

ubuntu18.04 安装配置记录：

1.配置语言输入环境

使用 ibus 框架安装 ibus-pinyin 输入法

1) 依次点击 `setup-region&Language`

点击最下方的“Manage Installed Languages”，会提示语言环境未安装完全，点击安装即可。

待安装完成后，可能会提示更新，按提示操作，如图 1

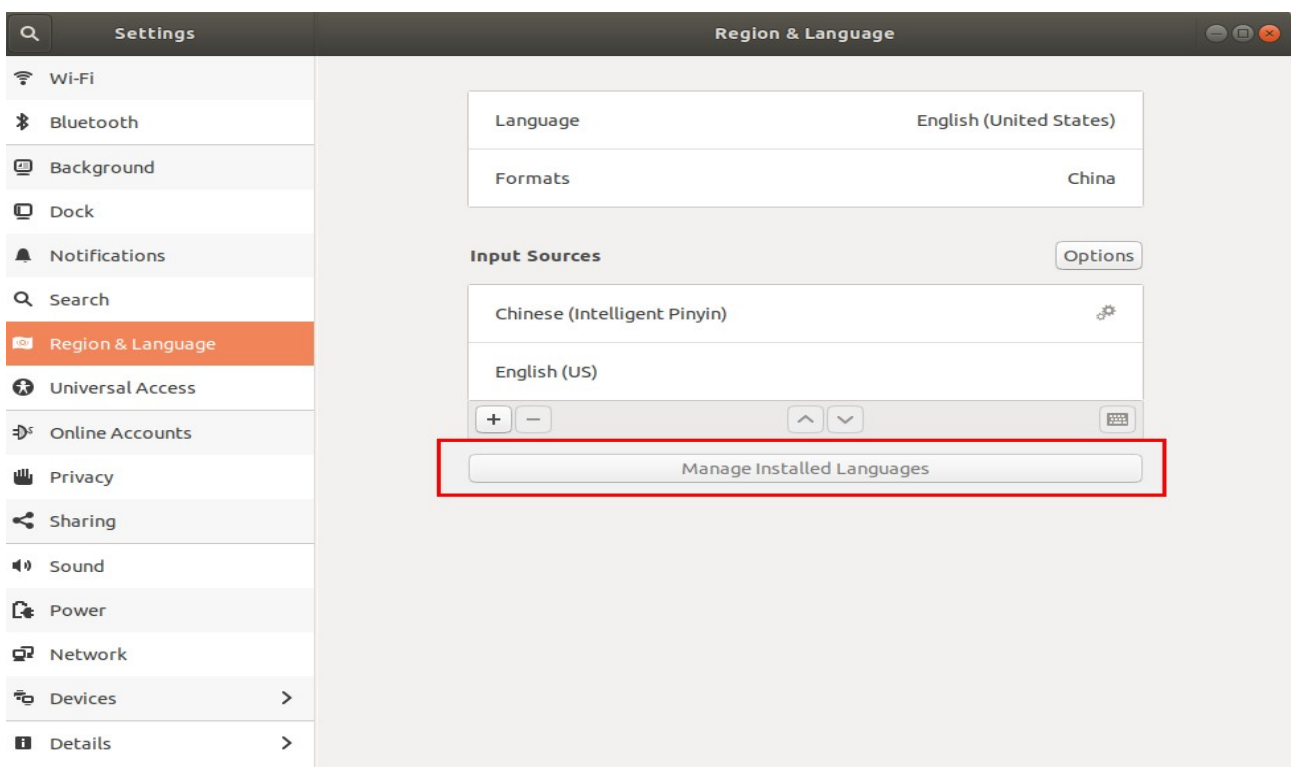


图 1

2) 待上述步骤做完，打开终端，输入如下命令，安装 ibus-pinyin 输入法（核心步骤，非常关键）

```
sudo apt-get install ibus-pinyin
```

