

The background features large, flowing, organic shapes in two shades of teal and blue. A dark teal shape occupies the top left, while a lighter blue shape is at the bottom. A medium teal shape is on the right side. The overall effect is modern and minimalist.

인공지능 개요



인공지능의 과거, 현재, 미래

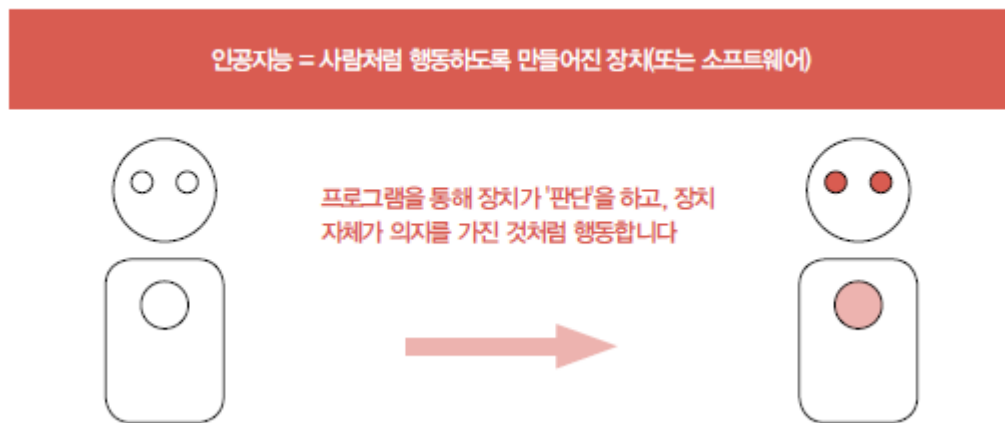
인공지능 연구의 과거와 현재를 살펴보고, 앞으로 어떻게 진화해 나갈지 살펴보기

SECTION 01 인공지능이란

1.1 개발자에게 맞는 ‘인공지능’의 정의

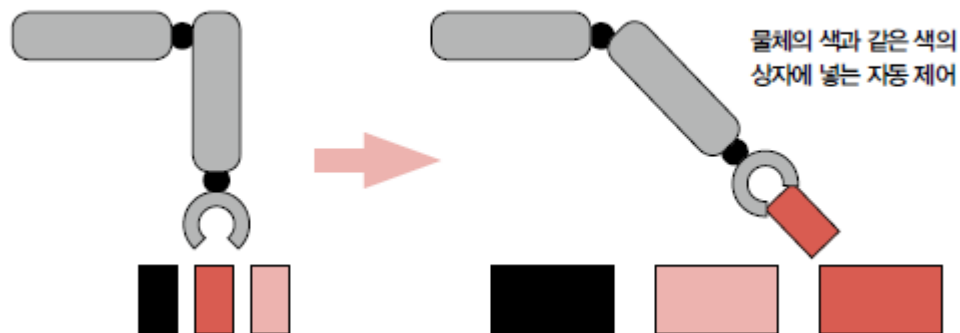
- 개발자 관점의 인공지능 = 사람처럼 행동하도록 만들어진 장치(또는 소프트웨어)

그림 1-1 개발자에게 맞는 인공지능의 정의



- 과거의 인공지능 붐에는 사람의 눈앞에 제공되는 것은 자동 제어의 결과물이었음.

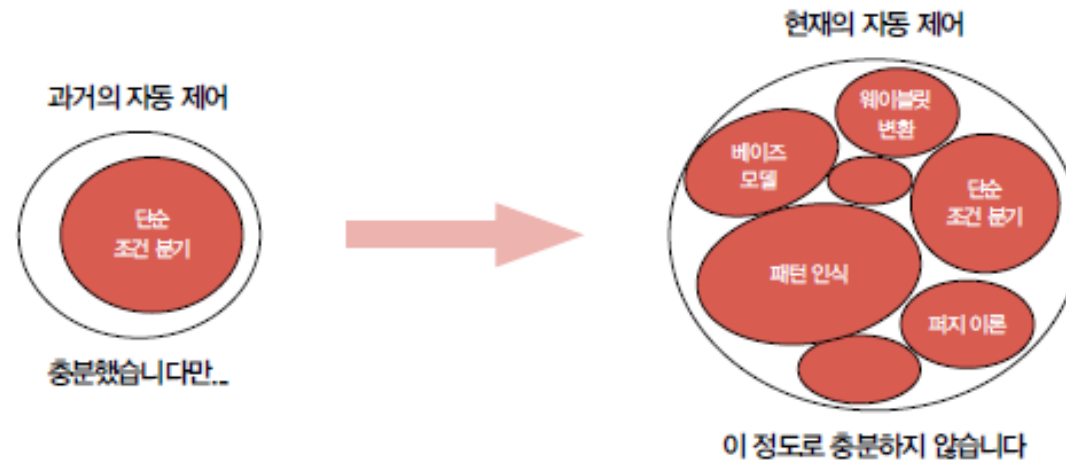
그림 1-2 자동 제어의 대표적인 예



SECTION 01 인공지능이란

- 오늘날에는 복잡한 이론을 자동 제어에 적용하더라도 ‘인공지능’이라고 하지 않음.

그림 1-3 인공지능과 자동 제어의 연관성



SECTION 02 인공지능의 여명기

2.1 '인공지능'의 탄생

- 인공지능이라는 용어는 1956년 열렸던 다트머스 회의에서 처음 언급되었음.
- 수학·컴퓨터 과학 이론을 바탕으로 한 연구에는 기계와 지능을 고찰한 앨런 튜링이 크게 공헌함.
- 생리학적 측면에서는 생리학, 기계 공학, 제어 공학을 융합해 다루는 분야인 사이버네틱스와 인공지능의 핵심 이론 중 하나인 '신경망Neural Network'을 연구함.
- 생리학 분야 연구로는 정보 전달 모델링에 관한 이론과 시냅스 가소성이 있음.

그림 1-4 시냅스 가소성

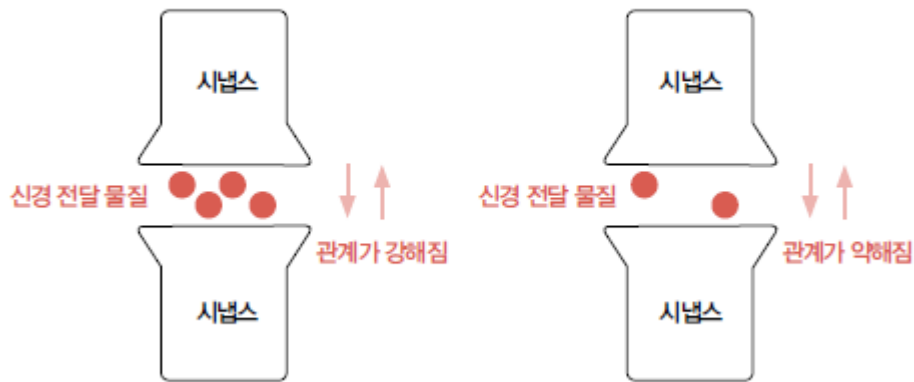
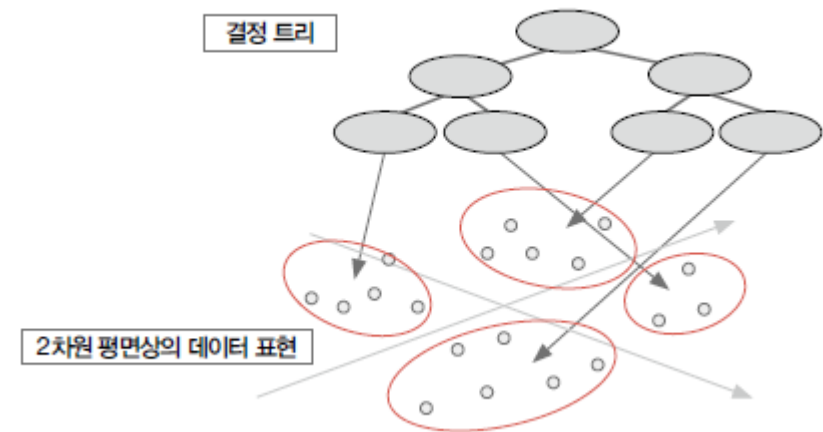


그림 1-5 결정 트리



SECTION 02 인공지능의 여명기

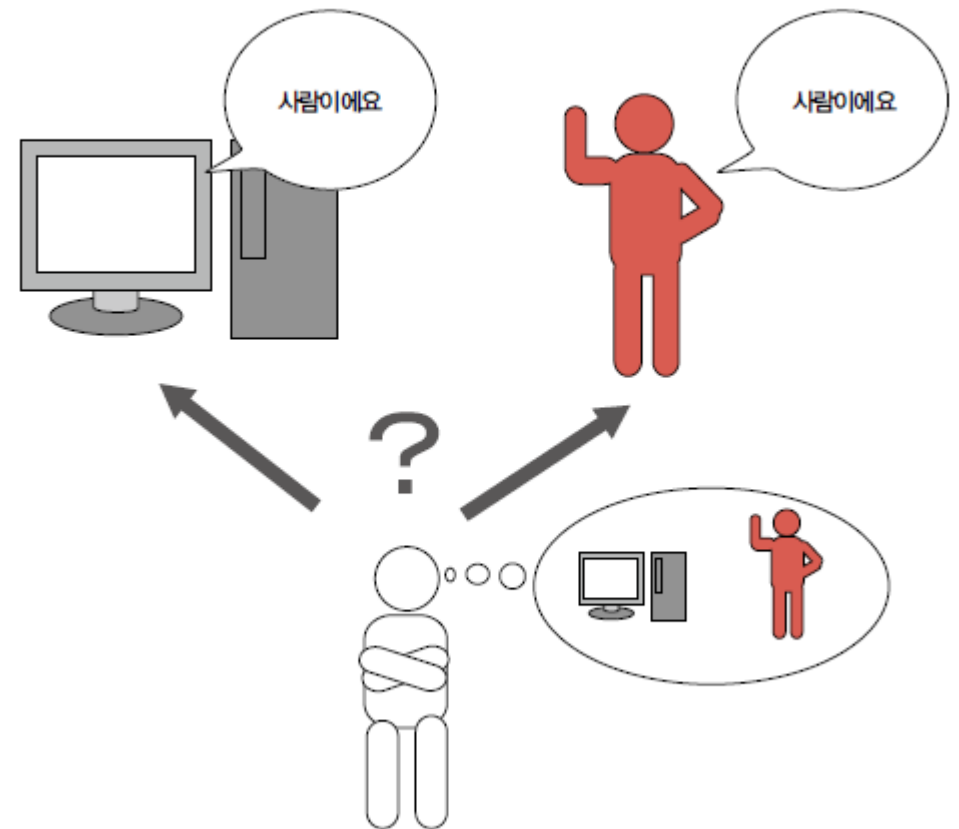
2.2 튜링 테스트와 인공지능

- 인공지능은 기계가 계산한 결과로 얻은 답이 '사람이 낸 답을 대신'할 수 있어야 함.
- 튜링 테스트 = 기계인지 사람(생각하는 존재)인지를 판별하는 시험

그림 1-6 사람과 자동 운항의 관계



그림 1-7 튜링 테스트



SECTION 03 인공지능의 발전 흐름

• 인공지능의 발전 흐름

그림 1-8 1960년 이후 인공지능의 발전 흐름



SECTION 03 인공지능의 발전 흐름

3.1 1960~1980년: 전문가 시스템과 1차 인공지능 붐

- 1950년대 이후 다수의 조건 분기를 사용하는 규칙 기반 자동 판정 프로그램이 발전하기 시작함.
- 새로운 사실을 탐색하는 추론 엔진과 일반인도 기계가 판단한 지식 탐구의 결과를 참고할 수 있도록 하는 전문가 시스템이 등장함.
- 전문가 시스템은 전문가가 실행하는 조건 판단을 프로그램화해 문제를 처리하는 시스템으로 유명한 전문가 시스템으로는 Dendral이 있음.

NOTE_ Dendral

Dendral은 1965년 스탠퍼드 대학교의 에드워드 파이겐바움(Edward Feigenbaum) 등이 개발하기 시작한 인공지능 프로젝트입니다. 아직 알려지지 않은 유기화합물에 질량 분석법을 적용해 화합물의 구조를 파악해서 분석합니다. 원래 화학자가 할 일을 자동화한 세계 최초의 전문가 시스템으로 알려져 있습니다.

