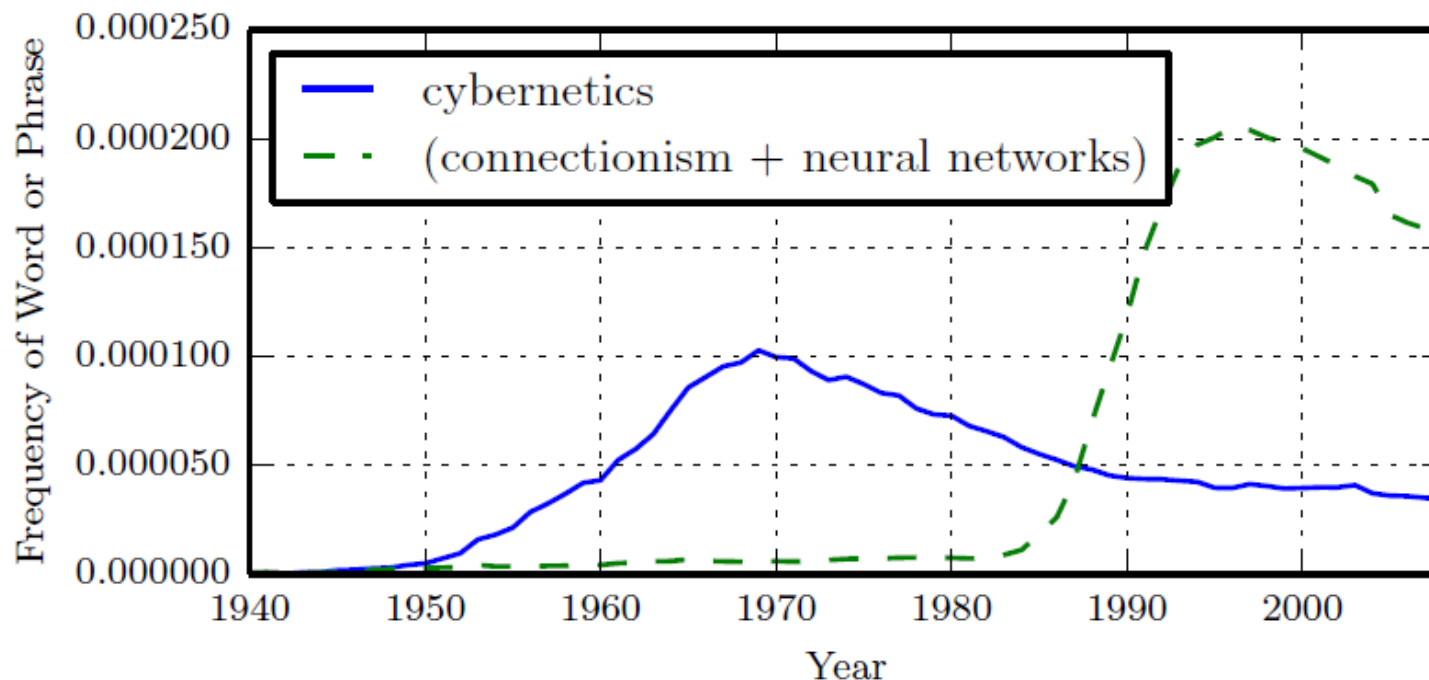
인공지능의 시작을 알린 촉발 시대

1.8.1 인공지능과 기계 학습의 간략한 역사

■ 인공지능의 역사

- 1940-1960: **인공두뇌학**cybernetics
- 1980-1990: **결합설**connectionism (or parallel distributed processing)
- 2006-현재: **딥러닝**

