CS5242 : Neural Networks and Deep Learning

Lecture 2: Linear Algebra and PyTorch

Semester 1 2021/22

Xavier Bresson

https://twitter.com/xbresson

Department of Computer Science

National University of Singapore (NUS)

Outline

- Review of linear algebra :
  - Vectors
  - Matrices
  - Matrix-vector multiplication
  - Matrix-matrix multiplication
  - Inner and outer products
  - Tensors
- PyTorch

Outline

- **Review of linear algebra :**
  - **Vectors**
  - Matrices
  - Matrix-vector multiplication
  - Matrix-matrix multiplication
  - Inner and outer products
  - Tensors
- PyTorch

Vectors

Vector Addition:

$$\begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} + \begin{bmatrix} 5 \\ -1 \\ 2 \end{bmatrix} = \begin{bmatrix} 6 \\ 0 \\ 4 \end{bmatrix}$$

Scalar Multiplication:

$$2.\begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 4 \end{bmatrix}$$

Inner Product

- **a.k.a. scalar product :**

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = x_1 y_1 + x_2 y_2 + x_3 y_3$$

Example : $$\begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = 1 + 2 + 6 = 9$$

Inner Product

- **Vector representation :**

$$\vec{\mathbf{x}} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \qquad \vec{\mathbf{y}} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

$$\vec{\mathbf{x}} \cdot \vec{\mathbf{y}} = x_1 y_1 + x_2 y_2 + x_3 y_3 + x_4 y_4$$

Outline

- **Review of linear algebra :**
  - Vectors
  - **Matrices**
  - Matrix-vector multiplication
  - Matrix-matrix multiplication
  - Inner and outer products
  - Tensors
- PyTorch

Matrices

$$W = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$ is a 3 x 2 matrix

Height/rows

Width/columns

$$W = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} & w_{15} \\ w_{21} & w_{22} & w_{23} & w_{24} & w_{25} \\ w_{31} & w_{32} & w_{33} & w_{34} & w_{35} \end{bmatrix}$$ is a 3 x 5 matrix

Transpose of a matrix

$$W = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 5 & 0 & 5 & 0 \\ 2 & 2 & 2 & 2 & 2 \end{bmatrix}$$

is a 3 x 5 matrix

$$W^T = \begin{bmatrix} 1 & 0 & 2 \\ 1 & 5 & 2 \\ 1 & 0 & 2 \\ 1 & 5 & 2 \\ 1 & 0 & 2 \end{bmatrix}$$

is a 5 x 3 matrix

Matrix operations

Matrix Addition:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} + \begin{bmatrix} 10 & 10 \\ 20 & 20 \\ 30 & 30 \end{bmatrix} = \begin{bmatrix} 11 & 12 \\ 23 & 24 \\ 35 & 36 \end{bmatrix}$$

Scalar multiplication:

$$2.\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} = \begin{bmatrix} 2 & 4 \\ 6 & 8 \\ 10 & 12 \end{bmatrix}$$

Outline

- **Review of linear algebra :**
  - Vectors
  - Matrices
  - **Matrix-vector multiplication**
  - Matrix-matrix multiplication
  - Inner and outer products
  - Tensors
- PyTorch

Matrix-Vector Multiplication

$$\begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} & w_{15} \\ w_{21} & w_{22} & w_{23} & w_{24} & w_{25} \\ w_{31} & w_{32} & w_{33} & w_{34} & w_{35} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} (\mathrm{row}_1) \cdot \vec{\mathbf{x}} \\ (\mathrm{row}_2) \cdot \vec{\mathbf{x}} \\ (\mathrm{row}_3) \cdot \vec{\mathbf{x}} \end{bmatrix}$$

3 x 5 matrix

input vector has 5 entries

output vector has 3 entries

Example 1

$$W = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \qquad \vec{\mathbf{x}} = \begin{bmatrix} 1 \\ -2 \\ 2 \end{bmatrix}$$

$$W\vec{\mathbf{x}} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} 1 \\ -2 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 - 4 + 6 \\ 4 - 10 + 12 \end{bmatrix} = \begin{bmatrix} 3 \\ 6 \end{bmatrix}$$

Example 2

$$W = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \qquad \vec{\mathbf{x}} = \begin{bmatrix} 1 \\ -2 \\ 2 \end{bmatrix} \qquad \vec{\mathbf{b}} = \begin{bmatrix} 7 \\ 10 \end{bmatrix}$$

$$W\vec{\mathbf{x}} + \vec{\mathbf{b}} \quad = \quad \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} 1 \\ -2 \\ 2 \end{bmatrix} + \begin{bmatrix} 7 \\ 10 \end{bmatrix} \quad = \quad \begin{bmatrix} 3 \\ 6 \end{bmatrix} + \begin{bmatrix} 7 \\ 10 \end{bmatrix} \quad = \quad \begin{bmatrix} 10 \\ 16 \end{bmatrix}$$

Outline

- **Review of linear algebra :**
  - Vectors
  - Matrices
  - Matrix-vector multiplication
  - **Matrix-matrix multiplication**
  - Inner and outer products
  - Tensors
- PyTorch

Example 1

$$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 3 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 & 1 \\ 2 & 0 & 1 & 1 \\ 3 & 2 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 6 & 4 & 3 & 3 \\ 8 & 4 & 5 & 5 \end{bmatrix}$$

(2 x 3) (3 x 4) = (2 x 4)

(Row 2) . (col 3) = (entry 2,3)

Matrix-Matrix multiplication = batch of Matrix-Vector multiplication

$$\begin{bmatrix} 1 & 1 \\ 2 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 5 \\ 1 \\ 3 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 \\ 2 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 4 \end{bmatrix} = \begin{bmatrix} 6 \\ 0 \\ 4 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 \\ 2 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 5 & 6 \\ 1 & 0 \\ 3 & 4 \end{bmatrix}$$

