인공지능의 이해
머신러닝/딥러닝의 이해

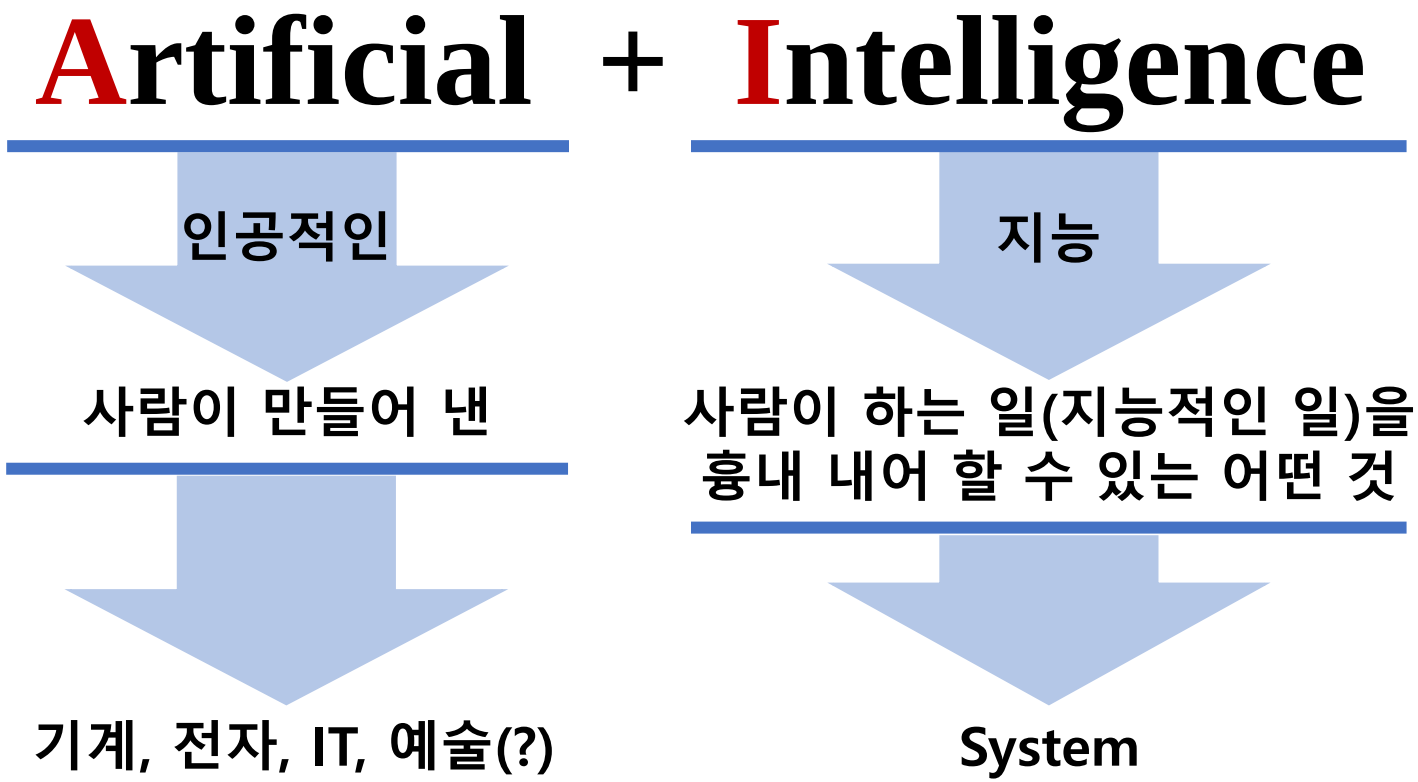
Yang Seokhwan

• AI 개요

- 인공지능의 기본 개념과 용어
- 인공지능의 역사 및 배경
- 머신러닝/인공신경망/딥러닝 개요
- 딥러닝 주요 알고리즘(모델)

AI 개요

- 인공지능(Artificial Intelligence, AI)이란 무엇인가?



AI 개요

- 인공지능이란
  - 다양한 기술을 이용하여
  - 사람이 하는 일을 흉내 내어 처리할 수 있는 시스템
- 다양한 기술에는
  - 기계, 전자, 컴퓨터 등 공학적인 기술과
  - 예술로 표현할 수 있는 창의성이 포함됨 (?)

정의(?)가 애매한 것 같은데 왜 그렇죠?
"지능"의 정의가 아직 명확하지 않음
아직 인간의 지능에 대해서는 밝혀지지 않은
영역이 많아서 명확하게 정의할 수 없음

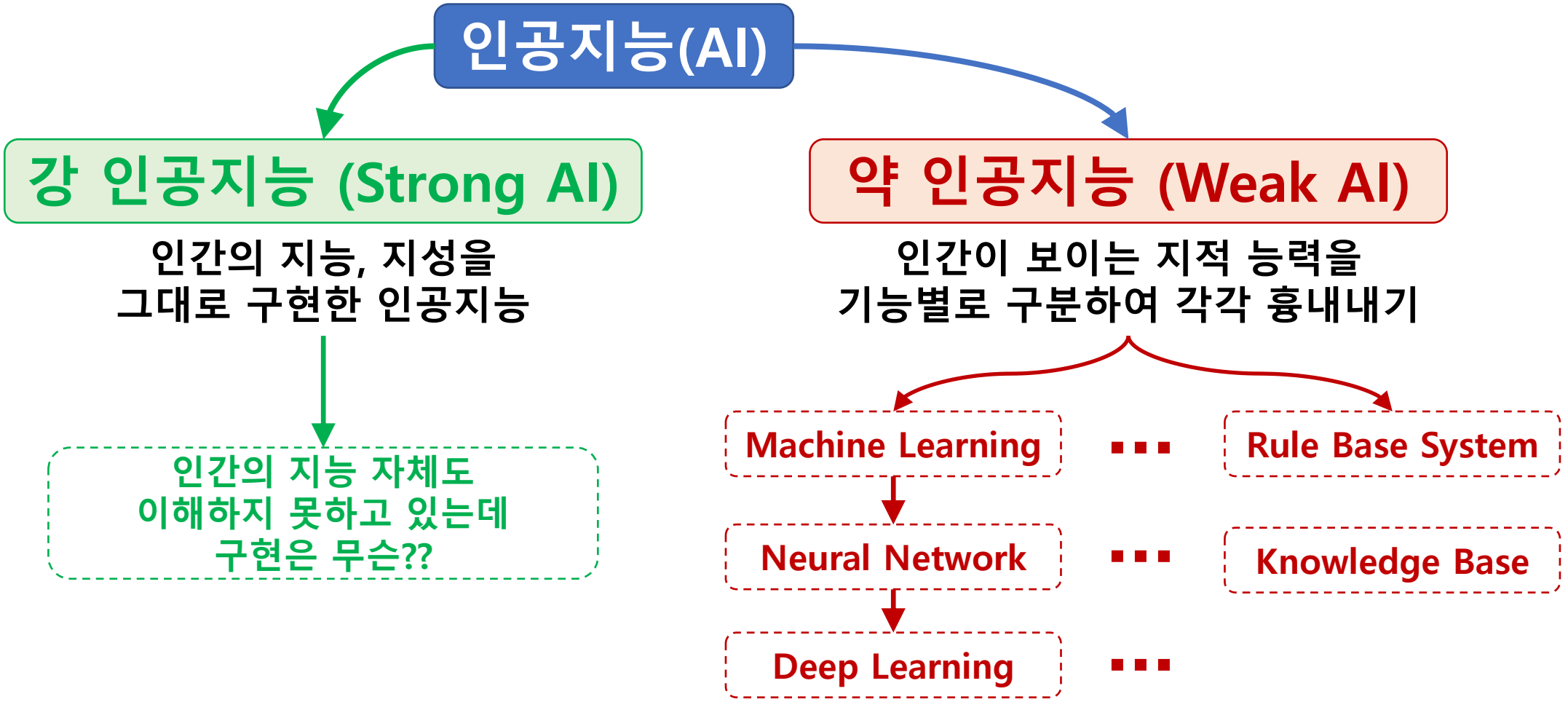
AI 개요

- **인간의 지능에 대하여 명확하게 밝혀지거나 정의되지 않음에 따라**
  - **인공지능의 기술 구현 방향은 인간을 흉내내는 것으로 귀결됨**
    - 시각: Computer Vision → 영상 인식, 분류, 영역구분 등
    - 청각 / 발성: Audio 처리기술 → AI Speaker 등
    - 촉각, 후각, 미각: 센서 기술 연구 수준에서 머물고 있음. 최근 성과가 조금씩 나오는 중
    - 사고: 현재 구현 불가능 → 데이터 처리, 의사결정, 언어처리 등으로 우회하여 구현

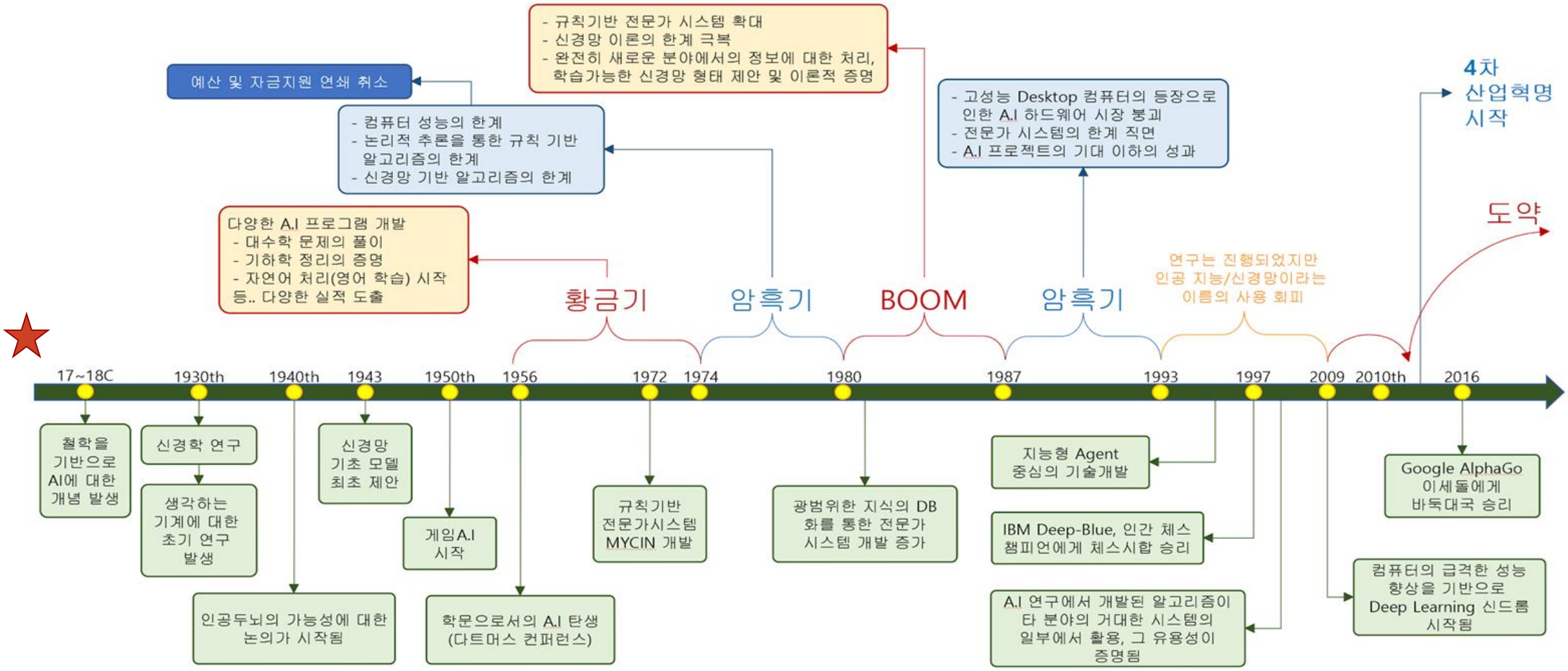
→ 주변에서 쉽게 볼 수 있는 인공지능의 연구/산업 분야

AI 개요

- 인공지능의 구분



AI 개발의 역사


- 규칙기반 전문가 시스템 확대
- 신경망 이론의 한계 극복
- 완전히 새로운 분야에서의 정보에 대한 처리, 학습가능한 신경망 형태 제안 및 이론적 증명
예산 및 자금지원 연쇄 취소
- 컴퓨터 성능의 한계
- 논리적 추론을 통한 규칙 기반 알고리즘의 한계
- 신경망 기반 알고리즘의 한계
- 고성능 Desktop 컴퓨터의 등장으로 인한 A.I 하드웨어 시장 붕괴
- 전문가 시스템의 한계 직면
- A.I 프로젝트의 기대 이하의 성과
4차 산업혁명 시작
다양한 A.I 프로그램 개발
- 대수학 문제의 풀이
- 기하학 정리의 증명
- 자연어 처리(영어 학습) 시작
등.. 다양한 실적 도출
도약
황금기
암흑기
BOOM
암흑기
연구는 진행되었지만 인공 지능/신경망이라는 이름의 사용 회피
17~18C
1930th
1940th
1943
1950th
1956
1972
1974
1980
1987
1993
1997
2009
2010th
2016
철학을 기반으로 AI에 대한 개념 발생
신경학 연구
생각하는 기계에 대한 초기 연구 발생
인공두뇌의 가능성에 대한 논의가 시작됨
신경망 기초 모델 최초 제안
게임A.I 시작
학문으로서의 A.I 탄생 (다트머스 컨퍼런스)
규칙기반 전문가시스템 MYCIN 개발
광범위한 지식의 DB화를 통한 전문가 시스템 개발 증가
지능형 Agent 중심의 기술개발
IBM Deep-Blue, 인간 체스 챔피언에게 체스시합 승리
A.I 연구에서 개발된 알고리즘이 타 분야의 거대한 시스템의 일부에서 활용, 그 유용성이 증명됨
Google AlphaGo 이세돌에게 바둑대국 승리
컴퓨터의 급격한 성능 향상을 기반으로 Deep Learning 신드롬 시작됨

AI 개발의 역사

- **일반적으로 알려지지 않은, 오래된 역사**
  - **기원전 1세기**
    - AI의 대전제: 인간의 사고를 기계로 재현할 수 있어야 한다 → 기계적 추론 / 형식추론
    - 형식추론: 그리스, 중국, 인도 등 세계 각지의 철학자들에 의해 기원전 1세기부터 정립
  - **13~14세기**
    - 스페인 마요르카 섬 출신 철학자 라몬 룰
      - 다수의 논리 기계 개발
      - 자신의 기계들이 간단하고 부인 불가능한 진실을 기계적이고 논리적인 방식으로 조합해 세상의 모든 지식을 만들 수 있다고 주장 → 17세기 독일 철학자 라이프니츠에 영향

AI 개발의 역사

- **17~18세기**
  - 라이프니츠, 홉스, 데카르트 등의 철학자
    - 인간의 모든 추론을 대수, 기하 등 수학적이고 기계적인 방식으로 체계화하여 번역할 수 있는지 연구
    - 20세기까지 연구가 이어짐
      - 부울, 프레게, 러셀, 화이트헤드, 힐버트 등

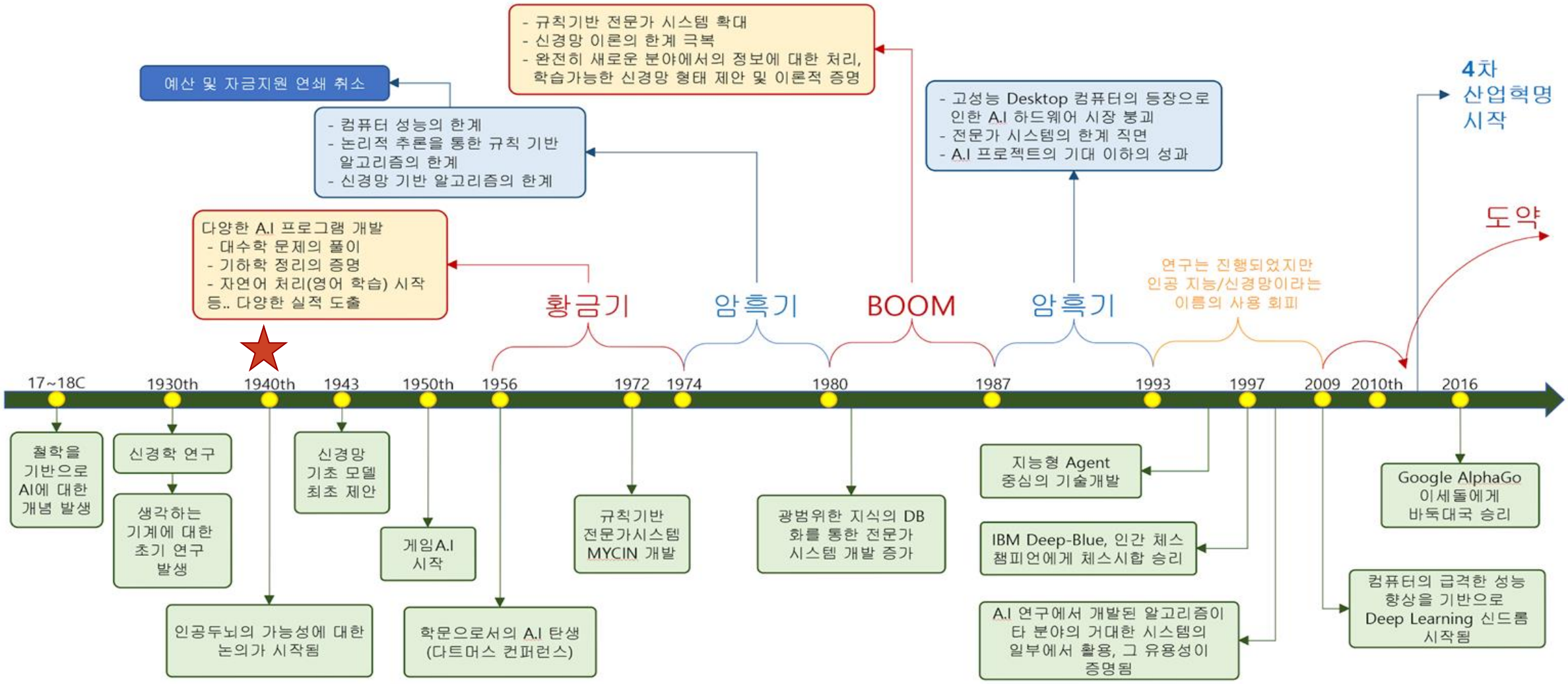
AI 개발의 역사

- **20세기 초, 중반**
  - 괴델(불완전성 정리), 앨런 튜링(튜링 기계), 알론조 처치(람다 대수)
  - 연구의 결론:
    - 인간의 추론을 수학적 추론으로 바꾸는 데는 한계가 있음

      → 수학적 추론으로 바꿀 수 없는 추론 존재
    - 가능한 범위 안의 모든 수학적 추론은 기계의 언어로 번역 가능

      → AI의 가능성에 대한 열쇠가 됨
- **1930년대**
  - 생각하는 기계에 대한 초기 연구가 시작됨
  - 신경학 연구의 활성화

AI 개발의 역사

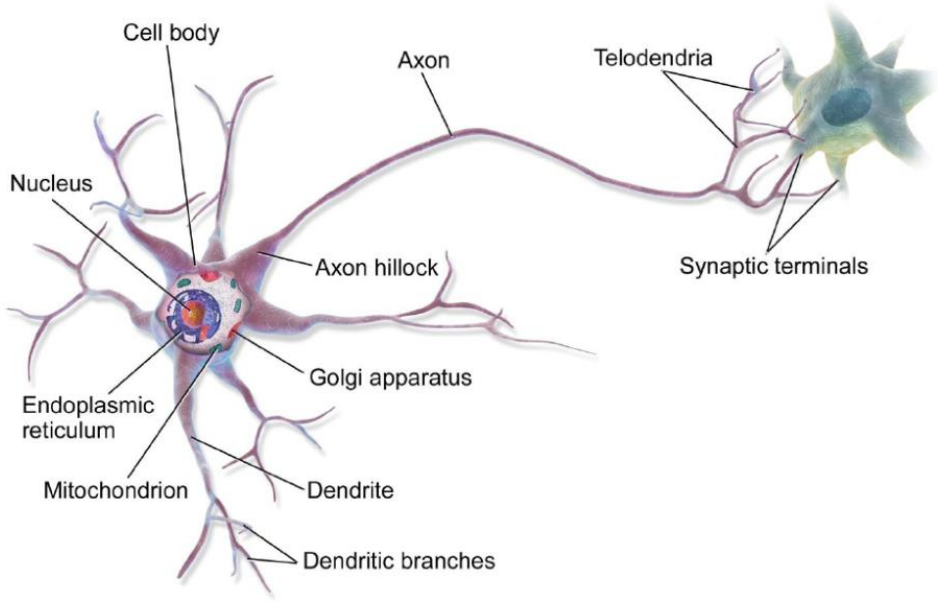

- 규칙기반 전문가 시스템 확대
- 신경망 이론의 한계 극복
- 완전히 새로운 분야에서의 정보에 대한 처리, 학습가능한 신경망 형태 제안 및 이론적 증명
예산 및 자금지원 연쇄 취소
- 컴퓨터 성능의 한계
- 논리적 추론을 통한 규칙 기반 알고리즘의 한계
- 신경망 기반 알고리즘의 한계
- 고성능 Desktop 컴퓨터의 등장으로 인한 A.I 하드웨어 시장 붕괴
- 전문가 시스템의 한계 직면
- A.I 프로젝트의 기대 이하의 성과
4차 산업혁명 시작
도약
다양한 A.I 프로그램 개발
- 대수학 문제의 풀이
- 기하학 정리의 증명
- 자연어 처리(영어 학습) 시작
등.. 다양한 실적 도출
황금기
암흑기
BOOM
암흑기
연구는 진행되었지만 인공 지능/신경망이라는 이름의 사용 회피
17~18C
1930th
1940th
1943
1950th
1956
1972
1974
1980
1987
1993
1997
2009
2010th
2016
철학을 기반으로 AI에 대한 개념 발생
신경학 연구
생각하는 기계에 대한 초기 연구 발생
인공두뇌의 가능성에 대한 논의가 시작됨
신경망 기초 모델 최초 제안
게임A.I 시작
학문으로서의 A.I 탄생 (다트머스 컨퍼런스)
규칙기반 전문가시스템 MYCIN 개발
광범위한 지식의 DB화를 통한 전문가 시스템 개발 증가
지능형 Agent 중심의 기술개발
IBM Deep-Blue, 인간 체스 챔피언에게 체스시합 승리
A.I 연구에서 개발된 알고리즘이 타 분야의 거대한 시스템의 일부에서 활용, 그 유용성이 증명됨
Google AlphaGo 이세돌에게 바둑대국 승리
컴퓨터의 급격한 성능 향상을 기반으로 Deep Learning 신드롬 시작됨

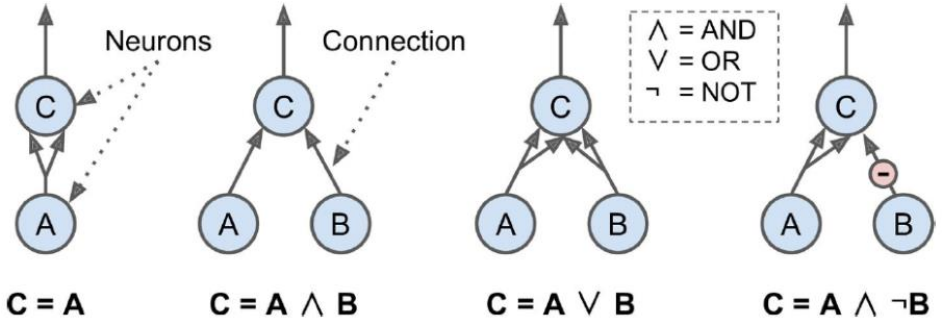
AI 개발의 역사

- **잘 알려진 AI 개발의 역사**
  - **1940년대**
    - 인공 두뇌의 가능성에 대한 논의가 시작됨
    - **1943년: 최초의 신경망 기초 모델 제안**

AI 개발의 역사

- **1943년**
  - 신경 생리학자 워런 맥컬록, 수학자 월터 피츠: 최초의 신경망 기초 모델 제안
  - 명제 논리를 사용하여 동물 뇌의 생물학적 뉴런이 복잡한 계산을 위해 어떻게 상호작용하는지에 대한 간단한 계산 모델 제공 → 최초의 인공 신경망 구조





AI 개발의 역사

- **1950년: 앨런 튜링, 튜링 테스트 창안**
  - **기계의 응답을 인간의 응답과 구분할 수 없다면 그 기계는 지능을 가지고 있다고 보아야 한다.**
- **1956년: 미국 다트머스 대학교 컨퍼런스**
  - 존 매커시: **인공지능(Artificial Intelligence)이라는 용어 첫 사용**
  - **학문으로서의 AI 연구가 본격적으로 시작됨**
- **1957년: 프랭크 로젠블럿, 퍼셉트론 제안**
  - 2중 컴퓨터 학습 네트워크에 기반한 패턴인식이 가능한 **초기 인공신경망 퍼셉트론** 제안

AI 개발의 역사

- **인공지능 연구 방향의 분류를 기준으로 봤을때**
  - 기호주의 학파: 역 연역법 → 규칙기반 전문가 시스템 중심으로 발전
  - 연결주의 학파: 신경망 → 신경망 알고리즘을 통한 학습 중심으로 발전
  - 진화주의 학파: 유전자 알고리즘
  - 베이즈 학파: 베이즈 추론 기반의 통계
  - 유추주의 학파: 서포트 벡터 머신

이상의 다섯 가지 분류는 뒤에서 살펴봄

인공지능 연구의 황금기

- 대수학 문제의 풀이
- 기하학 정리의 증명
- 자연어처리 시작
- 다양한 AI 프로그램 개발
- 규칙기반 전문가 시스템 MYCIN 개발

등.. 다양한 실적 도출

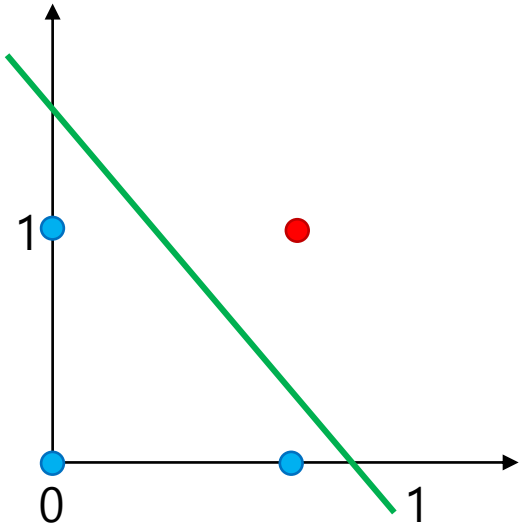
AI 개발의 역사

- **1969년: 마빈 민스키, 시모어 페퍼트 - 퍼셉트론즈**
  - 로젠블럿의 퍼셉트론 이후 신경망이 무엇이든지 할 수 있는 듯이 보였던 시절
  - 수백개의 알고리즘이 제시되었고, 학습 머신에 대한 관심과 열기가 대단하였으나
  - MIT의 마빈 민스키, 시모어 페퍼트의 "퍼셉트론즈(Perceptrons)" 논문 출간
    - **퍼셉트론은 단순한 XOR 연산조차 할 수 없다**
    - 이후 신경망에 대한 연구는 급격한 침체기에 빠지며
    - **인공지능의 겨울**로 불리는 암흑기를 이끌어냄

- 그러나 엄밀하게 말하면 인공지능(AI)의 겨울은 없었음. 단지 신경망 연구의 겨울이 있었을 뿐...
- 마빈 민스키 교수가 주축이 된 "기호주의 학파"의 인공지능 연구는 호황을 맞이함
  (인공지능이라는 이름을 많이 쓰지 않게 되면서 인공지능의 겨울로 알려짐)

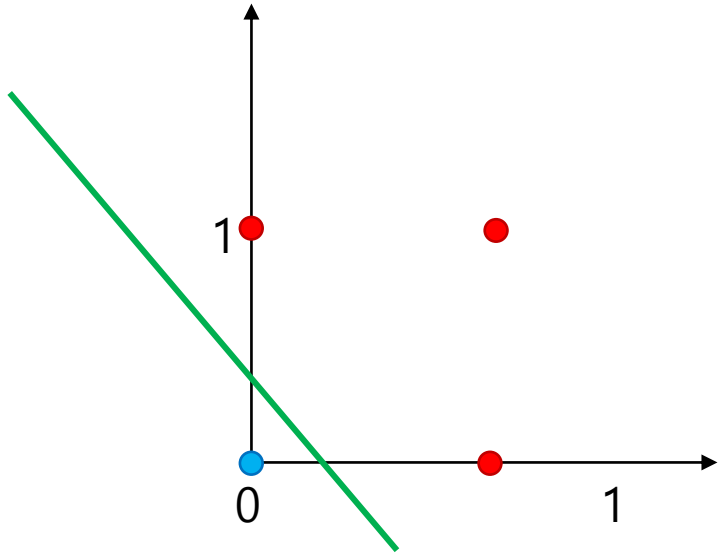
AI 개발의 역사

- 퍼셉트론은 단순한 XOR 연산조차 할 수 없다



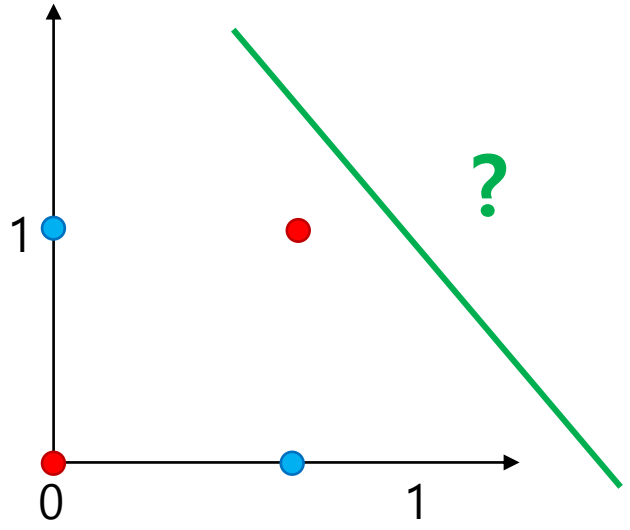
AND 연산

0 & 0 = 0
0 & 1 = 0
1 & 0 = 0
1 & 1 = 1



OR 연산

0 | 0 = 0
0 | 1 = 1
1 | 0 = 1
1 | 1 = 1



XOR 연산

0 ^ 0 = 0
0 ^ 1 = 1
1 ^ 0 = 1
1 ^ 1 = 0

AI 개발의 역사

- 인공지능(신경망) 연구 침체의 실질적인 원인
  - 컴퓨터 성능의 한계
  - 논리적 추론을 통한 규칙기반 알고리즘의 한계

    → 불똥은 신경망으로..

