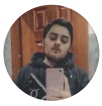
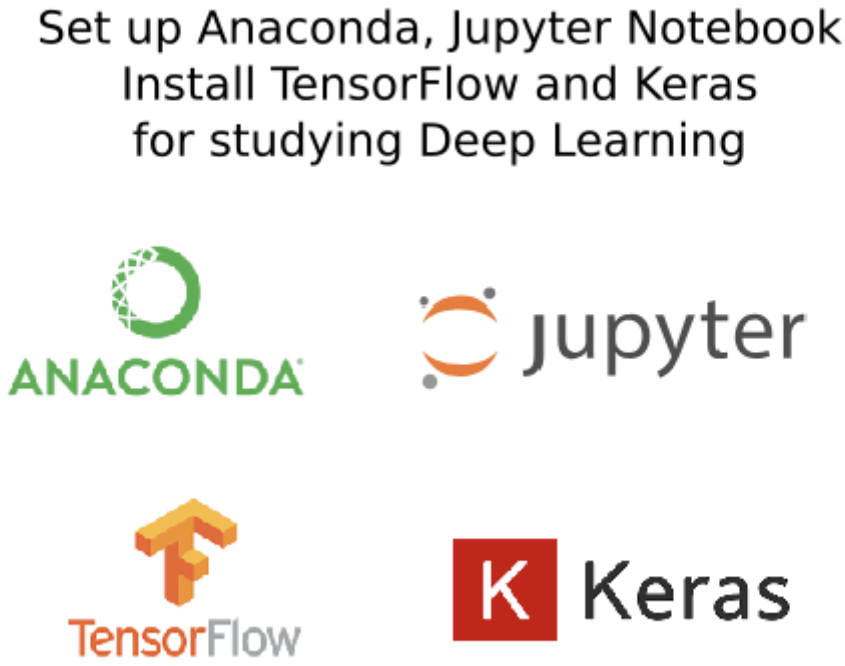


Hussnain Fareed Follow

Feb 24, 2018 · 6 min read · · Listen

Setup a Python Environment for Machine Learning and Deep Learning



Interest in **Machine Learning and Deep Learning** has exploded over the past decade. You see machine learning in computer science programs, industry conferences, and in many applications in daily life.

So, I find many beginners facing problems while installing libraries and setting up the environment. As I have faced the first time when I was trying. So this guide is totally for beginners.

In this story, I will tell you how you can easily set up a python environment on your system. I am using Windows but this guide is also suitable for Ubuntu & Linux users.

After completing this tutorial, you will have a working Python environment to begin learning and developing machine learning and deep learning software.

PC Hardware Setup

First of all to perform machine learning and deep learning on any dataset, the software/program requires a computer system powerful enough to handle the computing power necessary. So the following is required:

1. **Central Processing Unit (CPU)** — Intel Core i5 6th Generation processor or higher. An AMD equivalent processor will also be optimal.
2. **RAM** — 8 GB minimum, 16 GB or higher is recommended.
3. **Graphics Processing Unit (GPU)** — NVIDIA GeForce GTX 960 or higher. AMD GPUs are not able to perform deep learning regardless. For more information on NVIDIA GPUs for deep learning please visit https://developer.nvidia.com/cuda-gpus.
4. **Operating System** — Ubuntu or Microsoft Windows 10. I recommend updating Windows 10 to the latest version before proceeding forward.

Note: In the case of laptops, the ideal option would be to purchase a gaming laptop from any vendor deemed suitable such as Alienware, ASUS, Lenovo Legion, Acer Predator, etc.

Let's just get straight to the installation process. we are gonna hit the rock 😉

Download NVIDIA GeForce Experience

Table of Contents

In this tutorial, we will cover the following steps:

1. Download Anaconda
2. Install Anaconda & Python
3. Start and Update Anaconda
4. Install CUDA Toolkit & cuDNN
5. Create an Anaconda Environment
6. Install Deep Learning API's (TensorFlow & Keras)

Step 1: Download Anaconda

In this step, we will download the Anaconda Python package for your platform.

Anaconda is a free and easy-to-use environment for scientific Python.

