Using Anaconda and the Jupyter Notebook

netLabs!UG

April 8, 2019

Contents

1 Getting started with Anaconda

The Anaconda Python/R distribution comes installed with over 1500+ data science and machine learning packages.[1] The environment has two access options, through the graphical user interface (GUI) or through the terminal.

1.1 Accessing the Environment - Windows (GUI)

1. Open the start menu and type `Anaconda Navigator`.
2. Once the Navigator is open, hover to the `Applications on` tab to select the environment, the default environment is `base(root)`.

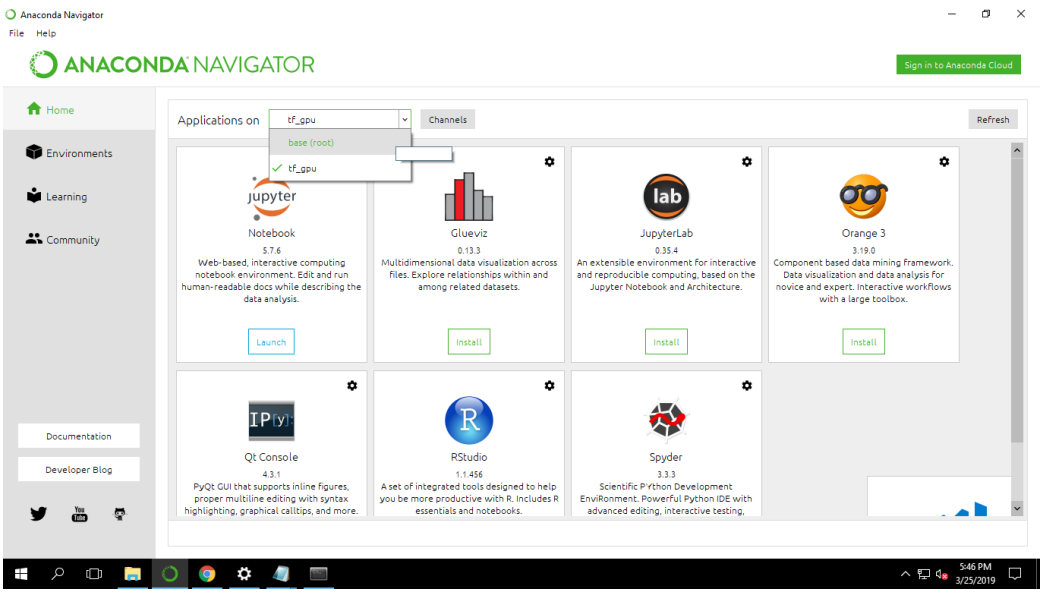

Figure 1: Selecting an environment to use.

