



# STAT 453: Introduction to Deep Learning and Generative Models

---

Ben Lengerich

Lecture 07: Cloud computing resources

September 24, 2025



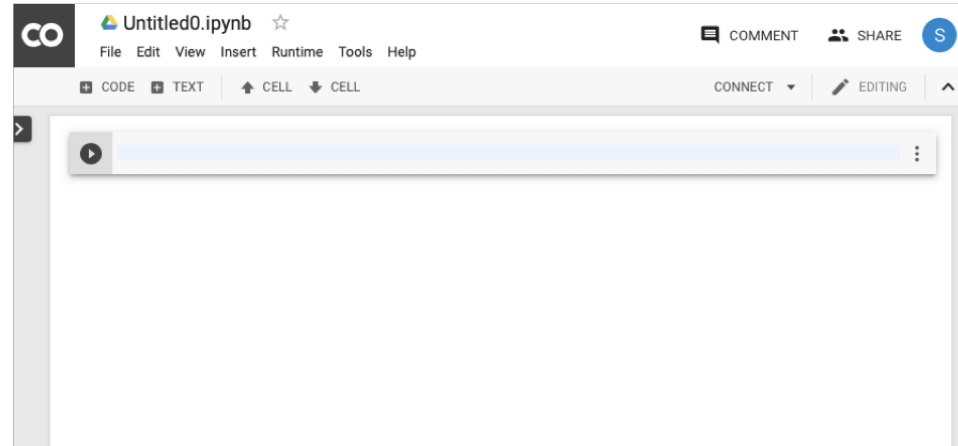
# Today: Computing partial derivatives with PyTorch

