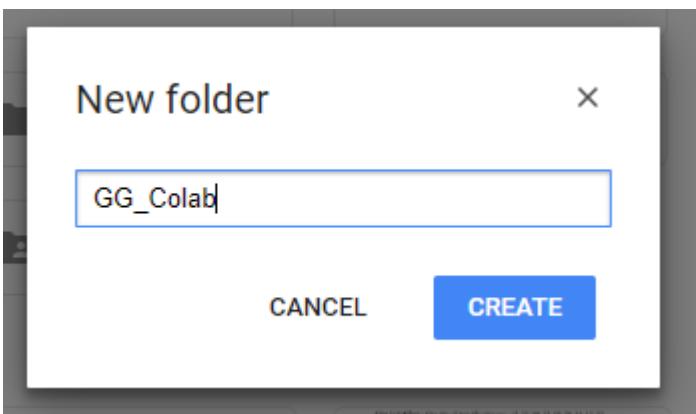


Sử dụng GPU trong Google Colaboratory

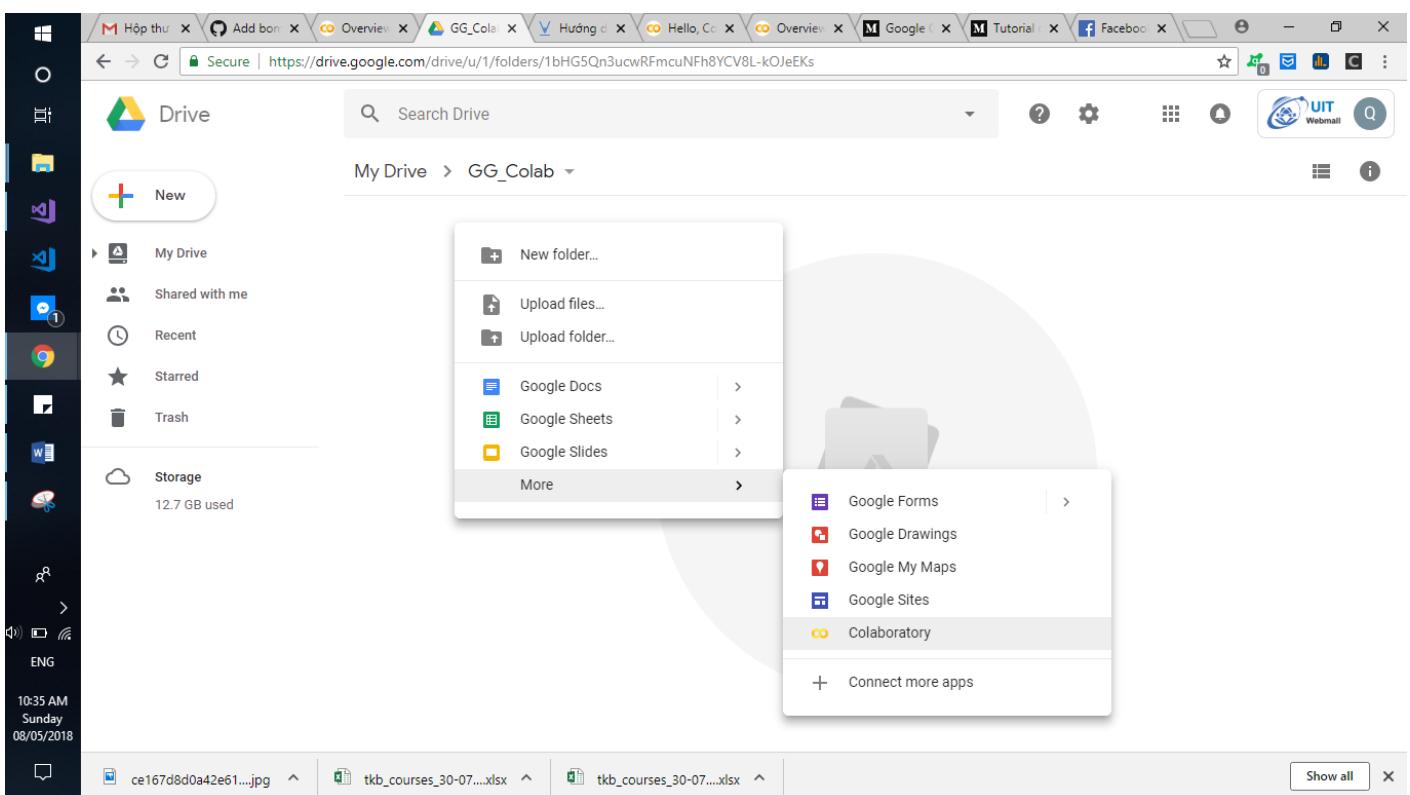
Google Colaboratory (GG Colab) là một dịch vụ điện toán đám mây được phát triển bởi Google và nó hoàn toàn miễn phí. Trong thời gian gần đây, Google vừa hỗ trợ chạy GPU trên máy ảo giúp cho việc nghiên cứu, sử dụng với deep learning và xử lý ảnh trở nên thuận tiện hơn. Bài viết này sẽ hướng dẫn cách sử dụng GPU trên Google Colaboratory