3. Select `Environments` from the left side panel, to view the packages installed on the selected environment.

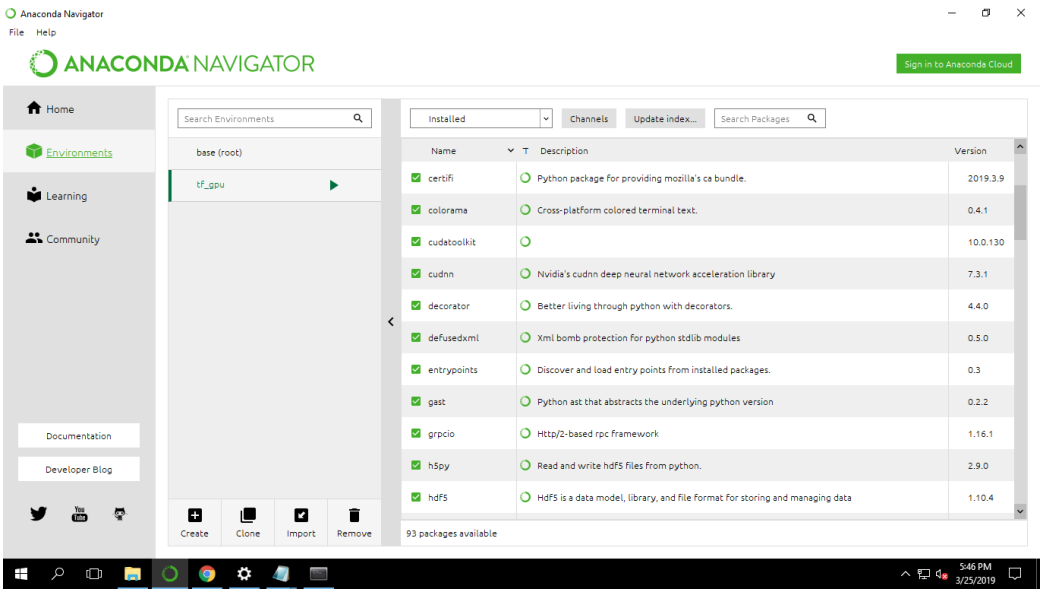

Figure 2: Packages installed on `tf_gpu` environment.

4. To use any applications associated with a selected environment, select home then click the launch button of the application.

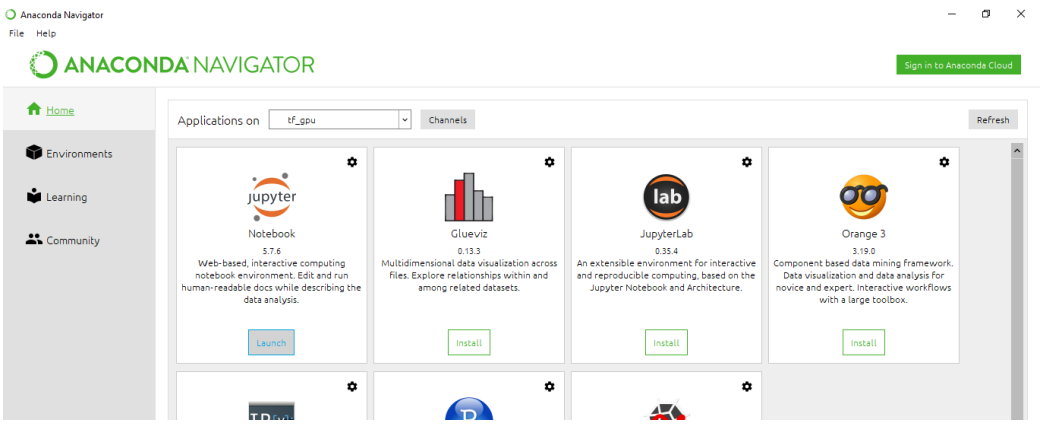

Figure 3: Launching the Jupyter Notebook application in the `tf_gpu` environment.

