

기계학습이란

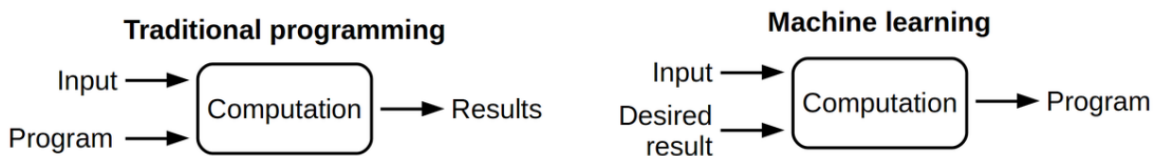
정의

- 학습
 - 경험의 결과로 나타나는, 비교적 지속적인 행동의 변화나 그 잠재력의 변화 또는 지식을 습득하는 과정
- 기계학습
 - 사무엘(인공지능 초창기), 컴퓨터가 **경험을 통해 학습**할 수 있도록 프로그래밍할 수 있다면, 세세하게 프로그래밍해야하는 번거로움에서 벗어날 수 있다

현대적 정의

- 어떤 컴퓨터프로그램이 T라는 작업을 수행할 때, 경험을 통해 성능이 개선된다면 이 프로그램은 학습을 한다고 말할 수 있음
 - **최적의 알고리즘**을 찾는 행위
 - 경험 **E**를 통해, 주어진 작업 **T**에 대한 성능 **P**의 향상
- 사례 데이터, 즉 과거 경험을 이용하여 성능 기준을 최적화하도록 프로그래밍 하는 작업
- 성능을 개선하거나 정확하게 **예측**하기 위해 경험을 이용하는 계산학 방법들

기계학습 VS 전통적인 프로그래밍



지식기반 ► 기계학습 대전환

- 인공지능의 탄생
 - 컴퓨터의 뛰어난 능력
 - 사람이 어려워하는 일을 아주 쉽게 수행한다
 - 컴퓨터에 대한 기대감
 - 사람이 쉽게하는 일도 컴퓨터가 할 수 있지 않을까?
 - **1950년대에 '인공지능' 개념 첫 등장**
- 초창기 : 지식기반 방식
 - **경험적인 지식** 혹은 **사실**을 인위적으로 컴퓨터에 부여하여 학습
- 큰 깨달음
 - **지식기반의 한계**
 - 사람은 변화가 심한 장면을 아주 쉽게 인식하지만, **왜 그렇게 인식**하는지 서술하지는 못함
- 주도권 전환
 - 지식 기반 ► 기계학습
 - 데이터 중심 접근방식

개념

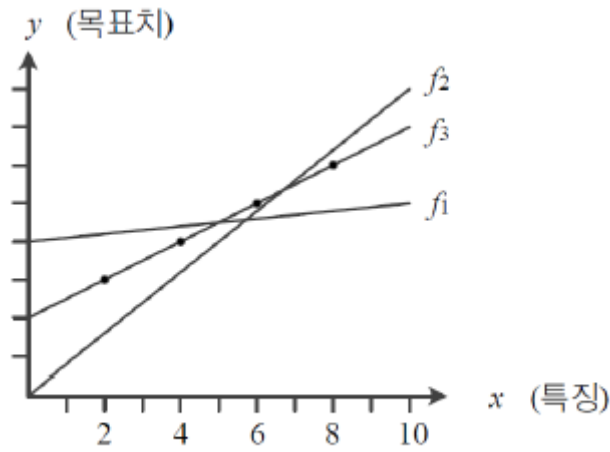


그림 1-4 간단한 기계 학습 예제

- 교사학습의 예
 - 가로축은 시간, 세로축은 이동체의 위치
 - 관측한 4개의 점이 데이터
- 예측 문제
 - 임의의 시간이 주어질 때, 이동체의 위치는?
 - 회귀문제와 분류문제
 - 회귀 : 목표치가 실수
 - 분류 : 목표치가 부류 혹은 종류의 값
- 훈련집합
 - 가로축은 특징, 세로축은 목표치
 - 관측한 4개의 점이 훈련집합

