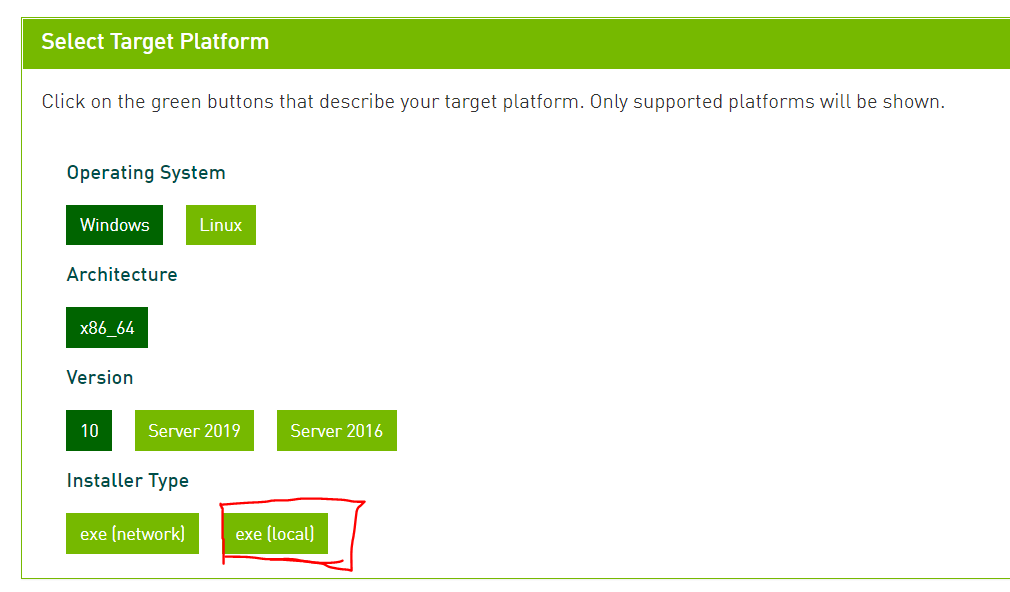
Installing CUDA Toolkit and cuDNN for Deep Learning:

Step 1:

Download Cuda toolkit 10.0



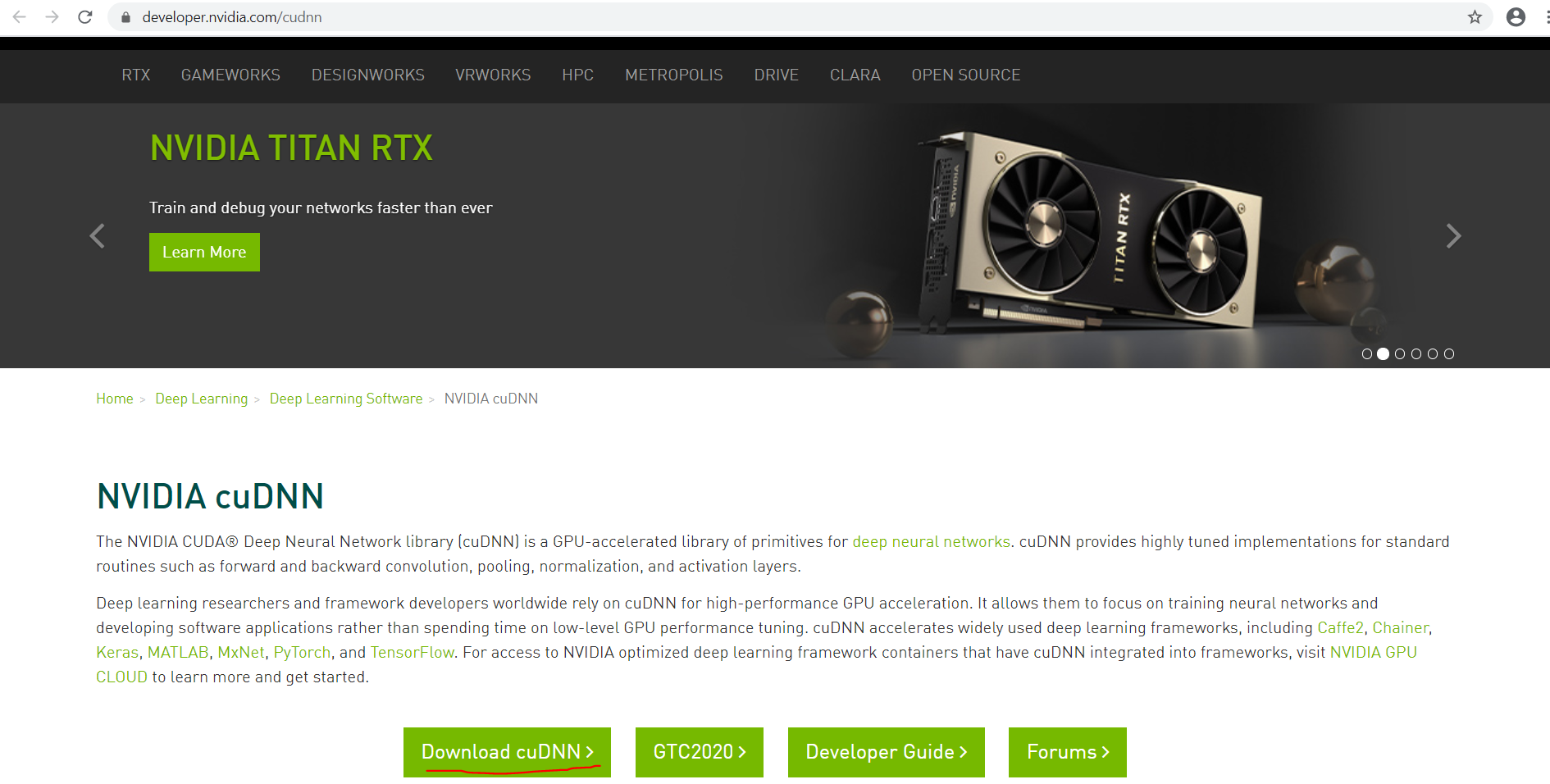
After downloaded the file look like below and the file size comes around 2.0 GB