1.2 Accessing the Environment - Windows (Command Prompt)

1. Open the start menu and type `Anaconda Prompt`.

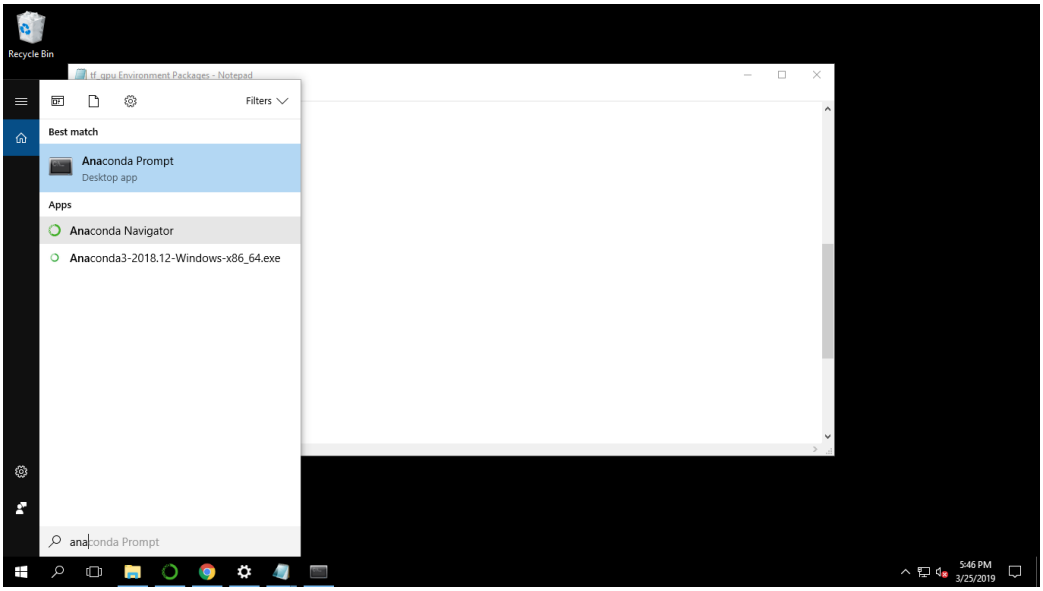

Figure 4: Accessing the environment through command prompt.

2. Once the command prompt is open, type `Anaconda-Navigator`, to open the Anaconda Navigator.

3. To open the Jupyter notebook, type `jupyter notebook`.

1.3 Accessing the Environment - Linux

1. Open the terminal and enter the following command.

```
$ anaconda-navigator
```

2. Select the launch option in the box with Jupyter Notebook.

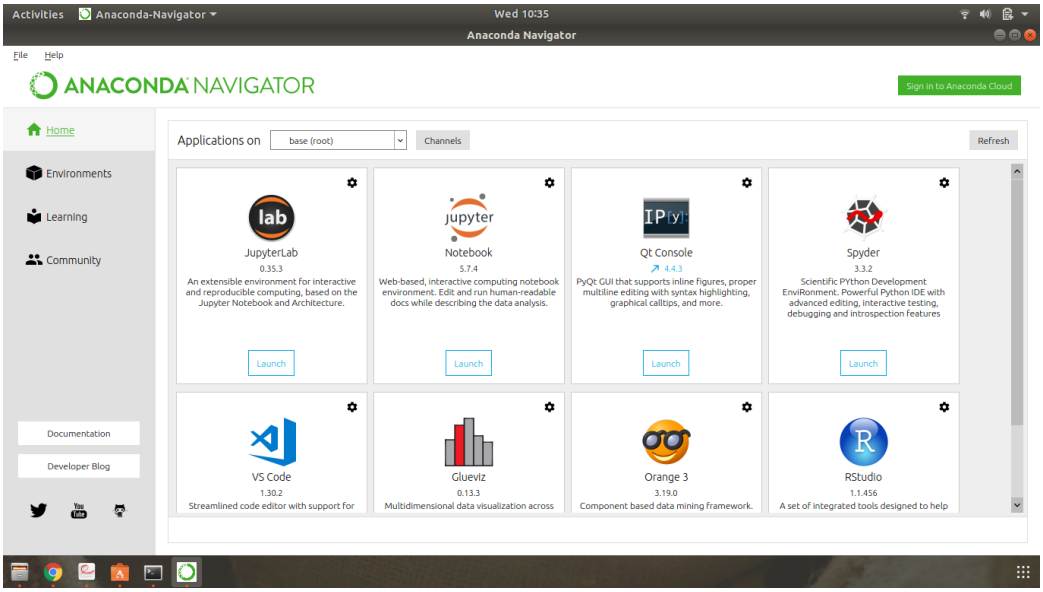

Figure 5: This command opens the navigation window as shown above.

Figure 6: The home page of the Jupyter Notebook showing the document directory tree.

2 Creating and using virtual environments

1. Open the terminal or `Anaconda Prompt` and enter the following command.[1]

   ```
   $ conda create --name your_env_name python=python_version -y
   ```

   python_version allows the selection of the python version to use for the particular environment.

2. The created environment can be accessed using the following command.

   ```
   $ conda activate your_env_name
   ```

3. To install packages within a given environment, anaconda offers two options. The command used depends on where the repository is located.