---

1. Google Colab
2. Center for High-Throughput Computing

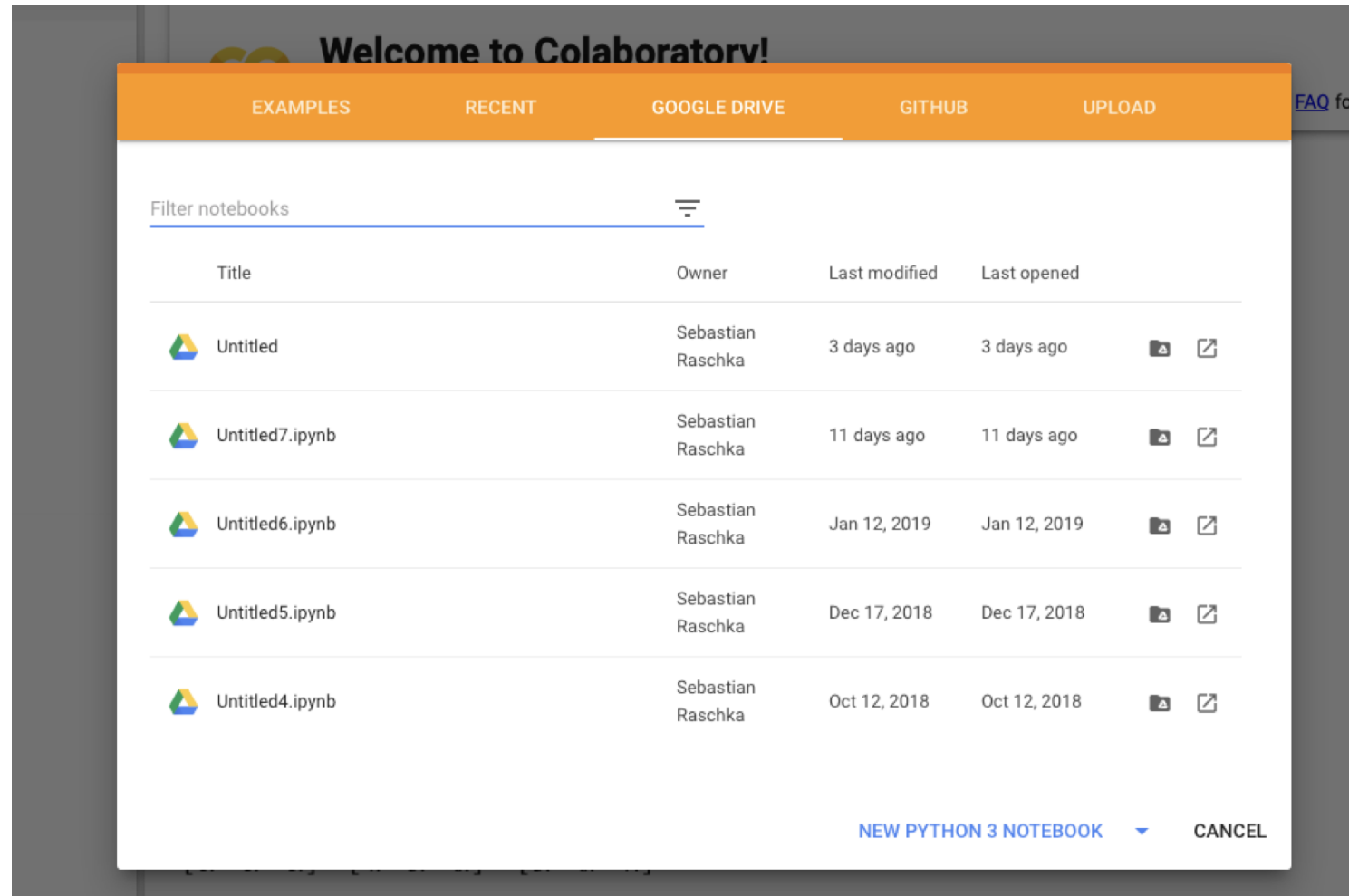
# Google Colab

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



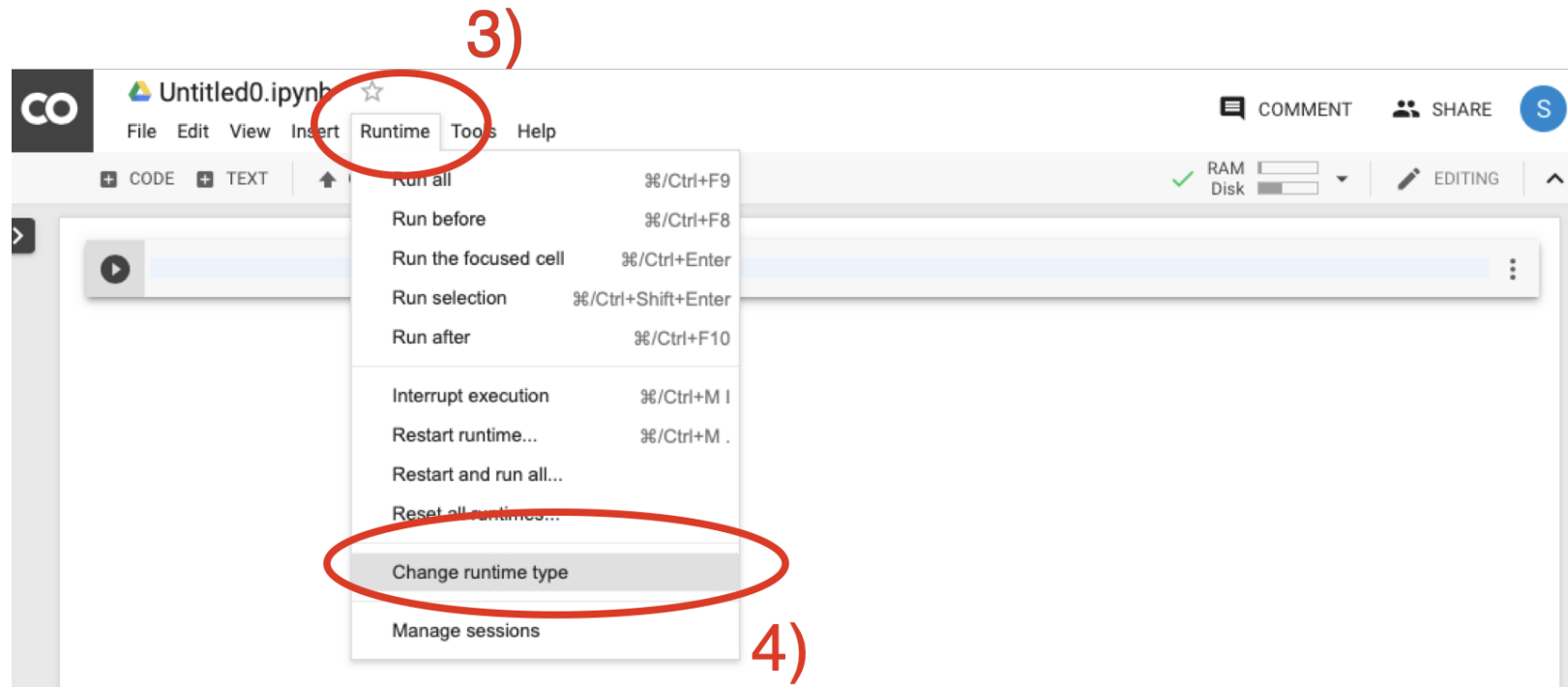
- Free Google-flavored Jupyter Notebooks in the Cloud
- For each notebook, they spin up a custom (Linux-based) computing instance
- Computations limited to ~12 h though; you won't lose your notebook, but computations will be interrupted
- Maybe useful for quick testing/experimenting/sharing (but maybe tedious as you need to reinstall packages each time)

