

win10 + WSL + TensorFlow1.4 + Python3.5.2+Ubuntu16 内核

by Xian2207, 13689903575, wszhangxian@126.com

第一部分: WSL(Windows Subsystem Linux 即 windows 内置 linux 子系统)安装

步骤 1: 右键点开始, 选择设置-更新与安全-针对开发人员选项, 选择开发人员模式



步骤 2: 打开控制面板, 依次选择程序 - 启用或关闭 Windows 功能 - 适用于 linux 的 windows 子系统 - 确定



步骤 3: **重启电脑**

步骤 4: **win+R** 键打开运行, 输入 **cmd** 敲回车键进入 dos 环境, 输入 **lxrun /install /y** 注意中间的空格和反斜杠方向不要错;

```
CA: C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.16299.125]
(c) 2017 Microsoft Corporation。保留所有权利。

C:\Users\user>lxrun /install /y
```

步骤 5: 安装成功, 输入 **bash**, 后面就是 linux 的系统环境, 虽然共用 dos 环境, 但系统可是调用 linux 的

```
C:\Users\user>lxrun /install /y
警告: lxrun.exe 仅用于配置适用于 Linux 的 Windows 子系统的旧分发版。
可以通过访问 Windows 应用商店来安装分发版:
https://aka.ms/wslstore

这将在 Windows 上安装由 Canonical 分发的 Ubuntu, 其授权所遵循的条款请参见此链接:
https://aka.ms/uowterms

正在从 Windows 应用商店下载... 100%
正在提取文件系统, 这将需要几分钟的时间...
安装成功!
文档在以下网址提供: https://aka.ms/wsldocs

C:\Users\user>bash
root@DESKTOP-NUAMBCJ:/mnt/c/Users/user#
```

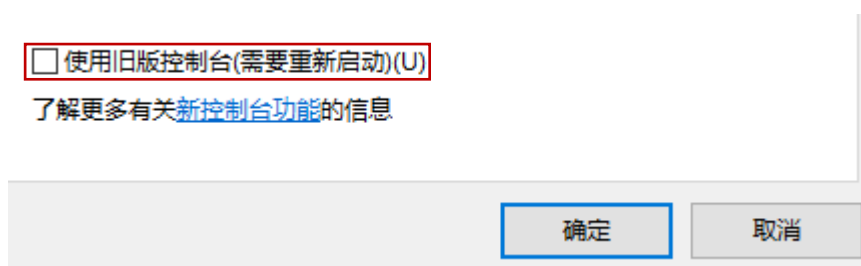
至此, linux 安装成功!

安装时出现的问题:

1 如果曾经装过 WSL, 且在 c:/Users/user/AppData/Local 文件夹下找不到 lxss 文件, 暴力删除后, 参考见: <https://blog.csdn.net/u013779141/article/details/25274807> 在 DOS 环境下输入 lxrun/install 或 Cmder 会出现“控制台不收支持, 若使用此功能, 必须禁用旧版控制台”和以下

```
C:\Users\user>lxrun/install
警告: lxrun.exe 仅用于配置适用于 Linux 的 Windows 子系统的旧分发版。
可以通过访问 Windows 应用商店来安装分发版:
https://aka.ms/wslstore
```

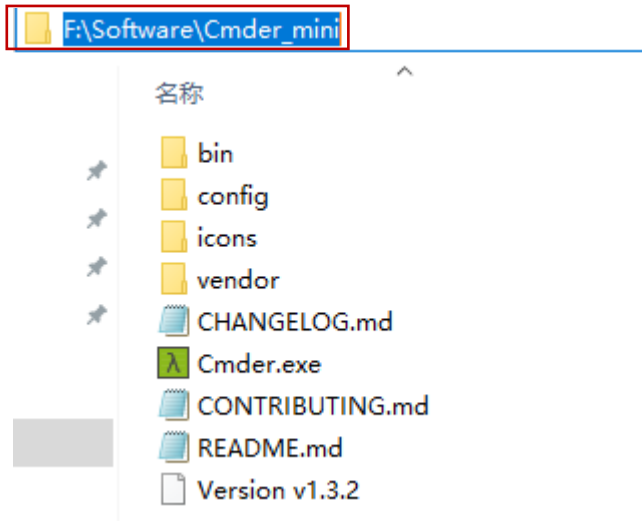
解决方式: 将 cmder 等额外的控制台删除, 然后 win+R 进入 windows 控制台, 点击控制台左上角, 点击属性, 取消勾选, 点击确定。



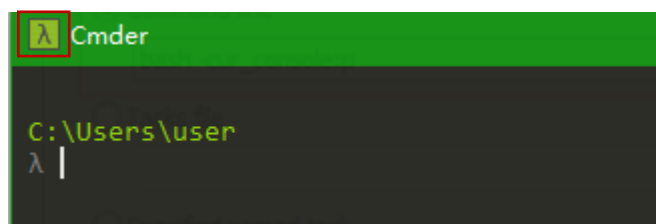
第二部分：安装 cmdr 控制台，方便未来操作

首先：下载适合 linux 运行的命令窗口，类似于 dos 环境，下载链接：<http://cmdr.net/>

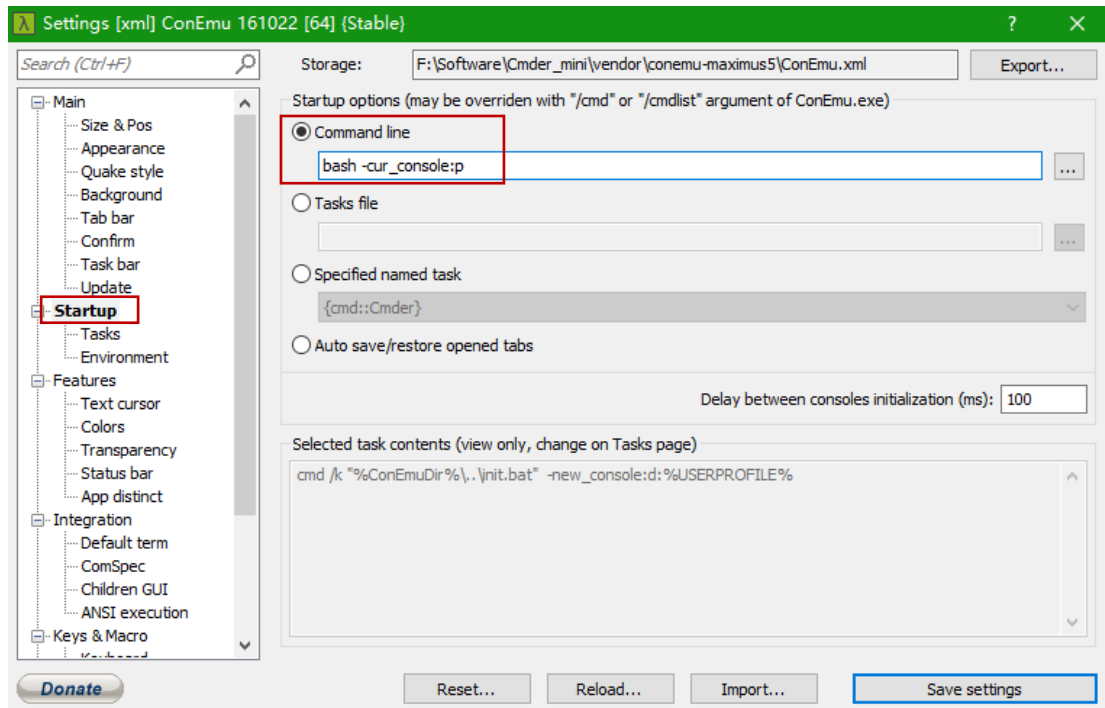
把 zip 文件夹放在你指定的位置，然后解压即可，例如我的放在：`F:\Software\Cmder_mini`;



