# Tensorflow GPU

Tensorflow GPU is a variant of the tensorflow library that makes use of available Nvidia GPU hardware to greatly increase the performance capabilities of the models. The gains to performance range between 5x to 15x depending on available hardware.

Follow these steps to setup tensorflow GPU:

## Setup Nvidia CUDA architecture:

This is the core component for tensorflow to utilize a machine's GPU. The primary dependency is that the device should have a GPU that is Nvidia CUDA compatible and has a compute capability of 3.0 or higher. The setup consists of 2 primary components.

1. CUDA® Toolkit 9.0

To setup the CUDA Toolkit, download the installer from this link.

Run the executable once downloaded and follow the instructions for an 'express installation'. Once successfully completed, restart your computer.

2. cuDNN v7.0

To download cuDNN, access this link.

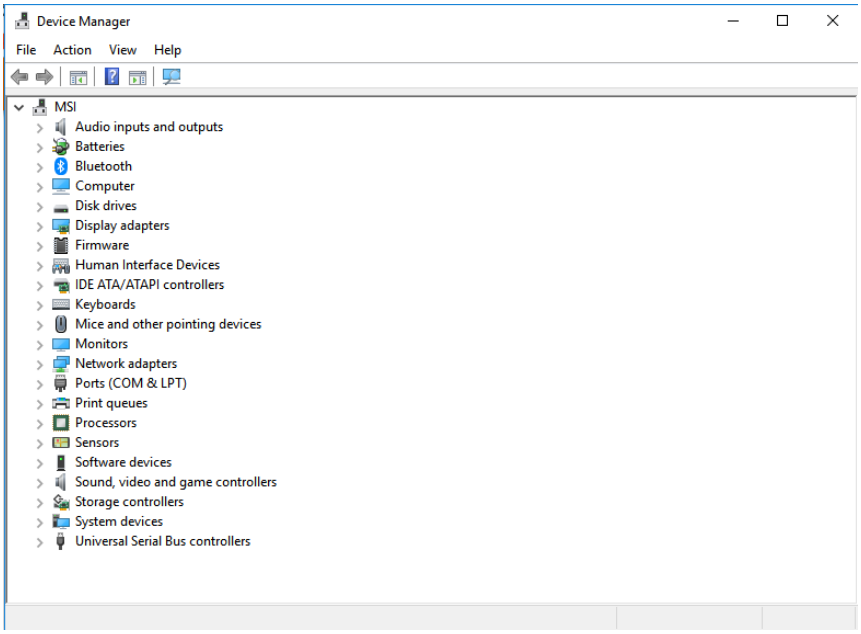
To be able to download the files, you will need to sign up for a developer account on Nvidia's website here.

Once the download is complete, extract the compressed file into the following directory

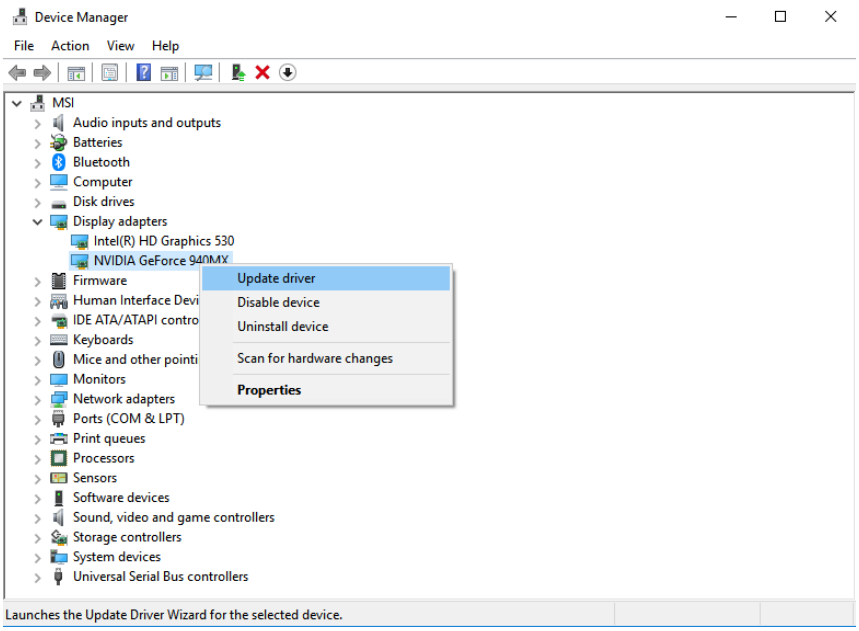
```
C:\cuda
```

## Update GPU Drivers

To perform this step open the start menu and type "Device Manager". Open the device manager and you will see the following window.