→ 인공지능의 역사적인 급증historical waves



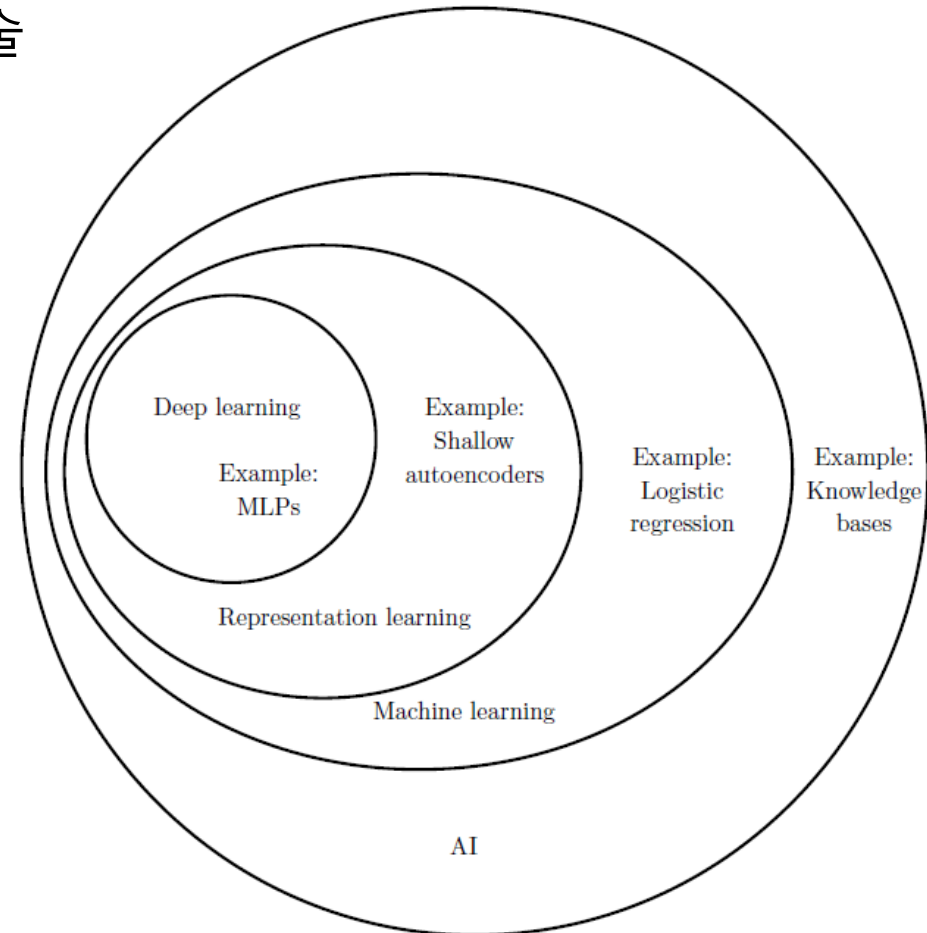
1.8.2 기술 추세

■ 리뷰 논문

- [LeCun2015, Jordan2015, Jones2014]

