머신 러닝

2023 Fall

강미선

I. 인공지능, 머신러닝, 딥러닝의 개념

1) 인공지능

- 인간이 가진 지적 능력을 컴퓨터를 통해 구현하는 기술
- 인공지능의 구분
  - » 강인공지능(Strong AI) : 인간의 능력을 초월한 성능을 가진 AI
  - » 약인공지능(Weak AI) : 특정 영역에서 도구로 사용하기 위해 설계된 AI

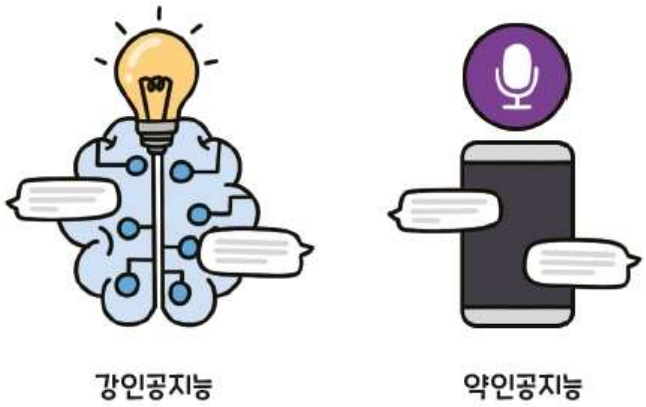


그림 8-1 강인공지능과 약인공지능

I. 인공지능, 머신러닝, 딥러닝의 개념

2) 머신러닝

- 컴퓨터를 인간처럼 학습하게 함으로써 인간의 도움 없이도 컴퓨터 스스로가 새로운 규칙을 발견할 수 있도록 하는 기술.
- 머신러닝은 기본적으로 알고리즘을 이용해 데이터를 분석하고, 분석을 통해 학습하며, 학습한 내용을 기반으로 판단이나 예측을 함
- 머신러닝이 스스로 학습하여 데이터를 처리하는 과정
  - ① 빅데이터를 입력
  - ② 데이터를 분석하여 모델을 만듦
  - ③ 모델을 이용하여 의사결정 및 예측 등을 수행

I. 인공지능, 머신러닝, 딥러닝의 개념

2) 머신러닝

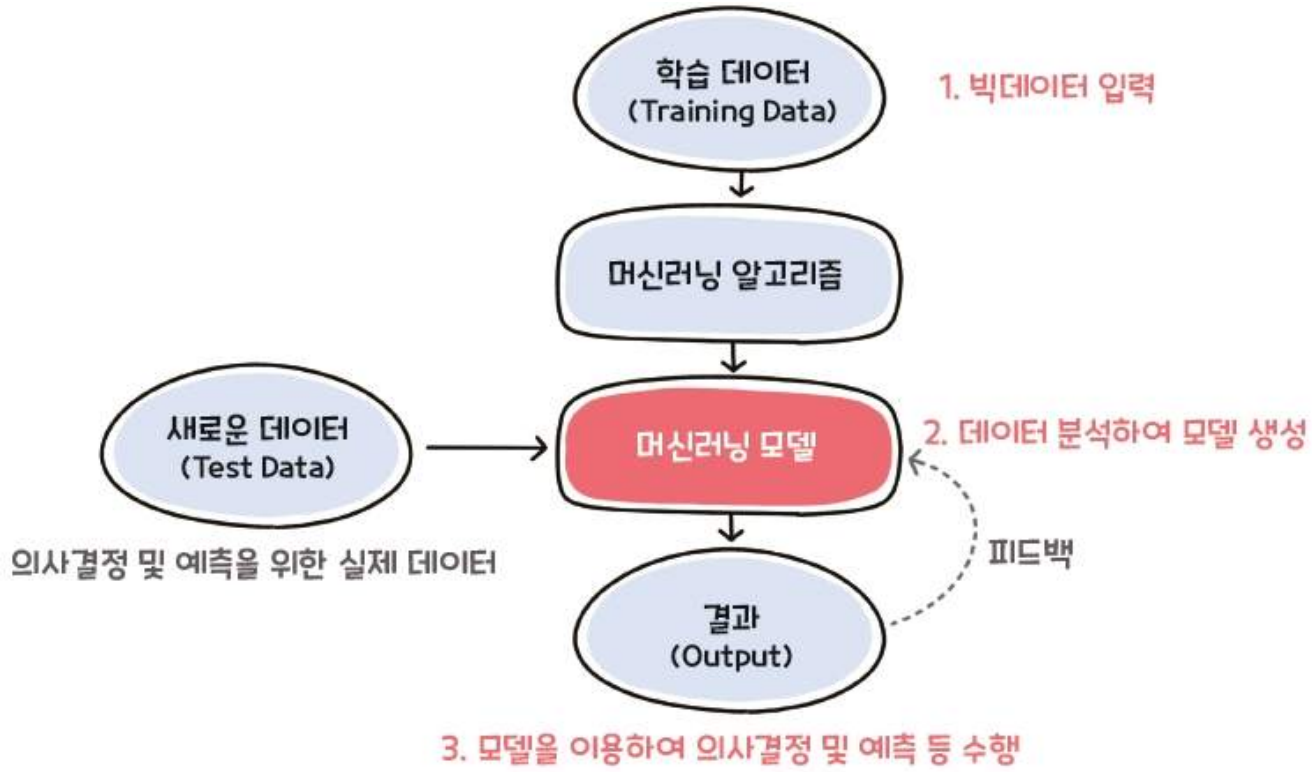


그림 8-2 머신러닝 학습 절차

I. 인공지능, 머신러닝, 딥러닝의 개념

3) 딥러닝

- **인공신경망(ANN, Artificial Neural Network)**
  - 여러 뉴런이 서로 연결되어 있는 구조의 네트워크

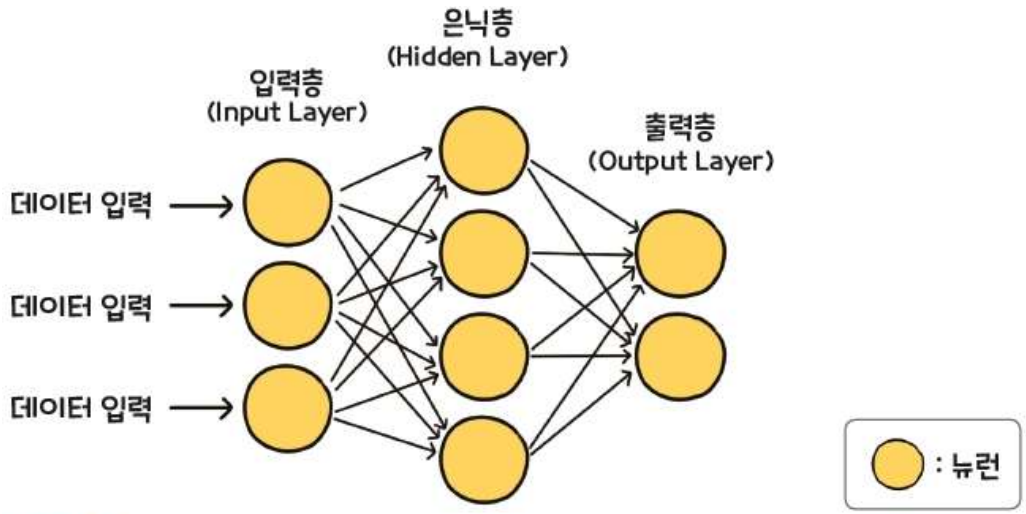


그림 8-3 인공신경망 구조

I. 인공지능, 머신러닝, 딥러닝의 개념

3) 딥러닝

- **딥러닝(Deep Learning)**
  - 여러 은닉층을 가진 인공신경망을 사용하여 머신러닝 학습을 수행하는 기술
  - 딥러닝의 '딥(Deep)'은 연속된 신경망 층(layer)을 깊게(deep) 쌓는다는 의미
  - 이 신경망이 깊어질수록 성능이 향상됨

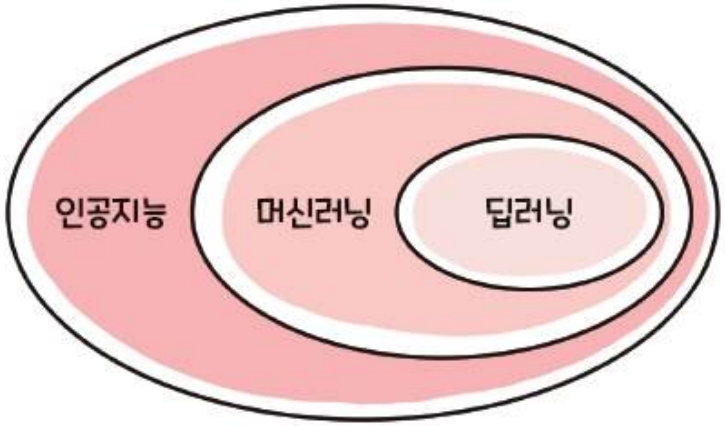


그림 8-4 인공지능, 머신러닝, 딥러닝의 관계

II. 머신러닝과 딥러닝의 차이점

1) 인간의 개입 유무

- 머신러닝은 사람이 학습 데이터에 레이블(정답)을 알려주거나 데이터의 특징을 추출하는 등 어느 정도 개입
- 딥러닝은 인간의 개입 없이 컴퓨터 스스로 학습함