- 1.Install Anaconda (Python 3.6 version) Download

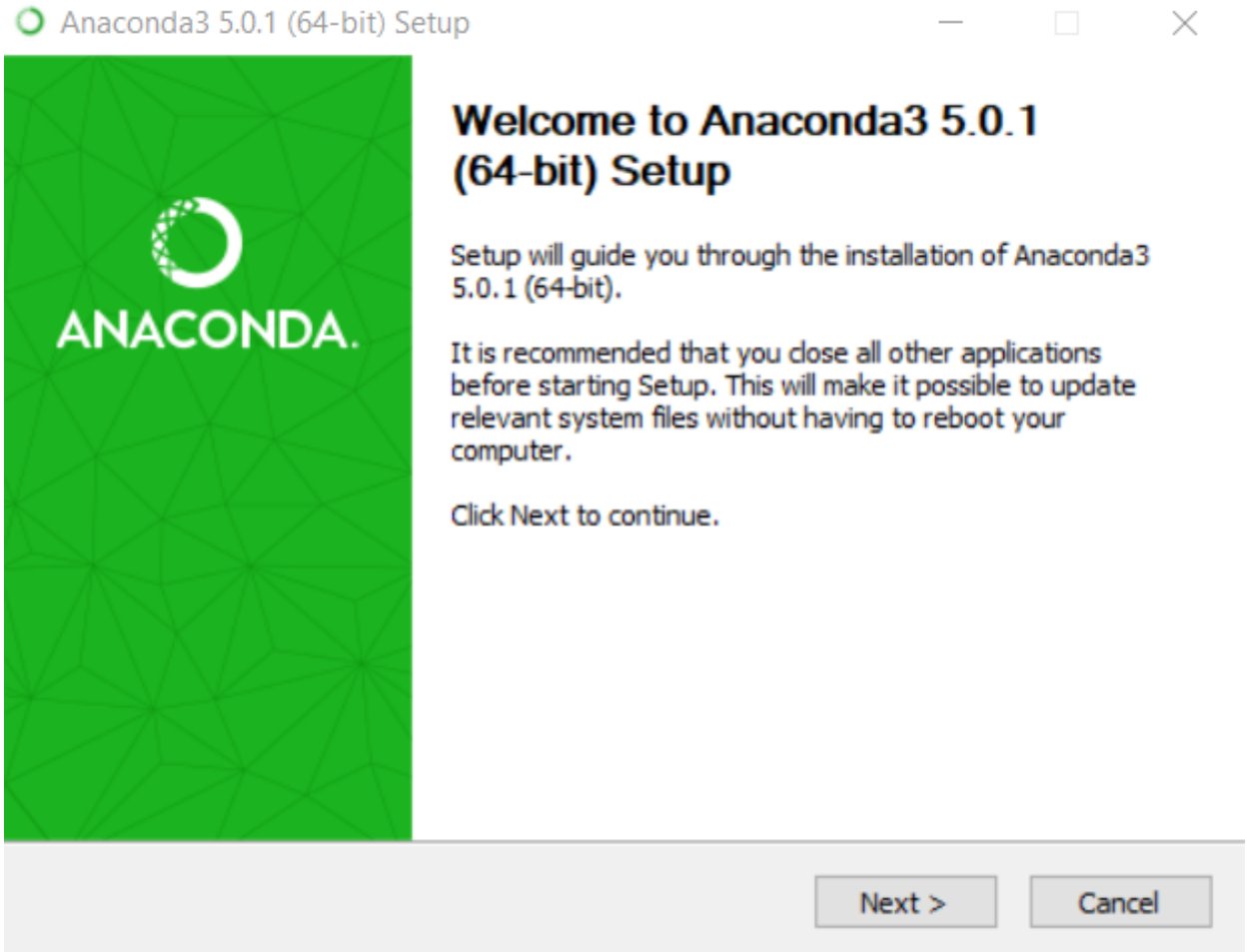
Step 2: Install Anaconda

In this step, we will install the Anaconda Python software on your system.

Installation is very easy and quick once you download the setup. Open the setup and follow the wizard instructions.

#Note: *It will automatically install Python and some basic libraries with it.*

It might take 5 to 10 minutes or some more time according to your system.



Step 3: Update Anaconda

Open Anaconda Prompt to type the following command(s). Don't worry Anaconda Prompt works the same as cmd.