1. TẠO FILE GOOGLE COLABORATORY:

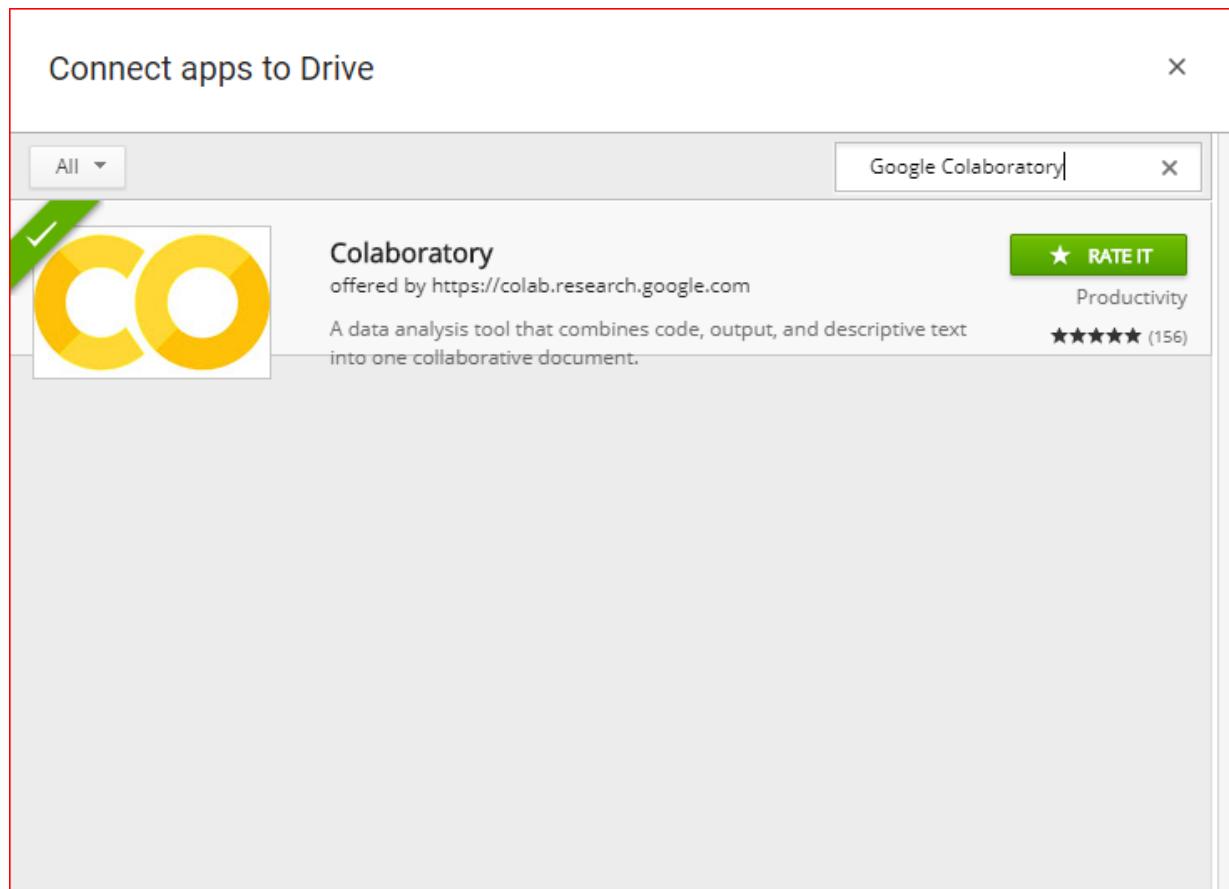
File **.ipynb** của GG Colab thường được liên kết với Google Drive giúp cho việc sắp xếp và quản lý trở nên thuận tiện hơn. Đầu tiên, mở Google Drive và tạo một folder lưu trữ các file GG Colab. Ở đây tôi đặt tên folder là “**“GG_Colab”**”



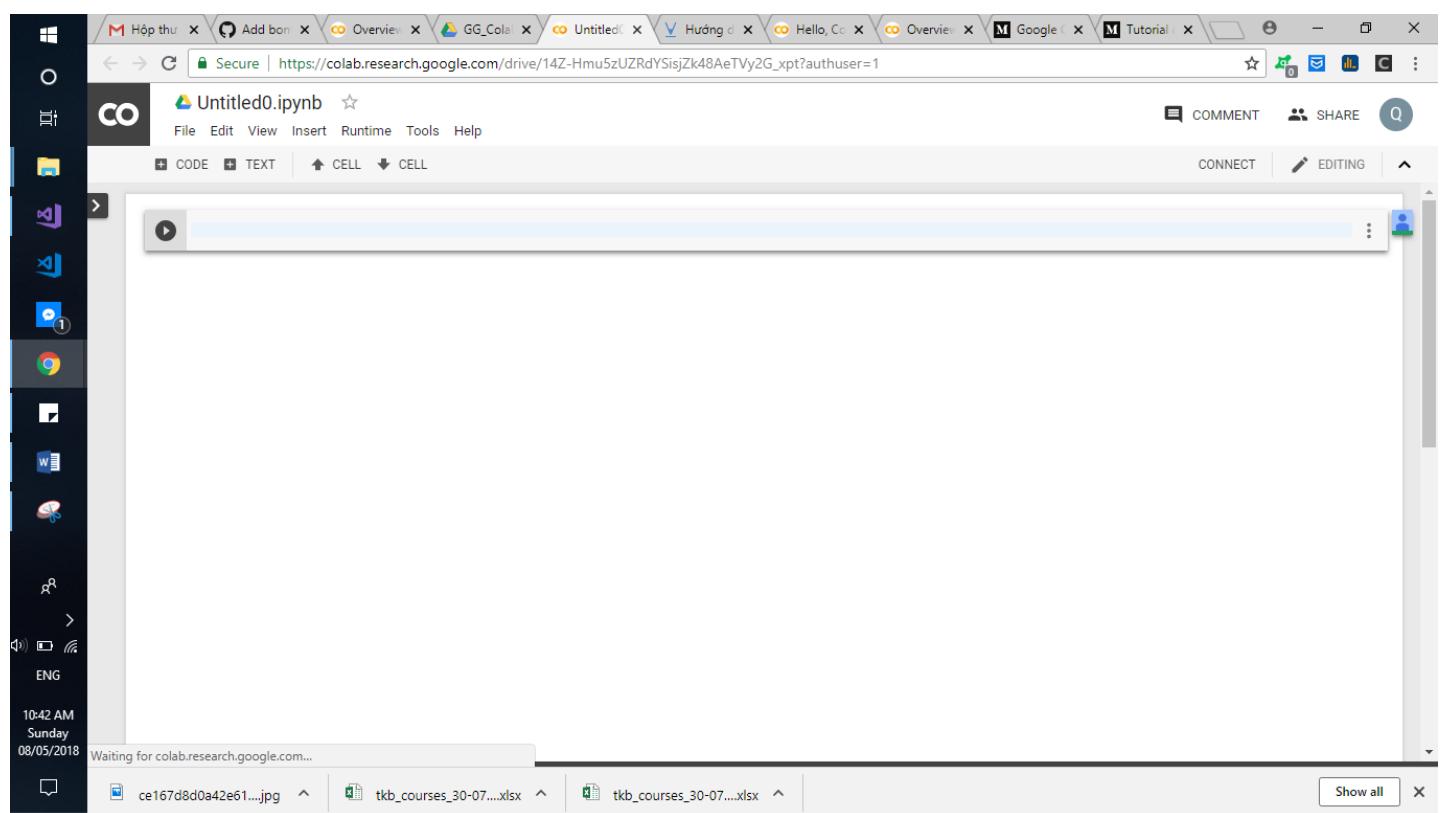
Vào folder vừa tạo, bây giờ ta tạo mới một file .ipynb để sử dụng. Chuột phải chọn “**More > Colaboratory**”



