

Geoff Hulten  
Ghulten@cs.washington.edu

2,000 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.

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.

## 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 **Anaconda Prompt** application. It will bring up a window like this:



Run the command:

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

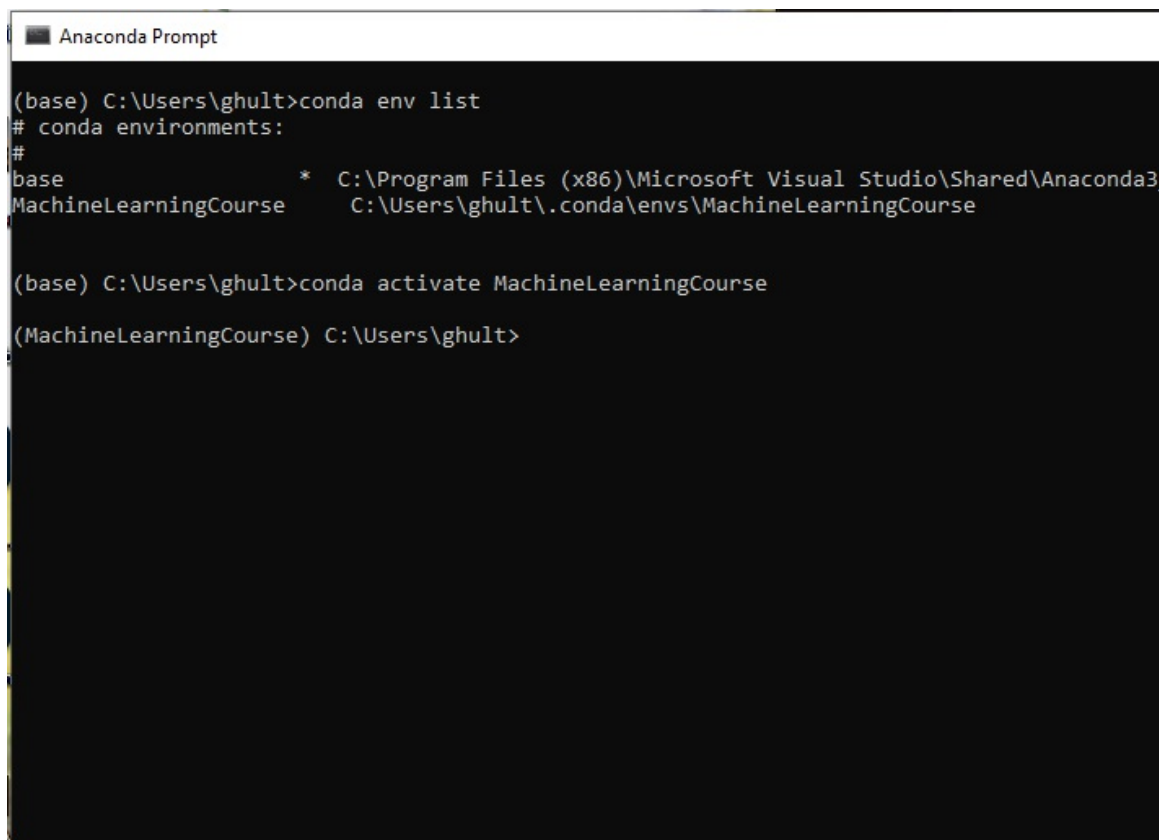
This will prompt you to proceed, then will run for a while, creating your 'MachineLearningCourse' environment and installing python and some supporting packages.

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.

