KiSTi 한국과학기술정보연구원 Korea Institute of Science and Technology Information

2023

인공지능특론 (Advanced Artificial Intelligence)

11주 신경망 소개

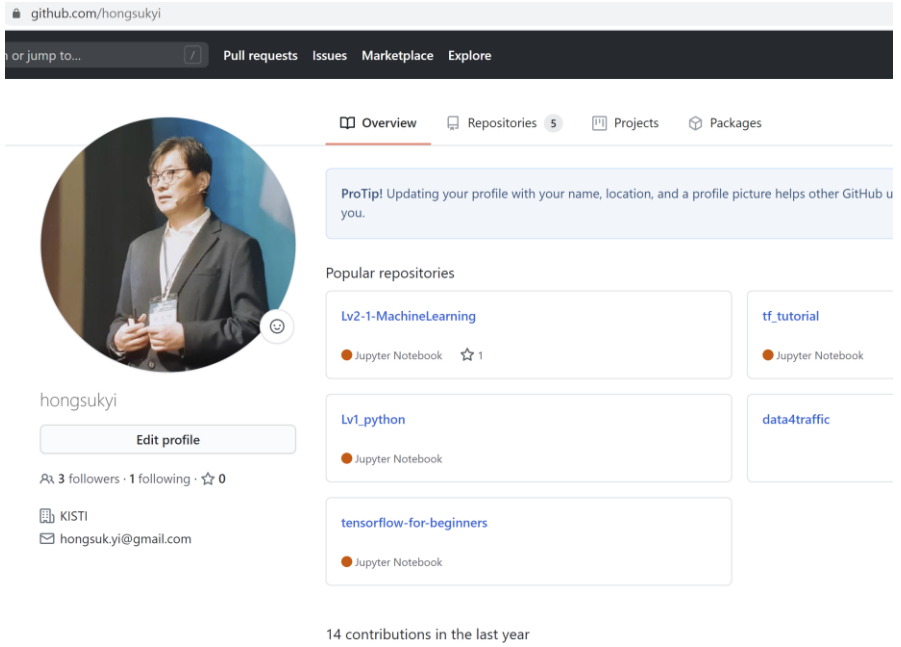
Hongsuk Yi KISTI

강사소개

- 성명 : 이홍석
- 소속
  - ✓KISTI 데이터기반문제해결연구단 책임연구원
  - ✓과학기술연합대학원(UST-KISTI) 응용AI전공 교수
- 연구내용
  - ✓딥러닝기반 도심지 교통혼잡 예측 (2018~2021)
  - ✓ 지능형 인프라 기술 연구 (2018~2019)
  - ✓도심지 교통예측을 위한 트랜스포머 모델 연구(2022~2025)
- 컴퓨팅 기술
  - ✓GPU 기반 가속컴퓨팅 기술 : CUDA, OpenCL
  - ✓슈퍼컴퓨팅 병렬처리 기술 : MPI, OpenMP, ManyCore Computing
  - ✓Deep Learning 기술 : Tensorflow, PyTorch, Keras, Theano

강사소개

- 교육: 머신러닝, 딥러닝, 인공지능, 빅데이터, MPI/OpenMP, CUDA, OpenCL 등
  - ✓딥러닝 워크숍 개최 (2015)
  - ✓한국교통학회 텐서플로우 튜토리얼 강좌 (2016~2018)
  - ✓교통학회/KIRD/KISTI 과학데이터스쿨 강의 진행 (2018~2022)
  - ✓교통 빅데이터 교육 (2022)
  - ✓UST 인공지능 특론 (전공필수) (2022)
  - ✓UST 인공지능 이해 (교양) (2021~2022)
  - ✓UST 기계학습의 이해, 인공지능 기초 (2019~2022)
- 참고자료
  - ✓인공지능, 딥러닝 강의 자료 모음 (github.com/hongsuk.yi)



전공강좌 강의계획서

KISTI SINCE 1962

❖ 강좌의 특성 및 학습목적

- ✓인공지능 기술에 대한 기초지식 및 활용역량 배양을 목적
- ✓머신러닝부터 심층 신경망까지 개념과 이론을 소개
- ✓파이썬 Scikit-Learn 패키지를 활용해 실습

❖ 본 강좌의 대상

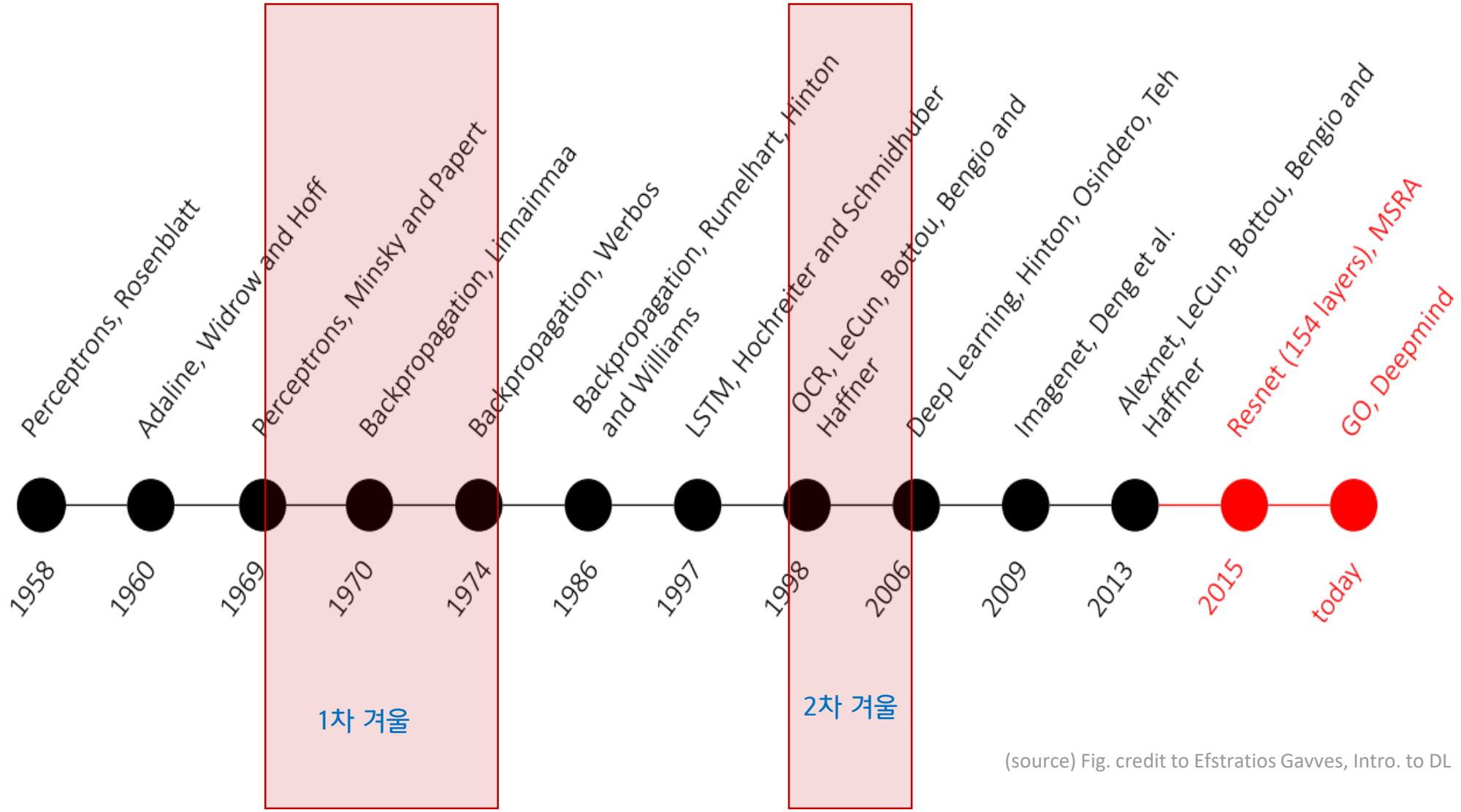
- ✓Python 언어에 대한 기초지식 및 프로그램 경험자
- ✓기계학습 또는 딥러닝에 대한 사전 지식이 없는 학생

인공지능 특론 강의 계획(수정)

- 11주 : 인공지능 소개 및 실습 데이터 소개
- 12주 : DNN 소개 및 응용
- 13주 : RNN 소개 및 응용
- 14주 : LSTM 소개 및 응용
- 15주 : CNN 소개 및 응용

인공지능 역사

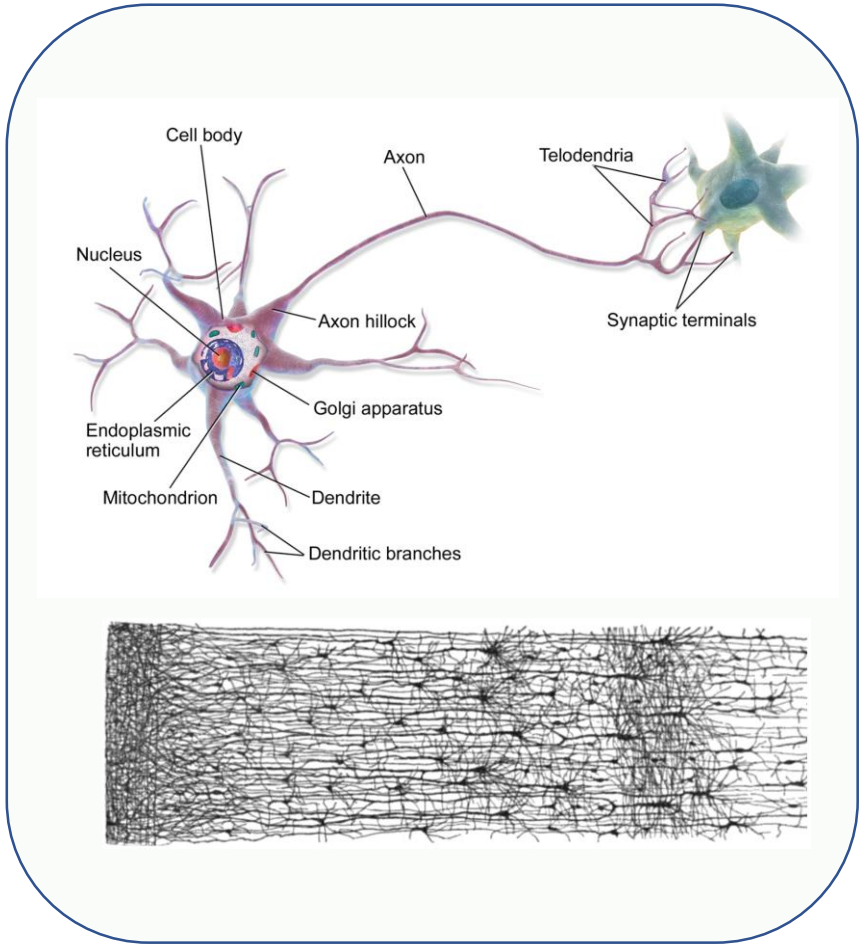
인공지능 역사

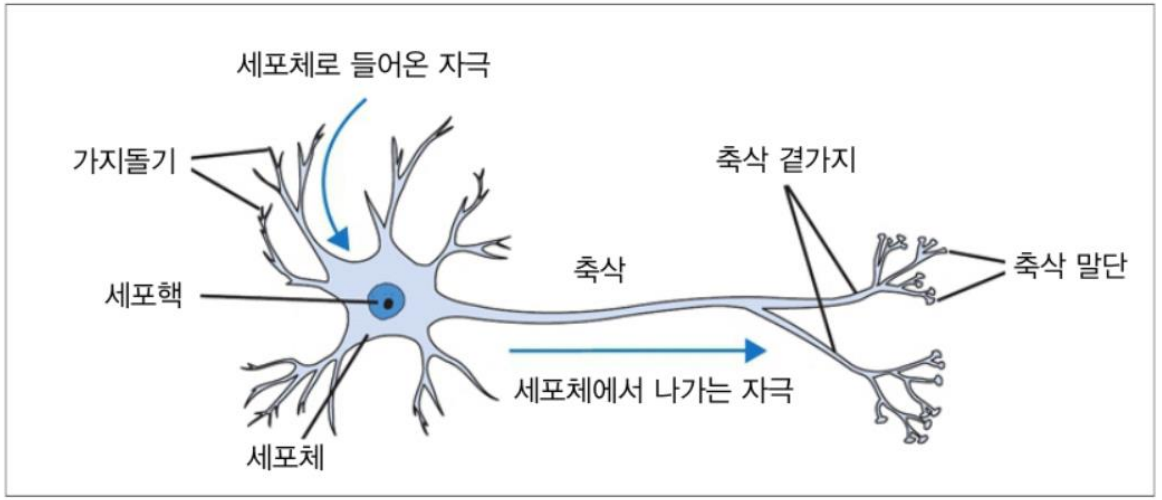


(source) Fig. credit to Efstratios Gavves, Intro. to DL

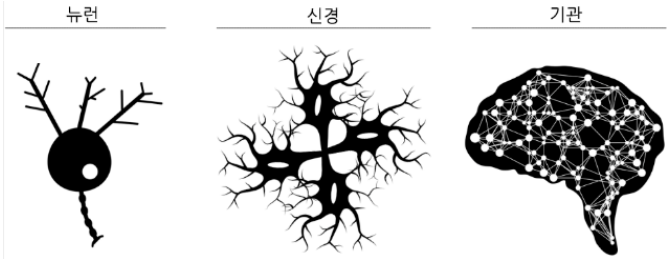
생물학적 뉴런(신경세포)의 구조와 신경망

❖ 생물학적 신경세포 ~사람의 뇌는 1천억(100 billion)개의 뉴런으로 구성





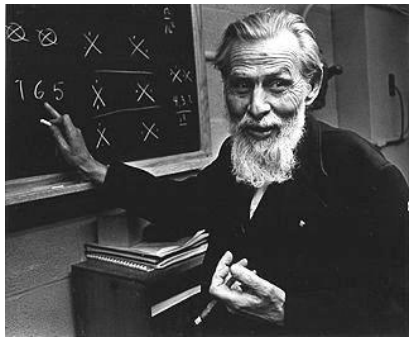
생물학적 신경망(*http://cs231n.github.io/neural-networks-1*)



(source) https://towtow-ai.tistory.com/5

From Biological to Artificial Neurons

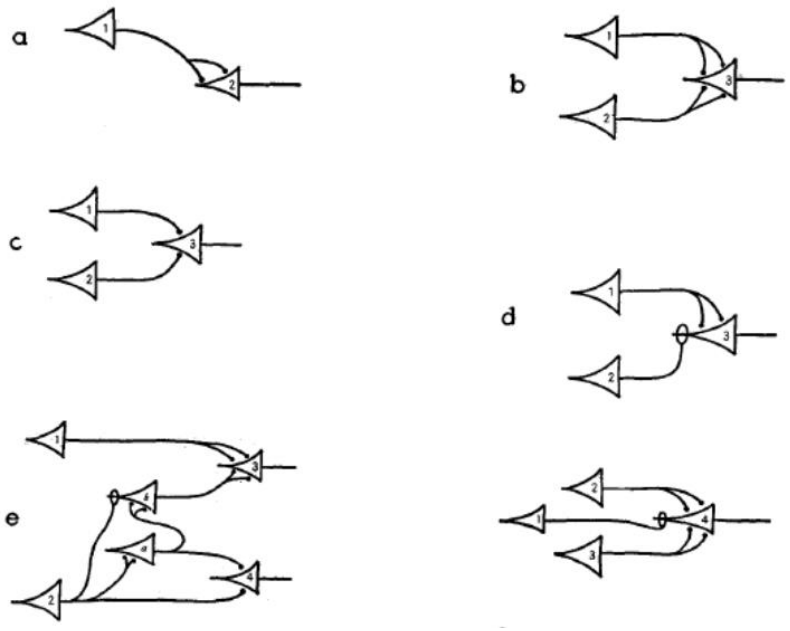
- ANNs were first introduced back in 1943 by McCulloch and Pitts



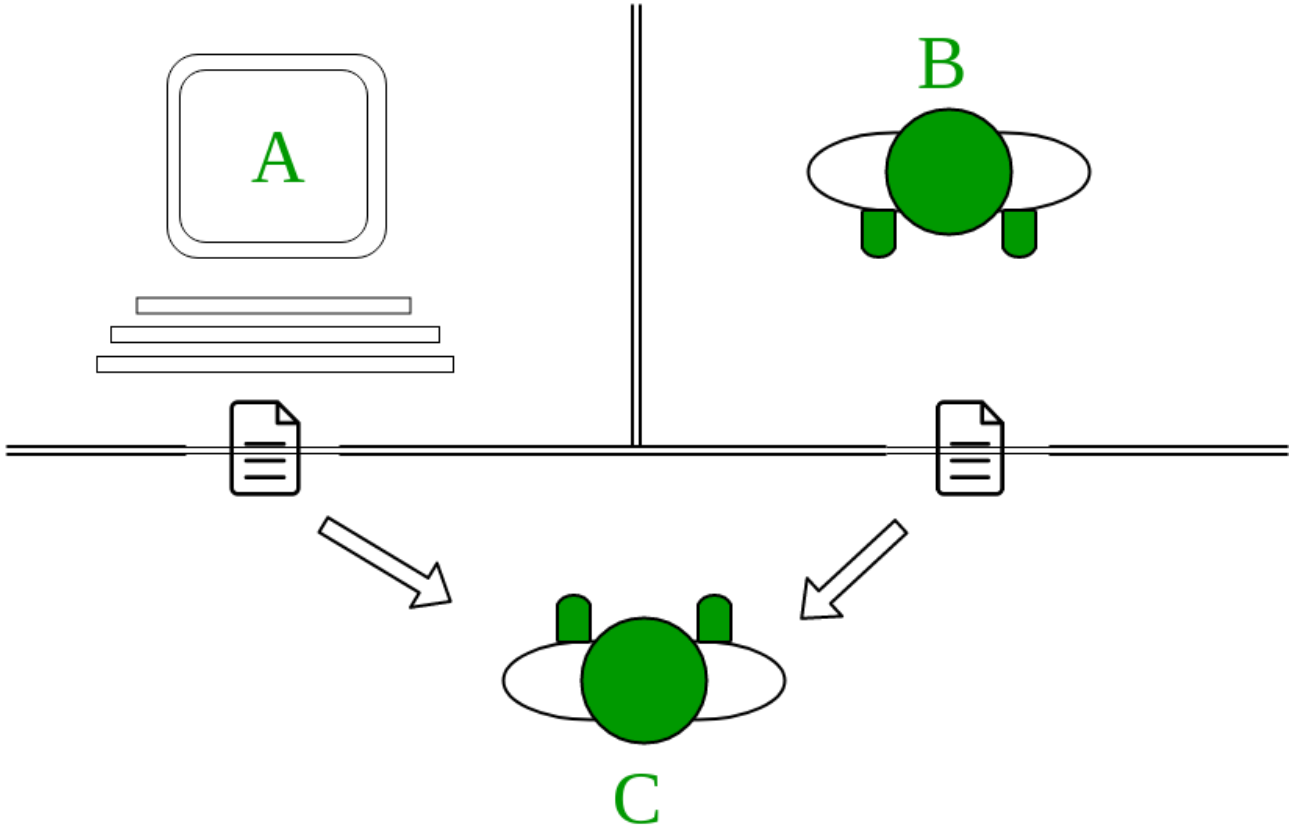
맥컬록

피츠

제안한 최초의 인공신경망 개념



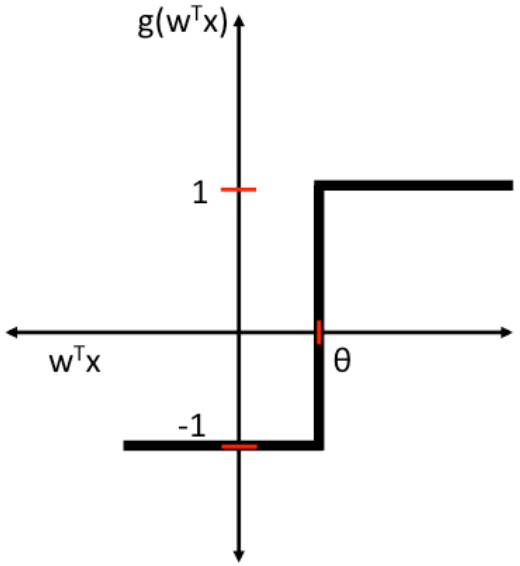
- Turing Test (1950) : Alan Turing is called the father of computer science



