Term Project - Part I

Python Installation
&
Deep Learning Environment

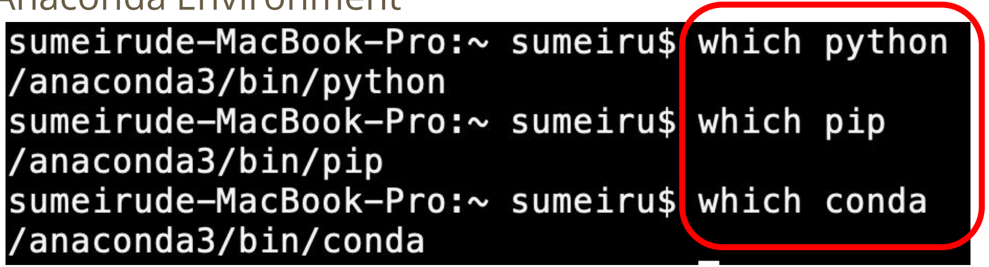
Python

- In this semester, TAs would recommend using Python which is a programming language to do a term project

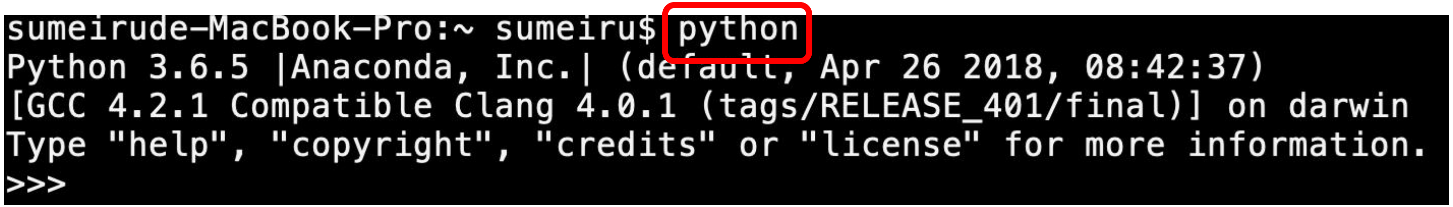
- Before announcing details about term project, please install and self-learn Python

1. How to install Python ?

Python Installation

- Anaconda
  - https://www.anaconda.com/distribution/
  - Test Anaconda Environment
    - 

```
sumeirude-MacBook-Pro:~ sumeiru$ which python
/anaconda3/bin/python
sumeirude-MacBook-Pro:~ sumeiru$ which pip
/anaconda3/bin/pip
sumeirude-MacBook-Pro:~ sumeiru$ which conda
/anaconda3/bin/conda
```

