

AI 라이브러리 활용

이 찬 우

학습목표 첫 딥러닝

- 1 | 딥러닝 실행을 위한 준비 사항
- 2 | 딥러닝 작업 환경 만들기
- 3 | 미지의 일을 예측하는 원리
- 4 | 폐암 수술 환자의 생존율 예측하기
- 5 | 딥러닝 개괄 잡기

1 첫 딥러닝

- 인공지능 > 머신러닝 > 딥러닝



- 인공지능의 큰 범주 안에 머신러닝이 속하고,
머신러닝의 일부분이 딥러닝
→ 딥러닝을 배우려면 반드시 머신러닝의 기초 개념을 알아야 함

그림 1-1 인공지능, 머신러닝, 딥러닝의 관계

1 | 딥러닝 실행을 위한 준비 사항

데이터

- 딥러닝은 데이터를 이용해 예측 또는 판별을 수행

컴퓨터(CPU? GPU?)

- 일반 CPU에서 동작시킬지 고속 그래픽 처리에 특화된 GPU에서 동작시킬지 선택

프로그램

- 딥러닝을 구동할 수 있게 프로그래밍을 해야함

2 | 딥러닝 작업 환경 만들기

- 딥러닝을 만들고 작동시키는 방법
 - 자신의 컴퓨터에 필요한 프로그램을 설치해 사용
 - 다른 하나는 구글 코랩(Google Colab)을 이용
- 구글 코랩 :

파이썬 개발에 쓰이는 주피터 노트북(Jupyter Notebook) 환경을 구글 클라우드에 마련해 놓은 것
- 구글 코랩을 이용하면 구글 프로세서를 이용해 빠르고 쉽게 딥러닝 코드를 테스트할 수 있음

2 | 딥러닝 작업 환경 만들기

- 다만, 내 컴퓨터가 아니므로 원하는 버전의 패키지를 매번 설치해야 함
- 작업 중이던 데이터를 잃어버릴 수 있다는 점을 주의해야 함
- 주피터 노트북이나 파이참(Pycharm) 등을 내 컴퓨터에 설치하면 데이터를 잃어버릴 걱정 없이 딥러닝 코딩을 할 수 있음

2 | 딥러닝 작업 환경 만들기

- 딥러닝을 작동시킬 때 사람들이 가장 많이 사용하는 대표적인 언어는 파이썬임
- 파이썬만 설치하는 것보다는 파이썬이 포함된 아나콘다(Anaconda)라는 통합 패키지를 설치하는 것이 훨씬 좋음

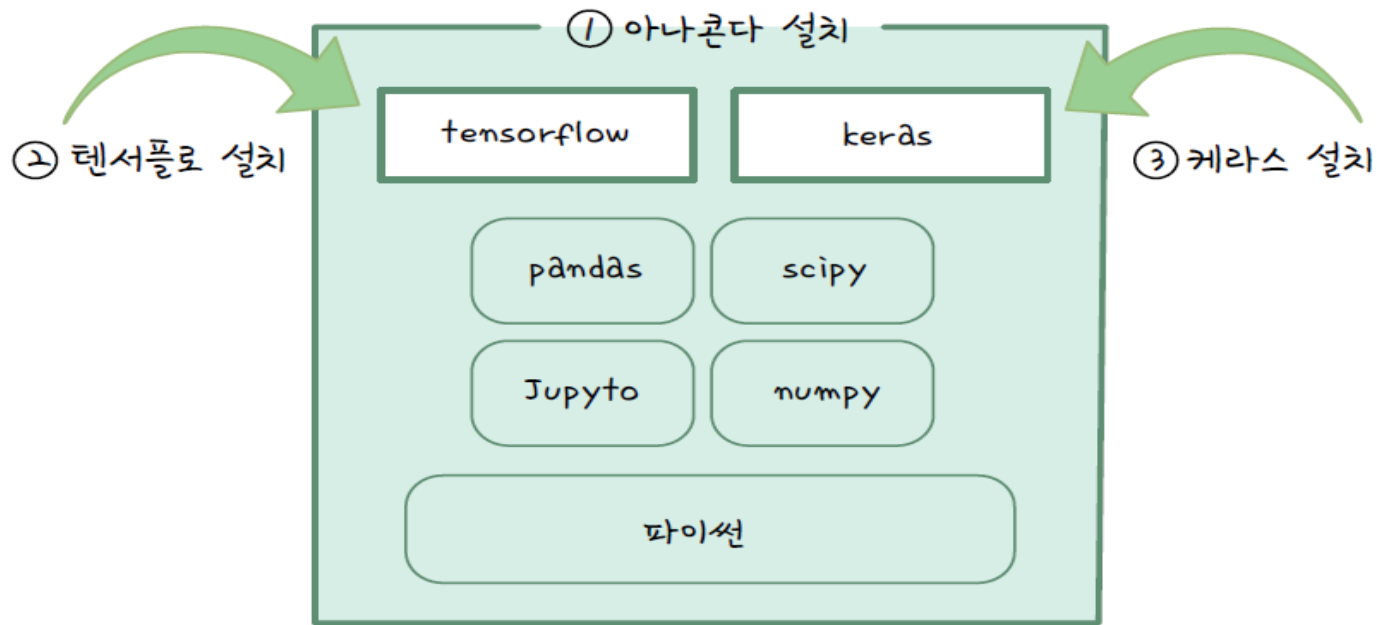
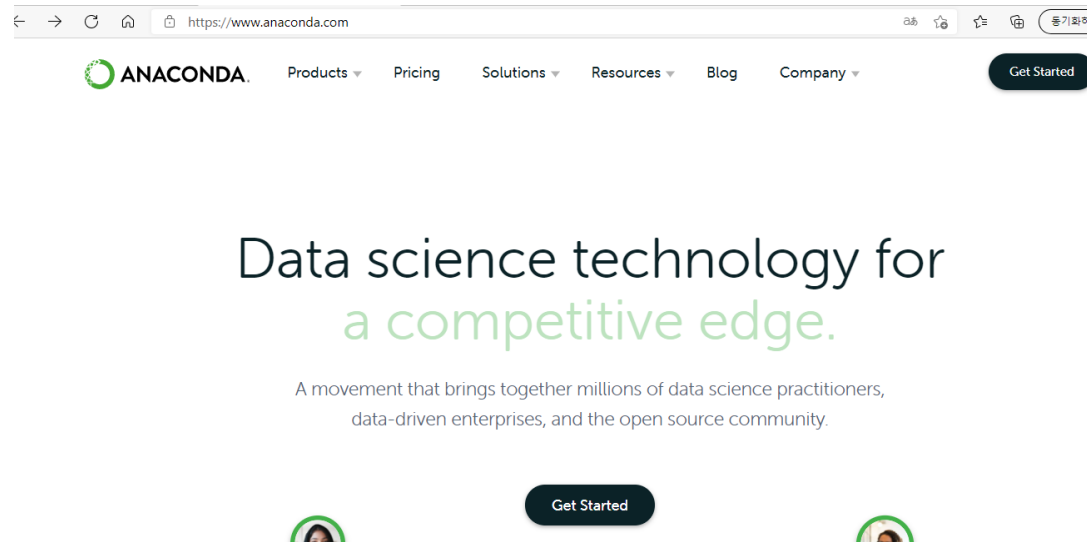


그림 1-2 내 컴퓨터에서 딥러닝을 구동하기 위한 세 가지 설치 작업

2 | 딥러닝 작업 환경 만들기

아나콘다 설치하기

- 아나콘다는 인터넷에서 누구나 다운로드해서 설치할 수 있음
 - www.anaconda.com



아나콘다 홈페이지 접속



Individual Edition
Open Source Distribution



Commercial Edition
Commercial Package Manager



Team Edition
Package Repository



Enterprise Edition
Full Data Science Platform



Professional Services
Data Experts Work Together

Get Started



Individual Edition

Your data science toolkit

With over 20 million users worldwide, the open-source Individual Edition (Distribution) is the easiest way to perform Python/R data science and machine learning on a single machine. Developed for solo practitioners, it is the toolkit that equips you to work with thousands of open-source packages and libraries.