SECTION 03 인공지능의 발전 흐름

3.2 1980~2000년: 2차 인공지능 붐과 신경망의 암흑기

- 1980년대에는 반도체 개발 비용이 낮아지면서 CPU, RAM, 캐시 메모리 용량이 늘어난 대규모 집적 회로를 만들 수 있게 되었음.
 - 국가 차원에서 “컴퓨터의 성능 향상을 고려하는 새로운 인공지능 연구”라는 관점으로 연구 이뤄짐.
 - 2차 인공지능 붐은 신경망 연구가 발전했던 시기임.
-
- 두 가지 문제로 인기가 시들해진 상태를 퍼셉트론의 다층화(다층 퍼셉트론)와 오차역전파법으로 두 가지 문제점을 해결함.
 - 1 명제 중 1개만 참일 경우를 판단하는 베타적 논리합을 다룰 수 없음.
 - 2 앞에서 설명한 사고범위 문제를 계산할 정도로 컴퓨터 연산 수준이 높지 않음.
 - 1980년대의 컴퓨터 연산 성능으로는 사고범위 문제를 해결하기 어렵다는 한계에 도달함.

SECTION 03 인공지능의 발전 흐름

3.3 2000~2010년: 통계 기반 머신러닝과 분산 처리 기술의 발전

- 컴퓨팅 연산 성능을 개선하는 분산 처리 기술도 발전함.
- 통계를 이용해 문제를 해결하는 방법은 크게 어떤 기준으로 데이터를 나누는 분류와 데이터로 앞으로 필요한 결과를 도출하는 예측으로 나눌 수 있음.
- 통계 기반 머신러닝 연구가 활발해진 계기는 1990년대 베이즈 정리를 출발점에 둔 베이즈 통계학의 재조명임.

그림 1-9 머신러닝을 이용하는 대표적인 작업: 분류와 예측

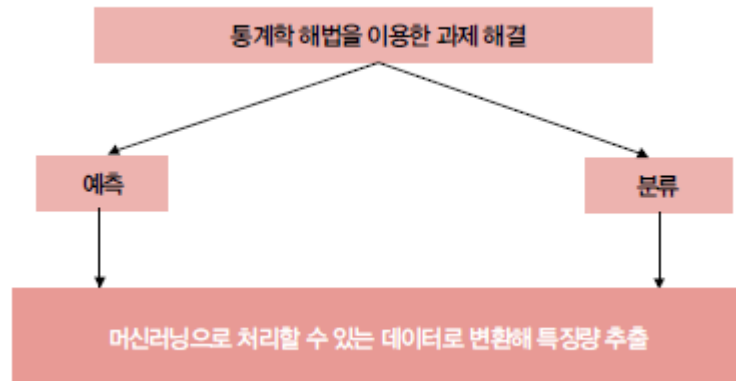
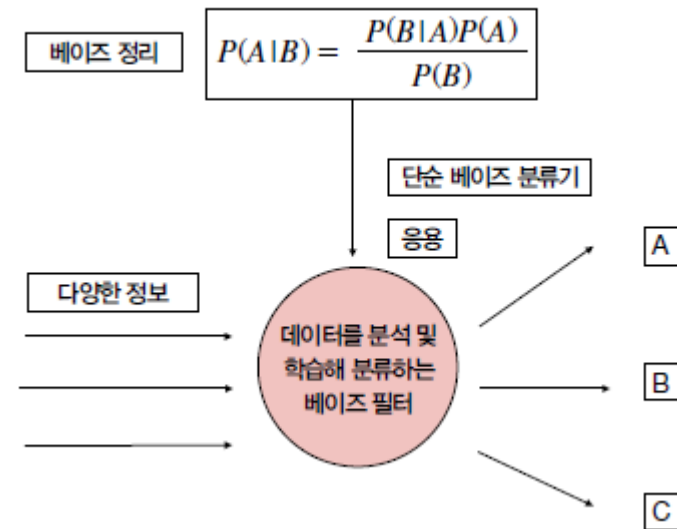


그림 1-10 베이즈 정리와 베이즈 필터



SECTION 03 인공지능의 발전 흐름

- 컴퓨팅 연산 성능이 향상된 이유는 1990년대 후반 고속 인터넷망 보급과 함께 대용량 이미지나 동영상 등이 만들어지기 시작했고, 이를 처리하거나 분석할 필요성이 생겼기 때문임.
- 대용량 이미지나 음성 데이터 분석은 컴퓨터 한 대의 컴퓨팅 연산 성능으로 처리하기가 힘들. 그 결과 하드웨어와 소프트웨어 모두를 고려하는 데이터 분산 처리 기술을 주목하기 시작함.

그림 1-11 색상 수에 따른 데이터 크기 비율

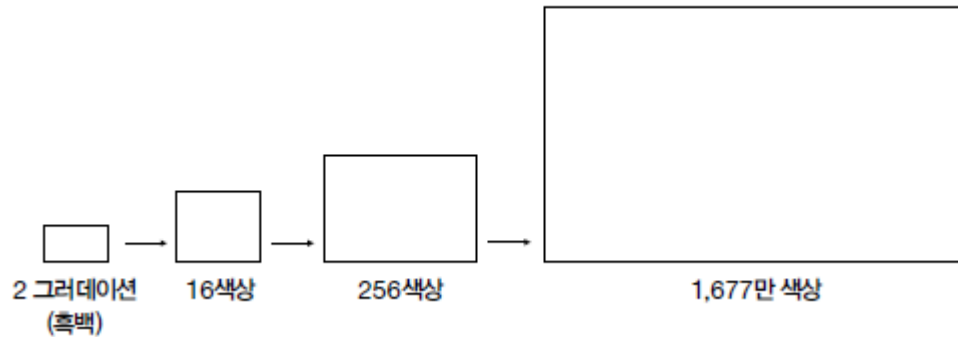
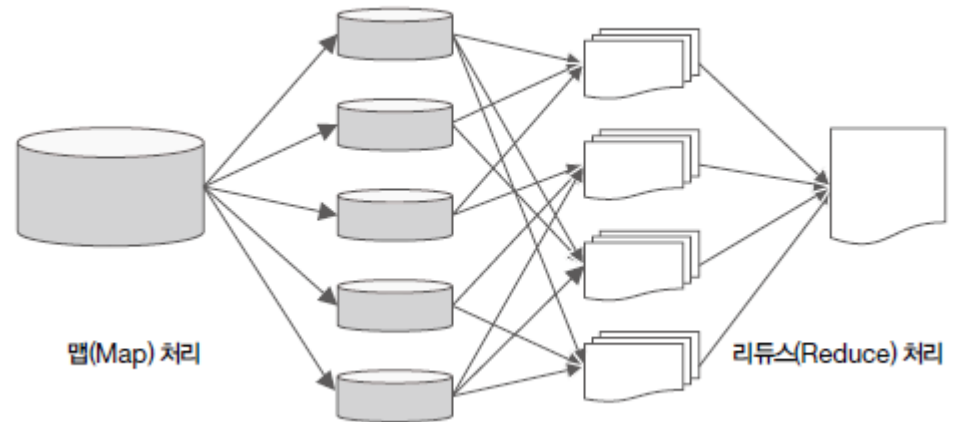


그림 1-12 맵리듀스 아키텍처



- 무어의 법칙에 따른 하드웨어 성능 향상과 분산 처리 기술이 결합하면서 2000년대 중반부터 다시 신경망 연구가 활발해짐. 이는 2006년 오토인코더AutoEncoder의 등장과 함께 딥러닝의 시대로 연결됨.

SECTION 03 인공지능의 발전 흐름

3.4 2010년 이후: 심층 신경망 기반 이미지 인식 성능 향상과 3차 인공지능 붐

- 2000년대 분산 처리 기술과 신경망 연구가 결합하면서 신경망 기반 머신러닝의 이미지 인식의 정확도가 다시 더 좋아지게 됨.
- 이 사실을 알린 상징적인 사건은 'ILSVRC 2012' 임.
- 2015년에는 일반적인 사람의 이미지 인식 오류율인 5% 아래의 오류율을 갖는 딥러닝 기반 이미지 인식 프로그램까지 등장함.

그림 1-13 ILSVRC 2012 - 이미지 인식 프로그램 성능 평가 순위¹⁸

IMAGENET Large Scale Visual Recognition Challenge 2012 (ILSVRC2012)			
Held in conjunction with PASCAL Visual Object Classes Challenge 2012 (VOC2012)			
Back to Main page			
Task 1			
Team name	Filename	Error (5 guesses)	Description
토론토 대학 팀	SuperVision	test-prede-111-146,2004-121-137-105-108,2011-100F	0.15115
	SuperVision	test-prede-131-137-145-135-145M.txt	0.10422
도쿄 대학 팀	DB	prod_PVX_MLACs_weighted.txt	0.26172
			Weighted sum of scores from each classifier with SIFT+PV, LBP+PV, GIST+PV and

그림 1-14 이미지 인식 엔진의 적용 사례



- 음성 인식과 자연어 처리에도 딥러닝을 활용하기 시작.
(대표적인 예로 챗봇프로그램과 구글과 네이버에서 제공하는 번역 서비스 등이 있음)

SECTION 03 인공지능의 발전 흐름

3.5 다양하게 활용하는 인공지능 연구

- 인공지능 연구는 다양한 분야에서 이용하는 추세임.

구분	설명
자동차 업계	• 이미지 인식 인공지능 연구를 자율 주행 기술과 결합하는 노력을 기울이고 있음
광고 업계	• 많은 웹 사이트에서 이용자에게 꼭 맞는 광고나 뉴스 기사를 추천(머신러닝 시스템이 계산한 예측 결과의 표시)하려고 머신러닝 시스템(추천 엔진)을 사용함.
비즈니스 인텔리전스 도구	• 1970년대부터 BI 도구를 탑재한 컴퓨터를 의사결정 지원 시스템으로 활용했으며, 다룰 수 있는 데이터양이 늘고 컴퓨터 처리 능력이 향상되면서 BI 도구 역시 경영 현장의 요구에 맞춰 정확한 예측을 할 수 있도록 진화해왔음.
챗봇	• 챗봇은 자연어 처리의 성능 향상을 추구함. • 자연스러운 대화나 번역을 할 수 있도록 기술을 발전시키는 것임. • 2000년대 후반 대량의 텍스트 데이터를 처리하고 특징을 추출하는 컴퓨터 자원의 확충과 추출된 특징을 표현하는 모델을 만들어 내면서 결실을 맺음.
의료 지원	• 의료 분야에서 인공지능을 이용하는 예로는 이미지 진단을 응용한 암 병변의 조기 발견, 손목 밴드형 계측 장치를 사용한 건강 관리 시스템 등이 있음. • 앞으로는 국가 차원에서 데이터를 이용한 맞춤 의학을 운용할 가능성도 있음.
로봇 산업	• 머신러닝을 포함한 인공지능 연구로 한계를 극복하려는 시도가 계속되었음. • 스스로 학습해서 자율적인 움직임 제어를 얻는 뉴로모픽 컴퓨터가 개발되어 왔음. • 앞으로는 강화 학습 알고리즘을 도입한 로봇 개발을 진행할 것으로 예상함.

3.6 인공지능의 미래 - '의식'을 지닌 인공지능

- 인공지능의 미래로 삼는 목표는 “사람 같은 ‘의식’을 지닌 인공지능을 만들자”임.
- 빅데이터와 디지털 클론
 - 빅데이터와 인공지능을 연결하는 한 가지 예로 사람의 사고 능력, 취미, 취향을 디지털 세계에 재현하려는 디지털 클론이 있음.
 - 디지털 클론 기술의 발전에는 이미지에서 표정을 추측, 감정에 대응할 수 있는 센싱 기술과 빅데이터 기술의 연결이 중요함.
 - 기술이 발전한 후에는 더 많은 센서에서 모이는 데이터와 인공지능 기술을 연결해 인격의 재현 등도 시도할 가능성이 큼.
- 기술적 특이점과 인공지능의 윤리적 관점
 - 작은 단위의 인공지능 프로그램을 서로 연계할 수 있게 되면 ‘의식’을 지닌 인공지능이 탄생, 문제를 해결할 수 있을지도 모름.
 - 그 과정에서 인공지능이 올바른 윤리 의식을 갖고 움직일지는 아무도 장담할 수 없음.
 - 올바른 윤리적 판단 아래 문제를 설정하고 해결하도록 제어하는 부분은 사람만의 영역으로 영원히 남을 수도 있음.



머신러닝/딥러닝 개요

머신러닝에 대해서

- 머신러닝은 '데이터에서 법칙성을 추출하는 통계적 방법'의 하나.
- 머신러닝에는 법칙을 추출하여 예측, 분류하는 다양한 모델(알고리즘).
- 머신러닝은 손글씨 문자 식별, 물체 식별, 질병 진단 등 다양한 분야에 걸쳐 응용.
- 딥러닝도 머신러닝의 일부로, 신경망 모델의 한 형태.



