Prerequisites and Pytorch Installation Guide

May 8, 2021

1 Prerequisites and PyTorch Installation

A walkthrough to install PyTorch for Python 3.6. If you already have PyTorch on your machine and feel confident in your installation, you can go ahead and ignore this notebook.

For the purposes of this project, it is recommended to use Anaconda as your Python distribution and for managing your environments. If you'd rather use some other distribution or version of Python and feel confident in your abilities, you may also choose to just follow the instructions here: https://pytorch.org/get-started/locally/.

If you do not want to run computations on your Local Machine, you could always go ahead and work on Google Colab (https://colab.research.google.com/) It is an environment similar to Jupyter Notebooks and would run code on Google's dedicated GPU.

1.0.1 Install Anaconda

Download and install Anaconda. A more detailed version of the below steps specifically for Windows (macOS and Linux instructions are similar) can be found here if you need additional assistance.

1. Go to the Anaconda install page and click on the Python 3.7 installer matching your operating system (e.g. Windows, macOS, Linux) and system architecture (likely 64-bit X86).
2. Open the installer from your downloads folder and run it by double clicking on it.
3. Follow the installer's instructions.

Note: there is an option at one point to add Anaconda to PATH. While the installer recommends against this option (it's possible it may interfere with your other software), we typically recommend doing so, as it makes accessing Anaconda commands easier. The rest of this project assumes that this option has been selected.

1.0.2 Set-up a conda environment

1. Open a Command Prompt (Windows) or a terminal (macOS or Linux)

   Note: If you chose not to add Anaconda to PATH during Anaconda installation above, you will need to use the Anaconda Command Prompt instead.

2. Create a conda environment for PyTorch: `Shell conda create -n pytorch python=3.6`

3. Activate your new environment:

   ```
   conda activate pytorch
   ```

4. Install some supporting dependencies `Shell` `conda install h5py imageio jupyter matplotlib numpy tqdm`

5. Install either the CPU version or GPU version of PyTorch:

   CPU: *(Recommended as sufficient for this class)* "'Shell # Windows/Linux conda install pytorch torchvision cpuonly -c pytorch

   # macOS conda install pytorch torchvision -c pytorch "'

   GPU: (You'll also need to install some Nvidia software) "'Shell # Windows/Linx, assuming CUDA 10.1 conda install pytorch torchvision cudatoolkit=10.1 -c pytorch

   # macOS # Trickier. May require installing from source. See PyTorch docs. "'

   Note: If you have GPUs available, you should use the GPU version for any serious research or application, as it can be significantly faster. For the purposes of this demo though, the CPU version is sufficent.

1.0.3 Verify PyTorch Installation

1. While in your `pytorch` environment, enter the Python shell with:

   ```
   python
   ```

   Verify that your python version says Python 3.6.[x].

2. Enter this program into your Python shell:

   ```
   import torch
   print(torch.__version__)
   x = torch.rand(2, 3)
   print(x)
   ```

   Your command line should return something like:

   ```
   1.3.1
   tensor([[0.3651, 0.7264, 0.9823],
           [0.5398, 0.2145, 0.3616]])
   ```

   The version returned will depend on what the current stable version of PyTorch is when you perform your install. As long as it's 1.[x].[y], these notebooks should work fine. The numbers in the tensor `x` will be random, so as long as the print returns a 2x3 `tensor` of random values, you're in good shape.

[1]:

```
import torch
print(torch.__version__)
x = torch.rand(2,3)
print(x)
```

```
1.4.0
tensor([[0.6034, 0.7559, 0.2226],
        [0.0051, 0.2992, 0.9300]])
```

```
[ ]:
```