Tutorial of Colab

Chaojian Li

January 2019

1 Why Colab

- Free GPU/TPU and No environment pre-setting
- Share with team members via Google Drive easily

2 Before start: Jupyter Notebook

Colab is based on Jupyter Notebook, an open-source web application that allows you to create documents contain live code, equations, visualization and narrative text. As it is based on a server-client structure, it can run on a server and be visited by other users remotely in their own browsers. Although it supports up to 40 programming languages, people mainly use Python with it, which was used to be called IPython Notebook, especially using to write tutorial and course project with live code, so others can read the documents and run the code step by step at the same time.

If you already have pip installed, you can use this to install Jupyter Notebook:

```
pip install jupyter
```

If you use Anaconda, Jupyter Notebook should already be installed. Simply enter this in your terminal, you will start it.

```
jupyter notebook
```

3 Usage in Public Server

3.1 New Library Installation

3.1.1 apt

If you want a new library from apt source, you can add this before your program code:

```
!apt-get install # your lib name#
```

Example:

```
!apt-get -qq install python-cartopy python3-cartopy
import cartopy
```

3.1.2 pip

If you want a new library from pip source, you can add this before your program code:

```
!pip install #your lib name#
```

Example:

```
!pip install -q matplotlib-venn

from matplotlib_venn import venn2
_ = venn2(subsets = (3, 2, 1))
```

3.2 Running traditional python code

Many Deep learning researchers don't use Jupyter Notebook, so their code cannot run on Colab directly as .ipynb file. But we do be able to run those traditional python program on Colab.

As Colab is provided by Google and can interact with Google Drive actively. So we can upload the directory to our Google Drive and mount it in the Colab's virtual machine. Here is an example.

3.3 Upload the project directory

Here we choose a project in Github, an exercise for Tsinghua lecture: Neuromorphic Computing Theory and System. We first download it as a .zip file in our local machine and unzip it. Then we open the Google Drive in our browsers.Choose New - Upload directory, and choose the project directory.

3.4 Check GPU in Colab

Colab provides both free GPU(K80) and TPU. As many open-source deep learning code are based on GPU, so here we check GPU for example.

First, we should choose Edit - Notebook settings to set python version and GPU usage. One account can only use one 11G K80, so only gpu:0 is accessible. Run the code below

```
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))
```

If we see:

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

That means we are able use GPU from Google now.

3.5 Mount Google Drive in Colab

As we want to load and save data in Colab, it is not convient to change files in the virtual machine's file system, so we can mount our Google Drive in the runtime's virtual machine. In this way, we can easily change the files via Google Drive.

We can run the code below to do this.

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

During its running, we will be asked to authorize the mounting by Google Drive. We can run the code below to check if we mount it successfully.

```
with open('/content/gdrive/My Drive/foo.txt', 'w') as f:
  f.write('Hello Google Drive!')
!cat /content/gdrive/My\ Drive/foo.txt
```

The foo.txt can also be viewed and edited in Google Drive.

3.6 Training with GPU

In this example, we must run some pre-processing scipts for data preparing. Run the code below to prepare mnist and fashion-mnist.

- Notice: the code below cannot run in the same code block, they should be in separate blocks.

```
cd /content/gdrive/My Drive/CNN4lecture-master/data/mnist
```

```
!python download_and_convert_mnist.py
```

```
cd /content/gdrive/My Drive/CNN4lecture-master/data/fashion
```

```
!python download_and_convert_fashion.py
```

The downloaded datasets can be viewed in Google Drive. Then we can train it.

```
cd /content/gdrive/My Drive/CNN4lecture-master/source
```

```
!python top.py
```

The output information is the same as we run in our local machine. And Colab supports working after the browser tab closed. We can reconnect to the notebook and view the running program.

4 Usage in Local Machine/ Private Server

Although it is easy can convenient to use GPU from Google in Colab, but we are limited to use a single K80. And many people are trying to run their code using these free GPU/TPU, so the same project may run faster in our laptops' GPU. If we have a powerful local machine or private server, we can follow this instruction to set up a Colab server. It is nearly the same as setting up a remote Jupyter Notebook Server via SSH Tunneling.

5 Conclusion and comments

Colab is a very good cloud product to make sure our code will run correctly in our private servers when we only have our laptops(No deep learning environments) because it has installed many deep learning libraries and allows us to install new libraries easily. However, it is obvious that it is not suitable to train a big network with the free GPU/TPU. And if we use our private server to do such a task, there is no difference between the Colab and a remote Jupyter.

So I will choose Colab if I need to finish a course project with others(we can edit the same notebook at the same time) and debug some modules before uploading to my private server. In a nutshell, as Colab is based on Jupyter Notebook, it is extremely suitable for form a report with live code in a single file, but if we want to form a well-organized code base, the traditional method with the auxiliary of Colab may be better.