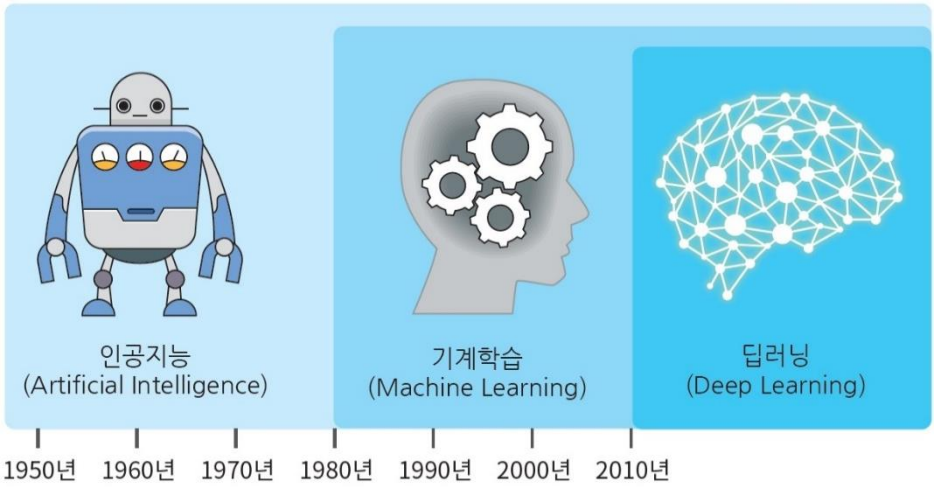
Chapter 01 컴퓨터와 데이터

0 인공지능


인공지능
(Artificial Intelligence)
기계학습
(Machine Learning)
딥러닝
(Deep Learning)
1950년
1960년
1970년
1980년
1990년
2000년
2010년

최근의 인공지능 활약

- 1997년 IBM의 딥블루: 체스시합에서 세계 챔피언이었던 카스퍼로프를 상대로 승리(인간을 넘어선 최초의 컴퓨터)
- 2011년 IBM의 왓슨: 퀴즈쇼 "제퍼디"에서 우승 차지
- 2016년 알파고(AlphaGo):구글의 인공지능 바둑 프로그램- 이세돌과의 경기에서 4-1로 승리, 2017년 1월 마스터(Master): 업그레이드된 알파고

인공지능 컴퓨터

- 1997년 IBM의 딥블루(Deep Blue)라는 컴퓨터가 세계 체스 챔피언인 개리 캐스파로프를 꺾으면서 다시 주목

https://www.youtube.com/watch?v=KF6sLCeBj0s

인공지능 컴퓨터

- 2011년에는 IBM의 왓슨(Watson)이 세계 최고의 퀴즈쇼인 제퍼디(Jeopardy)에서 그동안 전설적인 퀴즈 왕으로 꼽혔던 2명의 퀴즈 왕들을 상대로 한 대결에서 승리

https://www.youtube.com/watch?v=WFR3lOm_xhE&list=RDYgYSv2KSyWg&index=4

알파고

- https://www.youtube.com/watch?v=8tq1C8spV_g

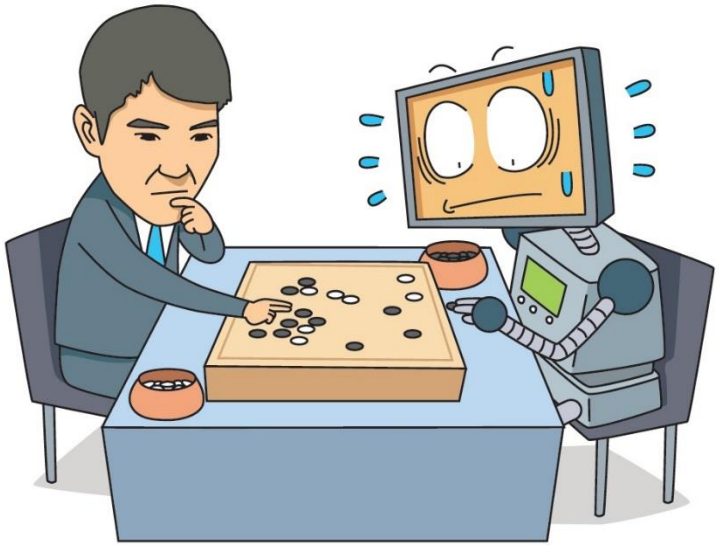


그림 1-1 알파고

알파고의 변신

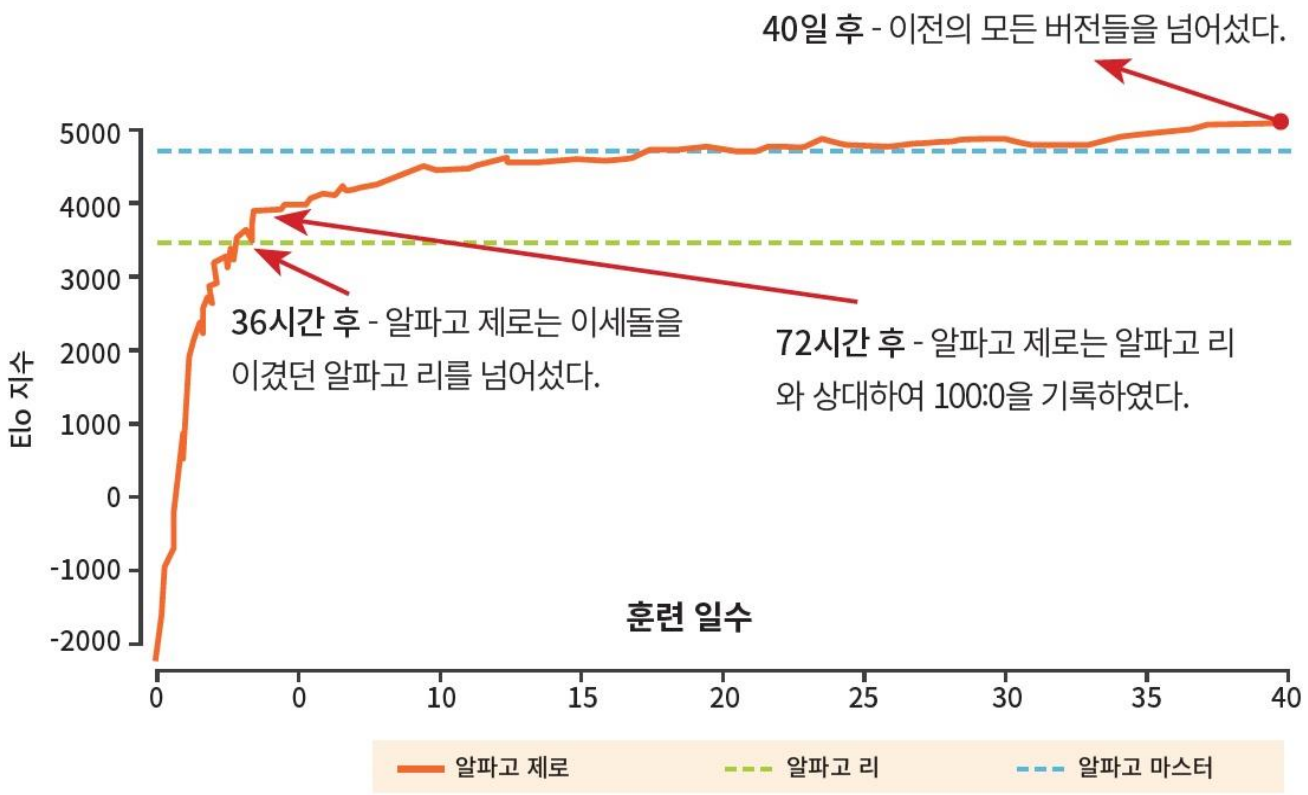


그림 1–2 알파고 제로(*출처: 딥 마인드)

이세돌-알파고 리 / 커제-알파고 마스터 => 알파고 제로(reinforcement learning system 적용)

자율 주행 자동차

- 인공지능 탑재 자율주행 자동차는 길 선택, 주행, 정차 모두 인공지능이 판단

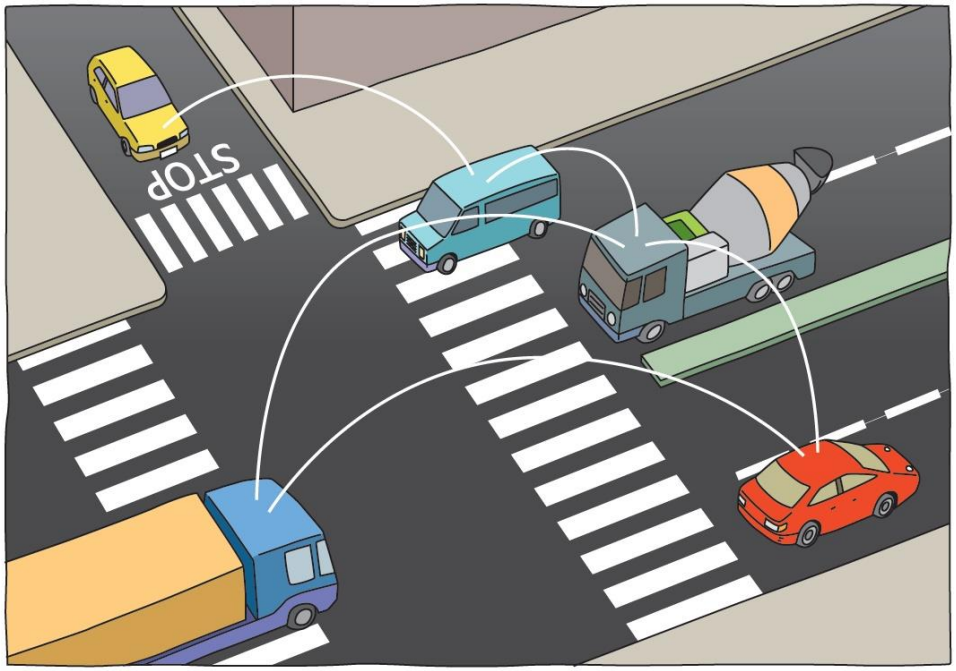


그림 1–36 연결된 자율주행 자동차의 개념

인공 지능은 어디에 필요할까?

- 음성인식: 필요한 것을 말하면 인터넷에 연결하여 자동주문한다.
  ex)Amazon의 알렉사

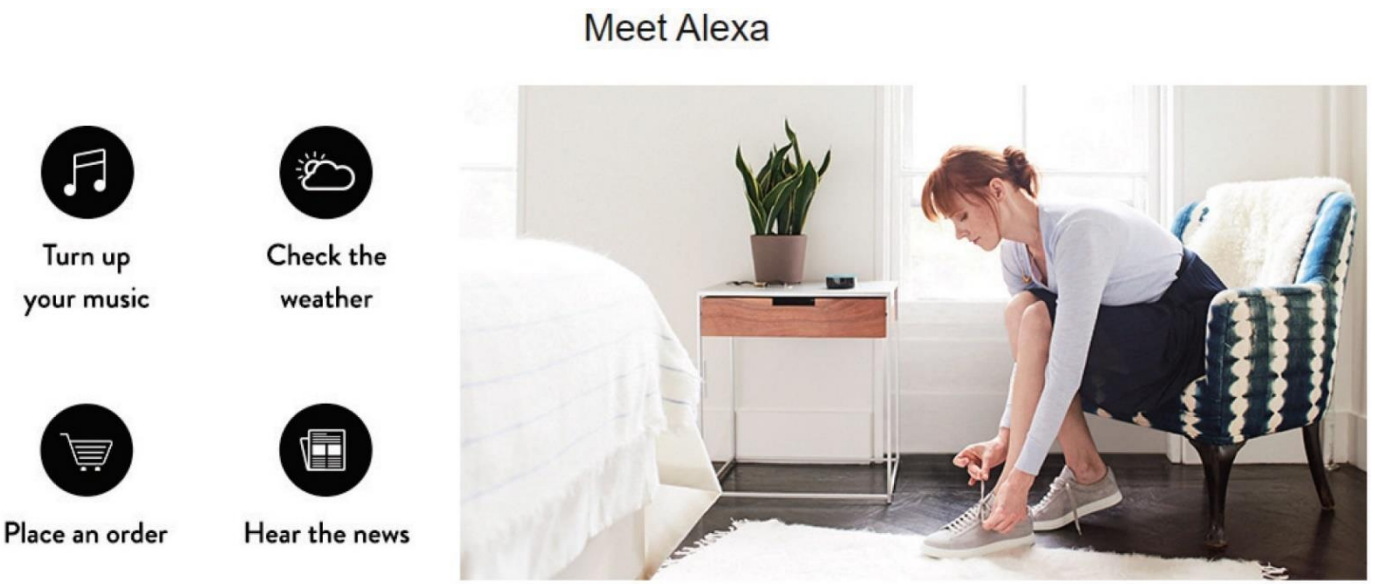


그림 1–5 아마존의 알렉사(*출처: 아마존)