Outline

- Review of linear algebra :
  - Vectors
  - Matrices
  - Matrix-vector multiplication
  - Matrix-matrix multiplication
  - Inner and outer products
  - Tensors
- PyTorch

Inner and outer products

$$\vec{\mathbf{x}} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \quad \vec{\mathbf{y}} = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}$$

$$\vec{\mathbf{x}} \cdot \vec{\mathbf{y}} = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 8 \end{bmatrix}$$

Inner product

(1 x 3) (3 x 1) = (1 x 1)

$$\vec{\mathbf{x}} \otimes \vec{\mathbf{y}} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \begin{bmatrix} 0 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 2 \\ 0 & 2 & 4 \\ 0 & 3 & 6 \end{bmatrix}$$

Outer product

(3 x 1) (1 x 3) = (3 x 3)

Matrix-Matrix multiplication $=$ sum of outer products

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 2 & 2 & 0 & 1 \\ 1 & 2 & 3 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 1 & 1 \\ 2 & 0 \\ 3 & 1 \end{bmatrix} = \begin{bmatrix} 6 & 3 \\ 5 & 3 \\ 0 & 2 \end{bmatrix}$$

$$\overbrace{\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 0 \end{bmatrix}}^{\text{This outer product}} + \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \begin{bmatrix} 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 2 & 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 3 & 1 \end{bmatrix} = \begin{bmatrix} 6 & 3 \\ 5 & 3 \\ 0 & 2 \end{bmatrix}$$

Outline

- Review of linear algebra :
  - Vectors
  - Matrices
  - Matrix-vector multiplication
  - Matrix-matrix multiplication
  - Inner and outer products
  - Tensors
- PyTorch

Tensors

- **Generalization of matrices to higher dimensions:**

X

No size

A scalar entry of X

=

0-D tensor

X

5 x 6 x 9

3-D tensor

X[i,j,k]

X

5 x 6

Matrix

=

2-D tensor

X[i,j]

X

Size is 5

A row vector of X

=

1-D tensor

X[i]

X

5 x 6 x 9 x 5 x 9 x 3

6-D tensor

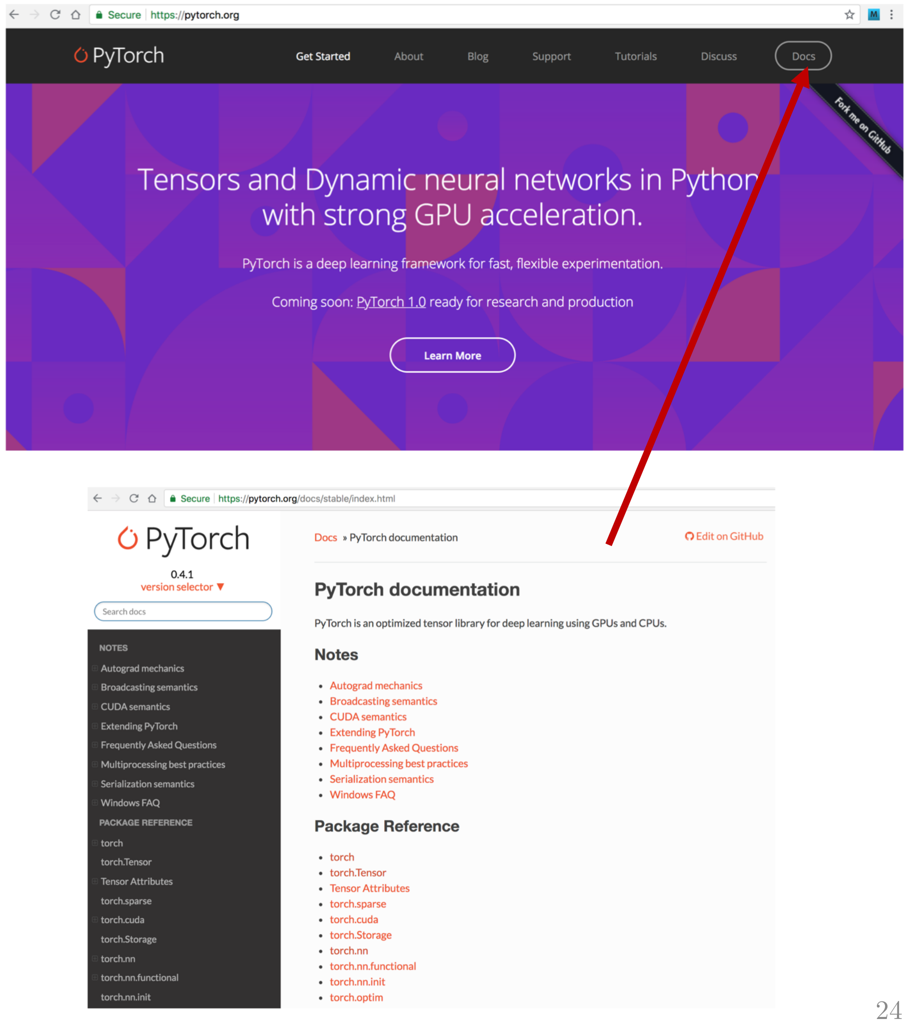
X[i,j,k,l,m,n]

Outline

- Review of linear algebra :
  - Vectors
  - Matrices
  - Matrix-vector multiplication
  - Matrix-matrix multiplication
  - Inner and outer products
  - Tensors
- **PyTorch**

PyTorch

- PyTorch is an open source deep learning library in Python.
- Developed by FAIR (Facebook Artificial Intelligence Research).
- Original authors: Soumith Chintala, Adam Paszke, Sam Gross, Gregory Chanan.
- Initial release: October 2016
- PyTorch provides two high-level features:
    - Tensor computation (like numpy) with strong GPU acceleration
    - Deep Neural Networks built on an autodiff system