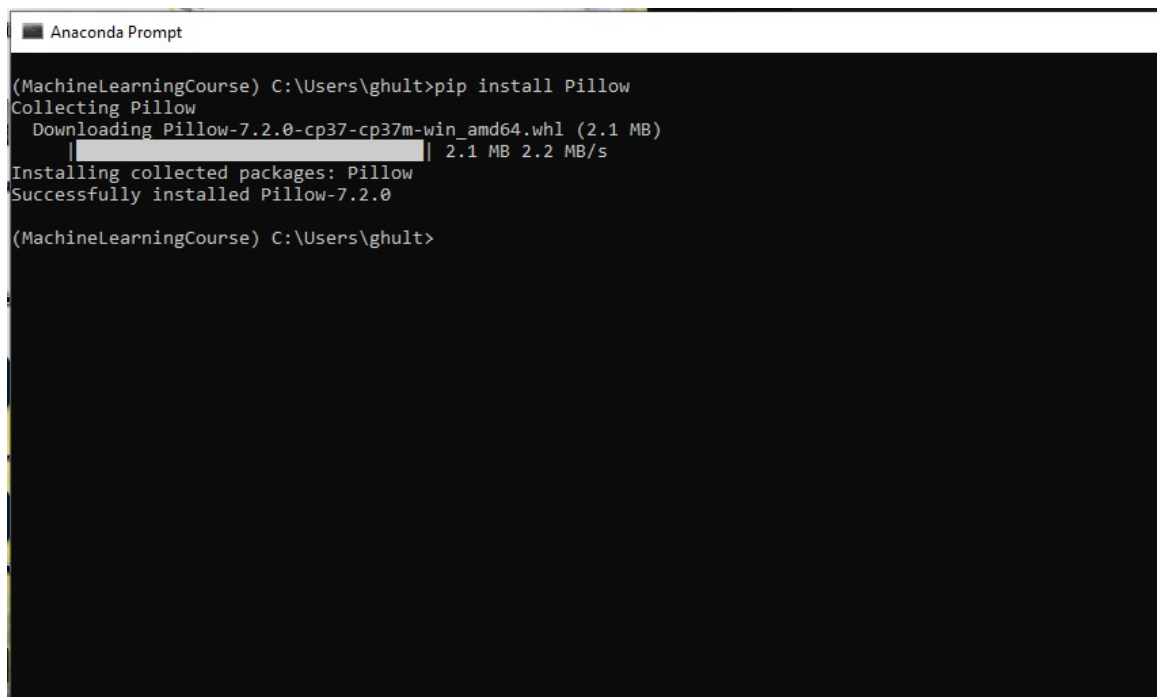
## **PIL**

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  
<http://pillow.readthedocs.io/en/3.0.x/installation.html>

With your MachineLearningCourse environment active, install Pillow by running:

```
> pip install Pillow
```

A screenshot of an Anaconda Prompt window. The title bar reads "Anaconda Prompt". The command prompt shows the user running the command `pip install Pillow`. The output indicates that Pillow is being collected and the specific wheel file `Pillow-7.2.0-cp37-cp37m-win_amd64.whl` (2.1 MB) is being downloaded at 2.2 MB/s. It then shows the installation of the collected packages and a successful completion message for Pillow-7.2.0. The prompt returns to the user's shell.

```
(MachineLearningCourse) C:\Users\ghult>pip install Pillow
Collecting Pillow
  Downloading Pillow-7.2.0-cp37-cp37m-win_amd64.whl (2.1 MB)
    | 2.1 MB 2.2 MB/s
Installing collected packages: Pillow
Successfully installed Pillow-7.2.0

(MachineLearningCourse) C:\Users\ghult>
```

## 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
```



```
Anaconda Prompt

(MachineLearningCourse) C:\Users\ghult>pip install matplotlib
Collecting matplotlib
  Downloading matplotlib-3.2.2-cp37-cp37m-win_amd64.whl (9.2 MB)
    | 9.2 MB 3.3 MB/s
Collecting numpy>=1.11
  Downloading numpy-1.19.0-cp37-cp37m-win_amd64.whl (13.0 MB)
    | 13.0 MB 6.8 MB/s
Collecting cyclar>=0.10
  Using cached cyclar-0.10.0-py2.py3-none-any.whl (6.5 kB)
Collecting python-dateutil>=2.1
  Using cached python_dateutil-2.8.1-py2.py3-none-any.whl (227 kB)
Collecting kiwisolver>=1.0.1
  Downloading kiwisolver-1.2.0-cp37-none-win_amd64.whl (57 kB)
    | 57 kB 2.7 MB/s
Collecting pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1
  Downloading pyparsing-2.4.7-py2.py3-none-any.whl (67 kB)
    | 67 kB 3.0 MB/s
Collecting six
  Downloading six-1.15.0-py2.py3-none-any.whl (10 kB)
Installing collected packages: numpy, six, cyclar, python-dateutil, kiwisolver, pyparsing, matplotlib
Successfully installed cyclar-0.10.0 kiwisolver-1.2.0 matplotlib-3.2.2 numpy-1.19.0 pyparsing-2.4.7 python-
1 six-1.15.0