그림 1-1 머신러닝이란

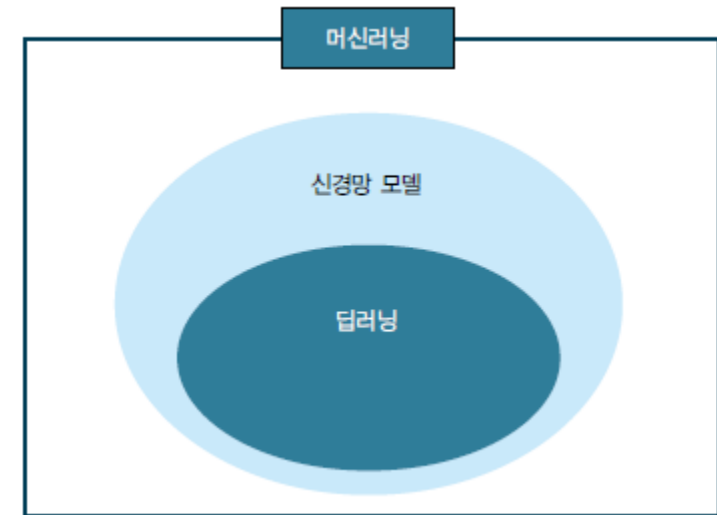


그림 1-2 머신러닝, 신경망 모델, 딥러닝의 관계

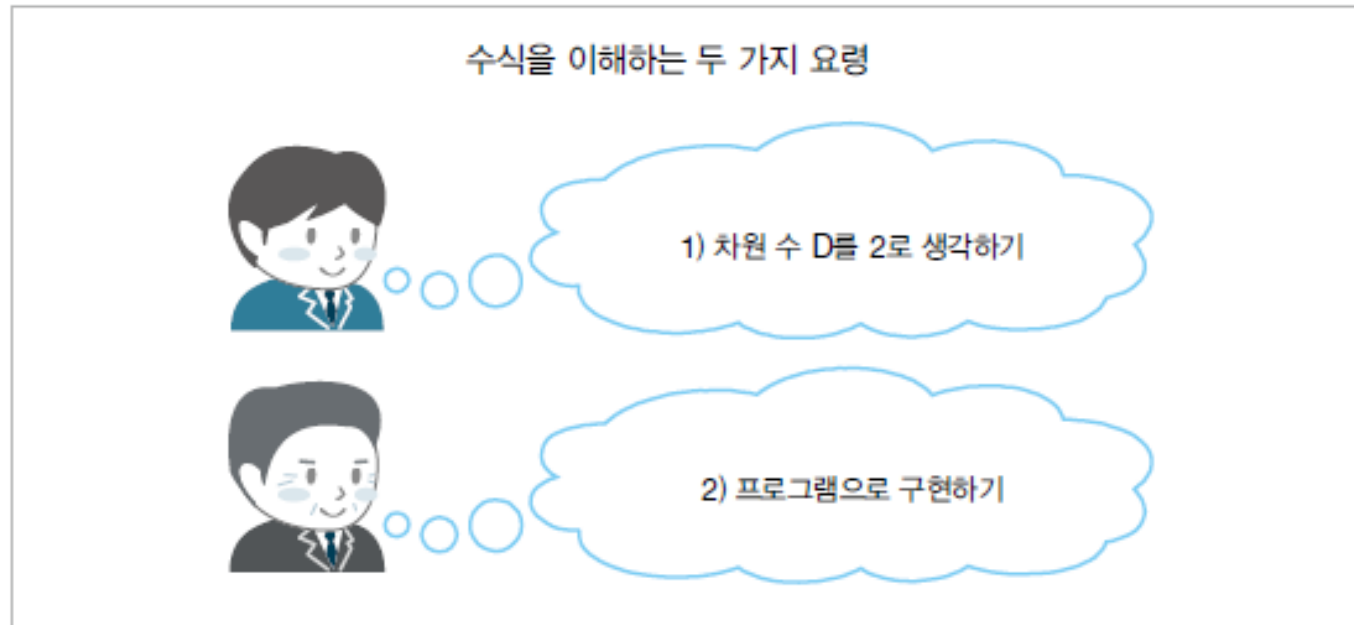
머신러닝에 대해서

1.1 머신러닝 습득의 방법

머신러닝을 쉽게 이해할 수 있는 두 가지 방법

- 어려운 수식을 간단하게 만드는 방법. 차원 D 를 2로 생각하는 것.
- 자신이 이해했는지 확인하는 방법. 프로그램으로 구현해보는 것.

그림 1-3 수식을 이해하는 방법

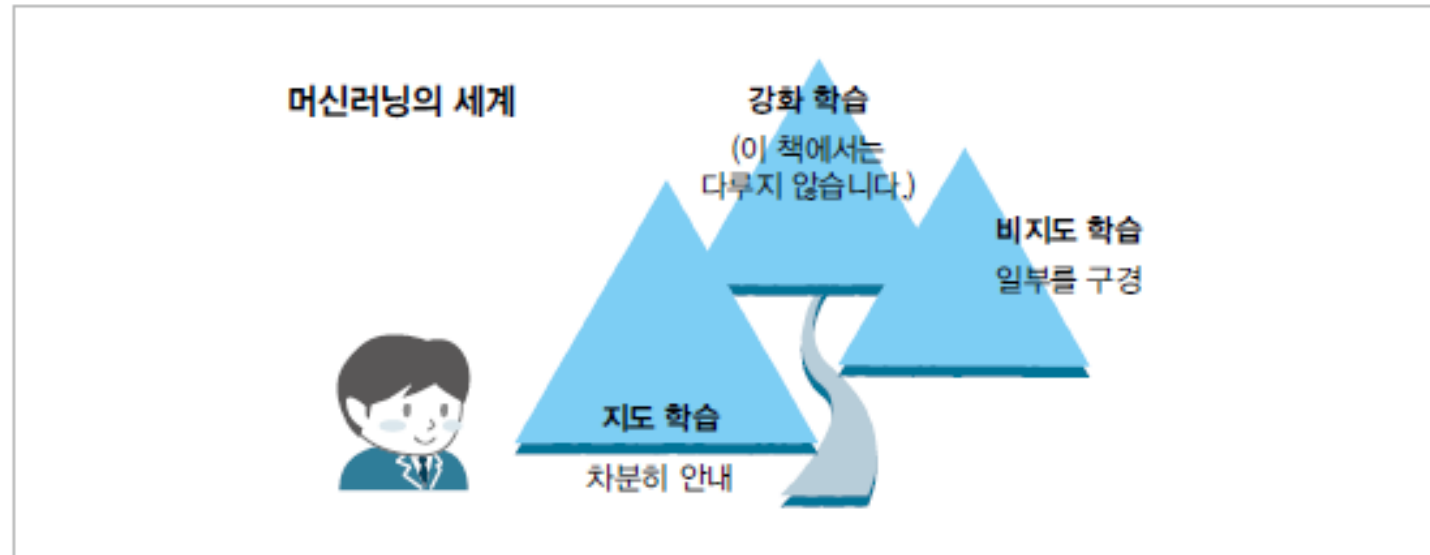


머신러닝에 대해서

1.2 머신러닝 문제의 분류

머신러닝의 문제는 크게 3가지로 분류

- 첫 번째는 지도 학습 Supervised Learning 문제
 - 입력에 대한 적절한 출력을 구하는 문제
- 두 번째는 비지도 학습 Unsupervised Learning 문제
 - 비지도 학습은 입력 정보의 특징을 찾는 문제
- 세 번째는 강화 학습 문제
 - 강화 학습은 장기나 체스와 같이 마지막 결과가 가장 좋은 행동을 찾는 문제.



왜 머신러닝인가?

◦ 결정 규칙을 직접 만들 때의 단점

- 결정에 필요한 로직은 한 분야나 작업에 국한됨. 따라서, 작업이 조금만 변경되더라도 전체 시스템을 다시 개발해야함.
- 규칙을 설계하려면 그 분야 전문가들이 내리는 결정 방식에 대해 잘 알아야 함.

◦ 머신러닝으로 풀 수 있는 문제

- 지도학습의 예
 - 편지 봉투에 손으로 쓴 우편번호 숫자 판별
 - 의료 영상 이미지에 기반한 종양 판단
 - 의심되는 신용카드 거래 감지
- 비지도 학습의 예
 - 블로그 글의 주제 구분
 - 고객들을 취향이 비슷한 그룹으로 묶기
 - 비정상적인 웹사이트 접근 탐지

◦ 문제와 데이터 이해하기

- 머신러닝 프로세스에서 가장 중요한 과정은 사용할 데이터를 이해하고 그 데이터가 해결해야할 문제와 어떤 관련이 있는지를 이해하는 것임

왜 파이썬인가?

- 파이썬(Python)은 데이터 과학 분야를 위한 표준 프로그래밍 언어
 - 파이썬은 범용 프로그래밍 언어의 장점은 물론 매트랩MATLAB과 R 같은 특정 분야를 위한 스크립팅 언어의 편리함을 함께 갖춘
 - 다양한 도구: 데이터 적재, 시각화, 통계, 자연어 처리, 이미지 처리 등에 필요한 라이브러리 존재
 - 터미널이나 주피터 노트북(Jupyter Notebook) 같은 도구로 대화하듯 프로그래밍할 수 있음
 - 머신러닝과 데이터 분석은 데이터 주도 분석이라는 점에서 근본적으로 반복 작업, 따라서 반복 작업을 빠르게 처리하고 손쉽게 조작할 수 있는 도구가 필수
 - 범용 프로그래밍 언어로서 파이썬은 복잡한 그래픽 사용자 인터페이스(GUI)나 웹 서비스도 만들 수 있으며 기존 시스템과 통합하기도 좋음

- 오픈 소스인 사이킷런(scikit-learn)은 자유롭게 사용하거나 배포 가능
 - 잘 알려진 머신러닝 알고리즘들은 물론 알고리즘을 설명한 풍부한 문서도 제공
 - <http://scikit-learn.org/stable/documentation>
 - 사이킷런은 매우 인기가 높고 독보적인 파이썬 머신러닝 라이브러리임
 - 산업 현장이나 학계에도 널리 사용되고 많은 튜토리얼과 예제 코드를 온라인에서 쉽게 찾을 수 있음
 - 사이킷런은 다른 파이썬의 과학 패키지들과도 잘 연동됨
- 사이킷런 설치
 - scikit-learn은 두 개의 다른 파이썬 패키지인 넘파이(NumPy)와 사이파이(SciPy)를 사용
 - 그래프를 그리려면 맷플롯립(matplotlib)을, 대화식으로 개발하려면 아이파이썬(Ipython)과 주피터 노트 북도 설치해야 함
 - 필요한 패키지들을 모아 놓은 파이썬 배포판을 설치하는 방법을 권장
 - Anaconda: 대용량 데이터 처리, 예측 분석, 과학 계산을 위한 파이썬 배포판
 - Enthought Canopy: 과학 계산을 위한 파이썬 배포판
 - Python(x,y): 윈도우 환경을 위한 과학 계산을 위한 무료 파이썬 배포판

필수 라이브러리와 도구들

- 주피터 노트북

- 주피터 노트북은 프로그램 코드를 브라우저에서 실행해주는 대화식 환경을 제공

- NumPy

- 파이썬으로 과학 계산을 하려면 꼭 필요한 패키지임. 다차원 배열을 위한 기능과 선형 대수 연산과 푸리에 변환 같은 고수준 수학 함수와 유사(pseudo) 난수 생성기를 포함

- SciPy

- 과학 계산을 위한 함수를 모아놓은 파이썬 패키지임. SciPy는 고성능 선형 대수, 함수 최적화, 신호 처리, 특수한 수학 함수와 통계 분포 등을 포함한 많은 기능을 제공

- matplotlib

- 파이썬의 대표적인 과학 계산을 위한 그래프 라이브러리임. 선 그래프, 히스토그램, 산점도 등을 지원하며 출판에 쓸 수 있을 만큼의 고품질 그래프를 그려줌