    - 

```
sumeirude-MacBook-Pro:~ sumeiru$ python
Python 3.6.5 |Anaconda, Inc.| (default, Apr 26 2018, 08:42:37)
[GCC 4.2.1 Compatible Clang 4.0.1 (tags/RELEASE_401/final)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

- Miniconda
  - https://docs.conda.io/en/latest/miniconda.html

Command on python

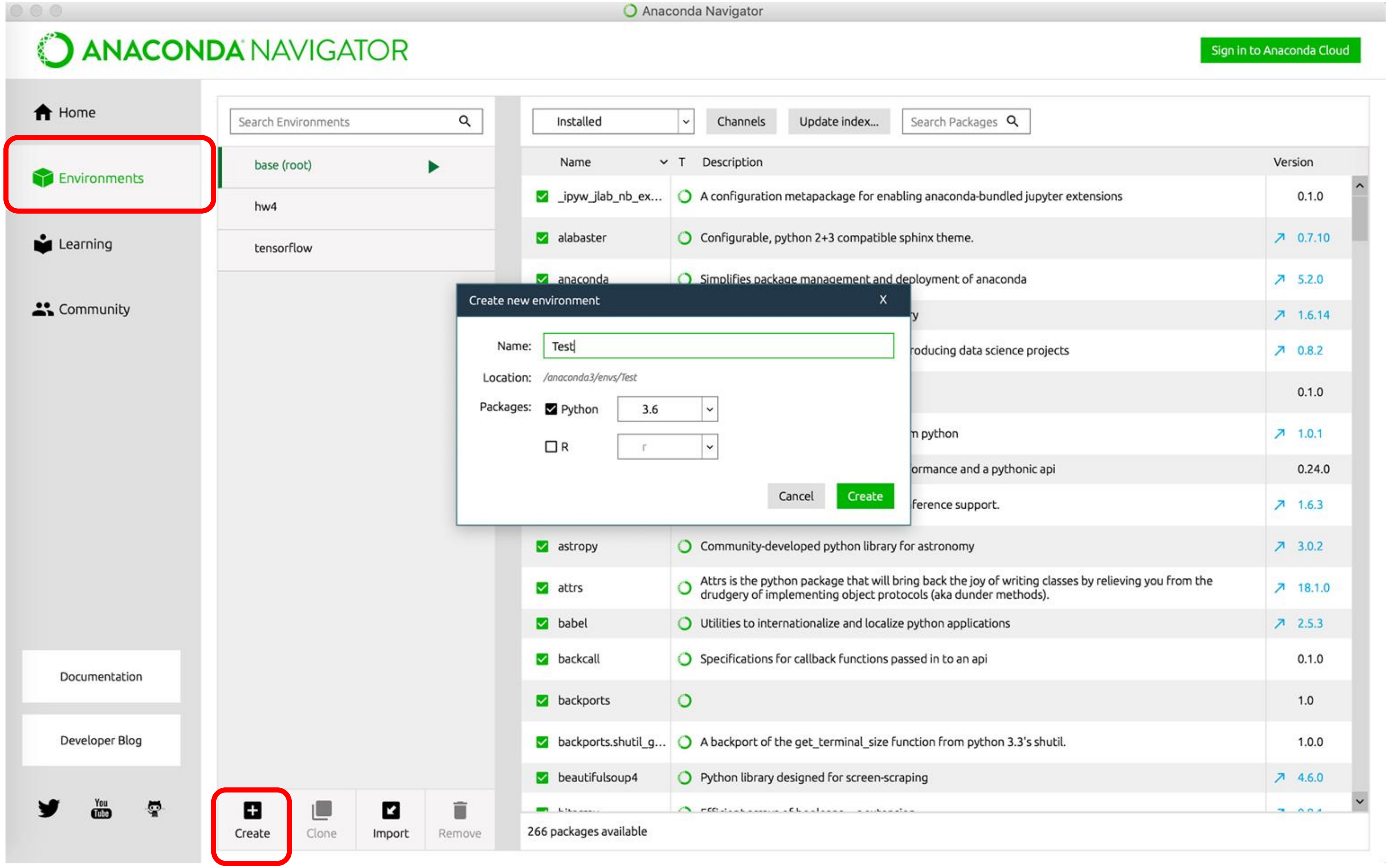
- Go to Terminal (Mac/Linux) or Anaconda Prompt (Windows)

- Install Package
  - pip install {package name}
  - conda install {package name}

- Check how many packages you have
  - pip freeze

Virtual Environment

- Develop projects
    - Manage various needs effectively
    - Avoid packages version collision



base (root)
hw4
tensorflow

Name	T	Description	Version
☑ _ipyw_jlab_nb_ex...	○	A configuration metapackage for enabling anaconda-bundled jupyter extensions	0.1.0
☑ alabaster	○	Configurable, python 2+3 compatible sphinx theme.	↗ 0.7.10
☑ anaconda	○	Simplifies package management and deployment of anaconda	↗ 5.2.0
		y	↗ 1.6.14
		roducing data science projects	↗ 0.8.2
			0.1.0
		n python	↗ 1.0.1
		ormance and a pythonic api	0.24.0
		ference support.	↗ 1.6.3
☑ astropy	○	Community-developed python library for astronomy	↗ 3.0.2
☑ attrs	○	Attrs is the python package that will bring back the joy of writing classes by relieving you from the drudgery of implementing object protocols (aka dunder methods).	↗ 18.1.0
☑ babel	○	Utilities to internationalize and localize python applications	↗ 2.5.3
☑ backcall	○	Specifications for callback functions passed in to an api	0.1.0
☑ backports	○		1.0
☑ backports.shutil_g...	○	A backport of the get_terminal_size function from python 3.3's shutil.	1.0.0
☑ beautifulsoup4	○	Python library designed for screen-scraping	↗ 4.6.0

266 packages available

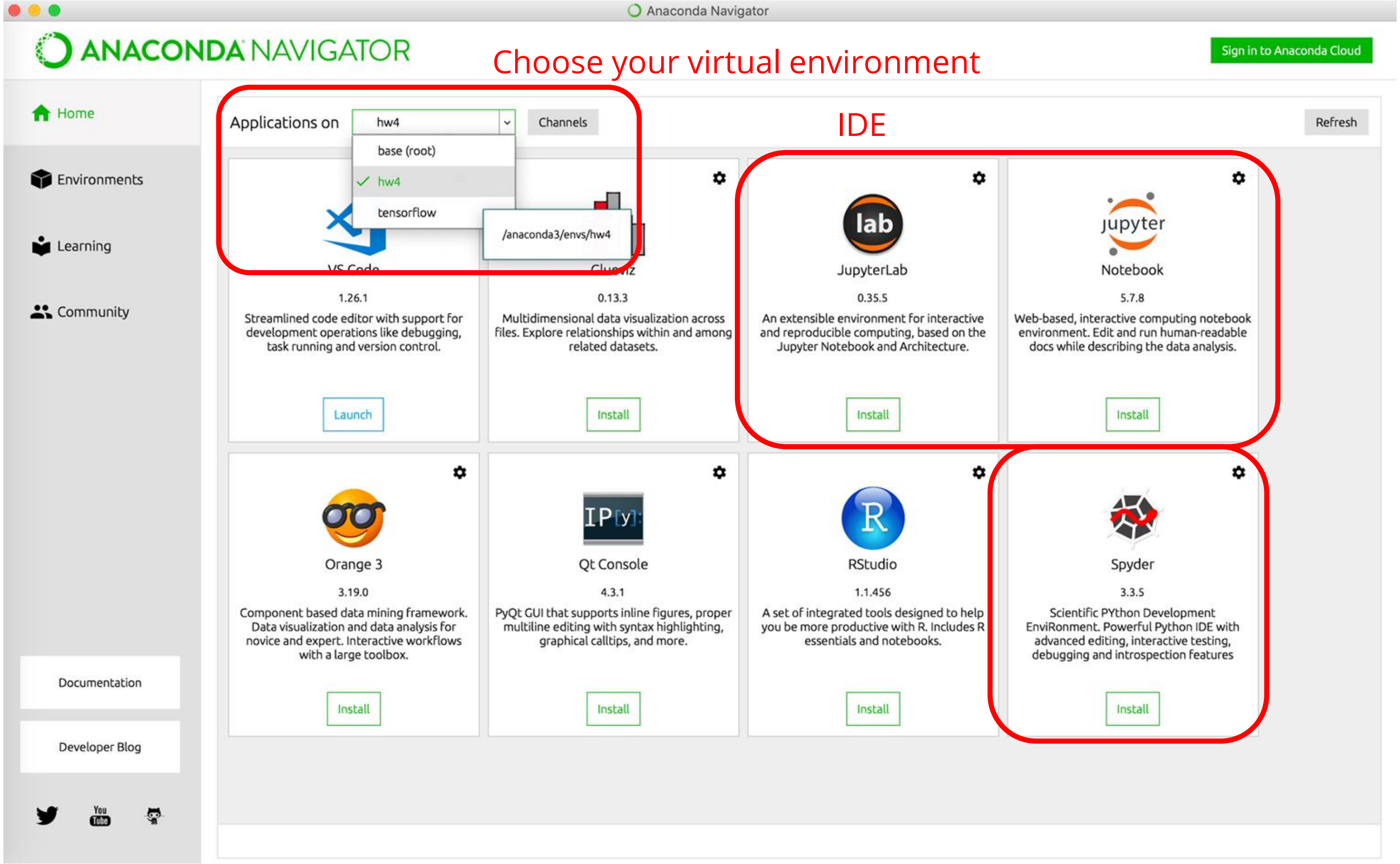
Create new environment X

Name: Test

Location: /anaconda3/envs/Test

Packages: ☑ Python 3.6

□ R r



Choose your virtual environment

IDE

VS Code 1.26.1
Streamlined code editor with support for development operations like debugging, task running and version control.

Glueviz 0.13.3
Multidimensional data visualization across files. Explore relationships within and among related datasets.

JupyterLab 0.35.5
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Notebook 5.7.8
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Orange 3 3.19.0
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

Qt Console 4.3.1
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

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

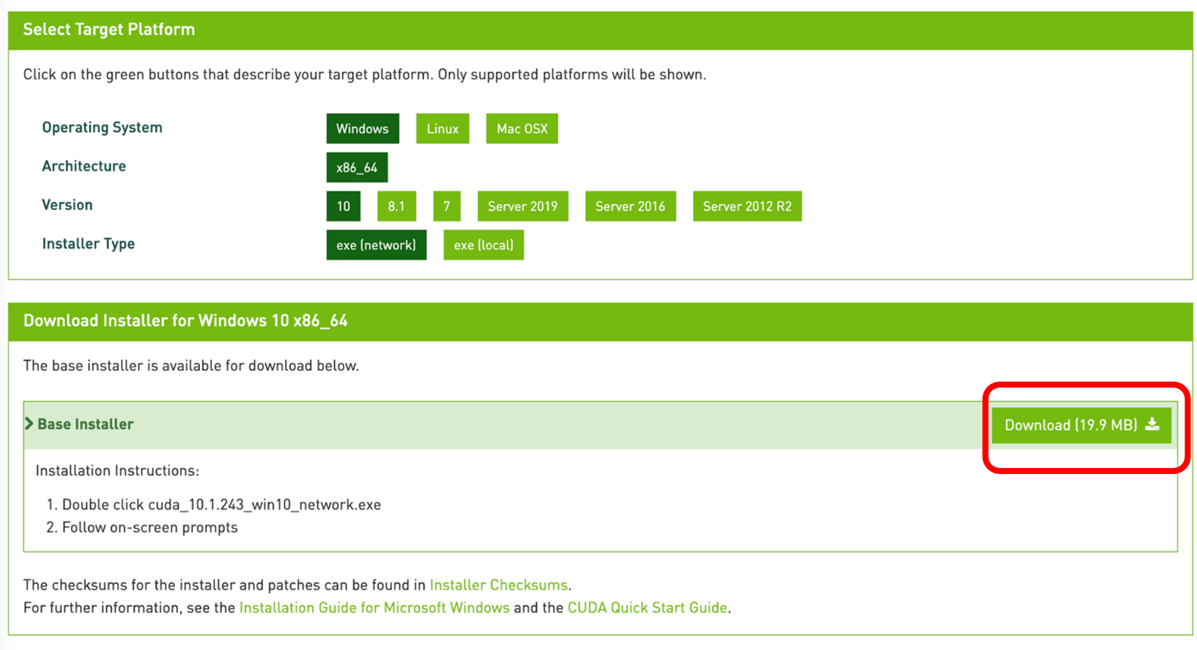
Spyder 3.3.5
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

2. How to create deep learning environment ?

2-1. If you have your own GPU

Download Cuda

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



Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown.

Operating System [✓ Windows] Linux Mac OSX

Architecture [✓ x86_64]

Version [✓ 10] 8.1 7 Server 2019 Server 2016 Server 2012 R2

Installer Type [✓ exe (network)] exe (local)

Download Installer for Windows 10 x86_64

The base installer is available for download below.

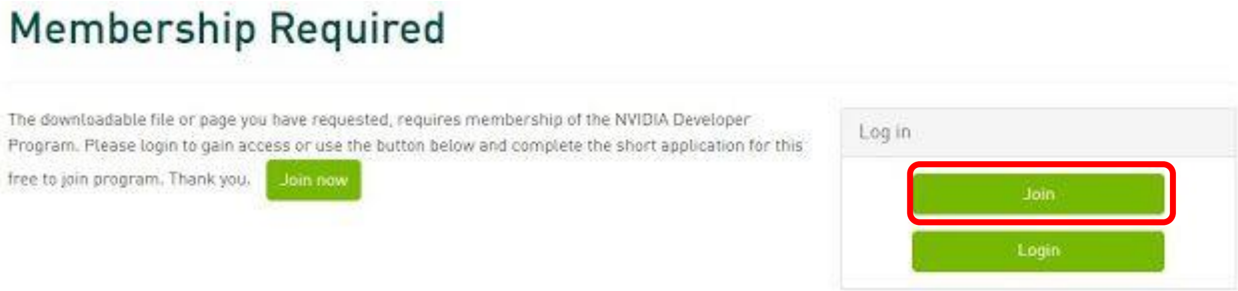
Installation Instructions:

1. Double click cuda_10.1.243_win10_network.exe
2. Follow on-screen prompts

The checksums for the installer and patches can be found in Installer Checksums.
For further information, see the Installation Guide for Microsoft Windows and the CUDA Quick Start Guide.

Download Cudnn

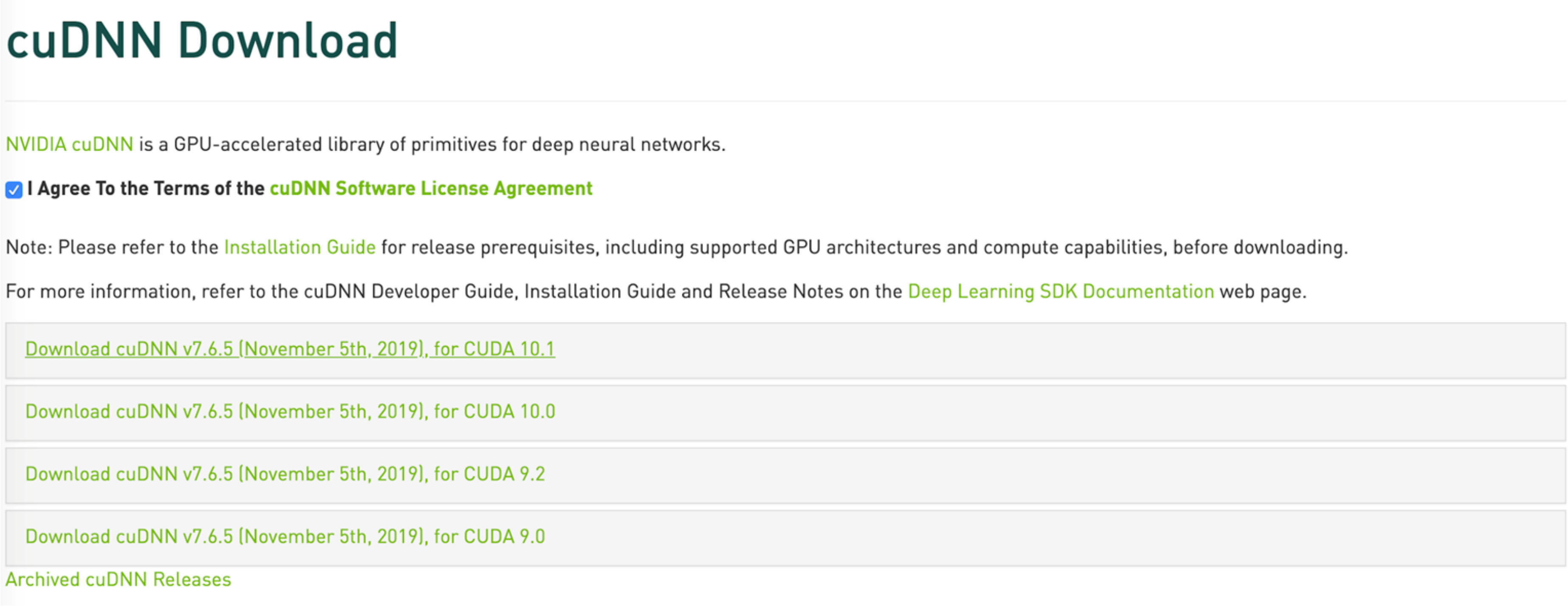
- https://developer.nvidia.com/cudnn
- Register an account



Membership Required

The downloadable file or page you have requested, requires membership of the NVIDIA Developer Program. Please login to gain access or use the button below and complete the short application for this free to join program. Thank you.

- Go to E-mail to activate your account



cuDNN Download

NVIDIA cuDNN is a GPU-accelerated library of primitives for deep neural networks.

☑ **I Agree To the Terms of the cuDNN Software License Agreement**

Note: Please refer to the Installation Guide for release prerequisites, including supported GPU architectures and compute capabilities, before downloading.

For more information, refer to the cuDNN Developer Guide, Installation Guide and Release Notes on the Deep Learning SDK Documentation web page.

Download cuDNN v7.6.5 (November 5th, 2019), for CUDA 10.1

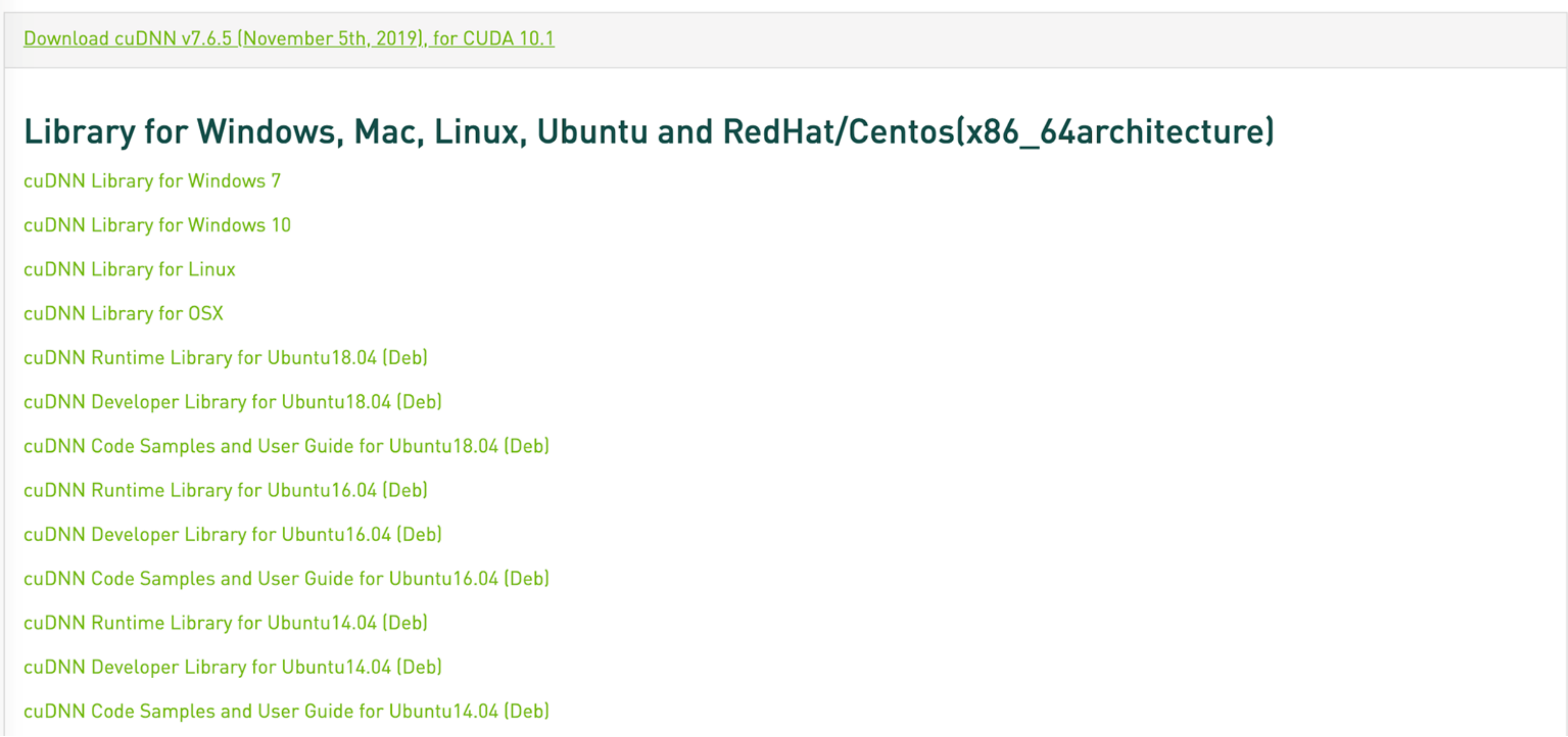
Download cuDNN v7.6.5 (November 5th, 2019), for CUDA 10.0

Download cuDNN v7.6.5 (November 5th, 2019), for CUDA 9.2

Download cuDNN v7.6.5 (November 5th, 2019), for CUDA 9.0

Archived cuDNN Releases

- Download cudnn for your CUDA version



Download cuDNN v7.6.5 (November 5th, 2019), for CUDA 10.1

Library for Windows, Mac, Linux, Ubuntu and RedHat/Centos(x86_64architecture)

cuDNN Library for Windows 7

cuDNN Library for Windows 10

cuDNN Library for Linux

cuDNN Library for OSX

cuDNN Runtime Library for Ubuntu18.04 (Deb)

cuDNN Developer Library for Ubuntu18.04 (Deb)

cuDNN Code Samples and User Guide for Ubuntu18.04 (Deb)

cuDNN Runtime Library for Ubuntu16.04 (Deb)

cuDNN Developer Library for Ubuntu16.04 (Deb)

cuDNN Code Samples and User Guide for Ubuntu16.04 (Deb)

cuDNN Runtime Library for Ubuntu14.04 (Deb)

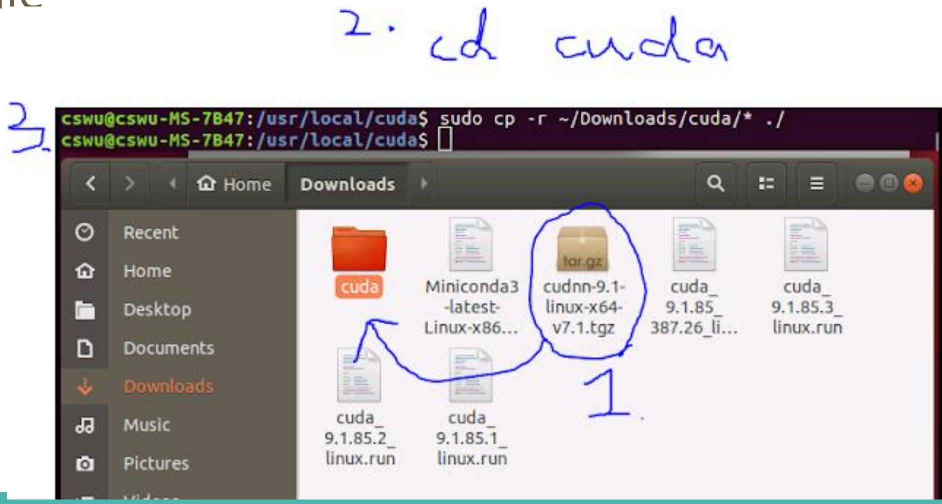
cuDNN Developer Library for Ubuntu14.04 (Deb)

cuDNN Code Samples and User Guide for Ubuntu14.04 (Deb)

Cudnn

Cudnn is a compressed file

1. Unzip the Cudnn file
2. Find the Cuda file (previously downloaded)
3. Put Cudnn unzipped file into Cuda file



2. cd cuda

3.

