



가톨릭대학교
THE CATHOLIC UNIVERSITY OF KOREA

머신러닝 개요

미디어기술콘텐츠학과
강호철

학습

- 사람의 학습

- 중, 고생 교과목
- 수영, 자전거 타기 등 운동

- 동물의 학습

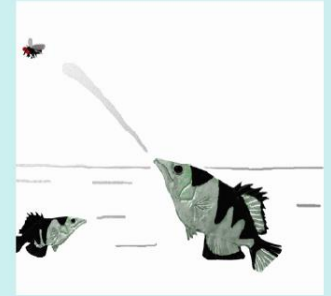
- 물총 물고기의 목표물 맞추기 능력 향상

- 기계의 학습

- 기계도 학습을 할 수 있을까?
- 기계학습 학문은 결국 기계가 학습 하기 위한 방법론에 대한 연구



(a) 자전거 타기 학습



(b) 물총물고기의 사냥

그림 1-1 사람과 동물의 학습

기계 학습 정의

- 학습이란?

- 표준 국어 대사전

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

- 기계 학습이란?

- 초창기 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 학습 정의

- 기계 학습이란?
 - 현대적 정의

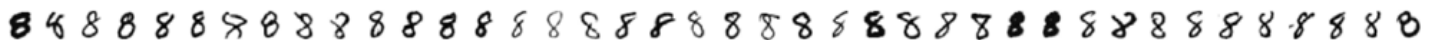
“A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E . 어떤 컴퓨터 프로그램이 T 라는 작업을 수행한다. 이 프로그램의 성능을 P 라는 척도로 평가했을 때 경험 E 를 통해 성능이 개선된다면 이 프로그램은 학습을 한다고 말할 수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

- 인공지능의 탄생
 - 컴퓨터의 뛰어난 능력
 - 사람이 어려워하는 일을 아주 쉽게 함
 - $80932.467076 \times 0.390324$ 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 분류 일도 잘 하지 않을까
 - 1950년대에 인공지능이라는 분야 등장
- 초창기는 지식 기반 방식이 주류
 - 예) 숫자 8 인식 문제



지식기반 방식에서 기계학습으로 전환

■ 지식 기반 방식 vs. 기계학습



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

지식기반 방식에서 기계학습으로 전환

- 인공지능의 주도권 전환
 - 지식기반에서 기계학습으로 전환
 - 기계학습의 특징?
 - 데이터 기반 접근 방식



그림 1-3 기계 학습으로 만든 최첨단 인공지능 제품들

기계학습 개념

- 기계학습 예제
 - 가로축: 시간
 - 세로축: 이동체 위치

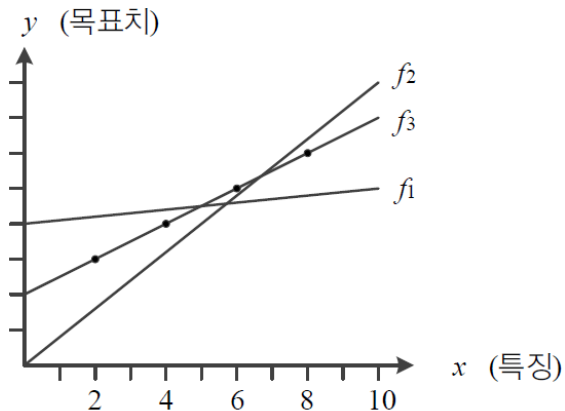


그림 1-4 간단한 기계 학습 예제

- 예측 문제
 - 임의의 시간에 대한 이동체 위치 예측
 - 회귀 vs. 분류 차이점?

기계학습 개념

- 기계학습 예제
 - 가로축: 시간 (특징)
 - 세로축: 이동체 위치 (목표치)

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

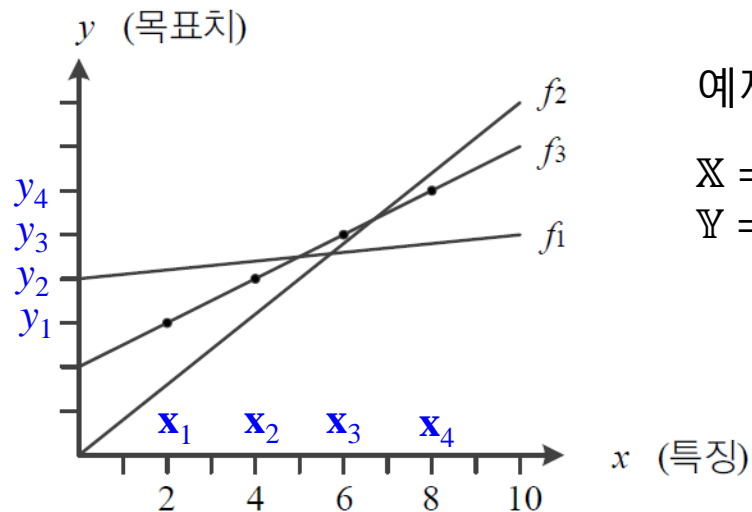


그림 1-4 간단한 기계 학습 예제

예제의 훈련집합

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

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

기계학습 개념

- 기계학습 예제

- 데이터 모델링 (가정)
- 직선 모델의 수식

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

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

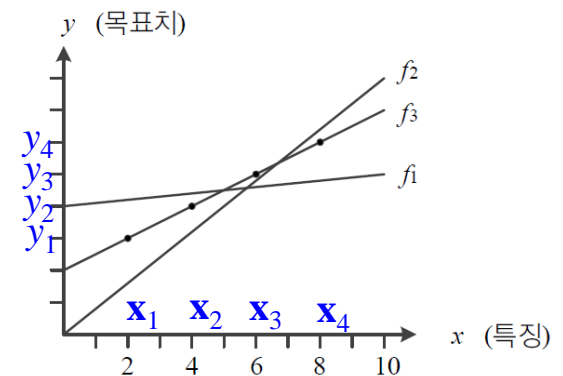


그림 1-4 간단한 기계 학습 예제

기계학습 개념

- 기계학습 예제

- 학습을 마치면 새로운 데이터에 대한 예측에 사용
- 10초 때 이동체 위치는?

