

如何在一台离线主机上配置深度学习环境？

原创 空字符 月来客栈 2020-09-08 09:58

收录于合集

《Tensorflow入门指南》 11 # 《效率工具》 24

1 前言

什么样的环境配置需要整整两天呢？答，在一台内网服务器上且没有内部Python可用源的情况下安装CUDA驱动+Tensorflow需要整整两天。

前段时间同事申请的一台主机上周到了，说让笔者来配置一下环境。拿到账号后第一时间登陆上去，并使用了 `nvidia-smi` 来查看是否配有显卡。结果提示找不到该命令，初步判断GPU凉了。今天早上同事走过来问我环境弄好了没，速度怎么样？笔者答到，配置好了，速度一般般。同事又问到，和你笔记本比呢？我说差不多，然后补充了一句，没GPU快不起来。结果同事回了一句，主机配了GPU。纳尼？配了？于是笔者赶快用命令查看了一下：

```
lspci | grep -i vga
```

结果只是输出了这么一句提示：

```
VGA compatible controller: ASPEED Technology, Inc. ASPEED Graphics Family (rev 41)
```

然而并没有发现GPU。笔者心想，难道是命令不对？于是又试试了这个命令：

```
lspci | grep -i nvidia
```

结果终于发现了新大陆，提示装有一块Tesla P100的显卡。

```
3D controller: NVIDIA Corporation GP100GL [Tesla P100 PCIe 16GB]
```

然后接下来的两天就泡在了GPU环境的配置上。

2 环境配置

由于主机在内网中，所以无法通过第三方源来安装各种包；但更无语的是居然没有供内部使用的Python源。这就意味着后续所有需要用到的依赖包都得手动从网上下载，然后再拷贝到主机中进行离线安装。接下来，笔者就开始一步步的介绍如何进行环境的安装。同时，这里默认已经安装好了对应的Python，没有的话可自行安装。

2.1 版本选择

在正式安装之前，笔者首先来大致介绍一下CUDA，CUDA Toolkit，cuDNN三者间的关系，这样也便于后续的安装理解。

- CUDA基本就等于CUDA Toolkit，CUDA Toolkit里面包含了较新的显卡驱动以及其它一些所需要用到的东西，所以安装了CUDA Toolkit后也就不需要再去单独安装驱动了；
- cuDNN是英伟达专门为深度学习所开发的用于深度神经网络的GPU加速库，像Tensorflow、Pytorch之类的深度学习框架要使用到GPU加速，就需要用这个cuDNN库。

在下载安装包之前，最重要的一个工作就是确定所需要安装的CUDA Toolkit版本号。怎么确定呢？最好的方法就是以你所需要的Tensorflow版本或者Pytorch版本来反推CUDA Toolkit的版本。在这个网站[1]的最后面，可以看到Tensorflow官方给出的版本依赖关系：

GPU

| 版本 | Python 版本 | 编译器 | 构建工具 | cuDNN | CUDA |
|-----------------------|-------------|-----------|--------------|-------|------|
| tensorflow-2.1.0 | 2.7、3.5-3.7 | GCC 7.3.1 | Bazel 0.27.1 | 7.6 | 10.1 |
| tensorflow-2.0.0 | 2.7、3.3-3.7 | GCC 7.3.1 | Bazel 0.26.1 | 7.4 | 10.0 |
| tensorflow_gpu-1.14.0 | 2.7、3.3-3.7 | GCC 4.8 | Bazel 0.24.1 | 7.4 | 10.0 |
| tensorflow_gpu-1.13.1 | 2.7、3.3-3.7 | GCC 4.8 | Bazel 0.19.2 | 7.4 | 10.0 |

图 1. Tensorflow与CUDA版本对应图

由于笔者一开始并没有注意到这个问题，直接选择安装了10.1版本的CUDA，并且也配套安装了7.6版本的cuDNN。最后的结果就是在Tensorflow1.11上调用GPU的时候显示缺少 `libcudnn.so.9` 这个文件。看到提示这个问题的时候笔者才意识到一开的版本选择出现了问题，只有CUDA 9.0才支持Tensorflow1.11。但是看到1.14的Tensorflow也支持10.0版本的CUDA，于是就侥幸的试了一下安装好的环境能不能在Tensorflow 1.14上运行。好在经测试后发现可以，不然前面的功夫就白费了。