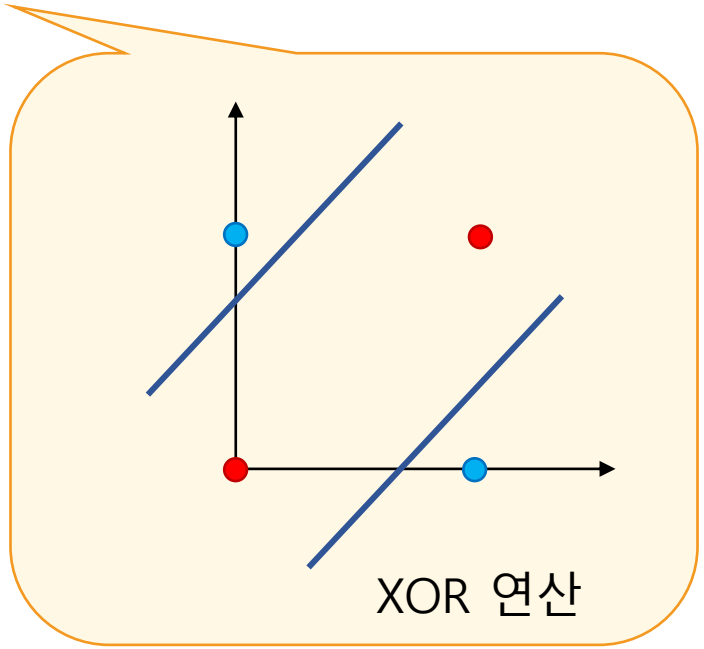
    → 신경망 기반 연구가 침체되면서 오히려 규칙기반 알고리즘은 호황을 맞이함

    → 대신 인공지능의 이름보다는 알고리즘, 컴퓨팅 기술 등의 이름으로 추진됨
  - 신경망 기반 알고리즘의 한계 (예: XOR)
  - 예산 및 연구 자금 지원의 연쇄적인 취소

→ 이런 이유들이 복합적으로 작용하여 침체기를 이끌어 냄

AI 개발의 역사

- **인공지능 연구의 2차 BOOM**
  - DBMS 시스템의 발달로 인한 광범위한 지식의 DB 구축
    → 규칙 기반 전문가 시스템 개발 증가
  - 신경망 알고리즘의 한계 극복 → 다층 신경망 알고리즘, 역전파 알고리즘 제안
  - 완전히 새로운 분야에서의 정보에 대한 처리,
    학습가능한 신경망 형태의 제안 및 이론적 증명
  - 고성능 하드웨어 기반의 AI 기술 시장 성장



AI 개발의 역사

- **인공지능 연구의 2차 침체기**
  - 고성능 데스크탑 컴퓨터 등장 → AI 하드웨어 시장 붕괴
  - 전문가 시스템의 한계 직면
    - 규칙 기반 시스템의 기반 구조, 개념의 한계 따른 개발, 운영의 과부하
  - AI 프로젝트의 기대 이하의 성과

  - 이후 연구가 진행되긴 했지만 지원 상황은 열악함
  - 간간히 성과는 나왔으나 크게 주목받지 못함
  - 특히 **신경망이라는 표현이 들어가면 거의 무조건 지원대상에서 배제**됨

AI 개발의 역사

- **2000년대 이후**
  - PC 성능의 급성장으로 인하여 AI, 신경망 분야의 성과가 시작됨
  - 기존의 다층 신경망이 수많은 층을 가진 신경망으로 발전, 딥 뉴럴 네트워크가 등장함
    (딥 뉴럴 네트워크, 딥러닝: **신경망이라는 이름을 쓰면 연구과제 탈락하니까 이름을 바꿈**)
  - **이론상으로 증명되었으나 컴퓨터의 성능, 학습데이터의 부족으로 인하여 실현되지 못했던 신경망 모델**이 **PC의 고성능화 및 빅데이터의 확산에 힘입어 급격한 성장**을 보임

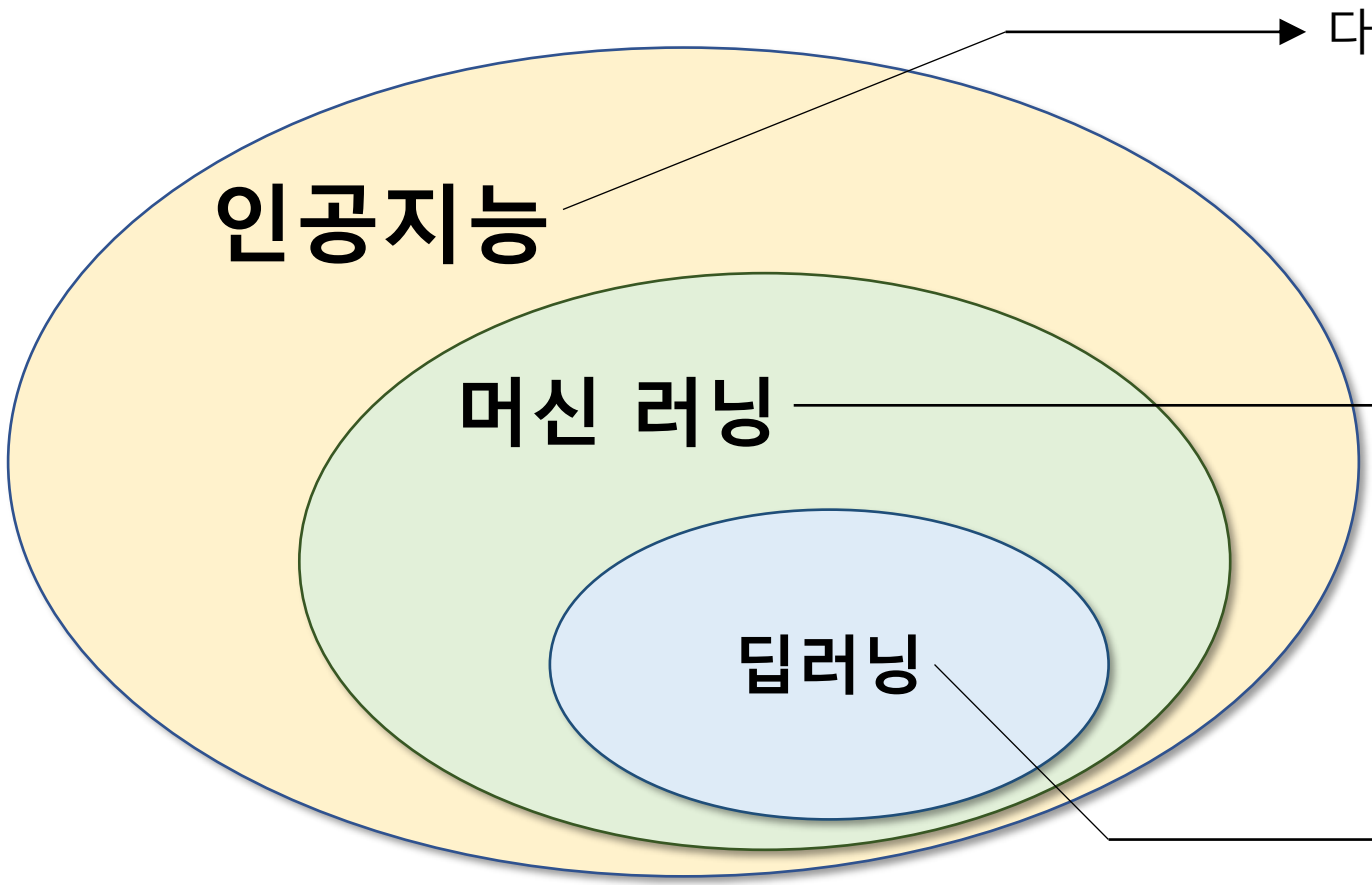
AI 개발의 역사

- **2012년**
  - 딥러닝 기술의 이미지넷 석권
  - 이미지 넷 이미지 인식대회(ILSVRC)에서 최대 75%미만에 머무르던 이미지 인식 성공률
  - 2012년 딥러닝 기반의 Alex Net 개발을 기점으로 84% 돌파
  - 2015년 ResNet 부터는 인간의 인식 성공률인 95%를 뛰어넘음(96.4%)
  - 2017년 SENet의 인식 성공률은 97.7%
  - 현재의 딥러닝 기반 모델의 이미지 인식 성공률은 평균 98%를 유지

AI 개발의 역사

- **2016년 이후**
  - **딥러닝과 강화학습을 이용한 알파고의 등장 (2016.03)**
    - 인간 최고 수준의 이세돌과 대국, 4:1로 알파고의 압승
    - 알파고 제로
      - 기보 등 사전 구축된 데이터 없이 단지 바둑 점수 규칙만을 기반으로 스스로 학습
    - 알파 제로 (알파고 제로의 범용 버전)
      - 게임 법칙만 입력하면 스스로 학습하고 승률을 높이는 방법까지 찾아냄
      - 단 30시간 학습으로 알파고 제로 격파
    - 이후, 스타 제로, 뮤 제로 등 다양한 모델 등장
  - **4차 산업혁명, GPU를 이용한 고성능 PC 등의 환경 개선으로 인하여 다양한 모델 등장**

AI 기술의 관계성

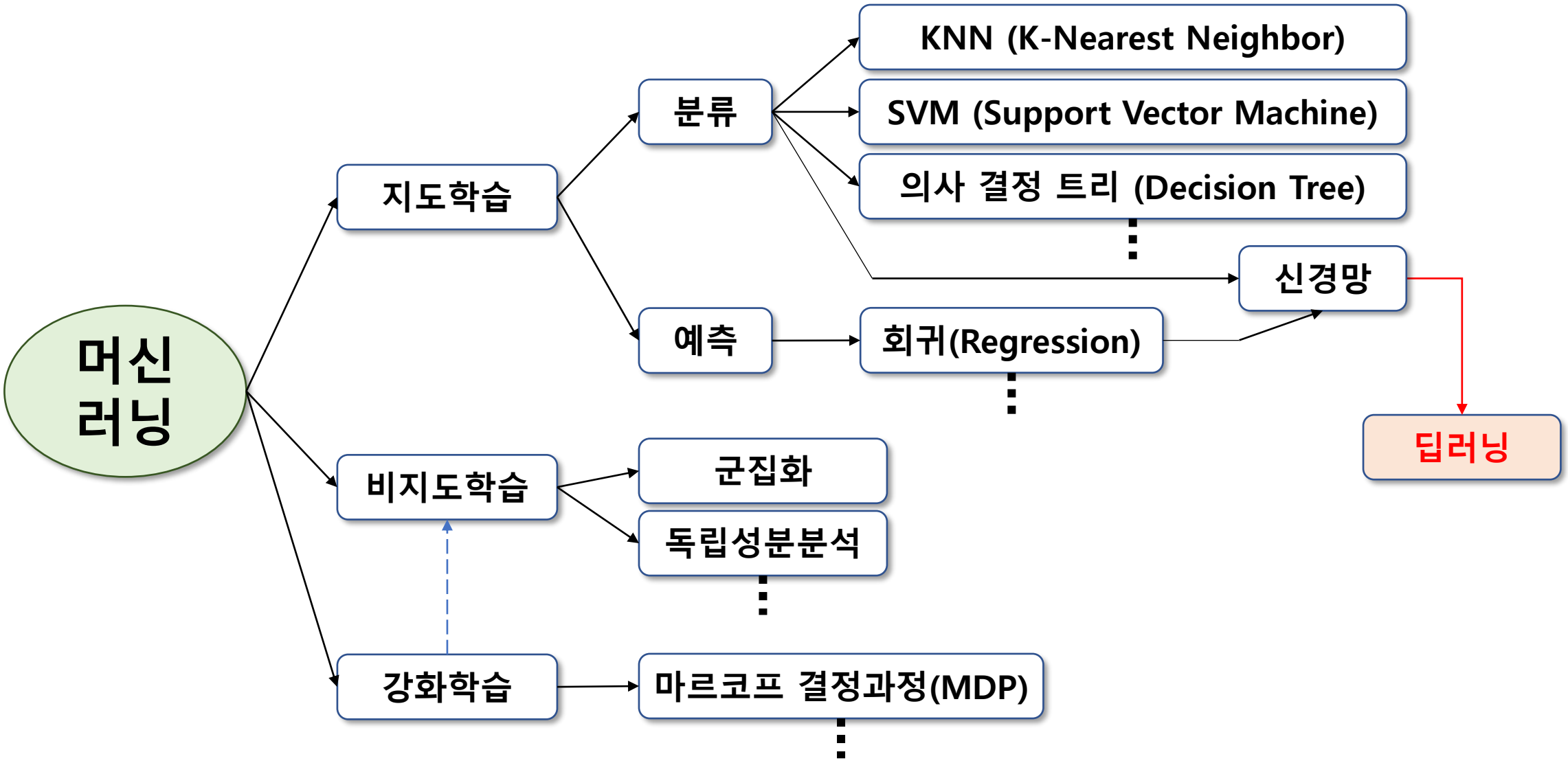


다양한 모델이 연구/개발됨

학습용 데이터를
학습 모델에 적용하여
데이터가 가지는 규칙성과 패턴을
찾아내는 기술

심층 신경망을 기반으로
데이터를 학습하고 처리하는
머신 러닝의 한 종류

AI 기술의 관계성

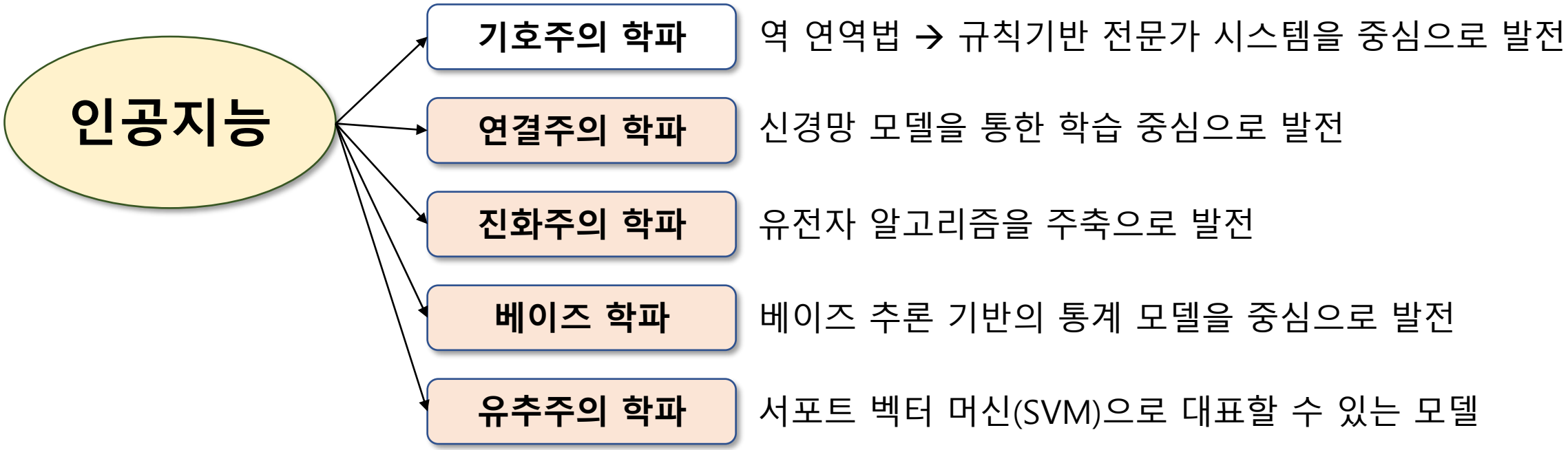

머신 러닝
지도학습
분류
KNN (K-Nearest Neighbor)
SVM (Support Vector Machine)
의사 결정 트리 (Decision Tree)
예측
회귀(Regression)
신경망
딥러닝
비지도학습
군집화
독립성분분석
강화학습
마르코프 결정과정(MDP)

최근의 인공지능 기술의 변화 추세

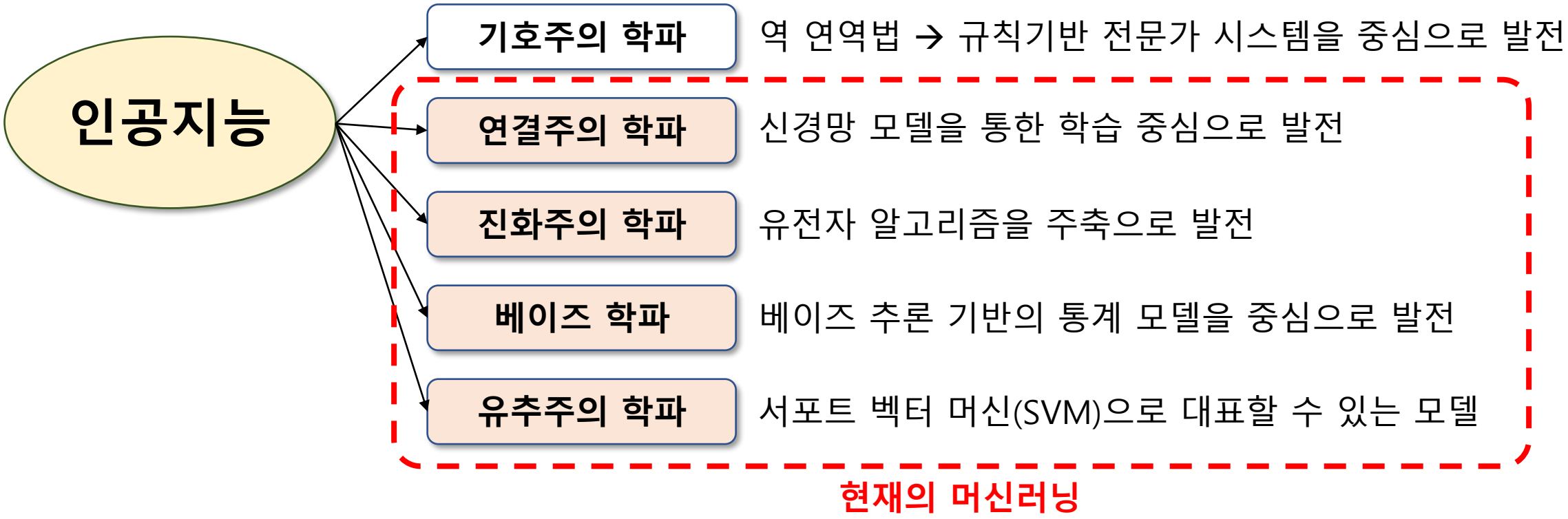
- 다양한 모델의 통합
  - 매우 다양한 모델이 존재함
  - 기존의 모델이 딥러닝 모델로 대체되는 경우가 많음
  - 기존 모델과 딥러닝 모델이 결합하는 경우도 많음

- 경계해야 할 점
  - 딥러닝 모델은 만능이 아니다 → 딥러닝 만능주의를 경계할 것
  - 딥러닝 모델이 적합한 경우는 따로 있다

인공지능의 유형


인공지능
기호주의 학파
역 연역법 → 규칙기반 전문가 시스템을 중심으로 발전
연결주의 학파
신경망 모델을 통한 학습 중심으로 발전
진화주의 학파
유전자 알고리즘을 주축으로 발전
베이즈 학파
베이즈 추론 기반의 통계 모델을 중심으로 발전
유추주의 학파
서포트 벡터 머신(SVM)으로 대표할 수 있는 모델

인공지능의 유형


인공지능
기호주의 학파
역 연역법 → 규칙기반 전문가 시스템을 중심으로 발전
연결주의 학파
신경망 모델을 통한 학습 중심으로 발전
진화주의 학파
유전자 알고리즘을 주축으로 발전
베이즈 학파
베이즈 추론 기반의 통계 모델을 중심으로 발전
유추주의 학파
서포트 벡터 머신(SVM)으로 대표할 수 있는 모델
현재의 머신러닝

인공지능의 유형

- **기호주의**
  - **흄의 귀납문제 :**
    - 우리가 본 것에서 시작한 일반화를 → 보지 못한 것에까지 적용하는 일을..
      어떻게 하면 정당화 할 수 있을까?
      → 정당화 할 수 있는 근거가 없다면 → 우리가 본 몇 가지 사례를 → 법칙으로 발전시킬 수 없음
  - **기호주의란 경험으로 얻은 지식, 사고 체계를 논리적으로 귀결 시키는 과정**
  - **이러한 과정을 시스템으로 구현하는 것이 기호주의의 머신 러닝**

인공지능의 유형

- 기호주의 머신 러닝을 위한 하나의 예
- 데이트 신청의 성공 패턴

주일, 데이트 종류는 결정적인 조건이 아님

날씨, TV프로그램 중 중요한 조건이 있다고 추측

1~3 중에서 날씨가 쌀쌀한 경우가 있다면?
4의 경우, 날씨가 온화했다면?
1~3 중에서 TV 프로그램이 좋았다면?
4의 경우, TV 프로그램이 따분했다면?

경우	주일	데이트 종류	날씨	오늘 밤 TV 프로그램	데이트 성사 여부
1	주중	저녁식사	온화	따분함	승낙
2	주말	클럽	온화	따분함	승낙
3	주중	클럽	온화	따분함	승낙
4	주말	클럽	쌀쌀함	좋음	거절
5	**주말**	**클럽**	**쌀쌀함**	**따분함**	**?**

- 이처럼 데이터의 패턴을 분석, 그 결과를 예측할 수 있도록 학습하는 것이 기호주의의 머신 러닝

인공지능의 유형

- **기호주의 머신 러닝의 문제점**
  - **우리가 발견한 패턴이 실제로 존재하는가? → 통계적 검증 필요, 수많은 데이터와 경우의 수**
  - **데이터에 적합한 단순한 가설을 선택한다면? → 사람이 편한 것이지 정확도, 성능 향상은 없음**
  - **기호주의 머신 러닝은 아는 것이 너무 적은 상태에서 학습 시작 → 결승점 도달 실패 확률 높음**
  - **역연역법을 통해서 논리적으로 예측하는 방안 → 나름대로 좋은 성과를 거둠. 그러나...**
    - 너무 많은 규칙을 관리해야 함 → 계산량 문제 **→ 해결? → 의사결정트리 (스무고개놀이) 등**
    - 잡음(무관한 데이터)에 쉽게 오류를 일으킴
    - 가장 큰 문제: **실제 개념은 규칙의 모음으로 간결하게 정의되는 일이 거의 없다는 사실**

인공지능의 유형

- **연결주의**
  - **심리학자 “헵”의 규칙을 기반으로 만들어진 유형**
    - 헵: 신경과학자보다 먼저 신경세포의 연결방식을 제안한 심리학자
    - 헵의 규칙이란?
      - **시냅스의 앞과 뒤에서 동시에 신경세포가 흥분할 때, 해당 시냅스의 효율이 강화됨**
      - 적당한 추측을 기반으로 심리학과 신경과학의 착상들을 통합해 놓음

인공지능의 유형

- 연결주의 머신 러닝의 개념
  - 각 개념(데이터)와 기억은 두뇌에서 세포의 모임으로 나타난다
  - 개념(데이터)은 모든 곳에 조금씩 저장되어 있다.
  - 두뇌는 수십억의 신경세포가 동시에 동작하며 많은 계산을 수행한다. 그러나 각 신경세포는 1초에 1000번 정도 반응하므로 계산이 느리다(병렬시스템)
  - 신경세포에는 수천개의 신경 접합부가 있다. 등..

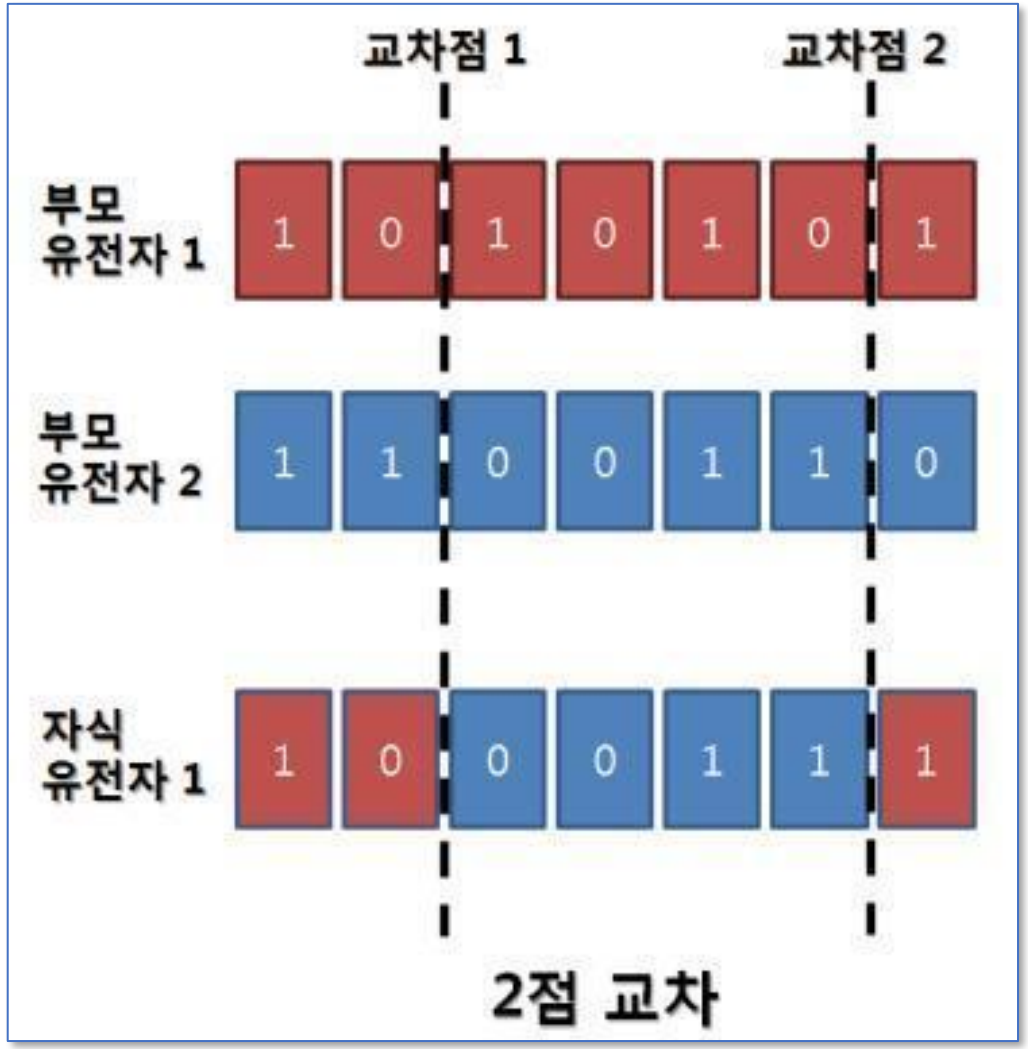
→ 두뇌가 어떻게 만들어지는가 이해해야 두뇌를 시뮬레이션(모의 실험)할 수 있으며 인공지능(머신 러닝)은 두뇌를 재 구축함으로써 구현 가능하다

인공지능의 유형

- 연결주의 머신 러닝의 문제점
  - 두뇌, 신경의 구조, 작용 등에 대하여 아직 모르는 부분이 너무 많음
  - 연결주의 등장 당시의 기준으로 컴퓨터의 성능이 지나치게 낮음 (구현 불가능)
  - 병렬처리해야 하는 데이터가 너무 많음 등...