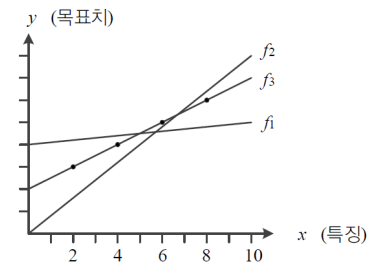


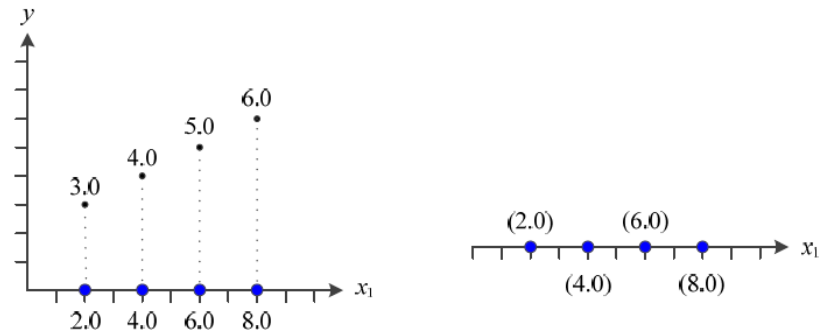
그림 1~4 간단한 기계 학습 예제

- 기계학습의 궁극적 목표

- 학습 결과를 새로운 데이터에 적용 했을 때 오류 최소화
- 테스트 집합에 대한 성능을 높이는 일반화 능력 고도화

특징 공간

■ 1차원 특징공간



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

■ 2차원 특징공간

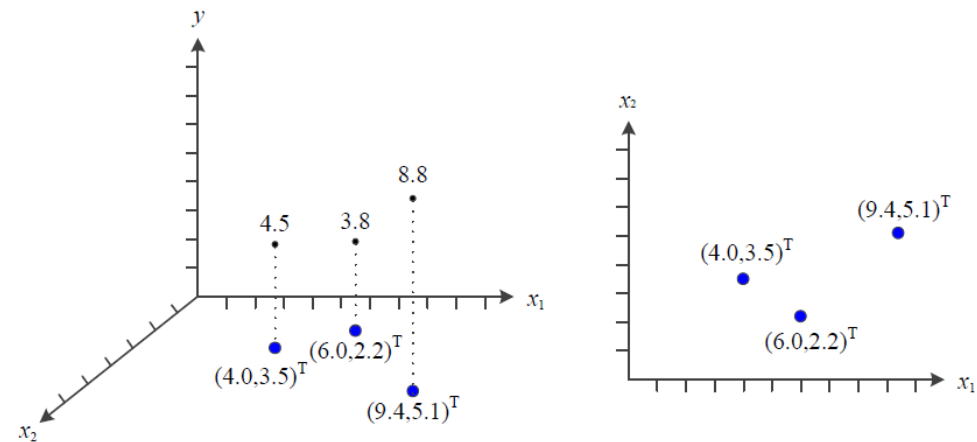
■ 특징 벡터 표기

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

■ 예시

■ $\mathbf{x}=(\text{몸무게}, \text{키})^T, y=\text{장타율}$

■ $\mathbf{x}=(\text{체온}, \text{두통})^T, y=\text{감기 여부}$



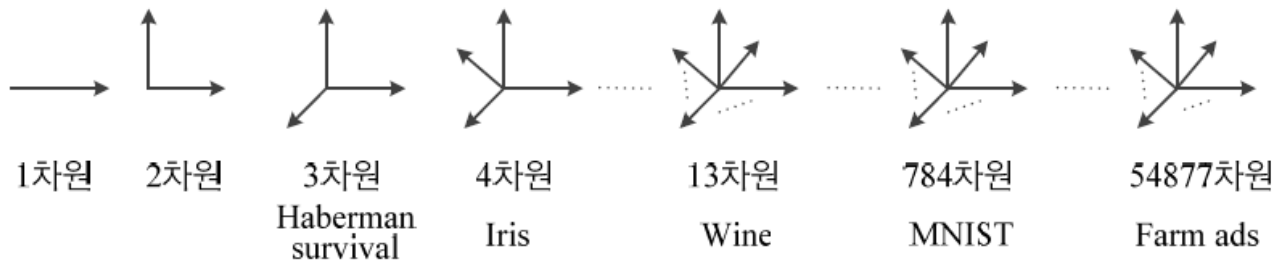
(b) 2차원 특징 공간(왼쪽: 특징 벡터와 목표값을 축으로 표시, 오른쪽: 특징 벡터만 축으로 표시)

그림 1-5 특징 공간과 데이터의 표현

특징 공간

■ 다차원 특징공간

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



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

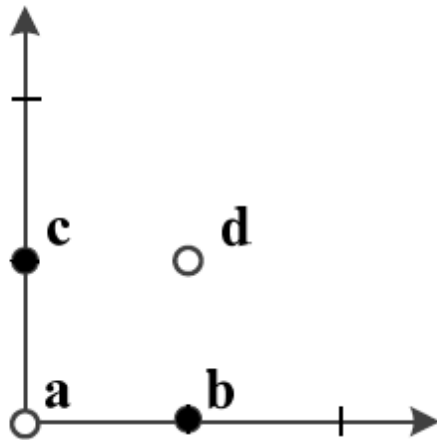
Farm ads: $\mathbf{x} = (\text{단어1}, \text{단어2}, \dots, \text{단어54877})^T$

그림 1-6 다차원 특징 공간

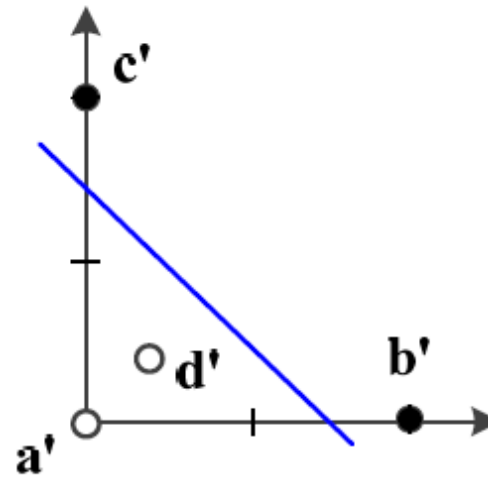


특징 공간 변환과 표현 학습

- 선형 분리 불가능 특징공간
 - 직선 모델을 적용하면 75% 정확 률이 한계



(a) 원래 특징 공간

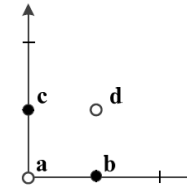


(b) 분류에 더 유리하도록 변환된 새로운 특징 공간

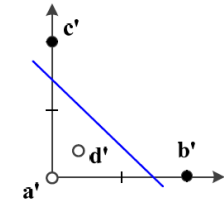
특징 공간 변환과 표현 학습

■ 특징 공간 변환

- 직선 모델을 적용하면 75% 정확률이 한계



(a) 원래 특징 공간



(b) 분류에 더 유리하도록 변환된 새로운 특징 공간

원래 특징 벡터 $\mathbf{x} = (x_1, x_2)^T \rightarrow$ 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 학습