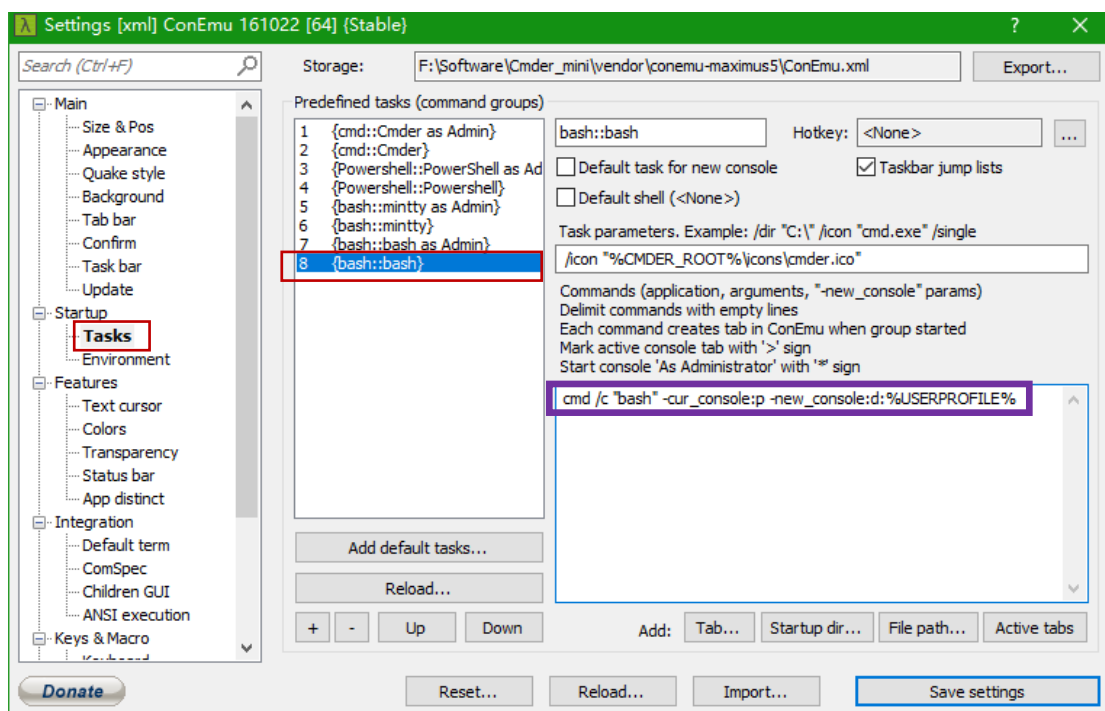
其次，双击 Cmder.exe，点击入这个标志，依次选择 `setting-Startup-Command line`



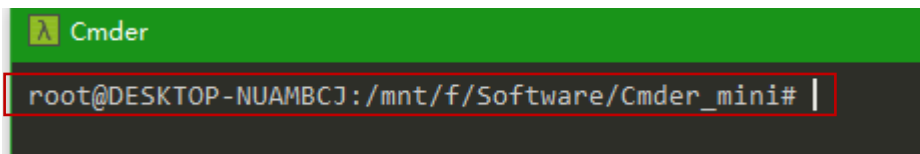
输入：`bash -cur_console:p` 完了 `save setting` 退出。



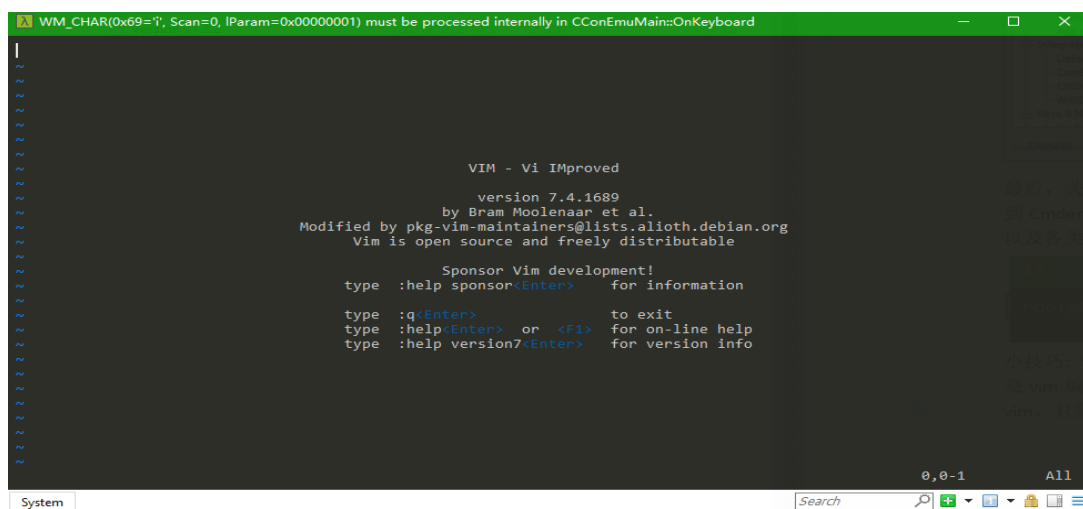
再次点击入标志, 依次选择 **setting-Startup-Task-8{bash::bash}**, 在 Start console 'as Administrator' 的大白框内用 `cmd /c "bash" -cur_console:p -new_console:d:%USERPROFILE%` 替换原来已存的字符串 (紫框) ;



