

# TensorFlow

툴 설치 메뉴얼

## 0 | 설치할 Tool

### TOOL

- Python : script 기반 프로그래밍 언어
- Anaconda : Python 프로그래밍 환경/개발툴
- Pycharm : 전문 개발자용 Python IDE (IDLE보다 더 편하고 효율적으로 사용할 수 있는 개발 도구)

## 0 | 설치할 라이브러리

### 딥러닝을 구동시키는 라이브러리

(딥러닝을 쉽게 다루기 위해서 가장 안정적인 성능을 보이는 텐서플로와 가장 쉽게 사용하게 해주는 케라스를 사용)

- TensorFlow : 구글에서 만든 머신러닝 라이브러리
- Keras : 파이썬으로 구현된 라이브러리(케라스의 기본 엔진은 텐서플로임)

### 기타 라이브러리

- pandas : Data analysis Lib
- scikit-learn : Machine Learning Lib
- numpy : Numeric Lib
- scipy : Math, science, engineering Lib
- matplotlib : Graphic, plotting, color Lib
- spyder : anaconda scientific Lib

## 0 | 파이썬 설치하기

### 윈도우에서 파이썬을 설치하는 방법

<http://www.python.org> 홈페이지에 접속

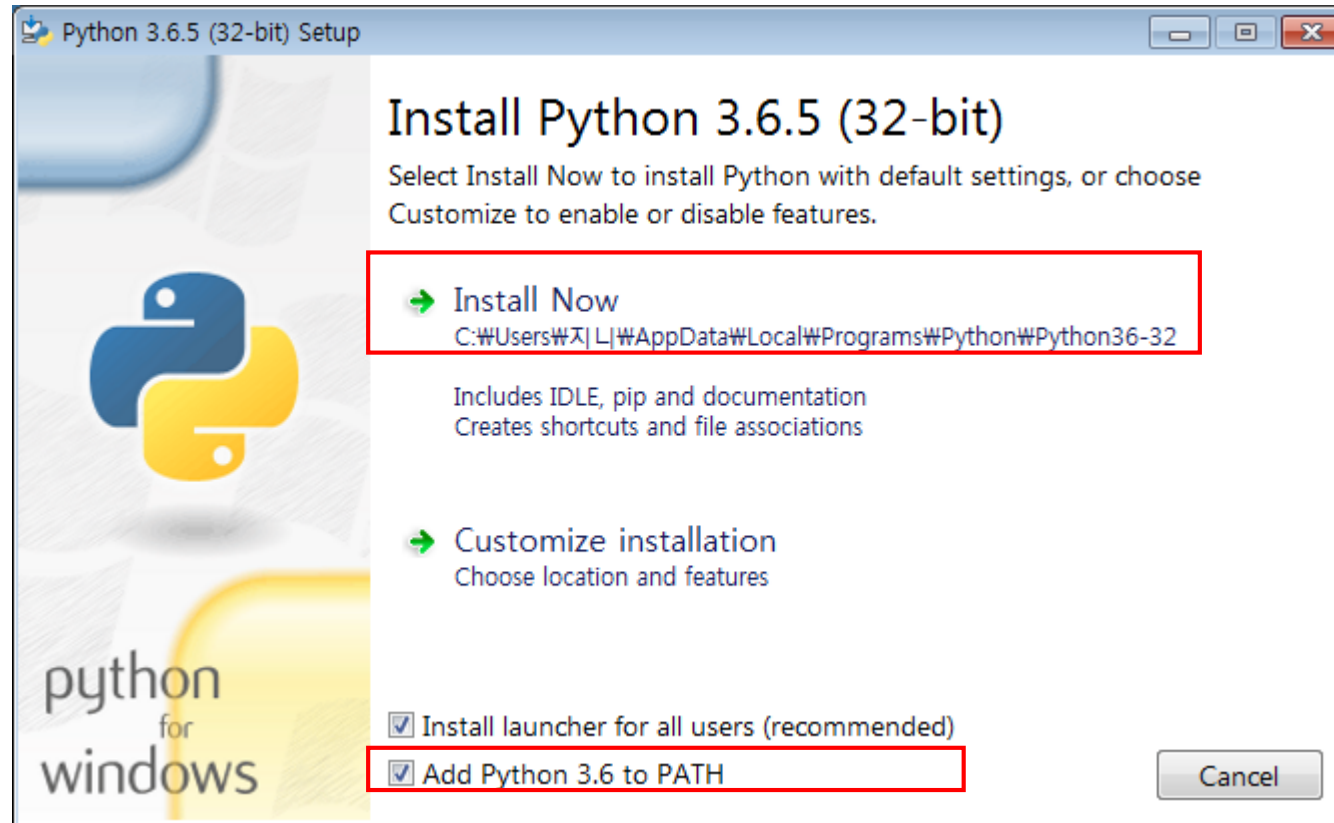
Downloads 메뉴 클릭 - 윈도우용 Python 설치 파일 내려 받기 (최신버전 파이썬 선택)



## 0 | 파이썬 설치하기

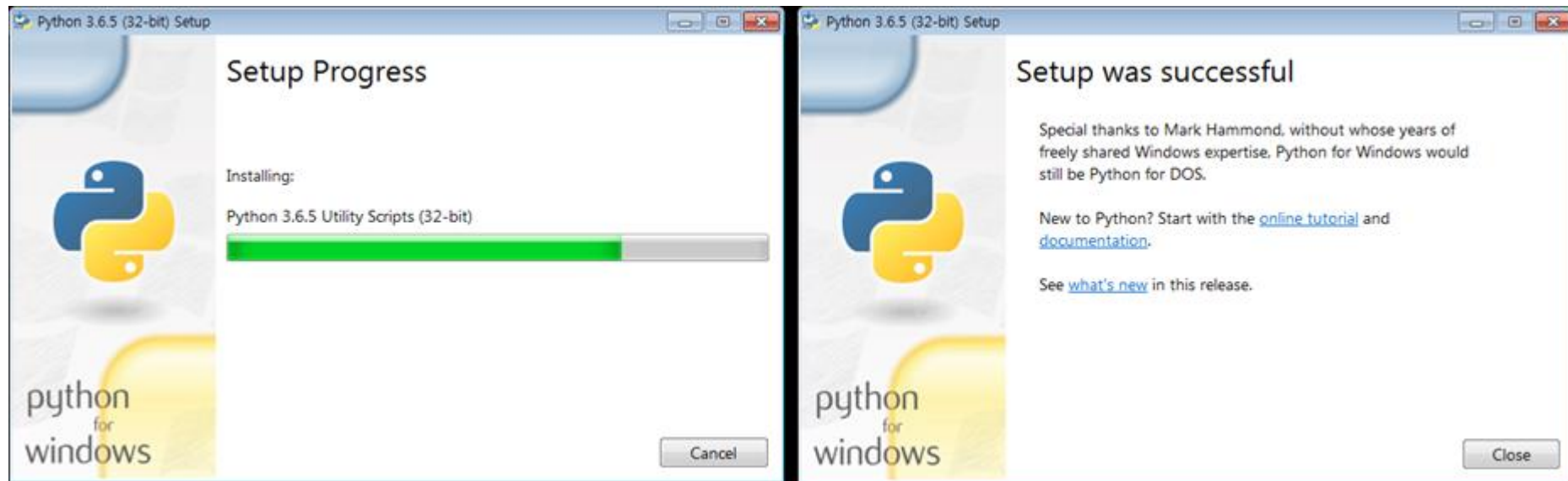
내려 받은 실행파일을 설치한다.

파이썬이 어느곳에서든 실행될 수 있도록 Add Python 3.6 to PATH 옵션을 반드시 선택해야 한다.



## 0 | 파이썬 설치하기

설치가 완료되면 Close를 클릭하여 종료한다.

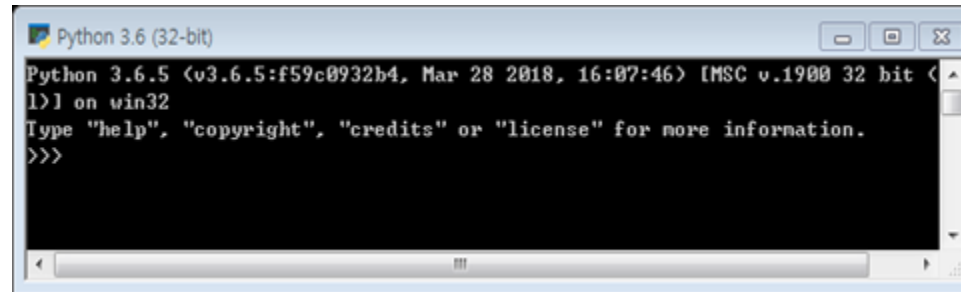


## 0 | 파이썬 설치하기

[시작]메뉴에서 [모든 프로그램-> Python3.X ->Python3.X(XX-bit)]를 선택한다.

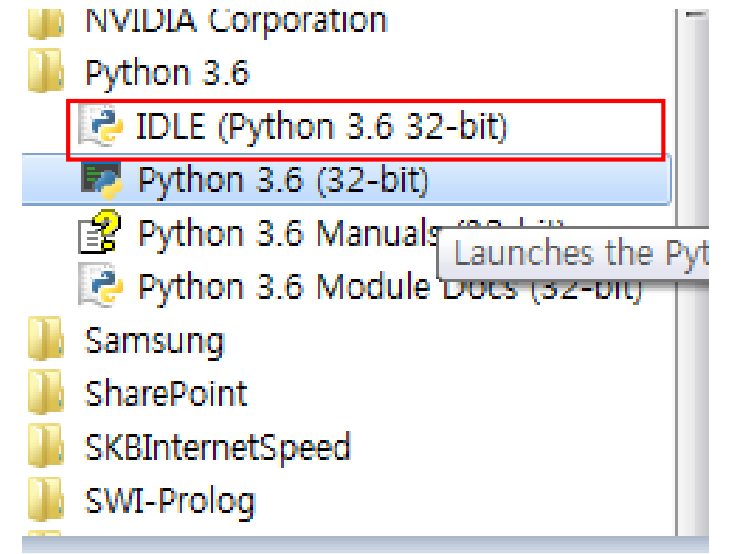
대화형 인터프리터(파이썬 셸)

>>>



종료

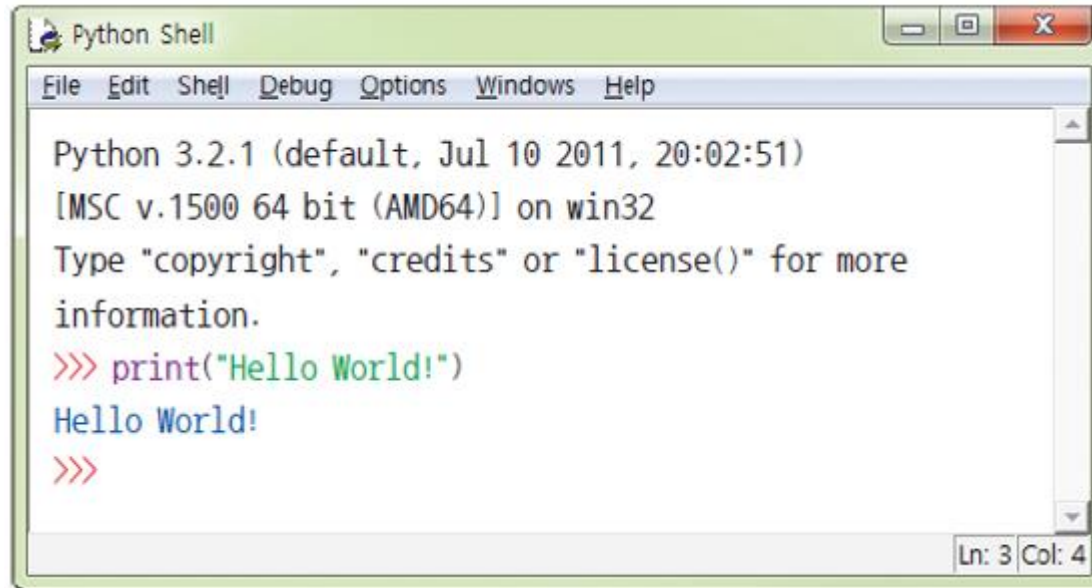
Ctrl + Z -> Enter를 누른다.



## 0 | 파이썬 실행하기

### 파이썬으로 출력하는 Hello World!

IDLE 실행 후 코드를 프롬프트에 입력한다.



```
Python Shell
File Edit Shell Debug Options Windows Help
Python 3.2.1 (default, Jul 10 2011, 20:02:51)
[MSC v.1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more
information.
>>> print("Hello World!")
Hello World!
>>>
```

Ln: 3 Col: 4



## 1 | 딥러닝 실행을 위한 준비 사항

### 내 컴퓨터의 시스템 정보 확인하기

- 텐서플로는 64비트 윈도우만을 지원하므로 사용하는 PC가 64비트인지 확인
- 시작 > 검색 > 제어판 > 시스템 순서로 시스템 패널을 열어 확인

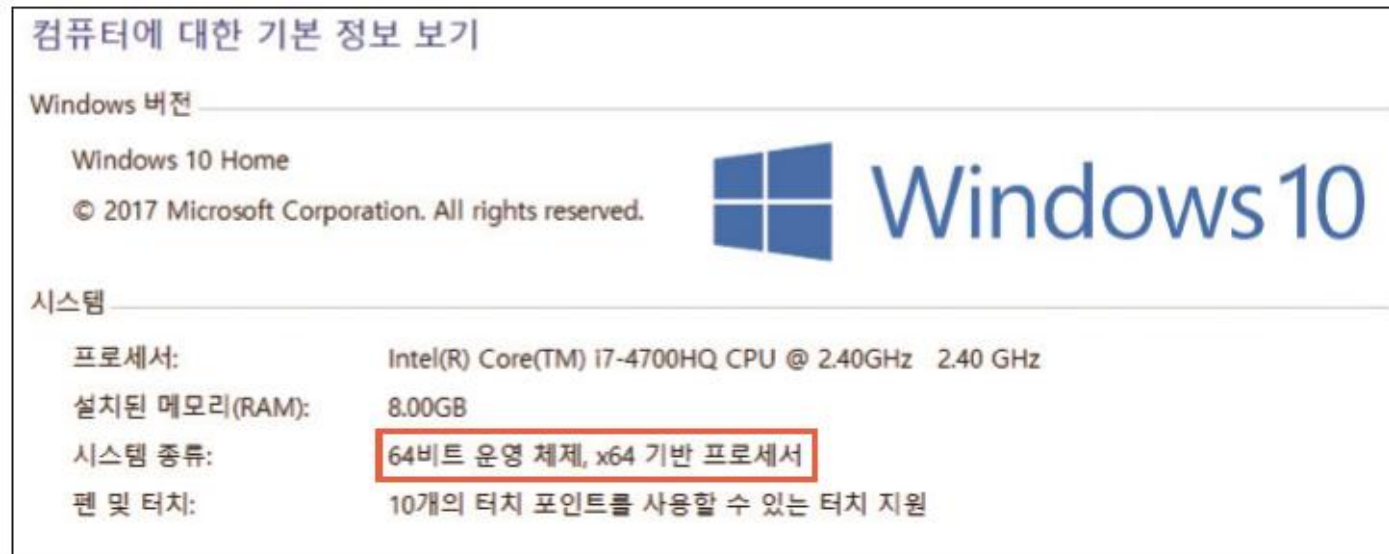


그림 1-2 내 컴퓨터의 시스템 정보 확인

## 1 | 딥러닝 실행을 위한 준비 사항

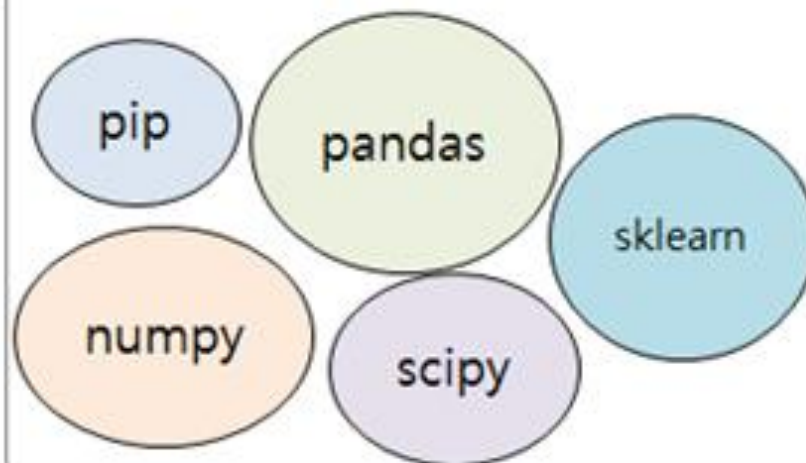
### CPU? GPU?

