**特飞内部情报系统Windows**

**开发环境配置**

日期： 2019-12-18

部门： 特种飞机部

作者： 薛远奎

**目录**

[安装anaconda 1](#_Toc1043)

[Jupyter notebook配置 2](#_Toc12795)

[安装VS code 3](#_Toc3005)

[测试 4](#_Toc18742)

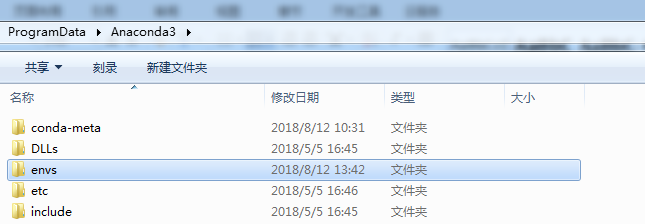
# 版本说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 日期 | 作者 | 内容变动 | 备注 |
| 1 | 2019.12.18 | 薛远奎 | 创建 | V1.0 |
| 2 | 2019.01.14 | 薛远奎 | 增加opencv安装指导  增加GPU安装指导 |  |
|  |  |  |  |  |

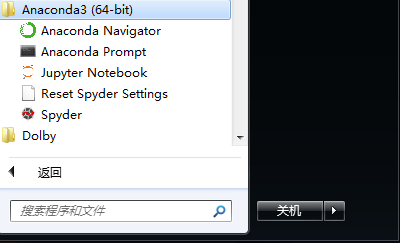
# Anaconda

## 安装anaconda

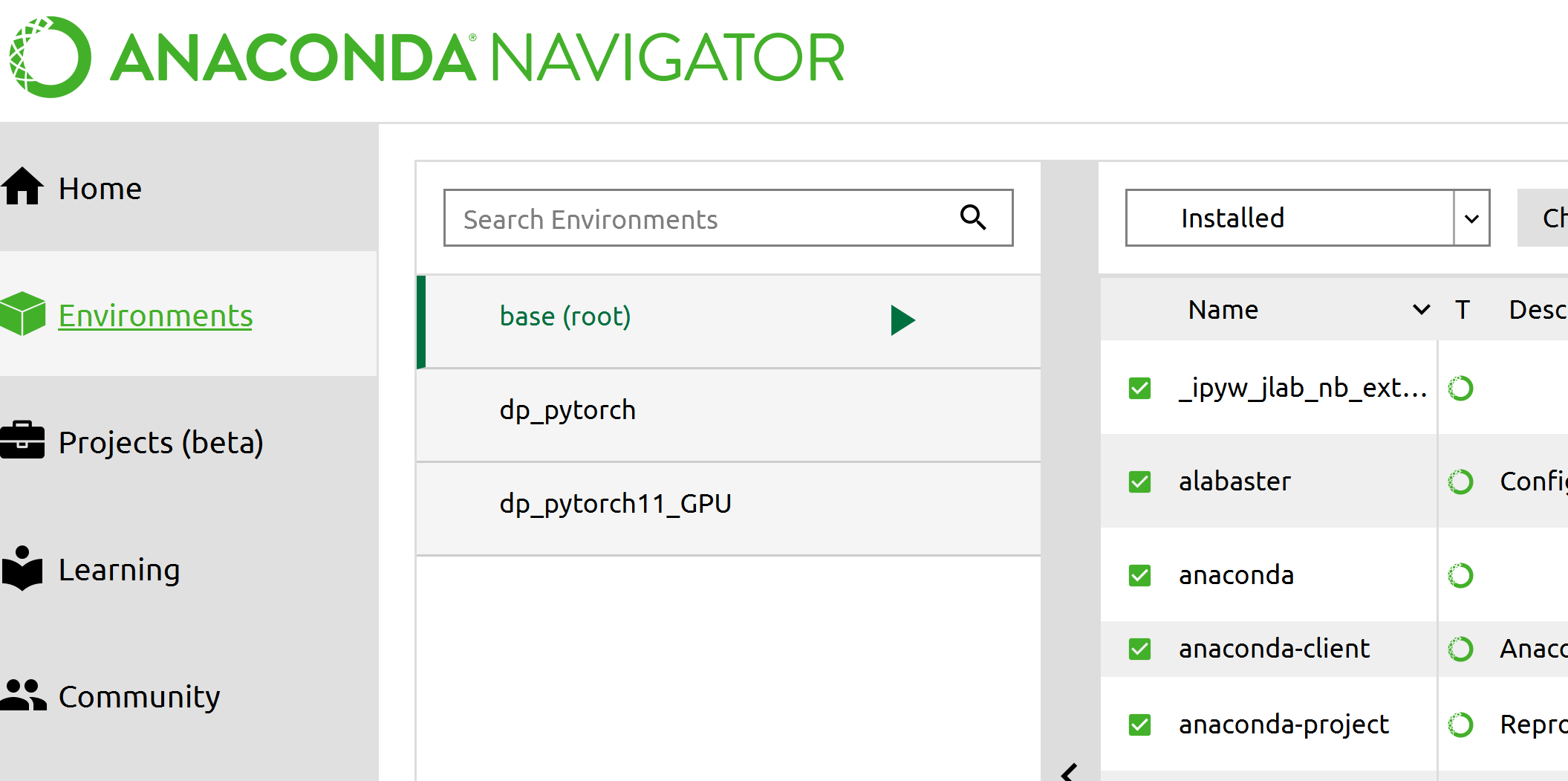
1. 双击Anaconda3-5.1.0-Windows-x86\_64.exe安装包，进行anaconda3的安装。
2. 找到anaconda的安装路径（例如我的安装路径为C:\ProgramData\Anaconda3），会在安装路径下看到一个envs文件夹。
3. 将dp\_pytorch11\_GPU文件夹拷贝到envs文件夹中。



