



[인공지능 입문]

Part 01. 인공지능의 이해

Chapter 01. 인공지능의 개요

목차

1. 인공지능 소개
2. 인공지능의 중요성과 시장성
3. 인공지능의 연대기
4. 인공지능의 분류

01

인공지능의 역사

01. 인공지능의 역사

- 인공지능의 시작부터 최근까지의 연대를 시간 흐름순으로 정리

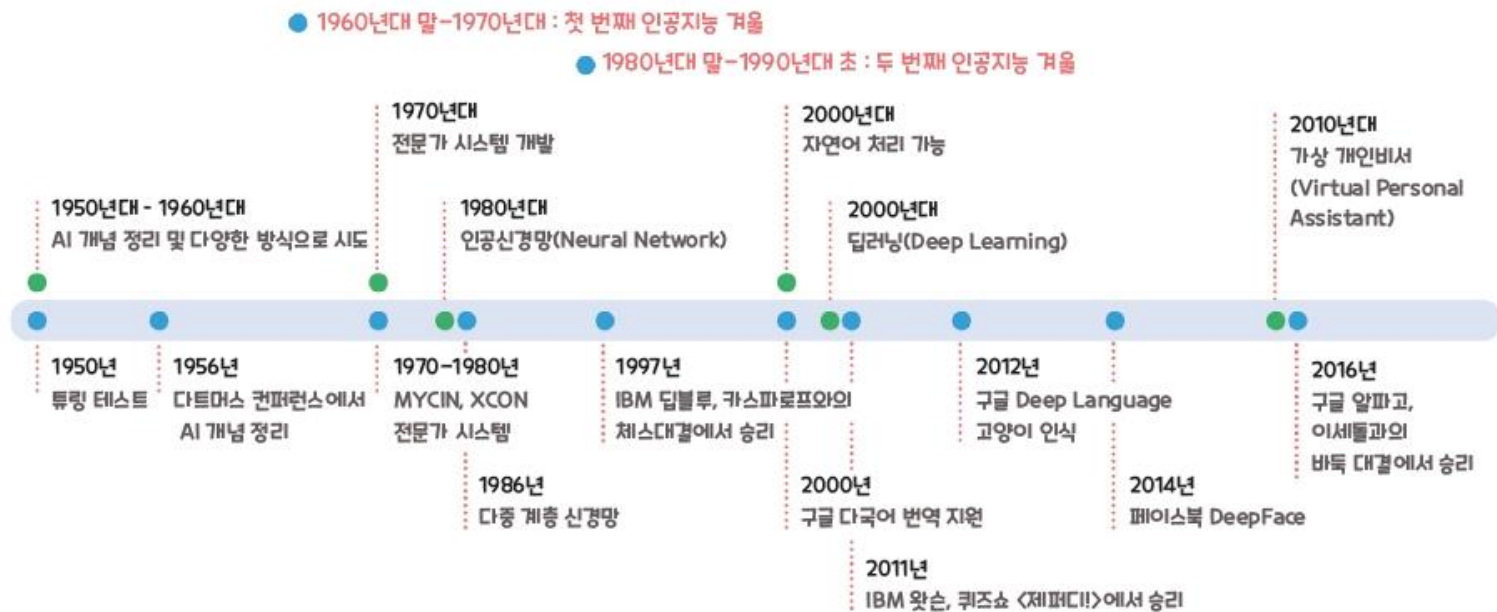


그림 1-17 인공지능 연대

01. 인공지능의 역사

• 인공지능 기술의 라이프 사이클

– 태동기-1차 암흑기-성장기-2차 암흑기-성숙기

표 1-2 인공지능 기술의 라이프사이클

구분	주요 연혁	설명
태동기	인공지능의 시작 (1950~1956년)	앨런 튜링은 「계산 기계와 지능」이라는 논문에서 기계가 생각할 수 있는지 테스트하는 방법과 지능적 기계의 개발 가능성 및 학습하는 기계 등에 대해 정의함.
	데이터 기반 분석 체계 구성 (1956~1974년)	10명의 과학자가 참여한 다트머스 컨퍼런스에서 '인공지능' 용어가 처음으로 사용됨.
1차 암흑기	첫 번째 겨울 (1974~1980년)	인공지능 시스템이 애초의 기대를 절대로 만족시키지 못할 것이라는 인식이 투자 삭감으로 이어지면서 첫 번째 인공지능 겨울이 시작됨.
성장기	전문가 시스템(1980~1987년)	전문가들의 지식을 데이터베이스로 구성한 규칙 기반 시스템이 탄생함.
2차 암흑기	두 번째 겨울 (1987~1993년)	전문가들의 지식 중 어떤 것을 추출해야 하는지에 대한 정의 부정확, 전문가 시스템의 개발 및 유지보수의 비효율성으로 인해 두 번째 인공지능 겨울이 시작됨.
성숙기	뉴럴 네트워크 (1993~2000년)	기존의 뉴럴 네트워크와는 달리 입력층과 출력층 사이에 은닉층이 추가되면서 복잡한 연산이 가능해짐.
	머신러닝과 딥러닝 (2000~2010년)	컴퓨터 스스로 데이터를 학습해서 문제를 해결할 수 있게 되었고, 학습한 데이터를 다른 문제에 재활용할 수 있게 됨. 또한 대량의 데이터 처리에 필요한 빅데이터의 확산과 하드웨어(GPU) 성능도 향상됨.
	알파고의 등장(2010~현재)	인공지능 기술의 발전 가능성에 대한 기대감이 상승하면서 관심이 높아짐.

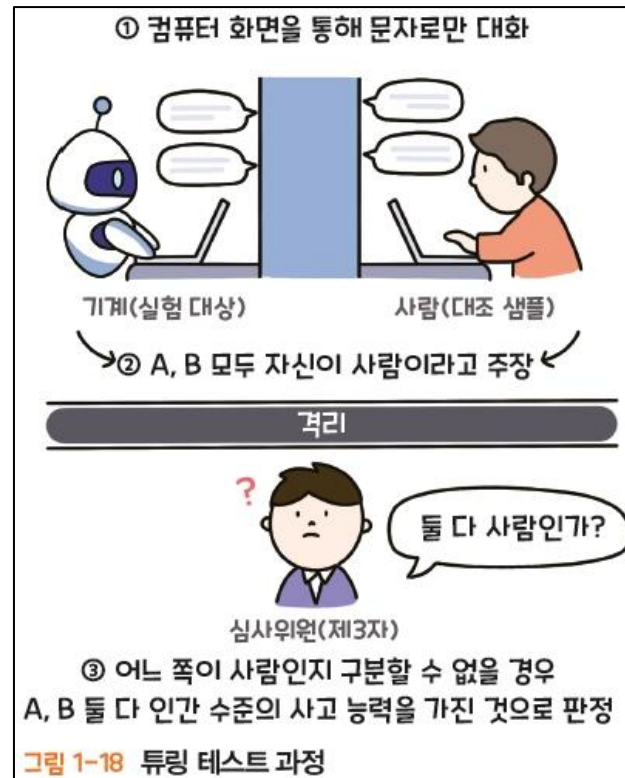
01. 인공지능의 역사

I. 1차 태동기 : 인공지능의 시작(1950~1956년)

- 튜링 테스트(Turing Test)

- 기계가 사람처럼 지능적으로 동작할 수 있는지 판단하는 테스트

- 튜링 테스트의 과정 →



01. 인공지능의 역사

I. 1차 태동기 : 인공지능의 시작(1950~1956년)

- 캡차(CAPTCHA)

- 2000년 미국 카네기 멜론 대학교의 연구원들이 사람과 로봇을 구별하기 위해 만든 튜링 테스트로, 로봇은 구별하기 난해한 문자를 제시하여 이를 맞추면 사람으로 인정하는 방식



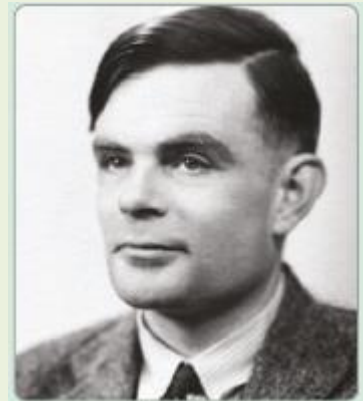
그림 1-19 캡차 튜링 테스트의 예

01. 인공지능의 역사

I. 1차 태동기 : 인공지능의 시작(1950~1956년)

하나 더 알기 앨런 튜링

- 앨런 튜링(Alan Turing)은 알고리즘과 수학 원리를 이용한 튜링 머신으로 컴퓨터 과학 발전에 큰 공헌을 하였으며, 튜링 테스트를 고안해 낸 것으로 유명함



01. 인공지능의 역사

II. 2차 태동기 : 데이터 기반 분석 체계 구성(1956~1974년)

- 인공지능을 컴퓨터 과학의 세부 영역으로 이끈 학자들은 '다트머스 컨퍼런스'에 참석했던 10명의 학자들이었음



존 맥카시



클로드 섀넌



마빈 민스키



나다니얼 로체스터



레이 솔로모노프



올리버 셀프리지



트렌처드 모어



아서 사무엘



앨런 뉴얼



허버트 사이먼

이들을 인공지능의 아버지라고 부릅니다.

그림 1-20 다트머스 컨퍼런스에 참석한 10명의 과학자들

01. 인공지능의 역사

II. 2차 태동기 : 데이터 기반 분석 체계 구성(1956~1974년)

- 1956년에 다트머스 대학교에서 열린 워크숍에서 이들은 인공지능(AI)이라는 이름과 연구개발 목표, 추진 방향 등을 제안하고 토론하였음
- '인공지능'이라는 단어는 '지능을 가진 기계'의 이름을 고민하던 중, 존 맥카시가 '인공지능'이라는 이름을 제안하면서 사용되기 시작했음
- 지금과 같은 수준의 인공지능을 다루는 것이 아니라 경우의 수를 계산하는 수준이었지만, 이들에 의해 지금 정도의 인공지능이 완성되었다고 할 수 있음

01. 인공지능의 역사

III. 1차 암흑기 : 인공지능의 첫 번째 겨울(1974~1980년)

- 허버트 사이먼은 "앞으로 20년 안에 기계는 사람이 할 수 있는 일은 무엇이든 할 수 있게 될 것"이라고 선언



1958년, 허버트 사이먼과 앨런 뉴얼 :

"10년 내에 디지털 컴퓨터가 체스 세계 챔피언을 이길 것이다. 그리고 10년 내에 디지털 컴퓨터는 중요한 새로운 수학적 정리를 발견하고 증명할 것이다."



1965년, 허버트 사이먼 :

"20년 내에 기계가 사람이 할 수 있는 모든 일을 하게 될 것이다."



1967년, 마빈 민스키 :

"이번 세기에 인공지능을 만드는 문제는 거의 해결될 것이다."



1970년, 마빈 민스키 :

"3~8년 안에 우리는 평균 정도의 인간 지능을 가진 기계를 가지게 될 것이다."

01. 인공지능의 역사

III. 1차 암흑기 : 인공지능의 첫 번째 겨울(1974~1980년)

- 다트머스 컨퍼런스 이후 많은 과학자들이 인공지능 개발을 위한 연구에 뛰어들었지만, 기대와 달리 연구 성과가 낮아 인공지능에 대한 투자가 적어지면서 재정적 위기를 맞이함
- 낙관의 거품이 걷히자 인공지능은 비판의 대상이 되었고 인공지능 역사의 첫 번째 겨울(First AI Winter)이 시작됨

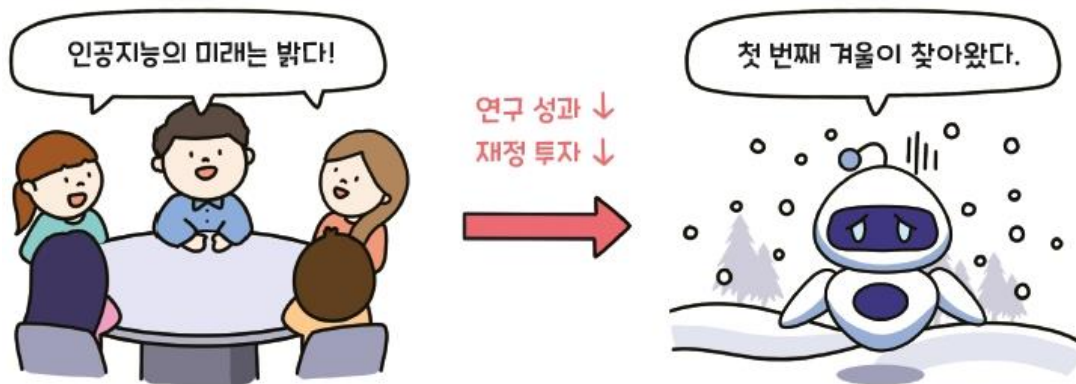


그림 1-22 인공지능의 첫 번째 겨울

01. 인공지능의 역사

IV. 성장기 : 전문가 시스템(1980~1987년)

- 전문가 시스템(Expert System)

- 인간이 특정 분야에 대하여 가지고 있는 전문적인 지식을 정리하고 표현하여 컴퓨터에 기억시킴으로써 일반인도 전문지식을 이용할 수 있도록 하는 시스템