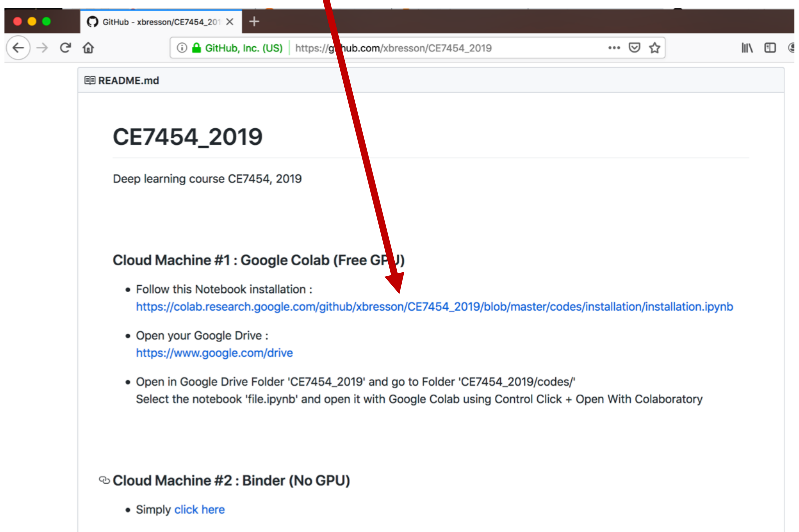


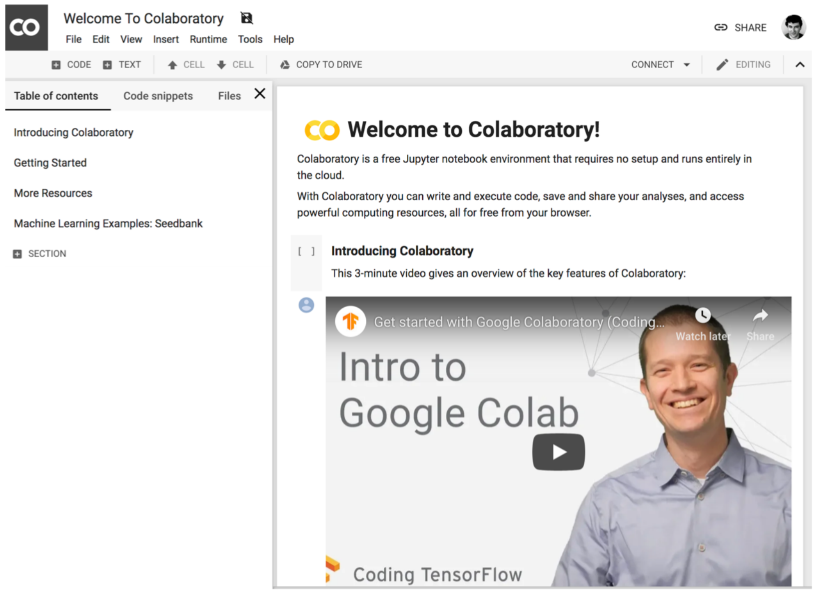
Cloud machine #1 (Free GPU)

- Follow these instructions :
  - Go to the GitHub folder of the course :

    https://github.com/xbresson/CS5242_2021

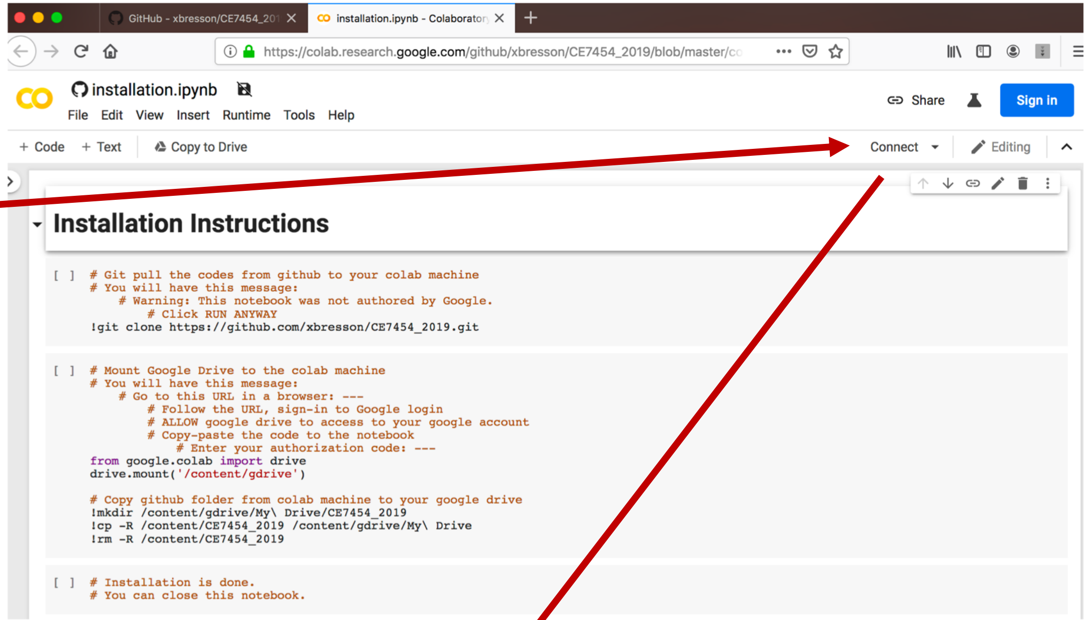
    Click on this link.

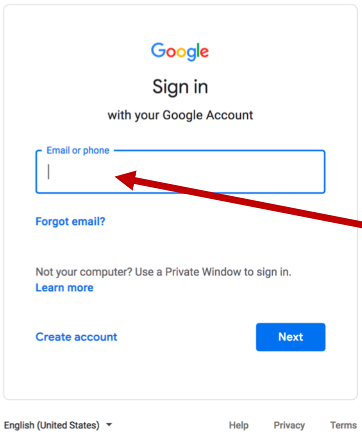


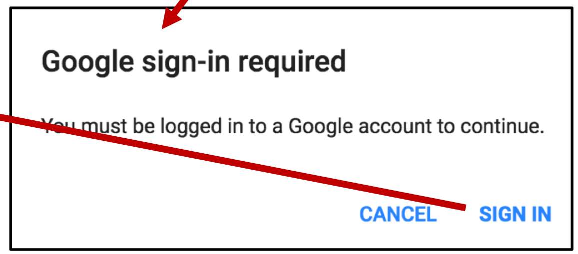


Cloud machine #1 (Free GPU)

- Click on CONNECT.
- It will ask you to sign-in with your Gmail account.



