

# 제 1 장 인공지능이란?

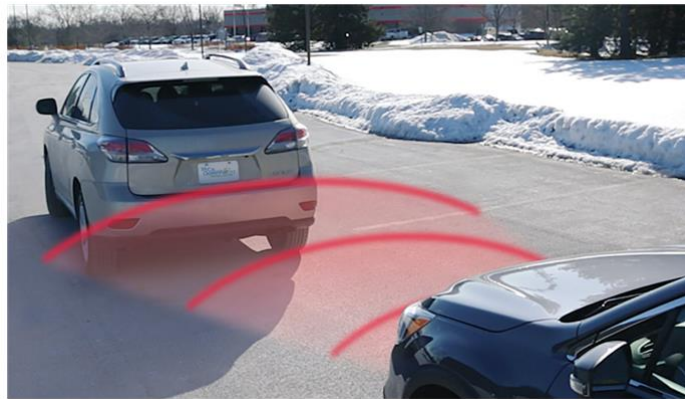
# 인공지능의 시대

- 1997년. IBM 딥블루 : 체스시합에서 세계 챔피언이었던 카스퍼로프를 상대로 승리(인간을 넘어선 최초의 컴퓨터)
- 2011년. IBM 왓슨 : 퀴즈쇼 “제퍼디”에서 우승 차지
- 2016년. 알파고(AlphaGo) : 구글의 인공지능 바둑 프로그램 - 이세돌과의 경기에서 4-1로 승리
- 2017년. 마스터(Master) : 업그레이드된 알파고
- 인간 vs 컴퓨터...'숨막히는' 반세기 대결 역사



# 인공지능의 시대

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



## Tesla FSD Beta 시승기



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

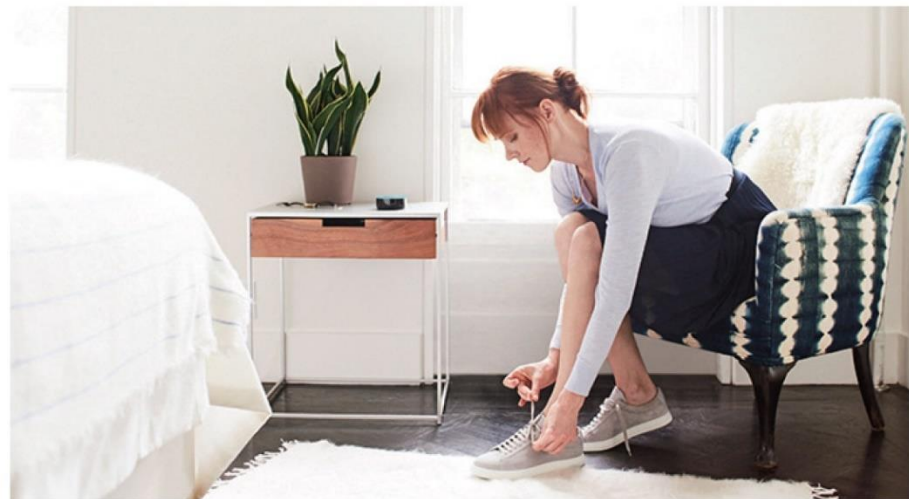
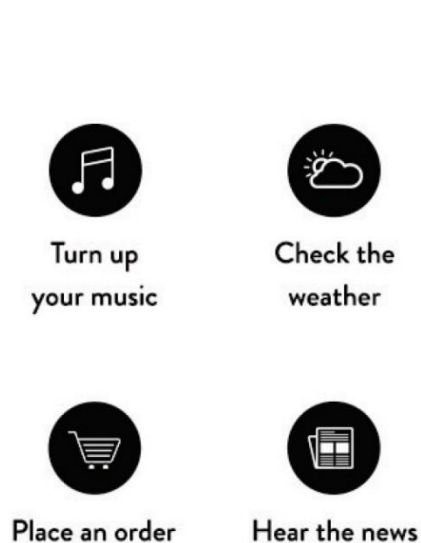


그림 1-5 아마존의 알렉사(\*출처: 아마존)



