Geoff Hulten
Ghulten@cs.washington.edu

1,900 words.

# Lecture Notes CSEP 546 -- 2020

by Geoff Hulten

# Foreword

Welcome.

This is my third opportunity to teach graduate level introduction to machine learning and I'm very grateful for it.

A bit about me. I got my PhD in machine learning from UW in 2005. Since then I've worked as a professional machine learning scientist and machine learning engineer in product groups at two of the big tech companies. I've been part of shipping machine learning into dozens of 'Internet Scale' systems including building models, managing teams of modelers, and architecting how machine learning interfaces with and augments the systems I worked on.

This course is targeted toward engineers and program

managers who want to learn the fundamentals of machine learning, but who also want to learn how to use machine learning in practice. About 50% of the content will address core learning algorithms and machine learning concepts. About 25% of the content will focus on how to use these algorithms & concepts to produce high-quality models in practice. And about 25% of the content will focus on how to architect machine learning systems for success at scale.

When you have completed this course you will understand all of the key concepts of machine learning; you will have implemented basic versions of the most important machine learning algorithms; you will have a basis of hands-on experience with the three learning approaches that have the most influence in modern professional machine learning; you will have some intuition of when to use (and when not to use) machine learning in your applications; and you will be prepared to architect machine learning-based systems that amplify the strengths of machine learning while compensating for its weaknesses.

Every year this course is evolving. This year my goals (compared to last year) are to:

1) Make the assignments smoother -- more focused on the

concept to learn; clearer to do; and easier (and fairer) to grade.

2) Refocus ML Engineering topics (which dragged a bit) around ML architecture, design patterns, & case studies.

3) Address several places where key concepts were weak (e.g. bounds, boosting, reinforcement learning, probabilistic modeling).

I'm also beginning the process of creating lecture notes to augment (or potentially eventually replace) one or both of the text books.

I look forward to taking this journey with all of you. Please ask lots of questions. Please point out where I make mistakes. And please give suggestions on how I could make this course better in the future.

Thank you,

Geoff Hulten

# Contents

# Chapter One: Getting Started

## Coding Environment

For assignments we will use Python with support from a few open source libraries & systems.

The recommended environment is vscode. You can use other environments, but you'll have to figure out how to install and debug problems on your own.

Note that versions & software change over time. I'll keep an updated list of tools and links online at: HERE HERE HERE.

### VSCode

VSCode is a free cross-platform code editor that works well with Python.

You can download it from: http://code.visualstudio.com/download

And you can find introductory instructions on how to use it at: http://code.visualstudio.com/docs/introvideos/basics

## Git

Git is a widely used source control mechanism.

The framework code for the assignments in this course is hosted in a git repository at: http://github.com/ghulten/MachineLearningCourse

The only thing you'll need to do with git is download the support code. Maybe the easiest way to do that is to visit the course repository and under the 'clone' button, select 'Download ZIP'.

Or you can install git, and clone the repository all official like -- the way the cool kids would do it.

Install git services from: http://git-scm.com

You may also like to use GitHub Desktop, which you can get from: http://desktop.github.com

You can enable Git in VSCode by following these instructions: http://code.visualstudio.com/docs/editor/github

Also: anyone who submits improvements in the spirit of the assignments (simple, light-weight, elegant demonstration of machine learning concepts -- not over-optimized or abstracted code) will get up to 2 bonus points toward the final grade.

## Python

Python is widely used in machine learning and data science to process, prepare, & explore data; and to stitch together ML tools into experiments/work flows.

I provide python framework code to get you started on the assignments, deal with data loading, etc.

Keep in mind that we are only going to be programming very basic versions of the various algorithms, and there will be very little need for optimization, so you won't need to be a python master to succeed at this course.

Please use python version 3.x (the latest python 3 version at the time of your install).

You can install python via Anaconda: http://anaconda.com/products/individual

This includes python and an environment manager that lets you cleanly work with open source tools and libraries. I'll provide detailed instructions for setting up your environment this way.