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

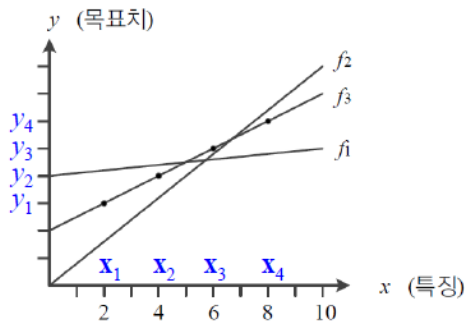


그림 1-4 간단한 기계 학습 예제

[그림 1-4] 예제의 훈련집합

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

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

- 관찰된 데이터를 어떻게 설명할 것인가?
 - 눈대중으로 봤을 때 점들이 직선을 이룬다 ▶ 모델로 직선을 선택을 가정
 - 직선 모델의 수식
 - $y = wx + b$ (매개변수 w, b)
- 기계학습은
 - 가장 정확하게 예측할 수 있는 최적의 매개변수를 찾는 작업
 - 처음에는 최적값을 모르니 임의의 값에서 시작, 점점 성능을 개선해 최적에 도달

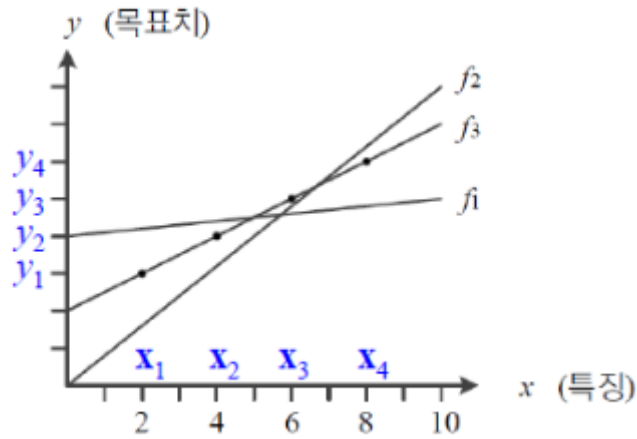


그림 1-4 간단한 기계 학습 예제

- $f_1 \blacktriangleright f_2 \blacktriangleright f_3$ (성능개선)
 - $w = 0.5, b = 2$
- 학습을 마치면
 - 새로운 특징(x 값)에 대응되는 목표치(y 값)의 예측에 사용
 - $y = 0.5x + 2 \blacktriangleright x$ 가 10일때 y 가 7이라고 예측
- 궁극적인 목표
 - 훈련집합에 없는 새로운 샘플에 대한 오류 최소화
 - 새로운 샘플 집합 = 테스트 집합(Test Set)
 - 일반화 능력 : 테스트 집합에 대한 높은 성능
- 필수요소
 - 내, 외부적 규칙 존재
 - 수학적으로 설명 불가능
 - 데이터가 있어야 함