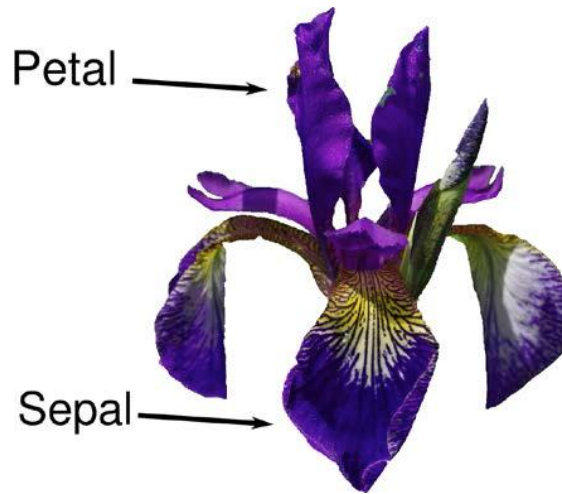
- pandas

- 데이터 처리와 분석을 위한 파이썬 라이브러리임

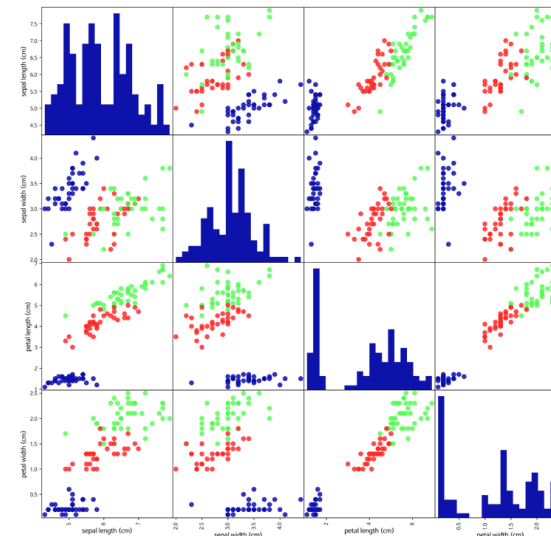
- mglearn

첫 번째 애플리케이션: 붓꽃의 품종 분류

- 어떤 품종인지 구분 해놓은 측정 데이터를 이용해 새로 채집한 붓꽃의 품종을 예측하는 머신러닝 모델을 만들어 보기
 - 데이터 적재
 - 성과 측정: 훈련 데이터와 테스트 데이터
 - 가장 먼저 할 일: 데이터 살펴보기
 - 첫 번째 머신러닝 모델: k-최근접 이웃 알고리즘
 - 예측하기
 - 모델 평가하기



▲그림 1-2 붓꽃의 부위



▲그림 1-3 클래스 레이블을 색으로 구분한 Iris 데이터셋의 산점도 행렬

딥러닝 개요

• 딥러닝의 발전 동향

- 컴퓨터를 인간처럼 학습시켜 스스로 규칙을 형성할 수 있도록 하는 알고리즘과 기술을 개발하는 기계학습은 인공지능의 한 분야.
예) 수신한 e-메일의 스팸 여부, 입력된 고양이와 개의 사진 구분과 같은 분류 문제, 주식 가격 예측, 기온 예측, 강수량 예측과 같은 회귀분석문제 등을 포함한 패턴인식 문제 등.
- 현재 검색엔진, 광고, 마케팅, 로봇, 인사 활동, 게임, 드론, 자율주행 자동차 등에 기계학습 방법론 활용. 기계학습 대표적인 방법론: 신경망, 의사결정나무, 베이지안 망, 서포트 벡터 기계, 강화학습 등

• 신경망 연구의 역사

- 1943년 매컬로크와 피츠가 시작.
- 1957년 로젠블라트는 전방향 연결 구조, 분류 기능의 퍼셉트론이라는 신경망을 개발.
- 1959년 위드로와 호프는 간단한 뉴런에 근거하여 adaptive linear elementadaline 개발 adaline을 여러 개 합성하여 Madaline이라는 신경망을 구성.
- 1960년대 중반 민스키와 페퍼트의 퍼셉트론에 대한 기능적 비판 이후 신경망에 대한 관심 줄어듦.
- 1986년 역전파 알고리즘이 소개되면서 신경망 관련 연구는 절정기.
- 1990년대 중반 서포트 벡터 기계가 등장으로 기계학습의 대세를 이루었고 신경망은 암흑기를 겪음.

딥러닝 역사

- 토론토대학교의 힌튼 교수가 신경망의 단점을 예비학습으로 해결할 수 있음을 발표. 딥러닝은 압도적인 성능으로 각종 기계학습 관련 대회 우승을 휩쓸었으며, 현재는 다른 기계학습방법을 통해 영상처리, 음성인식 등을 연구한 연구자들이 딥러닝으로 대동단결하는 양상을 보임.

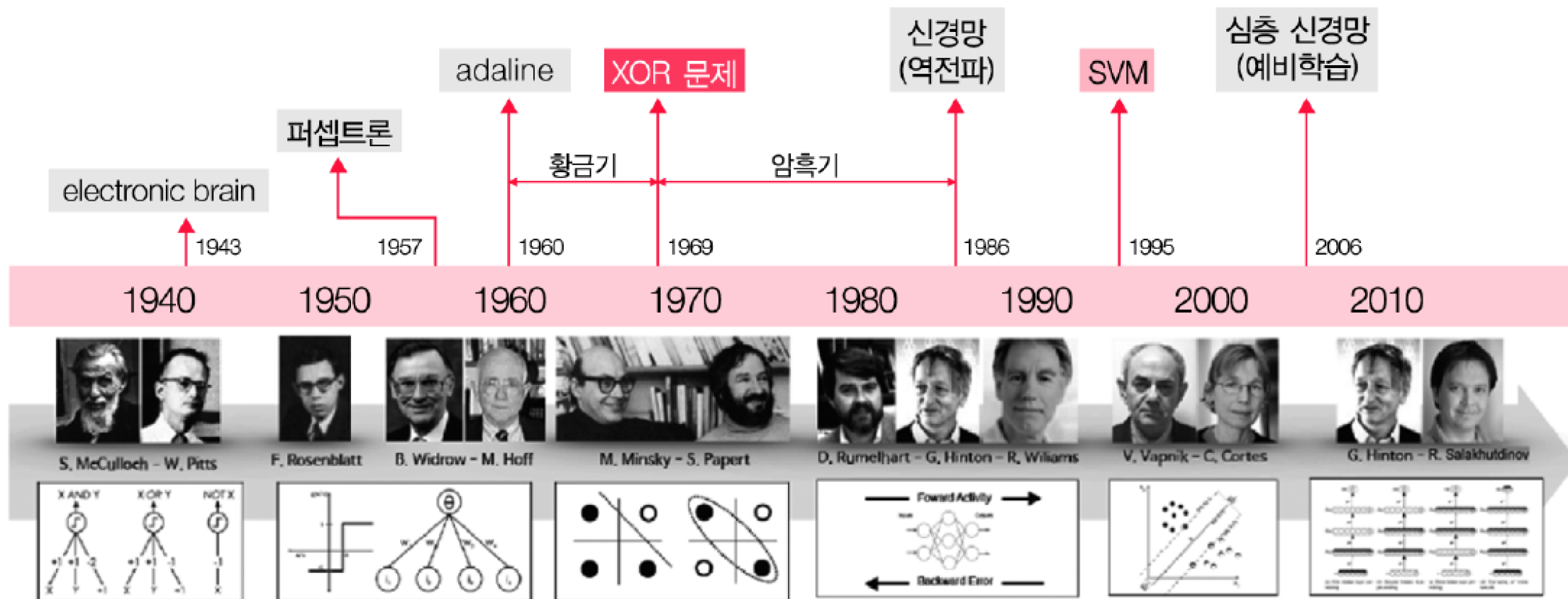


그림 1-1 신경망 연구의 역사

심층 신경망

- 심층 신경망은 입력층과 출력층 사이에 여러 개의 은닉층으로 이루어진 신경망.
심층 신경망은 1개의 은닉층을 가진 천층 신경망과 마찬가지로, 복잡한 비선형 관계를 모형화할 수 있음.
- 딥러닝(심층 신경망 학습에 활용되는 기계학습 방법론)이 관심을 받게 된 이유
 1. 기존 신경망의 단점 극복
 2. 하드웨어 성능의 발전
 3. 데이터의 기하급수적 증가

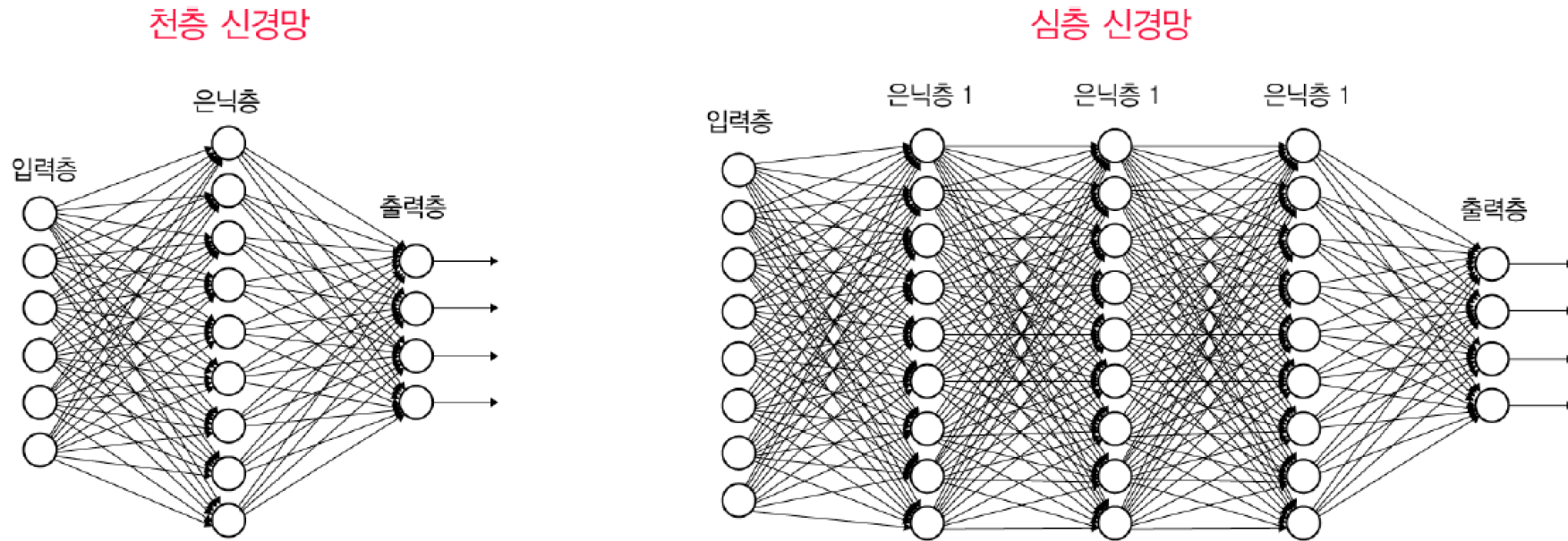


그림 1-2 천층 신경망과 심층 신경망

딥러닝 소프트웨어

- 기계학습은 컴퓨터가 자료를 분석할 수 있게 하는 알고리즘을 개발/프로그래밍하는 것이 핵심. 구글, 페이스북, 애플, 야후 등 글로벌 IT업체들이 기계학습을 연구.
- 딥러닝을 위해 많이 활용되는 소프트웨어
 - 텐서플로: 구글 브레인팀 개발, 현재 가장 많은 사람이 사용
 - Caffe: UC Berkeley에서 관리
 - Theano: 딥러닝 알고리즘을 파이썬으로 쉽게 구현할 수 있도록 해줌
 - 토치: 페이스북과 구글 딥마인드가 사용
 - CNTK: 마이크로소프트 개발
 - Matlab: 상용 소프트웨어
 - R: 통계분석

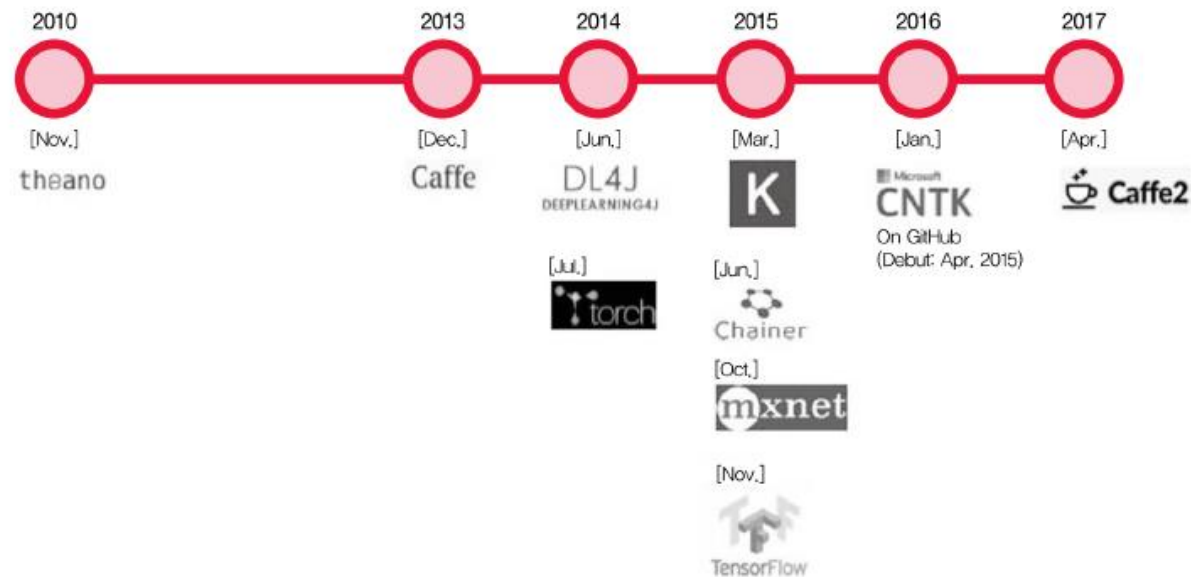


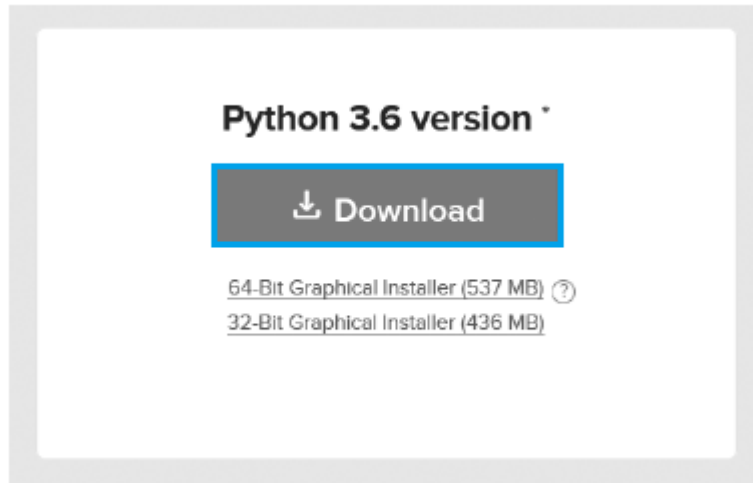
그림 1-3 딥러닝 소프트웨어의 역사



실습 환경 설치

파이썬 설치

- 파이썬은 아나콘다Anaconda를 통해 설치하는 것이 좋음.
- 아나콘다에서 제공하는 패키지를 사용하면 패키지(라이브러리)를 함께 설치할 수 있음
- 아나콘다의 다운로드 사이트(<https://www.anaconda.com/download/>)
- 파이썬 3.x 버전을 다운로드 함.
- 다운로드한 파일을 더블 클릭하여 설치를 시작