   ```
   $ conda install package_name
   ```
   or

   ```
   $ pip install package_name
   ```

4. To access the juypter notebook with support for the installed packages within a given environment. Ensure you are in the right environment and run the following command.

   ```
   $ jupyter notebook
   ```

5. To close a a virtual environment, type the following command

   ```
   $ conda deactivate
   ```

2.1 Commands for managing the environment

1. To list all the existing environments, type the following command;

   ```
   $ conda info --envs
   ```

2. To remove/delete a given environment, run the following command;

   ```
   $ conda remove --name your_env_name --all
   ```

   The *–all* flag is to remove all packages with in that environment.

[1] Creating an environment and installing the different pakages requires an internet connection preferably without firewalls

3. To see a list of all installed packages, enter the following command.
   `$ conda list`

4. To find out about using the environment, the following command is very helpful;

   `$ conda -h`

   The *-h* also *–help* means help, the command displays an exhaustive list of all the possible that can be used.

3 Using the Jupyter Notebook

The Jupyter Notebook can be accessed through the Graphical User Interface (GUI) or the terminal (`Anaconda Prompt`) as shown in the steps above.

3.1 Creating a notebook

1. To create a notebook, go to the New Tab, select Python 3.

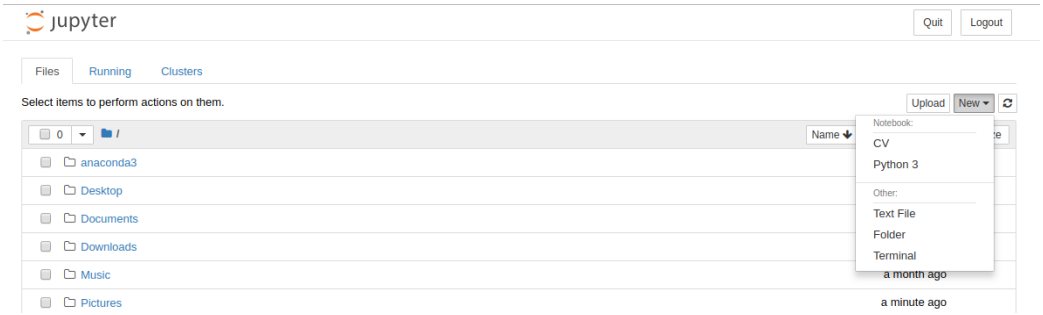

Figure 7: Creating a new notebook.

2. A new tab opens automatically with an active cell, enter your code.

```
print(''netLabs!UG'')
```

3. Press `Shift + Enter` to run the cell.

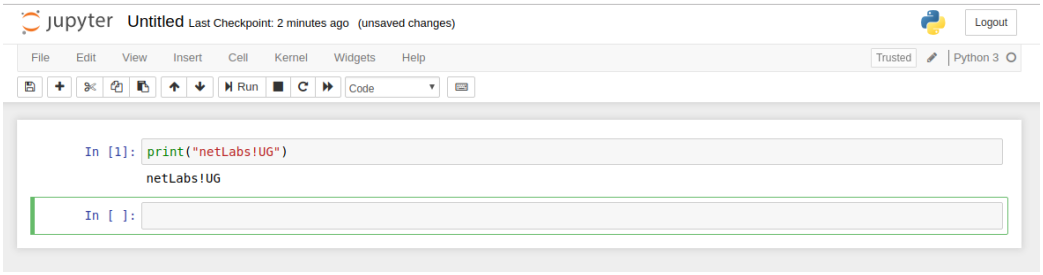

Figure 8: Running a cell.