(MachineLearningCourse) C:\Users\ghult>
```

## 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 snippet 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)

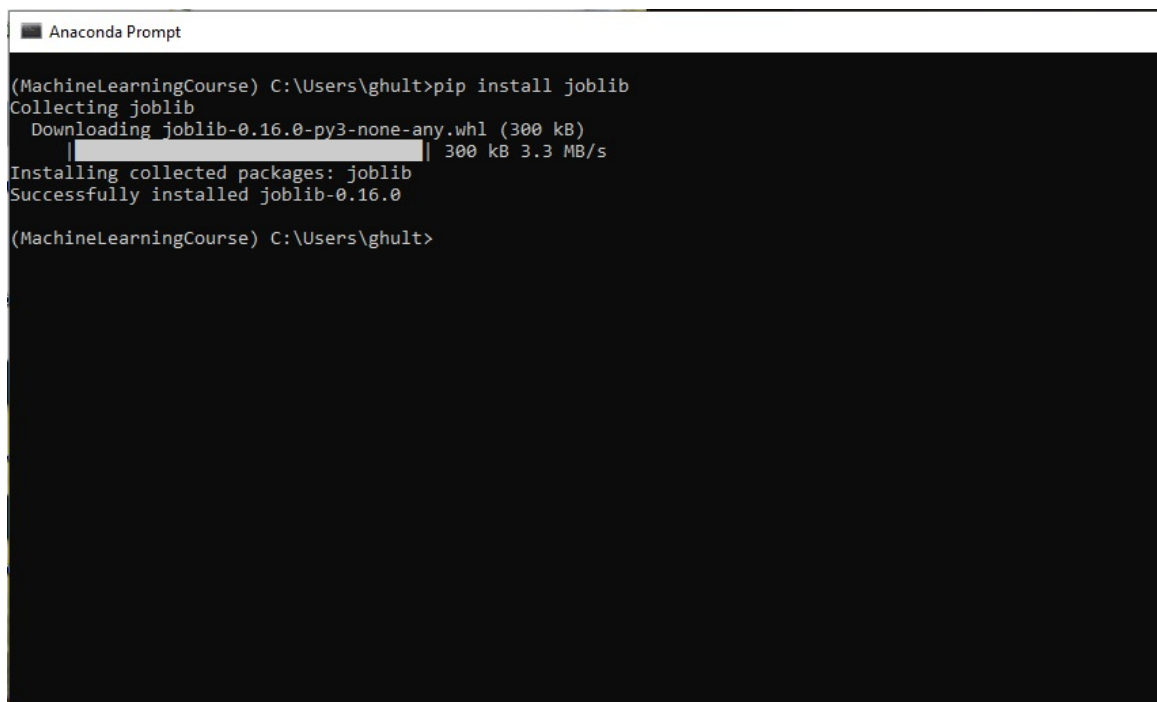
--
```

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
```

A screenshot of an Anaconda Prompt terminal window. The window title is "Anaconda Prompt". The terminal shows the command `(MachineLearningCourse) C:\Users\ghult>pip install joblib` being executed. The output shows the package being collected, downloaded (300 kB at 3.3 MB/s), and successfully installed. The prompt then returns to `(MachineLearningCourse) C:\Users\ghult>`.