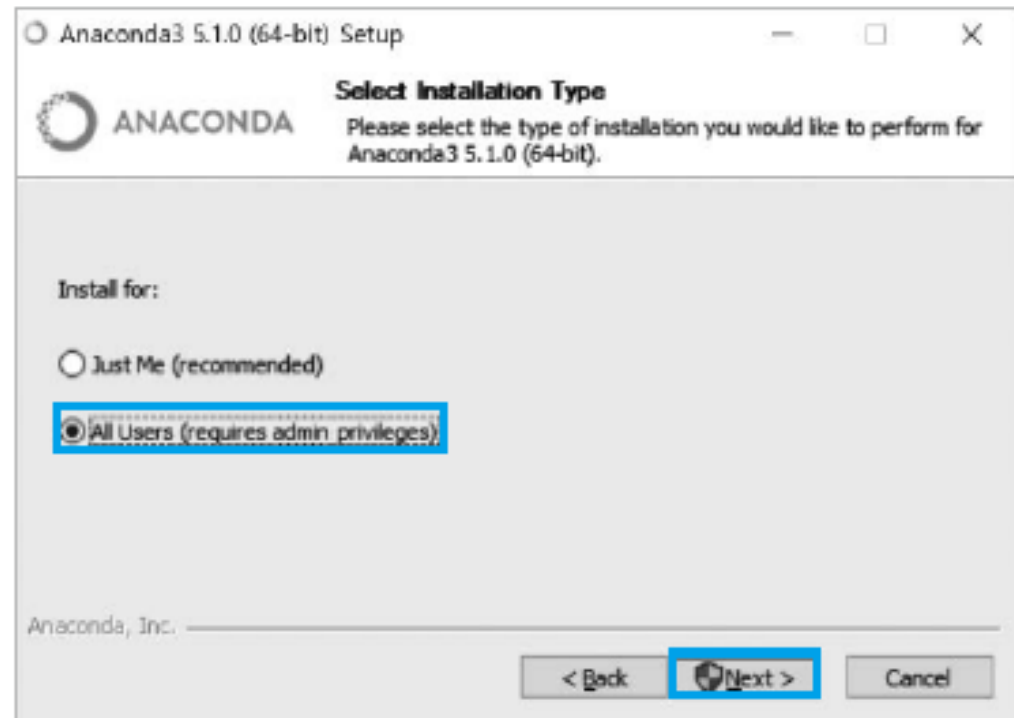
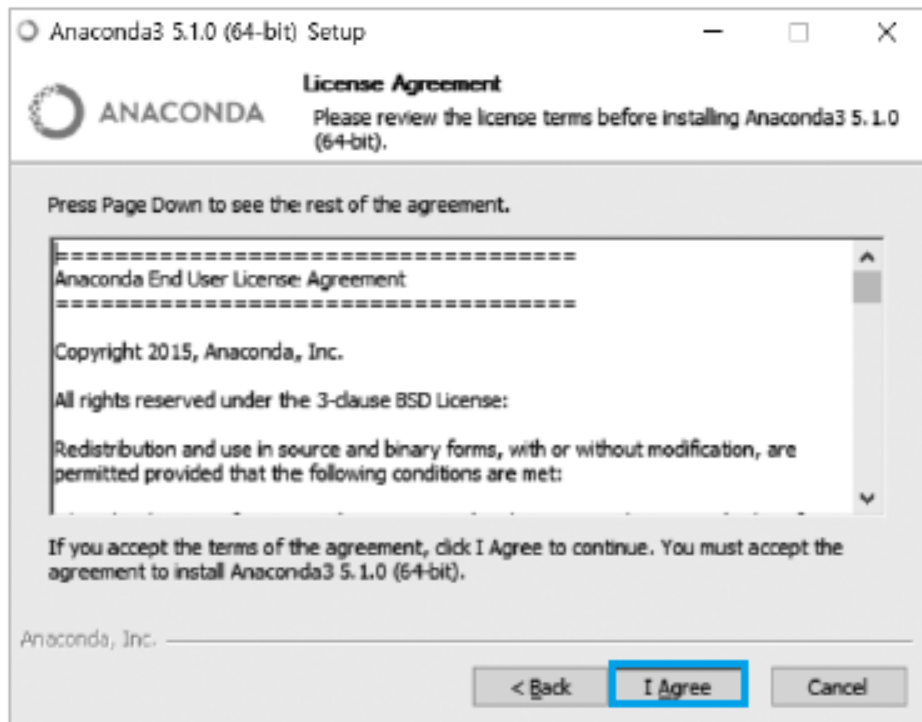
[아나콘다 다운로드 화면]



[Welcome to Anaconda3 5.1.0(64-bit) Setup] 화면(아나콘다 설치)

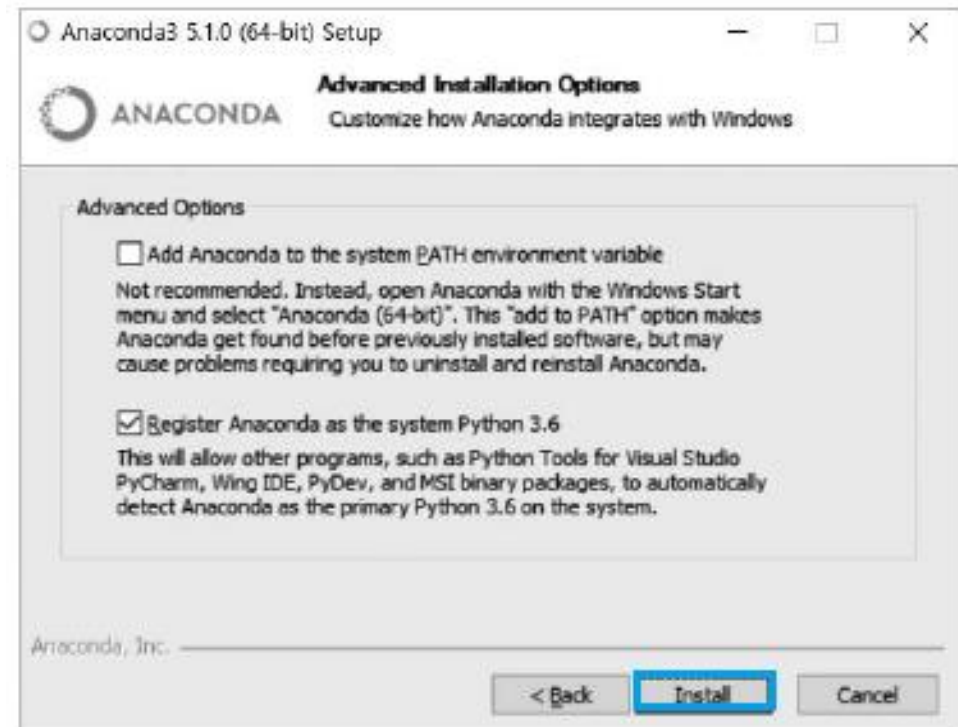
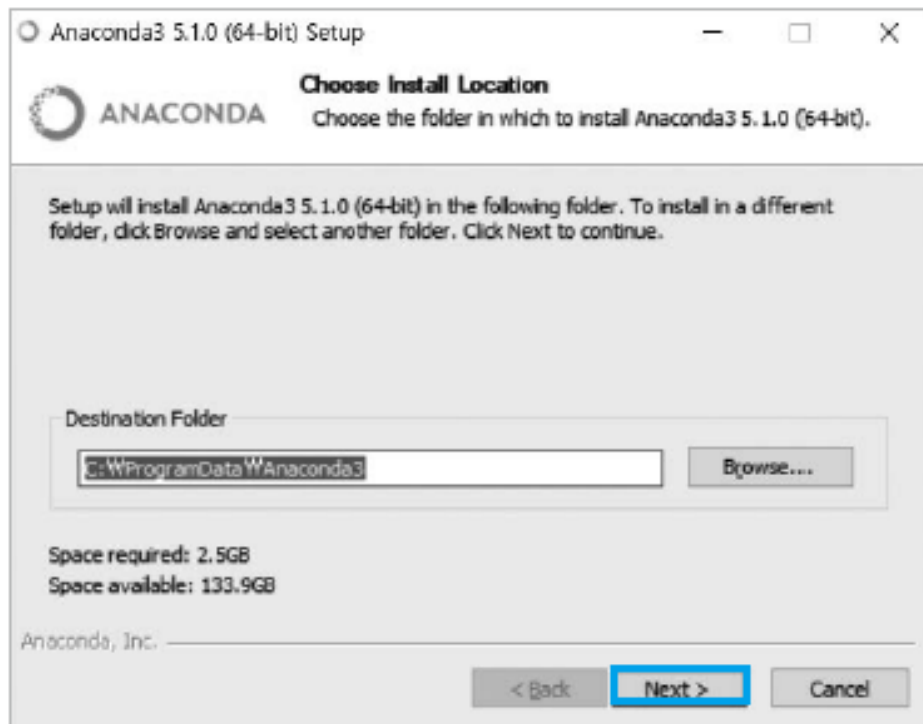
SECTION 02 파이썬 설치

- [License Agreement] 화면에서 라이선스에 동의하고 [I Agree] 버튼을 클릭.
- [Select Installation Type] 화면에서 사용할 유저를 'All Users'로 선택하고 [Next] 버튼을 클릭.