Download

Anaconda Installers

Windows 

Python 3.8

64-Bit Graphical Installer (457 MB)


32-Bit Graphical Installer (403 MB)

MacOS 

Python 3.8

64-Bit Graphical Installer (435 MB)

64-Bit Command Line Installer (428 MB)

Linux 

Python 3.8

64-Bit (x86) Installer (529 MB)

64-Bit (Power8 and Power9) Installer (279 MB)

2 | 딥러닝 작업 환경 만들기

- 다운로드한 인스톨러를 클릭하면 다음과 같은 설치 화면이 나타남



그림 1-5 아나콘다 설치 시작하기

2 | 딥러닝 작업 환경 만들기

- 아나콘다를 PATH 환경에 등록할지를 묻는 창이 나오면, Add Anaconda to my PATH environment variable에 체크한 뒤에 설치를 진행하는 것이 좋음

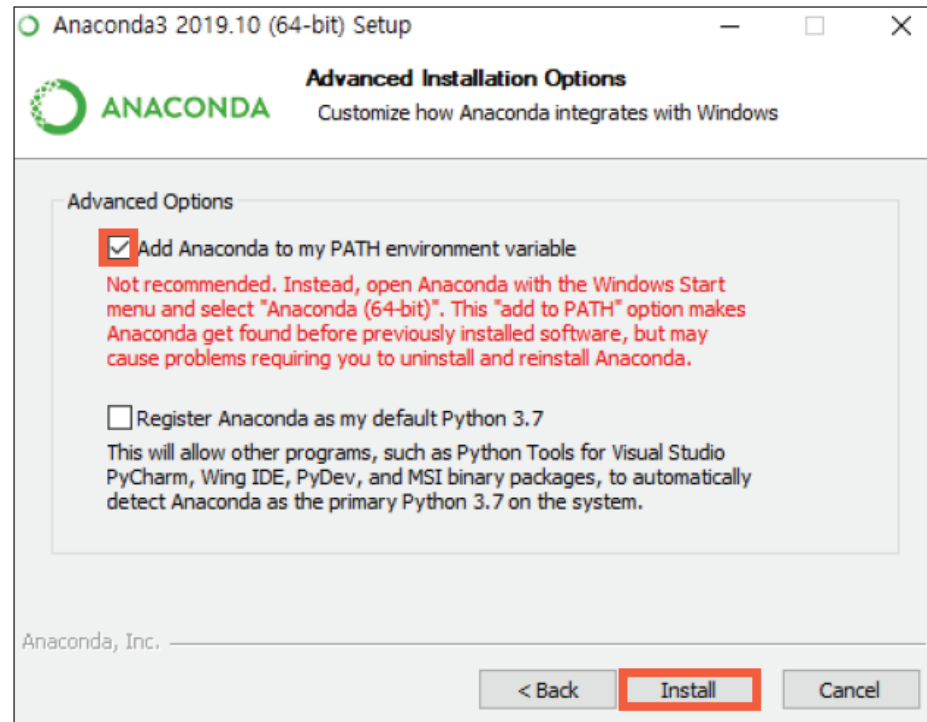


그림 1-6 아나콘다를 PATH 환경에 등록하기

2 | 딥러닝 작업 환경 만들기

- 설치가 완료되면 윈도우 시작 메뉴를 클릭한 다음 Anaconda 3(64-bit)를 클릭
- Anaconda Powershell Prompt를 선택함

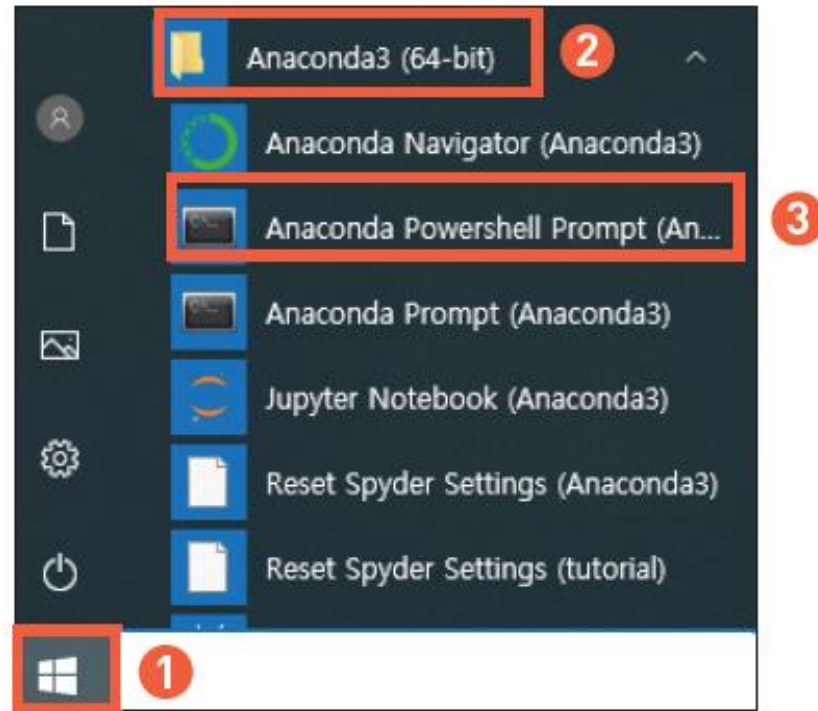
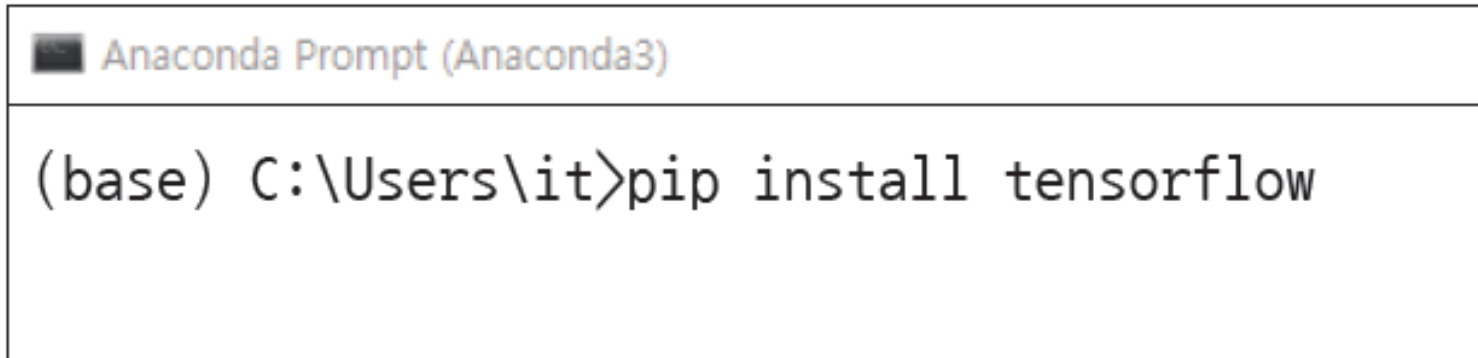


그림 1-7 아나콘다 프롬프트 실행

2 | 딥러닝 작업 환경 만들기

텐서플로 설치하기

- 아나콘다 프롬프트 창이 열리면 `pip install tensorflow`를 입력
- 텐서플로 설치를 진행함

A screenshot of the Anaconda Prompt window. The title bar reads "Anaconda Prompt (Anaconda3)". The command prompt shows the text "(base) C:\Users\it>pip install tensorflow" entered.

```
Anaconda Prompt (Anaconda3)

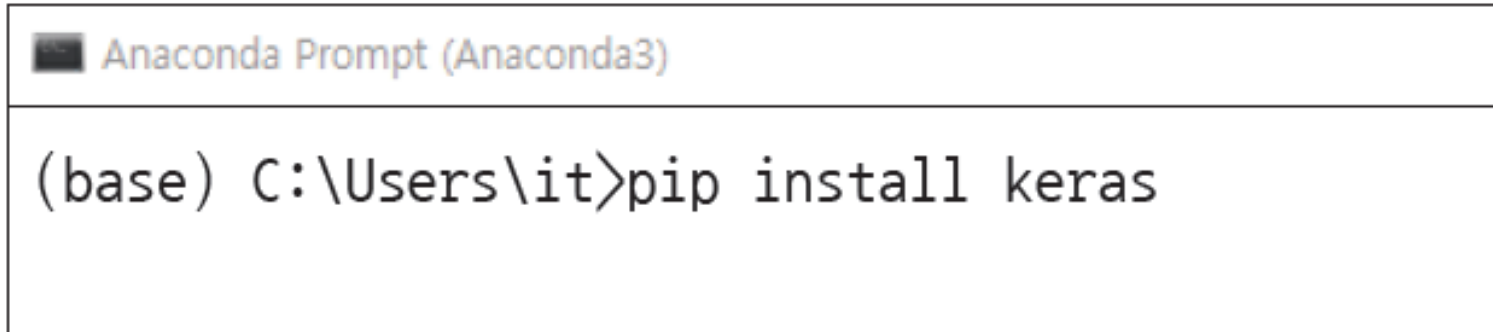
(base) C:\Users\it>pip install tensorflow
```

