Autumn 2023 Project Intro

go.osu.edu/aiclubsignup

Tools we will be working with

- Python 3.x
- PyTorch (Python library for Deep Learning)
- Jupyter Notebooks / Google CoLab
- If you want to run locally, you'll need a fairly powerful GPU (8GB VRAM or more), otherwise training and evaluation can be painfully slow. If you don't have one, Google Colab is just fine if not better.

Setting up your environment

Google Colab

Repository: https://github.com/inesh-l/project-series-2023-public-resources

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

Local Setup (Recommended for later in the semester)

1. Install a code editor of choice (I.E. Visual Studio Code, Jetbrains PyCharm)
2. Install Python for your operating system
3. Ensure that python is installed by entering “pip” or “python” in your command prompt/terminal.
4. All libraries will be installed via pip (e.g. “pip install matplotlib”)

Local - Windows

Git Website: https://git-scm.com/downloads

Python from the Microsoft Store (RECOMMENDED):
https://www.microsoft.com/store/productid/9NRWMJP3717K?ocid=pdpshare

- Alternatively, Python Website: https://www.python.org/

PyTorch Starting Locally: https://pytorch.org/get-started/locally/

nVidia CUDA (OPTIONAL):
https://developer.nvidia.com/cuda-11-8-0-download-archive

Local - macOS

Git Website: https://git-scm.com/downloads

Python Website: https://www.python.org/

PyTorch Starting Locally: https://pytorch.org/get-started/locally/

Local - Linux (INCLUDING WSL)

Install Git and Python from the package manager corresponding to your distribution.

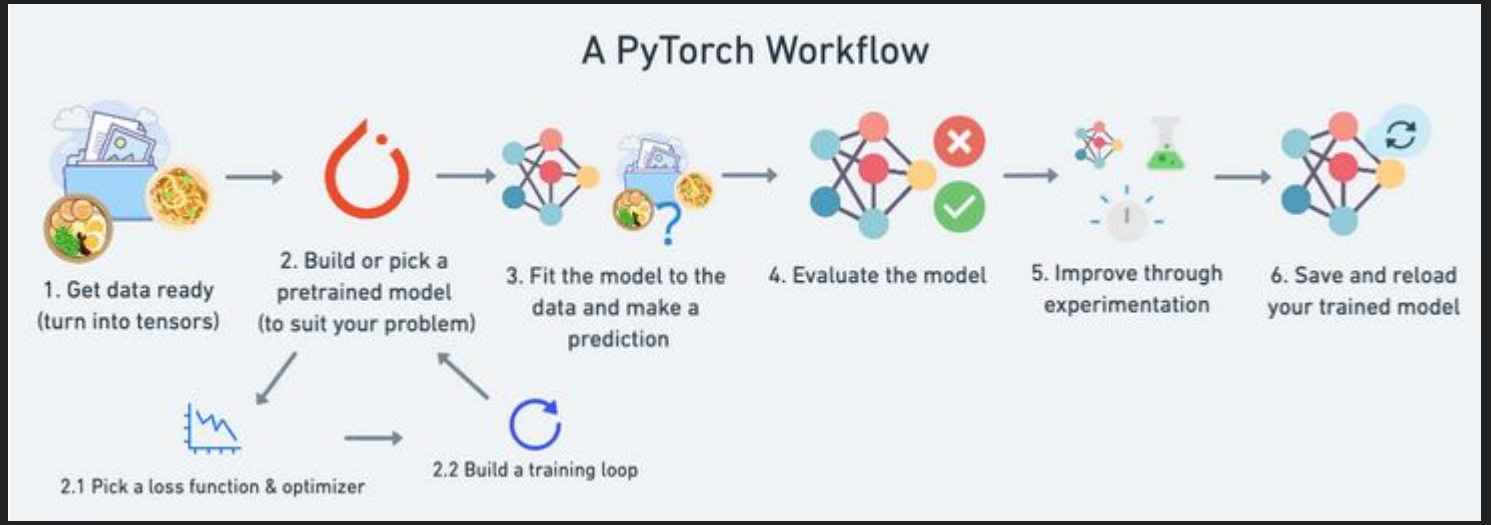
PyTorch Starting Locally: https://pytorch.org/get-started/locally/

nVidia CUDA (OPTIONAL):
https://developer.nvidia.com/cuda-11-8-0-download-archive

- If you're using WSL, go to Linux > x86_64 > WSL-Ubuntu.

Data in PyTorch

We are here


A PyTorch Workflow
1. Get data ready (turn into tensors)
2. Build or pick a pretrained model (to suit your problem)
3. Fit the model to the data and make a prediction
4. Evaluate the model
5. Improve through experimentation
6. Save and reload your trained model
2.1 Pick a loss function & optimizer
2.2 Build a training loop

Essentials

- All data in PyTorch is represented using “tensors”, or multi-dimensional arrays. This means that any image/text/audio data must be transformed into a tensor before being fed into a model.
- Datasets are represented in PyTorch using Python classes that extend the built in Dataset class.
- What a PyTorch dataset does:
  - Transforms local data into PyTorch tensors
  - Prepares data for use by other PyTorch functions

Create a PyTorch Dataset

Python has a built in support for many datasets

- Computer Vision: https://pytorch.org/vision/main/datasets.html
- Text Processing: https://pytorch.org/text/stable/datasets.html
- Some of these datasets download automatically when initialized in Python.

However, PyTorch also has documentation for creating a custom dataset class

https://pytorch.org/tutorials/beginner/basics/data_tutorial.html

Scroll down to “Creating a Custom Dataset for your files”

Make sure these packages are installed

```
#Used for importing our datasets
import torch
from torch import nn
from torchvision import datasets
from torchvision.transforms import ToTensor
from torch.utils.data import DataLoader

#Useful for visualizing data
import matplotlib.pyplot as plt
```

Next Meeting

To Do: Have a dataset ready for model building.

In the next meeting we will:

- Continue getting data ready for training and testing
- Explain what deep learning models are.
- Create our own deep learning model.

Feel free to ask any questions!

First Time Sign up

AI Club Website