Installation steps for tensorflow-GPU based

This installation steps refer to the Ubuntu x64 based version of the software, running on a CUDA Nvidia GPU. Everything was done according to the guide on the official site: https://www.tensorflow.org/install/install_linux#common_installation_problems

I . Setting up python with anaconda on a fresh ubuntu install

II. Installing all the NVIDIA stuff for GPU tensorflow

III. Installing tensorflow with anaconda

IV. Run programs

V. Possible problems

VI. Resources on ML/Deep learning/TF

I . Setting up python with anaconda on a fresh ubuntu install

Q: Why anaconda?
A: You already have the basic scientific packages for python like Numpy integrated. Plus, the name is cool.

Now, onto the installation steps:

1. curl -O https://repo.continuum.io/archive/Anaconda3-5.0.1-Linux-x86_64.sh into a folder of your choice
2. bash Anaconda3-5.0.1-Linux-x86_64.sh
3. Agree with license
4. Agree with env path updates
5. source ~/.bashrc
6. Check the install with conda command
7. You can now create environments for the installation

Q: What are environments?
A:https://realpython.com/blog/python/python-virtual-environments-a-primer/
If you have troubles with this, check https://www.digitalocean.com/community/tutorials/how-to-install-the-anaconda-python-distribution-on-ubuntu-16-04

II. Installing all the NVIDIA stuff for GPU tensorflow

First, to avoid wasting time: tensorflow says on its site that it supports CUDA toolkit 9.0. The link they provide redirects to toolkit 9.1, which doesn’t work yet as it should. Don’t download from that link, but search manually where to get toolkit 9.0 or access https://developer.nvidia.com/cuda-90-download-archive :)

Also, I went for the runfile installation since getting it from the repo automatically takes the most recent version, 9.1, which we don’t want.

Now, installation steps:

1. sudo sh cuda_9.0.176_384.81_linux.run
2. Follow instructions
3. If the driver is not installed, that’s because you already have on running and need to disable it. If it’s version 384+ you should be fine, else see chapter V for extra info

For CUDNN library, after creating an account on https://developer.nvidia.com/cudnn and downloading your desired version

1. sudo dpkg -i libcudnn7_7.0.3.11-1+cuda9.0_amd64.deb
2. sudo dpkg -i libcudnn7-dev_7.0.3.11-1+cuda9.0_amd64.deb
3. sudo dpkg -i libcudnn7-doc_7.0.3.11-1+cuda9.0_amd64.deb

For libcupti-dev library, just do sudo apt-get install libcupti-dev and you are good to go.
For more:
http://developer.download.nvidia.com/compute/cuda/9.0/Prod/docs/sidebar/CUDA_Quick_Start_Guide.pdf
http://docs.nvidia.com/deeplearning/sdk/cudnn-install/index.html#install-linux

III. Installing tensorflow with anaconda

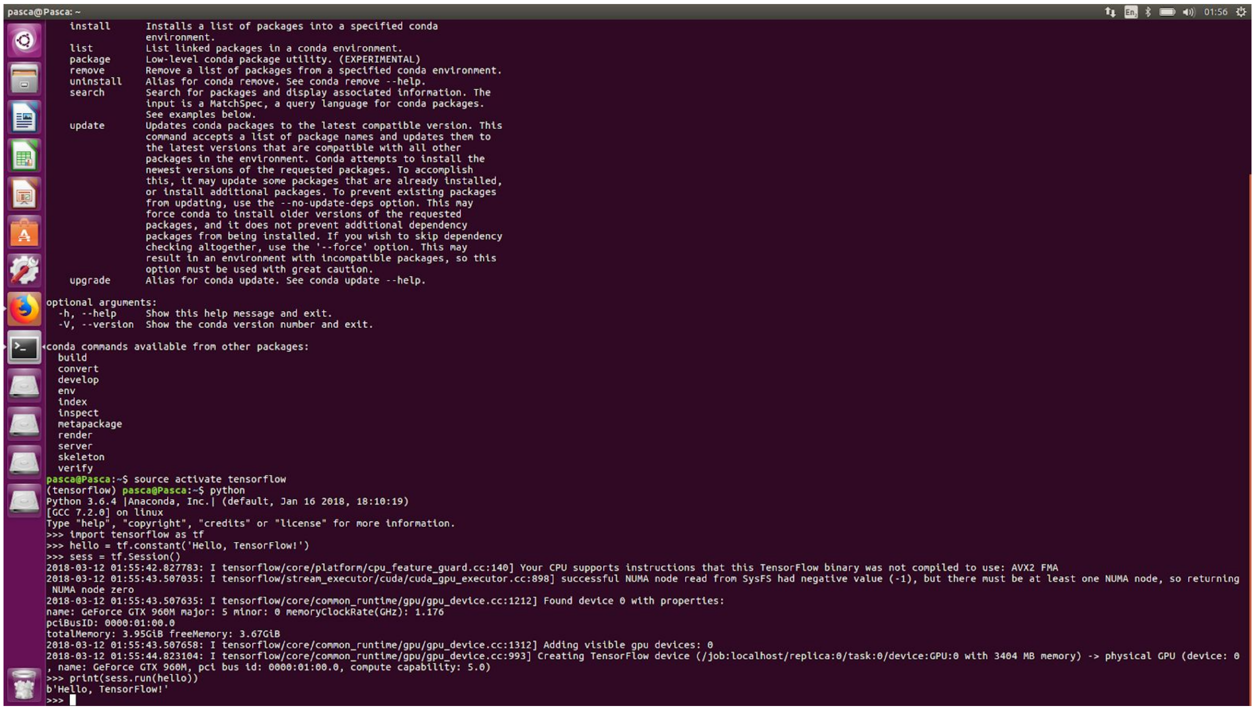
1. conda create -n tensorflow pip python=2.7 # or python=3.3, etc.
2. source activate tensorflow #activates the environment to install tensorflow which will be called tensorflow
3. pip install --ignore-installed --upgrade *tfBinaryURL,* taken from https://www.tensorflow.org/install/install_linux#the_url_of_the_tensorflow_python_package

