Intro to ML 22 / 23

Lab 01 - Introduction

Outline

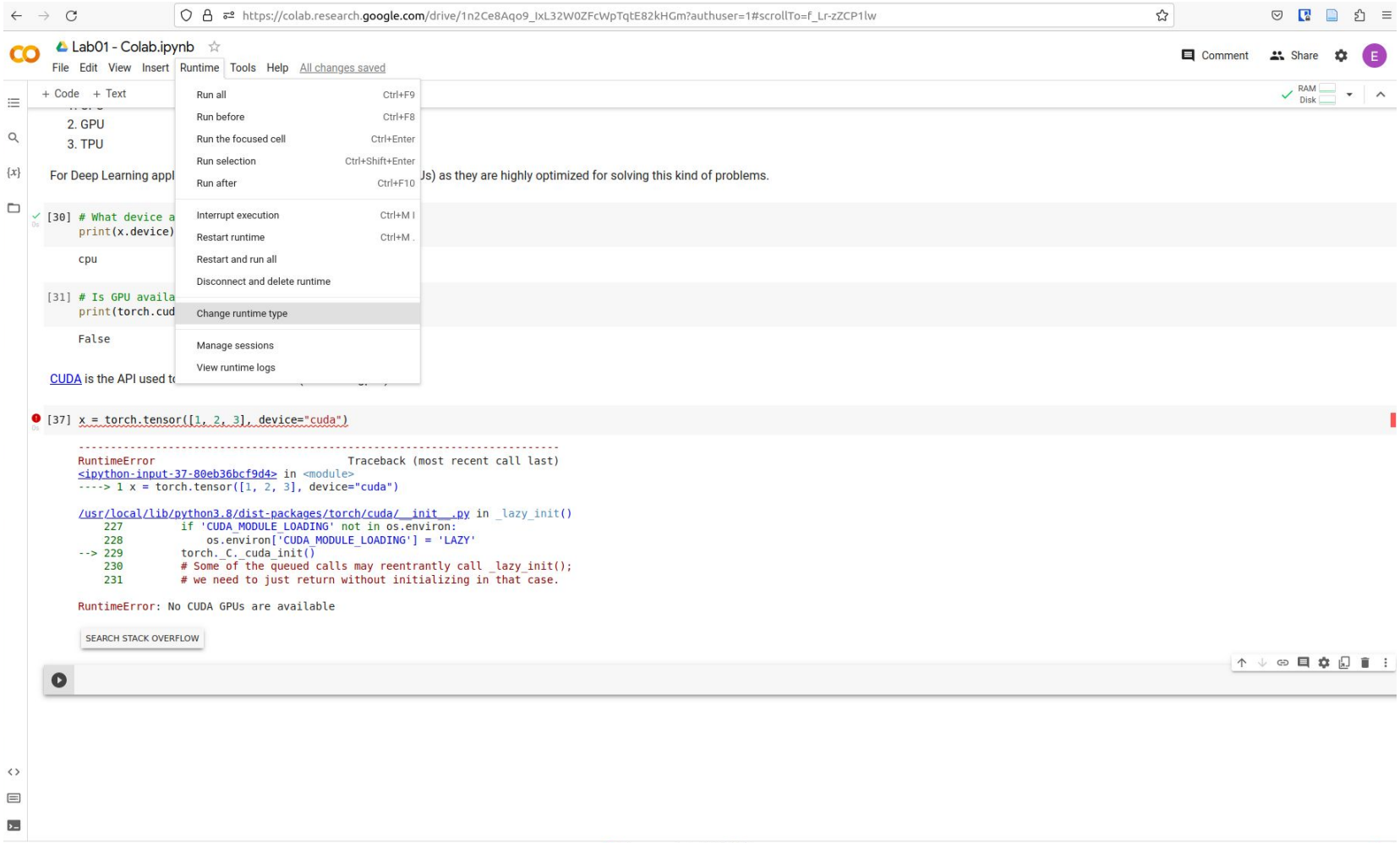
1. Colab
2. Anaconda / Package Manager
3. Azure Machine
4. IDE
5. Github
6. Classification with scikit-learn