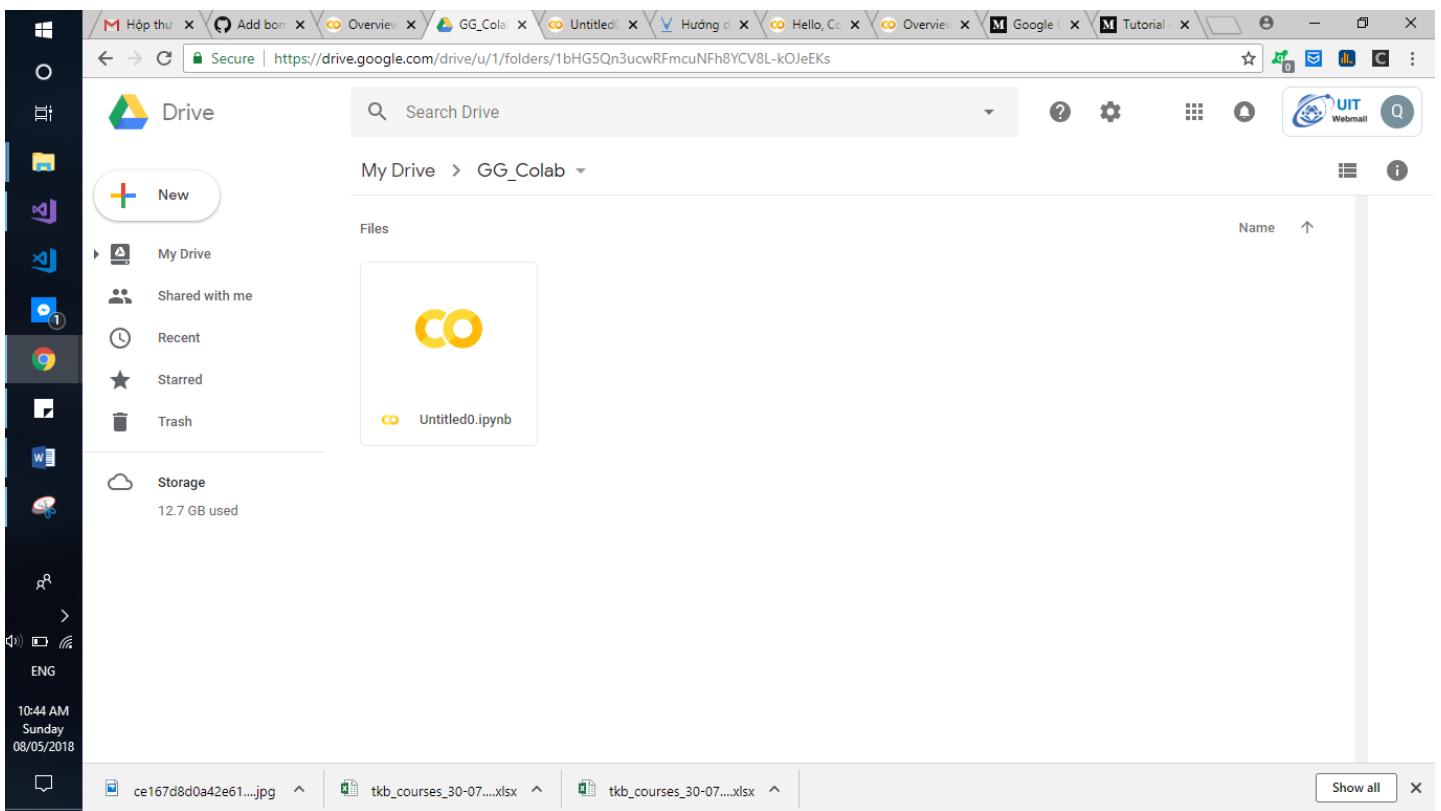
Lưu ý: nếu như không tìm thấy dòng Colaboratory nghĩa là drive của bạn chưa liên kết với GG Colab. Chọn Connect more apps, dialog tìm kiếm hiện ra, gõ “Google Colaboratory” hoặc “Colaboratory” để tìm kiếm.