3) 再次打开图 1 所示界面，点击“+”号，选中 pinyin，添加拼音输入法即可。如图 2, 3, 4

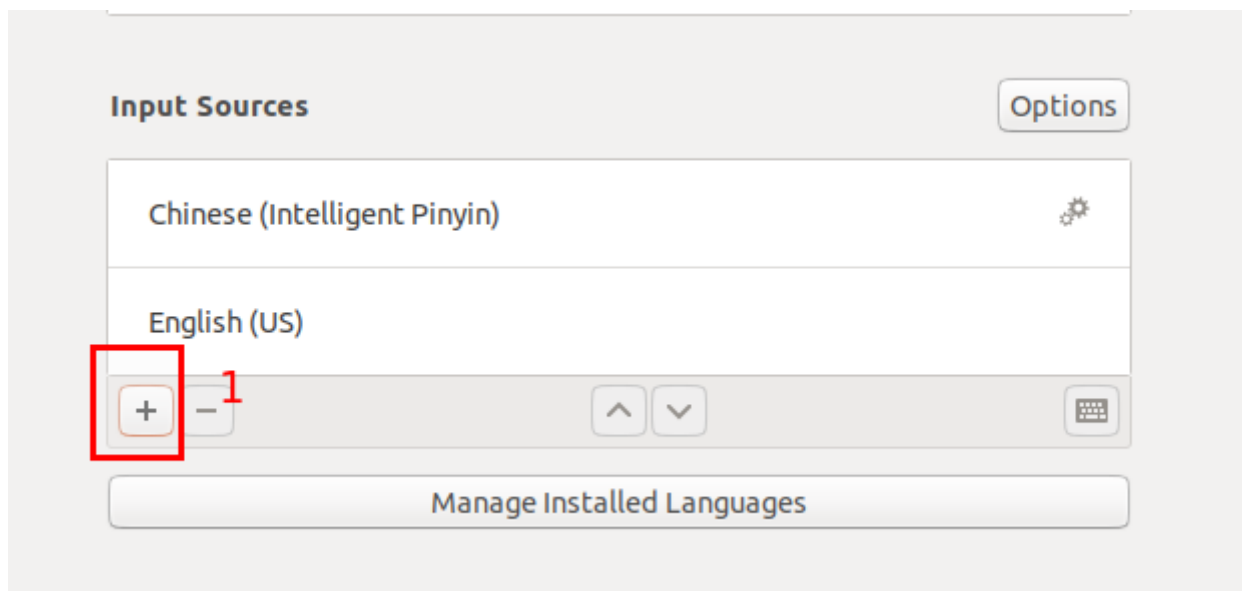


图 2



图 3



图 4

4) 按键盘上的“windows 徽标键”+ “空格”，切换输入法。按 CTRL+SHIFT 切换中英文。

2.安装 nvidia 驱动

0) 切记关掉 bios 中的“secure Boot”选项，或将其选择为“other os”