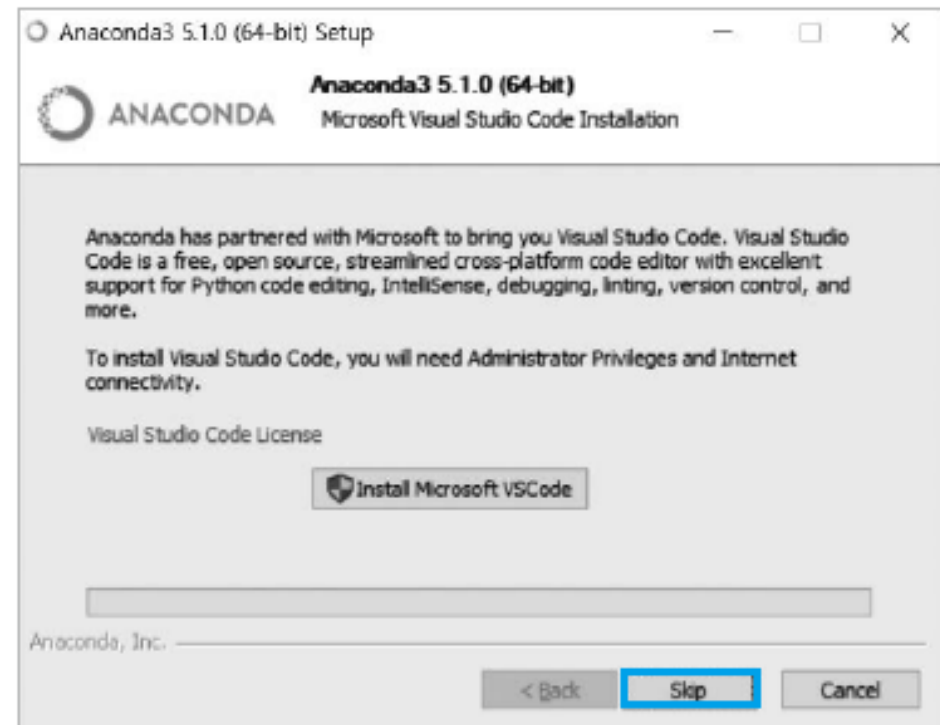
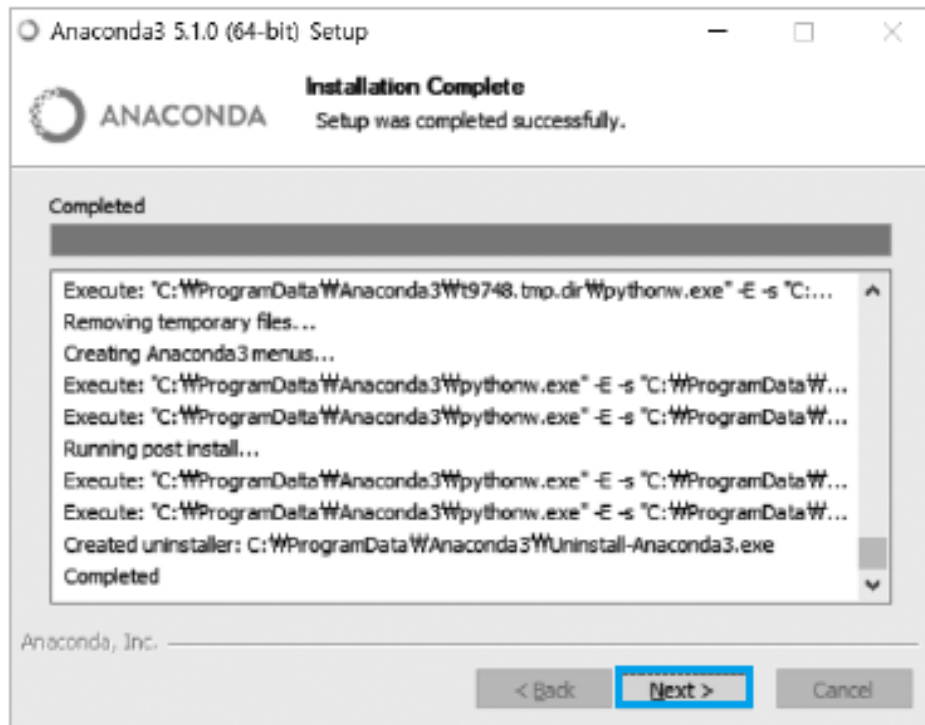
SECTION 02 파이썬 설치

- [Choose Install Location] 화면에서 설치 위치를 확인하고 [Next] 버튼을 클릭.
- [Advanced Installation Options] 화면이 나오면 기본 상태로 [Install] 버튼을 클릭.



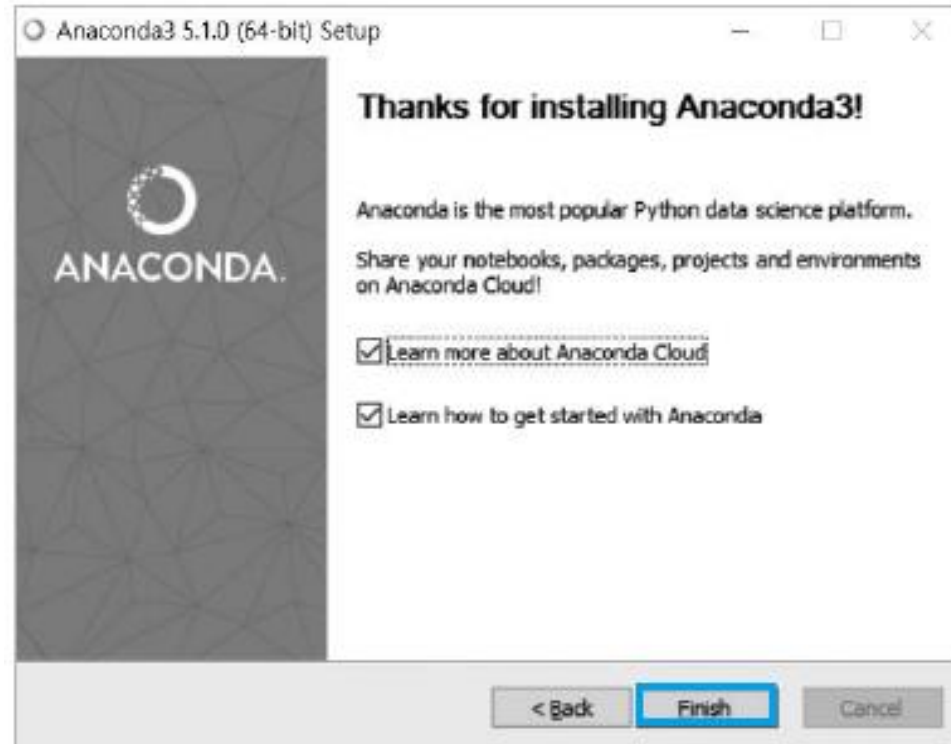
SECTION 02 파이썬 설치

- 아나콘다 설치가 완료되고 [Installation Complete] 화면이 나오면 [Next] 버튼을 클릭.
- Visual Studio Code5 설치 여부는 이 책에서는 사용하지 않으므로 [Skip] 버튼을 클릭.



SECTION 02 파이썬 설치

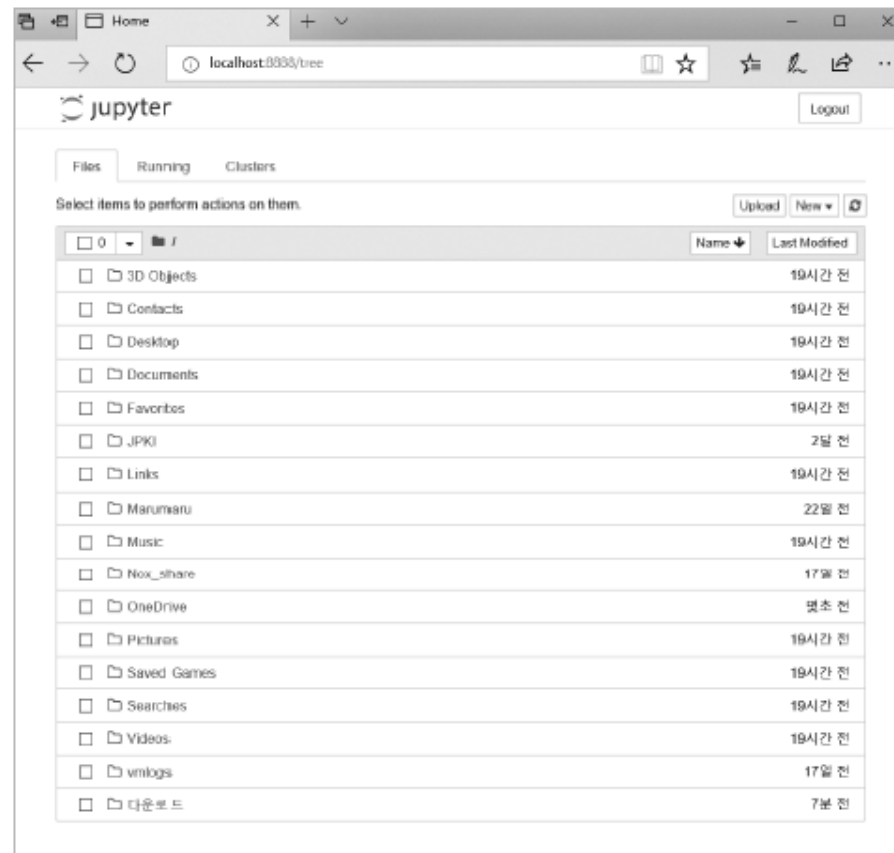
- [Thanks for installing Anaconda3!] 화면이 나타나면 [Finish] 버튼을 클릭하여 완료함.



SECTION 03 주피터 노트북

3.1 주피터 노트북의 이용

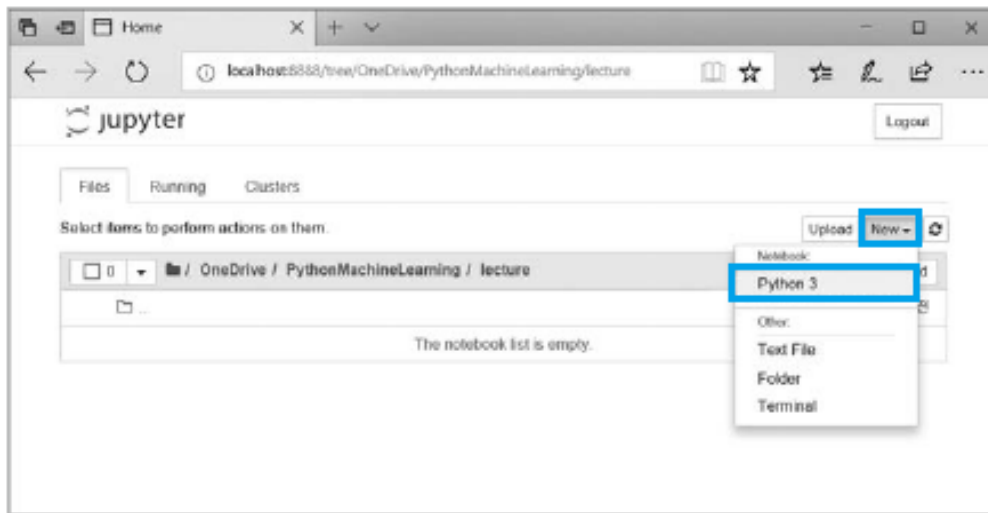
- 시작 메뉴에서 [Anaconda3] 폴더를 선택하고 [Jupyter Notebook]을 클릭함.
- 브라우저가 열리며 주피터 노트북 실행됨.



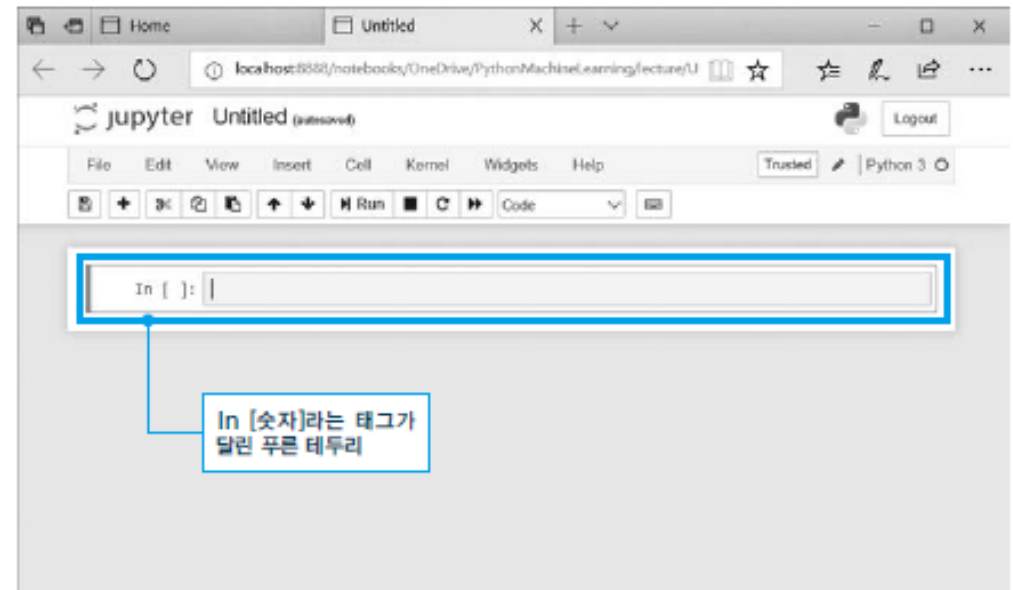
[주피터 노트북 실행시 기동 화면]

SECTION 03 주피터 노트북

- 작업할 폴더에 들어간 뒤, [New]→[Python 3]를 선택함.
- 새로운 탭이 열리고, 태그가 달린 푸른 테두리가 표시.
- 이곳은 '셀'이라고 부르며, 파이썬 프로그램을 작성하는 공간.