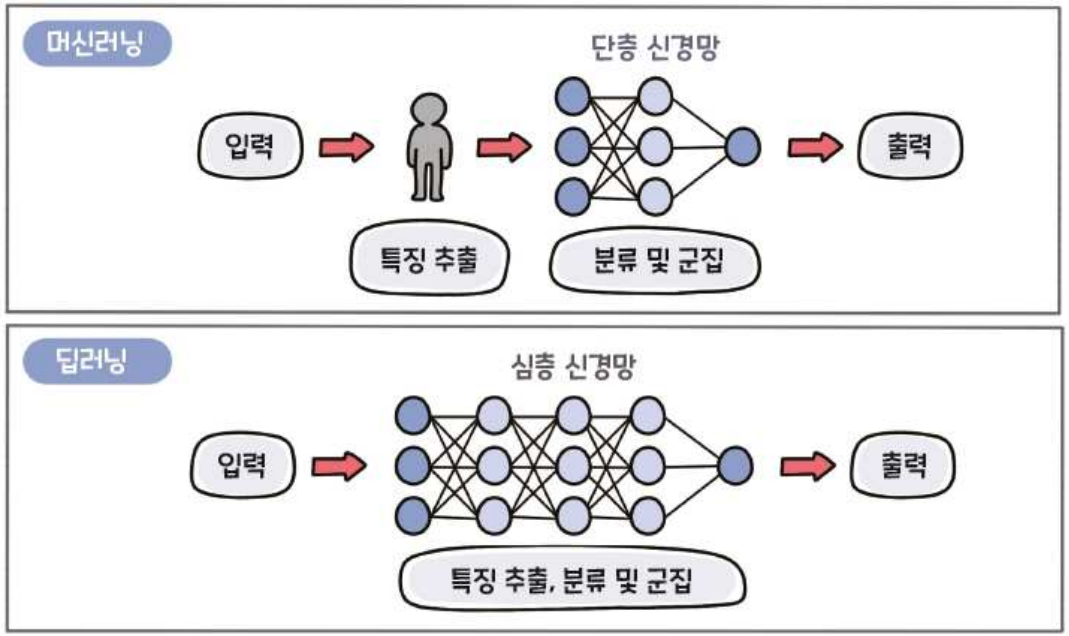
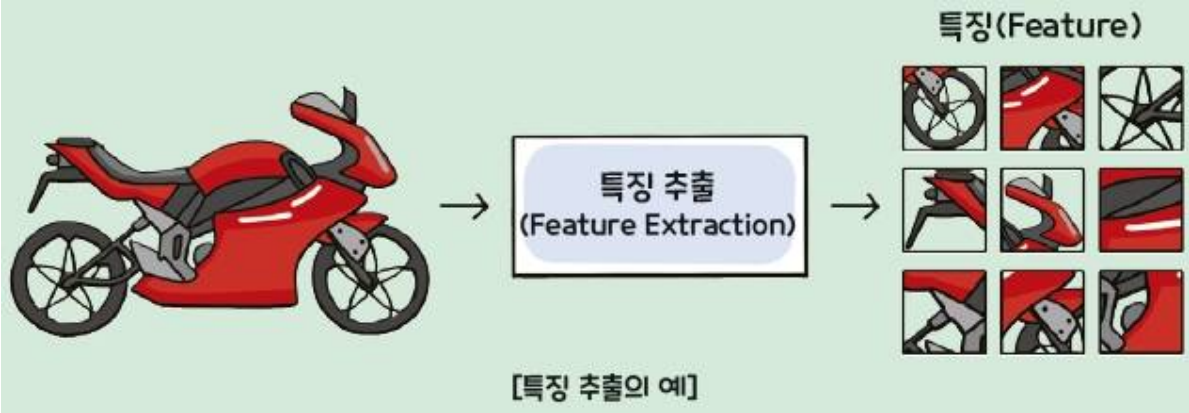


그림 8-5 인간의 개입 유무에 따른 머신러닝과 딥러닝의 차이

II. 머신러닝과 딥러닝의 차이점

하나 더 알기 **특징 추출**

- **특징 추출(Feature Extraction) :** 머신러닝에서 컴퓨터가 스스로 학습하려면 사람이 인지하는 데이터를 컴퓨터가 인지할 수 있는 데이터로 변환해야 하는데, 이 작업을 위해 데이터별로 어떤 특징을 가지는지 찾아내고 그것을 토대로 데이터를 벡터로 변환하는 것



[특징 추출의 예]

II. 머신러닝과 딥러닝의 차이점

2) 데이터 의존도(Data Dependencies)

- 딥러닝은 주어진 문제를 해결하기 위해 중요한 특징을 직접 추출
- 그래서 데이터의 양이 충분하지 않으면 정확한 특징을 추출할 수 없음
- 반면, 충분한 양의 데이터가 주어진다면 사람이 인지하지 못한 중요한 특징들까지 찾아낼 수 있을 정도로 좋은 성능 발휘

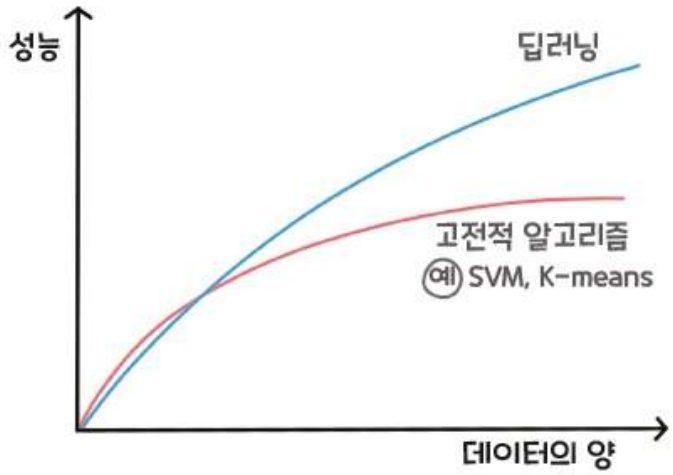


그림 8-6 데이터 양에 따른 성능 차이

II. 머신러닝과 딥러닝의 차이점

3) 심층신경망의 사용 여부

- 딥러닝은 심층신경망을 이용하여 입력 데이터에서 특징을 추출하고 스스로 결과(예측 혹은 분류)를 도출
- 심층신경망을 사용하는 것은 딥러닝만의 뚜렷한 특징

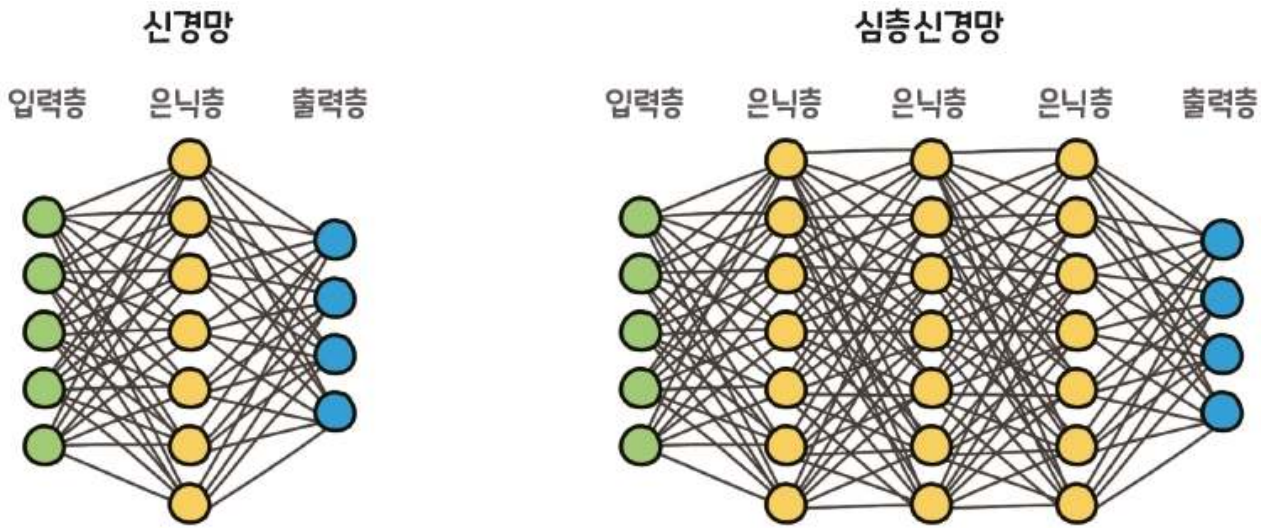


그림 8-7 신경망과 심층신경망의 차이

II. 머신러닝과 딥러닝의 차이점

3) 심층신경망의 사용 여부

표 8-1 머신러닝과 딥러닝의 차이점

구분	머신러닝	딥러닝
필요한 데이터의 양	적은 양의 데이터도 가능	빅데이터
정확도	낮음	높음
훈련 시간	짧은 시간 안에 가능	오래 걸림
하드웨어	CPU만으로도 가능	GPU
하이퍼파라미터 튜닝	제한적	다양한 방법으로 튜닝 가능

02. 머신러닝을 사용하는 이유

I. 기존 프로그래밍의 한계

- [그림 8-8]의 왼쪽 그림은 마케팅에 활용할 프로그램

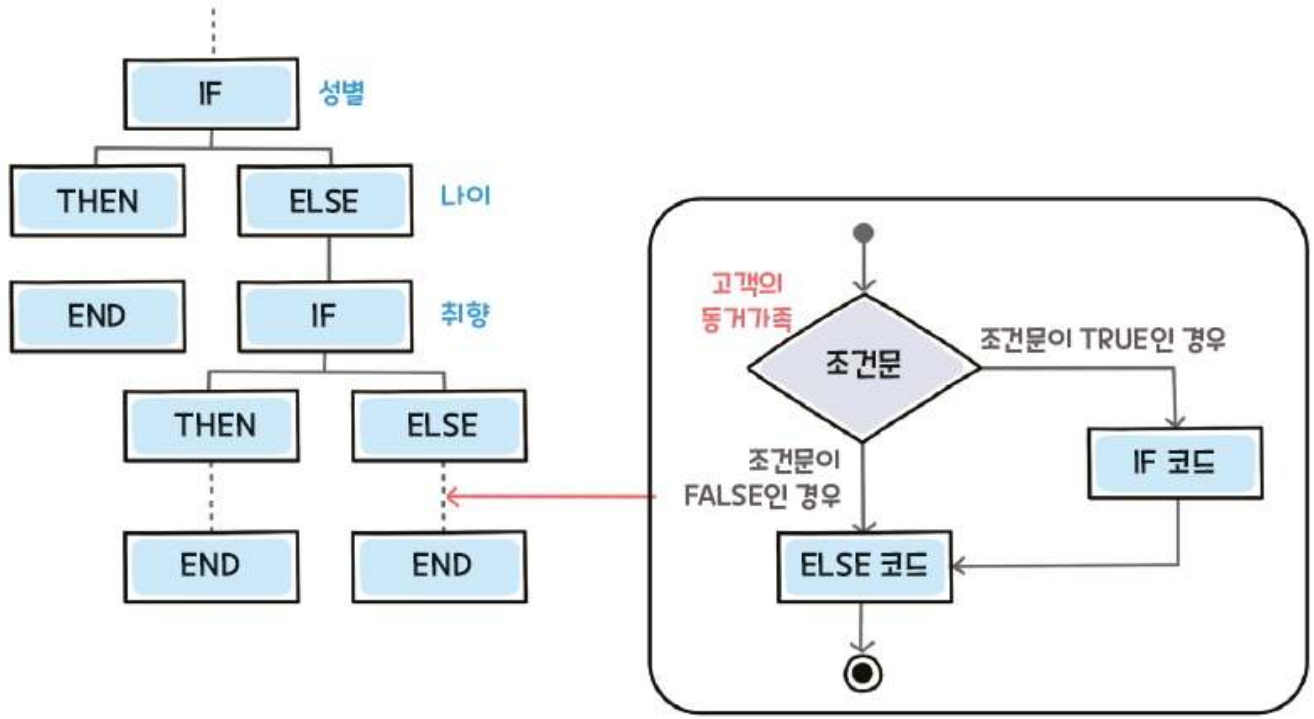


그림 8-8 기존 프로그램에 데이터 추가하기

02. 머신러닝을 사용하는 이유

I. 기존 프로그래밍의 한계

- 고객의 성별 · 나이 · 취향 등이 반영된 프로그램에 오른쪽 그림의 '고객의 동거가족' 변수를 추가하고 싶다면, 만들어져 있는 기존 프로그램을 변경해야 함
- 뿐만 아니라 기존 변수들과의 관련성까지 고려해 프로그램 전체를 수정해야 함
- '고객의 동거가족'에 대한 데이터를 데이터베이스에 저장하기 위한 수정 필요
- 즉, 변수가 하나 더 추가되었을 뿐인데도 프로그램에서 수정해야 할 부분들 이 상당히 많다는 것을 알 수 있음

