

How to Setup a Google Colab Notebook

Emmanuel Madrigal

June 9, 2020

Google Colab allows users to execute to write and execute Python code in the browser, with:

- Zero configuration required
- Free access to GPUs
- Easy sharing

1 What is a Jupyter Notebook?

A Jupyter notebook is an interactive environment that lets you write and execute code in Python and other languages. There are two main cell types in a notebook:

- A code cell contains code to be executed in the kernel and displays its output below.
- A markdown cell contains text formatted using Markdown and displays its output in-place when it is run.

2 What is Google Colaboratory?

Colaboratory, or “Colab” for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. More technically, Colab is a hosted Jupyter notebook service that requires no setup to use, while providing free access to computing resources including GPUs.

3 Setting up

Go to your drive and create a new Google Colab, see figure 1 for an example screenshot.

Alternatively you can go directly to `colab.research.google.com`.

4 Using Hardware Acceleration

One of the factors driving Machine Learning in the last few years is the availability of hardware acceleration and dedicated libraries to take advantage of this. Google Colab puts free devices which can greatly speed up these operations.

In order to select the type of acceleration follow these steps:

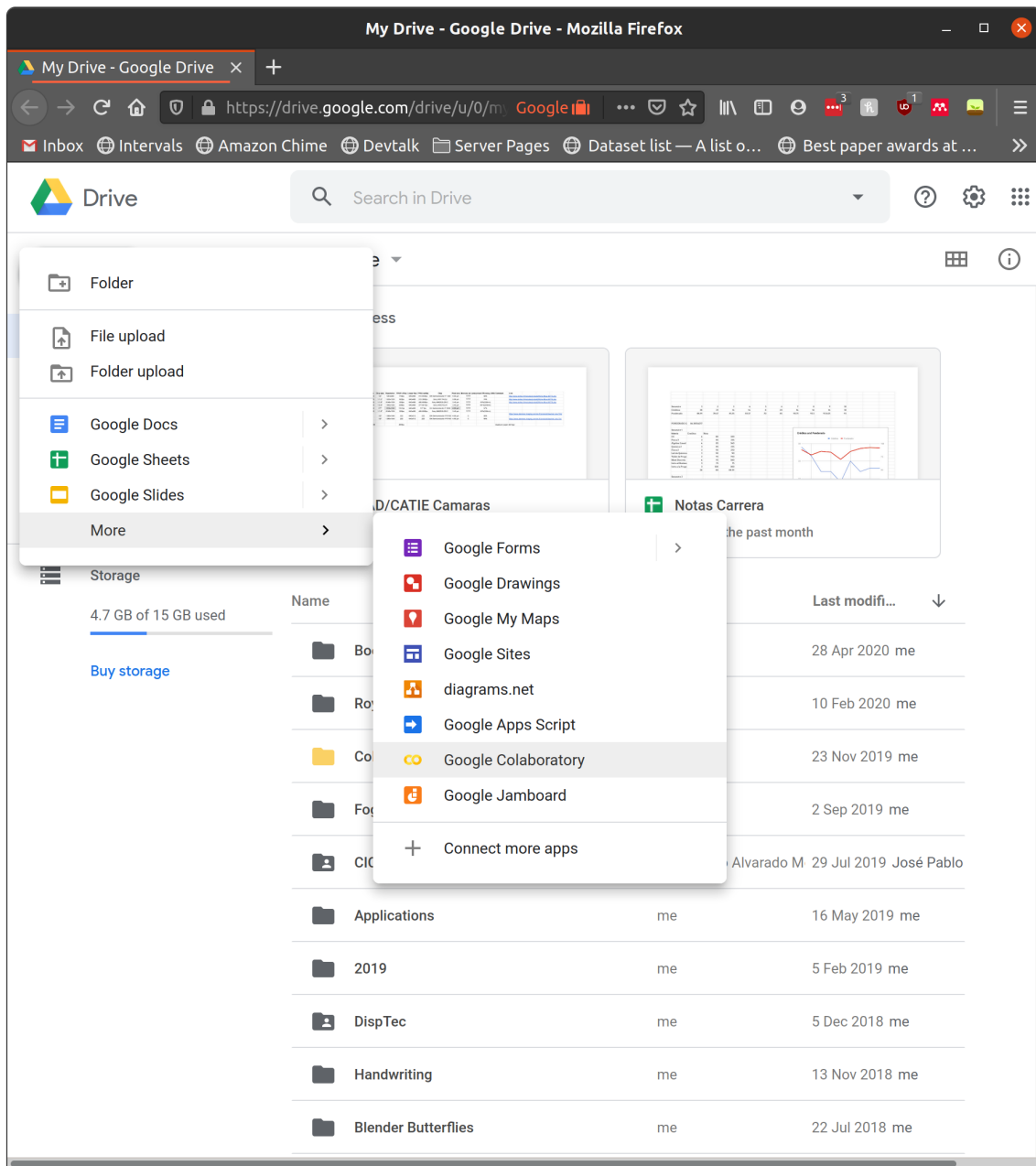


Figure 1: Sample Creation of a Colab Notebook

1. Go to the Runtime Menu
2. Select Change Runtime Type
3. Select either GPU or TPU

Be warned that although Colab offers several GPU's, the high-end GPUs can't be warranted which can actually make training slower.

4.1 Selecting a TPU

TPUs provided by colab are not available in the same virtual machine, but rather provided as a service. After selecting the TPU as a runtime, the environment variable `COLAB_TPU_ADDR` provides the TPU cluster address for your runtime:

```
import os

try:
    print (os.environ[ 'COLAB_TPU_ADDR' ])
except:
    print ( 'COLAB_TPU_ADDR is not available in the runtime ')
```

4.2 When to select GPU or TPU

GPUs are extremely efficient at matrix multiplication, which basically forms the core of machine learning. The strength of GPU lies in data parallelization, which means that instead of relying on a single core, as CPUs did before, a GPU can have many small cores.

TPU are Google's custom made tensor processing units. Built using 4 custom made ASICs or application-specific integrated circuit, specifically designed for machine learning using TensorFlow.

GPU should be used for mid-to-large datasets and models, image and video processing, and applications running CUDA or OpenCL. TPUs are recommended for matrix computations, dense vector processing, massive datasets and huge models.

5 Mounting Google Drive

Google Colab has a 12h time limit, after which it is reset. Checkpoints for models should be saved to a drive, datasets can also be easily loaded using this:

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

Then you'll see a link, click on that, allow access, copy the code that pops up, paste it in the box, and hit enter. If you don't see your drive in the side box on the left, just hit "refresh" and it should show up.

6 Importing libraries

Imports are pretty standard. For the most part, you can import your libraries by running import like you do in any other notebook.

```
import time
import PIL
from torch import nn, optim
import os
...
```

If a library is not available with an import, Google Colab supports pip:

```
!pip install -q keras
import keras
```

7 Further Reading

<https://research.google.com/colaboratory/faq.html> <https://colab.research.google.com/notebooks/welcome.ipynb> <https://www.predictiveanalyticsworld.com/machinelearningtimes/should-you-choose-a-gpu-or-a-tpu-to-train-your-machine-learning-models/10460/>