인공지능의 시대

- **강인공지능(strong AI):**
  - 인공지능의 강한 형태
  - 자의식이 있다
  - 일반적인 영역에서의 문제도 해결하지만, 명령받지 않은 일도 스스로 필요하다면 해결할 수 있다
  - ex)터미네이터의 스카이넷

그림 1–6 영화 터미네이터

인공지능의 시대

- **약인공지능(weak AI): Data 필수**
  - 인공지능의 약한 형태
  - 자의식이 없다
  - 특정한 영역에서 주어진 문제를 해결
  - ex) 알파고

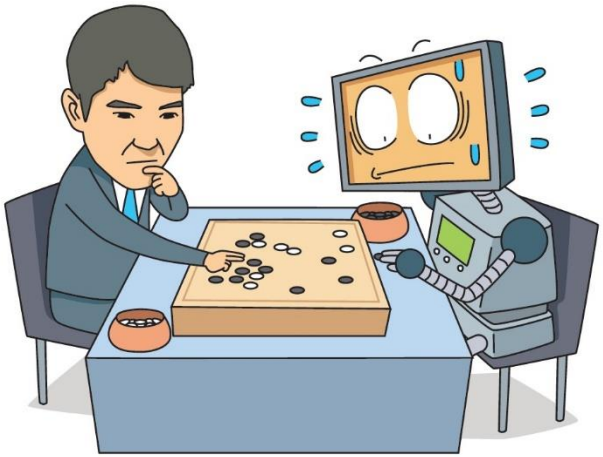


그림 1-1 알파고

인간과 인공지능

- 인간과 컴퓨터는 각각 장점과 약점을 가지고 있다. 인공지능이 탑재된 컴퓨터는 논리적으로 추론할 수도 있으며 학습도 가능하다. 인간은 계산은 늦지만 창의적으로 문제를 해결할 수 있다.
- 인간과 인공지능 컴퓨터는 좋은 동반자

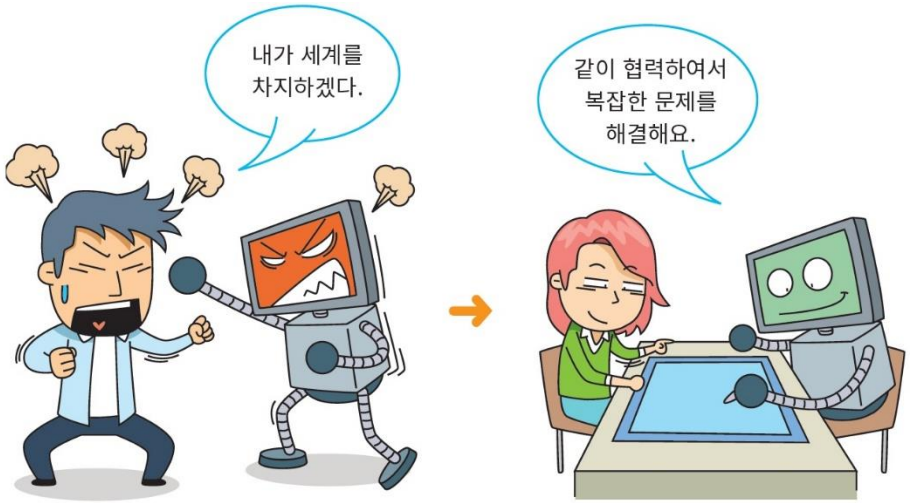


그림 1-7 인간과 인공지능

인공지능의 특징

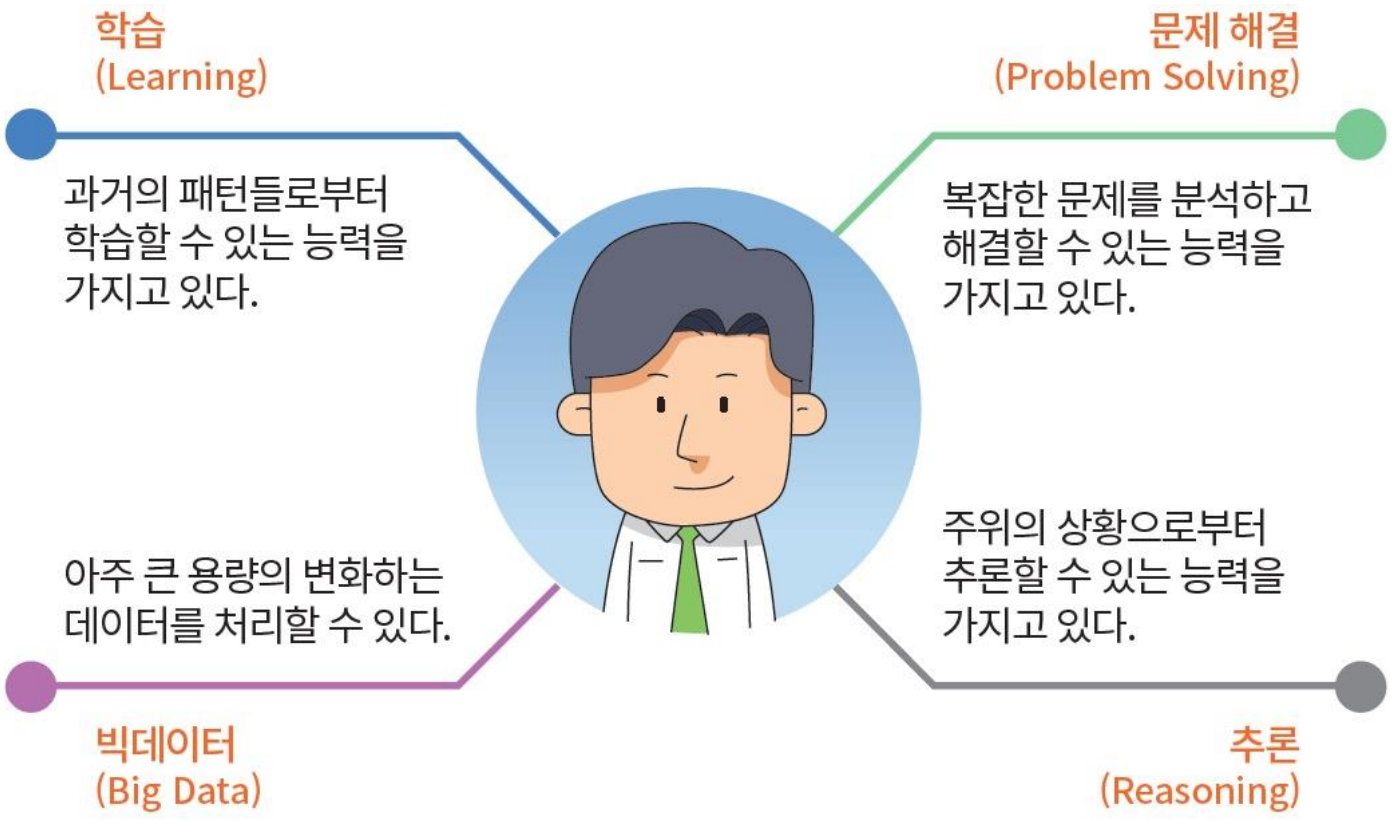


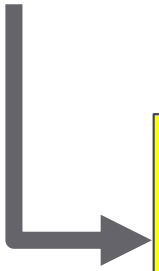
그림 1-8 인간의 지능

인공지능이란 무엇인가?

- 인공지능은 연구자들마다 정의가 다르다.
  - "인간처럼 사고하기"(Thinking Humanly) -Cognitive Science, 신경망
  - "합리적으로 사고하기"(Thinking Rationally)– 논리학, 추론
  - "인간처럼 행동하기"(Acting Humanly) - Turing Test, 로봇 공학
  - "합리적으로 행동하기"(Acting Rationally)- 에이전트: 목표를 성취하기 위해 행동, 추론을 포함

지능의 정의

1. 인간이 사물을 이해하고 학습하는 능력(learning)
2. 어떤 문제가 주어졌을 때, 합리적으로 사고하여 문제를 해결하는 능력(problem solving)



인공 지능이란 "인간의 인지적인 기능을 흉내 내어서 문제를 해결하기 위하여 학습하고 이해하는 기계(컴퓨터)"

인공지능 vs 머신러닝 vs 딥러닝

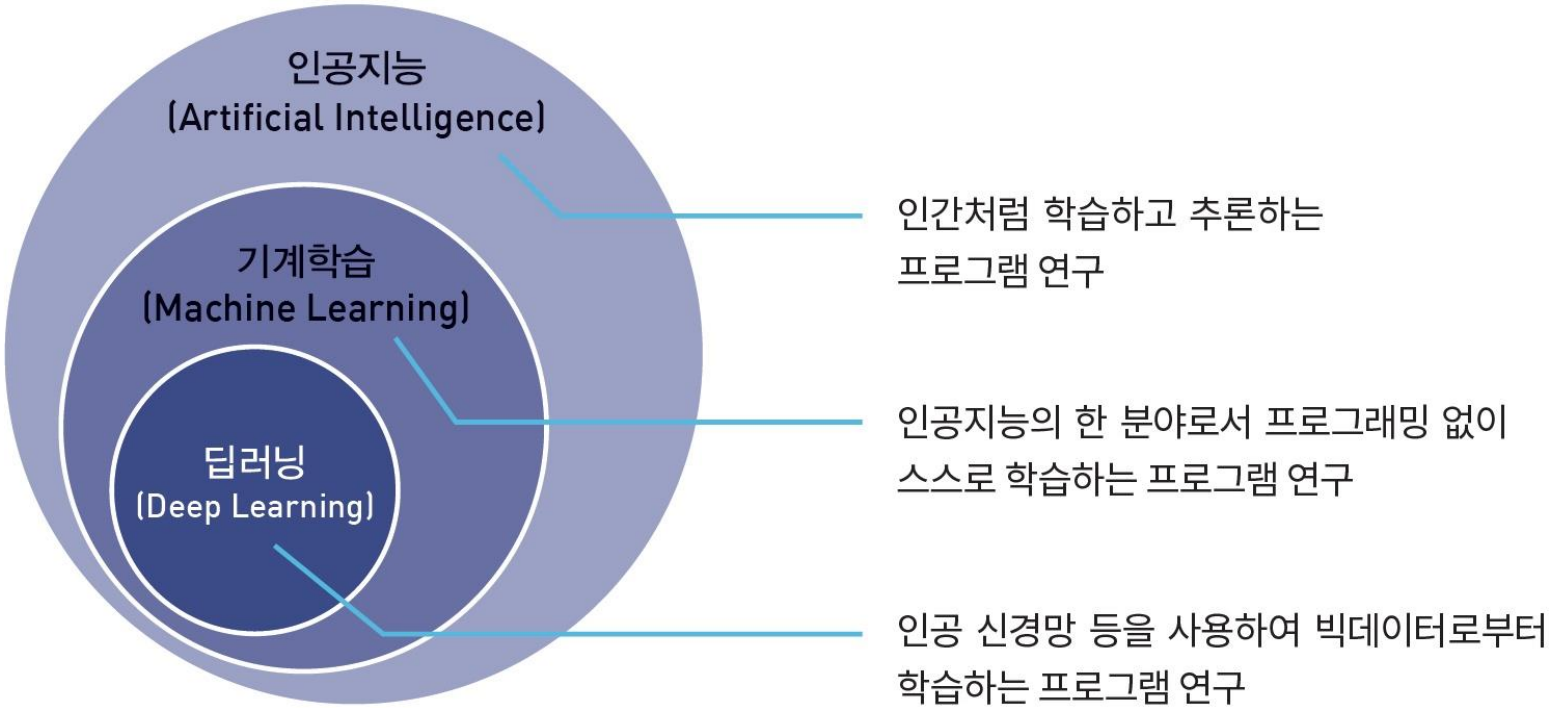


그림 1–9 인공지능, 기계학습, 딥러닝의 관계

머신러닝은 어디에 이용되는가?