II. 머신러닝의 유용성

- 하지만 빠른 의사결정이 필요한 시기에 [그림 8-8]의 방식이 적절하지 않음
- 그래서 이를 해결하기 위해 머신러닝 방식을 사용하는 것
- 대용량의 데이터와 많은 변수가 관련되어 있고 기존에 사용했던 규칙의 프로그램으로는 복잡한 작업이나 문제를 해결할 수 없을 때 머신러닝은 아주 유용한 해결책임

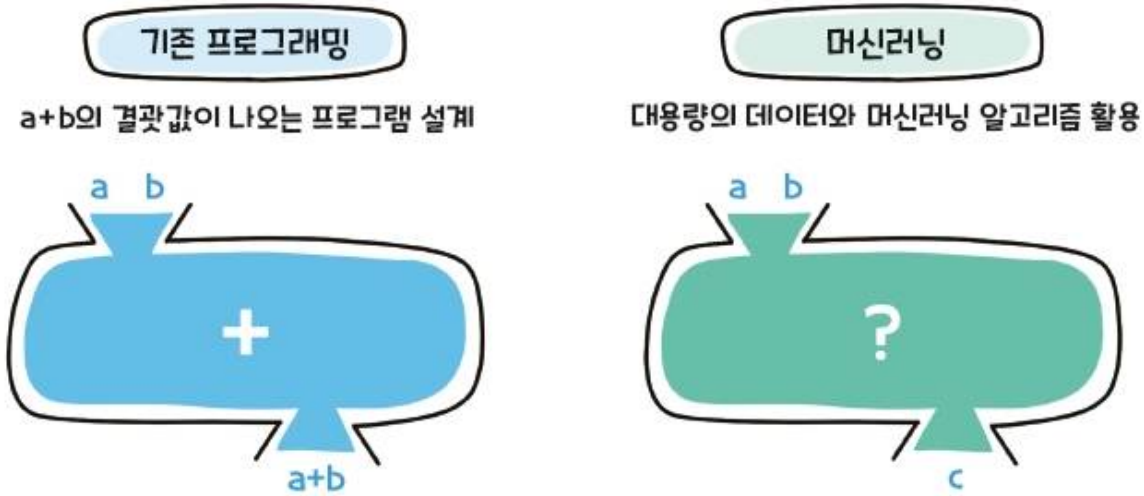


그림 8-9 기존 프로그래밍과 머신러닝의 차이

II. 머신러닝의 유용성

- 머신러닝은 다음과 같은 상황에서 사용하면 매우 유용함
  - 얼굴 인식이나 음성 인식과 같이 규칙 기반 프로그램으로 답을 낼 수 없는 복잡한 상황
  - 거래 기록에서 사기를 감지하는 경우와 같이 규칙이 지속적으로 바뀌는 상황
  - 주식 거래, 에너지 수요 예측, 쇼핑 추세 예측의 경우처럼 데이터 특징이 계속 바뀌는 상황

03. 머신러닝의 분류

- **머신러닝의 분류**
  - **지도학습 : 예측이나 분류를 위해 사용**
  - **비지도학습 : 군집을 위해 사용**
  - **강화학습 : 환경에서 취하는 행동에 대한 보상을 이용하여 학습을 진행**

그림 8-10 머신러닝 분류

- **지도학습(Supervised Learning)**
  - 문제와 답을 함께 학습함으로써 미지의 문제에 대한 올바른 답을 예측하는 학습
  - 지도학습에서 사용하는 모델로는 크게 예측과 분류가 있음

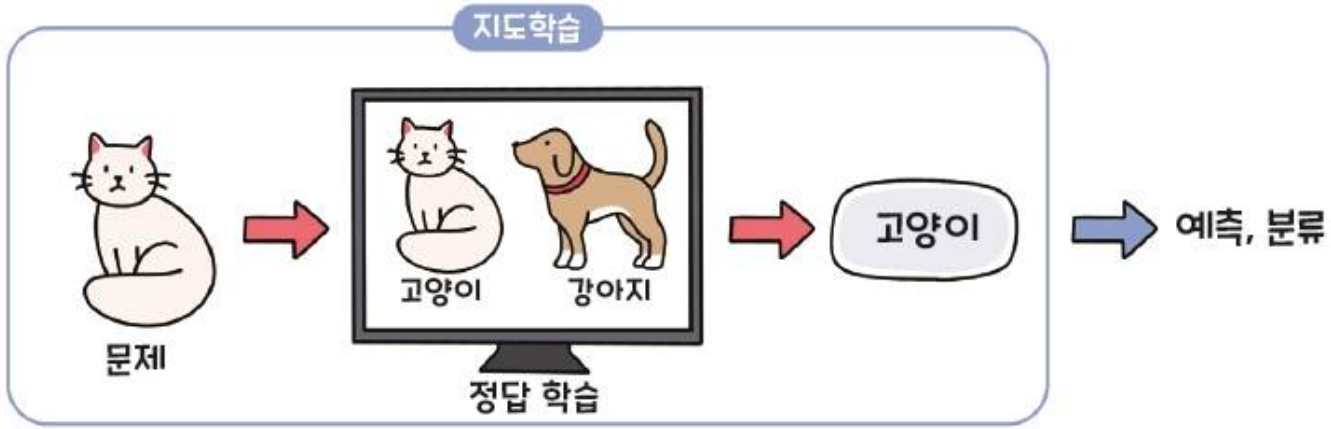


그림 8-11 지도학습

- **비지도학습(Unsupervised Learning)**
  - 지도학습과 다르게 조력자의 도움 없이 컴퓨터 스스로 학습하는 형태
  - 컴퓨터가 훈련 데이터를 이용하여 데이터들 간의 규칙성을 찾음

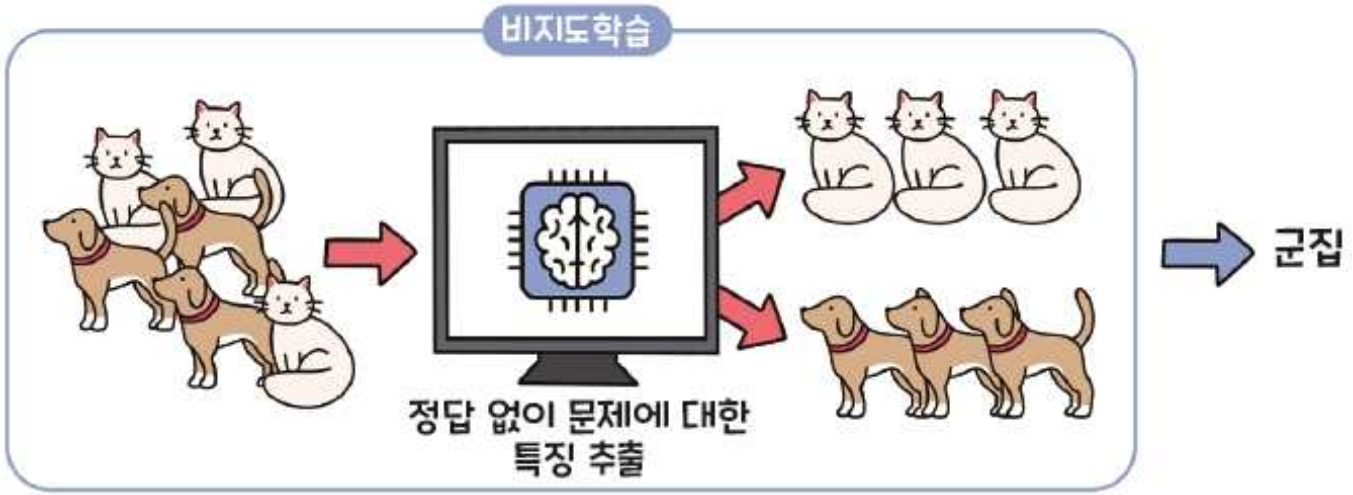


그림 8-12 비지도 학습

- **비지도학습(Unsupervised Learning)**
  - x(입력 데이터)와 y(지도학습에서 레이블)의 관계를 파악했던 지도학습과는 달리, 비지도학습은 x 간의 관계를 스스로 파악함
  - 즉, 지도학습과 다른 점은 y(레이블)의 차이
  - 비지도학습에서 사용하는 모델로는 군집(Clustering)이 있음

표 8-2 지도학습과 비지도학습 시 필요한 데이터

구분	지도학습	비지도학습
필요한 데이터 종류	x(학습 데이터), y(레이블)	x(학습 데이터)

- **강화 학습(Reinforcement Learning)**
  - 자신이 한 행동에 대해 보상(Reward)을 받으며 학습하는 것
  - 컴퓨터가 주어진 상태에 대해 최적의 행동을 선택하도록 학습하는 방법

그림 8-13 강화학습

- 강화학습을 이해하기 위해 알아야 할 개념들
  - 에이전트(Agent) : 주어진 문제 상황에서 행동하는 주체
  - 상태(State) : 현재 시점에서의 상황
  - 행동(Action) : 플레이어가 취할 수 있는 선택지
  - 보상(Reward) : 플레이어가 어떤 행동을 했을 때 따라오는 이득
  - 환경(Environment) : 문제 그 자체를 의미
  - 관찰(Observation) : 에이전트가 수집한(보고 듣는) 환경에 대한 정보

