



# MACHINE 기계 학습 LEARNING

오일석 지음

# PREVIEW

## ■ 사람의 학습

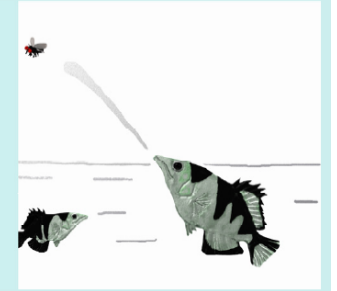
- 수학, 과학, 역사뿐 아니라 수영, 자전거 타기 등

## ■ 동물의 학습

- 예) 물총물고기의 목표물 맞추기 능력 향상



(a) 자전거 타기 학습



(b) 물총물고기의 사냥

그림 1-1 사람과 동물의 학습

## ■ 기계 학습

- 그렇다면 기계도 학습할 수 있을까?
- 경험을 통해 점점 성능이 좋아지는 기계를 만들 수 있을까?

## 1.1.1 기계 학습의 정의

- 학습이란? <표준국어대사전>

“경험의 결과로 나타나는, 비교적 지속적인 행동의 변화나 그 잠재력의 변화. 또는 지식을 습득하는 과정[국립국어원2017]”

- 기계 학습이란?

- 인공지능 초창기 사무엘의 정의

“Programming computers to learn from experience should eventually eliminate the need for much of this detailed programming effort. 컴퓨터가 경험을 통해 학습할 수 있도록 프로그래밍할 수 있다면, 세세하게 프로그래밍해야 하는 번거로움에서 벗어날 수 있다[Samuel1959].”

# 1.1.1 기계 학습의 정의

## ■ 기계 학습이란?

### ■ 현대적 정의

“A computer program is said to learn from experience  $E$  with respect to some class of tasks  $T$  and performance measure  $P$ , if its performance at tasks in  $T$ , as measured by  $P$ , improves with experience  $E$ .  
어떤 컴퓨터 프로그램이  $T$ 라는 작업을 수행한다. 이 프로그램의 성능을  $P$ 라는 척도로 평가했을 때 경험  $E$ 를 통해 성능이 개선된다면 이 프로그램은 학습을 한다고 말할 수 있다[Mitchell1997(2쪽)].”

“Programming computers to optimize a performance criterion using example data or past experience  
사례 데이터, 즉 과거 경험을 이용하여 성능 기준을 최적화하도록 프로그래밍하는 작업[Alpaydin2010]”

“Computational methods using experience to improve performance or to make accurate predictions  
성능을 개선하거나 정확하게 예측하기 위해 경험을 이용하는 계산학 방법들[Mohri2012]”

## 1.1.2 지식기반 방식에서 기계 학습으로의 대전환

### ■ 인공지능의 탄생

- 컴퓨터의 뛰어난 능력
  - 사람이 어려워하는 일을 아주 쉽게 함
  - $80932.46789076 \times 0.39001324$ 와 같은 곱셈을 고속으로 수행(현재는 초당 수십억개)
  - 복잡한 함수의 미분과 적분 척척
- 컴퓨터에 대한 기대감 (컴퓨터의 능력 과신)
  - 사람이 쉽게 하는 일, 예를 들어 고양이/개 구별하는 일도 잘 하지 않을까
  - 1950년대에 인공지능이라는 분야 등장

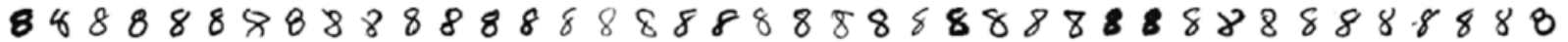
### ■ 초창기는 지식기반 방식이 주류

- 예) “구멍이 2개이고 중간 부분이 홀쭉하며, 맨 위와 아래가 둥근 모양이라면 8이다”

### 1.1.2 지식기반 방식에서 기계 학습으로의 대전환

## ■ 큰 깨달음

- 지식기반의 한계
- 단추를 “가운데 구멍이 몇 개 있는 물체”라고 규정하면 많은 오류 발생



**그림 1-2** 인식 시스템이 대처해야 하는 심한 변화 양상(8과 단추라는 패턴을 어떻게 기술할 것인가?)

- 사람은 변화가 심한 장면을 아주 쉽게 인식하지만, 왜 그렇게 인식하는지 서술하지는 못함

## 1.1.2 지식기반 방식에서 기계 학습으로의 대전환

- 인공지능의 주도권 전환
  - 지식기반 → 기계 학습
  - 기계 학습: 데이터 중심 접근방식



그림 1-3 기계 학습으로 만든 최첨단 인공지능 제품들



## 1.1.2 인공지능과 기계 학습의 간략한 역사

- 1843 에이더 “... 해석엔진은 꽤 복잡한 곡을 작곡할 수도 있다.”라는 논문 발표[Ada1843]
- 1950 인공지능 여부를 판별하는 튜링 테스트[Turing1950]
- 1956 최초의 인공지능 학술대회인 다트머스 콘퍼런스 개최. ‘인공지능’ 용어 탄생[McCarthy1955]
- 1958 로젠블랫이 퍼셉트론 제안[Rosenblatt1958]  
인공지능 언어 Lisp 탄생
- 1959 사무엘이 기계 학습을 이용한 체커 게임 프로그램 개발[Samuel1959]
- 1969 민스키가 퍼셉트론의 과대포장 지적. 신경망 내리막길 시작[Minsky1969] XOR
- 제1회 IJCAI(International Joint Conference on Artificial Intelligence) 개최
- 1972 인공지능 언어 Prolog 탄생
- 1973 Lighthill 보고서로 인해 인공지능 내리막길, 인공지능 겨울<sup>AI winter</sup> 시작
- 1974 웨어보스가 오류 역전파 알고리즘을 기계 학습에 도입[Werbos1974]
- 1975경 의료진단 전문가 시스템 Mycin – 인공지능에 대한 관심 부활
- 1979 『IEEE Transactions on Pattern Analysis and Machine Intelligence』 저널 발간
- 1980 제1회 ICML(International Conference on Machine Learning) 개최  
후쿠시마가 NeoCognitron 제안[Fukushima1980]
- 1986 『Machine Learning』 저널 발간  
『Parallel Distributed Processing』 출간  
다층 퍼셉트론으로 신경망 부활