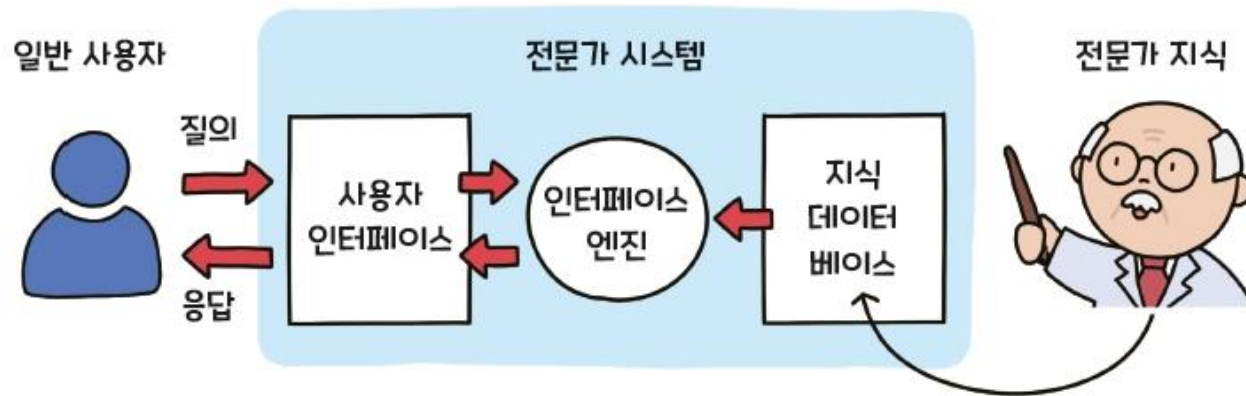


그림 1-23 전문가 시스템 추론 과정

01. 인공지능의 역사

V. 2차 암흑기 : 인공지능의 두 번째 겨울(1987~1993년)

- 전문가 시스템 유지비가 비싸고 전문가의 지식을 추출하는 데 병목현상이 발생하면서 시스템을 유지하는 것이 어려웠음
- 뿐만 아니라 데이터베이스에 축적되지 않은 질문들에 대해서는 예측할 수 없는 행동을 하는 전문가 시스템에 대해 사람들은 의구심을 가짐
- 이로 인해 인공지능의 **두 번째 겨울(Second AI Winter)**이 시작됨

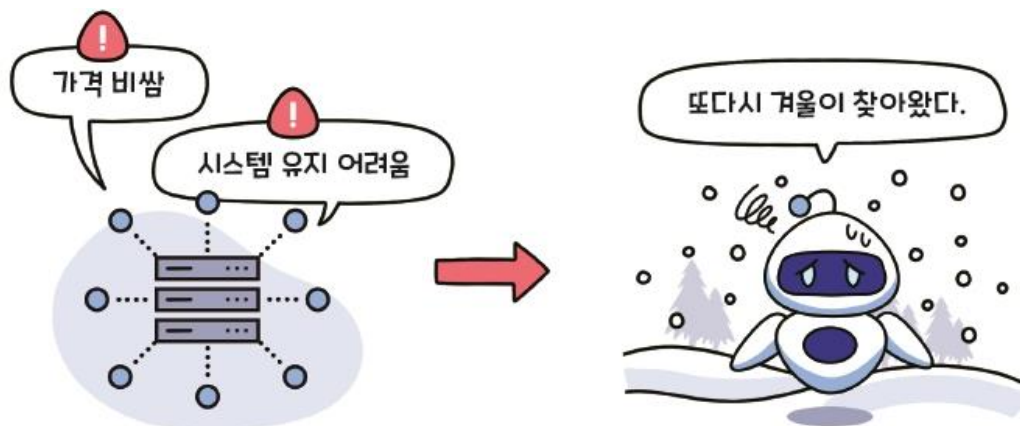


그림 1-24 인공지능의 두 번째 겨울

01. 인공지능의 역사

VI. 1차 성숙기 : 뉴럴 네트워크(1993~2000년)

- 1990년대 인공지능 연구는 인터넷과 함께 다시 한 번 중흥기를 맞이함
- 이전의 인공지능은 사람이 규칙을 만들어 시스템을 구현했다면,
→ 성숙기 단계의 인공지능은 공식을 스스로 만듦
→ 머신러닝과 딥러닝으로 발전

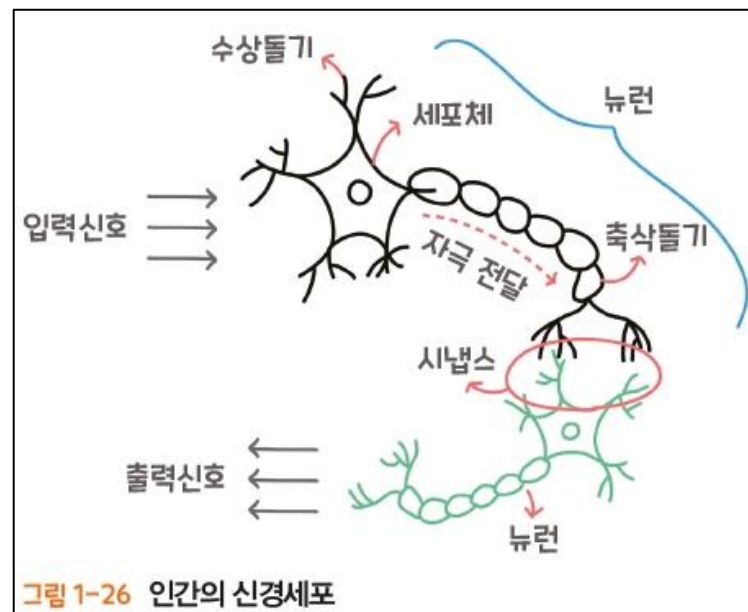


그림 1-25 머신러닝, 딥러닝을 통한 인공지능 체제 전환

01. 인공지능의 역사

VI. 1차 성숙기 : 뉴럴 네트워크(1993~2000년)

- 인간의 뇌 신경망을 모방한 인공신경망 연구는 인공지능 발전에 큰 영향을 미침
- 신경세포들의 연결로 이루어진 인간의 뇌는 시냅스를 통해 전기 자극을 전달함
- 이것을 그대로 모사한 인공신경망은 뉴런이 일정한 자극을 받으면 다음 뉴런으로 신호가 전달되는 방식으로 동작함



01. 인공지능의 역사

VI. 1차 성숙기 : 뉴럴 네트워크(1993~2000년)

- 인공지능 알고리즘의 구분
 - 입력층(Input Layer)
 - 은닉층 (Hidden Layer)
 - 출력층(Output Layer)

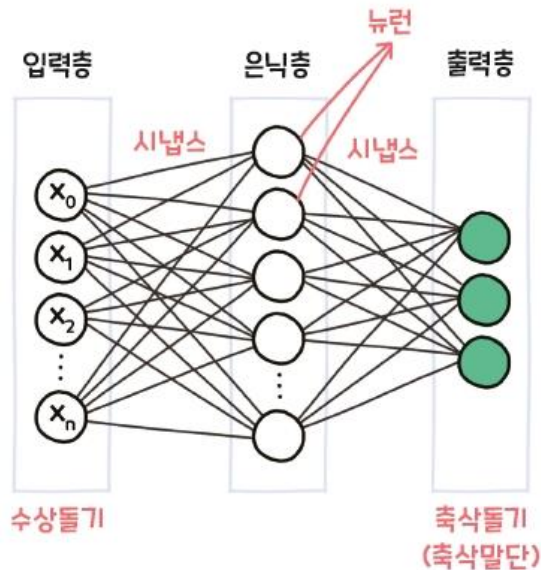


그림 1-27 인간의 신경세포를 모방한 인공지능망

인공신경망 알고리즘이란
유한한 시간 내에 특정 문제를 해결하기 위해
일련의 순서로 진행되는 계산 및 풀이 절차의
집합을 말합니다.



01. 인공지능의 역사

VI. 1차 성숙기 : 뉴럴 네트워크(1993~2000년)

- 입력층은 뇌의 수상돌기, 출력층은 축삭돌기(축삭말단)에 해당
- 입력층 사이의 연결 강도는 시냅스에 해당하므로 일정한 자극이 있을 때만 다음 층(인간의 뉴런)으로 전달
- 특히 2개 이상의 은닉층을 구성함으로써 복잡한 문제들을 쉽게 해결할 수 있게 되면서 인공지능은 성숙기를 맞이하게 됨

01. 인공지능의 역사

VII. 2차 성숙기 : 머신러닝과 딥러닝(2000~2010년)

1) 머신러닝(Machine Learning)

- 머신러닝(Machine Learning)

- 기본적으로 알고리즘을 이용해 데이터를 분석하고, 분석을 통해 학습하며, 학습한 내용을 기반으로 판단이나 예측

2) 딥러닝(Deep Learning)

- 딥러닝(Deep Learning)

- 인공신경망에서 발전한 형태의 인공지능으로, 인간 뇌의 뉴런과 유사한 입력층, 은닉층, 출력층을 활용해 데이터를 학습

01. 인공지능의 역사

VII. 2차 성숙기 : 머신러닝과 딥러닝(2000~2010년)

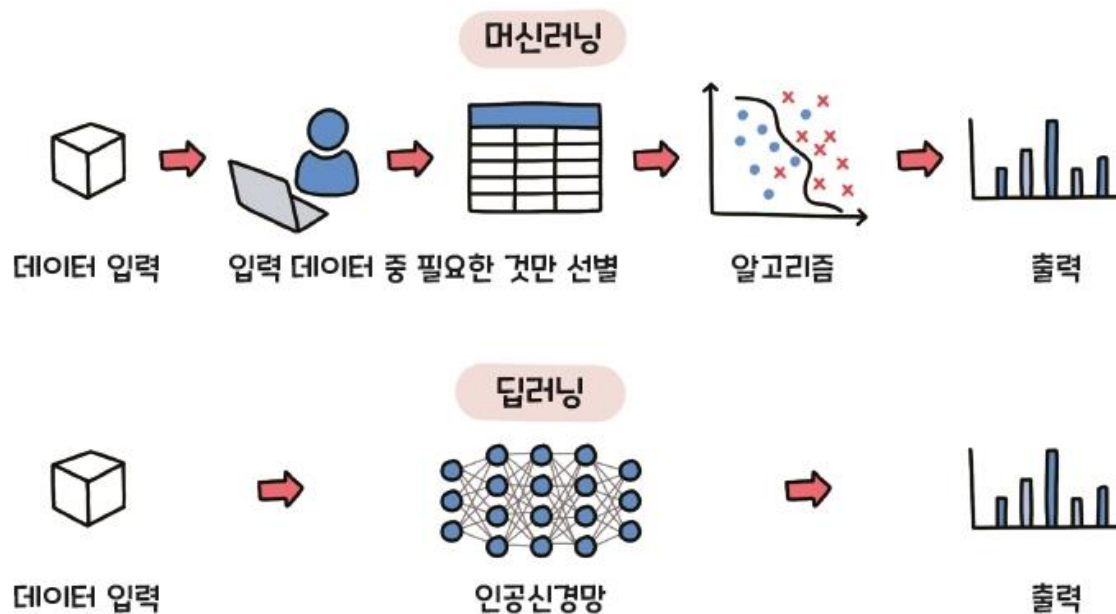


그림 1-28 머신러닝과 딥러닝의 학습 차이

01. 인공지능의 역사

VIII. 3차 성숙기 : 알파고의 등장(2010~현재)

- 영화 《신의 한 수 : 귀수 편》에는 다양한 스타일로 자신만의 바둑을 두는 바둑의 고수 등장
- 현실 속 바둑의 고수로는 알파고(AlphaGo)를 떠올릴 수 있음



그림 1-29 영화 《신의 한 수 : 귀수 편》

01. 인공지능의 역사

VIII. 3차 성숙기 : 알파고의 등장(2010~현재)

- 이세돌과 승부를 겨뤘던 알파고 리(AlphaGo Lee)는 나날이 성장하여 알파고 제로(AlphaGo Zero)로 완성됨
- 알파고 제로는 학습을 하는 데 인간의 기본 데이터가 필요하지 않음
- 알파고 제로는 알파고 리의 실력을 압도하는 기력(棋歷)을 불과 72시간만에 얻었으며, 알파고 리와의 경기에서 백전백승함

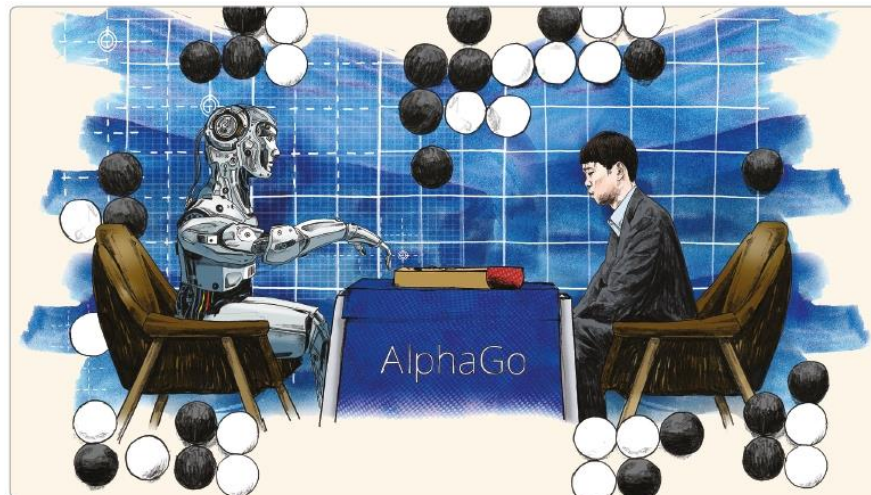


그림 1-30 알파고의 충격적인 등장과 인공지능에 대한 관심 © Financial Times

02

인공지능의 분류

01. 인공지능의 분류

- 인공지능은 지적 수준, 기능 발전, 구현 방식에 따라 분류할 수 있음

표 1-3 인공지능의 분류

분류	기준	상세 분류
지적 수준	인간과 같은 사고의 가능 여부	<ul style="list-style-type: none">• 약인공지능• 초인공지능• 강인공지능
기능 발전(레벨)	입력에 따른 출력이 변하는 에이전트 관점	<ul style="list-style-type: none">• 레벨 1 : 단순 제어 프로그램• 레벨 2 : 고전적 인공지능• 레벨 3 : 머신러닝• 레벨 4 : 딥러닝
구현 방식	지적 기능 구현 방식	<ul style="list-style-type: none">• 지식 기반 방법론(인지, 추론, 학습, 행동)• 데이터 기반 방법론(머신러닝, 데이터마이닝)