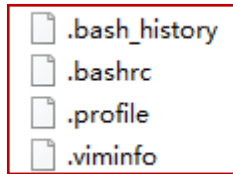
最后，关闭 Cmdr 窗口，可以 **exit+回车**，也可以鼠标点击**红叉**按钮。下次打开，直接定位到 Cmdr 的目录下（见下图），而 Cmdr 就是 linux 的操作界面，包括文本编辑，创建文件夹，安装 firefox 以及各类图形界面程序。



小技巧（1）：输入 **vi** 接着 **Tab** 键，注意 **vi** 之后按下键盘上的 **Tab** 键，你就会看到 **vim.basic**。它是 vim 编辑器的迷你版，类似于 notepad++和记事本.如果要进入 vim，输入 **vi+回车**即可；



按下 **I**(Insert 的 **I**)键, 屏幕上的说明不见了，下面是**--INSERT--**模式，可以进行文本编辑。



小技巧（3）：输入 clear，可以直接清屏。

第二部分: tensorflow1.4 安装

前提条件 1: 检查 win10+WSL 下的 python, 方法是打开 Cmder, 输入 python, 接着输入 python3

```
Cmder
root@DESKTOP-NUAMBCJ:/mnt/f/Software/Cmder_mini# python
The program 'python' can be found in the following packages:
 * python-minimal
 * python3
Try: apt install <selected package>
root@DESKTOP-NUAMBCJ:/mnt/f/Software/Cmder_mini# python3
Python 3.5.2 (default, Nov 17 2016, 17:05:23)
[GCC 5.4.0 20160609] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

可见, win10+WSL 默认安装了 Python3.5.2 版本。由于我的电脑没有 NIVIDA 显卡, 所以只能安装 CPU 版的 tensorflow。自己是否安装 NIVIDA 显卡(也叫显示适配卡-Display Adpater), 看链接: <https://jingyan.baidu.com/article/7082dc1c516c17e40b89bd4e.html> 如果有, 你可以安装 GPU 版本的 tensorflow, 计算能力更强更快;

前提条件 2: 根据官网安装 tensorflow, 我按照 google 的建议, 直接选择 virtualenv installation。因为不同版本的 python 可以并行不悖, 不影响 tensorflow 的运行, 安装过程见官网链接: https://www.tensorflow.org/install/install_linux

步骤 1: 安装 virtualenv

分析 1: 因为 WSL 在 C:\Users\user\AppData\Local\lxss 为了不影响这个 Linux 系统, 担心删除操作会影响这些 Linux 系统文件, 我把 virtualenv 安装在和我 Cmder 一样的安装目录内。未安装前, 是这样的:

名称	修改日期	类型
bin	2016/12/2 7:14	文件夹
config	2017/12/20 13:57	文件夹
icons	2016/12/2 7:14	文件夹
vendor	2016/12/2 7:15	文件夹
CHANGELOG.md	2016/12/2 7:14	MD 文件
Cmder.exe	2016/12/2 7:15	应用程序
CONTRIBUTING.md	2016/12/2 7:14	MD 文件
README.md	2016/12/2 7:14	MD 文件
Version v1.3.2	2016/12/2 7:15	2 文

现在打开 Cmder，根据上述官网链接，考虑我是 Python3，所以输入

```
sudo apt-get install python3-pip python3-dev python-virtualenv
```

再输入[Y]，即 Yes，继续安装

```
root@DESKTOP-NUAMBCJ:/mnt/f/Software/Cmdr_mini# sudo apt-get install python3-pip python3-dev python-virtualenv
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils build-essential cpp cpp-5 dpkg-dev fakeroot g++ g++-5 gcc gcc-5 libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan2 libatomic1 libc-dev-bin libc6-dev libc6-i386 libcilkrts5
  libbdp-gpl libbexpat1-dev libfakeroot libfile-cntllock-perl libgcc-5-dev libgomp1 libisl15 libitm1 liblsan0
  libmpc3 libmpx0 libpython-stdlib libpython2.7-minimal libpython2.7-stdlib libpython3-dev libpython3.5-dev
  libquadmath0 libstdc++-5-dev libstdc++6 libubsan0 linux-libc-dev make manpages-dev python python-minimal
  python-pip-whl python-pkg-resources python2.7 python2.7-minimal python3-setuptools python3-virtualenv python3-wheel
```

中途出现 E, Failed 提示, 即 Error:

```
E: Failed to fetch http://archive.ubuntu.com/ubuntu/pool/main/d/dpkg/dpkg-dev_1.18.4ubuntu1-1ubuntu1.1.dsc 404 Not Found [IP: 91.189.88.161 80]
E: Failed to fetch http://security.ubuntu.com/ubuntu/pool/main/e/expat/libexpat1-dev_2.1.0-11ubuntu1.1.dsc 404 Not Found [IP: 91.189.88.161 80]
E: Failed to fetch http://security.ubuntu.com/ubuntu/pool/main/p/python3.5/libpython3.5-dev_3.5.2-1ubuntu1.dsc 404 Not Found [IP: 91.189.88.161 80]
E: Failed to fetch http://security.ubuntu.com/ubuntu/pool/main/p/python3.5/python3.5-dev_3.5.2-1ubuntu1.dsc 404 Not Found [IP: 91.189.88.161 80]
E: Unable to fetch some archives, maybe run apt-get update or try with --fix-missing?
root@DESKTOP-NUAMBCJ:/mnt/f/Software/Cmder_mini#
```

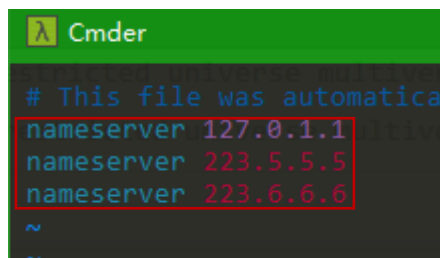
因为人在国内，数据源在国外，所以要么更改数据源，要么更改 DNS，即电脑 IP 地址域名。我选择后者，改到阿里数据源，具体操作是：输入 `sudo vi /etc/resolv.conf`

```
E: Failed to fetch http://security.ubuntu.com/ubuntu/pool/main/p/python3.5/python3.5-deb
b 404 Not Found [IP: 91.189.88.161 80]

E: Unable to fetch some archives, maybe run apt-get update or try with --fix-missing?
root@DESKTOP-NUAMBCJ:/mnt/f/Software/Cmder_mini# sudo vi /etc/resolv.conf
```

进入 vim，回忆我前面说的怎么进入 vim 的编辑状态，这里推荐按 **I** 键(Insert 的首写字母)，

直接进入编辑状态，然后修改成下面：

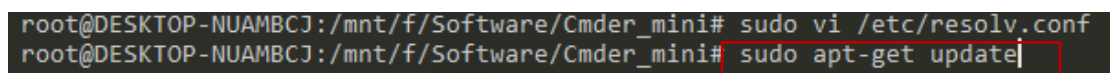


