

Colab Manipulations Session

IMSP 1st DATA Science school

We will make our first steps on using colab research google cloud resource.

Colab is a Cloud service from Google. Among many features, it offers:

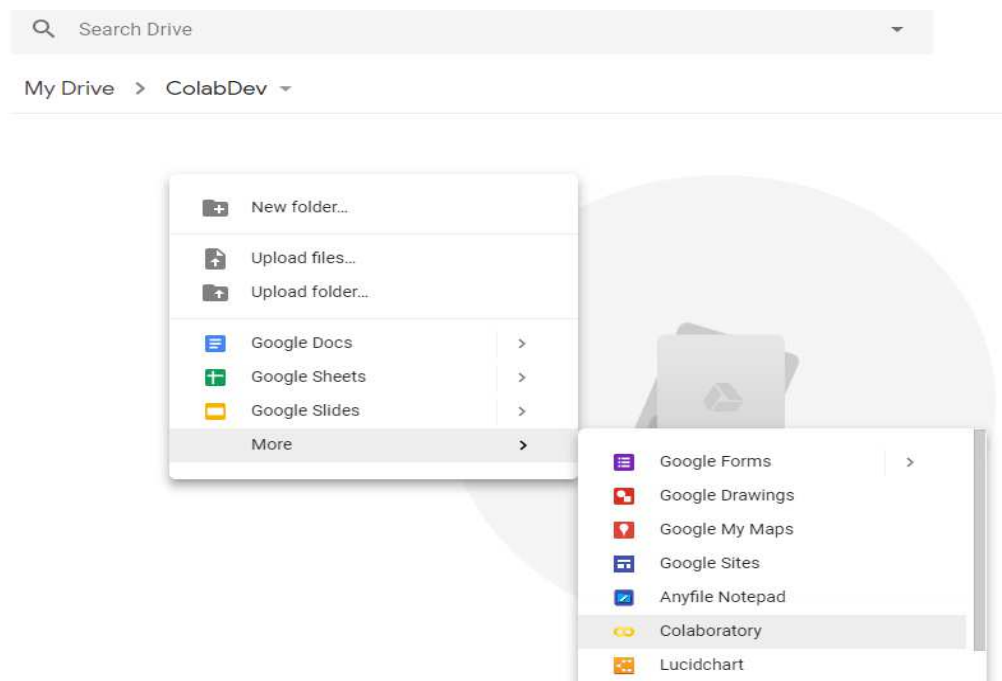
- Import of your Python codes and notebooks
- Configuration environment setting for different Frameworks like TensorFlow, Keras, PyTorch and OpenCV
- It offers also free GPU resources for improved execution time.

In this practical session you will configure a Colab environment for your own developments

1/ Create a New folder Named “ColabDev” in your Google Drive account. Colab will use your allocated Google Drive space to store your files. Hence you should rather use this place as a temporary folder and keep it light. Note that you can use the default Colab Notebooks folder in place.

2 / Creating a new colab Notebook File:

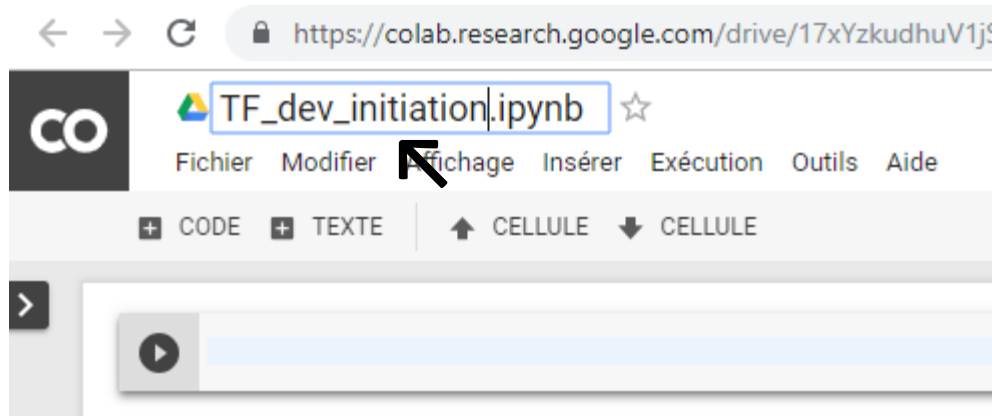
Right Click in the new created empty folder → More → Colaboratory



You are redirected to:

<https://colab.research.google.com/>

From the new window that opens up: Click on the File Name and rename it as below:



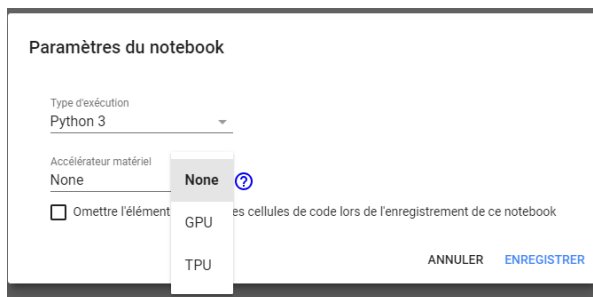
From Now you can insert your own code in the new created Notebook.