1. 安装完成之后，点击计算机开始，找到Anaconda3并打开Anaconda Navigator。

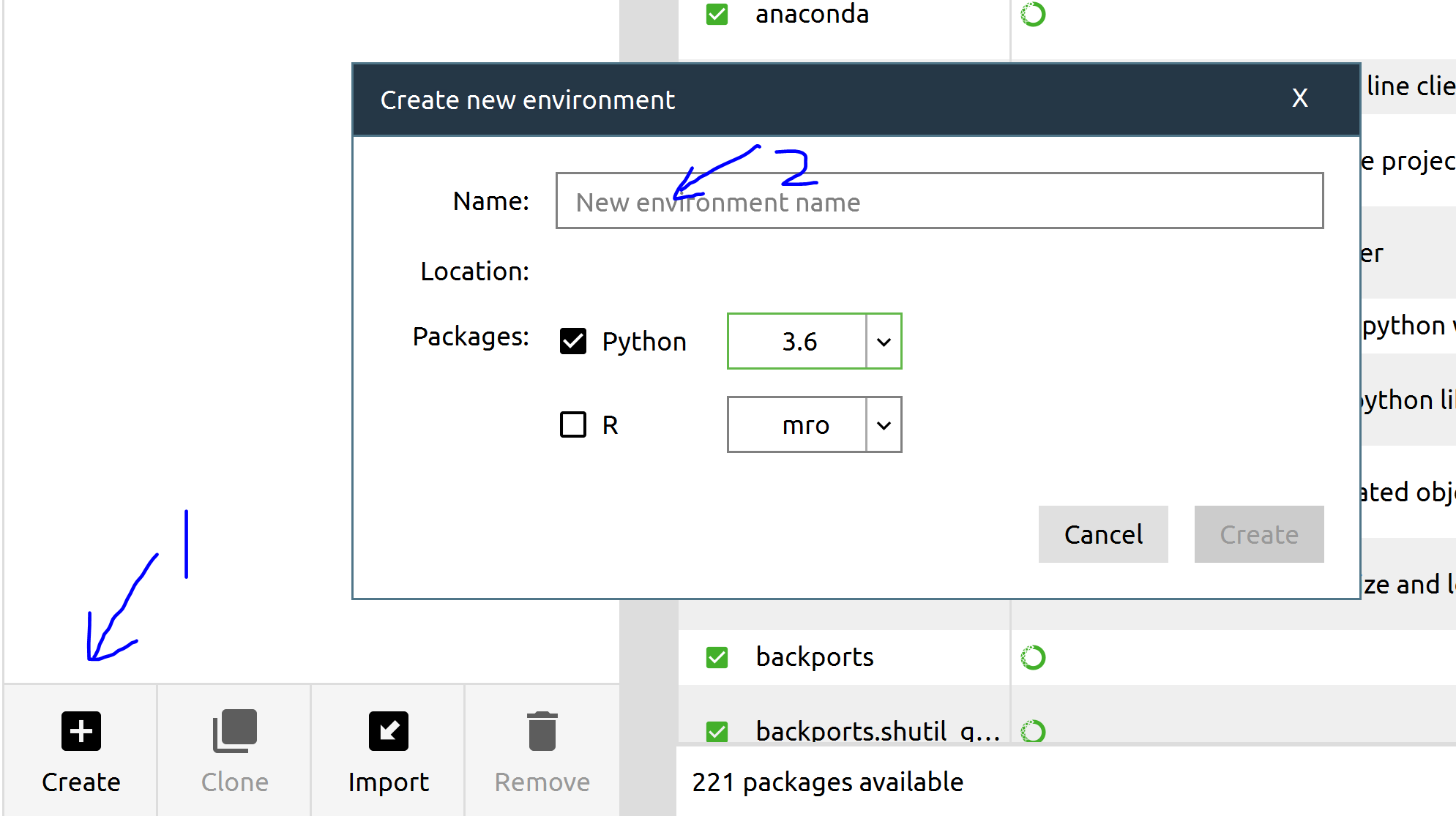


然后你应该会在左边菜单栏Environment中看到dp\_pytorch11\_GPU这个环境。



## 创建环境

如果你没有现成的环境包可用，那么需要新建一个，打开anaconda navigator，选择左边Environment点击Create，在弹出的对话框中输入环境名字即可。



备注：也可以完全在prompt中使用命令行新建环境，具体命令自查。

## Jupyter notebook配置

1. 打开anaconda prompt，输入activate dp\_pytorch11\_GPU激活pytorch的环境

|  |
| --- |
| C:\Users\Administrator>activate dp\_pytorch11\_GPU |

1. 输入如下命令安装ipykernel

|  |
| --- |
| (dp\_pytorch11\_GPU) C:\Users\Administrator> conda install ipykernel |

1. 将环境写入Notebook的kernel中

|  |
| --- |
| python -m ipykernel install --user --name dp\_pytorch11\_GPU --display-name "dp\_pytorch" |