```
# This file was automatically generated by NetworkManager.
nameserver 127.0.1.1
nameserver 223.5.5.5
nameserver 223.6.6.6
```

完了要保存退出，先按 Esc 键，输入:x，最后回车即可，又返回到 Cmder 界面，输入

Sudo apt-get update，完了回车执行这段命令；

参考链接：<http://blog.csdn.net/feinia8651/article/details/60332535>

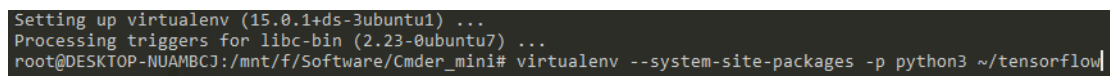


```
root@DESKTOP-NUAMBCJ:/mnt/f/Software/Cmder_mini# sudo vi /etc/resolv.conf
root@DESKTOP-NUAMBCJ:/mnt/f/Software/Cmder_mini# sudo apt-get update
```

完了，重新执行 `sudo apt-get install python3-pip python3-dev python-virtualenv` 即可下载；

步骤 2：创建 virtualenv 环境

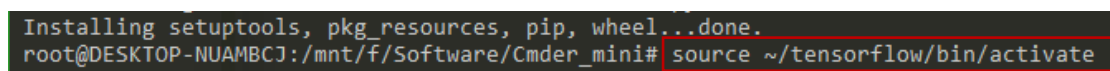
输入红框中的代码 `virtualenv --system-site-packages -p python3 ~/tensorflow`



```
Setting up virtualenv (15.0.1+ds-3ubuntu1) ...
Processing triggers for libc-bin (2.23-0ubuntu7) ...
root@DESKTOP-NUAMBCJ:/mnt/f/Software/Cmder_mini# virtualenv --system-site-packages -p python3 ~/tensorflow
```

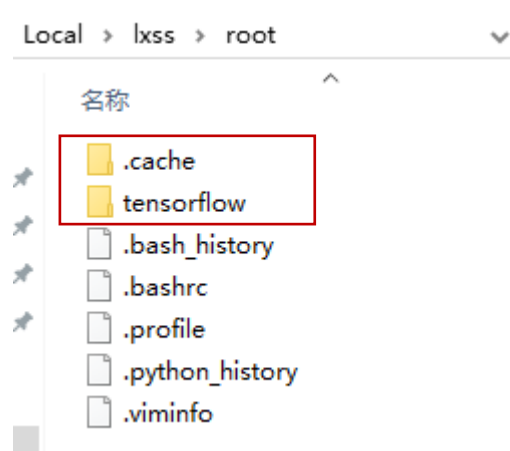
步骤 3：激活 virtualenv

输入 `source ~/tensorflow/bin/activate`







```
Installing setuptools, pkg_resources, pip, wheel...done.
root@DESKTOP-NUAMBCJ:/mnt/f/Software/Cmder_mini# source ~/tensorflow/bin/activate
```

此时打开 linux 安装路径 `C:\Users\user\AppData\Local\lxss\root`，现在多了两个文件夹



对比原来，说明上述步骤很成功的

-  .bash_history
-  .bashrc
-  .profile
-  .viminfo

此外，激活后，你会发现下面两个不同点，root 变为 tensorflow 了

```
root@DESKTOP-NUAMBCJ:/mnt/f/Software/Cmder_mini# source ~/tensorflow/bin/activate
(tensorflow) root@DESKTOP-NUAMBCJ:/mnt/f/Software/Cmder_mini# |
```

步骤 4: 安装 tensorflow in active virtualenv

输入 `pip3 install --upgrade tensorflow`

```
root@DESKTOP-NUAMBCJ:/mnt/f/Software/Cmder_mini# source ~/tensorflow/bin/activate
(tensorflow) root@DESKTOP-NUAMBCJ:/mnt/f/Software/Cmder_mini# pip3 install --upgrade tensorflow
```

步骤 5 验证 tensorflow 是否安装成功

安装完毕后，输入 `python`

```
Not uninstalling six at /usr/lib/python3/dist-packages, outside environment /root/tensorflow
Successfully installed bleach-1.5.0 enum34-1.1.6 html5lib-0.9999999 markdown-2.6.10 numpy-1
x-1.11.0 tensorflow-1.4.1 tensorflow-tensorboard-0.4.0rc3 werkzeug-0.13
(tensorflow) root@DESKTOP-NUAMBCJ:/mnt/f/Software/Cmder_mini# python
Python 3.5.2 (default, Nov 23 2017, 16:37:01)
[GCC 5.4.0 20160609] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> |
```

输入示例代码:

```
import tensorflow as tf
hello = tf.constant('Hello, TensorFlow!')
sess = tf.Session()
print(sess.run(hello))
```

它的正确结果是

If the system outputs the following, then you are ready to begin writing TensorFlow programs:

```
Hello, TensorFlow!
```

我的结果是

```
>>> import tensorflow as tf
>>> hello = tf.constant('Hello, TensorFlow!')
>>> sess = tf.Session()
2017-12-20 16:44:52.415886: I tensorflow/core/platform/cpu_fe
s TensorFlow binary was not compiled to use: SSE4.1 SSE4.2 AV
>>> print(sess.run(hello))
b'Hello, TensorFlow!'
>>>
```

不要紧，可能是 win10 版本所致，毕竟官网那个是 Ubuntu 环境，我们这里是 WSL。

步骤 6: 退出 virtualenv tensorflow

要先用 Python 的语法退出，`exit()`，然后输入 `deactivate` 退出 `virtualenv` 环境。以后每次进入 `tensorflow` 编程，都需要先激活 `virtualenv`，然后再重复步骤 5；退出时重复步骤 6。见成功退出的标志如下，又回到了 `root` 权限。

```
b'Hello, TensorFlow!'
>>> exit()
(tensorflow) root@DESKTOP-NUAMBCJ:/mnt/f/Software/Cmder_mini# deactivate
root@DESKTOP-NUAMBCJ:/mnt/f/Software/Cmder_mini# |
```

第三部分：Linux+tensorflow1.4+JDK+SDK+NDK 搭建环境

任务 1: 安装 Bazel (内含 JDK)

参考链接: <https://docs.bazel.build/versions/master/install-ubuntu.html#install-on-ubuntu> 网上安装五花八门，其实就三种安装方式。什么先安装依赖库，再安装 `bazel`，或者安装 `wheel` 都不要看，直接看上面的官网才对。我的是 Ubuntu16.04，所以直接按照推荐的过程来。

(1) 输入: `sudo apt-get install openjdk-8-jdk` 之后回车

