How to set up a deep learning environment

This article gives detailed instructions on how to set up a deep learning environment. *TensorFlow* (TF), developed by Google Brain, is the most well-known library used in production for deep learning models and it has a very large community. However, TensorFlow has a steep learning curve. On the other hand, *Keras* is a high-level API built on top of TensorFlow. It is more user-friendly and easy to use compared to TF, and you can familiarize yourself more quickly because it's more "pythonic". So why not just use Keras alone? You can. But in case you want to have more control over your model network, have access to better debugging, or develop novel networks and conduct some deep learning research in the future, TensorFlow is the way.

In this tutorial, we will help you set up a TensorFlow deep learning environment. Keras API is included in TensorFlow, but you can also download and use it separately.

Some considerations before installing: using GPU or not

A Graphics Processing Unit (GPU) is a microprocessing chip designed to handle graphics in computing environments and can have many more cores than a Central Processing Unit (CPU). More cores of GPUs allow for the better computation of multiple parallel processes. If you have access to GPU, how can you use it?

On your computer, if you installed TensorFlow using `pip`, the newest version will have the GPU support. For releases 1.15 and older, CPU and GPU packages are separate.

`pip install tensorflow` for CPU

`pip install tensorflow-gpu` for GPU

However, it is not that simple since you need to also install some NVIDIA® software requirements. We suggest following TensorFlow's installation instructions for GPU support.

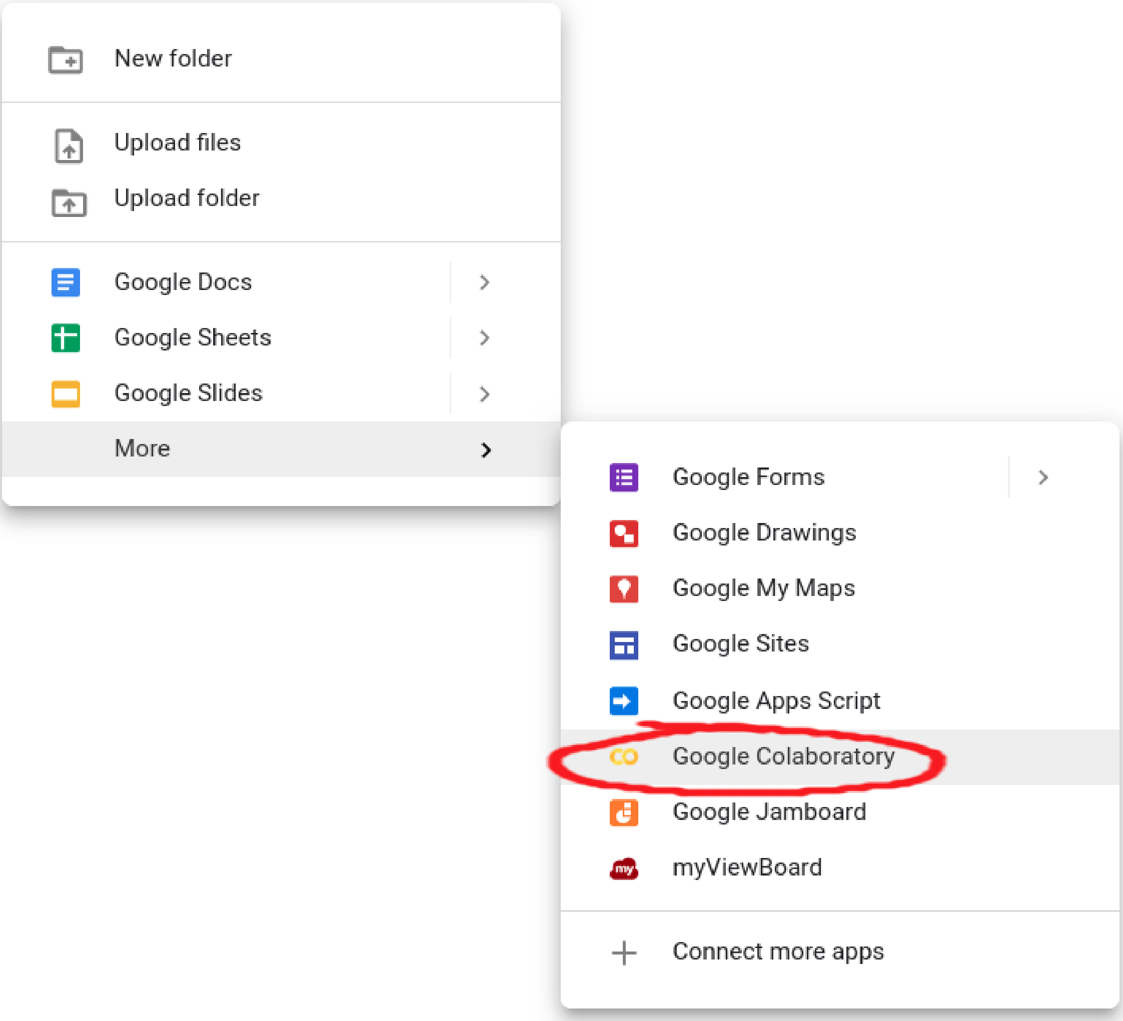
Different ways to using TensorFlow

There are multiple ways for you to run deep learning algorithms using TensorFlow:

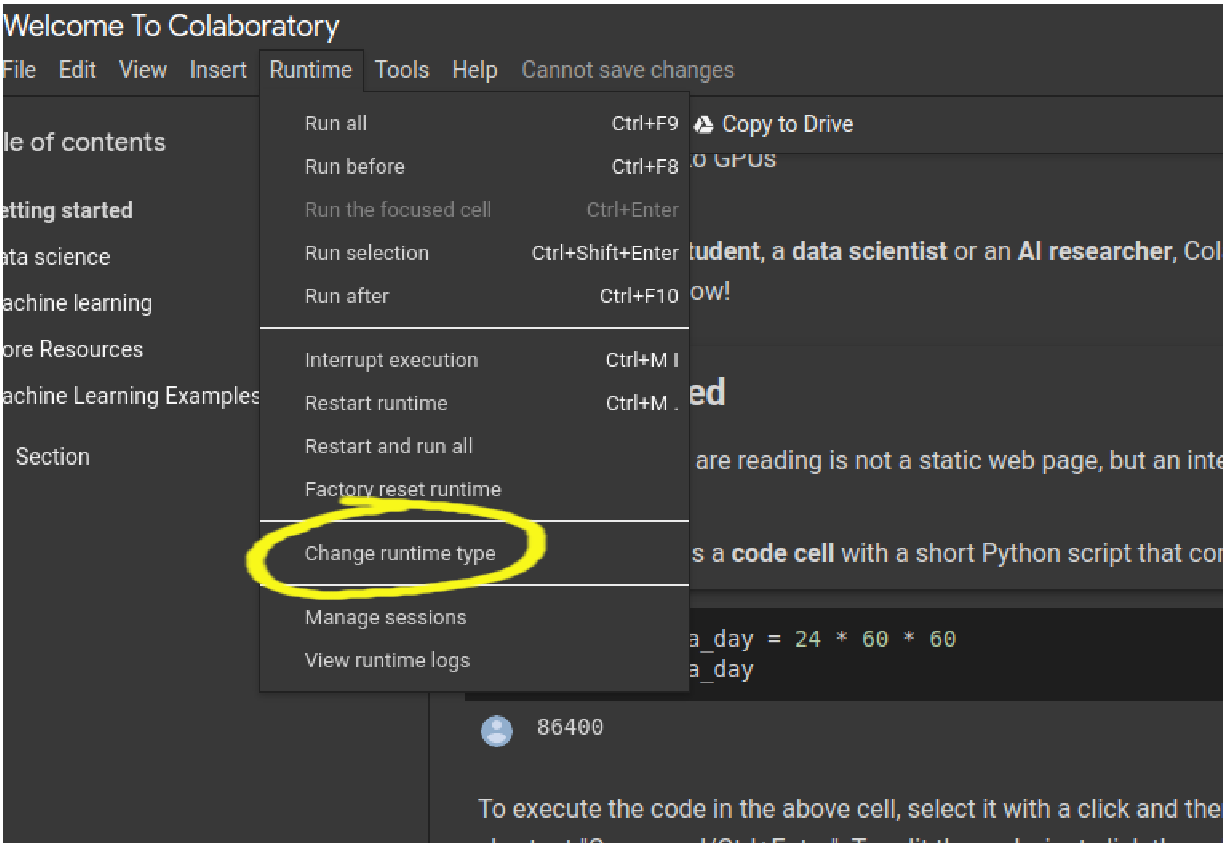
1. **(Easy-breezy) Use Google Colaboratory (Colab).** This is the easiest option for some initial exploration. No installation is necessary - run TensorFlow code directly in the browser with Colaboratory, Google's Jupyter Notebook environment that requires no setup to use and runs entirely in the cloud. There are two ways to start a Colaboratory notebook:

– Visit Colab

– From your Google Drive: *Right Mouse Click* > *More* > *Colaboratory* to start a new Colab Notebook.



Switching to using a GPU is easy in Colaboratory. In *Runtime* > *Change runtime type*, select GPU, and there you are. You even have access to a special Tensor Processing Unit (TPU), an application-specific circuit developed by Google specifically for neural network machine learning.



tudent, a **data scientist** or an **AI researcher**, Col

ow!