1. 再次运行Jupyter Notebook，点击New新建ipynb文件时，就会让你选择conda环境啦。

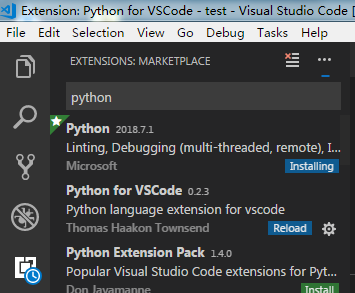


1. 点击New选择dp\_pytorch11\_GPU新建一个文件，并在第一个代码块中输入如下代码，按Shift+Enter快捷键运行：

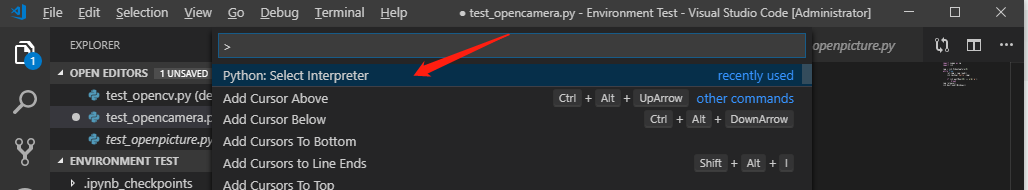
|  |
| --- |
| import torch  print('pytorch version:',torch.\_\_version\_\_)  x = torch.rand(5, 3)  print(x) |

# 安装VS code

1. 安装VSCodeSetup-x64-1.25.1.exe
2. 安装好之后打开Visual Studio Code，确保计算机连接好网络，选择最左边工具栏Extensions，安装Python和python for VSCode。



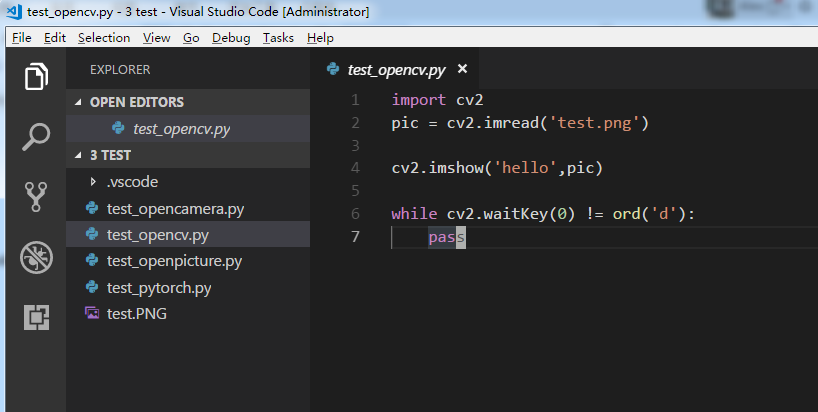
1. 使用快捷键ctrl + shift + p，打开编译器选择窗口。



然后选择你需要使用的编译器环境即可。

## 测试

1. 打开VS Code，选择File->Open Folder，打开一个测试程序的文件。
2. 打开对应的.py文件，F5运行



# Opencv

在Windows下安装python版本的opencv有两个选择，一种是下载别人编译好的安装包.whl，使用pip命令安装，另一种方式是使用在线安装，通过conda命令找到opencv的可用版本。

首先打开anaconda prompt激活需要配置的环境：

|  |
| --- |
| C:\Users\Administrator>activate dp\_pytorch11\_GPU |

使用如下命令，显示可用版本列表：

|  |
| --- |
| conda search opencv |
| (dp\_pytorch11\_GPU) C:\Users\admin>conda search opencv  Loading channels: done  Name Version Build Channel  opencv 3.3.1 py35h20b85fd\_1 defaults  opencv 3.3.1 py36h20b85fd\_1 defaults  opencv 3.4.1 py35h40b0b35\_3 defaults  opencv 3.4.2 py35h40b0b35\_0 defaults  opencv 3.4.2 py37h6fd60c2\_0 defaults  opencv 4.0.1 py38h2a7c758\_0 defaults |

使用如下命令，可以选择安装版本

|  |
| --- |
| conda install opencv=3.4.2 |

# Pytorch

如果没有pytorch环境，那么需要自己安装，本节假设你已经完成anaconda的安装，讲解如何安装pytorch。

## 安装依赖工具

如果不适用pytorch GPU版本仅仅适用CPU版本，那么这个步骤可以跳过，否则就需要安装依赖工具。

### Visual studio

首先需要下载安装visual studio，官网有community版本，可以免费安装。将C++相关的开发工具安装齐全！

### CUDA ToolKit

打开官网(https://developer.nvidia.com/cuda-downloads)，默认显示的是最新版的下载版本，可以点击Legacy Releases，选择Archived Releases下面的版本链接打开即可。