인공지능의 유형

- 진화주의
  - 진화론에서 출발하여 진화, 돌연변이 등을 통해 학습을 수행하는 유전자 알고리즘 중심 연구
  - 연결주의(신경망) 연구자였던 홀랜드가 생물학자 겸 통계학자인 로널드 피셔의 “자연선택의 유전 이론 " 논문을 접한 후 제안한 이론을 기반으로 함
  - 가장 적응력이 높은 유전자만 살아남고 살아남은 유전자가 가장 정확한, 또는 적절한 결과를 도출할 가능성이 크다



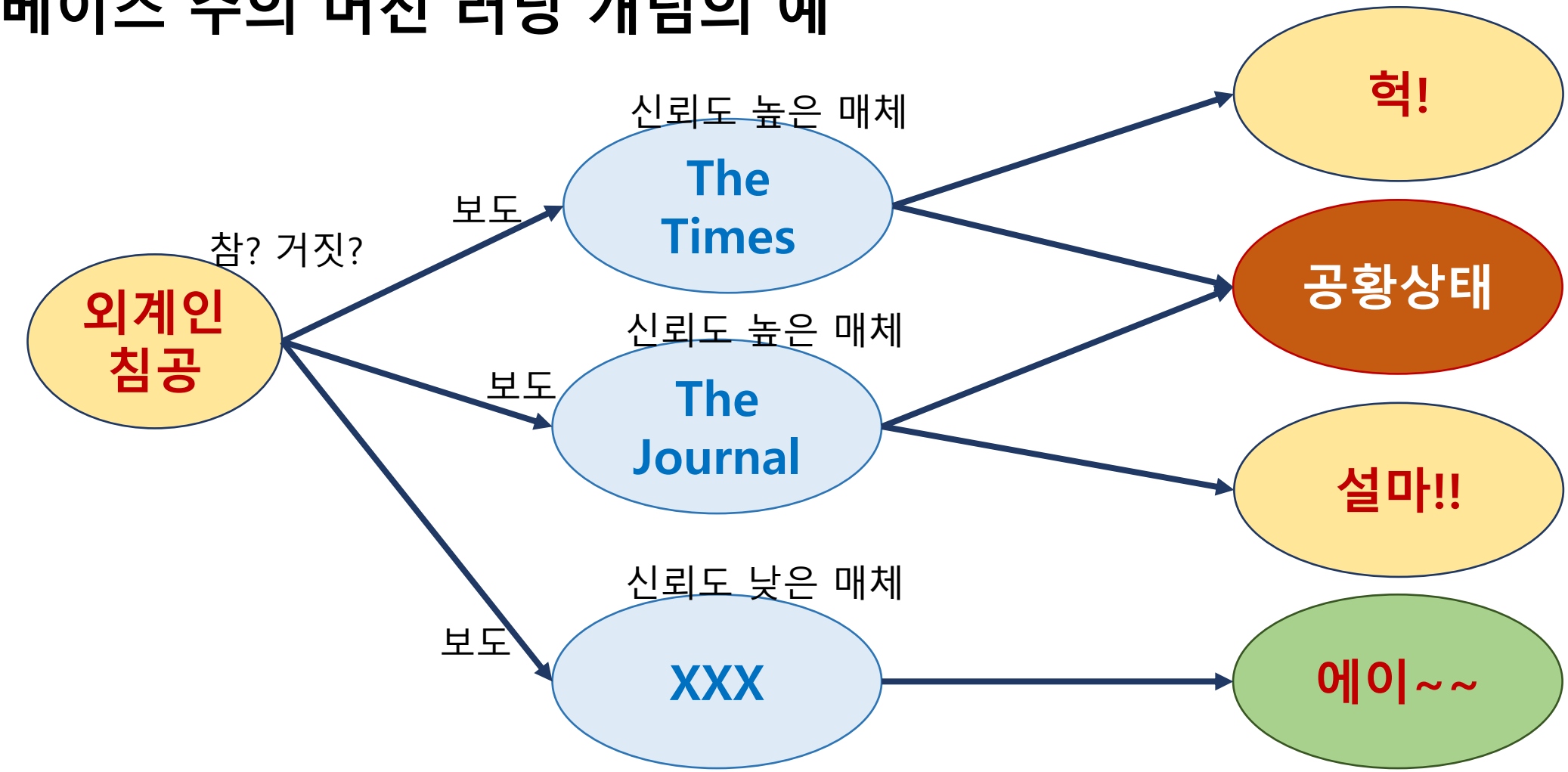
인공지능의 유형

- 베이즈 주의
  - 통계학의 일부인 베이즈 정리를 기반으로 한 머신 러닝
  - 어떤 원인에서 어떤 결과가 일어날 가능성이 더 높을 수록

    → 그 결과가 나타났을 때 그것이 원인일 가능성이 더 높다 (확률적인 기반)
  - 원인과 결과, 즉 인과관계에 대한 추론을 기반으로 학습, 예측을 진행함
  - 그런데... 인간은 언어 추리가 연관되면 베이즈 추론(베이즈 정리를 기반으로 하는 추론)을 매우 잘 하는 것은 아니다. 인간은 원인의 사전 확률을 무시하는 경향이 있다.

인공지능의 유형

- 베이즈 주의 머신 러닝 개념의 예



인공지능의 유형

- 유추주의
  - 사물, 현상에 대한 유추를 기반으로 학습을 진행하는 연구
  - 통계학에서 먼저 알고리즘화 되기 시작했으며 컴퓨터 과학 전 분야에서 많은 연구가 진행되고 있는 분야
  - 신경망, 기호주의, 유전자 알고리즘 등 다양한 머신 러닝 모델에도 영향을 끼침

인공지능의 유형

- **유추주의 개념의 예시**
  - **역사상 악명높은 사기꾼 – 프랭크 애버그네일 주니어**
  - **의학적인 교육은 전혀 받지 않은 채로 1960년대 후반 애틀란타에서 1년 가까이 의사로 행세함**
  - **아무도 모르게...**
  - **행위**
    - 빈 진료실에 들어감 → 아무것도 모르는 환자 입실 → 환자의 증상을 들음 → 캐비닛에 들어있는 환자들의 진료 기록 검색 → 유사한 다른 환자의 기록을 꺼내어 동일한 진단 내림 → 1년 가까이 아무도 의심하지 않음

인공지능의 유형

- 5가지 연구 유형에서...
  - 모든 것을 만족하는 마스터 알고리즘은 아직까지 나오지 않았다
  - 각 유형은 각각의 장, 단점을 가지며 어느 하나가 완벽한 모델은 없다
  - 최근의 추세는 각 유형이 서로 융합, 협력하여 서로의 단점을 보완하려는 움직임
  - 새로운 연구를 위해서도 서로의 장, 단점을 잘 알고 활용할 필요가 있음

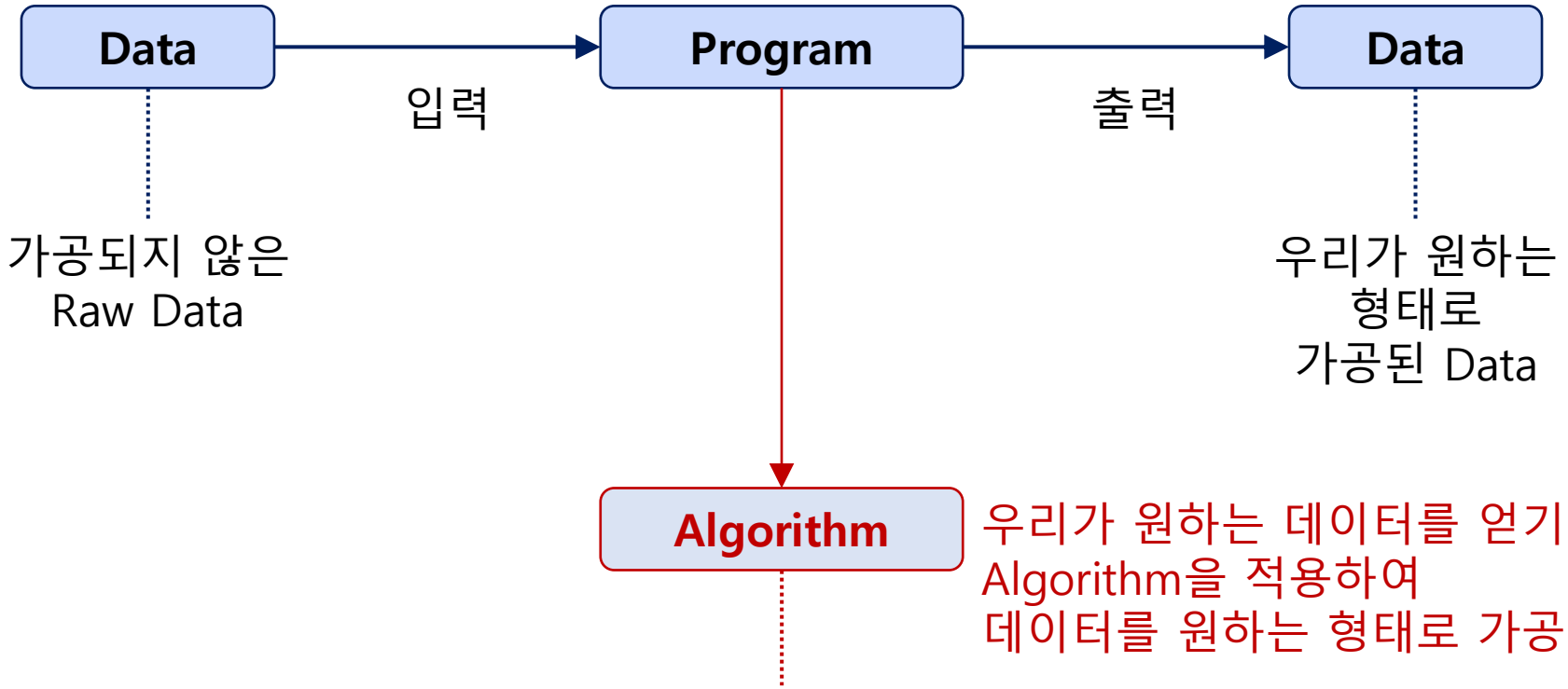
→ 앞에서 살펴본 "다양한 모델의 통합 추세"로 이어지게 됨

- AI와 프로그래밍

AI와 프로그래밍

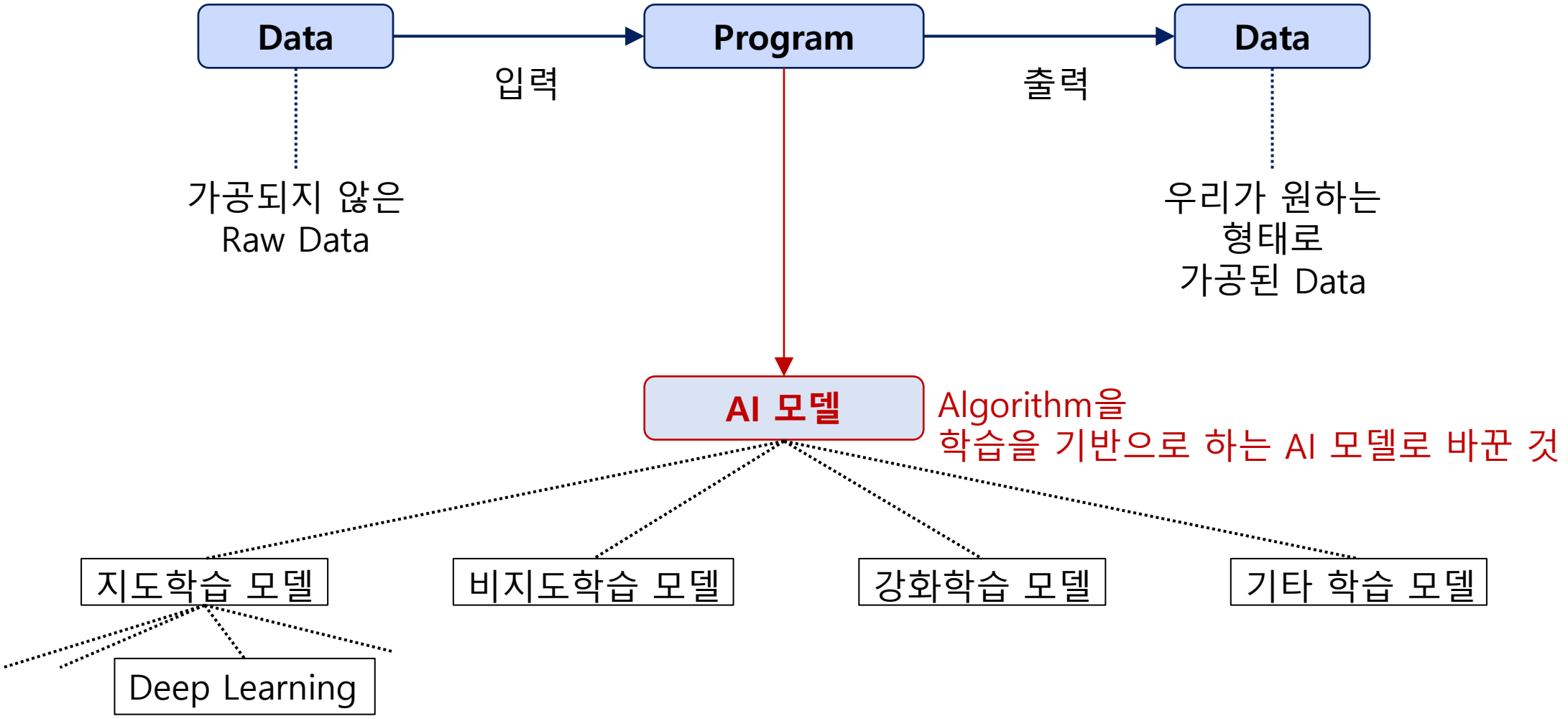
- AI 기술은 컴퓨터를 기반으로 프로그램의 형태로 구현됨
- 컴퓨터 프로그램이란?
  - 보유한 데이터를 입력하여
  - 우리가 원하는 결과 데이터를 만들어 내도록 지시하는
  - 명령어의 모임

AI와 프로그래밍

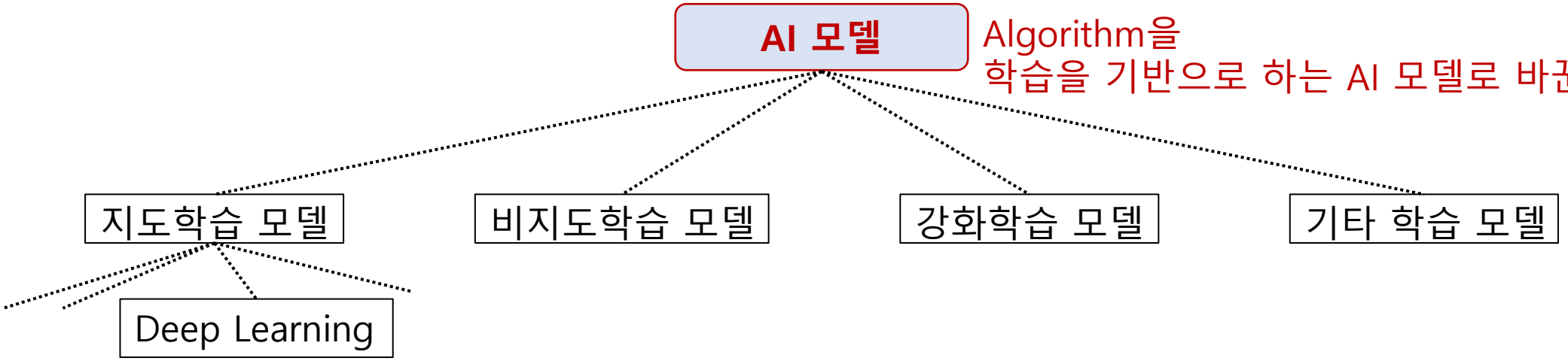


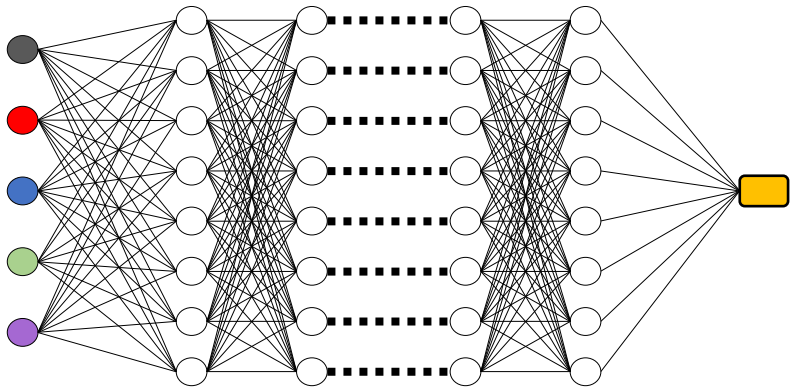
어떤 데이터가 입력되면 → 어떻게 가공해서(Algorithm) → 어떤 형태의 데이터를 출력하라

AI와 프로그래밍


Data
입력
Program
출력
Data
가공되지 않은
Raw Data
우리가 원하는
형태로
가공된 Data
AI 모델
Algorithm을
학습을 기반으로 하는 AI 모델로 바꾼 것
지도학습 모델
비지도학습 모델
강화학습 모델
기타 학습 모델
Deep Learning

AI와 프로그래밍





인공신경망을 기반으로 하는 딥러닝 모델의 설계와 구축

- 딥러닝 감잡기

딥러닝 적용 개념의 예를 살펴보기

- 문제 제기
  - 앞으로 다양한 딥러닝 관련 개념, 지식과 이론, 모델 등을 살펴보겠지만
  - “딥러닝을 어떻게 사용하라는 것인지?”를 이해하지 못하는 경우가 많음

- 간단한 예시를 통해 딥러닝 적용의 개념을 이해하자

잘 익은 귤 분류하기

- **문제 발생**
  - **시골 할머니 댁에 내려갔더니 놀지 말고 일이나 하라고 하신다...**
  - **수확한 귤을 분류하라고 하심**
    - 2개의 상자가 있음: 잘 익은 귤 상자 + 덜 익은 귤 상자
    - 일단 분류하자 → 하면 할 수록 귀찮음...
  - **사람이 분류하는 과정을 보면서 감을 잡고 AI 머신을 만들어서 분류하도록 시키자!!**
    - AI 머신에서는 프로그램을 작성해서 분류하게 된다

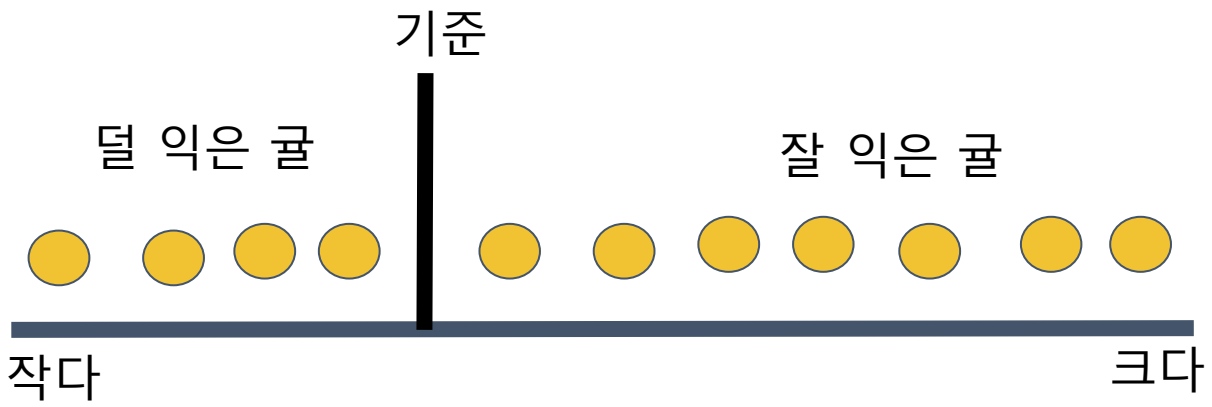
잘 익은 귤 분류하기

- **사람의 분류 과정**
  - **귤을 잔뜩 모아 놓고 분류를 시작함**
  - **잘 익은 귤과 덜 익은 귤을 각각 상자에 나눠 넣음 → 귀찮음**
  - **뭔가 고정된 분류 기준이 있으면 좋겠다...**
    - 잘 익은 귤을 보니 대체로 크고 덜 익은 귤은 작더라
    - 대충 크기를 재어보니 잘 익은 귤은 지름이 7.5cm 이상이다.
  - **귤의 지름이 7.5cm인 것을 기준으로 분류하자 → 프로그램 작성**

    → AI 머신의 분류 작업으로...

잘 익은 귤 분류하기

- AI 머신의 분류 과정
  - if size > 7.5 then right_box else left_box



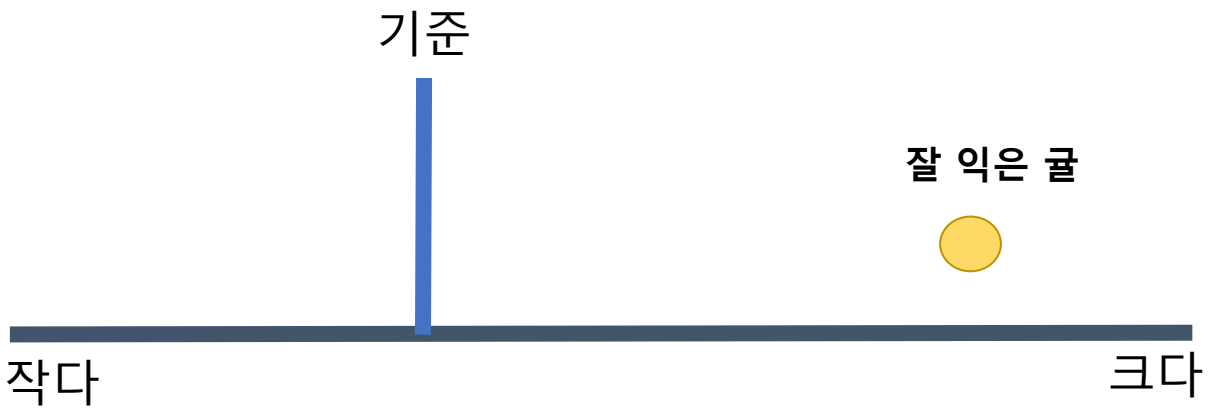
잘 익은 귤 분류하기

- AI 머신의 작업결과를 보고 있으니...
  - 가끔씩 7.5cm 미만의 귤 중에 잘 익은 것이 나온다.
  - 가끔씩 7.5cm 이상의 귤 중에서도 덜 익은 것이 나온다.
  - 하드 코딩한 AI 머신의 프로그램을 바꿔야할 것 같다....

    → 하드 코딩한 기준도 스스로 찾아내도록 하고 싶다..