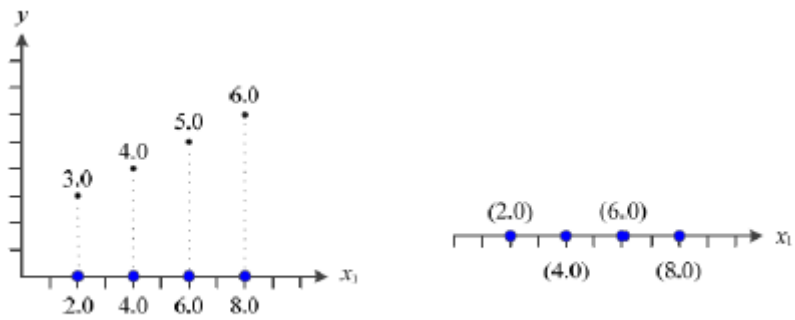
사람 VS 기계 학습

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

특징공간에 대한 이해

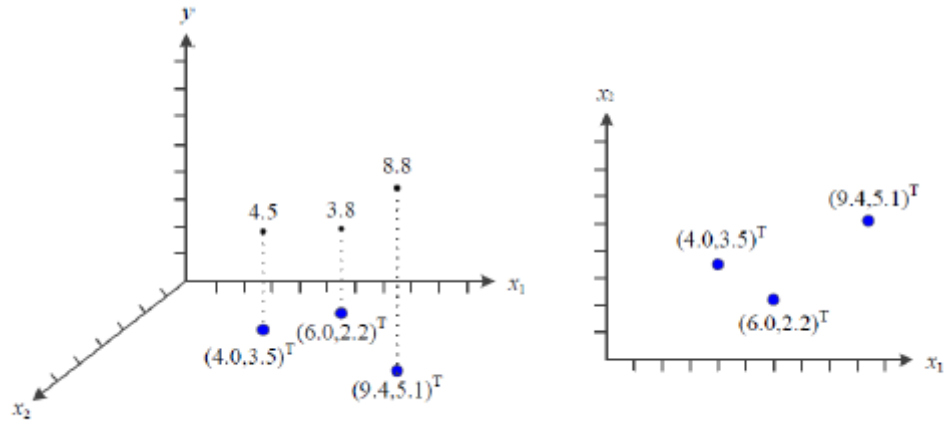
1차원과 2차원 특징공간

- 1차원 특징공간



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

- 2차원 특징공간



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

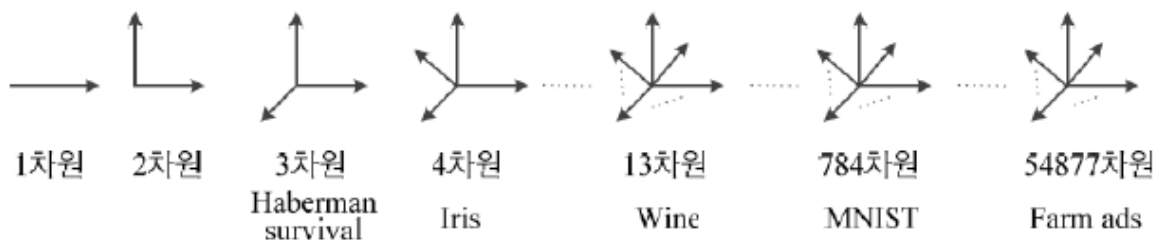
- 특징 벡터 표기

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

- 예시

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

다차원 특징공간



- x 값(특징)의 종류가 몇개인지에 따라 d 차원
 - 모든 데이터는 특징공간 안에 존재한다
- d -차원 데이터
 - 특징 벡터 표기

$$\mathbf{X} = (x_1, x_2, \dots, x_d)^T$$

- 학습모델

- 직선 모델인 경우 : 매개변수 수 = $d+1$

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

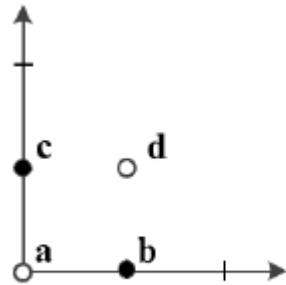
- 2차 곡선 모델인 경우 : 매개변수 수 = $d^2 + d + 1$

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

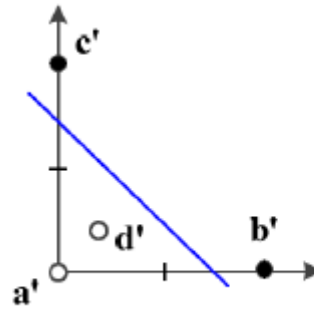
- 최적화를 잘 해야 함