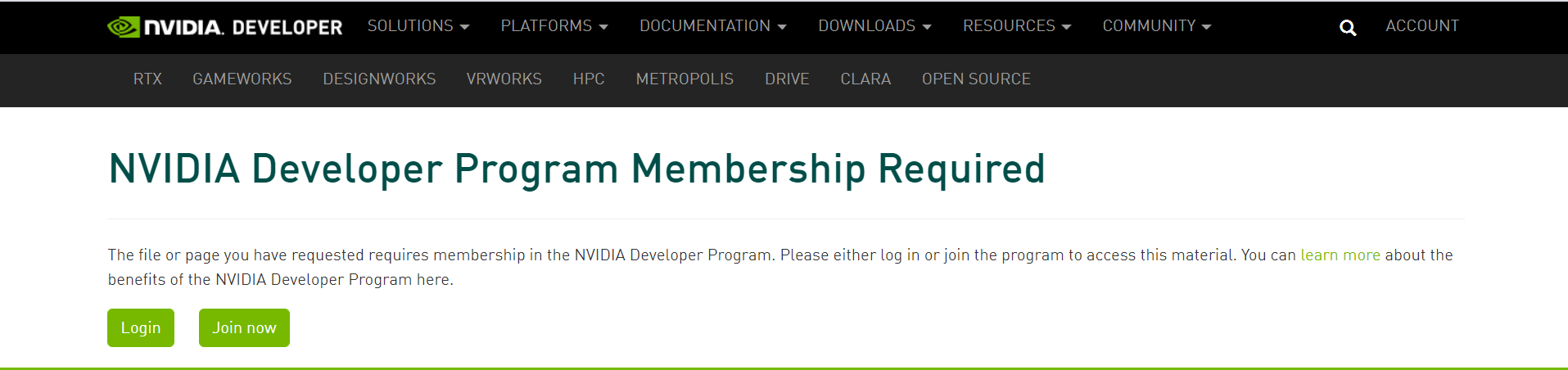


Install the Cuda toolkit.

