Colab Introduction & Benefit

Google Colab - What is Google Colab?

If you have used **Jupyter** notebook previously, you would quickly learn to use Google Colab. To be precise, Colab is a free Jupyter notebook environment that runs entirely in the cloud. Most importantly, it does not require a setup and the notebooks that you create can be simultaneously edited by your team members - just the way you edit documents in Google Docs. Colab supports many popular machine learning libraries which can be easily loaded in your notebook.

What Colab Offers You?

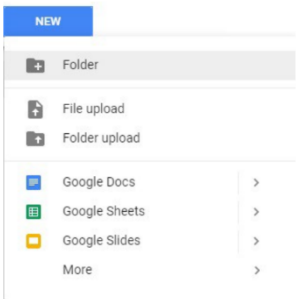
As a programmer, you can perform the following using Google Colab.

- Write and execute code in Python
- Document your code that supports mathematical equations
- Create/Upload/Share notebooks
- Import/Save notebooks from/to Google Drive
- Import/Publish notebooks from GitHub
- Import external datasets e.g. from Kaggle
- Integrate PyTorch, TensorFlow, Keras, OpenCV
- Free Cloud service with free GPU

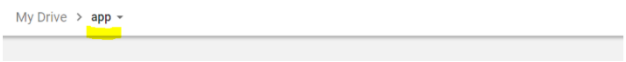
Creating and Renaming Notebook

Getting Google Colab Ready to Use

Creating Folder on Google Drive

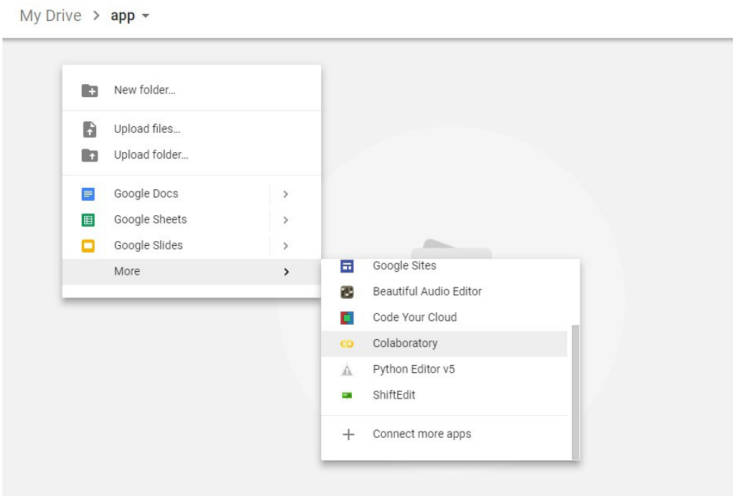


Since **Colab** is working on your own **Google Drive**, we first need to specify the folder we'll work. I created a folder named "**app**" on my **Google Drive**. Of course, you can use a different name or choose the default **Colab Notebooks** folder instead of **app folder**.

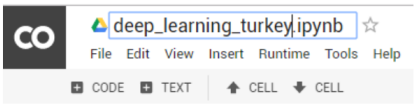


Creating New Colab Notebook

Create a new notebook via **Right click > More > Colaboratory**



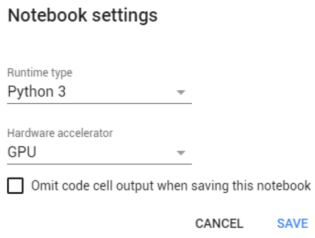
Rename notebook by means of clicking the file name.



Setting UP GPU

Setting Free GPU

It is so simple to alter default hardware **(CPU to GPU or vice versa)**; just follow **Edit > Notebook settings** or **Runtime>Change runtime type** and **select GPU** as **Hardware accelerator**.



Checking if notebook is running on GPU backend

```
[110] import tensorflow as tf

      device_name = tf.test.gpu_device_name()
      if device_name != '/device:GPU:0':
        raise SystemError('GPU device not found')
      print('Found GPU at: {}'.format(device_name))
```

```
Found GPU at: /device:GPU:0
```

Running cells

I will run some **Basic Data Types** codes from Python Numpy Tutorial.