01. 인공지능의 분류

I. 지적 수준에 따른 분류

- 약인공지능
- 강인공지능
- 초인공지능

표 1-4 지적 수준에 따른 인공지능 분류

분류	설명	사례
약인공지능	특정 문제해결에 전문화된 인공지능	스팸메일 필터링, 검색 서비스, 구글번역, 유튜브 영상 추천
강인공지능	모든 영역에서 인간과 같은 수준인 인공지능	영화 《터미네이터》에 등장하는 스카이넷, 비서로봇, 협동로봇 (공장로봇)
초인공지능	인류 전체의 지능을 초월하는 인공지능	‘인류가 앞으로 1,000년 동안 쓸 수 있는 신(新) 에너지원 만들기’와 같은 고차원의 명령 수행 기능

01. 인공지능의 분류

I. 지적 수준에 따른 분류

1) 약인공지능

- 약인공지능(Weak AI, Artificial Narrow Intelligence)
 - 한 분야의 특정한 일을 인간의 지시에 따라 수행하는 인공지능
 - 어떠한 목적에 최적화된 알고리즘으로, 적당한 규칙에 의해 구현됨
 - (예) 알파고는 바둑, 테슬라의 자율주행차는 자율주행에만 최적화



그림 1-31 약인공지능의 예 : 테슬라의 자율주행차

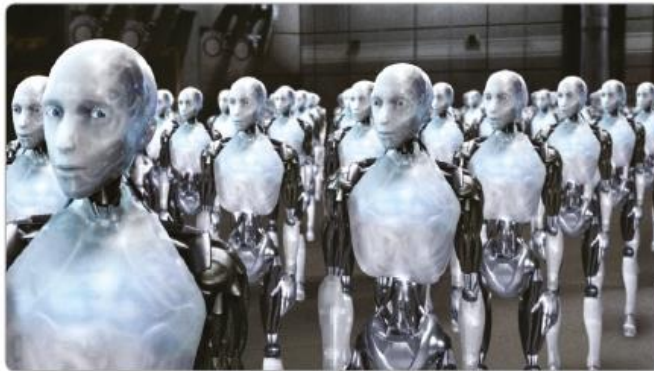
01. 인공지능의 분류

I. 지적 수준에 따른 분류

2) 강인공지능

- 강인공지능(Strong AI, Artificial General Intelligence)

- 약인공지능의 제한된 기능을 뛰어넘어 모든 산업 분야에서 범용적으로 사용되는 인공지능
- 지각력이 있고 스스로를 인식할 수 있는 존재



(a) 《아이, 로봇》



(b) 《터미네이터》

그림 1-32 영화 속에서 등장하는 강인공지능

01. 인공지능의 분류

I. 지적 수준에 따른 분류

3) 초인공지능

- 초인공지능(Artificial Super Intelligence)
 - 인간보다 몇 백배 이상 뛰어난 지능을 가진 존재
 - 연산 능력뿐만 아니라 과학 및 예술 분야에서 뛰어난 창의성을 발휘
 - 일반적인 지식과 사회적인 능력도 인간을 뛰어넘는 지능을 가짐



그림 1-33 초인공지능

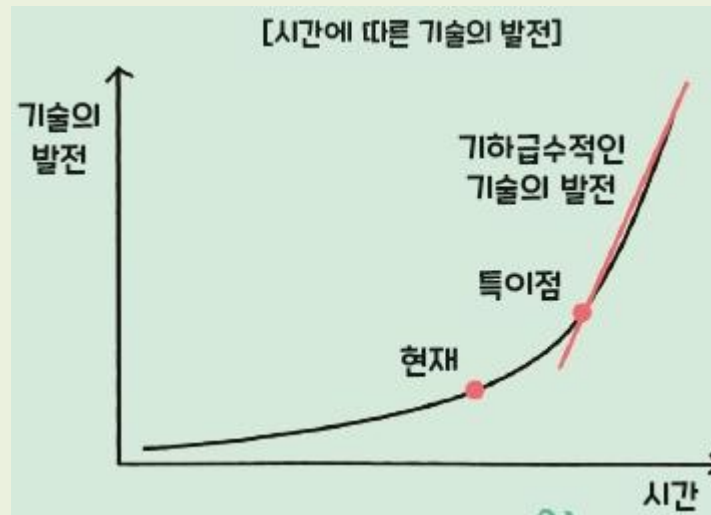
01. 인공지능의 분류

I. 지적 수준에 따른 분류

하나 더 알기

특이점

- **특이점(Singularity)** : 인공지능의 발전이 가속화되어 모든 인류의 지성을 합친 것보다 더 뛰어난 초인공지능이 출현하는 시점



01. 인공지능의 분류

II. 기능 발전에 따른 분류

- 단순 제어 프로그램
- 고전적 인공지능
- 머신러닝이 도입된 인공지능
- 딥러닝이 도입된 인공지능

표 1-5 기능 발전에 따른 인공지능 분류

© 마쓰오 유타카, 『인공지능과 딥러닝』

레벨	분류	설명	응용 제품
1	단순 제어 프로그램	<ul style="list-style-type: none">• 단순한 제어 프로그램• 엄밀한 의미로 인공지능은 아님	세탁기, 전자면도기 등
2	고전적 인공지능	<ul style="list-style-type: none">• 입력과 출력의 조합 수가 극단적으로 많은 경우• 추론 및 탐색을 하거나 기존의 지식베이스를 기반으로 판단	고전적 퍼즐
3	머신러닝이 도입된 인공지능	<ul style="list-style-type: none">• 빅데이터를 바탕으로 자동 판단• 전형적으로 머신러닝 알고리즘 이용	지도학습, 비지도학습, 강화학습
4	딥러닝이 도입된 인공지능	<ul style="list-style-type: none">• 입력 데이터 자체를 학습해 스스로 판단하고 예측	구글의 고양이 인식

01. 인공지능의 분류

II. 기능 발전에 따른 분류

1) 단순 제어 프로그램

- On/Off, Yes/No와 같이 필요한 변수가 두 가지 뿐인 가장 기초적인 단계의 인공지능
- (예) 세탁기, 청소기 등



그림 1-34 단순 제어 프로그램의 예 : 세탁기와 청소기

01. 인공지능의 분류

II. 기능 발전에 따른 분류

2) 고전적 인공지능

- 단순 제어 프로그램과 크게 다르지 않은 알고리즘
- 단순 제어 프로그램과의 차이는 패턴이 다양해진 것뿐
- 경우의 수에 따라 행동하기 때문에 '학습'이 아닌 '정해진 규칙'에 따라 행동
- (예) 로봇 청소기와 바둑 게임 프로그램 등



그림 1-35 고전적 인공지능의 예 : 로봇 청소기와 바둑 게임 프로그램

01. 인공지능의 분류

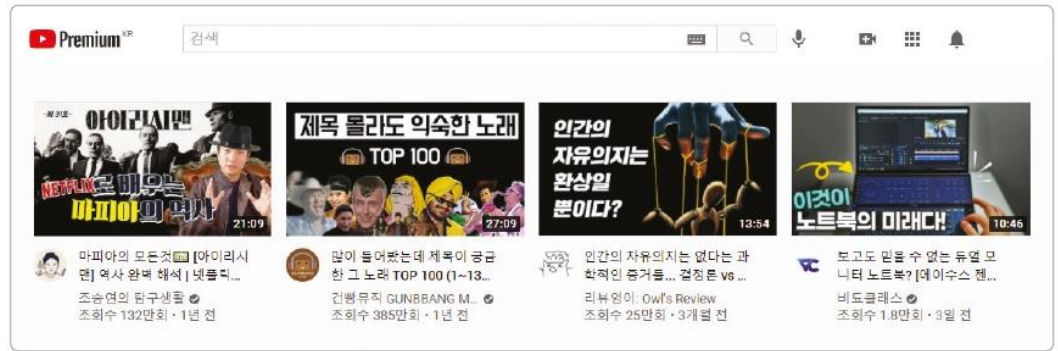
II. 기능 발전에 따른 분류

3) 머신러닝이 도입된 인공지능

- 적절한 판단을 위해 보유한 지식을 기반으로 추론 및 탐색하는 인공지능
- (예) 스마트폰의 얼굴인식, 유튜브의 맞춤 동영상 추천 시스템 등



(a) 스마트폰의 얼굴인식



(b) 유튜브의 맞춤 동영상 추천 시스템

그림 1-36 머신러닝이 도입된 인공지능의 예

01. 인공지능의 분류

II. 기능 발전에 따른 분류

4) 딥러닝이 도입된 인공지능

- 머신러닝보다 발전하여 데이터 자체를 스스로 학습하는 수준의 인공지능
- 머신러닝을 이용한 학습 과정에는 인간이 개입하지만, 딥러닝을 이용한 학습 과정에서는 기계가 스스로 해답을 찾아 분류나 예측을 수행

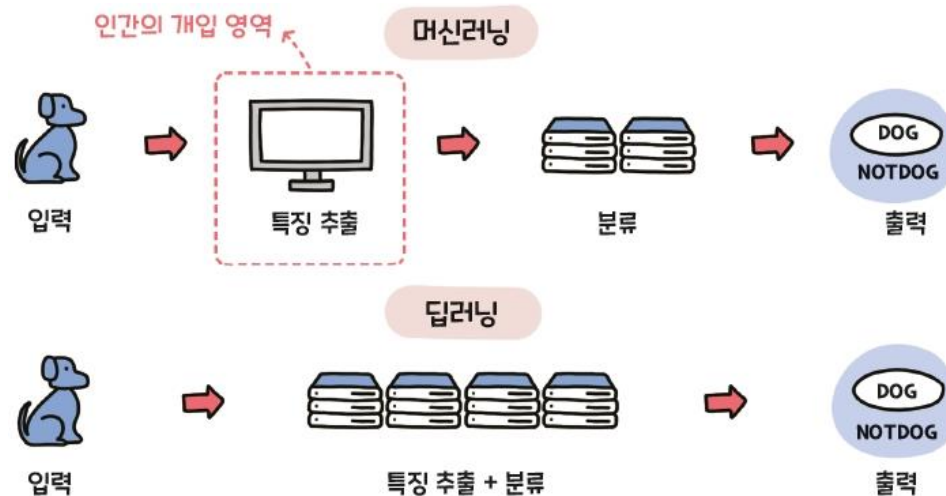


그림 1-37 머신러닝과 딥러닝 차이

01. 인공지능의 분류

II. 기능 발전에 따른 분류

4) 딥러닝이 도입된 인공지능

하나 더 알기

특징 추출

- **특징 추출(Feature Extraction)** : 머신러닝에서 컴퓨터가 스스로 학습하려면 사람이 인지하는 데이터를 컴퓨터가 인지할 수 있는 데이터로 변환해야 하는데, 이때 데이터별로 어떤 특징을 가지고 있는지를 찾아내고 그것을 토대로 데이터를 벡터로 변환하는 작업

01. 인공지능의 분류

II. 기능 발전에 따른 분류

4) 딥러닝이 도입된 인공지능

– (예) 자율주행차와 지능형 CCTV 등



(a) 자율주행차



(b) 지능형 CCTV

그림 1-38 딥러닝이 도입된 인공지능의 예

01. 인공지능의 분류

III. 구현 방식에 따른 분류

- 지식 기반 방법론
- 데이터 기반 방법론

표 1-6 구현 방식에 따른 인공지능 분류

분류	설명	핵심 기능 영역
지식 기반 방법론	저장된 지식을 기반으로 의사결정 수행	• 인지 • 추론 • 학습 • 행동
데이터 기반 방법론	데이터로부터 추출된 지식으로 의사결정 수행	• 머신러닝 • 데이터마이닝

01. 인공지능의 분류

III. 구현 방식에 따른 분류

1) 지식 기반 방법론

- 인지, 학습, 추론, 행동과 같은 인간의 지적 기능을 모방하기 위해 지적 기능들을 기호로 표현하고 이를 논리적인 규칙에 근거하여 처리함으로써 문제를 해결하고자 하는 방법이 시도되었음