- **강화 학습(Reinforcement Learning)**
  - 주어진 환경에서 에이전트가 선택한 행동에 따라 그 행동이 옳은 선택이면 상을 받고, 잘못된 선택이면 벌을 받음
  - 강화학습은 에이전트가 상태를 계속 주시하면서 보상이 높은 쪽으로 학습(행동)하게 됨

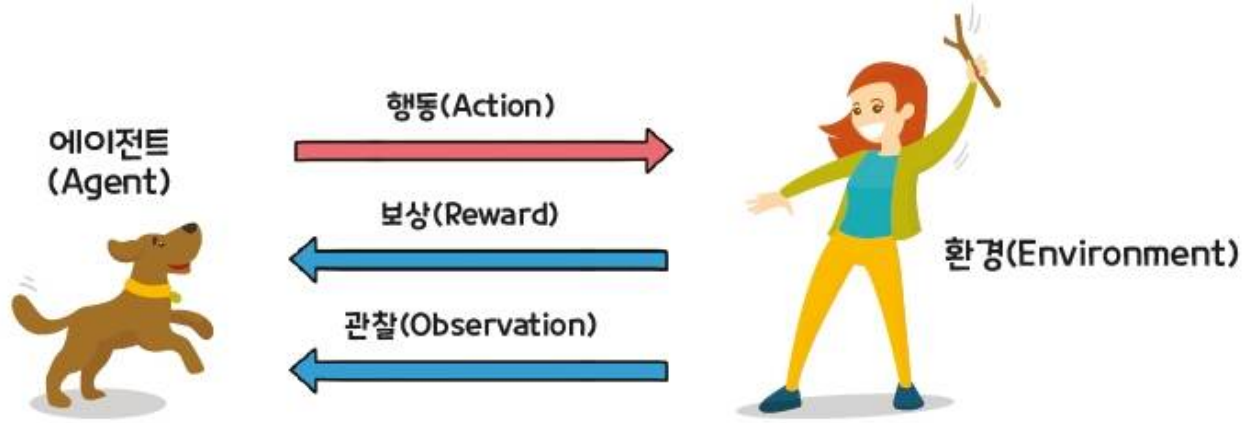


그림 8-14 강화학습 과정

04. 머신러닝 알고리즘의 유형

- 머신러닝 알고리즘의 유형
  - 지도학습 : 분류와 예측
  - 비지도학습 : 군집
  - 강화학습 : 큐러닝과 딥큐러닝

04. 머신러닝 알고리즘의 유형

- **분류(Classification)**
  - 레이블이 포함된 데이터를 학습하고 유사한 성질을 갖는 데이터끼리 분류한 후, 새로 입력된 데이터가 어느 그룹에 속하는지를 찾아내는 기법

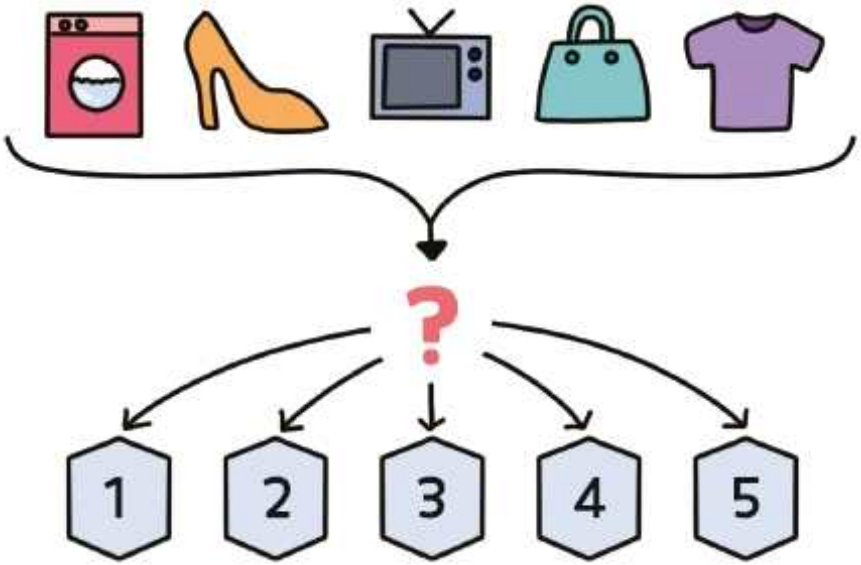


그림 8-15 분류

04. 머신러닝 알고리즘의 유형

- **분류의 종류**
  - **이진 분류(Binary Classification) : 데이터를 2개의 그룹으로 분류**
  - **다중 분류(Multiclass Classification) : 데이터를 3개의 그룹 이상으로 분류**

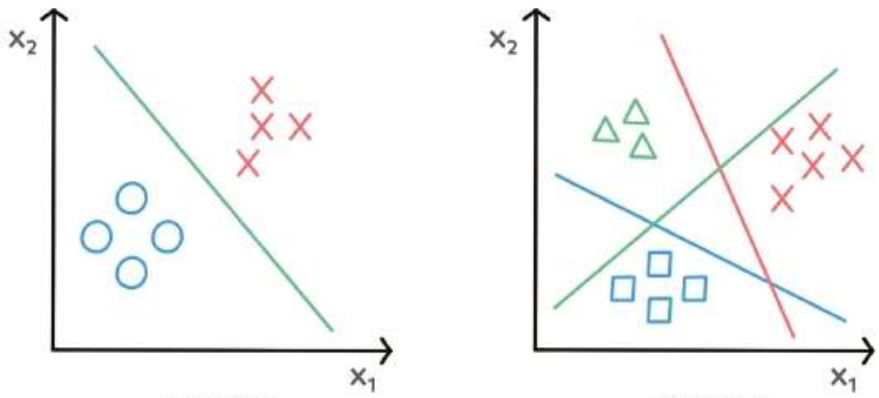


그림 8-16 이진 분류와 다중 분류

04. 머신러닝 알고리즘의 유형

- **분류에 해당하는 알고리즘**
  - **K-최근접 이웃(KNN)**
  - **서포트 벡터 머신(SVM)**
  - **의사결정나무 (Decision Tree)**
  - **로지스틱 회귀(Logistic Regression)**

04. 머신러닝 알고리즘의 유형

- **분류**

  1) **K-최근접 이웃(KNN, K-Nearest Neighbors)**

     – 새로운 데이터가 들어왔을 때 기존 데이터의 그룹 중 어떤 그룹에 속하는지 분류하는 알고리즘

     – (예) K=1일 때 신규 데이터가 입력되면 빨간 원으로 분류, K=3일 때 신규 데이터는 파란 삼각형으로 분류, K=9일 때도 파란 삼각형으로 분류됨

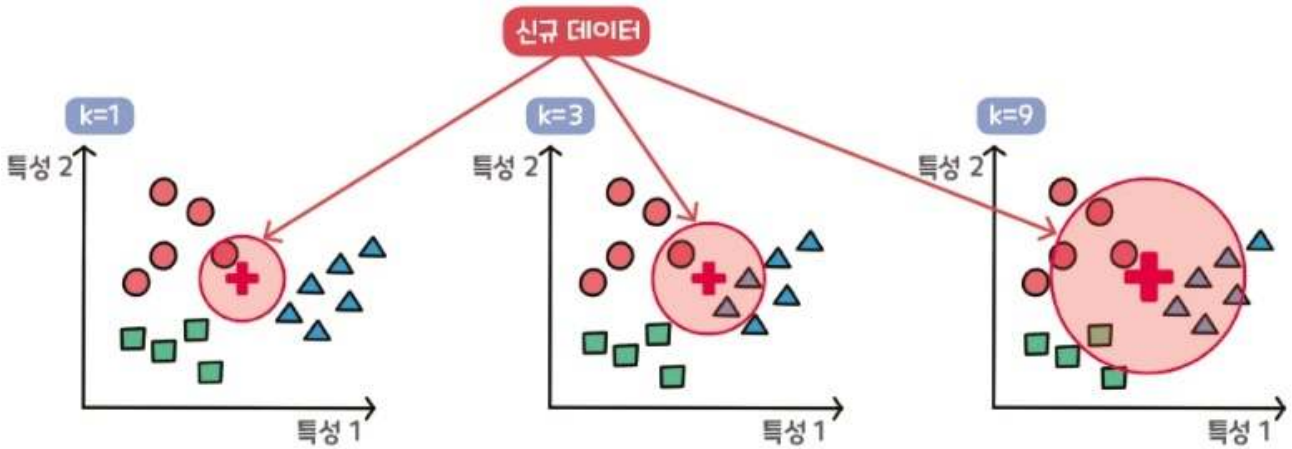


그림 8-17 K변화에 따른 분류

04. 머신러닝 알고리즘의 유형

- **분류**

  1) **K-최근접 이웃(KNN, K-Nearest Neighbors)**

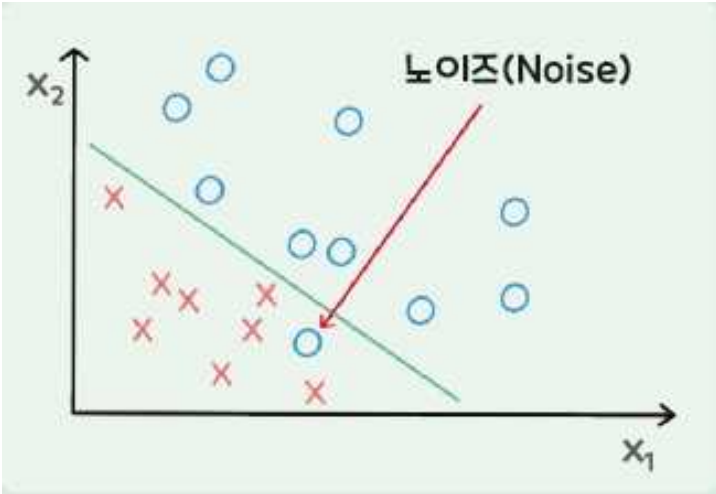
     - KNN은 학습 데이터 내에 존재하는 노이즈의 영향을 크게 받지 않으며, 학습 데이터 수가 많을 때 꽤 효과적인 알고리즘
     - 하지만 어떤 하이퍼파라미터가 분석에 적합한지는 불분명해, 데이터 각각의 특성에 맞게 연구자가 임의로 선정해야 한다는 단점이 있음