그림 1-8 텐서플로 설치

2 | 딥러닝 작업 환경 만들기

케라스 설치하기

- 다음과 같이 pip install keras를 입력하여 케라스 설치를 진행함



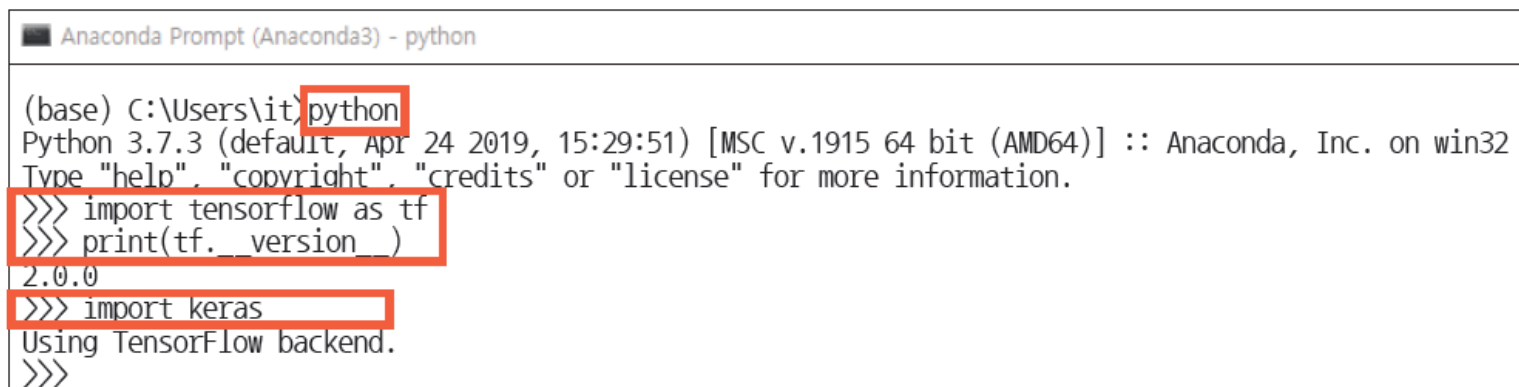
```
Anaconda Prompt (Anaconda3)  
(base) C:\Users\it>pip install keras
```

그림 1-9 케라스 설치

2 | 딥러닝 작업 환경 만들기

텐서플로 및 케라스 설치 확인하기

- 제대로 설치됐는지 확인하려면 다음과 같이 python을 입력
- `import tensorflow as tf`를 입력해 텐서플로를 불러옴
- `print(tf.__version__)`을 입력했을 때 텐서플로의 버전이 출력되면 설치가 끝남
- 케라스의 경우, `import keras`를 입력했을 때 Using TensorFlow backend라고 출력되면 됨



```
Anaconda Prompt (Anaconda3) - python

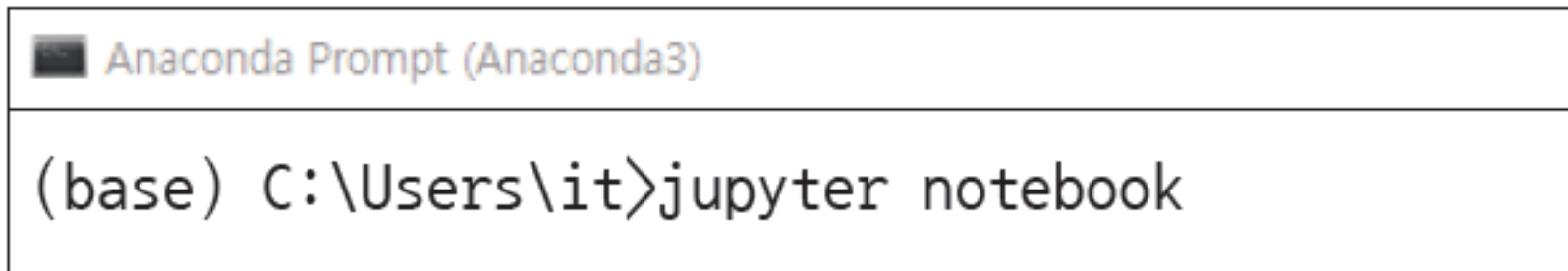
(base) C:\Users\it>python
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorflow as tf
>>> print(tf.__version__)
2.0.0
>>> import keras
Using TensorFlow backend.
>>>
```

그림 1-10 텐서플로 및 케라스 설치 확인하기

2 | 딥러닝 작업 환경 만들기

딥러닝 실행하기

- 딥러닝을 만들고 실행할 때 이용하는 대표적인 툴은 앞서 말한 주피터 노트북과 파이참 임
- 주피터 노트북은 대화형으로 결과를 바로 확인할 수 있음
- 웹 브라우저를 사용하기 때문에 가볍고 효율적이라는 장점이 있음
- 아나콘다 프롬프트에서 다음과 같이 jupyter notebook이라고 입력함



```
Anaconda Prompt (Anaconda3)  
(base) C:\Users\it>jupyter notebook
```

그림 1-11 주피터 노트북 실행

2 | 딥러닝 작업 환경 만들기

- 이제 다음과 같이 주피터 노트북 화면이 보일 것임

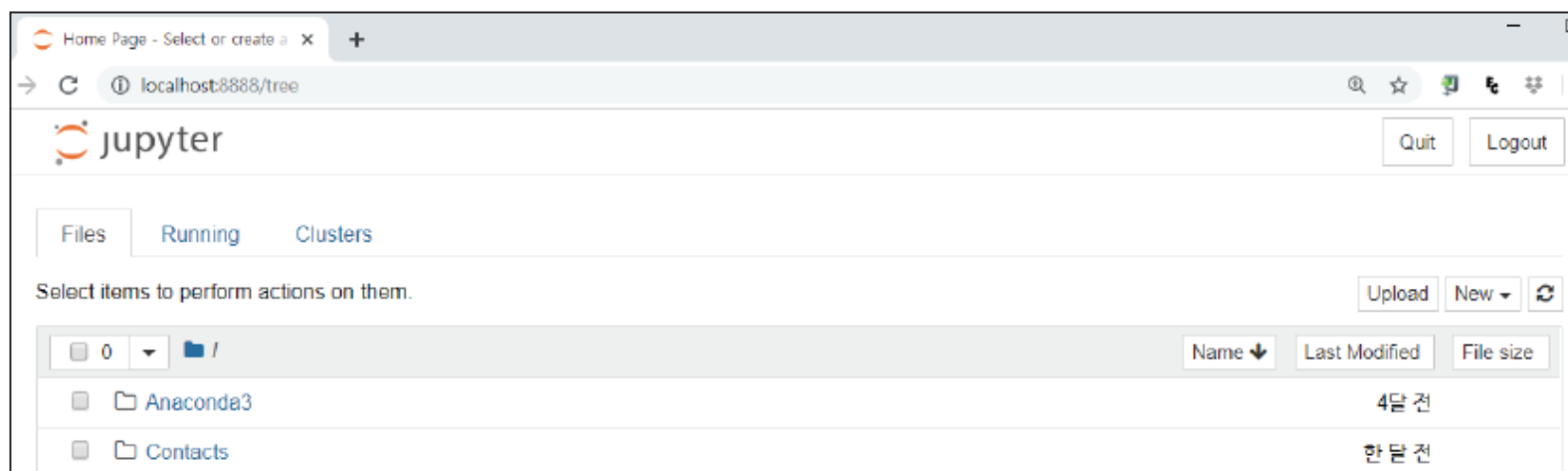
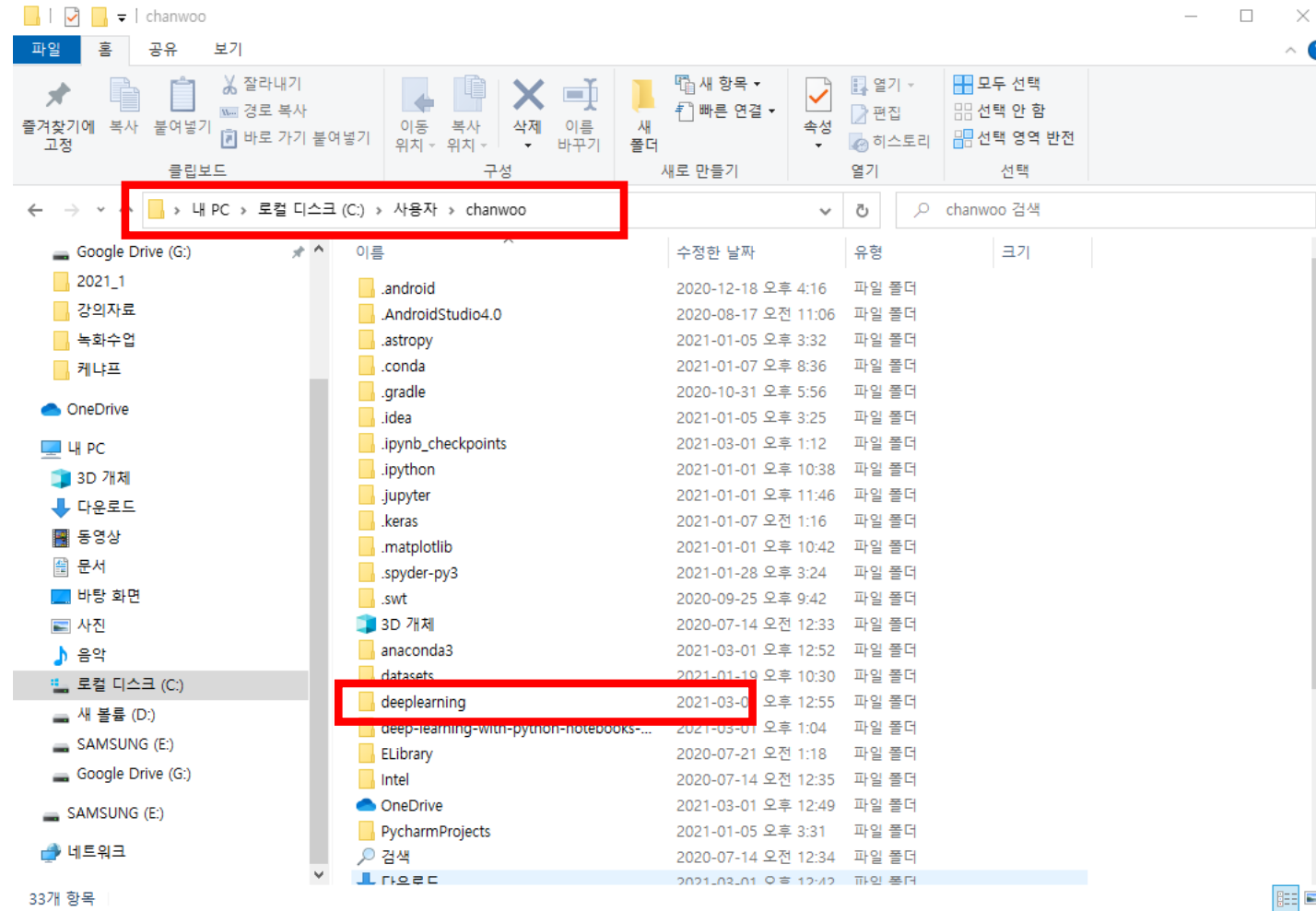