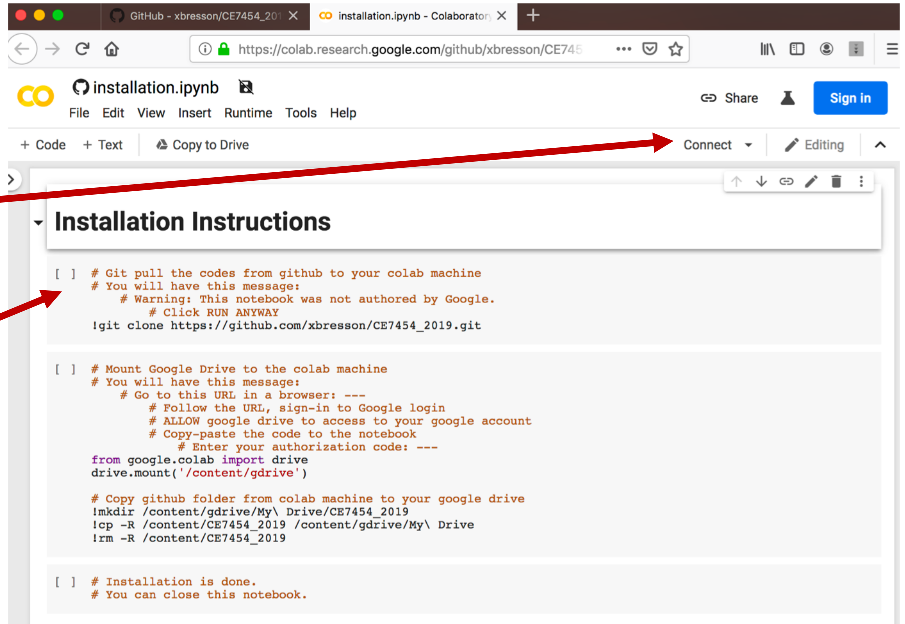


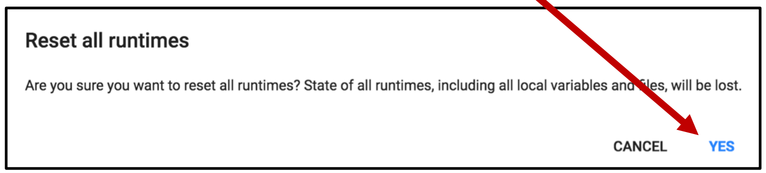


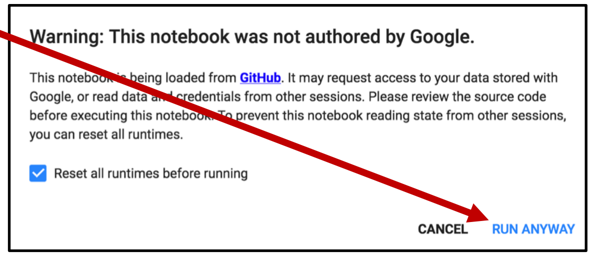
Cloud machine #1 (Free GPU)

- Click on CONNECT again to start the Google Cloud machine.

- Run the first cell to clone the codes from GitHub to the Google Cloud machine.
  - It will give a warning, click on RUN ANYWAY.
  - Answer YES to the next question RESET ALL RUNTIMES.

