

主讲人：正点原子团队

硬件平台：正点原子ATK-DLRV1126开发板

版权所有：广州市星翼电子科技有限公司

资料下载：[www.openedv.com/docs/index.html](http://www.openedv.com/docs/index.html)

教学平台：[www.yuanzige.com](http://www.yuanzige.com)

天猫店铺：[zhengdianyuanzi.tmall.com](http://zhengdianyuanzi.tmall.com)

技术论坛：[www.openedv.com/forum.php](http://www.openedv.com/forum.php)

公众平台：正点原子



## ■ Windows下搭建训练模型环境

- 1, 环境概述
- 2, 安装Anaconda3
- 3, 安装显卡驱动
- 4, 安装TensorFlow-GPU 2.5.0 (方式一)
- 5, 安装PyTorch-GPU 1.13.0 (方式一)
- 6, 安装CUDA Toolkit 11.2 (方式二)
- 7, 安装cuDNN 8.1 (方式二)
- 8, 配置环境变量 (方式二)
- 9, 安装TensorFlow-GPU 2.5.0 (方式二)

## 1、环境概述

在前面课程，我们在Ubuntu上安装了RKNN Toolkit用于将其它深度学习框架下的模型转换为RKNN模型，那个时候搭建的环境（安装RKNN Toolkit），我们叫做**模型转换环境**。

从本节课程开始，我们即将搭建训练模型的环境，注意：**模型训练环境**和**模型转换环境**是不一样的。



两种环境分别在不同的虚拟环境下，不建议放在一个虚拟环境，因为两种环境的依赖库可能会有冲突

版本依赖情况：

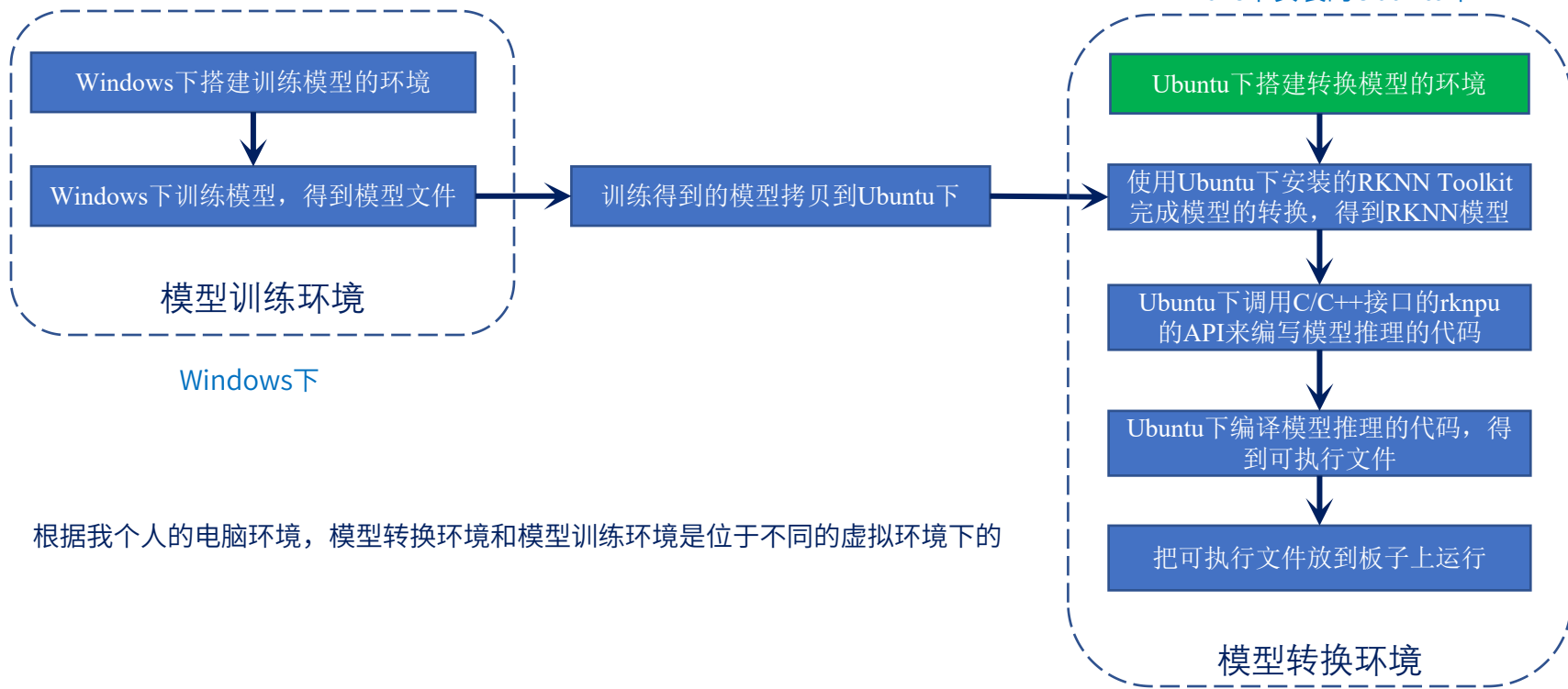
RKNN Toolkit的模型量化功能依赖于 TensorFlow 库，RKNN Toolkit的TensorFlow模型加载功能依赖于TensorFlow。