- 과제

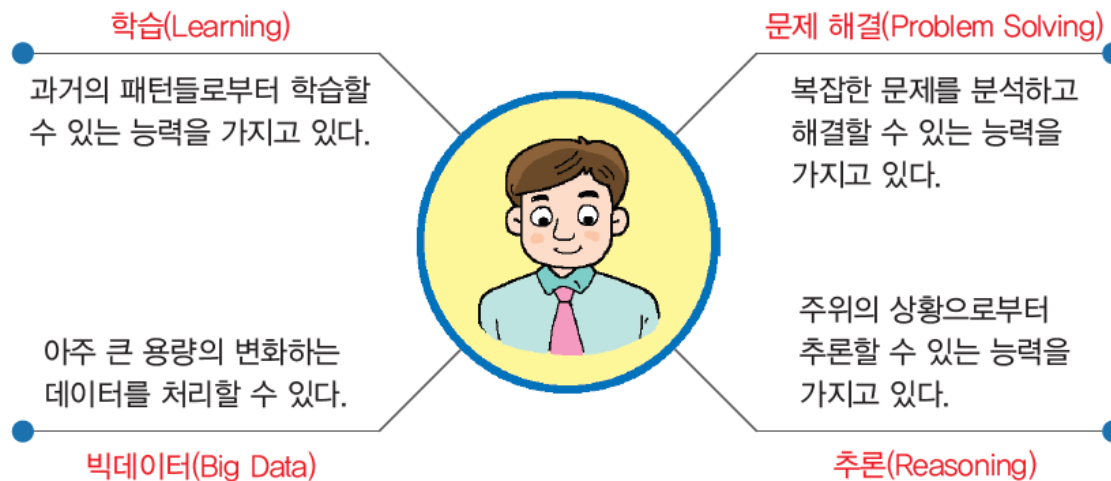
- 1) 자율주행기술 레벨 1,2,3,4 차이점
- 2) 우리나라 자율주행 기술 현황

유튜브, 문서 등을 보고 이해한 후 **A4 1장씩** 요약 정리

제출기한 :

# 인공지능의 정의

- 인간의 지능적 사고는 다음과 같은 특징을 가지고 있다.

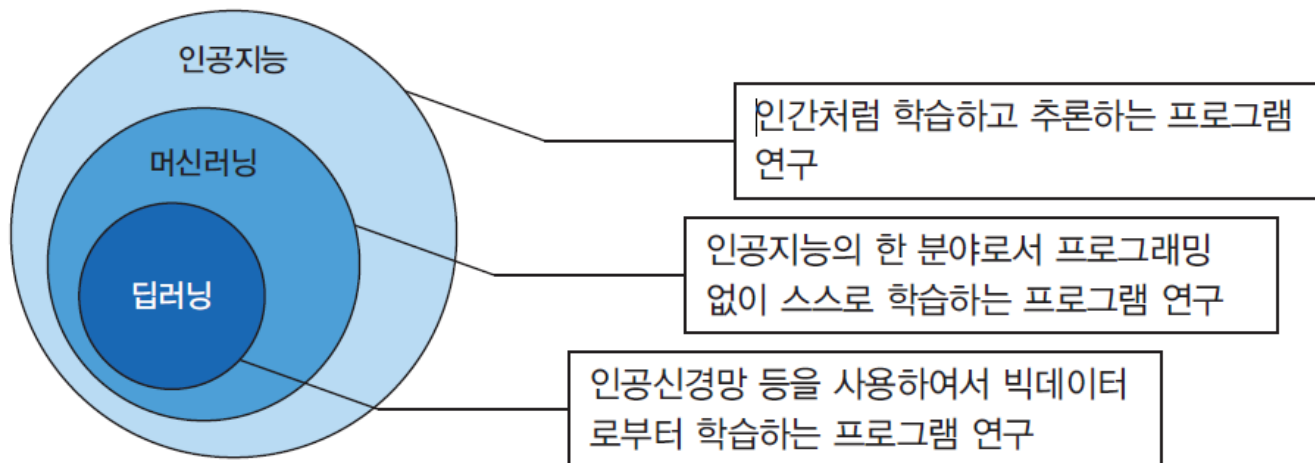


- 지능이란?
  - 인간이 사물을 이해하고 학습하는 능력(learning)
  - 어떤 문제가 주어졌을 때, 합리적으로 사고하여 문제를 해결하는 능력(problem solving)