I. 분류

① K-최근접 이웃

하나 더 알기 **노이즈(Noise)**

- **노이즈(Noise) :** 데이터에 무작위의 오류(Random Error) 또는 분산(Variance)이 존재하는 것임
- 예시 그래프를 보면 엑스(X)가 분류된 곳에 동그라미(O) 데이터가 하나 있는데, 이것이 노이즈 데이터임



04. 머신러닝 알고리즘의 유형

- **분류**

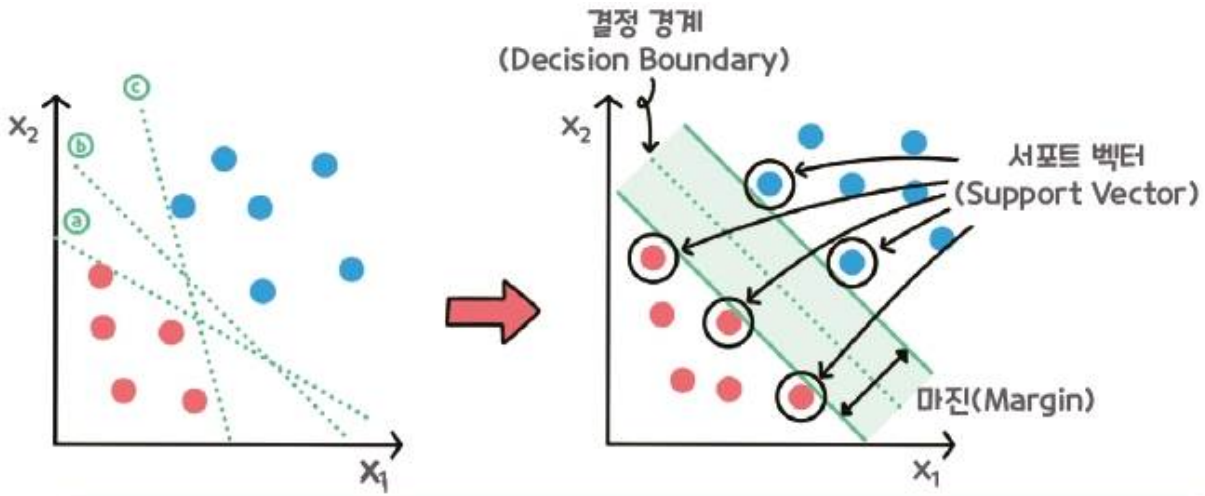
  2) 서포트 벡터 머신(SVM, Support Vector Machine)

  - 주어진 데이터가 어느 그룹에 속하는지 분류하는 모델
  - 두 분류 사이의 여백을 의미하는 마진을 최대화하는 방향으로 데이터를 분류
  - SVM은 마진을 극대화하는 선을 찾아 분류하므로 마진이 크면 클수록 새로운 데이터가 들어오더라도 잘 분류할 가능성이 높아짐
  - SVM은 사용 방법이 쉽고 예측 정확도가 높다는 장점
  - 하지만 모델 구축에 시간이 오래 걸리고 결과에 대한 설명력이 떨어지는 단점

04. 머신러닝 알고리즘의 유형

- **분류**

  2) 서포트 벡터 머신(SVM, Support Vector Machine)



· 결정 경계(Decision Boundary) : 분류를 위한 기준선
· 서포트 벡터(Support Vector) : 결정 경계와 가장 가까운 위치에 있는 데이터
· 마진(Margin) : 결정 경계와 서포트 벡터 사이의 거리

그림 8-18 SVM 분류

- **분류**

3) 의사결정나무(Decision Tree)

- 의사결정 규칙을 나무 형태로 분류하는 분석 방법
- [그림 8-19]와 같이 상위 노드에서 시작하여 분류 기준값에 따라 하위 노드로 확장하는 방식이 '나무'를 닮았다고 하여 '의사결정나무'라고 불림

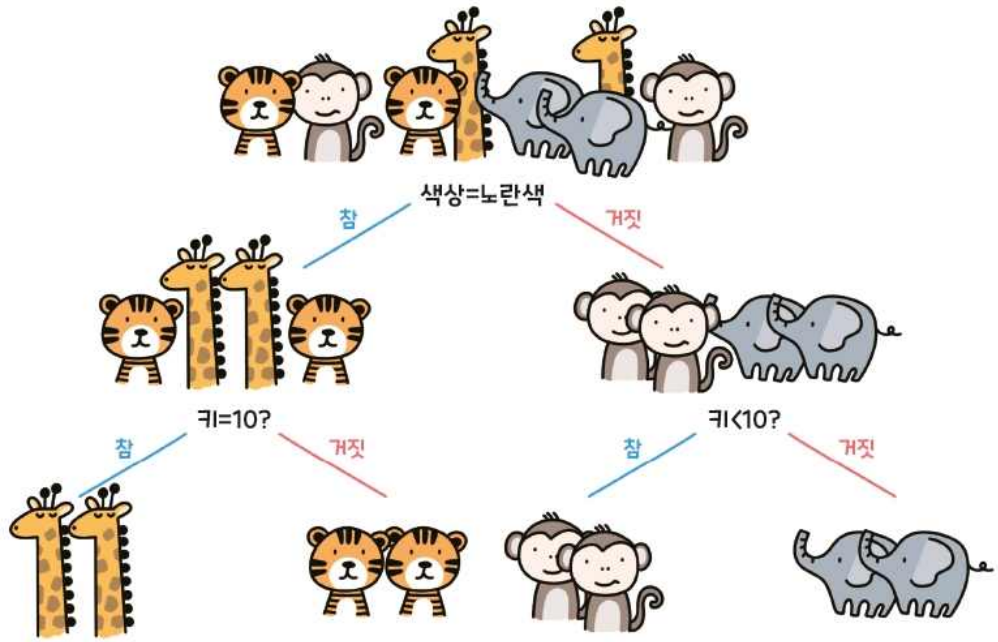


그림 8-19 의사결정나무 구조

04. 머신러닝 알고리즘의 유형

- **분류**

  3) **의사결정나무(Decision Tree)**

  - 의사결정나무는 분석 과정이 직관적이고 이해하기 쉬움
  - 인공신경망의 경우 분석 결과에 대한 설명이 어려운 블랙박스 모델인 반면, 의사결정나무는 분석 과정을 눈으로도 관측할 수 있음
  - 그래서 결과에 대한 명확한 설명이 필요할 때 많이 사용함

04. 머신러닝 알고리즘의 유형

- 분류

4) 로지스틱 회귀

- 로지스틱 회귀(Logistic Regression)
  - 데이터가 어떤 범주에 속할 확률을 0~1 사이의 값으로 정해놓고, 그 확률에 따라 가능성이 더 높은 범주에 속하는 것으로 분류해 주는 지도학습 알고리즘
- 회귀 (Regression)
  - 연속형 변수들에 대해 변수 간 관계를 추정하는 분석 방법이며, 선형 회귀는 독립변수와 종속변수가 직선의 형태를 취하는 관계

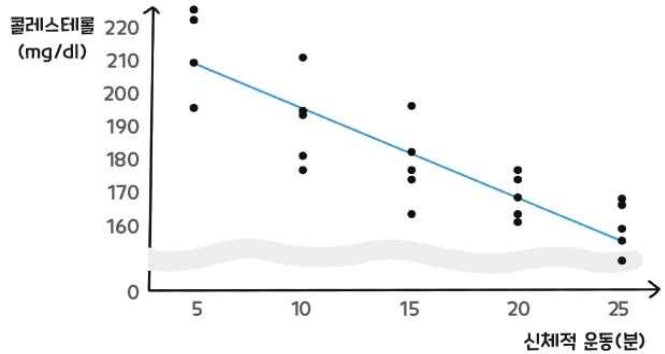


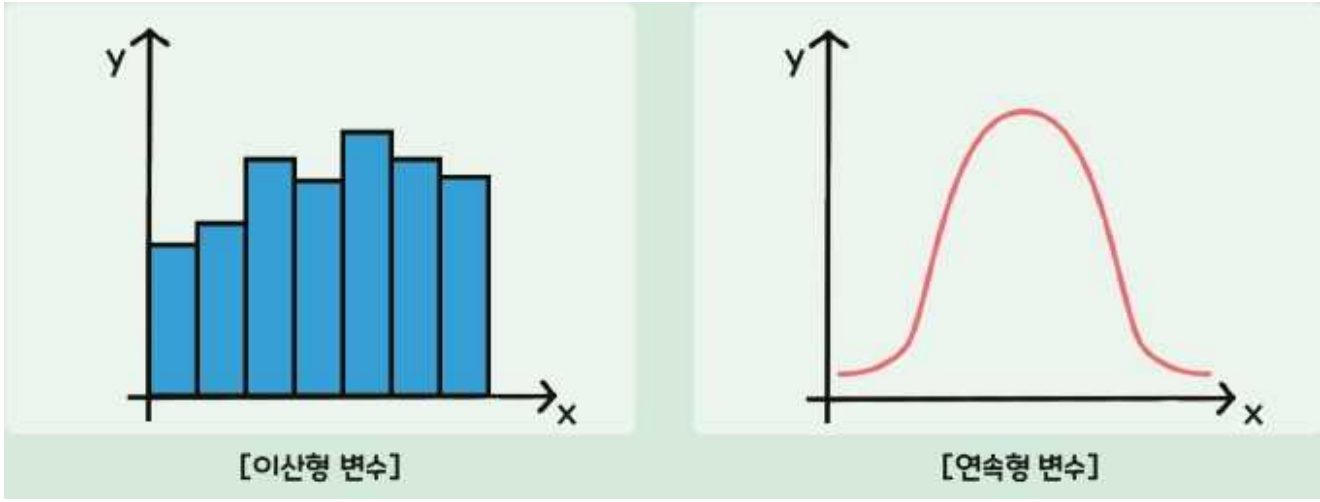
그림 8-20 직선의 형태의 예 : 콜레스테롤과 신체적 운동에 대한 관계

04. 머신러닝 알고리즘의 유형

- **분류**

하나 더 알기 **이분 변수, 이산형 변수, 연속형 변수**

1. **이분 변수** : 두 개의 값만을 가질 수 있는 변수(예 : 남/여, 있다/없다)
2. **이산형 변수** : 값들이 끊어지는 형태를 취하는 변수(예 : 주차된 자동차 수)
3. **연속형 변수** : 값들이 연속된 형태를 취하는 변수(예 : 키, 몸무게)