Mount Google Drive to Colab

When you run the code above, you should see a result like this:

Click the link, **copy** verification code and **paste** it to text box.

Colab Directory

```
[ ]  # Paths

     Colab directory -   '/content/'
     files_folder_drive_uploaded_or_mounted_on colab -  '/content/file_folder_drive'
```

Colab Shell Commands

```
In [14]: %cd ..
/home/jake/projects
```

```
#View contents of directory
!ls # for current directory
!ls path #for specified directory

#Make new directory
!mkdir dir_name #to make new dir. in current directory
!mkdir path/dir_name

#Remove/Delete
!rm -r dir_name
!rm -r path/dir_name

!rm file
!rm path/file
```

Download and Unzip Zip files

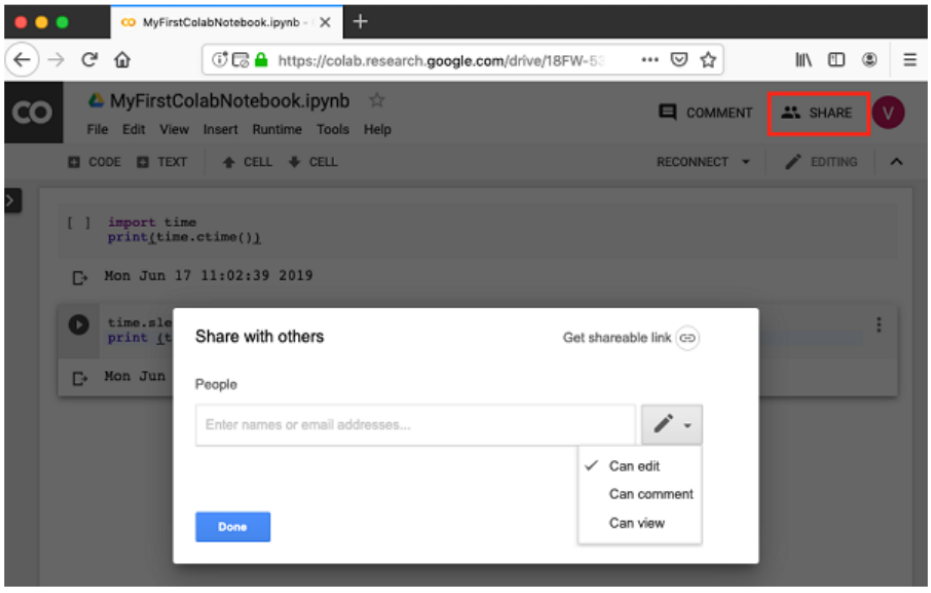
If you'd rather download a shared zip file link, you can use:

```
!wget
!unzip
```

For example:

```
!wget -cq https://s3.amazonaws.com/content.udacity-data.com/courses/nd188/flower_data.zip
!unzip -qq flower_data.zip
```

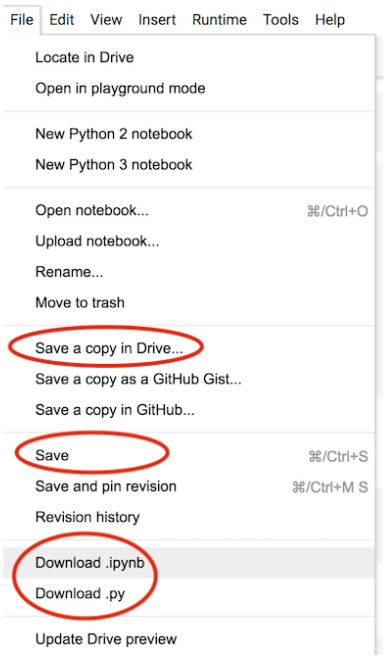
Sharing Notebook



Saving to Google Drive/Git Hub

Always be saving

Saving your work is simple! You can do a good ol' "command-s" or drop the "File" menu down to save. You can create a copy of your notebook by dropping "File" -> "Save a Copy in Drive." You can also download your workbook by going from "File" -> "download .ipyb" or "download .py."