- 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

- 특징점 간 차이공간 변환
 - 일반적인 metric
 - 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

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

데이터에 대한 이해

- 과학 기술의 발전 과정

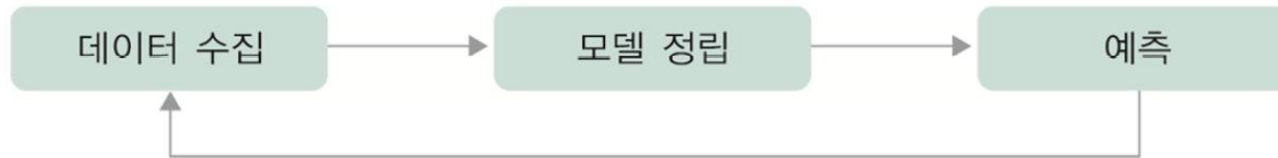


그림 1-8 과학기술의 발전 과정

- 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

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 매우 중요
- 다수의 공개 데이터베이스
 - 기계 학습의 초파리로 여겨지는 3가지 데이터베이스
 - Iris, MNIST, ImageNet
 - 위키피디아에서 'list of datasets for machine learning research'로 검색



데이터베이스의 중요성

■ Iris

- 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



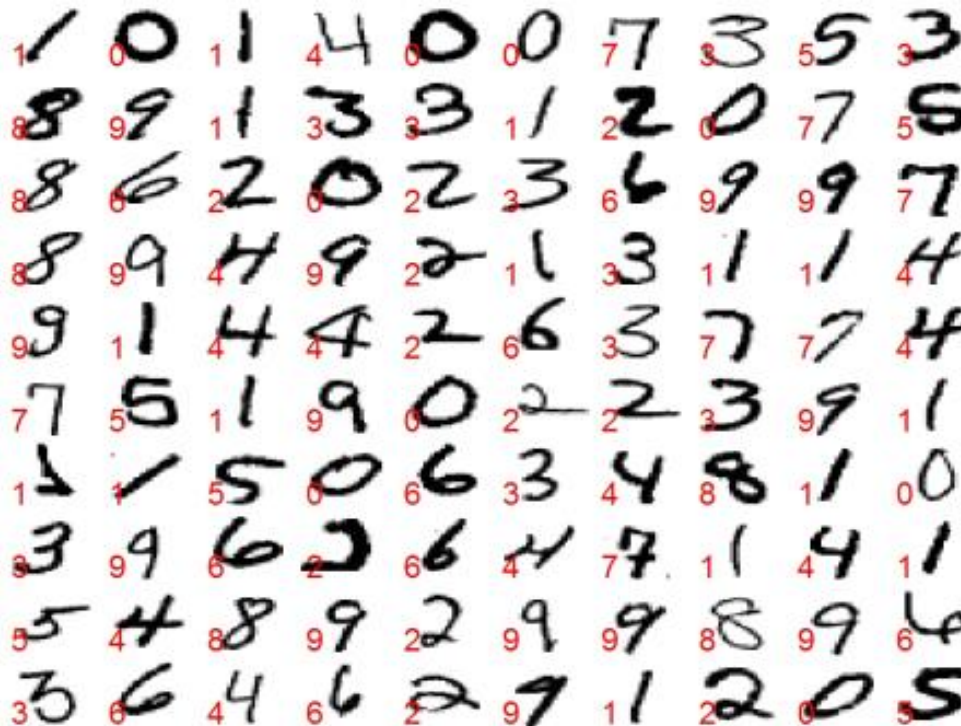
Versicolor
Virginica



데이터베이스의 중요성

■ MNIST

- 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

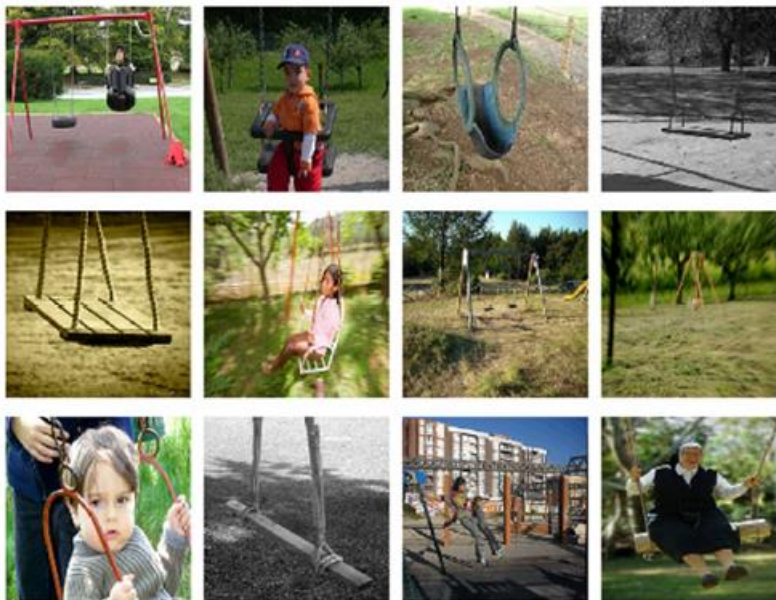
- Fashion MNIST



데이터베이스의 중요성

■ ImageNet

- 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.