注意这里下载的版本要与后续pytorch对应的GPU版本统一，如果pytorch的CUDA版本是10.1，那么这里就需要下载10.1版本的CUDA ToolKit。

下载安装包之后，右键选择管理员身份打开，安装即可。安装完毕打开cmd命令行窗口，输入:nvcc -V，有如下输出：

|  |
| --- |
| C:\Users\admin>nvcc -V  nvcc: NVIDIA (R) Cuda compiler driver  Copyright (c) 2005-2018 NVIDIA Corporation  Built on Sat\_Aug\_25\_21:08:04\_Central\_Daylight\_Time\_2018  Cuda compilation tools, release 10.0, V10.0.130 |

### CuDNN

打开官网https://developer.nvidia.com/rdp/cudnn-download，选择与cudatoolkit对应的版本，下载之后进行如下操作。

(1)解压：会生成cuda/include、cuda/lib、cuda/bin三个目录；

(2)分别将cuda/include、cuda/lib、cuda/bin三个目录中的内容拷贝到C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v7.5对应的include、lib、bin目录下即可。

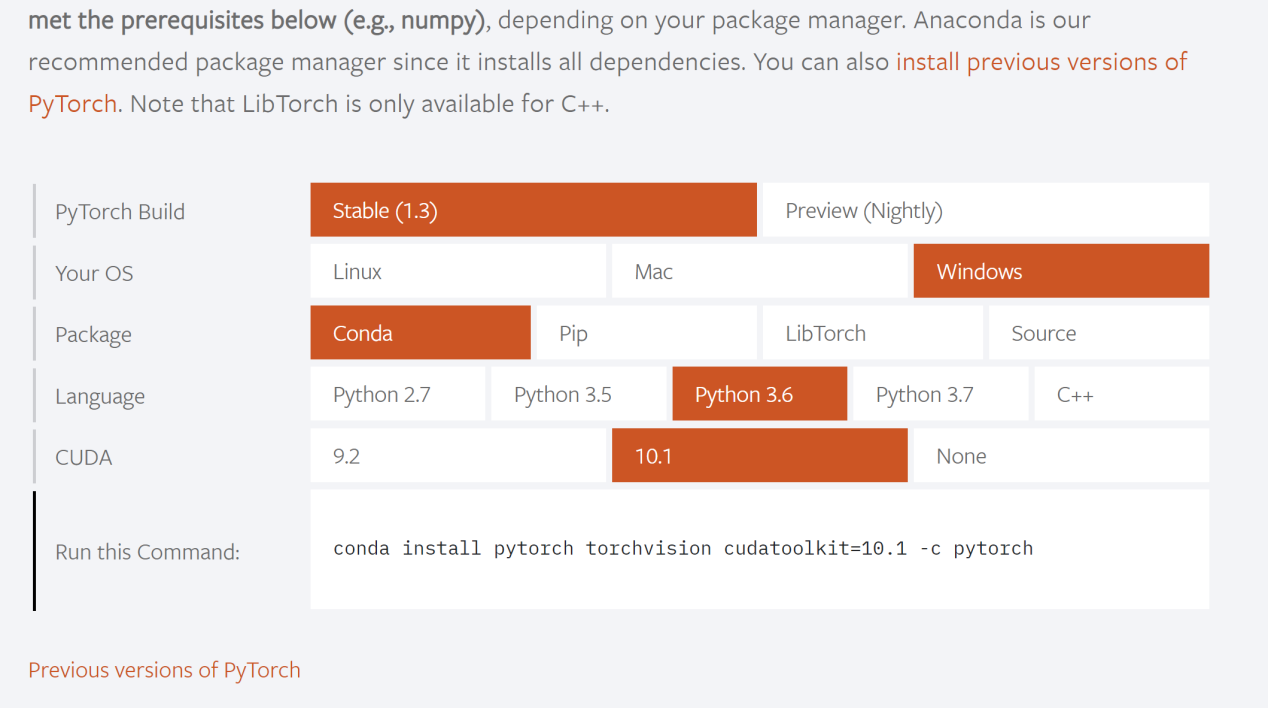
(3)将bin所在的目录添加到环境变量 PATH 中，“此电脑”→“高级系统设置”→“环境变量”→“系统变量”→“path”→“编辑”→“新建”加入该路径即可。