- 딥러닝을 일반 CPU에서 동작시킬지, 고속 그래픽 처리에 특화된 전용 프로세서인 GPU에서 동작시킬지 선택할 수 있음
- 모든 예제는 CPU와 GPU 어떤 환경에서도 잘 동작
- 딥러닝을 처음 접하는 사람은 CPU환경에서 먼저 학습해 보는 것도 좋다.
- 학습을 마친 후 대용량 데이터를 사용할 경우, GPU 작업 환경을 갖추길 추천

일반 파이썬



아나콘다



## 2 | 딥러닝 작업 환경 만들기

### 아나콘다 설치하기

- ① 아나콘다 사이트에서 아나콘다3 64비트 인스톨러를 내려받은 후 설치  
(파이썬 3.5 이상 버전을 선택)

- <https://www.continuum.io/downloads>



## 2 | 딥러닝 작업 환경 만들기

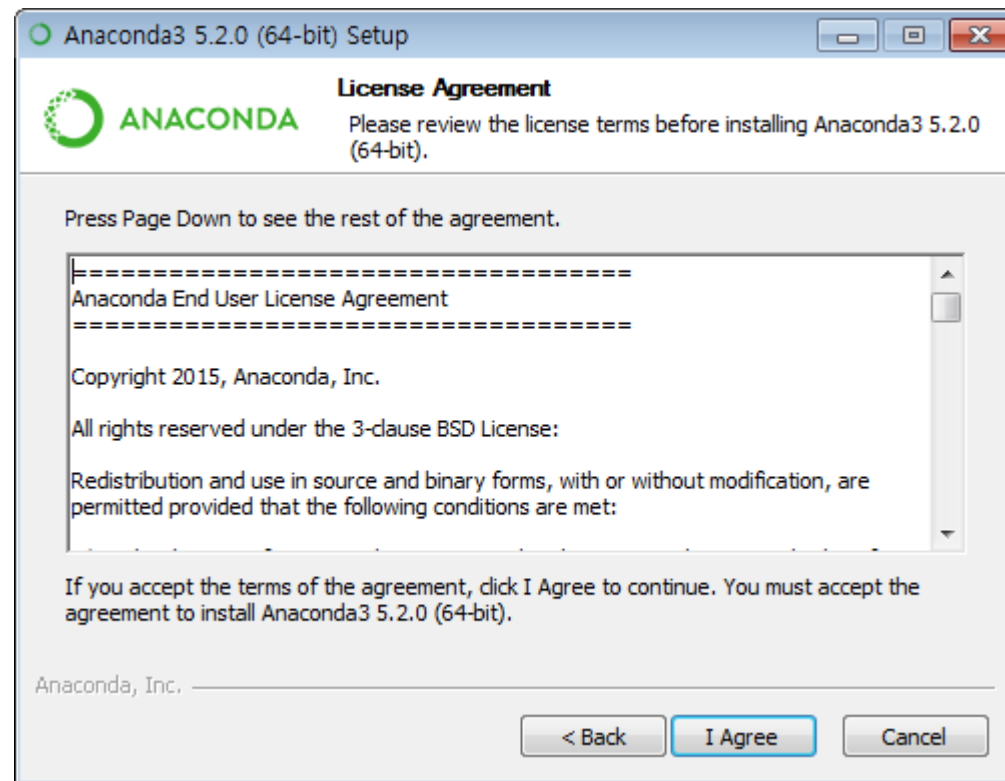
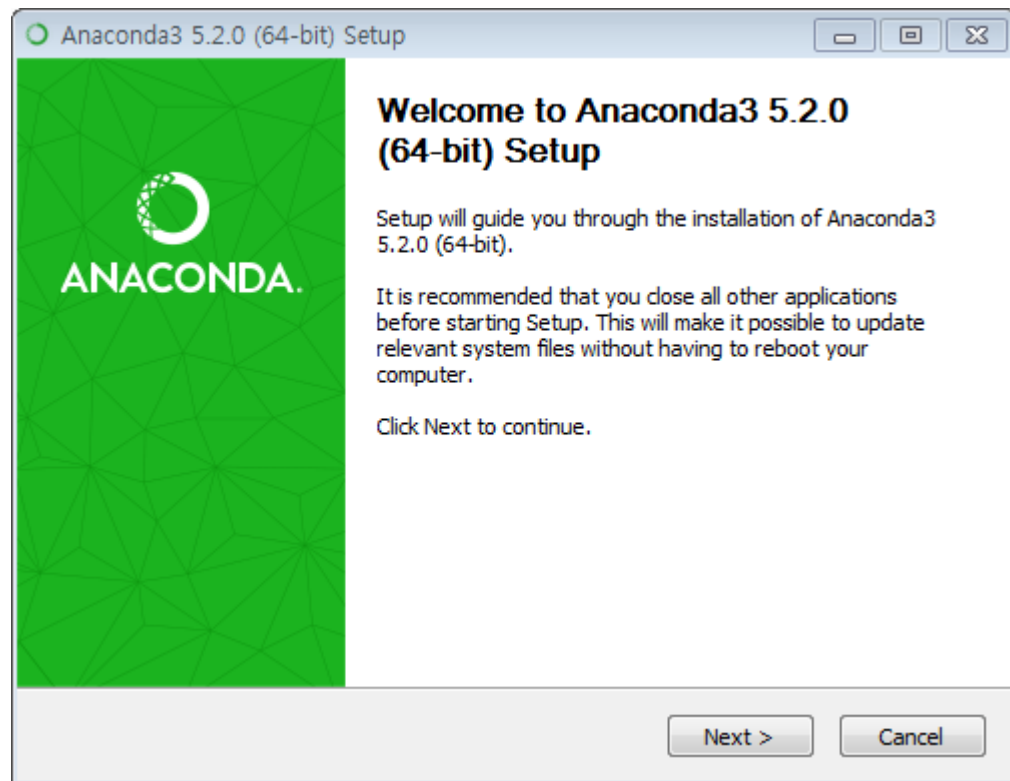


그림 1-3 아나콘다 설치

## 2 | 딥러닝 작업 환경 만들기

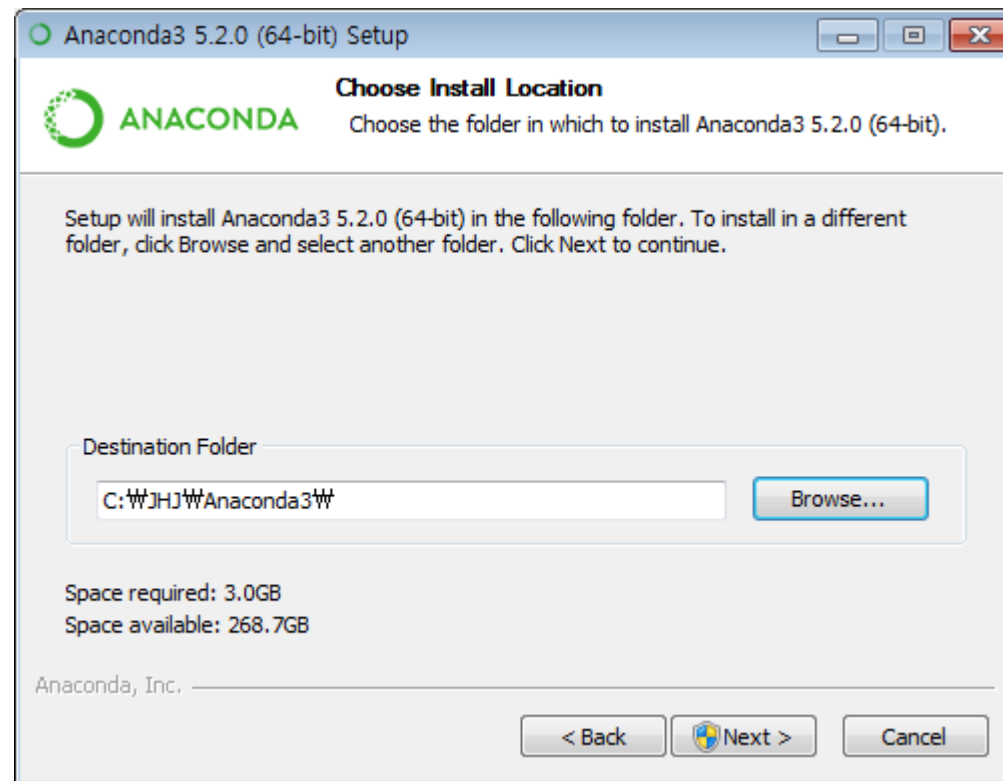
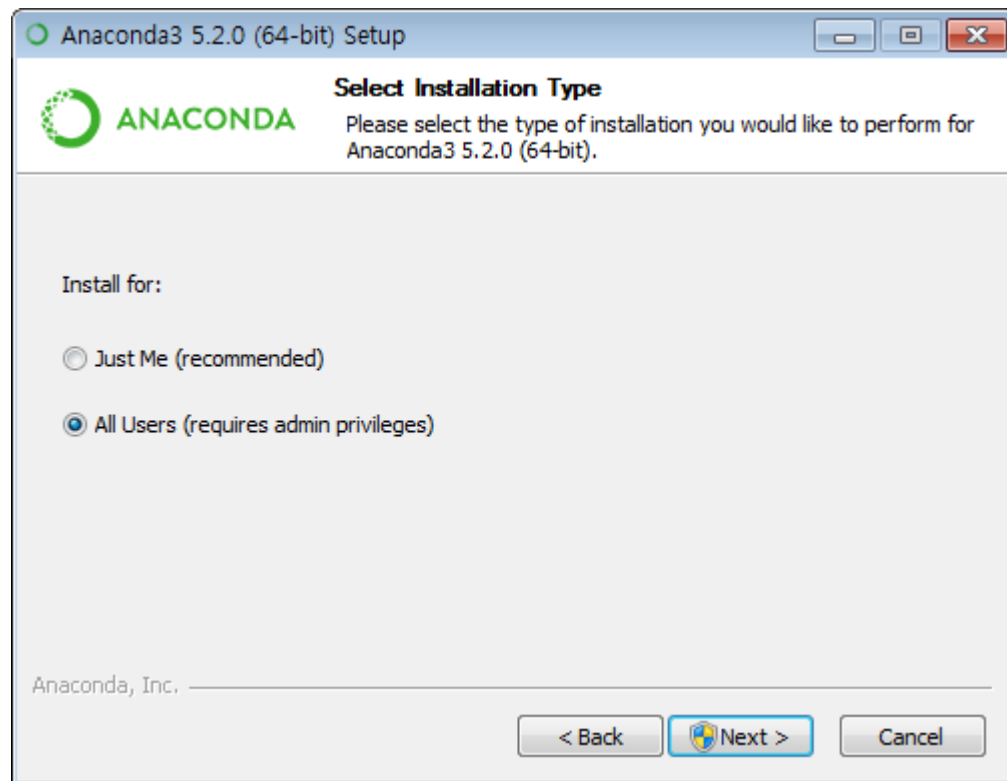


그림 1-4 아나콘다 설치

## 2 | 딥러닝 작업 환경 만들기

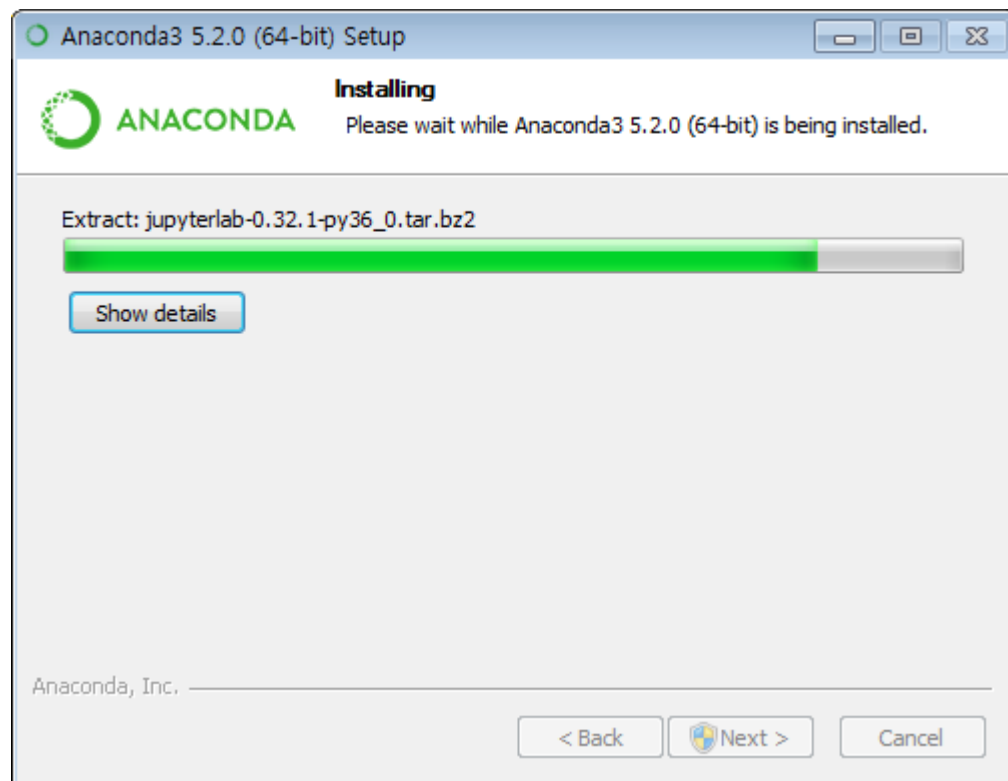
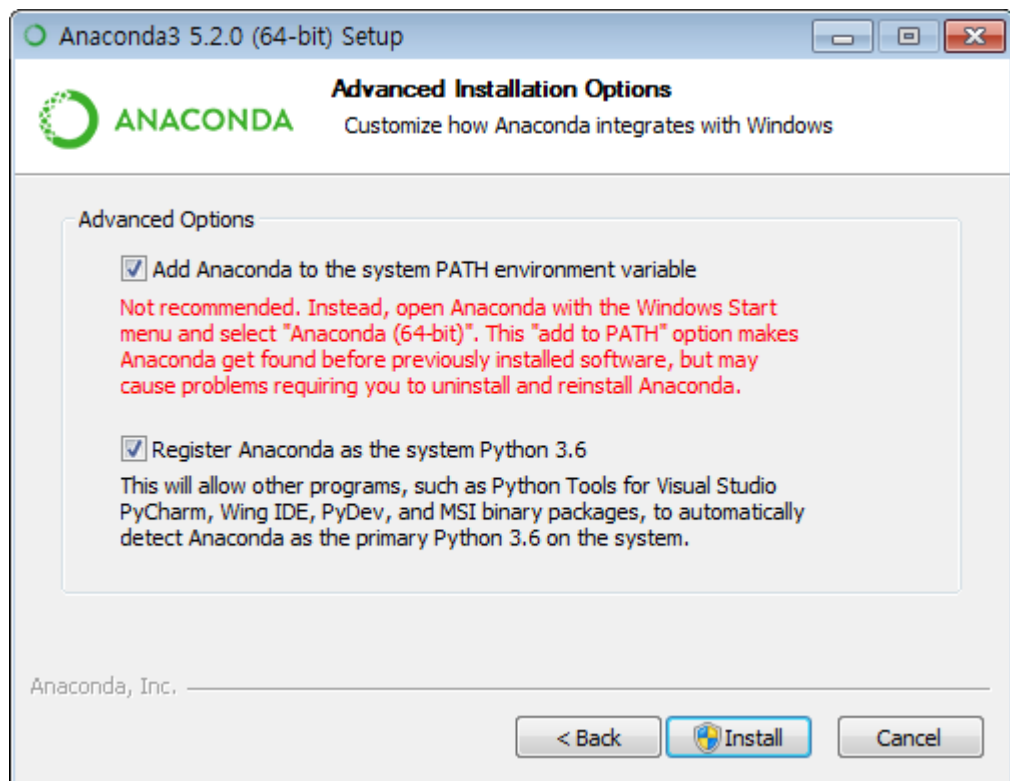


그림 1-5 아나콘다 설치

## 2 | 딥러닝 작업 환경 만들기

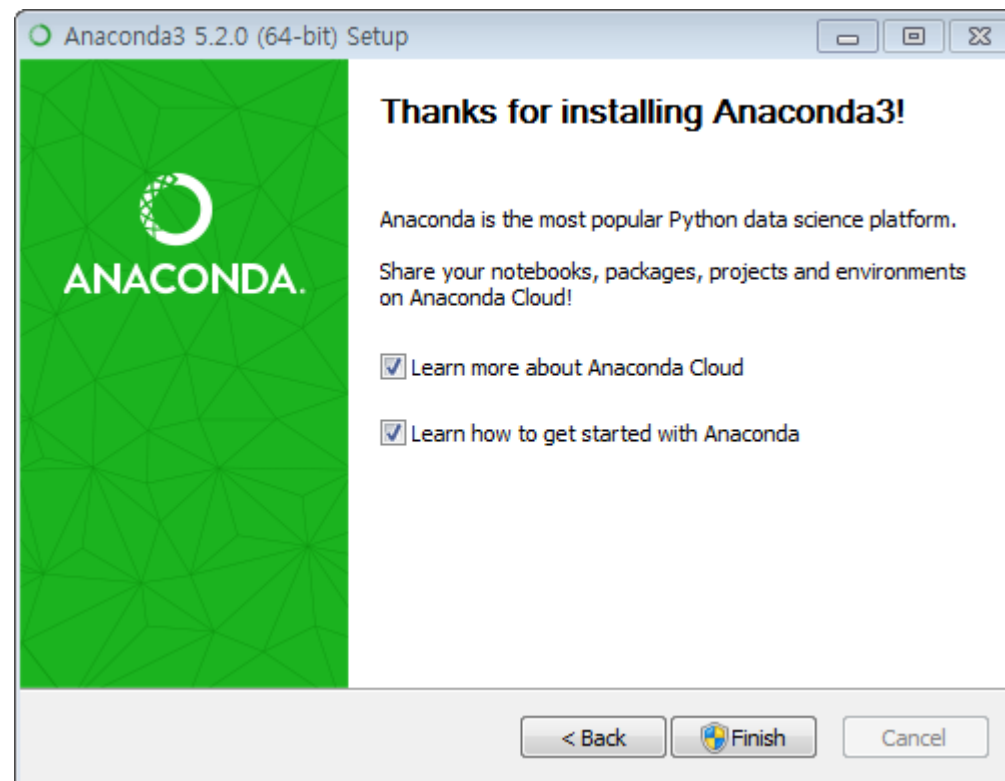


그림 1-6 아나콘다 설치



## 2 | 딥러닝 작업 환경 만들기

- ② 윈도우 버튼을 누른 후 방금 전에 설치한 아나콘다 프로그램 폴더 중 Anaconda Prompt 선택  
프로그램 폴더가 안 보이면 검색( ) 버튼을 누르고 Anaconda Prompt 입력