# Google Colab



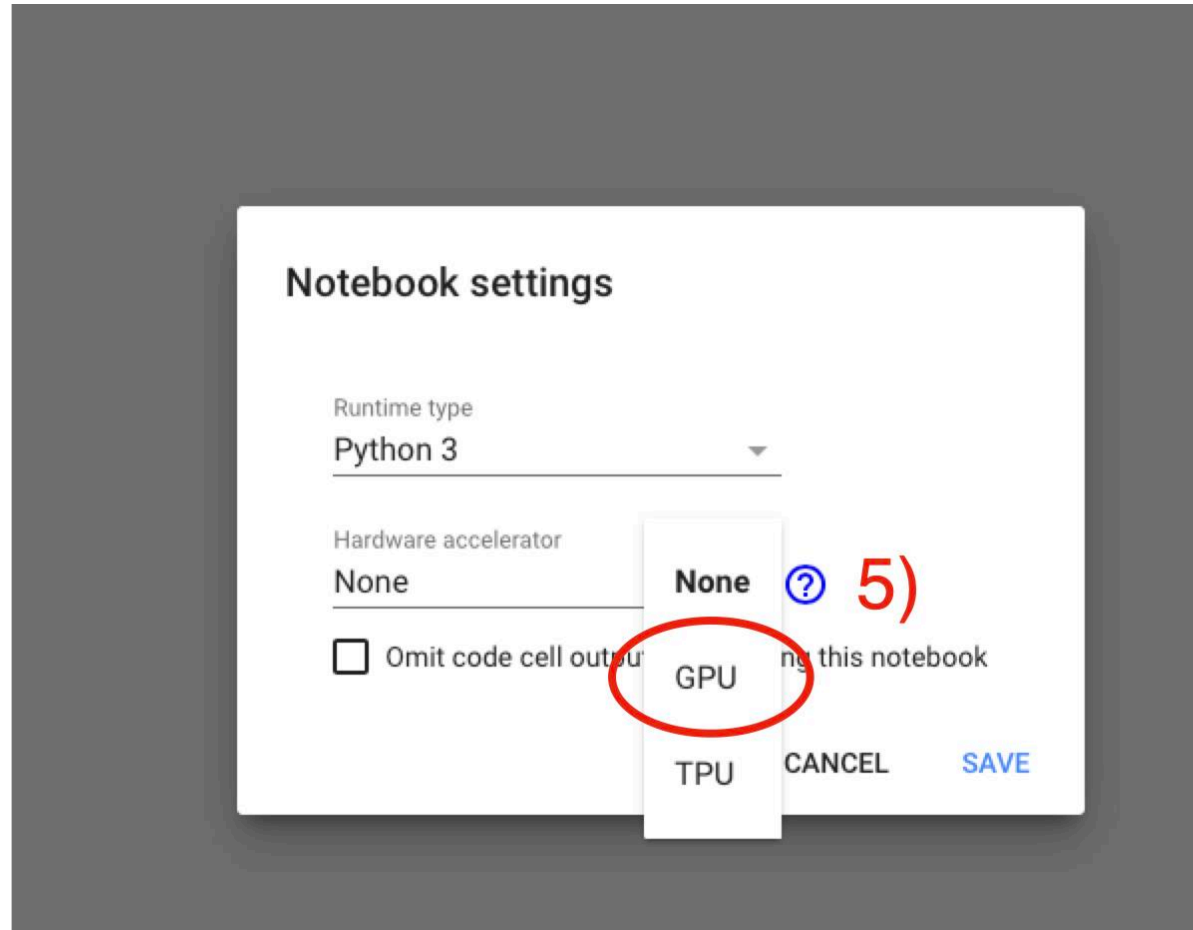
Menu appears if you visit <https://colab.research.google.com>