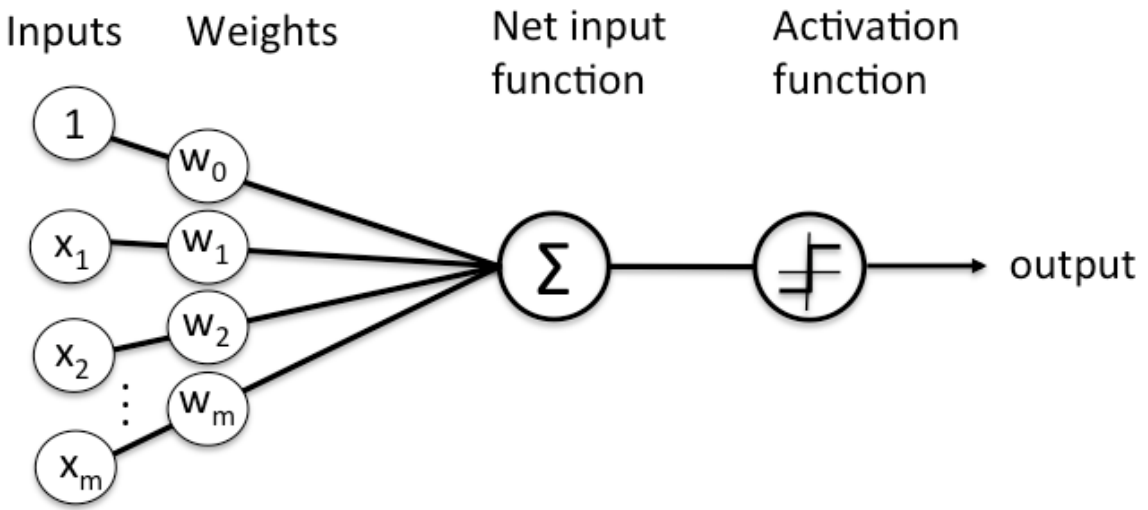
(source) https://www.geeksforgeeks.org/turing-test-artificial-intelligence/

The perceptron (1958)

- Rosenblatt's perceptron
  - ✓활성함수는 TLU(계단함수)



Unit step function.

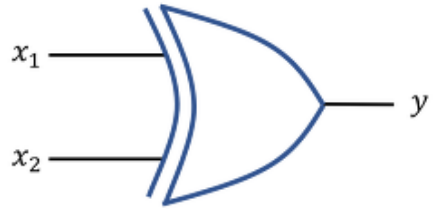


Schematic of Rosenblatt's perceptron.

(source) https://sebastianraschka.com/Articles/2015_singlelayer_neurons.html

인공지능 1차 겨울이 시작되다(1969)

❖ 문제의 시작 : XOR 게이트



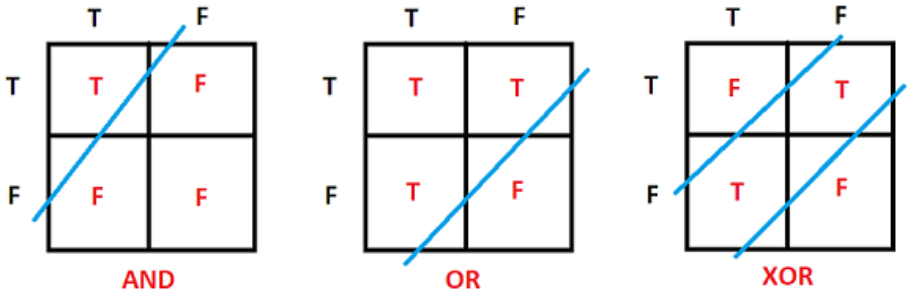
XOR Gate 의 회로기호

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

XOR Gate 의 입출력

```
XOR = !AND && OR
```

❖ 1개 퍼셉트론으론 간단한 선형 'XOR' 문제 풀수 없다(마빈 민스키, 1969)



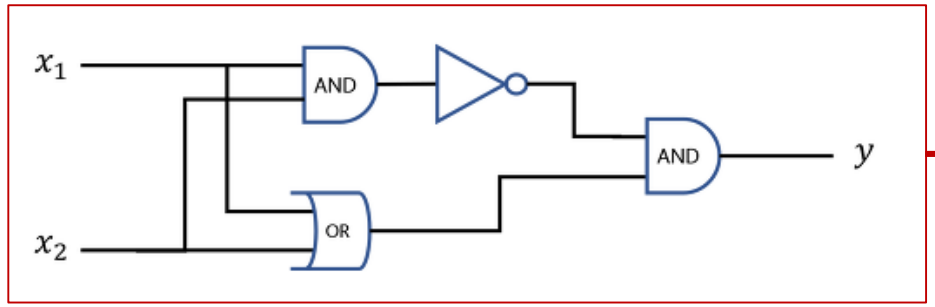
(source) https://gomguard.tistory.com/178

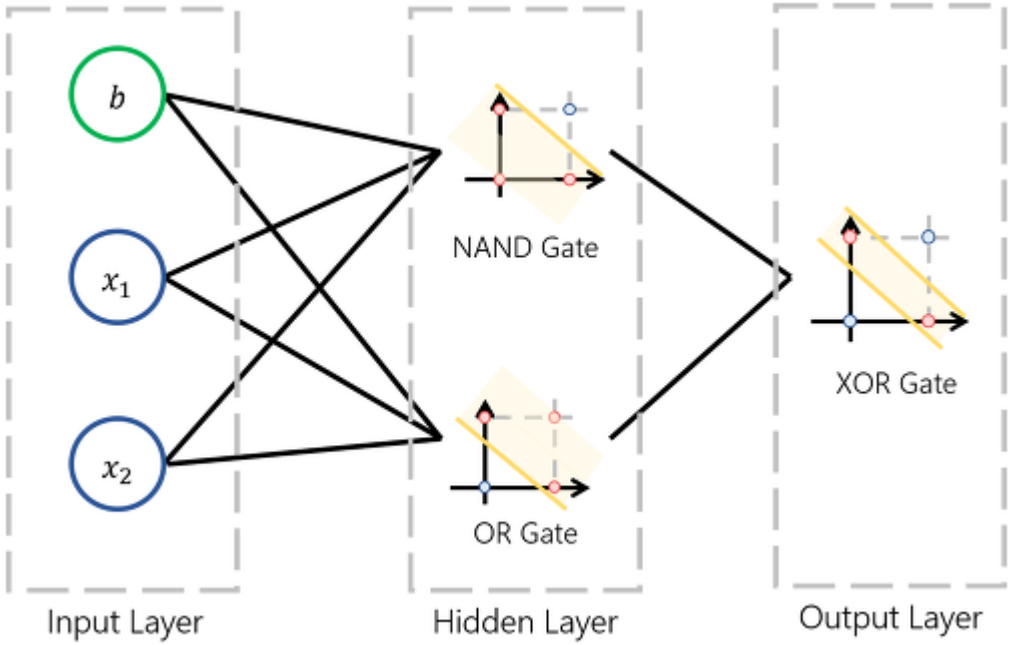
MLP : Multilayer Perceptron

❖ MLP로 'XOR' 문제 해결

- ✓ XOR 게이트는 'AND'와 'OR', 'NOT' 게이트의 조합
- ✓ Perceptron으논 'AND'나 'OR' 게이트 만들 수 있다.
- ✓ 여러 개의 Perceptron을 조합하면 'XOR' 게이트 만들 수 있다.

XOR = !AND && OR

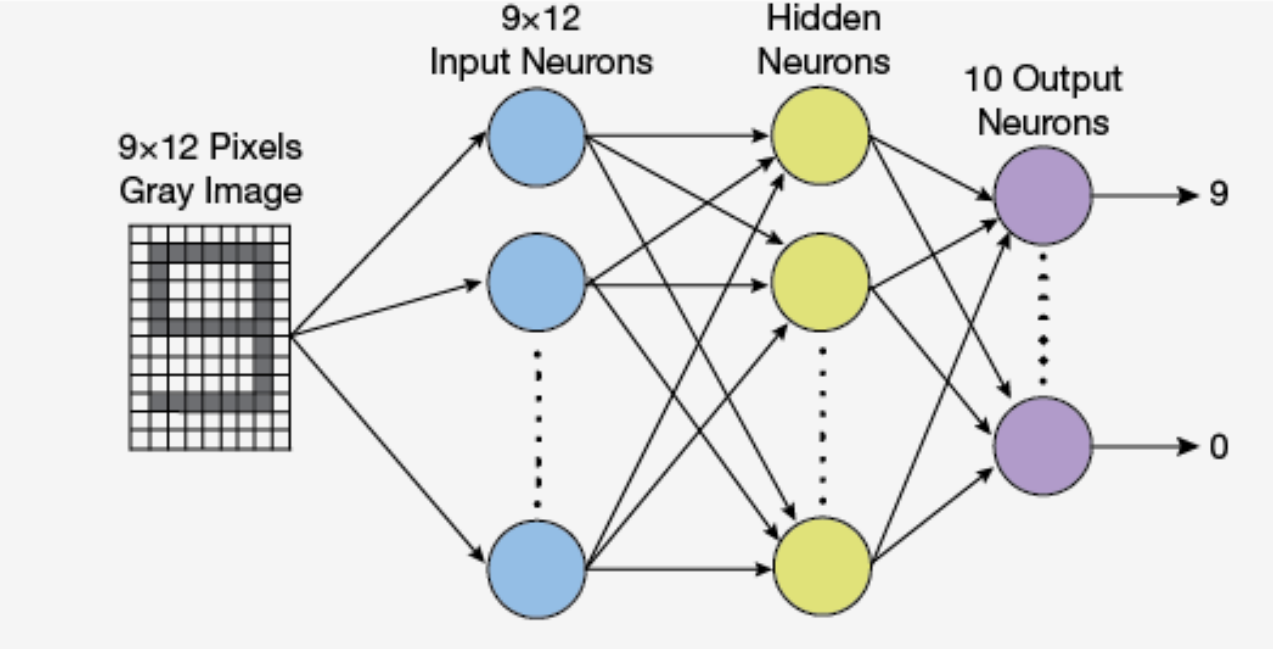




(source) https://gomguard.tistory.com/178

MLP의 성공사례

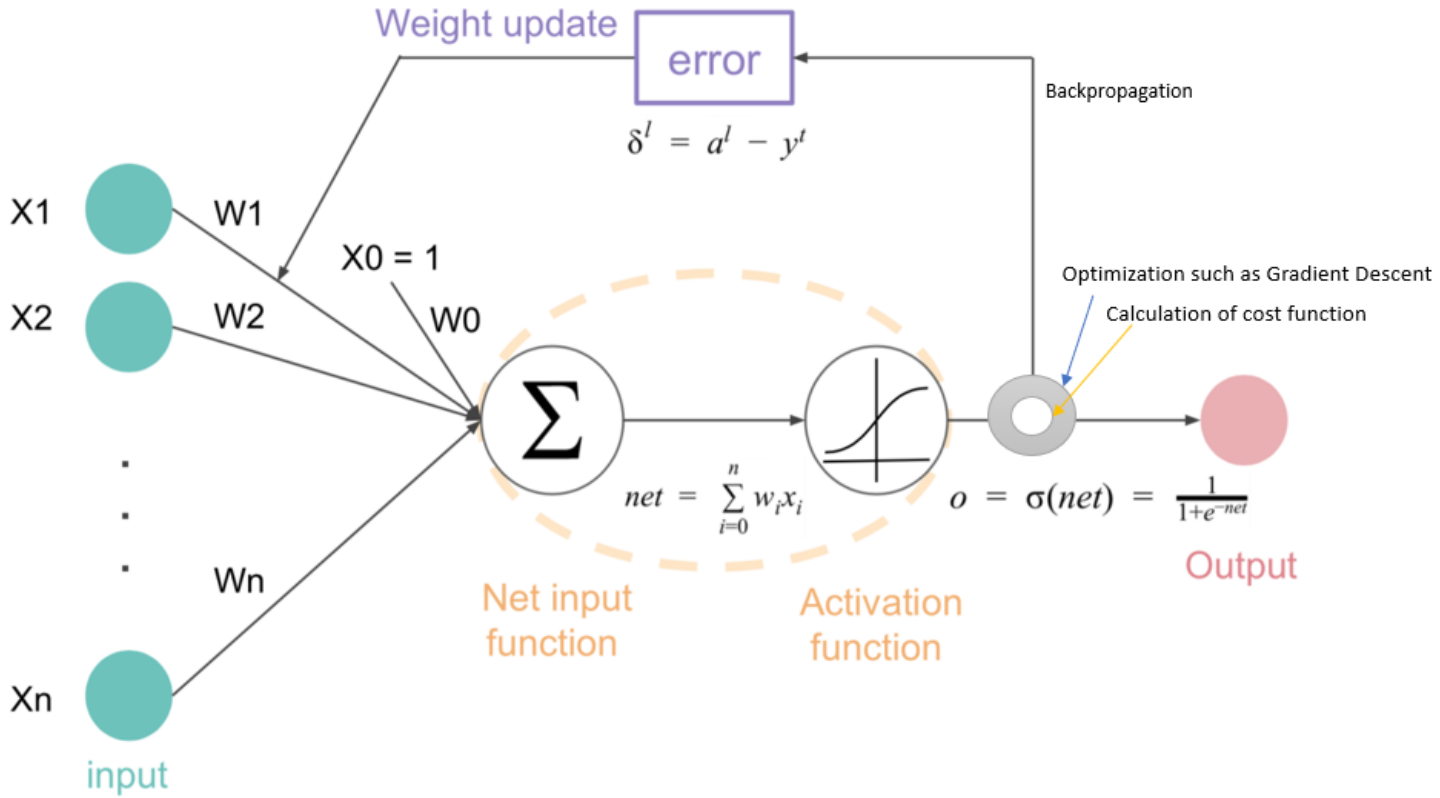
❖ 손글씨 분류 예제



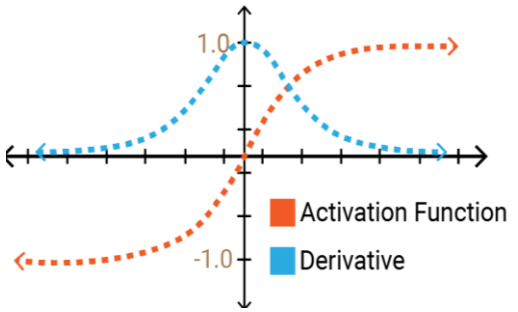
(source) https://www.aiche.org/resources/publications/cep/2018/june/introduction-deep-learning-part-1

Backpropagation로 AI 다시 봄이 시작된다

❖ 역전파알고리즘 (Geoffrey E. Hinton, 1986)



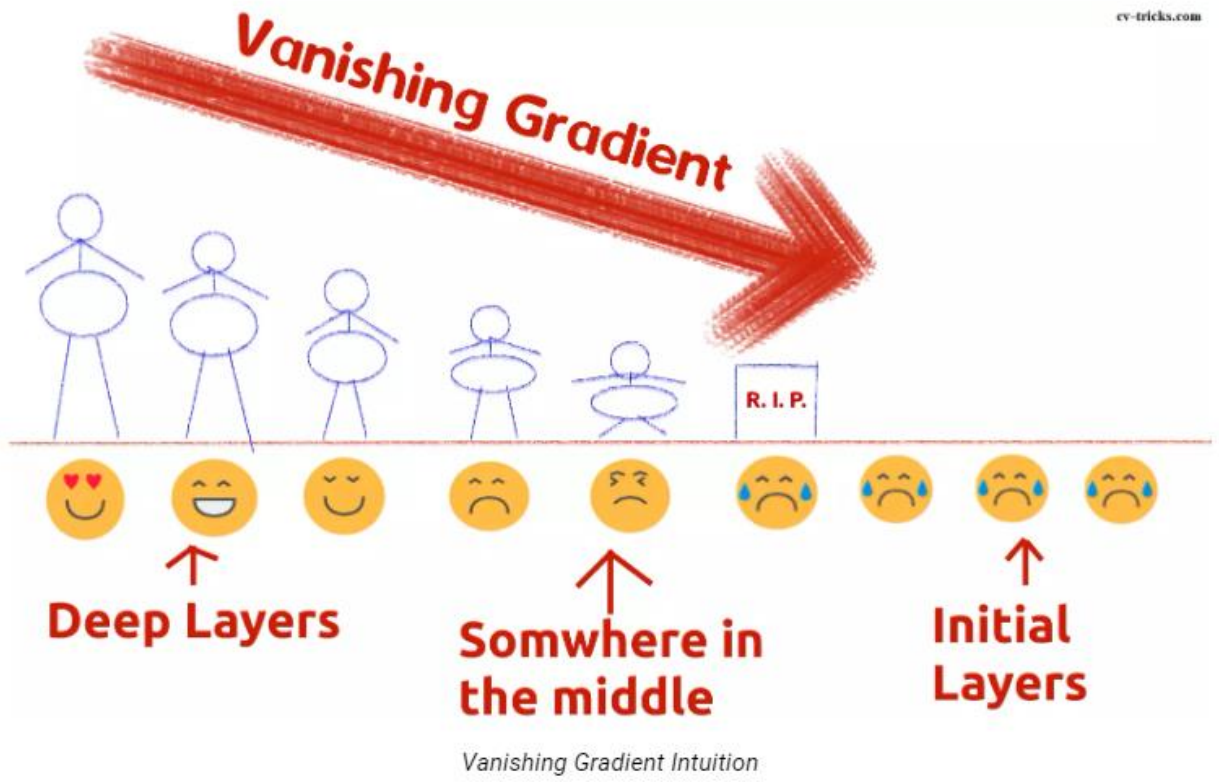
활성함수로 TLU(계단함수) 대신
미분가능한
Sigmoid 함수 사용



(source) https://datascience.stackexchange.com/questions/44703/how-does-gradient-descent-and-backpropagation-work-together