인공지능의 역사

1943 McCulloch와 Pittes가 신경망의 기초 모델을 정립

1950 앨런 튜링이 기계 지능의 테스트 방법으로 튜링 테스트를 소개

1955 다트무스 학술대회에서 '인공지능(AI)'이란 용어 탄생

1965 오늘날 채팅봇의 원조인 ELIZA가 탄생

1980S Feigenbaum이 인간 전문가의 의사결정을 흉내내는 전문가 시스템 개발

2016 알파고가 이세돌을 이김

2011~2014 시리나 구글 나우같은 음성인식 개인 비서가 등장

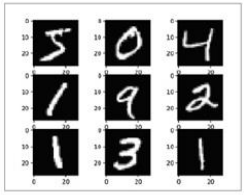
2011 IBM의 컴퓨터 왓슨이 게임쇼에서 인간을 누르고 우승

2009 구글이 자율주행 자동차를 선보임

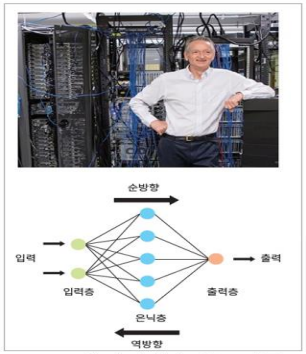
1997 IBM의 컴퓨터 딥블루가 체스 챔피언을 이김

그림 1-17 인공지능의 역사

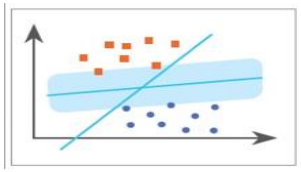
2012 ImageNet에서 우승



1998 필기체 인식



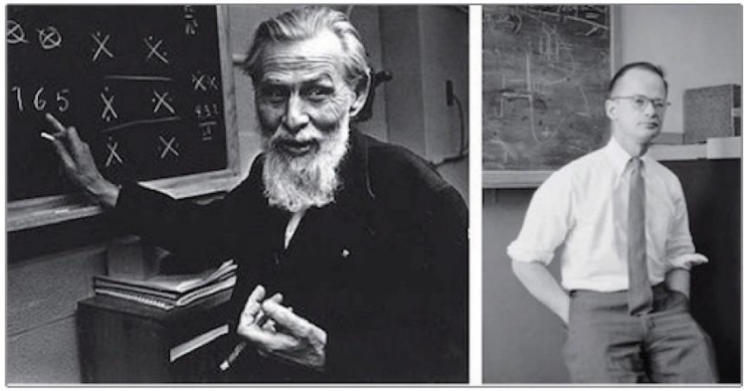
1985 역전파 학습 알고리즘



1995 SVM

인공지능의 태동

- 1943년에 Warren McCulloch과 Walter Pitts는 뉴런들의 간단한 네트워크를 분석하고 이것이 간단한 논리 기능을 수행할 수 있음을 보여주었다. 이것들은 나중에 연구자들이 인공 신경망이라고 부르게 되었다.



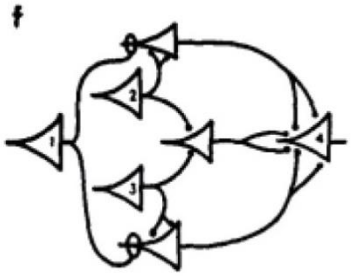


그림 1-18 Warren McCulloch와 Walter Pitts, 그들이 만들었던 신경망

퍼셉트론

- 인공 신경망의 초기 형태인 퍼셉트론(perceptron)을 Frank Rosenblatt가 개발하였다. Rosenblatt는 "퍼셉트론은 궁극적으로 언어를 배우고 결정하며 언어를 번역할 수 있게 될 것"이라고 예측하여 낙관적인 입장을 보였다. Minsky와 Papert의 1969년 저서 '퍼셉트론 (Perceptrons)' 이 발표되면서 갑작스럽게 중단되었다.

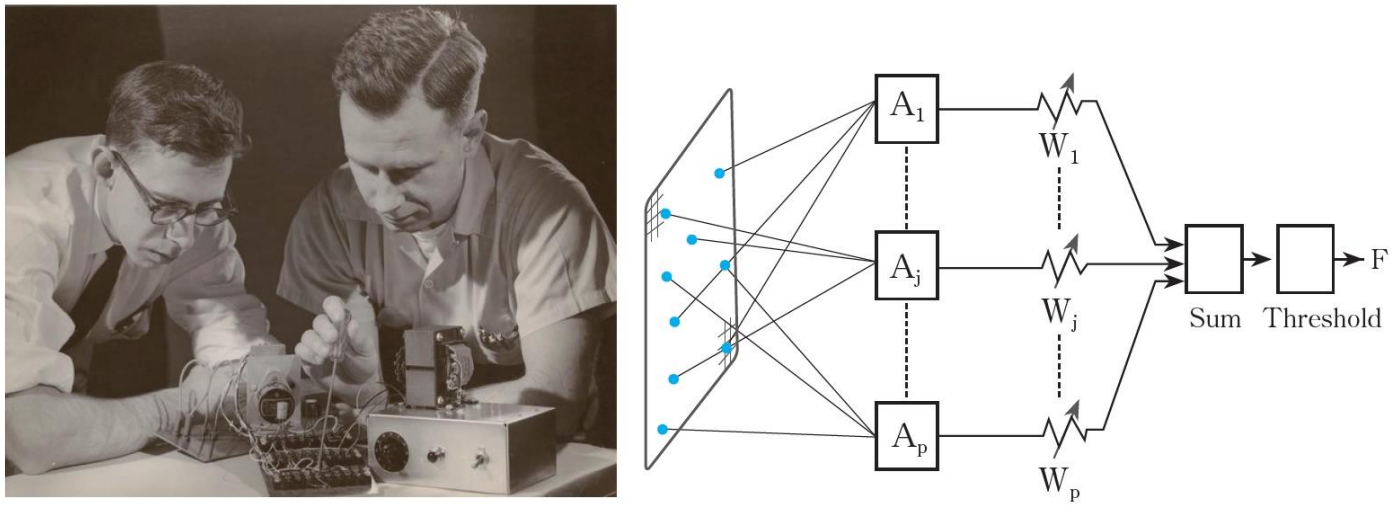


그림 1–20 Rosenblatt와 퍼셉트론

다트머스 학술 대회

- 1956년에 의해 다트머스 학술 회의가 Marvin Minsky와 John MacCarthy 등에 의하여 조직되었다

그림 1-21 다트머스 학술 회의 참가자들

첫 번째 AI 겨울

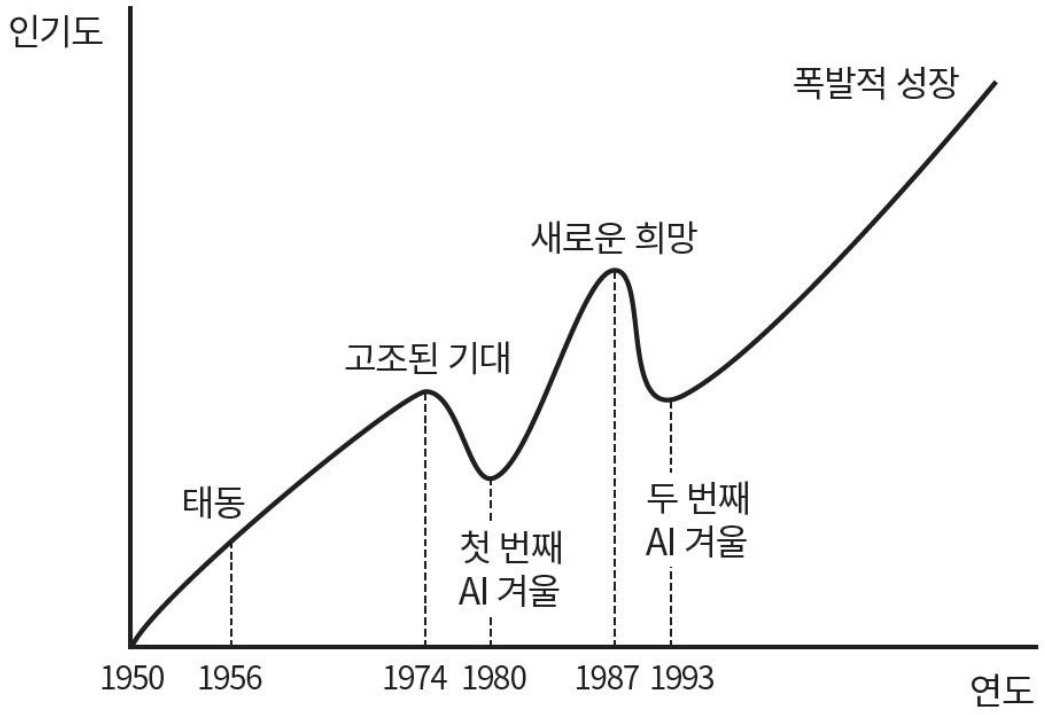


그림 1-24 인공지능의 부침

당시의 문제점

- 첫 번째로 1970년대에는 충분한 컴퓨팅 파워가 없었다. 실제로 유용한 결과를 내는데 필요한 CPU의 속도나 충분한 메모리가 없었다.

- 두 번째로 "장난감 문제"가 있다. 인공 지능 분야에서는 지수적 시간에만 풀 수 있는 많은 현실적인 문제가 있다. 따라서 이러한 현실적인 문제에 대한 최적의 솔루션을 찾는 데는 상상할 수 없는 양의 계산 시간이 필요하다.

- 세 번째로 컴퓨터 시각이나 자연어 처리와 같은 많은 인공 지능 응용 프로그램은 전 세계에 대한 엄청난 양의 정보를 필요로 한다. 1970년에는 아무도 이 정도의 데이터베이스를 만들 수 없었고 어떤 프로그램도 이 방대한 정보를 어떻게 학습해야 하는지를 알지 못했다.

전성 시대

- 연구자들은 이 세상의 모든 문제를 해결할 수 있는 시스템을 개발한다는 생각을 버렸다.
- 이에 새롭게 등장한 시스템이 "전문가 시스템(expert system)"이다.

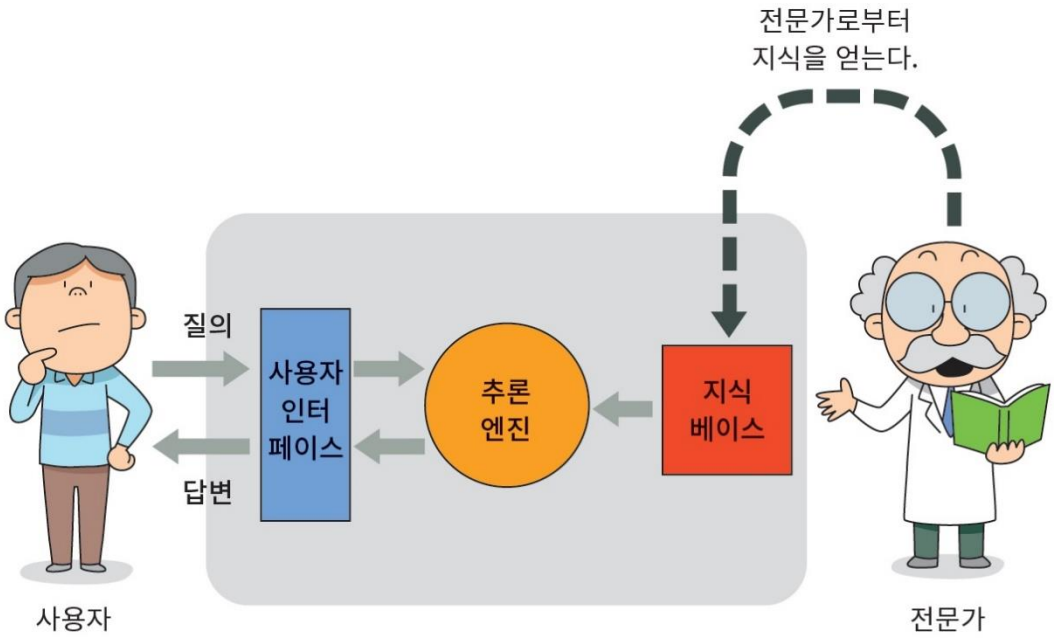


그림 1-27 전문가 시스템

전문가 시스템

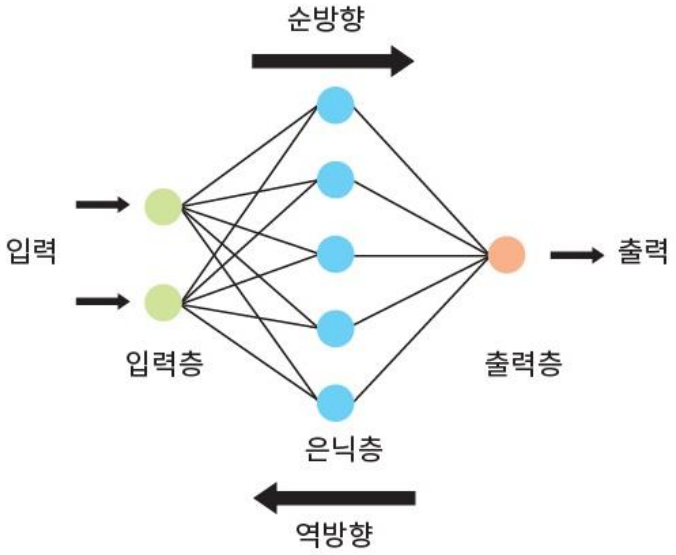
- DENDRAL은 분광계 수치로 화합물을 분석하는 전문가 시스템으로 스탠포드 대학교의 Edward Feigenbaum과 그의 학생들에 의해 개발되었다.

