

pytorch环境安装

说明

pytorch分为cpu版本和gpu版本，若安装cpu版本可直接使用 `pip install pytorch==版本号` 或者 `conda install pytorch==版本号`，若安装gpu版本，则需要注意以下几点

1. pytorch 版本必须同时匹配 cudatoolkit, torchvision版本

版本匹配的问题参考[Previous PyTorch Versions | PyTorch](#)

2. torchvision单独安装时，需严格指定匹配pytorch的版本，否则torchvision会解析自己的依赖，重新下载安装cpu版本的pytorch。torchvision版本和pytorch 对应关系参考[torchvision · PyPI](#)

torch	torchvision	python
main / nightly	main / nightly	>=3.8, <=3.10
1.13.0	0.14.0	>=3.7.2, <=3.10
1.12.0	0.13.0	>=3.7, <=3.10
1.11.0	0.12.0	>=3.7, <=3.10
1.10.2	0.11.3	>=3.6, <=3.9
1.10.1	0.11.2	>=3.6, <=3.9
1.10.0	0.11.1	>=3.6, <=3.9
1.9.1	0.10.1	>=3.6, <=3.9
1.9.0	0.10.0	>=3.6, <=3.9
1.8.2	0.9.2	>=3.6, <=3.9
1.8.1	0.9.1	>=3.6, <=3.9
1.8.0	0.9.0	>=3.6, <=3.9
1.7.1	0.8.2	>=3.6, <=3.9
1.7.0	0.8.1	>=3.6, <=3.8
1.7.0	0.8.0	>=3.6, <=3.8
1.6.0	0.7.0	>=3.6, <=3.8
1.5.1	0.6.1	>=3.5, <=3.8
1.5.0	0.6.0	>=3.5, <=3.8
1.4.0	0.5.0	=2.7, >=3.5, <=3.8
1.3.1	0.4.2	=2.7, >=3.5, <=3.7
1.3.0	0.4.1	=2.7, >=3.5, <=3.7
1.2.0	0.4.0	=2.7, >=3.5, <=3.7
1.1.0	0.3.0	=2.7, >=3.5, <=3.7
<=1.0.1	0.2.2	=2.7, >=3.5, <=3.7

3. 测试已安装的pytorch 是cpu版本还是gpu版本方法:

执行python,输入如下命令,若能执行成功,则说明gpu版安装成功

```
>>> import torch
>>> torch.rand(1,10).cuda()
tensor([[0.3678, 0.0037, 0.2274, 0.9019, 0.2401, 0.5550, 0.6322, 0.6329, 0.9908,
        0.8170]], device='cuda:0')
```

车道线分割项目为例

建议使用python 包管理工具conda

1. 创建虚拟环境(指定虚拟环境名, 和python 版本)

```
conda create -n seg python=3.8
```

2. 安装pytorch和cuda

```
conda install pytorch torchvision torchaudio cudatoolkit=11.1 -c pytorch-lts
-c conda-forge
```

3. 安装其他指定依赖

```
pip install -r requirements.txt
```

项目中用到的其他依赖, 可手动安装

```
pip install 包名
```