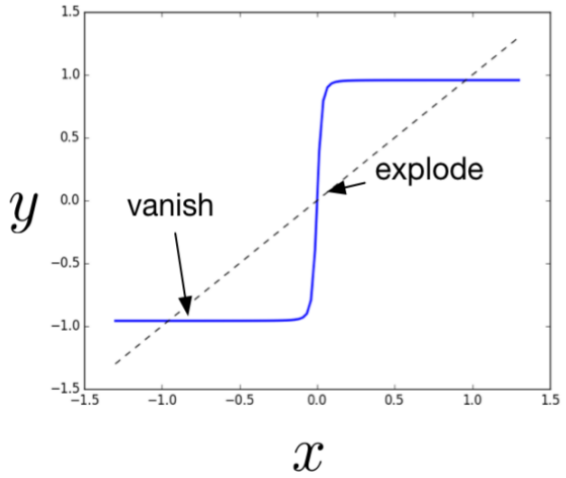
인공지능 2차 겨울 (1993~)

- RNN 처럼 많은 수의 은닉층이 있는 MLP에서는 VGP로 계산이 안됨
  - ✓긴 신경망 MLP에서는 값이 사라지는 VGP(Vanishing Gradient Problem) 문제 발생
  - ✓해결책으로 Long Short Term Memory (LSTM)가 등장함 (1997)



Vanishing Gradient Intuition

(문제의 원인) 시그모이드는 미분값이 사라짐과 발산함



(source) https://cv-tricks.com/keras/understand-implement-resnets/

인공지능 2차 겨울 극복기 (1993~2006)

KISTI SINCE 1962

❖ 문제점 및 해결

✓MLP의 오버피팅(Overfitting) 문제

- CNN의 가중치 공유
- Dropout (2014)

✓VGP 문제는

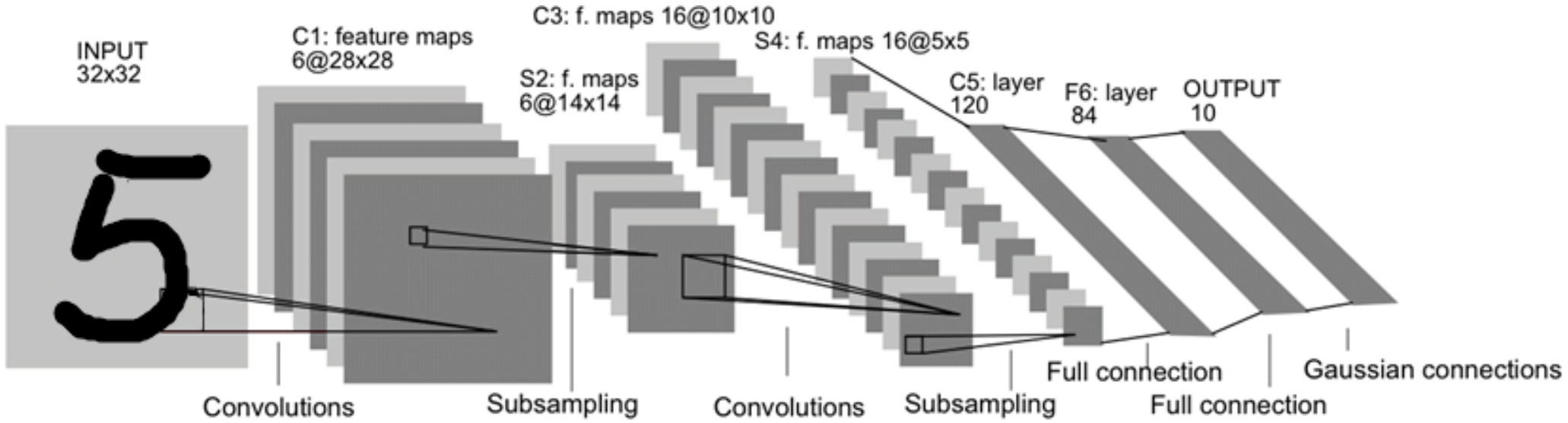
- 긴 MLP 문제로 값이 사람지는 것을 LSTM로 해결 시도

✓MLP의 계산 느린 문제는 GPU 등 가속기 등장

- Accelerator computing, such as GPU, TPU, Xeon Phi, etc. not yet

힘겨울 인공지능 겨울(II) 극복기: CNN 등장

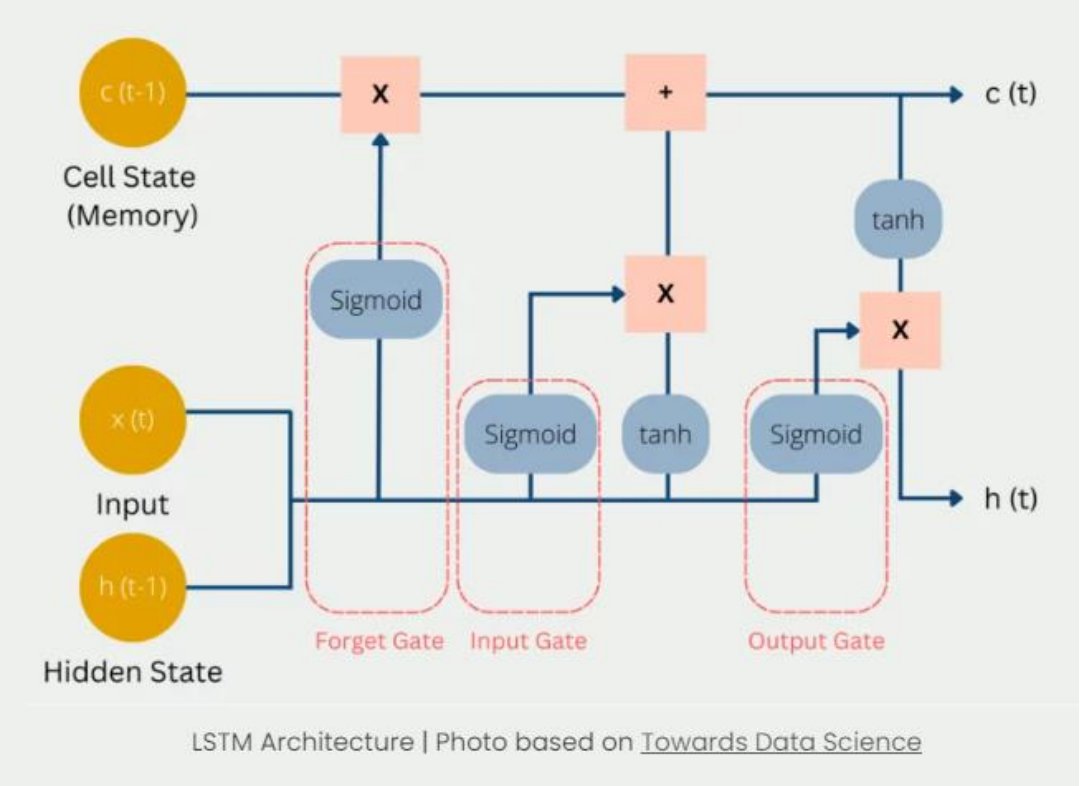
- CNN 알고리즘 적용 (LeNet-5, 1998)
  - ✓적은수의 가중치가 중요함
    - 가중치 공유로 파라미터 개수를 줄일 수 있음
    - 기존 밀집층의 Overfitting 문제 해결



(source) https://www.researchgate.net/figure/The-architecture-of-LeNet-5-23-a-CNN-used-for-digits-recognition-for-the-MNIST-dataset_fig2_321665783

힘겨울 인공지능 겨울(II) 극복기: LSTM 등장

- 순환신경망 RNN의 VGP를 해결한 LSTM (1997)
  - ✓긴 계산이 가능하도록 메모리를 도입하여 게이트로 조절



LSTM Architecture | Photo based on Towards Data Science

(source) https://databasecamp.de/en/ml/lstms

Deep Learning in Neural Networks : ReLU (2010~)

- ReLU 선형 활성함수가 중요한 기여를 한다.
  - ✓첫 사용은 이미 1969년 (Fukushima)

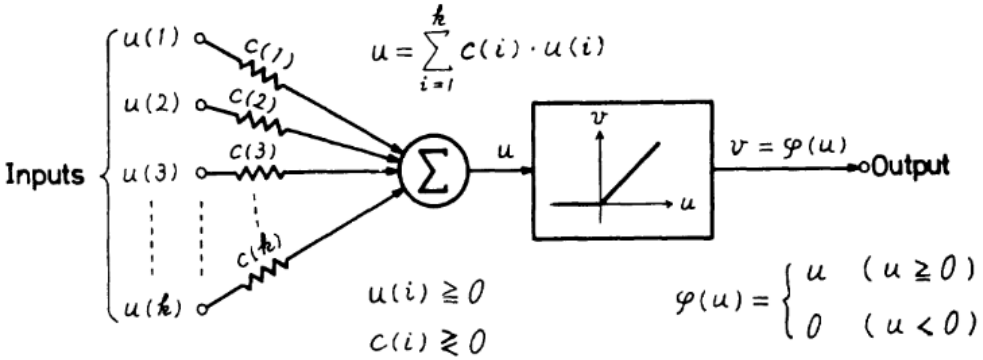
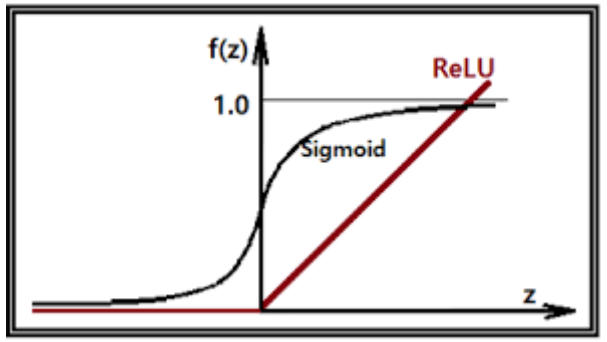


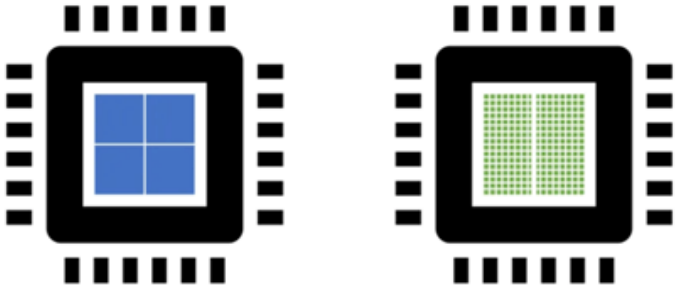
Fig. 3. Analog threshold element employed in the feature extractor network.



✓최근은 "Rectified Linear Units improve RBM' (G.Hinton, 2010년)

(source) https://stats.stackexchange.com/questions/447674/when-was-the-relu-function-first-used-in-a-neural-network

❖ GPU 디바이스에서 계산

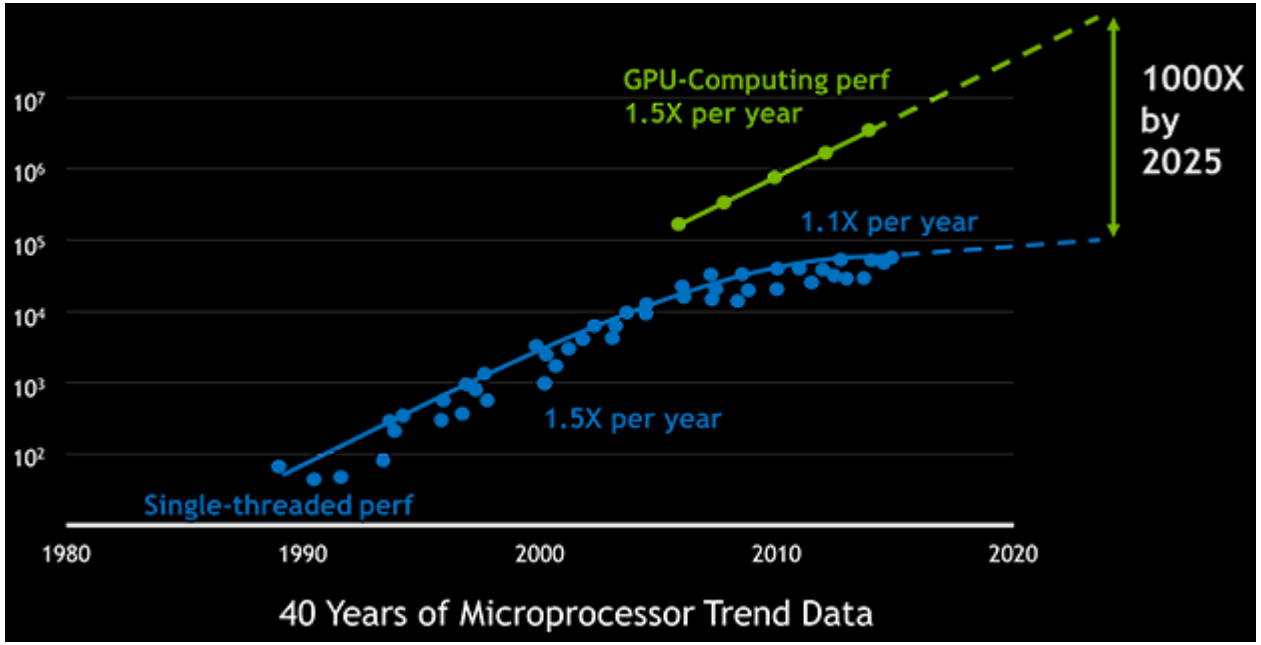


CPU	GPU
Central Processing Unit	Graphics Processing Unit
4-8 Cores	100s or 1000s of Cores
Low Latency	High Throughput
Good for Serial Processing	Good for Parallel Processing
Quickly Process Tasks That Require Interactivity	Breaks Jobs Into Separate Tasks To Process Simultaneously
Traditional Programming Are Written For CPU Sequential Execution	Requires Additional Software To Convert CPU Functions to GPU Functions for Parallel Execution

GPU vs CPU processing. Source: towardsdatascience.com

(source) https://www.cherryservers.com/blog/what-is-gpu-computing

Nvidia Investor Day 2017 Presentation.

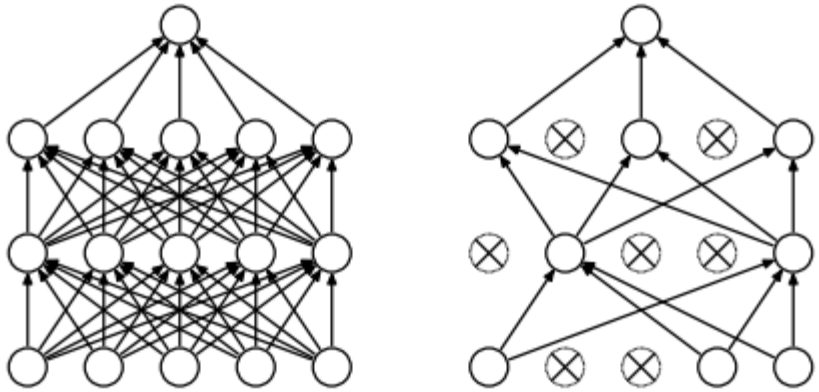


Huang's law extends Moore's law
"the performance of GPUs will more than double every two year"

Deep Learning in Neural Networks: DropOut

❖ 기존 MLP의 과적합 발생을 해소학 위해서 Dropout을 도입해보자

"A Simple Way to Prevent Neural Networks from Overfitting" (2014)



(a) Standard Neural Net (b) After applying dropout.

Figure 1: Dropout Neural Net Model. **Left**: A standard neural net with 2 hidden layers. **Right**: An example of a thinned net produced by applying dropout to the network on the left. Crossed units have been dropped.