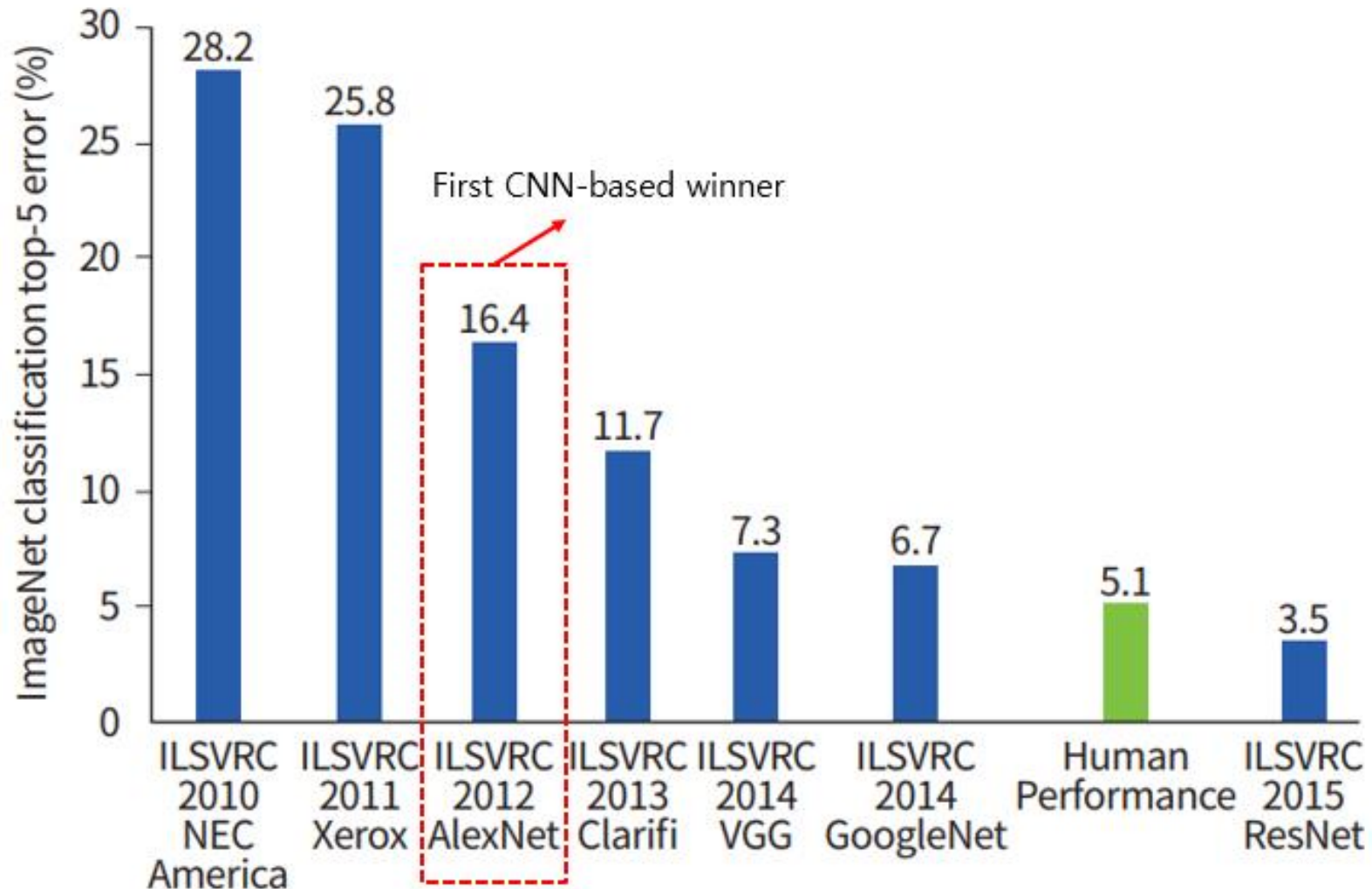


(a) 'swing' 부류



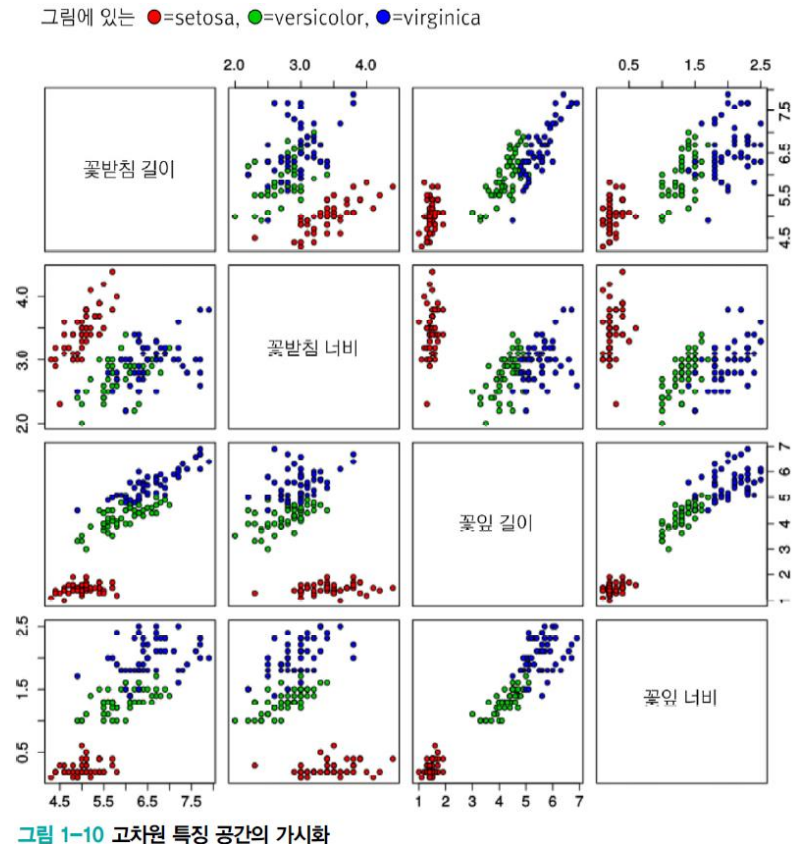
(b) 'Great white shark' 부류

데이터베이스의 중요성



데이터 가시화

- 4차원 이상의 초공간은 한꺼번에 가시화 불가능
- 여러 가지 가시화 기법
 - 2개씩 조합하여 여러 개의 그래프



- 고차원 공간을 저차원으로 변환하는 기법들을 사용하기도 함

기계학습에서의 학습 전개

- 선형 회귀 문제
 - 직선 모델 두 매개 변수 $\theta = (w, b)^T$

$$y = wx + b$$

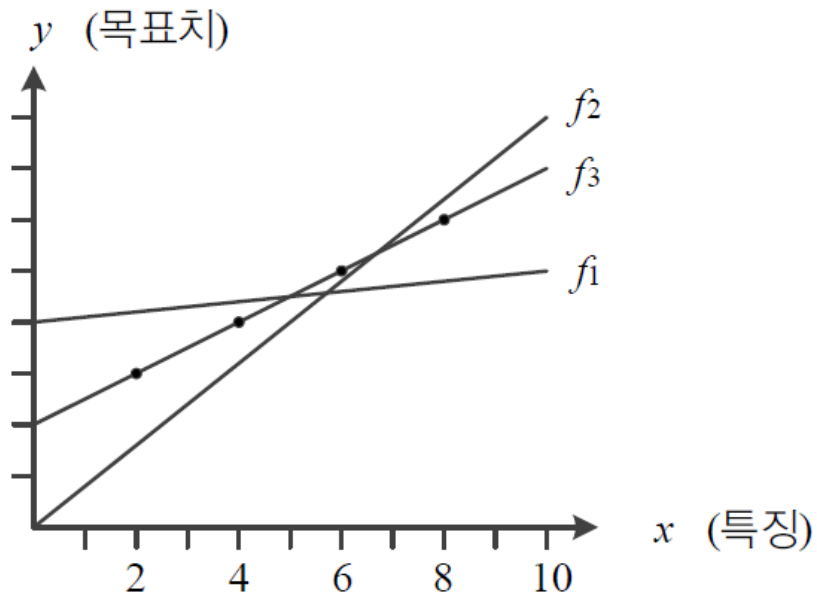


그림 1-4 간단한 기계 학습 예제

기계학습에서의 학습 전개

- 목적 함수 (비용 함수)
 - 선형회귀를 위한 목적 함수 J
 - 평균제곱오차 (MSE, Mean Squared Error)

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

- 초기에 최적 매개변수 값을 알 수 없으므로 난수로 $\Theta_1 = (w_1, b_1)^T$ 설정
→ $\Theta_2 = (w_2, b_2)^T$ 로 개선 → $\Theta_3 = (w_3, b_3)^T$ 로 개선 → Θ_3 는 최적해 $\hat{\Theta}$
 - 이때 $J(\Theta_1) > J(\Theta_2) > J(\Theta_3)$

기계학습에서의 학습 전개

- 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 $\theta_1 = (0.1, 4.0)^T$ 로 가정

$$x_1, y_1 \rightarrow (f_{\theta_1}(2.0) - 3.0)^2 = ((0.1 * 2.0 + 4.0) - 3.0)^2 = 1.44$$

$$x_2, y_2 \rightarrow (f_{\theta_1}(4.0) - 4.0)^2 = ((0.1 * 4.0 + 4.0) - 4.0)^2 = 0.16$$

$$x_3, y_3 \rightarrow (f_{\theta_1}(6.0) - 5.0)^2 = ((0.1 * 6.0 + 4.0) - 5.0)^2 = 0.16$$