- MYCIN은 전염성 질환을 진단하고 항생제를 처방하는 전문가 시스템이었다.

신경망의 부활

- 1982년 물리학자 John Hopfield는 완전히 새로운 방식으로 정보를 학습하고 처리할 수 있는 한 형태의 신경망(Hopfield Net)을 제안

- Geoffrey Hinton과 David Rumelhart는 "역전파(backpropagation)"라고 불리는 유명한 학습 방법을 대중화



두 번째 AI 겨울

- 전문가 시스템은 유용했지만 몇 가지 특수한 상황에서만 유용함이 밝혀졌다.
- 1980년대 후반, 미국의 전략적 컴퓨팅 구상(Strategic Computing Initiative)은 AI에 대한 기금을 잔인하게 삭감했다.

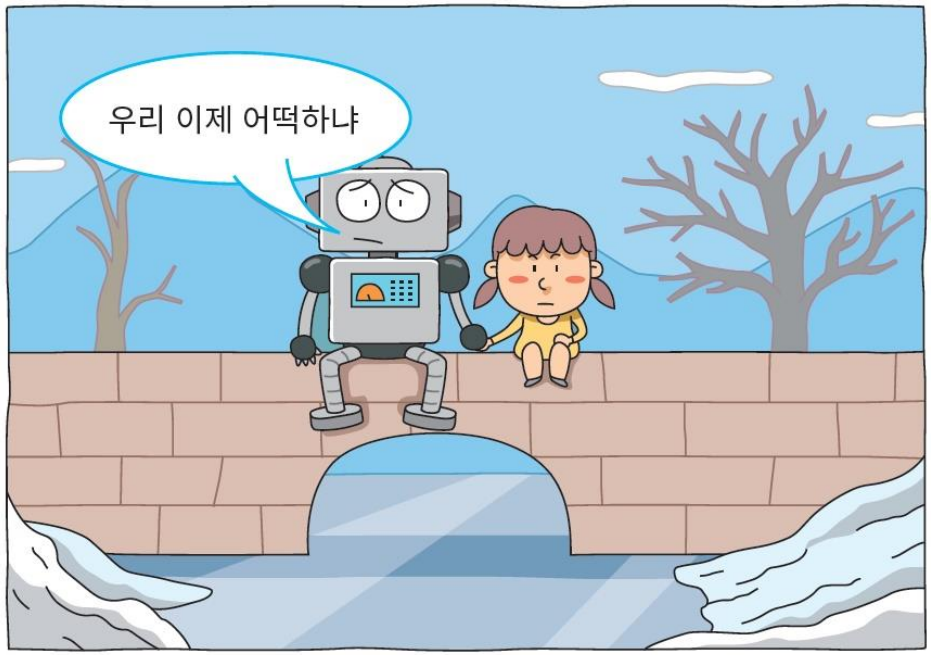


그림 1-30 AI의 겨울

인공지능의 응용 분야

- 자동차 업계에서는 이미지 인식 기술을 바탕으로 한 자율 주행 자동차 개발에 심혈을 기울이고 있다.

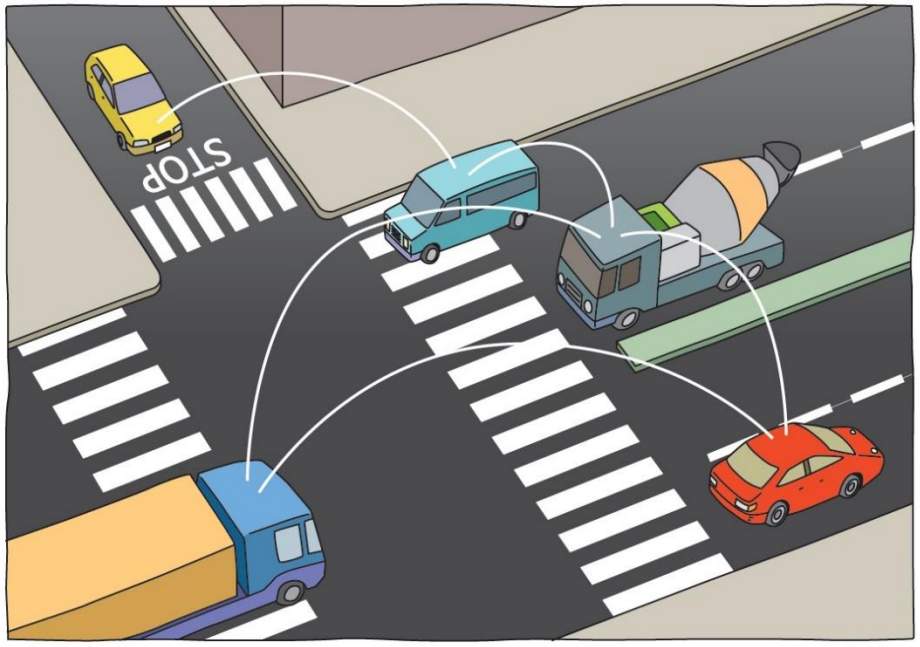


그림 1-36 연결된 자율주행 자동차의 개념

인공지능의 응용 분야(광고)

- 인공지능은 현재 사용자가 보고 있는 웹사이트의 컨텐츠와 가장 유사한 상품이나 기사를 추천한다.

그림 1–37 인공지능 추천 시스템

인공지능의 응용 분야(챗봇)

- 오늘날 챗봇은 Google Assistant 및 Amazon Alexa와 같은 가상 어시스턴트 , Facebook Messenger 또는 WeChat과 같은 메시징 앱 이나 웹 사이트를 통해 사용된다.

인공지능의 응용 분야(의료분야)

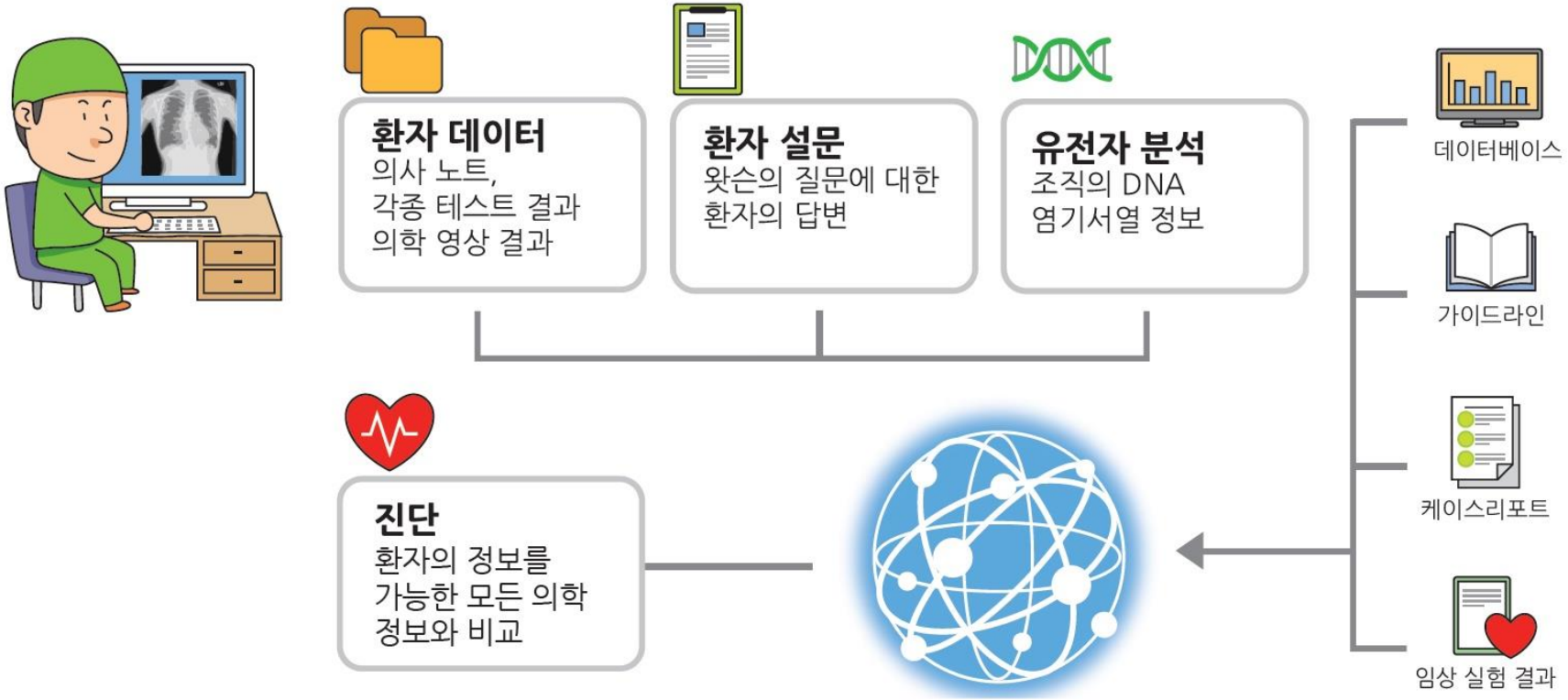


그림 1–39 왓슨을 사용한 의료 진단

1장 컴퓨터와 데이터

1.1 데이터를 지식으로 바꾸는 지능

1.1 데이터를 지식으로 바꾸는 지능

- **데이터를 지식으로 바꾸는 지능**
  - 현대 기술 시대에는 정형 또는 비정형 데이터가 매우 풍부함
  - 20세기 후반에 데이터에서 지식을 추출하여 예측하는 자기 학습(self-learning) 알고리즘과 관련된 **인공 지능**(Artificial Intelligence, AI)의 하위 분야로 머신 러닝이 출현
  - 사람이 수동으로 대량의 데이터를 분석하여 규칙을 유도하고 모델을 만듦
  - 머신 러닝(Machine Learning)이 데이터에서 더 효율적으로 지식을 추출하여 예측 모델과 데이터 기반의 의사 결정 성능을 점진적으로 향상시킬 수 있음

