



1장 나의 첫 딥러닝

목차

- 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 나의 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)라

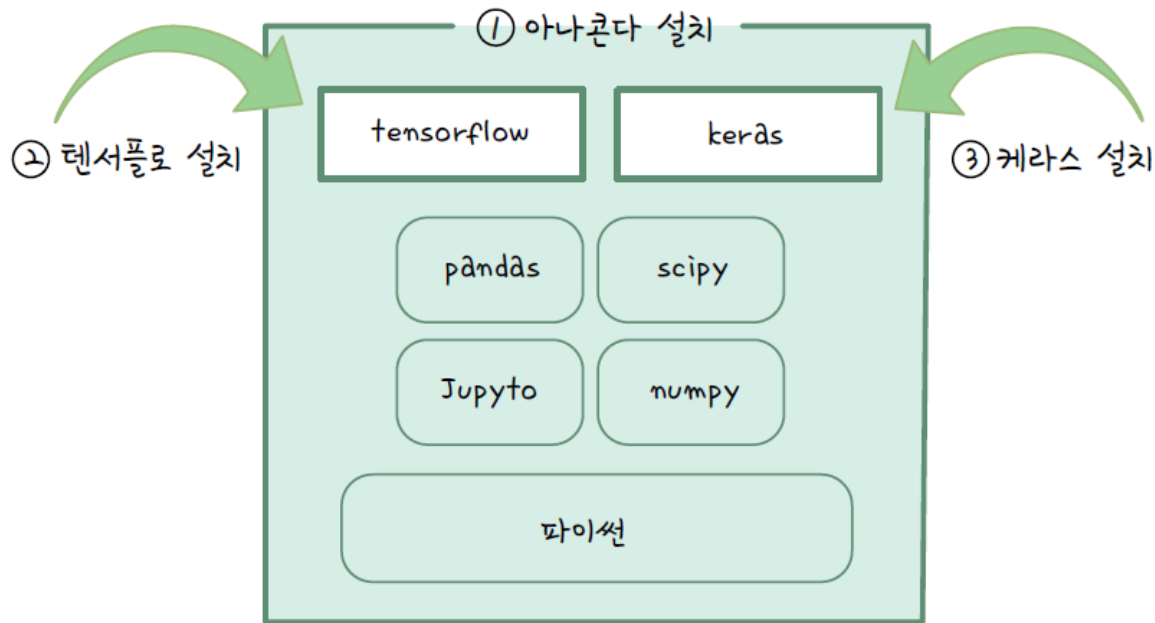
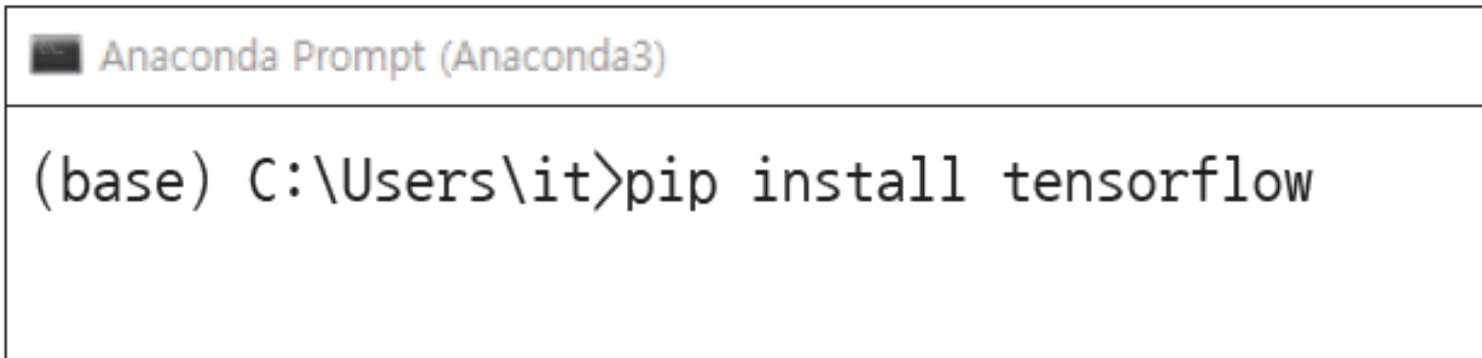


그림 1-2 내 컴퓨터에서 딥러닝을 구동하기 위한 세 가지 설치 작

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 current directory as "C:\Users\it" and the command "pip install tensorflow" has been entered.