RKNN Toolkit的PyTorch模型加载功能，依赖于 PyTorch。

综上所述，建议：**模型转换环境下的TensorFlow和PyTorch版本，尽量和模型训练环境下的TensorFlow和PyTorch版本一致，若不一致，容易出问题。**

## 1、环境概述

根据本人的电脑环境，整个模型部署的操作流程大概是这样的：



## 1、环境概述

### ◆本人电脑环境

操作系统：Windows 10

硬件配置：

CPU：Intel i7-11800H

GPU：NVIDIA GeForce RTX 3080 Ti（显存12GB，英伟达官网上显示[算力8.6](#)）

内存：32G

硬盘：

VMWare下安装的Ubuntu不支持CUDA，没办法用到GPU，所以本人训练模型是在Windows下完成的。

若在PC安装的是Ubuntu实体机，则Ubuntu可以调用到CUDA，可以使用GPU。

## 1、环境概述

◆ TensorFlow-GPU和CUDA、cuDNN以及Python版本确定

四者对应关系请参考：

[https://www.tensorflow.org/install/source\\_windows](https://www.tensorflow.org/install/source_windows)

Version	Python version	Compiler	Build tools	cuDNN	CUDA
tensorflow_gpu-2.10.0	3.7-3.10	MSVC 2019	Bazel 5.1.1	8.1	11.2
tensorflow_gpu-2.9.0	3.7-3.10	MSVC 2019	Bazel 5.0.0	8.1	11.2
tensorflow_gpu-2.8.0	3.7-3.10	MSVC 2019	Bazel 4.2.1	8.1	11.2
tensorflow_gpu-2.7.0	3.7-3.9	MSVC 2019	Bazel 3.7.2	8.1	11.2
tensorflow_gpu-2.6.0	3.6-3.9	MSVC 2019	Bazel 3.7.2	8.1	11.2
tensorflow_gpu-2.5.0	3.6-3.9	MSVC 2019	Bazel 3.7.2	8.1	11.2
tensorflow_gpu-2.4.0	3.6-3.8	MSVC 2019	Bazel 3.1.0	8.0	11.0
tensorflow_gpu-2.3.0	3.5-3.8	MSVC 2019	Bazel 3.1.0	7.6	10.1
tensorflow_gpu-2.2.0	3.5-3.8	MSVC 2019	Bazel 2.0.0	7.6	10.1

冷知识：

GeForce RTX 30系显卡，算力达到8.x，  
目前支持的是CUDA 11.0及以上版本

根据本人电脑环境，以及根据CUDA选择11.2版本，TensorFlow-GPU选择2.5.0版本，cuDNN选择8.1版本，Python可选择3.6~3.9版本。