Expand the display adapter and right click on the Nvidia GPU. Then click on “Update Driver”.




Follow the instructions to auto update the driver for your GPU.

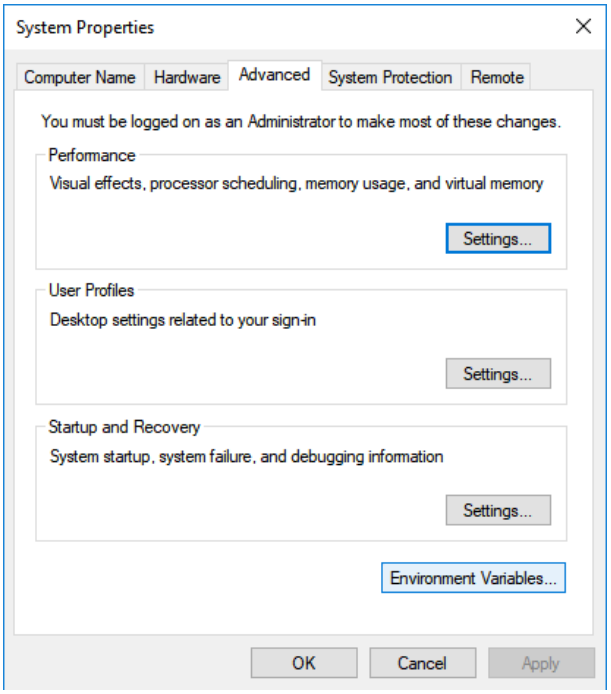
## Update Environment Variables

This is a critical step for setup of Tensorflow. A few variables need to be added to the global ‘PATH’ variable.

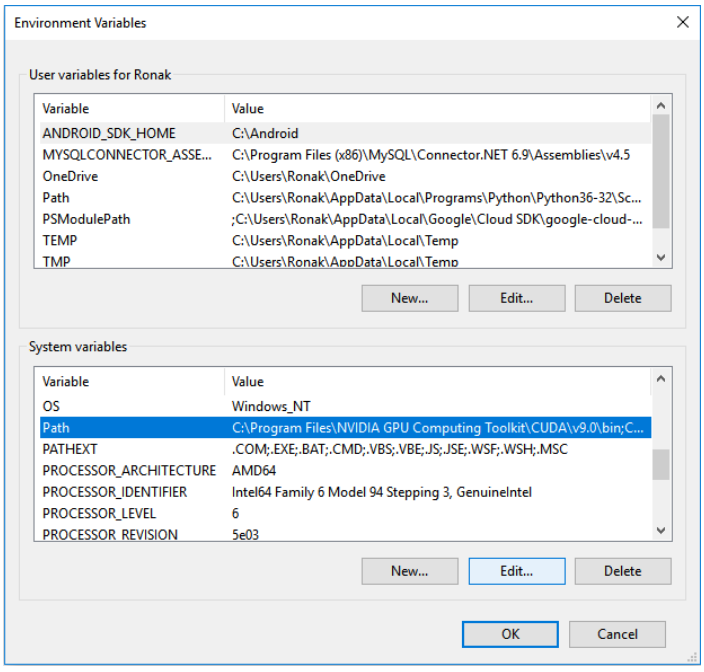
To do this, open the start menu and type “environment variables”.

Select “Edit the system environment variables”.