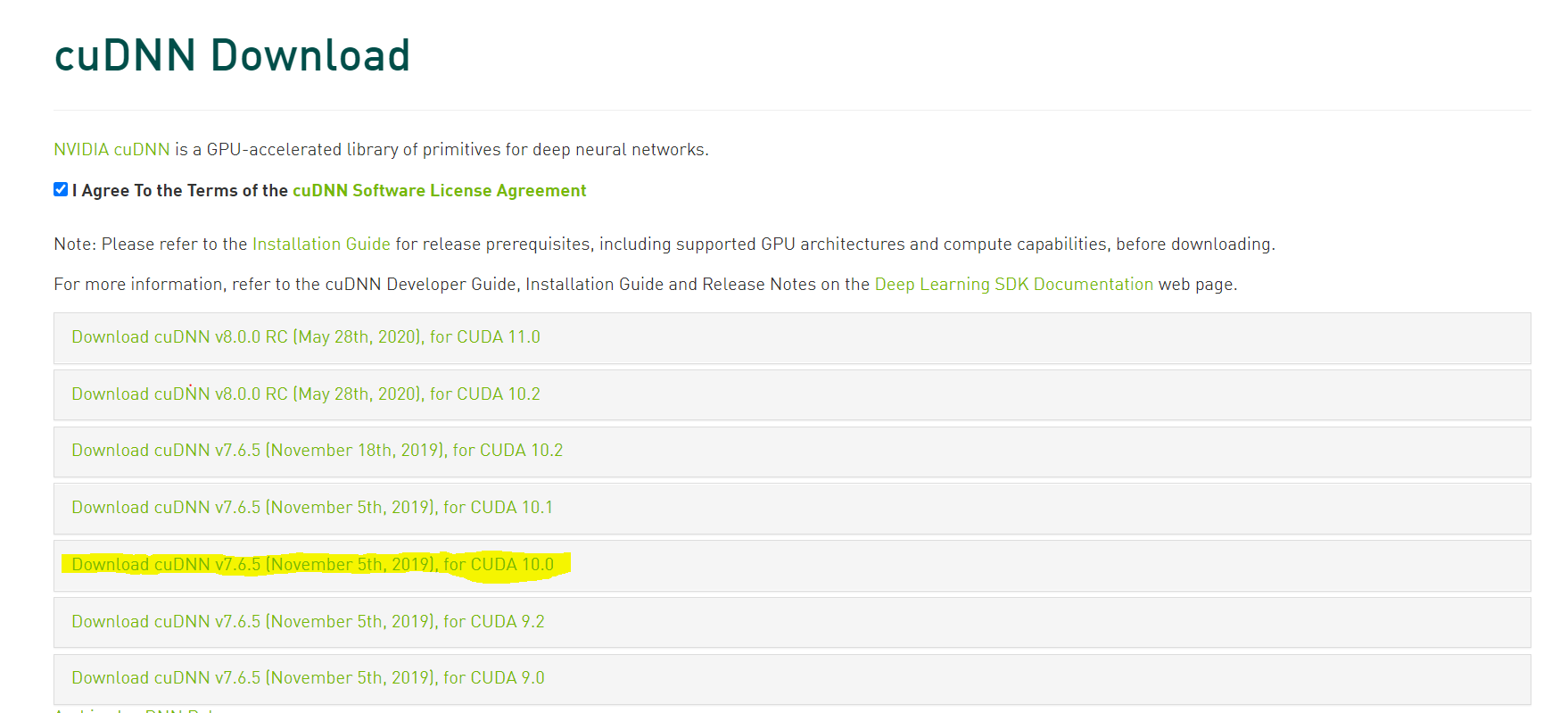
Step 2:

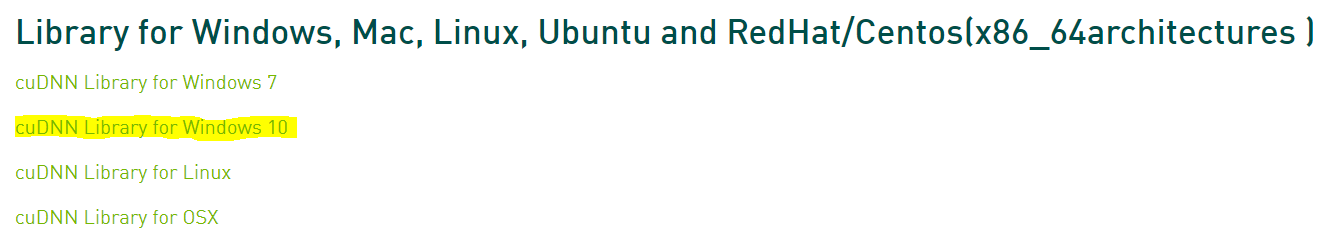
Next we need to install Nvidia Cuda Deep neural network library (NVIDIA cuDNN)





If already an existing user login with your credentials and download the package. If you are a new user create a new account then verify the account and then download the package.