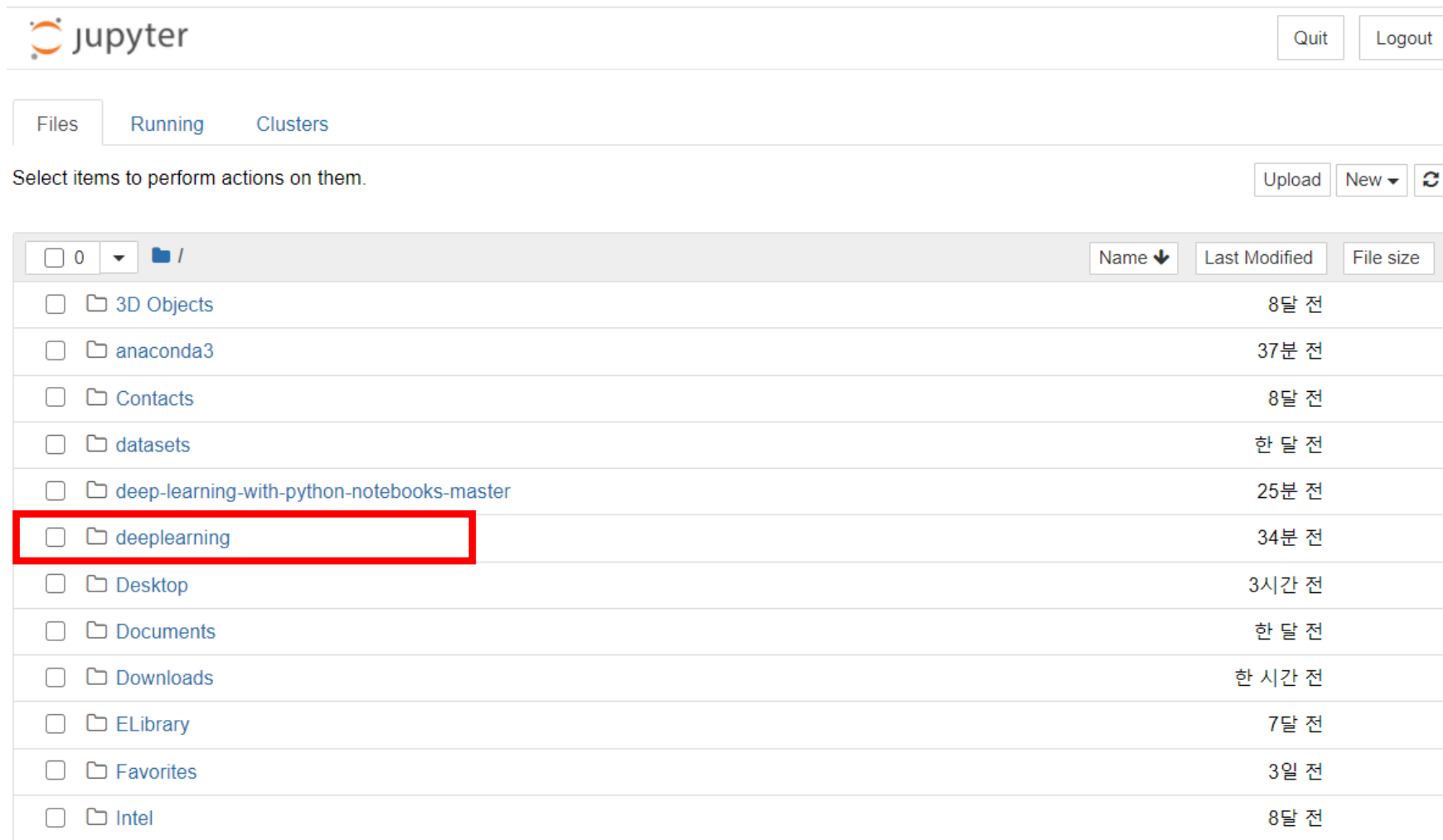
그림 1-12 주피터 노트북 화면

- 교과서 9 page 참고 파일 다운로드 후 압축을 풀고 그 폴더 전체를
아래 보이는 것 처럼 내PC>로컬디스크(C)>사용자>각자PC> 에 deeplearning 폴더를 카피



2 | 딥러닝 작업 환경 만들기

- deeplearning > run_project를 차례대로 클릭
- 예제 소스가 있는 폴더로 이동함



The image shows the JupyterLab interface. At the top, there's a header with the Jupyter logo and 'Quit' and 'Logout' buttons. Below the header, there are tabs for 'Files', 'Running', and 'Clusters'. The 'Files' tab is active. Below the tabs, there's a message 'Select items to perform actions on them.' and buttons for 'Upload', 'New', and a refresh icon. The main area is a file browser showing a list of folders. The 'deeplearning' folder is highlighted with a red box. The file browser has a table-like structure with columns for 'Name', 'Last Modified', and 'File size'.

	Name ↓	Last Modified	File size
<input type="checkbox"/>	0		
<input type="checkbox"/>	3D Objects	8달 전	
<input type="checkbox"/>	anaconda3	37분 전	
<input type="checkbox"/>	Contacts	8달 전	
<input type="checkbox"/>	datasets	한 달 전	
<input type="checkbox"/>	deep-learning-with-python-notebooks-master	25분 전	
<input type="checkbox"/>	deeplearning	34분 전	
<input type="checkbox"/>	Desktop	3시간 전	
<input type="checkbox"/>	Documents	한 달 전	
<input type="checkbox"/>	Downloads	한 시간 전	
<input type="checkbox"/>	ELibrary	7달 전	
<input type="checkbox"/>	Favorites	3일 전	
<input type="checkbox"/>	Intel	8달 전	