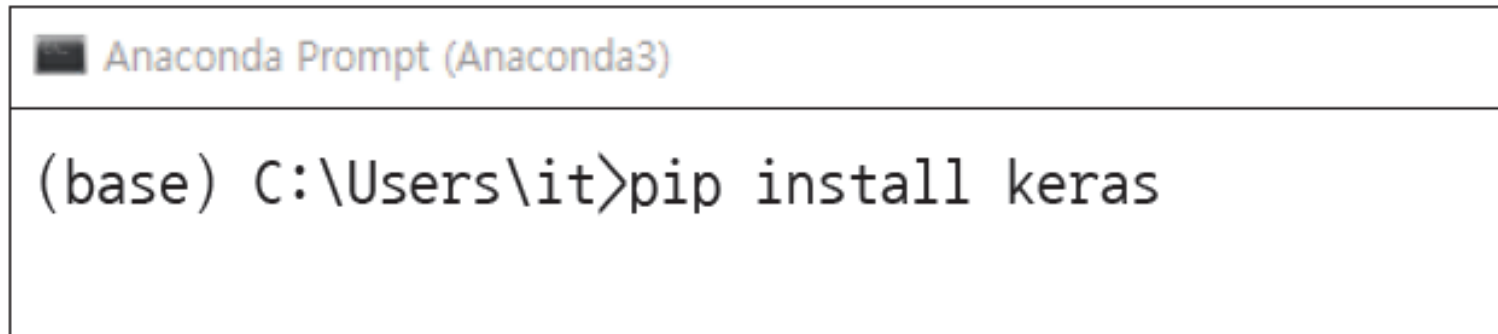
```
Anaconda Prompt (Anaconda3)  
(base) C:\Users\it>pip install tensorflow
```

그림 1-8 텐서플로 설치

2 딥러닝 작업 환경 만들기

케라스 설치하기

- 다음과 같이 pip install keras를 입력하여 케라스 설치를 진행함

A screenshot of the Anaconda Prompt (Anaconda3) window. The title bar reads "Anaconda Prompt (Anaconda3)". The command prompt shows the current environment as "(base)" and the directory as "C:\Users\it". The command entered is "pip install keras".

```
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 backend

```
Anaconda Prompt (Anaconda3) - python

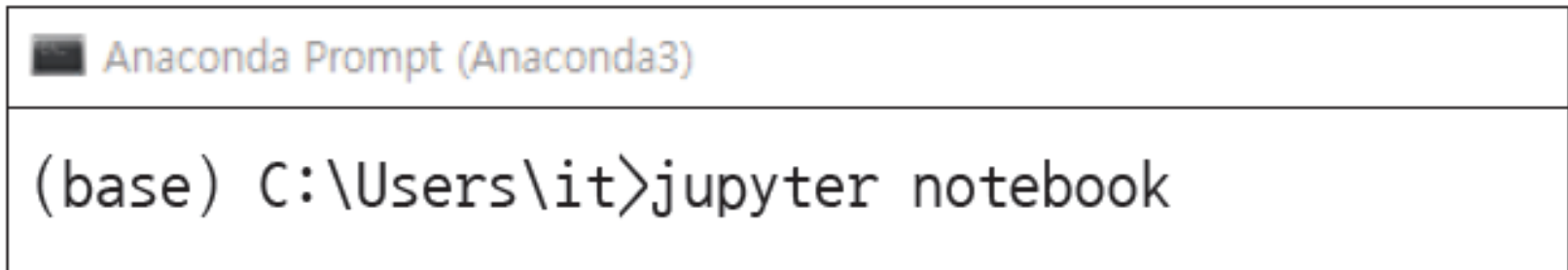
(base) C:\Users\it>python
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorflow as tf
>>> print(tf.__version__)
2.0.0
>>> import keras
Using TensorFlow backend.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