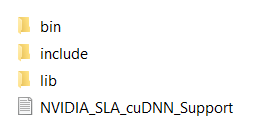




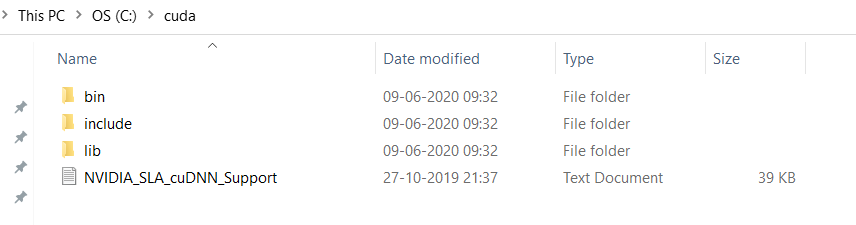
After the OS version which you have chosen and the download automatically starts and the downloaded file look like below and the size of the file around 286MB



Now extract the file and you will be able to see the below mentioned files inside the extracted cuDNN zip file.



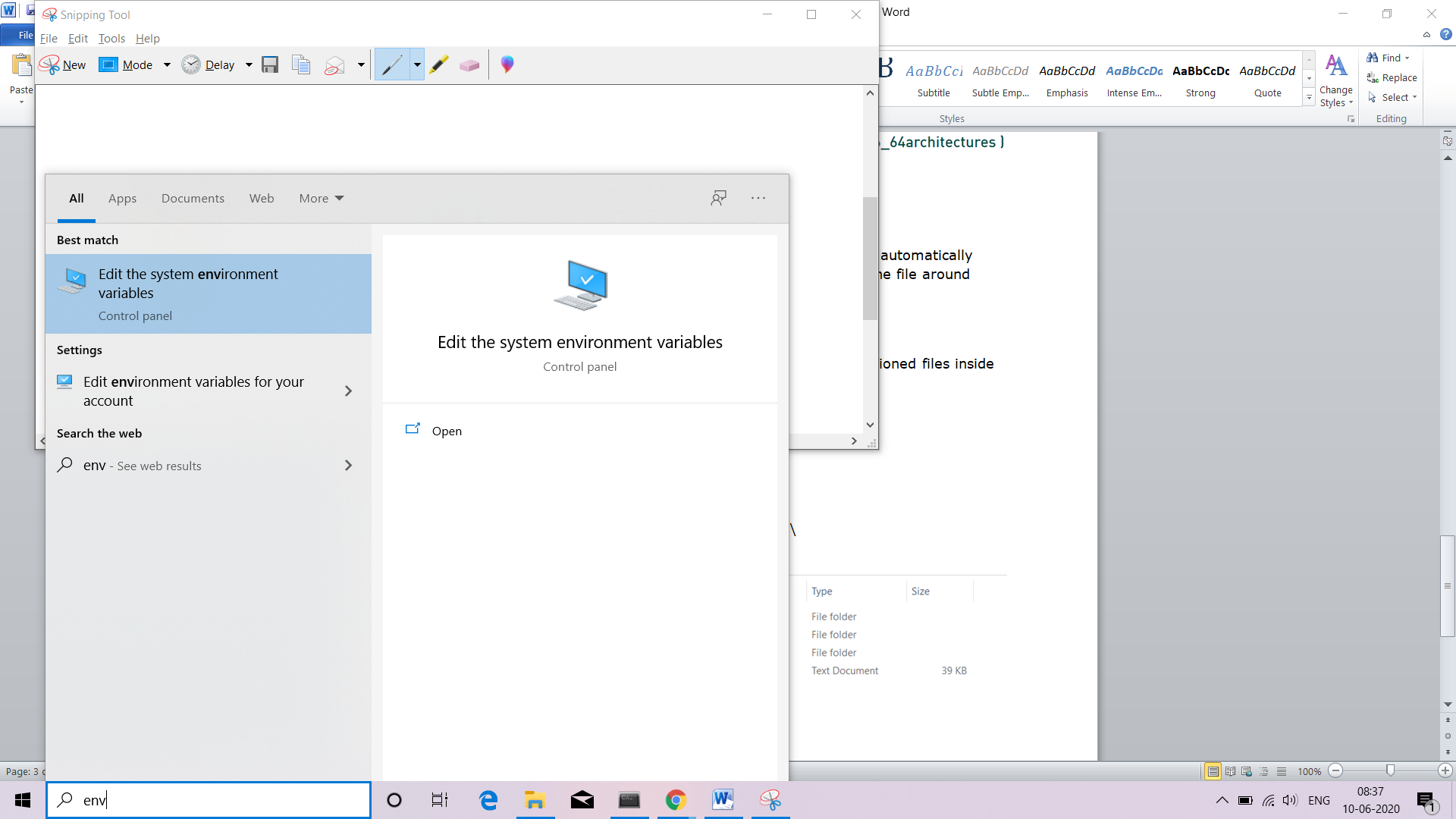
Now copy the three folders and paste it into the c:\