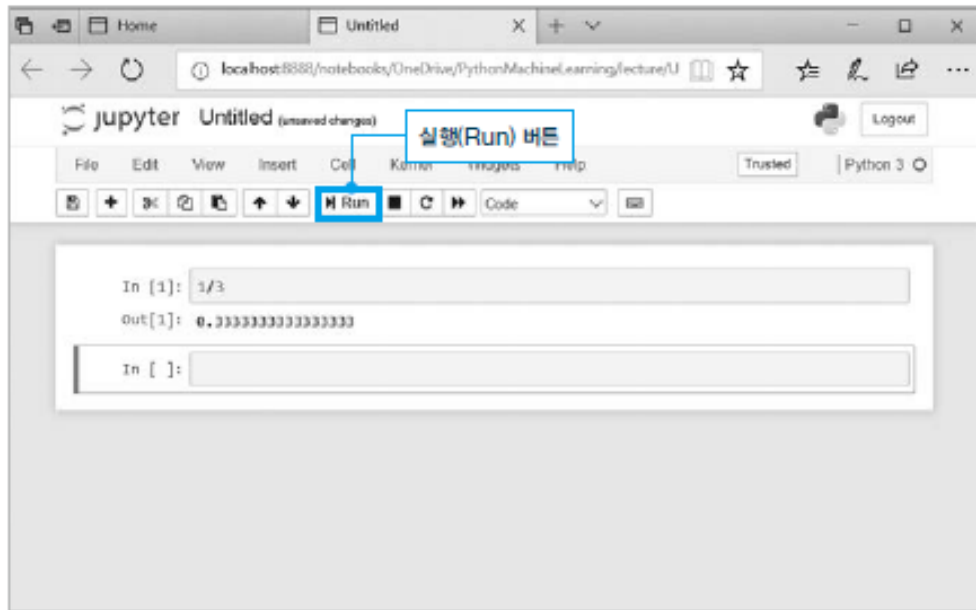
[주피터 노트북으로 파이썬을 실행]



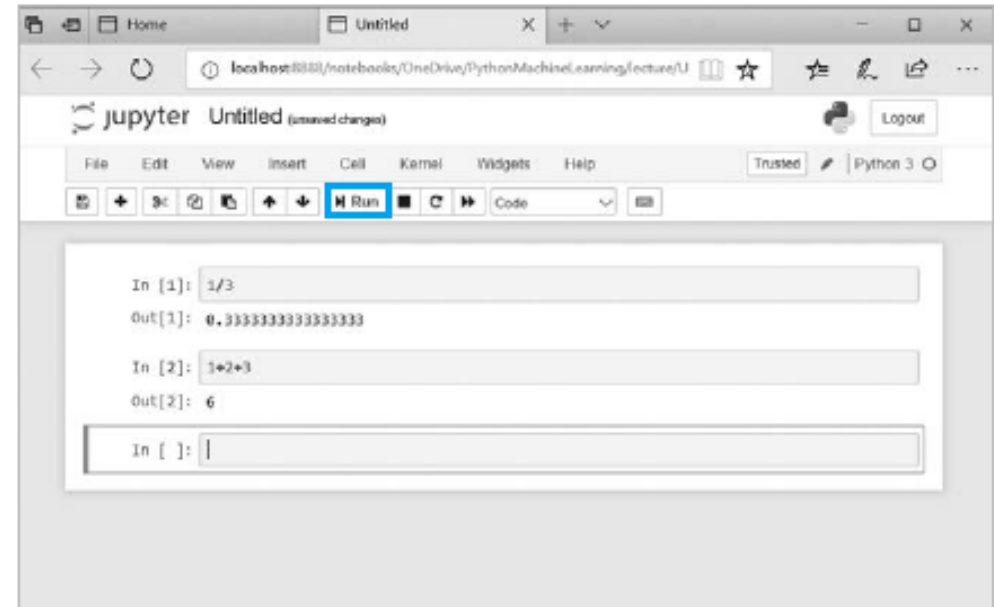
[주피터 노트북의 셀]

SECTION 03 주피터 노트북

- 셀에 '1 / 3'을 입력하고 메뉴의 [Run] 버튼을 클릭. 셀 하단에 결과가 표시되어 다음 셀이 추가됨.
- 다음 셀에 다시 수식을 쓰고, [Run] 버튼을 클릭하면 빠르게 계산을 계속할 수 있음.



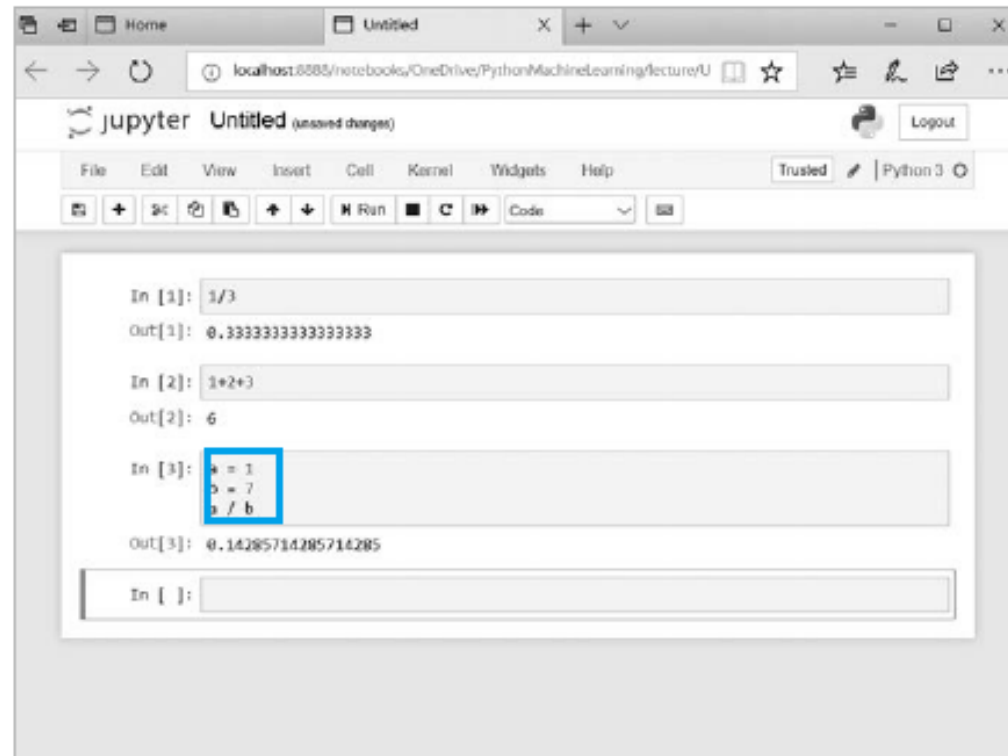
[첫 번째 계산 결과]



[두 번째 계산 결과]

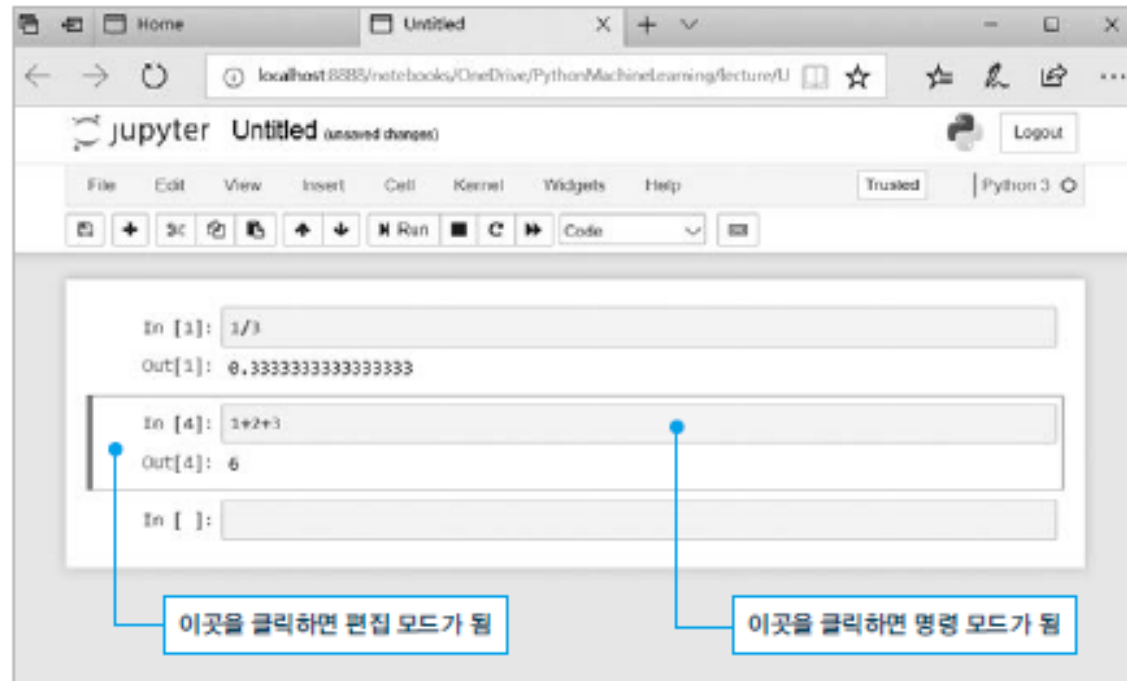
SECTION 03 주피터 노트북

- 하나의 셀에 여러 개의 명령문을 쓸 수도 있음.



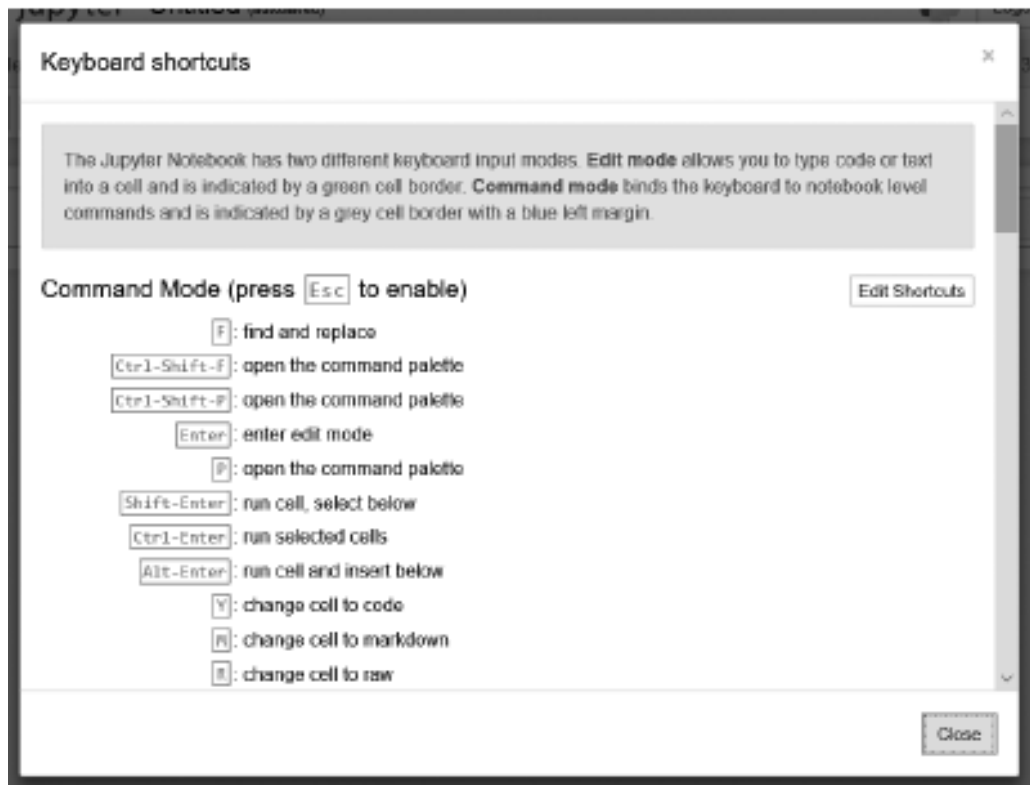
SECTION 03 주피터 노트북

- 셀에는 명령 모드(Command Mode)와 편집 모드(Edit Mode) 두 가지 모드가 있음.
- 편집 모드는 셀에 수식을 입력할 수 있는 모드.
- 명령 모드는 셀 삭제/복사/추가 등으로 셀자체를 조작할 수 있는 모드.
- 명령 모드로 바꾼 뒤, [H]키를 누르면 각각의 모드에 대한 소개와 단축키 목록이 표시됨.