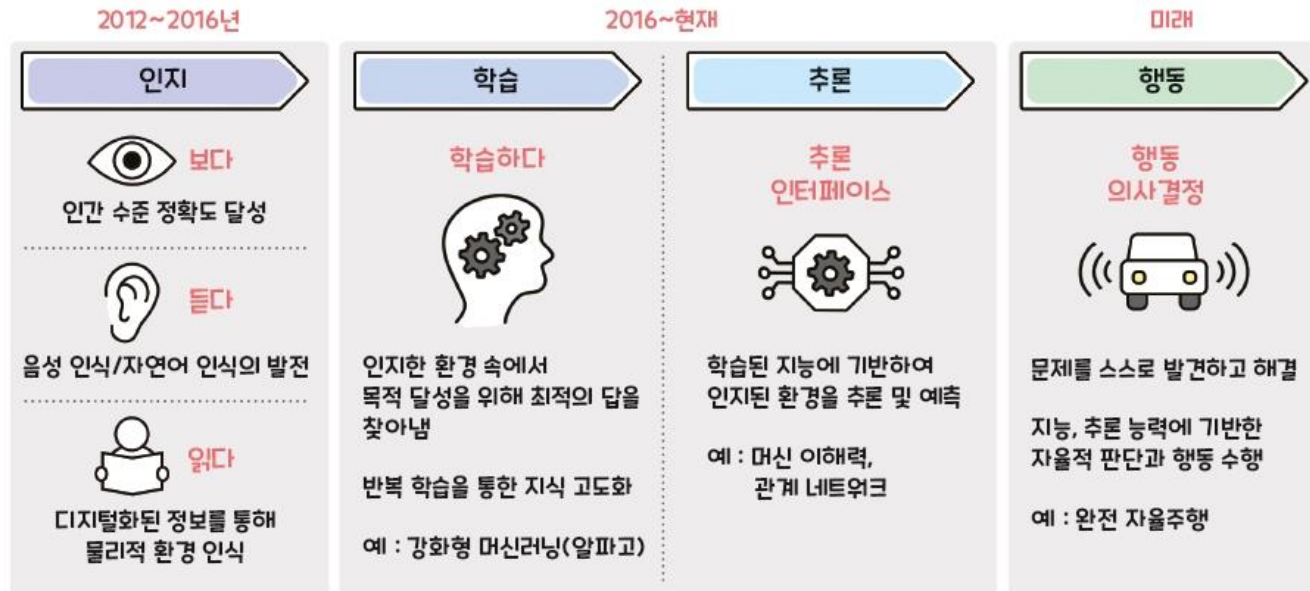


그림 1-39 지식 기반 방법론의 발전 과정

01. 인공지능의 분류

III. 구현 방식에 따른 분류

1) 지식 기반 방법론

■ 인지(Recognition)

- 컴퓨터가 특정 대상을 보고 듣고 읽을 수 있게 하려는 것
- 인지의 종류 : 자연어 처리, 통번역, 글씨/음성 인식, 이미지/동영상 인식 기술 등

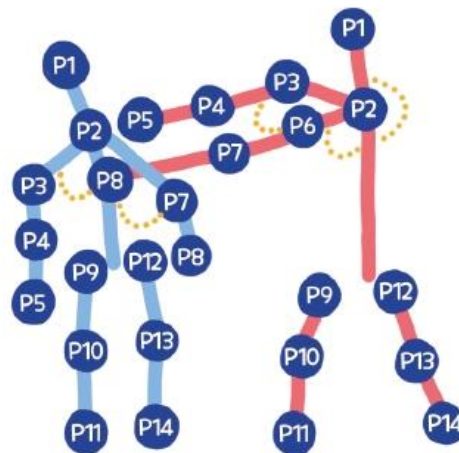
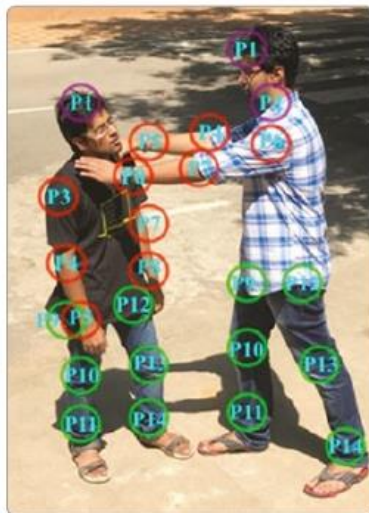


그림 1-40 드론 카메라를 통한 사람의 형상 인식

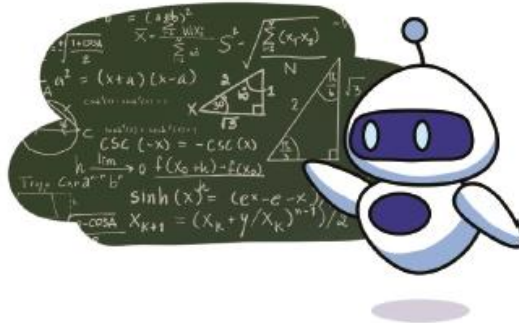
01. 인공지능의 분류

III. 구현 방식에 따른 분류

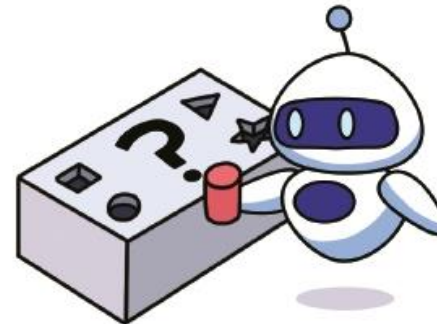
1) 지식 기반 방법론

▪ 학습(Learning), 추론(Mechanism)

- 데이터를 이용하여 규칙(Rule)을 만들고 결론을 도출하는 영역
- 머신러닝과 딥러닝이 여기에 속함



(a) 학습 영역



(b) 추론 영역

그림 1-41 학습 영역과 추론 영역

01. 인공지능의 분류

III. 구현 방식에 따른 분류

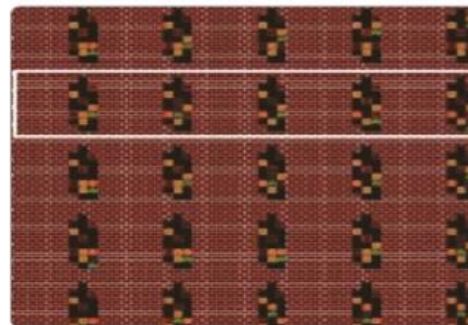
1) 지식 기반 방법론

■ 행동(Behavior)

- 기계가 인간과 같이 사고하고 행동하는 영역으로, 매 순간 결정(Decision)을 내리고 행동할 수 있도록 전략(Policy)과 보상(Reward) 개념 사용
- 딥마인드는 이러한 과정을 '소코반(Sokoban)' 게임을 이용하여 증명



(a) 현재 상황



(b) 현재 상황에서 행동 수행 시 발생할 미래 상황들

그림 1-42 인공지능의 행동 영역 증명: 소코반 게임

01. 인공지능의 분류

III. 구현 방식에 따른 분류

2) 데이터 기반 방법론

- 제공된 데이터로부터 연역적으로 지식을 추출하여 문제를 해결하는 방법론

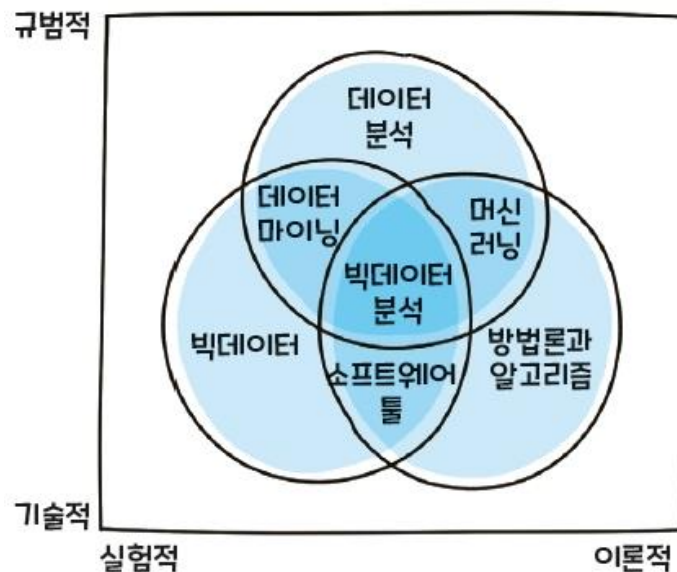


그림 1-43 데이터마이닝과 머신러닝의 관계

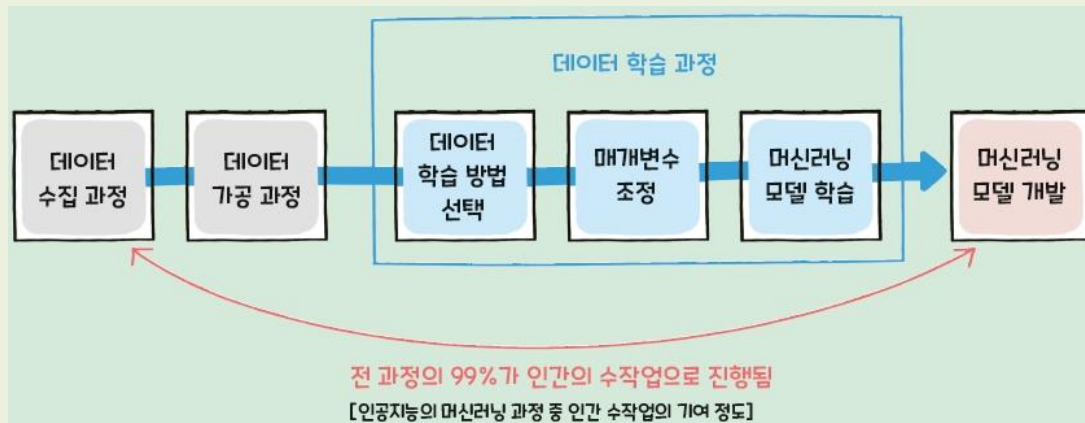
01. 인공지능의 분류

III. 구현 방식에 따른 분류

하나 더 알기 인공지능의 한계

1) 머신러닝의 99%는 인간의 수작업에 의존

- 머신러닝을 유지하기 위해 많은 양의 데이터를 분류하고 정리하는 것이 필요
- 데이터를 분류하고 적절한 알고리즘을 제공하는 과정에는 사람이 개입해야 함
- 결국 인공지능의 핵심은 인간의 노동력에 기반한다고 할 수 있음



01. 인공지능의 분류

III. 구현 방식에 따른 분류

하나 더 알기 인공지능의 한계

2) 편향된 데이터로 학습할 경우 인공지능(AI)도 편견 형성

- MS의 챗봇 '테이(Tay)'에 네티즌들이 인종차별 같은 모욕적인 단어들을 주입시킨 결과, 테이는 불쾌한 발언들을 쏟아내기 시작함
- 이 사건을 통해 인간이 어떤 유형의 데이터를 제공하는지에 따라 인공지능도 편견을 형성할 수 있다는 결론이 나옴



왜 기계가 편향되어질까요?

기술을 프로그래밍하는 데 있어 적절한 주의가 없으면 프로그래머의 편견도 결과에 영향을 미칠 가능성이 있습니다. 즉, 교육 데이터 세트를 공격하는 악의적인 해커 때문에 기계가 변형될 수 있는 것입니다.



03

머신러닝의 개요

01. 머신러닝이란?

- 머신러닝은 데이터에서부터 학습하도록 컴퓨터를 프로그래밍하는 과학(또는 예술)
 - "머신러닝은 명시적인 프로그래밍 없이 컴퓨터가 학습하는 능력을 갖 추게 하는 연구 분야"
 - 아서 새뮤얼(Arthur Samuel), 1959
 - "어떤 작업 T에 대한 컴퓨터 프로그램의 성능을 P로 측정했을 때 경험 E로 인해 성능이 향상됐다면, 이 컴퓨터 프로그램은 작업 T와 성능 측정 P에 대해 경험 E로 학습한 것"
 - 톰 미첼(Tom Mitchell), 1997
 - 스팸 필터는 (사용자가 스팸이라고 지정한) 스팸 메일과 일반 메일의 샘플을 이용해 스팸 메일 구분법을 배울 수 있는 머신러닝 프로그램의 하나

01. 머신러닝이란?

- 머신러닝은 데이터에서부터 학습하도록 컴퓨터를 프로그래밍하는 과학(또는 예술)
 - 기본 용어
 - 훈련 세트(training set): 시스템이 학습하는 데 사용하는 샘플
 - 훈련 사례(training instance(혹은 샘플): 각 훈련 데이터,
 - » 이 경우 작업 T는 새로운 메일이 스팸인지 구분하는 것
 - » 경험 E는 훈련 데이터(training data)
 - » 성능 측정 P는 직접 정의해야 하며, 이 성능 측정을 정확도accuracy라고 부르며 분류 작업에 자주 사용됨

02. 왜 머신러닝을 사용하는가?