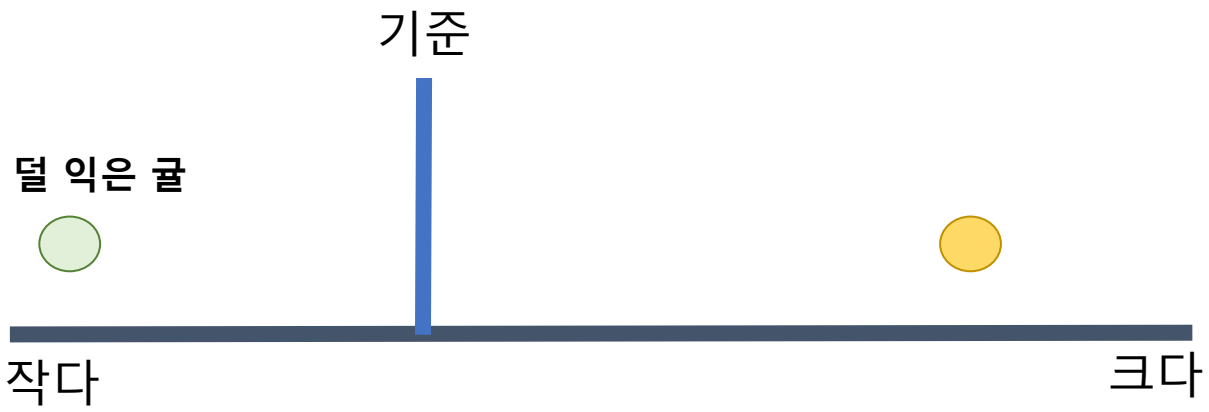
잘 익은 귤 분류하기

- **AI 머신의 작업 내용**
  - **귤의 분류 기준 찾기**



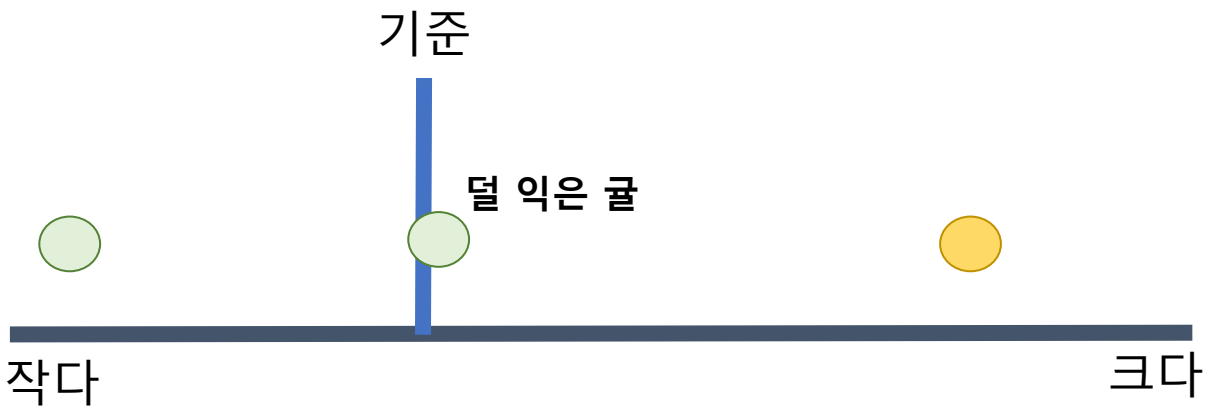
잘 익은 귤 분류하기

- **AI 머신의 작업 내용**
  - **귤의 분류 기준 찾기**



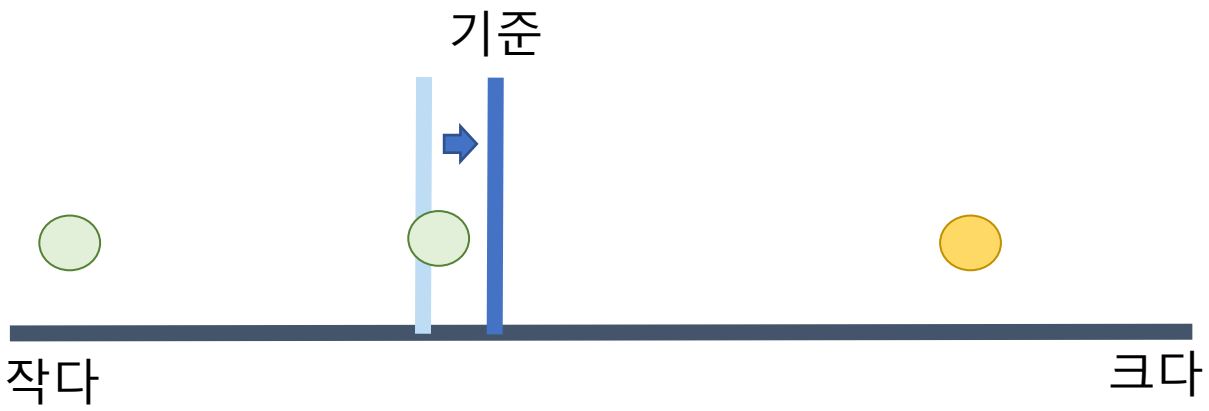
잘 익은 귤 분류하기

- **AI 머신의 작업 내용**
  - **귤의 분류 기준 찾기**



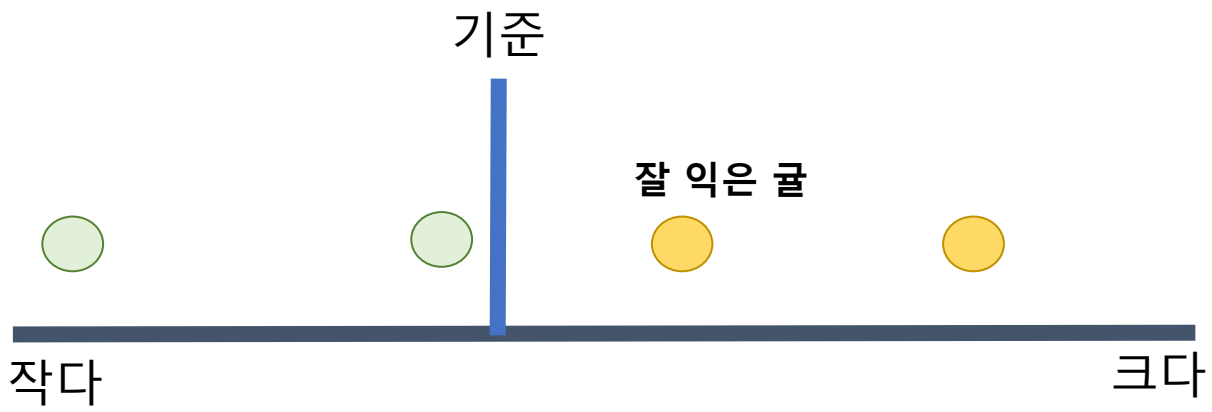
잘 익은 귤 분류하기

- **AI 머신의 작업 내용**
  - **귤의 분류 기준 찾기**



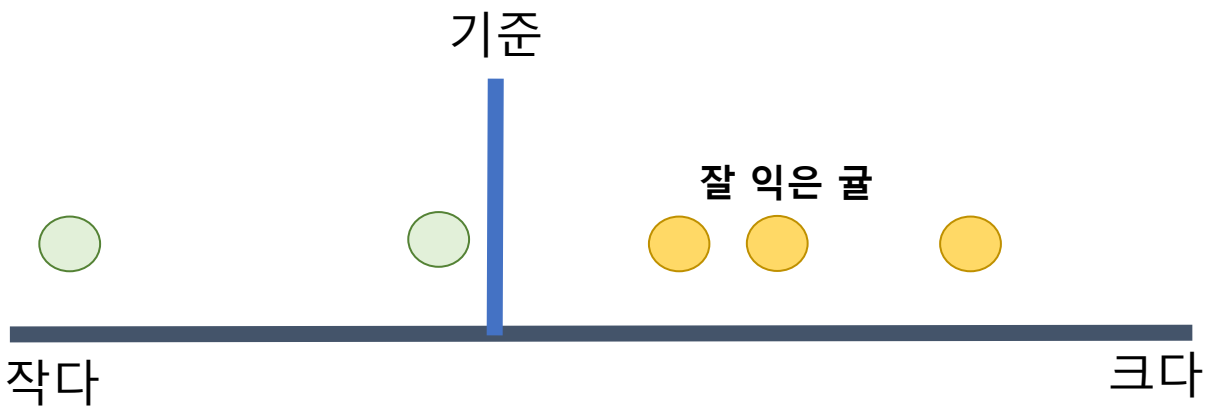
잘 익은 귤 분류하기

- **AI 머신의 작업 내용**
  - **귤의 분류 기준 찾기**



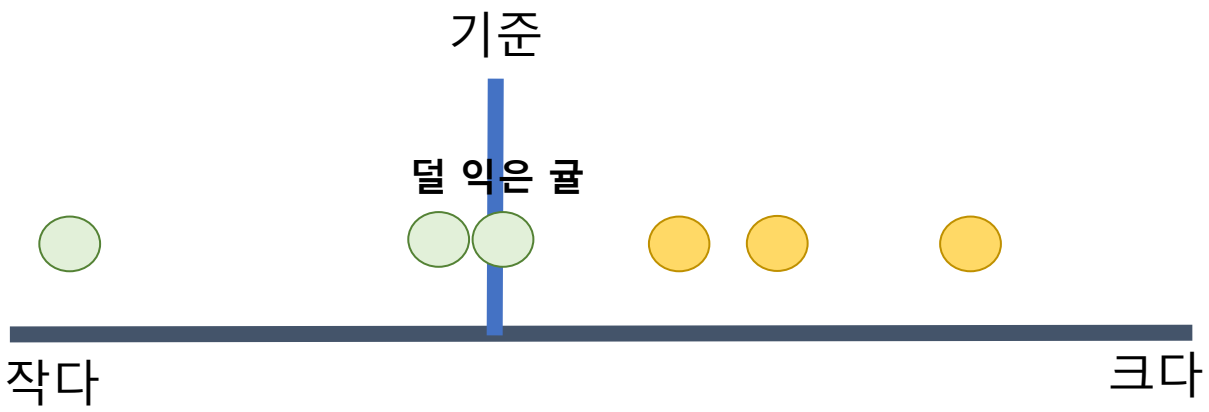
잘 익은 귤 분류하기

- **AI 머신의 작업 내용**
  - **귤의 분류 기준 찾기**



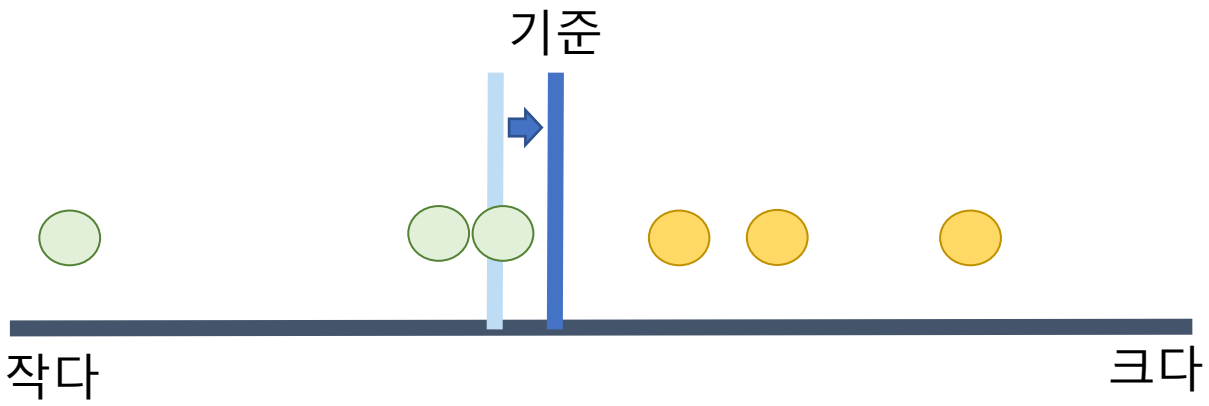
잘 익은 귤 분류하기

- **AI 머신의 작업 내용**
  - **귤의 분류 기준 찾기**



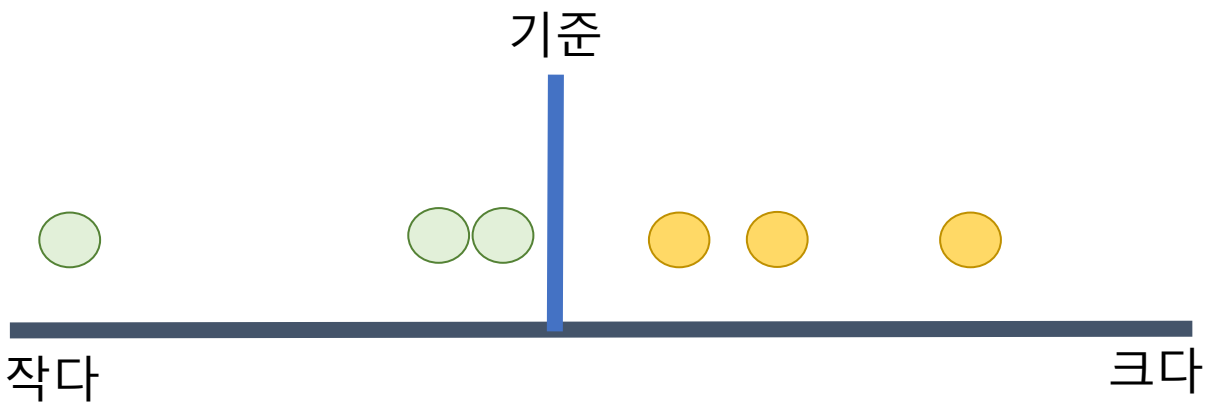
잘 익은 귤 분류하기

- **AI 머신의 작업 내용**
  - **귤의 분류 기준 찾기**



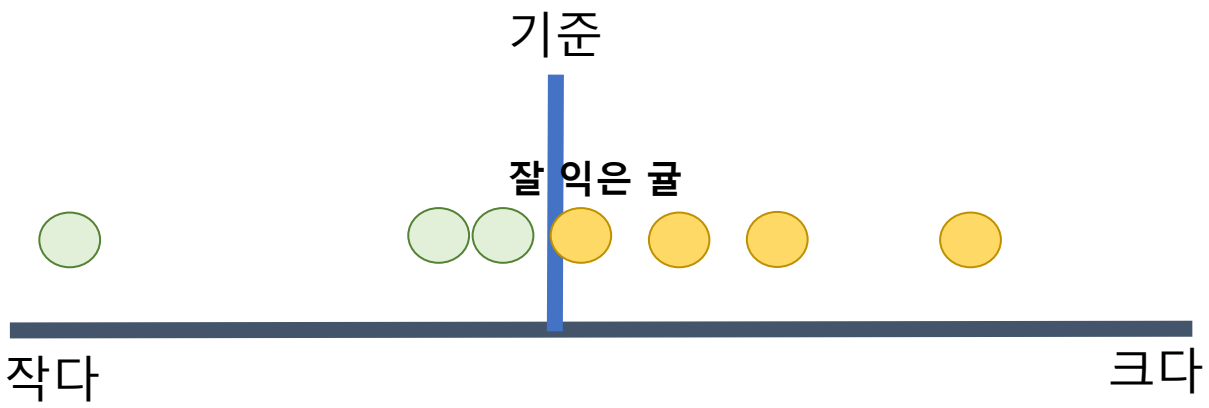
잘 익은 귤 분류하기

- **AI 머신의 작업 내용**
  - **귤의 분류 기준 찾기**



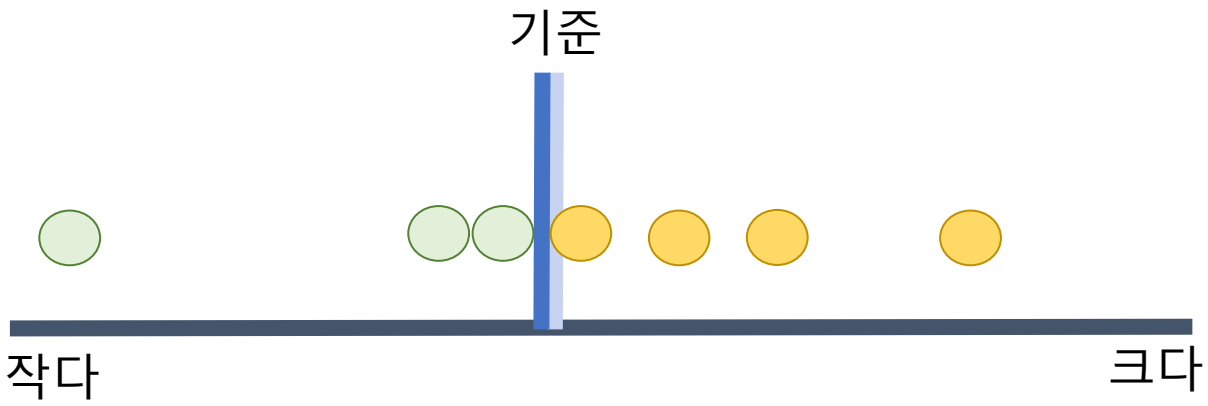
잘 익은 귤 분류하기

- **AI 머신의 작업 내용**
  - **귤의 분류 기준 찾기**



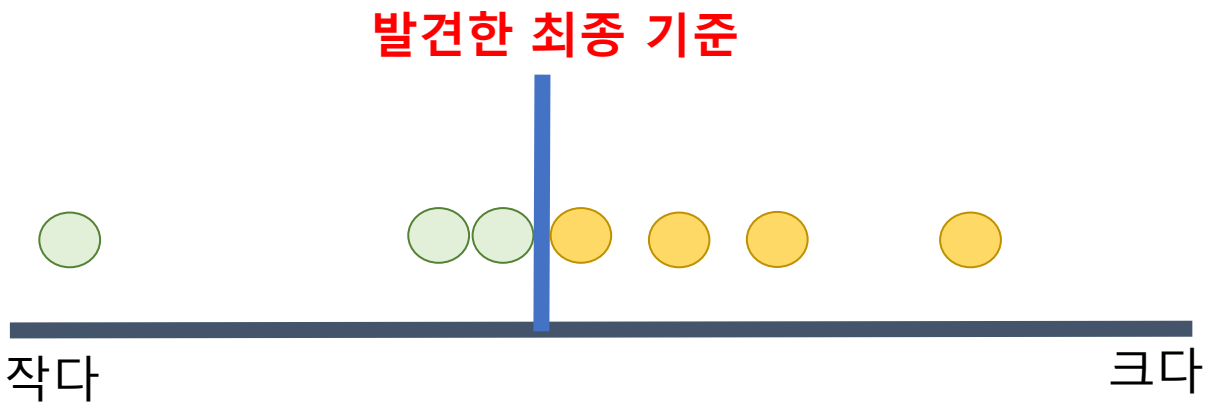
잘 익은 귤 분류하기

- **AI 머신의 작업 내용**
  - **귤의 분류 기준 찾기**



잘 익은 귤 분류하기

- **AI 머신의 작업 내용**
  - **귤의 분류 기준 찾기**



잘 익은 귤 분류하기

- 사람이 찾은 기준과 직접 하드 코딩한 프로그램(알고리즘) → ①
- AI 머신이 찾은 기준과 기준을 찾을 때 사용한 알고리즘 → ②
- ②의 경우가 딥러닝을 적용한 알고리즘의 개념임
- 서로 다른 방식을 사용하여 기준을 찾았지만 ①과 ②의 목적과 사용 방향은 동일함

- 딥러닝(Deep Learning) 모델

딥러닝이란?

- **딥러닝은 신경망 모델이다!!!**
  - **그 이유는?**
  - **예전에 완성된 신경망 모델이 그대로 사용된다.**
    - 부족했던 PC 성능이 급격히 향상되었다.
    - 부족했던 데이터가 이제는 넘쳐나는 시대로 바뀌었다.
    - 예전의 신경망을 구현, 검증하는 것이 가능해 졌다.
    - 구현해보니 안되는 줄 알았던 모델이 아주 잘 돌더라!!
    - 하는 김에 크기를 엄청나게 키워보자 → 역시 잘 되더라 → 딥러닝!!!
  - **좀 과격한 표현이지만 대체로 맞는 말임**

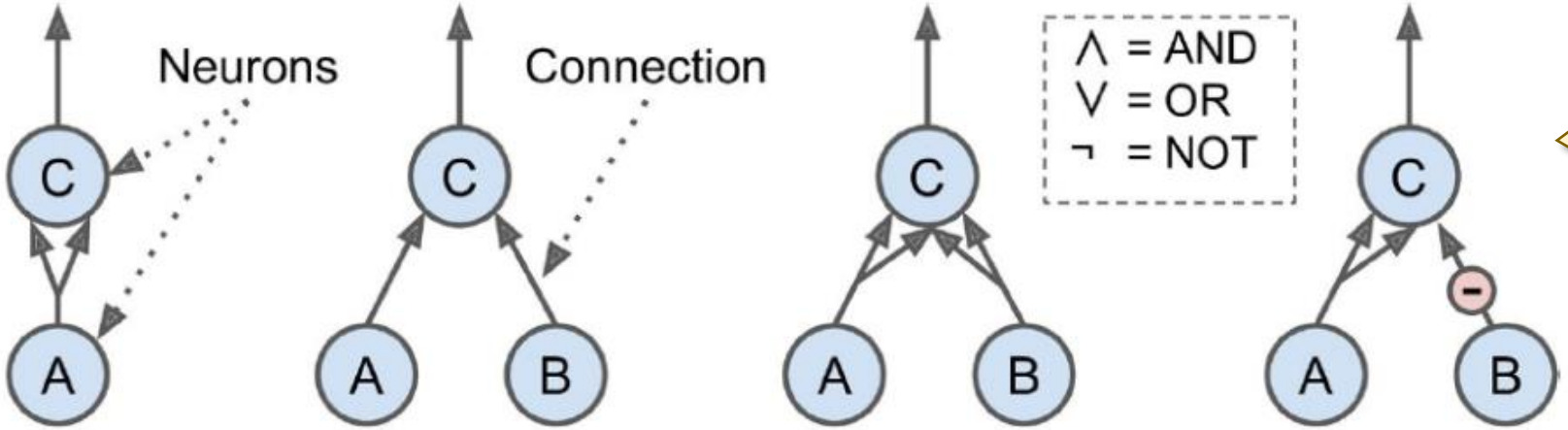
신경망 모델이란?

- 신경세포의 간단하고 효과적인 처리 방식에 착안해 구현된 머신 러닝 알고리즘(모델)의 한 종류
- 신경세포의 형태와 동작을 극도로 단순화 시킨 뉴런 모델을 다수 연결하여 망, 즉 네트워크를 만든 모델

  (= 다량의 뉴런(Neuron)들이 층(Layer)으로 연결되어 간단한 계산과 연결 방식을 통해 복잡한 문제를 해결하는 모델)
- 뉴런의 동작 방식은 컴퓨터 프로그램의 방식에 비해 다양한 장점을 지님

신경망 모델의 아이디어 기반

- 1943년 워런 맥컬록, 월터 피트의 최초의 신경망 모델이 시초



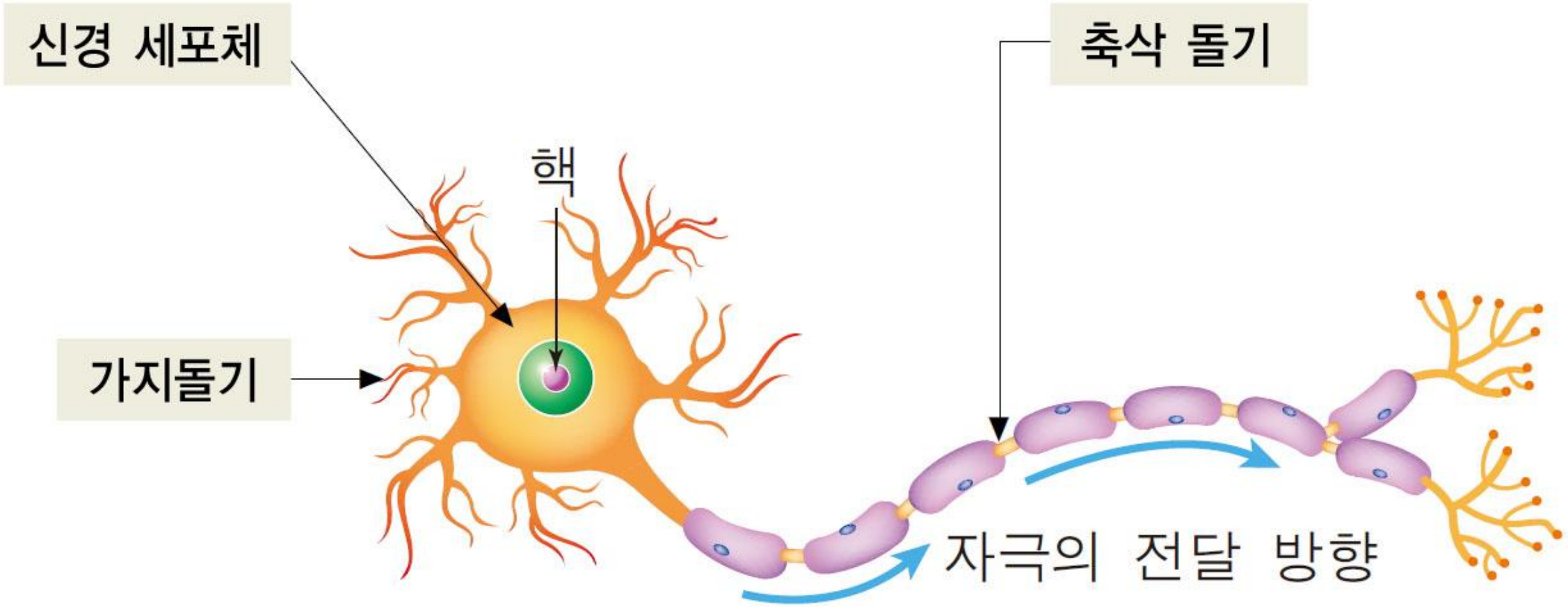
입력되는 데이터들의

연결형태에 따라서

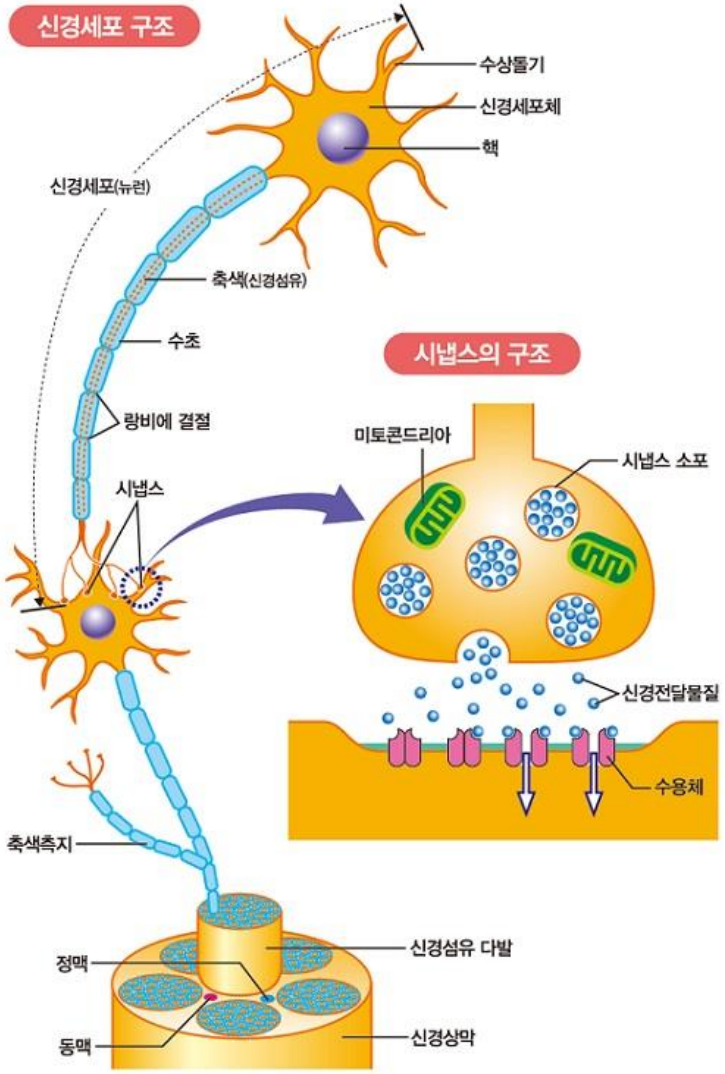
결과가 결정된다

- 헵의 규칙이 신경망 모델의 동작을 정의하는 기반이 됨
  - 시냅스의 앞과 뒤에서 동시에 신경세포가 흥분할 때, 해당 시냅스의 효율이 강화된다.

신경세포의 구조



신경세포의 연결 형태

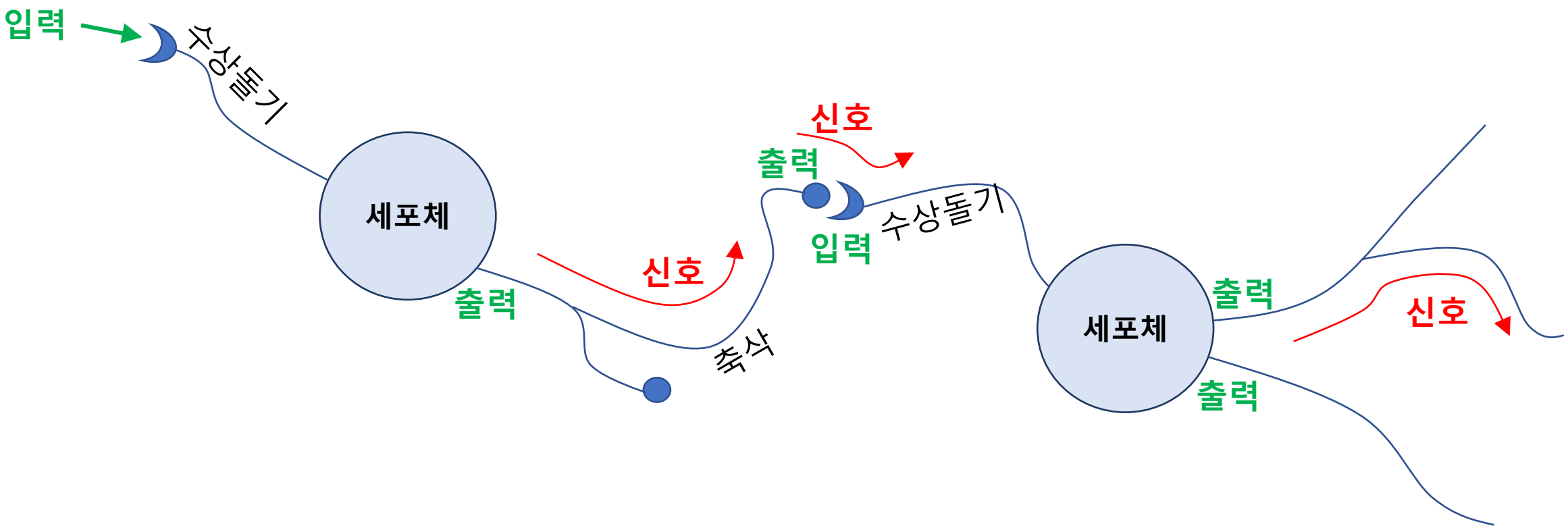


- 신호의 전달은 전기로 이루어짐
- 신경세포의 말단에는 시냅스가 존재
- 시냅스 사이의 신호 전달은 신경전달물질이라는 화학물질을 통해전달됨
- 신경전달물질
  - 세로토닌
  - 도파민
  - 엔돌핀
  - 아드레날린 등..

신경세포의 동작

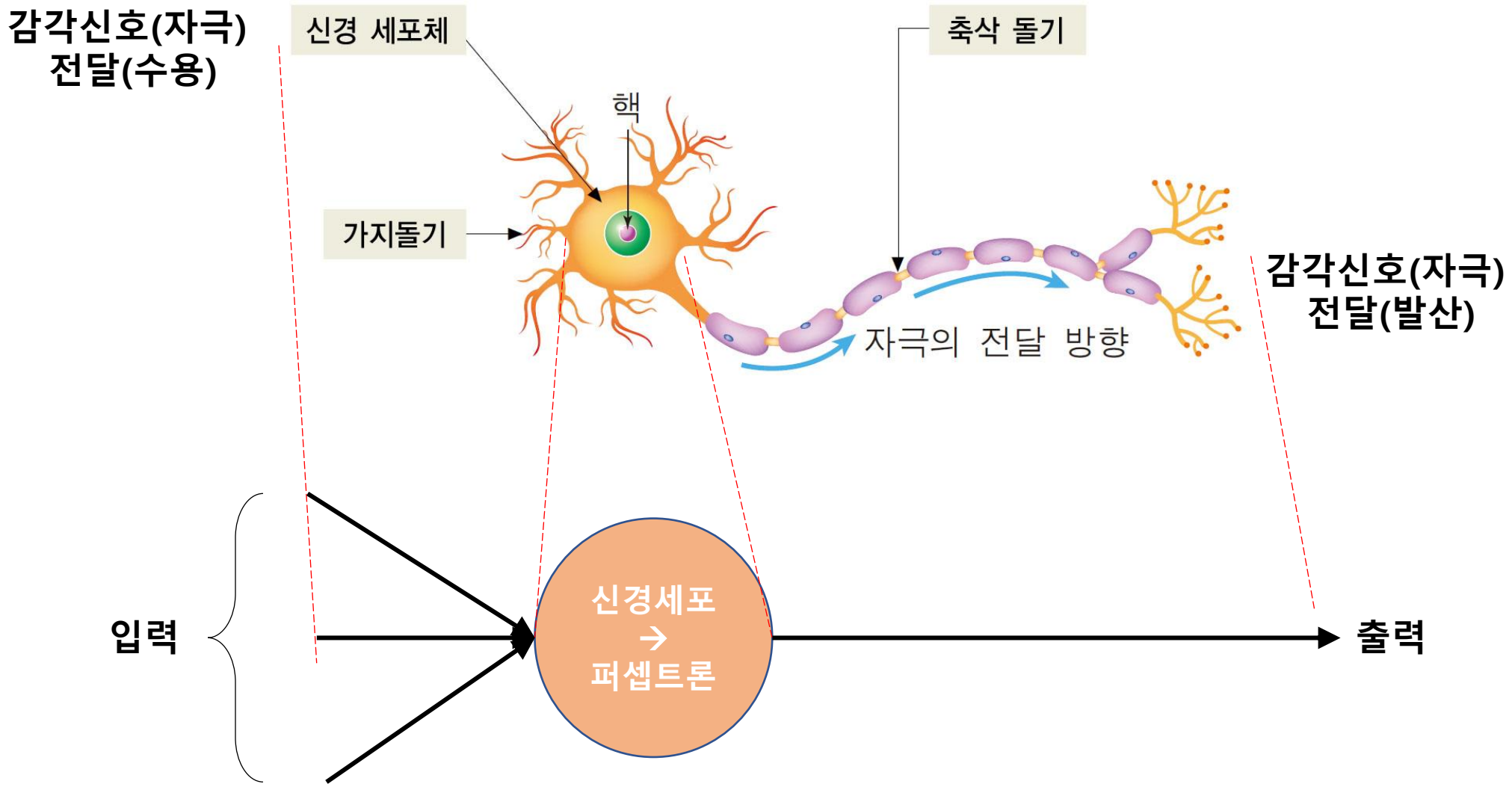
- 신경체계를 구성하는 수많은 신경세포들
- 다양한 감각기관을 통하여 (전기)신호를 발생, 전달
- 각 신경세포는 수많은 시냅스를 통해 신호를 전달 받음
- 전달 받은 신호는 대체로 무시하지만.. 동시에 전달된 신호의 합이 임계값을 넘으면 활성화(발산, 흥분한다 라고 표현함)
- 활성화 된 신경세포는 활성화 패턴에 따라 신경전달물질 분비
- 이웃 신경세포는 신경전달물질을 수용하면서 이온화 작용, 화학작용을 통하여 전기 신호 발생
- 처리 단계 반복

신경세포에서의 데이터 흐름

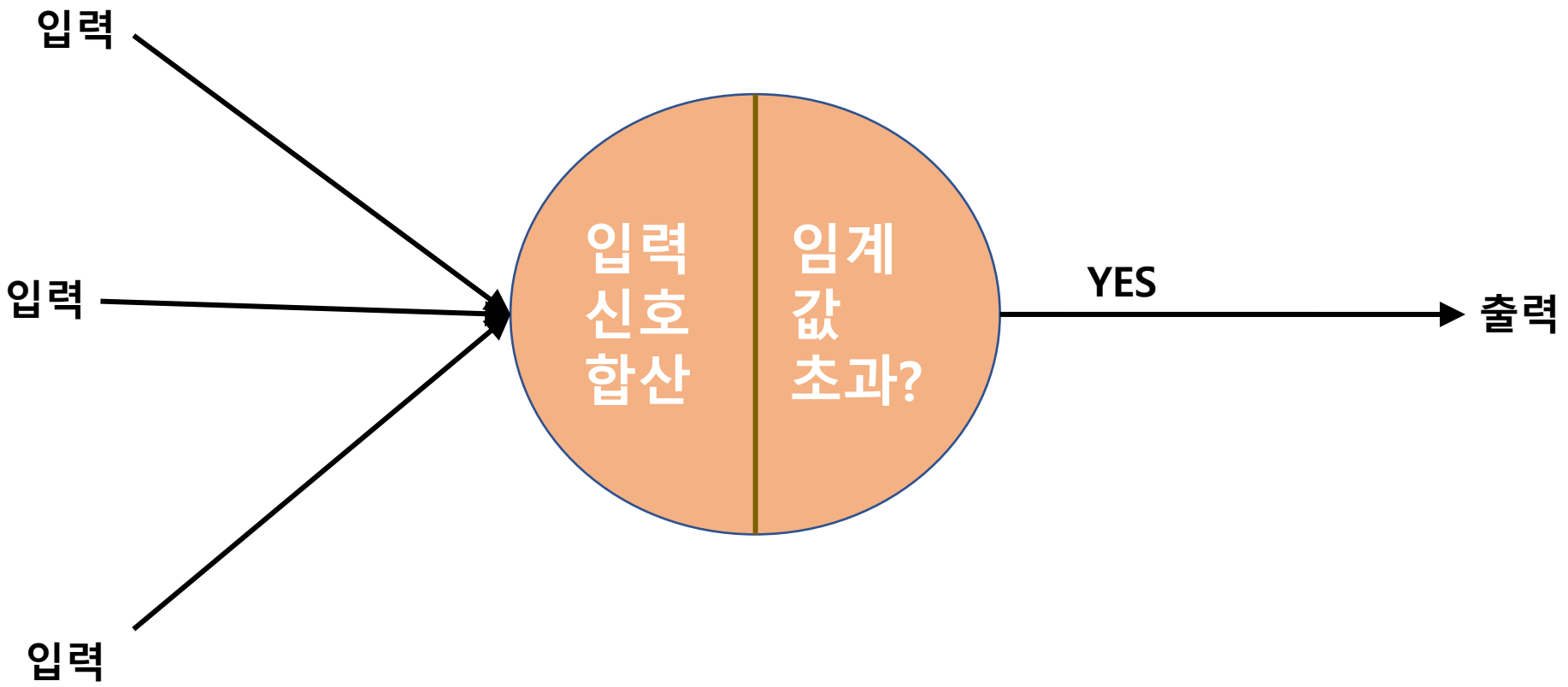


신경 세포 사이의 입 / 출력 흐름

신경세포, 신경망을 어떻게 단순화하였나?