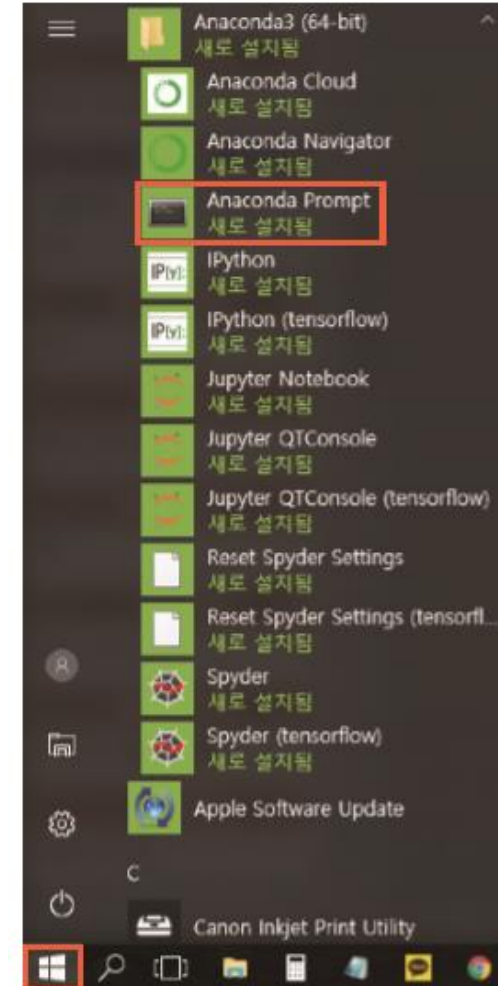


그림 1-7 Anaconda Prompt를 선택

## 2 | 딥러닝 작업 환경 만들기

- ③ 설치를 완료한 후에 Prompt창에 **conda --v** 명령어가 작동하는지 확인한다.

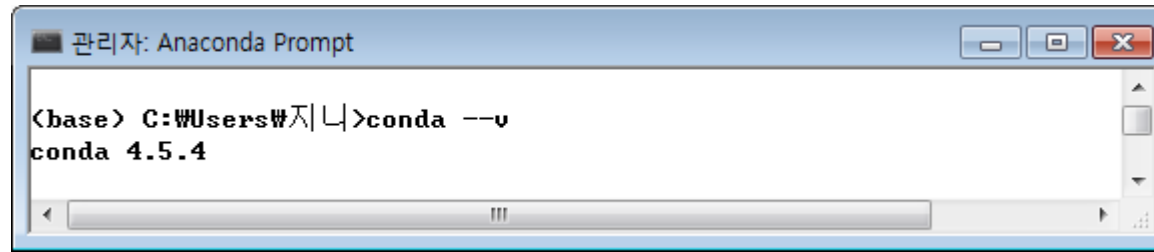


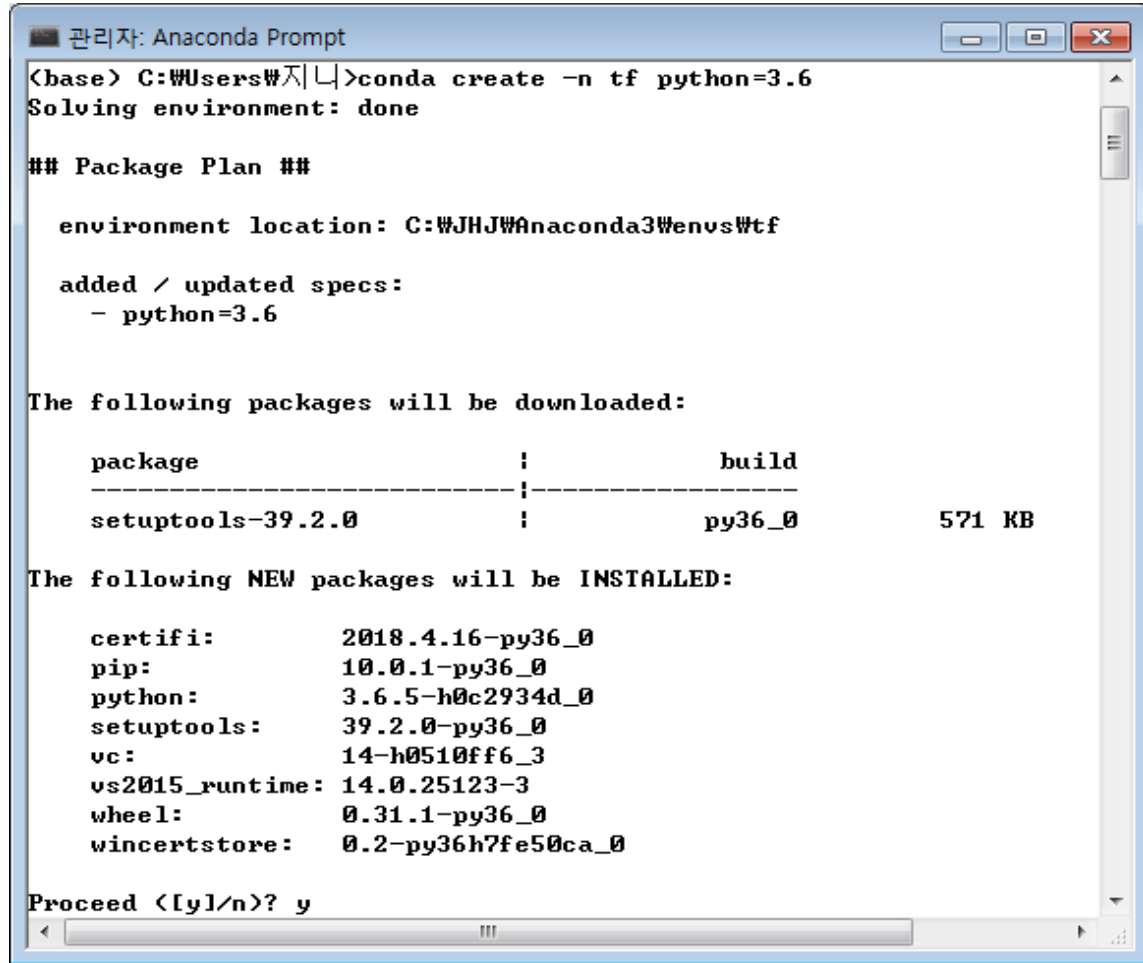
그림 1-8 아나콘다 환경 설정

- ④ conda environment를 만든다.

이 과정은 python의 virtualenv와 비슷하다. 안전하게 시스템의 python library가 꼬이지 않게 격리해서 만드는 과정이다.

**conda create -n tf python =3.6**

## 2 | 딥러닝 작업 환경 만들기



```
(base) C:\Users\WJ\>conda create -n tf python=3.6
Solving environment: done

## Package Plan ##

  environment location: C:\WJHJW\Anaconda3\envs\tf

added / updated specs:
- python=3.6

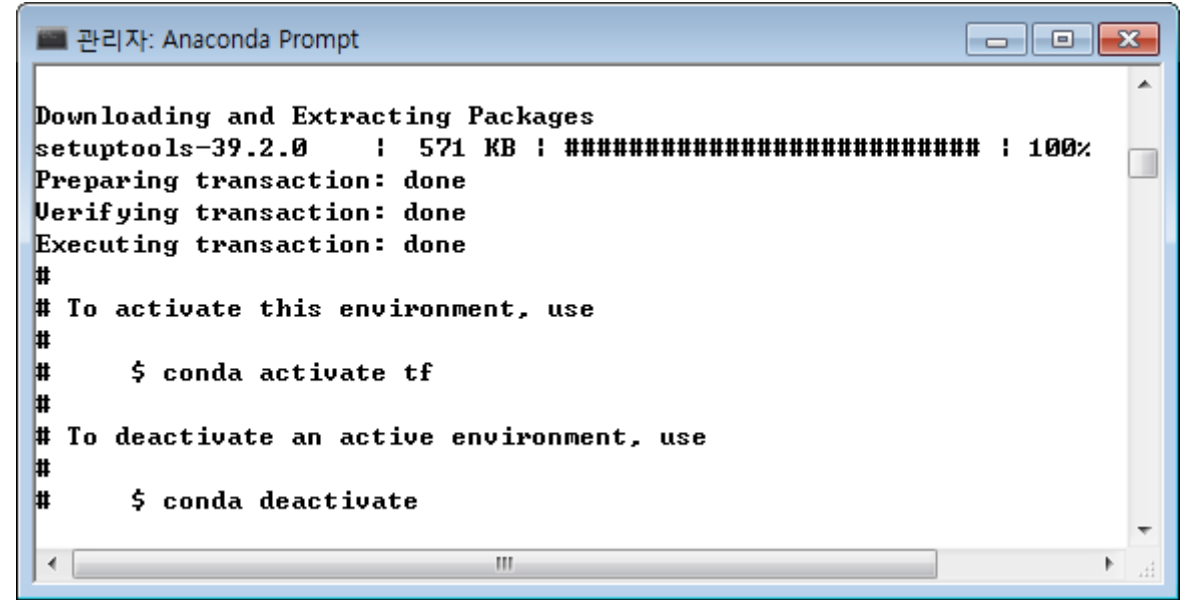
The following packages will be downloaded:

package                        | build                | 571 KB
-----|-----|
setuptools-39.2.0              | py36_0               |

The following NEW packages will be INSTALLED:

certifi: 2018.4.16-py36_0
pip: 10.0.1-py36_0
python: 3.6.5-h0c2934d_0
setuptools: 39.2.0-py36_0
vc: 14-h0510ff6_3
vs2015_runtime: 14.0.25123-3
wheel: 0.31.1-py36_0
wincertstore: 0.2-py36h7fe50ca_0

Proceed ([y]/n)? y
```



```
(base) C:\Users\WJ\>conda activate tf

Downloading and Extracting Packages
setuptools-39.2.0 | 571 KB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

#
# To activate this environment, use
#
#   $ conda activate tf
#
# To deactivate an active environment, use
#
#   $ conda deactivate
```

중간에 Proceed([y]/n)? 라는 질문이 나타나면 y를 입력하고 Enter를 누른다.  
환경 설정을 하는데 시간이 좀 걸릴 수도 있다.

그림 1-9 아나콘다 환경 설정

## 2 | 딥러닝 작업 환경 만들기

### 텐서플로 설치하기

- ① 아나콘다 명령 프롬프트에서 **activate tf** 라고 입력하고 Enter를 눌러 명령을 실행(앞서 생성한 tf 환경을 활성화하는 명령)  
명령을 실행하면 (tf) 표시가 프롬프트 맨 앞에 나타남

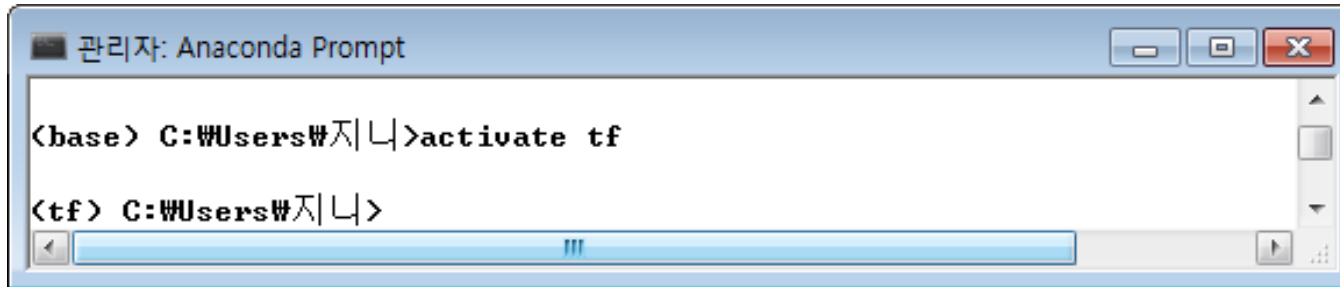
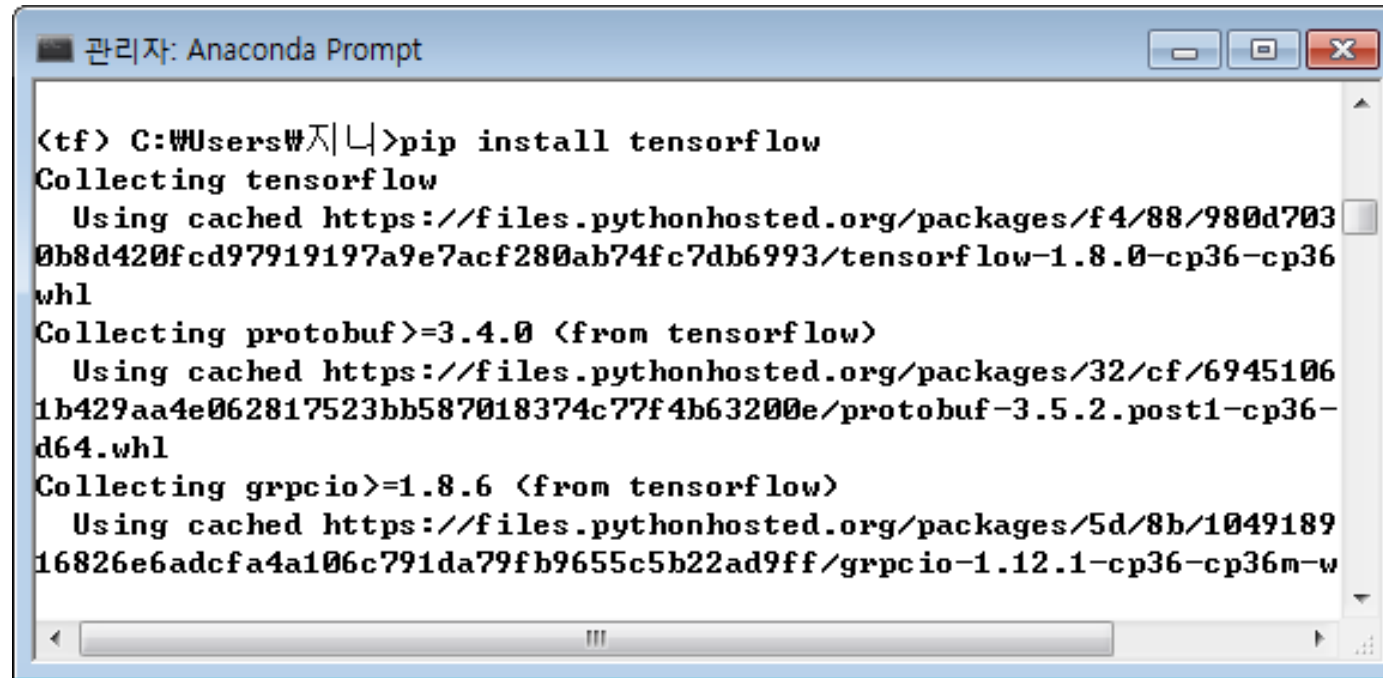


그림 1-10 tf 환경을 활성화

## 2 | 딥러닝 작업 환경 만들기

② **pip install tensorflow** 를 입력하고 Enter를 눌러 텐서플로를 설치



```
관리자: Anaconda Prompt

<tf> C:\Users\지니>pip install tensorflow
Collecting tensorflow
  Using cached https://files.pythonhosted.org/packages/f4/88/980d7030b8d420fcd97919197a9e7acf280ab74fc7db6993/tensorflow-1.8.0-cp36-cp36-
  whl
Collecting protobuf<=3.4.0 (from tensorflow)
  Using cached https://files.pythonhosted.org/packages/32/cf/69451061b429aa4e062817523bb587018374c77f4b63200e/protobuf-3.5.2.post1-cp36-
  d64.whl
Collecting grpcio<=1.8.6 (from tensorflow)
  Using cached https://files.pythonhosted.org/packages/5d/8b/104918916826e6adcfa4a106c791da79fb9655c5b22ad9ff/grpcio-1.12.1-cp36-cp36m-w
```