因此最好的做法就是一开始就根据需要使用Tensorflow或者Pytorch版本来选择CUDA的版本（也就是CUDA Toolkit 对应的版本）。

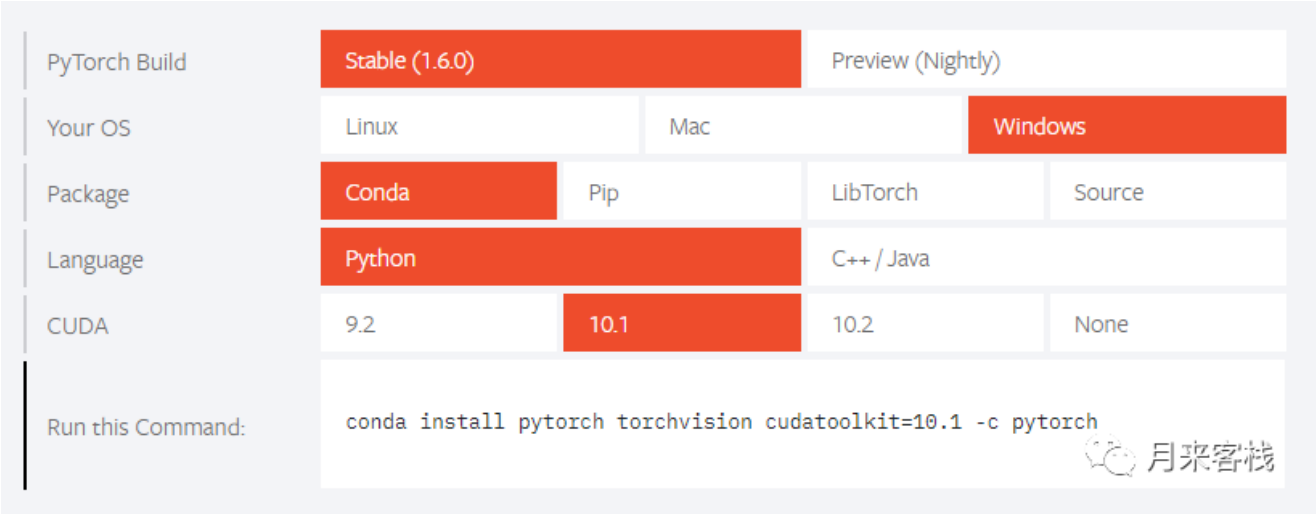


图 2. Pytorch与CUDA版本对应图

2.2 CUDA Toolkit安装

在确定好CUDA Toolkit版本后， 点击链接进入下载CUDA下载页面[2]， 可以看到最新的为11.0版本的CUDA Toolkit。

Latest Release

[CUDA Toolkit 11.0 Update1 \(Aug 2020\), Versioned Online Documentation](#)

Archived Releases

[CUDA Toolkit 11.0 \(May 2020\), Versioned Online Documentation](#)

[CUDA Toolkit 10.2 \(Nov 2019\), Versioned Online Documentation](#)

[CUDA Toolkit 10.1 update2 \(Aug 2019\), Versioned Online Documentation](#)

[CUDA Toolkit 10.1 update1 \(May 2019\), Versioned Online Documentation](#)

[CUDA Toolkit 10.1 \(Feb 2019\), Online Documentation](#)

[CUDA Toolkit 10.0 \(Sept 2018\), Online Documentation](#)



图 3. CUDA版本图

根据我们上面我们上面确定的版本号（本文为10.1）， 点击图3中对应版本的链接即可跳转到下一个页面， 如图4所示。

CUDA Toolkit 10.1 original Archive

Select Target Platform ⓘ

Click on the green buttons that describe your target platform. Only supported platforms will be shown.