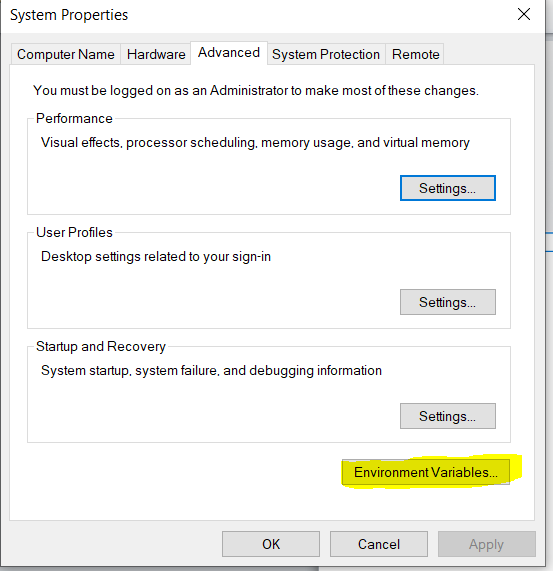
Step 3:

Setup the path variable

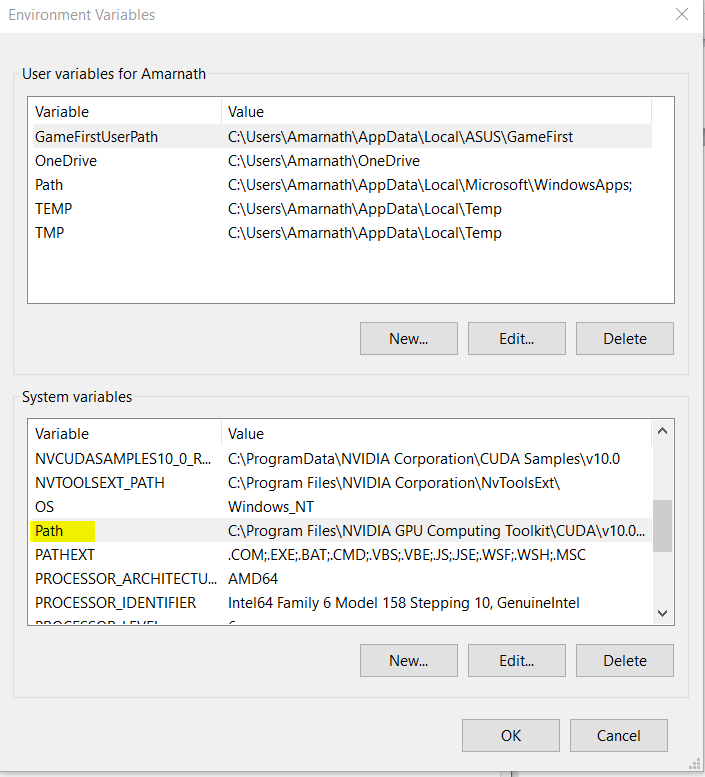
Now go to the environment variable



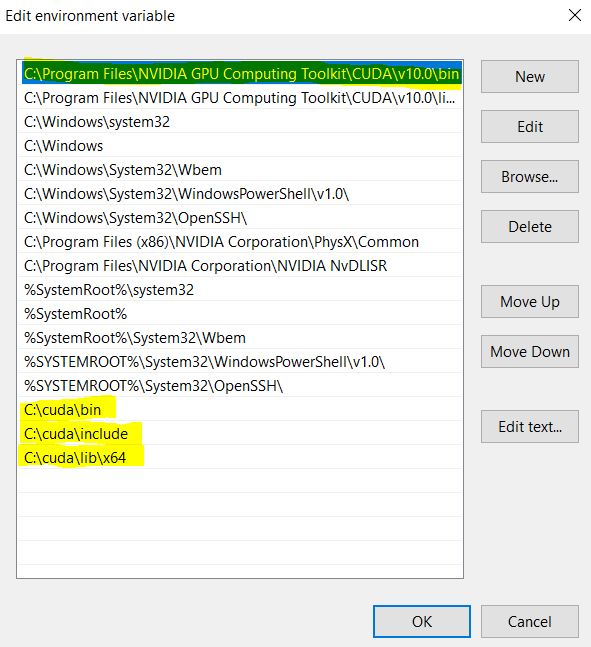
Then click Environment variable



Under system variables double click the path which displayed as below.



