Running the labs outside JupyterHub

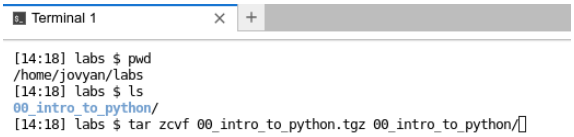
Contents

*If you choose to do the labs on your local machine during the semester, or on Colab to use the GPU, please **make sure that your notebooks run fine on JupyterHub** once you upload them for submission.*

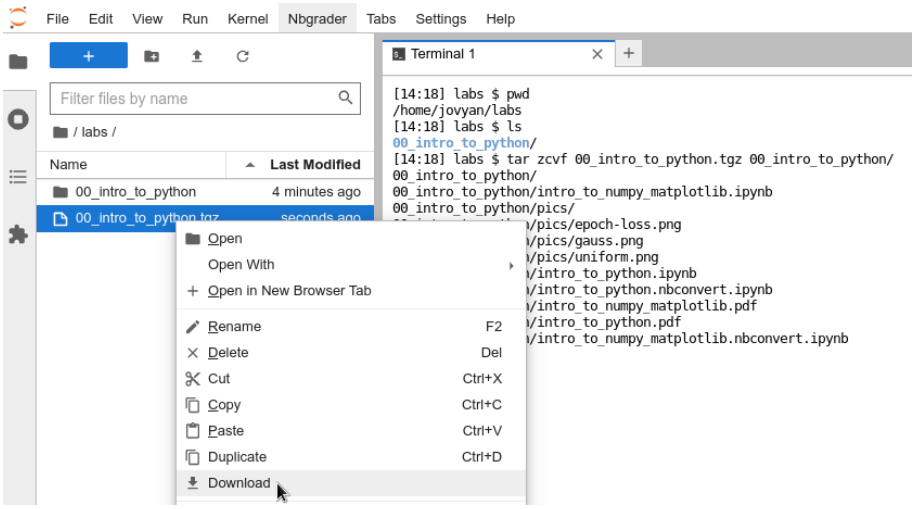
1 Downloading the labs from JupyterHub

1. Open a terminal in JupyterHub and go to the directory containing the lab you want to download.
2. Compress your lab folder with the following command (adapt it with the proper lab name):

```
tar zcvf your-lab.tgz your-lab/
```



3. Download file `your-lab.tgz` to your laptop:



4. Uncompress it:

```
tar zxvf your-lab.tgz
```

5. Note that you could download all your labs at once by compressing the main folder containing all your labs. If you have a lot of data inside the lab folder, you may want to clean the folder before creating the archive.

2 Running the labs locally

This section describes how you can run the labs on your local machine.

2.1 Installing miniconda

2.1.1 On Linux and macOS

We download the installation script in `Downloads/` and install miniconda in the home directory:

First, select the correct option for you system:

- Linux:

```
MINICONDA_SCRIPT=Miniconda3-latest-Linux-x86_64.sh
```

- macOS/arm (M1):

```
MINICONDA_SCRIPT=Miniconda3-latest-MacOSX-arm64.sh
```

- macOS/x86:

```
MINICONDA_SCRIPT=Miniconda3-latest-MacOSX-x86_64.sh
```

Then, install with:

```
cd ${HOME}/Downloads
wget "https://repo.anaconda.com/miniconda/${MINICONDA_SCRIPT}"
bash ${MINICONDA_SCRIPT} -b -p ${HOME}/miniconda3
```

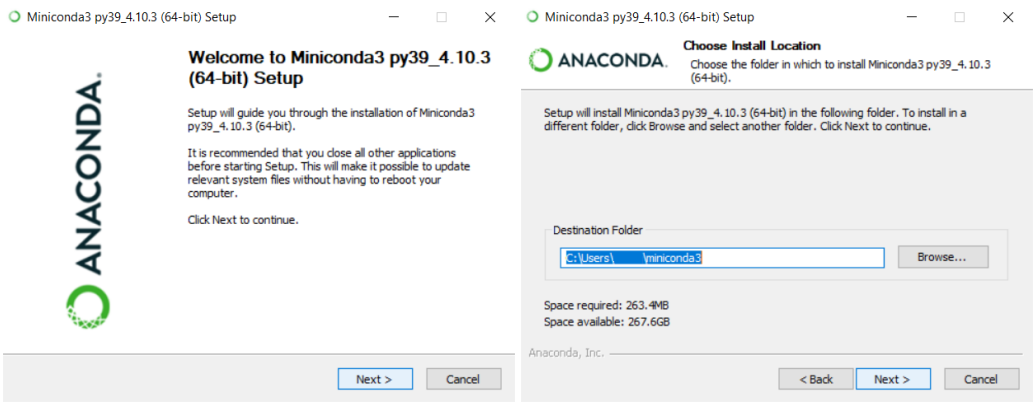
A folder named `miniconda3` is now present in your home directory, and this is where your conda environments will be created.

