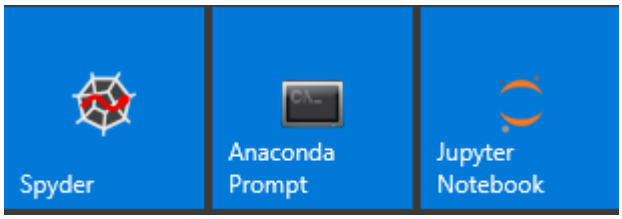
一、 CPU 版(win 10)

安装 anaconda,

下载地址
https://www.anaconda.com/download/
或选择国内清华镜像 https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/

ananconda 安装完成

好之后，会生成多个程序，主要有 3 个，



安装 tensorflow cpu 版

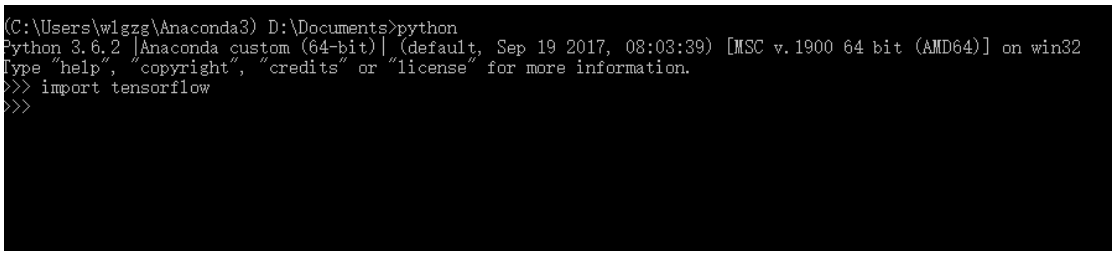
打开 Anaconda Prompt，输入：

```
conda install -c conda-forge tensorflow
```

中途 n/ y,输入 y
(若安装失败，再重复一次)

验证

打开 Anaconda Prompt，输入 python
再输入 import tensorflow,若无其他情况，即为安装正常



后续升级

打开 Anaconda Prompt，输入：

```
conda install -c conda-forge tensorflow
```

即可完成升级

二、 GPU 版(win 10)

要求

有独立英伟达显卡，且显卡计算能力不低于 3

显卡能力查询： https://developer.nvidia.com/cuda-gpus

GeForce GTX 1050	6.1	GeForce GTX 965M	5.2
GeForce GTX TITAN X	5.2	GeForce GTX 960M	5
GeForce GTX TITAN Z	3.5	GeForce GTX 950M	5.0
GeForce GTX TITAN黑色	3.5	GeForce 940M	5.0
GeForce GTX TITAN	3.5	GeForce 930M	5.0
GeForce GTX 980 Ti	5.2	GeForce 920M	3.5

软件准备

vs

若计算中具有

Microsoft Visual C++ 2012 Redistributable (x86) - 11.0....	Microsoft Corporation	2010/11/5	17.3 MB	11.0.60610.1
Microsoft Visual C++ 2013 Redistributable (x64) - 12.0....	Microsoft Corporation	2017/10/14	20.5 MB	12.0.30501.0
Microsoft Visual C++ 2013 Redistributable (x86) - 12.0....	Microsoft Corporation	2017/10/14	17.1 MB	12.0.30501.0

则无需处理，否则需要安装 vs2015(http://dl.msdn.com/sg/vs2015.com_chs.iso?t=a6943b4c-ccc7-4c11-9da2e8114bef4194&e=1512615707&h=9c061eaaed8667a27a38b90ad55e56b2)

cuda8.0

https://developer.nvidia.com/compute/cuda/8.0/prod/local_installers/cuda_8.0.44_win10-exe

cudnn 6.0

https://developer.nvidia.com/compute/machine-learning/cudnn/secure/v6/prod/8.0_20170307/cudnn-8.0-windows10-x64-v6.0-zip

软件安装

依次将上述软件安装（所有软件按默认安装）；
在 cuda 安装完成后，将 cudnn 解压后的文件复制到 cuda 的安装路径中，一般为：
C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v8.0
安装完成。