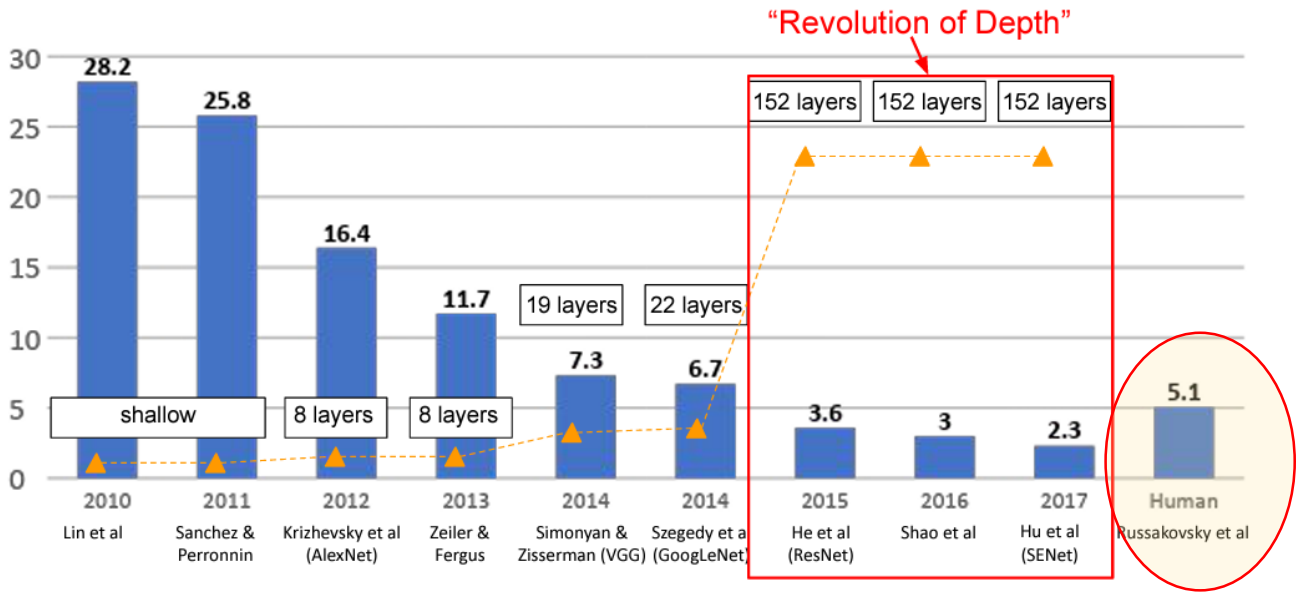
(source) https://www.cs.toronto.edu/~rsalakhu/papers/srivastava14a.pdf

인공지능 성공 사례 : ILSVRC winners

❖ History of CNN

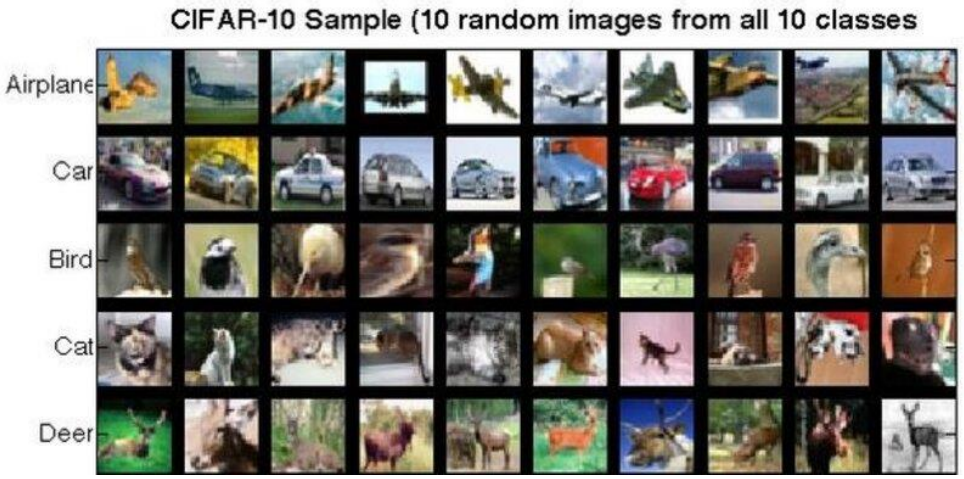
✓ ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners

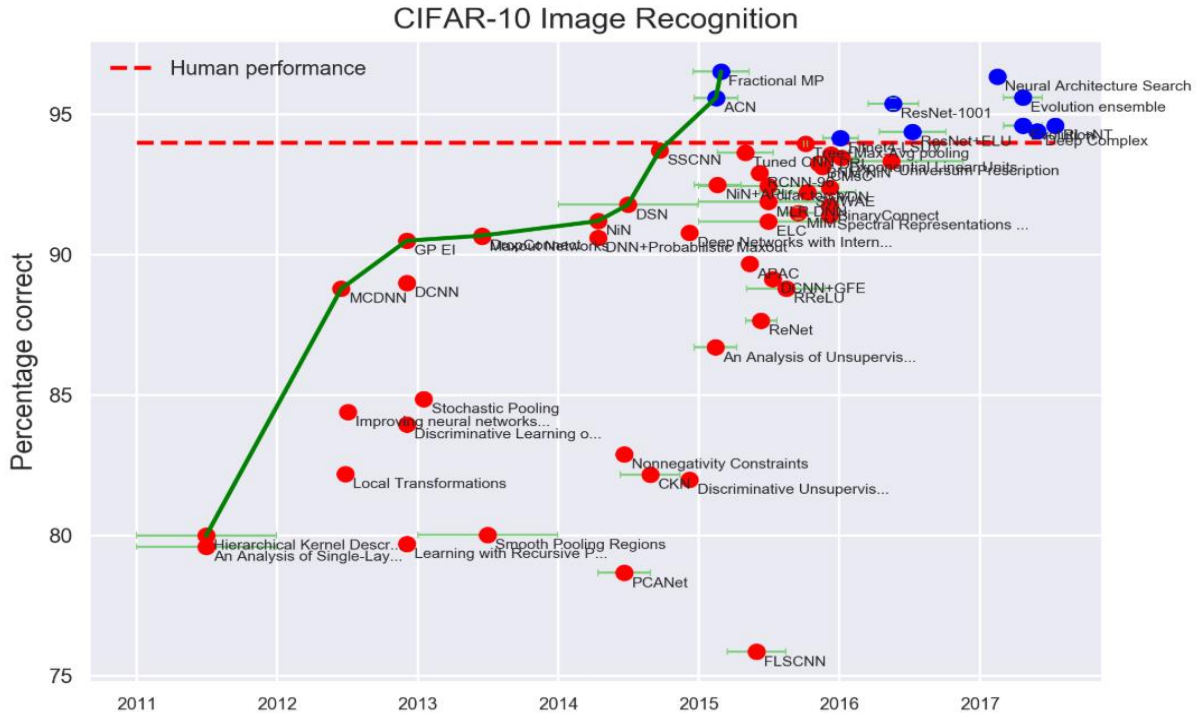
- Human Error ~ 5.1%



(source) https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf

인공지능 성공 사례 : CNN 알고리즘 등장 (2015~)

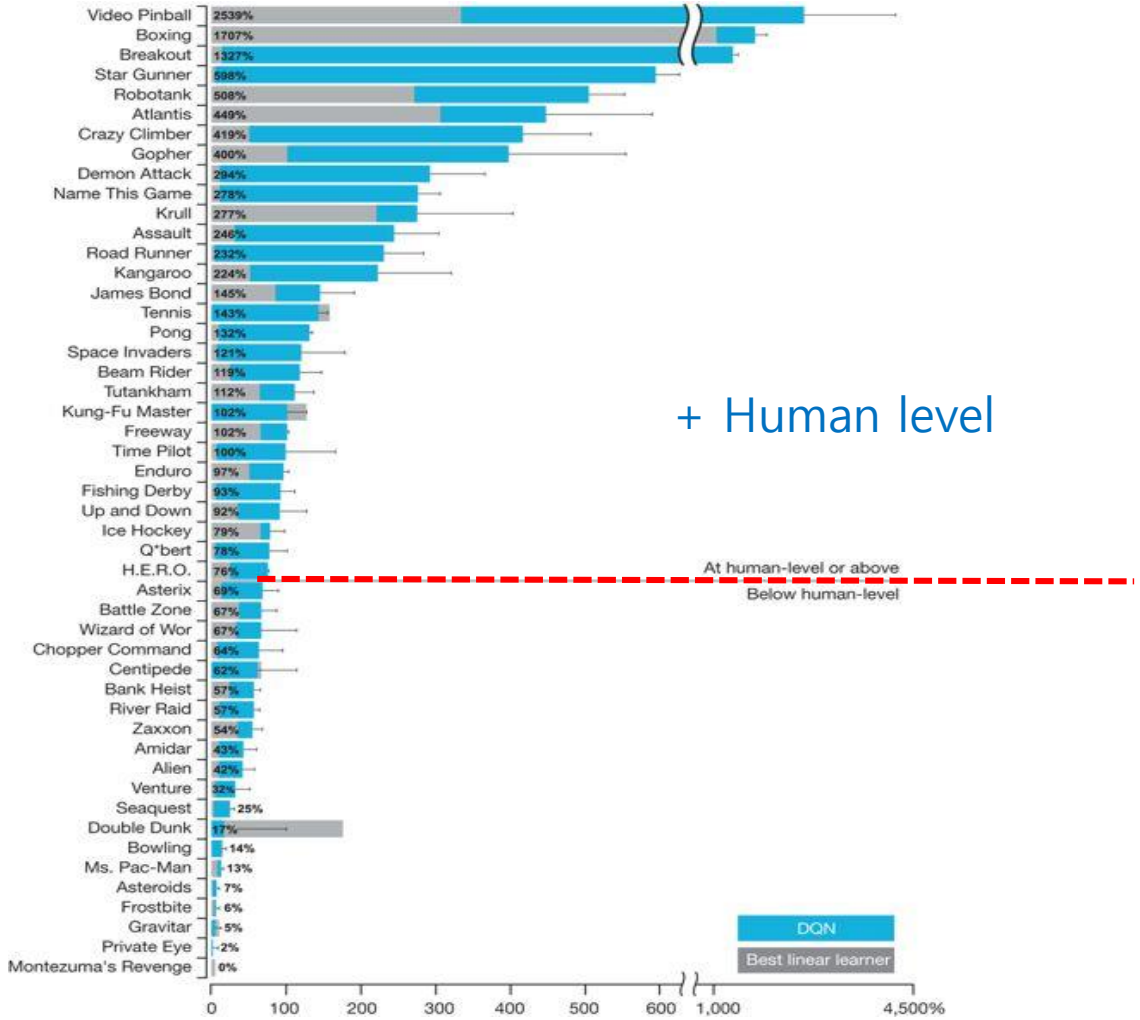




https://www.eff.org/ai/metrics

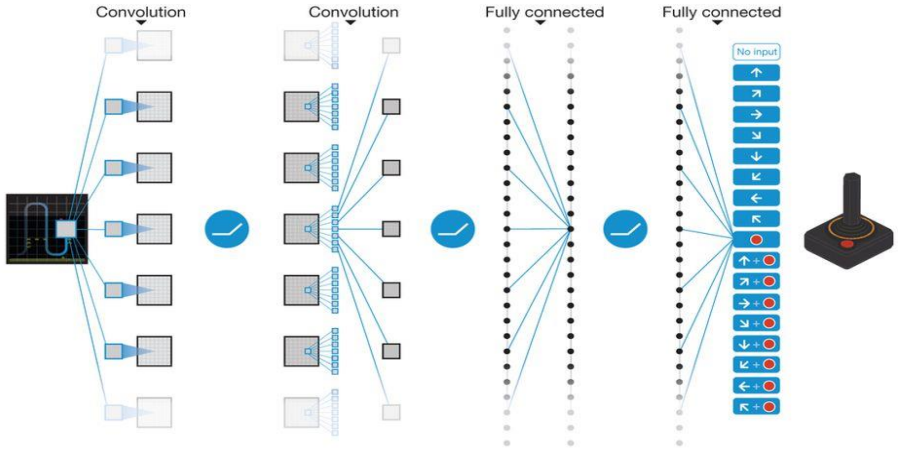
(source) https://www.eff.org/ai/metrics

인공지능 성공 사례 : 딥마인드 강화학습 (2015)



+ Human level

Google Deep Mind, Nature 2015



(source) Human-level control through deep reinforcement learning, nature 2015 (Mnih et.al.)

인공지능 성공 사례 : AlphaGo (2017)

- AlphaGo
  - ①AlphaGo-Supervised
  - ②AlphaGo-Lee
  - ③AlphaGo-Zero

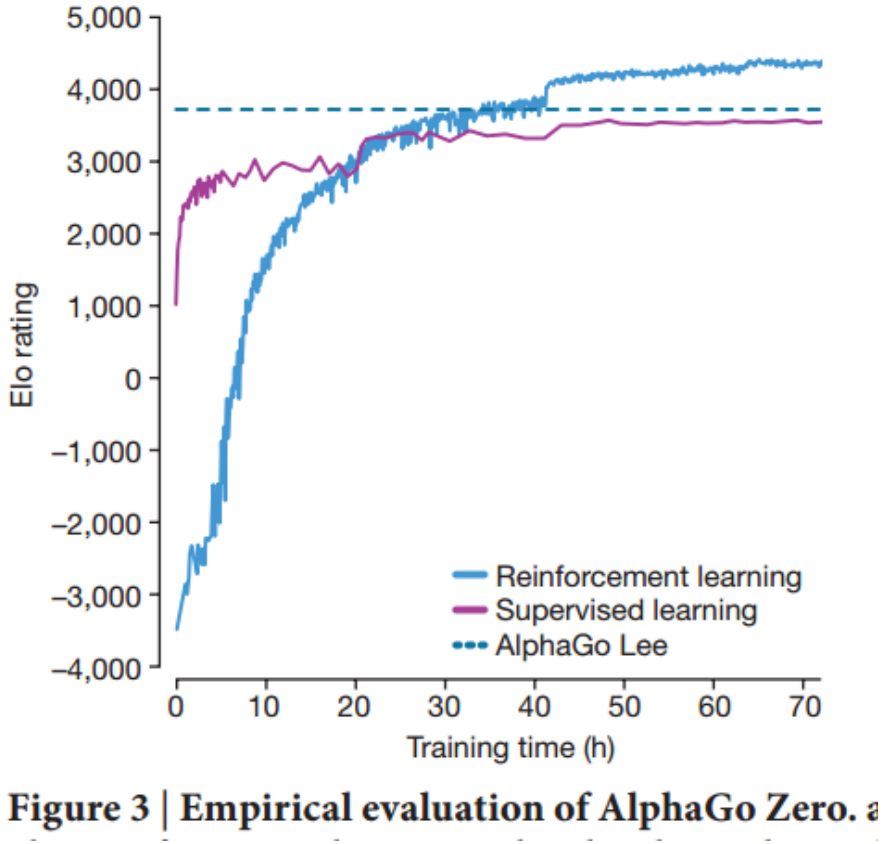
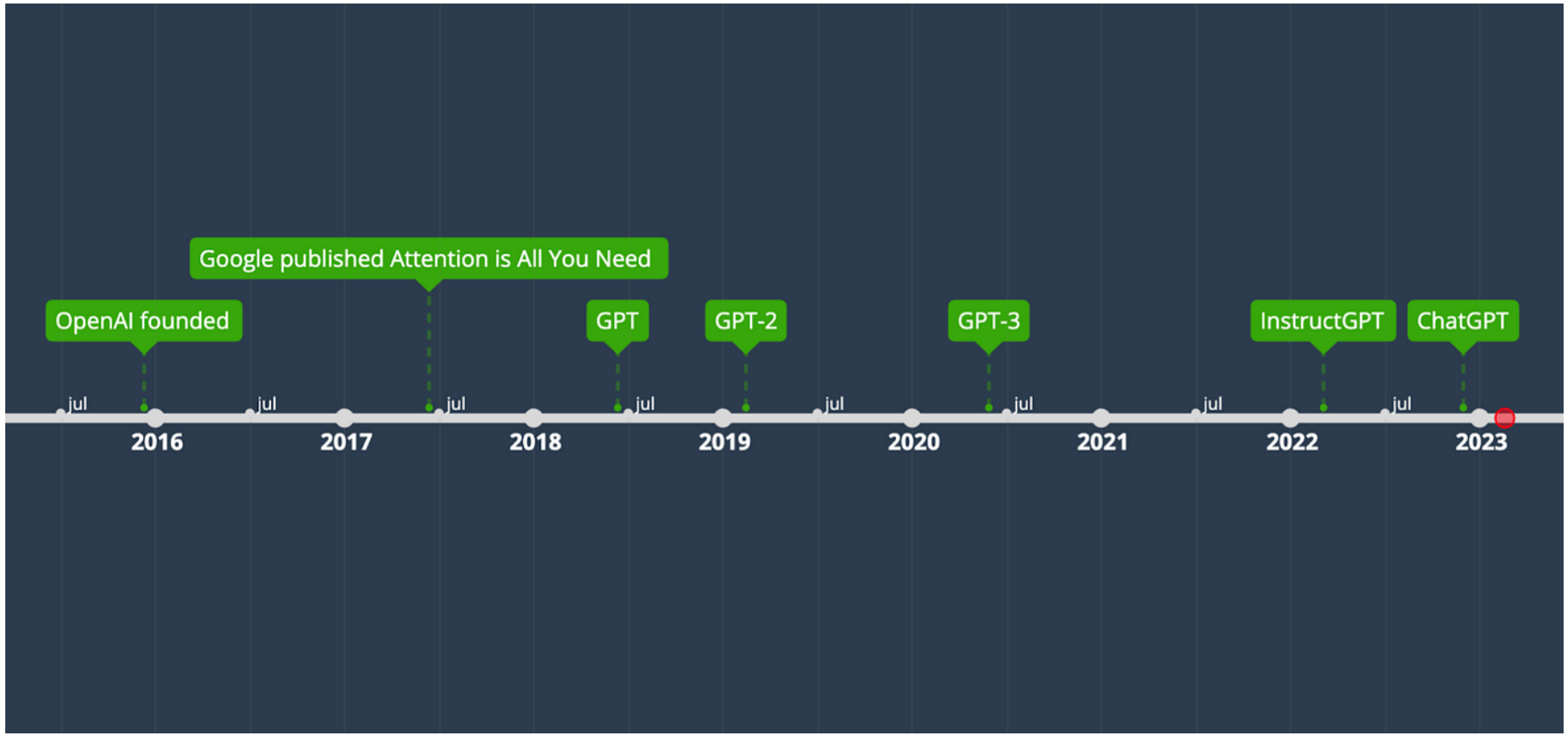


Figure 3 | Empirical evaluation of AlphaGo Zero. a

(Source) Mastering the game of Go without human knowledge , David Silver, et al. Nature(2017)