```
(MachineLearningCourse) C:\Users\ghult>pip install joblib
Collecting joblib
  Downloading joblib-0.16.0-py3-none-any.whl (300 kB)
    | 300 kB 3.3 MB/s
Installing collected packages: joblib
Successfully installed joblib-0.16.0

(MachineLearningCourse) C:\Users\ghult>
```

## Linking VSCode

The final step is to link VSCode to the 'MachineLearningCourse' environment, so that whenever you 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).

## Setting PYTHONPATH

The PYTHONPATH variable lets python know where to search for code modules (like the all the code provided by the framework). You need to set it correctly or the various parts of the framework won't know where to find one another.

If you're using VSCode you can automatically get a correct PYTHONPATH by 'opening' the folder that contains your local copy of the 'MachineLearningCourse' git repository. On my system this is: 'C:\users\ghult\GitHub'.

You can do this by launching VSCode and selecting:

**File->Open Folder...**

Then browse to the folder that contains your local copy of the 'MachineLearningCourse' git repository (your version of 'C:

\users\ghult\GitHub') and click 'select folder'. This will access the MachineLearningCourse\.vscode\launch.json config file to do the right thing.

You can also set the python path manually using a command like (with the correct path):

```
export  
PYTHONPATH="${PYTHONPATH}:/users/ghult/GitHub/"
```

## 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 ]


[Get Started](#)
[Ecosystem](#)
[Mobile](#)
[Blog](#)
[Tutorials](#)
[Docs](#)
[Resources](#)

## START LOCALLY

Select your preferences and run the install command. Stable represents the most current supported version of PyTorch. This should be suitable for many users. Preview is available latest, not fully tested and supported, 1.7 builds that are generated nightly. Please ensure **met the prerequisites below (e.g., numpy)**, depending on your package manager. Anaconda is recommended package manager since it installs all dependencies. You can also [install PyTorch](#). Note that LibTorch is only available for C++.

PyTorch Build	Stable (1.6.0)		Preview (Nightly)
Your OS	Linux	Mac	Windows
Package	Conda	Pip	LibTorch
Language	Python		C++ / Java
CUDA	9.2	10.1	10.2
Run this Command:	<pre>conda install pytorch torchvision cudatoolkit=10.2 -c pytorch</pre>		

Shortcuts

Prerequisites

- macOS Version
- Python
- Package Manager

Installation

- Anaconda
- pip

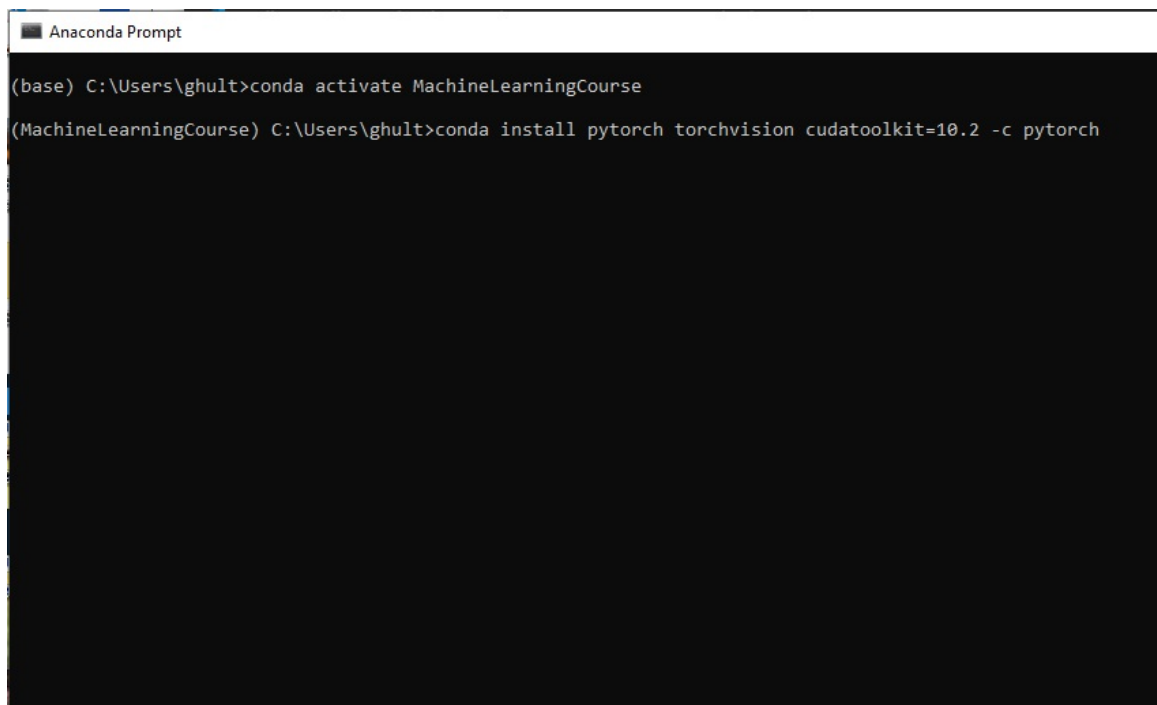
Verification

Building from source

Prerequisites

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
```



```

Anaconda Prompt

(base) C:\Users\ghult>conda activate MachineLearningCourse
(MachineLearningCourse) C:\Users\ghult>conda install pytorch torchvision cudatoolkit=10.2 -c pytorch