You can also install python without anaconda support via: https://www.python.org/downloads/. But this isn't the way I'm doing it, so support for this approach won't be as good.

Next, install the Python extension for Visual Studio Code from: https://marketplace.visualstudio.com/items?itemName=ms-python.python.

When you open a python file in VS code for the first time it (may) ask you to select a python interpreter. If it does, select whichever one you just installed.

You can learn more about how to use the python language at:

https://www.learnpython.org/

## Anaconda and Packages

Now we'll go through how to set up an anaconda environment for the course, install the basic packages you'll need, and link it to your vscode setup. You can find much more detail at: http://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html

But for a focused walk through, start by launching the A**naconda Prompt** application. It will bring up a window like this:


Anaconda Prompt
(base) C:\Users\ghult>

Run the command:

**> conda create -n MachineLearningCourse python=3.7**

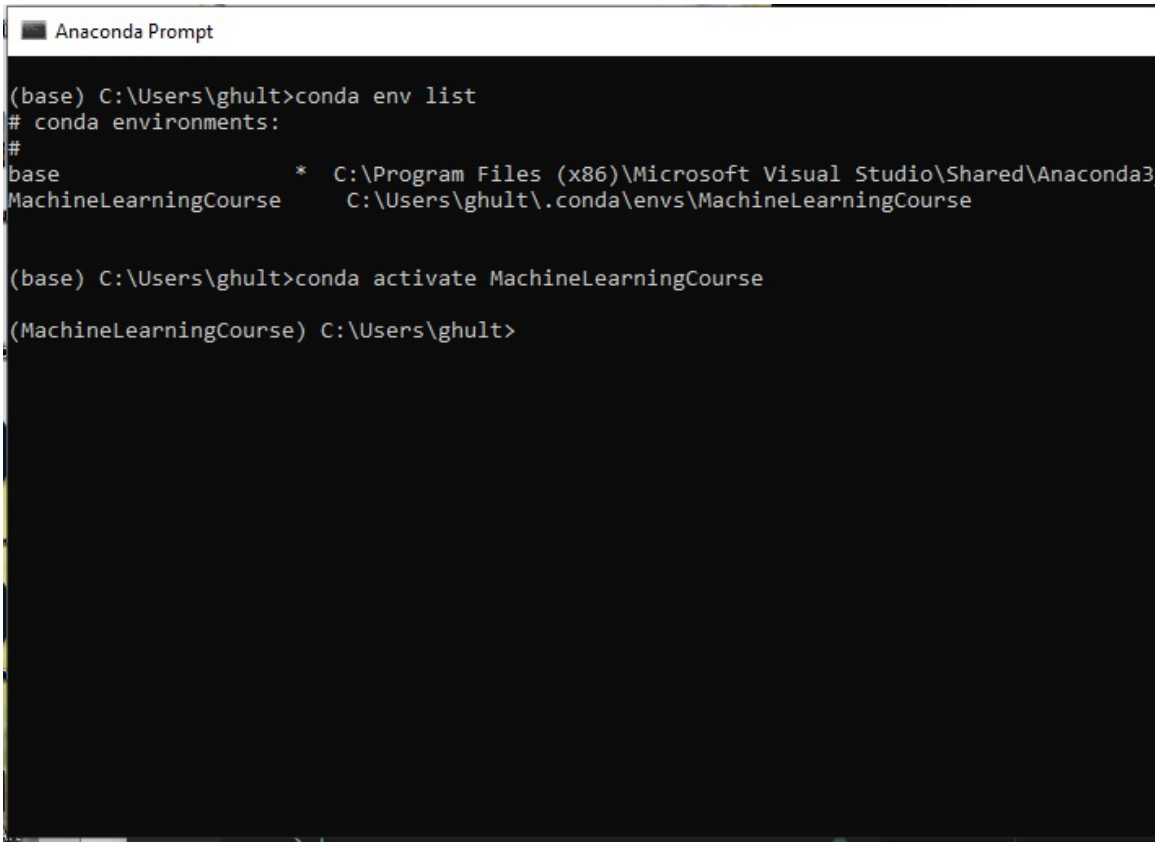
This will prompt you to proceed, then will run for a while, creating your 'MachineLearningCourse' environment and installing python and some supporting packages.