## 1、环境概述

◆ PyTorch-GPU和CUDA、Python、torchvision、版本确定

PyTorch-GPU、Python和torchvision对应关系请参考：

<https://github.com/pytorch/vision#installation>

PyTorch-GPU、Python和CUDA关系以及安装方法（在线安装）：

<https://pytorch.org/blog/deprecation-cuda-python-support/>

<https://pytorch.org/get-started/locally/>

<https://pytorch.org/get-started/previous-versions/>

PyTorch-GPU历史版本安装包下载地址（离线安装）：

[https://download.pytorch.org/whl/torch\\_stable.html](https://download.pytorch.org/whl/torch_stable.html)

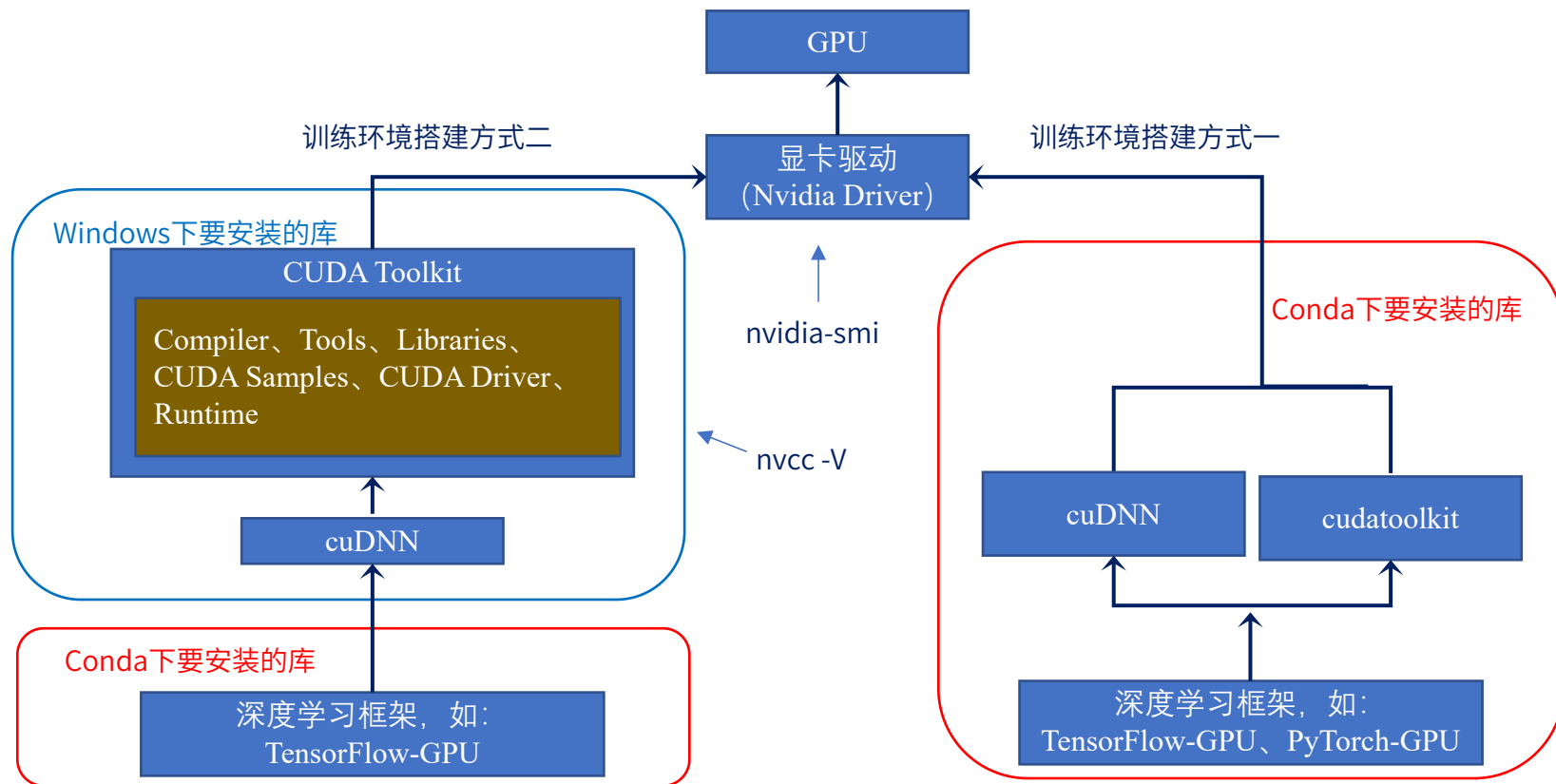
根据PyTorch官网的介绍，PyTorch-GPU选择可以1.13.0版本（本人测试暂时没问题）