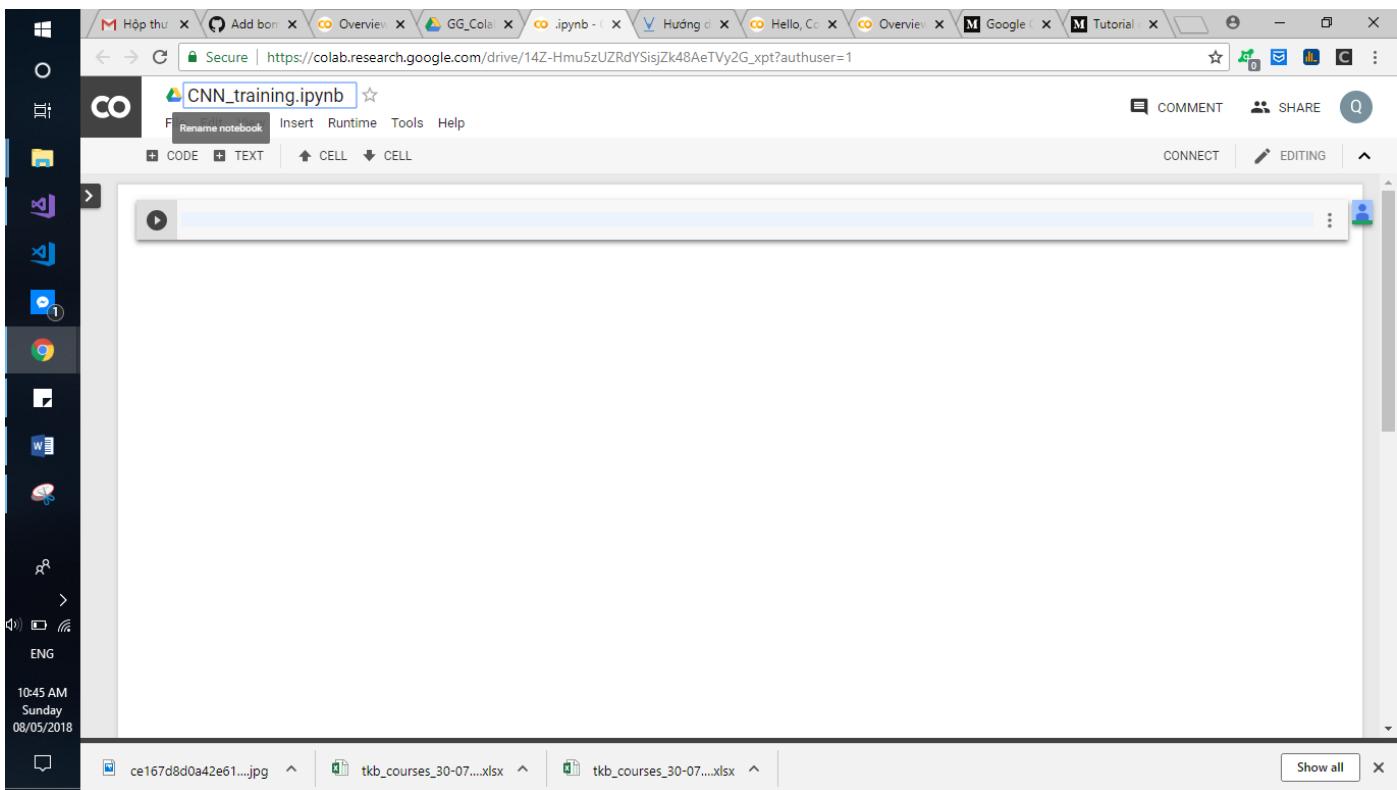
Sau khi tạo mới, giao diện GG Colab hiện ra như sau:



Đồng thời, phía cửa sổ Google Drive cũng hiện ra một file như sau:



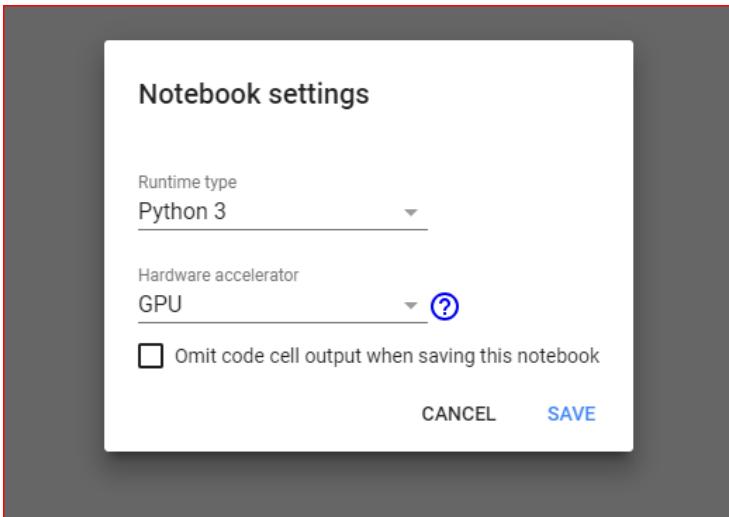
Tiếp theo, ta đổi tên file này lại cho phù hợp với mục đích sử dụng. Ở đây, tôi đặt tên file là “**CNN_training**”



2. THIẾT LẬP

Đầu tiên, để có thể sử dụng được GPU, ở menu chọn “**Runtime > Change**” runtime type.