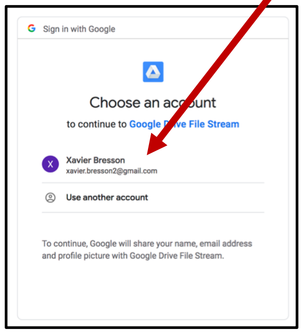


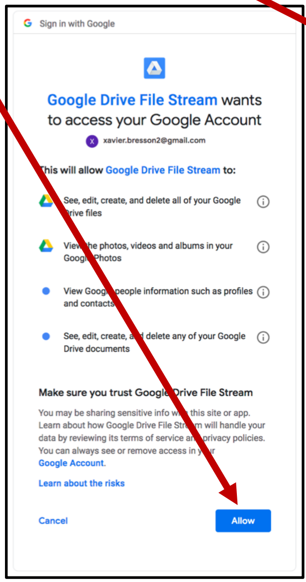


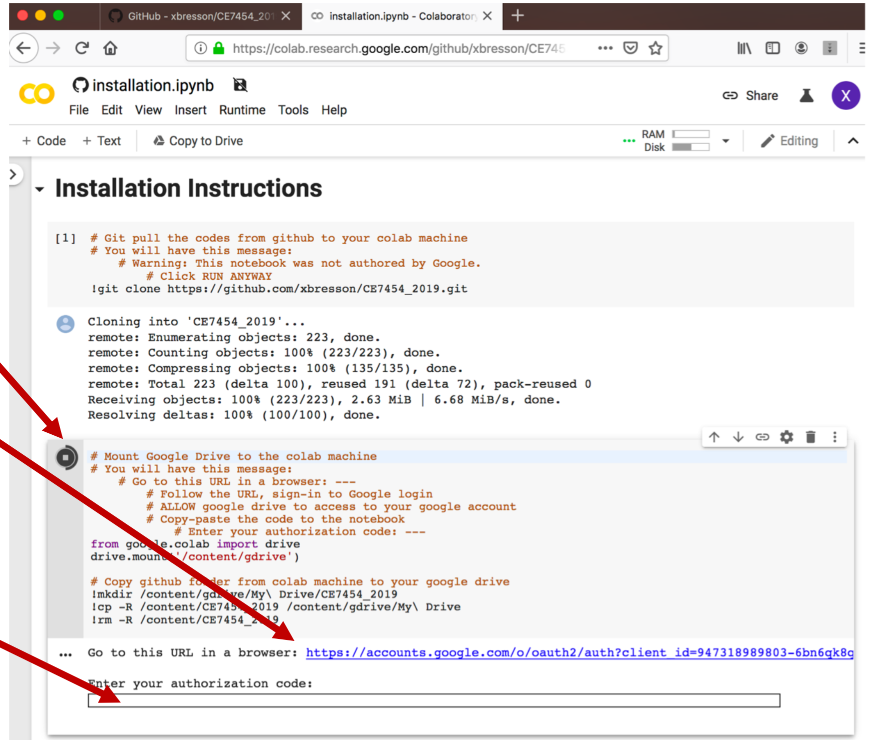


Cloud machine #1 (Free GPU)

- Run the second cell to mount your Google Drive to the Google Cloud machine (all your codes will be saved in Google Drive).
    - Click on the provided URL.
    - Select your Gmail account.
    - ALLOW Google Drive File Stream.
    - Copy-paste the code to the notebook (Enter your authorization code) and press Return.



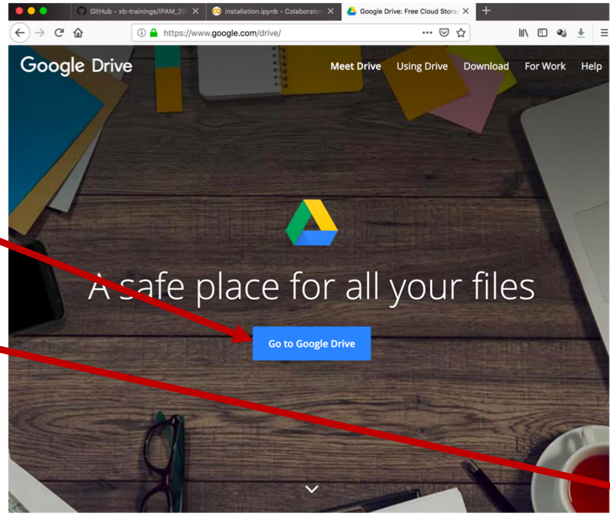


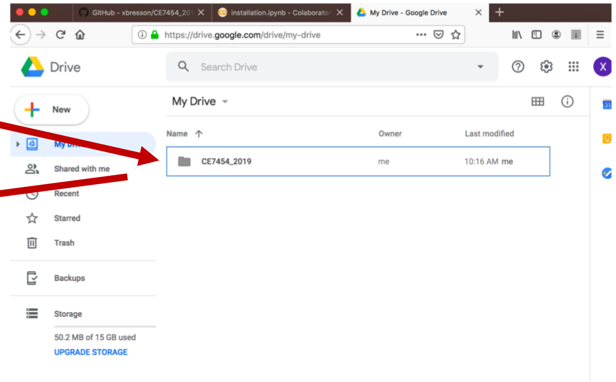


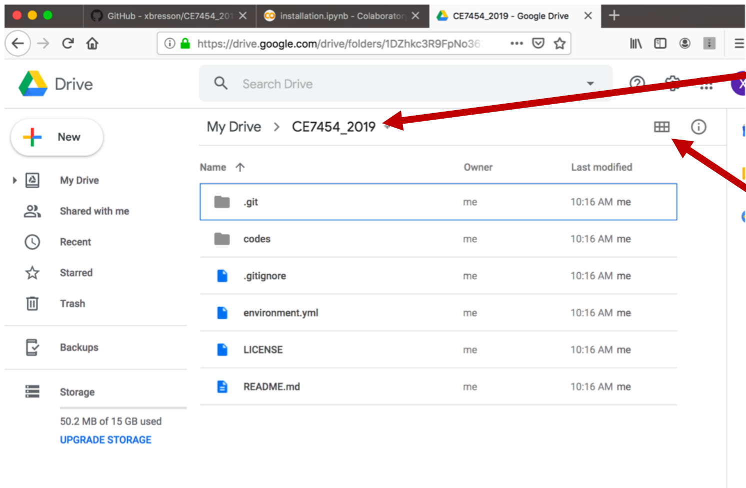
Cloud machine #1 (Free GPU)