Link to Colab Notebook

Colab

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

- Make a copy of the notebook on your Drive
- Jupyter notebook environment hosted by Google
- No setup required (basically)
- Allows running code on GPU (12 hour maximum of GPU runtime)


https://colab.research.google.com/drive/1n2Ce8Aqo9_IxL32W0ZFcWpTqtE82kHGm?authuser=1#scrollTo=f_Lr-zZCP1lw
Lab01 - Colab.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Run all Ctrl+F9
Run before Ctrl+F8
Run the focused cell Ctrl+Enter
Run selection Ctrl+Shift+Enter
Run after Ctrl+F10
Interrupt execution Ctrl+M I
Restart runtime Ctrl+M .
Restart and run all
Disconnect and delete runtime
Change runtime type
Manage sessions
View runtime logs
2. GPU
3. TPU
For Deep Learning appl
Js) as they are highly optimized for solving this kind of problems.
[30] # What device a
print(x.device)
cpu
[31] # Is GPU availa
print(torch.cud
False
CUDA is the API used t
[37] x = torch.tensor([1, 2, 3], device="cuda")
RuntimeError Traceback (most recent call last)
<ipython-input-37-80eb36bcf9d4> in <module>
----> 1 x = torch.tensor([1, 2, 3], device="cuda")
/usr/local/lib/python3.8/dist-packages/torch/cuda/__init__.py in _lazy_init()
227 if 'CUDA_MODULE_LOADING' not in os.environ:
228 os.environ['CUDA_MODULE_LOADING'] = 'LAZY'
--> 229 torch._C._cuda_init()
230 # Some of the queued calls may reentrantly call _lazy_init();
231 # we need to just return without initializing in that case.
RuntimeError: No CUDA GPUs are available
SEARCH STACK OVERFLOW
0s completed at 6:23 PM

Lab01 - Colab.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

2. GPU
3. TPU

For Deep Learning applications, is better to run code on GPUs (or TPUs) as they are highly optimized for solving this kind of problems.

```
[30] # What device are we using?
     print(x.device)

     cpu
```

```
[31] # Is GPU available?
     print(torch.cuda.is_available())

     False
```

CUDA is the API used to access NVIDIA GPUs (i.e. all the gpus)

```
[37] x = torch.tensor([1, 2, 3], device="cuda")

     ---------------------------------------------------------------------------
     RuntimeError                              Traceback (most recent call last
     <ipython-input-37-80eb36bcf9d4> in <module>
     ----> 1 x = torch.tensor([1, 2, 3], device="cuda")

     /usr/local/lib/python3.8/dist-packages/torch/cuda/__init__.py in _lazy_in
         227         if 'CUDA_MODULE_LOADING' not in os.environ:
         228             os.environ['CUDA_MODULE_LOADING'] = 'LAZY'
     --> 229         torch._C._cuda_init()
         230         # Some of the queued calls may reentrantly call _lazy_init
         231         # we need to just return without initializing in that case

     RuntimeError: No CUDA GPUs are available

     SEARCH STACK OVERFLOW
```

Notebook settings

Hardware accelerator

GPU

GPU class

Standard

Want access to premium GPUs?

Purchase additional compute units

Omit code cell output when saving this notebook

Cancel Save

0s completed at 6:23 PM

Anaconda

Anaconda is a package and environment manager

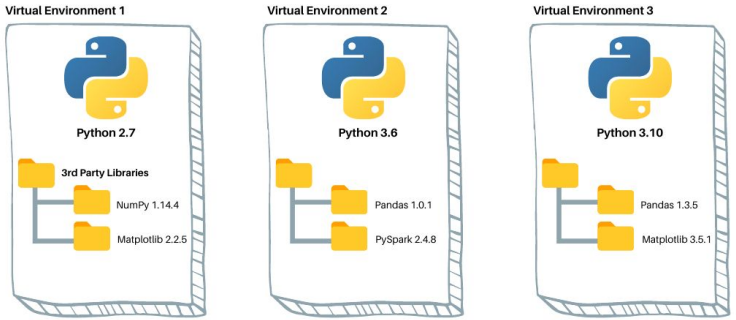
- Create virtual environments
- Easily install packages and dependencies

Install Anaconda

Anaconda

What is a virtual environment?