Ở phần “Runtime type” tôi sử dụng Python 3; phần “Hardware accelerator” chọn GPU để có thể sử dụng hỗ trợ GPU từ Google. Sau đó chọn “Save”



Trước khi có thể sử dụng GG Colab để thực hiện việc training, ta cần chạy đoạn code phía dưới để thiết lập, cài đặt một số thư viện cần thiết và xác thực tài khoản

```
!apt-get install -y -qq software-properties-common python-software-properties module-init-tools
!add-apt-repository -y ppa:alessandro-strada/ppa 2>&1 > /dev/null
!apt-get update -qq 2>&1 > /dev/null
!apt-get -y install -qq google-drive-ocamlfuse fuse
from google.colab import auth
auth.authenticate_user()
from oauth2client.client import GoogleCredentials
creds = GoogleCredentials.get_application_default()
import getpass
!google-drive-ocamlfuse -headless -id={creds.client_id} -secret={creds.client_secret} < /dev/null 2>&1 | grep URL
vcode = getpass.getpass()
!echo {vcode} | google-drive-ocamlfuse -headless -id={creds.client_id} -secret={creds.client_secret}
```

Sau khi nhập đoạn code vào cell, ta tiến hành run đoạn code, GG Colab sẽ tiến hành cài đặt các thư viện cần thiết như hình bên dưới

```
!apt-get install -y -qq software-properties-common python-software-properties module-init-tools
!add-apt-repository -y ppa:alessandro-strada/ppa 2>&1 > /dev/null
!apt-get update -qq 2>&1 > /dev/null
!apt-get -y install -qq google-drive-ocamlfuse fuse
from google.colab import auth
auth.authenticate_user()
from oauth2client.client import GoogleCredentials
creds = GoogleCredentials.get_application_default()
import getpass
!google-drive-ocamlfuse -headless -id={creds.client_id} -secret={creds.client_secret} < /dev/null 2>&1 | grep URL
vcode = getpass.getpass()
!echo {vcode} | google-drive-ocamlfuse -headless -id={creds.client_id} -secret={creds.client_secret}

...
Preconfiguring packages ...
Selecting previously unselected package cron.
(Reading database ... 18408 files and directories currently installed.)
Preparing to unpack .../00-cron_3.0pl1-128ubuntu5_amd64.deb ...
Unpacking cron (3.0pl1-128ubuntu5) ...
Selecting previously unselected package libapparmor1:amd64.
Preparing to unpack .../01-libapparmor1_2.11.0-2ubuntu17.1_amd64.deb ...
Unpacking libapparmor1:amd64 (2.11.0-2ubuntu17.1) ...
Selecting previously unselected package libdbus-1-3:amd64.
Preparing to unpack .../02-libdbus-1-3_1.10.22-1ubuntu1_amd64.deb ...
Unpacking libdbus-1-3:amd64 (1.10.22-1ubuntu1) ...
Selecting previously unselected package dbus.
Preparing to unpack .../03-dbus_1.10.22-1ubuntu1_amd64.deb ...
Unpacking dbus (1.10.22-1ubuntu1) ...
Selecting previously unselected package dirmngr.
Preparing to unpack .../04-dirmngr_2.1.15-1ubuntu8.1_amd64.deb ...
Unpacking dirmngr (2.1.15-1ubuntu8.1) ...
Selecting previously unselected package distro-info-data.
Preparing to unpack .../05-distro-info-data_0.36ubuntu0.2_all.deb ...
Unpacking distro-info-data (0.36ubuntu0.2) ...
Selecting previously unselected package libkmod2:amd64.
Preparing to unpack .../06-libkmod2_24-1ubuntu2_amd64.deb ...
```

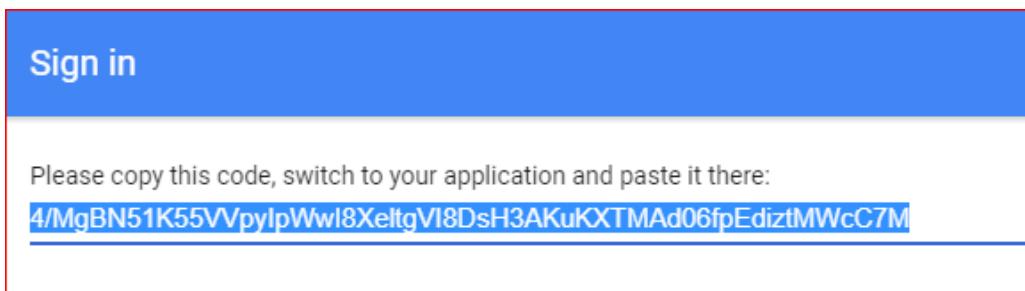
Sau đó, một địa chỉ URL và ô mã xác thực hiện ra yêu cầu xác thực, nhấn vào URL sẽ chuyển hiện ra một tab mới, ta chọn “Allow”.

```
(Reading database ... 19816 files and directories currently installed.)
Preparing to unpack .../libfuse2_2.9.7-1ubuntu1_amd64.deb ...
Unpacking libfuse2:amd64 (2.9.7-1ubuntu1) ...
...
Selecting previously unselected package fuse.
Preparing to unpack .../fuse_2.9.7-1ubuntu1_amd64.deb ...
Unpacking fuse (2.9.7-1ubuntu1) ...
Selecting previously unselected package google-drive-ocamlfuse.
Preparing to unpack .../google-drive-ocamlfuse_0.6.21-0ubuntu2_amd64.deb ...
Unpacking google-drive-ocamlfuse (0.6.21-0ubuntu2) ...
Setting up libfuse2:amd64 (2.9.7-1ubuntu1) ...
Processing triggers for libc-bin (2.26-0ubuntu2.1) ...
Setting up fuse (2.9.7-1ubuntu1) ...
Setting up google-drive-ocamlfuse (0.6.21-0ubuntu2) ...
Go to the following link in your browser:
```

https://accounts.google.com/o/oauth2/auth?redirect_uri=urn%3Aietf%3Awg%3Aoauth%3A2.0%3Aoob&prompt=select_account&response_type=code&client_id=4/MgBN51K55VvpIwWwl8XelqVI8DsH3AKuKXTMAd06fpEdiztMWcC7M