04. 머신러닝 알고리즘의 유형

- **분류**

4) 로지스틱 회귀

- 로지스틱 회귀는 선형 회귀와는 다르게 종속변수가 범주형 데이터
- 즉, 입력 데이터가 주어졌을 때 해당 데이터의 결과가 0과 1 사이의 값을 가짐
- 결과값이 정해진 범주 내에서 나오므로 확률적인 의미에서 사건 발생 가능성을 예측하는 데 사용할 수 있음
- 선형 회귀는 종속변수로 올 수 있는 값에 대한 제약이 없는 반면, 로지스틱 회귀의 종속변수는 값이 제한적이라는 것에 주목해야 함

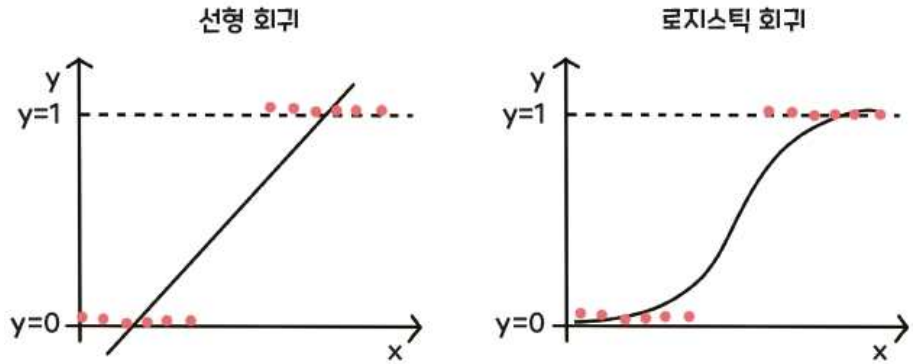


그림 8-21 선형 회귀와 로지스틱 회귀

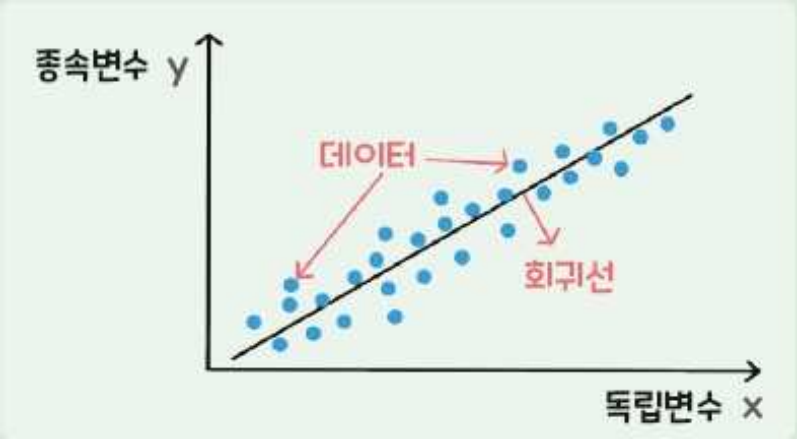
04. 머신러닝 알고리즘의 유형

- **분류**

④ 로지스틱 회귀

하나 더 알기 **회귀**

- 회귀는 연속형 변수를 예측하는 데 사용되는데, 즉 연속적인 숫자나 실수를 예측하는 것(예 : 주식 및 부동산 가격 예측 등)
- 회귀는 종속변수와 독립변수 간의 관계를 살펴볼 때 유용하게 사용



04. 머신러닝 알고리즘의 유형

- 군집화
  - 군집(Cluster, 클러스터)
    - 비슷한 특징을 가진 데이터들의 집단
  - 군집화(Clustering, 클러스터링)
    - 데이터가 주어졌을 때 그 데이터들을 유사한 정도에 따라 군집으로 분류하는 것
    - [그림 8-22]의 왼쪽 그래프를 보면 다양한 데이터들이 서로 섞여 있지만, 군집화 과정을 진행하면 오른쪽 그래프와 같이 비슷한 데이터끼리 군집으로 묶임

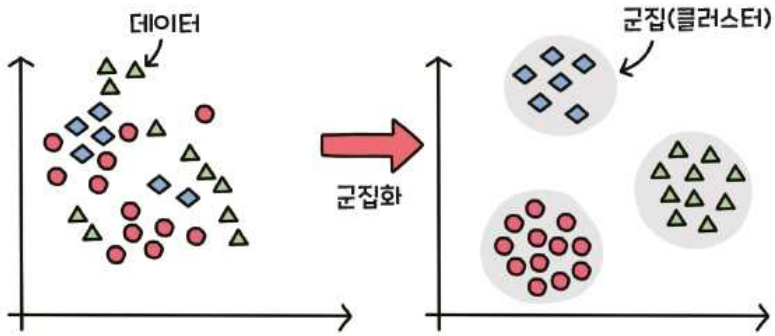


그림 8-22 군집화

04. 머신러닝 알고리즘의 유형

- **군집화**

  1) **k-평균 군집화(K-Means Clustering)**

     - **'K'는 주어진 데이터로부터 묶여질 그룹(군집의 수)**
     - **'Means'는 각 군집의 중심과 데이터들의 평균 거리를 의미**
     - **클러스터의 중심을 중심점(Centroids)이라고 함**

- **군집화**

1) k-평균 군집화(K-Means Clustering)

- (a) : 일반적인 데이터 분포입니다.
- (b) : 데이터셋에서 K개의 중심점을 임의로 지정하는데, 여기에서는 K=2의 값으로 중심점 2개를 설정했습니다.
- (c) : 데이터들을 가장 가까운 중심점에 할당합니다.
- (d) : (c)에서 할당된 결과를 바탕으로 중심점을 새롭게 지정합니다.
- (e) : 중심점이 더 이상 변하지 않을 때까지 (c)~(d) 과정을 반복합니다.
- (f) : 최종적인 군집이 형성됩니다.

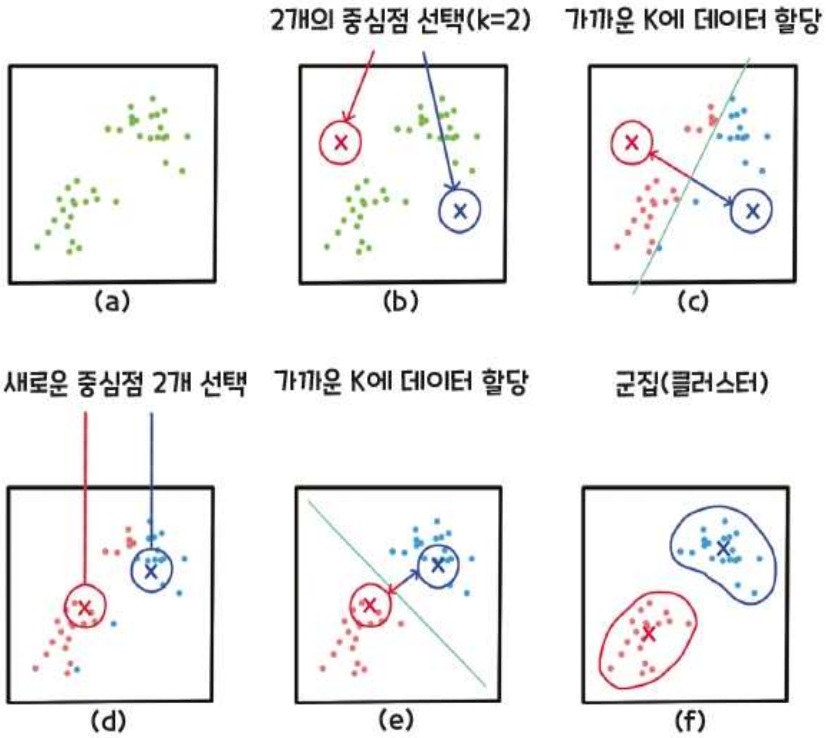


그림 8-23 K-평균 군집화 과정

- 군집화
  - 2) 밀도기반 클러스터링(DBSCAN)
    - 밀도를 기반으로 군집화하는 매우 유용한 군집 알고리즘
    - 밀도기반 클러스터링은 데이터들의 분포
    - 밀도기반 클러스터링을 이해하기 위한 관련 용어
      - » ε(Epsilon, 거리) : 하나의 점으로부터의 반경
      - » minPth(Minimum Points, 최소점) : 군집을 이루기 위한 최소한의 데이터 수

- 군집화

  2) 밀도기반 클러스터링(DBSCAN)

  - 밀도기반 클러스터링의 진행 과정(ε=5cm, minPth=4라고 가정)

① 1단계 : 한 점을 중심으로 반경 5cm거리에 4개의 데이터가 있는지(minPth=4를 만족하는지) 확인

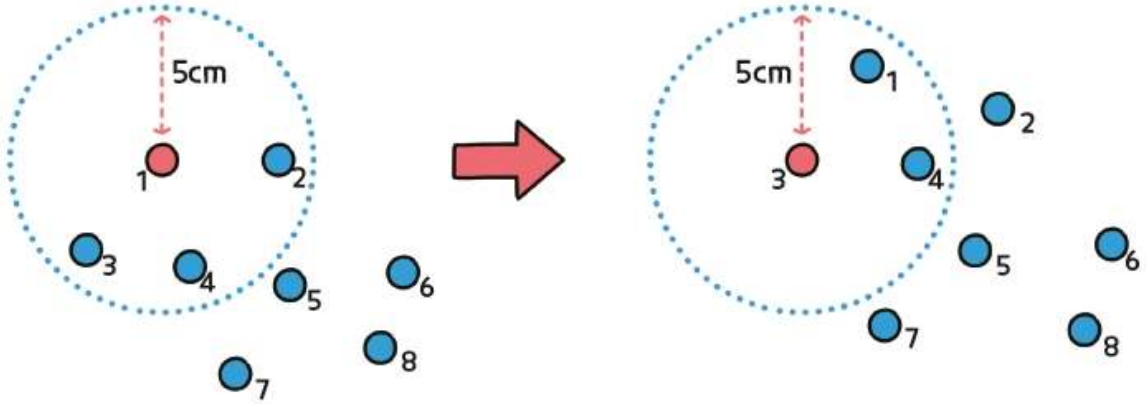


그림 8-24 최소점 확인 및 중심점 이동

- 군집화

  2) 밀도기반 클러스터링(DBSCAN)

