Tutorial 1: course environment

CIT-690E: Deep Learning, Spring 22

Ammar Sherif[1]

Nile University

March 7, 2022

[1] This is an adapted presentation from Eng. Ahmed Hosny, GitHub

Outline

Contacts

- Lab Session: after the lecture
- Q&A on Teams, or Moodle
- Office Hours: room 220 on Tuesday 16:30
  online: confirmation via email
- Email: ASherif [at] nu.edu.eg

What is Google Colab?

Google Colab

Cloud-based notebook service hosted by Google, which is equivalent to **Jupyter Lab**. It enables you write and execute your code on pre-configured machines.

What is Google Colab?

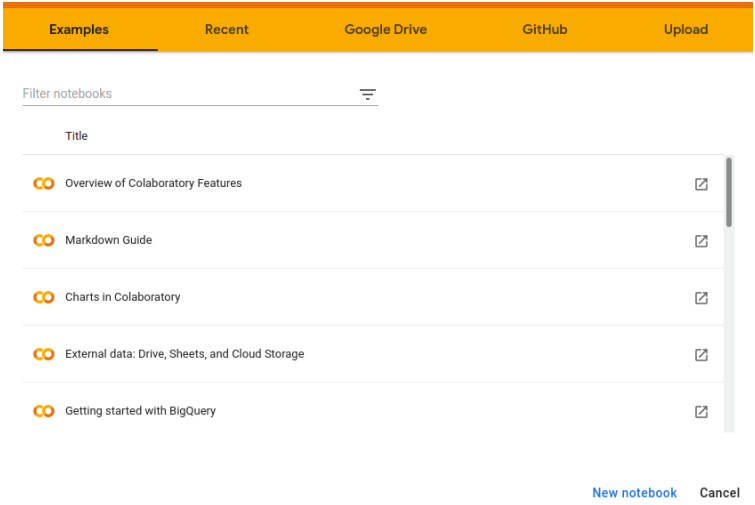
Google Colab

Cloud-based notebook service hosted by Google, which is equivalent to **Jupyter Lab**. It enables you write and execute your code on pre-configured machines.

- Notebooks
- Access to Google Drive
- Free GPUs and TPUs
- pre-installed packages for DL: `PyTorch`

Creating a Notebook

You can create a notebook via `https://colab.research.google.com/`.



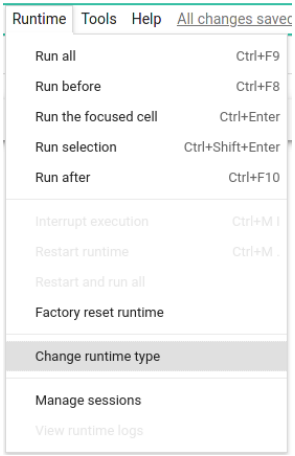
Creating a Notebook

You can create a notebook via `https://colab.research.google.com/`.

- Create a new notebook using `New notebook`
- Open a notebook hosted on your `Google Drive`
- Open a notebook from GitHub
- Upload a local notebook

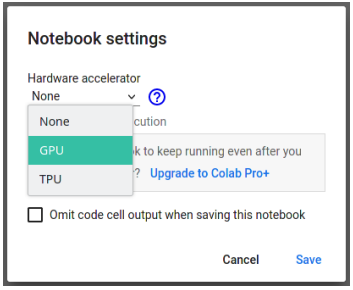
Runtime Environments

1. Click `Runtime -> Change runtime type`



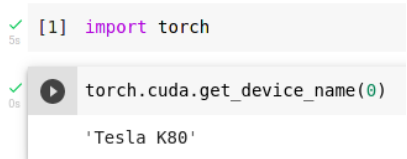
Runtime Environments

1. Click `Runtime -> Change runtime type`
2. Choose your runtime type



Runtime Environments

1. Click `Runtime -> Change runtime type`
2. Choose your runtime type
3. Try it out printing the GPU device. This need not be the same for you.