```
cswu@cswu-MS-7B47:/usr/local/cuda$ sudo cp -r ~/Downloads/cuda/* ./
cswu@cswu-MS-7B47:/usr/local/cuda$ 
```

1.

PYTORCH

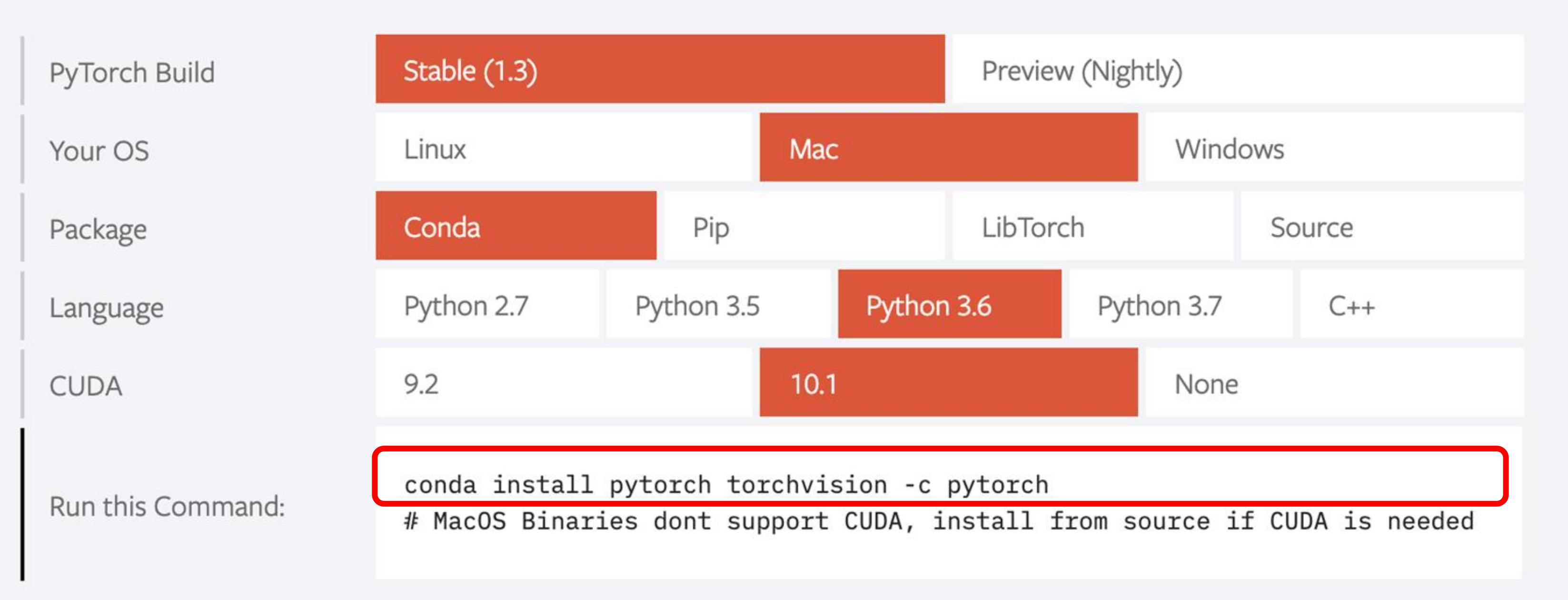
There are a lot of deep learning packages, such as Tensorflow, pytorch, and so on.

TensorFlow

K Keras

Deep Learning Package - Pytorch

- https://pytorch.org/



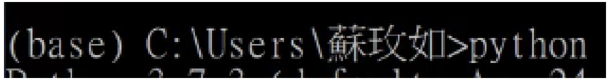
PyTorch Build	[✓ Stable (1.3)]	Preview (Nightly)			
Your OS	Linux	[✓ Mac]	Windows		
Package	[✓ Conda]	Pip	LibTorch	Source	
Language	Python 2.7	Python 3.5	[✓ Python 3.6]	Python 3.7	C++
CUDA	9.2	[✓ 10.1]	None		

Run this Command:

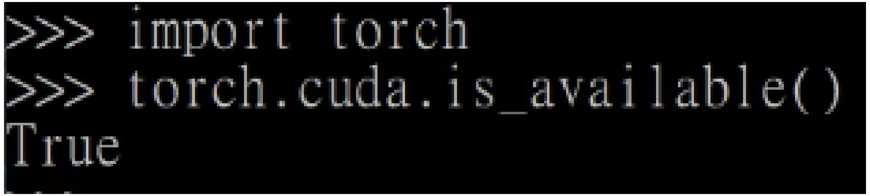
```
conda install pytorch torchvision -c pytorch
# MacOS Binaries dont support CUDA, install from source if CUDA is needed
```

Test Pytorch

- Go to your Power Shell / cmd / Terminal



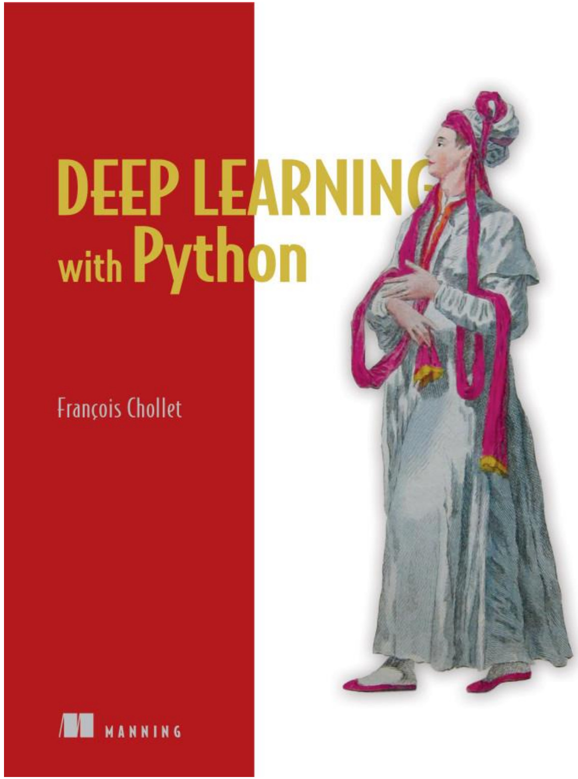
```
(base) C:\Users\蘇玫如>python
```