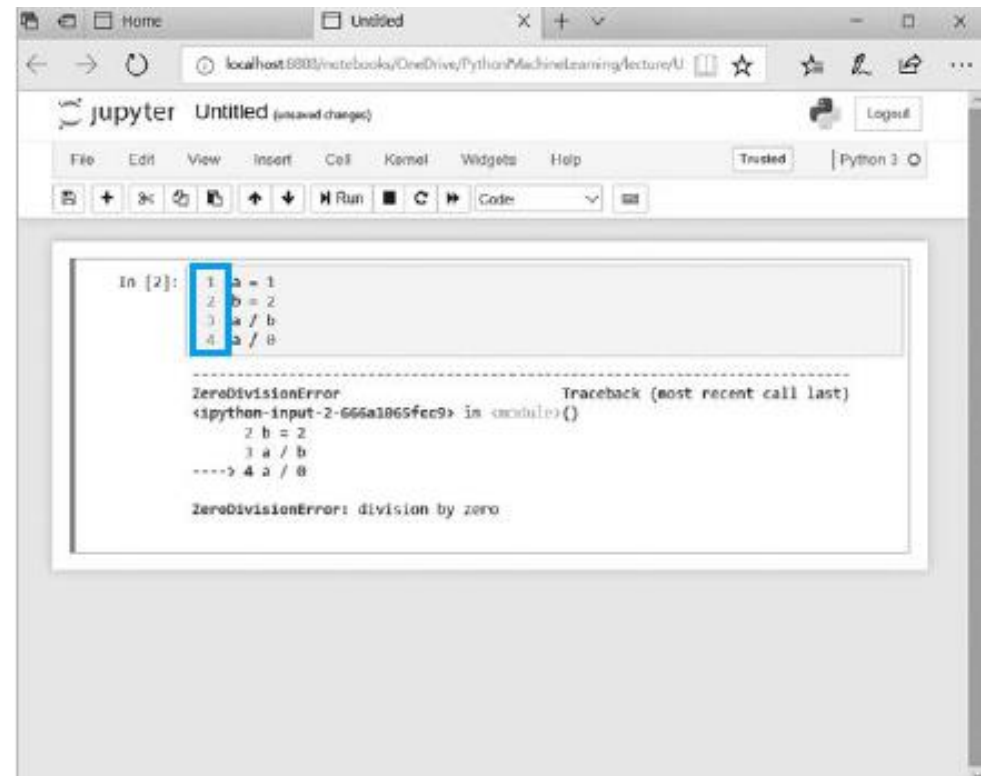


SECTION 03 주피터 노트북

- 명령 모드로 바꾼 뒤, [H]키를 누르면 각각의 모드에 대한 소개와 단축키 목록이 표시됨.
- 명령 모드에서 [A]나 [B]키를 누르면 선택한 셀의 위나 아래에 새 셀이 추가됨.



[각 모드에서의 기능과 단축키 목록(일부분)]

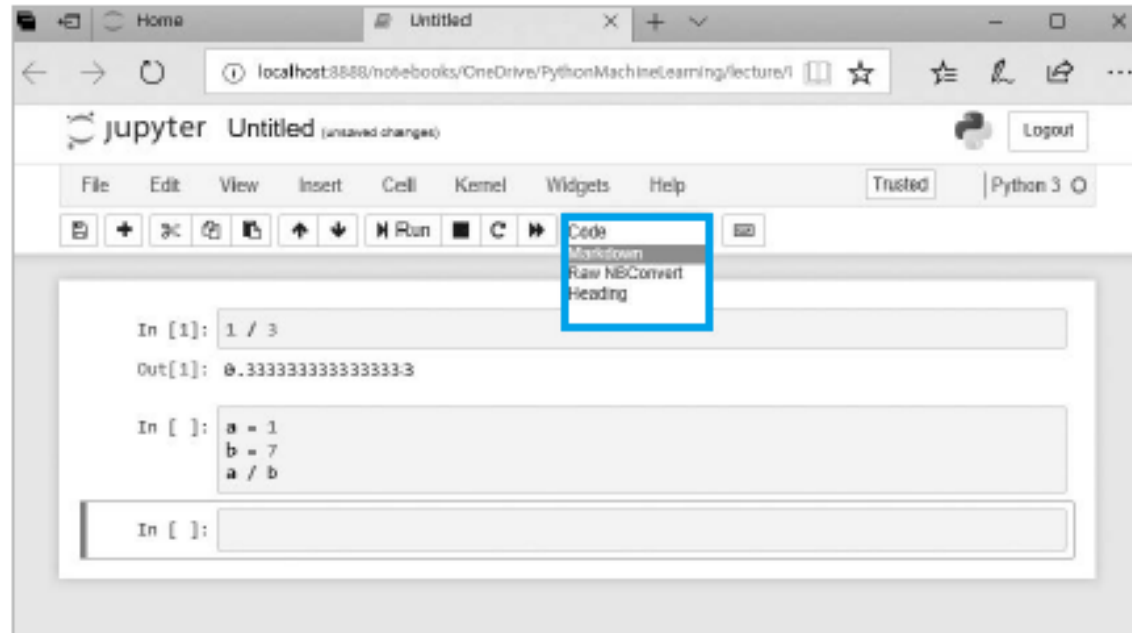


[행 번호 표시]

SECTION 03 주피터 노트북

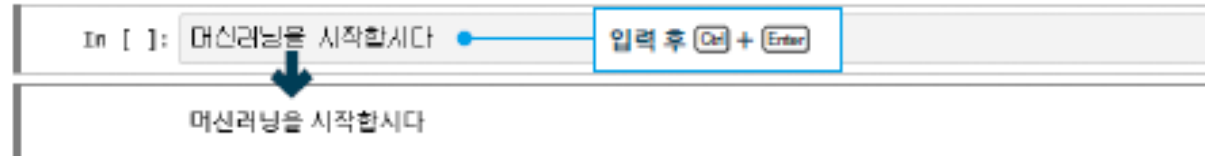
3.2 마크다운 형식으로 입력

- 주피터 노트북에서는 프로그램 코드뿐만 아니라 텍스트 메모를 추가할 수 있음.
- 우측 상단의 드롭다운 메뉴에서 [Code]→[Markdown]을 선택하여 메모 작성 모드로 바꿔본다.

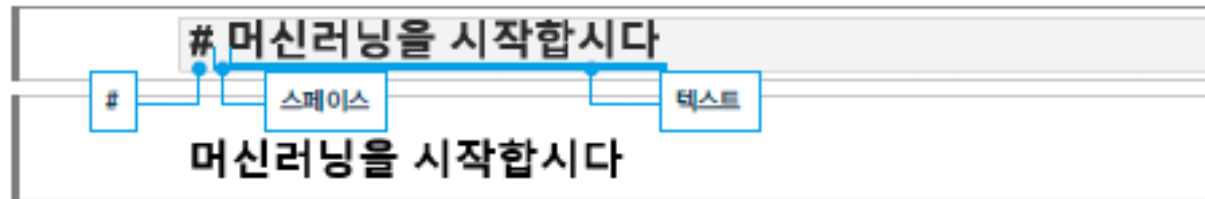


SECTION 03 주피터 노트북

- 마크다운 모드가 되어 일반적인 문장을 입력할 수 있음.
- 문자열의 처음에 [#]키를 누르고, 스페이스 바를 누르면 제목을 입력할 수 있음



- 예를 들어 '머신러닝을 시작합시다'를 입력하고 Ctrl + Enter 를 누르면 테두리가 사라지고 일반적인 문장으로 표시.

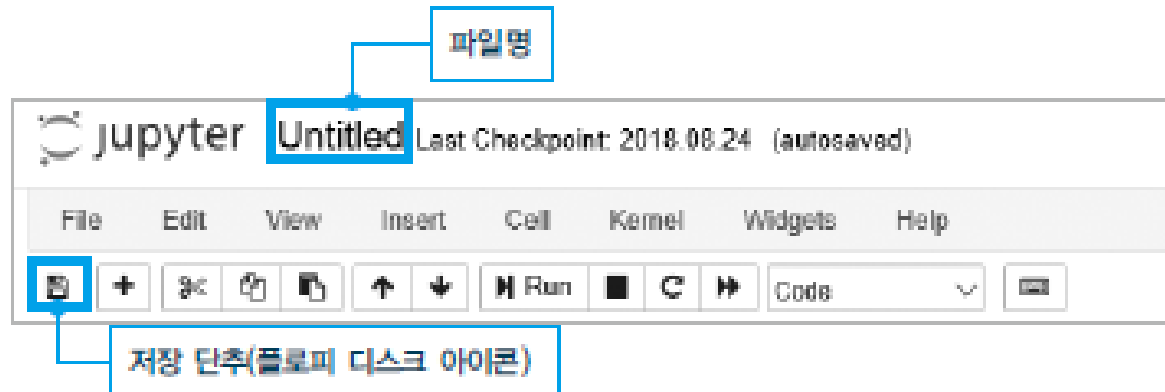


- Ctrl + Enter 키를 누르면 큰 글자로 표시됨. 제목은 ##이나 ###로 제목의 단계(크기)를 내릴 수 있음.

SECTION 03 주피터 노트북

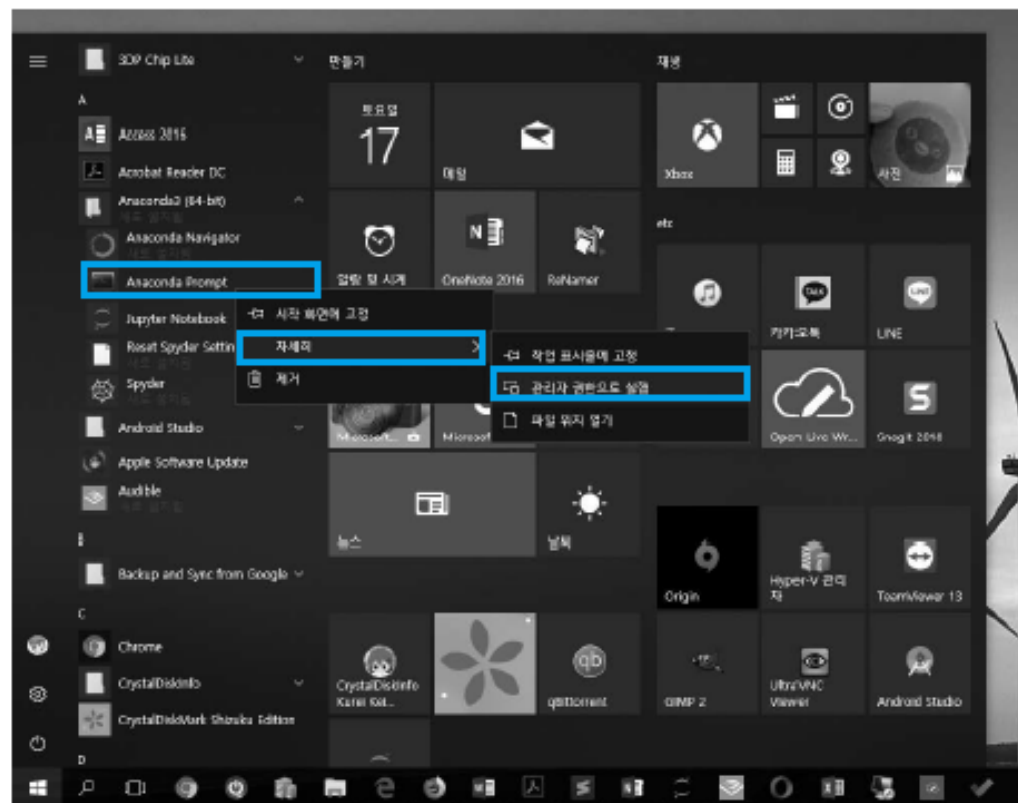
3.3 파일명의 변경

- 파일명은 기본적으로 'Untitled'로 되어 있음.
- 변경하려면 'Untitled'를 클릭하여 원하는 이름을 입력하면 됨.
- 저장 버튼(플로피 디스크 아이콘)를 클릭하면 파일이 저장됨.



SECTION 04 케라스와 텐서플로 설치

- 윈도우의 시작 메뉴에서 [Anaconda3]→[Anaconda Prompt]를 오른쪽 버튼으로 클릭하고, [자세히]→[관리자 권한으로 실행]을 실행함.



SECTION 04 케라스와 텐서플로 설치

- 'cd 폴더명' 명령으로 작업 폴더로 이동, pip install 명령으로 텐서플로를 설치함.
- 텐서플로의 설치가 완료되면 pip install 명령으로 케라스를 설치함.

```
> pip install tensorflow
```

```
관리자: Anaconda Prompt
(base) C:\WINDOWS\system32>cd C:\Users\Wardmond\OneDrive\PythonMachineLearning\lecture
(base) C:\Users\Wardmond\OneDrive\PythonMachineLearning\lecture>pip install tensorflow_
```

[텐서플로의 설치]

```
> pip install keras
```

```
관리자: Anaconda Prompt
(base) C:\Users\Wardmond\OneDrive\PythonMachineLearning\lecture>pip install keras
```

[케라스 설치]

SECTION 04 케라스와 텐서플로 설치

- 셀에 다음과 같이 입력하고 Ctrl + Enter 키를 누릅니다
- 마지막 줄에 'Using TensorFlow backend'라는 표시가 나오면 성공적으로 설치됨.
- 다음과 같이 `keras.__version__` 을 입력하면 버전을 확인할 수 있음. 밑줄(_)이 두 개씩 있음.

```
In import keras
```

```
Out Using TensorFlow backend.
```

[설치 확인]

```
In keras.__version__
```

```
Out '2.1.5'
```

[버전 확인]