그림 1-10 텐서플로 및 케라스 설치 확인

2 딥러닝 작업 환경 만들기

딥러닝 실행하기

- 딥러닝을 만들고 실행할 때 이용하는 대표적인 툴은 앞서 말한 주피터 노트북과 파이참임
- 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

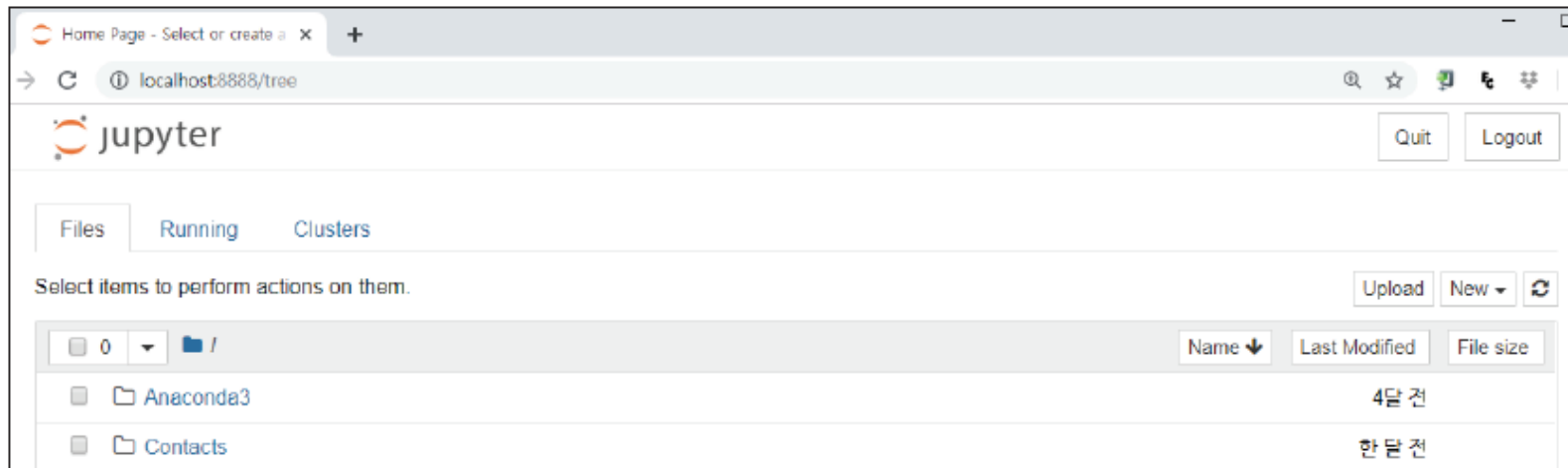


그림 1-12 주피터 노트북 화면

2 딥러닝 작업 환경 만들기

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

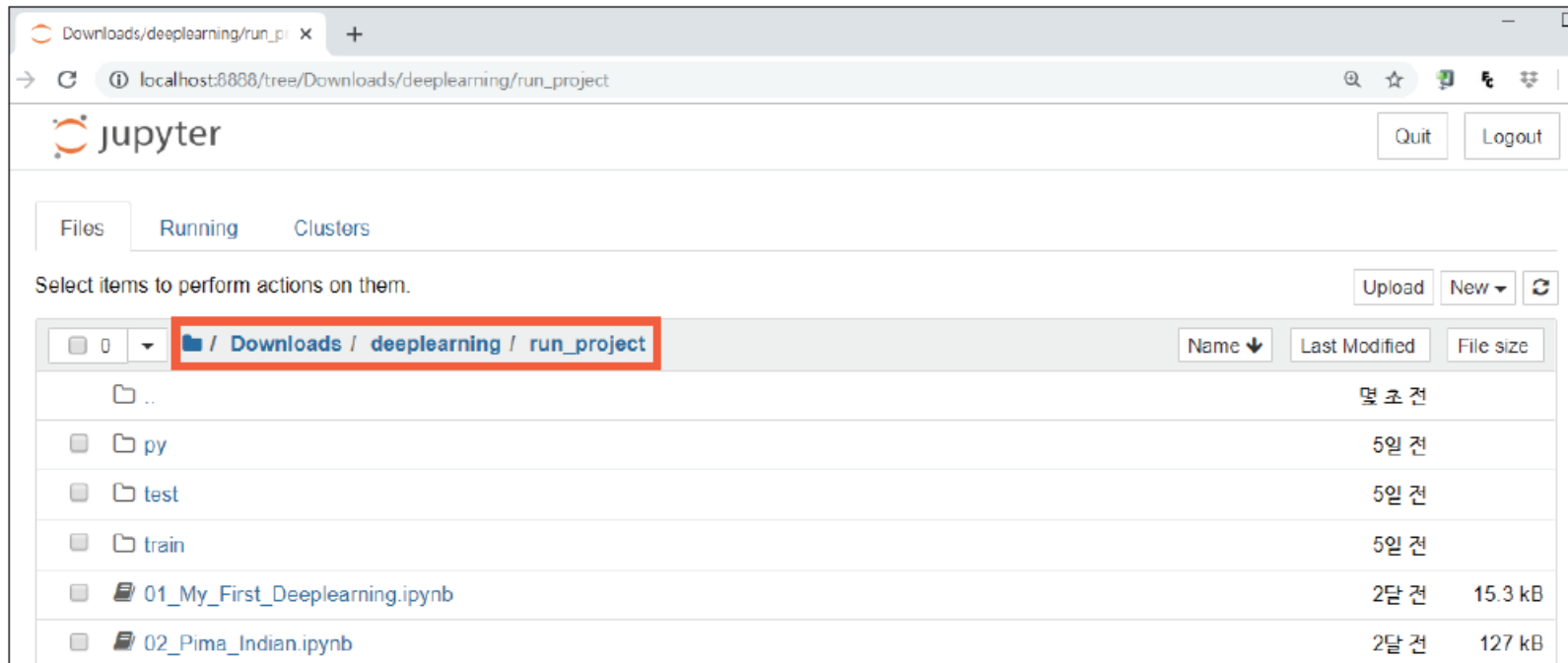