신경세포를 어떻게 단순화하였나?



퍼셉트론 모델의 형태

퍼셉트론 모델의 문제점? 한계?

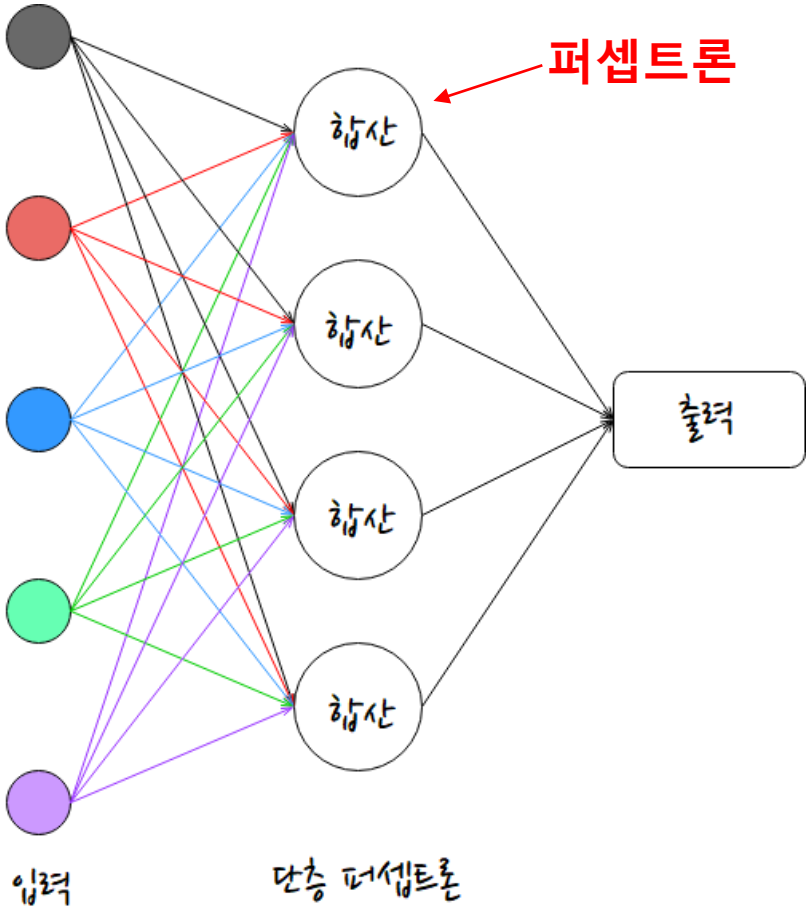
- 퍼셉트론 모델은 실제 신경세포의 형태를 극도로 단순화한 모델
- 전기신호의 전달, 합산, 발산을 모델링 함

- 그러나... 신경세포는 전기신호만 전달하는 것이 아님
- 신경세포의 활성화는 전기신호의 합산으로 이루어지더라도 해당 신호를 전달할 때는 다양한 신경전달물질이 활동함
- 동일하게 전기신호는 전달되더라도 신경전달물질의 종류에 따라 사람의 반응은 달라짐
  → 이런 부분은 전혀 반영되지 않은 모델

- 그러나 기능의 모방을 목표로 하는 약 인공지능의 시점에서는 충분히 활용가치가 높은 모델

신경망의 구성

- 단층 퍼셉트론(SLP, Single Layer Perceptron)

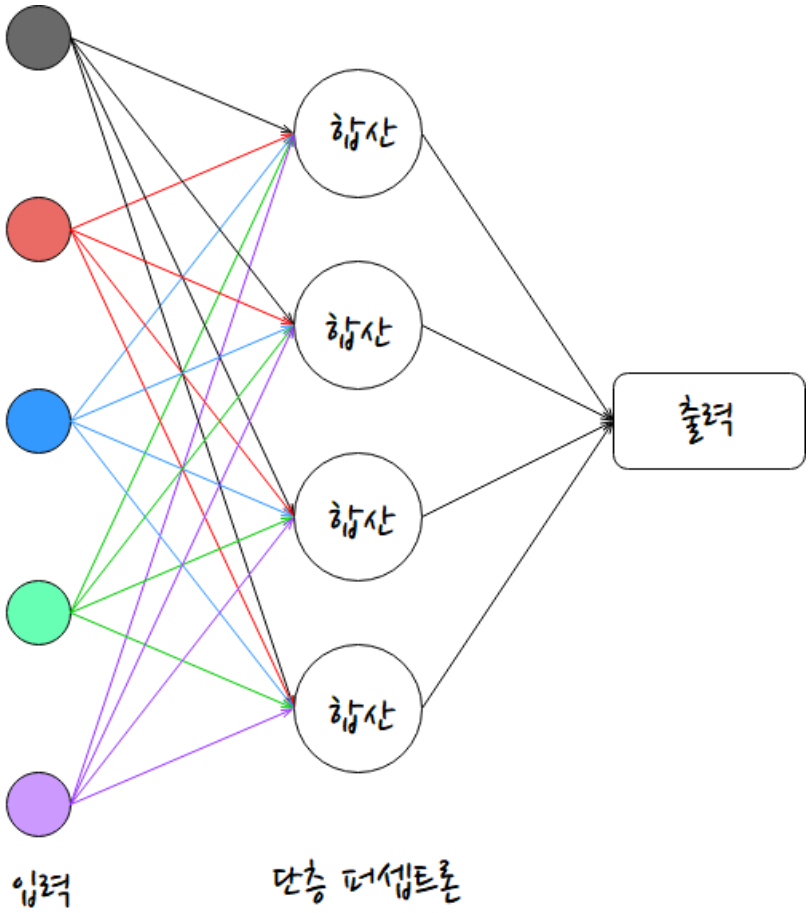


- 다수의 퍼셉트론이 하나의 층을 이루고 있는 형태
- 센서 데이터 등 다양한 데이터를 각 퍼셉트론의 입력으로 전달
- 각 퍼셉트론은 입력된 데이터를 모아서 합산
- 합산 결과가 임계 값을 넘으면 1, 넘지 않으면 0 출력
- 입력층에서 각 퍼셉트론으로 진행되는 통로에는 가중치 적용

  (가중치는 모든 통로가 각각 다르게 적용될 수 있음)
- 왼쪽 그림에서 4개의 퍼셉트론이 각각 1, 0, 0, 1 이라는 결과

  를 낸다면 최종 출력은 1001 이라는 2진수 값이 나오는 형태

단층 퍼셉트론의 문제점



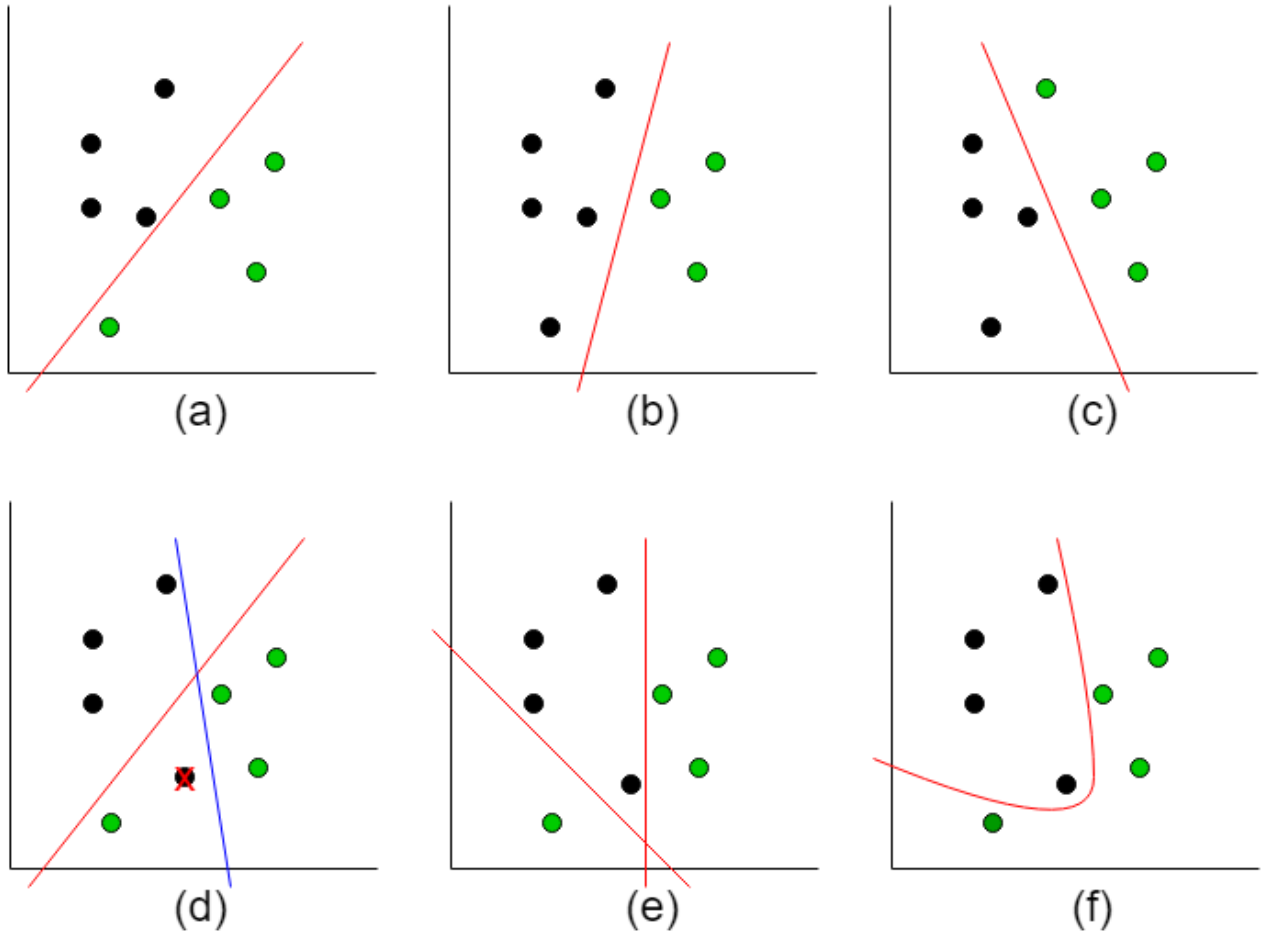
- 단층 퍼셉트론의 구조를 보면...
  - 가중치를 변경할 수 있는 방법이 없다

    → 학습이라는 개념이 없다
  - 한 번 생성된 후에는 아무런 변형이 없는 단순한 분류 알고리즘에 불과함
  - 한 층의 변경가능한 퍼셉트론만 존재

    → 1개의 선을 그어 분리 가능한 패턴만 분류 가능

단층 퍼셉트론의 문제점

- 직선으로 데이터 분류하기

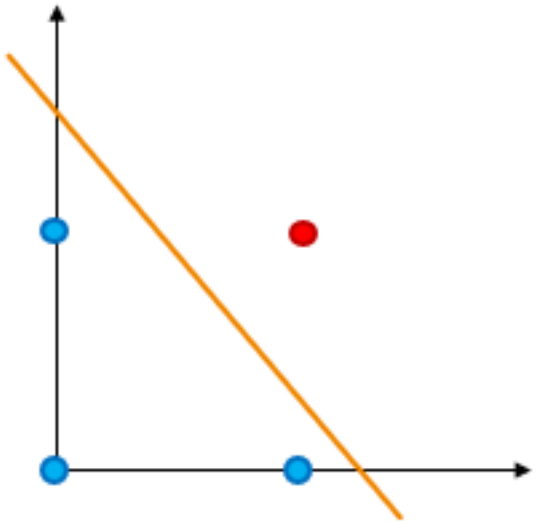


XOR 불가능 문제도

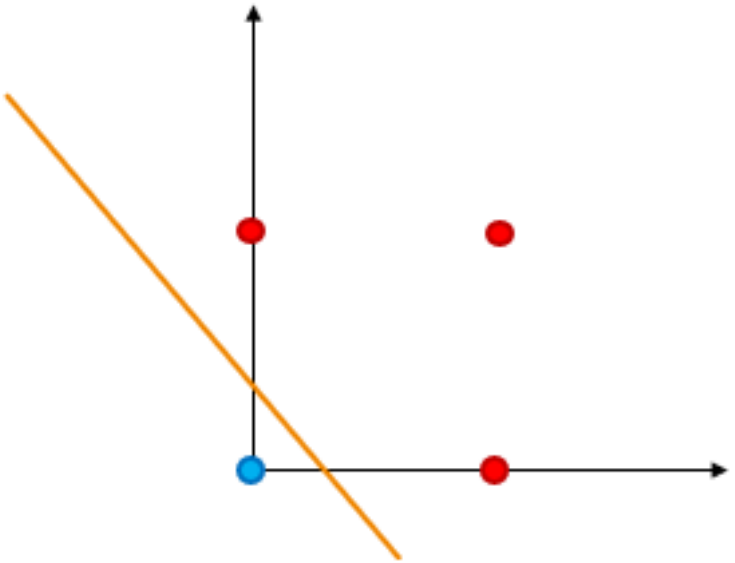
여기에서 확인됨

단층 퍼셉트론의 문제점

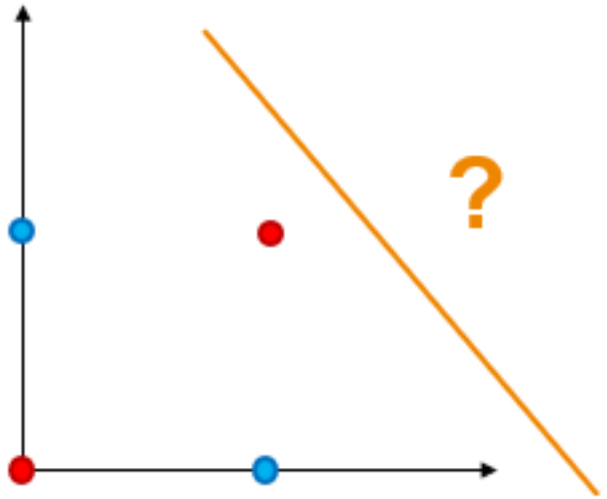
- **XOR 문제**



AND 연산



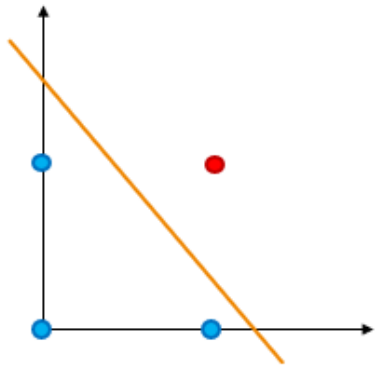
OR 연산



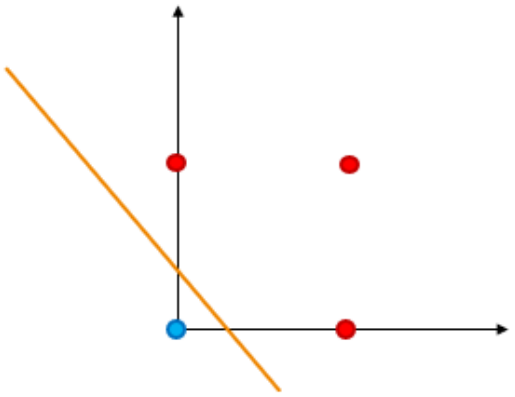
XOR 연산

단층 퍼셉트론의 문제점

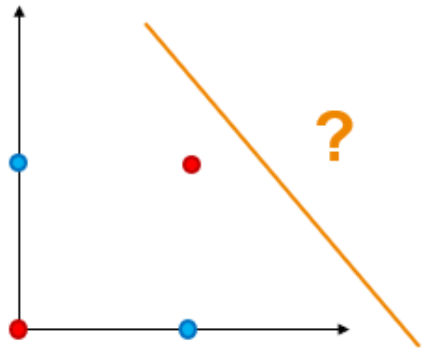
- **XOR 문제**



AND 연산

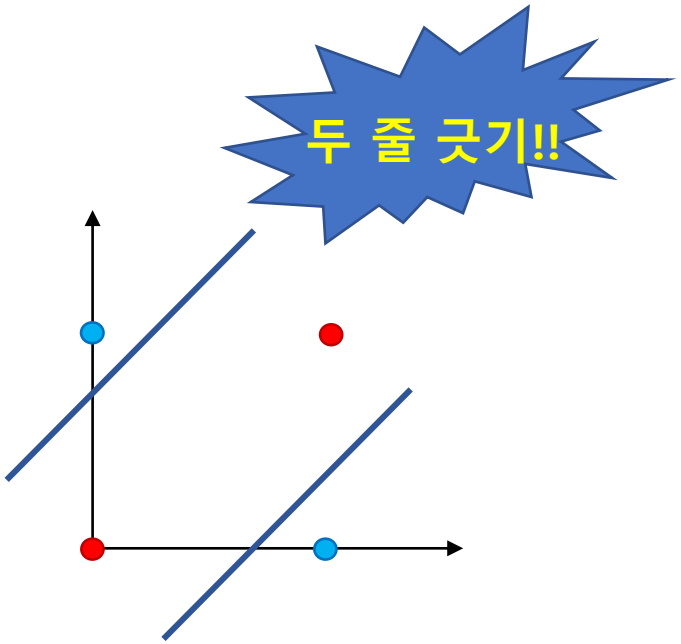


OR 연산



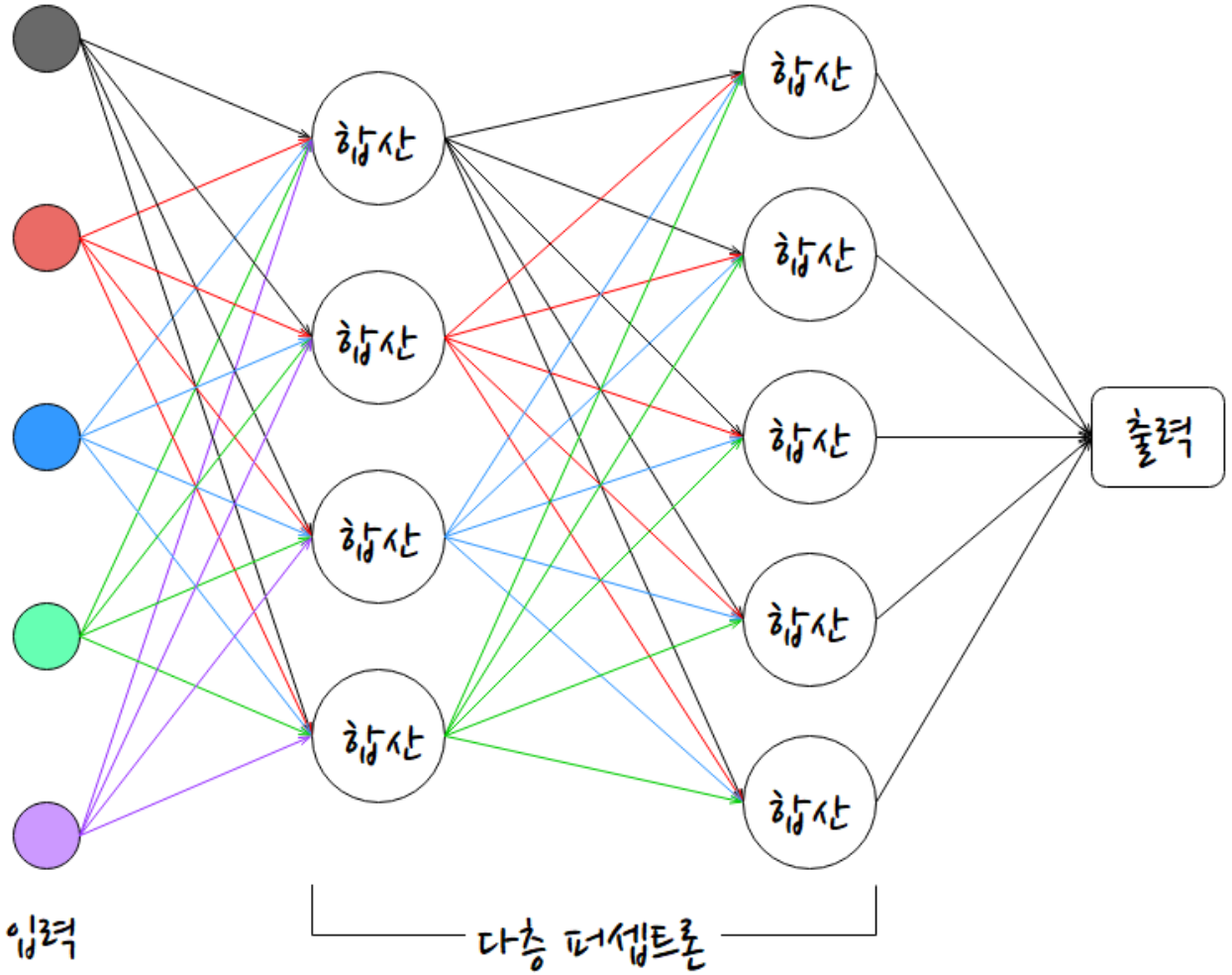
XOR 연산

그런데 이렇게 하면?



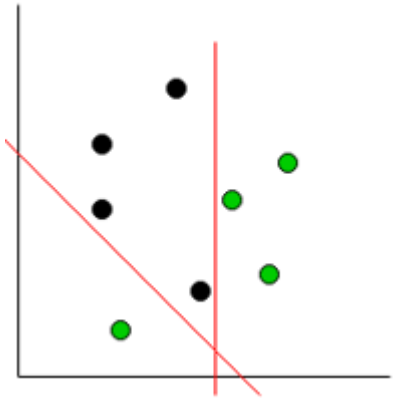
XOR 연산

다층 퍼셉트론(Multi Layer Perceptron)



단층 퍼셉트론의

가장 큰 문제 해결

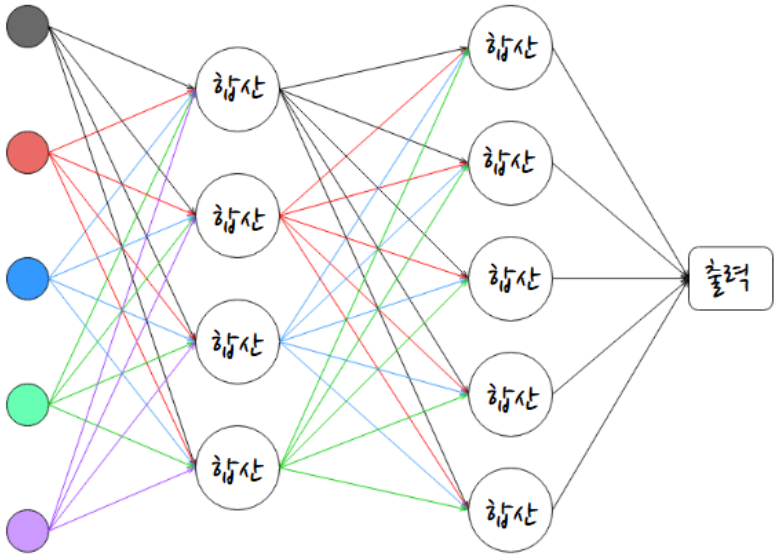


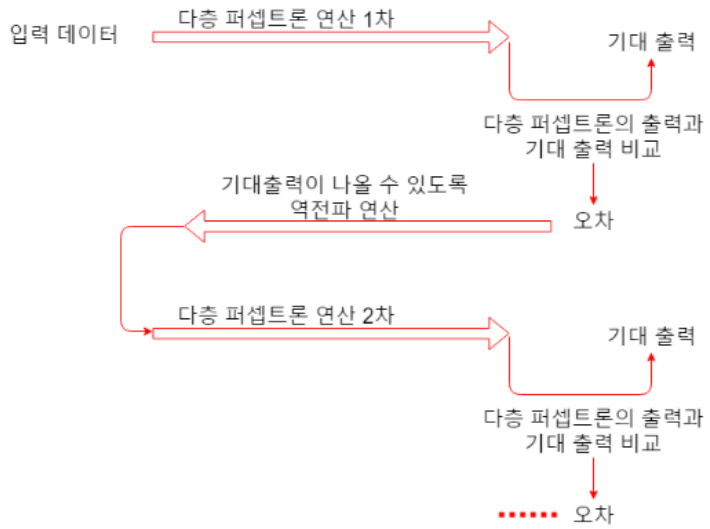
다층 퍼셉트론(Multi Layer Perceptron)

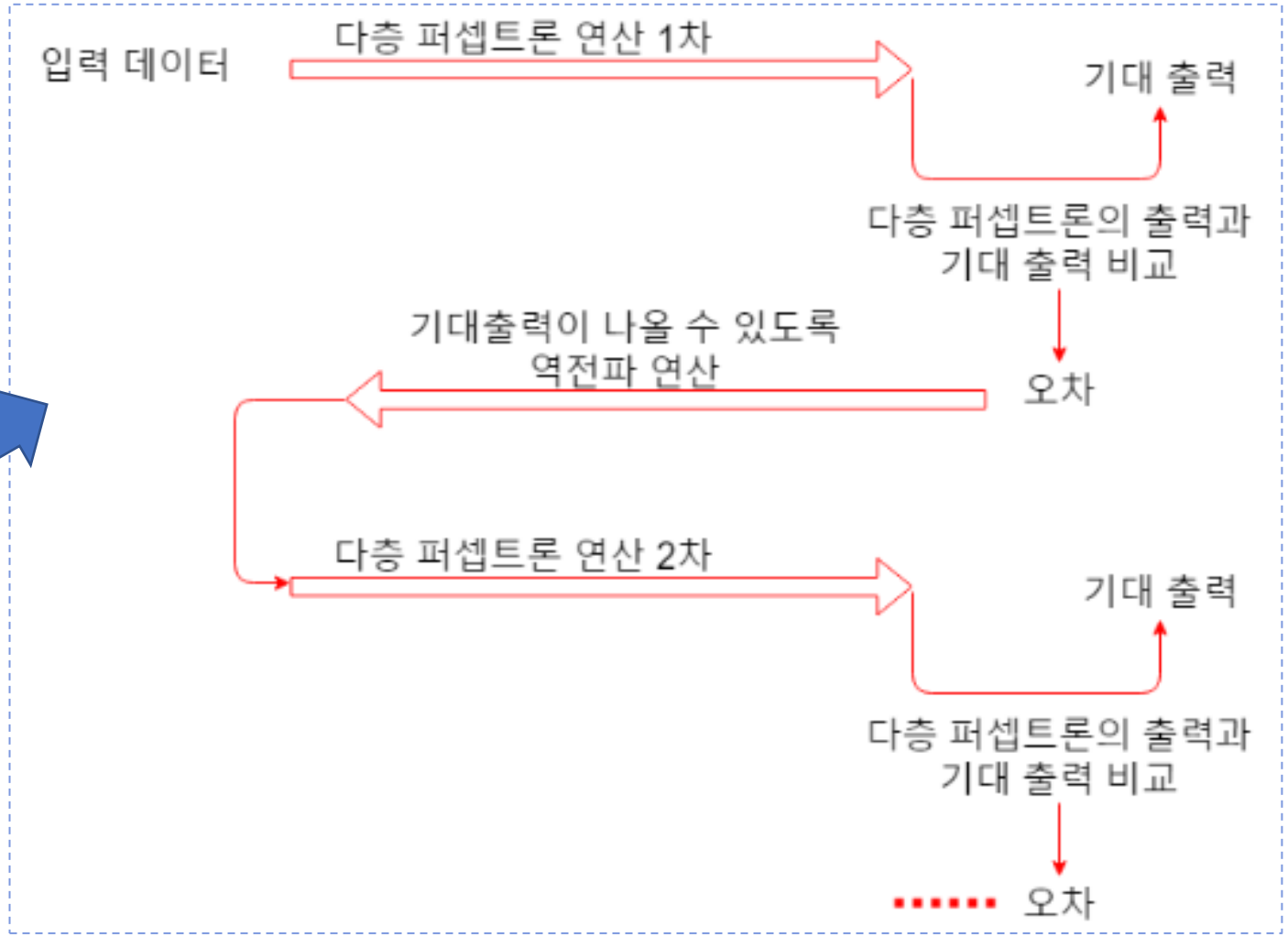
- 또 문제점
  - 단층 퍼셉트론과 마찬가지로 각 통로의 가중치를 변경할 방법이 없다
  - 한 번 생성되면 변경 불가능한 분류 알고리즘
  - 역시 학습의 개념이 없다 → 인공지능이 아닌 단순한 분류 알고리즘

- 해결책은?
  - 수행할 때마다 예전 데이터를 들고 와서 가중치를 수정해 주면 어떨까?
  - 그럼 아예 앞뒤로 왔다 갔다 반복하면서 가중치를 바꾸어 주면 어떨까?
  - Back Propagation (역전파) 알고리즘 제안

