Installing a Python Based Machine Learning Environment in Windows 10

Frank Ceballos Follow
Dec 18, 2018 · 5 min read

Overview

When I first got into machine learning it took me a few hours to figure how to properly set my Python environment. Out of frustration, I

decided to write this post to help anyone going through the process. We will start by installing Anaconda Navigator which will allow us to create independent environments, this will come really handy. Additionally, with Anaconda we can easily install compatible Python modules with very simple commands. Finally, we can use Anaconda as an Python development environment. If you follow the step-by-step procedure shown below, you will have installed Tensorflow, Keras, and Scikit-learn in no time. The set of instructions described below will show you how to create Anaconda environments to install any Python module.

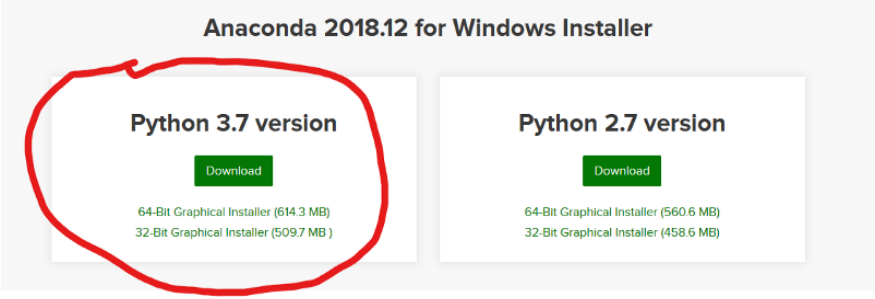
```
The following set of instructions were compiled from across
the web and written for a Windows 10 OS. Last tested on
02/09/2019.
```

Installing Anaconda

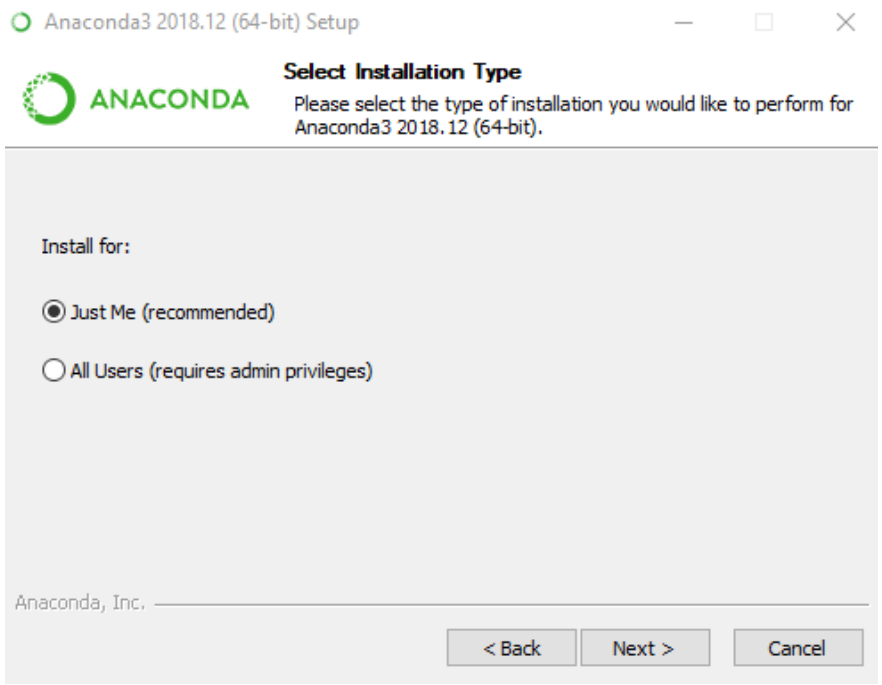
In order to start building your machine learning (ML) models with Python, we will start by installing Anaconda Navigator. Anaconda provides an efficient and easy way to install Python modules on your machine. So let's get started.