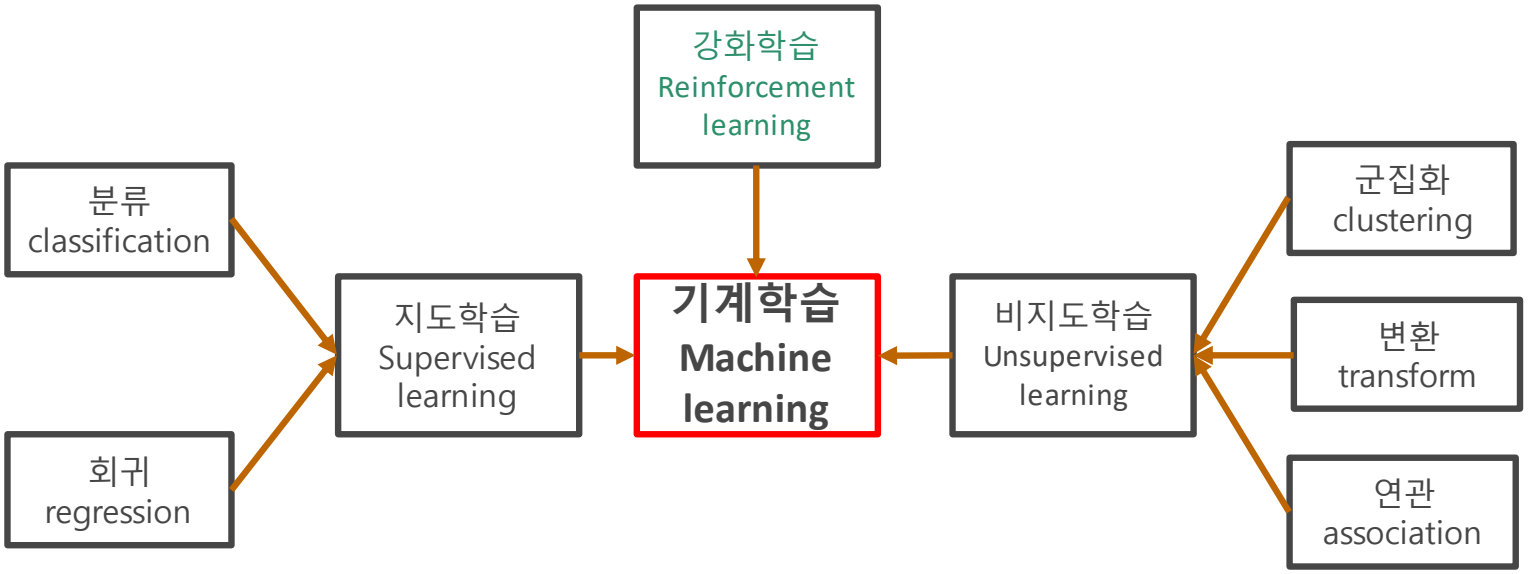
1.1 데이터를 지식으로 바꾸는 지능적인 시스템 구축

- **데이터를 지식으로 바꾸는 지능적인 시스템 구축**
  - 컴퓨터 과학 연구에서 머신 러닝은 점점 더 중요해지며, 우리 일상생활에서도 아주 큰 역할을 하고 있음
  - 머신 러닝 덕택에 견고한 이메일 스팸 필터, 편리한 텍스트와 음성 인식 소프트웨어, 믿을 수 있는 웹 검색 엔진, 체스 대결 프로그램을 사용
  - 아마도 곧 안전하고 효율적인 자율 주행 자동차도 사용할 수 있을 것
  - 또한, 의료 애플리케이션에서도 큰 진전이 있었음
  - 예를 들어 연구자들은 딥러닝 모델을 사용하여 피부암을 거의 사람 수준의 정확도로 진단 할 수 있다는 것을 보였음(https://www.nature.com/articles/nature21056)
  - 최근에는 딥마인드(DeepMind)의 연구원들이 또 하나의 이정표를 세웠음
  - 딥러닝으로 3D 단백질 구조를 예측하여 처음으로 물리학 기반 방식의 성능을 뛰어넘었음(https://deepmind.com/blog/alphafold/)

1.2 머신 러닝의 세 가지 종류

1.2 머신 러닝의 세 가지 종류

- **머신 러닝의 세 가지 종류**
  - **지도 학습** (supervised learning),
  - **비지도 학습**(unsupervised learning),
  - **강화 학습**(reinforcement learning)

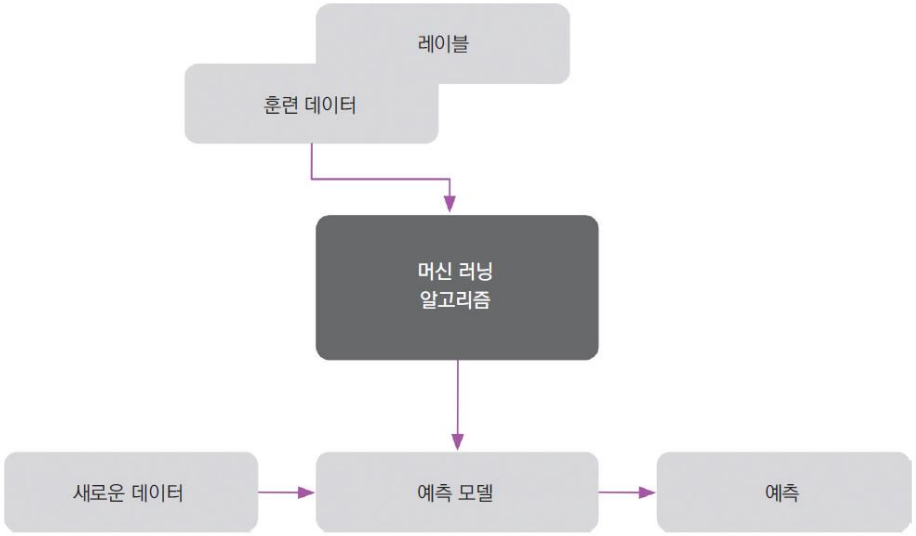


1.2 머신 러닝의 세 가지 종류

▼ 그림 1-1 머신 러닝의 세 가지 학습 종류

1.2 머신 러닝의 세 가지 종류

- **지도 학습으로 미래 예측**
  - 지도 학습의 주요 목적은 레이블(label)된 훈련 데이터에서 모델을 학습하여 본 적 없는 미래 데이터에 대해 예측을 만드는 것
  - 여기서 **지도**(supervised)는 희망하는 출력 신호(레이블)가 있는 일련의 샘플(데이터 입력)을 의미
  - 그림 1 - 2는 전형적인 지도 학습 작업 흐름을 나타냄
  - 레이블된 훈련 데이터(training data)가 머신 러닝 알고리즘에 전달되어 예측 모델을 훈련하고, 그 다음 새로운 레이블되지 않은 데이터 입력에 대해 예측을 수행



1.2 머신 러닝의 세 가지 종류

- **지도 학습으로 미래 예측**
  - 스팸 메일을 필터링하는 예를 생각해 보자
  - 레이블된 이메일 데이터셋에서 지도 학습 머신 러닝 알고리즘을 사용하여 모델을 훈련할 수 있음
  - 이 데이터셋은 스팸 또는 스팸이 아닌 이메일로 정확하게 표시되어 있음
  - 훈련된 모델은 새로운 이메일이 두 개의 범주(category) 중 어디에 속하는지 예측
  - 이메일 스팸 필터의 예처럼 개별 클래스 레이블이 있는 지도 학습을 **분류**(classification)
  - 지도 학습의 또 다른 종류는 연속적인 값을 출력하는 **회귀**(regression)

1.2 머신 러닝의 세 가지 종류

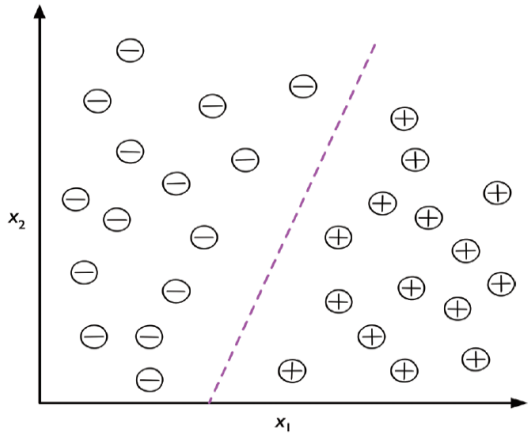
- **지도 학습으로 미래 예측**

  분류: 클래스 레이블 예측

  - 분류는 지도 학습의 하위 카테고리
  - 과거의 관측을 기반으로 새로운 샘플의 범주형 클래스 레이블을 예측하는 것이 목적
  - 클래스 레이블은 이산적(discrete)이고 순서가 없어 샘플이 속한 그룹으로 이해할 수 있음
  - 앞서 언급한 스팸 메일 감지는 전형적인 **이진 분류**(binary classification) 작업의 예
  - 스팸과 스팸이 아닌 이메일 두 개의 클래스 사이를 구분하려고 머신 러닝 알고리즘이 일련의 규칙을 학습

1.2 머신 러닝의 세 가지 종류

- **지도 학습으로 미래 예측**
  - 그림 1-3은 30개의 훈련 샘플이 있는 이진 분류 작업의 개념을 나타냄
  - 15개의 샘플은 **음성 클래스**(negative class)로 레이블(뺄셈 기호)되어 있음
  - 다른 15개의 샘플은 **양성 클래스**(positive class)로 레이블(덧셈 기호)되어 있음
  - 각 샘플이 두 개의 x_1, x_2 값에 연관되어 있으므로 2차원 데이터셋
  - 지도 학습 알고리즘을 사용하여 두 클래스를 구분할 수 있는 규칙을 학습
  - 이 규칙은 점선으로 나타난 **결정 경계**(decision boundary)
  - 새로운 데이터의 x_1, x_2 값이 주어지면 두 개의 범주 중 하나로 분류



1.2 머신 러닝의 세 가지 종류

- **지도 학습으로 미래 예측**
  - 두 개 이상의 클래스 레이블을 가진 경우가 많음
  - 지도 학습 알고리즘으로 학습한 예측 모델은 훈련 데이터셋에 있는 클래스 레이블을 새로운 샘플에 할당할 수 있음
  - 이런 **다중 분류**(multiclass classification)의 전형적인 예는 손으로 쓴 글자 인식
  - 글자("A", "B", "C" 등)는 예측하려는 대상이며 순서가 없는 범주나 클래스 레이블로 표현
  - 알파벳 각 글자를 손으로 쓴 이미지 샘플을 모아서 훈련 데이터 셋을 구성
  - 새로운 글자를 입력으로 제공하면 예측 모델이 일정한 정확도로 알파벳 글자를 예측할 것
  - 0에서 9까지 숫자가 훈련 데이터셋에 없다면 이 머신 러닝 시스템은 숫자를 인식하지 못할 것

1.2 머신 러닝의 세 가지 종류

- **지도 학습으로 미래 예측**

  회귀(Regression): 연속적인 출력 값 예측

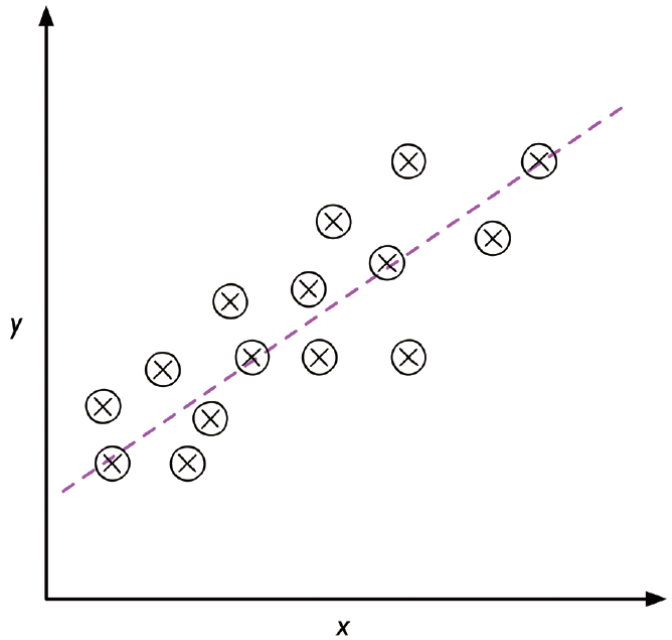
  - 이전 절에서 분류 작업은 범주형 순서가 없는 레이블을 샘플에 할당하는 것이라고 배웠음
  - 두 번째 지도 학습의 종류는 연속적인 출력 값을 예측하는 **회귀** 분석
  - 회귀는 **예측 변수**(predictor variable)(**설명 변수**(explanatory variable), 독립 변수(independence variable)와 연속적인 **반응 변수**(response variable)(**결과**(outcome), 종속변수(dependence variable)가 주어졌을 때 출력 값을 예측하기 위해 두 변수 사이의 관계를 찾음

1.2 머신 러닝의 세 가지 종류

- **지도 학습으로 미래 예측**
  - 머신 러닝 분야에서는 예측 변수를 보통 "특성(feature, 속성(attribute))"이라고 부르며, 반응 변수를 "타깃(target, 레이블Label)"이라고 부름
  - 시험 공부에 투자한 시간과 최종 점수 사이에 관계가 있다면 두 값으로 훈련 데이터를 만들고 모델을 학습할 수 있음
  - 이 모델은 시험에 응시하려는 학생들이 공부한 시간을 이용하여 시험 점수를 예측