$$x_4, y_4 \rightarrow (f_{\theta_1}(8.0) - 6.0)^2 = ((0.1 * 8.0 + 4.0) - 6.0)^2 = 1.44$$

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



기계학습에서의 학습 전개

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

$$x_1, y_1 \rightarrow (f_{\theta_2}(2.0) - 3.0)^2 = ((0.8 * 2.0 + 0.0) - 3.0)^2 = 1.96$$

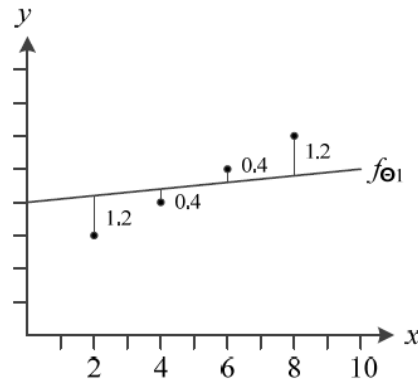
$$x_2, y_2 \rightarrow (f_{\theta_2}(4.0) - 4.0)^2 = ((0.8 * 4.0 + 0.0) - 4.0)^2 = 0.64$$

$$x_3, y_3 \rightarrow (f_{\theta_2}(6.0) - 5.0)^2 = ((0.8 * 6.0 + 0.0) - 5.0)^2 = 0.04$$

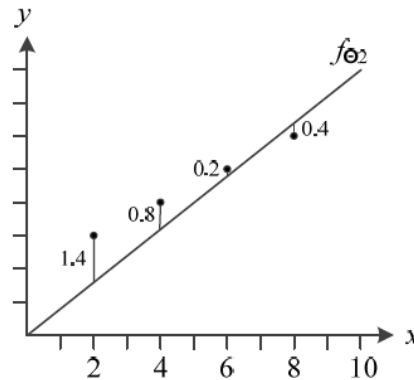
$$x_4, y_4 \rightarrow (f_{\theta_2}(8.0) - 6.0)^2 = ((0.8 * 8.0 + 0.0) - 6.0)^2 = 0.16$$

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

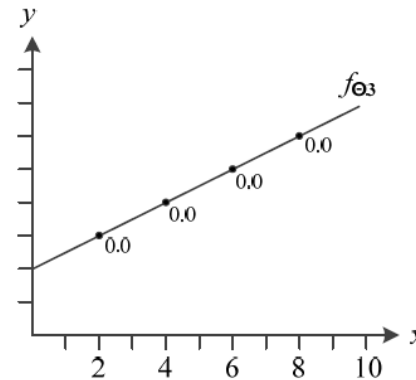
- θ_2 을 개선하여 $\theta_3 = (0.5, 2.0)^T$ 가 되었다고 가정
- 이때 $J(\theta_3) = 0.0$ 이 되어 θ_3 은 최적값 $\hat{\theta}$ 이 됨



(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

기계학습에서의 학습 전개

- 기계 학습 공식화

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

- 알고리즘

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

기계학습에서의 학습 전개

- 현실은....