为了避免TensorFlow-GPU和PyTorch-GPU的环境发生冲突，分别在两个虚拟环境下安装TensorFlow-GPU和PyTorch-GPU。

PyTorch-GPU版本以及TensorFlow-GPU版本可以根据项目情况进行更改。



## 1、环境概述



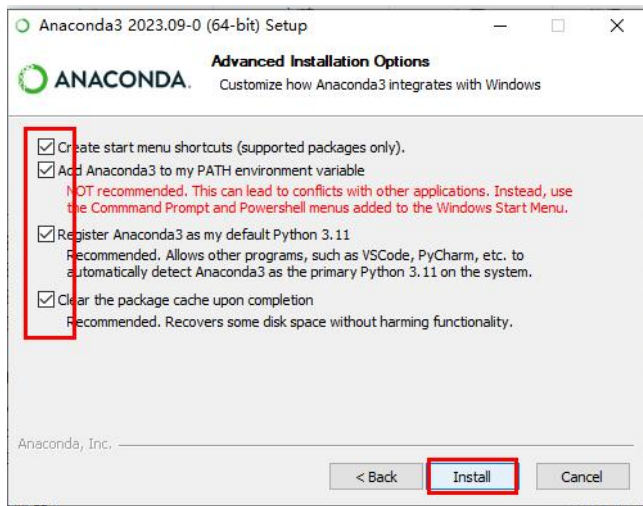
## 2、安装Anaconda

Anaconda官网：

<https://www.anaconda.com/download>

注意事项：

- 注意勾选自动配置环境变量PATH，选择将Anaconda3添加到环境变量里，否则后期只能手动配置环境变量；
- Anaconda安装完毕后，在使用Anaconda3之前，最好重启电脑；



## 3、安装显卡驱动

下载NVIDIA的显卡驱动：

<https://www.nvidia.cn/Download/Find.aspx?lang=cn>

注意，根据个人使用的点卡型号选择对应版本的显卡驱动，建议安装最新版本的显卡驱动（因为省事，后期不用更新）

### NVIDIA 驱动程序下载

官方高级驱动搜索 | NVIDIA

产品类型:

GeForce

产品系列:

GeForce RTX 30 Series

产品家族:

GeForce RTX 3080 Ti

操作系统:

Windows 10 64-bit

Windows 驱动程序类型:

DCH ?

语言:

Chinese (Simplified)

最新:

推荐/认证 ?

本人电脑使用的显卡是NVIDIA GeForce RTX 3080 Ti

搜索

名称

GeForce Game Ready 驱动程序 WHQL

GeForce Game Ready 驱动程序 WHQL

GeForce Game Ready 驱动程序 WHQL

版本

546.01

545.92

545.84

发布日期

31.10.2023

26.10.2023

17.10.2023

## 4、安装TensorFlow-GPU 2.5.0（方式一）

- 创建、进入虚拟环境

```
conda create -n py3.6-tensorflow-gpu-2.5.0 python=3.6
conda env list
conda activate py3.6-tensorflow-gpu-2.5.0
conda list
```

- 安装依赖库cudatoolkit 11.2.2和cudnn 8.1.0

```
conda install cudatoolkit=11.2.2 -c conda-forge
conda install cudnn=8.1.0 -c conda-forge
```

- 安装TensorFlow-GPU 2.5.0

```
pip install tensorflow-gpu==2.5.0 -i https://pypi.douban.com/simple/
```

- 查看TensorFlow-GPU 2.5.0是否可用

```
import tensorflow as tf
import sys
print(sys.version)
print(tf.version)
gpu_list=tf.config.list_physical_devices('GPU')
print("\ngpu_list:",gpu_list)
gpu_available=tf.test.is_gpu_available()
print("\nTensorFlow-gpu is available? = ",gpu_available)
```