# 인공지능의 정의

- 인공지능(Artificial Intelligence) : 컴퓨터에 지능을 부여하기 위한 연구 분야
- 머신러닝(Machine Learning) : 인공지능의 한 분야.
  - 컴퓨터에 학습 기능을 부여하기 위한 연구 분야
  - 데이터를 학습하는 알고리즘
  - 학습할 수 있는 데이터가 많아지면 알고리즘 성능이 향상
  - 어떤 문제에 대하여 명시적 알고리즘을 설계하고 프로그래밍하는 것이 어렵거나 불가능한 경우에 사용
  - 스팸 이메일 필터링, 네트워크 침입자 작동 검출, **OCR**, 컴퓨터 비전 등



# 인공지능의 정의

- 딥러닝(deep learning)
  - 신경망(neural network)의 학습 알고리즘
  - 인간의 두뇌를 본떠서 기계로 만들려고
  - 최근의 인공지능 붐은 전적으로 딥러닝의 성공 때문

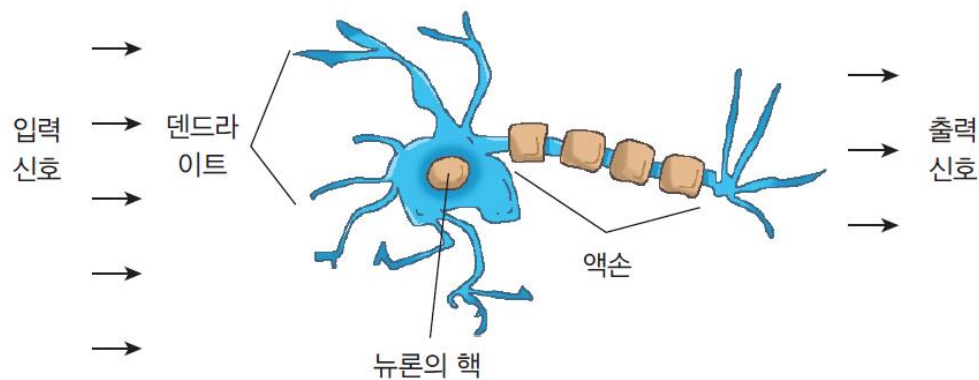


그림 1-8 생물학적 뉴런의 구조



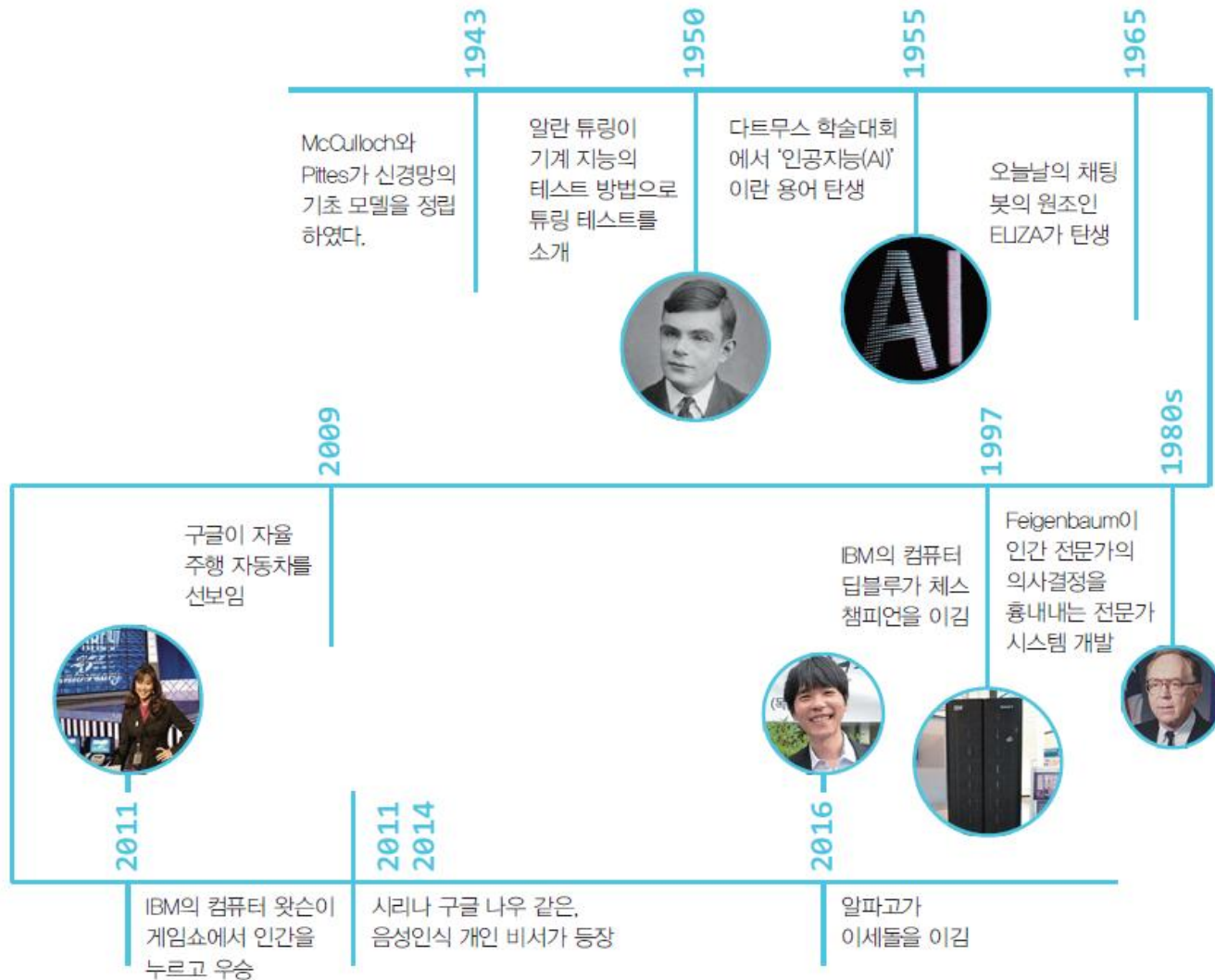
# 인공지능의 역사



한눈에 보는 인공지능의 역사

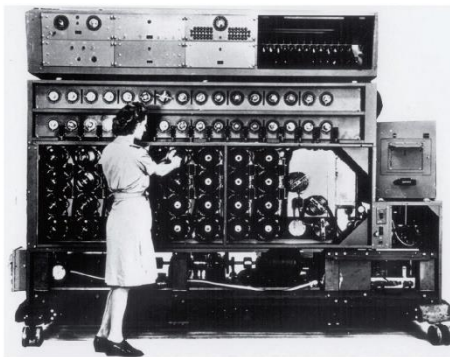


# 인공지능의 역사



# 튜링 테스트

- 1950년. 영국의 수학자. 알란 튜링(Alan Turing)