2 | 딥러닝 작업 환경 만들기

- 01_My_First_Deeplearning.ipynb를 클릭하면 파일이 열림

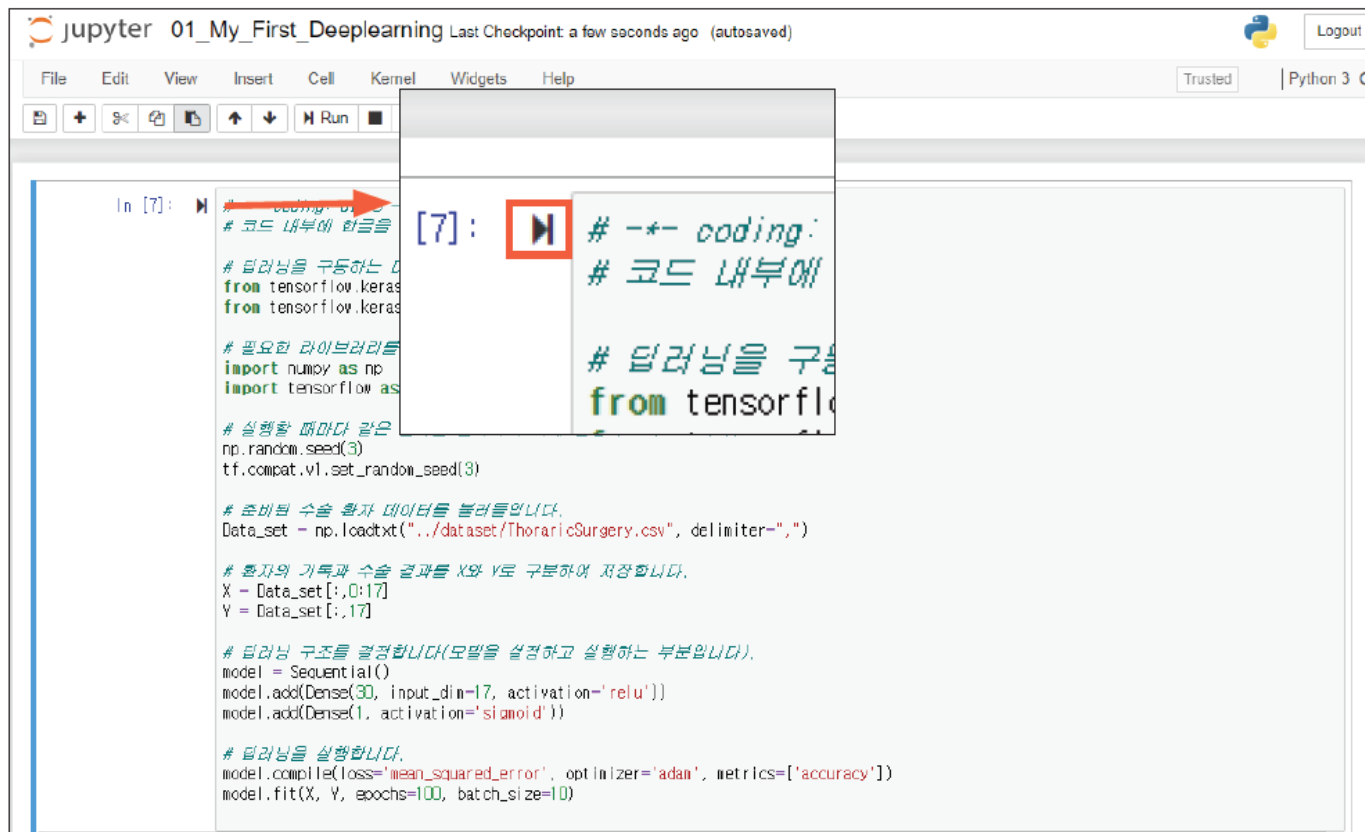
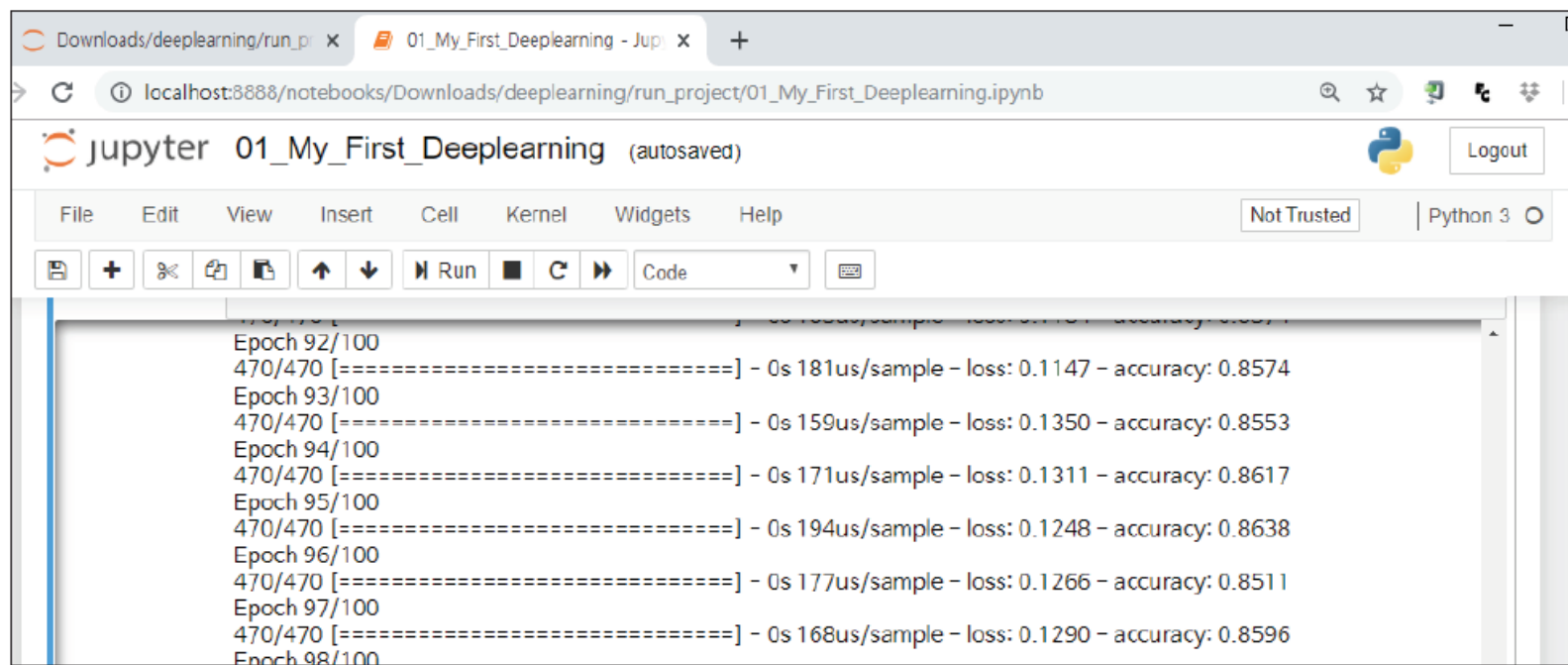


그림 1-14 주피터 노트북으로 딥러닝 실행하기

2 | 딥러닝 작업 환경 만들기

- 그림 1-15과 같이 하단에 실행 결과가 출력되면 정상적으로 실행된 것임



The screenshot shows a web browser window with two tabs: 'Downloads/deeplearning/run_p...' and '01_My_First_Deeplearning - Jup...'. The address bar shows 'localhost:8888/notebooks/Downloads/deeplearning/run_project/01_My_First_Deeplearning.ipynb'. The Jupyter interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help), a toolbar with icons for file operations and execution, and a status bar indicating 'Not Trusted' and 'Python 3'. The main area displays the execution output of a code cell, showing training progress for epochs 92 through 98. Each epoch's output includes the number of samples (470/470), a progress bar, and performance metrics (loss and accuracy).

```
Epoch 92/100
470/470 [=====] - 0s 181us/sample - loss: 0.1147 - accuracy: 0.8574
Epoch 93/100
470/470 [=====] - 0s 159us/sample - loss: 0.1350 - accuracy: 0.8553
Epoch 94/100
470/470 [=====] - 0s 171us/sample - loss: 0.1311 - accuracy: 0.8617
Epoch 95/100
470/470 [=====] - 0s 194us/sample - loss: 0.1248 - accuracy: 0.8638
Epoch 96/100
470/470 [=====] - 0s 177us/sample - loss: 0.1266 - accuracy: 0.8511
Epoch 97/100
470/470 [=====] - 0s 168us/sample - loss: 0.1290 - accuracy: 0.8596
Epoch 98/100
```

그림 1-15 01_My_First_Deeplearning.ipynb 실행 결과

- 아나콘다에 텐서플로 설치 하는 방법에 대한 자세한 내용 참조

CPU 버전

GPU 버전에 따라 참조 하여 설치하고 테스트 부분까지 참고 바랍니다.