1) 使用自带的软件管理安装 nvidia 驱动

步骤 1：点击左上角的 Activities，在搜索框里搜索“软件与更新”或者“Software & Update”，支持模糊搜索，如图 5，图 6。

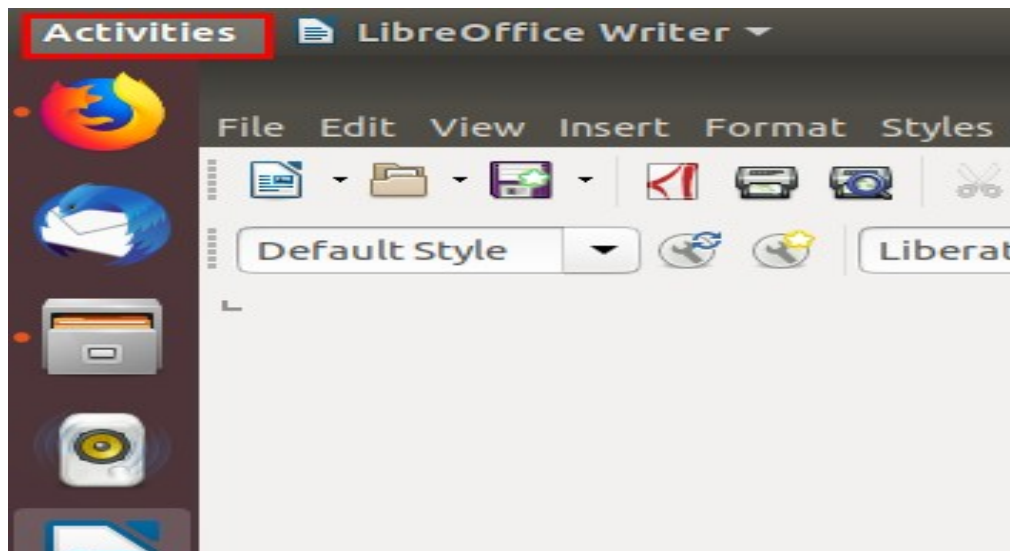


图 5



图 6

步骤 2：在打开的窗口中，点击“Additional Drivers”选项卡，安装自动识别的 Nvidia 驱动，如图 7。

注意，若有双显卡，则显示器一定要接到 Nvidia 的独立显卡接口上。

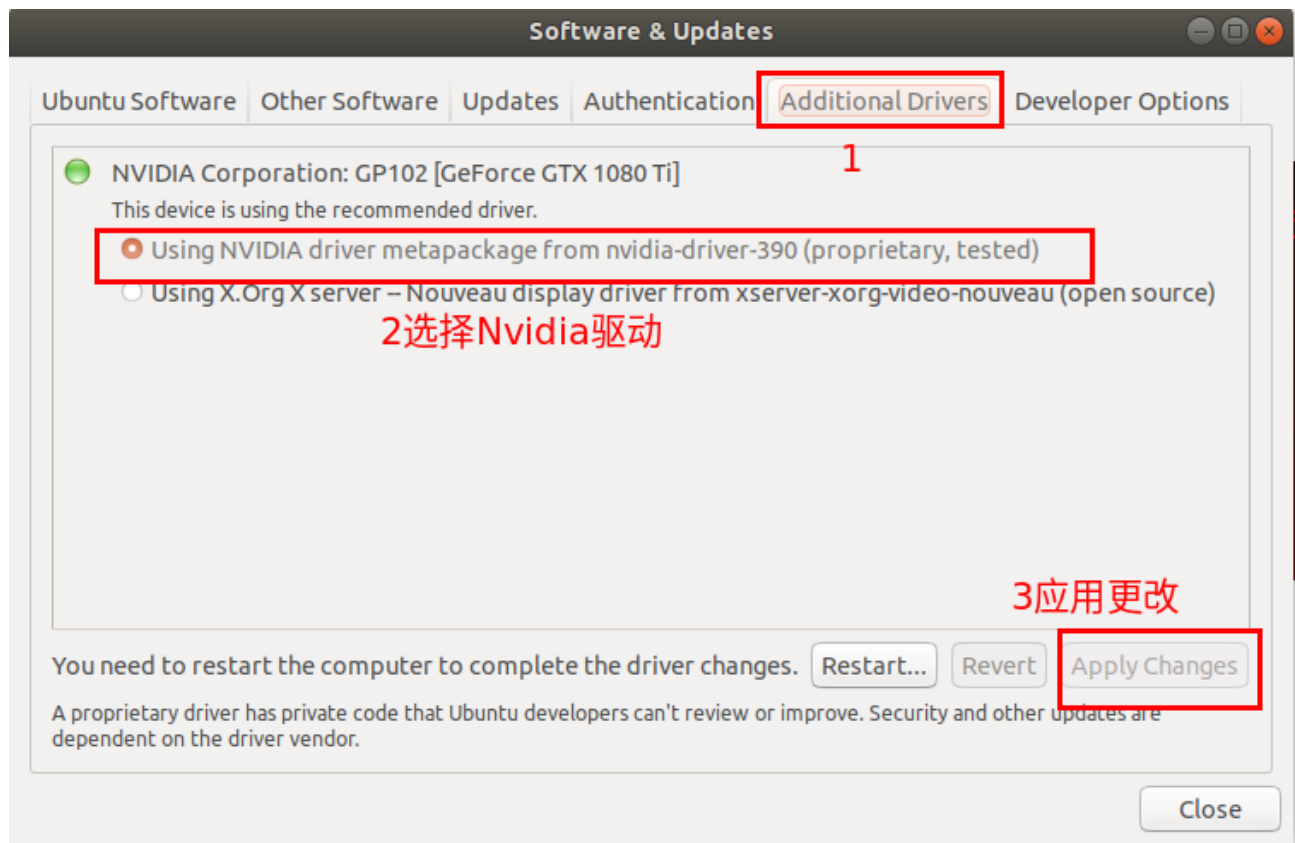


图 7

步骤三：重启电脑，终端运行 `nvidia-smi` 命令，出现图 8 界面表示成功。

```

deep-learn@deeplearn-System-Product-Name: ~
File Edit View Search Terminal Help
deep-learn@deeplearn-System-Product-Name:~$ nvidia-smi
Sun Dec  2 15:26:13 2018

+-----+
| NVIDIA-SMI 390.77                Driver Version: 390.77                |
+-----+-----+
| GPU   Name           Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
+-----+-----+
|    0  GeForce GTX 108...  Off   | 00000000:01:00:0  On   |           N/A       |
| 33%   31C    P8      11W / 250W |  433MiB / 11175MiB |      0%      Default |
+-----+-----+

+-----+
| Processes:                                     GPU Memory |
|  GPU       PID    Type    Process name                     Usage      |
+-----+-----+
|    0      1114     G   /usr/lib/xorg/Xorg                     18MiB      |
|    0      1149     G   /usr/bin/gnome-shell                 50MiB      |