Make sure the NVIDIA GPU toolkit automatically updated in the environment variable and we need to add manually the three cuDNN library (Bin, Include, lib x64 path) into the environment variable



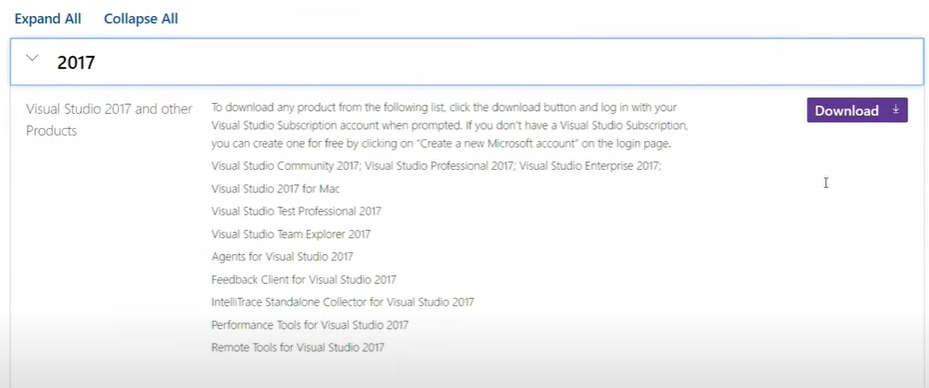
Then click ok after adding the cuDNN libraries

Step 4:

Install Visual studio community version.