## 打开环境

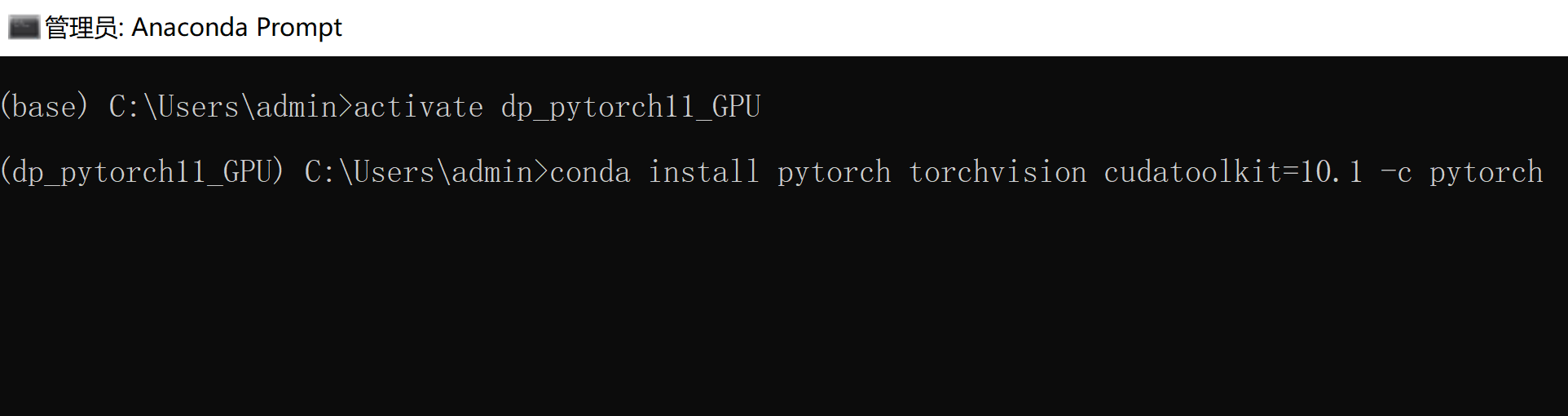
根据第2节教程，新建一个环境，然后打开anaconda prompt窗口，激活你的环境。如果环境已经具备，那么只需要激活即可。

## 安装pytorch

首先打开pytorch官网：<https://pytorch.org/>，往下拉找到如下配置框：



选择你创建环境使用的python版本、本地GPU对应的CUDA版本（如果没有GPU就选择None），选择好配置项后，即可生成安装命令（复制Run this command文本框内的内容），复制此命令至anaconda prompt窗口即可！



备注：

1. 使用这种方式将安装最新版本的pytorch，具体版本见配置框Stable后的内容，上面图示的配置框中pytorch版本为1.3。
2. 如果需要下载历史版本（例如pytorch1.1.0），则可以点击上面配置框下方红色部分Previous version of Pytorch，然后找到相关安装命令即可。