# Google Colab



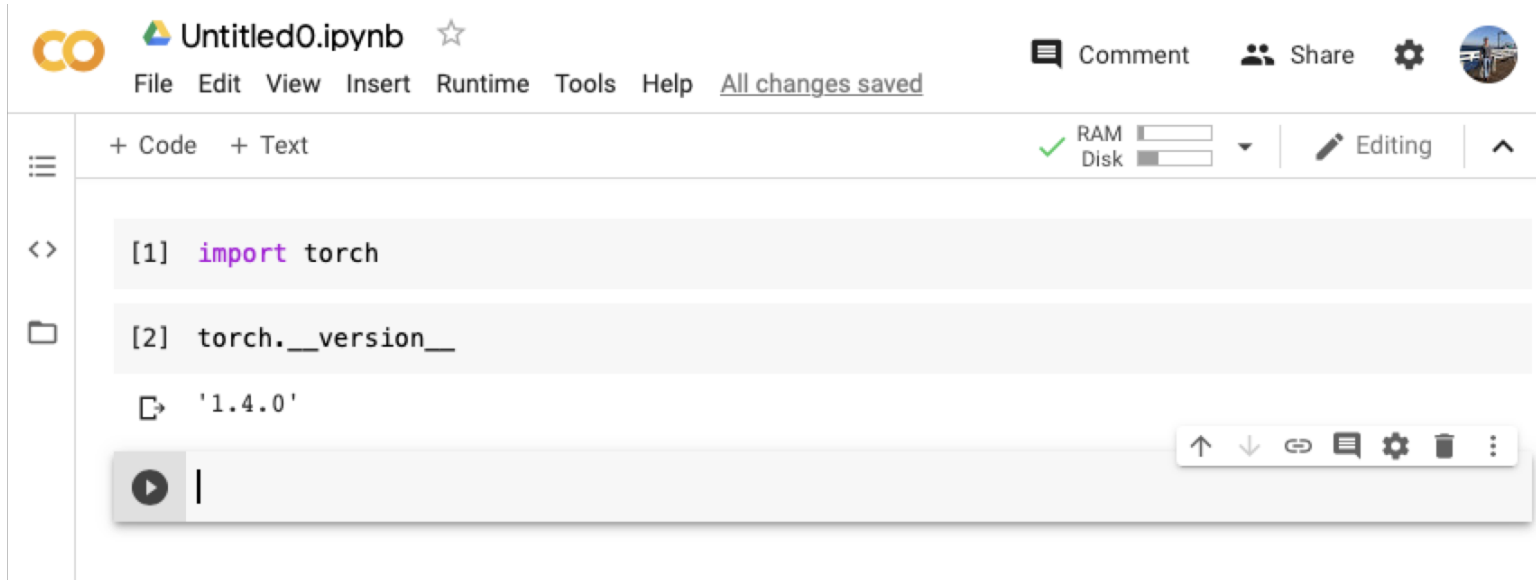
Follow these steps for running code on GPU later (default is CPU)

# Google Colab



Follow these steps for running code on GPU later (default is CPU)

