Environment Setup

DS-GA 1011: Lab 1

Objectives

By the end of this lab, you should have the following installed:

- Anaconda / Miniconda
- Python 3.6+
- Pytorch 0.4
- Jupyter Notebooks / JupyterLab
- Sublime / PyCharm / other editor

Optional:

- Git
- cuDNN / CUDA (for machines with GPUs)

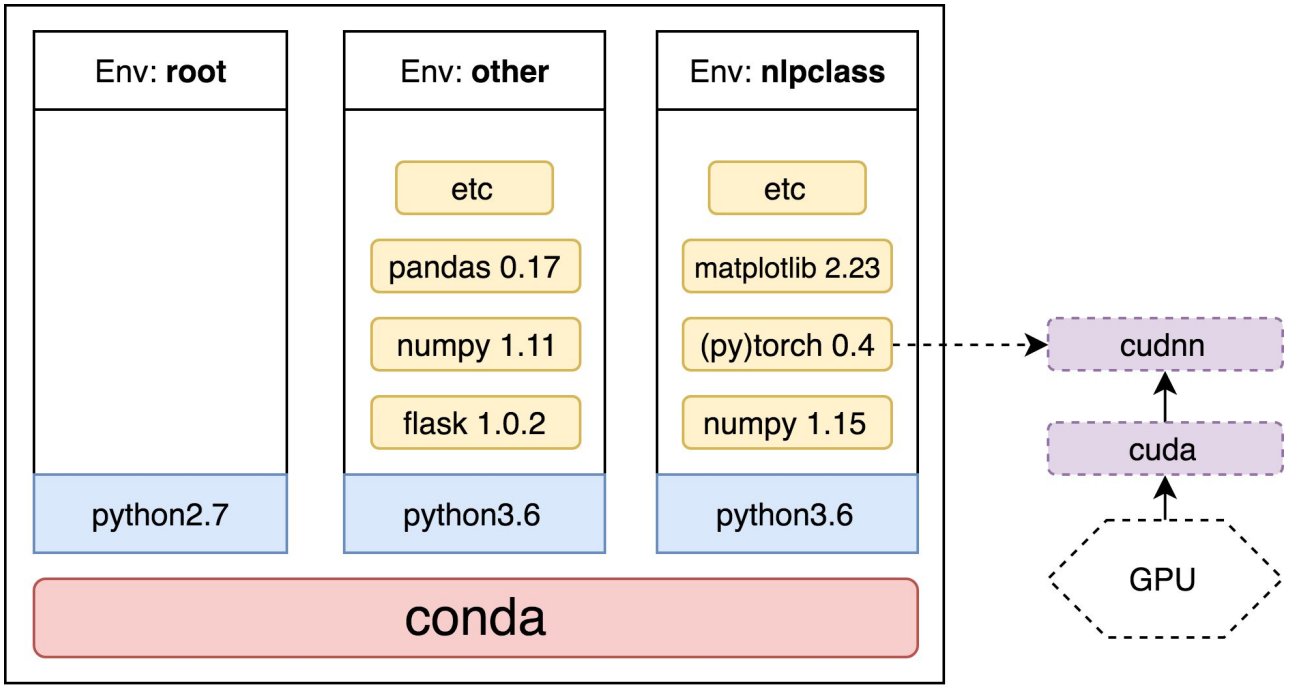
Heads-up

- Linux / MacOS users: You'll be fine, try to make your setup as clean and minimal as possible.

- Windows users: You'll likely run into issues. Ask for help on Piazza - other Windows users will likely have similar problems.

If you need help, let us know! Try not to leave this lab without a fully working setup.

Setup Overview



Conda Installation

For people starting fresh (never installed anaconda/miniconda, or "what is conda").

