4.Install TensorFlow GPU

Before we install TensorFlow GPU, we need to complete configuration of CUDA.
About configuration of CUDA, please refer to 【1.Preparation tutorial】

1. Install pip

We need to input command:

sudo apt-get install python3-pip python3-dev

After installation, pip is version 9.01, you need to upgrade it to the latest version. After upgrading, the pip version is 19.1.1. There will be a small bug after the upgrade, you need to manually change it.

We need to input command:

python3 -m pip install --upgrade pip #upgrade pip

```
nano@nano-desktop:~$ python3 -m pip install --upgrade pip
Collecting pip
  Downloading https://files.pythonhosted.org/packages/5c/e0/be401c003291b56efc55
aeba6a80ab790d3d4cece2778288d65323009420/pip-19.1.1-py2.py3-none-any.whl (1.4MB)
    100% |                                | 1.4MB 84kB/s
Installing collected packages: pip
Successfully installed pip-19.1.1
```

sudo vim /usr/bin/pip3 #Open pip3 file

Replace

```
from pip import main
if __name__ == '__main__':
    sys.exit(main())
```

to

```
from pip import __main__
if __name__ == '__main__':
    sys.exit(__main__._main())
```

After modification is complete, we need to input command:

pip3 -V

```
nano@nano-desktop:~$ pip3 -V
pip 19.1.1 from /home/nano/.local/lib/python3.6/site-packages/pip (python 3.6)
```

2.We need to input command to install some software package:

sudo apt-get install python3-numpy

(It is an extension library of Python language, which supports a large number of dimensional arrays and matrix operations, and also provides a large number of mathematical function libraries for array operations.)

sudo apt-get install python3-scipy

(Scipy is a common software package used in the fields of mathematics, science, and engineering, which can handle interpolation, integration, optimization, image

processing, numerical solution of ordinary differential equations, signal processing, etc.)

sudo apt-get install python3-pandas

(Pandas is a tool based on NumPy, which is created to solve data analysis tasks. Pandas includes a large number of libraries and some standard data models, and provides the tools needed to efficiently operate large data sets. Pandas provides A large number of functions and methods that enable us to process data quickly and easily.)

sudo apt-get install python3-matplotlib

(Matplotlib is a 2D plotting library for Python that generates publishing-quality graphics in a variety of hardcopy formats and a cross-platform interactive environment)

sudo apt-get install python3-sklearn

(Simple and efficient data mining and data analysis tools)

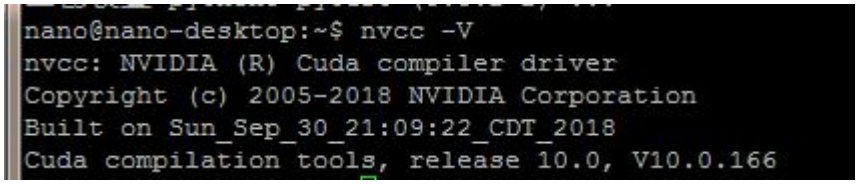
3.Install TensorFlow GPU version

1)Check if CUDA is installed properly

We need to input command:

nvcc -V

If you can see the CUDA version number, as shown below, it is installed correctly.



2)Install the required package

We need to input command:

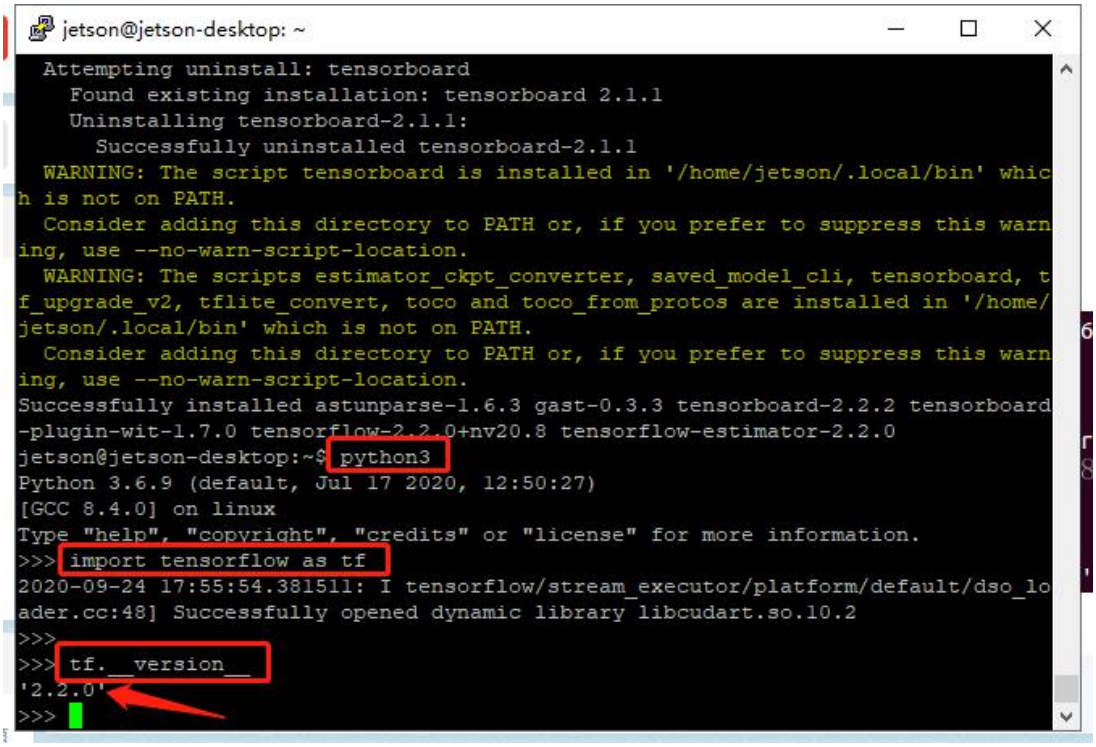
sudo apt-get install python3-pip libhdf5-serial-dev hdf5-tools