- “기계가 생각할 수 있을까?”
- 튜링 테스트
  - 인간, 컴퓨터, 질문자가 각각 독립된 방에 있고 원격 터미널만을 사용하여 통신
  - 질문자는 방 안에 누가 있는 지 볼 수 없고 음성을 들을 수도 없다.
  - 질문자는 누가 인간이고 누가 컴퓨터인지를 알아내기 위하여 질문을 하게 된다.



# 퍼셉트론(perceptron)

- Frank Rosenblatt
- 인공지능의 초기형태
- 1969. Minsky & Papert.
  - 퍼셉트론이 할 수 있는 작업에는 심각한 제한이 있고 엄청나게 과장되어 있다.는 것을 증명 -> 신경망 연구 중단 초래
  - 후대 연구자들이 퍼셉트론의 제한점을 극복할 수 있는 방법을 찾아내어 신경망 분야를 부활

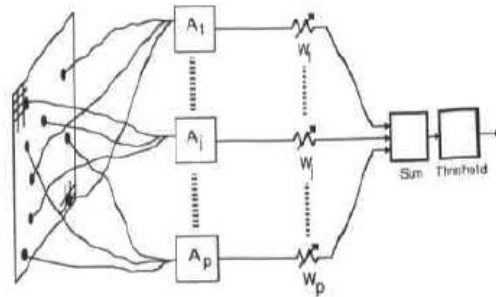
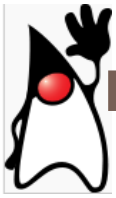


그림 1-13 Rosenblatt와 퍼셉트론



# 다트머스 학술회의

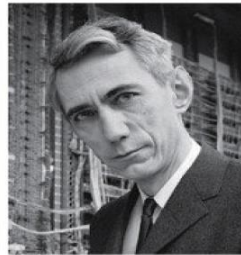
- 1956년 Marvin Minsky와 John MacCarthy 등에 의하여 조직
- ‘인공지능’이라는 이름이 만들어 지고 탄생