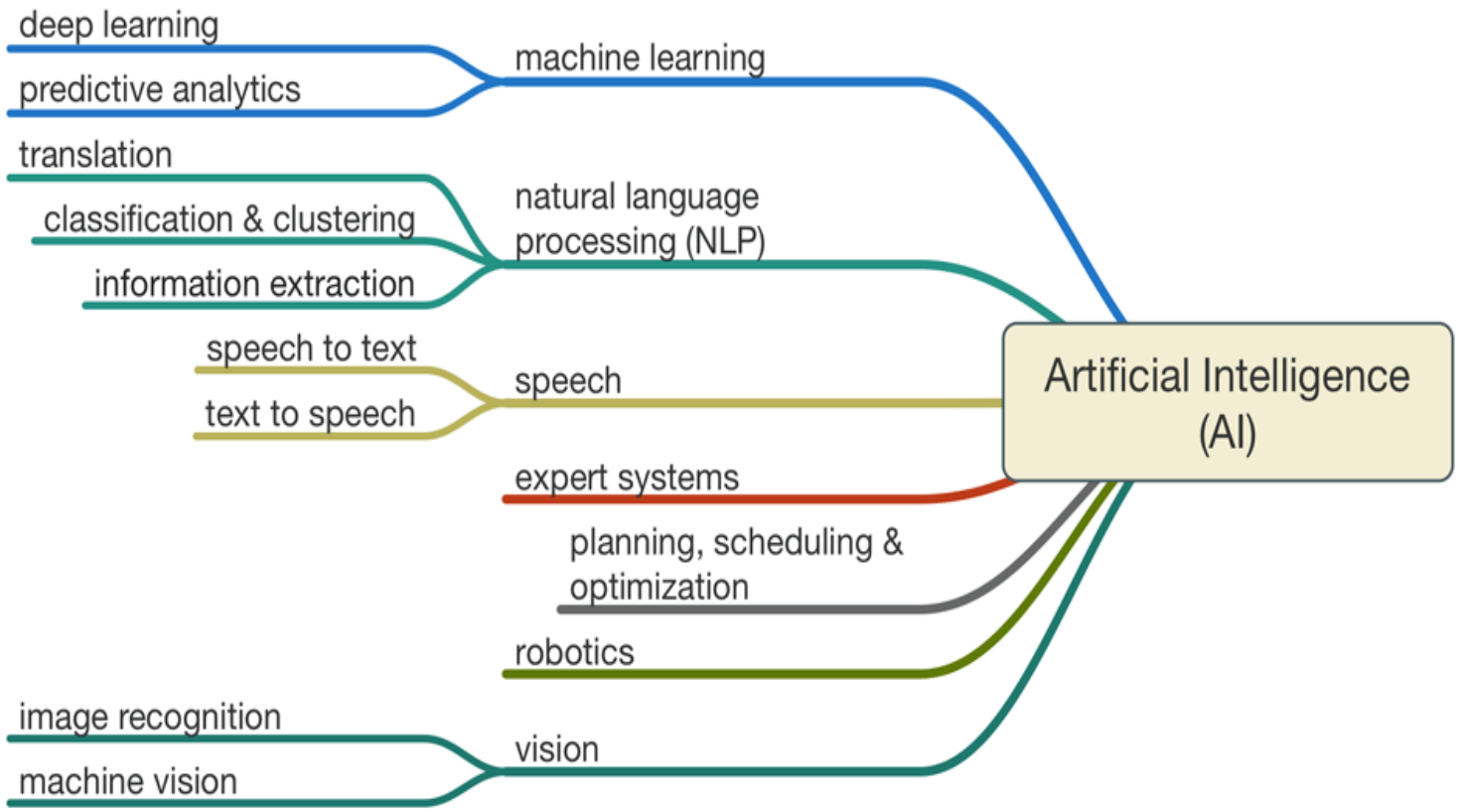
인공지능 성공 사례 : ChatGPT



(source) https://bootcamp.uxdesign.cc/how-chatgpt-really-works-explained-for-non-technical-people-71efb078a5c9?gi=8e09468637c6

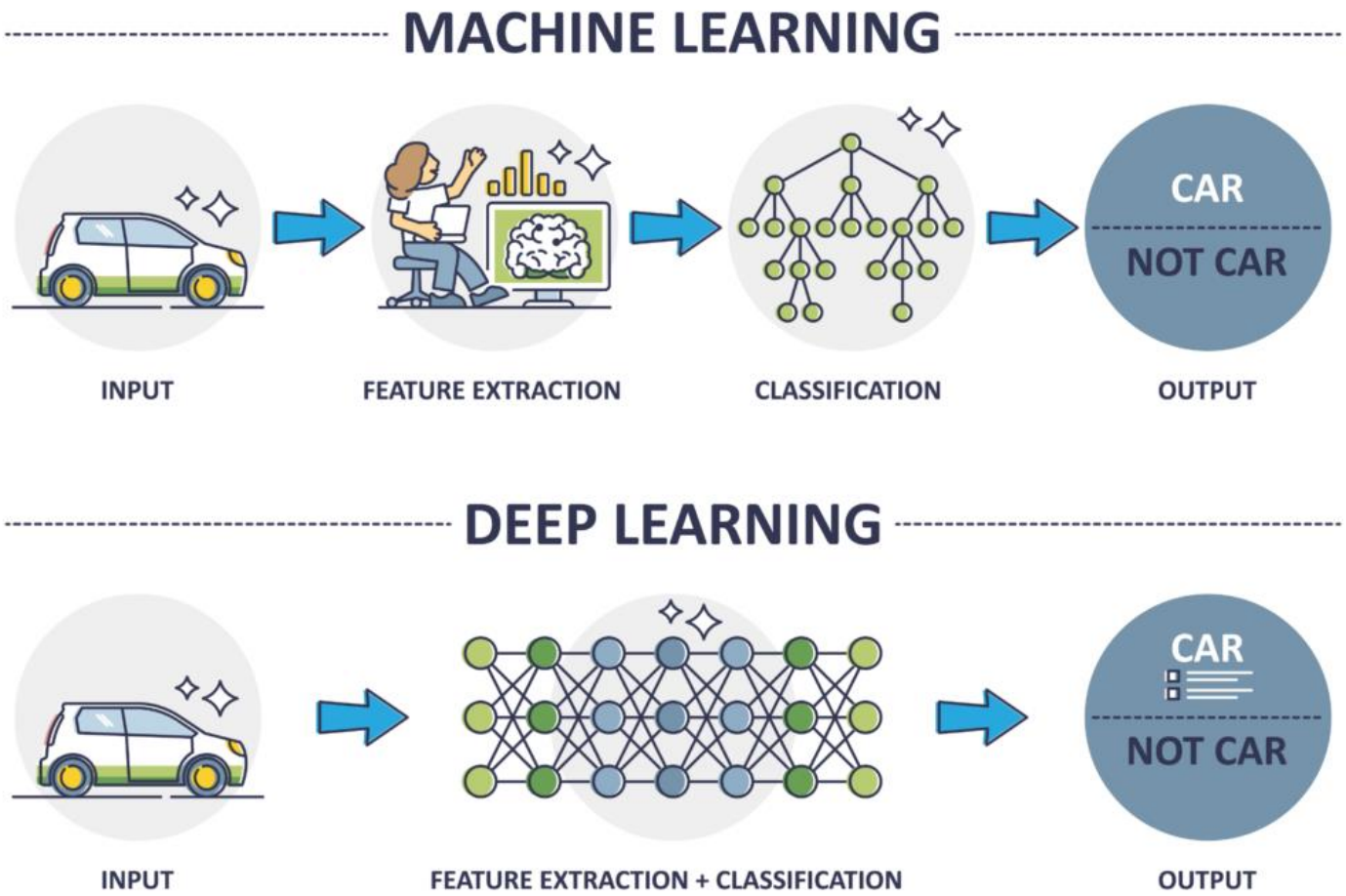
딥러닝 및 코랩 소개

인공지능의 분류



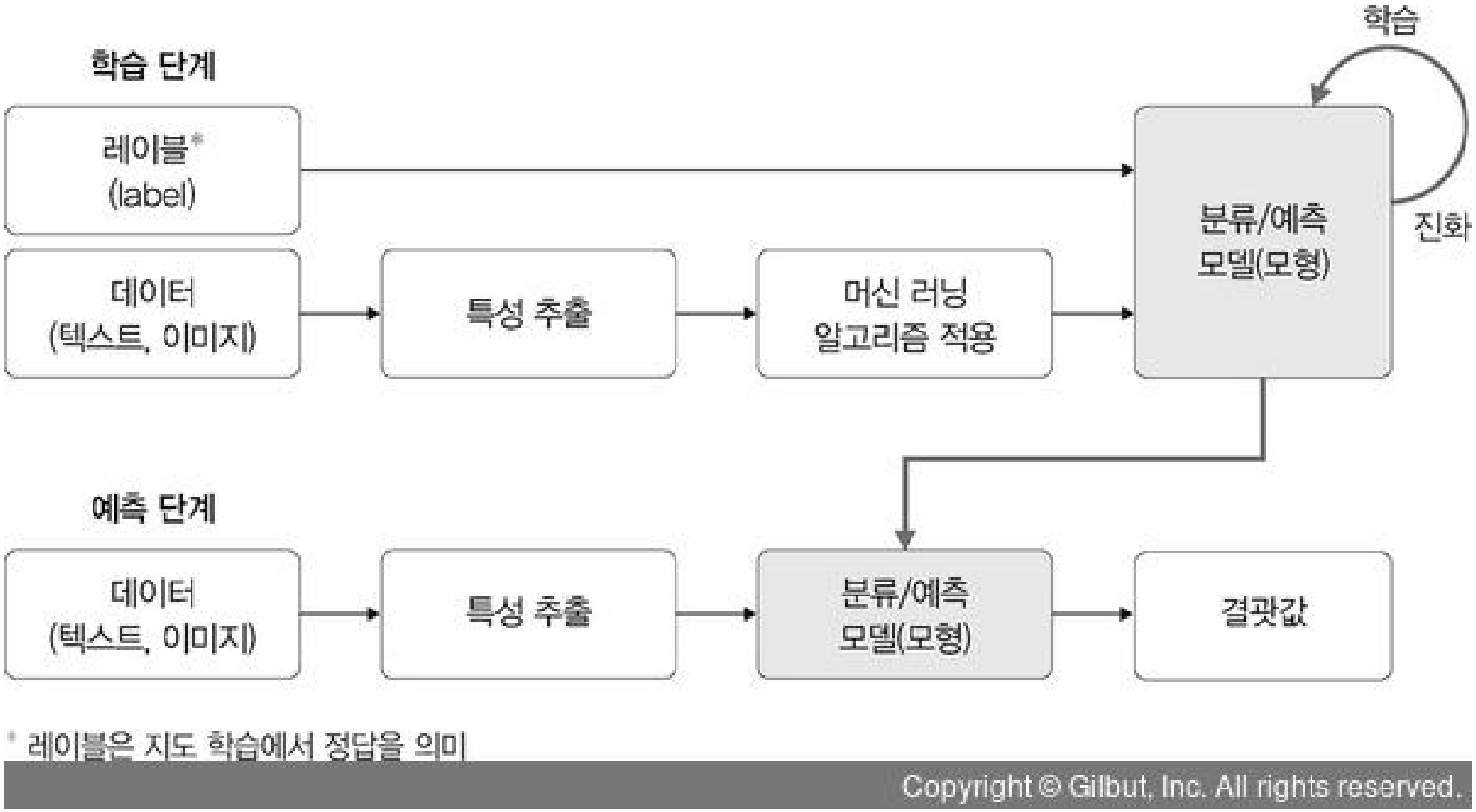
자료: https://www.eyerys.com/articles/paving-roads-artificial-intelligence-its-either-us-or-them

딥러닝과 머신러닝 비교



(source) https://www.ait.de/en/deep-learning/

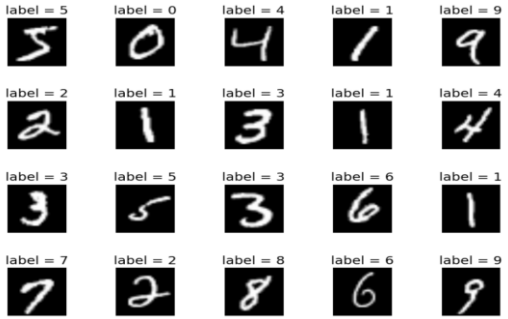
머신러닝 학습 과정 : 특성추출



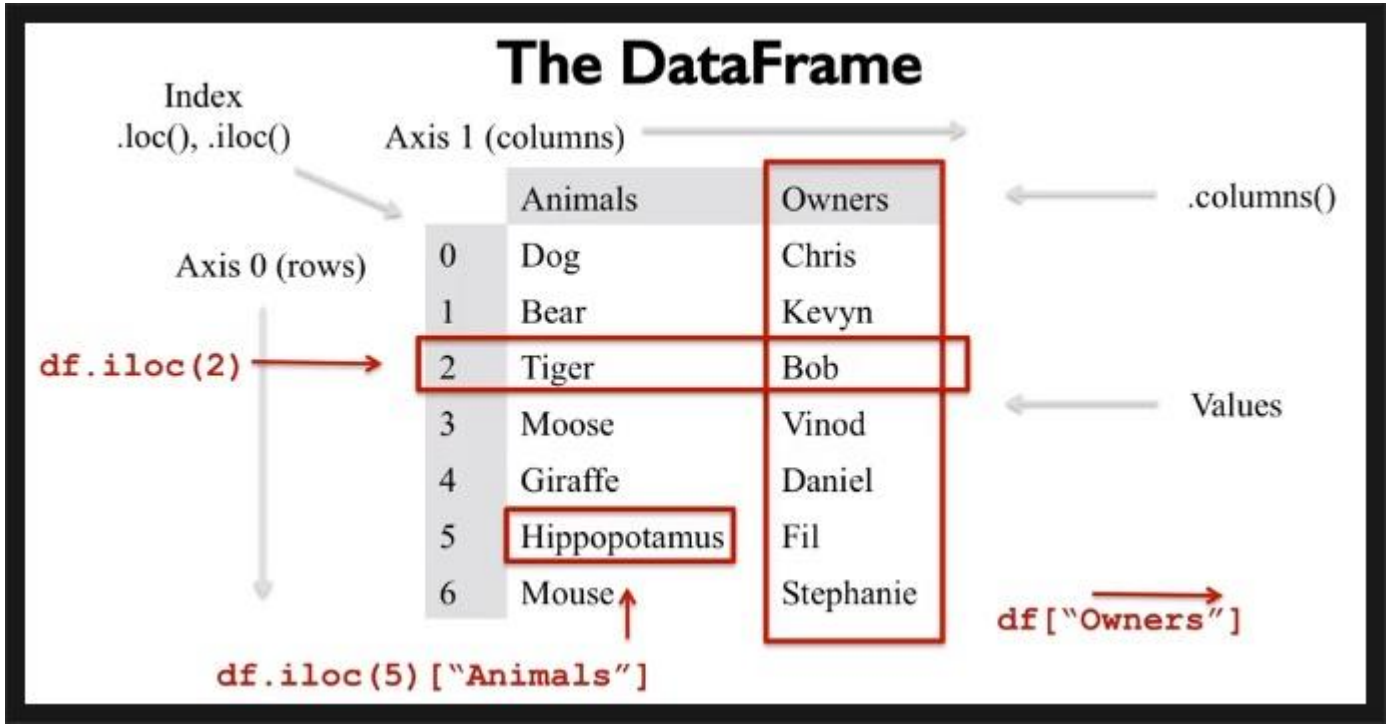
이미지 출처: https://thebook.io/080263/0006/

지도학습과 레이블(Label)

MNIST

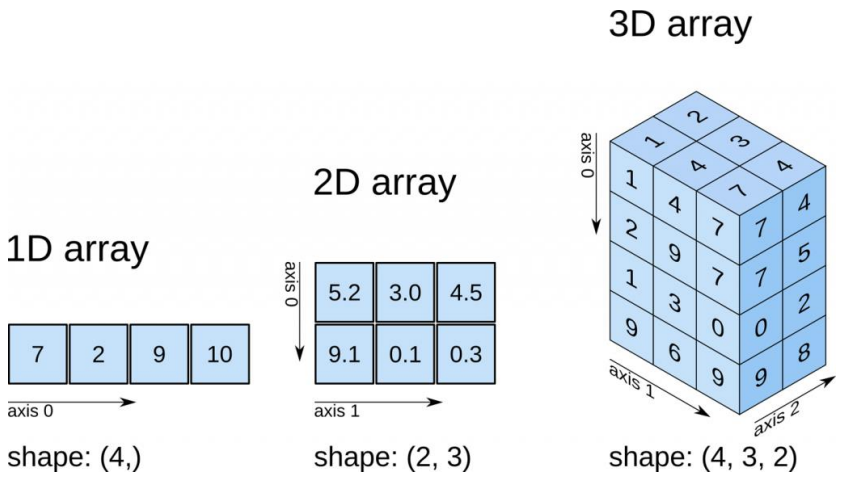


Import pandas as pd

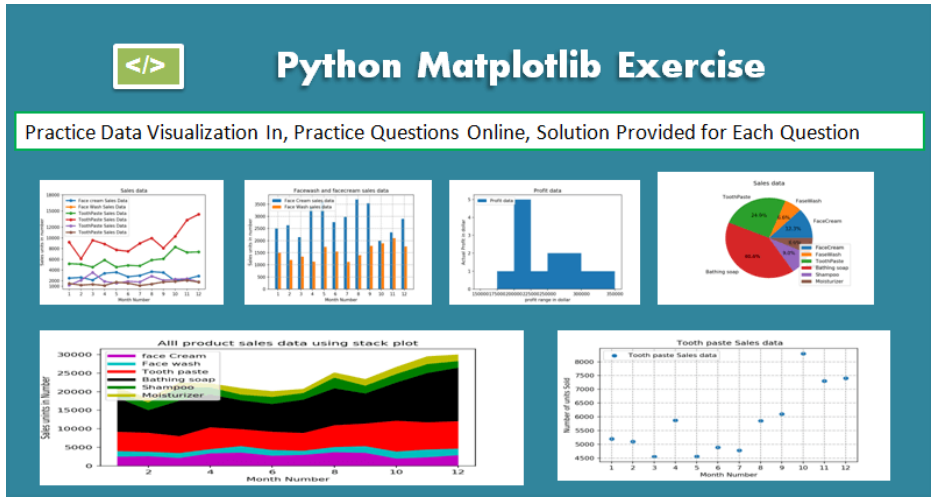


Import numpy as np

```
11
12 import numpy
13 import pylab
14
15 t = numpy.arange(0.0, 1.0+0.01, 0.01)
16 s = numpy.cos(2*2*numpy.pi*t)
17 pylab.plot(t, s)
18
19 pylab.xlabel('time (s)')
20 pylab.ylabel('voltage (mV)')
21 pylab.title('About as simple as it gets, folks')
22 pylab.grid(True)
23 pylab.savefig('simple_plot')
24
25 pylab.show()
```

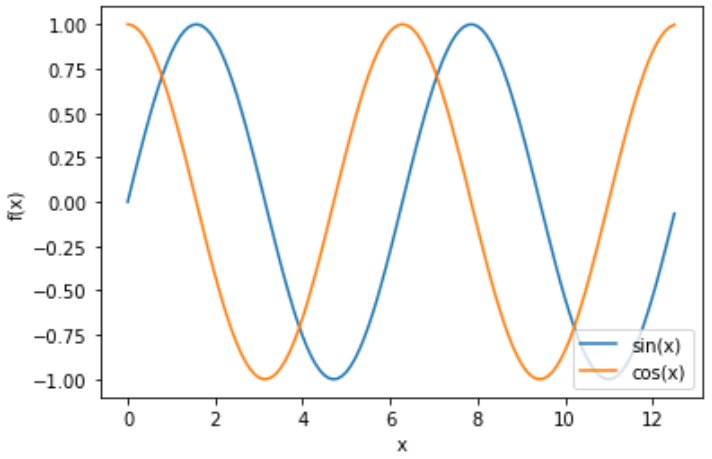


Import matplotlib.pyplot as plt




```
In [1]: import numpy as np
        import matplotlib.pyplot as plt

        x = np.arange(0, np.pi*4, 0.1)
        y1 = np.sin(x)
        y2 = np.cos(x)

        plt.plot(x,y1, label='sin(x)')
        plt.plot(x,y2, label='cos(x)')

        plt.xlabel('x')
        plt.ylabel('f(x)')

        plt.legend(loc='lower right')
```