- Open your Google Drive :
  https://www.google.com/drive

- Go folder CS5242_2021_codes





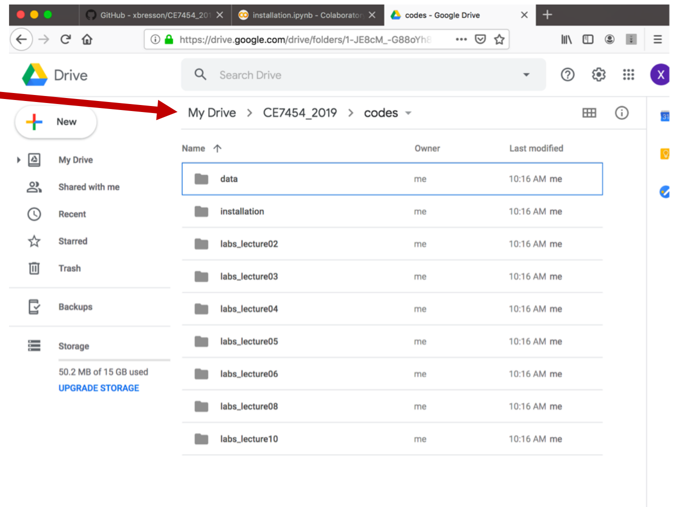


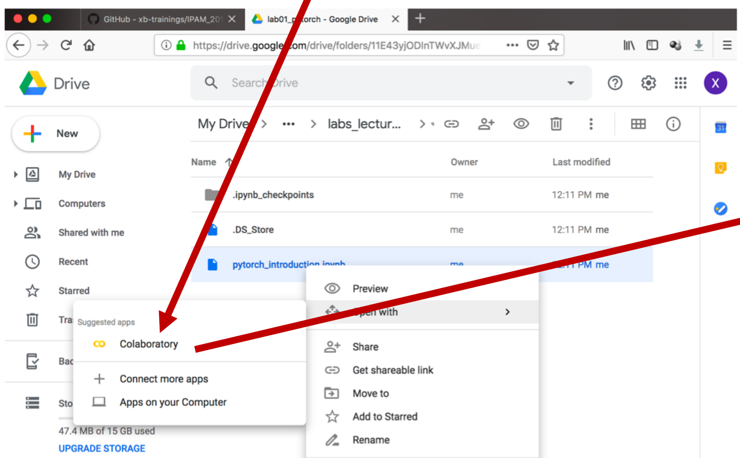
Click here for List View

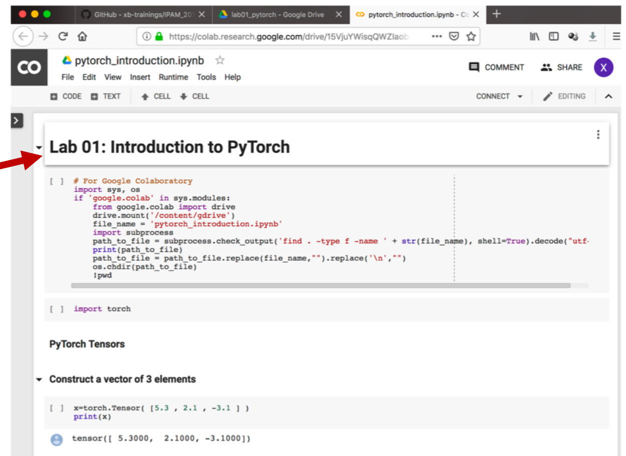
Cloud machine #1 (Free GPU)

- Go folder CS5242_2021_codes/codes

- Let us open the notebook pytorch_introduction.pynb in folder codes/labs_lecture02.
  - Select the notebook and open it using Control Click + Open With Colaboratory

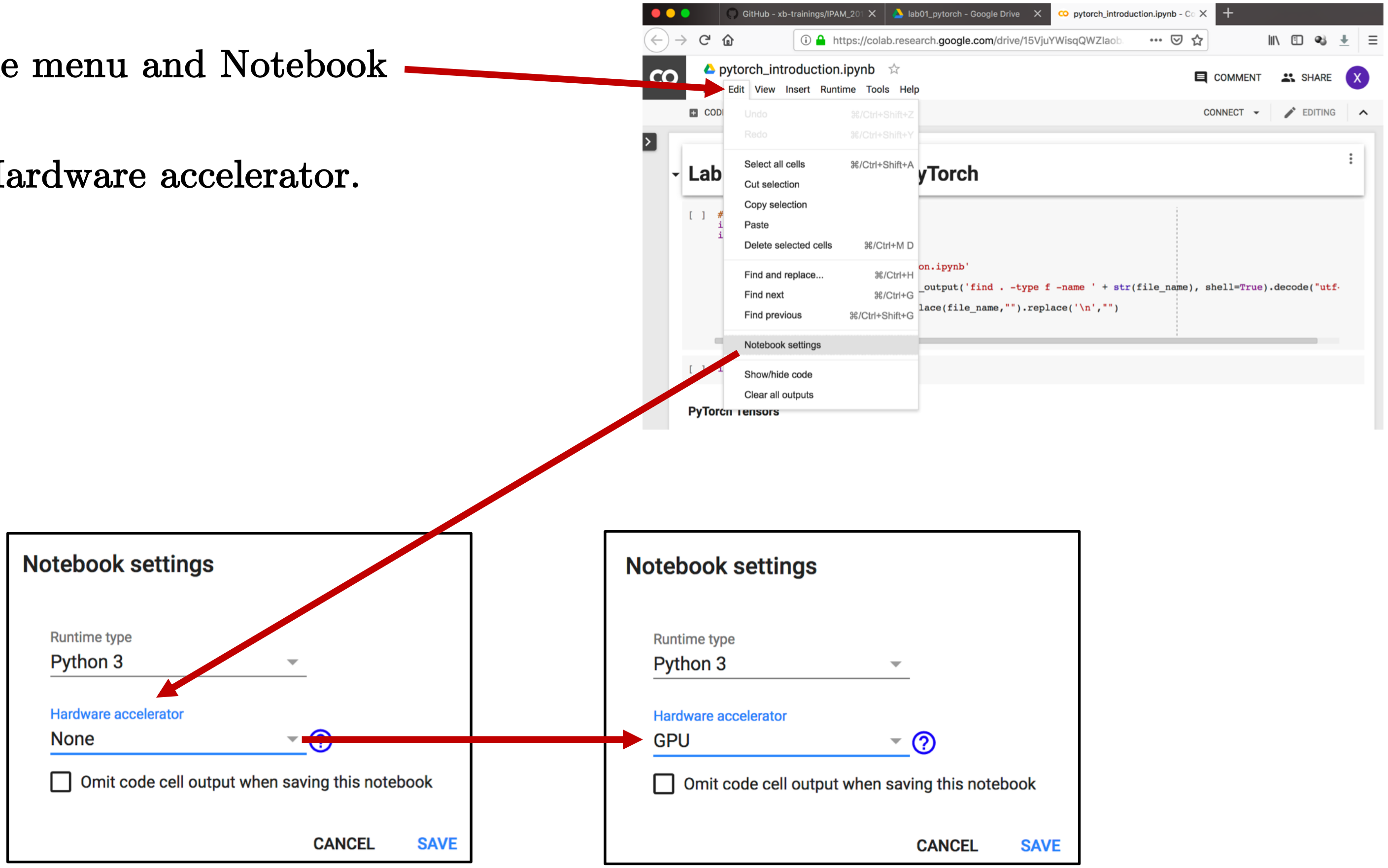






Cloud machine #1 (Free GPU)

- GPU accelerator :
  - Select Edit in the menu and Notebook settings.
  - Select GPU in Hardware accelerator.