- 전통적 프로그래밍 기법으로는 규칙이 점점 길고 복잡해지므로 유지 보수하기 매우 힘들
 - 머신러닝 기법에 기반을 둔 스팸 필터는 일반 메일에 비해 스팸에 자주 나타나는 패턴을 감지하여 어떤 단어와 구절이 스팸 메일을 판단하는데 좋은 기준인지 자동으로 학습합니다
- 전통적인 방식으로는 너무 복잡하거나 알려진 알고리즘이 없는 분야(예:음성인식)

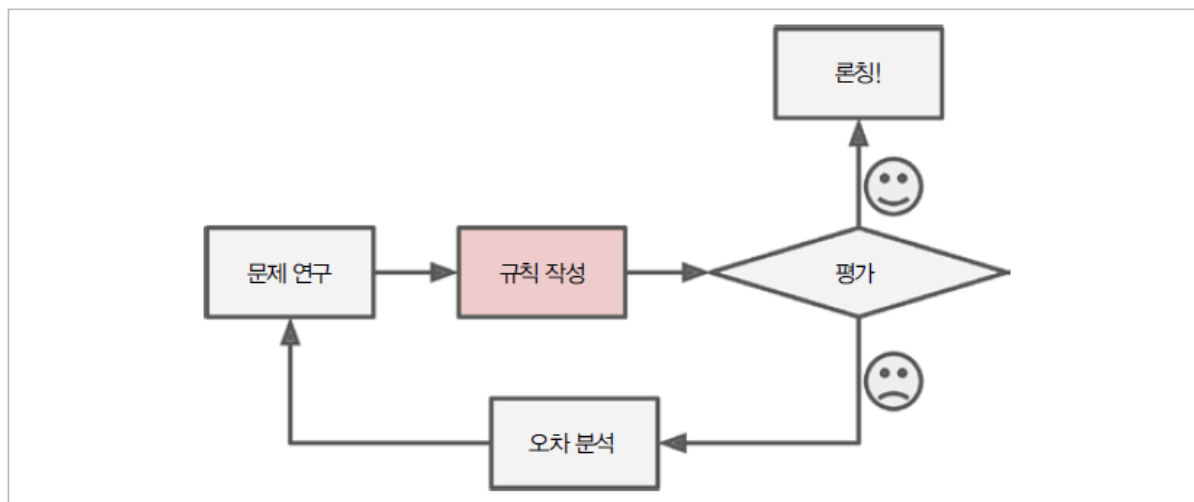


그림 1-1 전통적인 접근 방법

02. 왜 머신러닝을 사용하는가?

■ 머신러닝의 장점

- 기존 솔루션으로는 많은 수동 조정과 규칙이 필요한 문제: 하나의 머신러닝 모델이 코드를 간단하게 만들고 전통적인 방법보다 더 잘 수행되도록 할 수 있음
- 전통적인 방식으로는 해결 방법이 없는 복잡한 문제: 가장 뛰어난 머신러닝 기법으로 해결 방법을 찾을 수 있음
- 유동적인 환경: 머신러닝 시스템은 새로운 데이터에 적응 가능
- 복잡한 문제와 대량의 데이터에서 통찰 얻기

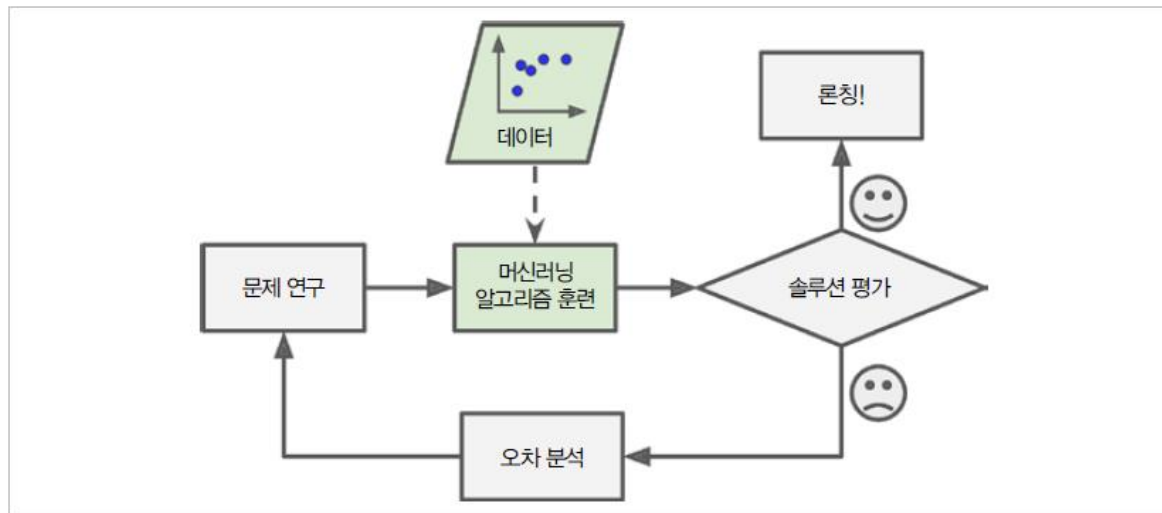


그림 1-2 머신러닝 접근 방법

03. 머신러닝 시스템의 종류

■ 넓은 범주의 분류

- 사람의 감독하에 훈련하는 것인지 그렇지 않은 것인지: 지도, 비지도, 강화 학습
- 실시간으로 점진적인 학습을 하는지 아닌지: 온라인 학습과 배치 학습
- 단순히 알고 있는 데이터 포인트와 새 데이터 포인트를 비교하는 것인지 아니면 과학자처럼 훈련 데이터셋에서 패턴을 발견하여 예측 모델을 만드는지: 사례 기반 학습과 모델 기반 학습

• 지도 학습과 비지도 학습

- 지도 학습
- 비지도 학습
- 강화 학습

• 배치 학습과 온라인 학습

- 배치 학습
- 온라인 학습

• 사례 기반 학습과 모델 기반 학습

- 사례 기반 학습
- 모델 기반 학습

03. 머신러닝 시스템의 종류

■ 지도 학습과 비지도 학습

- 지도 학습

알고리즘에 주입하는 훈련 데이터에 레이블(label)이라는 원하는 답이 포함된다.

- 분류
- 특성(예측 변수)
- 회귀

- 중요한 지도학습 알고리즘들

- k-최근접 이웃
- 선형 회귀
- 로지스틱 회귀
- 서포트 벡터 머신
- 결정 트리와 랜덤 포레스트
- 신경망

03. 머신러닝 시스템의 종류

■ 지도 학습과 비지도 학습

- 비지도 학습

- 훈련 데이터에 레이블이 없어서, 시스템이 아무런 도움 없이 학습해야 한다.

- 중요한 비지도학습 알고리즘들

- 군집

- » k-평균

- » DBSCAN

- » 계층 군집 분석

- » 이상치 탐지와 특이치 탐지

- » 원-클래스 SVM

- » 아이솔레이션 포레스트

- 시각화와 차원 축소

- » 주성분 분석(PCA)

- » 커널 PCA

- » 지역적 선형 임베딩

- » t-SNE

03. 머신러닝 시스템의 종류

■ 지도 학습과 비지도 학습

- 비지도 학습

- 훈련 데이터에 레이블이 없어서, 시스템이 아무런 도움 없이 학습해야 한다.

- 중요한 비지도학습 알고리즘들

- 군집

- » k-평균

- » DBSCAN

- » 계층 군집 분석

- » 이상치 탐지와 특이치 탐지

- » 원-클래스 SVM

- » 아이솔레이션 포레스트

- 시각화와 차원 축소

- » 주성분 분석(PCA)

- » 커널 PCA

- » 지역적 선형 임베딩

- » t-SNE

03. 머신러닝 시스템의 종류

■ 배치 학습과 온라인 학습

- 배치 학습

- 시스템이 점진적으로 학습할 수 없다.

- 온라인 학습

- 데이터를 순차적으로 한 개씩 또는 미니배치(mini-batch)라 부르는 작은 묶음 단위로 주입하여 시스템을 훈련한다.

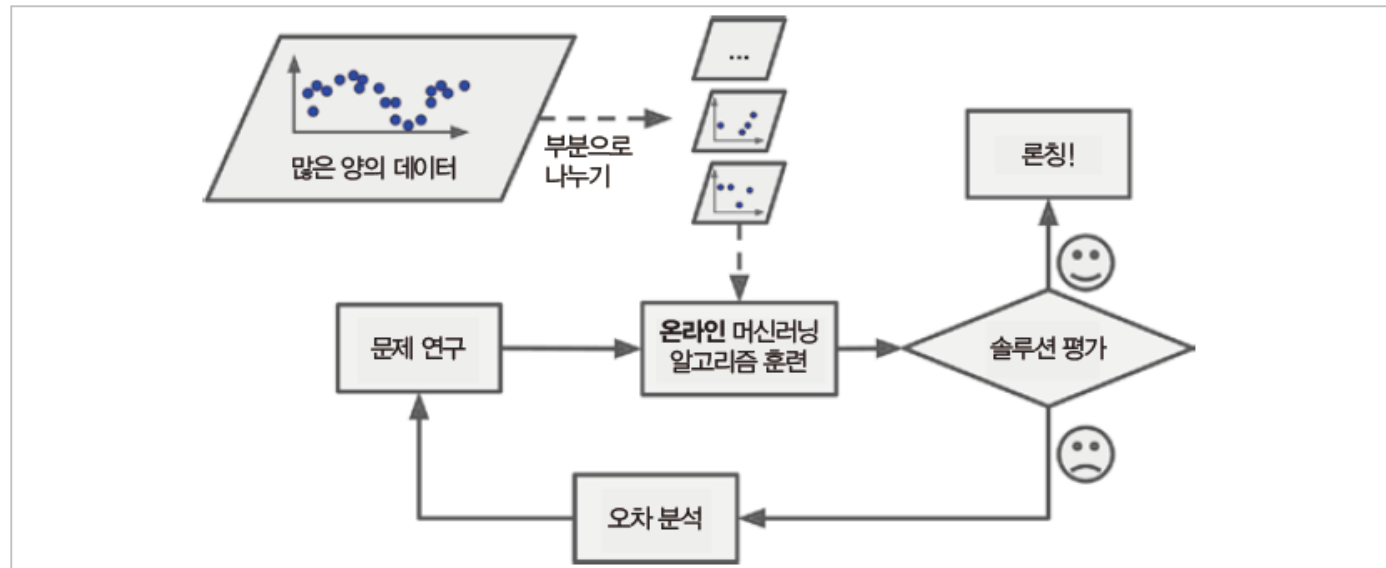
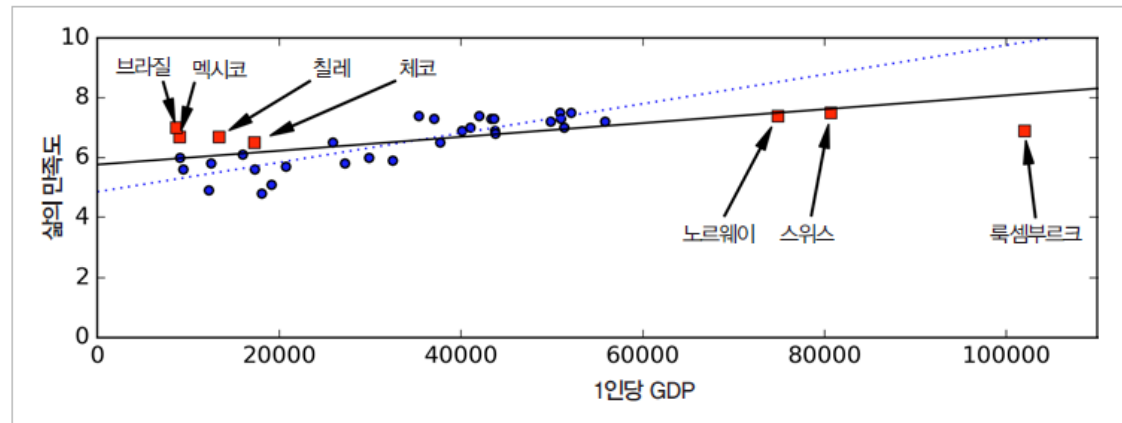


그림 1-14 온라인 학습을 사용한 대량의 데이터 처리

04. 머신러닝의 주요 도전 과제

- 머신러닝의 주요 작업은 학습 알고리즘을 선택해서 어떤 데이터에 훈련시키는 것
 - '나쁜 데이터'
 - 충분하지 않은 양의 훈련 데이터
 - 대표성 없는 훈련 데이터
 - 낮은 품질의 데이터
 - 관련 없는 특성
 - '나쁜 알고리즘'
 - 훈련 데이터 과대적합
 - 훈련 데이터 과소적합



▲ 그림 1-21 대표성이 더 큰 훈련 샘플

05. 테스트와 검증

- 훈련 세트와 테스트 세트 두 개로 나누어 검증
 - 데이터의 80%를 훈련에 20%는 테스트용으로 분리하며, 데이터셋 크기에 따라 비율이 다름
 - 훈련 세트를 사용해 모델을 훈련하고 테스트 세트를 사용해 모델을 테스트
 - 새로운 샘플에 대한 오류 비율: 일반화 오차 또는 외부 샘플 오차
 - 테스트 세트에서 모델을 평가함으로써 이 오차에 대한 추정값으로, 이전에 본 적이 없는 새로운 샘플에 모델이 얼마나 잘 작동할지 예측
 - » 훈련 오차가 낮지만(즉, 훈련 세트에서 모델의 오차가 적음) 일반화 오차가 높다면 이는 모델이 훈련 데이터에 과대적합되었다는 뜻
 - 하이퍼파라미터 튜닝과 모델 선택
 - 홀드아웃 검증(holdout validation): 간단하게 훈련 세트의 일부를 떼어내어 여러 후보 모델을 평가하고 가장 좋은 하나를 선택
 - » 검증 세트가 작을 경우, 반복적으로 교차 검증 수행
 - 데이터 불일치
 - NFL(No Free Lunch): 데이터에 관해 완벽하게 어떤 가정도 하지 않으면 한 모델을 다른 모델보다 선호할 근거가 없음 – 데이비드 월퍼트, 1996