Operating System

WindowsLinuxMac OSX

Architecture ⓘ

x86_64ppc64le

Distribution

FedoraOpenSUSERHELCentOSSLESUbuntu

Version

76

Installer Type ⓘ

runfile (local)rpm (local)rpm (network)

月来客栈

图 4. CUDA Toolkit 下载页面

在图4所在的页面中，我们根据当前主机系统的情况来选择合适的安装包。同时可以发现，在这个页面中还有对于主机系统的选择。笔者这里使用到的是CentOS 7.7版本的系统，假如笔者使用的是CentOS 8.0版本的系统也就意味着不能安装CUDA Toolkit 10.1，同时根据图1可知，也就意味着不能使用1.x版本的Tensorflow。

因此，总结起来就是，在选择CUDA Toolkit版本的时候，既要考虑到该版本是否支持后续所要使用的深度学习框架的版本，同时还要考虑到是否支持当前主机系统的版本。

在下载完成安装包后（大约2.4G）将其拷贝到主机上，然后执行如下命令即可（如果是选择的其它Linux发行版的系统，以及其它形式的安装包，按照该页面所给出的安装方法安装就行）：

```
sudo sh cuda_10.1.105_418.39_linux.run
```

大约30秒左右就能看到如下所示的步骤，输入 `accept` 回车继续。

```
End User License Agreement
-----

Preface
-----

The Software License Agreement in Chapter 1 and the Supplement
in Chapter 2 contain license terms and conditions that govern
the use of NVIDIA software. By accepting this agreement, you
agree to comply with all the terms and conditions applicable
to the product(s) included herein.

NVIDIA Driver

Description

This package contains the operating system driver and

Do you accept the above EULA? (accept/decline/quit): 月来客栈
accept
```

图 5. CUDA Toolkit安装示意图

紧接着，就会弹出如下所示的一个示意图，直接选择Install，按回车即可。

```
CUDA Installer
- [X] Driver
    [X] 418.39
+ [X] CUDA Toolkit 10.1
    [X] CUDA Samples 10.1
    [X] CUDA Demo Suite 10.1
    [X] CUDA Documentation 10.1
Install
Options 月来客栈
```

图 6. CUDA Toolkit安装图

在这之后，CUDA Toolkit也就安装成功了，通过 `nvidia-smi` 命令，即可查看相关显卡的信息。如果输入命令后提示找不到该命令，可以重启试试，如果还是不行则表示没有安装成功。