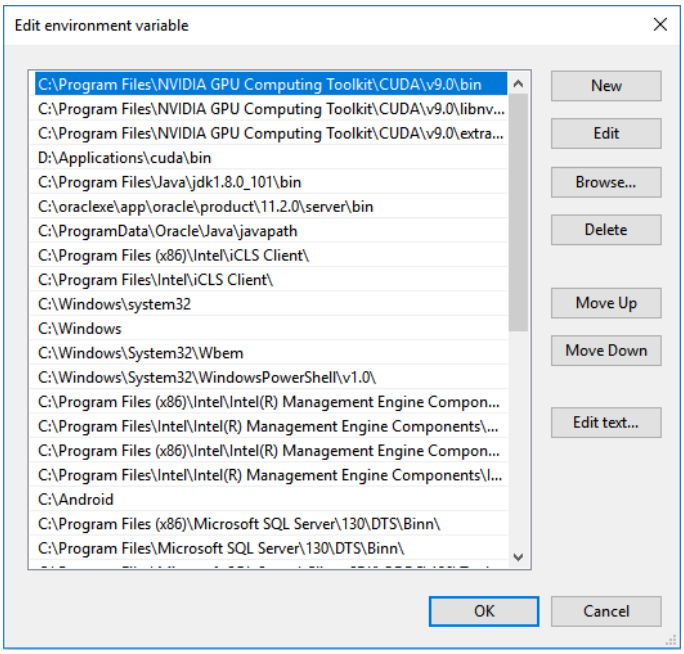
The following window appears.




Click on “Environment Variables” and a new window will appear.




Select the ‘Path’ variable from the System variables box and then click on “Edit”.



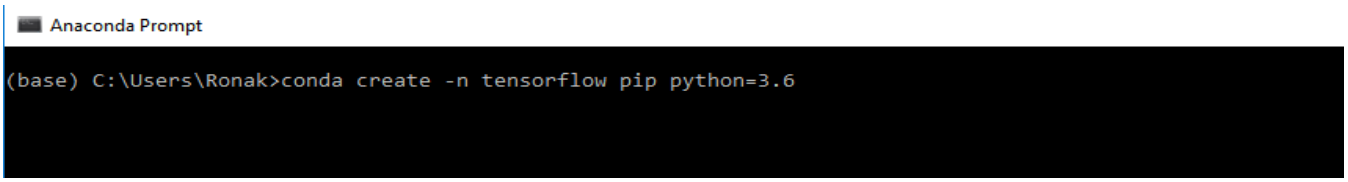

Once in the edit window, click on New and add the following 4 values one by one.

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.0\bin
C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.0\libnvvp
C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.0\extras\CUPTI\libx64
C:\cuda\bin

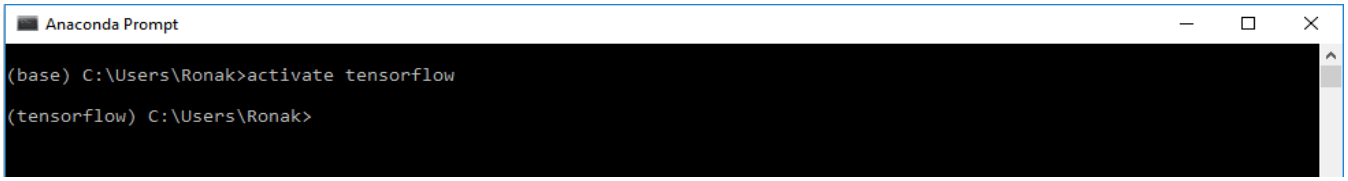
Click on Ok to save the environment variables and restart your computer.

## Setup Anaconda Environment

1. Create a virtual environment on Anaconda




2. Activate this environment to access it



## Install Libraries

While this is a straightforward step; the instructions are being detailed due to a conflict with the Keras library being used by this project.

Within the 'tensorflow' environment created in the previous step, install the following libraries using the commands listed.

```
conda install numpy
conda install pandas
conda install matplotlib
conda install sklearn
conda install keras
```

The 'Keras' library installs tensorflow as one of its dependencies. We do not need this version of tensorflow as it will conflict with the GPU configuration. To avoid this, use the following command to remove the tensorflow library.

```
pip uninstall tensorflow
```