- 训练一个模型看看TensorFlow-GPU 2.5.0是否可用

## 5、安装PyTorch-GPU 1.13.0（方式一）

- 创建、进入虚拟环境

```
conda create -n py3.8-pytorch-gpu-1.13.0 python=3.8
conda activate py3.8-pytorch-gpu-1.13.0
python -V
```

- 安装PyTorch-GPU 1.13.0

在线安装参考：<https://pytorch.org/get-started/previous-versions/>

在线安装命令：

```
// 可以使用conda安装
conda install pytorch==1.13.0 torchvision==0.14.0 torchaudio==0.13.0 pytorch-cuda=11.7 -c pytorch -c nvidia
// 或者使用pip安装
pip install torch==1.13.0+cu117 torchvision==0.14.0+cu117 torchaudio==0.13.0 --extra-index-url https://download.pytorch.org/whl/cu117
```

离线安装，安装包下载链接：[https://download.pytorch.org/whl/torch\\_stable.html](https://download.pytorch.org/whl/torch_stable.html)

离线安装命令：

```
// 先把安装包下载下来，再使用pip命令去安装
pip install torch-1.13.0+cu117-cp38-cp38-win_amd64.whl torchvision-0.14.0+cu117-cp38-cp38-win_amd64.whl torchaudio-0.13.0+cu117-cp38-cp38-win_amd64.whl
```

## 5、安装PyTorch-GPU 1.13.0（方式一）

### ● 验证安装的PyTorch-GPU 1.13.0

```
# 在虚拟环境py3.8-pytorch-gpu-1.13.0下执行如下命令检查PyTorch-GPU 1.13.0是否可用
python
import torch
import torchvision
print(torch.__version__)
print(torchvision.__version__)

print(torch.version.cuda)
print(torch.backends.cudnn.version())
print(torch.cuda.is_available())
```

以YOLOV5 V7.0版本训练为例子验证安装的PyTorch-GPU 1.13.0是否可用。

```
# 先安装依赖库
pip install -r requirements.txt -i https://pypi.douban.com/simple --trusted-host pypi.douban.com

# 为了节省时间，训练一个epoch看看
# 适用于cmd、Git Bash终端的命令：
python train.py --data coco128.yaml --weights " " --cfg yolov5s.yaml --img 640 --epoch 1

# 适用于Powershell终端的命令：
python train.py --data coco128.yaml --weights [string]::Empty --cfg yolov5s.yaml --img 640 --epoch 1
```

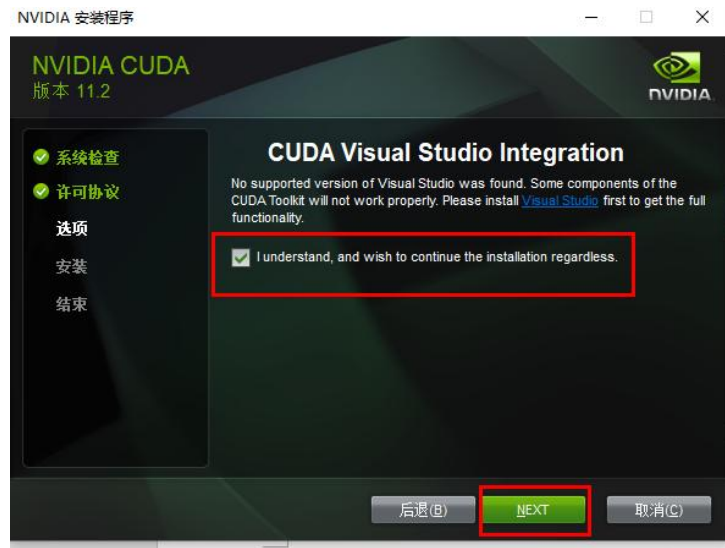
## 6、CUDA Toolkit 11.2（方式二）

CUDA下载链接：

<https://developer.nvidia.com/cuda-toolkit-archive>

选择默认安装就行（默认安装在C盘）；

安装过程选择“精简”安装即可；



## 7、安装cuDNN 8.1（方式二）

cuDNN下载链接：

<https://developer.nvidia.com/rdp/cudnn-download>

- 下载cuDNN需要英伟达官网账号（或者需要登录邮箱），自己注册一个即可；
- CUDA安装目录默认在C盘： C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.2
- 将下载下来的cuDNN包解压，然后将bin,lib,include文件夹中的文件复制粘贴到CUDA的安装目录下：