2.3 cuDNN安装

在安装完CUDA Toolkit后，点击[3]进入cuDNN下载页面：

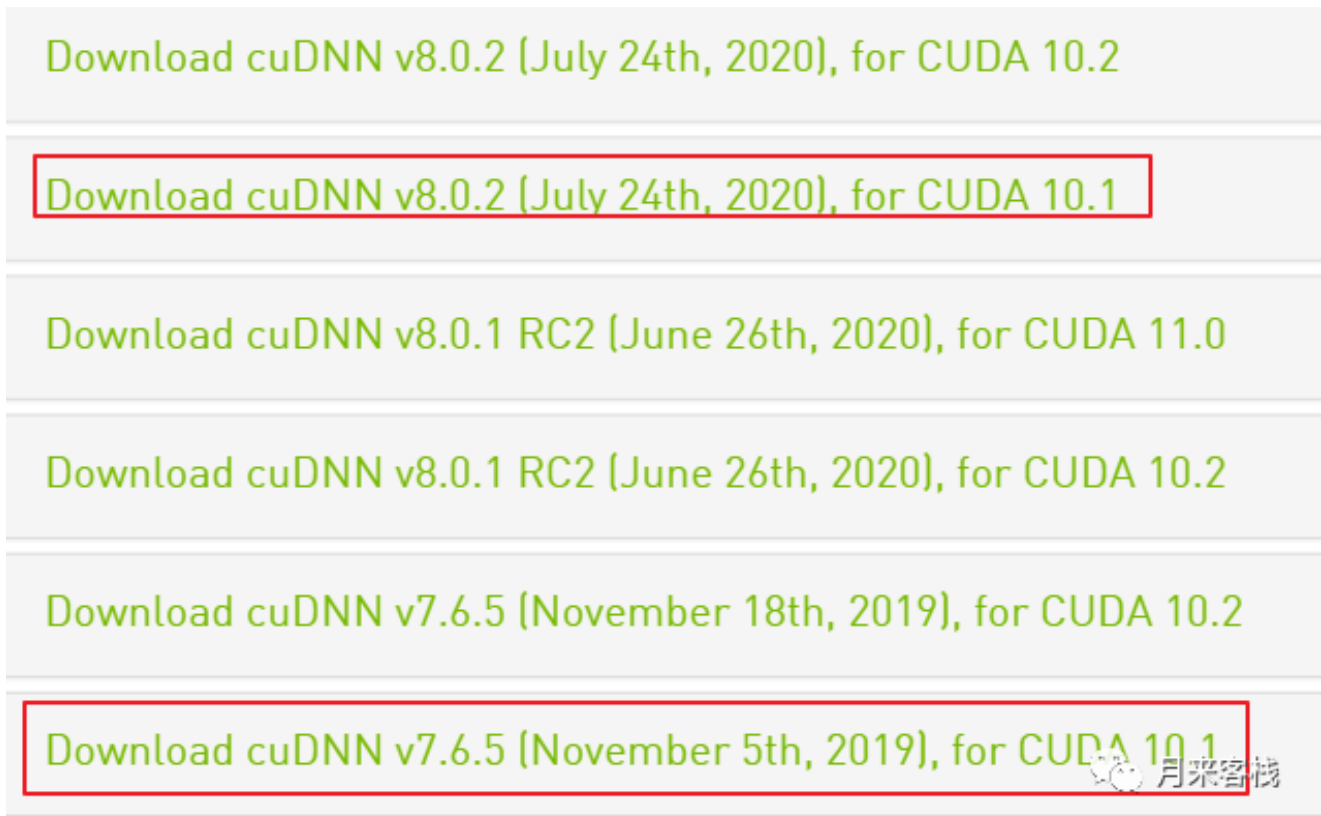


图 7. cuDNN版本选择

可以发现有不同版本的cuDNN都可以支持CUDA 10.1，那我们到底应该选择哪一个呢？根据图1所示，cuDNN的版本大致应该是7.4左右，因此笔者在这里选择的是7.6.5的版本。你也可以接着往下看找到7.4版本的cuDNN进行下载。通常来说，只要大的版本号一致，就没啥太大的问题，当然如果不行那就得重新安装。

在确定好版本号，点击相应链接进入到下载页面。但麻烦点的是，在下载cuDNN时需要注册一个账号登录后才能下载。在下载好安装包后同样将其拷贝到主机上，然后以如下步骤进行安装：

(1) 切换到安装包的目录并解压安装包

```
tar -xzf cudnn-10.1-linux-x64-v7.6.5.32.tgz
```

(2) 进入解压后的文件夹中并复制相关文件到cuda的安装目录

```
sudo cp include/cudnn.h /usr/local/cuda-10.1/include  
cp lib64/libcudnn* /usr/local/cuda-10.1/lib64
```

注意，如果你安装的是其它版本的CUDA Toolkit，则 `local` 下的文件夹不一定是 `cuda-10.1`，要灵活处理。

（3）修改文件权限

```
sudo chmod a+r /usr/local/cuda-10.1/include/cudnn.h
sudo chmod a+r /usr/local/cuda-10.1/lib64/libcudnn*
```