Performance Comparison

As shown in [Colab vs Laptop (Radečić, 2020)], **Fasion-MNIST** dataset was used for comparison between training on local laptops and Colab on the below architecture, of $\approx 150K$ trainable parameters:

```
CNN(
  (conv1): Conv2d(1, 8, kernel_size=(3, 3), stride=(1, 1))
  (conv2): Conv2d(8, 20, kernel_size=(3, 3), stride=(1, 1))
  (fc1): Linear(in_features=500, out_features=256, bias=True)
  (fc2): Linear(in_features=256, out_features=64, bias=True)
  (fc3): Linear(in_features=64, out_features=10, bias=True)
)
```

Performance Comparison

As shown in [Colab vs Laptop (Radečić, 2020)], **Fasion-MNIST** dataset was used for comparison between training on local laptops and Colab on the below architecture, of $\approx 150K$ trainable parameters:

```
CNN(
  (conv1): Conv2d(1, 8, kernel_size=(3, 3), stride=(1, 1))
  (conv2): Conv2d(8, 20, kernel_size=(3, 3), stride=(1, 1))
  (fc1): Linear(in_features=500, out_features=256, bias=True)
  (fc2): Linear(in_features=256, out_features=64, bias=True)
  (fc3): Linear(in_features=64, out_features=10, bias=True)
)
```

Laptops used

- Lenovo ThinkPad T480s: Intel i7–8550U @ 1.8 GHz, 16GB RAM
- Lenovo Legion Y540: Intel i5–9300H @ 2.4 GHz, Nvidia GeForce 1650 4GB, 16GB RAM
- MacBook Pro 2019 13":Intel i5-QC @ 1.4 GHz, 8GB RAM

Performance Comparison

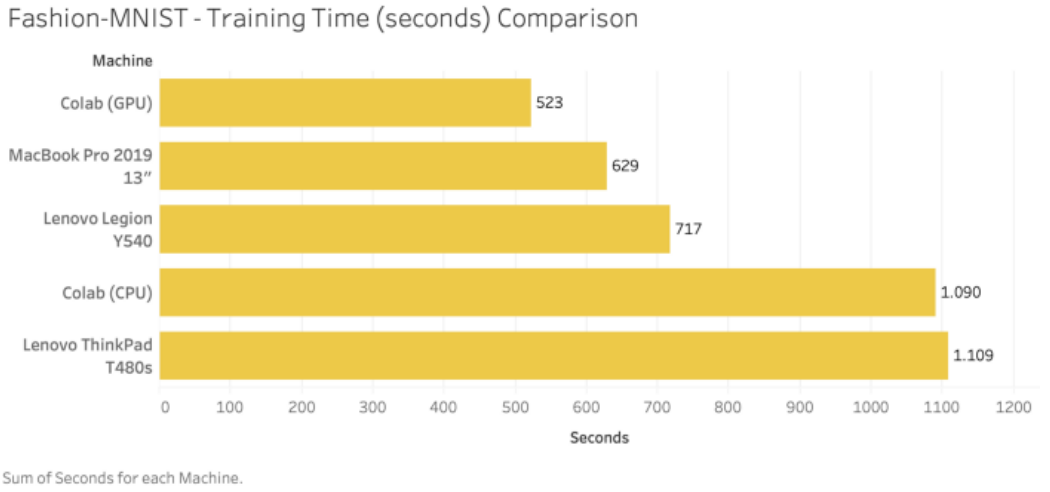


Figure: Results from [Colab vs Laptop (Radečić, 2020)]

Colab Pro Comparison

Radečić in [Colab Pro (Radečić, 2020)] compared between both using Pneumonia Dataset from Kaggle.

Colab (Free)

- Free
- Intel(R) Xeon(R) CPU @ 2.20GHz

Colab Pro

- $10/mo
- Intel(R) Xeon(R) CPU @ 2.30GHz

Colab Pro Comparison

Radečić in [Colab Pro (Radečić, 2020)] compared between both using Pneumonia Dataset from Kaggle.