John MacCarthy



Marvin Minsky



Claude Shannon



Ray Solomonoff



Alan Newell



Herbert Simon



Arthur Samuel



Oliver Selfridge



Nathaniel Rochester



Trenchard More

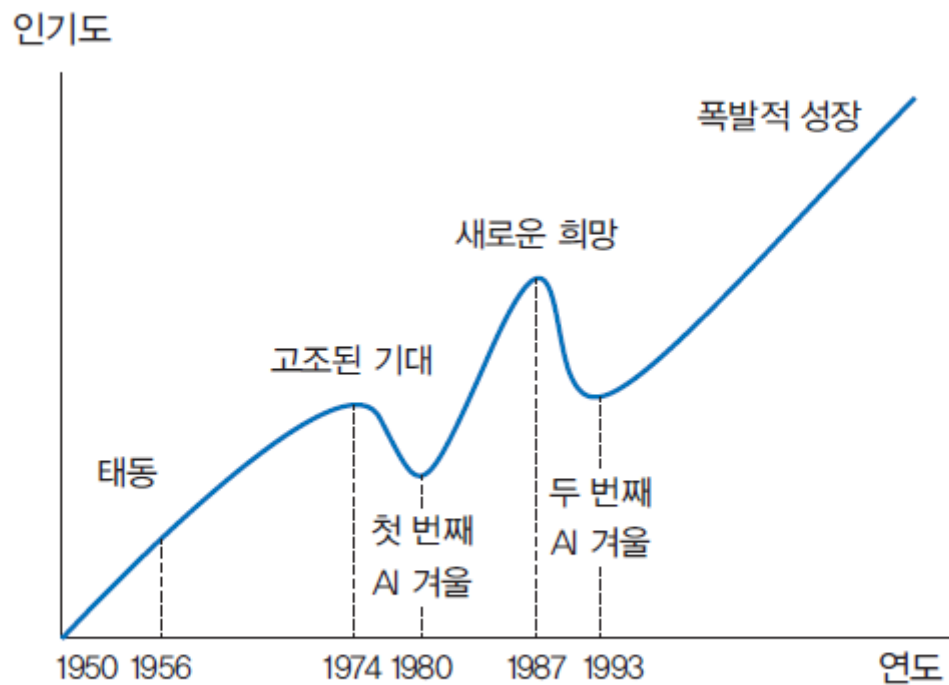


그림 1-17 인공지능의 부침

- 1970년대 문제점(한계)

(1) 충분한 컴퓨팅 파워가 없었다.

실제로 유용한 결과를 내는데 필요한 **CPU**의 속도나 충분한 메모리가 없었다.

(2) 상상할 수 없는 양의 계산 시간이 필요하다.

입력의 크기를  $n$ 이라 하면, 쉬운 문제들은  $n$ 의 다항식에 비례하는 시간 안에 풀 수 있지만 어려운 문제들은  $n$ 의 지수에 비례하는 시간이 필요하다.

(3) 엄청난 양의 정보를 필요로 한다.

1970년에는 아무도 이 정도의 데이터베이스를 만들 수 없었고 어떤 프로그램도 이 방대한 정보를 어떻게 학습해야 하는지를 알지 못했다.

(4) 새로운 논리 시스템이 필요.

논리 그 자체의 구조를 변경하지 않고는 일반적인 추론을 구현하기 어렵다는 것을 발견했다.



# 전문가 시스템(Expert System)

- 문제 영역을 제한해야 한다. 이 세상의 모든 문제를 해결할 수 있는 시스템을 개발한다는 생각을 버렸다.
- 전문가 시스템 : 전문가로부터 습득한 ‘규칙(rule)’들을 사용하여 특정 영역에 대해 질문이나 문제를 해결하는 프로그램

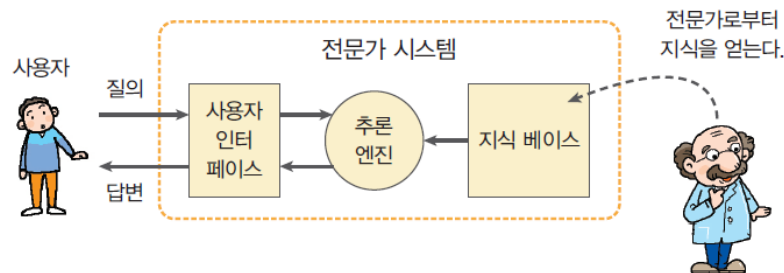


그림 1-19 전문가 시스템

- if-then 형태의 지식 표현 방법

규칙 1: IF 신호등이 녹색  
THEN 길을 건넌다.