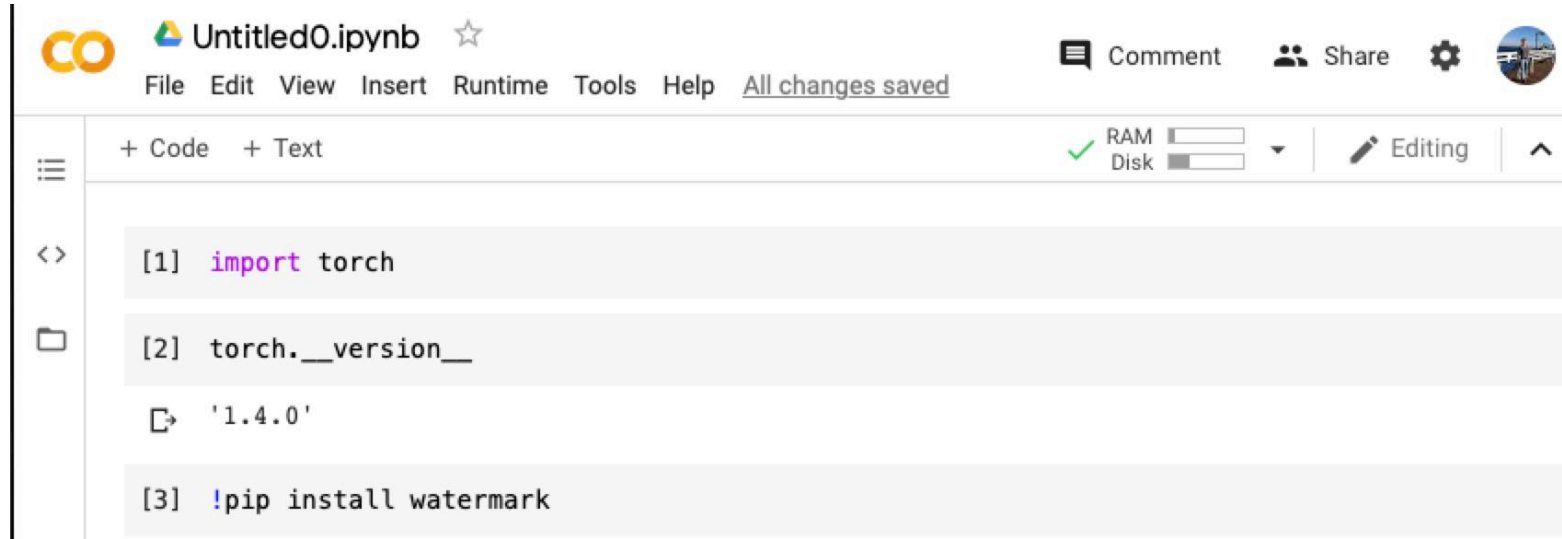
# Google Colab



The screenshot shows the Google Colab interface for a notebook titled 'Untitled0.ipynb'. The top menu bar includes 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help', with a status indicator 'All changes saved'. On the right, there are buttons for 'Comment', 'Share', and a settings gear. Below the menu, a toolbar shows '+ Code' and '+ Text' buttons, along with RAM and Disk usage indicators (both at 0%) and an 'Editing' mode button. The notebook content consists of two code cells. The first cell contains the code `[1] import torch`. The second cell contains the code `[2] torch.__version__` and has been executed, displaying the output `'1.4.0'`. A third, empty code cell is visible at the bottom with a play button icon on the left and a toolbar on the right.