Back Propagation (역전파) 알고리즘

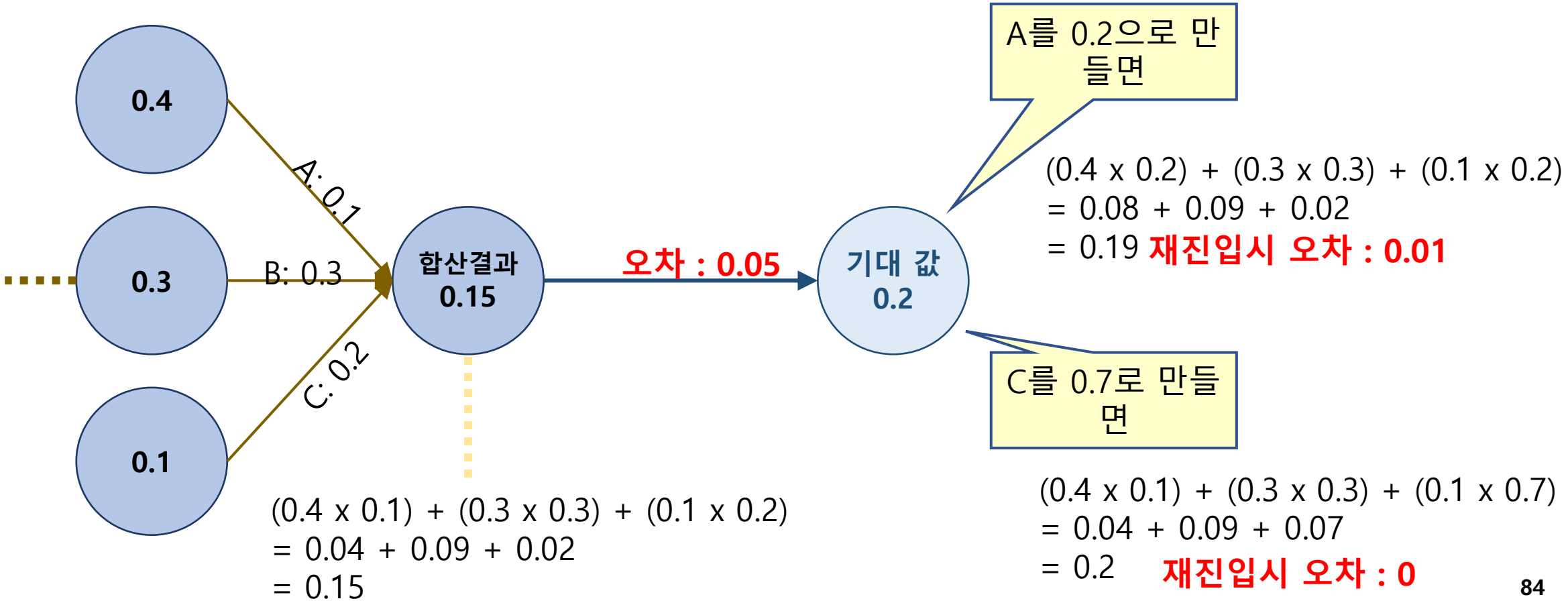






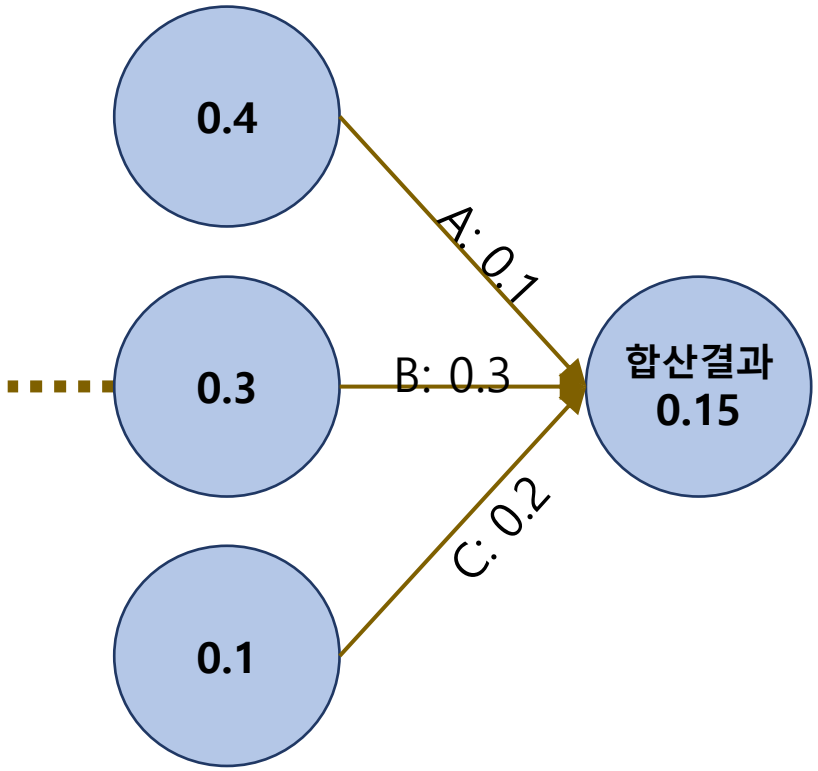
Back Propagation (역전파) 알고리즘

- 역전파 시 어떻게 가중치를 조절하는가?
  - 조절 내용: 은닉층을 거친 결과값과 기대한 결과값의 오차를 줄이는 방향으로 수정



Back Propagation (역전파) 알고리즘

- 역전파 시 어떻게 가중치를 조절하는가?
  - 조절 방법

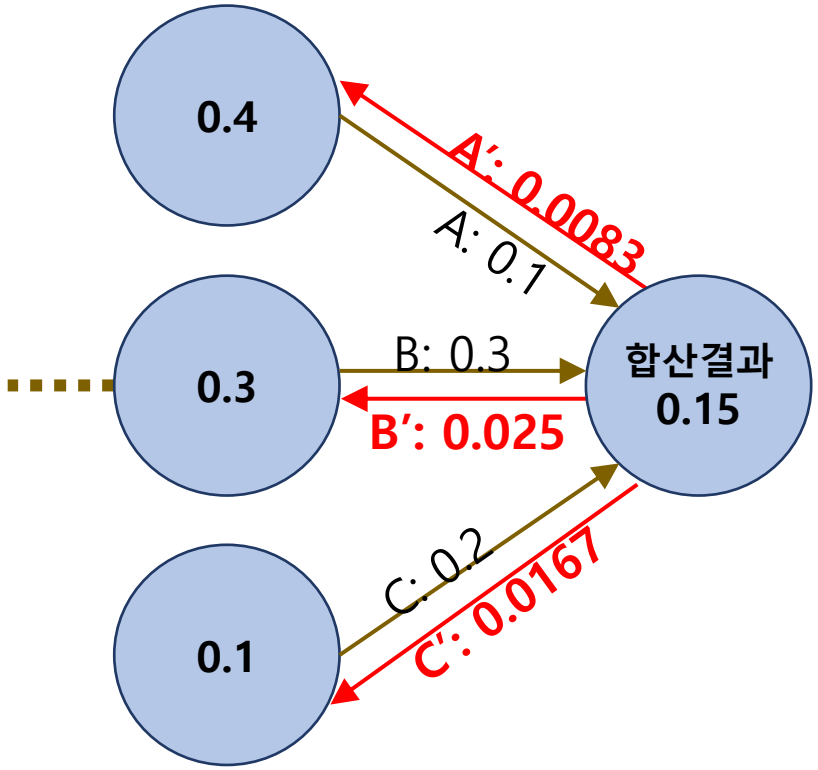


오차 값 0.05를 줄이기 위해서

- 가중치의 비율은 1 : 3 : 2
- 오차 0.05를 1 : 3 : 2 으로 나눈다
- A′ = 0.0083
- B′ = 0.025
- C′ = 0.0167
- A′ + B′ + C′ = 0.05

Back Propagation (역전파) 알고리즘

- 역전파 시 어떻게 가중치를 조절하는가?
  - 조절 방법

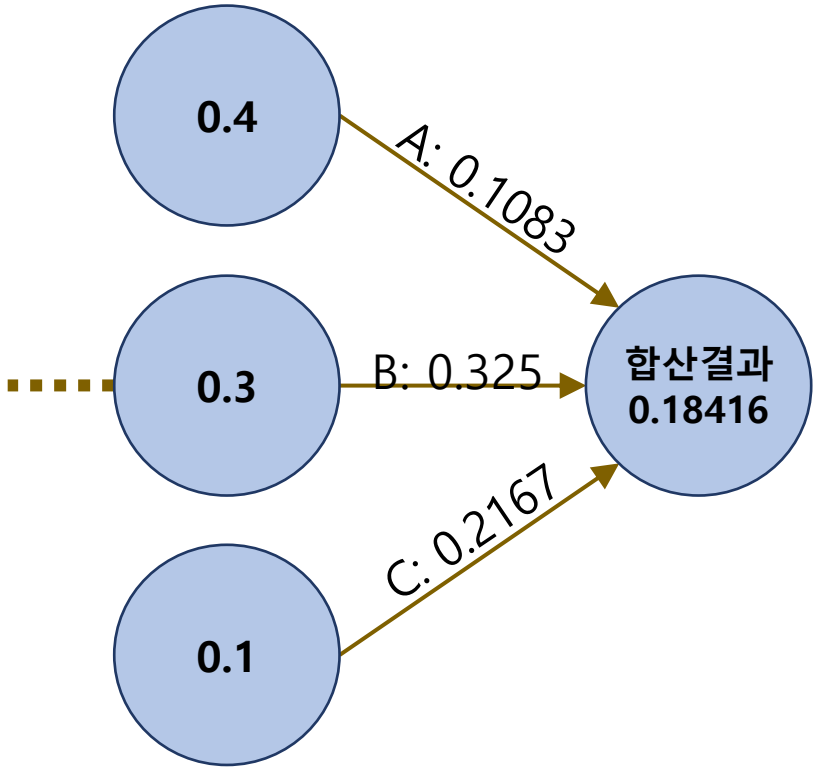


되돌려준 후 반영한다

- A = 0.1 + 0.0083 = 0.1083
- B = 0.3 + 0.025 = 0.325
- C = 0.2 + 0.0167 = 0.2167

Back Propagation (역전파) 알고리즘

- 역전파 시 어떻게 가중치를 조절하는가?
  - 조절 방법



반영 후 다시 계산

- A = 0.1 + 0.0083 = 0.1083
- B = 0.3 + 0.025 = 0.325
- C = 0.2 + 0.0167 = 0.2167
- 0.4 x A + 0.3 x B + 0.1 x C

  = 0.04332 + 0.0975 + 0.04334 = 0.18416

→ 기대 값 0.2와 비교하여 → 오차 0.01584

→ 기존의 오차 0.05보다 줄어듦

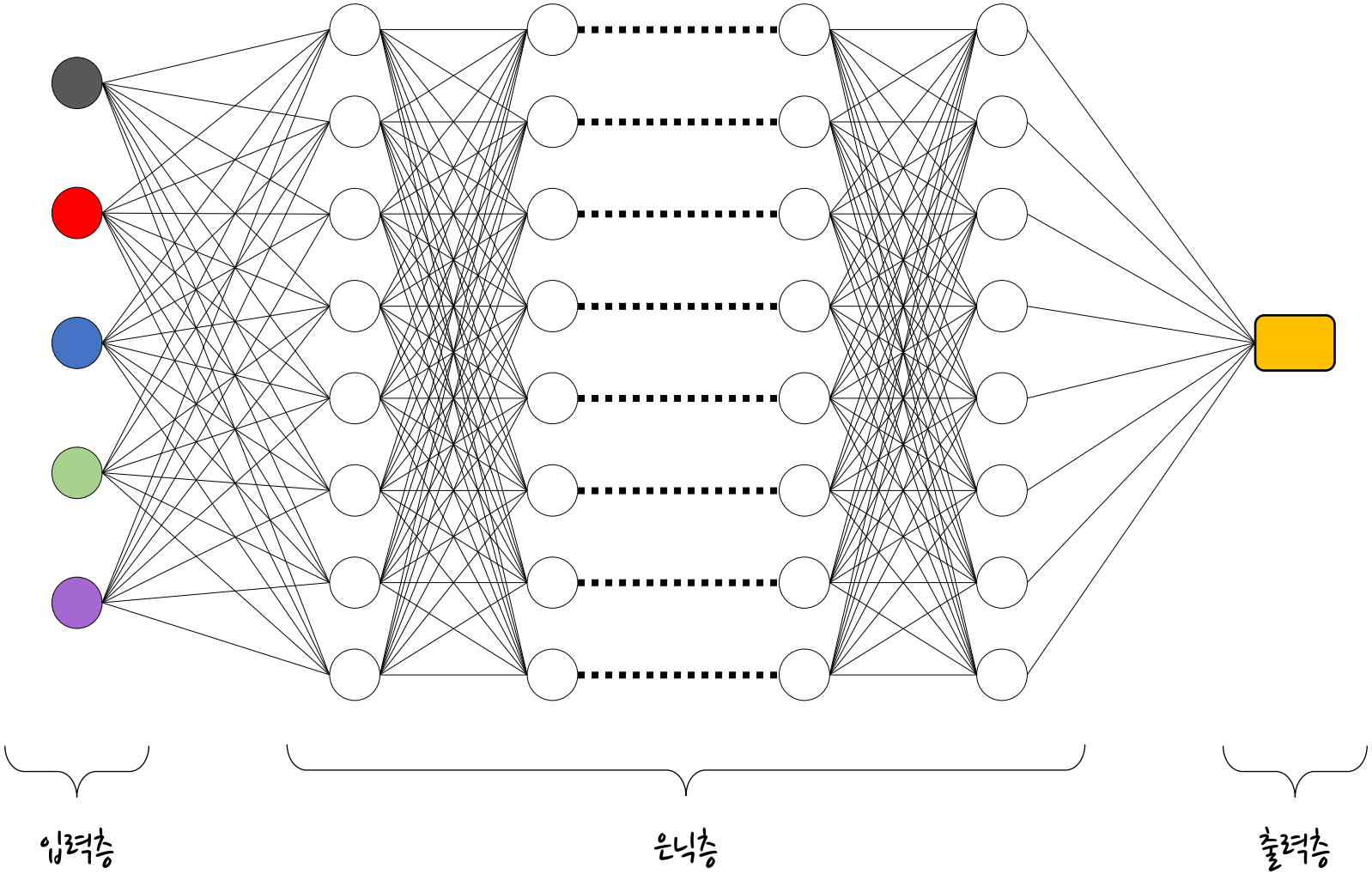
Back Propagation (역전파) 알고리즘

- Back Propagation (역전파) 알고리즘을 구현한 다층 신경망에서
- 그 층을 훨~씬 많이 만들어서
- 수 많은 분류 작업을 수행할 수 있게 한다면?
- 다층 신경망(Multilayer Neural Network)

  → 심층 신경망(Deep Neural Network) 으로 진화

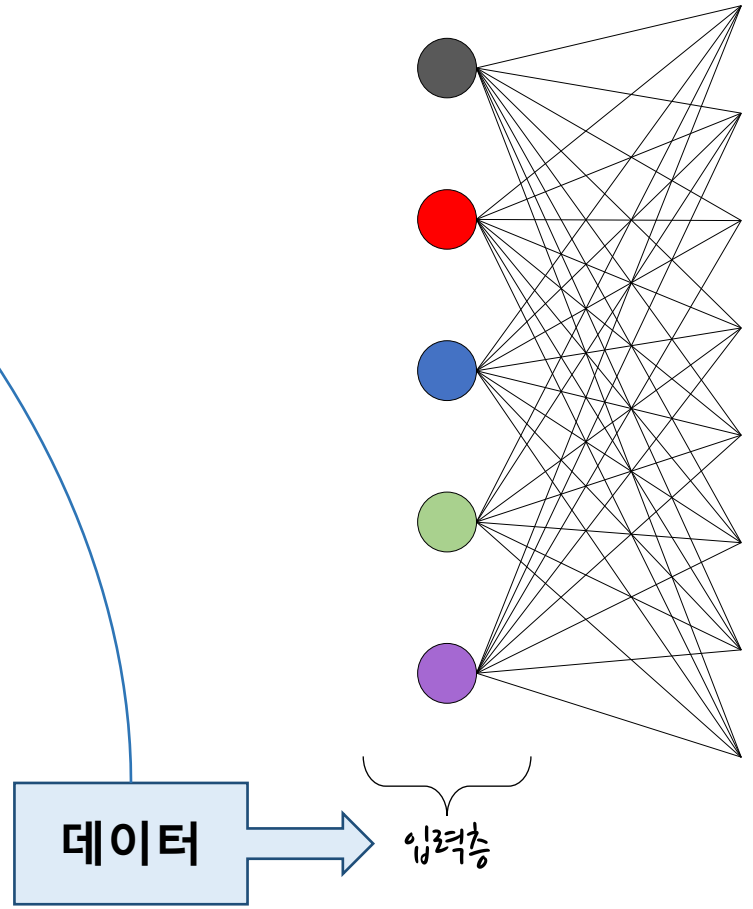
- 심층 신경망을 이용한 학습 모델 = 딥 러닝(Deep Learning)

가장 기본적인 딥러닝 모델: 전체 구조



가장 기본적인 딥러닝 모델: 입력층

어떤 데이터들이 입력되는가?

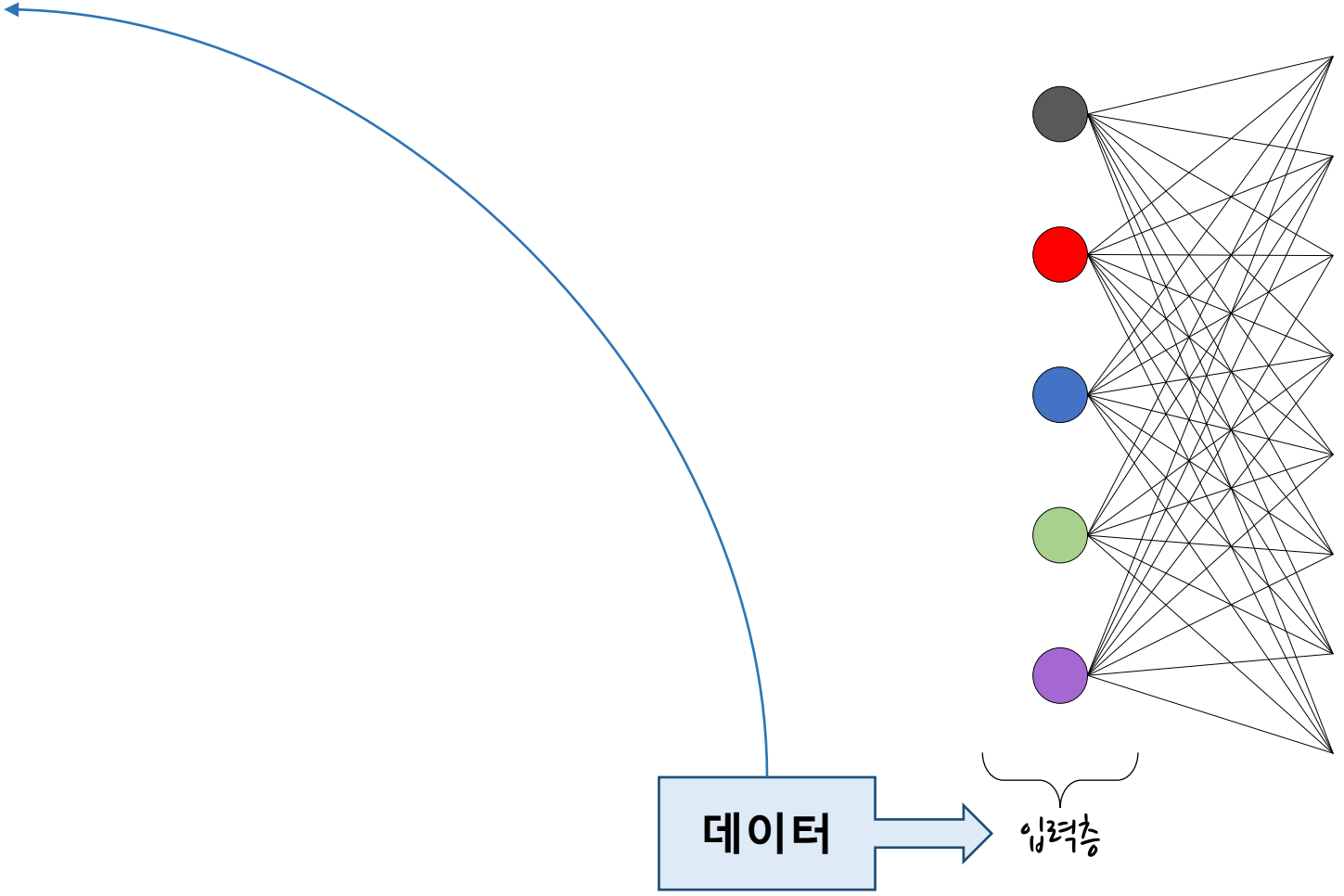


가장 기본적인 딥러닝 모델: 입력층

어떤 데이터들이 입력되는가?

- 이미지(사진) 데이터
- 동영상 데이터
- 센서 데이터
- 주식 데이터
- 기상관측 데이터
- 등...

- SNS 데이터
- Web Scraping 데이터
- 등...



가장 기본적인 딥러닝 모델: 입력층

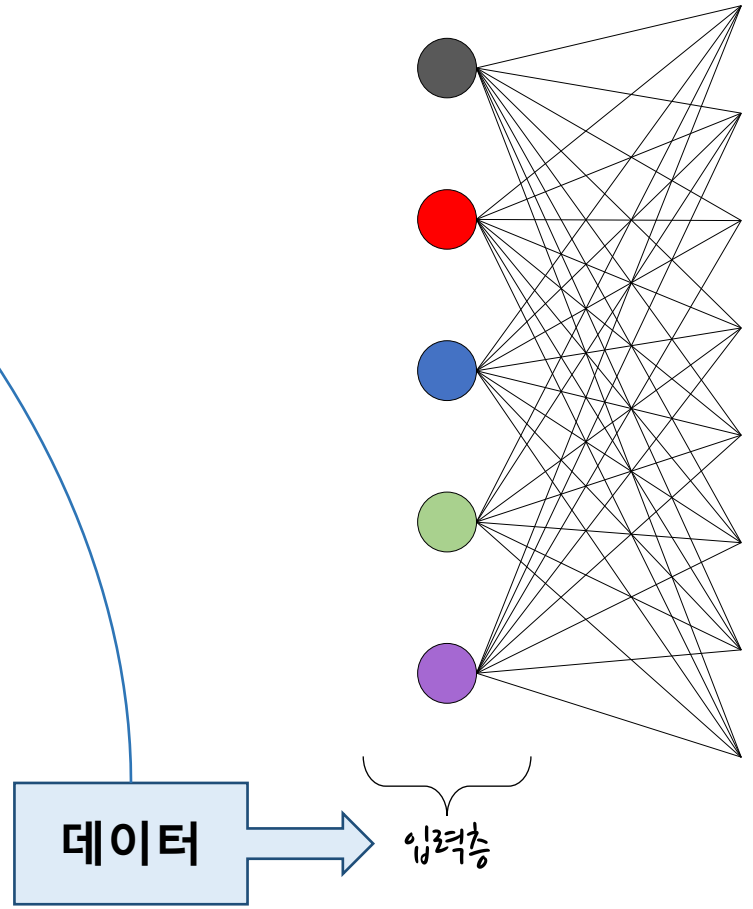
어떤 데이터들이 입력되는가?

- 이미지(사진) 데이터
- 동영상 데이터
- 센서 데이터
- 주식 데이터
- 기상관측 데이터
- 등…

수치 데이터

- SNS 데이터
- Web Scraping 데이터
- 등…

문자(열) 데이터 → 수치화



가장 기본적인 딥러닝 모델: 입력층

어떤 데이터들이 입력되는가?

- 이미지(사진) 데이터
- 동영상 데이터
- 센서 데이터
- 주식 데이터
- 기상관측 데이터
- 등…

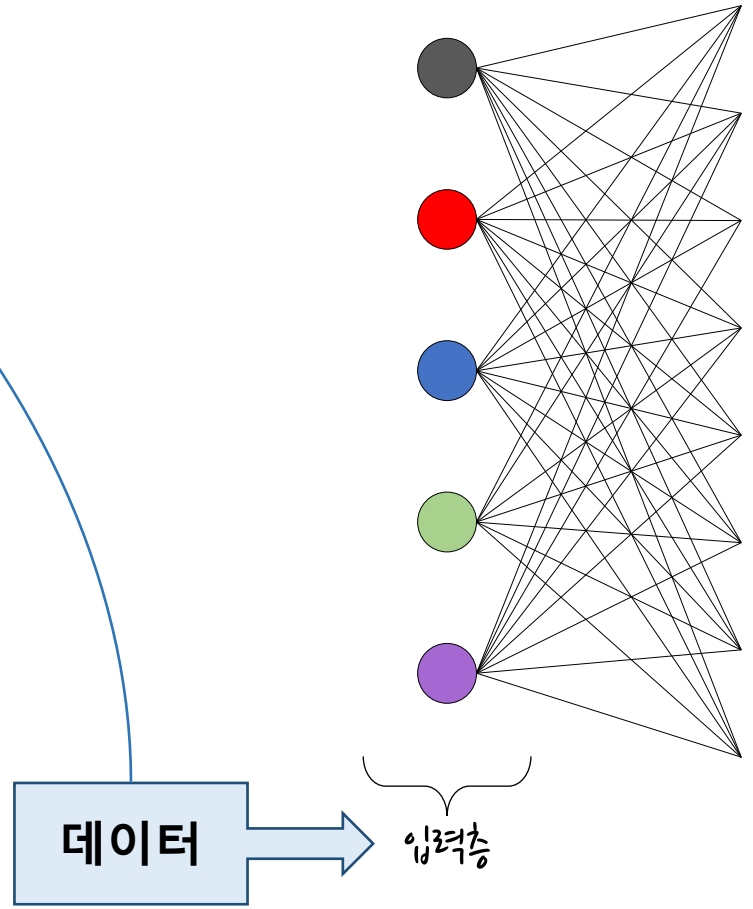
수치 데이터

- SNS 데이터
- Web Scraping 데이터
- 등…

문자(열) 데이터 → 수치화

딥러닝에서 사용되는 데이터는

기본적으로 수치 데이터



가장 기본적인 딥러닝 모델: 입력층

어떤 데이터들이 입력되는가?

- 이미지(사진) 데이터
- 동영상 데이터
- 센서 데이터
- 주식 데이터
- 기상관측 데이터
- 등…

수치 데이터

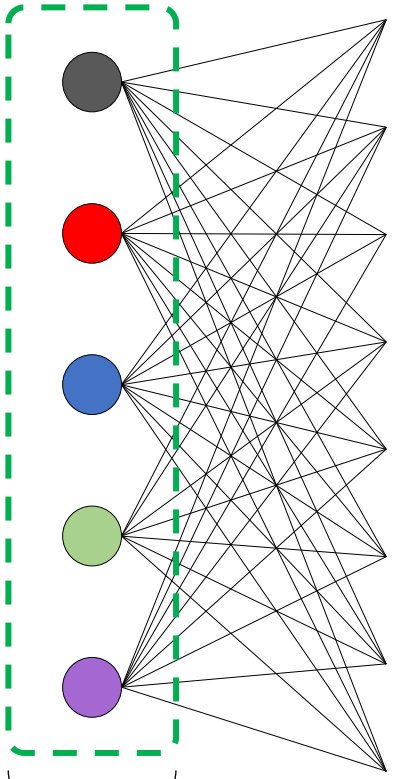
- SNS 데이터
- Web Scraping 데이터
- 등…

문자(열) 데이터 → 수치화

딥러닝에서 사용되는 데이터는

기본적으로 수치 데이터

각 노드에 하나씩 입력
→ 한 줄로 이어진 데이터



데이터 → 입력층

가장 기본적인 딥러닝 모델: 입력층

어떤 데이터들이 입력되는가?