|    0      1416     G   /usr/lib/xorg/Xorg                     223MiB     |
|    0      1529     G   /usr/bin/gnome-shell                 135MiB     |
|    0      2143     G   /usr/lib/firefox/firefox              2MiB       |
+-----+

deep-learn@deeplearn-System-Product-Name:~$ █

```

图 8

3.安装 cuda9.0+cudnn+tensorflow1.12 环境

(tensorflow 只支持到 cuda9.0)

python 环境安装

- 1) `sudo apt update`
- 2) `sudo apt install python3-dev python3-pip`

cuda9.0 安装

(官方教程: <https://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html>)

0) 下载 cuda9.0: (黄色底纹表示这是同一条命令, 否则每一行为一条命令)

wget

https://developer.nvidia.com/compute/cuda/9.0/Prod/local_installers/cuda_9.0.176_384.81_linux-run

1) `sudo chmod +x cuda_9.0.176_384.81_linux.run`

2) `sudo ./cuda_9.0.176_384.81_linux.run`

3) `gedit ~/.bashrc`(或者 `vi ~/.bashrc`)

4)添加如下两行(环境变量):

```
export PATH=/usr/local/cuda-9.0/bin:$PATH
```

```
export LD_LIBRARY_PATH=/usr/local/cuda-9.0/lib64:$LD_LIBRARY_PATH
```

5)立即生效:

```
source ~/.bashrc
```

6) 打印版本:

```
nvcc -V
```

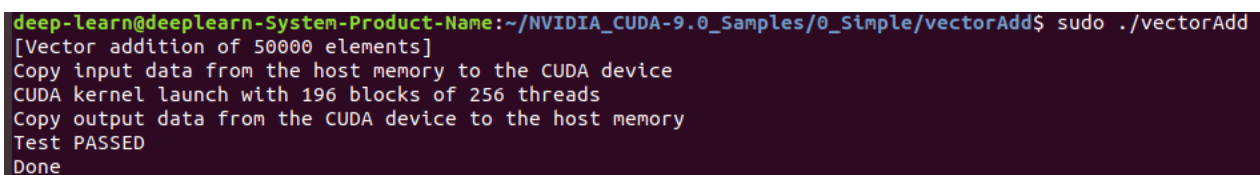
7) 测试 sample 能否成功

```
cd ~/NVIDIA_CUDA-9.0_Samples/0_Simple/vectorAdd
```

```
make
```

```
sudo ./vectorAdd
```