4 Extras

4.1 Using TensorFlow

1. Launch the Jupyter Notebook in the `tf_gpu` environment.
2. Open a new notebook.
3. Enter your code.

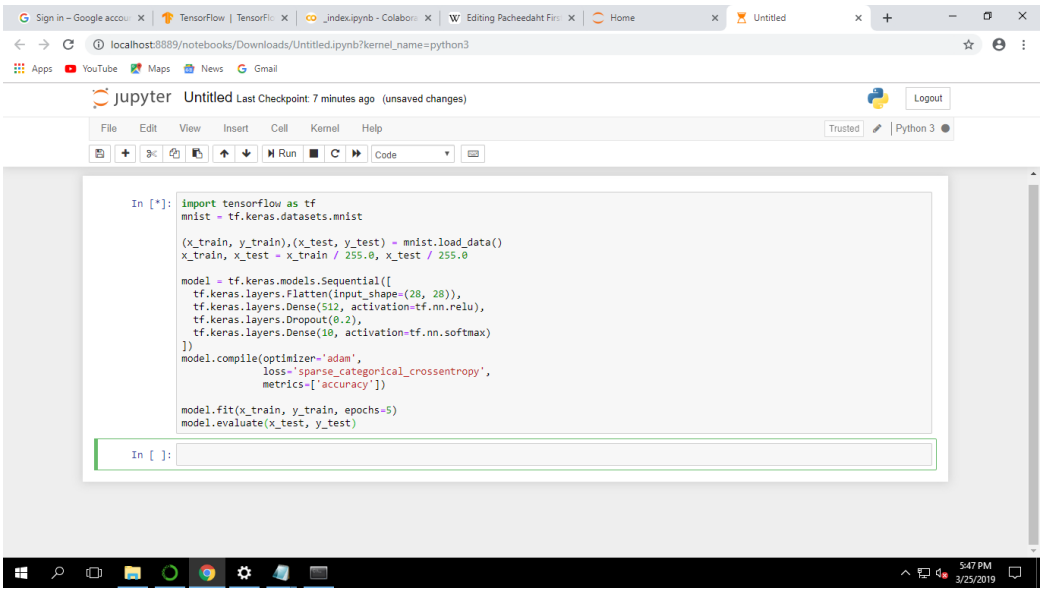

Figure 9: Tensorflow test code. [2]

4. Run the code using `Shift + Enter` or select run from the upper tab.

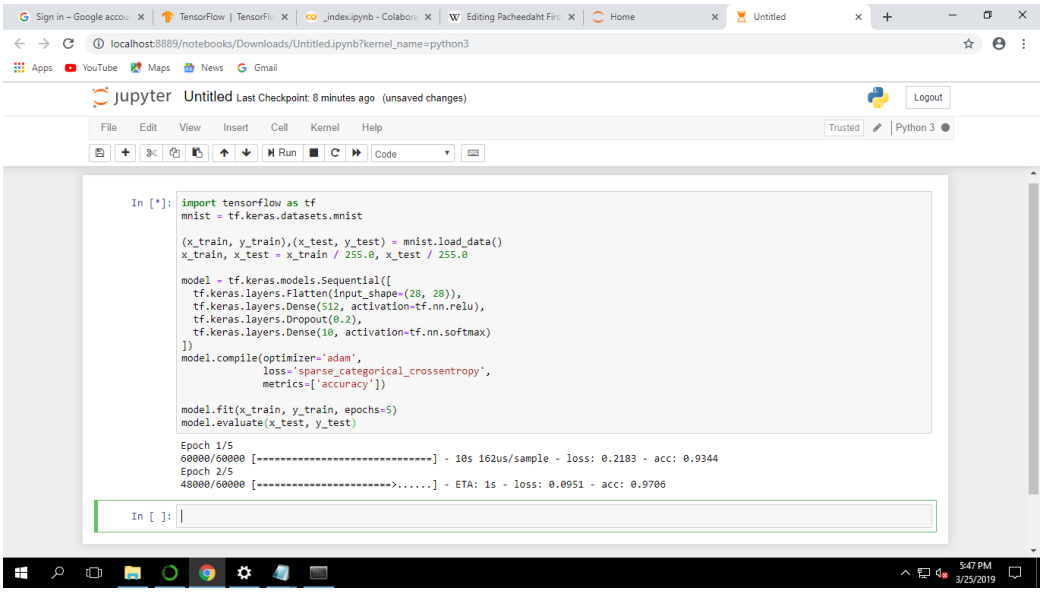

Figure 10: Running the entered code.