```

## Gym

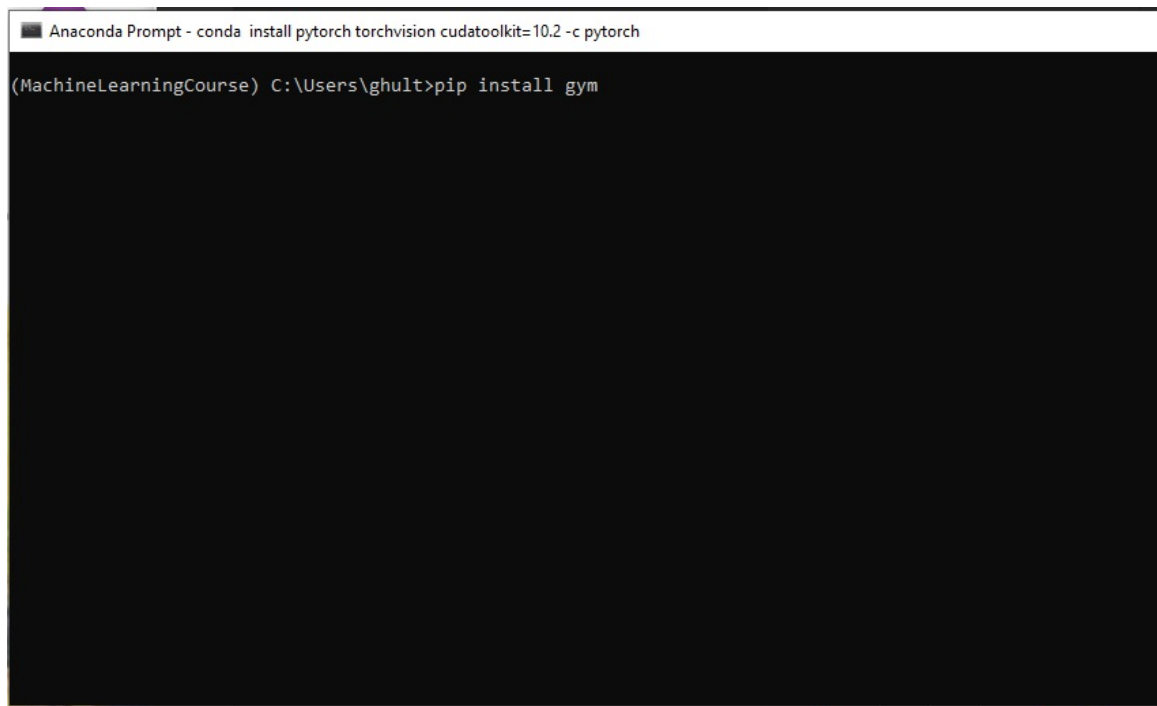
Gym is a toolkit for experimenting with reinforcement learning algorithms. It allows you to interact with a number of

environments 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