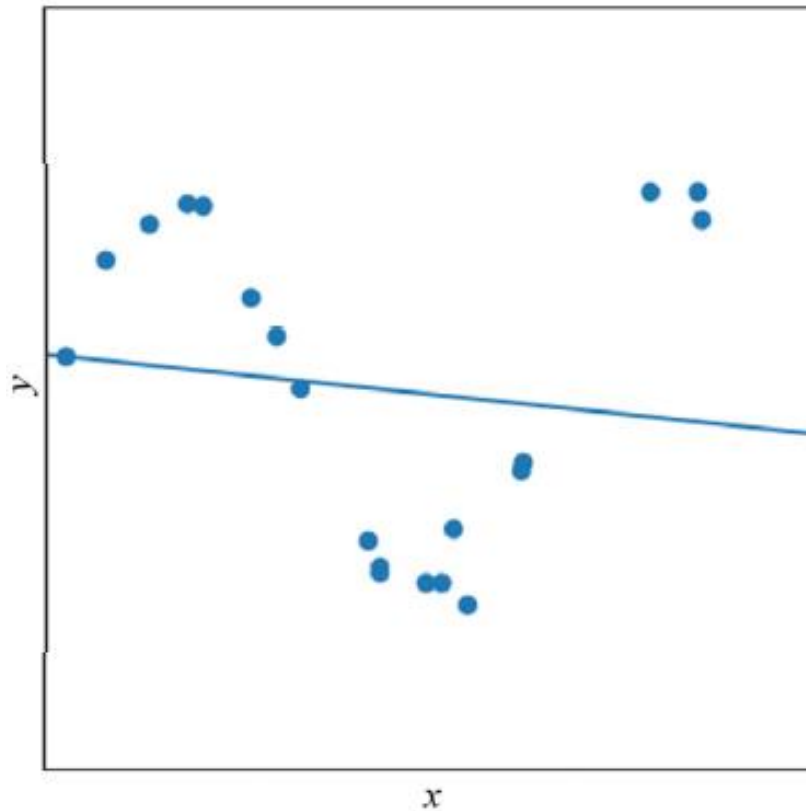


그림 1-12 선형 모델의 한계

과소적합, 과잉적합

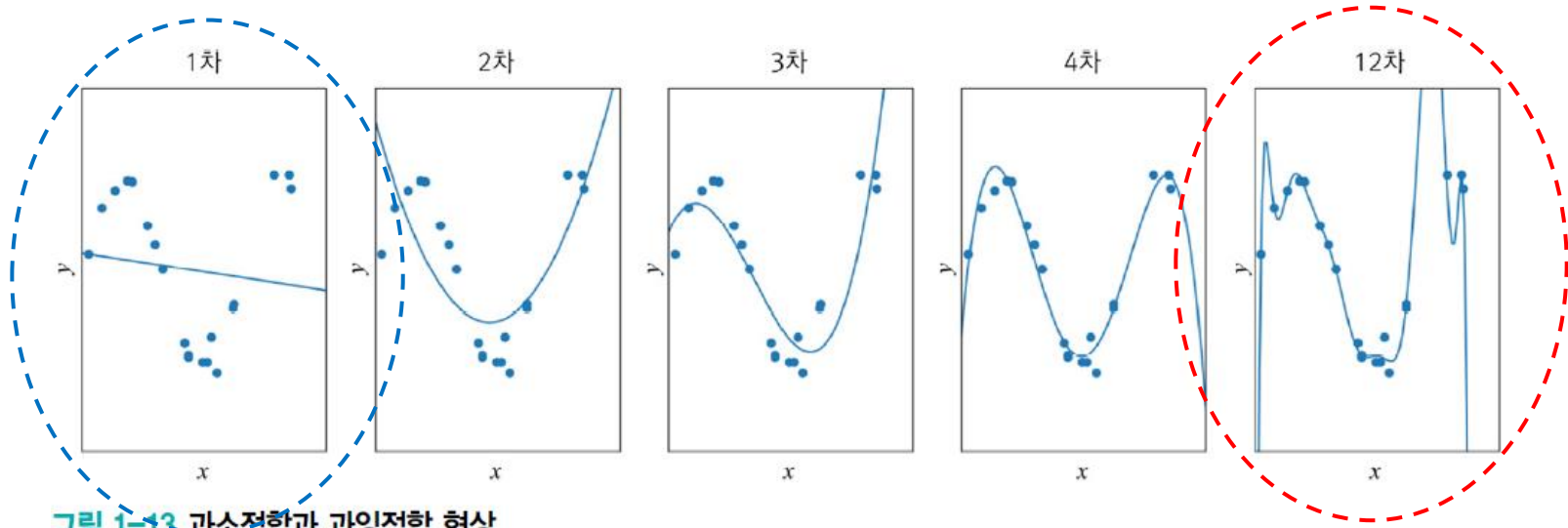


그림 1-13 과소적합과 과잉적합 현상

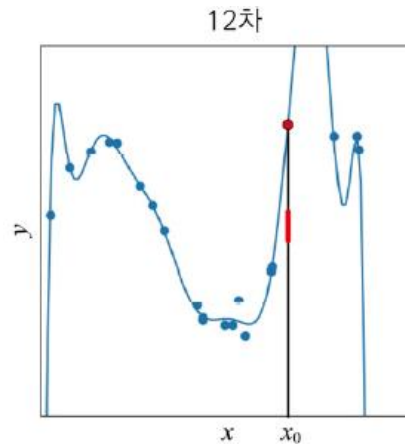


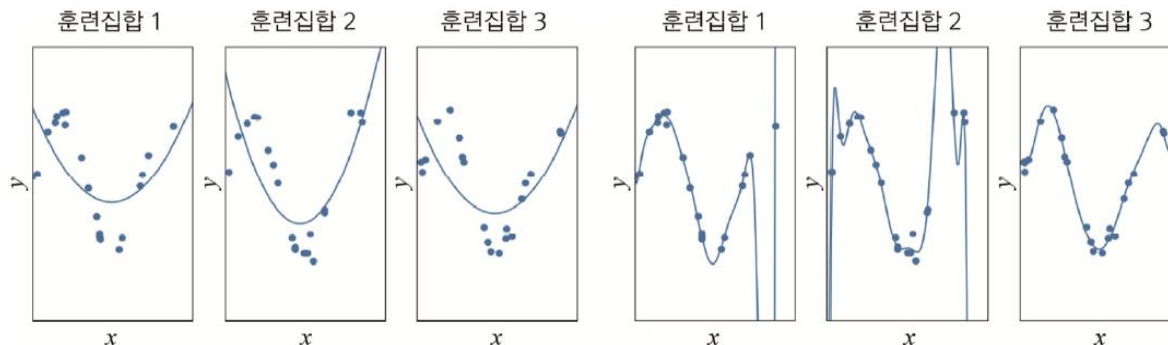
그림 1-14 과잉적합되었을 때 부정확한 예측 현상

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

바이어스와 분산

- 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 단순한 모델은 바이어스는 크고 분산은 작음
 - 복잡한 모델은 바이어스는 작고 분산은 큼
 - 바이어스와 분산은 트레이드오프 관계



(a) 2차 모델(바이어스는 크고, 분산은 작음)

(b) 12차 모델(바이어스는 작고, 분산은 큼)