到此，cuDNN就安装完成了。

2.4 Tensorflow框架安装

由于是离线安装，所以我们首先要去网站[4]下载对应的GPU安装包，具体步骤如下：

（1）搜索安装包



图 8. Tensorflow-gpu下载

（2）选择安装包

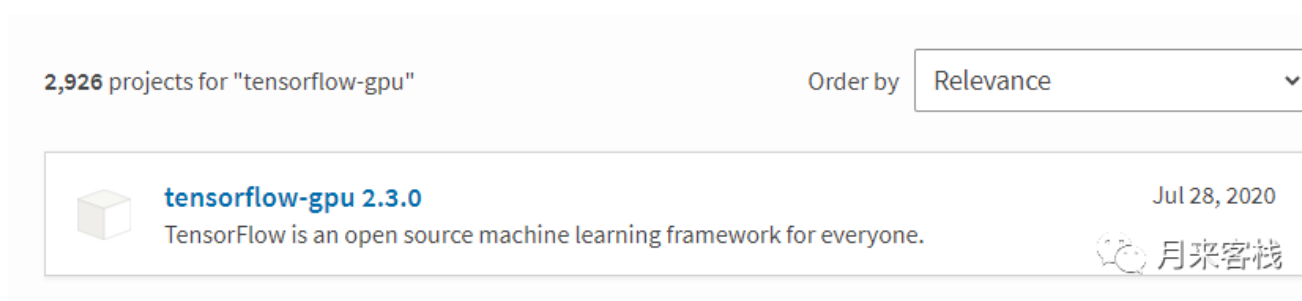


图 9. 选择安装包

（3）查找指定版本



图 10. 查找版本

(4) 选择指定版本安装包

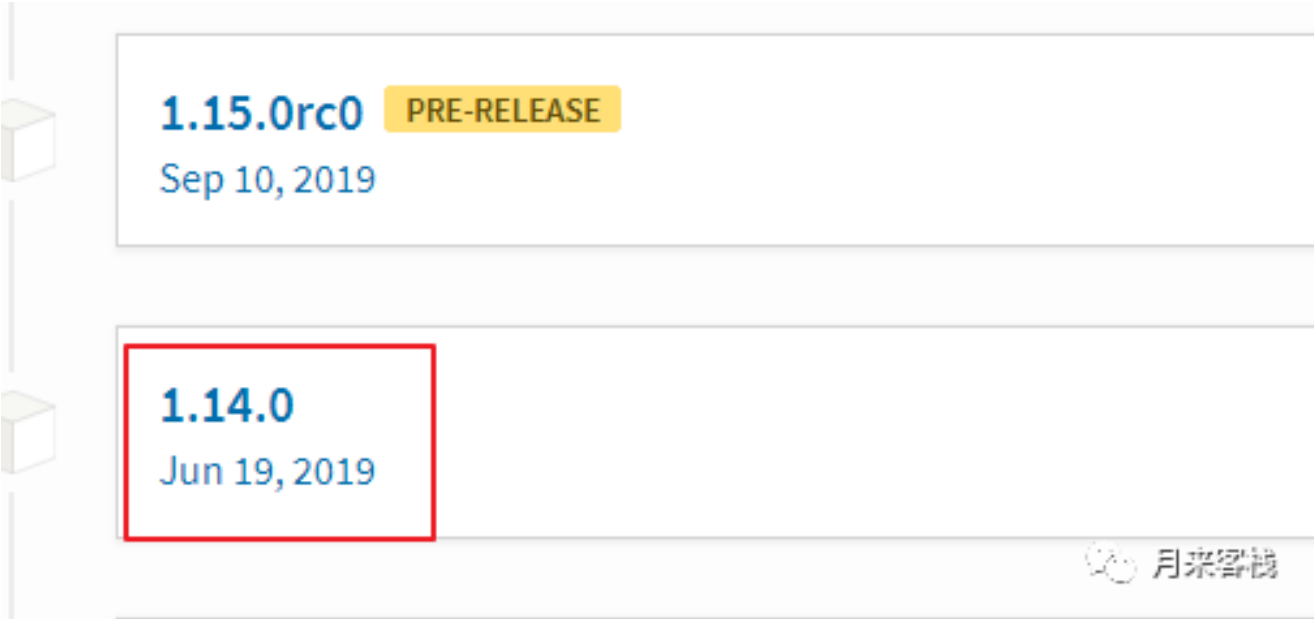


图 11. 选择指定版本

(5) 下载对应版本

tensorflow-gpu 1.14.0

```
pip install tensorflow-gpu==1.14.0
```

TensorFlow is an open source machine learning framework

Navigation

Project description

Release history

Download files

Project description

TensorFlow is an
allows easy depl
clusters of server

Originally develop
comes with stron

图 12. 下载对应版本

(6) 选择对应平台安装包

| | | | |
|--|-------|------|--------------|
| tensorflow_gpu-1.14.0-cp36-cp36m-manylinux1_x86_64.whl (377.0 MB) | Wheel | cp36 | Jun 19, 2019 |
| tensorflow_gpu-1.14.0-cp36-cp36m-win_amd64.whl (287.7 MB) | Wheel | cp36 | Jun 19, 2019 |
| tensorflow_gpu-1.14.0-cp37-cp37m-manylinux1_x86_64.whl (377.1 MB) | Wheel | cp37 | Jun 19, 2019 |

图 13. 选择对应平台安装包

其中 `cp36` 表示python3.6。

这样，我们便下载得到了tensorflow-gpu 1.14版本的离线安装包。同时，对于 `xxx.whl` 格式的安装包我们通过 `pip install xxx.whl` 安装即可。但是，在安装tensorflow-gpu的过程中，你并不会一次就安装成功。这是因为tensorflow-gpu会依赖于其它的包，例如tensorboard、

2.5 测试

```
import tensorflow as tf

with tf.device('/cpu:0'):
    a = tf.constant([1,2.,3.],shape=[3],name='a')
    b = tf.constant([1,2.,3.],shape=[3],name='b')

with tf.device('/gpu:0'):
    c = a + b