结果如图 9:



```
deep-learn@deeplearn-System-Product-Name:~/NVIDIA_CUDA-9.0_Samples/0_Simple/vectorAdd$ sudo ./vectorAdd
[Vector addition of 50000 elements]
Copy input data from the host memory to the CUDA device
CUDA kernel launch with 196 blocks of 256 threads
Copy output data from the CUDA device to the host memory
Test PASSED
Done
```

图 9

cuda7.3 安装

(官方教程:<https://docs.nvidia.com/deeplearning/sdk/cudnn-install/index.html#installlinux-tar>)

0) 下载 cudnn:

```
wget https://developer.nvidia.com/compute/machine-learning/cudnn/secure/v7.3.0/prod/9.0_2018920/cudnn-9.0-linux-x64-v7.3.0.29
```

- 1) `tar -xzf cudnn-9.0-linux-x64-v7.3.1.20.tgz`
- 2) `sudo cp lib64/* /usr/local/cuda/lib64`
- 3) `sudo cp include/* /usr/local/cuda/include/`
- 4) `sudo chmod a+r /usr/local/cuda/include/cudnn.h`
`/usr/local/cuda/lib/libcudnn*`

tensorflow 安装

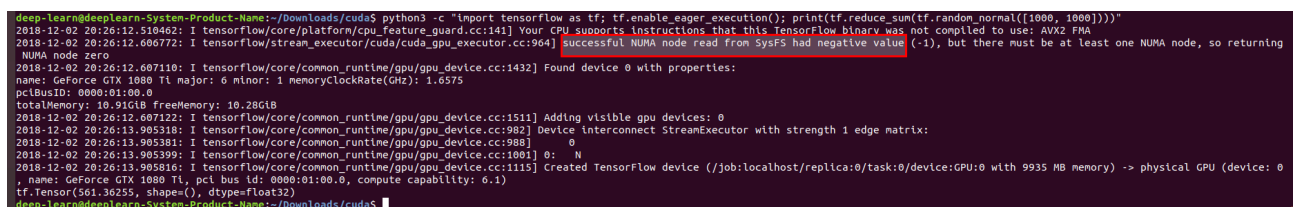
(官方教程: <https://tensorflow.google.cn/install/pip>)

1) `pip3 install --user --upgrade tensorflow-gpu`

2) 测试:

```
python3 -c "import tensorflow as tf; tf.enable_eager_execution();  
print(tf.reduce_sum(tf.random_normal([1000, 1000])))"
```

若成功界面如图 10:



```
deep-learn@deep-learn-System-Product-Name:~/Downloads/cuda$ python3 -c "import tensorflow as tf; tf.enable_eager_execution(); print(tf.reduce_sum(tf.random_normal([1000, 1000])))"  
2018-12-02 20:26:12.510462: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 FMA  
2018-12-02 20:26:12.606772: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:964] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning  
NUMA node zero  
2018-12-02 20:26:12.607110: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1432] Found device 0 with properties:  
name: GeForce GTX 1080 Ti major: 6 minor: 1 memoryClockRate(GHz): 1.6575  
pciBusID: 0000:01:00.0  
totalMemory: 10.91GiB freeMemory: 10.28GiB  
2018-12-02 20:26:12.607122: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1511] Adding visible gpu devices: 0  
2018-12-02 20:26:13.905318: I tensorflow/core/common_runtime/gpu/gpu_device.cc:982] Device interconnect StreamExecutor with strength 1 edge matrix:  
2018-12-02 20:26:13.905381: I tensorflow/core/common_runtime/gpu/gpu_device.cc:988] 0  
2018-12-02 20:26:13.905399: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1081] 0: N  
2018-12-02 20:26:13.905816: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1115] Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 with 9935 MB memory) -> physical GPU (device: 0  
, name: GeForce GTX 1080 Ti, pci bus id: 0000:01:00.0, compute capability: 6.1)  
tf.Tensor(561.36255, shape=(), dtype=float32)  
deep-learn@deep-learn-System-Product-Name:~/Downloads/cuda$
```