Clone & Run Git Hub Repo

Cloning Github Repo to Google Colab

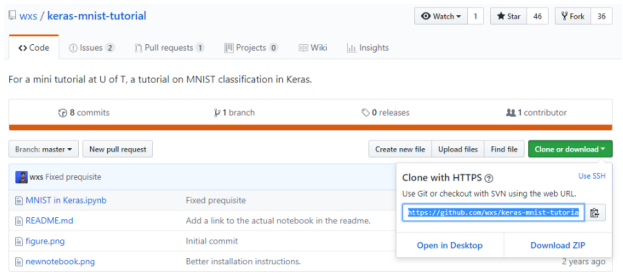
It is easy to clone a Github repo with Git.

Step 1: Find the Github Repo and Get "Git" Link

Find any Github repo to use.

For instance: https://github.com/wxs/keras-mnist-tutorial

Clone or download > Copy the link!



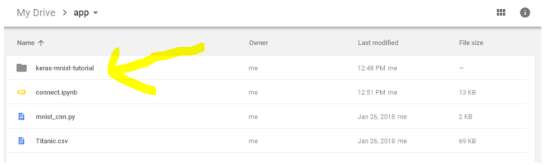
2. Git Clone

Simply run:

```
!git clone https://github.com/wxs/keras-mnist-tutorial.git
```

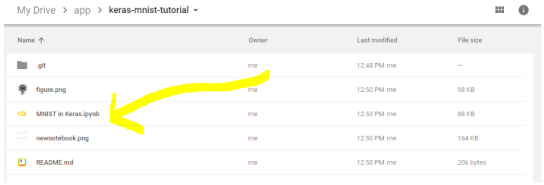
3. Open the Folder in Google Drive

Folder has the same with the Github repo of course :)



4. Open The Notebook

Right Click > Open With > Colaboratory



5. Run

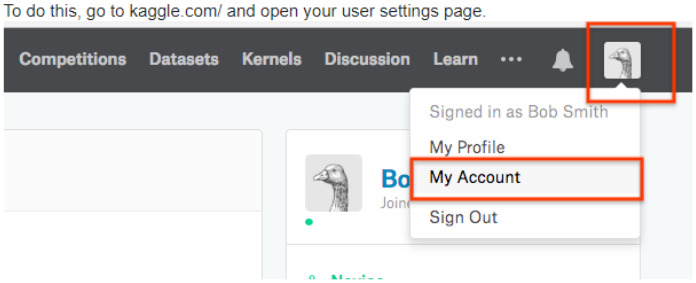
Now you are able to run Github repo in Google Colab.

Use Kaggle Datasets In Colab

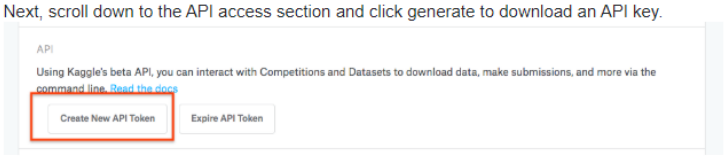
Step-by-step --

1. Create an API key in Kaggle.

   To do this, go to kaggle.com/ and open your user settings page.



2. Next, scroll down to the API access section and click generate to download an API key.



   This will download a file called `kaggle.json` to your computer. You'll use this file in Colab to access Kaggle datasets and competitions.

3. Navigate to https://colab.research.google.com/.
4. Upload your `kaggle.json` file using the following snippet in a code cell:

```
from google.colab import files
files.upload()
```

5. Install the kaggle API using `!pip install -q kaggle`
6. Move the `kaggle.json` file into `~/.kaggle`, which is where the API client expects your token to be located:

```
!mkdir -p ~/.kaggle
!cp kaggle.json ~/.kaggle/
```

7. Now you can access datasets using the client, e.g., `!kaggle datasets list`.

References

https://www.tutorialspoint.com/google_colab/google_colab_quick_guide.htm

https://towardsdatascience.com/getting-started-with-google-colab-f2fff97f594c

https://colab.research.google.com/drive/1l09j_Yv3H016EqHyrJUe_0mNah1M80qf#scrollTo=5xlf41nj04Cc

https://medium.com/deep-learning-turkey/google-colab-free-gpu-tutorial-e113627b9f5d

https://stackoverflow.com/questions/49310470/using-kaggle-datasets-in-google-colab