sess = tf.Session()
cc = sess.run(c)

print(cc)
```

```
Could not load dynamic library 'libcudart.so.10.0'; dlerror: libcudart.so.10.0: cannot open shared
Could not load dynamic library 'libcusparses.so.10.0'; dlerror: libcusparses.so.10.0: cannot open s
.....
```

```
export CUDA_HOME=/usr/local/cuda
export LD_LIBRARY_PATH=/usr/local/cuda/lib64:$LD_LIBRARY_PATH:/usr/local/cuda/lib64:/usr/local/cuda/lib64
export PATH=/usr/local/cuda/bin:$PATH
```

10/12

3 总结

在这篇文章中，笔者介绍了如何在一台断网的主机上安装深度学习环境。其主要思想就是找到我们需要的，并且匹配的安装包进行下载，然后拷贝到主机上进行离线安装。同时，需要清楚的是上述整个安装过程其实主要就归结为三个包的安装：CUDA Toolkit、cuDNN以及Tensorflow-gpu。当然在安装的过程中可能出现一些意想不到的报错，但只需要搜索相关错误提示一般都能找到对应的解决方案。

本次内容就到此结束，感谢您的阅读！如果你觉得上述内容对你有所帮助，欢迎关注并传播本公众号！若有任何疑问与建议，请添加笔者微信'nulls8'加群进行交流。青山不改，绿水长流，我们月来客栈见！

引用

[1] Tensorflow与CUDA版本查询：

https://www.tensorflow.org/install/source#common_installation_problems

[2] CUDA Toolkit下载页面：<https://developer.nvidia.com/cuda-toolkit-archive>

[3] cuDNN下载页面：<https://developer.nvidia.com/rdp/cudnn-archive>

[4] Python安装包下载：<https://pypi.org/>

[5]<https://www.freesion.com/article/9403156806/>

引用

[\[1\]你还在使用破解版的Pycharm?](#)

[\[2\]你从未如此顺滑地进行笔记](#)

[\[3\]Pycharm配置Anaconda虚拟环境](#)

[\[4\]优雅的安装和使用Anaconda](#)

发现人生乐趣

人生中总有些东西值得分享



长按二维码关注
@月来客栈

点击「在看」让有用的内容被更多人看见！



收录于合集 #《Tensorflow入门指南》 11

下一篇 · 《Tensorflow1.x入门指南》 API使用索引目录，持续更新中.....

喜欢此内容的人还喜欢

无名之辈，我是谁
月来客栈