- This is nice! It appears that PyTorch is already pre-installed now (it wasn't always the case)

# Google Colab



The screenshot shows the Google Colab interface for a notebook titled "Untitled0.ipynb". The top menu bar includes "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help", with a status message "All changes saved". On the right, there are icons for "Comment", "Share", and a user profile. Below the menu, a toolbar shows "+ Code" and "+ Text" buttons, along with RAM and Disk usage indicators and an "Editing" mode selector. The notebook contains three code cells:

```
[1] import torch
```

```
[2] torch.__version__
```

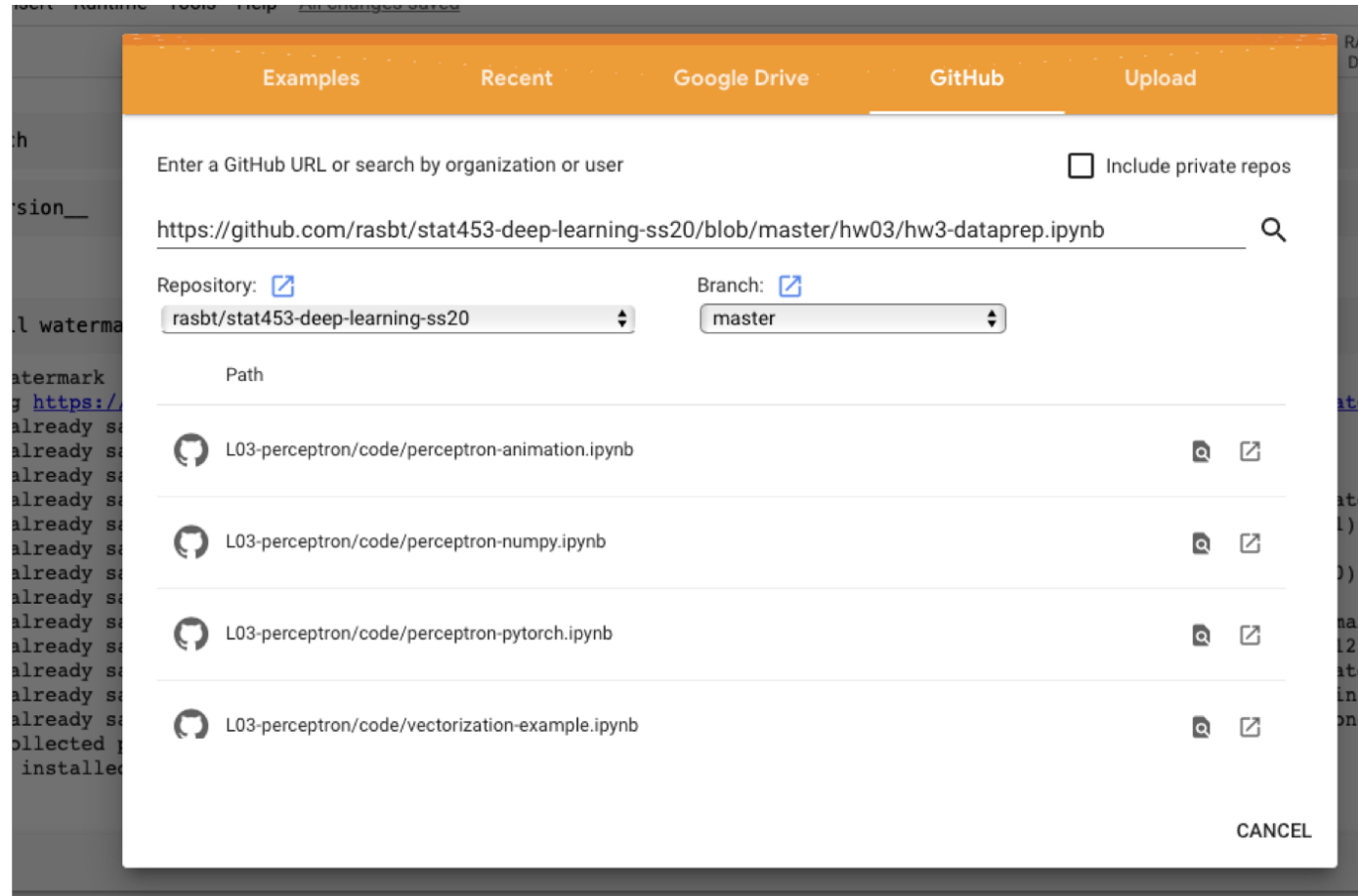
```
'1.4.0'
```

```
[3] !pip install watermark
```