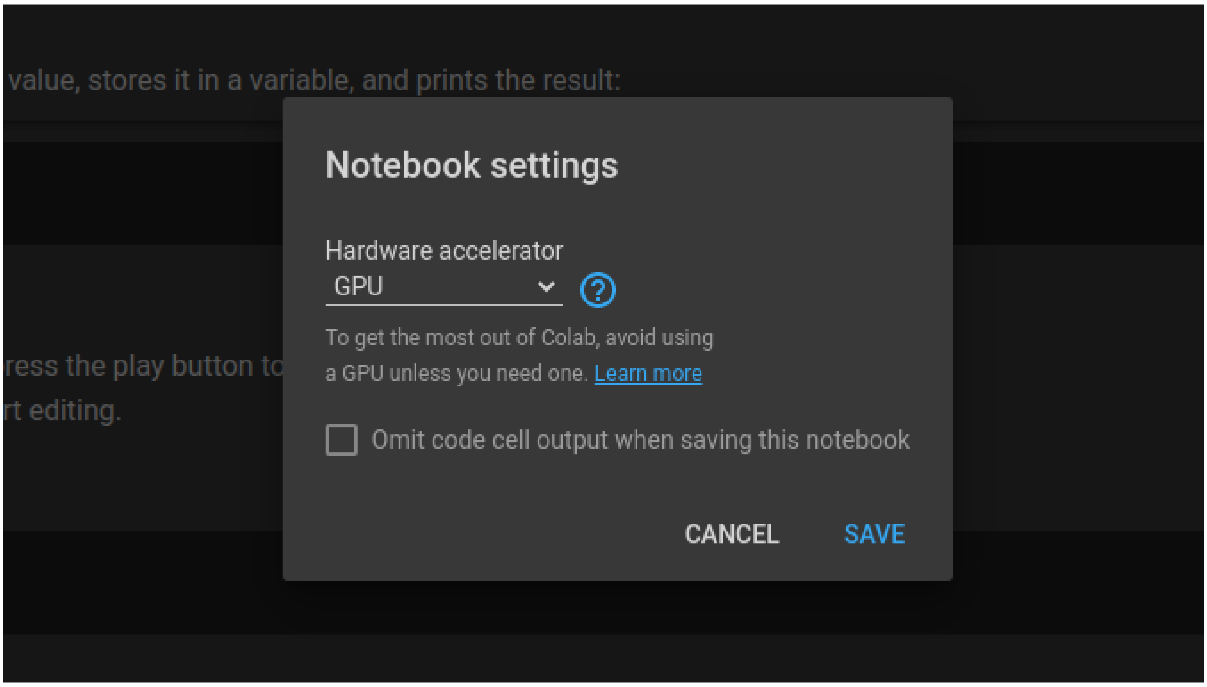
are reading is not a static web page, but an inte

s a **code cell** with a short Python script that co

```
a_day = 24 * 60 * 60
a_day
```

```
86400
```

To execute the code in the above cell, select it with a click and the



value, stores it in a variable, and prints the result:

ress the play button to

rt editing.

Hardware accelerator
GPU

To get the most out of Colab, avoid using a GPU unless you need one. Learn more

□ Omit code cell output when saving this notebook

1. **Your computer, your rules.** To install TensorFlow on your computer, you first need to have Python installed (version 3.5–3.8) and a Python’s package manager *pip* (version >19.0.), which should have come with your Python version. TensorFlow is installed in the same way as any other package you add to your Python. It’s always

good to install new large packages in a virtual environment to keep it separate from your base distribution and we will provide commands for that as well.

(Suggested) Installing with *pip*:

Create a virtual environment with _venv_ (install it if needed):

``python3 -m venv my_tensorflow``

Activate the virtual environment

``source my_tensorflow/bin/activate``

First upgrade pip:

``pip install --upgrade pip``

Use _pip_ to install TensorFlow in your virtual environment:

``pip install --upgrade tensorflow``

Note: with the new versions of TF (version > 2.0) by installing with *pip*, you don't need to specifically install the GPU supported version: *tensorflow-gpu*.

The procedure with Anaconda/Miniconda:

Create a virtual environment using the following command:

`conda create -n my_tensorflow` (instead of _my_tensorflow_ you can give any name)

Activate the environment

`conda activate my_tensorflow`

Now that you are in `my_tensorflow` virtual environment you can download TensorFlow (for CPU) with:

`conda install -c anaconda tensorflow`

If you wish to use TensorFlow with GPU support use:

`conda install -c anaconda tensorflow-gpu`

You can complete the steps above in Anaconda Navigator which provides a Graphical User Interface (GUI) if you don't want to use the command line. Be careful about the distinction between the CPU and GPU version when using Anaconda.

1. **Takes some footwork: use Docker.** Docker is a software platform for building applications based on containers — small and lightweight execution environments that make shared use of the operating system kernel but otherwise run in isolation from one another. For this, you need to install Docker on your machine. Depending on your operating system, procedures can differ. Using Docker is the easiest way to set up GPU support.

How to check if you successfully downloaded TensorFlow?

You can test your installation by entering Python's interactive mode and typing the following code:

```
import tensorflow as tf
print(tf.__version__) #will print the version of your TensorFlow
```

The community often uses `tf` as a module handle, and by also using it will enable you to copy-paste some code from the Internet as you are learning, and it will already be compatible with your import handle.

How to access the functionalities of Keras?

As we mentioned, Keras is built on top of TensorFlow. You can access its functionalities directly if you install the Keras Python package:

```
import keras
```

or through TensorFlow:

```
from tensorflow import keras
```