安装 tensorflow cpu 版

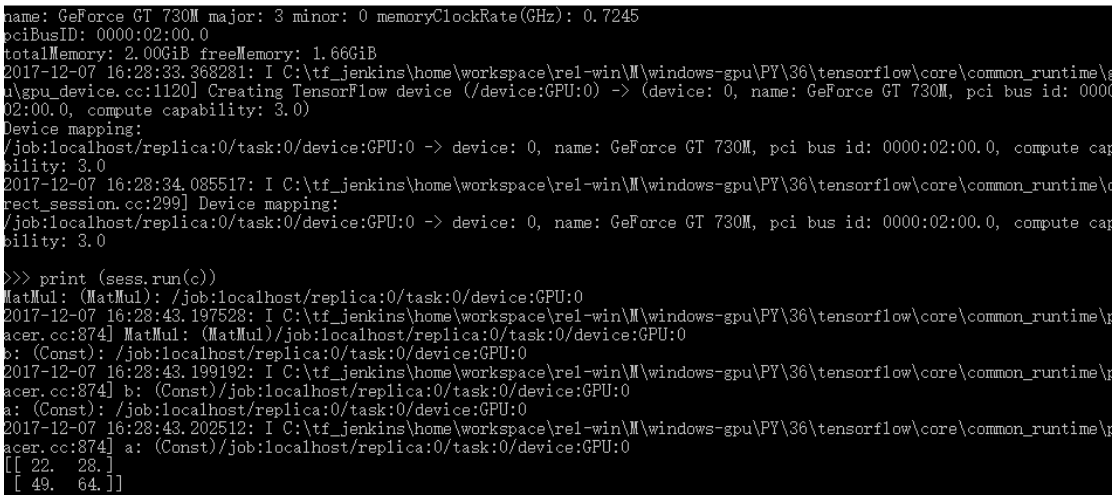
打开 Anaconda Prompt，输入：
`conda install -c aaronzs tensorflow-gpu 中途 n/ y,输入 y`
(若安装失败，再重复一次)

验证

打开 Anaconda Prompt，输入：python
再次输入：

```
import tensorflow as tf
a = tf.constant([1.0, 2.0, 3.0, 4.0, 5.0, 6.0], shape=[2, 3], name='a')
b = tf.constant([1.0, 2.0, 3.0, 4.0, 5.0, 6.0], shape=[3, 2], name='b')
c = tf.matmul(a, b)
# 新建 session with log_device_placement 并设置为 True.
sess = tf.Session(config=tf.ConfigProto(log_device_placement=True))
# 运行这个 op.
print (sess.run(c))
```

结果如下



CPU 与 GPU 计算性能对比

```
In [3]: tf.reset_default_graph()

In [4]: runfile('C:/Users/wlgzg/Desktop/test.py', wdir='C:/Users/wlgzg/Desktop')
Iter 0, Minibatch Loss= 2.247075, Training Accuracy= 0.68750
Iter 50, Minibatch Loss= 0.488892, Training Accuracy= 0.83594
Optimization Finished!
Running time: 975.5876262737131 Seconds

In [5]: tf.reset_default_graph()

In [6]: runfile('C:/Users/wlgzg/Desktop/test.py', wdir='C:/Users/wlgzg/Desktop')
Iter 0, Minibatch Loss= 1.218077, Training Accuracy= 0.78906
Iter 50, Minibatch Loss= 0.178999, Training Accuracy= 0.93750
Optimization Finished!
Running time: 186.88791837744725 Seconds

In [7]: tf.reset_default_graph()

In [8]: runfile('C:/Users/wlgzg/Desktop/test.py', wdir='C:/Users/wlgzg/Desktop')
Iter 0, Minibatch Loss= 2.441218, Training Accuracy= 0.74219
Iter 50, Minibatch Loss= 0.239160, Training Accuracy= 0.92188
Optimization Finished!
Running time: 186.95831634669048 Seconds
```

结果 1 为强制使用 cpu，结果 2 为强制使用 gpu,结果 3 为默认使用 gpu