Enter verification code:

Sau đó copy đoạn mã xác thực bỏ vào và nhấn Enter



GG Colab sẽ tiến hành xác thực thêm một lần nữa, làm tương tự như lần trước

```
!apt-get install -y -qq software-properties-common python-software-properties module-init-tools
!add-apt-repository -y ppa:alessandro-strada/ppa 2>&1 > /dev/null
!apt-get update -qq 2>&1 > /dev/null
!apt-get -y install -qq google-drive-ocamlfuse fuse
from google.colab import auth
auth.authenticate_user()
from oauth2client.client import GoogleCredentials
creds = GoogleCredentials.get_application_default()
import getpass
!google-drive-ocamlfuse -headless -id={creds.client_id} -secret={creds.client_secret} < /dev/null 2>&1 | grep URL
vcode = getpass.getpass()
!echo {vcode} | google-drive-ocamlfuse -headless -id={creds.client_id} -secret={creds.client_secret}

...
Please, open the following URL in a web browser: https://accounts.google.com/o/oauth2/auth?client_id=32555940559.apps.googleusercontent.com&redirect
```

Sau khi chương trình hiện ra dòng phía dưới thì xem như ta đã thiết lập và xác thực thành công

Please enter the verification code: Access token retrieved correctly. Đồng thời, góc bên phải phía trên sẽ hiện ra biểu tượng kết nối thành công

```
!apt-get install -y -qq software-properties-common python-software-properties module-init-tools
!add-apt-repository -y ppa:alessandro-strada/ppa 2>&1 > /dev/null
!apt-get update -qq 2>&1 > /dev/null
!apt-get -y install -qq google-drive-ocamlfuse fuse
from google.colab import auth
auth.authenticate_user()
from oauth2client.client import GoogleCredentials
creds = GoogleCredentials.get_application_default()
import getpass
!google-drive-ocamlfuse -headless -id={creds.client_id} -secret={creds.client_secret} < /dev/null 2>&1 | grep URL
vcode = getpass.getpass()
!echo {vcode} | google-drive-ocamlfuse -headless -id={creds.client_id} -secret={creds.client_secret}

...
Please, open the following URL in a web browser: https://accounts.google.com/o/oauth2/auth?client_id=32555940559.apps.googleusercontent.com&redirect
.....
Please, open the following URL in a web browser: https://accounts.google.com/o/oauth2/auth?client_id=32555940559.apps.googleusercontent.com&redirect
Please enter the verification code: Access token retrieved correctly.
```

3. TRAINING DỮ LIỆU MNIST VỚI GOOGLE COLABORATORY

Trong phần này, ta sử dụng GG Colab để train một mạng CNN đơn giản của bộ MNIST. Sau này, chúng ta thường làm việc với các file .py được lưu trữ trên drive nên trước tiên, mount Google Drive trên GG Colab bằng lệnh:

```
!mkdir -p drive
```

```
!google-drive-ocamlfuse drive
```

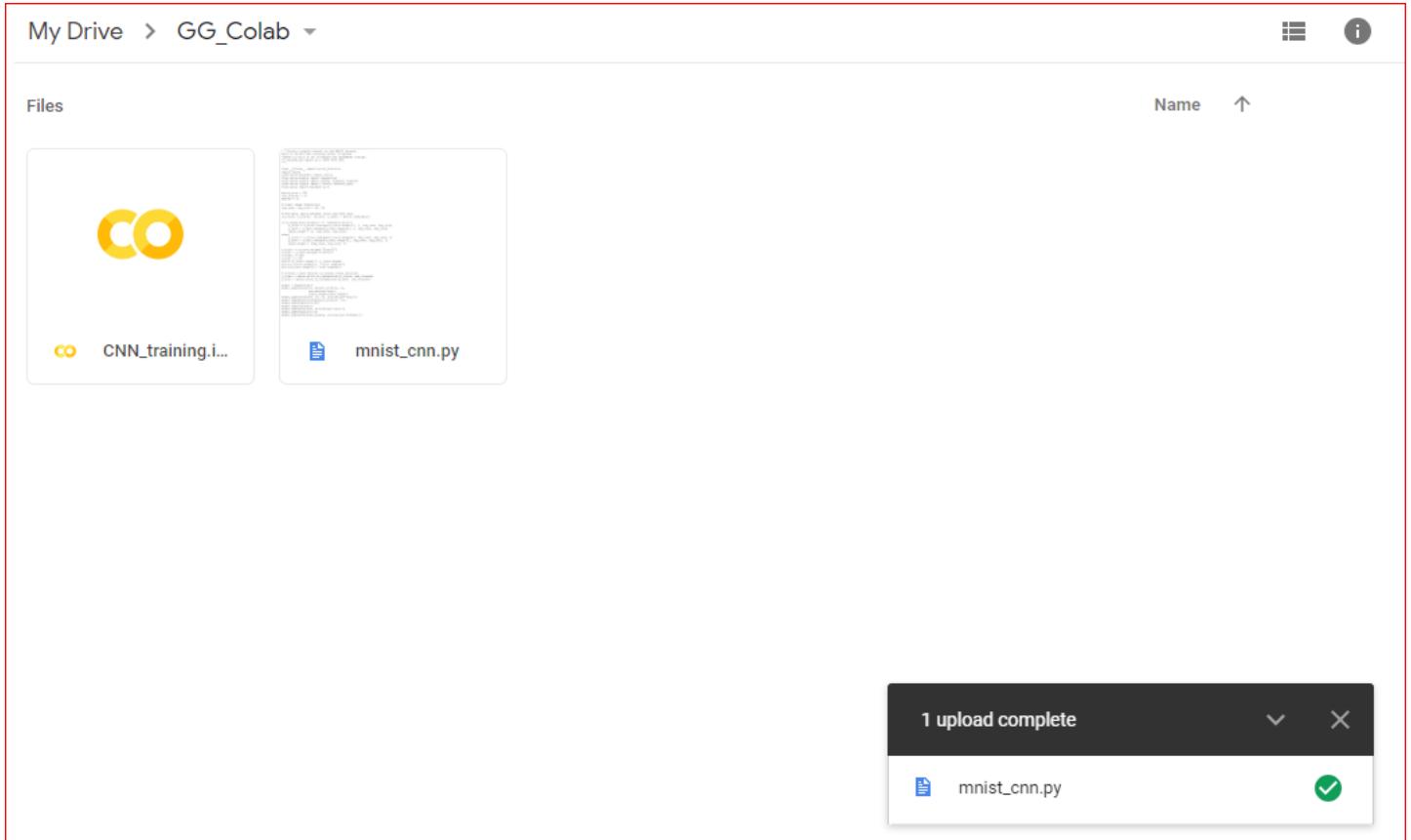
Tiếp theo cài đặt Keras. Keras là một library hỗ trợ build CNN được viết dựa trên tensorflow nhưng đơn giản, dễ tiếp cận và sử dụng hơn.