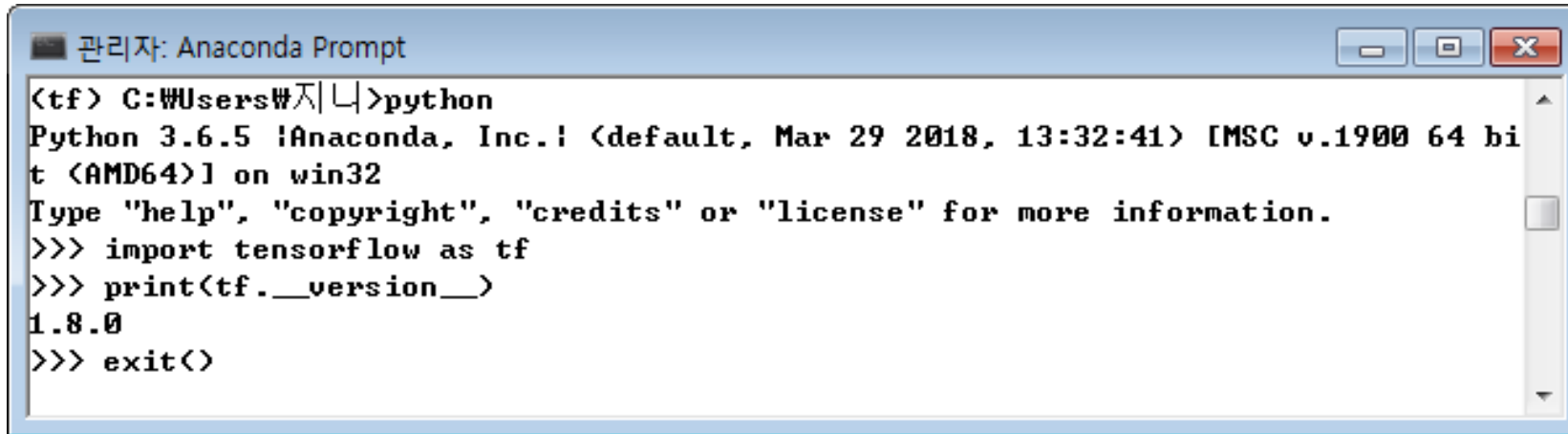
그림 1-11 텐서플로 설치

## 2 | 딥러닝 작업 환경 만들기

③ 텐서플로가 제대로 설치됐는지 확인하려면 **python**을 실행한 다음

**import tensorflow as tf**를 입력

그리고 **print(tf.\_\_version\_\_)**을 입력했을 때 텐서플로의 버전이 출력되면 설치가 완료된 것

A screenshot of an Anaconda Prompt window. The title bar reads '관리자: Anaconda Prompt'. The command prompt shows the following sequence of commands and outputs:

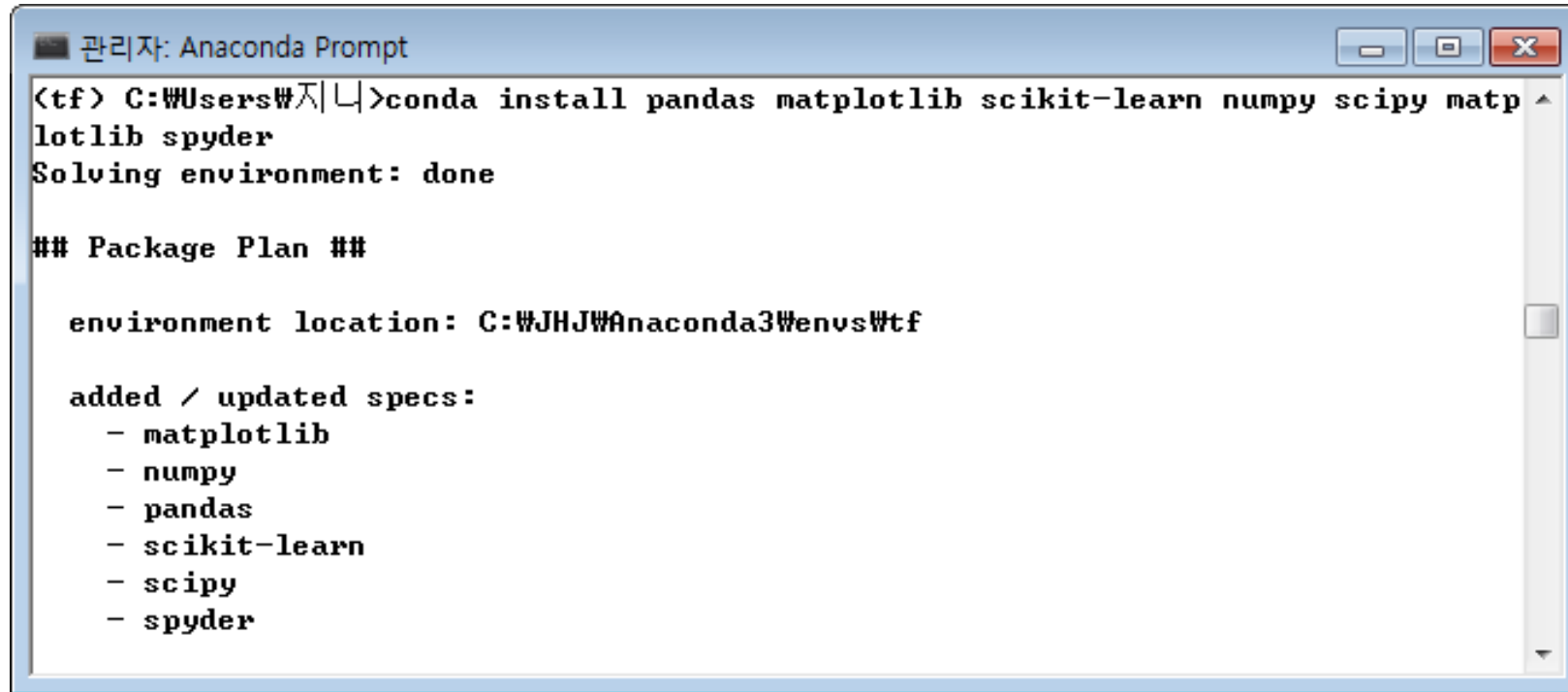
```
<tf> C:\Users\zz\Anaconda Prompt>python
Python 3.6.5 |Anaconda, Inc.| (default, Mar 29 2018, 13:32:41) [MSC v.1900 64 bi
t (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorflow as tf
>>> print(tf.__version__)
1.8.0
>>> exit()
```

그림 1-12 텐서플로 설치 여부 확인

## 2 | 딥러닝 작업 환경 만들기

필요한 라이브러리 설치

④ `conda install pandas matplotlib scikit-learn numpy scipy matplotlib spyder`



```
관리자: Anaconda Prompt
<tf> C:\Users\김지니>conda install pandas matplotlib scikit-learn numpy scipy matplotlib spyder
Solving environment: done

## Package Plan ##

  environment location: C:\JHJ\Anaconda3\envs\tf

added / updated specs:
- matplotlib
- numpy
- pandas
- scikit-learn
- scipy
- spyder
```

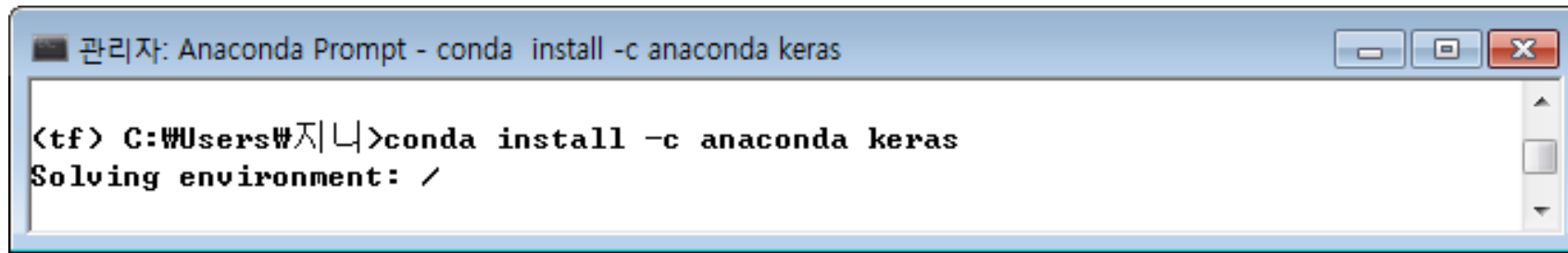
그림 1-13 필요한 라이브러리 설치

## 2 | 딥러닝 작업 환경 만들기

### 케라스 설치하기

① **pip install keras** 라고 입력하여 케라스를 설치

설치 안될 경우에는 **conda install -c anaconda keras**



```
관리자: Anaconda Prompt - conda install -c anaconda keras

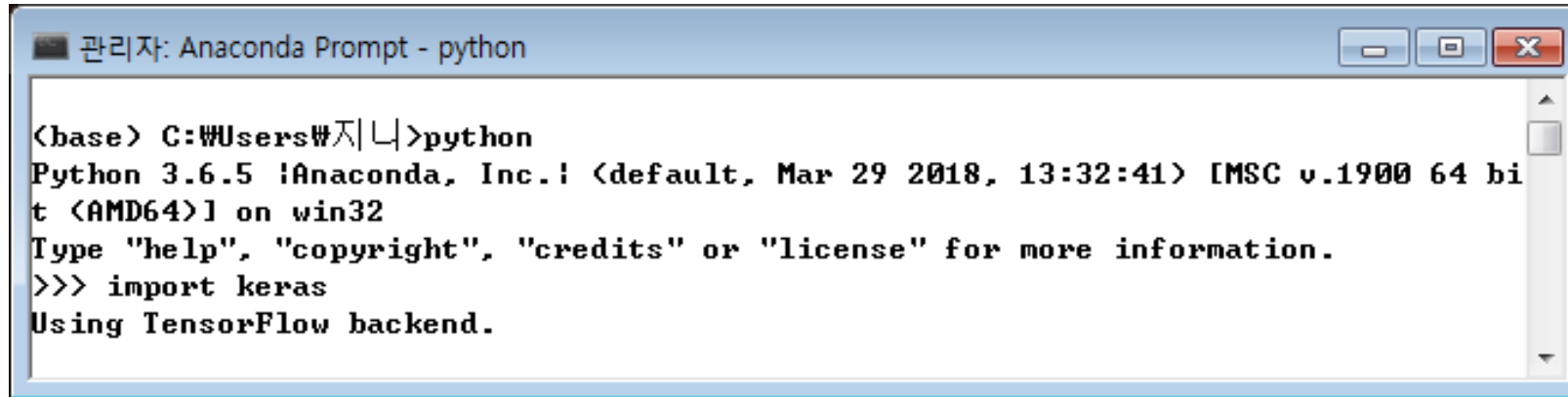
<tf> C:\Users\지니>conda install -c anaconda keras
Solving environment: /
```

그림 1-14 케라스 설치



## 2 | 딥러닝 작업 환경 만들기

- ② 케라스가 설치됐는지 확인하려면 마찬가지로 **python**을 실행한 다음
- ③ **import keras**를 입력했을 때 다음과 같이 출력되면 설치가 완료된 것



```
관리자: Anaconda Prompt - python

<base> C:\Users\지니>python
Python 3.6.5 |Anaconda, Inc.| (default, Mar 29 2018, 13:32:41) [MSC v.1900 64 bi
t (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import keras
Using TensorFlow backend.
```