```
root@DESKTOP-NUAMBCJ:/mnt/f/Software/Cmder_mini# sudo apt-get install openjdk-8-jdk
```

(2) 安装好后，输入: `echo "deb [arch=amd64] http://storage.googleapis.com/bazel-apt stable jdk1.8" | sudo tee /etc/apt/sources.list.d/bazel.list`

```
done.
done.
root@DESKTOP-NUAMBCJ:/mnt/f/Software/Cmder_mini# echo "deb [arch=amd64] http://storage.googleapis.com/bazel-
dk1.8" | sudo tee /etc/apt/sources.list.d/bazel.list
```

(3) 再输入: `curl https://bazel.build/bazel-release.pub.gpg | sudo apt-key add -`

```
root@DESKTOP-NUAMBCJ:/mnt/f/Software/Cmder_mini# echo "deb [arch=amd64] http://storage.googleapis.com/bazel-apt stable
dk1.8" | sudo tee /etc/apt/sources.list.d/bazel.list
deb [arch=amd64] http://storage.googleapis.com/bazel-apt stable jdk1.8
root@DESKTOP-NUAMBCJ:/mnt/f/Software/Cmder_mini# curl https://bazel.build/bazel-release.pub.gpg | sudo apt-key add -
```

(4) 再输入: `sudo apt-get update && sudo apt-get install bazel`

```
OK
root@DESKTOP-NUAMBCJ:/mnt/f/Software/Cmdr_mini# sudo apt-get update && sudo apt-get install bazel
```

(5) 再输入: **sudo apt-get upgrade bazel**

```
done.
done.
Processing triggers for resolvconf (1.78ubuntu5) ...
root@DESKTOP-NUAMBCJ:/mnt/f/Software/Cmdr_mini#
```

(6) 再输入 **bazel version** 验证是否安装成功

```
root@DESKTOP-NUAMBCJ:/mnt/f/Software/Cmdr_mini# bazel version
Extracting Bazel installation...
Build label: 0.9.0
Build target: bazel-out/k8-fastbuild/bin/src/main/java/com/google/dev
Build time: Tue Dec 19 09:31:58 2017 (1513675918)
Build timestamp: 1513675918
Build timestamp as int: 1513675918
root@DESKTOP-NUAMBCJ:/mnt/f/Software/Cmdr_mini# |
```


任务 2: 更新数据源

参考: http://blog.csdn.net/qq_33746131/article/details/52966547

参考: <http://blog.csdn.net/sunnyliqian/article/details/50179915>

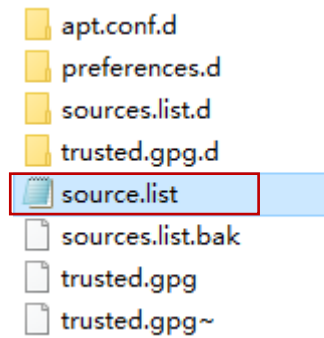
分析: 如果不更新数据源, 有时候用 Cmdr 下载东西就下载不下来, 提示 HTTP sends request, ...Not Found, 或者 connecting time ...超时等。总之, 由于我们在国内, 最好更新数据源, 比如用阿里, 搜狐, 清华的, 他们的数据源直接链接国外, 我们从而达到下载目的。

步骤 1: 输入 **cd /etc/apt/** 切换目录

```
 Cmdr
root@DESKTOP-NUAMBCJ:/mnt/f/Software/Cmdr_mini# cd /etc/apt/
```

步骤 2: 输入 **cp sources.list sources.list.bak** 备份, 步骤 3 可以帮你看到;

步骤 3: 定位 **C:\Users\user\AppData\Local\lxss\rootfs\etc\apt**, 删除 **source.list** 这个文件;

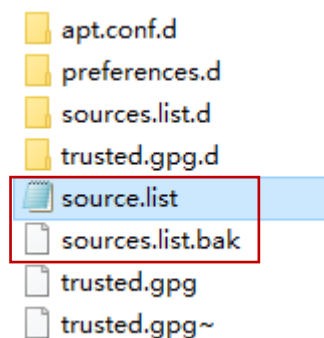


步骤 4: 输入 `sudo vim source.list` 进入 vim (Cmder 切换到 vim 环境), 按下 `I` 键进入编辑状态, 输入数据源, 完了 `Esc`, `--Insert--` 状态消失, 输入 `:x` (冒号 `x`, 即保存退出);

步骤 5: 回到 Cmder 终端环境后, 输入 `sudo apt-get update` 即可。

```
root@DESKTOP-NUAMBCJ:/mnt/f/Software/Cmder_mini# cd /etc/ap
root@DESKTOP-NUAMBCJ:/etc/apt# sudo cp sources.list source
root@DESKTOP-NUAMBCJ:/etc/apt# sudo vim source.list
root@DESKTOP-NUAMBCJ:/etc/apt# sudo apt-get update
Hit:1 http://storage.googleapis.com/bazel-apt stable InRel
Hit:2 http://archive.ubuntu.com/ubuntu xenial InRelease
Get:3 http://security.ubuntu.com/ubuntu xenial-security In
Get:4 http://archive.ubuntu.com/ubuntu xenial-updates InRe
Get:5 http://security.ubuntu.com/ubuntu xenial-security/un
Get:6 http://archive.ubuntu.com/ubuntu xenial-updates/main
Get:7 http://archive.ubuntu.com/ubuntu xenial-updates/univ
Fetched 1,640 kB in 12s (131 kB/s) http://mirrors.aliyun.c
Reading package lists... Done
root@DESKTOP-NUAMBCJ:/etc/apt#
```

小技巧: 再次打开下列文件夹 `C:\Users\user\AppData\Local\lxss\rootfs\etc\apt`, 你会看到 `source.list` 和 `sources.list.bak` 已经生成了, 但 `source.list` 是重新生成的;



此外, 如果你安装了记事本或者 Notepad++, 可以打开 `source.list`, 会发现里面的你刚刚添加的数据源 `deb` 开头的数据源, 至于 `#Ali Yun Database` 这个注释, 是我自己添加的以防将来还要添加数据源, 以示区分。你可以选择不添加, 都无所谓。