[아나콘다에 텐서플로 2 설치하기](https://copycoding.tistory.com/341)

<https://copycoding.tistory.com/341>

<https://alinear.tistory.com/3>

3 | 미지의 일을 예측하는 원리

- 머신러닝 :

기존 데이터를 이용해 아직 일어나지 않은 미지의 일을 예측하기 위해 만들어진 기법

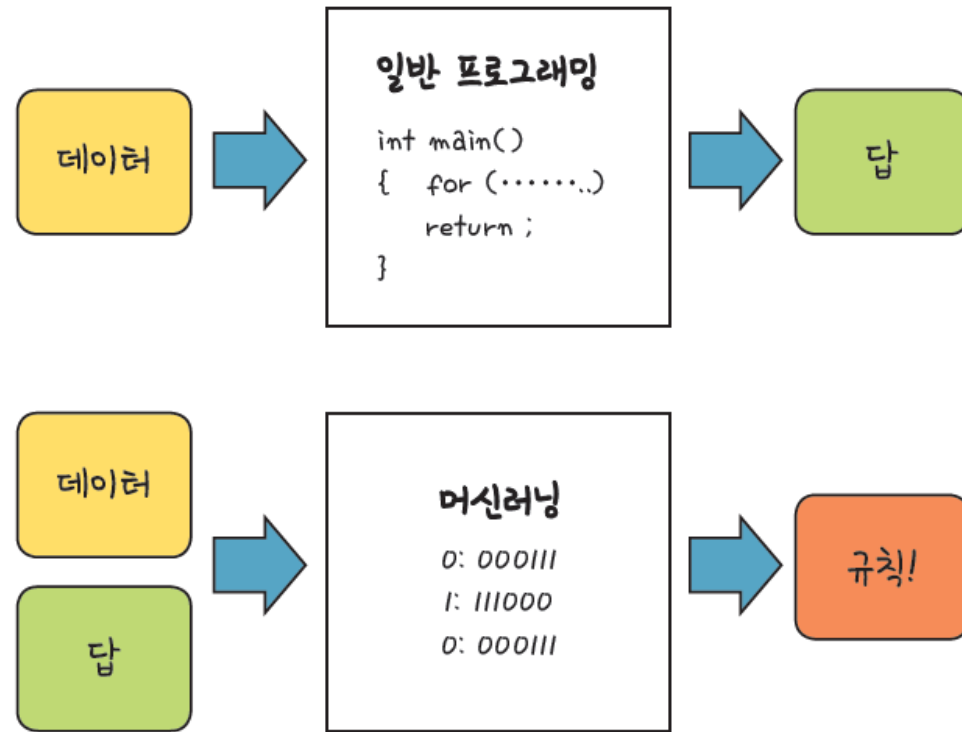


그림 1-16 머신러닝과 일반 프로그래밍 비교

3 | 미지의 일을 예측하는 원리

- 학습(training) :

데이터가 입력되고 패턴이 분석되는 과정

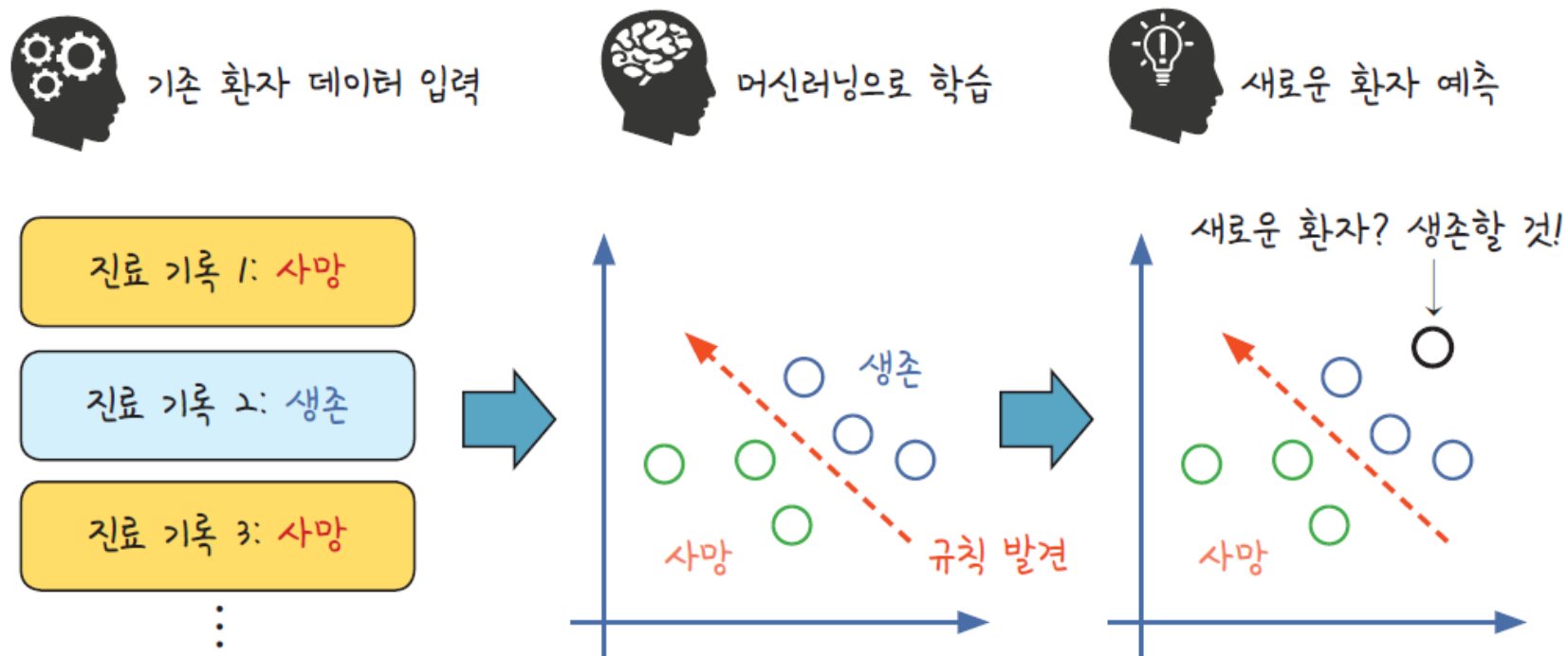


그림 1-17 머신러닝의 학습 및 예측 과정

4 | 폐암 수술 환자의 생존율 예측하기

- 불과 몇 년 전만 해도 딥러닝 알고리즘을 만들려면 코드를 길게 작성해야 했지만, 지금은 단 몇 줄의 코드로도 완벽히 구동할 수 있게 됨

코드 1-1 폐암 수술 환자의 생존율 예측하기

- 예제 소스: run_project/01_My_First_Deeplearnig.ipynb

```
# 딥러닝을 구동하는 데 필요한 케라스 함수 호출  
from tensorflow.keras.models import Sequential  
from tensorflow.keras.layers import Dense  
  
# 필요한 라이브러리 불러오기  
import numpy as np  
import tensorflow as tf
```

4 | 폐암 수술 환자의 생존율 예측하기

```
# 실행할 때마다 같은 결과를 출력하기 위해 설정하는 부분
```

```
np.random.seed(3)
```

```
tf.random.set_seed(3)
```

```
# 준비된 수술 환자 데이터를 불러오기
```

```
Data_set = np.loadtxt("../dataset/ThoraricSurgery.csv", delimiter=",")
```

```
# 환자의 기록과 수술 결과를 X와 Y로 구분하여 저장
```

```
X = Data_set[:,0:17]
```

```
Y = Data_set[:,17]
```

4 | 폐암 수술 환자의 생존율 예측하기

```
# 딥러닝 구조를 결정(모델을 설정하고 실행)
```

```
model = Sequential( )
```

```
model.add(Dense(30, input_dim=17, activation='relu'))
```

```
model.add(Dense(1, activation='sigmoid'))
```

```
# 딥러닝 실행
```

```
model.compile(loss='mean_squared_error', optimizer='adam',  
metrics=['accuracy'])
```

```
model.fit(X, Y, epochs=100, batch_size=10)
```

4 | 폐암 수술 환자의 생존율 예측하기

실행
결과



Train on 470 samples

Epoch 1/100

470/470 [=====] - 0s 812us/sample - loss:
0.1495 - accuracy: 0.8404

Epoch 2/100

470/470 [=====] - 0s 106us/sample - loss:
0.1447 - accuracy: 0.8511

(중략)

Epoch 99/100

470/470 [=====] - 0s 108us/sample - loss:
0.1199 - accuracy: 0.8638

Epoch 100/100

470/470 [=====] - 0s 106us/sample - **loss: 0.1192**
- **accuracy: 0.8638**

4 | 폐암 수술 환자의 생존율 예측하기

- loss는 예측이 실패할 확률, accuracy는 예측이 성공할 확률임
- 예측 성공률은 데이터를 분석해 데이터를 확장하거나, 딥러닝 구조를 적절하게 바꾸는 등의 노력으로 더 향상될 수 있음
- 그뿐만 아니라 학습에 사용되지 않은 데이터를 따로 모아 테스트를 해 보면서 이 예측 성공률이 정말로 가능한지를 확인하는 과정까지 거치게 됨
- 이러한 '최적화 과정'을 진행하려면 딥러닝의 구동 원리를 이해해야 함