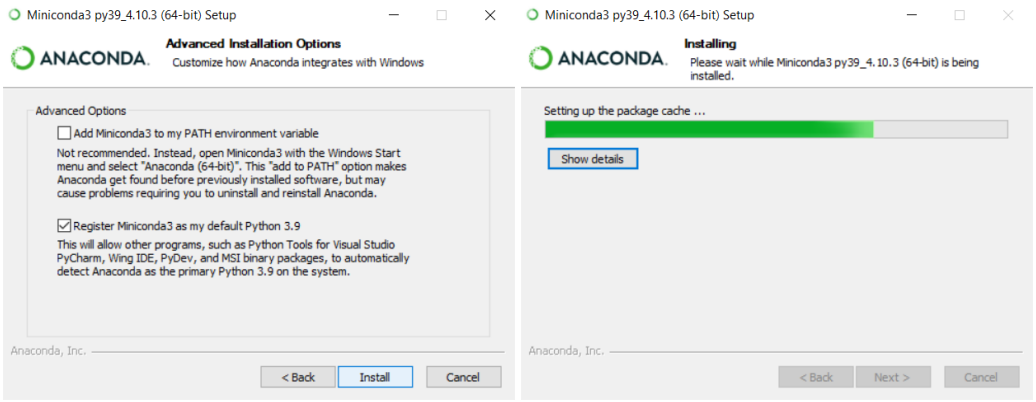
2.1.2 On Windows

Download the Windows installer (probably your machine is in 64-bits) from this link:

https://docs.conda.io/en/latest/miniconda.html

Double click on the installer `Miniconda3-latest-Windows-x86_64.exe` and follow the instructions: Agree to the terms and conditions, install "just for you", and don't set the `PATH` variable:





2.2 Activating the conda `(base)` environment

2.2.1 On Linux and macOS

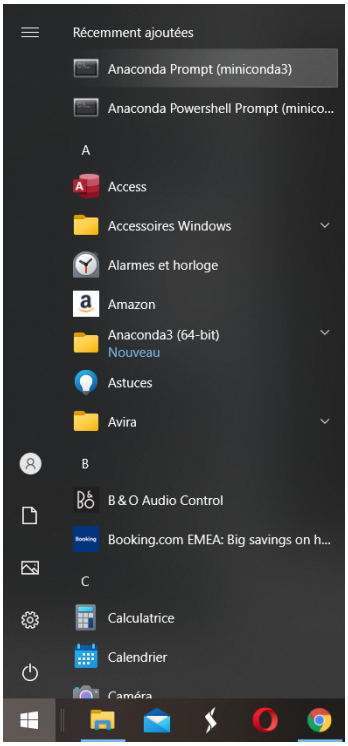
If you installed miniconda3 in `${HOME}/miniconda3/` as suggested above, you can activate your conda environment, in a new terminal with:

```
eval "$(${HOME}/miniconda3/bin/conda shell.bash hook)"
```

If you chose a different location, adapt the path accordingly. After that, your prompt should start with `(base)`.

2.2.2 On Windows

You can start a command line interface by clicking on the Windows button and click on "Anaconda3 Prompt (miniconda3)":



2.3 Creating the (labs) conda environment

Create a file named `labs.yaml` containing:

```
name: labs
dependencies:
  - python=3.10
  - pytorch::pytorch=2.0
  - pytorch::torchvision
  - pytorch::torchaudio
  - pytorch::pytorch-cuda=11.8 # Only Linux and Windows
  - nvidia::cuda=11.8 # Only Linux and Windows
  - jupyter
  - matplotlib
  - pandoc
  - numpy
  - scipy
  - scikit-learn
  - scikit-learn-intelex
  - pip
  - pip:
    - opencv-python
```

> *If you copy paste this content to a text file, make sure that* ***the leading spaces are there****:*
> - ***2 spaces*** *for conda packages like* `python=3.10`*, and*
> - ***4 spaces*** *for the lines below* `pip:`*.*
>
> *Your* `.yaml` *file should look like above.*

You may need to adapt the PyTorch and CUDA versions. Please check the compatibilities between your NVIDIA driver and the CUDA version.