- conda is a package and environment manager for Python, which allows you to easily manage multiple environments and has prepackaged libraries/dependencies.
- **Everyone**:
  - Download corresponding installer from: https://conda.io/miniconda.html (Python 3.6)
- **MacOS / Linux users**:
  - Run `bash [installer_file_that_ends_with.sh]`
  - Go with default arguments, let the setup script modify your `.bashrc`
  - MacOS users may need to first install XCode.
- **Windows users**:
  - Run the installer executable
  - I recommend choosing to modify the PATH variable (easier system-wide usage of Conda), but you can also go with the default.
- run `conda list` after to confirm that installation succeeded

For more details: https://conda.io/docs/user-guide/install/index.html

Conda Environment Setup

- Open the respective terminal for your OS:
  - Terminal (Linux/MacOS), Powershell (Windows) or Anaconda prompt (Windows without modifying path)
- Run `conda create -n nlpclass python=3.6`
  - This creates a standalone conda environment for this class
  - *Try to never install anything in the root environment*
- Run `conda activate nlpclass`
  - This activates the environment for this class
- Run `conda install jupyter notebook matplotlib scikit-learn`
- Go to https://pytorch.org/, select the configuration corresponding to your machine, and run the command
  - e.g. `conda install pytorch torchvision -c pytorch`
  - This installs PyTorch from the PyTorch conda channel

Useful Conda commands

- `conda install [package-name]`
  - Install package(s) into current environment
- `conda list`
  - Shows packages installed in current environment
- `conda info --envs`
  - Shows conda environments on your system

Jupyter

- Jupyter provides a web interface for interacting with Python kernels
  - Great for plotting results, writing notes, code demonstrations, quick scripting
  - Not great for software engineering!
- In the folder cloned above, with the `nlpclass` environment activated, run `jupyter notebook`

- **Demonstration**

- Optional: Run Jupyter persistently in a tmux session! (MacOs/Linux only)
  - Run `tmux new -s nlpclass` (opens a side session)
  - Activate environment, open notebook, etc
  - Press "Ctrl+B" and then "D" (your notebook session is still running, you can even close the console)
  - Run `tmux attach -t nlpclass` (reopens the `nlpclass` tmux session)

JupyterLab

- JupyterLab is a next-gen version Jupyter that makes it more akin to RStudio
  - Incorporates many more features (shared kernels, output panels, etc)
  - It is still in development (version 0.3), and most people still use stand-alone notebooks

- To install, run: `conda install -c conda-forge jupyterlab`

- To run, run: `jupyter lab`

- **Demonstration**

System / Python Setup

- To import packages in Python, the root folder of the package needs to be on your PYTHONPATH.
- You can see it in `import sys; print(sys.path)`
- The current working directory is always included in the PYTHONPATH.
- There are several common ways to modify your PYTHONPATH.
  1. Modify at the system level
     - e.g. Adding `export PYTHONPATH=/my/new/path:$PYTHONPATH` to `.bashrc`
  2. Modify on the fly
     - Running `export PYTHONPATH=/my/new/path:$PYTHONPATH` before running your code / starting your notebook server.
  3. Modify for just that command
     - Running `PYTHONPATH=/my/new/path:$PYTHONPATH python`
  4. Modify in-session:
     - In a Python session, run:
       - ```
         import sys
         sys.path += ["/my/new/path"]
         ```
         ```

Editors

- **Sublime**: Free, lightweight and excellent text editor

- **PyCharm**: Full-fledged IDE
  - Community edition available for free
  - Professional licence available for free to students
  - Learn to set up *projects*

- **Vim/Emacs**:
  - If you know them, you already know if you want to use them. You also know which one of the two is superior :)

Optional: Git

- Git is a version control system
  - GitHub is a website that's built on Git

- Install Git to your machine following instructions on https://git-scm.com/book/en/v2/Getting-Started-Installing-Git

- (Forthcoming) In your work folder for this class, run: `git clone git@github.com:nyu-dl/DS-GA-1011-Labs-2018.git`
  - Or `git clone https://github.com/nyu-dl/DS-GA-1011-Labs-2018.git`

- Diving into Git internals will take too long for this lab - come to Office Hours!

Optional: CUDA / cuDNN Setup (if you have a GPU)

CUDA

- (First!) Install NVIDIA Drivers: https://www.nvidia.com/Download/index.aspx
- Download the appropriate version and install from https://developer.nvidia.com/cuda-downloads

cuDNN (from *tar* archive)

- Go to https://developer.nvidia.com/cudnn, sign up, and download.
- Unzip the cuDNN archive.
- Copy the required files to your CUDA installation folder (see below)

Finally, install PyTorch with the corresponding CUDA/cuDNN versions.

More details: https://docs.nvidia.com/deeplearning/sdk/cudnn-install/index.html.