5 | 딥러닝의 개괄 잡기

첫 번째 부분: 데이터 분석과 입력

- 데이터를 불러와서 사용할 수 있게 만들어 주는 부분임

```
# 필요한 라이브러리를 불러옴
```

```
import numpy as np
```

```
(중략)
```

```
# 준비된 수술 환자 데이터 불러들이기
```

```
Data_set = np.loadtxt("../dataset/ThoracicSurgery.csv",  
                        delimiter=",")
```

```
# 환자의 기록과 수술 결과를 X와 Y로 구분하여 저장
```

```
X = Data_set[:,0:17]
```

```
Y = Data_set[:,17]
```

5 | 딥러닝의 개괄 잡기

- 파이썬
 - 초보자부터 전문가까지 사용자 폭이 넓은 프로그래밍 언어임
 - 다양한 플랫폼에서 사용할 수 있음
 - 특히 라이브러리가 풍부하여 연구 기관 및 산업계에서 두루 사용되고 있음
- 라이브러리
 - 특정 기능을 담은 작은 프로그램(모듈, module)을 말함
 - 라이브러리를 불러올 때 사용하는 명령어가 import임

```
import numpy as np
```

→ 넘파이(numpy)라는 라이브러리를 np라는 이름으로 불러오라는 뜻

5 | 딥러닝의 개괄 잡기

- 넘파이는 수치 계산을 위해 만들어진 라이브러리로 데이터 분석에 많이 사용됨

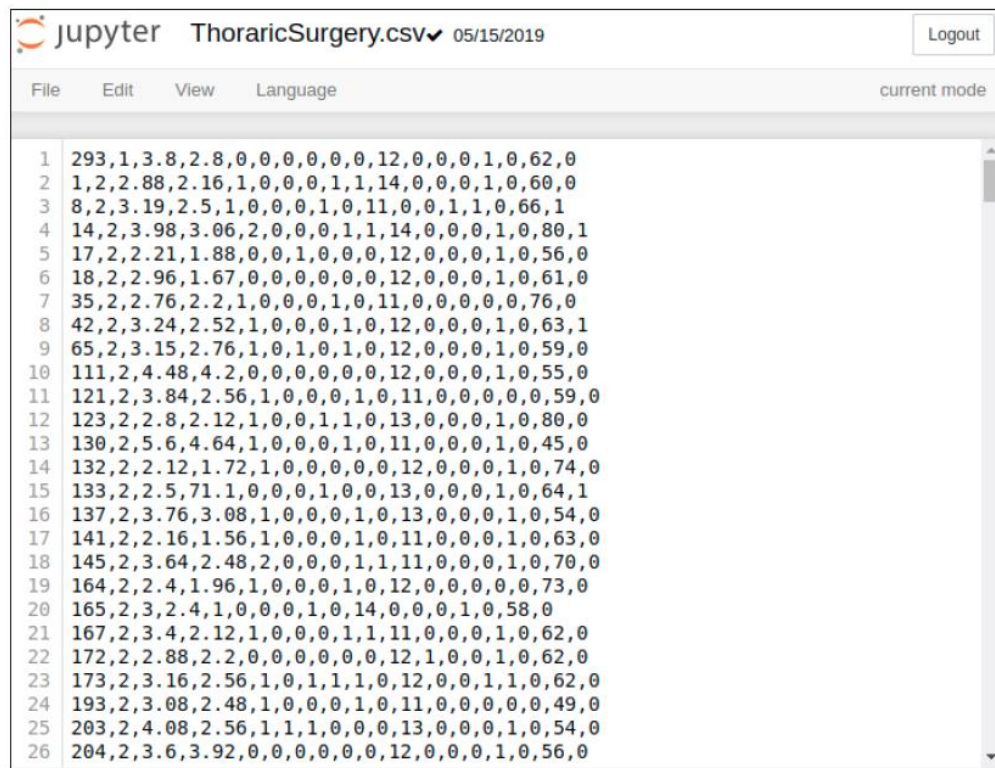
```
Data_set = numpy.loadtxt("../dataset/ThoracicSurgery.csv",  
delimiter=",")
```

5 | 딥러닝의 개괄 잡기

- 머신러닝에서 알고리즘이나 좋은 컴퓨터 환경만큼 중요한 것이 바로 제대로 된 데이터를 준비하는 일
- 매번 데이터를 정밀히 관찰하고 효율적으로 다루는 연습을 하는 것이 중요함

5 | 딥러닝의 개괄 잡기

- 이 파일은 폴란드의 브로츠와프 의과대학에서 2013년 공개한 폐암 수술 환자의 수술 전 진단 데이터와 수술 후 생존 결과를 기록한 실제 의료 기록 데이터임

A screenshot of a Jupyter Notebook interface. The top bar shows the Jupyter logo, the filename 'ThoracicSurgery.csv', and the date '05/15/2019'. There is a 'Logout' button on the right. Below the top bar is a menu bar with 'File', 'Edit', 'View', and 'Language'. The main area displays a list of 26 rows of data from the 'ThoracicSurgery.csv' file. Each row contains a line number followed by a comma-separated list of numerical values. The values are mostly integers, with some decimal points. The data appears to be a combination of categorical and numerical features.

```
jupyter ThoracicSurgery.csv 05/15/2019 Logout
File Edit View Language current mode

1 293,1,3.8,2.8,0,0,0,0,0,0,12,0,0,0,1,0,62,0
2 1,2,2.88,2.16,1,0,0,0,1,1,14,0,0,0,1,0,60,0
3 8,2,3.19,2.5,1,0,0,0,1,0,11,0,0,1,1,0,66,1
4 14,2,3.98,3.06,2,0,0,0,1,1,14,0,0,0,1,0,80,1
5 17,2,2.21,1.88,0,0,1,0,0,0,12,0,0,0,1,0,56,0
6 18,2,2.96,1.67,0,0,0,0,0,0,12,0,0,0,1,0,61,0
7 35,2,2.76,2.2,1,0,0,0,1,0,11,0,0,0,0,0,76,0
8 42,2,3.24,2.52,1,0,0,0,1,0,12,0,0,0,1,0,63,1
9 65,2,3.15,2.76,1,0,1,0,1,0,12,0,0,0,1,0,59,0
10 111,2,4.48,4.2,0,0,0,0,0,0,12,0,0,0,1,0,55,0
11 121,2,3.84,2.56,1,0,0,0,1,0,11,0,0,0,0,0,59,0
12 123,2,2.8,2.12,1,0,0,1,1,0,13,0,0,0,1,0,80,0
13 130,2,5.6,4.64,1,0,0,0,1,0,11,0,0,0,1,0,45,0
14 132,2,2.12,1.72,1,0,0,0,0,0,12,0,0,0,1,0,74,0
15 133,2,2.5,71.1,0,0,0,1,0,0,13,0,0,0,1,0,64,1
16 137,2,3.76,3.08,1,0,0,0,1,0,13,0,0,0,1,0,54,0
17 141,2,2.16,1.56,1,0,0,0,1,0,11,0,0,0,1,0,63,0
18 145,2,3.64,2.48,2,0,0,0,1,1,11,0,0,0,1,0,70,0
19 164,2,2.4,1.96,1,0,0,0,1,0,12,0,0,0,0,0,73,0
20 165,2,3,2.4,1,0,0,0,1,0,14,0,0,0,1,0,58,0
21 167,2,3.4,2.12,1,0,0,0,1,1,11,0,0,0,1,0,62,0
22 172,2,2.88,2.2,0,0,0,0,0,0,12,1,0,0,1,0,62,0
23 173,2,3.16,2.56,1,0,1,1,1,0,12,0,0,1,1,0,62,0
24 193,2,3.08,2.48,1,0,0,0,1,0,11,0,0,0,0,0,49,0
25 203,2,4.08,2.56,1,1,1,0,0,0,13,0,0,0,1,0,54,0
26 204,2,3.6,3.92,0,0,0,0,0,0,12,0,0,0,1,0,56,0
```

그림 1-17 ThoracicSurgery.csv 파일 확인

5 | 딥러닝의 개괄 잡기

줄 항목	속성																	클래스
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	293	1	3.8	2.8	0	0	0	0	0	0	12	0	0	0	1	0	62	0
2	1	2	2.88	2.16	1	0	0	0	1	1	14	0	0	0	1	0	60	0
3	8	2	3.19	2.5	1	0	0	0	1	0	11	0	0	1	1	0	66	1
...
470	447	8	5.2	4.1	0	0	0	0	0	0	12	0	0	0	0	0	49	0