- 이미지(사진) 데이터
- 동영상 데이터
- 센서 데이터
- 주식 데이터
- 기상관측 데이터
- 등…

수치 데이터

- SNS 데이터
- Web Scraping 데이터
- 등…

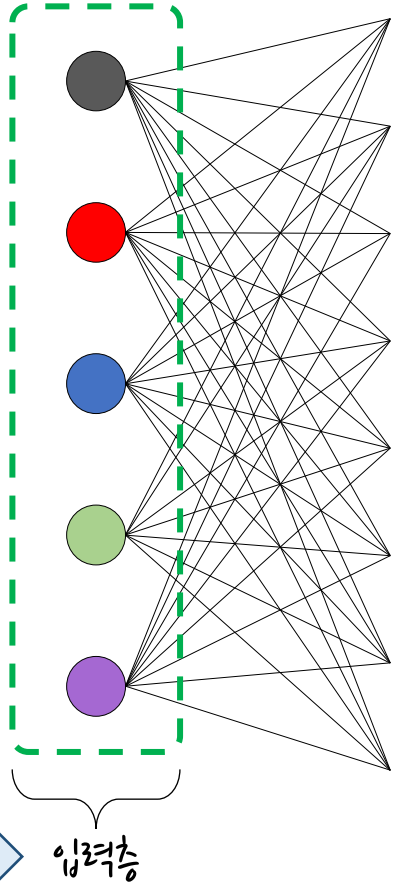
문자(열) 데이터 → 수치화

딥러닝에서 사용되는 데이터는

기본적으로 수치 데이터

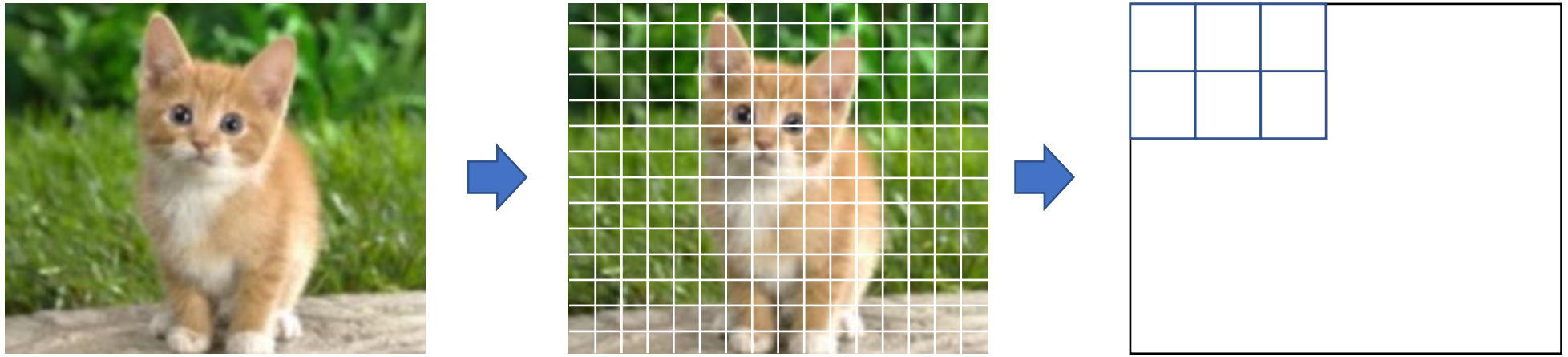
1차원 배열로 이루어진 수치 데이터

각 노드에 하나씩 입력
→ 한 줄로 이어진 데이터



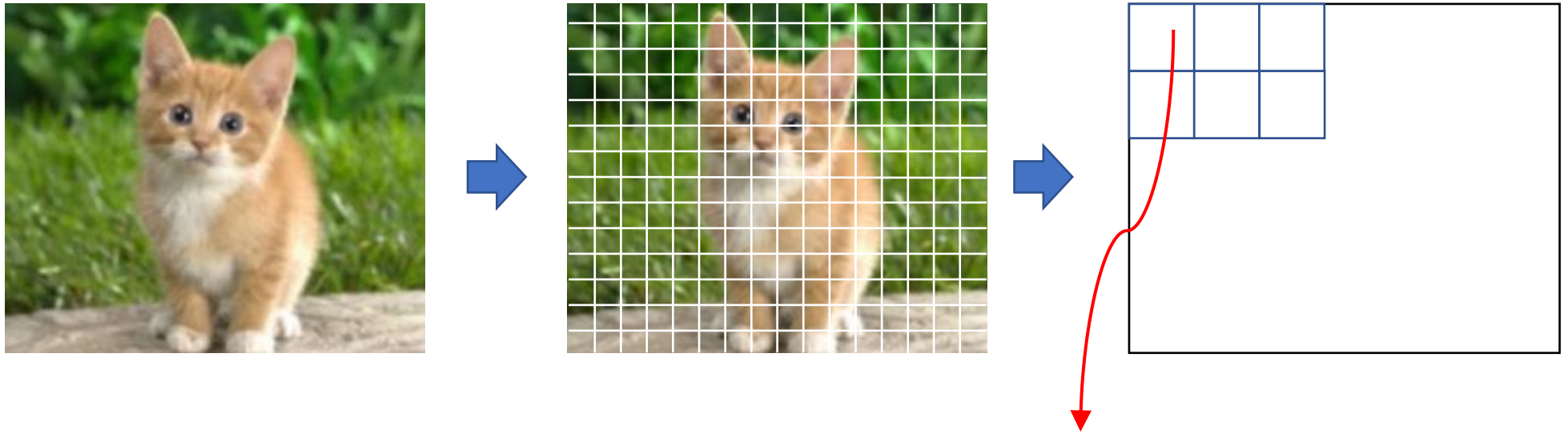
가장 기본적인 딥러닝 모델: 입력층

- 이미지 / 영상 데이터가 왜 수치 데이터인가?
  - 이미지 데이터는 색깔을 가진 수많은 점이 가로x세로 크기의 2차원 배열 속에 모인 데이터



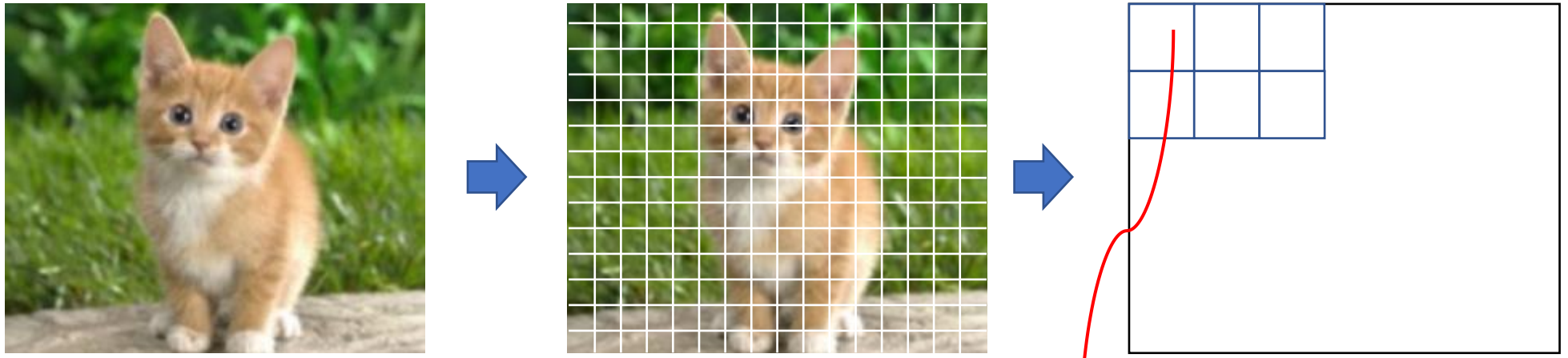
가장 기본적인 딥러닝 모델: 입력층

- 이미지 / 영상 데이터가 왜 수치 데이터인가?
  - 이미지 데이터는 색깔을 가진 수많은 점이 가로x세로 크기의 2차원 배열 속에 모인 데이터



가장 기본적인 딥러닝 모델: 입력층

- 이미지 / 영상 데이터가 왜 수치 데이터인가?
  - 이미지 데이터는 색깔을 가진 수많은 점이 가로x세로 크기의 2차원 배열 속에 모인 데이터



배열 데이터를 1차원으로 변환하여 입력 데이터로 사용

32832 = #008040 = #00 #80 #40 =

R G B

동영상 데이터는 다수의 이미지가 순서대로 연결된 것

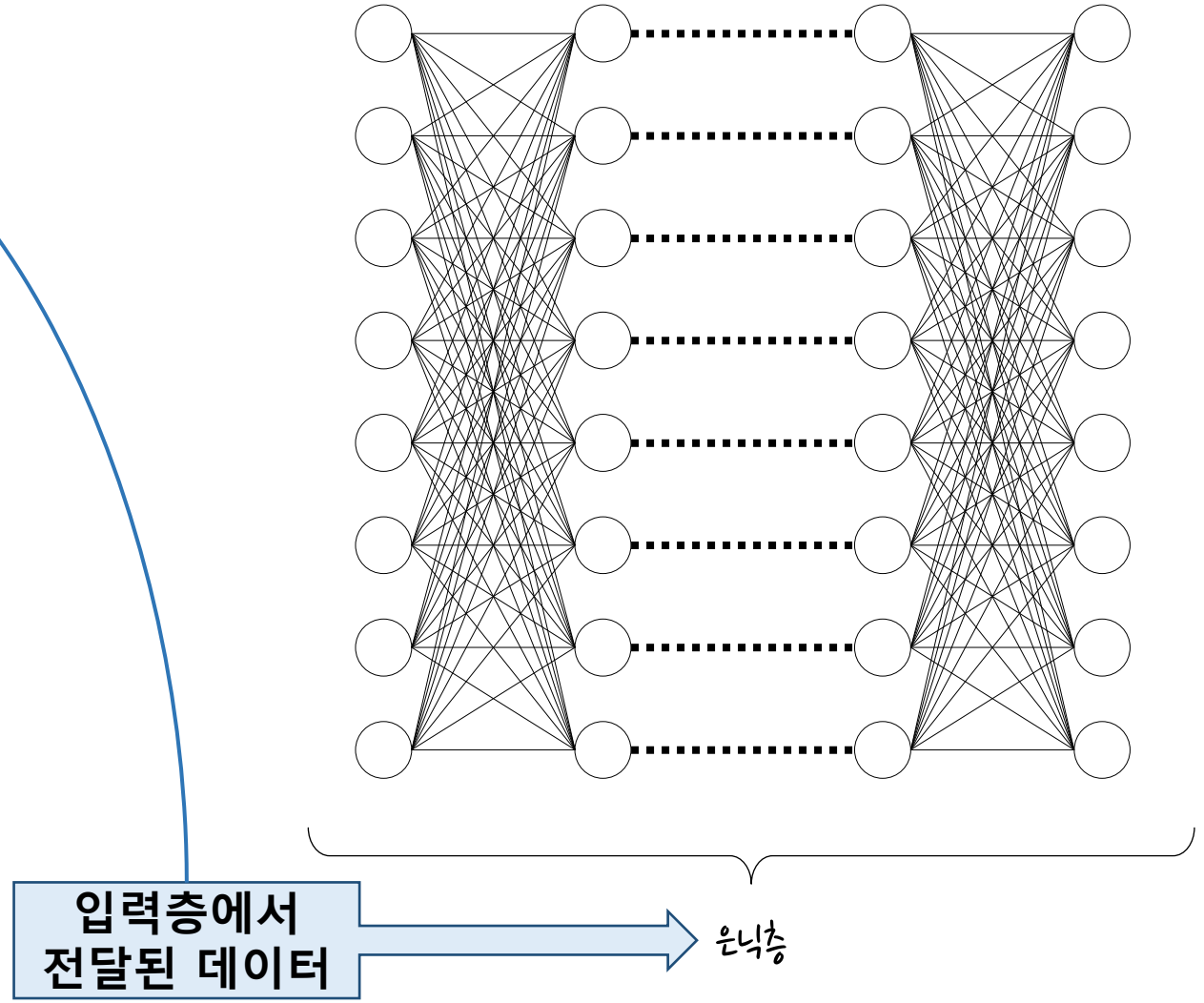
가장 기본적인 딥러닝 모델: 은닉층

데이터는 어떻게 전달되는가?

- 입력층에서 입력된 데이터는
- 각 데이터가 첫 번째층의 모든 뉴런에
- 동일하게 전달됨

- 각 층의 모든 뉴런이 가진 데이터는
- 다음 층의 모든 뉴런에
- 동일하게 전달됨

실제의 신경망에서는 신경전달물질이 퍼져 나가는 범위 안의 신경세포에만 신호가 전달됨 (모델링의 한계)

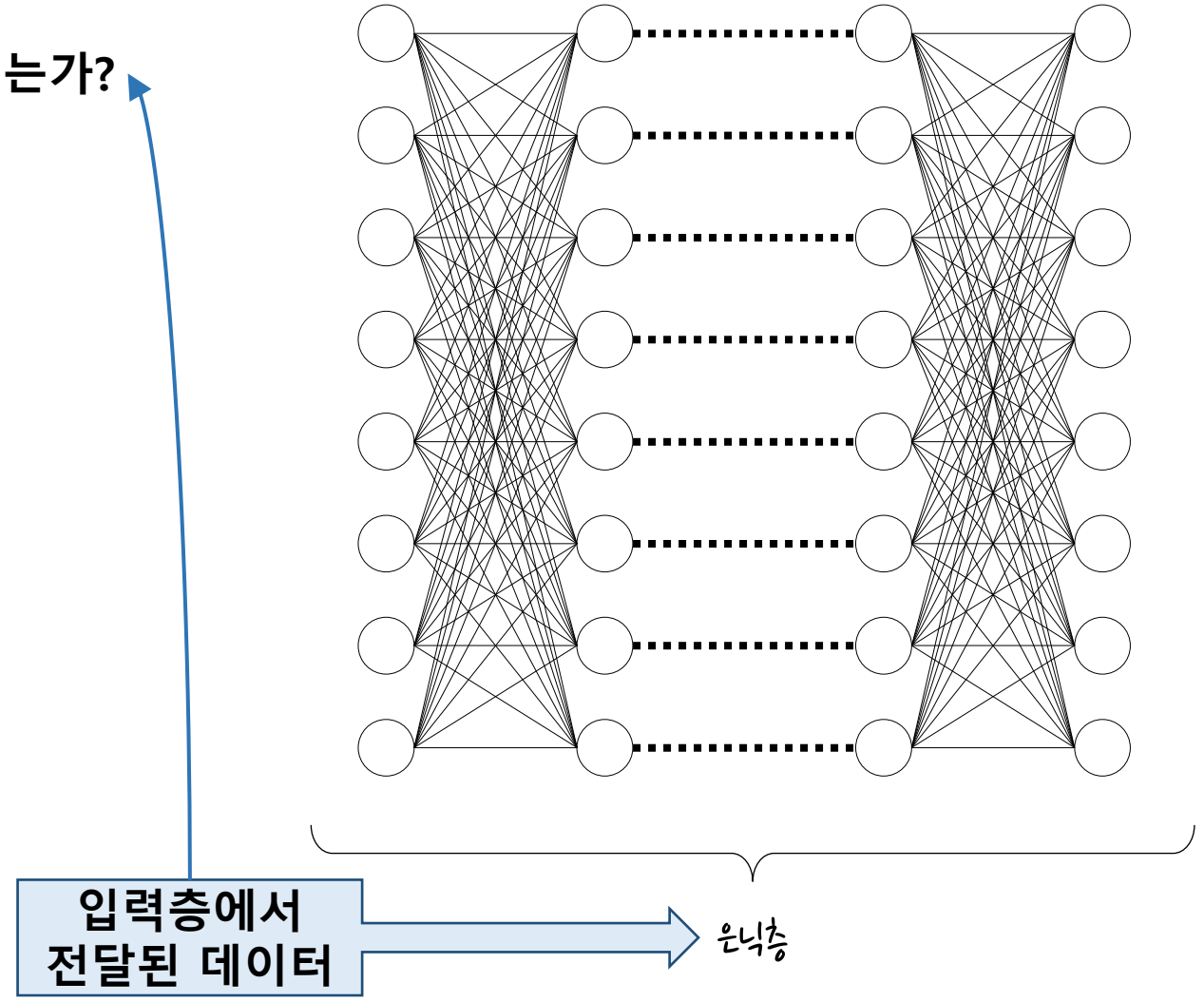


가장 기본적인 딥러닝 모델: 은닉층

각 뉴런에 전달된 데이터는 어떻게 가공되는가?

- 각 층의 뉴런은
- 입력되는 모든 데이터를
- 다 더한 후(합산)

- 합산 결과를 활성화 함수에 적용하고
- 활성화 함수(F1) 적용 결과를
- 다음 층의 뉴런으로 전달

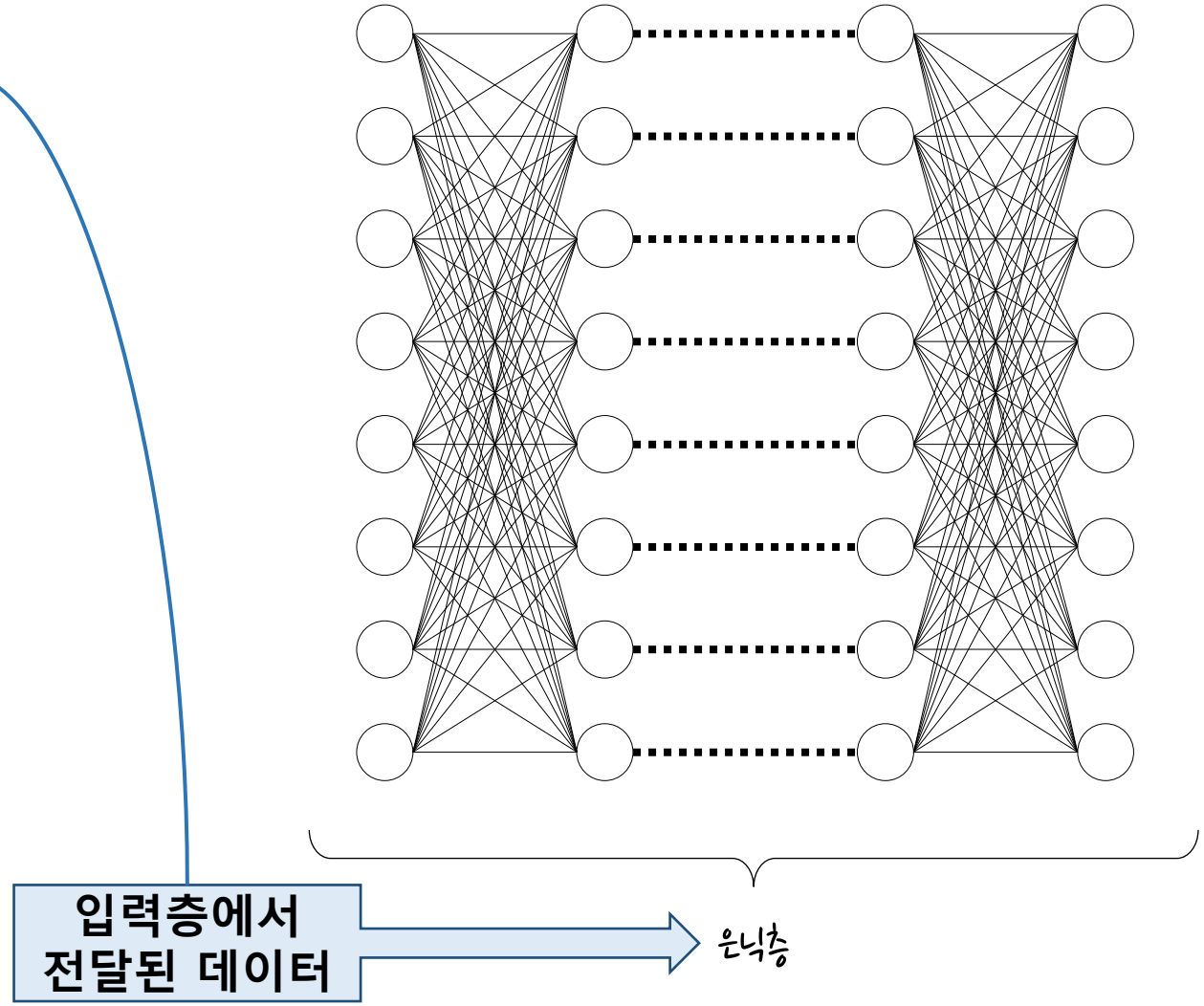


가장 기본적인 딥러닝 모델: 은닉층

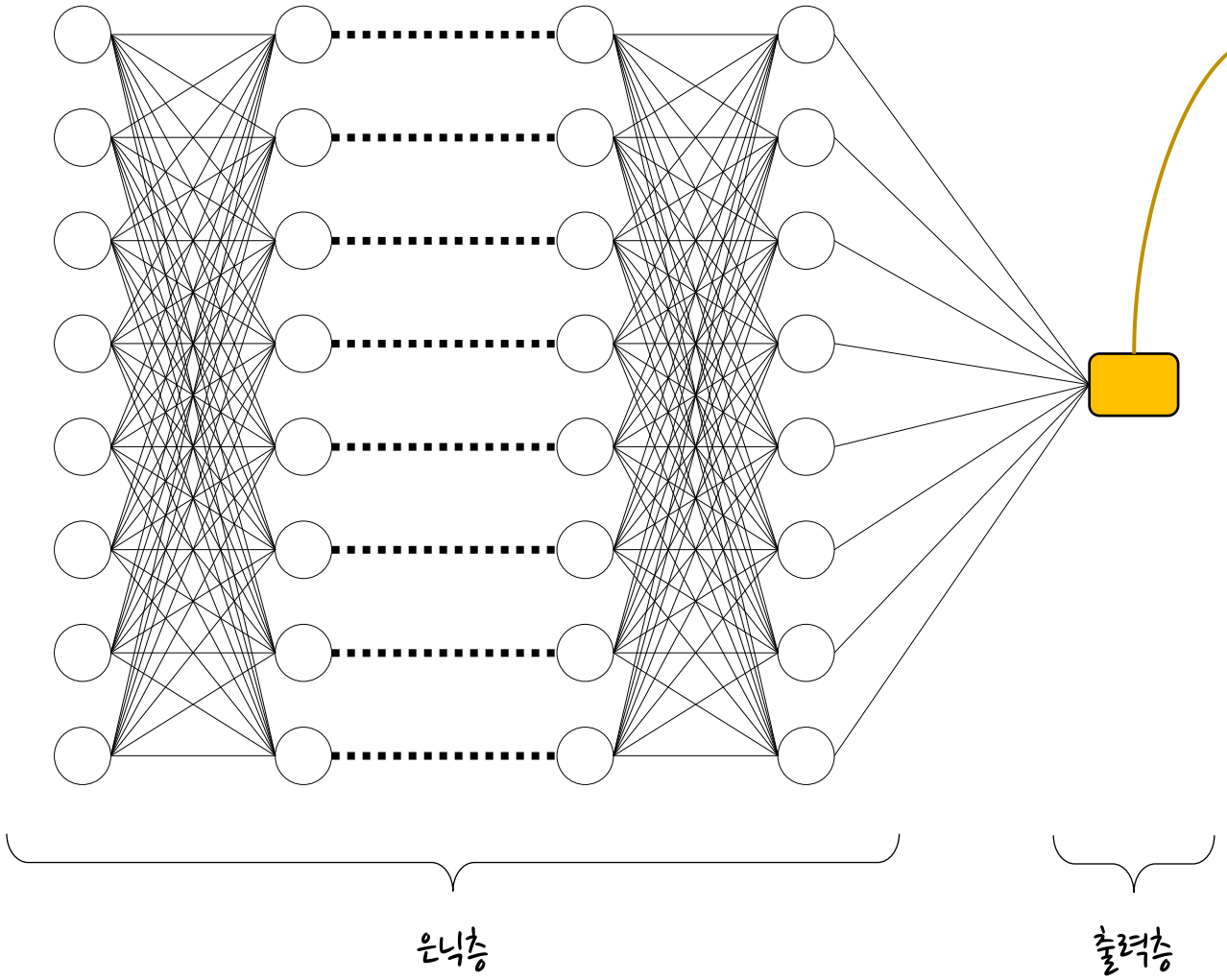
은닉층은 데이터에 어떤 작업을 하는가?

- 각 층에 입력된 데이터는
- 다음 층의 뉴런으로 가는 경로에 설정된 가중치를
- 현재의 뉴런이 가진 데이터에 곱하여
- 그 결과를 다음 층의 뉴런으로 전달

- 모든 층의 데이터 이동에서
- 가중치의 곱셈과 입력 값의 합산은
- 동일하게 적용됨



가장 기본적인 딥러닝 모델: 은닉층

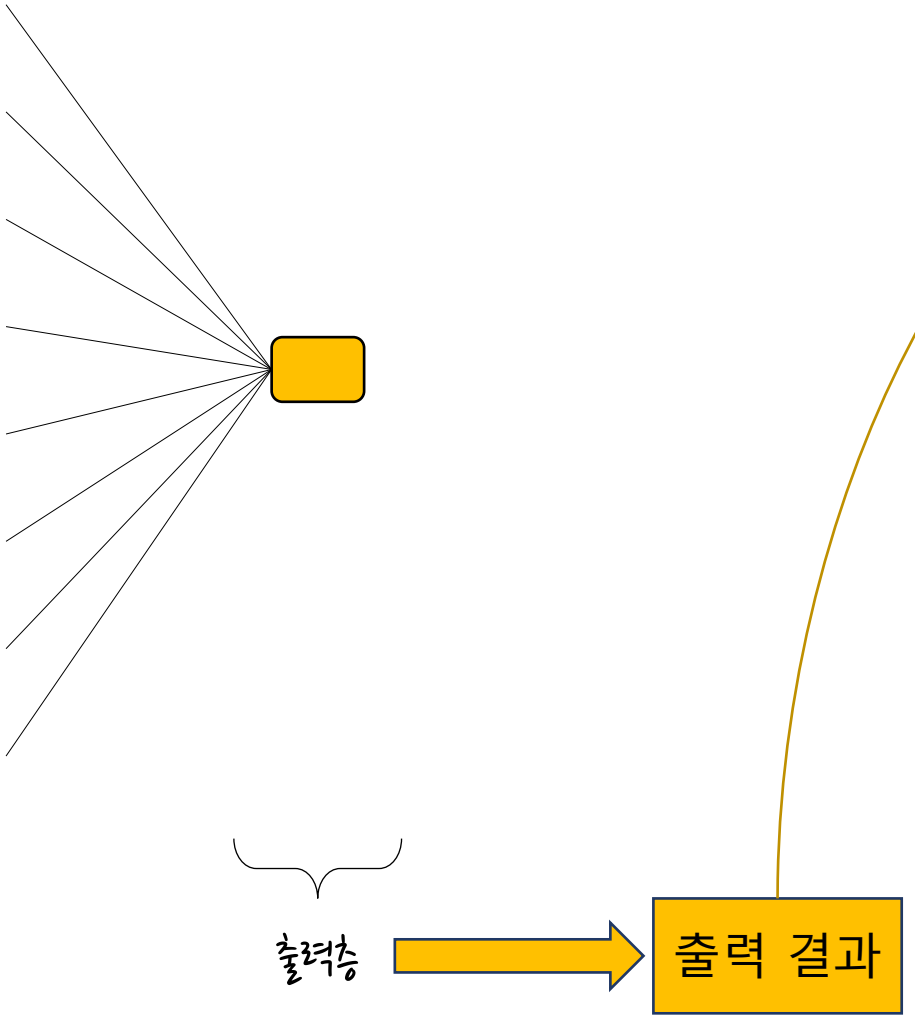


은닉층과 출력층은 함께 어떤 작업을 하는가?

- 은닉층의 마지막 층의 뉴런들은
- 자기에게 입력된 데이터를
- 합산 후
- 활성화 함수(F1)를 적용하고
- 적용 결과를
- 출력층의 출력 뉴런으로 전달

- 출력 뉴런은
- 전달된 모든 데이터를
- 순서대로 나열하여(벡터화)
- 활성화 함수(F2)를 적용하고
- 적용 결과(벡터값)를
- 결과로서 출력함

가장 기본적인 딥러닝 모델: 출력층



출력층에서는 어떤 결과가 나오는가?