Installing Tensorflow, Keras, and Scikit-learn

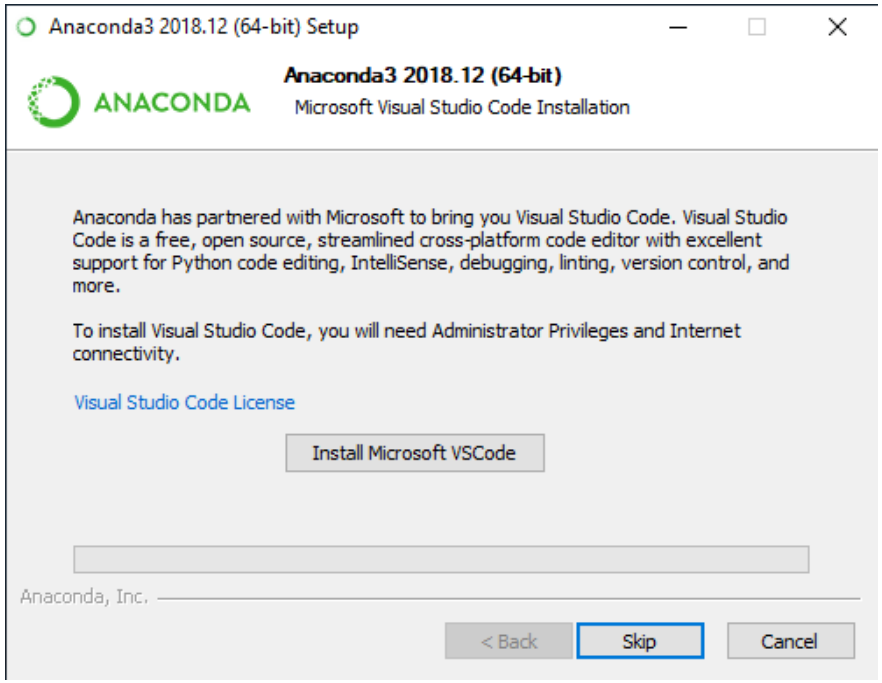
1. Download and install the latest version of Anaconda Navigator for your operating system.



2. Proceed with the installation wizard but skip the step where you need to download and install VS, we will do this later. Additionally, make sure to install Anaconda Navigator to a single user. At the time this set of instructions were written, installing VS with the setup wizard made my Anaconda installation fail. Additionally, installing Anaconda for All Users might lead to problems. For example, you won't be able to install any modules because Anaconda won't have the necessary privileges.

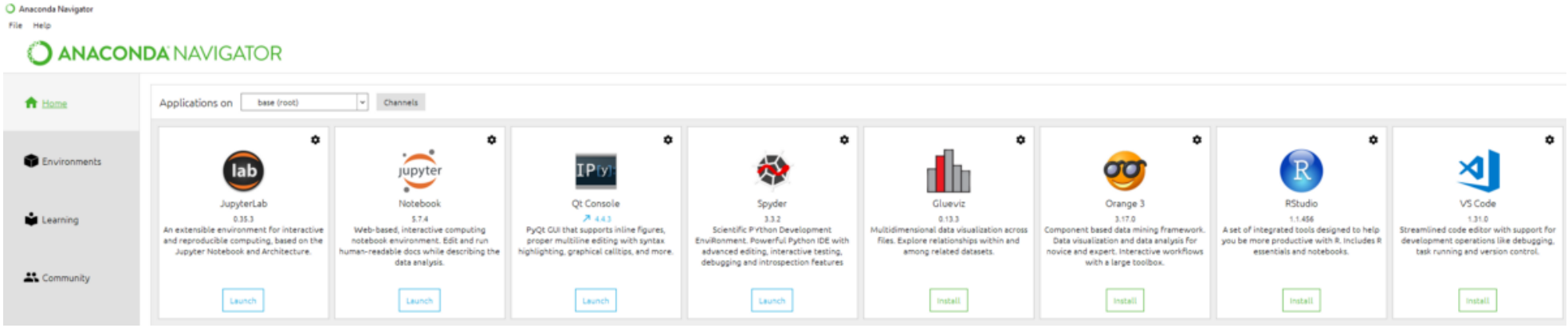


Make sure you install Anaconda to the current user else you might face problems down the road.