표 1-1 폐암 수술 환자의 의료 기록 데이터

5 | 딥러닝의 개괄 잡기

- 1번째 항목부터 17번째 항목까지를 '속성(attribute)'이라고 함
- 정답에 해당하는 18번째 항목을 '클래스(class)'라고 함
- 딥러닝을 구동시키려면 '속성'만을 뽑아 데이터셋을 만들고, '클래스'를 담은 데이터셋을 또 따로 만들어 줘야 함

5 | 딥러닝의 개괄 잡기

- 속성 데이터셋 X를 다음과 같이 생성함

```
X = Data_set[:,0:17]
```

- 클래스 데이터셋 Y는 18번째 항목을 이용해 다음과 같이 만들어 줌

```
Y = Data_set[:,17]
```

5 | 딥러닝의 개괄 잡기

두 번째 부분: 딥러닝 실행

- 이제 딥러닝을 실제로 실행하는 부분임

딥러닝을 구동하는 데 필요한 케라스 함수 호출

```
from tensorflow.keras.models import Sequential
```

```
from tensorflow.keras.layers import Dense
```

(중략)

5 | 딥러닝의 개괄 잡기

모델을 설정하고 실행하는 부분, 딥러닝 구조를 결정

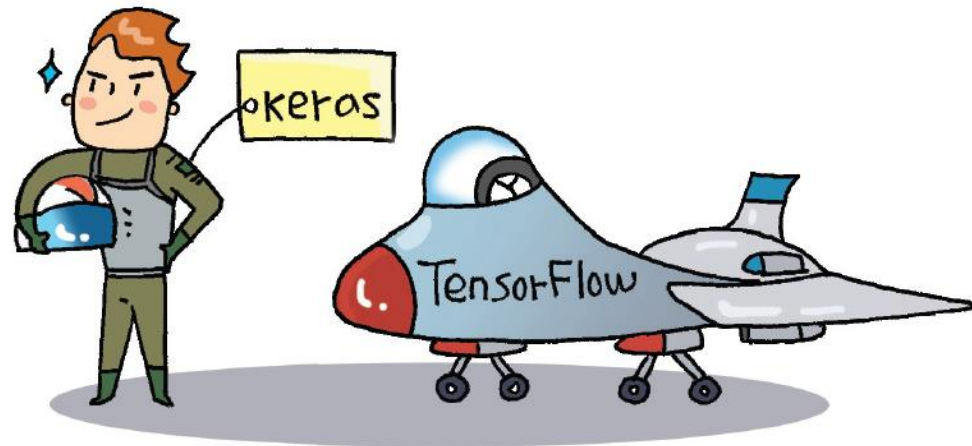
```
model = Sequential()  
model.add(Dense(30, input_dim=17, activation='relu'))  
model.add(Dense(1, activation='sigmoid'))
```

딥러닝 실행

```
model.compile(loss='mean_squared_error', optimizer='adam',  
metrics=['accuracy'])  
model.fit(X, Y, epochs=100, batch_size=10)
```


5 | 딥러닝의 개괄 잡기

- 주로 케라스(keras)를 사용해 딥러닝을 실행시킴
- 텐서플로(tensorFlow)가 미리 설치되어 있어야 함
- 딥러닝 프로젝트를 '여행'으로 비유해 본다면 텐서플로는 목적지까지 빠르게 이동시켜주는 '비행기'에 해당함
- 케라스는 비행기의 이륙 및 정확한 지점까지의 도착을 책임지는 '파일럿'에 비유할 수 있음



5 | 딥러닝의 개괄 잡기

- 설치가 모두 올바르게 되었다면 다음과 같은 방법으로 설치된 케라스 라이브러리를 불러올 수 있음

```
from tensorflow.keras.models import Sequential  
from tensorflow.keras.layers import Dense
```

5 | 딥러닝의 개괄 잡기

- Sequential() 함수는 딥러닝의 구조를 한 층 한 층 쉽게 쌓아 올릴 수 있게 해 줌
- Sequential() 함수를 선언하고 나서 model.add() 함수를 사용해 필요한 층을 차례로 추가하면 됨

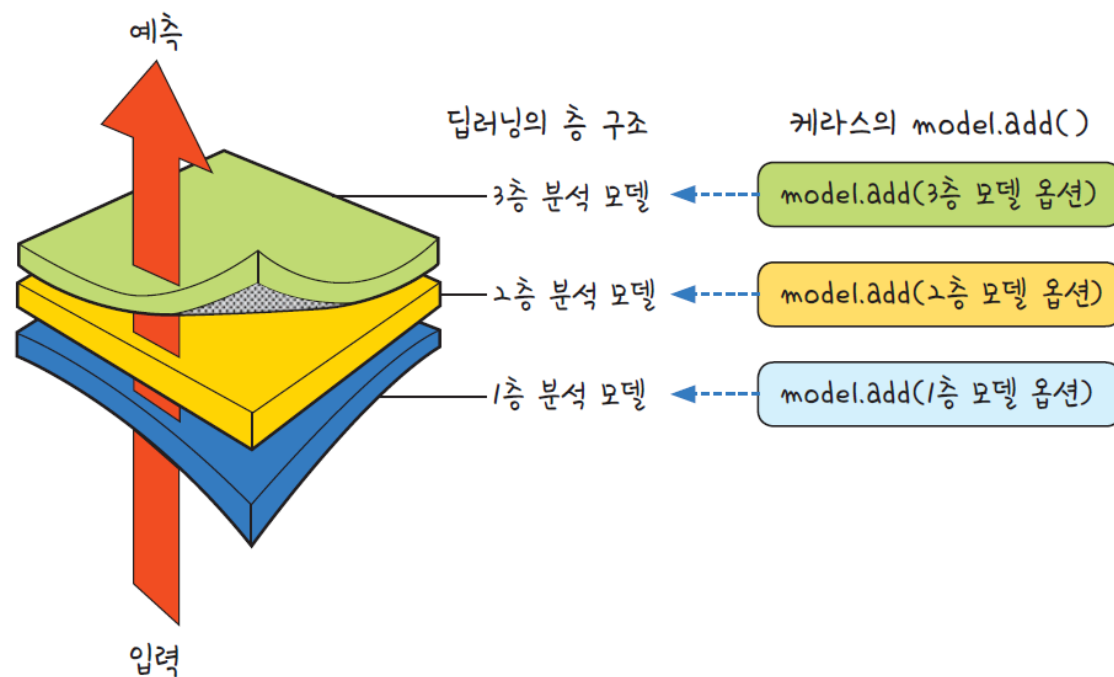


그림 1-19 딥러닝의 층 구조와 케라스

5 | 딥러닝의 개괄 잡기

- 우리가 살펴보고 있는 01_My_First_DeepLearning.ipynb에서는 `model.add()` 함수를 이용해 두 개의 층을 쌓아 올림

```
model = Sequential()  
model.add(Dense(30, input_dim=17, activation='relu'))  
model.add(Dense(1, activation='sigmoid'))
```

- 층을 몇 개 쌓을지는 데이터에 따라 그때 그때 결정함
- 케라스의 가장 큰 장점 중 하나는 `model.add()` 함수를 이용해 필요한 만큼의 층을 빠르고 쉽게 쌓아 올릴 수 있다는 것임

5 | 딥러닝의 개괄 잡기

- `model.add()` 함수안에는 `Dense()` 함수가 포함되어 있음
- `dense`는 '조밀하게 모여있는 집합'이란 뜻으로, 여기서는 각 층이 제각각 어떤 특성을 가질지 옵션을 설정하는 역할을 함
- 딥러닝의 구조와 층별 옵션을 정하고 나면 `compile()` 함수를 이용해 이를 실행 시킴

```
model.compile(loss='mean_squared_error', optimizer='adam', metrics
=['accuracy'])
model.fit(X, Y, epochs=30, batch_size=10)
```

5 | 딥러닝의 개괄 잡기

'블랙박스'를 극복하려면?

- 딥러닝이 어떤 원리로 작동되고, 딥러닝 내부에서 어떤 방식으로 결과를 도출하는지를 알지 못하면 딥러닝은 속 안을 들여다 볼 수 없는 캄캄한 '블랙박스'가 되어 버림



5 | 딥러닝의 개괄 잡기

'블랙박스'를 극복하려면?

- 이것이 바로 딥러닝이라는 '블랙박스'를 열어 그 안에서 구동되는 여러 가지 원리를 공부해야 하는 이유임

정리 학습 첫 딥러닝

- 1 | 딥러닝 실행을 위한 준비 사항
- 2 | 딥러닝 작업 환경 만들기
- 3 | 미지의 일을 예측하는 원리
- 4 | 폐암 수술 환자의 생존율 예측하기
- 5 | 딥러닝 개괄 잡기