In your `(base)` environment (see section 2.3), create the `(labs)` environment with:

```
(base) conda env create -f labs.yaml
```

This command may take several minutes (15 min) because it may need to download the packages if not already done during the installation of a previous environment.

2.4 Running the notebooks

We assume that you want to run locally a notebook named `your-notebook.ipynb`. In your `(base)` environment (see section 2.3) activate the `(labs)` environment with:

```
(base) conda activate labs
```

and you should get a prompt starting with `(base)`.

You can launch Jupyter with:

```
(labs) cd where-your-notebook-is/
(labs) ls
your-notebook.ipynb
(labs) jupyter notebook
```

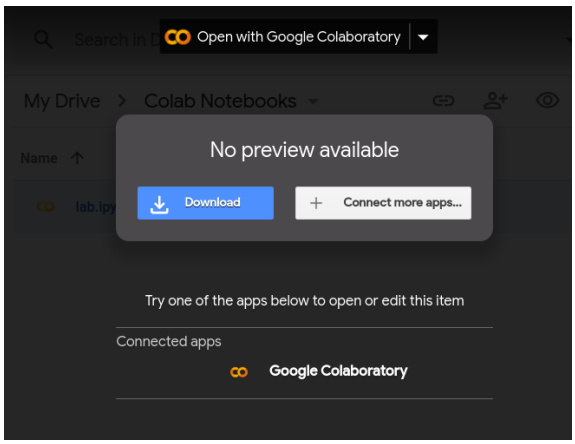
and open your notebook by double clicking on it, and run it.

3 Running the labs on colab

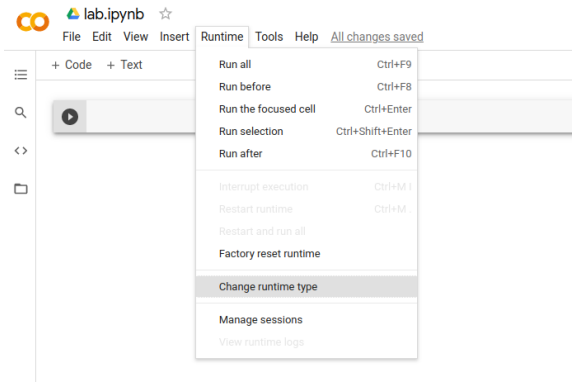
This section describes how you can run a lab using a GPU on Google colab.

Connect to your Google Drive account and upload the notebook you want to run.

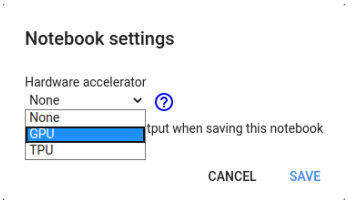
Open the notebook by double clicking on it and selecting `Open with Google Colaboratory`:



By default, the instance of colab is using a CPU, and a GPU instance can be chosen instead by clicking `Runtime > Change runtime type`,

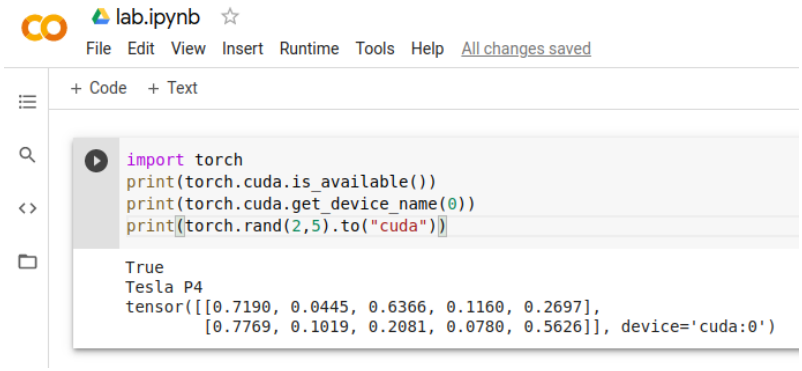


and then selecting `GPU`:



Once the instance is restarted, you may try the following piece of code and check that the GPU is active indeed:

```
import torch
print(torch.cuda.is_available())
print(torch.cuda.get_device_name(0))
print(torch.rand(2,5).to("cuda"))
```



Note that the images of the documentation won't appear in the notebook under Colab, because there are included as HTML tags, where Colab used images in base 64. However, the Matplotlib graphs are displayed.