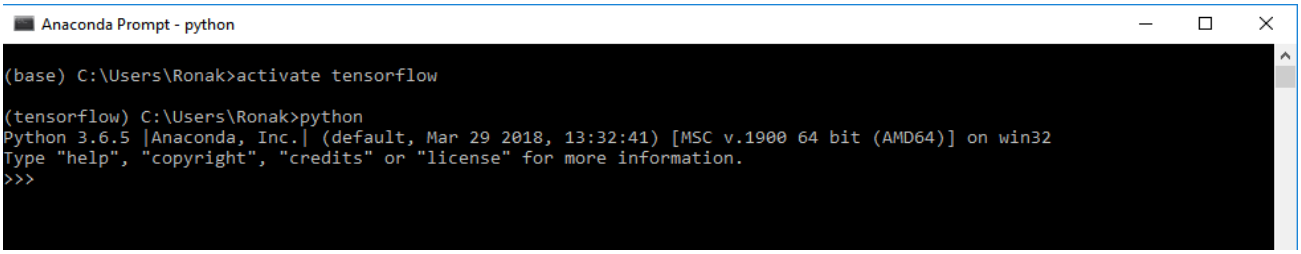
**Note:** Do not use 'conda' to uninstall tensorflow as it will uninstall keras as well.

Once the tensorflow library is uninstalled, use the following command to setup the tensorflow GPU version.

```
pip install --ignore-installed --upgrade tensorflow-gpu
```

Once the installation is complete, use the following commands to verify the complete configuration.

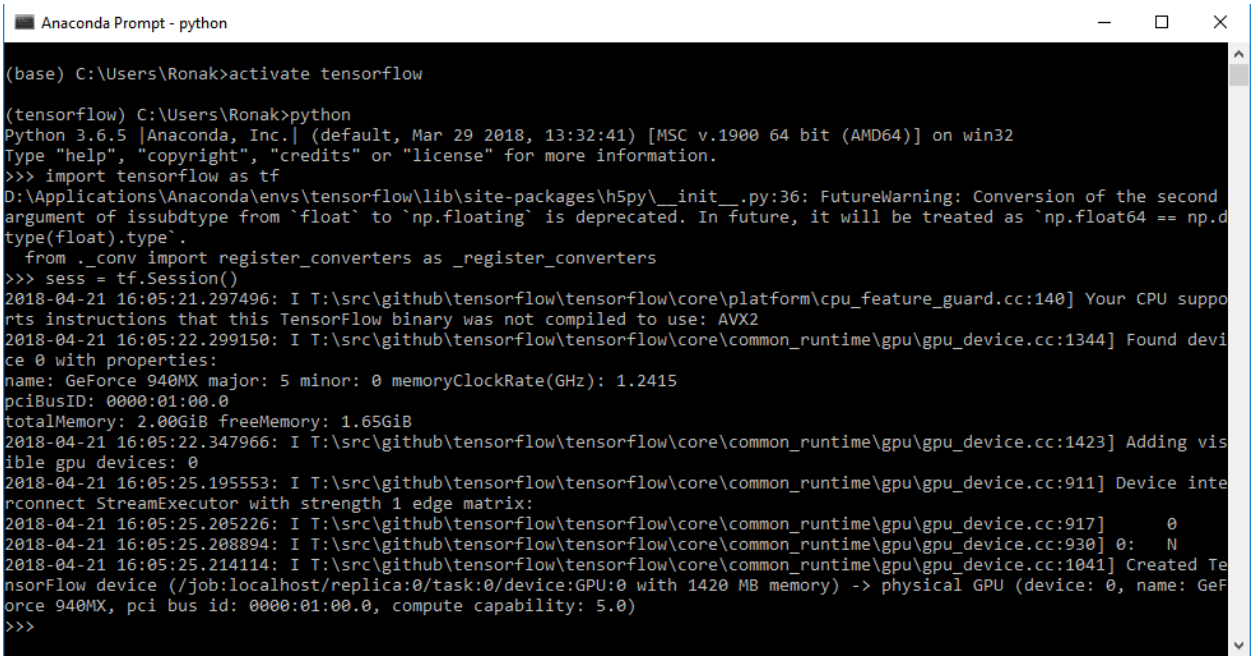
Access the python command line within anaconda.




Type in the following lines of code.

```
import tensorflow as tf
sess = tf.Session()
```

The commands will return information about the GPU available and its various parameters like memory and clock speed.




This means that the setup was successful and configuration is now complete.