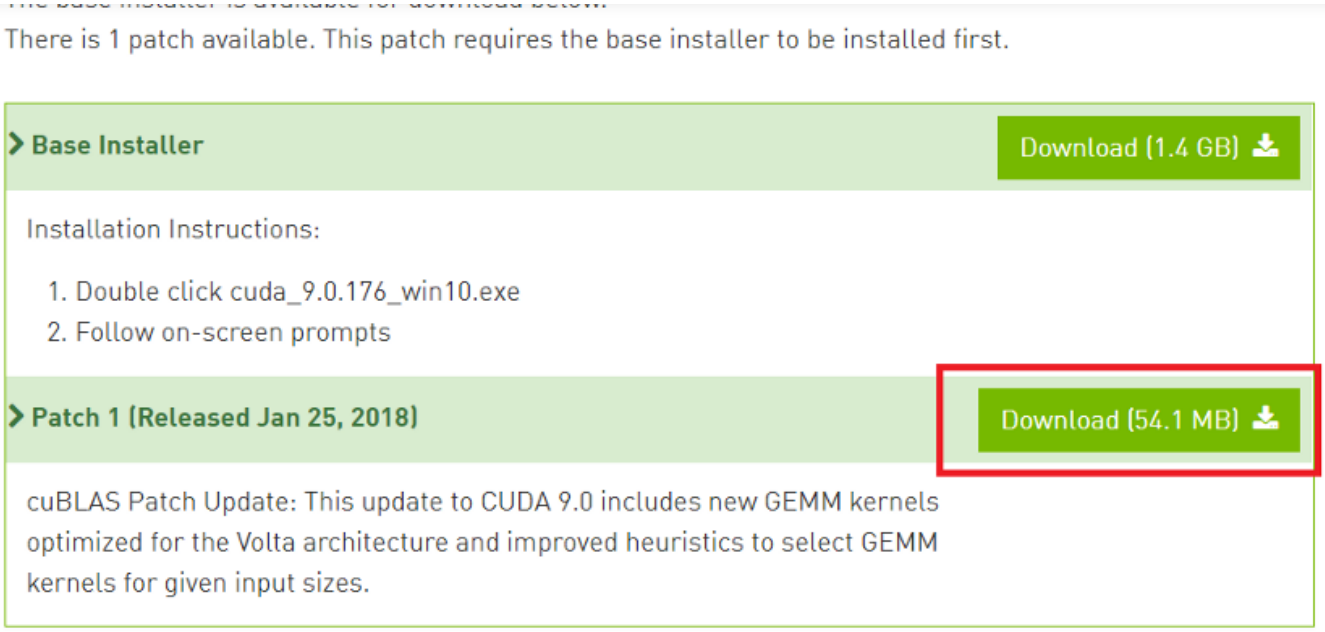
```
conda update conda
```

Choose your version depending on your Operating System and GPU.

#Version Support: Here is a guide to check that if your version supports your Nvidia Graphic Card

For downloading other versions you can follow this link: https://developer.nvidia.com/cuda-toolkit-archive

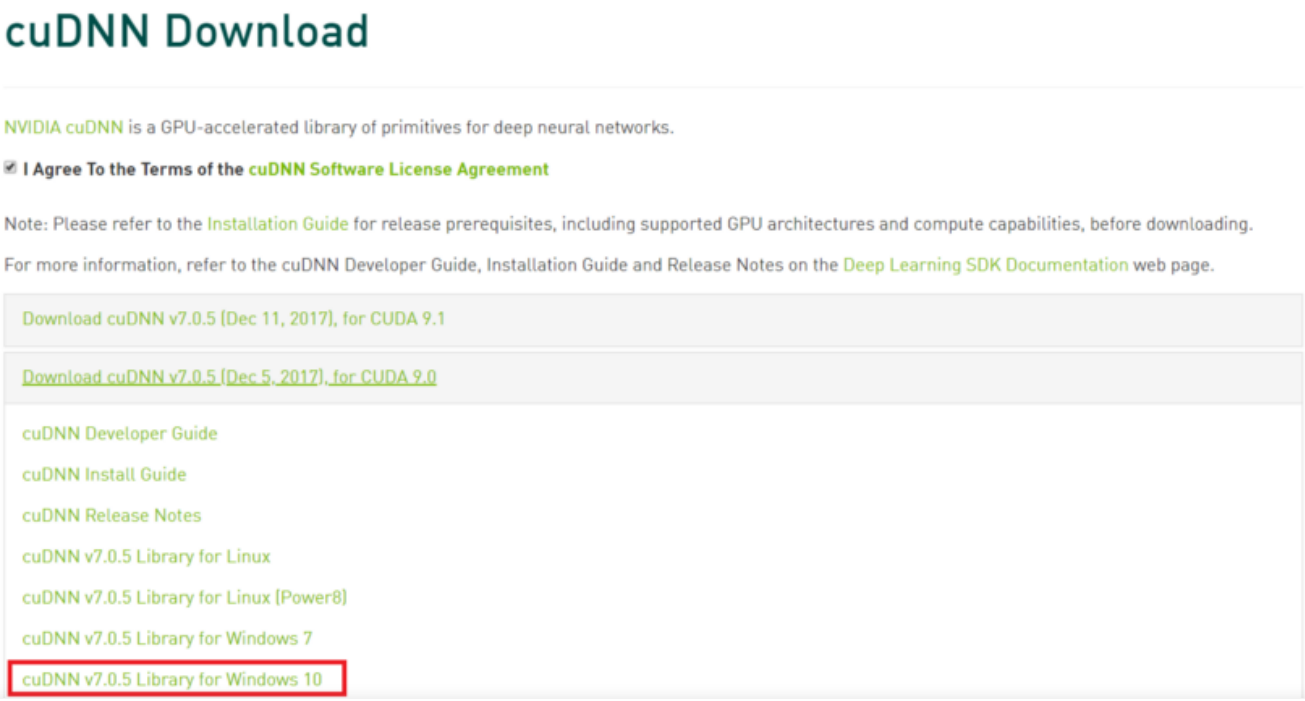
#Note: *CUDA 9.0 is recommended as TensorFlow is NOT compatible with CUDA Toolkit 9.1 and 9.2 versions. Kindly choose the CUDA version according to your Nvidia GPU version to avoid errors.*