图 10

4.安装 Matlab2015

1.切换到安装脚本所在目录：

```
cd Matlab2015b/Matlab
```

2.运行安装脚本

```
sudo ./install
```

3.在图形界面接受协议

4.选择“I have a installation key file”，填入序列号：

```
09806-07443-53955-64350-21751-41297
```

5.点击“Next”等待安装完成

6.复制 **Matlab_2015b_Linux64_Crack** 目录中的 R2015b/gln/glnxa64 下的所有文件到安装目录的对应位置：注：下面这是一条命令，不要分两次执行。

```
sudo cp Matlab_2015b_Linux64_Crack/R2015b/bin/glnxa64/*  
/usr/local/MATLAB/R2015b/bin/glnxa64/
```

7.在安装目录下创建“licences”目录：

```
sudo mkdir /usr/local/MATLAB/licenses
```

8.将 Matlab_2015b_Linux64_Crack 下的 license_standalone.lic 文件复制到上面创建的目录里：

注：先切换到上述目录再执行下面的命令。

```
sudo cp license_standalone.lic /usr/local/MATLAB/licenses/
```

9. 执行 matlab 程序继续激活 `sudo /usr/local/MATLAB/R2015b/bin/matlab`

10.选择手动不联网激活，如图 11

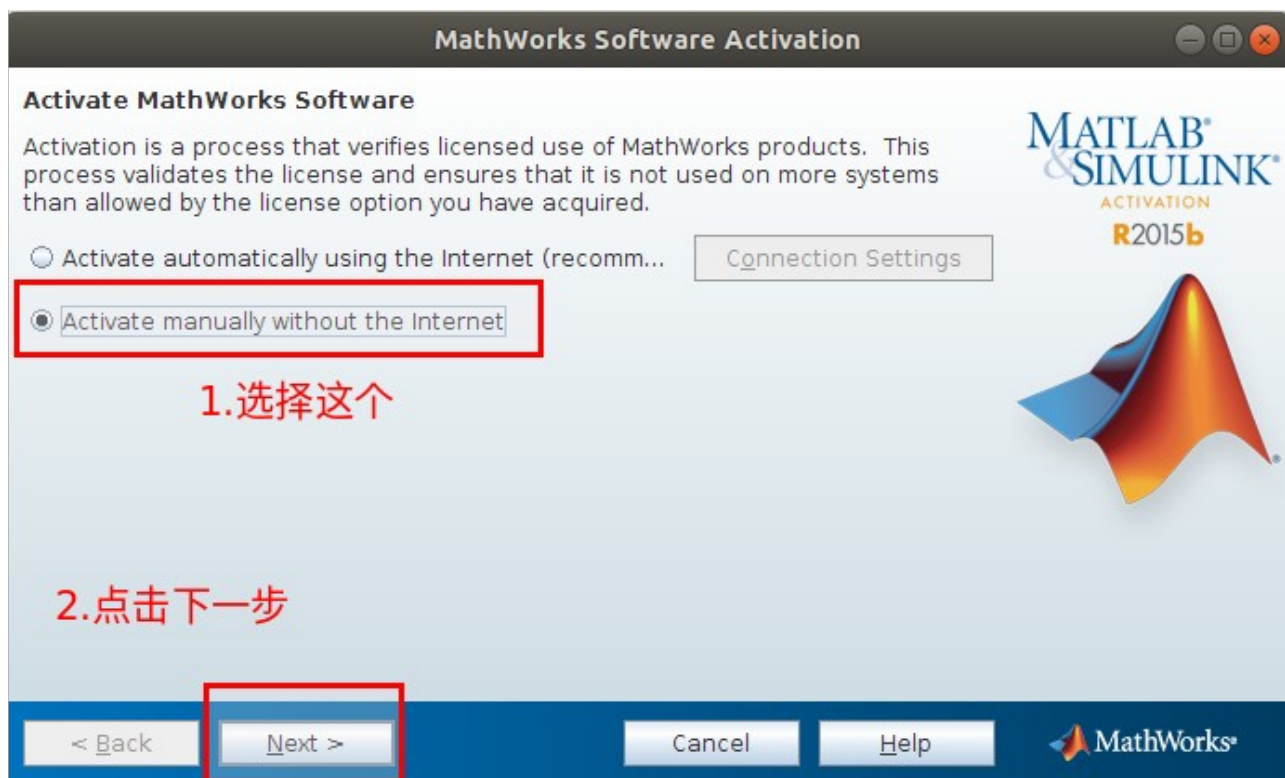


图 11

12. 选择激活文件的路径，如图 12



图 12

13.显示 **Activation is complete** ，激活完成，点击 finish 关闭程序。

14. 测试激活是否成功

```
sudo /usr/local/MATLAB/R2015b/bin/matlab
```

若出现图 13 所示错误，则执行以下命令

```
mv /usr/local/MATLAB/R2015b/sys/os/glnxa64/libstdc++.so.6  
/usr/local/MATLAB/R2015b/sys/os/glnxa64/libstdc++.so.6.old
```

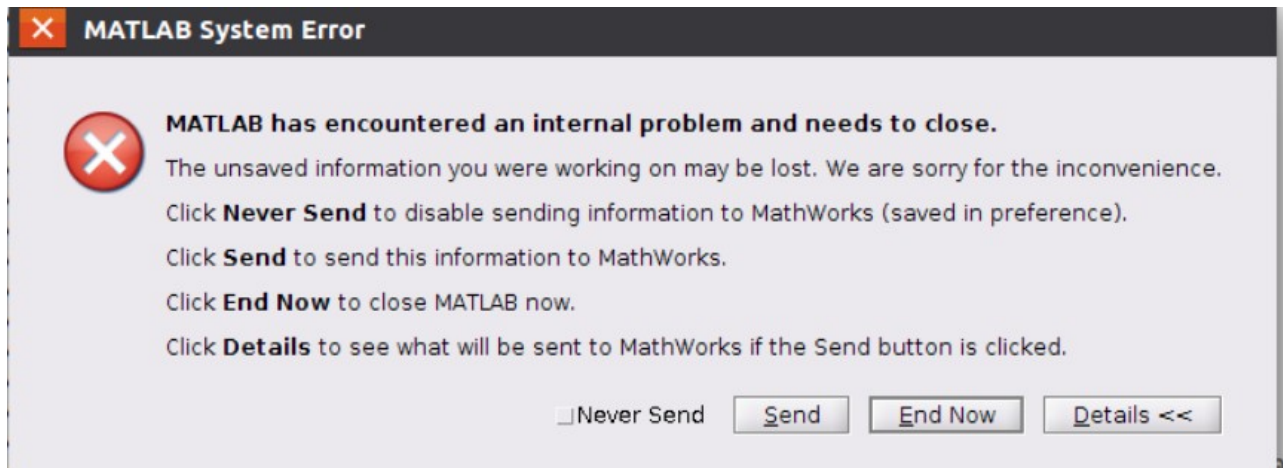


图 13

15. 添加 matlab 快捷方式到桌面

```
gedit ~/Desktop/Matlab.desktop
```

加入如下语句：

```
[Desktop Entry]
```

```
Name=Matlab 2015b
```

```
Exec=/usr/local/MATLAB/R2015b/bin/matlab -desktop
```

```
Icon=/home/deep-learn/Pictures/matlab.png #matlab.png 在 Crack 目录  
下，需复制到相应的目录下
```

```
Type=Application
```

```
Name[zh_CN]=Matlab 2015b
```