그림 1-15 모델의 바이어스와 분산 특성

바이어스와 분산

- 기계학습의 목표
 - 낮은 바이어스와 낮은 분산을 가진 예측기 제작이 목표

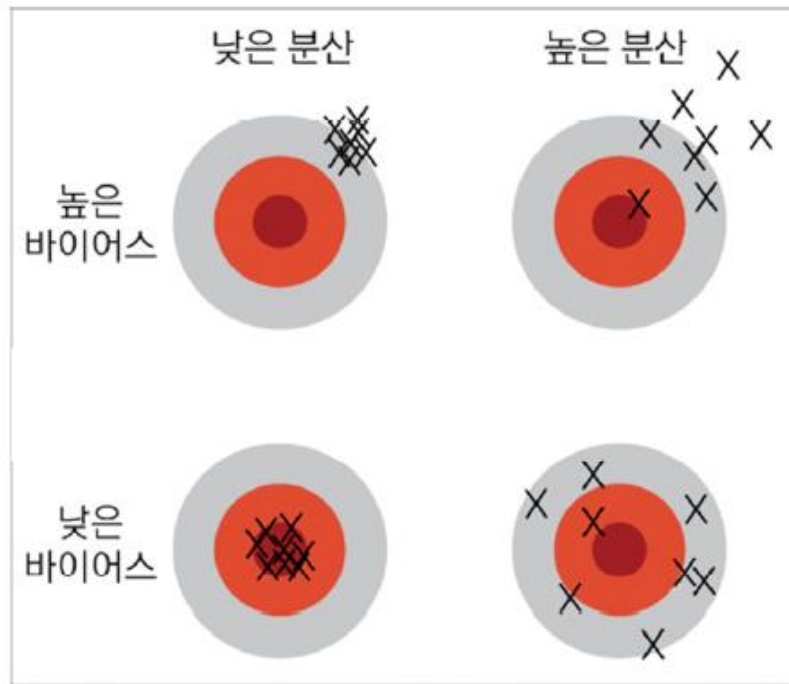


그림 1-16 바이어스와 분산

- 하지만 바이어스와 분산은 트레이드 오프 관계이므로 바이어스 희생을 최소로 유지하며 분산을 최대한 낮추는 전략 필요

모델 선택 알고리즘

- 검증집합을 이용한 모델 선택
 - 훈련집합, 테스트 집합과 다른 별도의 검증 집합을 가지고 모델 선택

알고리즘 1-2 검증집합을 이용한 모델 선택

입력: 모델집합 Ω , 훈련집합, 검증집합, 테스트집합

출력: 최적 모델과 성능

- 1 for (Ω 에 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.

모델 선택 알고리즘

- 교차검증
 - 별도의 검증집합을 구하기 어려운 경우 사용
 - 훈련집합을 등분하여 학습과 평가 과정 여러 번 반복 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.