#Note: People with version 9.0 Download can also install the given patch in any case of error while proceeding.

2. Download cuDNN Download

Download the latest version of cuDNN. Choose your version depending on your Operating System and CUDA. Membership registration is required. Don't worry you can easily create an account using your email.



```
C:\cudnn-9.0-windows10-x64-v7
```

Step 5: Add cuDNN into Environment Path

1. Open Run dialogue using (*Win + R)* and run the command **sysdm.cpl**
2. In Window-10 **System Properties**, please select the Tab **Advanced**.
3. Select Environment Variables
4. Add the following path to your Environment.

```
C:\cudnn-9.0-windows10-x64-v7\cuda\bin
```

Step 6: Create an Anaconda Environment

Here we will create a new anaconda environment for our specific usage so that it will not affect the root of Anaconda. Amazing!! isn't it? 😛

Open Anaconda Prompt to type the following commands.

1. Create a conda environment named "tensorflow" (you can change the name) by invoking the following command:

```
conda create -n tensorflow pip python=3.6
```

2. Activate the conda environment by issuing the following command:

```
activate tensorflow
 (tensorflow)C:>  # Your prompt should change
```

Step 7: Install Deep Learning Libraries


1. **TensorFlow**

TensorFlow is a tool for machine learning. While it contains a wide range of functionality, TensorFlow is mainly designed for deep neural network models.

=> For installing TensorFlow, Open Anaconda Prompt to type the following commands.

To install the GPU version of TensorFlow:

```
C:\> pip install tensorflow-gpu
```

To install the CPU-only version of TensorFlow:

```
C:\> pip install tensorflow
```

If your machine or system is the only CPU supported you can install CPU version for basic learning and practice.

=> You can test the installation by running this program on shell:

```
>>> import tensorflow as tf
>>> hello = tf.constant('Hello, TensorFlow!')
>>> sess = tf.Session()
>>> print(sess.run(hello))
```

For getting started and documentation you can visit ***TensorFlow*** website.

2. **Keras**

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano.


=> Let's try running ***Mnist_Mlp.Py*** in your prompt. you can use other examples as well.

Open Anaconda Prompt to type the following commands.

```
activate tensorflow
python mnist_mlp.py
```

For getting started and documentation you can visit ***Keras*** website.

Here is an implementation of **Keras Standard Fully Connected Neural Network using Python for Digit Recognition** I have done.

There are some other famous libraries like ***Pytorch, Theano, and Caffe2*** you can use as per your choice and use.

Congratulations! 😉 You have successfully created the environment using TensorFlow, Keras (with Tensorflow backend) over GPU on Windows!

If you enjoyed this story, please click the 👏 button and share to help others find it. You can also buy me a coffee to help me write more interesting content for you!