IV. Run programs

1. Python
2. Run a short program
3. Learn ML
4. Learn Tensorflow
5. ??
6. Make $$/Look smart

V. Possible problems

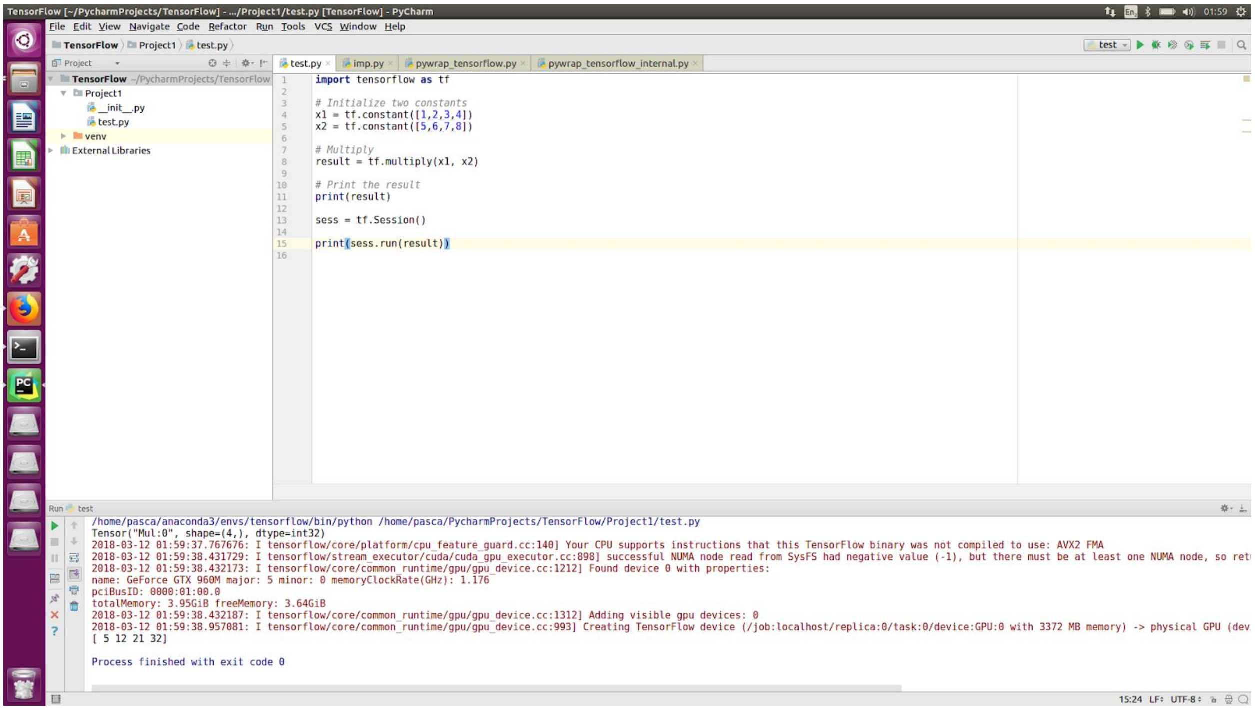
1. Have incompatible driver version with the NVIDIA toolkit -> get appropriate driver version according to toolkit documentation
2. Forget about updating LD paths for NVIDIA toolkit -> follow the steps exactly as in the toolkit documentation, post install steps
3. Can’t run TF from Pycharm -> make sure you use anaconda environment set up before(e.g. “Tensorflow”) and launch Pycharm from the bash because it doesn’t recognize env variables
4. Screen goes black after driver installation -> incompatible driver version, go text mode runlevel for the Linux, purge nvidia drivers and try an older one



```
    install      Installs a list of packages into a specified conda
                 environment.
    list         List linked packages in a conda environment.
    package      Low-level conda package utility. (EXPERIMENTAL)
    remove       Remove a list of packages from a specified conda environment.
    uninstall    Alias for conda remove. See conda remove --help.
    search       Search for packages and display associated information. The
                 input is a MatchSpec, a query language for conda packages.
                 See examples below.
    update       Updates conda packages to the latest compatible version. This
                 command accepts a list of package names and updates them to
                 the latest versions that are compatible with all other
                 packages in the environment. Conda attempts to install the
                 newest versions of the requested packages. To accomplish
                 this, it may update some packages that are already installed,
                 or install additional packages. To prevent existing packages
                 from updating, use the --no-update-deps option. This may
                 force conda to install older versions of the requested
                 packages, and it does not prevent additional dependency
                 packages from being installed. If you wish to skip dependency
                 checking altogether, use the '--force' option. This may
                 result in an environment with incompatible packages, so this
                 option must be used with great caution.
    upgrade      Alias for conda update. See conda update --help.

optional arguments:
  -h, --help     Show this help message and exit.
  -V, --version  Show the conda version number and exit.

conda commands available from other packages:
  build
  convert
  develop
  env
  index
  inspect
  metapackage
  render
  server
  skeleton
  verify
pasca@Pasca:~$ source activate tensorflow
(tensorflow) pasca@Pasca:~$ python
Python 3.6.4 |Anaconda, Inc.| (default, Jan 16 2018, 18:10:19)