모델 선택 알고리즘

- 부트스트랩
 - 난수를 이용한 샘플링 반복

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 ( $\Omega$ 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.
```

모델 선택의 한계와 해결책

- 모델 집합 Ω 으로 부터 최적의 모델 선택 문제
 - 현실에서는 아주 다양
 - 신경망, 강화 학습, 확률 그래피컬 모델, SVM, 트리 분류기 등이 선택 대상
 - 신경망을 채택하더라도 MLP, Deep MLP, CNN 등 아주 많음
 - 현실적 해결책
 - 경험적으로 큰 틀(model 혹은 hypothesis) 선택
 - 모델 선택 알고리즘으로 세부 모델 선택하는 전략 사용
 - 예) CNN을 사용하기로 정한 후, 은닉층 개수, 활성화함수, 모멘텀 계수 등을 정하는데 모델 선택 알고리즘을 적용함

규제

- 규제(regularization)를 하는 이유?
- 데이터 확대, 가중치 감소 등...



규제

- 데이터 확대
 - 데이터를 더 많이 수집하면 일반화 능력 향상

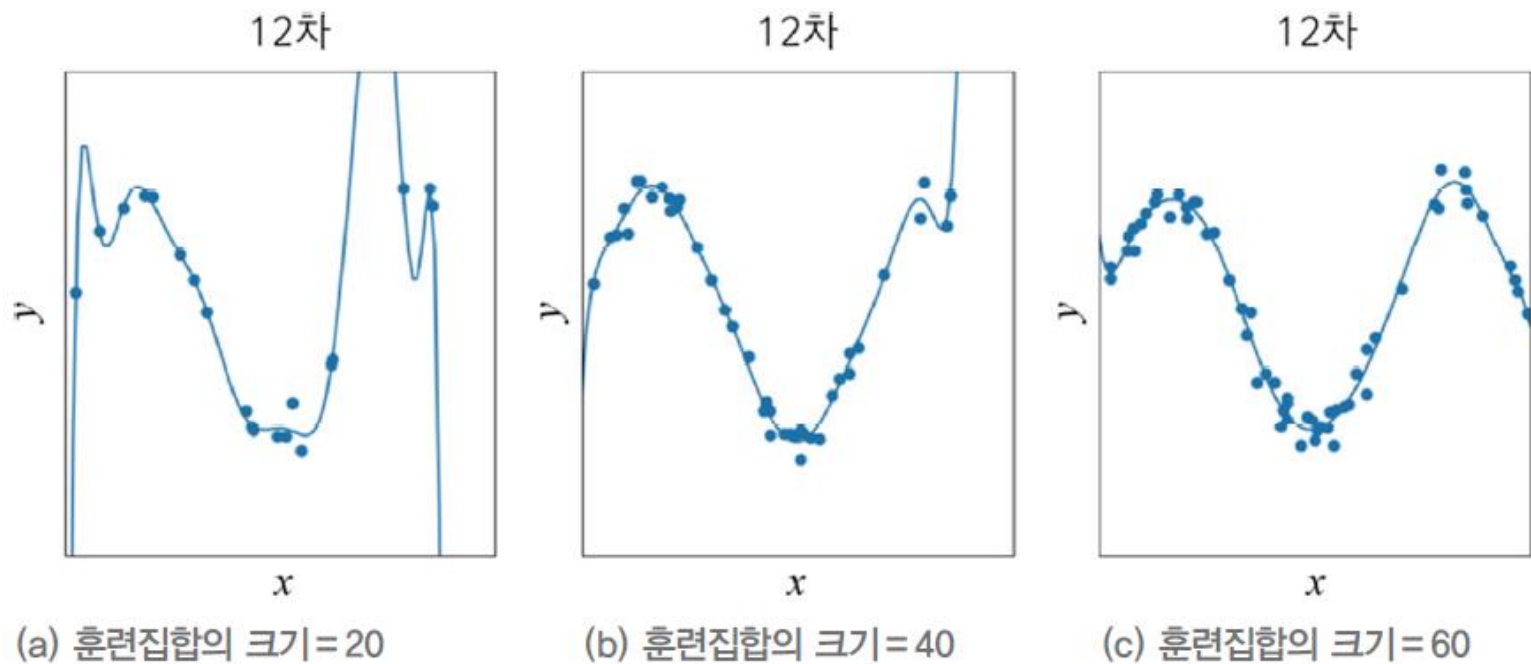


그림 1-17 데이터를 확대하여 일반화 능력을 향상함

규제

- 데이터 확대
 - 데이터 수집은 많은 비용이 발생
 - 데이터 모으기 + annotation
 - 인위적 데이터 확대

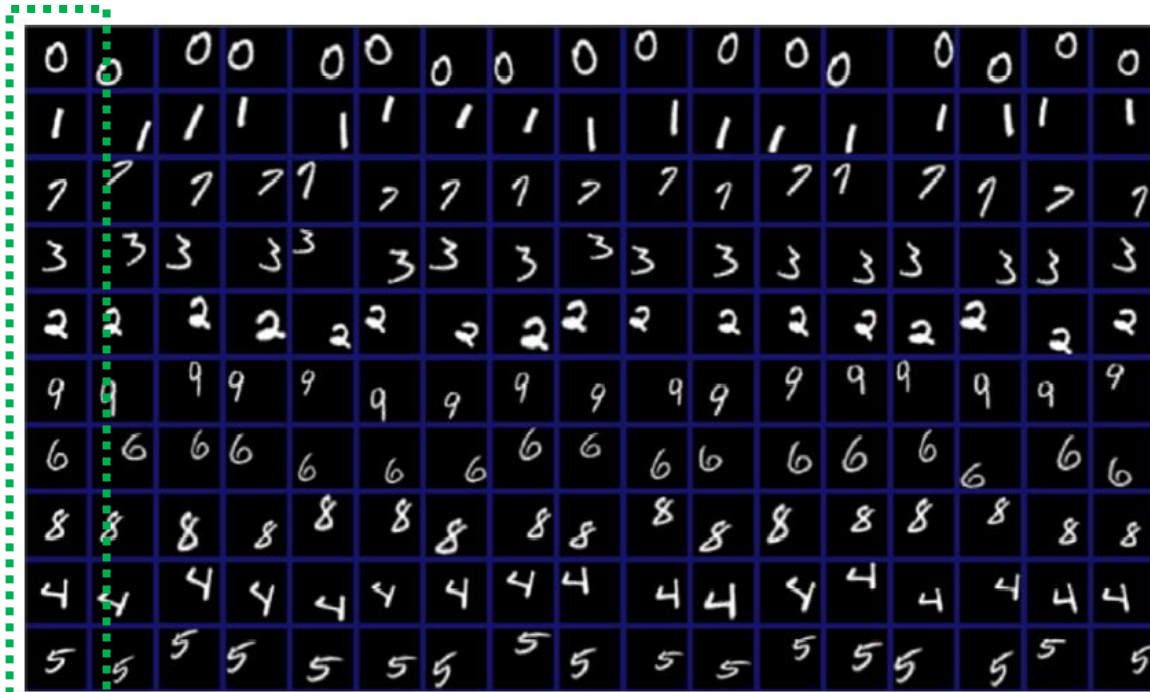


그림 5-24 필기 숫자 데이터의 다양한 변형⁸

규제

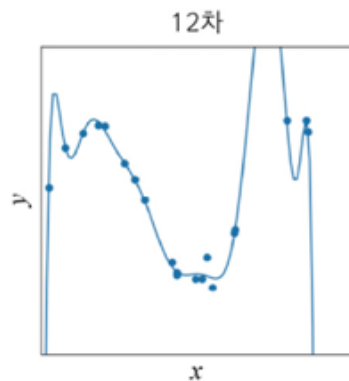
- 가중치 감쇠

- 앞 예제의 12차 곡선 식

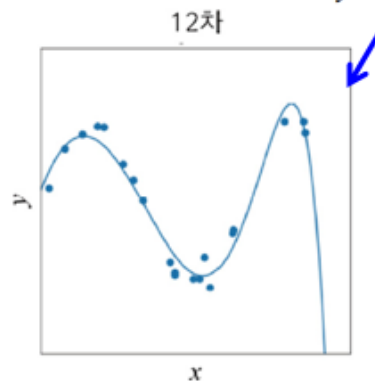
$$y = 1005.7x^{12} - 27774.4x^{11} + \dots - 22852612.5x^1 - 12.8$$

- 가중치 감쇠는 개선된 목적함수를 이용하여 가중치를 작게 조절하는 규제 기법으로 penalty term 추가

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



(a) 가중치 감쇠 적용 안 함[식 (1.8)의 목적함수]



(b) 가중치 감쇠 적용함[식 (1.11)의 목적함수]

$$y = 10.779x^{12} - 42.732x^{11} + \dots - 2.379x^1 + 0.119$$

그림 1-18 가중치 감쇠에 의한 규제 효과

기계학습의 유형

- 지도 학습
 - 특징 벡터 \mathbf{x} 와 목표값 y 가 모두 주어진 상황
 - 회귀와 분류 문제로 구분
- 비지도 학습
 - 특징 벡터 \mathbf{x} 는 주어지는데 목표값 y 가 주어지지 않는 상황
 - 군집화(clustering)
 - 밀도 추정, 특징 공간 변환 과업



기계학습의 유형

- 강화 학습
 - 목표값이 주어지는데, 지도 학습과 다른 형태임
 - 예) 바둑
 - 수를 두는 행위가 샘플인데, 게임이 끝나면 목표값 하나가 부여됨
 - 이기면 1, 패하면 -1을 부여
 - 게임을 구성한 샘플들 각각에 목표값을 나누어 주어야 함
- 준지도 학습
 - 일부는 X 와 Y 를 모두 가지지만, 나머지는 X 만 가진 상황
 - 인터넷 덕분에 X 의 수집은 쉽지만, Y 는 수작업이 필요하여 최근 중요성 부각

인공지능과 기계학습의 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 로젠블렛이 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



인공지능과 기계학습의 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 기계학습의 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]



참고자료

- Machine Learning 기계학습
 - 저자: 오일석
 - 출판사: 한빛아카데미, 2017