4.2 Directory Structure

1. To ensure uniformity, users should save their work in the `Students Projects` folder under the documents directory.

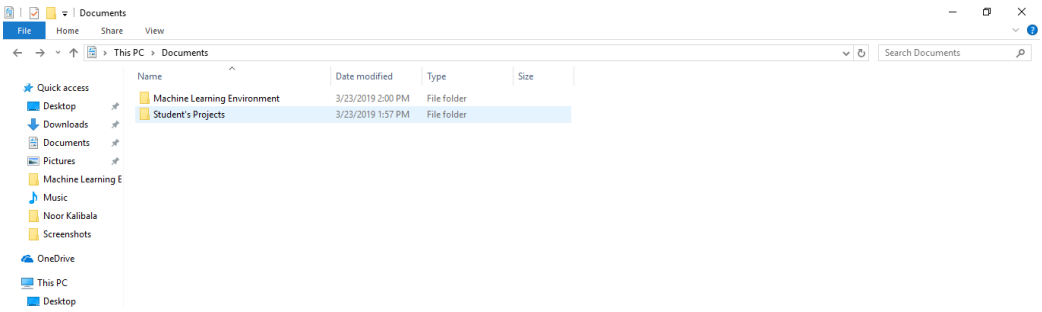

Figure 11: The documents directory folder.

2. Users are encouraged to label their folders in the following format, `users_name_{reg_no}`.

3. The `Machine Learning Environments` folder contains the list of packages installed on existing environments.

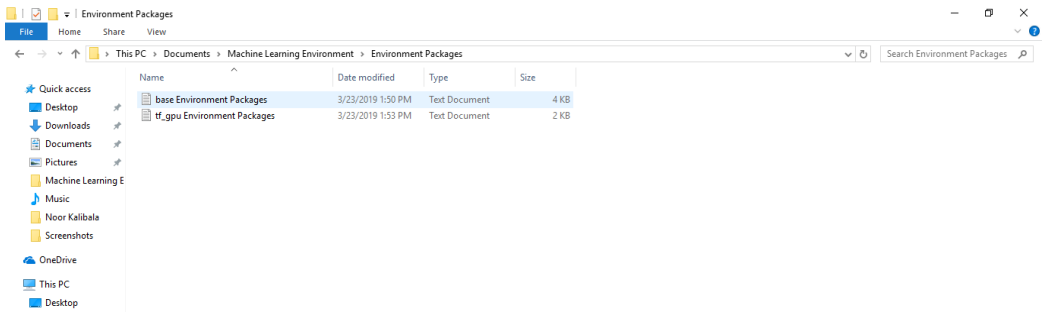

Figure 12: The lists containing the packages installed on the each environment.

5 Further Reading

Anaconda User Guide
`http://docs.anaconda.com/anaconda-cloud/user-guide/`
Official User Guide for the Anaconda environment, has links to resources on how to install, update, use and maintain the environment.

Jupyter Notebook Documentation
`https://jupyter-notebook.readthedocs.io/en/stable/`
This documentation covers a wide range of topics including but not limited to starting a Jupyter Notebook, creating notebooks, sharing notebooks among others topics.

Deep Learning Indaba Tutorials
`http://www.deeplearningindaba.com/videos.html`
Contains video and text tutorials on machine learning and deep learning including intuitive material on the mathematics behind the technologies.

Tensorflow Tutorials
`https://www.tensorflow.org/tutorials`
These tutorials cover most of the fundamentals of using TensorFlow, deep learning and machine learning in general.

Python Virtual Environments
`https://realpython.com/python-virtual-environments-a-primer/`
Delves into what virtual environments are, why they are needed and more.

References

[1] Anaconda Contributors. Anaconda Python/R Distribution - Anaconda. `https://www.anaconda.com/distribution/`.

[2] TensorFlow Contributors. Tensorflow Core — Tensorflow. `https://www.tensorflow.org/tutorials`.