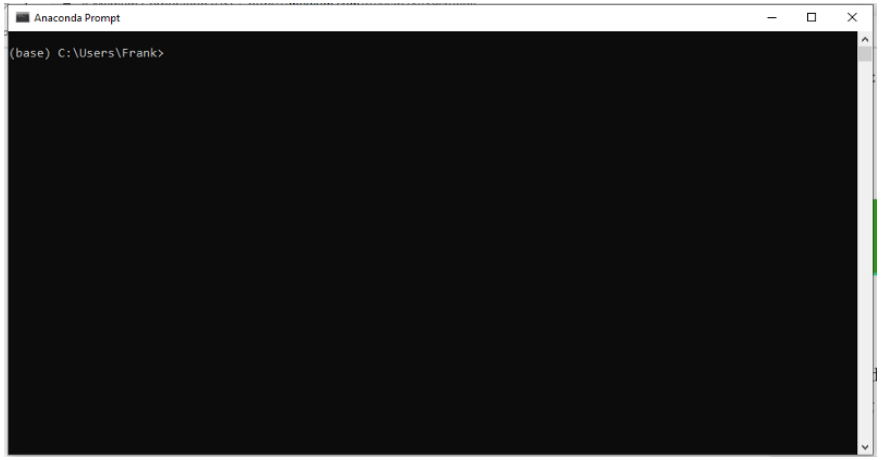
Skip this step. We will get it done in a bit.

3. Launch Anaconda Navigator and select the Home Tab, it should be selected by default. Find the VS Code Panel and click on the Install button. This will take a minute or two.



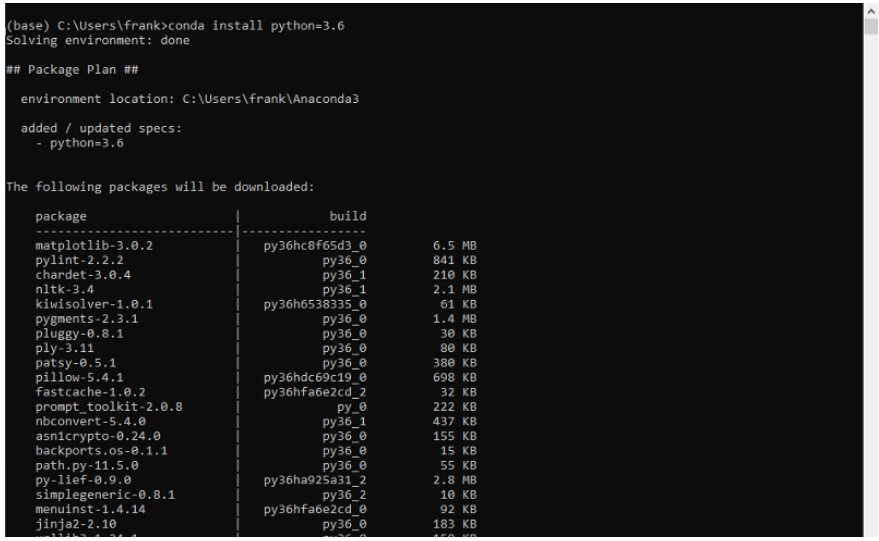
After you install VS Code, you will be able to see a Launch button under the VS Code panel.

4. Close Anaconda Navigator and launch Anaconda Prompt. Launch Anaconda prompt by searching for it in the windows search bar. The following terminal should open. Notice that this will open on the **base** Anaconda environment.



5. Downgrade Python to a Keras & Tensorflow compatible version. Anaconda will start to look for all the compatible modules for Python 3.6. This might take a few minutes. To downgrade to Python 3.6 use the following command:

```
conda install python=3.6
```



After the environment is resolved, Anaconda will show you all the packages that will be downloaded.

6. Create a new conda environment where we will install our modules to built our models using the GPU. To do so, execute the following command:

```
conda create --name PythonGPU
```

Note: Ensure that you have a NVIDIA graphics card. If you don't, install the CPU version of Keras.