When this is done you can get a list of the environments on your machine by running:

**> conda env list**

And you can activate the MachineLearningCourse environment by running:

**> conda activate MachineLearningCourse**


Anaconda Prompt
(base) C:\Users\ghult>conda env list
# conda environments:
#
base * C:\Program Files (x86)\Microsoft Visual Studio\Shared\Anaconda3
MachineLearningCourse C:\Users\ghult\.conda\envs\MachineLearningCourse
(base) C:\Users\ghult>conda activate MachineLearningCourse
(MachineLearningCourse) C:\Users\ghult>

You can tell that the MachineLearningCourse environment is active because the name appears in parenthesis in front of the C: in the prompt.

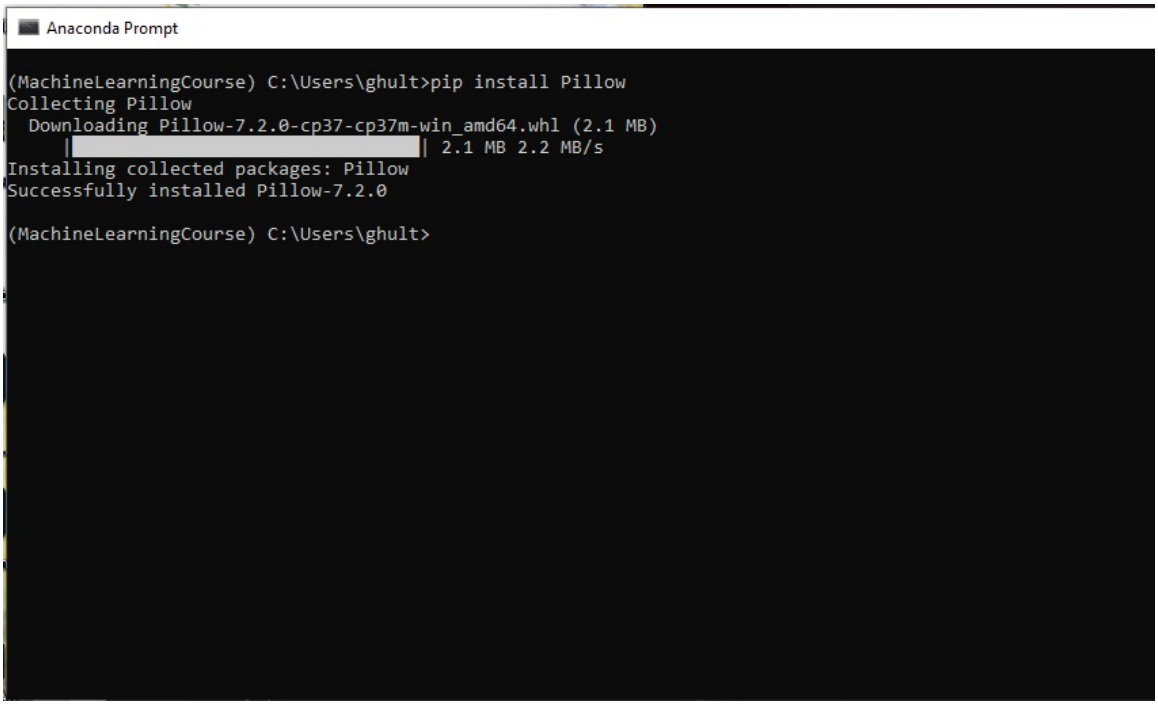
**PIL**

The Pillow Image Library supports reading and creating image files. We'll use it to help visualize our models' outputs. We'll also use it with a computer vision project later in the course.