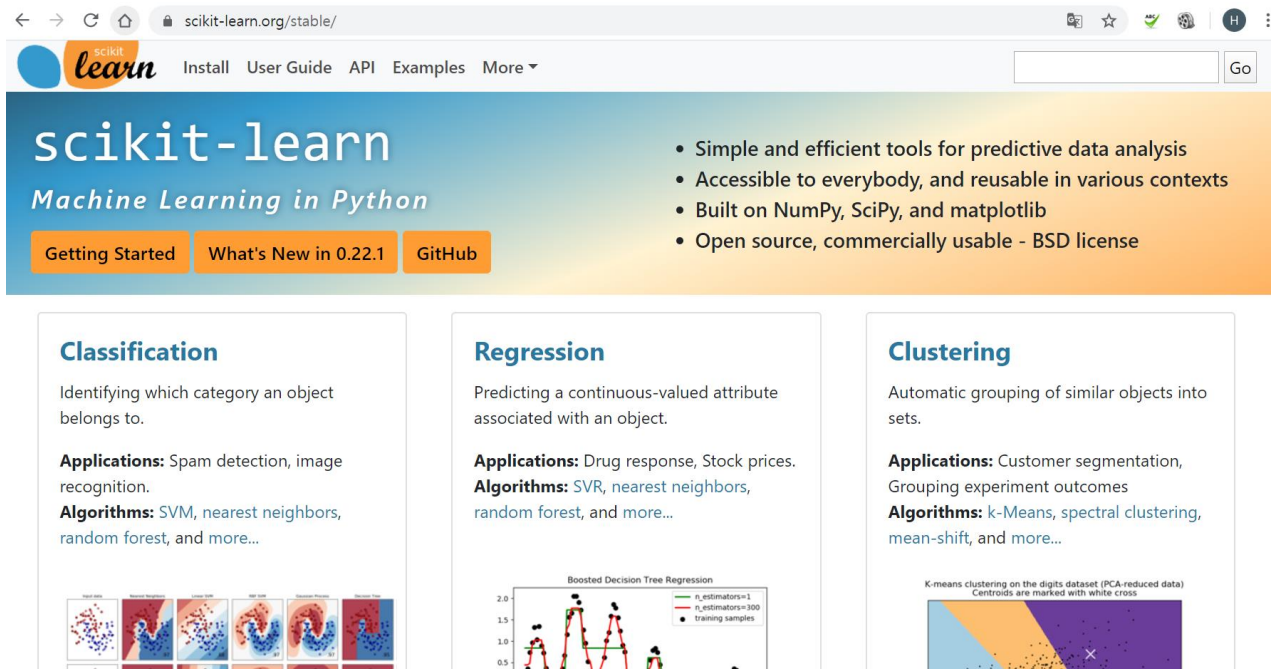
```
Out[1]: <matplotlib.legend.Legend at 0x23c8b058b08>
```



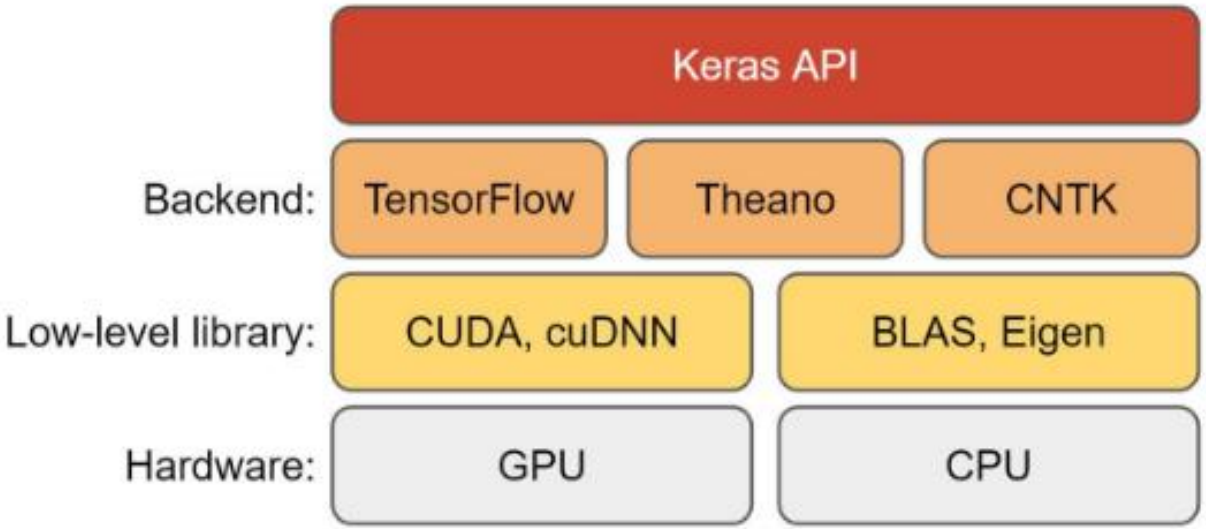
Scikit-learn

```
from sklearn.preprocessing import LabelEncoder

y = LabelEncoder().fit_transform(y)
```

파이썬 머신러닝 중에서 가장 많이 사용되는 라이브러리



Keras and Tensorflow

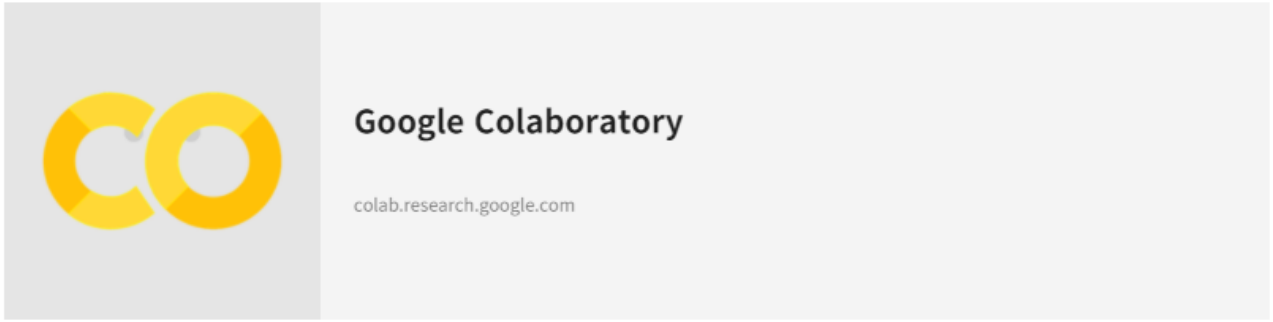


이미지 출처: https://www.digikey.de/en/maker/projects/getting-started-with-machine-learning-using-tensorflow-and-keras/0746640deea84313998f5f95c8206e5b

코랩 Colab이란?

- Colaboratory는 완전히 클라우드에서 실행되는 무료 Jupyter 노트 환경
  - ✓즉, **브라우저에서 클라우드 환경을 이용해 Python 코드를 실행할 수 있으며,**
  - ✓**무료로 GPU와 TPU 등의 컴퓨팅 자원을 함께 사용할 수 있다.**
  - ✓Colaboratory를 사용하면 브라우저를 통해 무료로 코드를 작성 및 실행하고
  - ✓분석을 저장 및 공유하며, 강력한 컴퓨팅 리소스를 이용할 수 있습니다.
- **Google Colaboratory 접속하기**

https://colab.research.google.com/



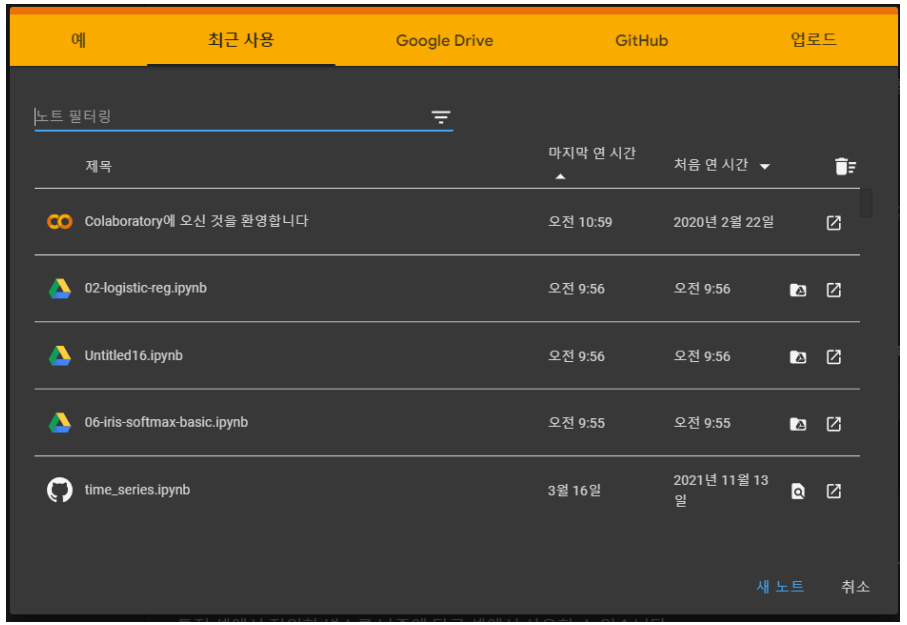
Google Colab 스펙

❖ Google Colab 스펙

- ✓CPU : 제온
- ✓Memory : 13GB
- ✓HDD : 320GB
- ✓GPU : NVIDIA Tesla K80

❖ 파일 생성/접근 방법

- ✓개인 구글 개정으로 접근
- ✓https://colab.research.google.com 접속
- ✓GOOGLE 새드라이브 PYTHON 노트탭 선택
- ✓실습은python3로 진행할 예정.



코랩 사용법

파일 이름 변경

Code cell, Text cell

- .ipynb 파일은 code cell과 text cell로 구성
- 각 셀 하단에 마우스를 대거나, 화면 좌상단 버튼으로 셀 추가 가능
- 셀 선택(마우스) 후 셀 우상단 삭제버튼으로 셀 삭제 가능

코랩 사용법

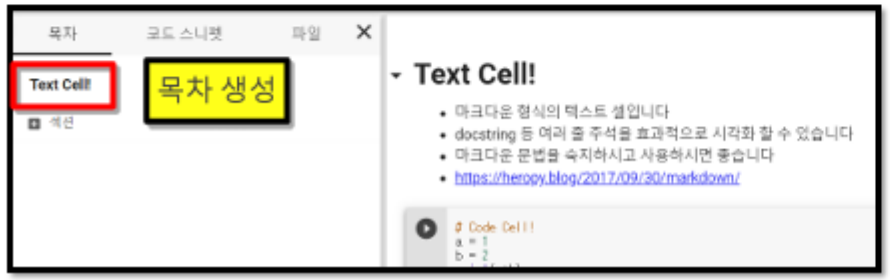
Code cell

- 일반적인 파이썬 코딩 방식과 동일
- 각 셀은 한번에 실행할 단위를 뜻함
- 실행 이후에도 메모리는 유지되어 다른 셀 실행 시 영향을 줌
  - 런타임 다시 시작 시 초기화

- 상단 메뉴의 런타임
  - 실행 중인 셀 중단
  - 런타임 다시 시작

코랩 사용법

- Text cell
  - 여러 줄 주석의 효과적인 시각화
  - 마크다운(Markdown) 문법
  - 자동 목차 생성



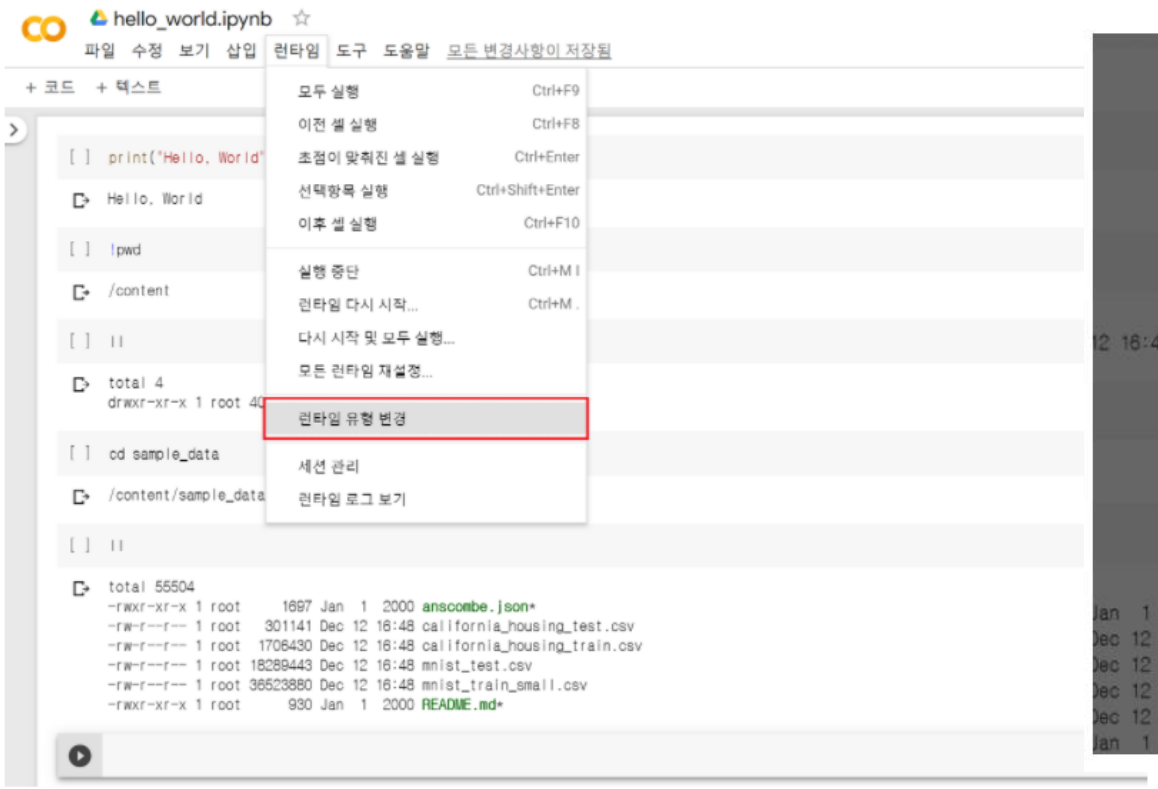
11

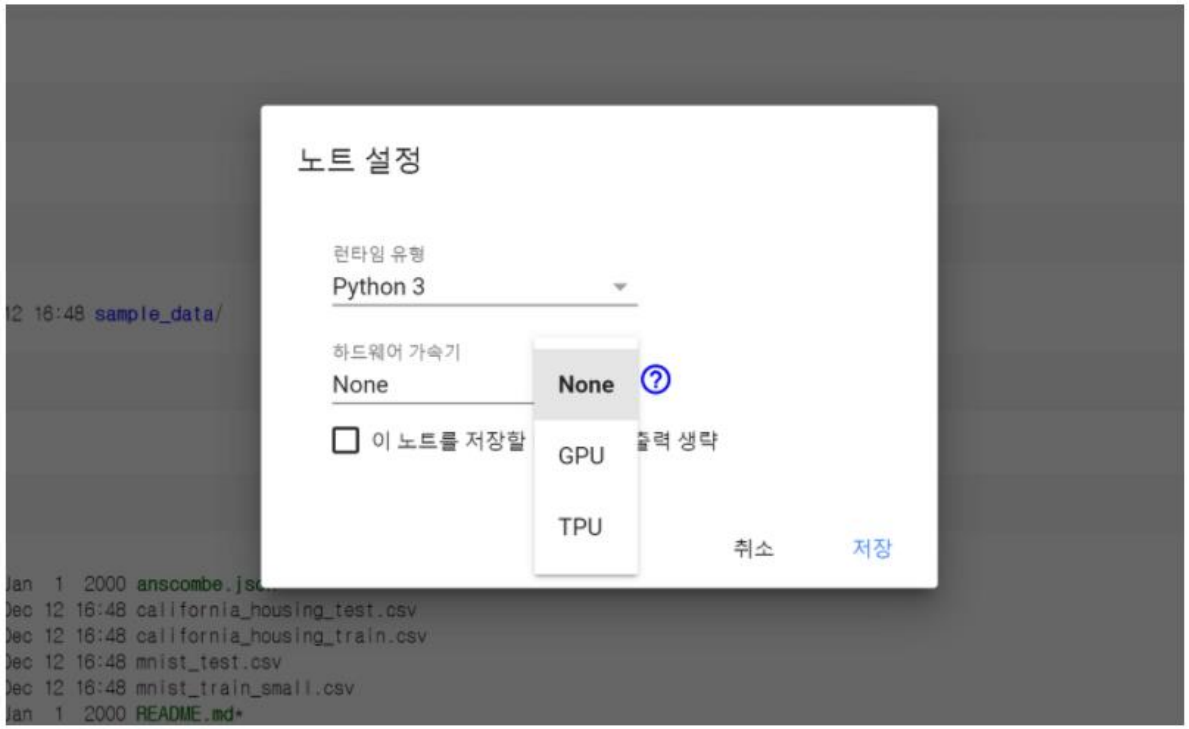
런타임 유형 GPU, TPU 설정하기

무료답지 않게 엄청난 서비스가 무료입니다.

GPU, TPU 등을 지원하기 때문에 요즘 핫한 Tensorflow등을 동작할 수 있습니다.