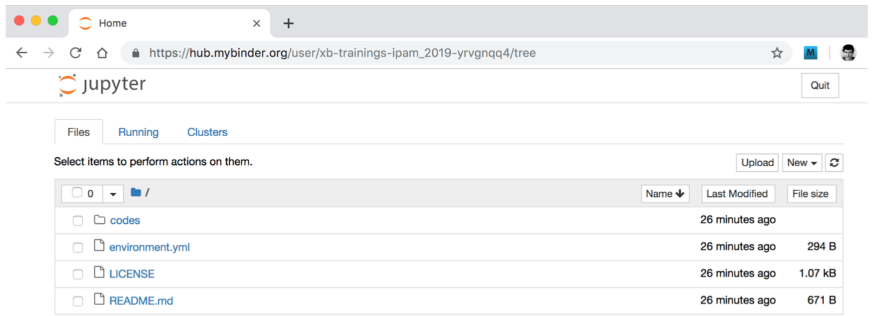
Cloud machine #2 (No GPU)

- Start your cloud machine :
  - Type this address in your favorite web browser :

    https://mybinder.org/v2/gh/xbresson/CS5242_2021/master

  - Binder creates mini-instances on Amazon Cloud.
  - Binder provides at least 12 hours of session time per user session.
  - Binder will automatically shut down user sessions that have more than 10 minutes of inactivity (if you leave your window open, this will be counted as "activity").

binder
(beta)



Your local machine

- Install Python and run the notebooks on your machine (it will use the GPU if you have it) :
  - Follow these instructions (easy steps) :

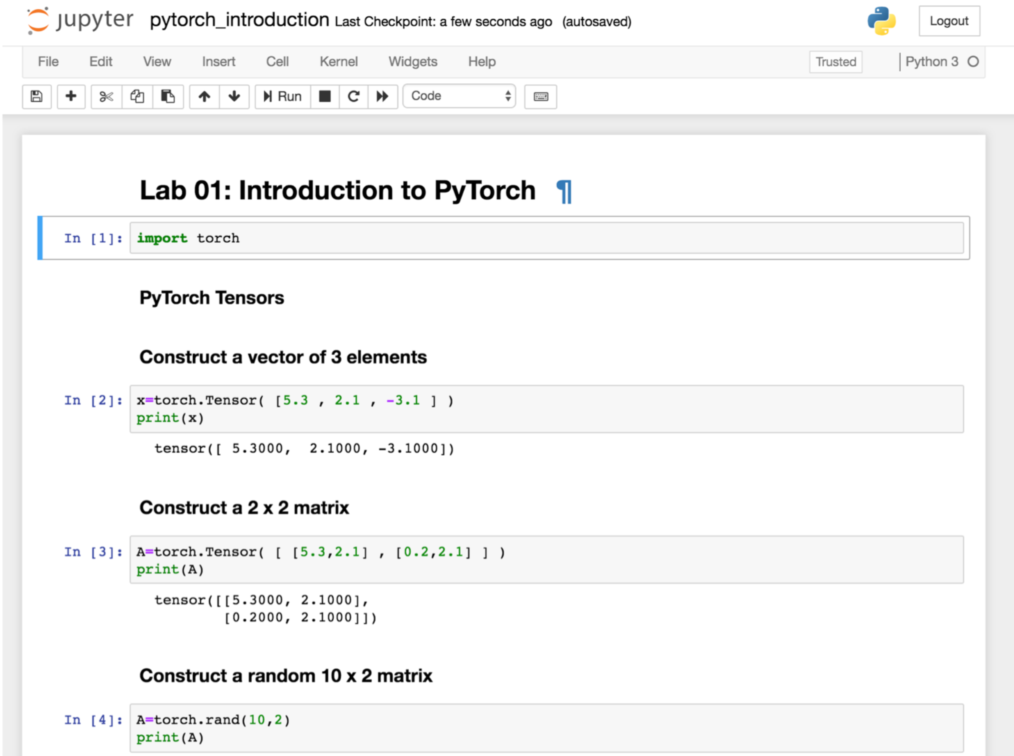
Running Python notebooks on your computer

Installation

```
# Conda installation for Linux and OSX
# Open a Terminal
curl -o ~/miniconda.sh -O https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86_64.sh # Linux
curl -o ~/miniconda.sh -O https://repo.continuum.io/miniconda/Miniconda3-latest-MacOSX-x86_64.sh # OSX
chmod +x ~/miniconda.sh
./miniconda.sh
source ~/.bashrc

# Conda installation for Windows
# Download and install : https://repo.anaconda.com/miniconda/Miniconda3-latest-Windows-x86_64.exe
# Open new Application "Anaconda Prompt Terminal"

# Clone GitHub repo
conda install git
git clone https://github.com/xbresson/CS5242_2021.git
cd CS5242_2021

# Install python libraries
conda env create -f environment.yml
source activate deeplearn_course

# Run the notebooks
jupyter notebook
```

Lab 01

- Introduction to PyTorch



Lab 01: Introduction to PyTorch ¶

In [1]:
```
import torch
```

PyTorch Tensors

Construct a vector of 3 elements

In [2]:
```
x=torch.Tensor( [5.3 , 2.1 , -3.1 ] )
print(x)
```

```
tensor([ 5.3000,  2.1000, -3.1000])
```