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

Ở đây, chúng ta sẽ chạy một file .py đã build sẵn CNN để nhận dạng dữ liệu MNIST.

Download file mnist_cnn.py ở địa chỉ https://github.com/keras-team/keras/blob/master/examples/mnist_cnn.py. Đây là file .py đã build sẵn model CNN để nhận dạng dữ liệu chữ viết tay MNIST.

Sau đó, upload vào folder GG_Colab mà ta đã tạo trong Google Drive



Như vậy, đường dẫn đến file “mnist_cnn.py” là “drive/GG_Colab/mnist_cnn.py”

Sau đó, chạy lệnh

```
!python3 drive/GG_Colab/mnist_cnn.py
```

Tiến hành training

```

python3 drive/GG_Colab/mnist_cnn.py

Using TensorFlow backend.
Downloading data from https://s3.amazonaws.com/img-datasets/mnist.npz
11493376/11490434 [=====] - 4s 0us/step
x_train shape: (60000, 28, 28, 1)
60000 train samples
10000 test samples
Train on 60000 samples, validate on 10000 samples
Epoch 1/12
2018-08-06 04:02:22.456174: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:897] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so return name: Tesla K80 major: 3 minor: 7 memoryClockRate(GHz): 0.8235
pcibusID: 0000:00:04.0
totalMemory: 11.17GiB freeMemory: 11.10GiB
2018-08-06 04:02:22.456771: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1471] Adding visible gpu devices: 0
2018-08-06 04:02:22.844789: I tensorflow/core/common_runtime/gpu/gpu_device.cc:952] Device interconnect StreamExecutor with strength 1 edge matrix:
2018-08-06 04:02:22.844860: I tensorflow/core/common_runtime/gpu/gpu_device.cc:958] 0
2018-08-06 04:02:22.844997: I tensorflow/core/common_runtime/gpu/gpu_device.cc:971] 0: N
2018-08-06 04:02:22.845376: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1084] Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 with 10763 MB memory) -> physical GPU (device: 0/1,0000000000000000 [=====] - ETA: 10s - loss: 0.4860 - acc: 0.84700000/60000 [=====] - 12s 203us/step - loss: 0.2641 - acc: 0.9189 - val_loss: 0.0601 - val_acc: 0.9852
Epoch 2/12
60000/60000 [=====] - 10s 162us/step - loss: 0.08917 - acc: 0.9725 - val_loss: 0.0443 - val_acc: 0.9852
Epoch 3/12
59520/60000 [=====] - ETA: 0s - loss: 0.0677 - acc: 0.979560000/60000 [=====] - 10s 161us/step - loss: 0.0676 - acc: 0.9795 - val_loss: 0.0338 - val_acc: 0.9851
Epoch 4/12
60000/60000 [=====] - 10s 161us/step - loss: 0.0546 - acc: 0.9836 - val_loss: 0.0322 - val_acc: 0.9881
Epoch 5/12
60000/60000 [=====] - 10s 164us/step - loss: 0.0489 - acc: 0.9850 - val_loss: 0.0308 - val_acc: 0.9887
Epoch 6/12
5888/60000 [=>.....] - ETA: 8s - loss: 0.0348 - acc: 0.989860000/60000 [=====] - 10s 162us/step - loss: 0.0409 - acc: 0.9877 - val_loss: 0.0295 - val_acc: 0.9910
Epoch 7/12
60000/60000 [=====] - 10s 161us/step - loss: 0.0385 - acc: 0.9883 - val_loss: 0.0265 - val_acc: 0.9910
Epoch 8/12
56192/60000 [=====] - ETA: 0s - loss: 0.0338 - acc: 0.989260000/60000 [=====] - 10s 162us/step - loss: 0.0337 - acc: 0.9891 - val_loss: 0.0275 - val_acc: 0.9910
Epoch 9/12
60000/60000 [=====] - 10s 162us/step - loss: 0.0325 - acc: 0.9902 - val_loss: 0.0282 - val_acc: 0.9897
Epoch 10/12
60000/60000 [=====] - 10s 162us/step - loss: 0.0302 - acc: 0.9906 - val_loss: 0.0293 - val_acc: 0.9907
Epoch 11/12
4736/60000 [=>.....] - ETA: 8s - loss: 0.0254 - acc: 0.991660000/60000 [=====] - 10s 163us/step - loss: 0.0286 - acc: 0.9911 - val_loss: 0.0287 - val_acc: 0.9910
Epoch 12/12
60000/60000 [=====] - 10s 164us/step - loss: 0.0265 - acc: 0.9918 - val_loss: 0.0254 - val_acc: 0.9918
Test loss: 0.025446533852072573
Test accuracy: 0.9918