```
>>> import torch
>>> torch.cuda.is_available()
True
```

If output is true, it represents your GPU is loaded

Deep Learning Package - Tensorflow & Keras



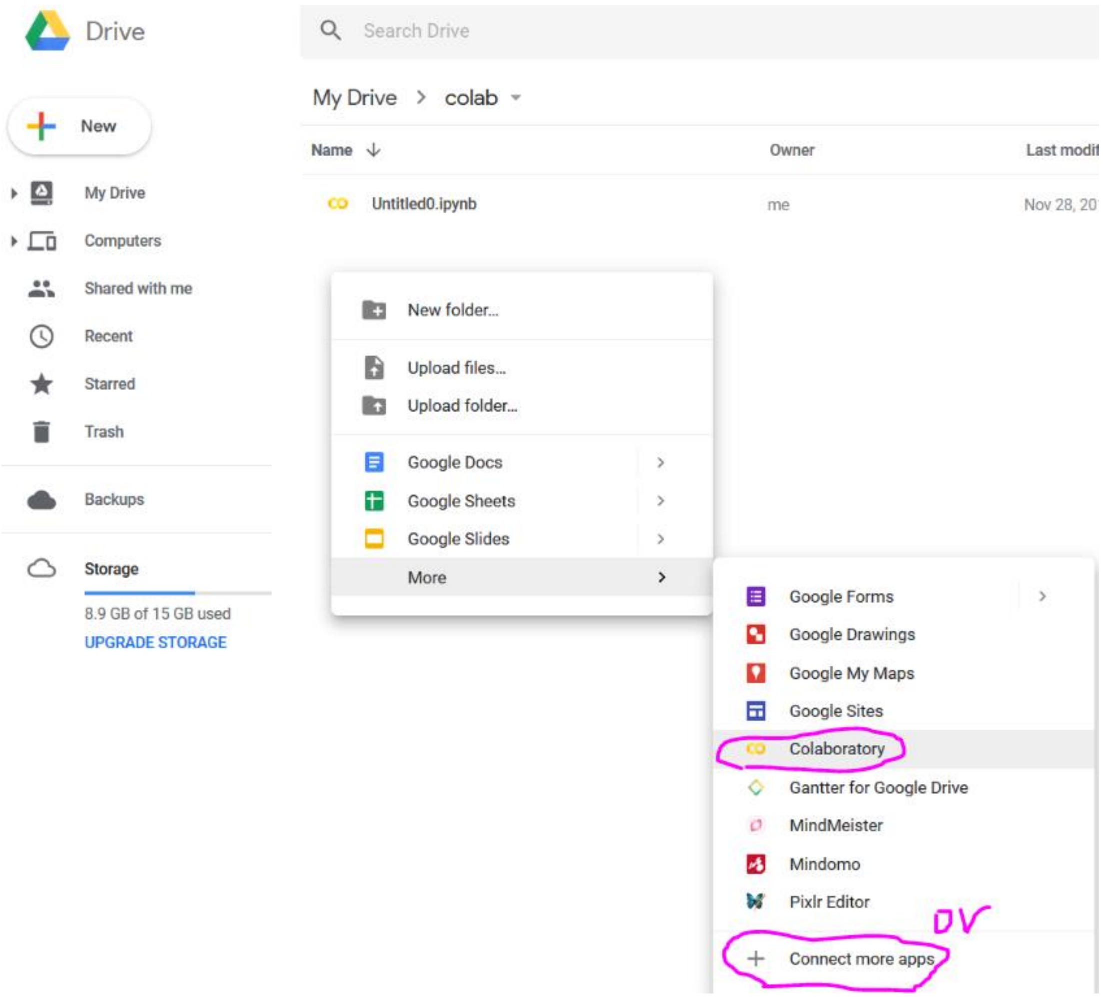
You can do your term project from now on

2-2. If you don’t have your own GPU

Google Colab

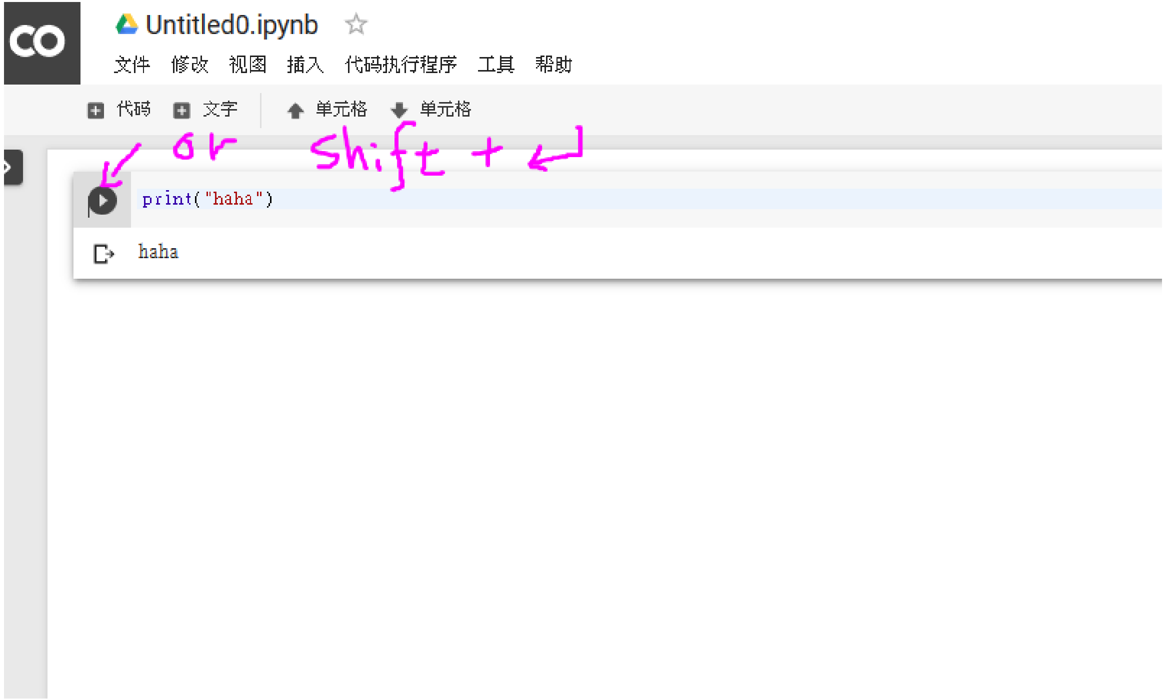
- Login your google drive

- Create a new colab file
    - Right click at empty place, in the menu: More>Colaboratory
    - If not found this option : More>Connect more apps > search and add Colaboratory



Name ↓	Owner	Last modi
Untitled0.ipynb	me	Nov 28, 20

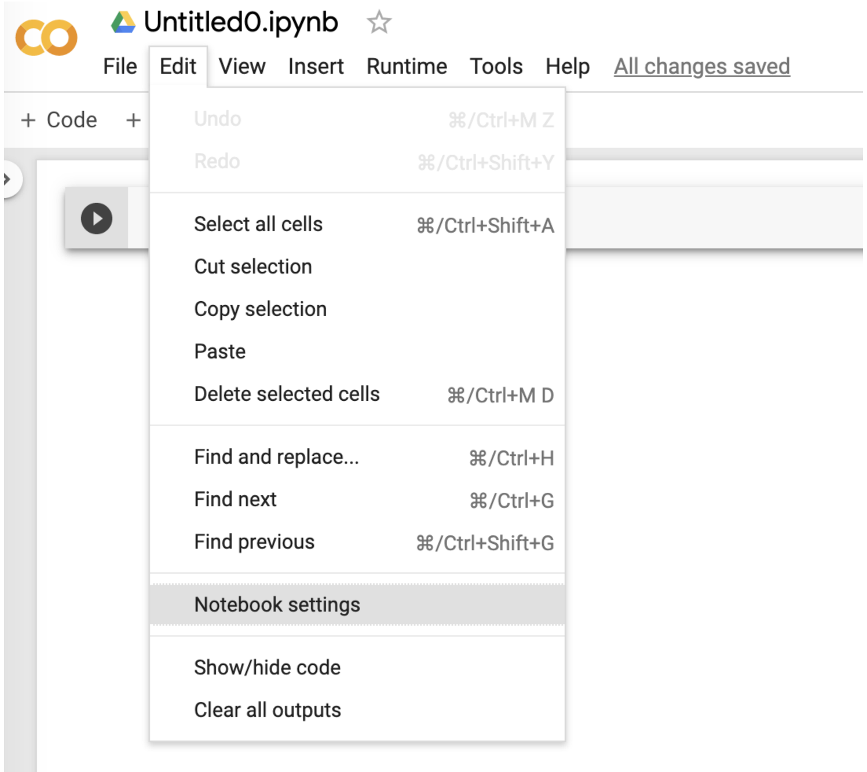
- Here you could input some code, then press the play button or shift+enter to execute the code.
- You could just consider it as a cloud jupyter notebook with tensorflow in your google drive.

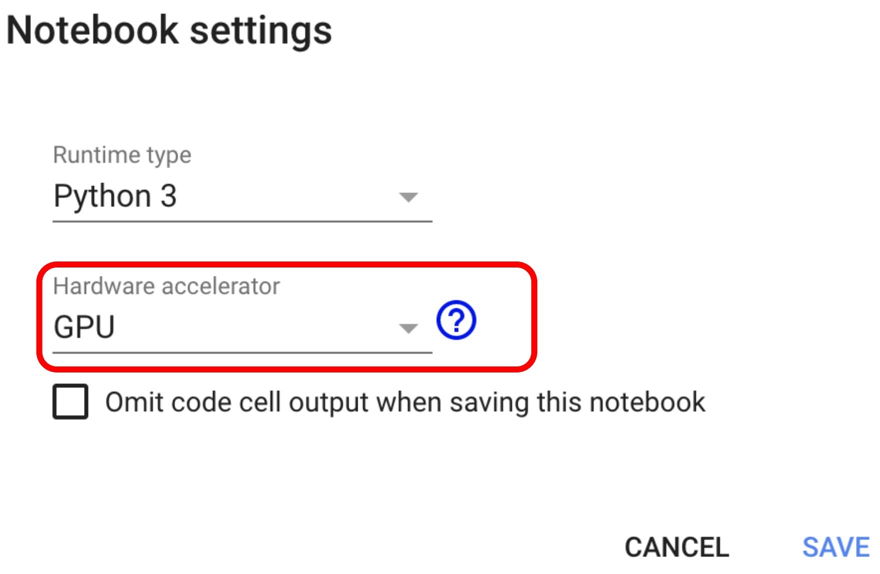