You can find more information at pillow.readthedocs.io/en/3.0.x/installation.html

With your MachineLearningCourse environment active, install Pillow by running:

**> pip install Pillow**




**matplotlib**

Matplot lib helps make plots and charts. You don't have to use it. You might prefer loading data into some other tool (like

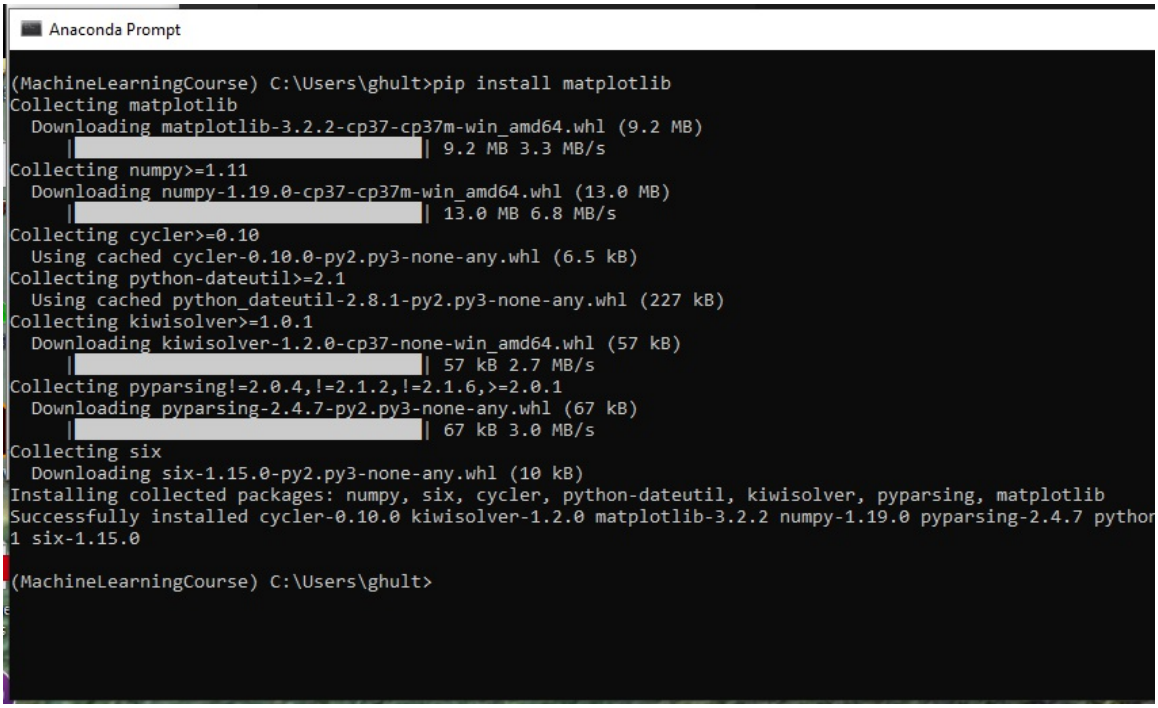
Excel) to make charts.

But if you do choose to use matplotlib you can find some tutorials at: http://matplotlib.org/tutorials/index.html

But you don't need to go learn a lot about it yet (or maybe ever) because I'll provide template code to produce charts for most of the things you'll need to complete the assignments.

You can find installation instructions at: http://matplotlib.org/users/installing.html

**> pip install matplotlib**




**JobLib**

Joblib is a basic parallel execution framework. Machine learning often involved running many (dozens or hundreds) of

slight variations of the same algorithm to find the best one. With joblib, it's easy to run these in parallel across the cores of your CPU to speed things up.