To take all the c++ libraries

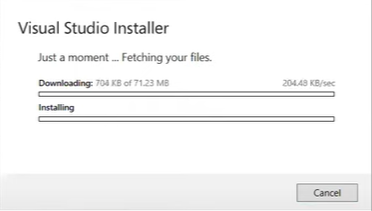




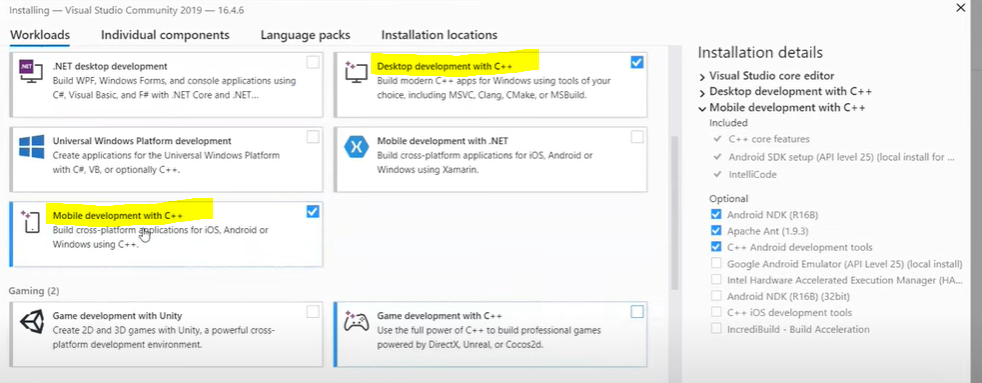
Then file looks like below



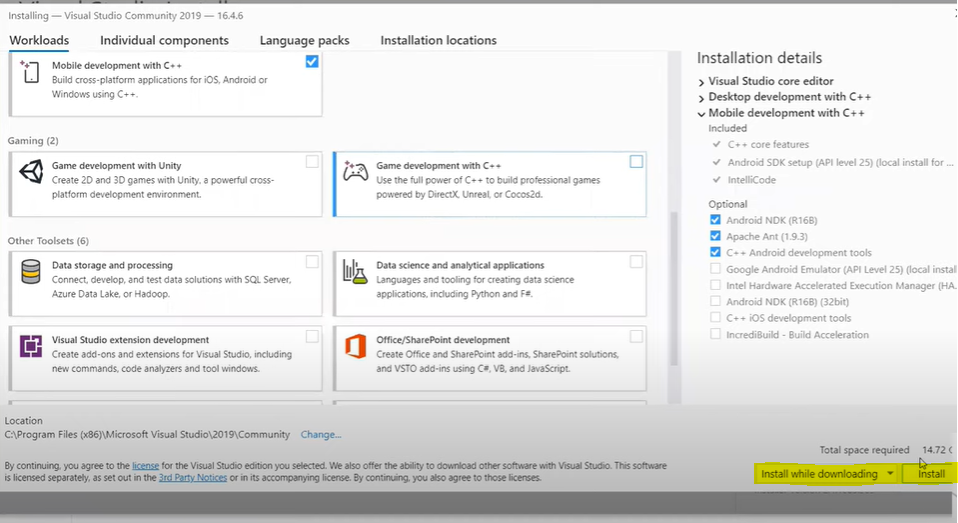
Now install



After some time then you will be able to see the screen like below and make sure our goal is to install the c++ libraries, so we need to select the desktop development c++ and mobile development with c++



To install this it requires around 14 GB space to install



Then click the desire option to install.

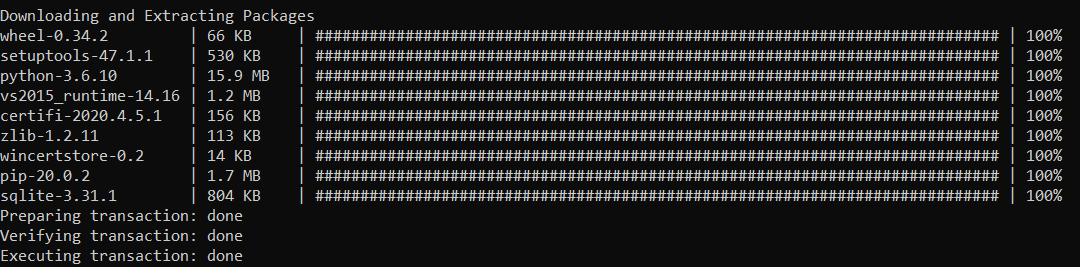
Step 5:

Create a new environment variable

Open Anaconda prompt



Make sure the Python version should be only 3.6



Environment created

Now we need to activate the environment

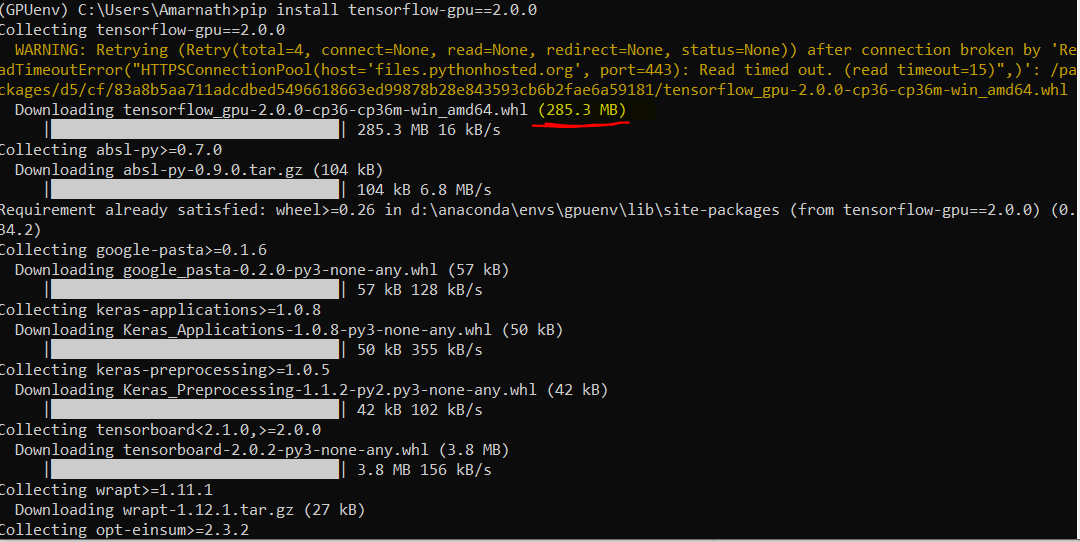


After activating the environment then we need to install the tensorflow with GPU in our newly created environment

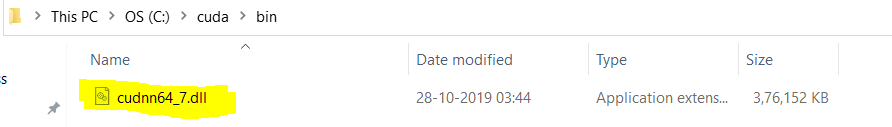


Here I have installed the tensorflow GPU for the version 2.0 if you want to install tensorflow for gpu version 1.15





After successful installation of Tensorflow with GPU in our environment then we need to check whether it is successfully installed or not. In order to do that open python in the command prompt and then type import tensorflow

Then you will be see this dll file. This is the dll file which we try to run inside the c:\ drive -- > bin which helps us to communicate with GPU cuda. 