① 把cuDNN的bin目录下的文件复制到CUDA安装目录 bin目录下：

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.2\bin

② 把cuDNN的include下的所有文件，复制到CUDA安装目录的include下：

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.2\include

③ 把cuDNN的lib\x64下的所有文件，拷贝到CUDA安装目录的lib\x64下：

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.2\lib\x64



## 8、配置环境变量（方式二）

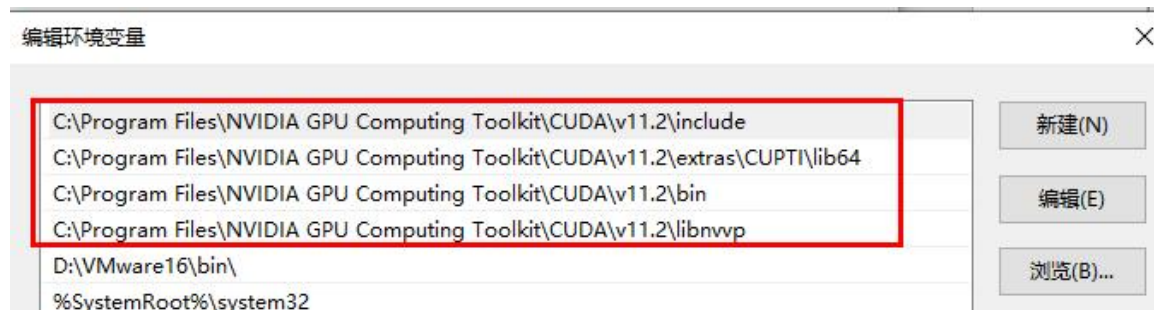
需要手动添加以下环境变量到系统变量里：

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.2\bin

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.2\libnvvp

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.2\extras\CUPTI\lib64

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.2\include



// 在cmd下执行如下2条命令，可查看CUDA驱动版本、CUDA Runtime版本以及GPU显卡的型号等信息

```
cd C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.2\extras\demo_suite
```

```
.\deviceQuery.exe
```

## 9、安装TensorFlow-GPU 2.5.0（方式二）

- 创建、进入虚拟环境

```
conda create -n py3.6-tf-gpu-2.5.0 python=3.6
conda env list
conda activate py3.6-tf-gpu-2.5.0
conda list
```

- 安装TensorFlow-GPU 2.5.0

```
pip install tensorflow-gpu==2.5.0 -i https://pypi.douban.com/simple/
pip install tensorflow-gpu==2.5.0 -i https://mirrors.aliyun.com/pypi/simple/
conda list
```

- 查看TensorFlow-GPU 2.5.0是否可用

```
import tensorflow as tf
import sys
print(sys.version)
print(tf.version)
gpu_list=tf.config.list_physical_devices('GPU')
print("\ngpu_list:",gpu_list)
gpu_available=tf.test.is_gpu_available()
print("\nTensorFlow-gpu is available? = ",gpu_available)
```

- 训练一个模型看看TensorFlow-GPU 2.5.0是否可用

遇到的问题：

在VSCode下的Jupyter Notebook调试Python代码时，虚拟环境下需要安装ipykernel后，方可在Jupyter Notebook 下运行代码：

```
// 使用pip命令安装（可通过-i选项指定源来安装）
```

```
pip install ipython ipykernel
```

```
// 若以上pip命令安装后还不行，可以试试使用conda命令安装看看：
```

```
// 安装ipykernel（注意是在哪个虚拟环境下安装）
```

```
conda install ipykernel --update-deps --force-reinstall
```

```
// 若要临时指定源来安装，可使用如下命令（命令中使用清华源，注意是在哪个虚拟环境下安装）
```

```
conda install ipykernel --update-deps --force-reinstall -c https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
```



版权所有：广州市星翼电子科技有限公司  
天猫店铺：<https://zhengdianyuanyi.tmall.com>