Here is a quick code snippit to give a sense of how easy it is:

--

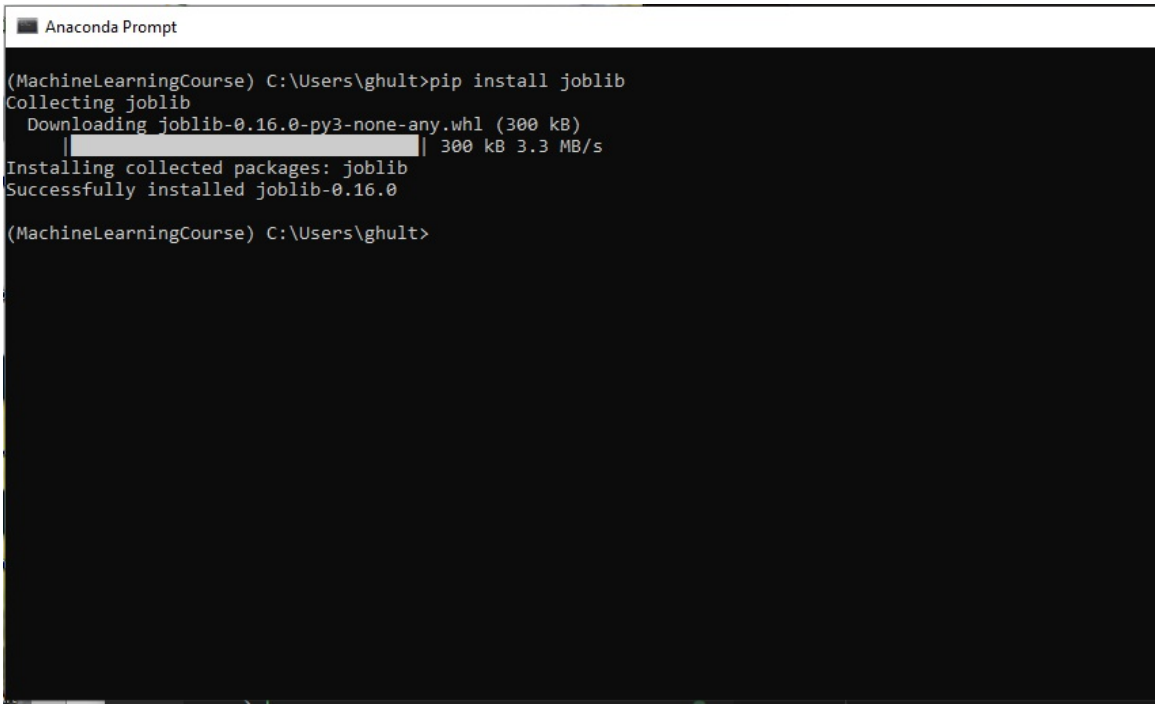
```
# Here is code to try a bunch of different parameter settings in serial
evaluations = [ Execute(parameters) for parameters in listOfAllParametersToTry ]

# Here is code to try the same set of parameters in parallel across 8 CPU cores
from joblib import Parallel, delayed
evaluations = Parallel(n_jobs=8)(delayed(Execute)(parameters) for parameters in listOfAllParametersToTry)
```

--

You can find more information at: http://joblib.readthedocs.io/en/latest/installing.html

With your 'MachineLearningCourse' environment active, install joblib by running:

**> pip install joblib**




**Linking VSCode**

The final step is to link VSCode to the 'MachineLearningCourse' environment, so that whenever your run

your programs you are doing it with access to your full setup.

You can find a lot of detail about this at: http://code.visualstudio.com/docs/python/environments

But the minimum you need to do to get going is:

1) Make sure you're using the command prompt as your terminal (and not powershell).

To do this, open the VS Code Command Palette (Ctrl+Shift+P) and search for 'Terminal: Select Default Shell' and choose 'Command Prompt'. You may need to restart VS Code for this to take effect.

2) Tell VS Code to use the interpreter from your MachineLearningCourse environment

Open the folder for your git workspace. In my case this is: c:\user\ghult\Docs\github\MachineLearningCourse

Open the Command Palette (Ctrl+Shift+P).

In the Command Palette's search box type: Python: Select Interpreter

And then select the MachineLearningCourse environment.