그림 1-15 케라스 설치 여부 확인

- ④ 설치를 마친 다음 테스트를 위해 “파이참 설치하기”로 넘어감

### 3 | 파이참 설치하기

① 파이참 내려받기 페이지에 접속하여 Community 버전을 선택한 후 내려받음

- <https://www.jetbrains.com/pycharm/download/>

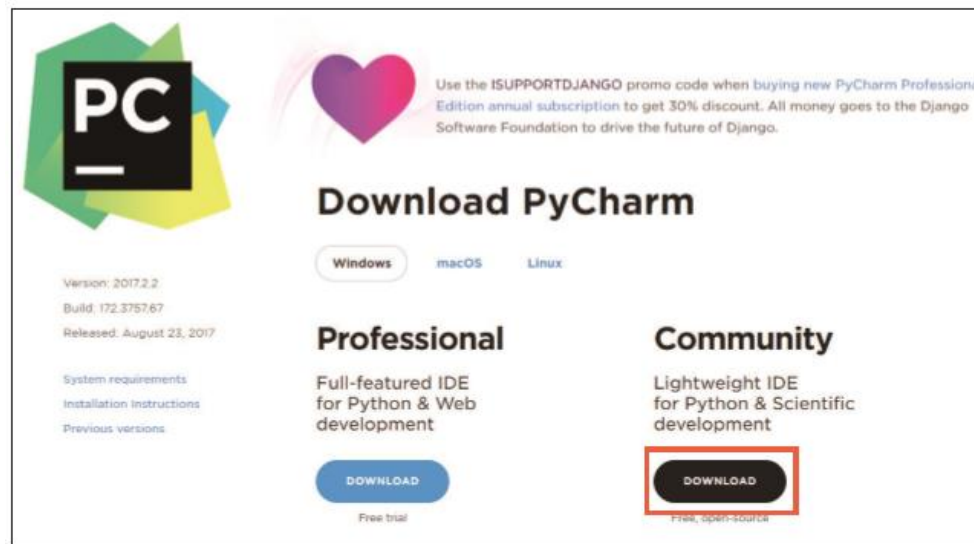


그림 1-16 파이참 내려받기 페이지에서 Community 버전 내려받기

### 3 | 파이참 설치하기

- ② 설치를 마쳤으면 파이참을 실행한 다음 Create New Project 버튼을 눌러 새 프로젝트를 만들

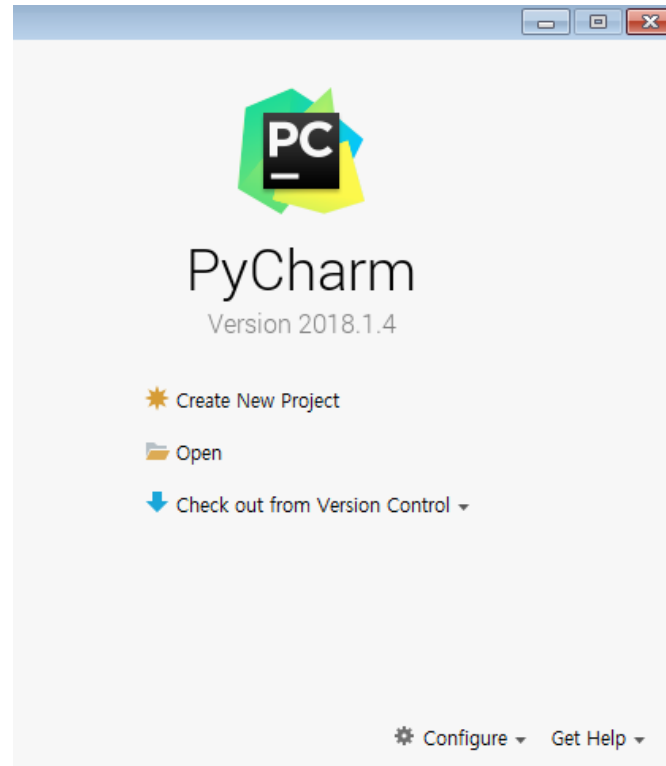


그림 1-17 파이참에서 새 프로젝트 생성

### 3 | 파이참 설치하기

- ③ Location 항목에 PycharmProject 폴더가 나오면 뒤에 `Wdeeplearning` 입력

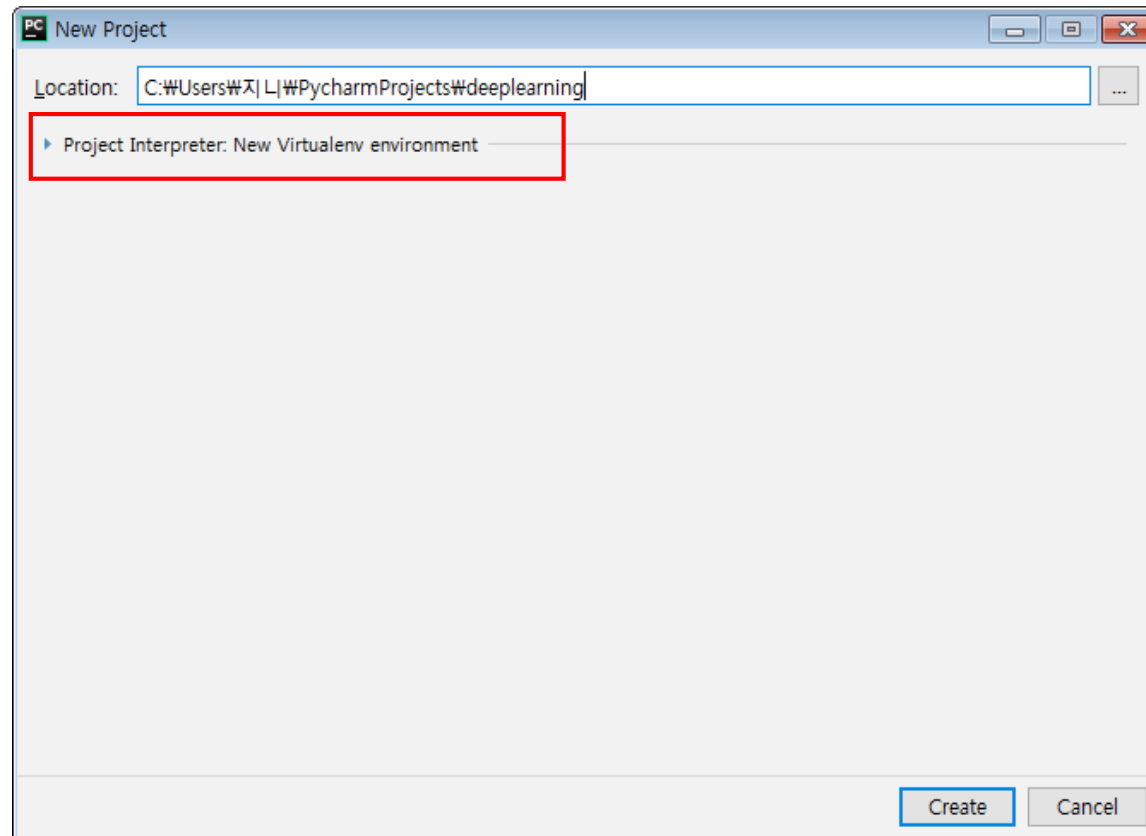


그림 1-18 작업 폴더 만들기

### 3 | 파이참 설치하기

#### ④ 아나콘다 환경을 불러오기 위해 Existing Interpreter 선택

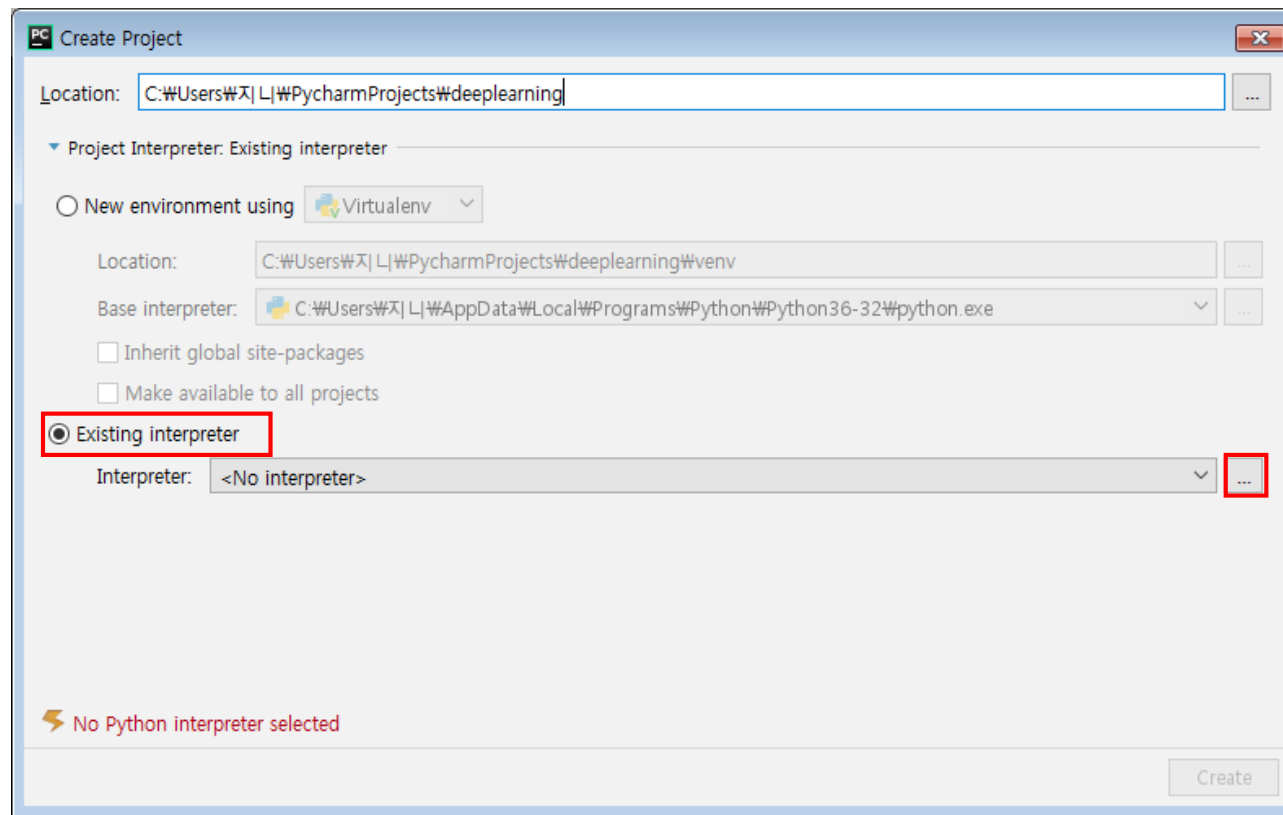


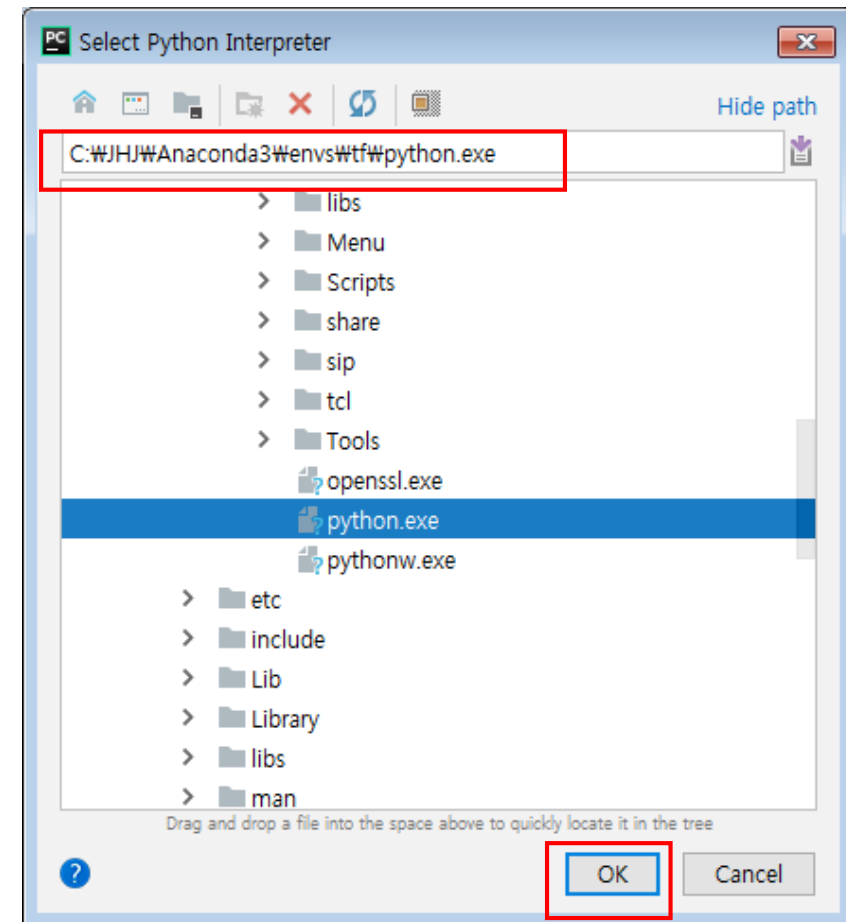
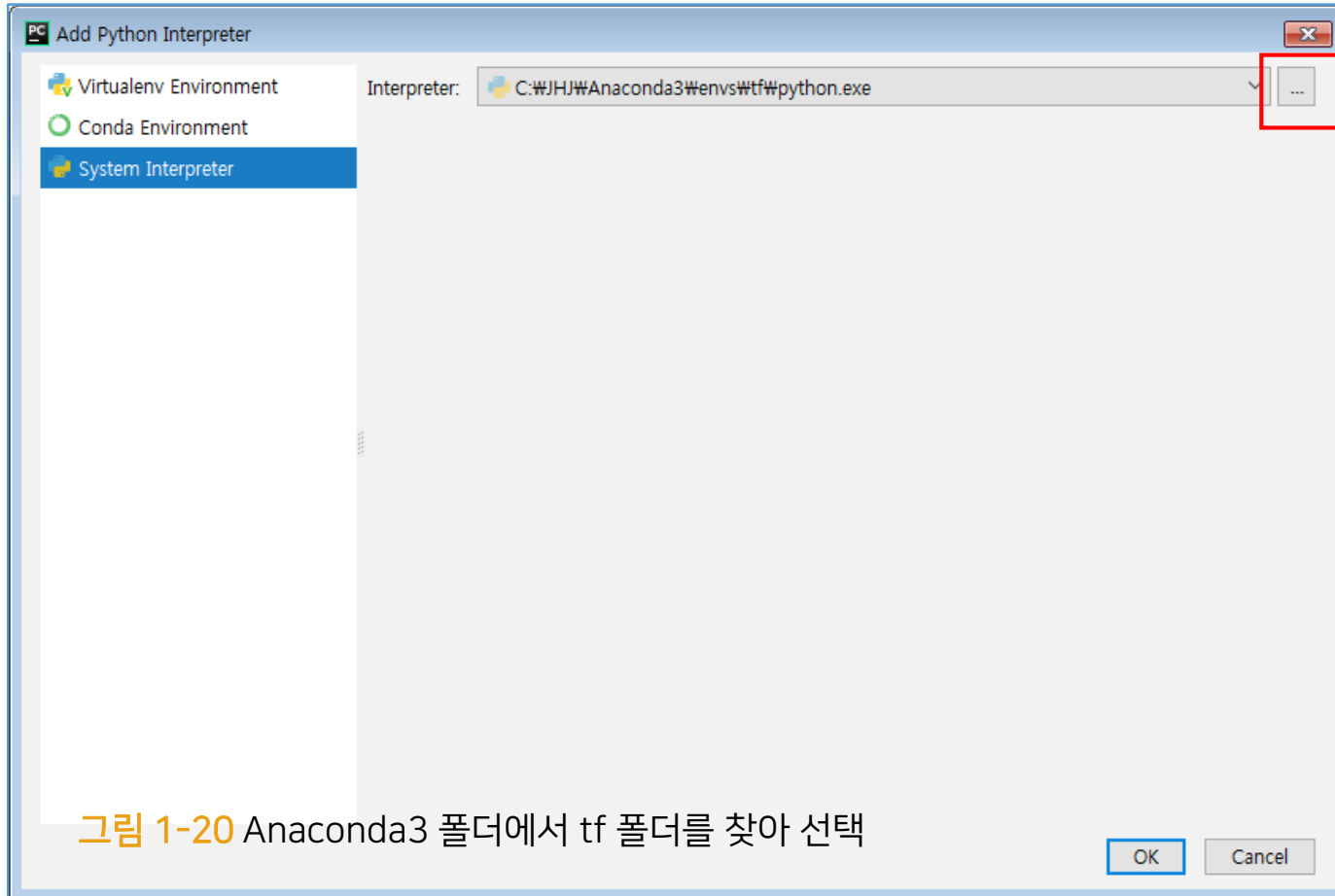
그림 1-19 아나콘다 환경 불러오기

### 3 | 파이참 설치하기

Anaconda 설치 시에 Default Path를 선택하면, "C:\WProgramData\WAnaconda3"를 직접 입력하고(시스템파일이어서 보이지 않음)  
Interpreter로  
"C:\WProgramData\WAnaconda3\envs\Wtutorial\Wtools\Wpython.exe"를 선택해야 함

⑤ tf 작업 환경이 Anaconda3 > envs > tf에 설치 되어 있음.

⑥ 그림 같이 tf 폴더 안에 있는 python.exe를 선택한 후 OK 버튼을 누름.



### 3 | 파이참 설치하기

⑦ Interpreter가 바뀌는 것을 확인(다소 시간이 걸릴 수 있음)

Create 버튼을 눌러 프로젝트 환경을 만들

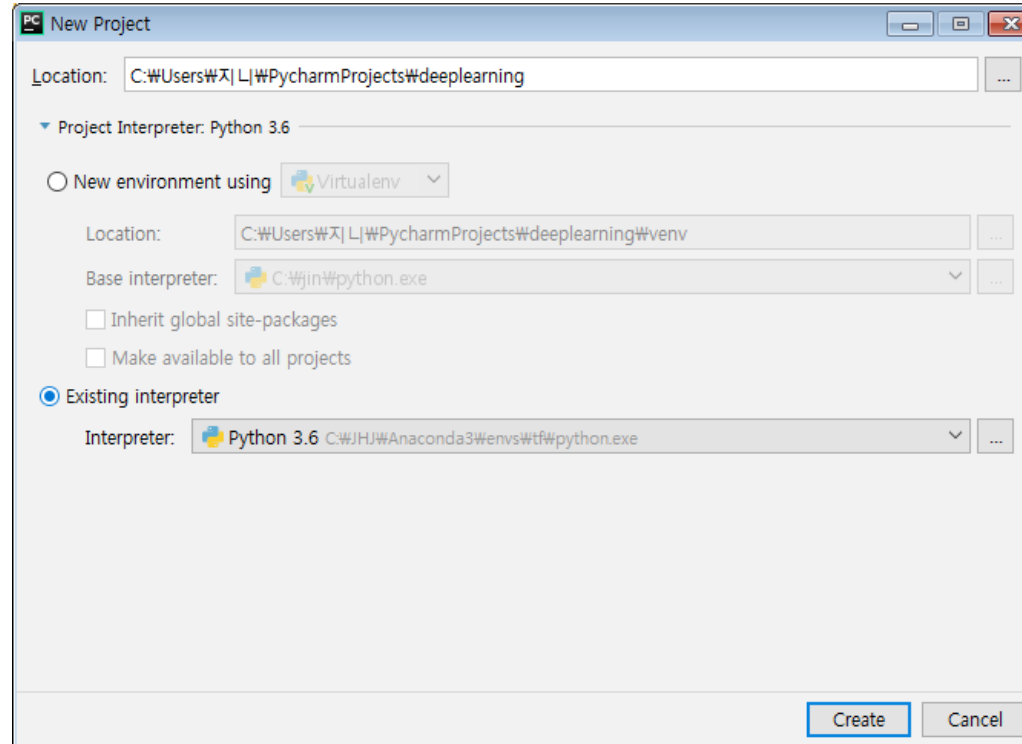


그림 1-21 Create 버튼 클릭

### 3 | 파이참 설치하기

- ⑧ 윈도우 탐색기를 열어 PycharmProjects 폴더에 들어가면 deeplearning 폴더가 생성된 것을 확인할 수 있음
- ⑨ deeplearning 폴더 안에 deep\_class 폴더 파일을 복사해 넣음