How to check whether our GPU is initialized or not in our system

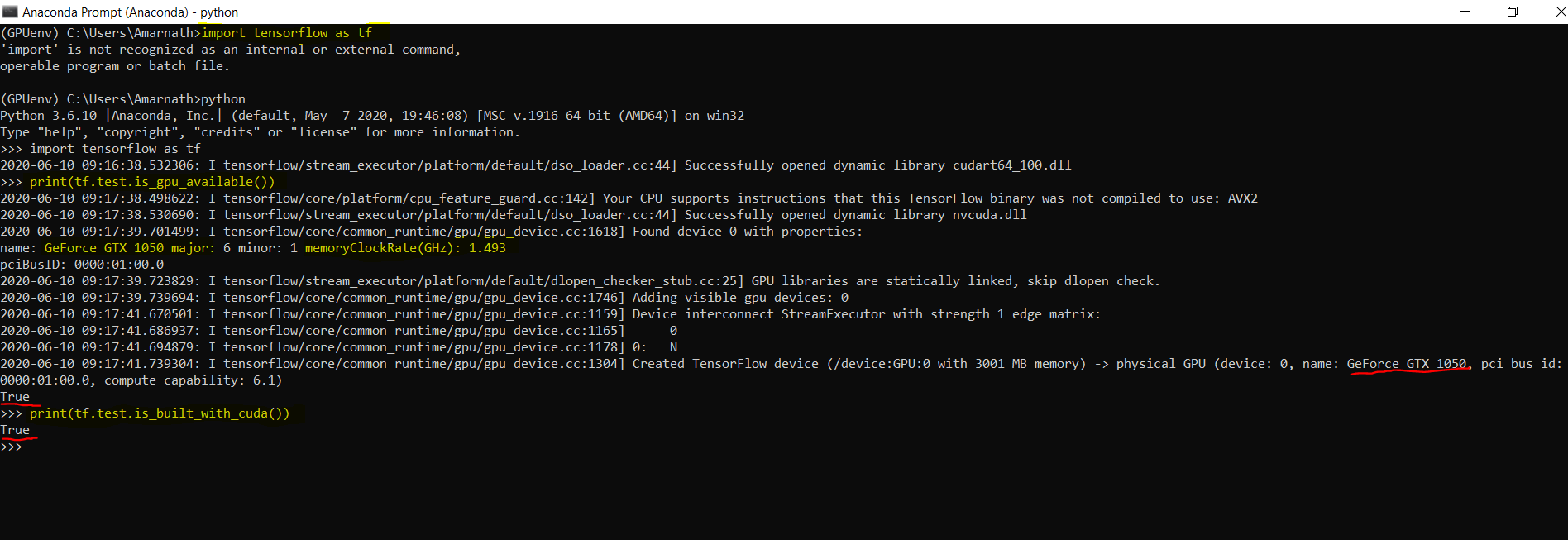
In order to do

Try to execute below 3 commands

1. import tensorflow as tf

2. print(tf.test.is\_gpu\_available())

3. print(tf.test.is\_built\_with\_cuda())



Cuda Toolkit: [https://developer.nvidia.com/cuda-10....](https://www.youtube.com/redirect?event=video_description&v=StH5YNrY0mE&q=https%3A%2F%2Fdeveloper.nvidia.com%2Fcuda-10.0-download-archive&redir_token=3HOqOKNcYerTmA42erGDWp3IDcR8MTU5MTg0ODEzOEAxNTkxNzYxNzM4)

cuDnn: [https://developer.nvidia.com/rdp/cudn...](https://www.youtube.com/redirect?event=video_description&v=StH5YNrY0mE&q=https%3A%2F%2Fdeveloper.nvidia.com%2Frdp%2Fcudnn-download&redir_token=3HOqOKNcYerTmA42erGDWp3IDcR8MTU5MTg0ODEzOEAxNTkxNzYxNzM4)