그림 1-13 예제 소스가 있는 폴더로 이동

2 딥러닝 작업 환경 만들기

- 01_My_First_DeepLearning.ipynb를 클릭하면 파일이 열림

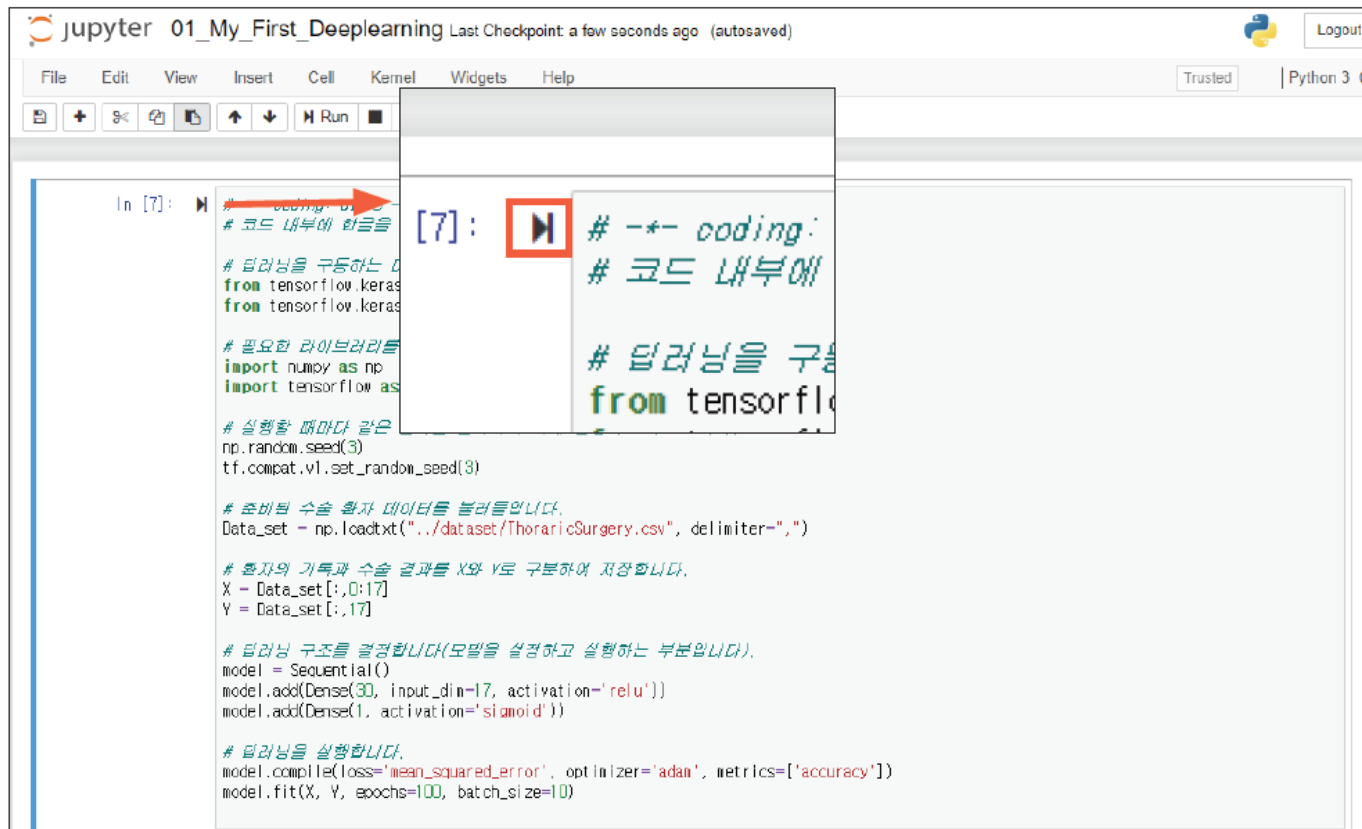
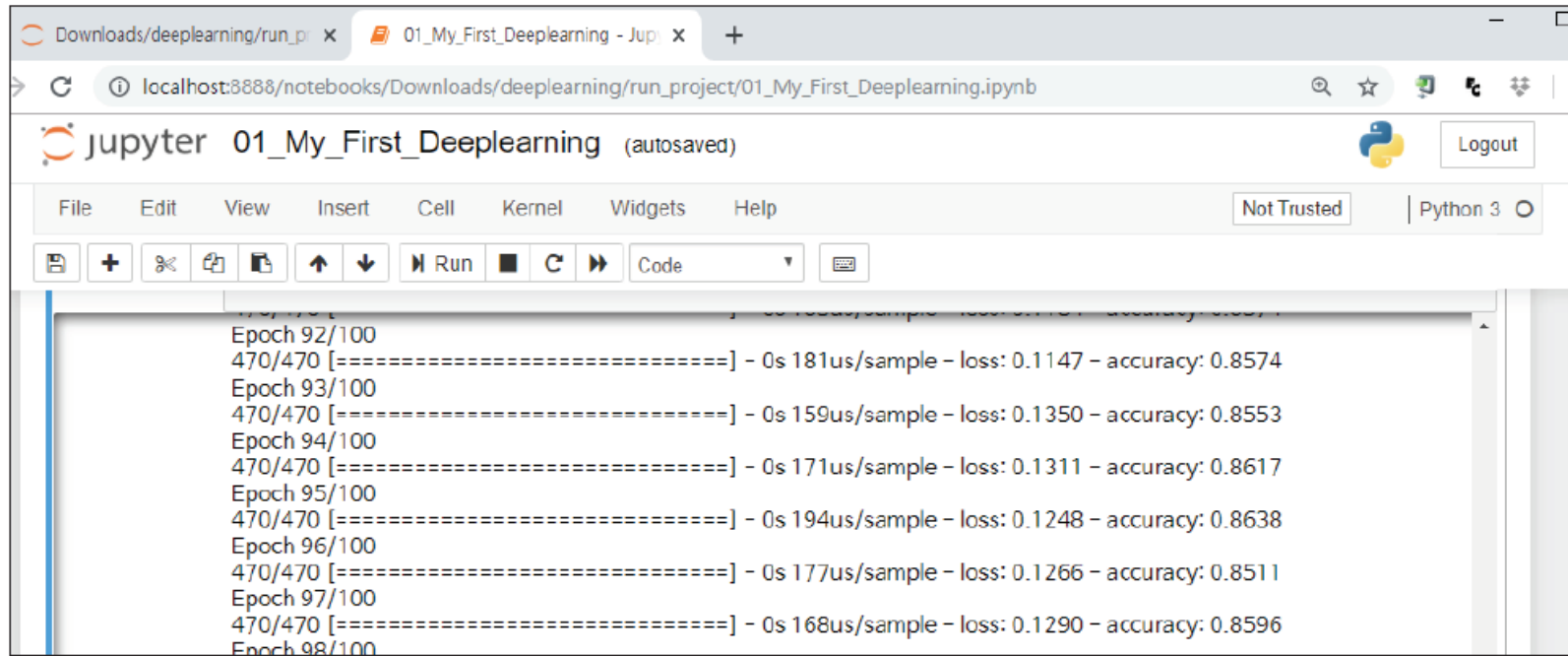