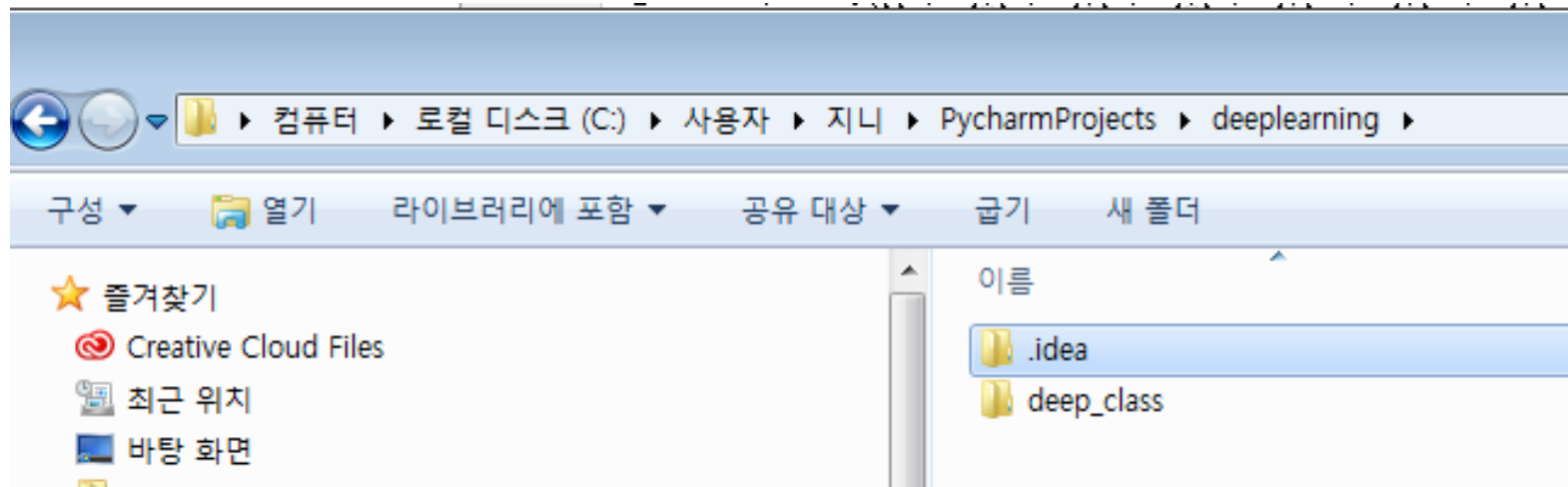


그림 1-22 deeplearning 폴더 생성 확인



### 3 | 파이참 설치하기

⑩ 다시 파이참을 실행. 소스 코드를 실행할 수 있게 설정된 것을 볼 수 있음

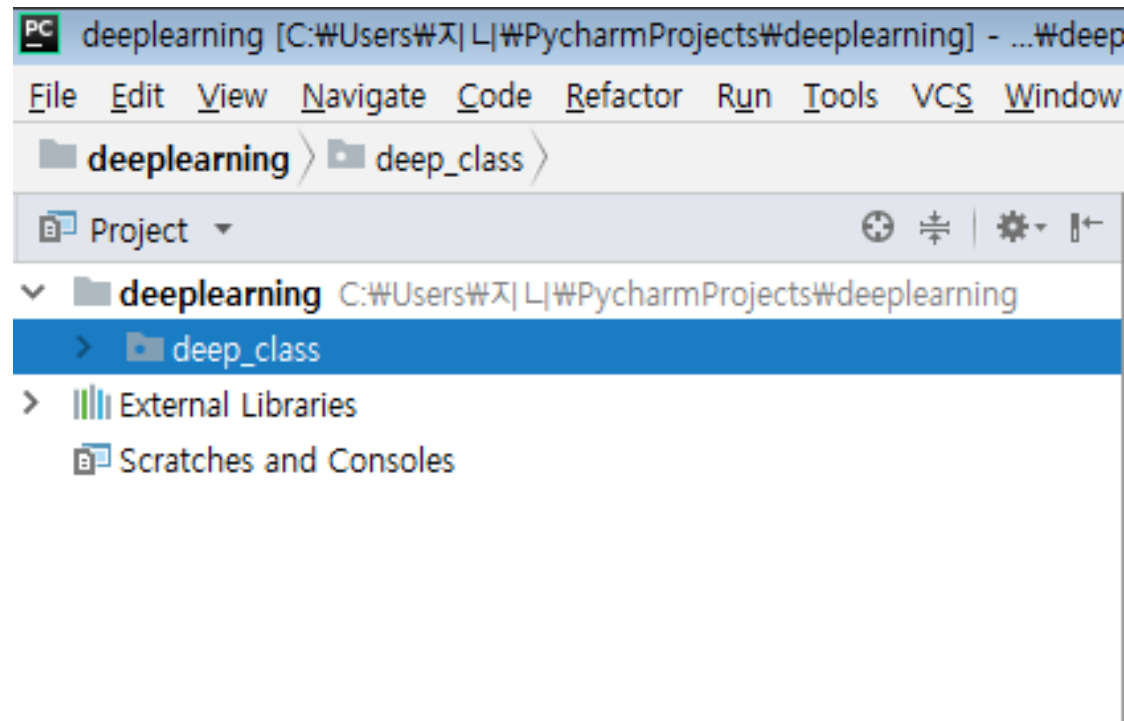


그림 1-23 파이참에서 소스 코드 폴더 목록 확인하기

## 4 | 딥러닝 실행하기

- ① 실습 코드를 불러오기 위해 파이참에서 보이는 deep\_class 폴더를 선택  
.py파일(파이썬 파일)을 선택하면 소스 파일이 열림

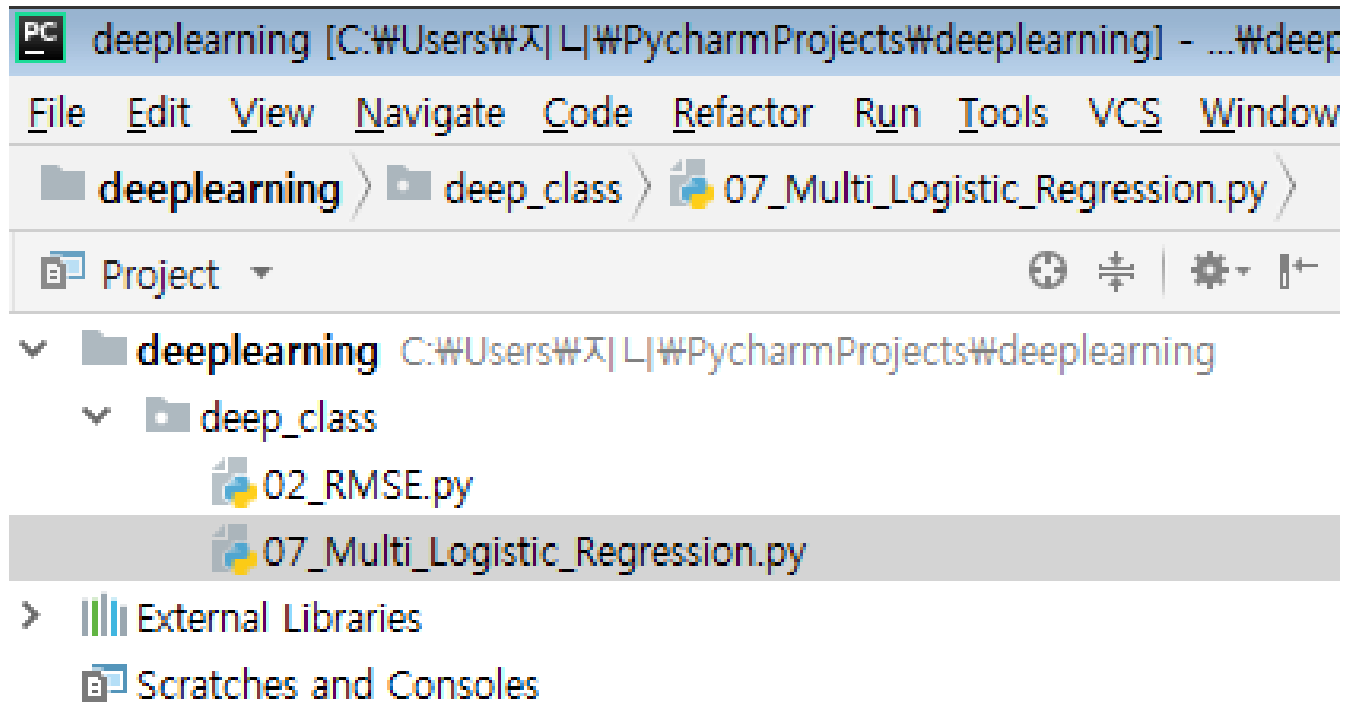


그림 1-24 소스 파일 불러오기

```

1  -*- coding: utf-8 -*-
2  import ...
3
4
5  # 실행할 때마다 같은 결과를 출력하기 위한 seed 값 설정
6  seed = 0
7  np.random.seed(seed)
8  tf.set_random_seed(seed)
9
10 # x,y의 데이터 값
11 x_data = np.array([[2, 3],[4, 3],[6, 4],[8, 6],[10, 7],[12, 8],[14, 9]])
12 y_data = np.array([0, 0, 0, 1, 1, 1, 1]).reshape(7, 1)
13
14 # 입력 값을 플레이스 홀더에 저장
15 X = tf.placeholder(tf.float64, shape=[None, 2])
16 Y = tf.placeholder(tf.float64, shape=[None, 1])
17
18 # 기울기 a와 bias b의 값을 임의로 정함.
19 a = tf.Variable(tf.random_uniform([2,1], dtype=tf.float64)) # [2,1] 의미: 들어오는 값은 2개, 나가는 값은 1개
20 b = tf.Variable(tf.random_uniform([1], dtype=tf.float64))
21
22 # y 시그모이드 함수의 방정식을 세움
23 y = tf.sigmoid(tf.matmul(X, a) + b)
24
25 # 오차를 구하는 함수
26 loss = -tf.reduce_mean(Y * tf.log(y) + (1 - Y) * tf.log(1 - y))
27
28 # 학습률 값
29 learning_rate=0.1
30

```

## 4 | 딥러닝 실행하기

② 메뉴의 Run > Run을 선택해 해당 파이썬파일 실행

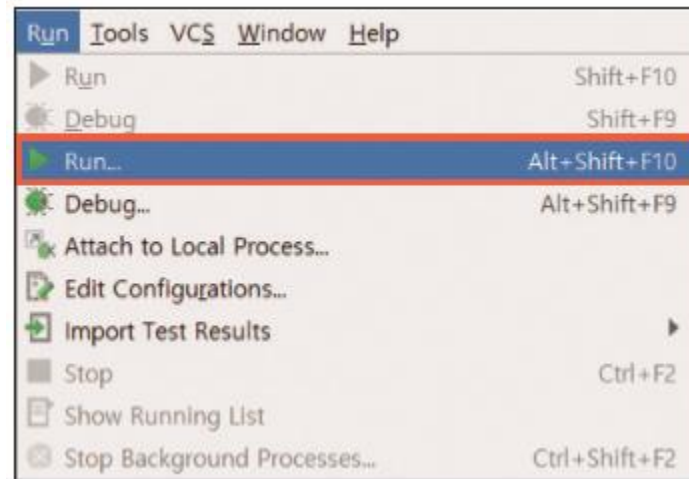
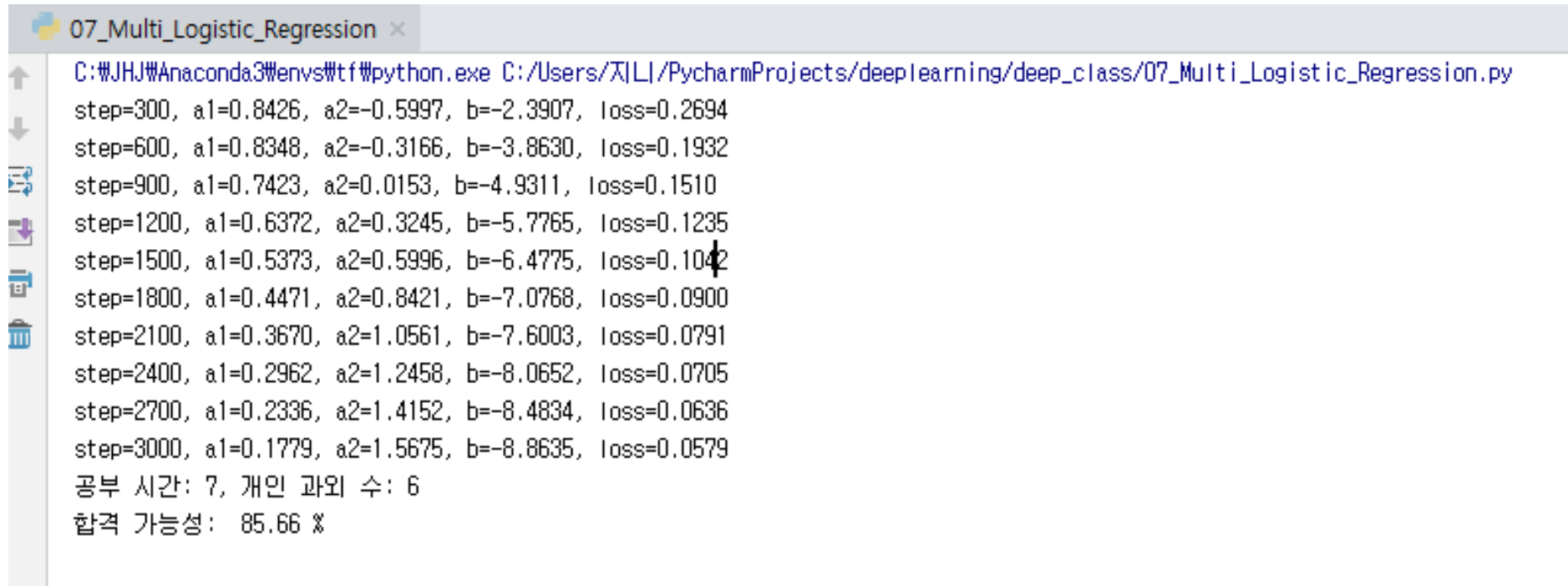


그림 1-25 파이썬 파일 실행

## 4 | 딥러닝 실행하기

③ 정상적으로 모두 설치되었다면 다음과 같이 코드가 성공적으로 실행됨



```
07_Multi_Logistic_Regression x
C:\JHJ\Anaconda3\envs\tf\python.exe C:/Users/지LI/PycharmProjects/deeplearning/deep_class/07_Multi_Logistic_Regression.py
step=300, a1=0.8426, a2=-0.5997, b=-2.3907, loss=0.2694
step=600, a1=0.8348, a2=-0.3166, b=-3.8630, loss=0.1932
step=900, a1=0.7423, a2=0.0153, b=-4.9311, loss=0.1510
step=1200, a1=0.6372, a2=0.3245, b=-5.7765, loss=0.1235
step=1500, a1=0.5373, a2=0.5996, b=-6.4775, loss=0.1042
step=1800, a1=0.4471, a2=0.8421, b=-7.0768, loss=0.0900
step=2100, a1=0.3670, a2=1.0561, b=-7.6003, loss=0.0791
step=2400, a1=0.2962, a2=1.2458, b=-8.0652, loss=0.0705
step=2700, a1=0.2336, a2=1.4152, b=-8.4834, loss=0.0636
step=3000, a1=0.1779, a2=1.5675, b=-8.8635, loss=0.0579
공부 시간: 7, 개인 과외 수: 6
합격 가능성: 85.66 %
```

그림 1-26 실행 결과 확인

# 여러 가지 환경 설정

---

- 1 | GPU 환경에서 설치하기
- 2 | 리눅스에서 설치하기

# 1 | GPU 환경에서 설치하기

## 내 컴퓨터에서 GPU 확인하기

- 시스템 패널을 열고 왼쪽에서 장치 관리자를 클릭

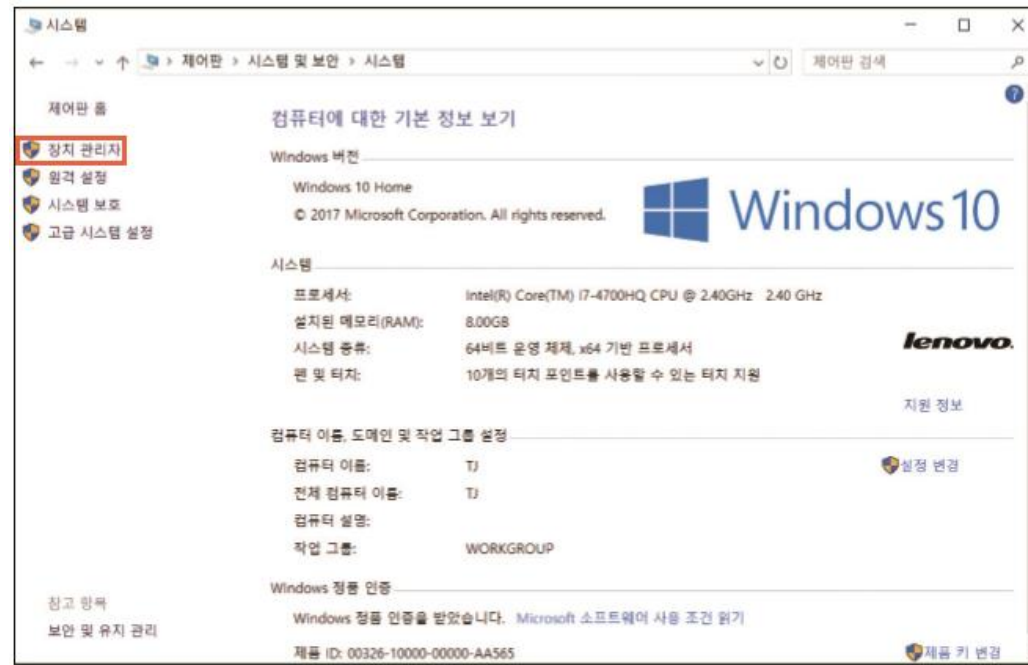


그림 D-1 시스템 패널 > 장치 관리자 선택

## 1 | GPU 환경에서 설치하기

- GPU가 장착된 컴퓨터라면 '디스플레이 어댑터' 항목에서 설치된 GPU 종류를 볼 수 있음

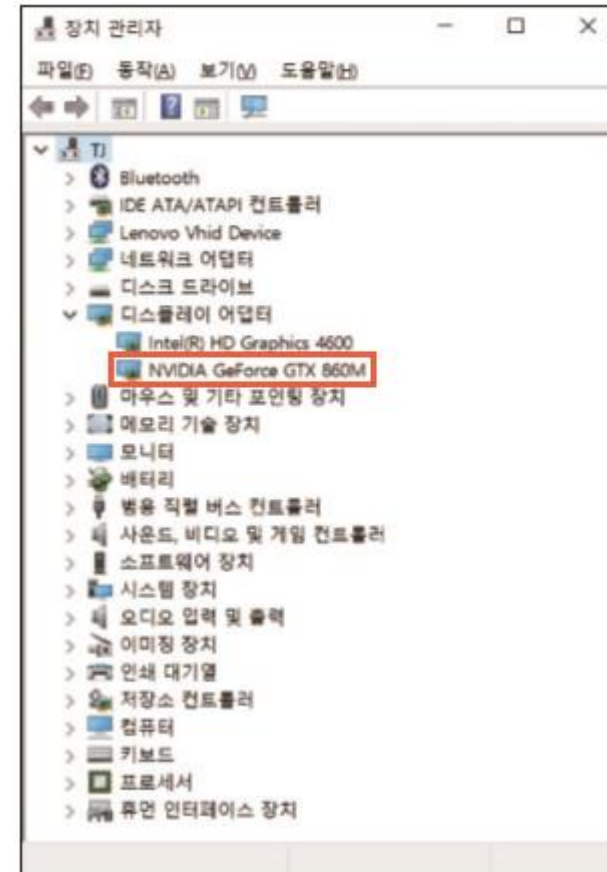


그림 D-2 PC에 장착된 GPU 선택



# 1 | GPU 환경에서 설치하기

## 아나콘다 설치하기

① 아나콘다 64비트 인스톨러를 내려받아 설치(파이썬 3.5 이상 버전, 64비트)