- it is an isolated environment virtually created on your machine

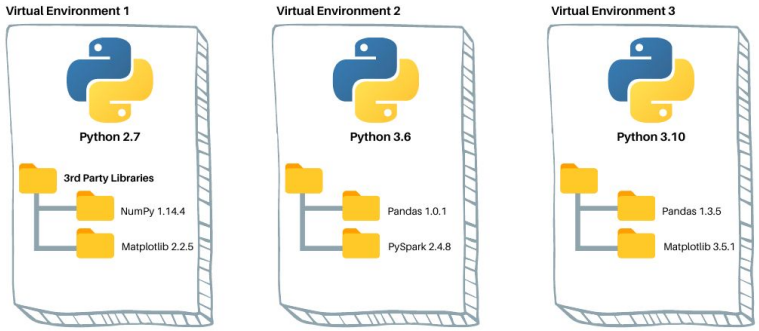


dataquest.io

Anaconda

Use virtual environment for each different project, it will make it easier:

- keep the dependencies separated
- share your configuration with others
- reproduce what you were doing



dataquest.io

Anaconda

Basic commands

- create new env: **conda create --name lab01**
- activate env: **conda activate lab01**

Install packages you want, anaconda will take care off all the dependencies

Azure

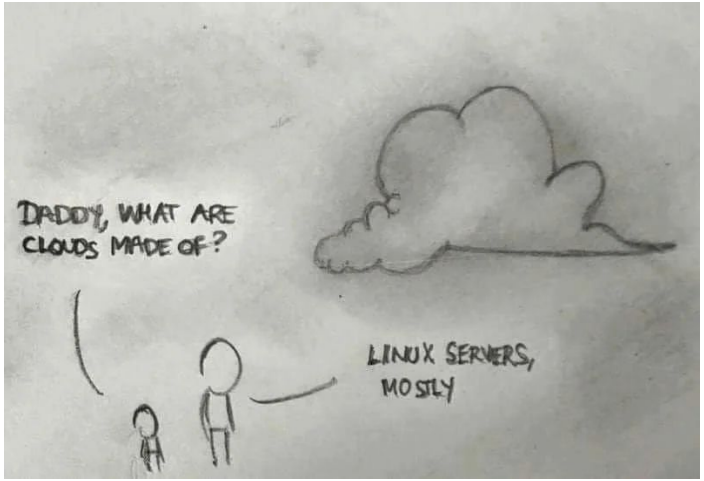
It is a cloud computing platform offered by Microsoft (other famous similar services are Google Cloud and Amazon AWS).

- You have access to a virtual machine on a cluster, you can run computational intensive code.
- You have to setup the machine
- You have access to 50 hrs of computing resources offered by DISI

Turn off your machine if you are not using it!

Azure

How to connect to the machine?



Once you are logged in, you can follow the anaconda step to set up your environment

Azure

Useful commands:

- ssh yourusername@server
- scp or rsync to transfer files to from the machine

  ```
  rsync -r /your/local/path username@server:/your/remote/path
  ```

- Other useful commands
  - cd, ls, mkdir, rm, cp
  - https://ubuntu.com/tutorials/command-line-for-beginners#1-overview
  - https://www.educative.io/blog/bash-shell-command-cheat-sheet

IDE

Integrated Development Environment: will let you easily work with the above stuff.

- VScode
- Pycharm

PyCharm

Install:
https://www.jetbrains.com/help/pycharm/installation-guide.html

Pycharm professional is free for students, you have to apply here:
https://www.jetbrains.com/community/education/#students

Github

- Versioning of your code
- Share the code
- Make sure it is not lost

Use github! [Basic commands]