```

GG Colab train 6000 samples, 12 epochs trong thời gian rất nhanh (mỗi epoch chỉ mất khoảng 10s)

4. CLONE PROJECT TỪ GIT

Trước tiên ta chuyên đường dẫn mặc định từ “drive” vào “**drive/GG_Colab**” thông qua lệnh:

```
import os
```

```
os.chdir("drive/GG_Colab")
```

Để clone một project bất kỳ từ git, sử dụng câu lệnh:

```
!git clone <git_url>
```

Ví dụ: Git clone project keras-mnist-tutorial từ <https://github.com/wxs/keras-mnist-tutorial>

For a mini tutorial at U of T, a tutorial on MNIST classification in Keras.

Branch: master ▾ New pull request

Clone with HTTPS ⓘ Use SSH
Use Git or checkout with SVN using the web URL.
<https://github.com/wxs/keras-mnist-tutorial>

Create new file Upload files Find file Clone or download

8 commits 1 branch 0 releases 1 contributor

MNIST in Keras.ipynb Fixed prerequisite

README.md Add a link to the actual notebook in the readme.

figure.png Initial commit

newnotebook.png Better installation instructions.

README.md

Just a little notebook based on the Keras MNIST example for a tutorial I'm giving. If you're looking at this on Github you can view a [static version of the notebook](MNIST in Keras.ipynb) in your browser.

Vào GG Colab và gõ lệnh

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

Sau đó run:

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



```
Cloning into 'keras-mnist-tutorial'...
remote: Counting objects: 26, done.
remote: Total 26 (delta 0), reused 0 (delta 0), pack-reused 26
Unpacking objects: 100% (26/26), done.
Checking connectivity: 26, done.
Checking out files: 100% (4/4), done.
```

Sau khi clone, trong Google Drive đã xuất hiện folder keras-mnist-tutorial mà chúng ta đã clone

Search Drive

My Drive > GG_Colab

Folders

- keras-mnist-tu...

Files

- CNN_training.i...
- mnist_cnn.py

Search Drive

My Drive > GG_Colab > keras-mnist-tutorial

Folders

- .git

Files

- figure.png
- MNIST in Keras...
- newnotebook.p...
- README.md

5. MỘT SỐ CÂU LỆNH TRONG GOOGLE COLABORATORY

Lệnh thiết lập ban đầu khi khởi chạy GG Colab: (1)

```
!apt-get install -y -qq software-properties-common python-software-properties module-init-tools
!add-apt-repository -y ppa:alessandro-strada/ppa 2>&1 > /dev/null
!apt-get update -qq 2>&1 > /dev/null
!apt-get -y install -qq google-drive-ocamlfuse fuse
```

```
from google.colab import auth
auth.authenticate_user()
from oauth2client.client import GoogleCredentials
creds = GoogleCredentials.get_application_default()
import getpass
!google-drive-ocamlfuse -headless -id={creds.client_id} -secret={creds.client_secret} </dev/nu
ll 2>&1 | grep URL
vcode = getpass.getpass()
!echo {vcode} | google-drive-ocamlfuse -headless -id={creds.client_id} -secret={creds.client_s
ecret}
```

Mount Google Drive: (2)

```
!mkdir -p drive
!google-drive-ocamlfuse drive
```

Install Keras: (3)

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

Kiểm tra xem GPU có đang hoạt động

```
import tensorflow as tf
tf.test.gpu_device_name()
```

Xem thông tin CPU và GPU

```
from tensorflow.python.client import device_lib
device_lib.list_local_devices()
```

Xem thông tin RAM

```
!cat /proc/meminfo
```

Xem thông tin chi tiết CPU

```
!cat /proc/cpuinfo
```

Cài đặt thư viện bất kỳ:

```
!pip install hoặc !apt-get install
```

6. MỘT SỐ LUU Ý

- Hiện tại, mỗi lần đăng nhập vào GG Colab chúng ta sẽ được chỉ định một máy ảo khác nhau. Do đó, ta cần thiết lập lại từ đầu.
- Thông thường, thứ tự cần thực hiện khi đăng nhập vào Google Colab sẽ là: chạy đoạn code (1) để cài đặt và thiết lập, (2) để mount Google Drive, (3) để cài đặt Keras và cài đặt các thư viện cần thiết khác.
- GPU hiện tại mà Google Colab sử dụng là Tesla K80, tuy nhiên ta chỉ có thể sử dụng trong một khoảng thời gian giới hạn (thường là ít hơn 8 hrs) nên GG Colab chỉ phù hợp khi cần xử lý những project có lượng data vừa và nhỏ. Tuy nhiên ta vẫn có thể chạy với dữ liệu lớn với một vài thiết lập.
- Để chạy file .py ta sử dụng lệnh

```
!python3 <link_to_pyfile>
```

- Để chạy file .ipynb, ta mở trực tiếp file bằng GG Colab

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

<https://colab.research.google.com/notebooks/welcome.ipynb#scrollTo=-Rh3-Vt9Nev9>