```
1 #Ali·Yun·database
2 deb·http://mirrors.aliyun
3 deb·http://mirrors.aliyun
  multiverse
4 deb·http://mirrors.aliyun
  multiverse
5 deb·http://mirrors.aliyun
  multiverse
6 deb·http://mirrors.aliyun
  multiverse
7 deb-src·http://mirrors.aliyun
8 deb-src·http://mirrors.aliyun
```

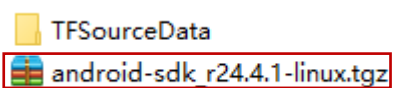
任务 3： 安装 SDK

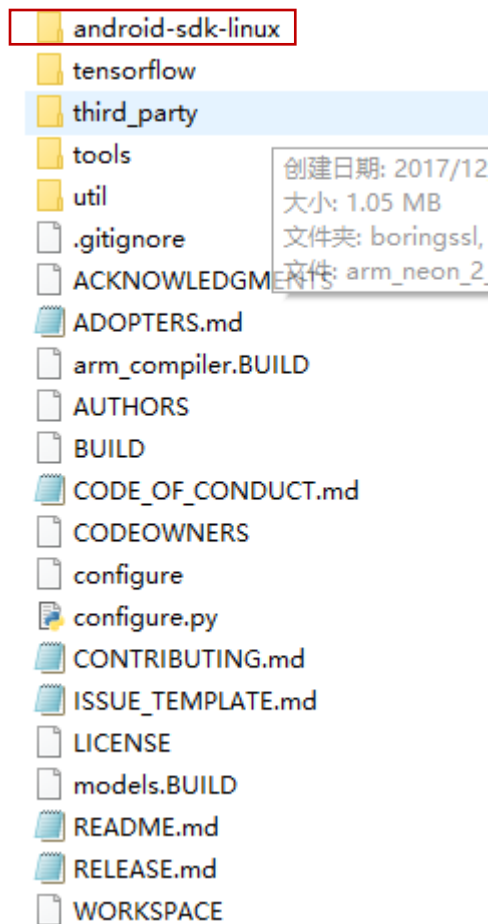
步骤 1： 执行 `cd /home` 切换到/home 的目录下；

步骤 2： 执行 `wget https://dl.google.com/android/android-sdk_r24.4.1-linux.tgz` 由于数据源没问题，所以直接下载到/home 目录下；

步骤 3： 执行 `mkdir TFSourceData` 创建 TFSourceData 目录；

步骤 4： 执行 `tar xvzf android-sdk_r24.4.1-linux.tgz -C /home/TFSourceData` 解压到/home 目录下的 TFSourceData 文件夹中；





步骤 5: 更新和安装 SDK

执行: `cd /home/TFSourceData/android-sdk-linux` 注意路径问题, 别出错

执行: `sudo tools/android update sdk --no-ui` 此路径下才能正确执行更新和安装 SDK

```
root@DESKTOP-NUAMBCJ:/home# cd /home/TFSourceData/android-sdk-linux
root@DESKTOP-NUAMBCJ:/home/TFSourceData/android-sdk-linux# sudo tools/android update sdk --no-ui
```

任务 4: 安装 NDK

步骤 1: 输入 `cd /home` 切换目录

步骤 2: 输入 `wget https://dl.google.com/android/repository/android-ndk-r12b-linux-x86_64.zip`
开始下载, 自动下载到/home 目录下

```
SDK: command not found
root@DESKTOP-NUAMBCJ:/home/TFSourceData/android-sdk-linux# cd /home
root@DESKTOP-NUAMBCJ:/home# wget https://dl.google.com/android/repository/android-ndk-r12b-linux-x86_64.zip
```

如果网络不好, 会出现


```
root@DESKTOP-NUAMBCJ:/home# wget http://dl.google.com/android/
--2017-12-21 17:59:49-- http://dl.google.com/android/repositor
Resolving dl.google.com (dl.google.com)... 203.208.43.99, 203.
Connecting to dl.google.com (dl.google.com)|203.208.43.99|:80.
Connecting to dl.google.com (dl.google.com)|203.208.43.110|:80
Connecting to dl.google.com (dl.google.com)|203.208.43.103|:80
Connecting to dl.google.com (dl.google.com)|203.208.43.104|:80
HTTP request sent, awaiting response... 404 Not Found
2017-12-21 18:00:52 ERROR 404: Not Found.
```

试试直接复制 https://dl.google.com/android/repository/android-ndk-r12b-linux-x86_64.zip

到浏览器下载；或者 http://dl.google.com/android/repository/android-ndk-r12b-linux-x86_64.zip
注意二者的区别，把 **s** 去掉

404. That's an error.

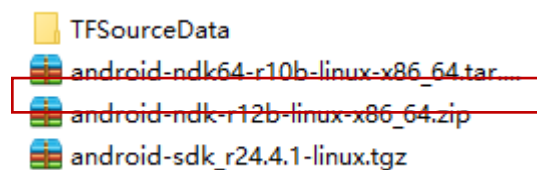
That's all we know.

说明这个网址资源不存在了，没法下载。此时百度或者 google，发现下面两个链接

<https://developer.android.google.cn/ndk/downloads/index.html>

<http://blog.csdn.net/davebobo/article/details/52446986>

都有资源，我建议用上面的，因为上面那个类似官网，下面是 CSDN 某个博客的。下载 old version r12b 后(高版本可能和 Bazel 不兼容，至少 Ubuntu14 不能用 r12 以上版的)，把 zip 文件放到/home 目录下



由于我先装的 r16 版本，所以踩了坑，这里用 linux 安装 r16 版例子来教你用 zip 解压

执行：`unzip android-ndk-r16b-linux-x86_64.zip -d /home/TFSourceData` 解压到 TFSourceData