규칙 2: IF 신호등이 빨간색  
THEN 멈춰서 기다린다.





# 전문가 시스템(Expert System)

- DENDRAL

- 분광계 수치로 화합물을 분석하는 전문가 시스템
- NASA에서 무인우주선을 화성 발사. 화성 토양을 분광계 수치로 분석

- MYCIN

- 전염성 질환을 진단하고 항생제를 처방하는 전문가 시스템
- 초보 의사보다 수준이 높았다

- 1982년. 물리학자 John Hopfield.
- 완전히 새로운 방식으로 정보를 학습하고 처리할 수 있는 한 형태의 신경망(Hopfield Net)을 제안
- Geoffrey Hinton과 David Rumelhart는 "역전파(backpropagation)"라고 불리는 유명한 학습 방법을 대중화

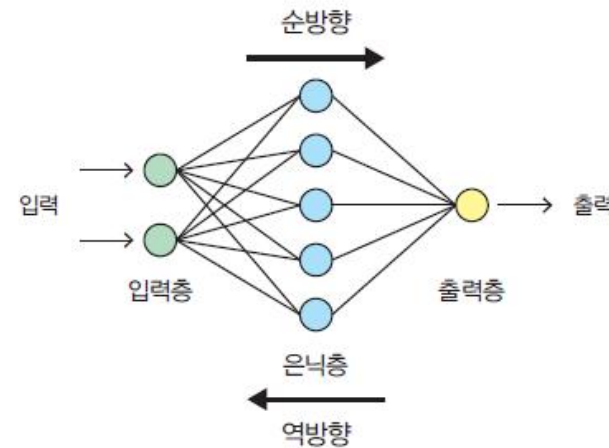


그림 1-21 Geoffrey Hinton과 역전파 알고리즘

# 두 번째 AI 겨울

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



- 딥러닝(deep learning)

- 많은 은닉층을 가진 ‘심층 신경망’에서 사용하는 학습 기술
- 2012년 영상 인식 대회에서 신경망은 다른 모든 방법을 큰 차이로 물리쳤다.
- 얇은 신경망에 비해 훨씬 더 복잡한 작업을 수행할 수 있다.

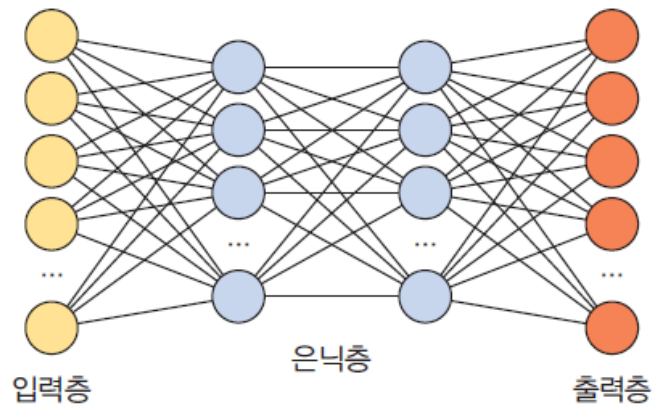


그림 1-23 심층 신경망

- 빅데이터(Big Data)

- 우리 주변의 디지털 환경에서 생산되는 방대한 데이터를 저장하고 활용할 수 있게 되었다. 예를 들어서 인터넷 쇼핑몰에서는 사용자가 돌아다닌 기록이 자동적으로 데이터로 저장된다.
- 빅데이터는 딥러닝의 훈련 데이터로 활용할 수 있다.