- In any case, if you'd like/need to install packages, you can do it as shown in the example above
- Note that in Jupyter Notebooks, the "!" indicates that what follows on that line is a "shell command" (you can think of a "shell" as the Linux & macOS command-line terminal, e.g., a Bash Shell)

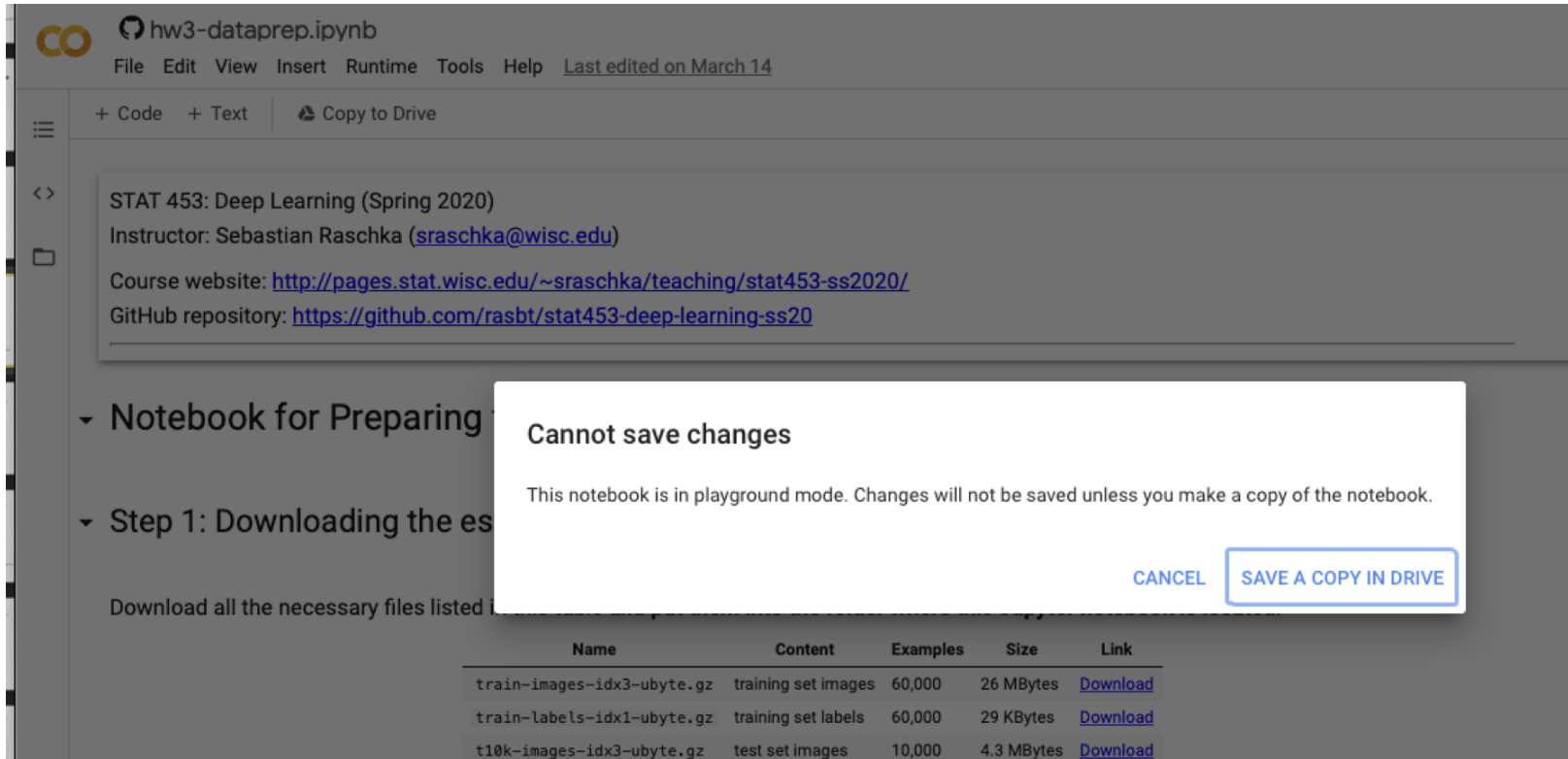


# Google Colab



- You can also upload Notebooks or directly paste GitHub links to notebooks

# Google Colab



The screenshot shows the Google Colab interface for a notebook titled "hw3-dataprep.ipynb". The top menu bar includes File, Edit, View, Insert, Runtime, Tools, and Help, with a note "Last edited on March 14". Below the menu, there are tabs for "+ Code" and "+ Text", and a "Copy to Drive" button. The notebook content area displays text about "STAT 453: Deep Learning (Spring 2020)", the instructor "Sebastian Raschka (sraschka@wisc.edu)", the course website "http://pages.stat.wisc.edu/~sraschka/teaching/stat453-ss2020/", and the GitHub repository "https://github.com/rasbt/stat453-deep-learning-ss20". A dialog box titled "Cannot save changes" is overlaid on the notebook, stating "This notebook is in playground mode. Changes will not be saved unless you make a copy of the notebook." and providing "CANCEL" and "SAVE A COPY IN DRIVE" buttons. Below the dialog, a table lists files for download:

Name	Content	Examples	Size	Link
train-images-idx3-ubyte.gz	training set images	60,000	26 MBytes	<a href="#">Download</a>
train-labels-idx1-ubyte.gz	training set labels	60,000	29 KBytes	<a href="#">Download</a>
t10k-images-idx3-ubyte.gz	test set images	10,000	4.3 MBytes	<a href="#">Download</a>

When you import a Notebook from a GitHub link, make sure to save it in your Google Drive if you plan to make edits, otherwise it will be gone later

# Mounting your Google Drive to the Notebook

1)

```
from google.colab import drive
drive.mount('/content/drive')
```

... Go to this URL in a browser: [https://accounts.google.com/o/oauth2/auth?client\\_id=9473189](https://accounts.google.com/o/oauth2/auth?client_id=9473189)

Enter your authorization code:

Then, click on the link and enter it in the field above

2)

```
from google.colab import drive
drive.mount('/content/drive')
```

... Go to this URL in a browser: [https://accounts.google.com/o/oauth2/auth?client\\_id=9473189](https://accounts.google.com/o/oauth2/auth?client_id=9473189)

Enter your authorization code:

.....

3)

```
from google.colab import drive
drive.mount('/content/drive')
```

Go to this URL in a browser: [https://accounts.google.com/o/oauth2/auth?client\\_id=9473189](https://accounts.google.com/o/oauth2/auth?client_id=9473189)

Enter your authorization code:

.....