If it doesn't show up in the list of options you can choose 'Enter interpreter path...' and choose the python.exe in the environment directly (you can find the path by executing 'conda env list' in an **Anaconda Prompt** window).

## Interlude

That's all you need to get started with the first assignments.

We'll refer back to the remaining sections of this chapter for additional tools and libraries as the course progresses. Or maybe you're an overachiever and want to install them all right now before doing anything else...

That's fine too.

## Cuda

Cuda is a library that lets you execute computational tasks on Nvidia GPUs. If you have an Nvidia GPU, this library is an optional way to speed up one of the neural network assignments later in the course.

NOTE: At this time (late 2020) PyTorch does not (easily) support the latest version of CUDA (11.0). Please install CUDA version 10.2 via the download archive link:

https://developer.nvidia.com/cuda-10.2-download-archive

[ At some point in the future PyTorch will support CUDA 11.0, then you can find instructions for installing here: https://developer.nvidia.com/cuda-toolkit ]

Keep in mind, that CUDA works by changing your video driver. If that sounds scary to you, you might prefer to skip it and just wait a little bit longer for your neural networks to converge.

## PyTorch

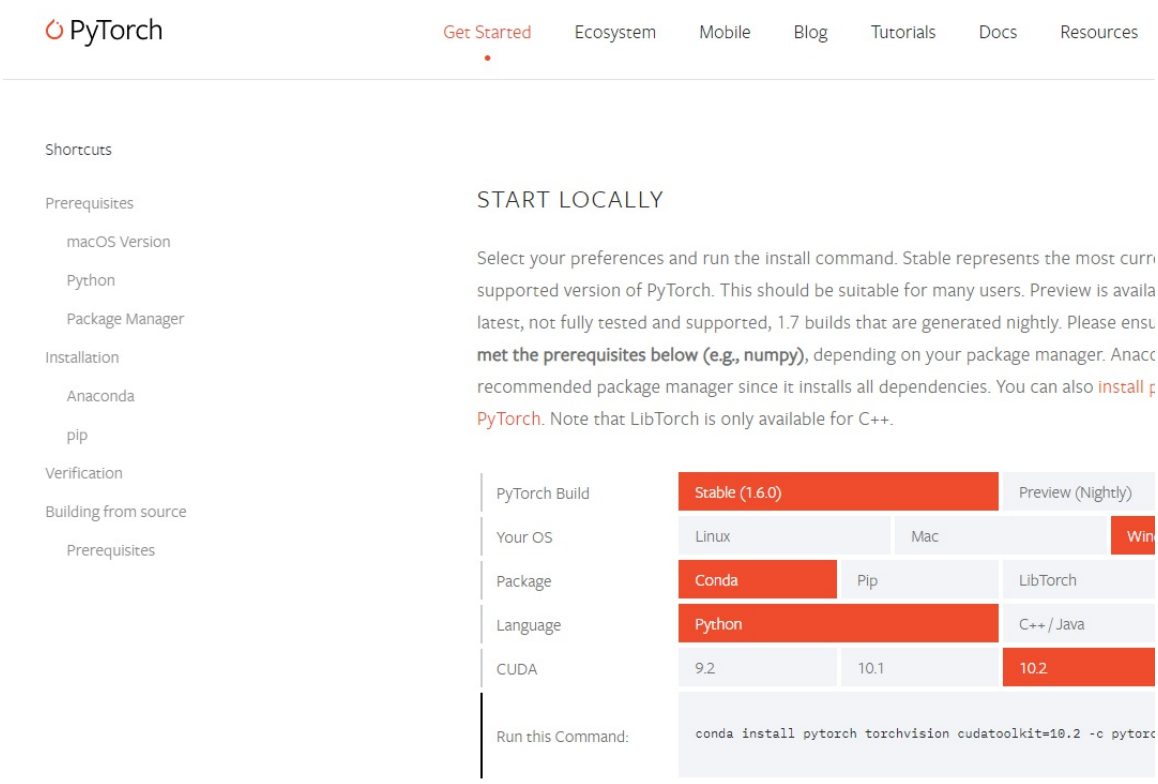
PyTorch is an open source framework for learning neural networks. We'll be using it for one assignment near the end of the course.