# Yolov3项目示例

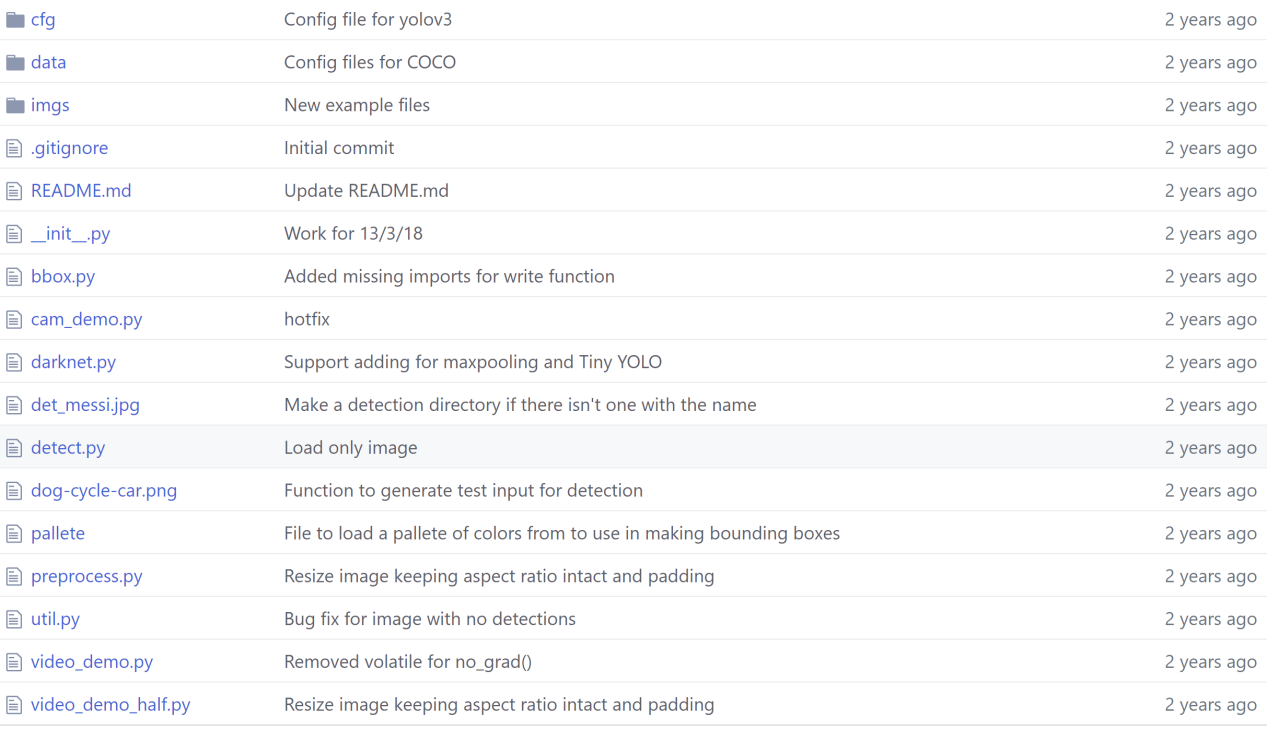
本章节将以Github上开源的项目为例，讲解该项目的文件结构和如何运行。

## 项目一：[ayooshkathuria](https://github.com/ayooshkathuria/pytorch-yolo-v3)

### [下载](https://github.com/ayooshkathuria/pytorch-yolo-v3)

本项目作者ayooshkathuria，可以在GitHub上找到其项目链接：<https://github.com/ayooshkathuria/pytorch-yolo-v3>。打开后将项目代码下载至本地。

项目文件的结构如下图：



根据项目文件可以看到，本项目没有提供训练的功能，可以进行本地图片的识别（detect.py），可以通过摄像头获取图像并识别（cam\_demo.py），可以打开本地视频并对其进行识别（video\_demo.py）。

### 配置

在文件夹cfg中，保存了yolov3的几种常用的模型网络结构，此外还需要下载对应的模型网络参数，例如yolov3模型的参数下载地址为：

<https://pjreddie.com/media/files/yolov3.weights>，可以将该地址复制到浏览器中，通过浏览器下载，但建议使用迅雷下载，比较稳定、快速，下载后将参数文件放到项目文件夹主目录即可。