Mounted at /content/drive

# Mounting your Google Drive to the Notebook

Once  
mounted,  
you can open  
dataset files

```
[16] # this code cell unzips the .gz files

import sys
import gzip
import shutil
import os

writemode = 'wb'
zipped_mnist = [f for f in os.listdir('/content/drive/My Drive/Colab Notebooks/hw03') if f.endswith('.gz')]
for z in zipped_mnist:
    path = os.path.join('/content/drive/My Drive/Colab Notebooks/hw03', z)
    with gzip.GzipFile(path, mode='rb') as decompressed, open(path[:-3], writemode) as outfile:
        outfile.write(decompressed.read())
```

```
with open(labels_path, 'rb') as lbpath:
    magic, n = struct.unpack('>II',
                             lbpath.read(8))
    labels = np.fromfile(lbpath,
                         dtype=np.uint8)

with open(images_path, 'rb') as imgpath:
    magic, num, rows, cols = struct.unpack(">IIII",
                                           imgpath.read(16))
    images = np.fromfile(imgpath,
                         dtype=np.uint8).reshape(len(labels), 784)

return images, labels

X_train, y_train = load_mnist('/content/drive/My Drive/Colab Notebooks/hw03', kind='train')
print('Rows: %d, columns: %d' % (X_train.shape[0], X_train.shape[1]))

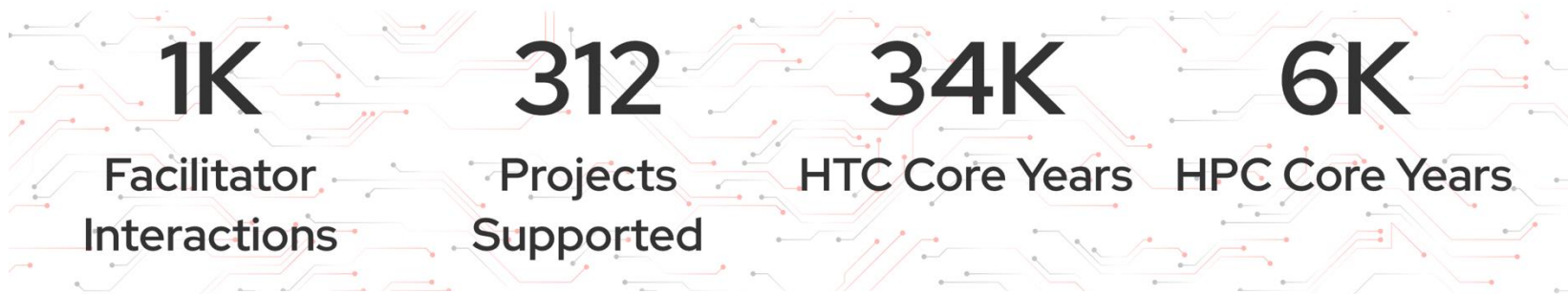
X_test, y_test = load_mnist('/content/drive/My Drive/Colab Notebooks/hw03', kind='t10k')
print('Rows: %d, columns: %d' % (X_test.shape[0], X_test.shape[1]))
```

```
↳ Rows: 60000, columns: 784
   Rows: 10000, columns: 784
```



# Center for High-Throughput Computing (CHTC)

Last Year Serving UW-Madison



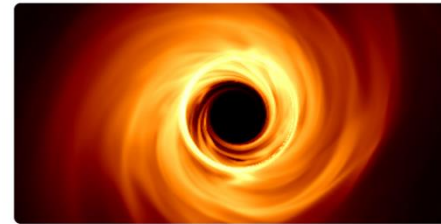
College	Projects Supported	HTC Core Years	HPC Core Years	Facilitator Interactions
Agricultural and Life Sciences	62	8,038	54	159
Education	5	216	-	10
Engineering	66	3,342	4,221	316
Law	1	181	-	1
Letters and Sciences	112	11,645	1,244	440
Medicine and Public Health	42	7,172	54	154
Off-Campus Collaborations	14	2,852	138	-
Pharmacy	5	75	258	7
Veterinary Medicine	5	21	-	-
Total	312	33,542	5,969	1,087

# Center for High-Throughput Computing (CHTC)

[chtc.cs.wisc.edu](https://chtc.cs.wisc.edu)



**HTCondor Celebrates 40 Years  
Powered by Community**  
July 25, 2025



**HTC Strikes Again**  
June 09, 2025

The Center for High Throughput Computing (CHTC), established in 2006, aims to bring the power of High Throughput Computing to all fields of research, and to allow the future of HTC to be shaped by insight from all fields.

Are you a UW-Madison researcher looking to expand your computing beyond your local resources? Request an account now to take advantage of the open computing services offered by the CHTC!

[Request Account](#)

## Upcoming Events



Questions?