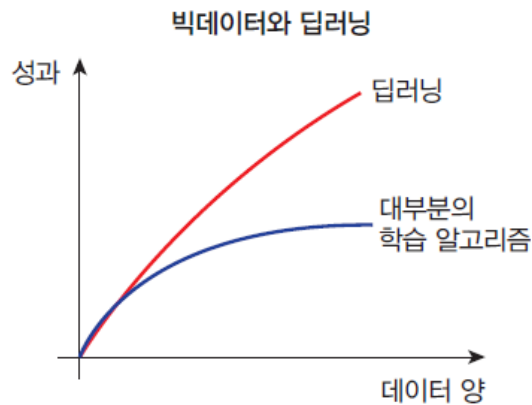
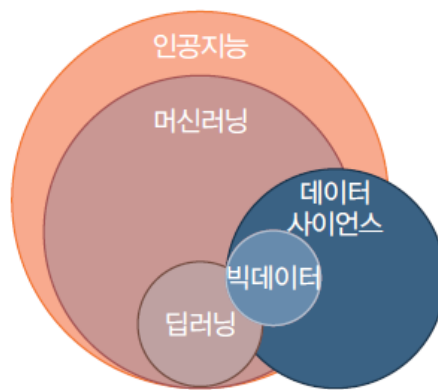


그림 1-24 빅데이터



# 인공지능의 역사 — 다시한번 보기



한눈에 보는 인공지능의 역사

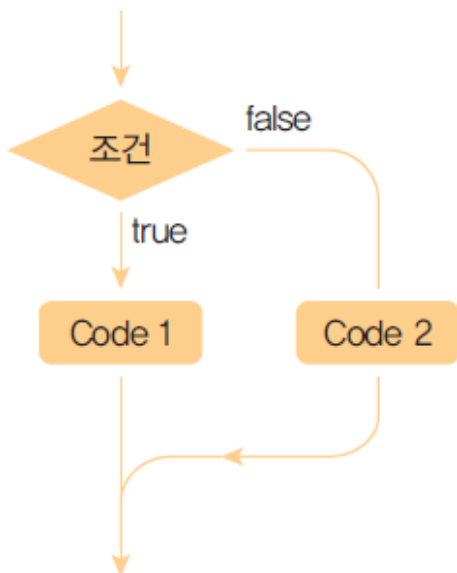
[추천] 인공지능의 역사 1부(신인류). 1시간27분. 깊은 내용

<https://www.youtube.com/watch?v=RBx0s-ugEh0>



# 규칙 기반 방법 VS 머신러닝 방법

규칙 기반 방법



머신러닝 방법

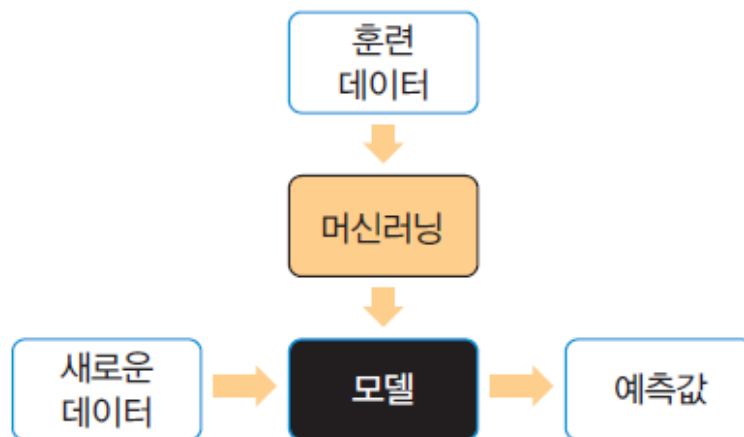


그림 1-26 규칙 기반과 머신러닝 방법의 비교



# 규칙 기반 방법 VS 머신러닝 방법

- 규칙 기반 방법
  - 전문가 시스템이나 전통적 프로그램에서 사용하는 방법
  - **IF ... THEN ...** 형태
  - 인간이 모든 지식을 코딩하여야 한다. 따라서 지식을 확장하기가 어렵다

규칙 #1      IF 동물이 육식동물이다. AND 황갈색이다. AND 갈기가 있다.  
              THEN 사자이다.

규칙 #2      IF 동물이 육식동물이다. AND 황갈색이다. AND 검은 줄무늬가 있다.  
              THEN 호랑이다.