그림 1-14 주피터 노트북으로 딥러닝 실행하기

2 딥러닝 작업 환경 만들기

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



The screenshot shows a Jupyter Notebook interface in a web browser. The browser tabs include '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 has a title bar 'jupyter 01_My_First_Deeplearning (autosaved)' and a 'Logout' button. Below the title bar is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. To the right of the menu bar is a 'Not Trusted' warning and 'Python 3'. Below the menu bar is a toolbar with icons for saving, adding cells, undo, redo, and running. The main area shows the execution output of a code cell, which displays training progress for epochs 92 through 98. Each epoch shows the number of samples processed (470/470), the time per sample, the loss, and the 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

3 미지의 일을 예측하는 원리

- 머신러닝 :

기존 데이터를 이용해 아직 일어나지 않은 미지의 일을 예측하기 위해

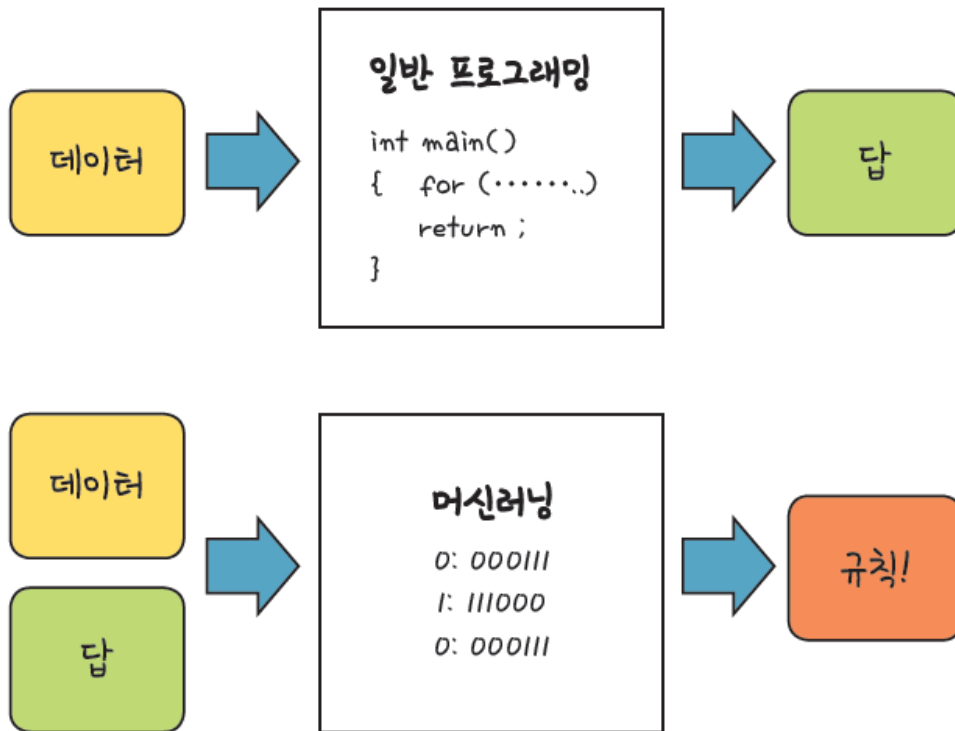


그림 1-16 머신러닝과 일반 프로그래밍 비교

3 미지의 일을 예측하는 원리

- 학습(training) :