3)Install TensorFlow GPU version

We need to input the following command:

pip3 install --extra-index-url
https://developer.download.nvidia.com/compute/redist/jp/v42
tensorflow-gpu==1.13.1+nv19.3 --user

4)Enter the following command to check whether tensorflow is successfully installed.



4. Install Keras

Keras is a high-level neural network API written in Python. It can run with TensorFlow, CNTK, or Theano as the backend.

We can input the following command to install Keras:

sudo pip3 install keras

After the installation is complete, we can input **python3** and check the installation results. When we input **import keras**, the following prompts “using TensorFlow backend”, it proves that Keras is installed successfully and uses TensorFlow as backend.

```
beckhans@Jetson:~$ python3
Python 3.6.7 (default, Oct 22 2018, 11:32:17)
[GCC 8.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import keras
Using TensorFlow backend.
>>>
```

If there is a problem like the one shown below, you need to update the version of numpy.

```
nano@nano-desktop:~$ python3
Python 3.6.7 (default, Oct 22 2018, 11:32:17)
[GCC 8.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import keras
Using TensorFlow backend.
ModuleNotFoundError: No module named 'numpy.core._multiarray_umath'
ImportError: numpy.core.multiarray failed to import

The above exception was the direct cause of the following exception:

Traceback (most recent call last):
  File "<frozen importlib._bootstrap>", line 968, in _find_and_load
SystemError: <class '_frozen_importlib._ModuleLockManager'> returned a result wi
th an error set
ImportError: numpy.core._multiarray_umath failed to import
ImportError: numpy.core.umath failed to import
2019-05-10 17:27:32.651672: F tensorflow/python/lib/core/bfloat16.cc:675] Check
failed: PyBfloat16_Type.tp_base != nullptr
Aborted (core dumped)
```

We can input the following command:

sudo pip3 install numpy==1.16.3

```
nano@nano-desktop: ~
nano@nano-desktop:~$ sudo pip3 install numpy==1.16.3
WARNING: The directory '/home/nano/.cache/pip/http' or its parent directory is n
ot owned by the current user and the cache has been disabled. Please check the p
ermissions and owner of that directory. If executing pip with sudo, you may want
 sudo's -H flag.
WARNING: The directory '/home/nano/.cache/pip' or its parent directory is not ow
ned by the current user and caching wheels has been disabled. check the permissi
ons and owner of that directory. If executing pip with sudo, you may want sudo's
 -H flag.
Collecting numpy==1.16.3
  Downloading https://files.pythonhosted.org/packages/93/48/956b9dcdddfcedb17058
39280e02cbfeb2861ed5d7f59241210530867d5b/numpy-1.16.3.zip (5.1MB)
    |                                | 5.1MB 40kB/s
Building wheels for collected packages: numpy
  Building wheel for numpy (setup.py) ... done
  Stored in directory: /home/nano/.cache/pip/wheels/8c/49/27/48d3f185a57ffcaabdb
366bd1d05e86d587222a85de3b48ec2
Successfully built numpy
Installing collected packages: numpy
  Found existing installation: numpy 1.13.3
    Uninstalling numpy-1.13.3:
      Successfully uninstalled numpy-1.13.3
Successfully installed numpy-1.16.3
nano@nano-desktop:~$
```