Colab의 GPU 정보 확인 방법:
!nvidia-smi





실습:
VDS(차량검지기) 데이터 가시화(I)

VDS 데이터 가시화

KISTI SINCE 1962

In [1]:
```
import pandas as pd
```

In [2]:
```
df = pd.read_csv('./vds.csv')
```

In [3]:
```
df.head()
```

Out[3]:

	Date	ToVol	SmVol	MeVol	LaVol	Speed	Occ.Rate
0	2017-04-02 0:00	43	34	9	0	50.3	1.90
1	2017-04-02 0:05	45	32	13	0	58.9	1.84
2	2017-04-02 0:10	46	34	12	0	50.6	1.87
3	2017-04-02 0:15	45	36	9	0	50.9	1.72
4	2017-04-02 0:20	27	13	13	1	62.2	1.12

VDS 데이터 가시화

KISTI SINCE 1962

In [4]:
```
df.tail(3)
```

Out[4]:

	Date	ToVol	SmVol	MeVol	LaVol	Speed	Occ.Rate
8061	2017-04-29 23:45	32	28	4	0	50.6	1.36
8062	2017-04-29 23:50	31	21	10	0	59.3	1.40
8063	2017-04-29 23:55	39	33	6	0	52.5	1.74

In [5]:
```
df.describe()
```

Out[5]:

	ToVol	SmVol	MeVol	LaVol	Speed	Occ.Rate
count	8064.000000	8064.000000	8064.000000	8064.000000	8064.000000	8064.000000
mean	110.459945	79.353299	29.948537	1.158110	49.327431	6.166941
std	63.954451	46.802106	19.081136	1.530192	7.921856	6.739946
min	6.000000	2.000000	0.000000	0.000000	9.100000	0.230000
25%	50.000000	35.000000	13.000000	0.000000	44.900000	2.140000
50%	122.000000	87.000000	29.000000	1.000000	48.500000	5.550000
75%	155.000000	111.000000	44.000000	2.000000	54.200000	7.290000
max	338.000000	250.000000	145.000000	16.000000	87.800000	82.100000

VDS 데이터 가시화

KISTI SINCE 1962

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8064 entries, 0 to 8063
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Date      8064 non-null   object
 1   ToVol     8064 non-null   int64
 2   SmVol     8064 non-null   int64
 3   MeVol     8064 non-null   int64
 4   LaVol     8064 non-null   int64
 5   Speed     8064 non-null   float64
 6   Occ.Rate  8064 non-null   float64
dtypes: float64(2), int64(4), object(1)
memory usage: 441.1+ KB
```

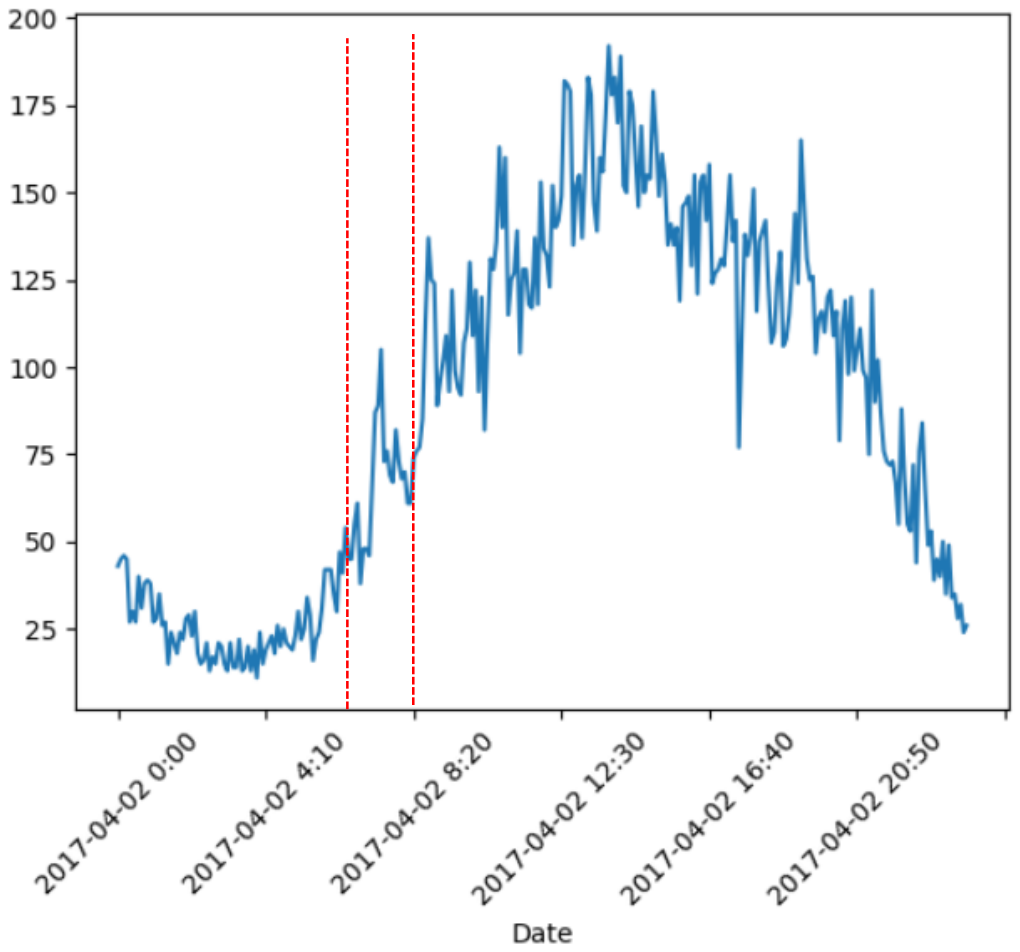
In [7]:
```
df.set_index('Date', inplace=True)
```

In [8]:
```
df.head()
```

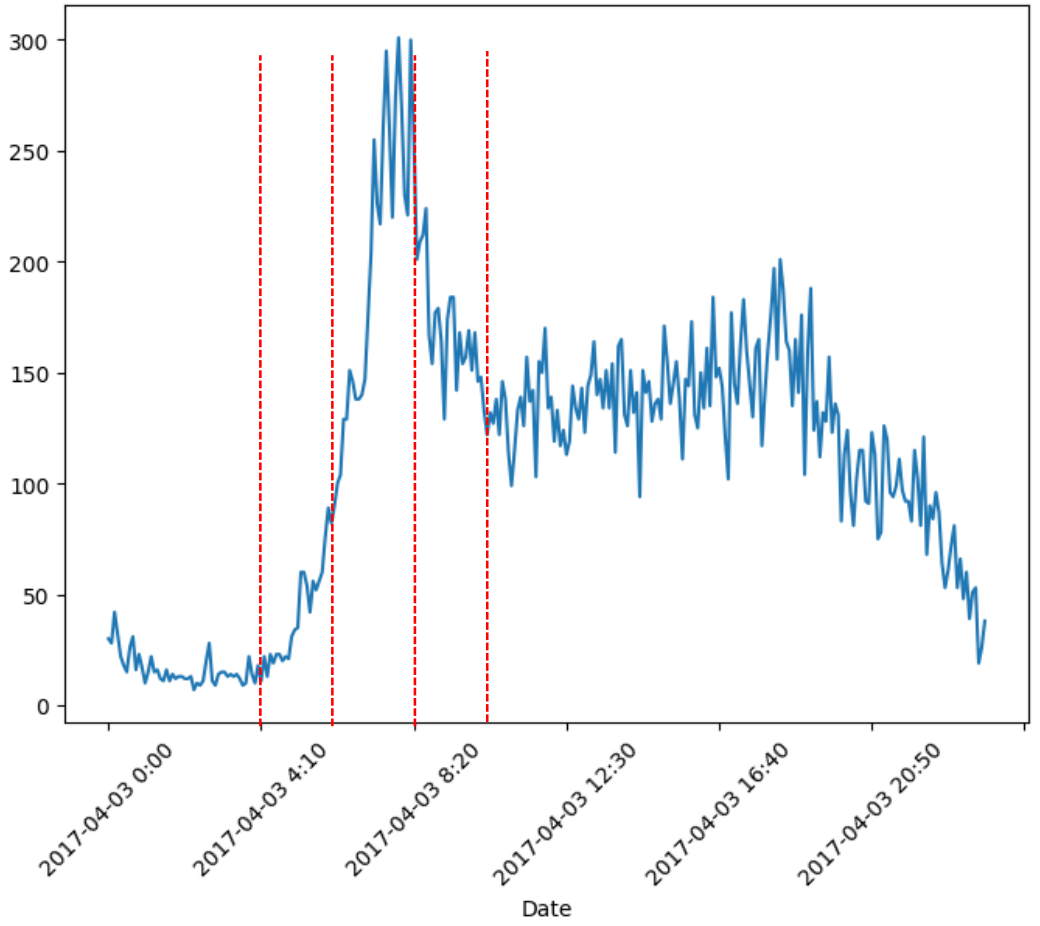
Date	ToVol	SmVol	MeVol	LaVol	Speed	Occ.Rate
2017-04-02 0:00	43	34	9	0	50.3	1.90
2017-04-02 0:05	45	32	13	0	58.9	1.84
2017-04-02 0:10	46	34	12	0	50.6	1.87
2017-04-02 0:15	45	36	9	0	50.9	1.72
2017-04-02 0:20	27	13	13	1	62.2	1.12

VDS 데이터 가시화

```
In [11]: df['ToVol'][:288].plot(rot=45)
```

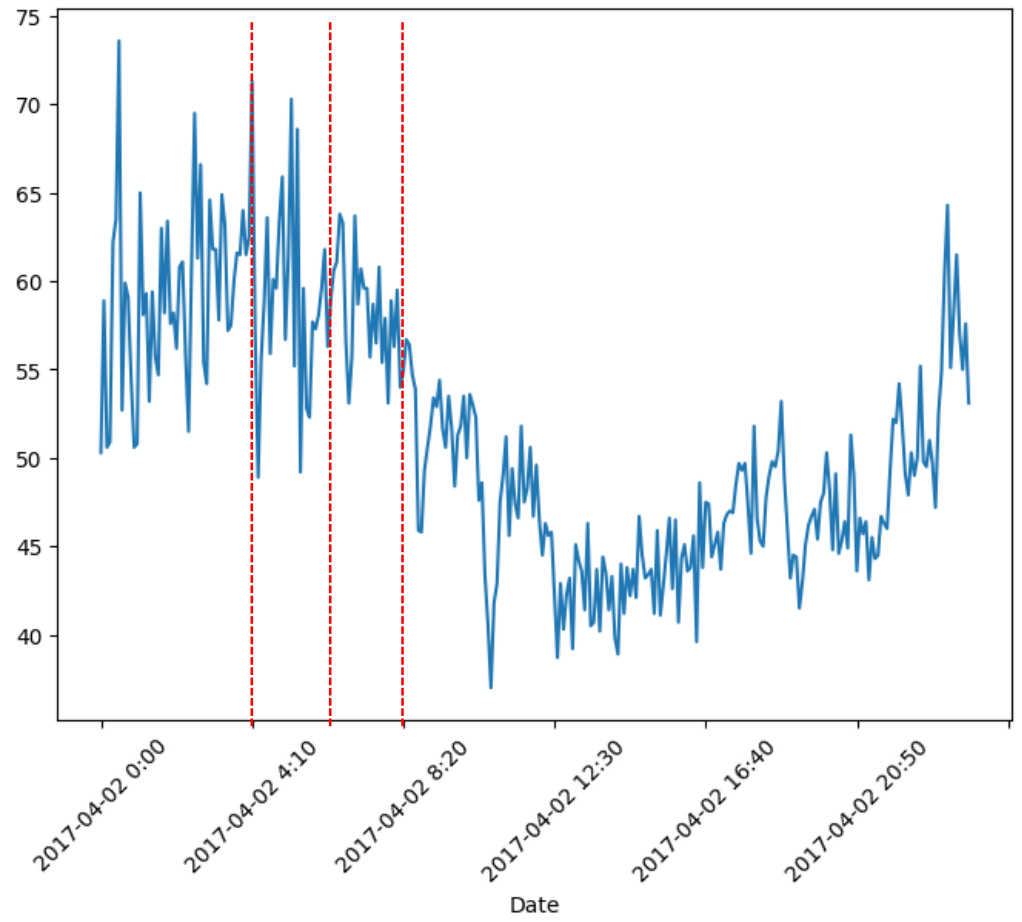



```
In [12]: df['ToVol'][288:576].plot(rot=45,figsize=(8,6))
```

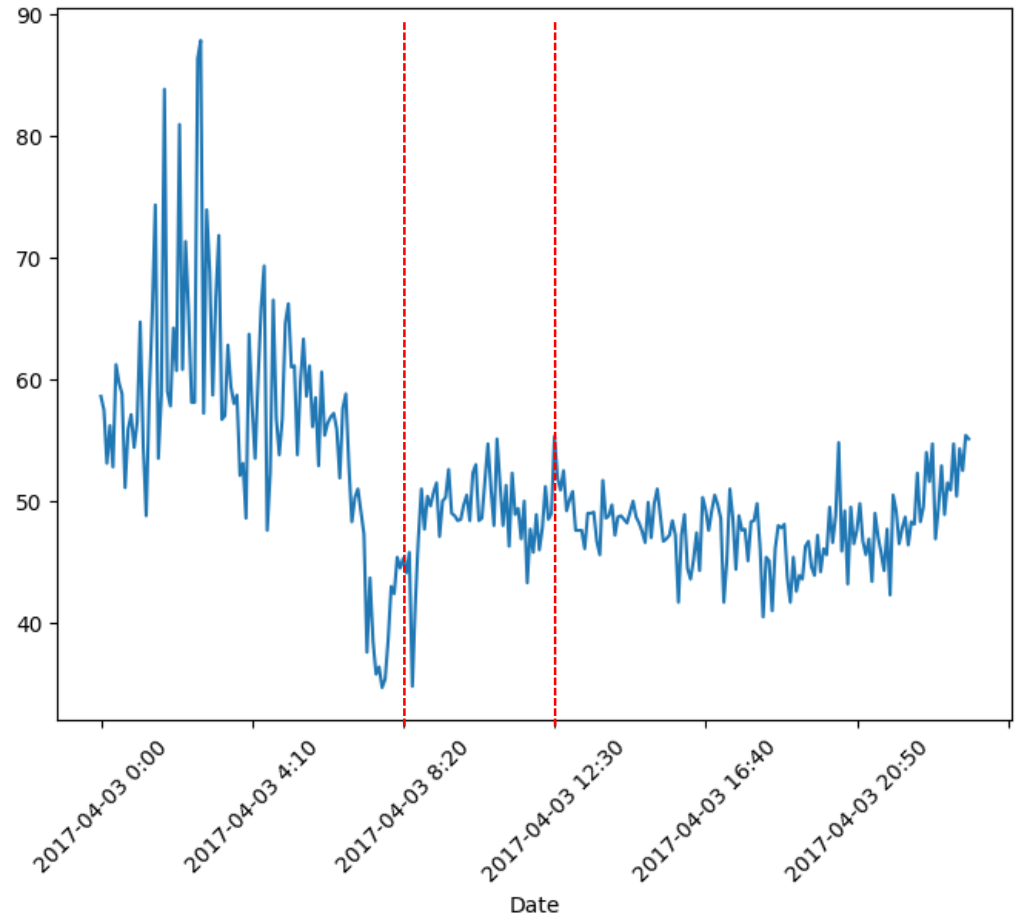


VDS 데이터 가시화

```
df['Speed'][:288].plot(rot=45,figsize=(8,6))
```

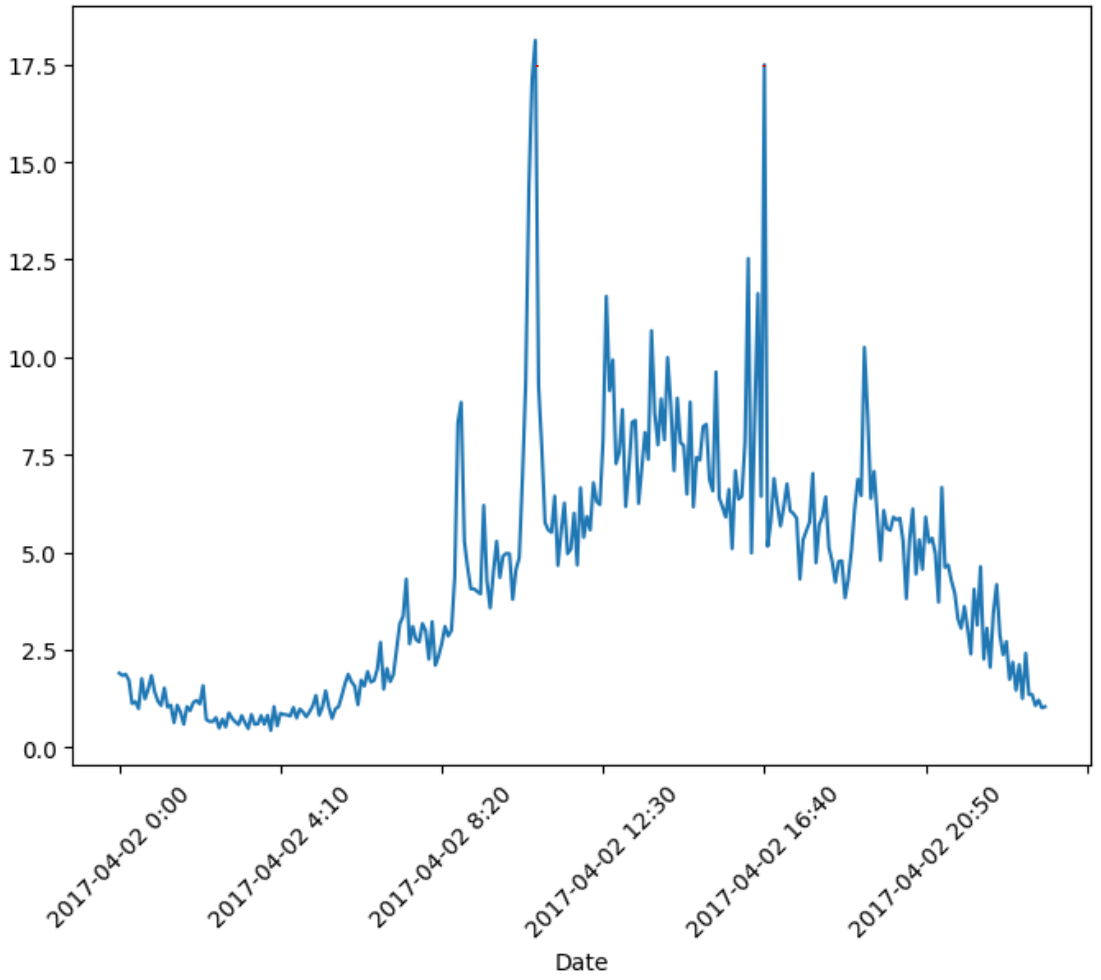