04

인공신경망의 역사

01. 인공지능경망의 역사

I. 퍼셉트론의 등장

- 1957년 프랭크 로젠블라트가 퍼셉트론(Perceptron)이라는 신경망 모델 발표
- 인간의 두뇌 움직임을 수학적으로 구성하여 당시 굉장한 이슈가 되었던 모델
- 퍼셉트론이 발표된 후 인공지능경망에 대한 기대 폭증



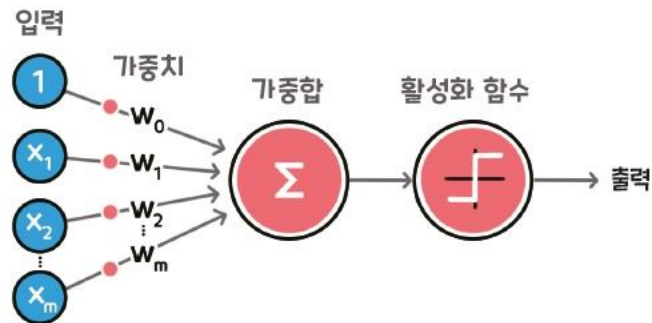
그림 9-1 퍼셉트론 관련 뉴욕타임즈 기사

01. 인공지능의 역사

I. 퍼셉트론의 등장

• 퍼셉트론(Perceptron)

- 신호(x_1, x_2)를 입력으로 받아 하나의 신호(y)를 출력하는 기계
- 신경망의 기원이 되는 알고리즘



- 입력(Input) : AND 또는 OR 연산을 위한 입력 신호
- 가중치(Weight) : 입력 신호에 부여되는 중요도로, 가중치가 크다는 것은 그 입력이 출력을 결정하는 데 큰 역할을 한다는 의미
- 가중합(Weighted Sum) : 입력값과 가중치의 곱을 모두 합한 값
- 활성화 함수(Activation Function) : 어떠한 신호를 입력받아 이를 적절하게 처리하여 출력해 주는 함수로, 가중합이 임계치(Threshold)를 넘어가면 1, 그렇지 않으면 0을 출력함
- 출력(Output) : 최종 결과(분류)

01. 인공지능의 역사

I. 퍼셉트론의 등장

- 초기 퍼셉트론을 이용한 문제해결은 AND와 OR 같은 간단한 연산이었음
- 당시 기계가 AND와 OR 연산을 스스로 풀 수 있으면 이를 조합해 어떤 문제든 풀어낼 수 있다고 생각했음
- [그림 9-3]을 보면, AND와 OR 연산의 경우 퍼셉트론을 통한 선형 분리가 가능한 것을 확인 가능

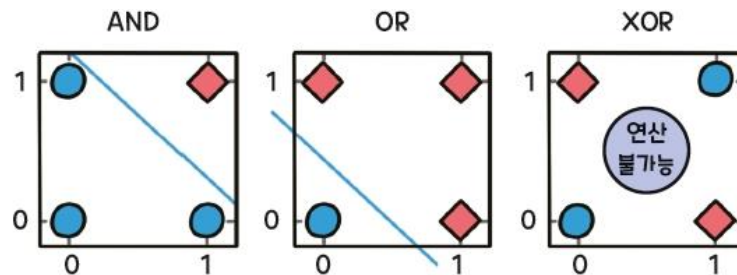


그림 9-3 퍼셉트론의 한계

선형 분리(Linearly Separable)는 AND, OR 연산과 같이 직선 하나로 데이터들이 두 개의 그룹으로 나누어지는 것을 의미합니다.

01. 인공신경망의 역사

I. 퍼셉트론의 등장

- 1969년 마빈 민스키와 시모어 페퍼트가 현재의 퍼셉트론으로는 XOR 연산이 절대 불가능하다는 가설을 수학적으로 증명
- 퍼셉트론을 여러 개 쌓아 올린 다층 퍼셉트론(MLP)을 통해 XOR 연산에 대한 문제는 해결될 수 있지만, '각각의 가중치(Weight)와 편향(Bias)을 학습시킬 방법이 없다'라는 결론을 내놓으면서 퍼셉트론을 이용한 학습의 한계를 언급
- 이후 인공신경망 연구는 침체기를 맞이하게 됨

01. 인공지능의 역사

II. 역전파의 고안

- 역전파(Backpropagation)

- 가중치와 편향을 학습하기 위한 방법으로 고안
- 신경망의 오차(예측값과 실제값의 차이)를 출력층에서부터 입력층으로 거꾸로 전파시켜 각 층(Layer)의 가중치와 편향을 업데이트함

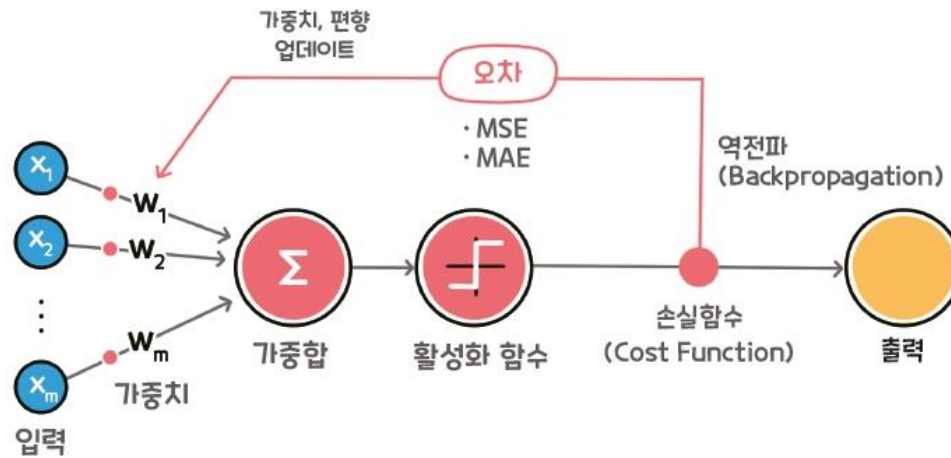


그림 9-4 역전파 과정

01. 인공신경망의 역사

II. 역전파의 고안

- 1974년 폴 워보스가 박사과정 논문에서 역전파를 처음 제안
- 1986년, 제프리 힌튼이 또 다른 역전파 방법을 고안
 - » 제프리 힌튼이 제안한 역전파가 XOR 문제뿐만 아니라 더 복잡한 문제도 해결할 수 있음이 증명되자 인공신경망은 다시 사람들의 관심을 끌기 시작
- 이후 인공신경망 연구는 큰 진전이 있었지만, 다음 이유로 긴 침체기 돌입
 - » 신경망의 깊이가 깊어질수록 원하는 결과를 얻을 수 없음
 - » 신경망 학습에 최적화된 하이퍼파라미터에 대한 이론적인 근거가 없음

01. 인공신경망의 역사

II. 역전파의 고안

- 신경망이 깊어질수록 학습력이 좋아져야 하는데 기대하는 결과가 나오지 않는 일이 지속적으로 발생
- 역전파를 수행할 때 출력층에서 멀리 떨어진 층에서는 기울기가 급속히 작아지는 기울기 소멸 문제(Vanishing Gradient)와 기울기가 너무 커지는 기울기 발산(Exploding Gradient) 문제가 발생해 학습력이 떨어진 것임

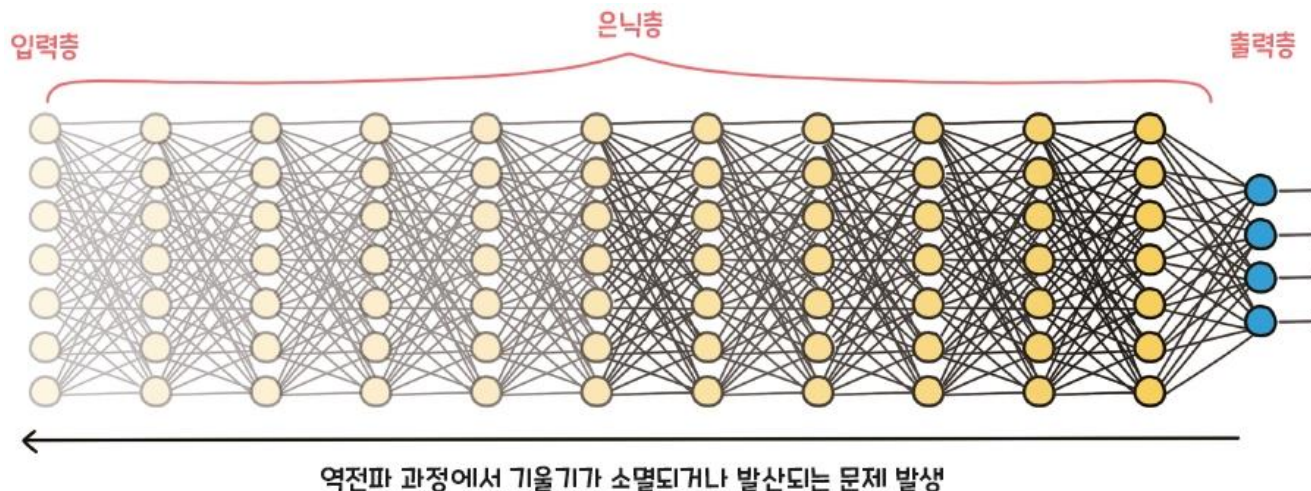


그림 9-5 기울기 소멸, 발산 문제

01. 인공지능경망의 역사

II. 역전파의 고안

하나 더 알기 오차 계산

회귀모델에서 오차를 계산하는 방법으로는 MSE와 MAE가 많이 사용

1. MSE(Mean Squared Error)

- 의미 : 예측값과 실제값의 차이인 오차들의 제곱에 대한 평균
- 수식 :
$$\frac{\sum(\text{예측값} - \text{실제값})^2}{n}$$
- 장점 : 직관적이고 단순함
- 단점 : ① 오류를 제곱하기 때문에 1 미만의 오류는 더 작아지고 그 이상의 오류는 더 커지는 값의 왜곡이 발생
② 스케일에 의존적

01. 인공지능경망의 역사

II. 역전파의 고안

하나 더 알기 오차 계산

2. MAE(Mean Absolute Error)

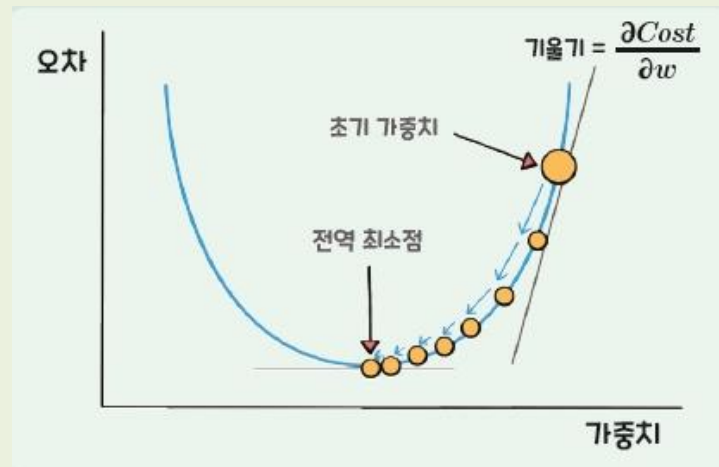
- 의미 : 예측값과 실제값의 차이인 오차들의 절댓값에 대한 평균
- 수식 :
$$\frac{\sum(|\text{예측값} - \text{실제값}|)}{n}$$
- 장점 : 매우 직관적인 지표
- 단점 : 스케일에 의존적

01. 인공신경망의 역사

II. 역전파의 고안

하나 더 알기 기울기 계산

- 경사 하강법은 오류가 작아지는 방향으로 가중치(w)값을 보정하기 위해 사용
- 경사 하강법은 최초 가중치에서부터 미분을 적용한 뒤 이 미분값이 계속 감소하는 방향으로 순차적으로 가중치값을 업데이트한 후 업데이트가 끝나는 전역 최소점에서의 w 를 반환



01. 인공지능경망의 역사

II. 역전파의 고안

하나 더 알기 기울기 계산

- 손실함수에는 지역 최소점(Local Minimum)과 전역 최소점(Global Minimum)이 있는데, 지역 최소점은 함수 일부 구간의 최솟값이며, 전역 최소점은 전체 구간에서의 최솟값임



01. 인공지능경망의 역사

III. 딥(Deep)의 출현

- 2006년, 역전파를 고안했었던 제프리 힌튼이 가중치의 초기값을 제대로 설정하면 깊이가 깊은 신경망도 학습이 가능하다는 연구를 선보임
- 2007년, 벤지오(Bengio) 팀이 오토인코더(Autoencoder)를 사용한 좀 더 간단한 사전훈련 방법을 제안
- 이때부터 인공지능경망(Neural) 대신 '딥(Deep)'이라는 용어를 사용하기 시작