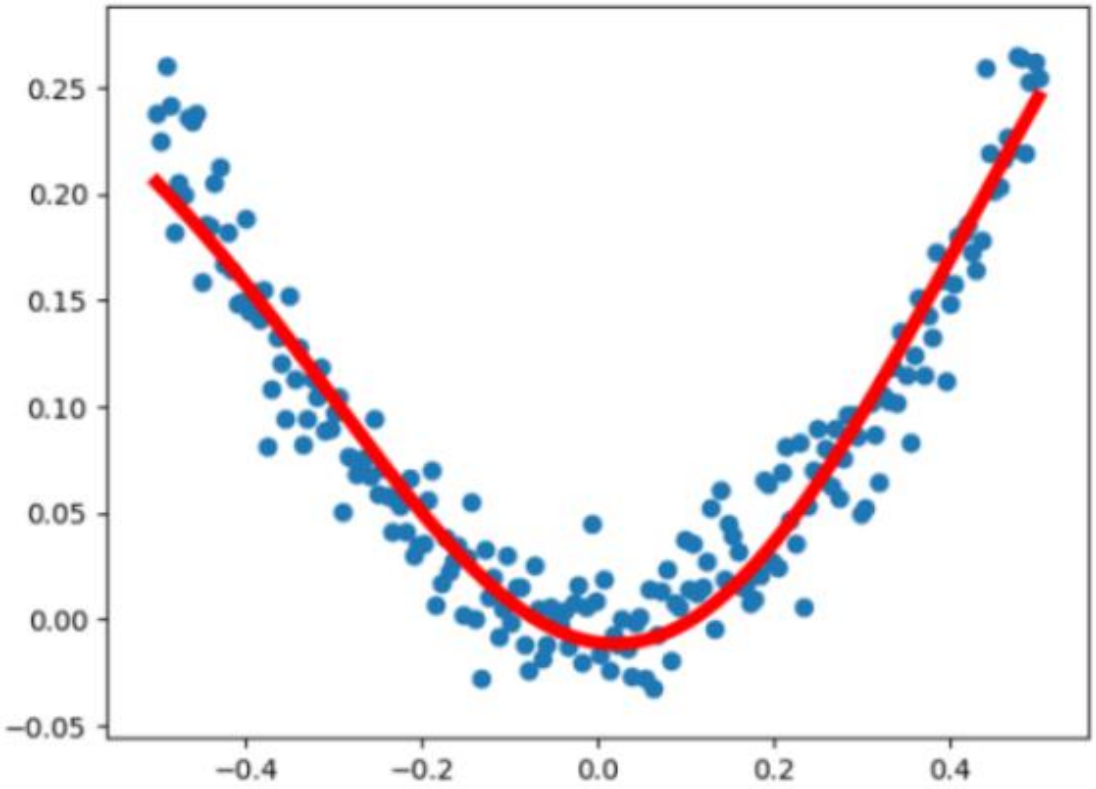
5. Test TensorFlow

We can use vi to create a new python file name: tensorflowDemo.py and then copy the following code into it. After saving, run it with **python3 tensorflowDemo.py**. This section must be run in a graphical interface, because a chart will appear.

```
import tensorflow as tf
import numpy as np
```

```
import matplotlib.pyplot as plt

x_data = np.linspace(-0.5, 0.5, 200)[:, np.newaxis]
noise = np.random.normal(0, 0.02, x_data.shape)
y_data = np.square(x_data) + noise

x = tf.placeholder(tf.float32, [None, 1])
y = tf.placeholder(tf.float32, [None, 1])

# Input layer one neuron, output layer one neuron, middle 10 neurons
# First layer
Weights_L1 = tf.Variable(tf.random.normal([1, 10]))
Biases_L1 = tf.Variable(tf.zeros([1, 10]))
Wx_plus_b_L1 = tf.matmul(x, Weights_L1) + Biases_L1
L1 = tf.nn.tanh(Wx_plus_b_L1)

# Second layer
Weights_L2 = tf.Variable(tf.random.normal([10, 1]))
Biases_L2 = tf.Variable(tf.zeros([1, 1]))
Wx_plus_b_L2 = tf.matmul(L1, Weights_L2) + Biases_L2
pred = tf.nn.tanh(Wx_plus_b_L2)

# Loss function
loss = tf.reduce_mean(tf.square(y - pred))

# Train
train = tf.train.GradientDescentOptimizer(0.1).minimize(loss)

with tf.Session() as sess:
    sess.run(tf.global_variables_initializer())
    for i in range(2000):
        sess.run(train, feed_dict={x: x_data, y: y_data})
        print("第{0}次， loss = {1}".format(i, sess.run(loss,feed_dict={x: x_data, y:
y_data})))
    pred_vaule = sess.run(pred, feed_dict={x: x_data})
    plt.figure()
    plt.scatter(x_data, y_data)
    plt.plot(x_data, pred_vaule, 'r-', lw=5)
    plt.show()
```

```
第1993次, loss = 0.000520005589351058
第1994次, loss = 0.0005199451697990298
第1995次, loss = 0.0005198844010010362
第1996次, loss = 0.0005198236322030425
第1997次, loss = 0.0005197632126510143
第1998次, loss = 0.0005197029677219689
第1999次, loss = 0.0005196425481699407
```


0.25
0.20
0.15
0.10
0.05
0.00
−0.05
−0.4
−0.2
0.0
0.2
0.4