```
df['Occ.Rate'][288:576].plot(rot=45,figsize=(8,6))
```

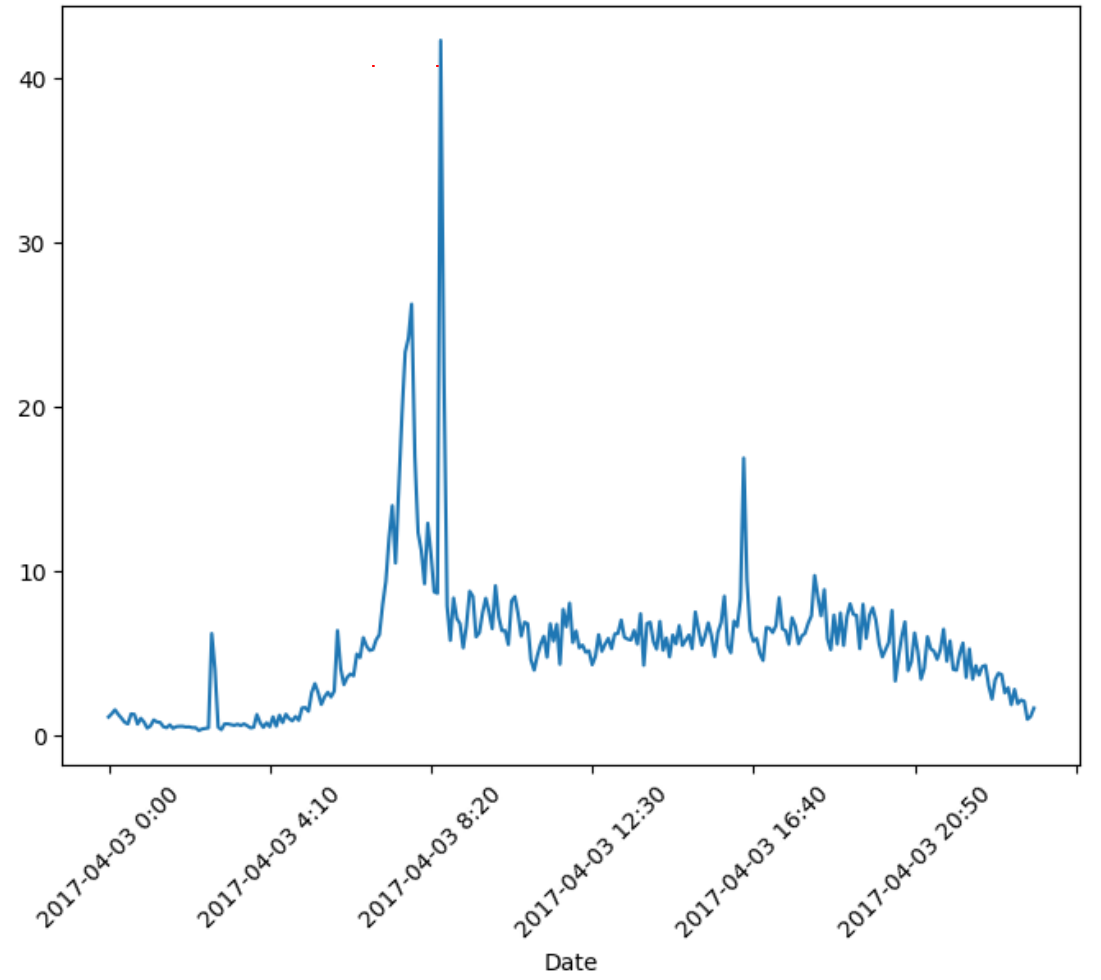


VDS 데이터 가시화

```
df['Occ.Rate'][:288].plot(rot=45,figsize=(8,6))
```




```
df['Occ.Rate'][288:576].plot(rot=45,figsize=(8,6))
```



ChatGPT 소개

ChatGPT: 개요

- ChatGPT는 Generative Pre-trained Transformer(GPT)와 Chat의 합성어이다.
- **ChatGPT**는 OpenAI가 개발한 프로토타입 대화형 인공지능 챗봇이다.
  - ✓ChatGPT는 대형 언어 모델 GPT-3의 개선판인 GPT-3.5를 기반으로 만들어졌으며,
  - ✓지도학습과 강화학습을 모두 사용해 파인 튜닝되었다.
    - ChatGPT는 2022년 11월 프로토타입으로 시작되었으며,
    - 다양한 지식 분야에서 상세한 응답과 정교한 답변으로 인해 집중 받았다.
  - ✓ 다만, 정보의 정확도는 중요한 결점으로 지적되고 있다.

https://ko.wikipedia.org/wiki/ChatGPT

ChatGPT: 학습

- ChatGPT는
  - ✓지도 학습과 강화 학습을 활용해 GPT-3.5를 기반으로 세밀하게 조정되었다.
    - 지도학습과 강화학습 모두 인간 트레이너들이 모델의 성능을 개선하기 위해 사용되었다.
  - ✓지도학습의 경우,
    - 인간 트레이너가 사용자와 ChatGPT 양쪽 모두를 연기하는 대화가 모델에 입력되었다.
  - ✓강화 단계에서는 인간 트레이너들이 먼저 모델이 이전 대화에서 만든 응답들에 순위를 매겼다.
    - 이 순위들은 PPO(Proximal Policy Optimization)를 이용하여 보상 모델을 만들기 위해 사용
    - 이 모델들은 마이크로소프트와 협업하여 마이크로소프트 애저 슈퍼컴퓨팅 인프라 상에서 훈련

https://ko.wikipedia.org/wiki/ChatGPT

다음 낱말 맞추기 연습

❖ ChatGPT-3 사전학습 (언어모델)

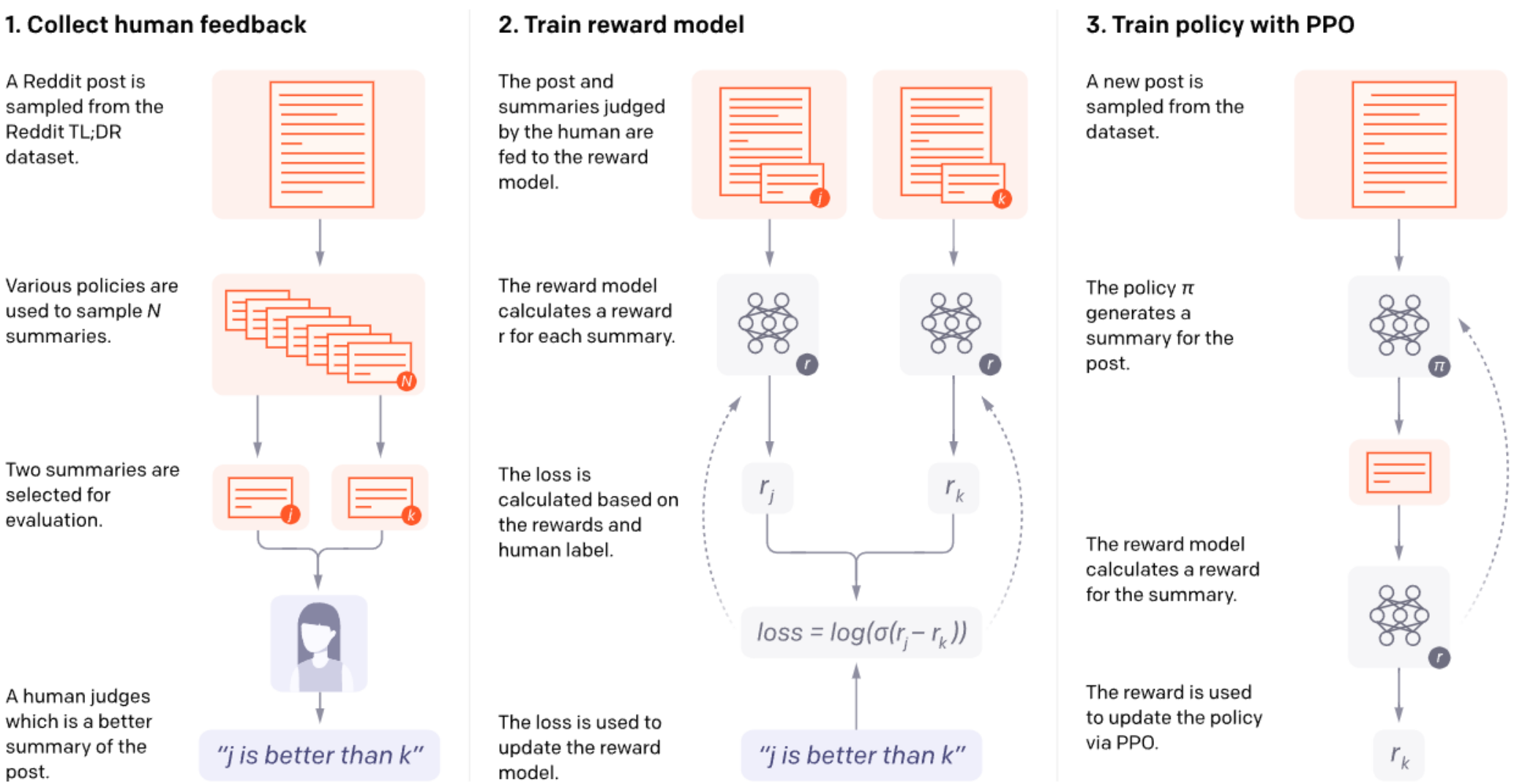
✓사전학습은 단지 단어끼리 관련성(연관성)이 높은 것

- GPT-3은 1억5천개 단어를 이용
- 위키피디아, 각종 책 등 45 테라바이트

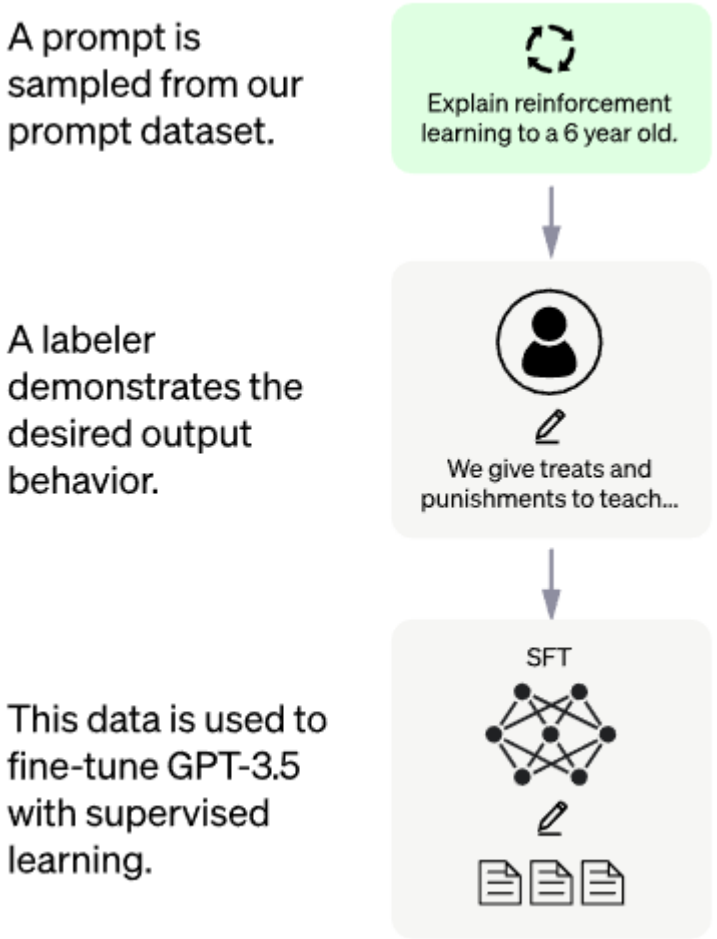
✓다음 낱말 맞추기에서 출발

- 예) 나는 밥을
- 다음 단어 맞추기는 검색어 자동 완성과 비슷

(source) https://www.donga.com/news/It/article/all/20230227/118085105/1

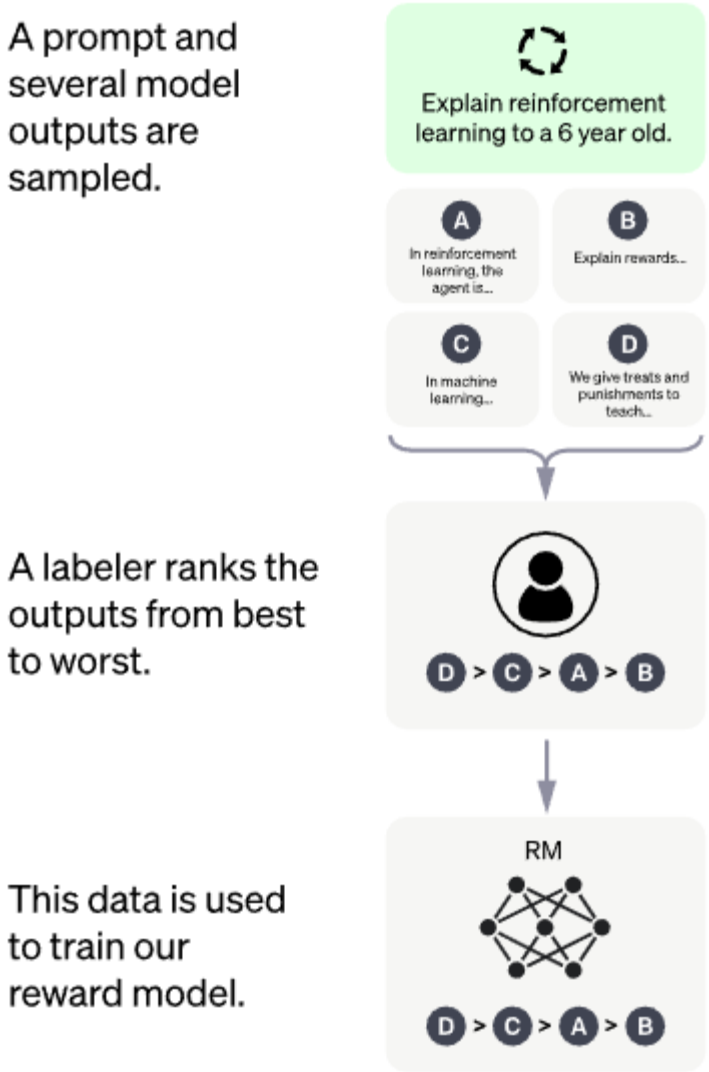

1. Collect human feedback
A Reddit post is sampled from the Reddit TL;DR dataset.
Various policies are used to sample N summaries.
Two summaries are selected for evaluation.
A human judges which is a better summary of the post.
"j is better than k"
2. Train reward model
The post and summaries judged by the human are fed to the reward model.
The reward model calculates a reward r for each summary.
The loss is calculated based on the rewards and human label.
The loss is used to update the reward model.
loss = log(σ(r_j − r_k))
"j is better than k"
3. Train policy with PPO
A new post is sampled from the dataset.
The policy π generates a summary for the post.
The reward model calculates a reward for the summary.
The reward is used to update the policy via PPO.

‘질문-답 ' 대본을 학습



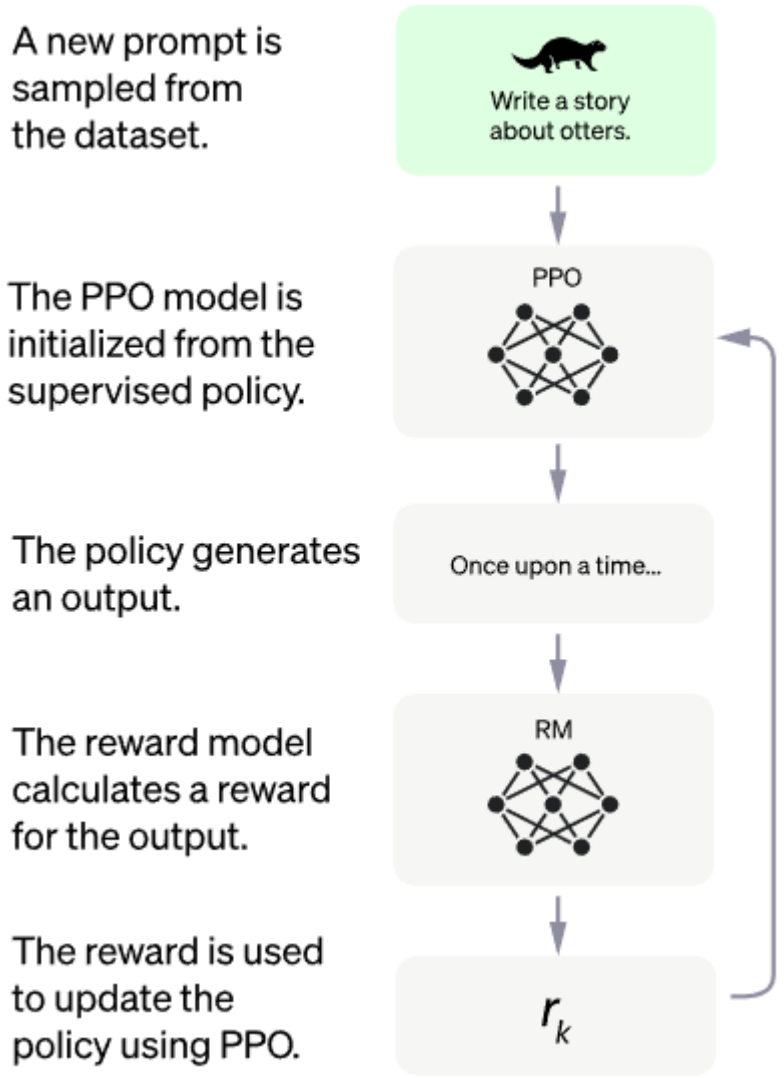
- 대화형 데이터를 수집하고 지도 정책을 훈련
  - ✓프럼프트(질문-답) 작성하자
    - 오픈AI는 40명의 전문 인력 고용
    - 민감한 이슈를 적절히 판별하고 답하는 능력자
    - 진실되고 유용하며 성실한 답변 능력자
    - 1만3천개 텍스트 (질문-답) 작성함
  - ✓레이블러(labeler)
    - 레이블러는 원하는 출력 형태를 보여준다
  - ✓STF 모델 훈련
    - 미세조정 지도학습

2단계 : 보상모델로 텍트스 등급



- ❖ 비교 데이터를 수집하고 보상 모델을 훈련
  - ✓프럼프트 출력이 샘플됨
    - 같은 프롬프트 실행하면 4개의 다른 답 출력
  - ✓다시 사람이 등급을 매김
    - 수 백만 사람들이 8세트 가겨으로 라벨링
- ❖ 챗GTP 모델의 지능은 어디서 오는 것일까?
  - ✓PPO 강화학습 모델

3단계 : InstructGPT



- 챗GTP 모델의 지능은 어디서 오는 것일까?
  - ✓강화학습을 사용하여 보상모델에 대한 정책 최적화
- 3단계는
  - ✓1단계 SFT 모델과 2단계 랭킹 모델을 사용하여 강화 학습을 통해서 Fine-Tuning
  - ✓PPO(Proximal Policy Optimization)
- 강화학습 과정
  - ✓InstructGPT는 새로운 프롬프트를 보고 결과를 추론
  - ✓이 결과를 보상모델이 평가하고 reward를 계산
  - ✓보상이 InstructGPT에 주어지고, 정책을 최적화하여 사람이 원하는 결과에 가장 가까운 것을 출력함

KISTI SINCE 1962

2023 TRUST KISTI

Korea Institute of Science and Technology Information