■ 딥러닝은 인공지능 실현에 핵심 기술

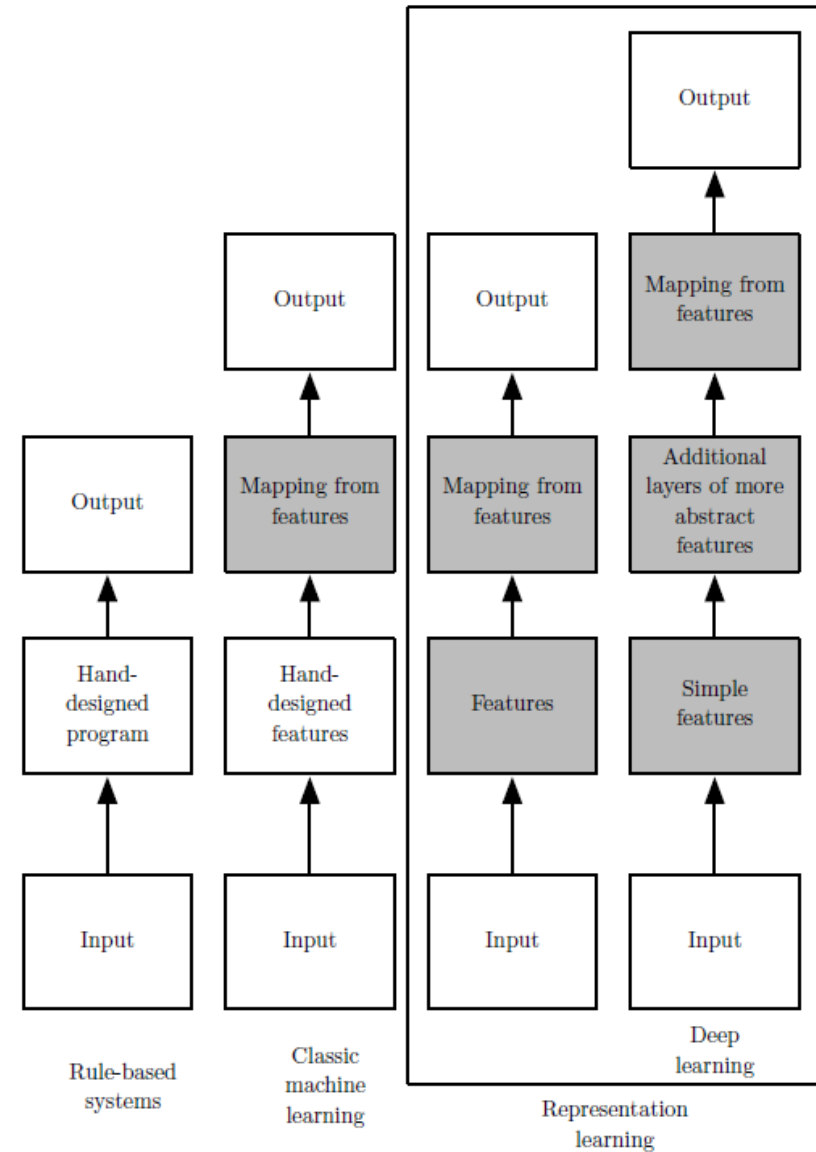
인공지능 범주



1.8.2 기술 추세

- 기계 학습 알고리즘과 응용의 **다양화**
- 서로 다른 알고리즘과 응용의 **융합**
- **딥러닝**이 기계 학습의 **주류**
- **표현 학습**이 중요해짐

학습의 주요 요소
회색박스는 데이터로부터 학습될 수 있는 요소



1.8.3 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
 - 오목 프로그램이나 바둑 프로그램은 규모만 다를 뿐 본질적으로 같은 일. 오목은 간단한 규칙으로 구현 가능하나 바둑은 효율적인 탐색 공간을 확인하는 보다 복잡한 기계 학습 알고리즘 사용
- 현재, 인공지능은 온통 수학과 컴퓨터 알고리즘 단계

1.8.3 사회적 전망

■ 인공지능의 단계

초인공지능(Super AI)

인공지능의 발전이 가속화되어 모든 인류의 지성을 합친 것보다 더 뛰어난 인공지능

인간을 넘어서는 특이점

강인공지능 (Strong AI = 인공일반지능)

인간이 할 수 있는 어떠한 지적인 업무도 성공적으로 해낼 수 있는 (가상적인) 기계의 지능

- 인간다운 인공지능으로의 변화 - 모든 상황에 두루 적용 가능
- 인간의 오감을 인지하는 컴퓨터를 이용해 직관적인 간접체험 가능
 - 거대하고 어려운 사회문제의 해결
 - 획기적으로 달라지는 의료활동
 - 인지능력과 통신망을 통합해 모든 시스템을 스마트하게 변화
 - 인간의 의사결정시 센서 데이터 활용
 - 지식 암기 기반 교육과정 소멸, 인공지능 과외교사 보편화
 - 입법, 행정, 사법 기능 대체 / 정부, 국가 기본제도와 운영 변화
- 인간의 삶을 이해하는 기계에 의한 초지능 시대 진입

약인공지능 (Weak AI)

인간이 지시한 명령의 틀 안에서만 일하기 때문에 예측과 관리가 용이

최근, 약인공지능의 빠른 발전
특정 분야의 일만 할 수 있도록 설계됨 → 종합적 판단에 한계를 보임