```
print("haha")
```

```
haha
```

Select GPU to Train Model




Notebook settings

Runtime type: Python 3

Hardware accelerator: GPU

□ Omit code cell output when saving this notebook

Tensorflow Demo on Colab

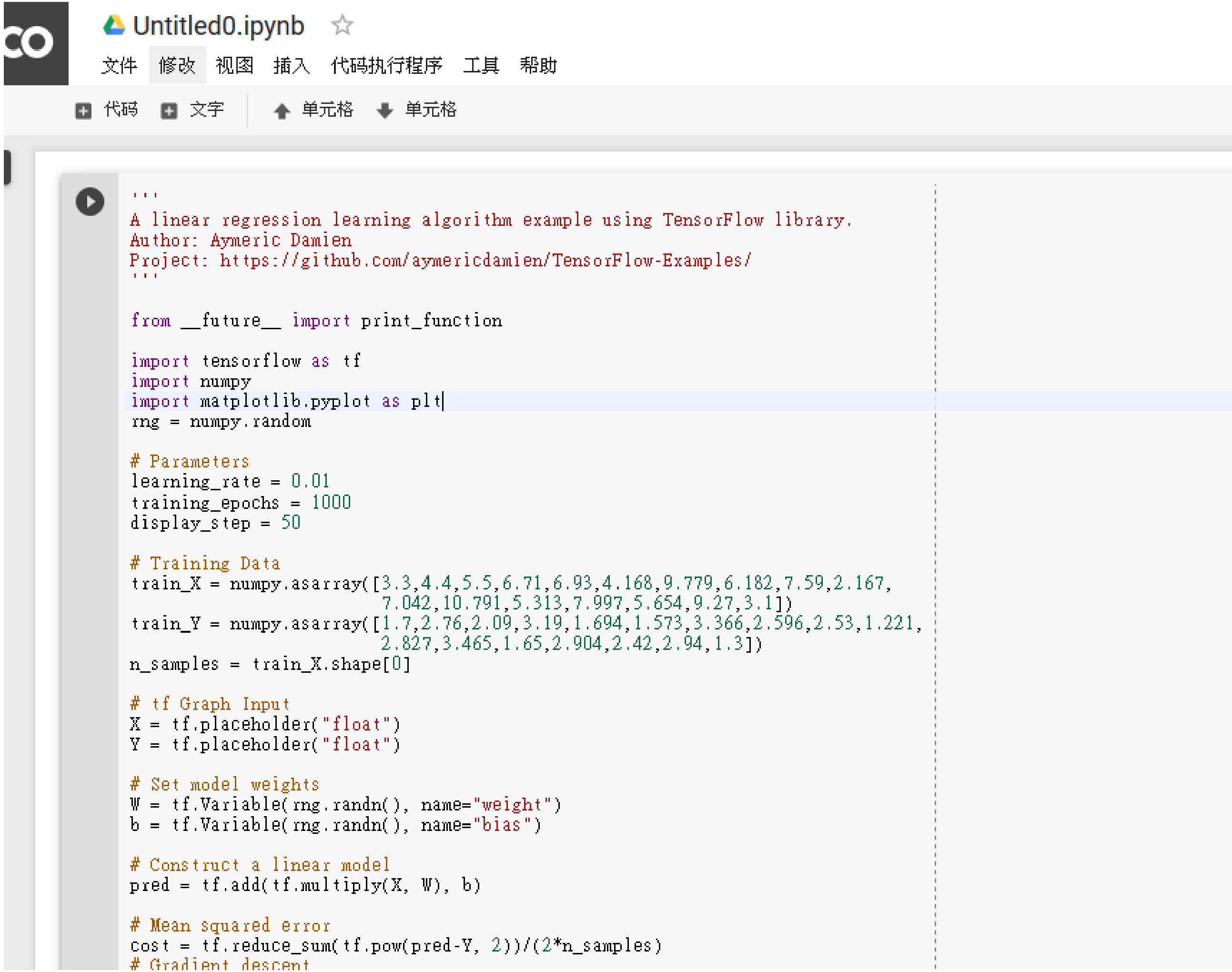
- a good example to let you study how to visualize the result of your model

1. Copy the test example code from here

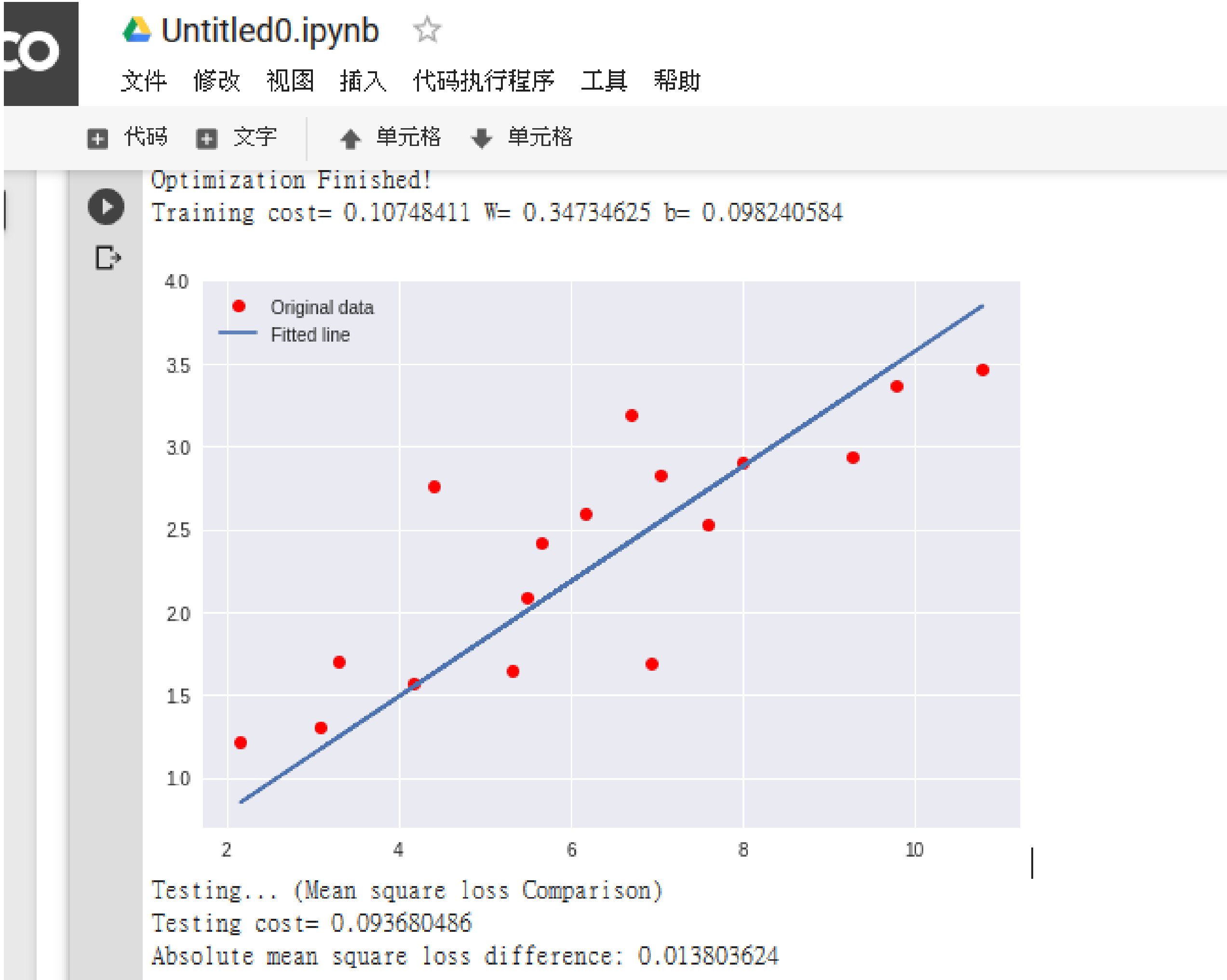
This is a simple linear regression model optimized by gradient descent.

2. Paste the code to an empty colab notebook and execute.

Notice that tensorflow, numpy and matplotlib are all pre-installed in colab.