```
Cmdr
root@DESKTOP-NUAMBCJ:/mnt/f/Software/Cmdr_mini# cd /home
root@DESKTOP-NUAMBCJ:/home# unzip android-ndk-r16b-linux-x86_64.zip -d /home/TFSourceData
```

出现下面这个说明 zip 不是 linux 默认安装的，需要自己安装一遍

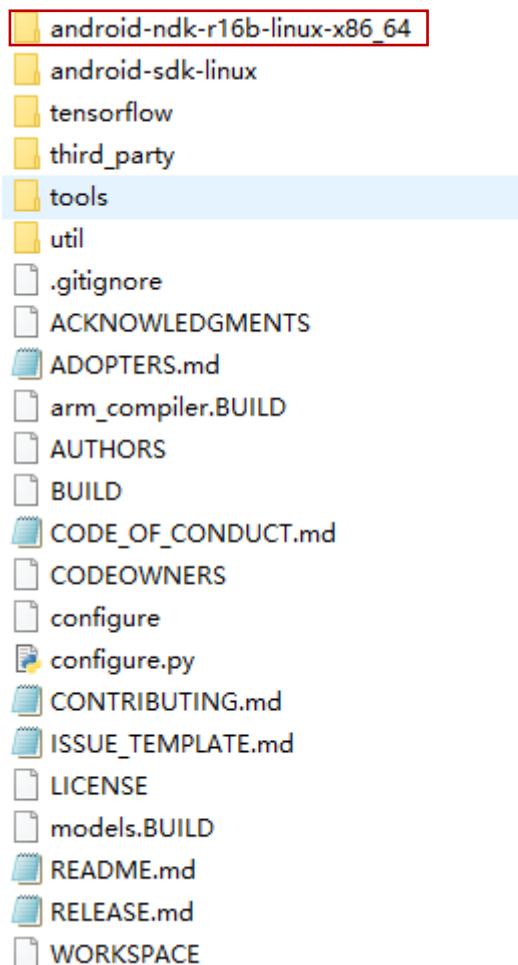
```
Cmdr
root@DESKTOP-NUAMBCJ:/mnt/f/Software/Cmdr_mini# cd /home
root@DESKTOP-NUAMBCJ:/home# unzip android-ndk-r16b-linux-x86_64.zip -d /home/TFSourceData
WARNING:root:could not open file '/etc/apt/sources.list'
The program 'unzip' is currently not installed. You can install it by typing:
apt install unzip
You will have to enable the component called 'main'
root@DESKTOP-NUAMBCJ:/home#
```

网上有 `yum install -y unzip zip` 和 `sudo apt-get install zip` 两种安装方式，但前者不适用于我们的 ubuntu 或 WSL 子系统，况且 Cmdr 提示用 `apt install unzip`，所以使用后者，执行：

`sudo apt-get install zip` 出现 E，即 Error，可能是 `/home` 目录下没有权限

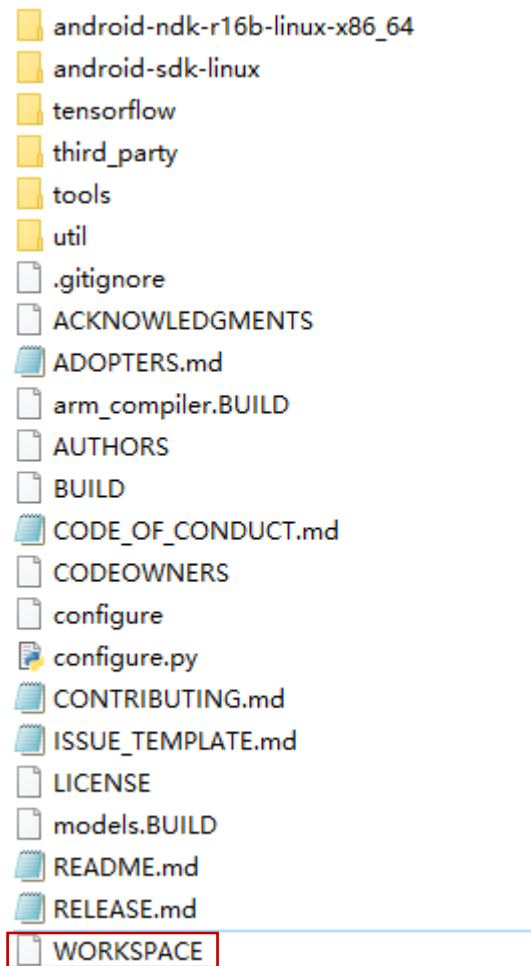
```
root@DESKTOP-NUAMBCJ:/home# sudo apt-get install zip
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package zip
```

执行：`cd -` 切换目录，再次执行 `sudo apt-get install zip` 也不行，说明不是权限，百度发现是数据源更新问题，需要把数据源再扩充下，这里我没有试，直接手动解压到 TFSourceData 文件下了，后面有时间我再重新添加数据源。

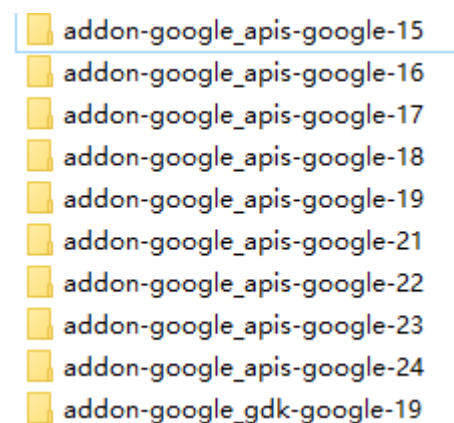


任务 5: 修改 workspace 文件

步骤 1: 定位到 C:\Users\user\AppData\Local\lxss\home\TFSourceData 文件夹, 打开 WORKSPACE 文件



修改见参考: <http://blog.csdn.net/cxq234843654/article/details/70861155> 总之里面代码的修改要和 NDK 和 JDK 版本相对应, 仔细找找文件夹, 都能找到答案。SDK api_level 我的最高是 23, 定位 C:\Users\user\AppData\Local\lxss\home\TFSourceData\android-sdk-linux\add-ons


















最后得到

```
android_sdk_repository(  
    . . . . name = "androidsdk",  
    . . . . #my highest api_level = 24  
    . . . . api_level = 23,  
    . . . . # Ensure that you have the build_tools_version be  
    . . . . # SDK manager as it updates periodically.  
    . . . . build_tools_version = "27.0.2",  
    . . . . # Replace with path to Android SDK on your system  
    . . . . path = "/home/TFSourcedata/android-sdk-linux",  
)
```

```
android_ndk_repository(  
    . . . . name="androidndk",  
    . . . . path="/home/TFSourcedata/android-ndk-r12b-linux-x86_64/android-ndk-r12b",  
    . . . . # This needs to be 14 or higher to compile TensorFlow.  
    . . . . # Please specify API level to >= 21 to build for 64-bit  
    . . . . # architectures or the Android NDK will automatically select biggest  
    . . . . # API level that it supports without notice.  
    . . . . # Note that the NDK version is not the API level.  
    . . . . api_level=14)
```

步骤 2: 修改 build.gradle

定位 C:\Users\user\AppData\Local\lxss\home\TFSourcedata\tensorflow\examples\android 找到
build.gradle 文件,

	assets	20
	bin	20
	gradle	20
	jni	20
	res	20
	sample_images	20
	src	20
	_init_.py	20
	AndroidManifest.xml	20
	BUILD	20
	build.gradle	20
	download-models.gradle	20
	gradlew	20
	gradlew.bat	20
	README.md	20

用记事本或 notepad++ 打开

```
android {
    compileSdkVersion 23
    buildToolsVersion '26.0.2'

    if (nativeBuildSystem == 'cmake') {
        defaultConfig {
            applicationId = 'org.tensorflow.demo'
            minSdkVersion 21
            targetSdkVersion 23
            ndk {
                abiFilters "${cpuType}"
            }
        }
    }
}
```

将 buildToolVersion 更改为 27.0.2