5. 安装 teamviewer

1) 官网下载 teamviewer14 for ubuntu

2) `sudo dpkg -i teamviewer-***.deb`(安装包的名字), 会提示缺少组件, 依赖关系有问题

3) 强制安装组件, 不修补依赖关系

```
sudo apt install -f
```

4) 再次安装 运行 teamviewer 安装包

```
sudo dpkg -i teamviewer-***.deb
```

5) 设置开机自启动

- 开启守护进程

```
sudo systemctl enable teamviewerd
```

- 设置图形界面开机启动, 如图 14

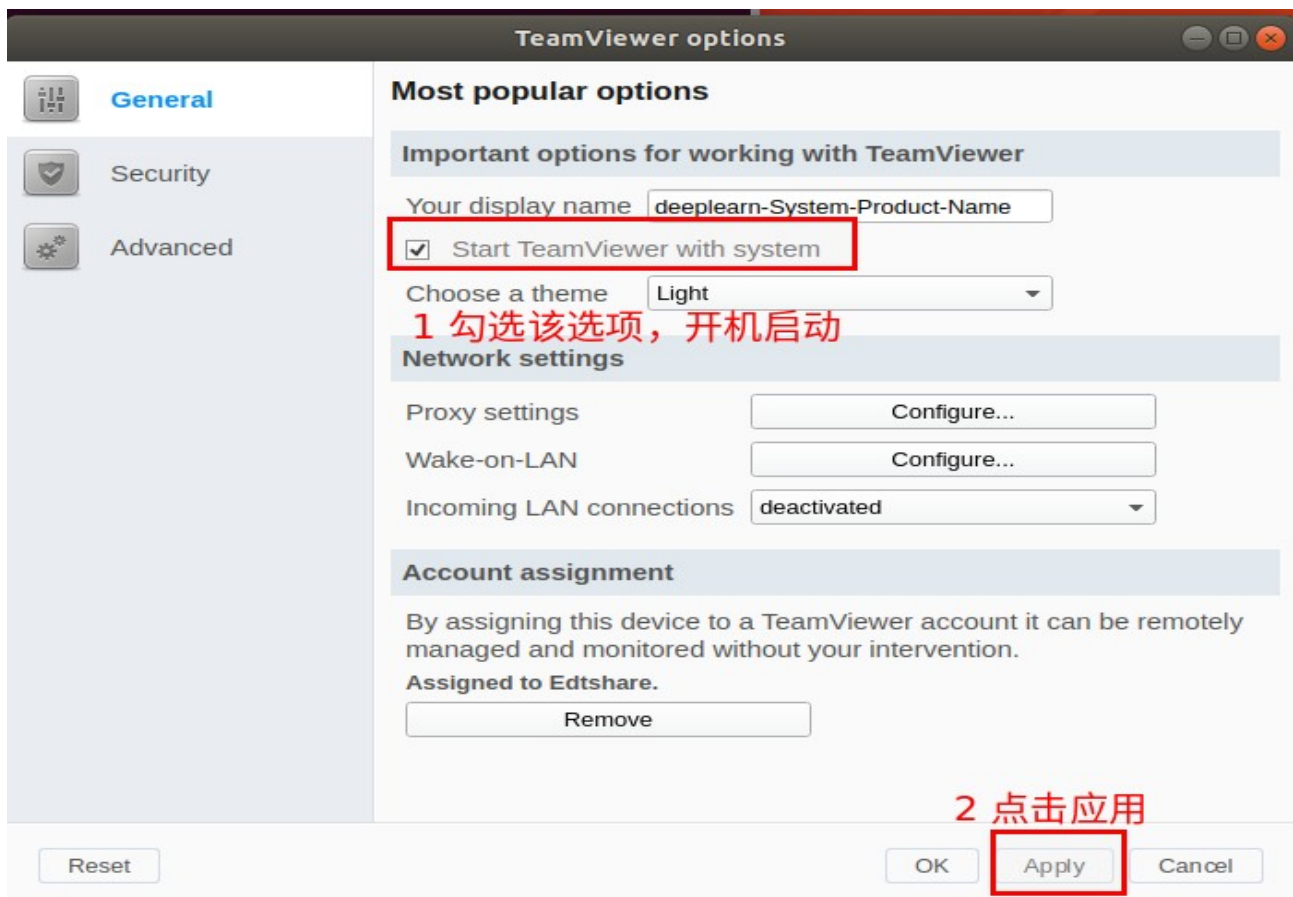


图 14