Colab (Free)

- Free
- Intel(R) Xeon(R) CPU @ 2.20GHz
- Disk Size: 34GB

Colab Pro

- $10/mo
- Intel(R) Xeon(R) CPU @ 2.30GHz
- Disk Size: 34GB

Colab Pro Comparison

Radečić in [Colab Pro (Radečić, 2020)] compared between both using Pneumonia Dataset from Kaggle.

Colab (Free)

- Free
- Intel(R) Xeon(R) CPU @ 2.20GHz
- Disk Size: 34GB
- Runtime Duration: 12 Hours

Colab Pro

- $10/mo
- Intel(R) Xeon(R) CPU @ 2.30GHz
- Disk Size: 34GB
- Runtime Duration: 24 Hours

Colab Pro Comparison

Radečić in [Colab Pro (Radečić, 2020)] compared between both using Pneumonia Dataset from Kaggle.

Colab (Free)

- Free
- Intel(R) Xeon(R) CPU @ 2.20GHz
- Disk Size: 34GB
- Runtime Duration: 12 Hours
- GPU: Tesla K80 (11GB)

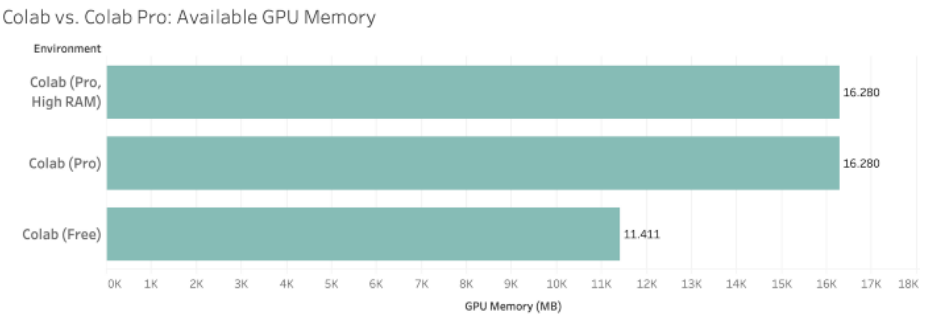
Colab Pro

- $10/mo
- Intel(R) Xeon(R) CPU @ 2.30GHz
- Disk Size: 34GB
- Runtime Duration: 24 Hours
- GPU: Tesla P100-PCIE-16GB

Colab Pro Comparison

Radečić in [Colab Pro (Radečić, 2020)] compared between both using Pneumonia Dataset from Kaggle.

Colab (Free) Colab Pro



Colab Pro Comparison

Radečić in [Colab Pro (Radečić, 2020)] compared between both using Pneumonia Dataset from Kaggle.

Colab (Free)

- Free
- Intel(R) Xeon(R) CPU @ 2.20GHz
- Disk Size: 34GB
- Runtime Duration: 12 Hours
- GPU: Tesla K80 (11GB)
- RAM: 12GB

Colab Pro

- $10/mo
- Intel(R) Xeon(R) CPU @ 2.30GHz
- Disk Size: 34GB
- Runtime Duration: 24 Hours
- GPU: Tesla P100-PCIE-16GB
- RAM: 12GB (Default) - 26GB (High-RAM)