```

android {
    compileSdkVersion 23
    buildToolsVersion '27.0.2'

    if (nativeBuildSystem == 'cmake') {
        defaultConfig {
            applicationId = 'org.tensorflow.demo'
            minSdkVersion 21
            targetSdkVersion 23
            ndk {
                abiFilters "${cpuType}"
            }
        }
    }
}

```

步骤 3: 将上面的 AndroidManifest.xml 打开, 确保和上面 SDK 的一致。

```

<uses-sdk
    android:minSdkVersion="21"
    android:targetSdkVersion="23" />

```

第四部分: WSL 下生成 apk 文件, 导入手机运行

步骤 1: 编译 so 库和 jar 包

参考: <http://blog.csdn.net/cxq234843654/article/details/71171293>

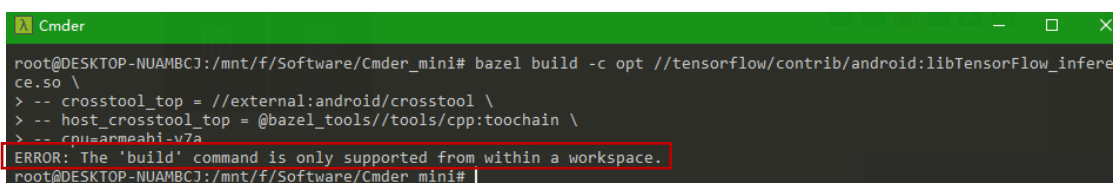
输入下面代码:

```

bazel build -c opt //tensorflow/contrib/android:libtensorflow_inference.so \
  -- crosstool_top=//external:android/crosstool \
  -- host_crosstool_top=@bazel_tools//tools/cpp:toolchain \
  -- cpu=armeabi-v7a

```

出现错误, 原因在于路径不对



```

Cmake
root@DESKTOP-NUAMBCJ:/mnt/f/Software/Cmake_mini# bazel build -c opt //tensorflow/contrib/android:libTensorFlow_inference.so \
> -- crosstool_top = //external:android/crosstool \
> -- host_crosstool_top = @bazel_tools//tools/cpp:toolchain \
> -- cpu=armeabi-v7a
ERROR: The 'build' command is only supported from within a workspace.
root@DESKTOP-NUAMBCJ:/mnt/f/Software/Cmake_mini#

```

输入: `source ~/tensorflow/bin/activate`, 注意路径切换到 TFSourceData 下执行红框代码

```

root@DESKTOP-NUAMBCJ:/mnt/f/Software/Cmder_mini# clear
root@DESKTOP-NUAMBCJ:/mnt/f/Software/Cmder_mini# source ~/tensorflow/bin/activate
(tensorflow) root@DESKTOP-NUAMBCJ:/mnt/f/Software/Cmder_mini# bazel build //TFSourceData/examples/android:tensorflow_demo
ERROR: The 'build' command is only supported from within a workspace.
(tensorflow) root@DESKTOP-NUAMBCJ:/mnt/f/Software/Cmder_mini# cd /home/TFSourceData
(tensorflow) root@DESKTOP-NUAMBCJ:/home/TFSourceData# bazel build //tensorflow/examples/android:tensorflow_demo

```

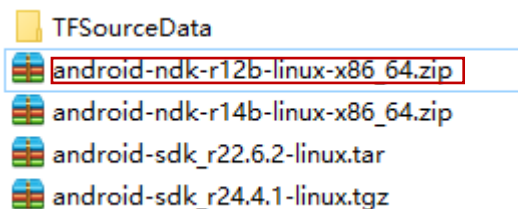
出现 ndk 版本不对

```

ERROR: in target '//external:android/crosstool': no such package '@androidndk//': /home/TFSourceData/android-ndk-r12b-linux-x86_64/android-ndk-r12b (No such file or directory)
ERROR: Analysis of target '//tensorflow/examples/android:tensorflow_demo' failed; build aborted: in target '//external:android/crosstool': no such package '@androidndk//': /home/TFSourceData/android-ndk-r12b-linux-x86_64/android-ndk-r12b (No such file or directory)
INFO: Elapsed time: 3.914s
FAILED: Build did NOT complete successfully (7 packages loaded)

```

重新下载 ndk r12b 版本即可，注意解压到 TFSourceData 目录下



再次执行 bazel build...没有 jar 包

```

ERROR: missing input file '@androidsdk//:tools/support/annotations.jar'
ERROR: /home/TFSourceData/tensorflow/examples/android/BUILD:64:1: //tensorflow/examples/android:tensorflow_demo failed to build
Use --verbose_failures to see the command lines of failed build steps.
ERROR: /home/TFSourceData/tensorflow/examples/android/BUILD:64:1: 1 input file failed
INFO: Elapsed time: 42.330s, Critical Path: 1.82s
FAILED: Build did NOT complete successfully

```

解决方案参考：

<https://stackoverflow.com/questions/24438748/fail-to-find-annotations-jar-after-updating-to-android-ndk-r12b>, 下载好后，安装到 TFSourceData\android-sdk-linux\tools\support 可能没有相关文件夹名字，自己建立文件夹，命名即可，比如我命名 support 文件夹，将下载好+命名的好的.jar 文件复制过去就行。

<http://central.maven.org/maven2/com/google/android/annotations/4.1.1.4/annotations-4.1.1.4.jar>

Renamed it in annotation.jar and moved to: android-sdk/tools/support/

Restarting Eclipse everything will be fine.

再次执行 bazel build....出现

```
CONFLICT: asset:WORKSPACE is provided with ambiguous priority from:
external/speech_commands/WORKSPACE
external/mobile_ssd/WORKSPACE
CONFLICT: asset:WORKSPACE is provided with ambiguous priority from:
external/stylize/WORKSPACE
external/speech_commands/WORKSPACE
CONFLICT: asset:WORKSPACE is provided with ambiguous priority from:
external/mobile_ssd/WORKSPACE
external/inception5h/WORKSPACE
ERROR: /root/.cache/bazel/_bazel_root/60d76693e697bfaa612d25696d1cfc23/external/nsync/BUILD:441:1: C++ compilation of
rule '@nsync//:nsync_cpp' failed (Exit 1)
```

conflict 没事，主要是 error

备注：Linux 训练好的模型移植到手机主要参考以下链接

<http://blog.csdn.net/cxq234843654/article/details/71171293>

<http://blog.csdn.net/cxq234843654/article/details/70861155>

http://blog.csdn.net/masa_fish/article/details/54585537

http://blog.csdn.net/masa_fish/article/details/54097796

<https://www.cnblogs.com/greentomlee/p/5656467.html>

<https://www.jianshu.com/p/f63320075e10>(重点)

<http://nilhcem.com/android/custom-tensorflow-classifier>(英文参考)