1.2 머신 러닝의 세 가지 종류

- **지도 학습으로 미래 예측**
  - 그림 1-4는 **선형 회귀**(linear regression)의 개념을 나타냄
  - 특성 x와 타깃 y가 주어지면 데이터 포인트와 직선 사이 거리가 최소가 되는 직선을 그을 수 있음
  - 일반적으로 평균 제곱 거리를 사용
  - 데이터에서 학습한 직선의 기울기와 절편(intercept)을 사용하여 새로운 데이터의 출력 값을 예측



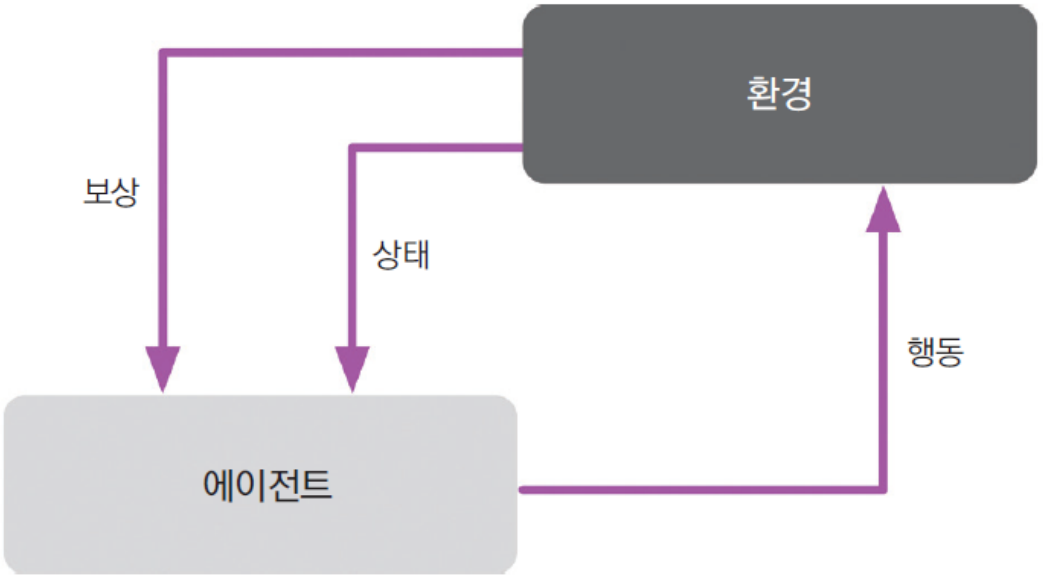
1.2 머신 러닝의 세 가지 종류

- **강화 학습으로 반응형 문제 해결**
  - **강화 학습**은 머신 러닝의 또 다른 종류
  - 강화 학습은 환경과 상호 작용하여 시스템(**에이전트**(agent)) 성능을 향상하는 것이 목적
  - 환경의 현재 상태 정보는 **보상**(reward) 신호를 포함하기 때문에 강화 학습을 지도 학습과 관련된 분야로 생각할 수 있음
  - 강화 학습의 피드백은 정답(ground truth) 레이블이나 값이 아님
  - 보상 함수로 얼마나 행동이 좋은지를 측정한 값
  - 에이전트는 환경과 상호 작용하여 보상이 최대화되는 일련의 행동을 강화 학습으로 학습
  - 탐험적인 시행착오(trial and error) 방식이나 신중하게 세운 계획을 사용
  - 강화 학습의 대표적인 예는 체스 게임
  - 에이전트는 체스판의 상태(환경)에 따라 기물의 이동을 결정
  - 보상은 게임을 종료했을 때 승리하거나 패배하는 것으로 정의할 수 있음

1.2 머신 러닝의 세 가지 종류

- **강화 학습으로 반응형 문제 해결**
  - 강화 학습에는 여러 하위 분류가 있음
  - 일반적인 구조는 강화 학습 에이전트가 환경과 상호 작용하여 보상을 최대화하는 것
  - 각 상태는 양의 보상이나 음의 보상과 연관됨
  - 보상은 체스 게임의 승리나 패배처럼 전체 목표를 달성하는 것으로 정의할 수 있음
  - 예를 들어 체스에서 기물의 이동으로 나타난 결과는 각기 다른 환경 상태로 생각할 수 있음

- environment,
- reward,
- state,
- action,
- agent



1.2 머신 러닝의 세 가지 종류

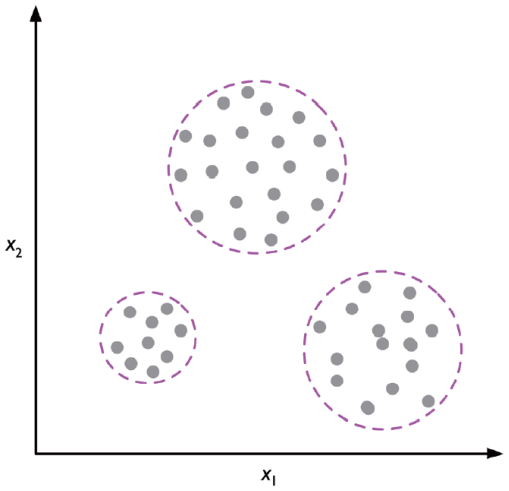
- **비지도 학습으로 숨겨진 구조 발견**
  - 지도 학습에서는 모델을 훈련할 때 사전에 옳은 답을 알고 있음
  - 강화 학습에서는 에이전트의 특정 행동을 보상하는 방법을 정의
  - 비지도 학습에서는 레이블되지 않거나 구조를 알 수 없는 데이터를 다룸
  - 비지도 학습 기법을 사용하면 알려진 출력 값이나 보상 함수의 도움을 받지 않고 의미 있는 정보를 추출하기 위해 데이터 구조를 탐색할 수 있음

1.2 머신 러닝의 세 가지 종류

- **비지도 학습으로 숨겨진 구조 발견**

  군집: 서브그룹 찾기

  - **군집**(clustering)은 사전 정보 없이 쌓여 있는 그룹 정보를 의미 있는 서브그룹(subgroup) 또는 **클러스터**(cluster)로 조직하는 탐색적 데이터 분석 기법
  - 분석 과정에서 만든 각 클러스터는 어느 정도 유사성을 공유하고 다른 클러스터와는 비슷하지 않은 샘플 그룹을 형성
  - 이따금 군집을 **비지도 분류**(unsupervised classification)라고 하는 이유가 여기 있음
  - 클러스터링은 정보를 조직화하고 데이터에서 의미 있는 관계를 유도하는 훌륭한 도구
  - 그림 1- 6은 군집이 어떻게 레이블되지 않은 유사도를 기반으로 세 개의 개별적인 그룹으로 조직



1.2 머신 러닝의 세 가지 종류

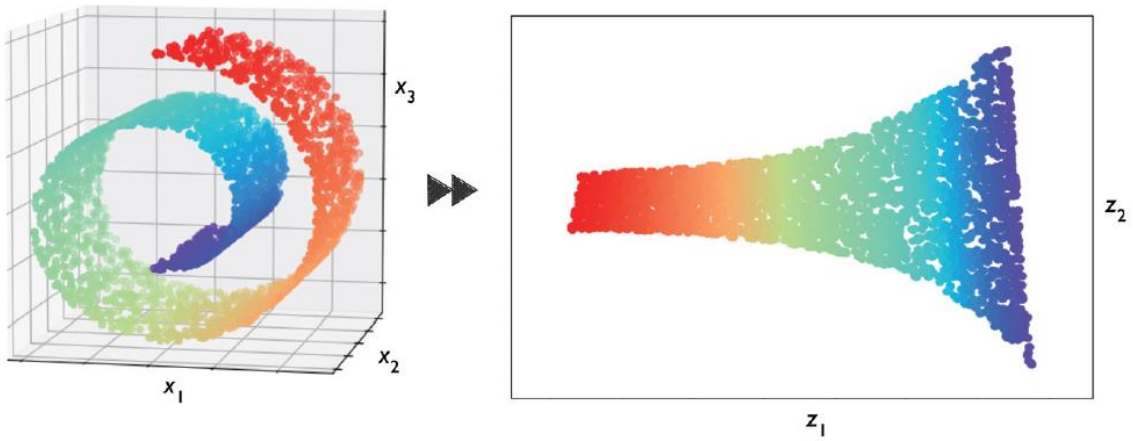
- **비지도 학습으로 숨겨진 구조 발견**

  차원 축소: 데이터 압축

  - 비지도 학습의 또 다른 하위 분야는 **차원 축소**(dimensionality reduction)
  - 고차원의 데이터를 다루어야 하는 경우는 흔함
  - 즉, 하나의 관측 샘플에 많은 측정 지표가 있음
  - 이로 인해 머신 러닝 알고리즘의 계산 성능과 저장 공간의 한계에 맞닥뜨릴 수 있음
  - 비지도 차원 축소는 잡음(noise) 데이터를 제거하기 위해 특성 전처리 단계에서 종종 적용하는 방법

1.2 머신 러닝의 세 가지 종류

- **비지도 학습으로 숨겨진 구조 발견**
  - 이런 잡음 데이터는 특정 알고리즘의 예측 성능을 감소시킬 수 있음
  - 차원 축소는 관련 있는 정보를 대부분 유지하면서 더 작은 차원을 가진 부분 공간(subspace)으로 데이터를 압축
  - 이따금 차원 축소는 데이터 시각화에도 유용
  - 예를 들어 고차원 특성을 1차원 또는 2차원, 3차원 특성 공간으로 투영하여 3D와 2D 산점도(scatterplot)나 히스토그램(histogram)으로 시각화
  - 그림 1-7은 비선형(nonlinear) 차원 축소를 적용하여 3D 스위스롤(Swiss Roll) 모양의 데이터를 새로운 2D 특성의 부분 공간으로 압축하는 예를 보여 줌



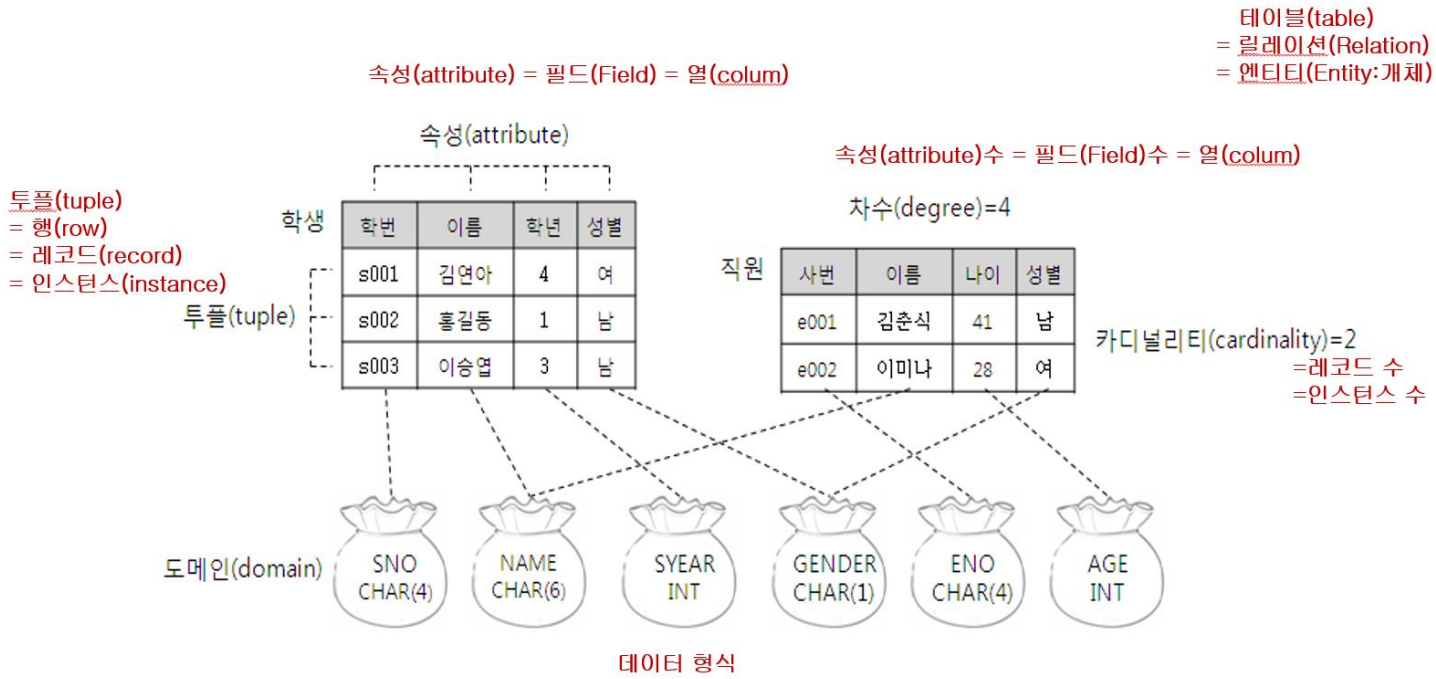
1.3 기본 용어와 표기법 소개

1.3 기본 용어와 표기법 소개

- **기본 용어와 표기법 소개**
  - 지금까지 지도 학습, 비지도 학습, 강화 학습 세 개의 머신 러닝 종류를 살펴보았음
  - 머신 러닝은 방대한 분야이고 여러 학문이 관련되어 있기 때문에 여러 다른 용어로 같은 개념을 설명하는 경우를 머지않아 만나게 될 것