  ② 2단계 : 이동한 중심점 3을 기준으로 1단계를 반복하는데, 3을 기준으로 반경 5cm 이내에 데이터가 4개 초과 있는지 확인하면, 역시 데이터의 수가 4보다 작으므로 이번에는 중심점을 4로 지정함

  ③ 3단계 : 4를 중심점으로 했을 때 데이터의 수가 4를 초과하므로 군집이 생성

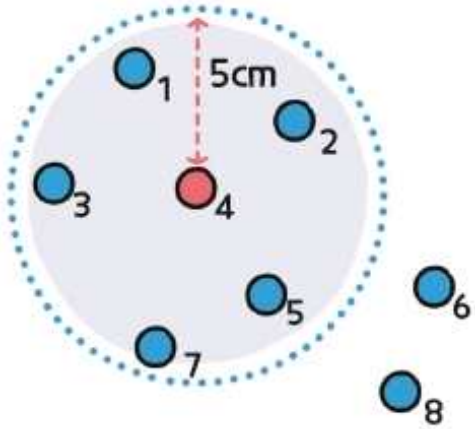


그림 8-25 군집 생성

- 군집화

  2) 밀도기반 클러스터링(DBSCAN)

  – K-평균 군집화와 달리 밀도기반 클러스터링은 클러스터 수를 지정할 필요가 없음

  – 더 중요한 것은 밀도기반 클러스터링은 K-평균 군집화가 찾을 수 없는 임의의 모양들을 가질 수 있다는 점임

  – 예를 들어, 밀도기반 클러스터링은 [그림 8-26]의 첫 번째 그림과 같이 다른 군집으로 둘러 싸인 상태에서 또 다른 군집을 가질 수 있음

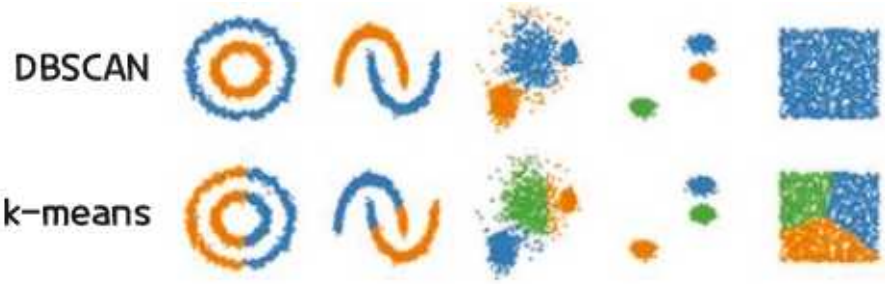


그림 8-26 밀도기반 클러스터링과 K-평균 군집화 비교

04. 머신러닝 알고리즘의 유형

- 강화학습 기술

1) 모델이 없는 알고리즘

- 모델기반 알고리즘은 현재의 상태에서 어떤 행동을 했을 때 다음의 상태가 될 확률을 의미함
- 예를 들어, [그림 8-27]과 같이 격자 공간에서 로봇이 상하좌우로 이동할 때 로봇의 다음 상태에 대해 직관적으로 파악할 수 있음
- 모델기반 알고리즘은 이처럼 행동에 따른 상태의 변화를 예측할 수 있어 최적의 솔루션을 얻을 수 있음

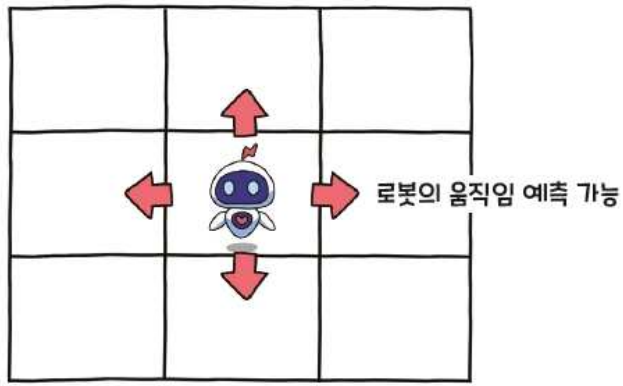


그림 8-27 모델기반 알고리즘의 예

- 강화학습 기술
  - 1) 모델이 없는 알고리즘
    - 모델이 없는 알고리즘
      - 에이전트가 행동을 통해 받게 되는 보상을 최대로 하는 정책(Policy)을 찾는 것
      - (예) 지뢰찾기 게임

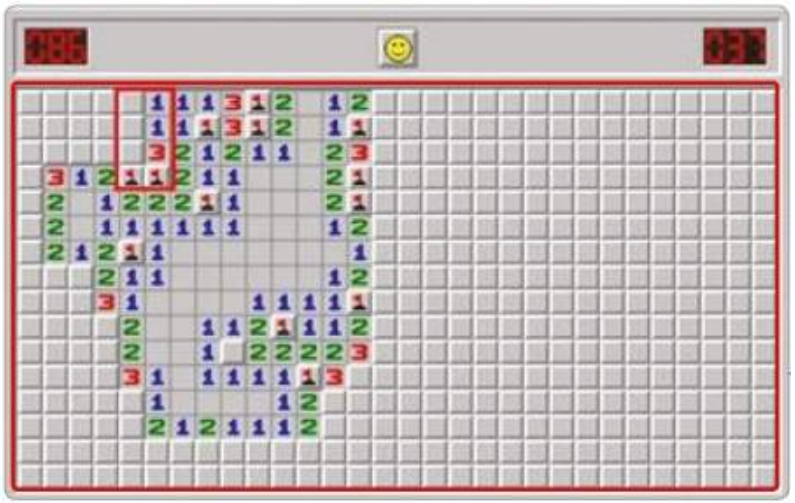


그림 8-28 모델이 없는 알고리즘의 예 : 지뢰찾기 게임

04. 머신러닝 알고리즘의 유형

- 강화학습 기술
  - 2) 큐러닝
    - 큐러닝(Q-Learning)
      - 특정 상태에서 어떤 결정을 내려야 미래 보상이 극대화될 것인지에 대한 정책을 지속적으로 업데이트하는 것
      - 모델 없이 학습하는 대표적인 강화학습 알고리즘
      - [그림 8-29]와 같이 S에서 시작하여 E로 끝나는 미로게임이 있다고 가정

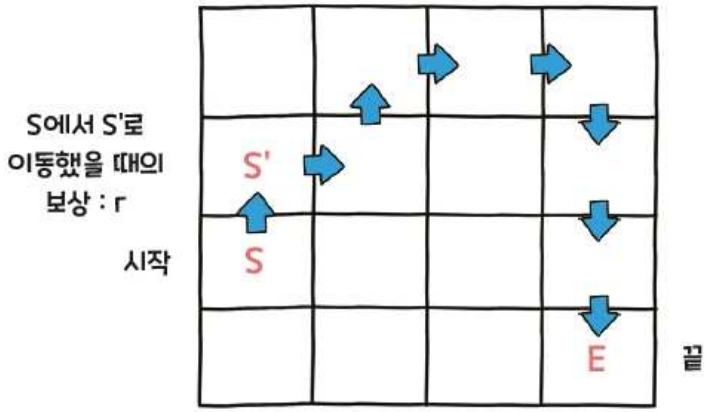


그림 8-29 큐러닝의 예

- 강화학습 기술

  2) 큐러닝

  - S에서 시작한 미로게임은 우왕좌왕하면서 우연히 E에 도달
  - 이때 에이전트는 S에서 S'로 이동했을 때 첫 번째 보상값(r)을 받음
  - 이후로는 E에 도달하기 위한 움직임을 계속하면서 보상값을 업데이트해 나갈 것
  - 이를 정리하면 다음과 같음

① 모든 환경 데이터값(상태, 행동)을 초기화함

② 현재 상태(S)를 확인함

③ 다음의 작업을 반복함

  - » S'로 이동
  - » 행동에 따른 보상값(r)을 받음
  - » 목적지에 도착할 때까지 이동과 보상값을 [그림 8-30]처럼 테이블에 기록

- **강화학습 기술**

  2) 큐러닝

그림 8-30 큐러닝 구조

04. 머신러닝 알고리즘의 유형

- 강화학습 기술

3) 딥큐러닝(Deep Q Leaning)

- 큐러닝에 신경망을 결합한 알고리즘
- 큐러닝에서는 보상값(r)을 업데이트하기 위해 테이블을 이용했다면, DQN에서는 네트워크(신경망)를 이용
- Q값은 전략에 따라 행동했을 때 미래의 보상들에 대한 기댓값의 총합
- 결국 큐러닝과 DQN 모두 Q값이 높은 쪽으로 행동하는 것을 목표로 함

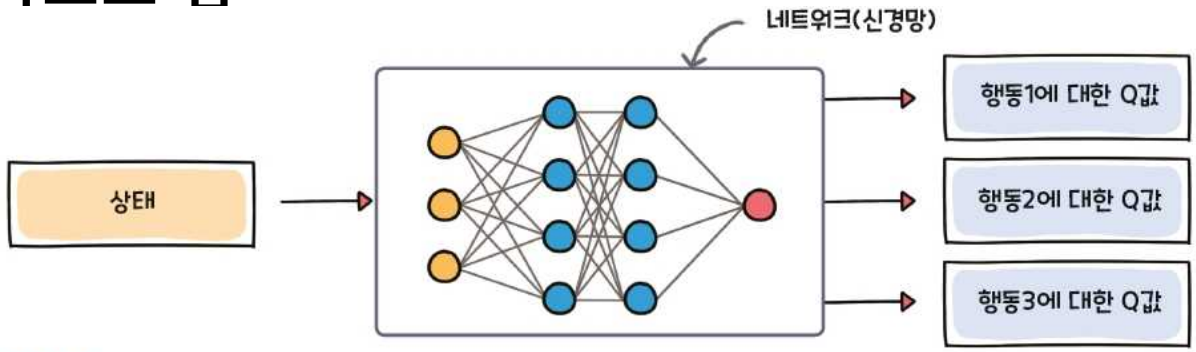


그림 8-31 딥큐러닝(DQN) 구조

05. 머신러닝의 주요 도전 과제

- **많은 데이터 확보**
  - 컴퓨터는 인간과 다르게 0, 1이라는 숫자만 인식할 수 있음
  - 즉, 이미지 안의 객체를 숫자로 표현해야 하고 객체 중에서도 자동차만을 인식할 수 있도록 별도의 처리(머신러닝의 가중치(weight)) 를 해주어야 한다는 뜻
  - 이렇게 복잡한 과정을 거치면서 다양한 유형의 자동차를 정확하게 인식하기 위해서는 수많은 데이터가 필요함