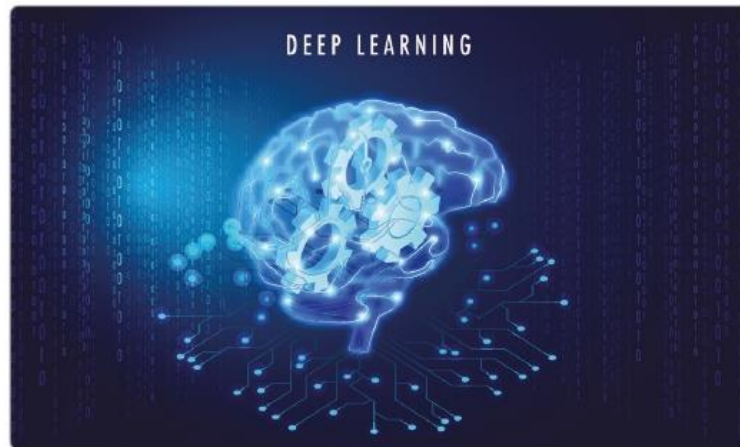


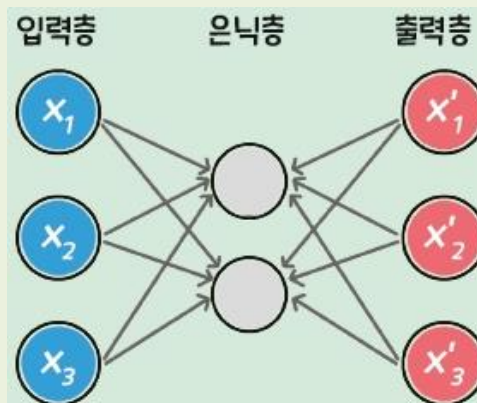
그림 9-6 딥러닝의 출현

01. 인공지능망의 역사

III. 딥(Deep)의 출현

하나 더 알기 오차 계산

- **오토인코더(Autoencoder)** : 입력층과 출력층이 동일한 네트워크에 데이터를 입력하여 비지도학습을 하는 것
- 인코더를 통해 입력 데이터에 대한 특징을 추출하고, 디코더를 통해 원본 데이터를 재구성하는 학습
- 가중치의 좋은 초기값을 얻는 목적으로 이용됨



05

딥러닝의 개요

01. 딥러닝의 개요

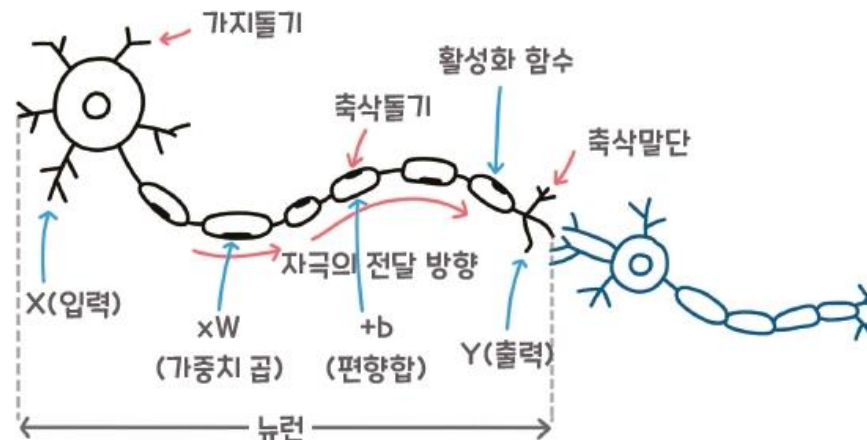
I. 인공신경망의 개념

- 딥러닝의 기원은 인공신경망
- 인간의 뇌에는 수많은 뉴런이 존재하고 그 뉴런들은 시냅스로 서로 연결되어 있는데, 이를 신경망(Neural Network)이라고 부름
- 인공신경망은 사람의 신경망 구조에서 착안해 만들어졌기 때문에 뉴런들의 연결, 즉 신경망을 인공적으로 흉내 낸 것

01. 딥러닝의 개요

I. 인공신경망의 개념

- 인간의 신경망을 이루는 기본 동작



- ① 가지돌기에서 신호를 받아들임
- ② 신호가 축삭돌기를 지나 축삭말단으로 전달됨
- ③ 축삭돌기를 지나는 동안 신호가 약해져서 축삭말단까지 전달되지 않거나 강하게 전달되기도 함
- ④ 축삭말단까지 전달된 신호는 다음 뉴런의 가지돌기로 전달됨
- ⑤ 수억 개의 뉴런 조합을 통해 손가락을 움직이거나 물체를 판별하는 등 다양한 조작과 판단 수행 가능

그림 9-7 인간의 신경망 동작 과정

01. 딥러닝의 개요

I. 인공신경망의 개념

- 인공신경망(ANN, Artificial Neural Network)

- 인간의 신경망을 흉내 낸 머신러닝 기법

- » 가지돌기 : 인공신경망의 입력층

- » 축삭말단 : 인공신경망의 출력층

- » 축삭말단에 이르기까지의 신호의 크기 : 인공신경망에서 가중치

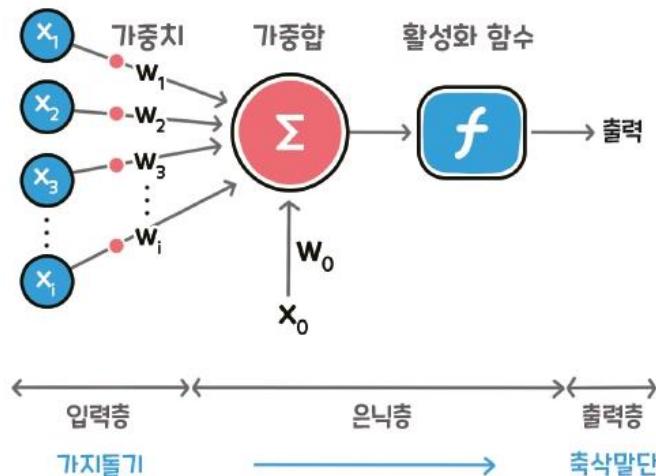


그림 9-8 인공신경망

인공신경망	인간의 뇌
	뉴런
	전기·화학적 신호가 흐르는 방향
가중치	신호의 크기

01. 딥러닝의 개요

II. 딥러닝의 개념

- 딥러닝(Deep Learning)

- 여러 층(특히 은닉층이 여러 개)을 가진 인공신경망을 사용하여 머신러닝 학습을 수행하는 것
- 심층학습이라고도 부름

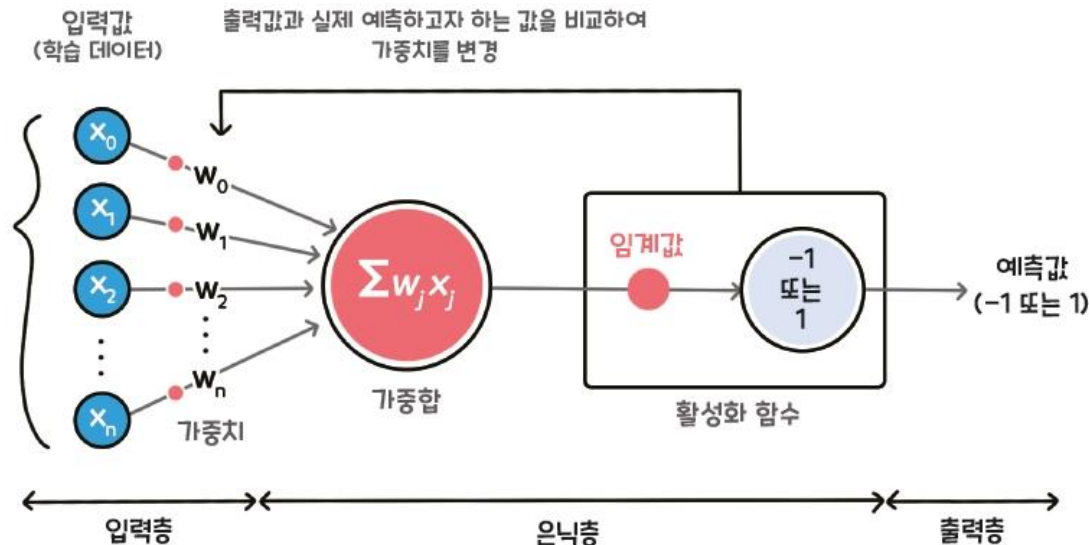


그림 9-9 인공신경망의 구조

01. 딥러닝의 개요

II. 딥러닝의 개념

표 9-1 딥러닝 용어 설명

구분	설명
입력층	학습하고자 하는 데이터를 입력받음
은닉층	모든 입력 노드로부터 입력값을 받아 가중합을 계산하고, 이 값을 활성화 함수에 적용하여 출력층에 전달
출력층	최종 결과 출력
가중치	입력 신호가 출력에 미치는 영향을 조절하는 매개변수로, 입력값의 중요도를 결정
편향	가중합에 더하는 상수로, 하나의 뉴런에서 활성화 함수를 거쳐 최종적으로 출력되는 값을 조절

01. 딥러닝의 개요

II. 딥러닝의 개념

1) 가중합

- 가중합(Weighted Sum)

- 입력값과 가중치를 곱한 뒤 편향(Bias)을 더한 값
- [그림 9-10]을 예로 들어 하나의 노드에 대한 가중합 구하기

» 가중합(y) = $(x_0 \times w_0) + (x_1 \times w_1) + (x_2 \times w_2) + b$

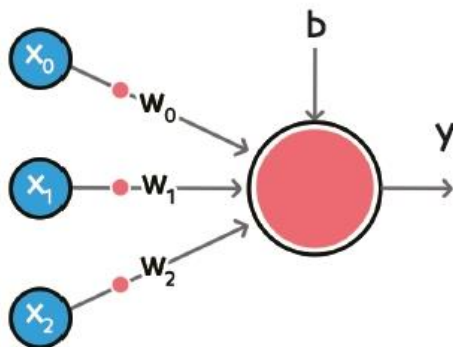


그림 9-10 하나의 노드에 대한 가중합

- 원: 노드
- 화살표: 신호의 흐름
- x_n : 외부에서 들어오는 신호(Input)
- w_n : 신호를 연결하는 가중치(Weight)
- b : 출력값을 조절하는 편향(Bias)
- y : 외부로 나가는 신호(Output)

01. 딥러닝의 개요

II. 딥러닝의 개념

2) 활성화 함수

- **활성화 함수(Activation Function)**
 - 입력 신호가 출력 결과에 미치는 영향도를 조절하는 매개변수
- **활성화 함수를 사용하는 이유**
 - 출력값을 0~1 사이의 값으로 반환해야 하는 경우에 사용
 - 비선형을 위해 사용

01. 딥러닝의 개요

II. 딥러닝의 개념

2) 활성화 함수

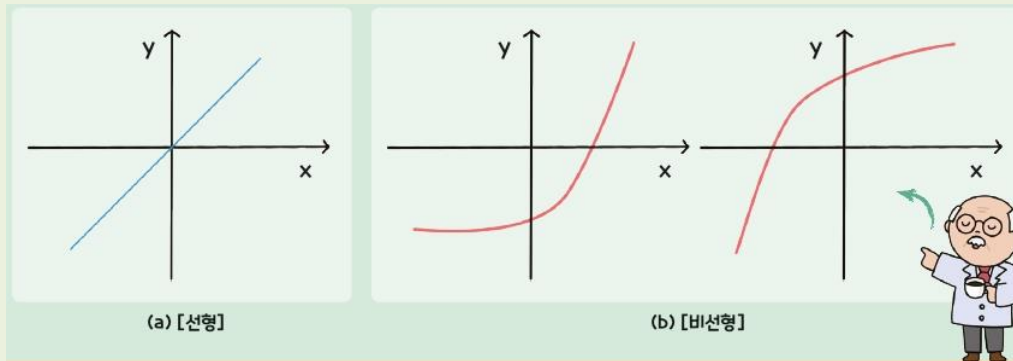
- 선형함수인 $[f(x) = W \times x]$ 식이 있다고 가정
 - 이때 은닉층을 2개 사용한다면, 식은 $[f(f(f(x))) = W \times W \times W \times x = kx]$ 로 단순하게 변형됨
 - 은닉층을 20개 사용한다고 해도 동일한 결과가 나옴
 - 즉, 선형함수로는 은닉층을 여러 개 추가해도 은닉층을 1개 추가한 것과 차이가 없기 때문에 정확한 결과를 위해선 비선형 함수, 즉 활성화 함수가 필요함

01. 딥러닝의 개요

II. 딥러닝의 개념

하나 더 알기 선형과 비선형

- **선형(Linear)** : 그림 (a)처럼 그래프가 직선(line)의 모양을 취하는 것으로, 1차 함수의 그래프 모양을 유지하는 것
- **비선형(Non-linear)** : 직선이 아닌 것으로, 그림 (b)와 같이 그래프에서 곡선 모양을 취하는 것을 비선형이라고 함