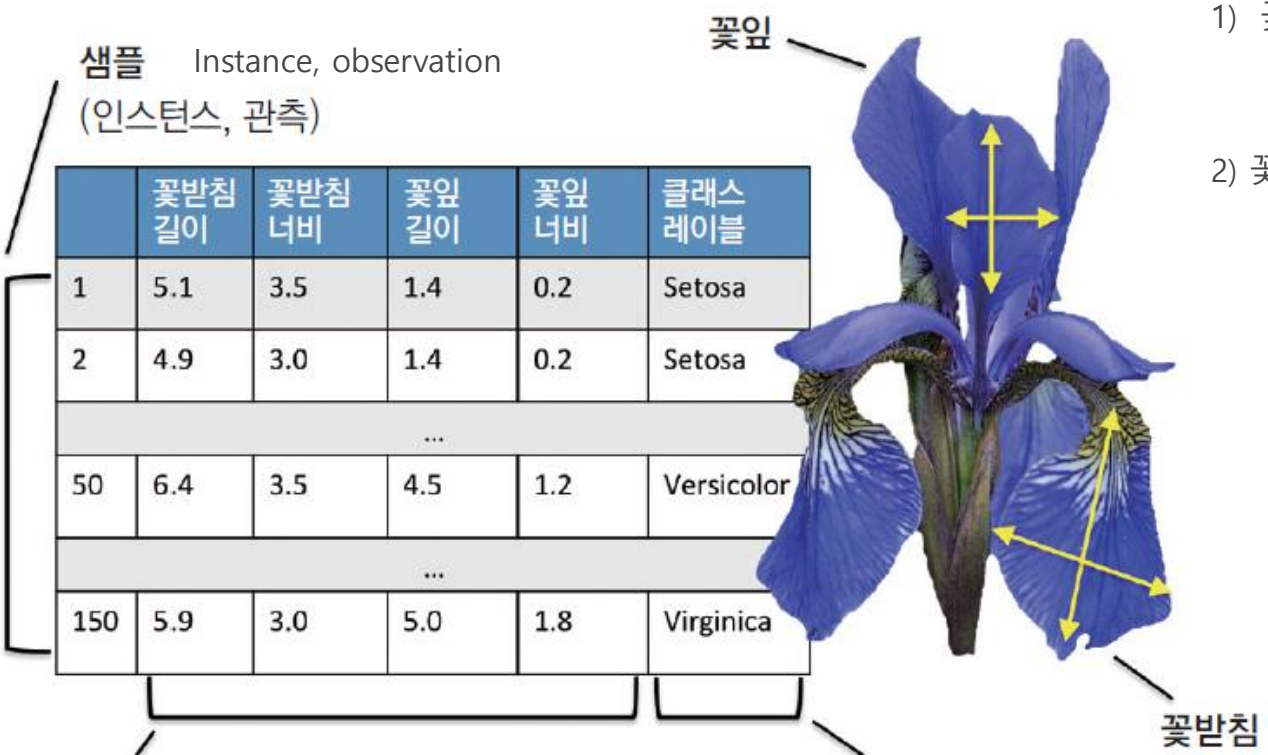
1.3 기본 용어와 표기법 소개

- **이 책에서 사용하는 표기법과 규칙**
  - 그림 1-8의 표는 머신 러닝 분야의 고전적인 예제인 붓꽃(Iris) 데이터셋 일부를 보여 줌
  - 붓꽃 데이터셋은 Setosa, Versicolor, Virginica 세 종류 150개의 붓꽃 샘플을 담고 있음
  - 각 붓꽃 샘플은 데이터셋에서 하나의 행(row)으로 표현
  - 센티[
    특성(



1.3 기본 용어와 표기법 소개

▼ **그림 1-8 붓꽃 데이터셋**



	꽃받침 길이	꽃받침 너비	꽃잎 길이	꽃잎 너비	클래스 레이블
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

1) 꽃잎 : Petal
   Petal - length
   Petal – width

2) 꽃받침 : Sepal
   Sepal – length
   Sepal - width

1.3 기본 용어와 표기법 소개

- **이 책에서 사용하는 표기법과 규칙**
  - 간단하게 표기하고 효율적으로 코드를 구현할 수 있도록 기초적인 선형대수학(linear algebra)을 사용
  - 다음 장부터는 행렬(matrix)과 벡터(vector) 표기로 데이터를 표현
  - 일반적인 관례에 따라서 샘플은 특성 행렬 **X**에 있는 행으로 나타내고, 특성은 열을 따라 저장
  - 150개의 샘플과 네 개의 특성을 가진 붓꽃 데이터셋은 150×4 크기의 행렬 $\boldsymbol{X} \in \mathbb{R}^{150\times4}$ 로 쓸 수 있음

$$\begin{bmatrix} x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & x_4^{(1)} \\ x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & x_4^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{(150)} & x_2^{(150)} & x_3^{(150)} & x_4^{(150)} \end{bmatrix}$$

1.3 기본 용어와 표기법 소개

- **머신 러닝 용어**
  - 머신 러닝은 여러 연구 분야의 과학자들이 관여하기 때문에 분야가 방대하고 관련된 학문이 많음
  - 이 과정에서 많은 용어와 개념이 재발견되거나 재정의되었고 이미 알고 있는 내용이 다른 이름으로 등장하기도 함

1.3 기본 용어와 표기법 소개

- **머신 러닝 용어**
  - 다음 목록에서 이 책과 다른 머신 러닝 책을 읽을 때 자주 등장하는 용어와 동의어를 정리

훈련 샘플: 데이터셋을 나타내는 테이블의 행

동의어로는 관측(observation), 레코드(record), 인스턴스(instance), 예시(example)가 있음(대부분의 경우 샘플은 훈련 예시의 집합을 의미)

훈련: 모델 피팅(fitting)

모수 모델(parametric model)의 경우 파라미터 추정(parameter estimation)과 비슷함

특성(x): 데이터 테이블이나 데이터 행렬의 열

동의어로는 예측 변수(predictor variable), 변수, 입력, 속성(attribute), 공변량(covariate)이 있음

1.3 기본 용어와 표기법 소개

- **머신 러닝 용어**
  - **타깃(y):** 동의어로는 결과(outcome), 출력(output), 반응 변수, 종속 변수(dependent
  - variable), (클래스) 레이블(label), 정답(ground truth)이 있음

  손실 함수(loss function): 종종 비용 함수(cost function)와 동의어로 사용

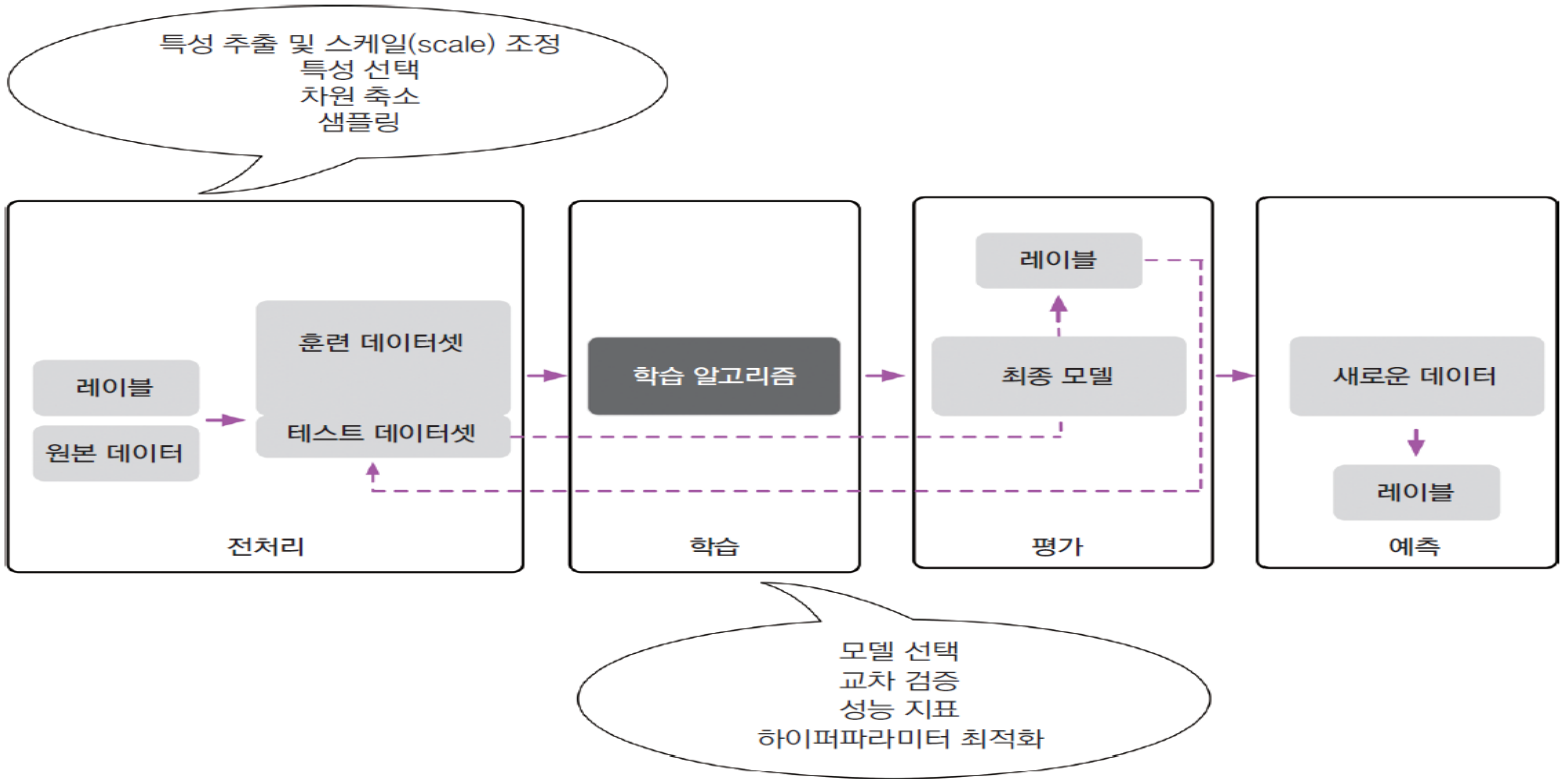
  일부 자료에서는 손실 함수를 하나의 데이터 포인트에 대해 측정한 손실로 사용하고, 비용

  함수는 전체 데이터셋에 대해 계산한 손실(평균 또는 합)로 사용

1.4 머신 러닝 시스템 구축 로드맵

1.4 머신 러닝 시스템 구축 로드맵

- **머신 러닝 시스템 구축 로드맵**
  - 이전 절에서 머신 러닝의 기초 개념과 세 종류의 학습 방법을 설명
  - 그림 1- 9는 예측 모델링에 머신 러닝을 사용하는 전형적인 작업 흐름을 보여 줌



1.4 머신 러닝 시스템 구축 로드맵

- **전처리: 데이터 형태 갖추기**
  - 머신 러닝 시스템을 구축할 수 있는 로드맵(roadmap)을 이야기해 보자
  - 주어진 원본 데이터의 형태와 모습이 학습 알고리즘이 최적의 성능을 내기에 적합한 경우는 매우 드뭄
  - 데이터 전처리는 모든 머신 러닝 애플리케이션에서 가장 중요한 단계 중 하나
  - 이전 절에서 예로 든 붓꽃 데이터셋을 생각해 보면 원본 데이터는 일련의 꽃 이미지들이고, 여기서 의미 있는 특성을 추출해야 함
  - 유용한 특성은 꽃의 색깔, 색조, 채도가 될 수 있음
  - 높이, 꽃의 길이와 너비도 가능함
  - 많은 머신 러닝 알고리즘에서 최적의 성능을 내려면 선택된 특성이 같은 스케일을 가져야 함
  - 특성을 [0, 1] 범위로 변환하거나 평균이 0이고 단위 분산을 가진 표준 정규 분포(standard normal distribution)로 변환하는 경우가 많음