If you want to use your CPU instead, execute the following command:

```
conda create --name PythonCPU
```

Follow the instructions displayed on the terminal. Conda environments give the user the liberty to install very specific modules that are independent habitats. Personally, I created two environments. One where I can built my models using the CPU and the other where I can built my models using the GPU. For more information about conda environments, I suggest you take a look at the official documentation.

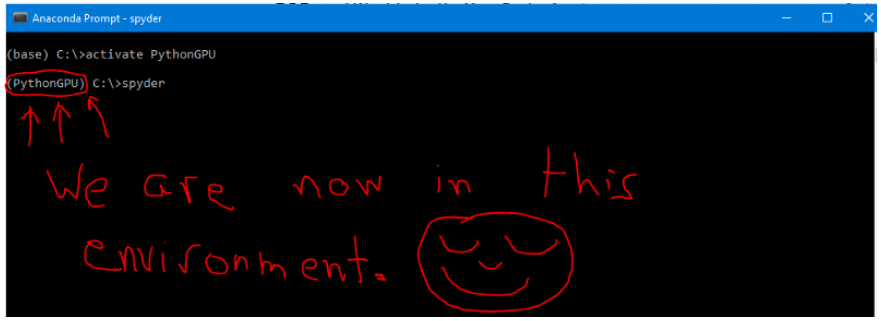
7. To activate the conda environment that was just created use:

`activate PythonGPU` or `activate PythonCPU`

To deactivate the environment use:

```
conda deactivate
```

Do not deactivate the environment yet, we are about to install all the good stuff.



8. To install Keras & Tensorflow GPU versions, the modules that are necessary to create our models with our GPU, execute the following command:

```
conda install -c anaconda keras-gpu
```

If you want to use your CPU to built models, execute the following command instead:

```
conda install -c anaconda keras
```

A lot of computer stuff will start happening.

9. Install **Spyder,** the editor that I currently used to code in Python.

```
conda install spyder
```

10. Install **Pandas**. Pandas is a library that is extremely powerful and allows you to easily read, manipulate, and visualize data.

```
conda install -c anaconda pandas
```

If you want to read Excel files with Pandas, execute the following commands:

```
conda install -c anaconda xlrd
```

```
conda install -c anaconda xlwt
```

11. Install the **Seaborn** library. Seaborn is an amazing library that allows you to easily visualize your data.

```
conda install -c anaconda seaborn
```

12. To install scikit-learn.

```
conda install -c anaconda scikit-learn
```

Adding Missing Modules

By now you should feel comfortable installing modules using the conda command. If you need a specific module, simply Google something along the following lines:

```
Anaconda LibraryNameYouWant Install
```

If you encounter any problems search the web. Is most likely that you're not the first person to encounter a given error.

Launching Spyder and Checking That All Modules Were Installed Correctly

To launch Spyder, first activate the conda environment you want (PythonCPU or PythonGPU) and execute the following command:

```
spyder
```

To ensure everything was installed correctly, execute the following lines of code on the python console:

```
import numpy as np # For numerical fast numerical
calculations
```

```
import matplotlib.pyplot as plt # For making plots
import pandas as pd # Deals with data
import seaborn as sns # Makes beautiful plots
from sklearn.preprocessing import StandardScaler # Testing
sklearn
import tensorflow # Imports tensorflow
import keras # Imports keras
```

```
Python 3.6.8 |Anaconda, Inc.| (default, Dec 30 2018, 18:50:55) [MSC v.1915 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.2.0 -- An enhanced Interactive Python.

In [1]: import numpy as np # For numerical fast numerical calculations
   ...: import matplotlib.pyplot as plt # For making plots
   ...: import pandas as pd # Deals with data
   ...: import seaborn as sns # Makes beautiful plots
   ...: from sklearn.preprocessing import StandardScaler # Testing sklearn
   ...: import tensorflow # Imports tensorflow
   ...: import keras # Imports keras
Using TensorFlow backend.
```

If you see no ModuleImport errors, you're now ready to start building machine learning based models using Keras, Tensorflow, and Scikit-Learn.

You can find my LinkedIn here.