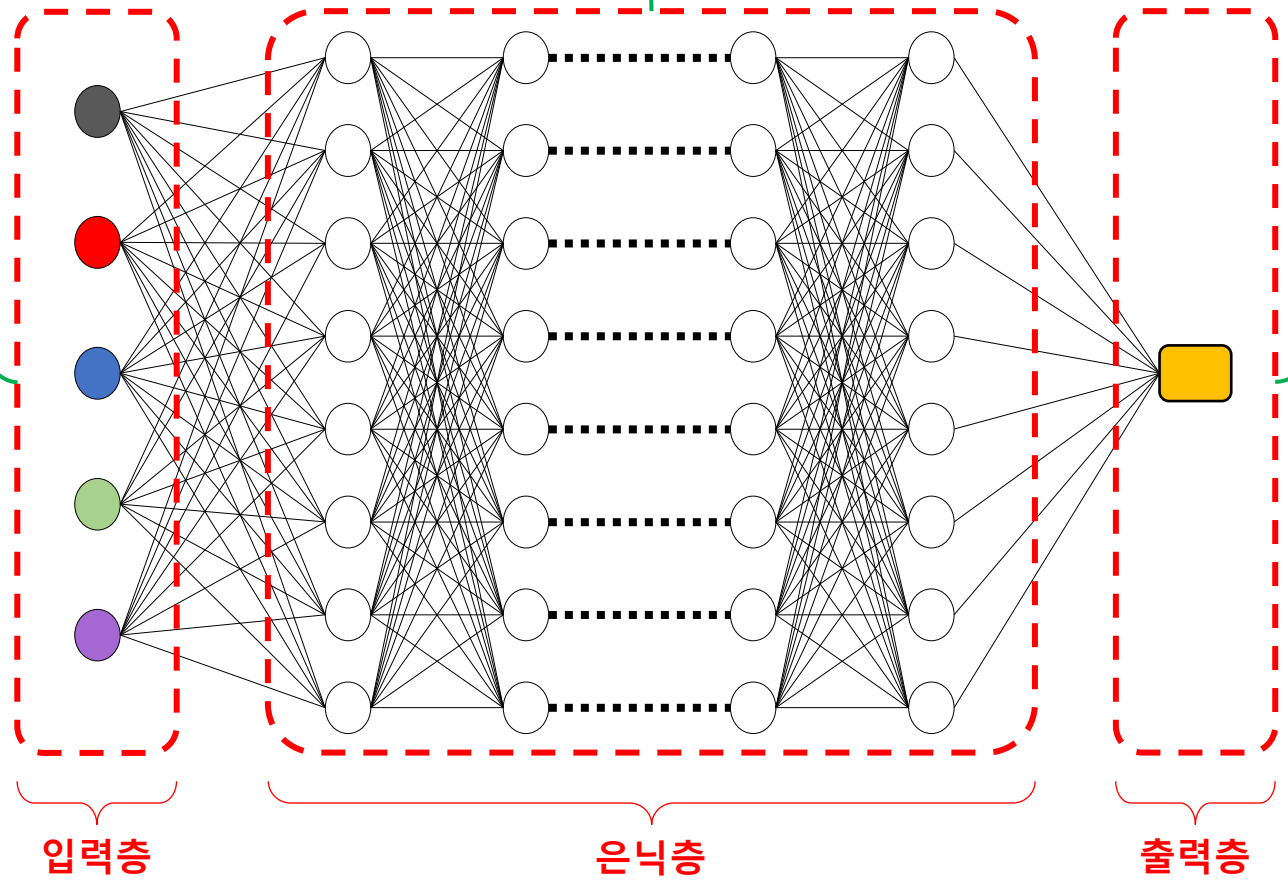
- 출력 뉴런은
- 전달된 모든 데이터를
- 순서대로 나열하여(벡터화)
- 활성화 함수(F2)를 적용하고
- 적용 결과(벡터값)를
- 결과로서 출력함

- 출력층의 결과는
- 미리 입력된 기대 결과값과 비교하여(지도학습)
- 일치하면 → 학습 완료
- **일치하지 않으면 → 진행의 반대 방향으로 다시 전달**

딥러닝 기본 모델

은닉층에서는 데이터의 연산(가중치x데이터, 합산, 활성화)과
역전파에 의한 오류 보정(가중치 조정) 작업 수행

입력 데이터는
수치로 구성된
1차원 배열



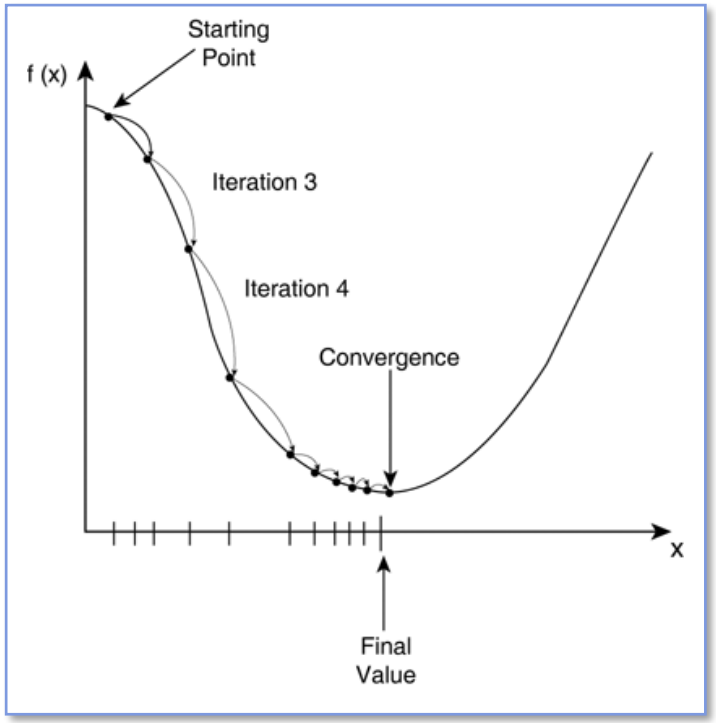
출력 데이터로
오차 검증.
오차발생 시 역전파,
통과 시 결과 도출

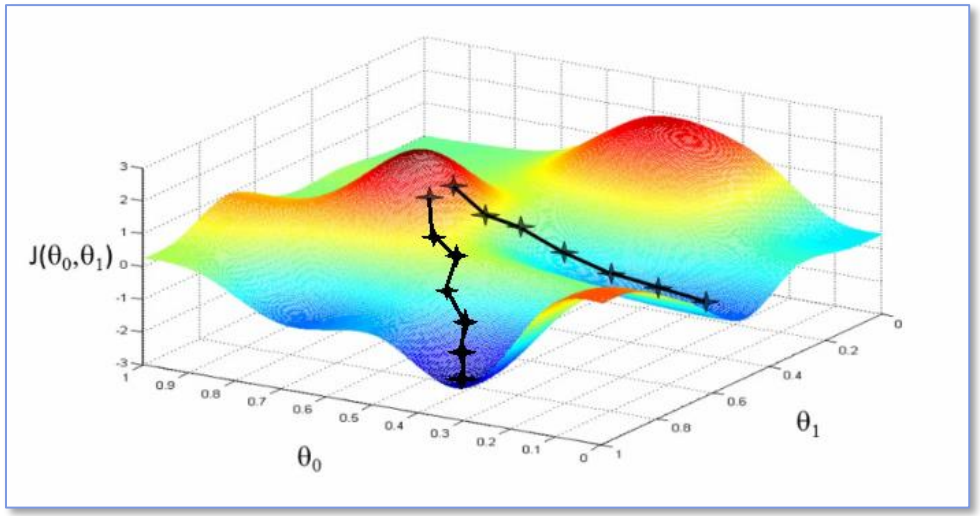
입력층 은닉층 출력층

딥러닝 기본 모델-가중치의 조정

- 가장 많이, 기본적으로 사용되는 가중치 조정 방법
  - 경사 하강법 (Gradient Descent)

초기값에 따라
최저점이
달라질 수 있다



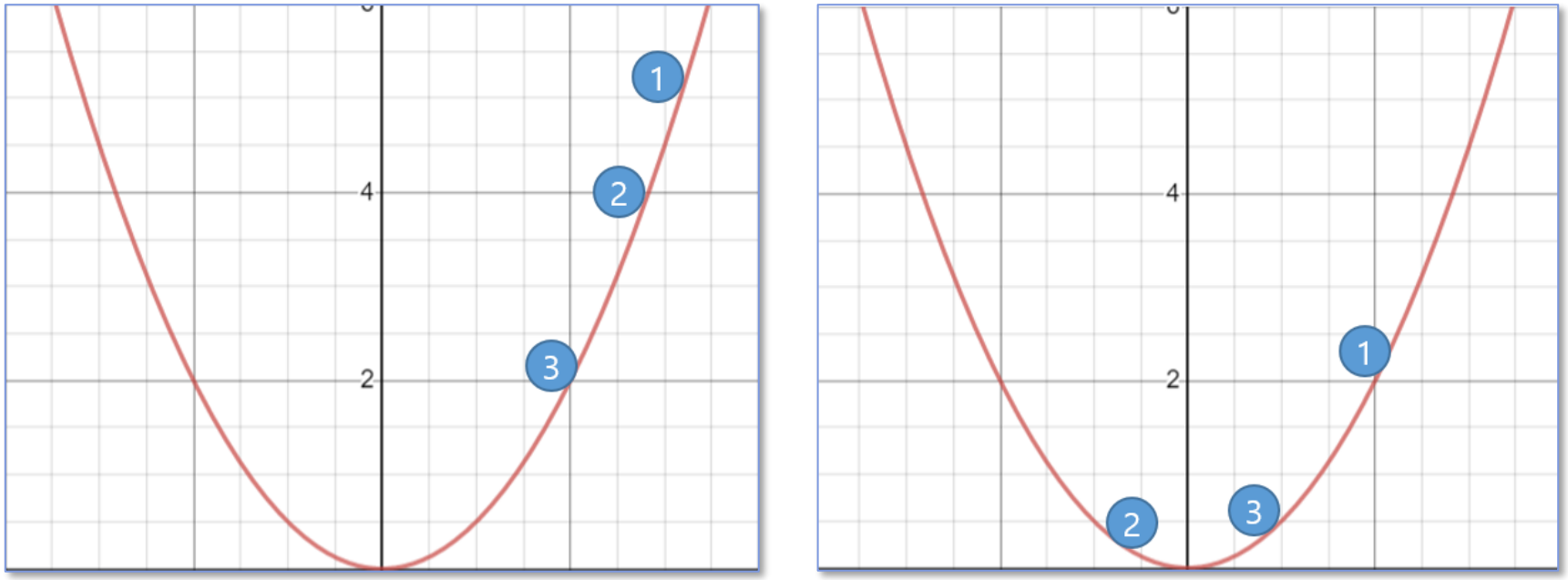


최저점은 하나가 아니다

이런 이유로...
같은 목적과 같은 데이터로 학습하더라도
모든 학습된 모델의 내부(가중치 집합)는 모두 다름

딥러닝 기본 모델-가중치의 조정

- 그 외의 가중치 조절 방법들..
  - SGD, Momentum, AdaGrad, RMSprop, Adam 등



공이 굴러가듯이 움직인다

어떻게 굴러가는지, 어디에서 튀어 오르는지, 관성의 적용을 어떻게 받는지... 등

이런 특징들을 활용 또는 참고하여 새로운 방식을 고려할 수도 있다

딥러닝 기본 모델-가중치의 조정

- 가중치 계산 방법으로 사용할 수 없는 것은?
  - 딥러닝 모델에서의 가중치 계산, 조정은 미분의 개념을 기반으로 움직임
  - 부드럽게 이어지지 않고 뾰족하거나 각진 형태로 인하여 미분이 불가능한 그래프의 형태를 가지는 계산 모델은 사용할 수 없음

딥러닝 기본 모델-활성화 함수

- **그럼 여기서... 각 퍼셉트론에서 임계 값을 넘으면 활성화는 어떻게 하나?**
- **활성화 함수(Activation Function)**
  - **신경망을 구성하는 각 퍼셉트론에서 임계 값을 넘었을 때 출력을 처리하는 함수**
  - **함수의 정의**
    - 입력: 이전 층의 디바이스 또는 퍼셉트론들로부터 전달되는 데이터
    - 함수의 동작: 입력 값의 합산 + 합산결과와 임계 값의 비교 + 출력 결정 (활성화 조건)
    - 출력: 퍼셉트론 층의 연산 결과값. 다음 층의 뉴런에 대한 입력 또는 최종 층의 출력

딥러닝 기본 모델-활성화 함수

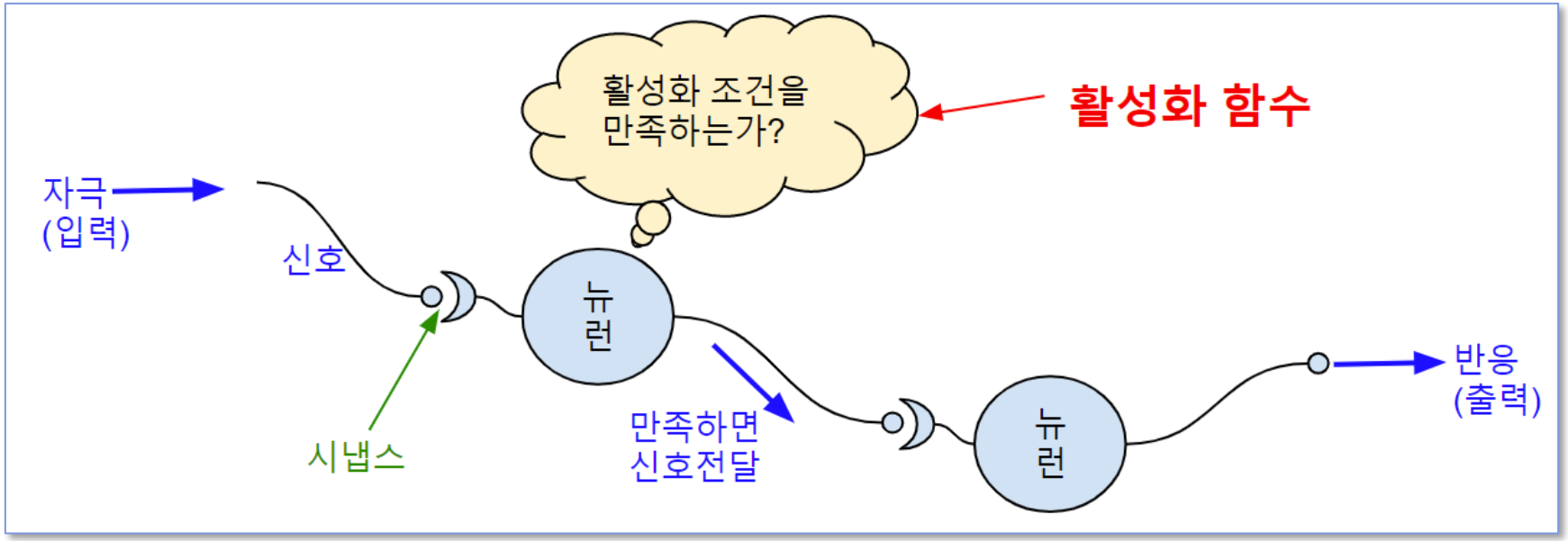
- 그런데 활성화 함수는 왜 필요한가?
  - 피부, 눈과 같은 감각기관이 어떤 자극을 받아 신호를 발생시키면
  - 그 신호는 축삭을 통해서 이동하고
  - 축삭의 말단에 있는 시냅스를 거쳐
  - 다음 뉴런으로 전달

딥러닝 기본 모델-활성화 함수

- **그런데!!!**
  - **전달되는 모든 신호(아주 미세한 신호부터 강한 신호까지)를 모두 다음 뉴런으로 전달한다면?**
    - 생활 자체가 어려워짐
    - 매우 비 효율적
  - **우리 몸에서 반응할 필요가 있는 수준까지만 신호를 전달하고 나머지의 신호는 무시한다!!**
    - 진화의 결과
    - 이 기준을 모델에 반영한 것이 활성화 함수

딥러닝 기본 모델-활성화 함수

- 즉, 활성화 함수란
  - 뉴런의 신호 흐름을 모델링 할 때 각각의 뉴런에 제한을 걸어 둔 것
  - 활성화 함수에서 적용한 기준에 따라 조건을 만족하는 경우에만 다음 뉴런으로 신호 전달



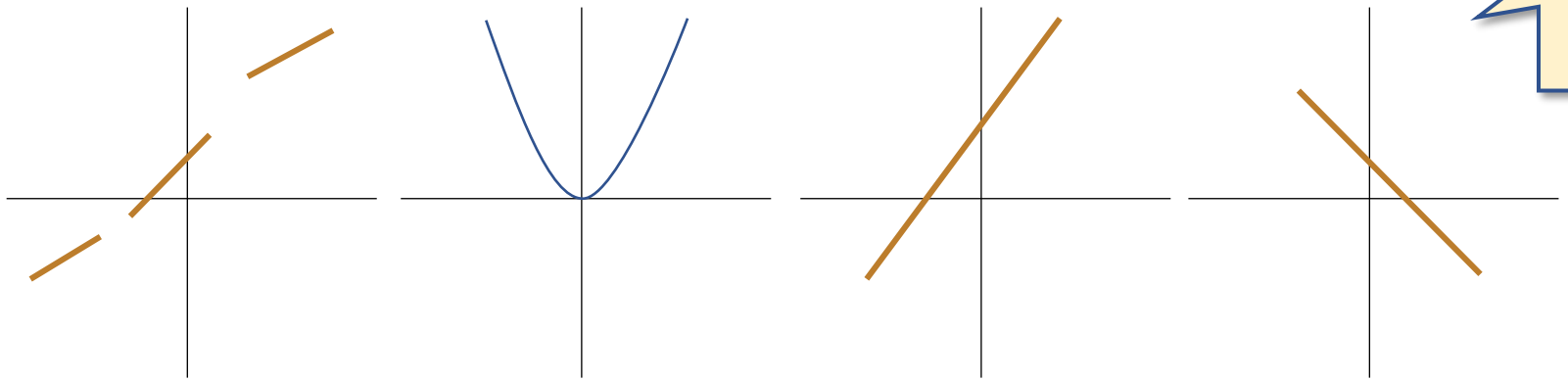
활성화 함수의 적용 개념

딥러닝 기본 모델-활성화 함수

- 그럼 활성화 함수는 왜 그렇게 많은 형태가 존재하는가?
  - 우리는 아직 우리의 뇌와 신경들이 어떻게 동작하고 서로 어우러지는지 정확하게 알지 못함
  - 따라서 우리가 AI로 해결하고자 하는 문제에 맞게 가장 효율적이고 적절한 활성화 함수를 계속 연구, 개발하여 활용하는 것

딥러닝 기본 모델-활성화 함수

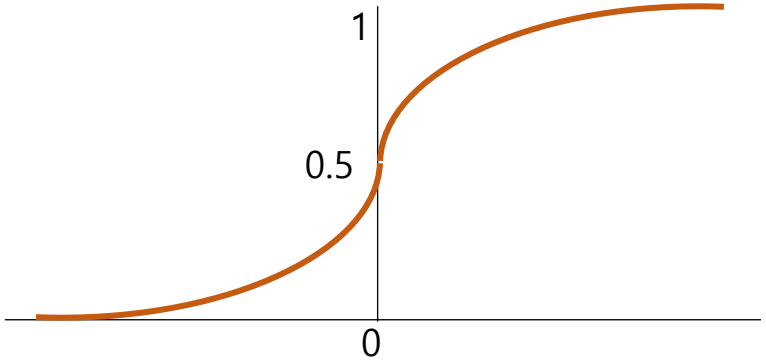
- 활성화 함수의 조건
  - 정의역(함수에 입력 가능한 값의 범위, 집합) 안에서 연속이며 무한해야 한다
  - 단조 함수여야 한다 (방향을 바꾸지 않아야 한다)
  - 비선형 함수여야 한다.
  - 계산 효율이 좋아야 한다.



이런 함수들은
부적합

딥러닝 기본 모델-활성화 함수

- 표준 은닉 활성화 함수
  - Sigmoid

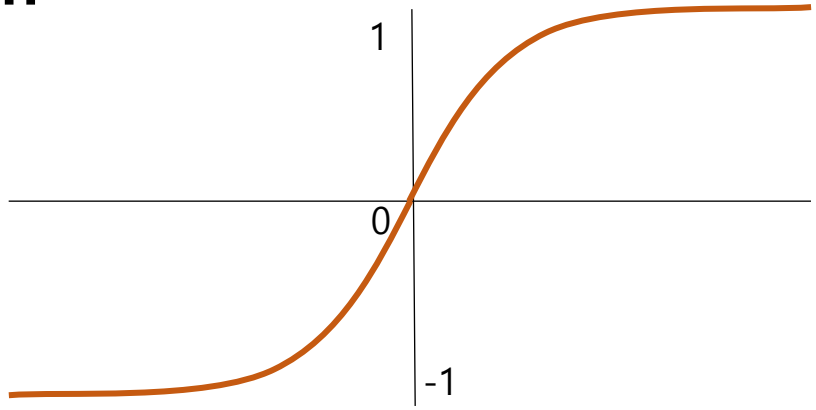


가장 많이 사용되어 왔고

가장 중요한 활성화 함수

(활성화 함수의 기본 형태)

  - tanh



은닉층에서는 sigmoid 보다 tanh 함수가 더 좋음

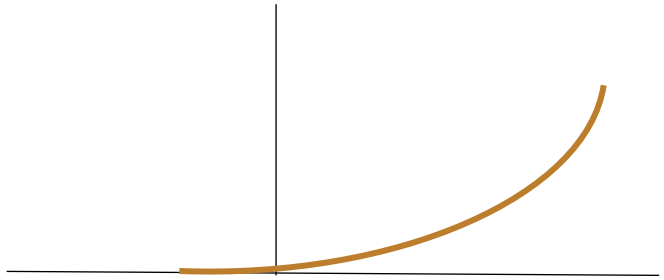
음의 상관관계도 지원

딥러닝 기본 모델-활성화 함수

- **표준 출력 계층 활성화 함수**
  - **신경망의 목적에 따라 최선의 선택이 달라진다**
    - 일반 데이터값 예측 → 활성화 함수 미적용
    - 서로 무관한 항목에 대한 예/아니오 확률 예측 → sigmoid
    - 여러 가능성 중 하나의 확률 예측 → softmax
  - **최근 가장 많이 쓰이는 활성화 함수**
    - Relu

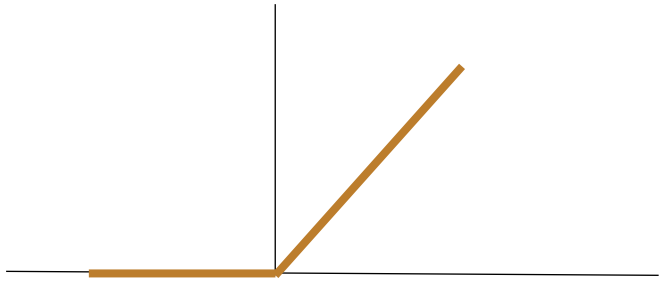
딥러닝 기본 모델-활성화 함수

- 표준 출력 계층 활성화 함수
  - Softmax



    - k개의 값이 존재할 때 각각의 값의 편차를 확대시켜 큰 값은 상대적으로 더 크게, 작은 값은 상대적으로 더 작게 만든 후, 정규화 시키는 함수
    - Softmax 함수를 거친 k개의 값의 총합은 1이 됨
    - 지수증가를 기반으로 하는 함수

  - Relu (Rectified Linear Unit, 정류된 선형 유닛)



    - 데이터가 0보다 작으면 무조건 0
    - 데이터가 0보다 크면 입력값 그대로!! (주로 선형증가의 형태가 된다)

딥러닝 기본 모델-오차의 측정

- **출력층에서는 왜 오차를 측정하는가?**
  - **신경망의 목적은 정확한 예측 결과를 얻는 것**
    - 분류를 위한 모델은? 분류 역시 어떤 클래스가 가장 잘 일치할 것인가..를 예측, 계산하여 그 값이 가장 큰 것을 선택하는 것이므로 동일하다고 볼 수 있음
  - **예측을 한 후에는 얼마나 잘 예측했는가 평가해야 함**
    - 평가 방법으로 오차의 측정을 사용 → 가장 간단하고 쉬운 방법이므로
  - **특히 가중치의 조정은 미분과 관련이 있다는 것은**

    → 오차의 값은 양수만 사용해야 한다는 의미

딥러닝 기본 모델-오차의 측정

- 오차의 값은 왜 양수만 사용하는가?
  - 미분은 거리, 넓이를 이용한 개념 → 거리 또는 넓이는 음수가 없음

- 어떻게 양수만으로 오차를 처리할 것인가?
  - 실제로 오차를 측정하면 양수, 음수 모두 나올 수 있지만 각 값을 거리의 개념으로 바꿔서 사용
  - 절대값, 제곱 등을 이용하여 양수로 변환함

딥러닝 기본 모델-손실 함수

- **손실 함수 (Loss Function)**
  - **출력 값과 정답(기대 값)의 오차를 정의하는 함수**
  - **손실 함수는 데이터의 특성에 따라 변형, 새로 제안해서 사용 가능**
  - **종류**
    - 평균 제곱 오차 (MSE, Mean Squared Error)
      - 가장 많이 사용됨. 출력 값과 기대 값의 차이를 제곱하여 평균한 값

$$E = \frac{1}{2}\sum_{k}(y_k - t_k)^2$$

      - 큰 오차는 더욱 크게, 작은 오차는 더욱 작게 → **처리할 때에는 큰 오차에 더 집중**

(전반적인 성능 향상에 더 좋음)

    - 교차 엔트로피 오차 (Cross Entropy Error)
      - 범주형 데이터의 분류에 주로 사용

$$E = -\sum_{k} t_k \log y_k$$

딥러닝 기본 모델-손실 함수

- 왜 손실함수를 사용하는가?
  - 학습의 궁극적인 목적은 높은 정확도를 끌어내는 매개변수를 찾는 것
  - 왜 “정확도“라는 지표를 놔두고 “손실함수의 값“이라는 우회적인 방법을 사용하는가?

딥러닝 기본 모델-손실 함수

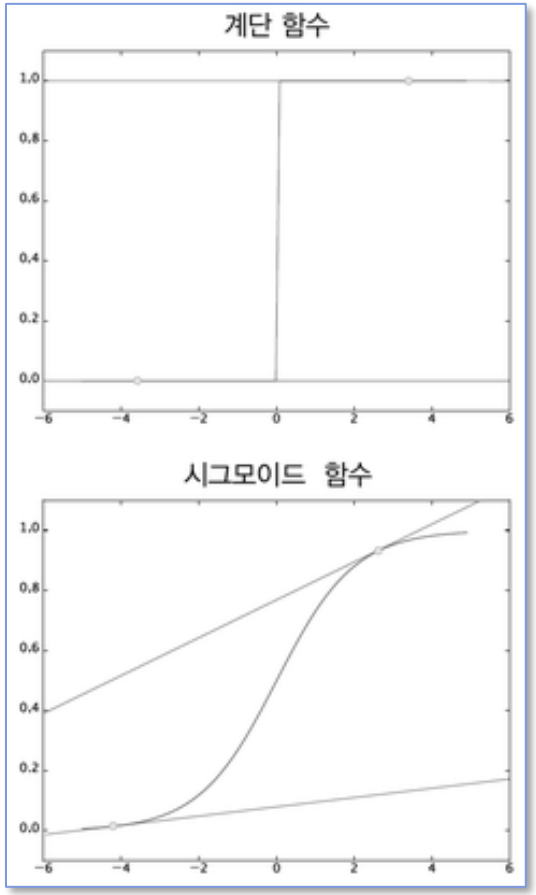
- **왜 손실함수를 사용하는가?** 신경망 학습에서의 미분의 역할을 생각해 보면...
  - **신경망 학습에서는 최적의 매개변수를 탐색할 때, 손실함수의 값을 가능한 작게 만드는 매개변수의 값을 찾음**
  - **이때 매개변수의 미분을 계산하고, 그 미분 값을 단서로 매개변수의 값을 서서히 갱신하는 과정을 반복함**
    - 손실함수의 미분 값이 음수 → 가중치 매개변수를 양의 방향으로 변화시켜 손실함수의 값을 줄일 수 있다.
    - 손실함수의 미분 값이 양수 → 가중치 매개변수를 음의 방향으로 변화시켜 손실함수의 값을 줄일 수 있다.
    - 손실함수의 미분 값이 0 → 어느 쪽으로도 움직이지 않으므로 갱신이 멈춘다.

딥러닝 기본 모델-손실 함수

- 정확도를 지표로 삼지 않는 이유
  - 미분 값이 대부분의 장소에서 0이 되어 매개변수를 갱신할 수 없다.
  - 정확도는 매개변수의 작은 변화에는 거의 반응을 보이지 않거나, 갑자기 변화한다.
  - 활성화 함수로 '계단 함수'가 아닌 '시그모이드 함수'를 사용하는 이유도 같다.

딥러닝 기본 모델-손실 함수

- 정확도를 지표로 삼지 않는 이유
  - 계단함수는 대부분의 장소에서 기울기가 0 이지만, 시그모이드 함수의 기울기(접선)는 0이 아니다.
  - 계단 함수는 한순간만 변화를 일으키지만, 시그모이드 함수의 미분은 연속적으로 변한다.
  - 즉 시그모이드 함수의 미분은 어느 장소에서도 0이 되지 않는다.
  - 이는 신경망 학습에서 중요한 성질로, 기울기가 0이 되지 않는 덕분에 신경망이 올바르게 학습할 수 있다.



딥러닝 기본 모델-과적합

- **과적합(Over Fitting)**
  - **주어진 데이터로 학습을 너무 많이 하면 오히려 역효과!!**
    - 학습에 입력된 데이터는 완벽에 가깝게 처리함
    - 학습에 입력되지 않은 데이터는 제대로 처리되지 않음
  - **원인: 잡음 데이터 (대부분)**
    - 불필요한 정보가 많이 포함된 데이터로 학습이 반복됨에 따라 불필요한 정보가 분류의 기준에 포함되어 버리는 것이 원인

딥러닝 기본 모델-과적합

- **과적합(Over Fitting)의 해결 방안**
  - **조기 종료**
    - 적당한 선에서 학습을 종료시킴 → 데이터의 정규화와 관련
  - **정규화 (데이터를 일반화 시키기)**
    - 필요한 신호는 학습하고 잡음은 제거하는 효과
    - 모델의 학습 난이도를 높임으로써 학습 데이터의 세부 사항(잡음 포함)에 대한 일반화를 활용하도록 하는 기법의 일부로 사용됨

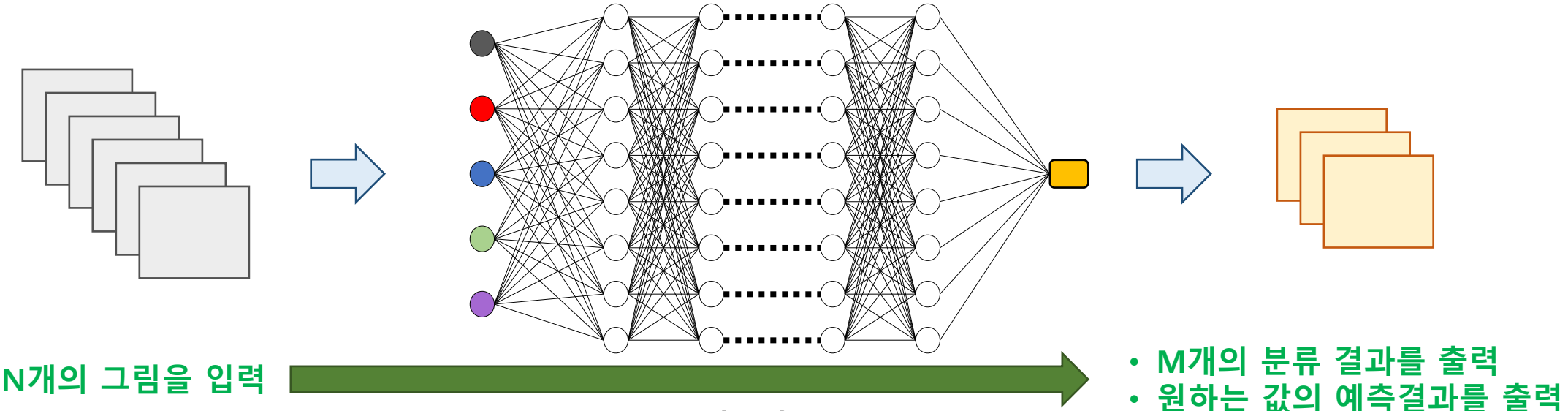
딥러닝 기본 모델-과적합

- **과적합(Over Fitting)의 해결 방안**
  - **Drop Out**
    - 학습 중에 무작위로 선택한 뉴런을 0으로 설정

      → 군데군데 망의 연결고리를 잘라 내어 대형 신경망이 소형 신경망처럼 동작하게 만듦
    - 소형 신경망에서는 과적합이 거의 발생하지 않음 (표현능력이 협소하기 때문)
    - 대형 신경망(딥러닝 모델)을 Drop Out을 통해 소형 신경망처럼 동작하게 하여 과적합 발생률을 떨어뜨리는 방법

딥러닝이란?

- 엄밀하게 따지면 딥러닝이란...
  - 다수의 노드에 적용되는 최적화 프로그램이다.



N개의 그림을 입력

- M개의 분류 결과를 출력
- 원하는 값의 예측결과를 출력

결과를 출력할 수 있도록

각 노드 사이의 경로에 대한 가중치를

조건을 모두 만족할 수 있는

값들의 집합으로 최적화 하는 프로그램

딥러닝이란?

- 그렇다면 이것은 인공지능이 아니지 않나?
  - 애초에 인간의 두뇌 역시 다양한 입력의 결과를 올바르게 출력하기 위한 최적화 과정을 처리함
  - 많이 사용될 수록 굵어지고 민감해지는 각 신경세포 간의 시냅스 결합 강도를
  - 뉴런 사이의 가중치로 모델링하여 구현한 것일 뿐!!