Construct a 2 x 2 matrix

In [3]:
```
A=torch.Tensor( [ [5.3,2.1] , [0.2,2.1] ] )
print(A)
```

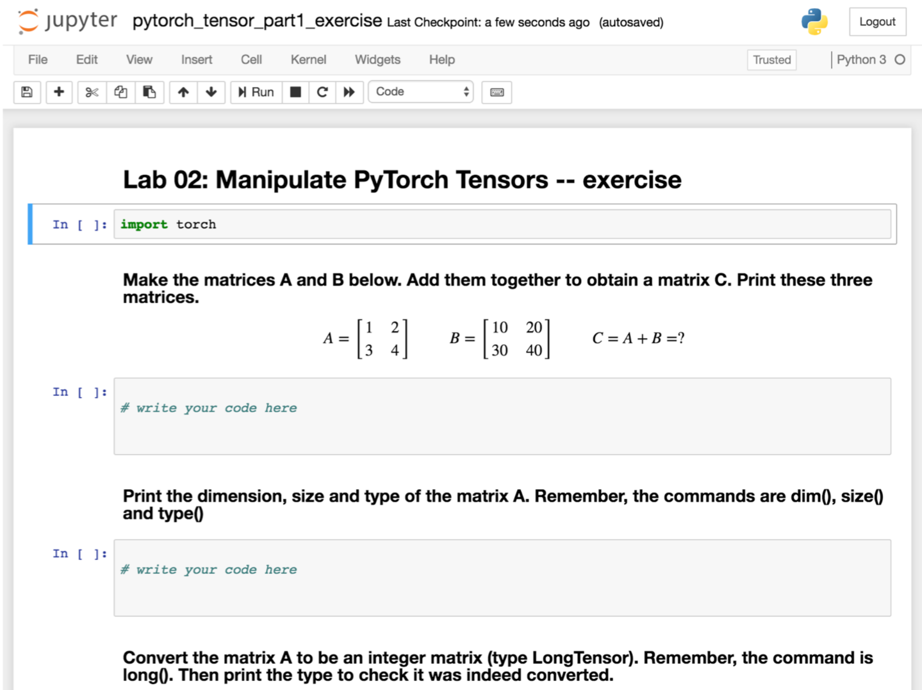
```
tensor([[5.3000, 2.1000],
        [0.2000, 2.1000]])
```

Construct a random 10 x 2 matrix

In [4]:
```
A=torch.rand(10,2)
print(A)
```

Lab 02

- Manipulating PyTorch tensors – basic linear algebra



Lab 03

- Manipulating PyTorch tensors – reshaping and slicing

jupyter pytorch_tensor_part2_exercise Last Checkpoint: a few seconds ago (autosaved)

Lab 03: Resizing and slicing in PyTorch -- exercise

In [ ]:
```
import torch
import utils
```

Make a 10 x 2 matrix random matrix A. Then store its third row (index = 2) in to a vector v. Then store the first 5 rows (index 0 to index 4) into a submatrix B. The important information is that B has a total of five rows. Print A, v and B.

In [ ]:
```
# write your code here
```

Extract entry (0,0) of the matrix A and store it into a PYTHON NUMBER x

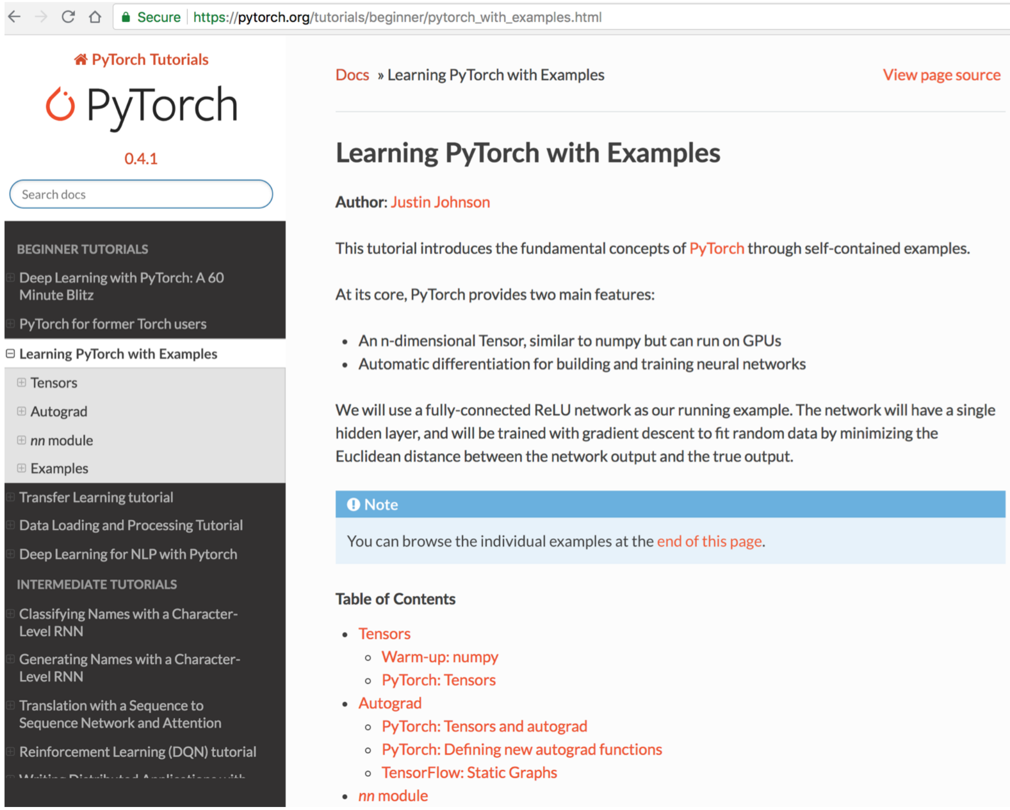
In [ ]:
```
# write your code here
```

Let's download 60,000 gray scale pictures as well as their label. Each picture is 28 by 28 pixels.

In [ ]:
```
from utils import check_mnist_dataset_exists
data_path=check_mnist_dataset_exists()

data=torch.load(data_path+'mnist/train_data.pt')
label=torch.load(data_path+'mnist/train_label.pt')
```

More tutorials

- **Learning PyTorch with Examples**



https://pytorch.org/tutorials/beginner/pytorch_with_examples.html

Questions?