First decide if you're going to use CUDA or not. In my environment, CUDA saves about a factor of 10 in runtime. Then

visit the PyTorch download page to find the correct command line for installing PyTorch into the anaconda environment your using:

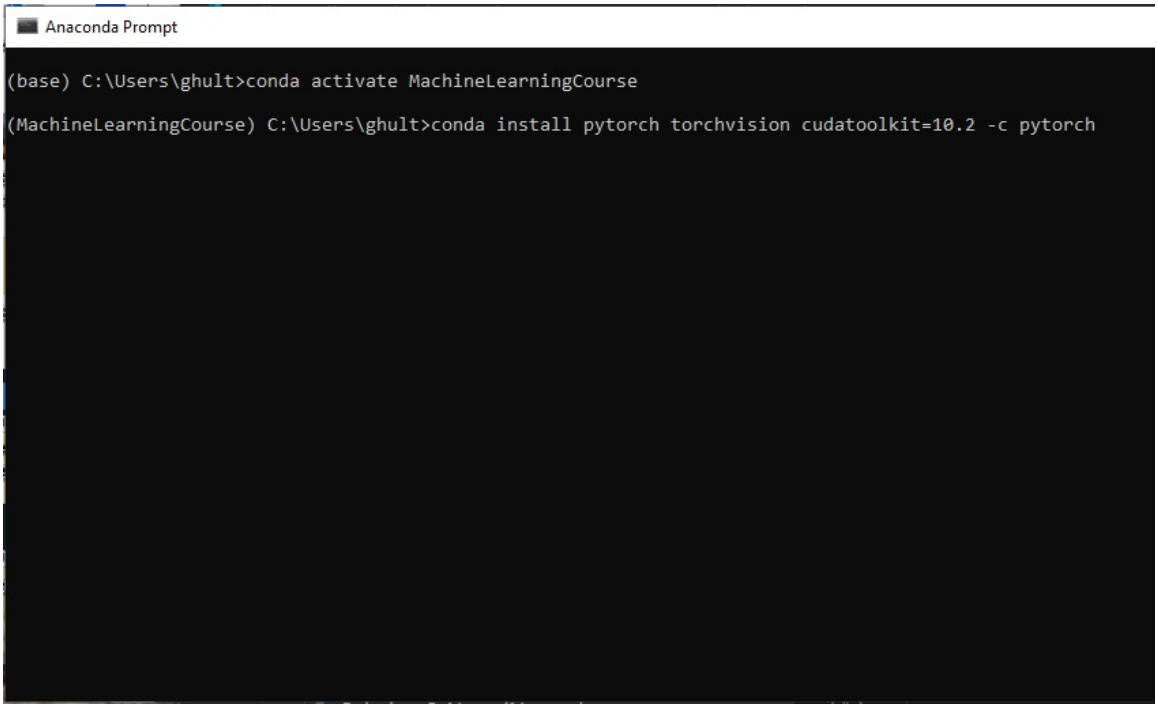
https://pytorch.org/get-started/locally/

[ NOTE as of this writing PyTorch doesn't (easily) support CUDA 11, so if you installed that by mistake, and there isn't an option for CUDA 11 on the PyTorch website, go back and install CUDA 10.2 ]