Colab Pro Comparison

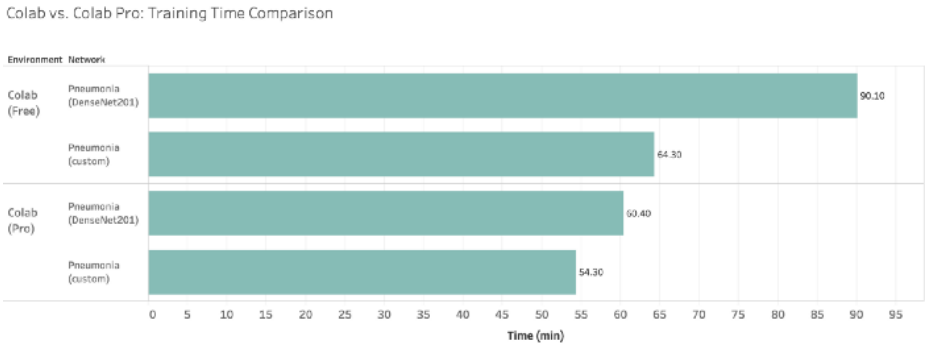
Radečić in [Colab Pro (Radečić, 2020)] compared between both using Pneumonia Dataset from Kaggle. He, also used a pre-trained **DenseNet201**, modifying last layer, and the below custom architecture to compare between them

```
CustomModel(
  (conv1): Conv2d(3, 32, kernel_size=(3, 3), stride=(1, 1))
  (conv2): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1))
  (conv3): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1))
  (conv4): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1))
  (conv5): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1))
  (fc1): Linear(in_features=6400, out_features=128, bias=True)
  (out): Linear(in_features=128, out_features=2, bias=True)
)
```

Colab Pro Comparison

Radečić in [Colab Pro (Radečić, 2020)] compared between both using Pneumonia Dataset from Kaggle.

Colab (Free) Colab Pro



Colab Pro Comparison

Radečić in [Colab Pro (Radečić, 2020)] compared between both using Pneumonia Dataset from Kaggle.

Colab (Free)	Colab Pro
• Free	• $10/mo
• Intel(R) Xeon(R) CPU @ 2.20GHz	• Intel(R) Xeon(R) CPU @ 2.30GHz

Colab Pro+

Google recently, started offering `Colab Pro+`, where your notebook can run even while you are ofline, for $50/mo, for 24 Hours max. Neverthless, still, the notebook disconnects when idle, no cells running.

Git & GitHub

Git

It is a Version Control system.

Git & GitHub

Git

It is a Version Control system.

It is used for the below cases:

- Synchronization and Collaboration

Git & GitHub

Git

It is a Version Control system.

It is used for the below cases:

- Synchronization and Collaboration
- Informative Changes Tracking, with messages

Git & GitHub

Git

It is a Version Control system.

It is used for the below cases:

- Synchronization and Collaboration
- Informative Changes Tracking, with messages
- Undo changes, and restoration

Git & GitHub

Git

It is a Version Control system.

It is used for the below cases:

- Synchronization and Collaboration
- Informative Changes Tracking, with messages
- Undo changes, and restoration
- Track Ownership

Git & GitHub

Git

It is a Version Control system.

It is used for the below cases:

- Synchronization and Collaboration
- Informative Changes Tracking, with messages
- Undo changes, and restoration
- Track Ownership
- Branching and Merging

Git & GitHub

Git

It is a Version Control system.

Installing Git

You can install git by visiting the download section: `https://git-scm.com/downloads` and pickup an installation for your OS.

Git & GitHub

Git

It is a Version Control system.

Installing Git

You can install git by visiting the download section: `https://git-scm.com/downloads` and pickup an installation for your OS. Also, although we are introducing some of the basics of git below, you are highly encouraged to check the free Pro Git book by Scott Chacon: `https://git-scm.com/book/en/v2`

Git & GitHub

Git

It is a Version Control system.

Git Commands

- `init`: initialize a git repository

Git & GitHub

Git

It is a Version Control system.

Git Commands

- `init`: initialize a git repository
- `add`: add file(s) to file index

Git & GitHub

Git

It is a Version Control system.

Git Commands

- `init`: initialize a git repository
- `add`: add file(s) to file index
- `status`: check status of the current working tree

Git & GitHub

Git

It is a Version Control system.