- <https://www.continuum.io/downloads>



## 1 | GPU 환경에서 설치하기



그림 D-3 아나콘다 설치

## 1 | GPU 환경에서 설치하기

- ② 윈도우 버튼을 누른 후 방금 전에 설치한  
아나콘다 프로그램 폴더 중  
Anaconda Prompt를 선택  
프로그램 폴더가 보이지 않을 경우,  
검색(🔍) 버튼을 누르고  
Anaconda Prompt를 입력해도 됨

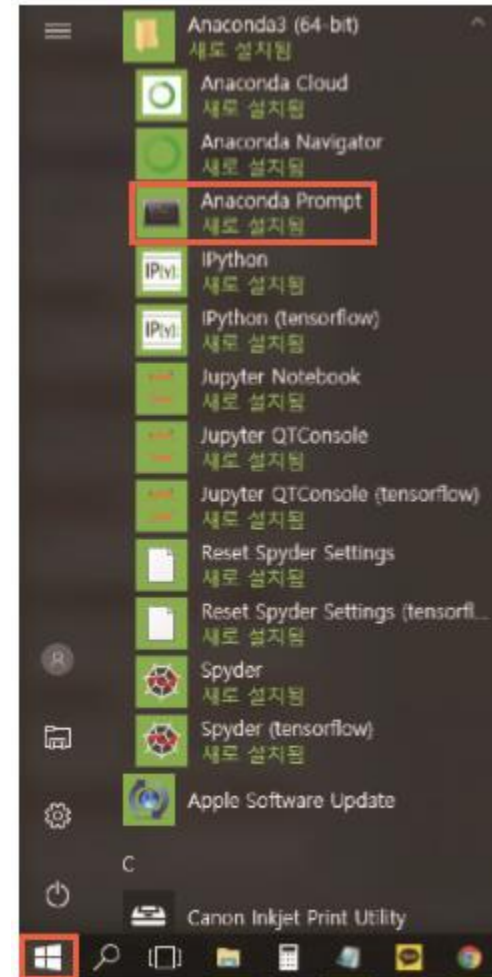


그림 D-4 Anaconda Prompt를 선택

## 1 | GPU 환경에서 설치하기

- ③ 명령 프롬프트가 나타나면 아나콘다 환경과 교재 실행을 위한 작업 환경을 tutorial이란 이름으로 생성하여 저장(파이썬 3.6로 환경 설정)

```
conda create -n tutorial python=3.6 numpy scipy matplotlib  
spyder pandas seaborn scikit-learn h5py
```

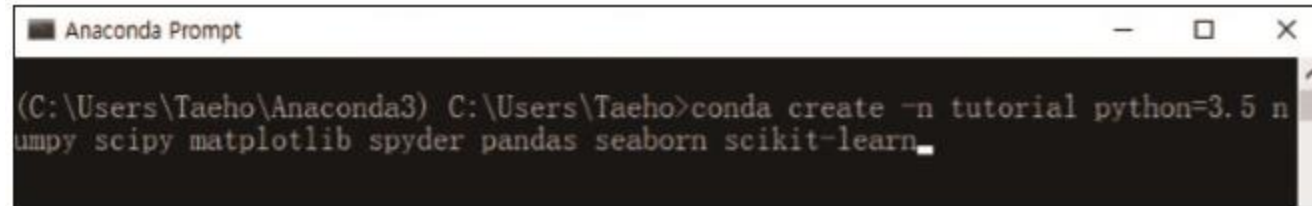


그림 D-5 아나콘다 환경 설정

## 1 | GPU 환경에서 설치하기

- ④ 다음으로 activate tutorial을 실행  
(tutorial) 표시가 프롬프트 맨 앞에 나타남



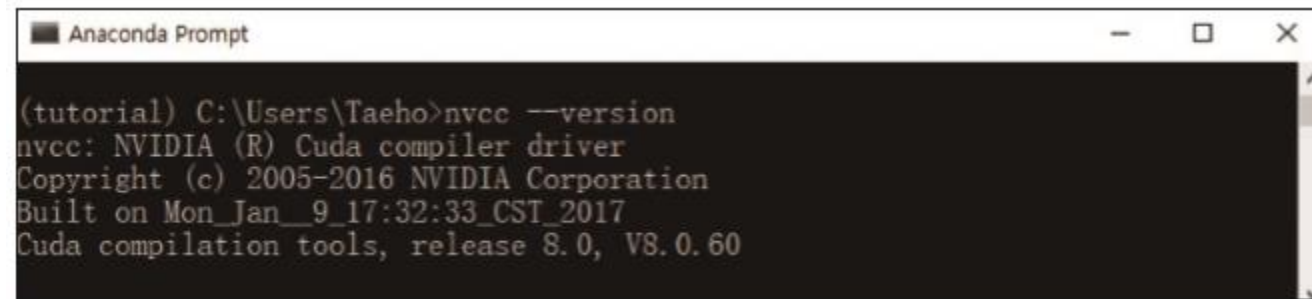
```
Anaconda Prompt
(C:\Users\Taeho\Anaconda3) C:\Users\Taeho>activate tutorial
(tutorial) C:\Users\Taeho>_
```

그림 D-6 저장된 환경을 활성화

## 1 | GPU 환경에서 설치하기

### GPU 구동을 위한 드라이버 설치하기

- ① 먼저 명령 프롬프트에서 `nvcc --version`을 실행하여 드라이버가 설치됐는지 확인해 보자



```
Anaconda Prompt

(tutorial) C:\Users\Taeho>nvcc --version
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2016 NVIDIA Corporation
Built on Mon_Jan__9_17:32:33_CST_2017
Cuda compilation tools, release 8.0, V8.0.60
```

그림 D-7 드라이버 설치 확인

## 1 | GPU 환경에서 설치하기

② 만일 그림 D-7과 같이 나오지 않는다면 NVIDIA 홈페이지에 접속해서 자신의 OS 환경에 맞는 CUDA를 내려받아 설치

- <https://developer.nvidia.com/cuda-downloads>

## 1 | GPU 환경에서 설치하기

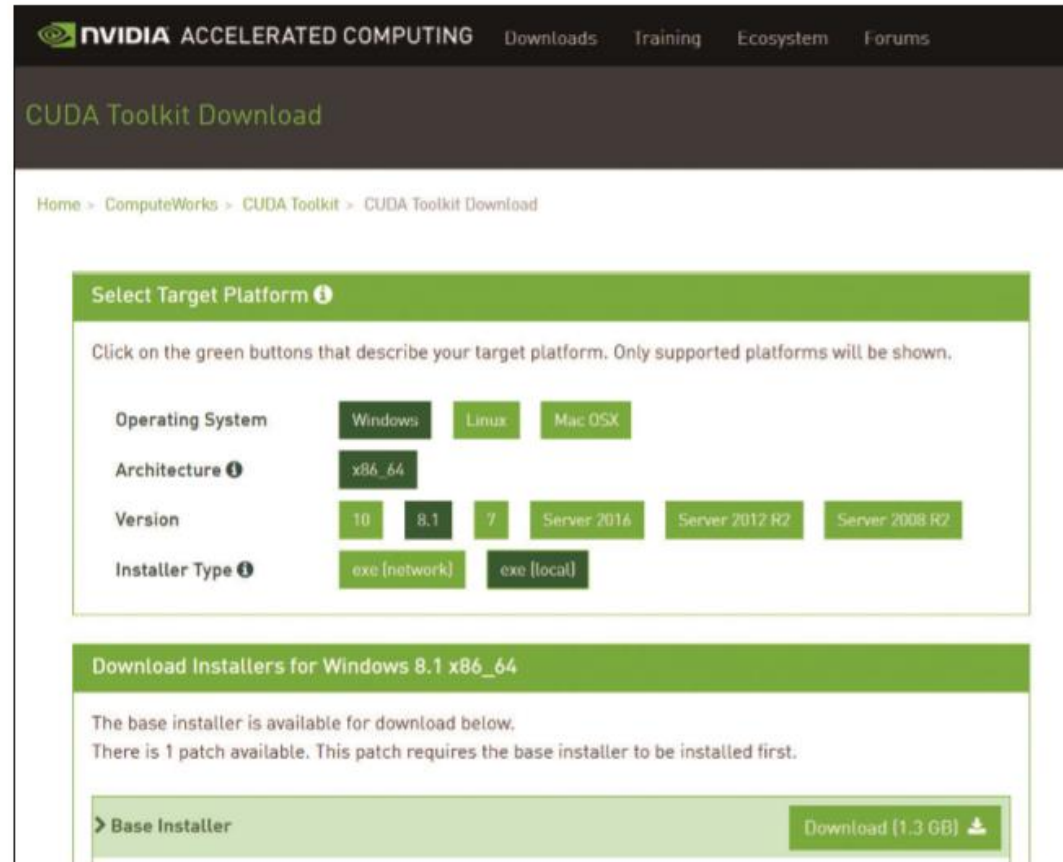


그림 D-8 NVIDIA 홈페이지 접속



## 1 | GPU 환경에서 설치하기

- ③ CUDA에 이어 아래 홈페이지에서 cuDNN v6.0(for CUDA 8.0)를 내려받음  
(간단한 회원가입이 필요함)

- <https://developer.nvidia.com/rdp/cudnn-download>

## 1 | GPU 환경에서 설치하기

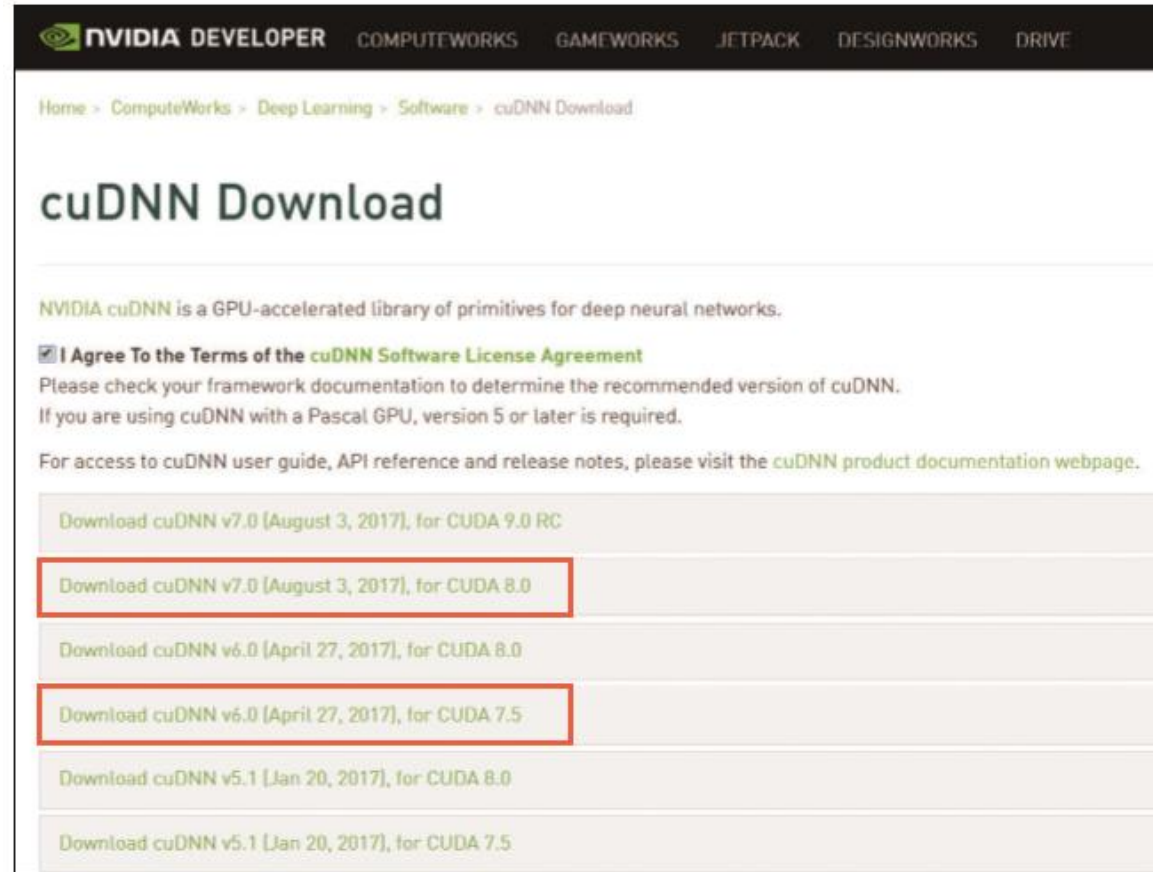


그림 D-9 드라이버 선택해서 내려받기

## 1 | GPU 환경에서 설치하기

- ④ 내려받은 cuDNN의 압축을 풀어 CUDA 디렉터리(C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v8.0)에 덮어쓰

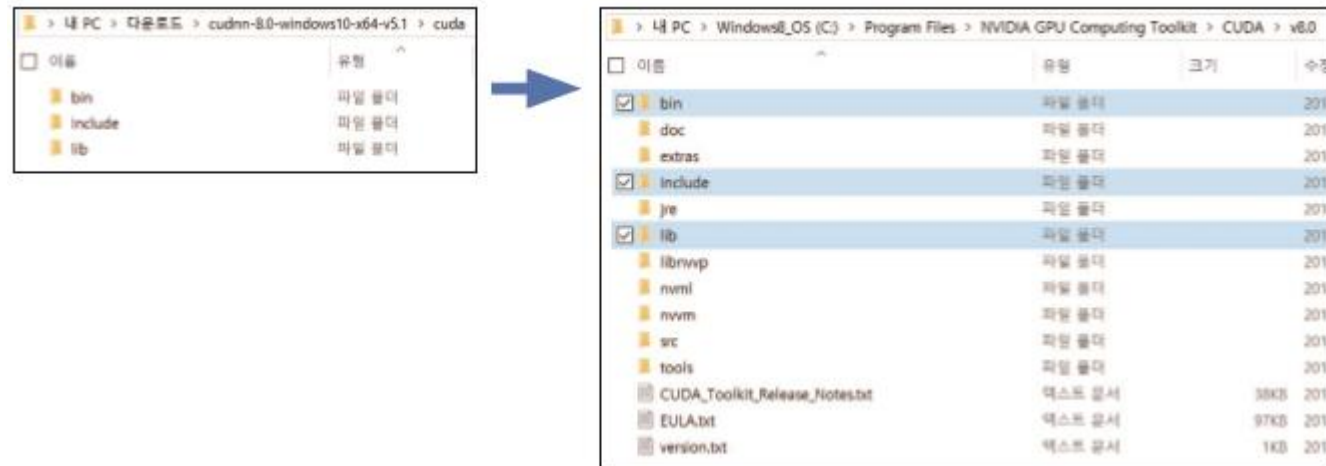
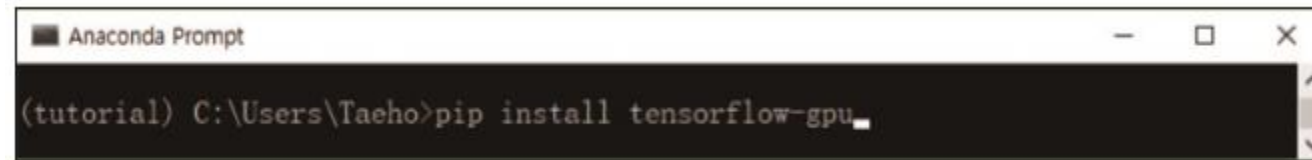


그림 D-10 cuDNN 파일압축 해제 및 덮어쓰우기

## 1 | GPU 환경에서 설치하기

- ⑤ pip install tensorflow-gpu를 입력해 최신 버전의 텐서플로를 설치

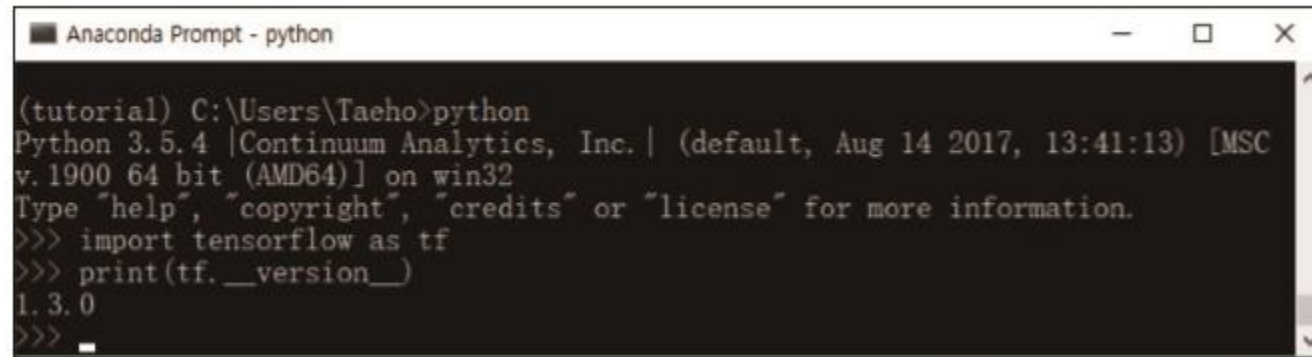


```
Anaconda Prompt
(tutorial) C:\Users\Taeho>pip install tensorflow-gpu
```