특징공간 변환과 표현문제

- 선형 분리 불가능한 원래 특징 공간
 - 직선모델 적용시 정확도 75프로 한계

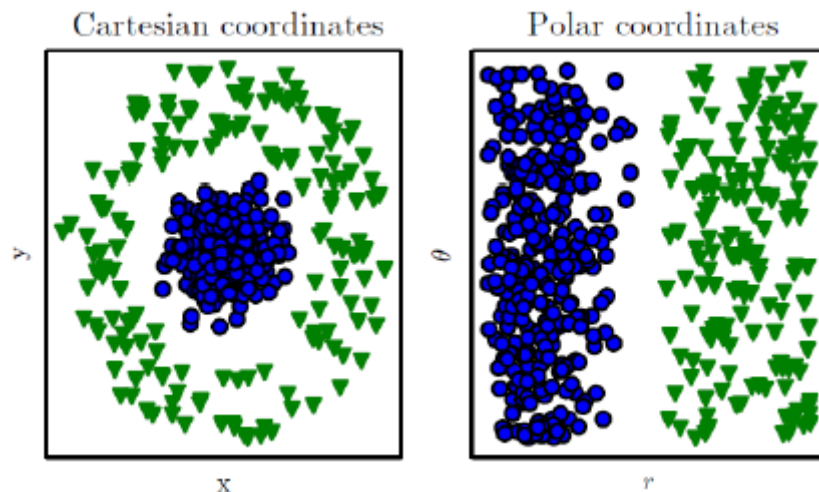


(a) 원래 특징 공간



(b) 분류에 더 유리하도록 변환된 새로운 특징 공간

- XOR 문제
- 좌표계를 변형시켜 선형에 가깝게 만든다
- 표현문제의 예



- 식으로 변환된 새로운 특징 공간
 - 공간변환을 통해 직선 모델로 100% 정확도

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

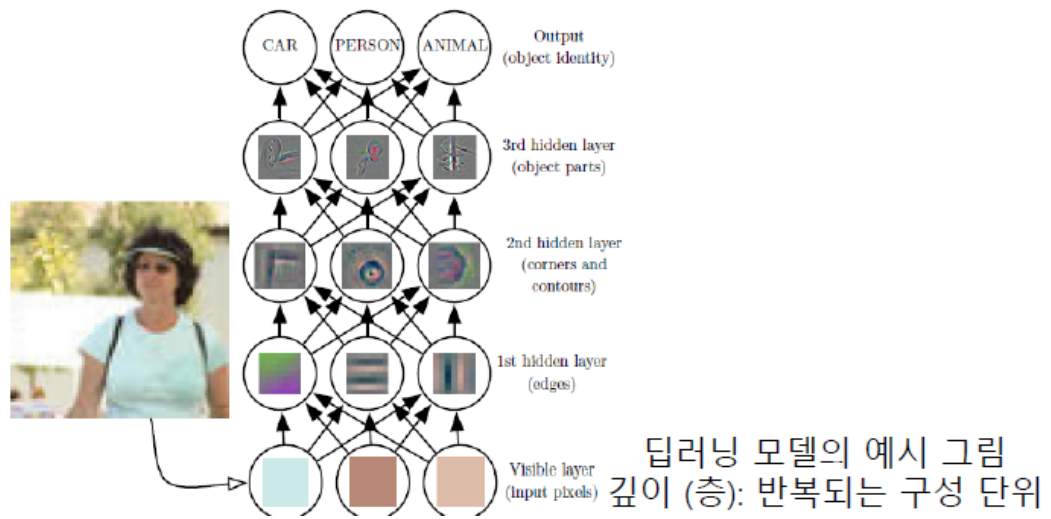
$$\mathbf{a} = (0,0)^T \rightarrow \mathbf{a}' = (0,0)^T$$

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

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

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

- 표현학습
 - 좋은 특징 공간을 자동으로 찾는 작업
 - 딥 러닝 (Deep Learning)
 - 다수의 은닉층을 가진 신경망을 이용하여 최적의 계층적인 특징 공간을 찾아냄
 - 아래쪽 은닉층은 저급 특징(선, 구석점 등), 위쪽은 고급 특징(얼굴, 바퀴 등) 추출



- 차원에 대한 몇가지 설명
 - 거리 : 차원에 무관하게 수식 적용 가능
 - 보통 2~3차원의 저차원에서 식을 고안해 고차원으로 확장 적용
- 차원의 저주(Curse of dimensionality)
 - 차원이 높아짐에 따라 발생하는 현실적인 문제들
 - 차원이 크면 클수록 데이터가 더 많이 필요하다