PyTorch
Get Started
Ecosystem
Mobile
Blog
Tutorials
Docs
Resources
Shortcuts
Prerequisites
macOS Version
Python
Package Manager
Installation
Anaconda
pip
Verification
Building from source
Prerequisites
START LOCALLY
Select your preferences and run the install command. Stable represents the most curr
supported version of PyTorch. This should be suitable for many users. Preview is availa
latest, not fully tested and supported, 1.7 builds that are generated nightly. Please ensu
met the prerequisites below (e.g., numpy), depending on your package manager. Anacc
recommended package manager since it installs all dependencies. You can also install p
PyTorch. Note that LibTorch is only available for C++.
PyTorch Build
Stable (1.6.0)
Preview (Nightly)
Your OS
Linux
Mac
Win
Package
Conda
Pip
LibTorch
Language
Python
C++ / Java
CUDA
9.2
10.1
10.2
Run this Command:
conda install pytorch torchvision cudatoolkit=10.2 -c pytorc

To install the version indicated with these settings, go to your conda prompt, activate the MachineLearningCourse environment and run the command:

**> conda install pytorch torchvision cudatoolkit=10.2 -c pytorch**




**Gym**

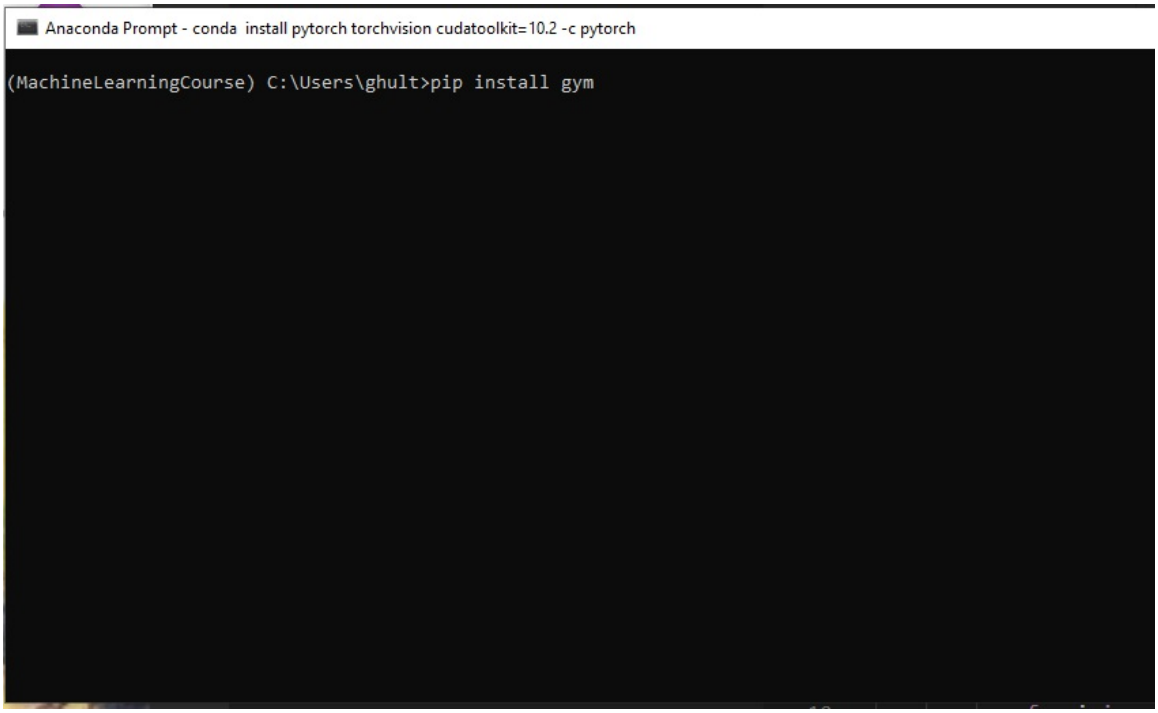
Gym is a toolkit for experimenting with reinforcement learning algorithms. It allows you to interact with a number of

enviornments using a consistent API and provides visualizations of your algorithms' progress at learning to interact with these environments.

You can find instructions for installing gym at:

https://gym.openai.com/docs/#installation

**> pip install gym**


Anaconda Prompt - conda install pytorch torchvision cudatoolkit=10.2 -c pytorch
(MachineLearningCourse) C:\Users\ghult>pip install gym

# Endnotes