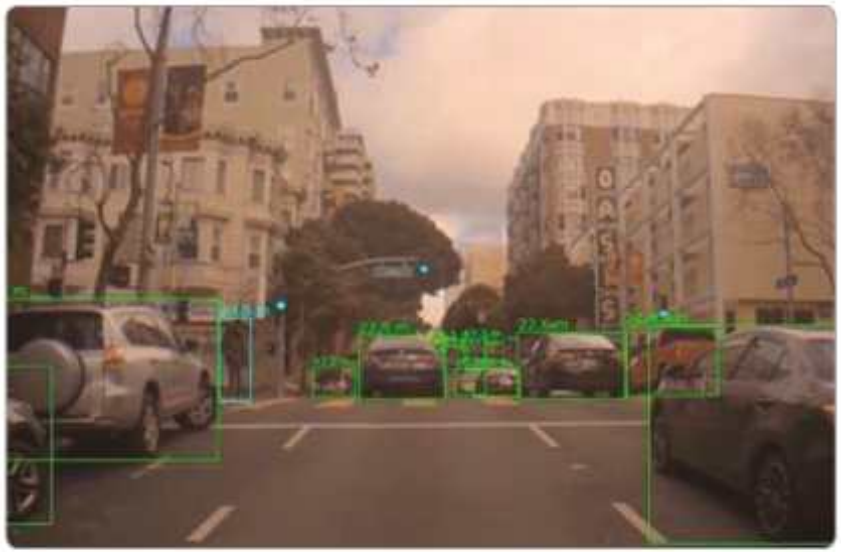


그림 8-32 인공지능의 자동차 인식

- **많은 데이터 확보**

하나 더 알기

가중치(Weight)

- **가중치(Weight) :** 입력 신호가 출력에 미치는 중요도를 조절하는 매개변수
- [그림 8-32]의 이미지에는 건물과 자동차들이 있음
- 각각의 객체에 단순히 숫자를 부여한다면 건물1=00001, 자동차1=00002, 건물2=00003, 자동차2=00004와 같이 부여되어 컴퓨터는 건물과 자동차를 구별할 수 없을 것
- 그래서 건물1=00001, 자동차1=11110과 같이 자동차에 더 높은 숫자를 부여하여 건물과 자동차를 구별할 수 있도록 한 것이 가중치임
- 실제 객체 인식 과정에서 사용되는 가중치는 더 복잡한 과정을 거치지만, 간단하게는 중요한 객체를 부각시키기 위한 값이라고 이해해도 좋음

05. 머신러닝의 주요 도전 과제

- 과적합 현상
  - 과적합(Overfitting)
    - 훈련 데이터를 너무 과하게 학습하여 실제 데이터를 분석할 때는 성능이 좋지 못한 것을 의미
    - 문제의 복잡도에 비해 데이터가 현저히 부족한 경우, 즉 문제가 정의 된 전체 공간을 학습 데이터가 아우르지 못하고 일부 경우에만 집중했을 때 발생함

05. 머신러닝의 주요 도전 과제

- **과적합 현상**

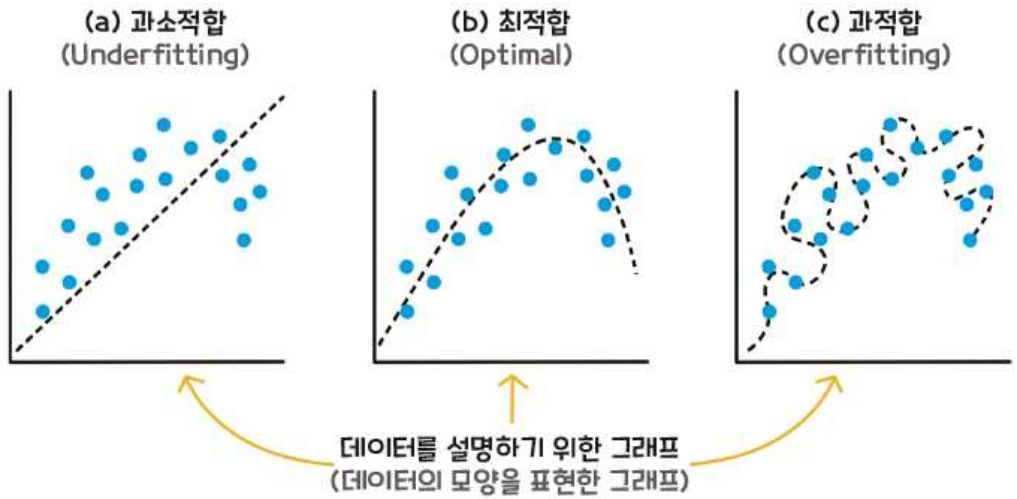


그림 8-33 과소적합, 최적합, 과적합 그래프

  - (a) 그래프 : 변수가 0인 쪽의 데이터 몇 개는 비교적 잘 근사하지만, 일정 시점 이후 데이터는 우하향하고 있음
  - (b) 그래프 : 데이터와 비슷하게 우상향하고 있어 제대로 반영하고 있다고 볼 수 있음

05. 머신러닝의 주요 도전 과제

- **과적합 현상**

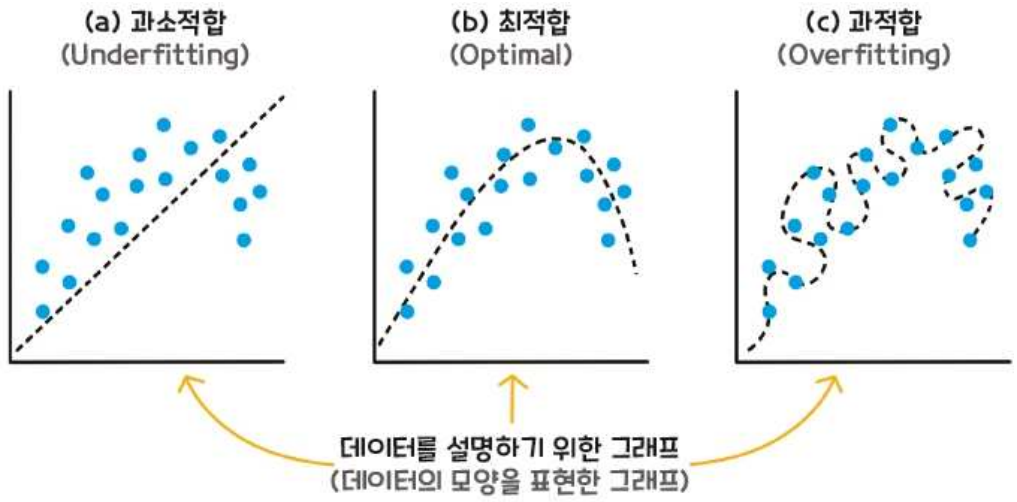


그림 8-33 과소적합, 최적합, 과적합 그래프

  - (c) 그래프 : 학습 데이터와 생성된 모델의 오차를 구해보면 0에 가까울 것임. 즉, 그래프가 모든 점을 지나고 있음
  - 그렇다면 (b) 그래프보다 (c) 그래프가 더 좋은 그래프일까? 어떤 그래프가 좋은 그래프인지 알아보기 위해서는 실제 데이터값을 불러오면 됨

05. 머신러닝의 주요 도전 과제

- **과적합 현상**

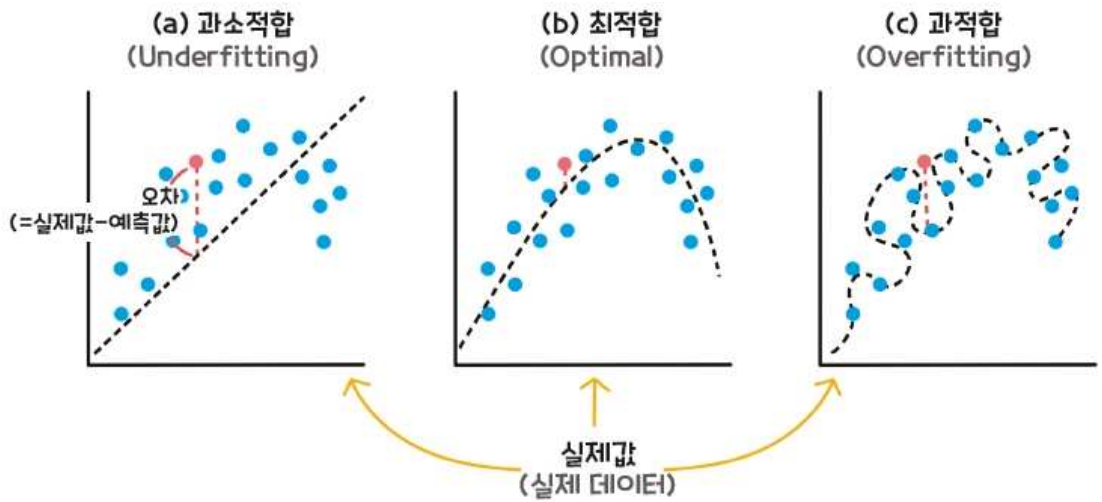


그림 8-34 과적합 판단 방법

  - [그림 8-34]와 같이 하나의 실제값을 불러왔을 때, 실제값과 모델이 내놓은 예측값의 차이인 오차가 가장 작은 그래프가 좋은 그래프라고 할 수 있음
  - 확인해 보면 (b) 그래프의 오차가 가장 작으므로 최적합(Optimal), 즉 가장 좋은 그래프라고 할 수 있음

05. 머신러닝의 주요 도전 과제

- **유연성 부족**
  - 머신러닝은 유연성이 부족함
  - 머신러닝은 데이터로 시작해서 데이터로 끝나는 기술
  - 다른 사람이 만들어 놓은 모델은 재활용이 가능하더라도 데이터는 공유 어려움
  - 실제로 공유 된 데이터를 사용할 수도 있지만 분석하고자 하는 변수 중 일부가 누락된 경우가 많기 때문에 공유 된 데이터를 이용한 분석은 그 목적에서 벗어난 경우가 많음
  - 따라서 제대로 학습을 하려면 원하는 결과를 위한, 목적에 맞는 '나만의 데이터'가 필요함
  - 결국 데이터가 없다면 머신러닝 알고리즘도, 딥러닝 알고리즘도 적용 어려움