그림 D-11 tensorflow-gpu 설치

## 1 | GPU 환경에서 설치하기

- ⑥ 텐서플로가 제대로 동작하는지 확인하려면 python을 실행한 다음 import tensorflow as tf를 입력 print(tf.\_\_version\_\_)을 입력했을 때 텐서플로의 버전이 출력되면 설치가 완료된 것



```
Anaconda Prompt - python

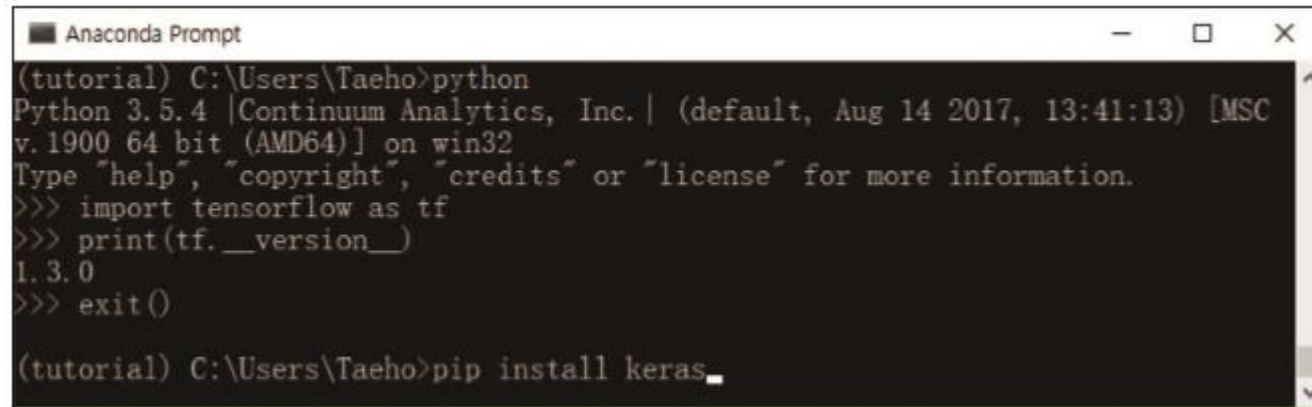
(tutorial) C:\Users\Taeho>python
Python 3.5.4 |Continuum Analytics, Inc. | (default, Aug 14 2017, 13:41:13) [MSC
v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorflow as tf
>>> print(tf.__version__)
1.3.0
>>> _
```

그림 D-12 텐서플로 설치 확인

## 1 | GPU 환경에서 설치하기

### 케라스 설치하기

- ① pip install keras라고 입력하여 케라스를 설치



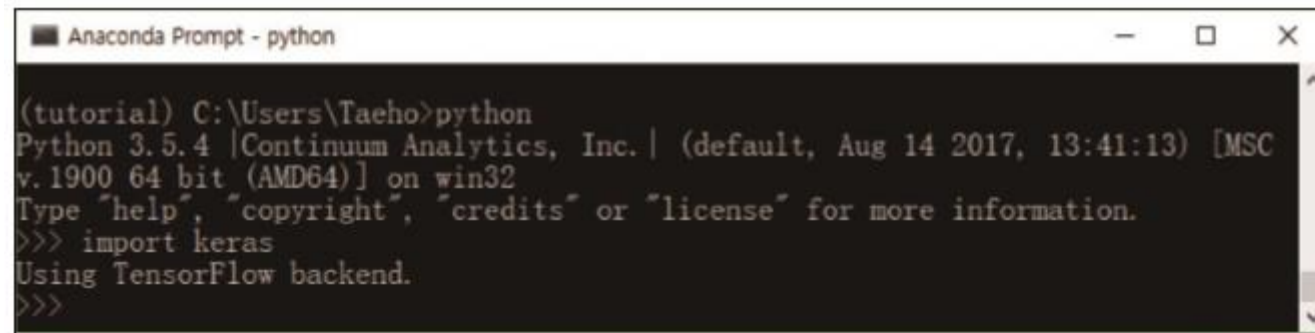
```
Anaconda Prompt
(tutorial) C:\Users\Taeho>python
Python 3.5.4 |Continuum Analytics, Inc. | (default, Aug 14 2017, 13:41:13) [MSC
v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorflow as tf
>>> print(tf.__version__)
1.3.0
>>> exit()

(tutorial) C:\Users\Taeho>pip install keras
```

그림 D-13 케라스 설치

## 1 | GPU 환경에서 설치하기

- ② 케라스가 설치됐는지 확인하려면 마찬가지로 파이썬을 실행한 다음 import keras를 입력해서 다음과 같이 출력되면 설치된 것

A screenshot of an Anaconda Prompt window titled "Anaconda Prompt - python". The window shows a Python 3.5.4 shell with the following text: "(tutorial) C:\Users\Taeho>python", "Python 3.5.4 |Continuum Analytics, Inc. | (default, Aug 14 2017, 13:41:13) [MSC v.1900 64 bit (AMD64)] on win32", "Type 'help', 'copyright', 'credits' or 'license' for more information.", ">>> import keras", "Using TensorFlow backend.", and ">>>". The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

```
(tutorial) C:\Users\Taeho>python
Python 3.5.4 |Continuum Analytics, Inc. | (default, Aug 14 2017, 13:41:13) [MSC
v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import keras
Using TensorFlow backend.
>>>
```

그림 D-13 케라스 설치 여부 확인

## 2 | 리눅스에서 설치하기

- 리눅스에서 설치하기 전 다음 사항을 점검해 보자
  - 1) 딥러닝 구동에 필요한 텐서플로는 현재 우분투 14.04 버전 이상을 지원합니다.  
또한, 64비트 컴퓨터에서 안정적으로 실행됩니다.
  - 2) CPU에서 동작시킬 것인지, GPU에서 동작시킬 것인지를 선택해야 합니다. CPU  
에서 설치하는 것이 쉽고 빠릅니다. GPU에서 구동하면 더 많은 데이터를 더 빨리 학습시킬 수 있습니다(실전 프로젝트는 GPU 환경에서 구동하는 것을 추천합니다).
  - 3) GPU 환경에서 실행하기를 원하는 경우 CUDA 8.0과 cuDNN 5.1이 미리 설치돼 있어야 합니다. 각각  
의 설치 방법은 NVIDIA 홈페이지([www.nvidia.com](http://www.nvidia.com))를 참고하세요.



## 2 | 리눅스에서 설치하기

### 설치 방법

- 아나콘다를 이용해서 설치하는 방법을 설명
  - 리눅스에서 설치하는 방법은 이 밖에도 virtualenv, 'native' pip, Docker 등 여러 가지가 있음
  - 다른 방법으로 설치하고 싶은 사람은 텐서플로 홈페이지의 리눅스 설치 안내 참고
- 
- [https://www.tensorflow.org/install/install\\_linux](https://www.tensorflow.org/install/install_linux)

## 2 | 리눅스에서 설치하기

① 아나콘다 내려받기 페이지에서 아나콘다 인스톨러를 내려받아 설치

- <https://www.continuum.io/downloads>

② 설치가 끝나면 터미널을 열고 다음과 같이 텐서플로 환경을 만들

```
$ conda create -n tensorflow
```

③ 만들어진 환경을 구동

```
$ source activate tensorflow  
(tensorflow)$ # 구동 후 이렇게 프롬프트가 바뀌어야 합니다.
```

## 2 | 리눅스에서 설치하기

- ④ pip install tensorflow-gpu를 입력하고 **Enter**를 눌러 텐서플로를 설치

```
(tensorflow)$ pip install tensorflow-gpu
```

- ⑤ 이때 제대로 설치되지 않을 경우, 아래와 같이 패키지 주소를 직접 적어서 설치

```
(tensorflow)$ pip install --ignore-installed --upgrade tfBinaryURL
```

이때 tfBinaryURL 부분에는 텐서플로 파이썬 패키지의 주소를 적음

패키지 주소는 다음과 같음

## 2 | 리눅스에서 설치하기

### 파이썬 2.7의 경우

- CPU 버전: [https://storage.googleapis.com/tensorflow/linux/cpu/tensorflow-1.3.0-cp27-none-linux\\_x86\\_64.whl](https://storage.googleapis.com/tensorflow/linux/cpu/tensorflow-1.3.0-cp27-none-linux_x86_64.whl)
- GPU 버전: [https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow\\_gpu-1.3.0-cp27-none-linux\\_x86\\_64.whl](https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow_gpu-1.3.0-cp27-none-linux_x86_64.whl)

### 파이썬 3.4의 경우

- CPU 버전: [https://storage.googleapis.com/tensorflow/linux/cpu/tensorflow-1.3.0-cp34-cp34m-linux\\_x86\\_64.whl](https://storage.googleapis.com/tensorflow/linux/cpu/tensorflow-1.3.0-cp34-cp34m-linux_x86_64.whl)
- GPU 버전: [https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow\\_gpu-1.3.0-cp34-cp34m-linux\\_x86\\_64.whl](https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow_gpu-1.3.0-cp34-cp34m-linux_x86_64.whl)

## 2 | 리눅스에서 설치하기

### 파이썬 3.5의 경우

- CPU 버전: [https://storage.googleapis.com/tensorflow/linux/cpu/tensorflow-1.3.0-cp35-cp35m-linux\\_x86\\_64.whl](https://storage.googleapis.com/tensorflow/linux/cpu/tensorflow-1.3.0-cp35-cp35m-linux_x86_64.whl)
- GPU 버전: [https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow\\_gpu-1.3.0-cp35-cp35m-linux\\_x86\\_64.whl](https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow_gpu-1.3.0-cp35-cp35m-linux_x86_64.whl)

### 파이썬 3.6의 경우

- CPU 버전: [https://storage.googleapis.com/tensorflow/linux/cpu/tensorflow-1.3.0-cp36-cp36m-linux\\_x86\\_64.whl](https://storage.googleapis.com/tensorflow/linux/cpu/tensorflow-1.3.0-cp36-cp36m-linux_x86_64.whl)
- GPU 버전: [https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow\\_gpu-1.3.0-cp36-cp36m-linux\\_x86\\_64.whl](https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow_gpu-1.3.0-cp36-cp36m-linux_x86_64.whl)

## 2 | 리눅스에서 설치하기

- ⑥ 텐서플로 설치 여부를 확인하고자 파이썬을 실행

```
$ python
```

- ⑦ 다음과 같이 파이썬 코드를 실행

```
import tensorflow as tf  
print(tf.__version__)
```

- ⑧ 케라스를 설치

```
pip install keras
```

## 2 | 리눅스에서 설치하기

- ⑨ 케라스 설치 여부 확인을 위해 파이썬을 실행

```
$ python
```

- ⑩ 다음과 같이 파이썬 코드를 실행시켜 봄

```
# Python  
import keras
```

- ⑪ 파이참을 통해 실행하려면 다음과 같이 파이참을 설치

```
$ sudo apt-get update  
$ sudo apt-get install pycharm
```

## 2 | 리눅스에서 설치하기

### ⑫ 파이참을 실행

```
$ pycharm-community
```

### ⑬ 파이참을 열어 소스 코드를 엮

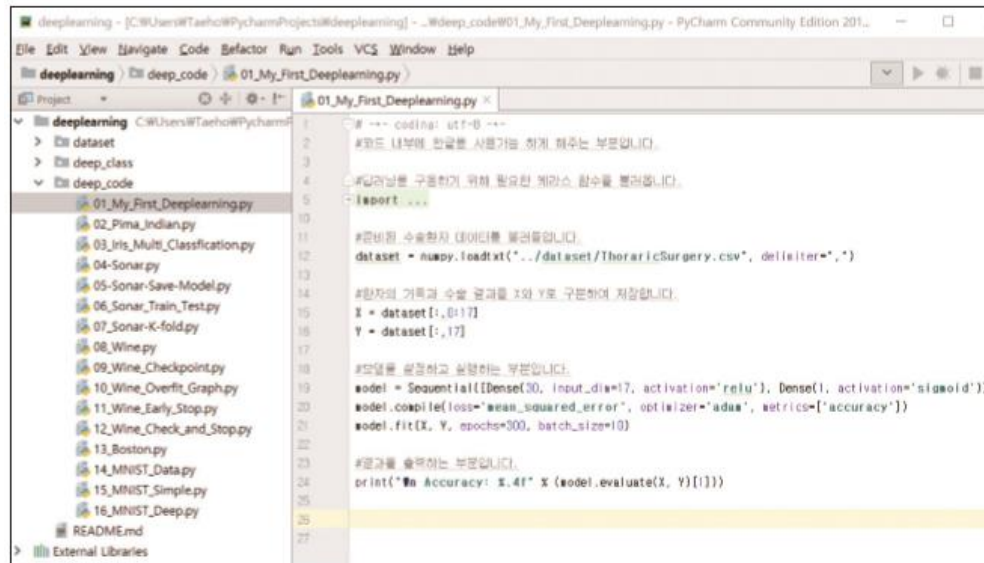


그림 D-15 파이참에서 열기



## 2 | 리눅스에서 설치하기

- ⑭ Shift + F10 또는 Run> Run 해당 파이썬 파일을 선택해서 실행

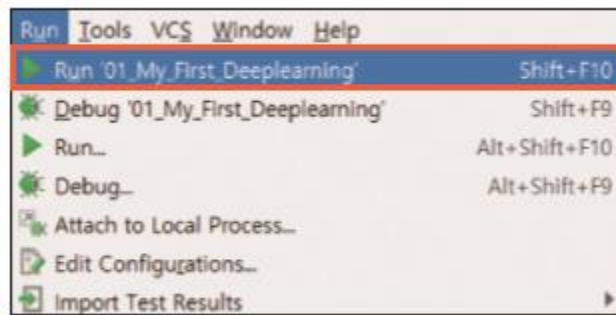
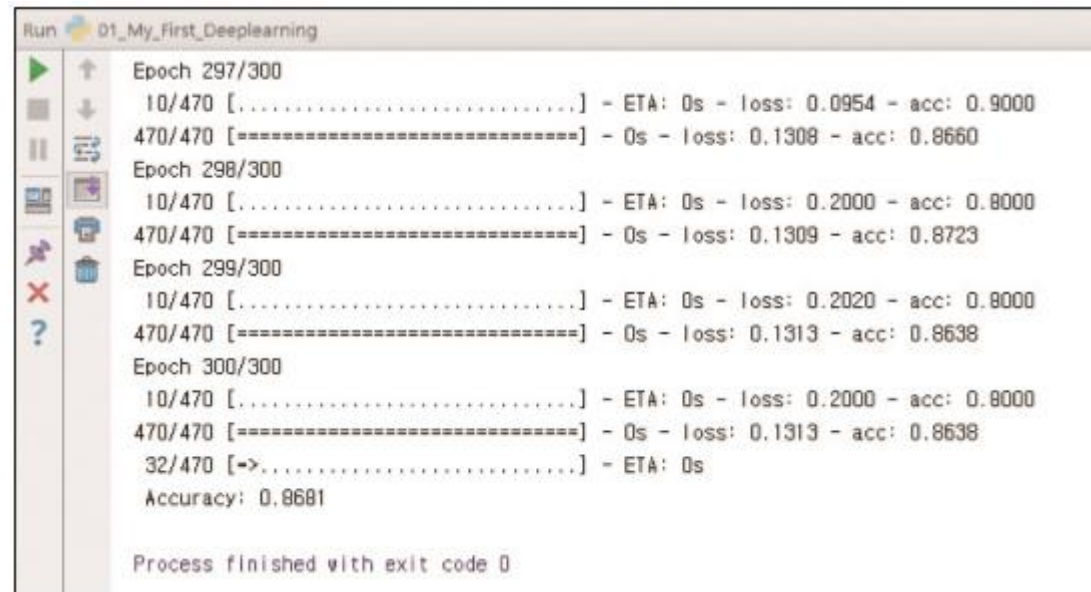


그림 D-16 Shift + F10으로 프로젝트 실행

## 2 | 리눅스에서 설치하기

⑮ 설치가 모두 끝나면 다음과 같이 코드가 성공적으로 실행됨



```
Run D1_My_First_DeepLearning
Epoch 297/300
 10/470 [.....] - ETA: 0s - loss: 0.0954 - acc: 0.9000
470/470 [=====] - 0s - loss: 0.1308 - acc: 0.8660
Epoch 298/300
 10/470 [.....] - ETA: 0s - loss: 0.2000 - acc: 0.8000
470/470 [=====] - 0s - loss: 0.1309 - acc: 0.8723
Epoch 299/300
 10/470 [.....] - ETA: 0s - loss: 0.2020 - acc: 0.8000
470/470 [=====] - 0s - loss: 0.1313 - acc: 0.8638
Epoch 300/300
 10/470 [.....] - ETA: 0s - loss: 0.2000 - acc: 0.8000
470/470 [=====] - 0s - loss: 0.1313 - acc: 0.8638
32/470 [==>.....] - ETA: 0s
Accuracy: 0.8681

Process finished with exit code 0
```

그림 D-17 실행 성공