Git Commands

- `init`: initialize a git repository
- `add`: add file(s) to file index
- `status`: check status of the current working tree
- `commit`: save snapshot as entry in history

Git & GitHub

Git

It is a Version Control system.

Git Commands

- `init`: initialize a git repository
- `add`: add file(s) to file index
- `status`: check status of the current working tree
- `commit`: save snapshot as entry in history
- `push`: push the changes to remote repository

Git & GitHub

Git

It is a Version Control system.

Git Commands

- `init`: initialize a git repository
- `add`: add file(s) to file index
- `status`: check status of the current working tree
- `commit`: save snapshot as entry in history
- `push`: push the changes to remote repository
- `pull`: pull changes from remote repository

Git & GitHub

Git

It is a Version Control system.

Git Commands

- `init`: initialize a git repository
- `add`: add file(s) to file index
- `status`: check status of the current working tree
- `commit`: save snapshot as entry in history
- `push`: push the changes to remote repository
- `pull`: pull changes from remote repository
- `clone`: clone a remote repository

Git & GitHub

Git

It is a Version Control system.

Git Commands

- `init`: initialize a git repository
- `add`: add file(s) to file index
- `status`: check status of the current working tree
- `commit`: save snapshot as entry in history
- `push`: push the changes to remote repository
- `pull`: pull changes from remote repository
- `clone`: clone a remote repository
- `log`: show the log of project

Git & GitHub

Git

It is a Version Control system.

Git Commands

- `init`: initialize a git repository
- `add`: add file(s) to file index
- `status`: check status of the current working tree
- `commit`: save snapshot as entry in history
- `push`: push the changes to remote repository
- `pull`: pull changes from remote repository
- `clone`: clone a remote repository
- `reset`: revert commits (soft, hard)

Git & GitHub

Git

It is a Version Control system. Therefore, as we see, `git` is a tool used on our local machines to communicate with local and remote repositories.

Git & GitHub

Git

It is a Version Control system. Therefore, as we see, `git` is a tool used on our local machines to communicate with local and remote repositories.

GitHub

GitHub is a cloud-based hosting platform for repositories, remote.

Git & GitHub

Git

It is a Version Control system. Therefore, as we see, `git` is a tool used on our local machines to communicate with local and remote repositories.

GitHub

GitHub is a cloud-based hosting platform for repositories, remote.

Further Resources

- `https://git-scm.com/docs`, the official documentation of `git`
- `https://docs.github.com/en`, the official documentation of `GitHub`
- `https://training.github.com/downloads/github-git-cheat-sheet.pdf`, cheat sheet for git while using GitHub.

PyTorch

- open-source Deep Learning library

PyTorch

- open-source Deep Learning library
- by Facebook

PyTorch

- open-source Deep Learning library
- by Facebook
- optimized for GPU and CPU

PyTorch

- open-source Deep Learning library
- by Facebook
- optimized for GPU and CPU
- supports Python, C++, and Java

PyTorch

- open-source Deep Learning library
- by Facebook
- optimized for GPU and CPU
- supports Python, C++, and Java
- Other products built upon it: Tesla Autopilot, Uber's Pyro, Hugging Face's Transformers, PyTorch Lightning, and Catalyst.

PyTorch

- open-source Deep Learning library
- by Facebook
- optimized for GPU and CPU
- supports Python, C++, and Java
- Other products built upon it: Tesla Autopilot, Uber's Pyro, Hugging Face's Transformers, PyTorch Lightning, and Catalyst.
- Primary data type: Tensors

References

Dario Radečić (2020)

Google Colab: How does it compare to a GPU-enabled laptop?

url: https://towardsdatascience.com/google-colab-how-does-it-compare-to-a-gpu-enabled-laptop-851c1e0a2ca9.

Dario Radečić (2020)

Colab Pro: Is it Worth the Money?

url: https://towardsdatascience.com/colab-pro-is-it-worth-the-money-32a1744f42a8.

Thank You