[GCC 7.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorflow as tf
>>> hello = tf.constant('Hello, TensorFlow!')
>>> sess = tf.Session()
2018-03-12 01:55:42.827783: I tensorflow/core/platform/cpu_feature_guard.cc:140] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 FMA
2018-03-12 01:55:43.507035: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:898] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning
 NUMA node zero
2018-03-12 01:55:43.507635: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1212] Found device 0 with properties:
name: GeForce GTX 960M major: 5 minor: 0 memoryClockRate(GHz): 1.176
pciBusID: 0000:01:00.0
totalMemory: 3.95GiB freeMemory: 3.67GiB
2018-03-12 01:55:43.507658: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1312] Adding visible gpu devices: 0
2018-03-12 01:55:44.823104: I tensorflow/core/common_runtime/gpu/gpu_device.cc:993] Creating TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 with 3404 MB memory) -> physical GPU (device: 0
, name: GeForce GTX 960M, pci bus id: 0000:01:00.0, compute capability: 5.0)
>>> print(sess.run(hello))
b'Hello, TensorFlow!'
>>>
```



```python
import tensorflow as tf

# Initialize two constants
x1 = tf.constant([1,2,3,4])
x2 = tf.constant([5,6,7,8])

# Multiply
result = tf.multiply(x1, x2)

# Print the result
print(result)

sess = tf.Session()

print(sess.run(result))
```

```
/home/pasca/anaconda3/envs/tensorflow/bin/python /home/pasca/PycharmProjects/TensorFlow/Project1/test.py
Tensor("Mul:0", shape=(4,), dtype=int32)
2018-03-12 01:59:37.767676: I tensorflow/core/platform/cpu_feature_guard.cc:140] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 FMA
2018-03-12 01:59:38.431729: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:898] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so ret
2018-03-12 01:59:38.432173: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1212] Found device 0 with properties:
name: GeForce GTX 960M major: 5 minor: 0 memoryClockRate(GHz): 1.176
pciBusID: 0000:01:00.0
totalMemory: 3.95GiB freeMemory: 3.64GiB
2018-03-12 01:59:38.432187: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1312] Adding visible gpu devices: 0
2018-03-12 01:59:38.957081: I tensorflow/core/common_runtime/gpu/gpu_device.cc:993] Creating TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 with 3372 MB memory) -> physical GPU (dev
[ 5 12 21 32]

Process finished with exit code 0
```

VI. Resources on ML/Deep learning/TF

1. https://machinelearningmastery.com/start-here/
2. https://www.datacamp.com/community/tutorials/tensorflow-tutorial
3. http://blog.kaggle.com/2017/11/27/introduction-to-neural-networks/
4. https://ai.google/education#?modal_active=none
5. https://developers.google.com/machine-learning/crash-course/prereqs-and-prework
6. Andrew NG's coursera courses