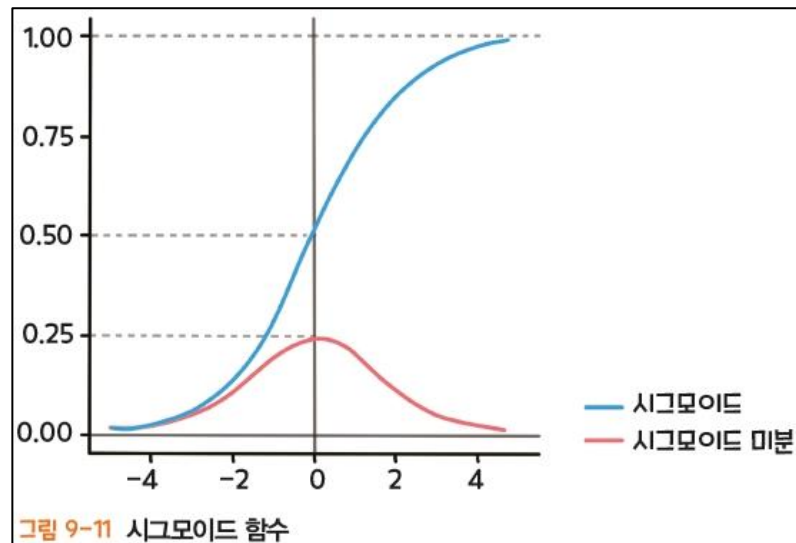
01. 딥러닝의 개요

II. 딥러닝의 개념

2) 활성화 함수

■ 시그모이드(Sigmoid) 함수

- x 값의 변화에 따라 0에서 1까지의 값을 출력하는 S자형 함수
- 로지스틱 함수라고도 부름



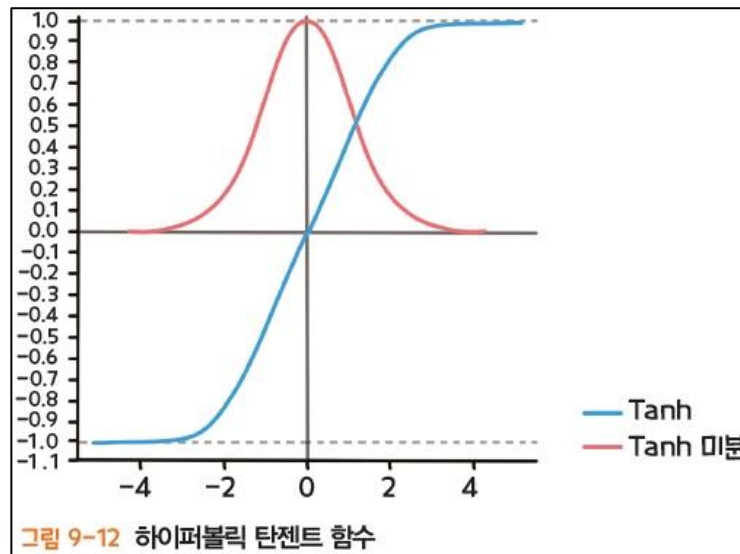
01. 딥러닝의 개요

II. 딥러닝의 개념

2) 활성화 함수

▪ 하이퍼볼릭 탄젠트(Hyperbolic Tangent) 함수

- 시그모이드 함수와 유사
- 하지만 $-1 \sim 1$ 의 값을 가지면서 데이터의 평균이 0이라는 점이 다름



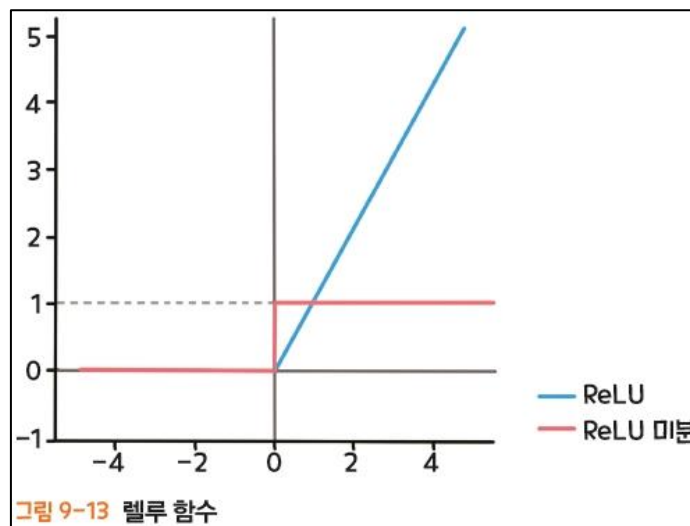
01. 딥러닝의 개요

II. 딥러닝의 개념

2) 활성화 함수

- 렐루(ReLU) 함수

- x 가 음의 값을 가지면 0을 출력하고, 양의 값을 가지면 x 를 그대로 출력
- 함수 형태도 $\max(0, x)$ 로 계산이 간단하여 학습 속도가 매우 빠름



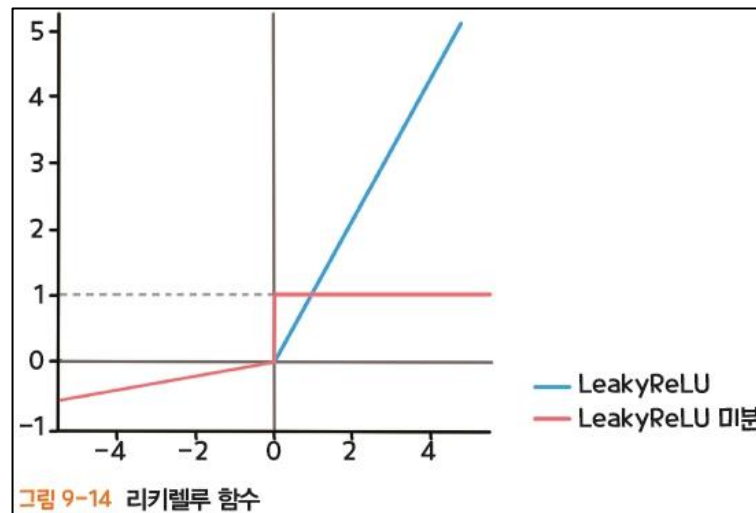
01. 딥러닝의 개요

II. 딥러닝의 개념

2) 활성화 함수

▪ 리키렐루(LeakyReLU) 함수

- 렐루 함수와 유사
- 차이점은 가중치 곱의 합이 0보다 작을 때의 값도 고려한다는 점
- 이 함수는 렐루 함수의 '죽은 렐루' 현상을 보완



01. 딥러닝의 개요

II. 딥러닝의 개념

2) 활성화 함수

■ 소프트맥스(SoftMax) 함수

- 입력받은 값이 0~1 사이의 값으로 출력되도록 정규화하여 출력의 총 합이 1이 되는 특성을 갖는 함수
- 딥러닝에서 출력 노드의 활성화 함수로 많이 사용



그림 9-15 소프트맥스 함수

01. 딥러닝의 개요

III. 딥러닝 학습

1) 순전파

- 순전파(Forward Propagation)

- 입력층에서 출력층 방향으로 연산이 진행되면서 최종 출력값(예측값)이 도출되는 과정

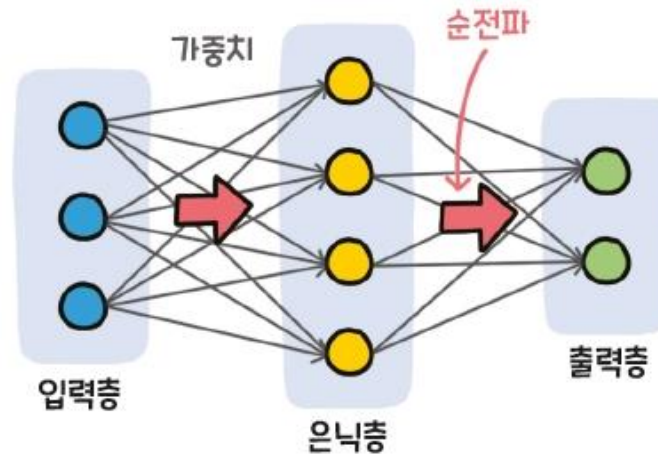


그림 9-16 순전파

01. 딥러닝의 개요

III. 딥러닝 학습

2) 손실함수

- 손실함수(Loss Function)

- 예측값과 실제값의 차이를 구하는 함수
 - » 두 값의 차이가 클 수록 손실함수의 값은 커짐
 - » 두 값의 차이가 작을수록 손실함수의 값도 작아짐

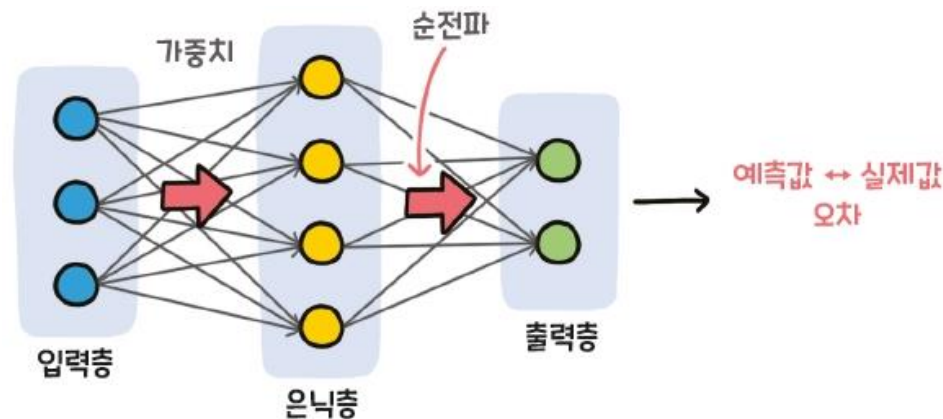


그림 9-17 손실함수

01. 딥러닝의 개요

III. 딥러닝 학습

3) 옵티마이저

- 옵티마이저(Optimizer)

- 딥러닝에서 학습 속도를 빠르고 안정적이게 만드는 것
- 전체 데이터(전체 배치)를 가지고 매개변수(가중치, 편향)의 값을 조정할 수 있고, 정해진 양의 데이터(미니 배치)만 가지고 매개변수의 값을 조정할 수 있음
 - » 배치(Batch) : 가중치 등 매개변수 값을 조정하기 위해 사용하는 데이터의 양

01. 딥러닝의 개요

III. 딥러닝 학습

3) 옵티마이저

- 일반적으로 전체 데이터를 한번에 학습하게 되면 계산량이 많아질 뿐만 아니라 속도도 느려질 수 있기 때문에 미니 배치를 이용한 학습을 함

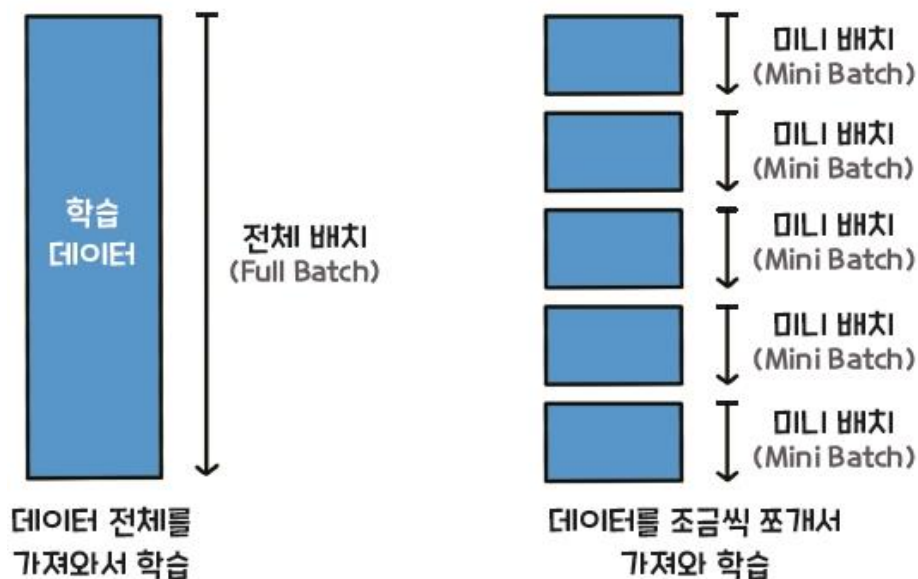


그림 9-18 옵티마이저

01. 딥러닝의 개요

III. 딥러닝 학습

4) 역전파

- 역전파(Backpropagation)

- 오차예측값과 실제값의 차이를 역방향으로 전파시키면서(출력층 → 은닉층 → 입력층) 가중치를 업데이트하는 것

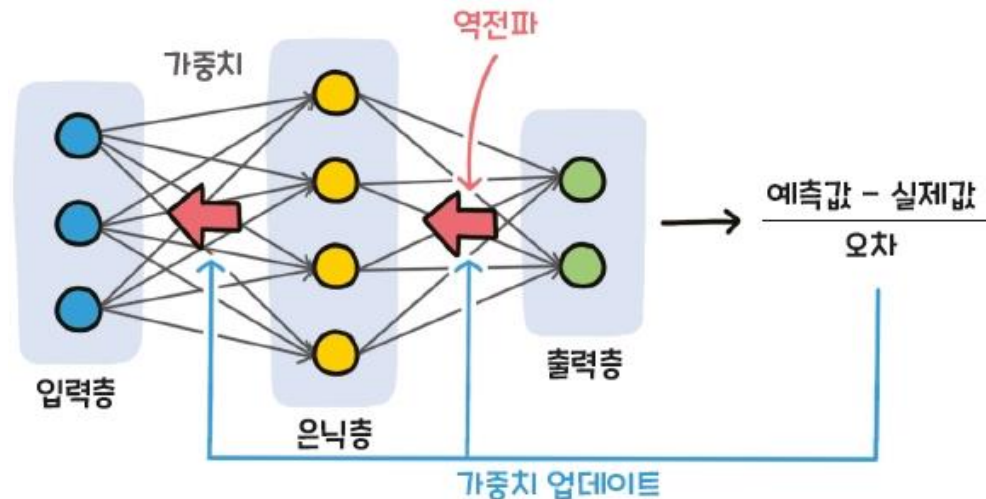


그림 9-19 역전파

01. 딥러닝의 개요

III. 딥러닝 학습

4) 역전파

- 역전파는 경사 하강법을 이용해 오차를 줄이는 방향으로 가중치를 수정함
- 가중치를 수정할 때에는 순전파에서 계산한 결과 $y = f(x)$ 의 편미분값을 오차에 곱해 출력층 → 은닉층 → 입력층 순서로 전달함



그림 9-20 가중치 수정 방법

Thank You !