```
'''
A linear regression learning algorithm example using TensorFlow library.
Author: Aymeric Damien
Project: https://github.com/aymericdamien/TensorFlow-Examples/
'''

from __future__ import print_function

import tensorflow as tf
import numpy
import matplotlib.pyplot as plt
rng = numpy.random

# Parameters
learning_rate = 0.01
training_epochs = 1000
display_step = 50

# Training Data
train_X = numpy.asarray([3.3,4.4,5.5,6.71,6.93,4.168,9.779,6.182,7.59,2.167,
                         7.042,10.791,5.313,7.997,5.654,9.27,3.1])
train_Y = numpy.asarray([1.7,2.76,2.09,3.19,1.694,1.573,3.366,2.596,2.53,1.221,
                         2.827,3.465,1.65,2.904,2.42,2.94,1.3])
n_samples = train_X.shape[0]

# tf Graph Input
X = tf.placeholder("float")
Y = tf.placeholder("float")

# Set model weights
W = tf.Variable(rng.randn(), name="weight")
b = tf.Variable(rng.randn(), name="bias")

# Construct a linear model
pred = tf.add(tf.multiply(X, W), b)

# Mean squared error
cost = tf.reduce_sum(tf.pow(pred-Y, 2))/(2*n_samples)
# Gradient descent
```



```
Optimization Finished!
Training cost= 0.10748411 W= 0.34734625 b= 0.098240584
```

```
Testing... (Mean square loss Comparison)
Testing cost= 0.093680486
Absolute mean square loss difference: 0.013803624
```

3. The output can be easily show by the plot library: matplotlib

Install Pytorch on Colab

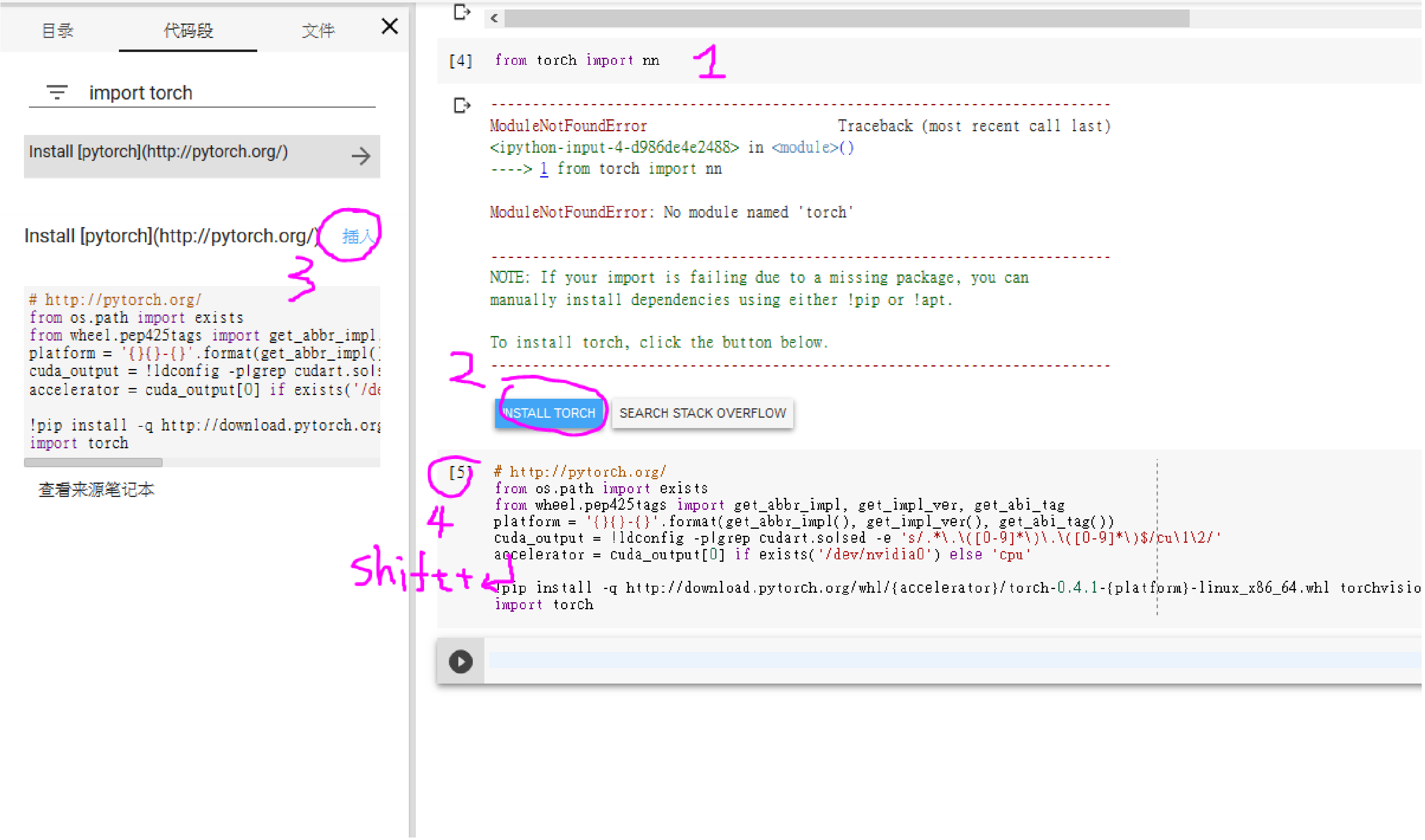
You can use pytorch instead of tensorflow.

1. Import torch directly, then the colab would ask you to install pytorch.

2. Press INSTALL TORCH, then a menu would show up in the left side.

3. Press Insert in the sidemenu, then a new code block would show up

4. Execute the new code block

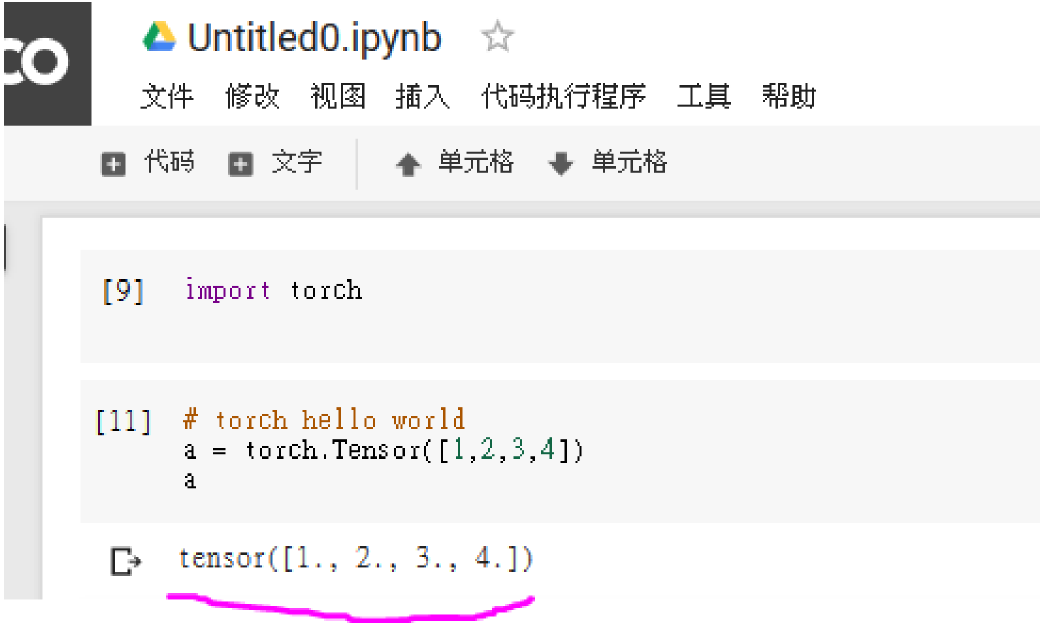


```
# http://pytorch.org/
from os.path import exists
from wheel.pep425tags import get_abbr_impl
platform = '{}{}-{}'.format(get_abbr_impl()
cuda_output = !ldconfig -p|grep cudart.so|s
accelerator = cuda_output[0] if exists('/de

!pip install -q http://download.pytorch.or
import torch
```

```
[4] from torch import nn
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-4-d986de4e2488> in <module>()
----> 1 from torch import nn

ModuleNotFoundError: No module named 'torch'

---------------------------------------------------------------------------
NOTE: If your import is failing due to a missing package, you can
manually install dependencies using either !pip or !apt.