## 1.1.2 인공지능과 기계 학습의 간략한 역사

- 1987      Lisp 머신의 시장 붕괴로 제2의 인공지능 겨울  
          UCI 리포지토리 서비스 시작  
          NIPSNeural Information Processing Systems 콘퍼런스 시작
- 1989      『Neural Computation』 저널 발간
- 1993      R 언어 탄생
- 1997      IBM 딥블루가 세계 체스 챔피언인 카스파로프 이김  
          LSTMLong short-term memory 개발됨
- 1998경    SVM이 MNIST 인식 성능에서 신경망 추월
- 1998      르쿤이 CNN의 실용적인 학습 알고리즘 제안[LeCun1998]  
          『Neural Networks: Tricks of the Trade』 출간
- 1999      NVIDIA 사에서 GPU 공개
- 2000      『Journal of Machine Learning Research』 저널 발간  
          OpenCV 최초 공개
- 2004      제1회 그랜드 챌린지(자율 주행)
- 2006      층별학습 탄생[Hinton2006a]
- 2007경    딥러닝이 MNIST 인식 성능에서 SVM 추월
- 2007      GPU 프로그래밍 라이브러리인 CUDA 공개

## 1.1.2 인공지능과 기계 학습의 간략한 역사

	어번 챌린지(도심 자율 주행)
	<u>Scikit-learn</u> 라이브러리 최초 공개
2009	Theano 서비스 시작
2010	<u>ImageNet</u> 탄생
	제1회 ILSVRC 대회
2011	IBM 왓슨이 제퍼디 우승자 꺾음
2012	MNIST에 대해 0.23% 오류율 달성
	<u>AlexNet</u> 발표 (3회 ILSVRC 우승)
2013	제1회 ICLR International Conference on Learning Representations 개최
2014	Caffe 서비스 시작
2015	<u>TensorFlow</u> 서비스 시작
	OpenAI 창립
2016	알파고와 이세돌의 바둑 대회에서 알파고 승리[Silver2016]
	『Deep Learning』 출간
2017	알파고 제로[Silver2017]

Home


Environments

Learning

Community


Applications on base (root) Channels

Refresh




**CMD.exe Prompt**  
0.1.1  
Run a cmd.exe terminal with your current environment from Navigator activated

Launch




**JupyterLab**  
2.1.5  
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Launch




**Notebook**  
6.0.3  
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Launch




**Powershell Prompt**  
0.0.1  
Run a Powershell terminal with your current environment from Navigator activated

Launch




**Qt Console**  
4.7.5  
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Launch




**Spyder**  
4.1.4  
Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Launch




**Glueviz**  
0.15.2  
Multidimensional data visualization across files. Explore relationships within and among related datasets.

Install



**Orange 3**  
3.26.0  
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

Install



**RStudio**  
1.1.456  
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

Install

**ANACONDA NAVIGATOR**

Thanks for installing Anaconda!

Anaconda Navigator helps you easily start important Python applications and manage the packages in your local Anaconda installation. It also connects you to online resources for learning and engaging with the Python, SciPy, and PyData community.

To help us improve Anaconda Navigator, fix bugs, and make it even easier for everyone to use Python, we gather anonymized usage information, just like most web browsers and mobile apps.

To opt out of this, please uncheck below (You can always change this setting in the Preferences menu).

☒ Yes, I'd like to help improve Anaconda.

Ok Ok, and don't show again

Documentation

Developer Blog



# ANACONDA NAVIGATOR

Home

Environments

Learning

Community

Documentation

Developer Blog



Applications on base (root) Channels

Refresh



CMD.exe Prompt

0.1.1

Run a cmd.exe terminal with your current environment from Navigator activated

Launch



JupyterLab

2.1.5

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Launch



Notebook

6.0.3

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Launch



Powershell Prompt

0.0.1

Run a Powershell terminal with your current environment from Navigator activated

Launch



Qt Console

4.7.5

PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Launch



Spyder

4.1.4

Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Launch



Glueviz

0.15.2

Multidimensional data visualization across files. Explore relationships within and among related datasets.

Install



Orange 3

3.26.0

Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

Install



RStudio

1.1.456

A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

Install

Spyder (Python 3.8)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\Wmgchun\spyder-py3\temp.py

```
1  -*- coding: utf-8 -*-
2  """
3  Spyder Editor
4  This is a temporary script file.
5  """
6
7
8
```

Usage

Here you can get help of any object by pressing Ctrl+H in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our [tutorial](#)

Variable explorer Help Plots Files

Console I/A

Python 3.8.3 (default, Jul 2 2020, 17:38:36) [MSC v.1916 64 bit (AMD64)]  
Type "copyright", "credits" or "license()" for more information.

IPython 7.16.1 -- An enhanced Interactive Python.

In [1]:

IPython console History

LSP Python: ready conda: base (Python 3.8.3) Line 1, Col 1 UTF-8 CRLF RW Mem 66%