데이터가 입력되고 패턴이 분석되는 과정

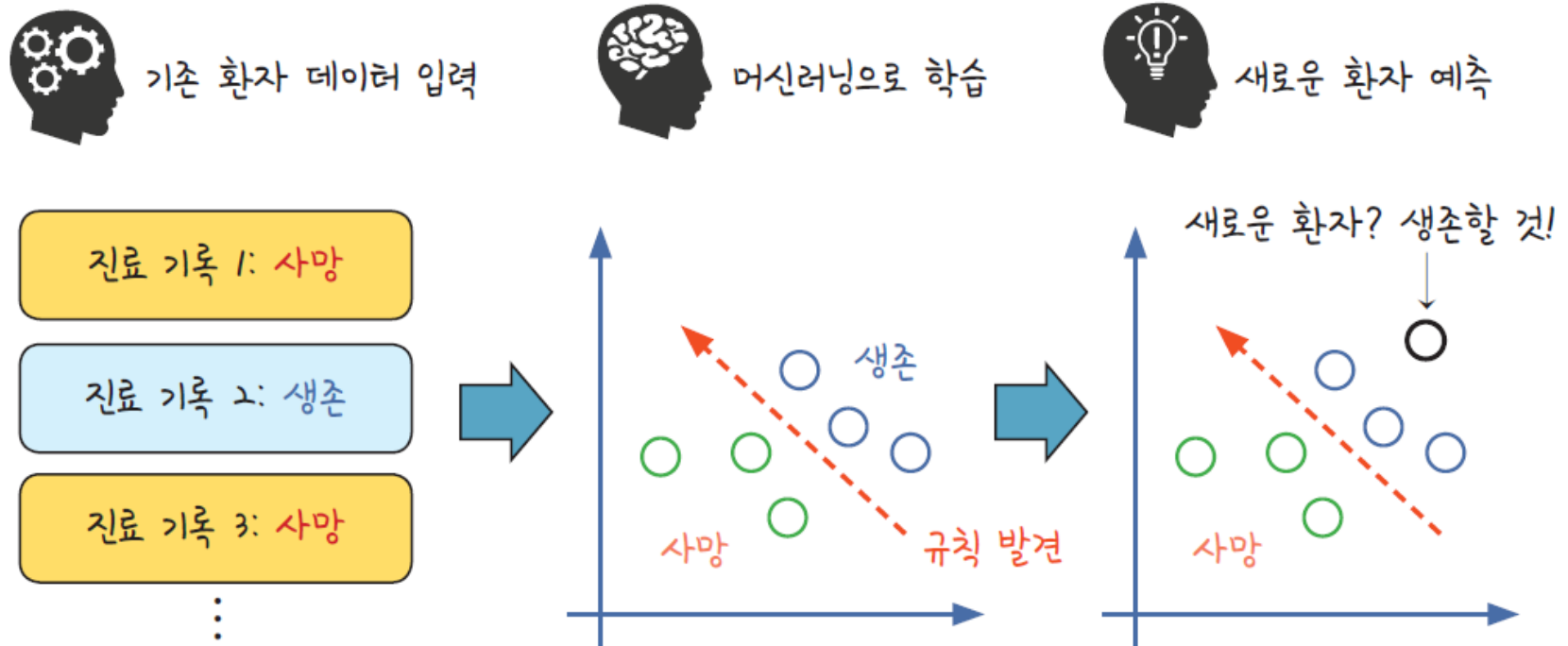


그림 1-17 머신러닝의 학습 및 예측 과정