To install torch, click the button below.
---------------------------------------------------------------------------
```

```
[5] # http://pytorch.org/
    from os.path import exists
    from wheel.pep425tags import get_abbr_impl, get_impl_ver, get_abi_tag
    platform = '{}{}-{}'.format(get_abbr_impl(), get_impl_ver(), get_abi_tag())
    cuda_output = !ldconfig -p|grep cudart.so|sed -e 's/.*\.\([0-9]*\)\.\([0-9]*\)$/cu\1\2/'
    accelerator = cuda_output[0] if exists('/dev/nvidia0') else 'cpu'

    !pip install -q http://download.pytorch.org/whl/{accelerator}/torch-0.4.1-{platform}-linux_x86_64.whl torchvisio
    import torch
```



```
[9]  import torch
```

```
[11] # torch hello world
     a = torch.Tensor([1,2,3,4])
     a
```

```
tensor([1., 2., 3., 4.])
```

- Delete all block then try torch.
- If success, “torch hello world” should work

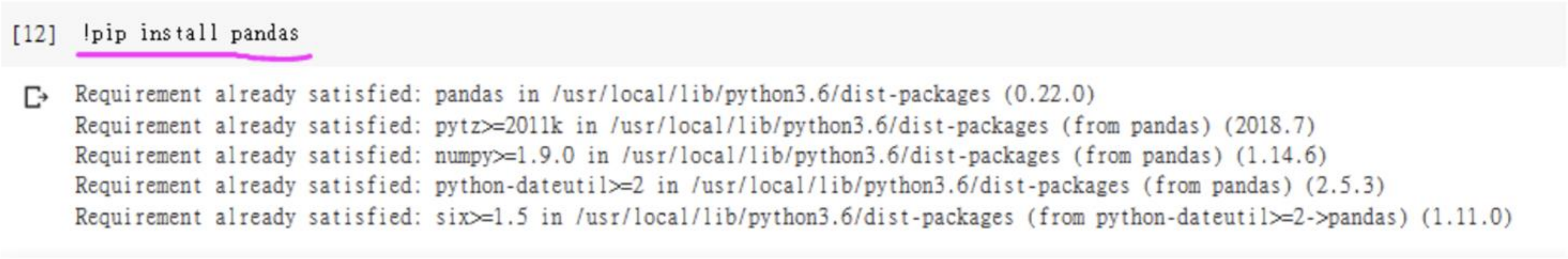
Install other python packages on Colab

just input:

```
!pip install {package name}
```

then execute.

ex. install pandas:



```
[12] !pip install pandas
```

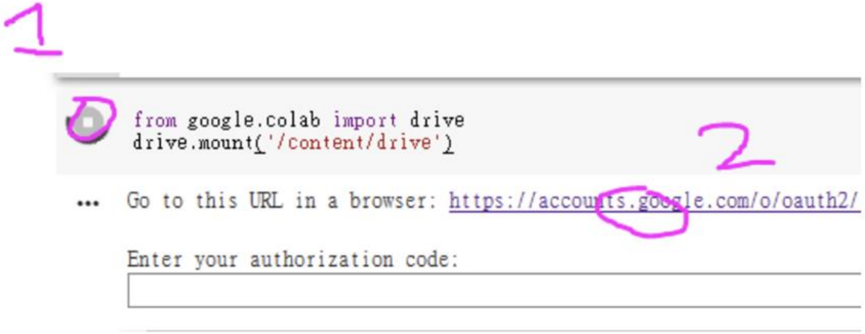
```
Requirement already satisfied: pandas in /usr/local/lib/python3.6/dist-packages (0.22.0)
Requirement already satisfied: pytz>=2011k in /usr/local/lib/python3.6/dist-packages (from pandas) (2018.7)
Requirement already satisfied: numpy>=1.9.0 in /usr/local/lib/python3.6/dist-packages (from pandas) (1.14.6)
Requirement already satisfied: python-dateutil>=2 in /usr/local/lib/python3.6/dist-packages (from pandas) (2.5.3)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.6/dist-packages (from python-dateutil>=2->pandas) (1.11.0)
```

Mount google drive folder to colab

1. Input this and execute

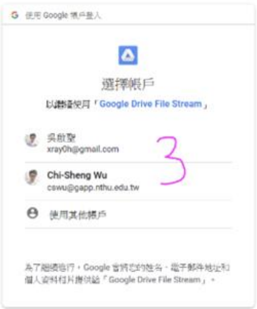
```
from google.colab import drive
drive.mount('/content/drive')
```

2.~4. follow the picture



```
from google.colab import drive
drive.mount('/content/drive')
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/

Enter your authorization code:
```



使用 Google 帳戶登入

選擇帳戶

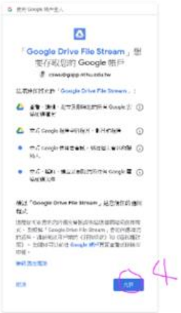
以繼續使用「Google Drive File Stream」

吳啟聖
xray0h@gmail.com

Chi-Sheng Wu
cswu@gapp.nthu.edu.tw

使用其他帳戶

為了繼續進行，Google 會將您的姓名、電子郵件地址和個人資料相片提供給「Google Drive File Stream」。



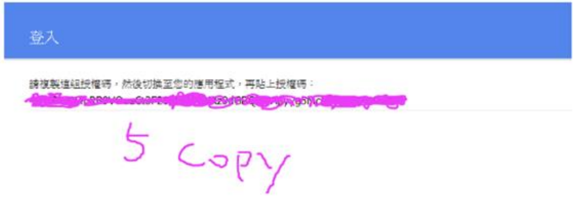
使用 Google 帳戶登入

「Google Drive File Stream」想要存取您的 Google 帳戶

cswu@gapp.nthu.edu.tw

[illegible]

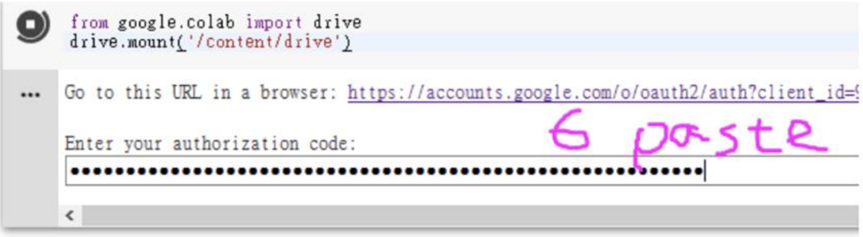
5.&6. **Paste auth code** to colab and **press enter**



登入

請複製這組授權碼，然後切換至您的應用程式，再貼上授權碼：

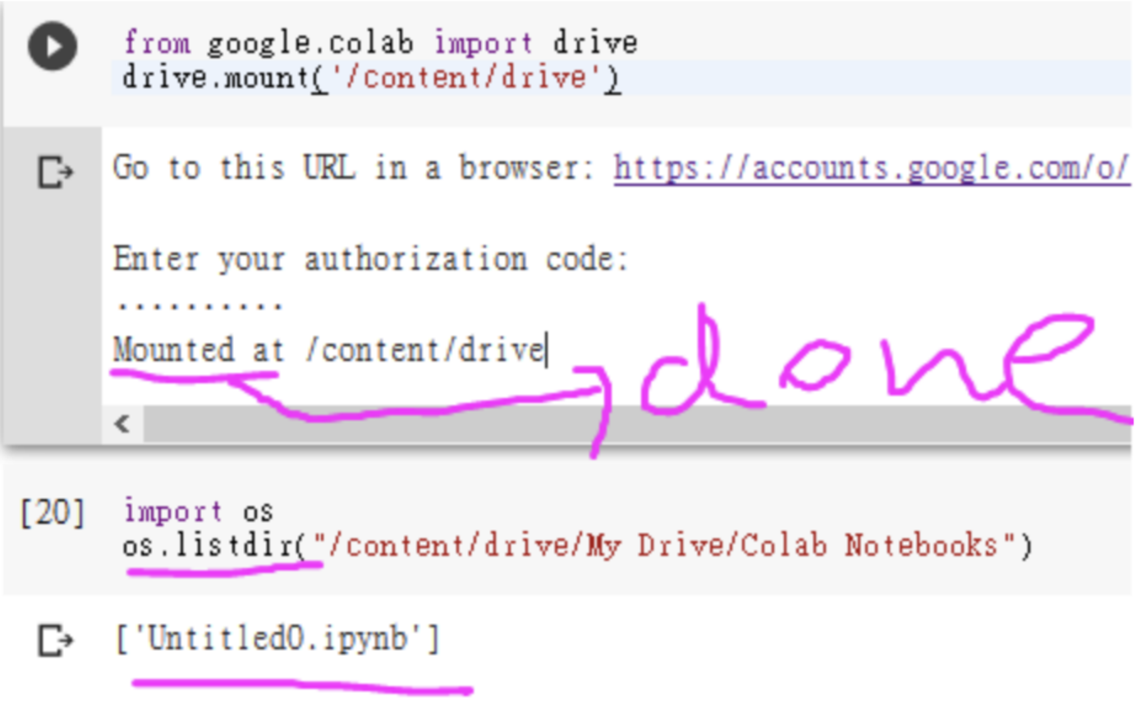
[redacted]



```
from google.colab import drive
drive.mount('/content/drive')
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=

Enter your authorization code:
••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
```

7. Test the mounted google drive folder by **os.listdir()**
   I just show the files in 'Colab Notebooks' here. You could find everything in **'/content/drive/My Drive'.**



```
from google.colab import drive
drive.mount('/content/drive')
```

```
Go to this URL in a browser: https://accounts.google.com/o/

Enter your authorization code:
..........
Mounted at /content/drive
```

done

```
[20] import os
     os.listdir("/content/drive/My Drive/Colab Notebooks")
```

```
['Untitled0.ipynb']
```

You can do your term project from now on

3. Self-learn Python

Checklist

- Make sure that your team could do all jobs in the checklists before TAs announce the term project
- https://paper.dropbox.com/doc/practice-of-final-project--AxvG4ZHbneVrAU0O2yZq~zqtAQ-2GNV5YSCyFXTJA96Q7PUS