Setting runtime resources:

- GPU

To change between GPU(Free) and CPU running material:

- Simply go to “Edit” → “NoteBook Settings”
- Click on “Enhance Device” like below



Now we can start running our Python Code on GPU

!!\ Note that the notebook files you create are managed like all other Google Documents. You can thus share them and edit with other users.

Out of the scope of this session:

- ⇒ To have a broader and well documented knowledge of Colab, go to the link below, you have a bunch of resources to start with.

<https://colab.research.google.com/notebooks/welcome.ipynb>

- ⇒ Look at this tutorial on linear regression for further practicing (low level APIs)

From the notebooks list → `first_steps_with_tensor_flow.ipynb`

Uses cases implementations

Initiation session:

Follow together the steps in the “TF_dev_initiation.ipynb - Colaboratory.pdf” File.

DL Uses cases

For the different use cases you have the choice of 3 running environments for “use case 1” and 2 running environments for “use case 2”.

Choice 1 → Colab environment: Nothing to install. Just upload the “Use_Case_1.ipynb” to your colabdev repository and start the running the session.

Choice 2 → Run on the **DataScience School virtual machines**. You will need to check that you have the required materials installed. See the requirements installation below.

Choice 3 → Run on your local environment. Preferred is Linux environment. Instructions are there for windows too but not recommended.

Caution: The manipulations described for the notebook were tested in Colab and Linux env for *Use case 1* and fully tested on Linux env for *Use case 2*.

Requirements installation

Ubuntu/Linux:

Install Anaconda, Python 3 version: <https://www.continuum.io/downloads>
Anaconda comes with several packages built in.

```
#Create the conda env for the project  
conda env create -f environment.yml -n my_env
```

```
#New package : update the packages list in environment.yml and run  
conda env update -f environment.yml -n my_env
```

```
#Install usefull jupyter notebook extensions  
pip install jupyter_contrib_nbextensions  
jupyter contrib nbextension install --user  
jupyter nbextensions_configurator enable --user
```

Windows:

Install Anaconda, Python 3 version: <https://www.continuum.io/downloads#windows>
Anaconda comes with matplotlib built in.

```
#Create the conda env for the project
conda env create -f environment.yml -n my_env
```

```
#New package: update the packages list in environment.yml and run
conda env update -f environment.yml -n my_env
```

```
#Install usefull jupyter notebook extensions
```

```
pip install jupyter_contrib_nbextensions
jupyter contrib nbextension install --user
jupyter nbextensions_configurator enable --user
```

In the Anaconda shell type command below for installing TensorFlow:

```
pip install --upgrade tensorflow
```

If you get the error "Could not find a version that satisfies the requirement (...)" try the following alternative:

```
conda config --add channels conda-forge
conda install tensorflow
```

Use case 1

Data visualization and preprocessing

We start with Data visualization and preprocessing before building out our model. You will find terms like pipeline or Data flow to define this process. It is crucial for the two following reasons:

- You need to know what type of data you are dealing with and which feature are important for the task you are trying to achieve.
- You need to prepare your data to be efficiently usable during the training phase. Not just load and train. Normalization, One hot encoding, String Classification, and Class indexing are among the usual preparation items.

You need to answers the following questions:

- What types of features are there: Numeric, categorical, or both?
- Do you need to perform normalization
- Are there null values
- What is the labels distribution, is there class imbalance?
- Do you have correlation among the features

Upload the Use_Case_1.ipynb to Colab to proceed.

Use case 2

To have more insight about how convolutional networks come around, go through the tutorial here:

<https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist>

This practical session will run in two phases

1. Build your ETL(Extract Transform and Load) pipeline
2. Feed the loaded data frame to the model
3. Build your model implementing convolution layers
4. Test you model
5. Make some predictions
6. Save and Deploy your model for serving
7. Build a client for online predictions to your deployed server

Open the Use_Case_2.ipynb in your jupyter notebook environment.