1.4 머신 러닝 시스템 구축 로드맵

- **전처리: 데이터 형태 갖추기**
  - 일부 선택된 특성은 매우 상관관계가 높아 어느 정도 중복된 정보를 가질 수 있음
  - 이때는 차원 축소 기법을 사용하여 특성을 저차원 부분 공간으로 압축
  - 특성 공간의 차원을 축소하면 저장 공간이 덜 필요하고 학습 알고리즘을 더 빨리 실행할 수 있음
  - 어떤 경우에는 차원 축소가 모델의 예측 성능을 높이기도 함
  - 데이터셋에 관련 없는 특성(또는 잡음)이 매우 많을 경우, 즉 신호 대 잡음비(Signal-to-Noise Ratio, SNR)가 낮은 경우

1.4 머신 러닝 시스템 구축 로드맵

- **전처리: 데이터 형태 갖추기**
  - 머신 러닝 알고리즘이 훈련 데이터셋에서 잘 작동하고 새로운 데이터에서도 잘 일반화되는지 확인하려면 데이터셋을 랜덤하게 훈련 데이터셋과 테스트 데이터셋으로 나누어야 함
  - 훈련 데이터셋에서 머신 러닝 모델을 훈련하고 최적화
  - 테스트 데이터셋은 별도로 보관하고 최종 모델을 평가하는 맨 마지막에 사용

1.4 머신 러닝 시스템 구축 로드맵

- **예측 모델 훈련과 선택**
  - 여러 머신 러닝 알고리즘은 각기 다른 문제를 해결하기 위해 개발
  - 데이비드 월퍼트(David Wolpert)의 공짜 점심 없음(no free lunch) 이론의 중요한 핵심 포인트는 아무런 대가도 치르지 않고 학습할 수는 없다는 것
  - 예를 들어 분류 알고리즘은 저마다 태생적인 편향이 있음
  - 현실에서는 가장 좋은 모델을 훈련하고 선택하기 위해 최소한 몇 가지 알고리즘을 비교해야 함
  - 여러 모델을 비교하기 전에 먼저 성능을 측정할 지표를 결정해야 함
  - 분류에서 널리 사용되는 지표는 **정확도**(accuracy)

1.4 머신 러닝 시스템 구축 로드맵

- **예측 모델 훈련과 선택**
  - 모델 선택에 테스트 데이터셋을 사용하지 않고 최종 모델을 평가하려고 따로 보관한다면, 테스트 데이터셋과 실제 데이터에서 어떤 모델이 잘 동작할지 어떻게 알 수 있을까?
  - 이 질문에 나온 이슈를 해결하기 위해 다양한 교차 검증 기법을 사용
  - 교차 검증에서는 모델의 일반화 성능을 예측하기 위해 훈련 데이터를 훈련 데이터셋과 검증 데이터셋으로 더 나눔
  - 머신 러닝 라이브러리들에서 제공하는 알고리즘의 기본 하이퍼 파라미터가 현재 작업에 최적이라고 기대할 수 없음
  - 이어지는 장에서 모델 성능을 상세하게 조정하기 위해 하이퍼 파라미터 최적화 기법을 많이 사용할 것

1.4 머신 러닝 시스템 구축 로드맵

- **예측 모델 훈련과 선택**
  - 하이퍼 파라미터(hyperparameter)는 데이터에서 학습하는 파라미터가 아니라 모델 성능을 향상하기 위해 사용하는 다이얼로 생각할 수 있음
  - 이어지는 장에서 실제 예제를 보면 명확하게 이해 할 수 있을 것

1.4 머신 러닝 시스템 구축 로드맵

- **모델을 평가하고 본 적 없는 샘플로 예측**
  - 훈련 데이터셋에서 최적의 모델을 선택한 후에는 테스트 데이터셋을 사용하여 이전에 본 적이 없는 데이터에서 얼마나 성능을 내는지 예측하여 일반화 오차를 예상
  - 이 성능에 만족한다면 이 모델을 사용하여 미래의 새로운 데이터를 예측할 수 있음
  - 이전에 언급한 특성 스케일 조정과 차원 축소 같은 단계에서 사용한 파라미터는 훈련 데이터셋만 사용하여 얻은 것임을 주목해야 함
  - 나중에 동일한 파라미터를 테스트 데이터셋은 물론 새로운 모든 샘플을 변환하는 데 사용
  - 그렇지 않으면 테스트 데이터셋에서 측정한 성능은 과도하게 낙관적인 결과가 됨

1.5 머신 러닝을 위한 파이썬

1.5 머신 러닝을 위한 파이썬

- **머신 러닝을 위한 파이썬**
  - 파이썬은 데이터 과학 분야에서 가장 인기 있는 프로그래밍 언어
  - 개발자와 오픈 소스 공동체가 매우 활발히 활동하고 있기 때문에 과학 컴퓨팅과 머신 러닝을 위한 유용한 라이브러리가 많이 개발되어 있음
  - 계산량이 많은 작업에서는 파이썬 같은 인터프리터 언어의 성능이 저수준 프로그래밍 언어보다 낮음
  - 포트란(Fortran)과 C 언어로 만든 저수준 모듈 위에 구축된 넘파이(NumPy)와 사이파이(SciPy) 같은 라이브러리 덕택에 다차원 배열에서 벡터화된 연산을 빠르게 수행할 수 있음

1.5 머신 러닝을 위한 파이썬

- **머신 러닝을 위한 파이썬**
  - 이 책에서는 사이킷런(Scikit-learn) 라이브러리로 대부분 머신 러닝 프로그래밍 작업을 하겠음
  - 사이킷런은 현재 가장 인기 있고 사용하기 쉬운 오픈 소스 머신 러닝 라이브러리 중 하나
  - 책의 후반부에서 머신 러닝의 하위 분야인 딥러닝(deep learning)을 다룰 때 최신 버전의 텐서플로 라이브러리를 사용
  - 이 라이브러리는 그래픽 카드를 활용하여 심층 신경망 모델을 매우 효율적으로 훈련

1.5 머신 러닝을 위한 파이썬

- **파이썬과 PIP에서 패키지 설치**
  - 파이썬은 세 개의 주요 운영 체제인 마이크로소프트 윈도(Microsoft Windows), macOS, 리눅스(Linux)에서 사용할 수 있음
  - 파이썬 공식 사이트(https://www.python.org)에서 문서와 설치 파일을 내려받을 수 있음

1.5 머신 러닝을 위한 파이썬

- **파이썬과 PIP에서 패키지 설치**
  - 책은 파이썬 3.7.2 버전과 그 이상에 맞추어져 있음
  - 현재 파이썬 3의 최신 버전을 사용하는 것이 좋음
  - 일부 코드는 파이썬 2.7과 호환될 수 있지만 파이썬 2.7의 공식 지원이 2019년 말에 끝났기 때문에 상당수의 오픈 소스 라이브러리는 이미 파이썬 2.7에 대한 지원을 중단(https://python3statement.org/)
  - 파이썬 3.7 또는 그 이상을 사용하길 권장
  - 책에서 사용할 패키지는 pip 설치 프로그램으로 설치할 수 있음
  - 이 프로그램은 파이썬 3.3 버전부터 파이썬 표준 라이브러리에 포함
  - 자세한 pip 설명은 온라인 문서(https://docs.python.org/3/installing/index.html)를 참고

1.5 머신 러닝을 위한 파이썬

- **파이썬과 PIP에서 패키지 설치**
  - 파이썬을 설치하고 난 후 터미널(Terminal)에서 pip 명령으로 필요한 파이썬 패키지를 설치할 수 있음

```
> pip install 패키지 이름
```

  - 설치한 패키지를 업데이트할 때는 --upgrade 옵션을 사용

```
> pip install --upgrade 패키지 이름
```

1.5 머신 러닝을 위한 파이썬

- **아나콘다 파이썬 배포판과 패키지 관리자 사용**
  - 과학 컴퓨팅을 위해서는 컨티넘 애널리틱스(Continuum Analytics)의 아나콘다(Anaconda) 파이썬 배포판을 권장
  - 아나콘다는 상업적 목적을 포함하여 무료로 사용할 수 있고 기업이 사용하기 충분한 수준의 파이썬 배포판
  - 데이터 과학, 수학, 공학용 파이썬 필수 패키지들을 모두 포함하고 있으며 주요 운영 체제를 모두 지원
  - 아나콘다 설치 파일은 https://www.anaconda.com/download/에서 내려받을 수 있음
  - 간단한 아나콘다 안내는 온라인 문서(https://docs.anaconda.com/anaconda/user-guide/getting-started/)를 참고

1.5 머신 러닝을 위한 파이썬

- **아나콘다 파이썬 배포판과 패키지 관리자 사용**
  - 아나콘다를 설치한 후 다음 명령으로 필요한 파이썬 패키지를 설치할 수 있음

```
> conda install 패키지 이름
```

  - 설치한 패키지를 업데이트할 때는 다음 명령을 사용

```
> conda update 패키지 이름
```

1.5 머신 러닝을 위한 파이썬

- **과학 컴퓨팅, 데이터 과학, 머신 러닝을 위한 패키지**
  - 책 전반에 걸쳐 데이터를 저장하고 조작하는 데 넘파이 다차원 배열을 주로 사용
  - 이따금 판다스(Pandas)도 사용
  - 판다스는 넘파이 위에 구축된 라이브러리이고 테이블 형태의 데이터를 아주 쉽게 다룰 수 있는 고수준 도구를 제공
  - 종종 정량적인 데이터를 시각화하면 이해하는 데 매우 도움이 됨
  - 이를 위해 많은 옵션을 제공하는 맷플롯립(Matplotlib) 라이브러리를 사용

1.5 머신 러닝을 위한 파이썬

- **과학 컴퓨팅, 데이터 과학, 머신 러닝을 위한 패키지**
  - 책에서 사용하는 주요 파이썬 패키지 버전은 다음과 같음
  - 여러분 컴퓨터에 설치된 패키지와 버전이 동일하거나 더 높은지 확인
  - 예제 코드를 정상적으로 실행하려면 버전을 맞추는 것이 좋음

    - NumPy 1.19.5
    - SciPy 1.4.1
    - Scikit-learn 0.23.2
    - Matplotlib 3.2.2
    - Pandas 1.1.5
    - TensorFlow 2.4.1

1.6 요약

1.6 요약

- **요약**
  - 이 장에서는 매우 넓은 시각으로 머신 러닝을 바라보았음
  - 큰 그림에 익숙해졌고 이어지는 장에서 자세히 살펴볼 주요 개념을 둘러보았음
  - 지도 학습의 주요 하위 분야 두 개는 분류와 회귀
  - 분류 모델은 샘플을 알려진 클래스로 분류
  - 회귀 분석은 타깃 변수의 연속된 출력을 예측
  - 비지도 학습은 레이블되지 않은 데이터에서 구조를 찾는 유용한 기법
  - 또 전처리 단계에서 데이터 압축에도 유용하게 사용

1.6 요약

- **요약**
  - 머신 러닝을 적용하는 전형적인 로드맵도 간략히 살펴보았음
  - 이를 바탕으로 이어지는 장에서 좀 더 깊은 주제와 실전 예제를 다루겠음
  - 마지막으로 파이썬 환경과 필수 패키지를 설치하고 업데이트해서 머신 러닝을 작업할 준비를 마쳤음
  - 책 뒷부분에서 머신 러닝 이외에도 데이터셋을 전처리하는 여러 기법을 소개
  - 머신 러닝 알고리즘의 성능을 최대로 끌어내는 데 도움이 될 것
  - 분류 알고리즘을 폭넓게 다루겠지만 회귀 분석과 군집 알고리즘도 살펴볼 것

1.6 요약

- **요약**
  - 이제 강력한 기술들이 가득한 머신 러닝의 세계로 떠나는 흥미로운 여행을 앞두고 있음
  - 한번에 한 걸음씩 각 장을 거치면서 점진적으로 지식을 쌓도록 하겠음
  - 다음 장에서 머신 러닝 초기의 분류 알고리즘 중 하나를 구현하면서 시작해 보자
  - 3장을 이해하는 데 도움이 될 것
  - 여기서는 사이킷런 오픈 소스 머신 러닝 라이브러리로 많은 고급 머신 러닝 알고리즘을 다룸