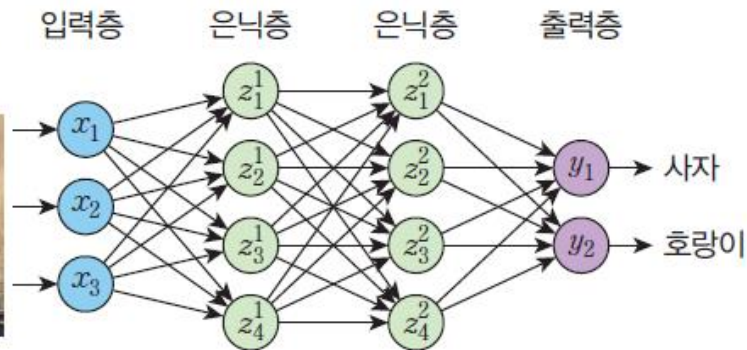




# 규칙 기반 방법 VS 머신러닝 방법

- 머신러닝 방법

- 사진들을 모아서 인공지능 시스템에 보여주면서 학습
- 데이터에 비례하여 성능을 향상시킬 수 있고
- 인간이 데이터로부터 특징을 추출할 필요가 없다.





# 인공지능의 응용 분야

- 자율 주행 자동차
- 맞춤형 광고
- 챗봇(chatbot)
- 의료분야
- 언어 번역
- 경영 분야
- 딥러닝 예술



# 인공지능의 응용 분야

- 과제

- 1) 일상생활에서 인공지능이 응용되고 있는 사례 조사

- 찾아서 읽고 써보고 응용 사례 설명
- 사용해본 소감 및 평가

- 2) 인공지능 응용 제안

- 인공지능이 이런 곳에 이런 기능을 제공하면 좋겠다

- 제출기간 :



# Lab: 머신러닝 체험하기 #1

- <https://transcranial.github.io/keras-js/#/>

The screenshot displays the Keras.js web application interface. The browser address bar shows the URL `transcranial.github.io/keras-js/#/mnist-cnn`. The page title is "Basic Convnet for MNIST".

On the left sidebar, under the "Keras.js" logo, there are sections for "DEMOS" and "LINKS". The "DEMOS" section lists various models: Basic Convnet (MNIST), Convolutional VAE (MNIST), AC-GAN (MNIST), ResNet-50 (ImageNet), Inception v3 (ImageNet), DenseNet-121 (ImageNet), SqueezeNet v1.1 (ImageNet), Bidirectional LSTM (IMDB), and Image Super-Resolution. The "LINKS" section includes "GitHub repo" and "MD.ai". The "CONTACT" section lists "Leon Chen" and "@transcranial".

The main content area features a drawing canvas with the instruction "Draw any digit (0-9) here". A large digit "7" is drawn on the canvas. To the right of the canvas is a "use GPU" toggle switch, which is currently turned on. Below the canvas is a "CLEAR" button.

To the right of the canvas is a bar chart showing the probability distribution for digits 0 through 9. The bar for digit 7 is the tallest, indicating the highest probability.

Below the drawing area, there are two sections showing the internal state of the model:

- Conv2D**: 32 3x3 filters, padding valid, 1x1 strides. This section displays a grid of 32 small images, each showing the digit "7" filtered by a different 3x3 kernel.
- Activation**: ReLU. This section displays a grid of 32 small images, each showing the digit "7" after ReLU activation.



# Lab: 머신러닝 체험하기 #2

- [Quick Draw](https://quickdraw.withgoogle.com/) (https://quickdraw.withgoogle.com/)





# Lab: 머신러닝 체험하기 #3

- 티처블 머신 (<https://teachablemachine.withgoogle.com/>)

The screenshot shows the Teachable Machine website in a web browser. The browser's address bar displays 'teachablemachine.withgoogle.com'. The website has a blue header with the 'Teachable Machine' logo and navigation links for 'About', 'FAQ', and a 'Get Started' button. The main content area features the title 'Teachable Machine' in large blue letters, followed by the subtitle 'Train a computer to recognize your own images, sounds, & poses.' Below this is a paragraph explaining that it's a fast, easy way to create machine learning models without coding. A 'Get Started' button is prominently displayed. To the right, there is a video of a person with a blue pose-detection overlay. Below the video, a small overlay shows progress bars for 'Tree' and 'Wings', with 'Wings' at 100%. At the bottom, there are logos for various frameworks like TensorFlow.js, p5.js, Coral, and Node.js. The footer section begins with the heading 'What is Teachable Machine?'.

Teachable Machine

Train a computer to recognize your own images, sounds, & poses.

A fast, easy way to create machine learning models for your sites, apps, and more – no expertise or coding required.

Get Started

Tree

Wings 100%

What is Teachable Machine?