

## 感知实验 2 环境安装说明文档

### 1. 安装 PyTorch CPU 版本

#### 1.1 PyTorch 是什么？

PyTorch 是一个基于 Python 的开源机器学习框架，由 Facebook（现 Meta）团队开发，广泛用于深度学习研究和应用。它以动态计算图为核心特性，支持灵活的模型构建与调试，并提供 GPU 加速计算能力，简化了神经网络的设计和训练流程。凭借易用性、丰富的工具库和活跃的社区生态，**PyTorch** 成为学术界和工业界实现自然语言处理、计算机视觉等深度学习相关任务的首选工具之一。

#### 1.2 安装 PyTorch

官网：<https://pytorch.org/>。为了方便，我们推荐同学们直接选择 torch1.11.0 CPU 版本。打开 Anaconda Prompt，输入 conda activate detect 进入 detect 环境。输入：**conda install pytorch==1.11.0 torchvision==0.12.0 torchaudio==0.11.0 cpuonly -c pytorch -y** 进行安装。如下所示，安装结束显示 done。

打开 detect 环境里的 Spyder 输入下方的示例代码验证安装情况。执行后有如下输出：

```
PyTorch版本: 1.11.0
CUDA是否可用: False
张量相加结果: tensor([5, 7, 9])

验证通过！所有包均已正确安装。
```

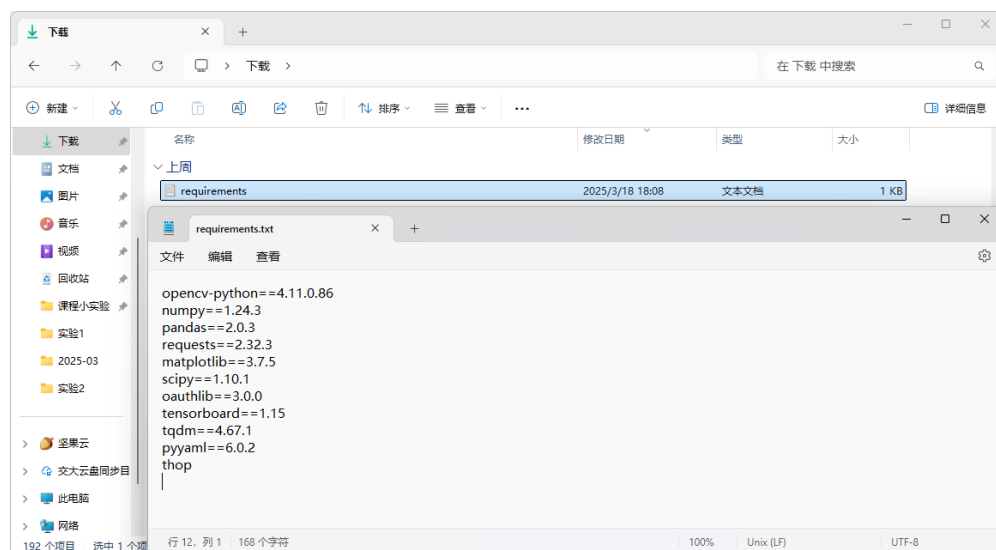
示例代码，用于验证 PyTorch 的安装情况：

```
01 import torch
02
03 # 检查 PyTorch 版本及 CPU 支持
04 print("PyTorch 版本:", torch.__version__)
05 print("CUDA 是否可用:", torch.cuda.is_available()) # 应输出 False
06
07 # 基础张量运算测试
08 a = torch.tensor([1, 2, 3])
09 b = torch.tensor([4, 5, 6])
10 c = a + b
11 print("张量相加结果:", c)
12
13 # 验证 torchvision 和 torchaudio 是否安装（不报错即可）
14 try:
15     import torchvision
16     import torchaudio
17     print("\n 验证通过！所有包均已正确安装。")
18 except ImportError as e:
19     print("\n 安装异常:", e)
```

## 2. 安装依赖库

在电脑安装 Anaconda 的磁盘某一路径下新建 requirements.txt 文件。为了方便，假设同学们 Anaconda 安装在 C 盘（系统盘），在 C 盘（系统盘）的用户下载文件夹里新建 requirements.txt 文件，复制粘贴如下依赖库内容：

```
opencv-python==4.11.0.86
numpy==1.24.3
pandas==2.0.3
requests==2.32.3
matplotlib==3.7.5
scipy==1.10.1
oauthlib==3.0.0
tensorboard==1.15
tqdm==4.67.1
pyyaml==6.0.2
thop
```



打开 Anaconda Prompt，输入 conda activate detect 进入 detect 环境，接着输入指令：  
**cd C:\Users\95811\Downloads** 进入 C 盘（系统盘）的用户下载文件夹，注意修改路径为自己电脑的下载目录路径，上述指令中的用户名为 95811，需要更改为自己电脑的用户名。  
（注：如果 Anaconda 安装在其他磁盘，比如 E 盘，需在 E 盘某一文件夹下执行相似操作。）  
输入安装指令：**pip install -r requirements.txt -i https://pypi.tuna.tsinghua.edu.cn/simple**  
打开 detect 环境里的 Spyder 输入下方的示例代码验证安装情况。执行后有如下输出：

```
验证结果：
库名          | 实际版本          | 状态
-----|-----|-----
opencv-python | 4.11.0             | ✓
numpy          | 1.24.3             | ✓
pandas         | 2.0.3              | ✓
requests       | 2.32.3             | ✓
matplotlib     | 3.7.5              | ✓
scipy          | 1.10.1             | ✓
oauthlib       | 3.0.0              | ✓
tensorboard    | 1.15.0             | ✓
tqdm           | 4.67.1             | ✓
pyyaml         | 6.0.2              | ✓
thop           | 0.1.1              | ✓

所有依赖库验证通过！
```

示例代码，用于验证感知实验 2 依赖库的安装情况：

```
01 import importlib
02 deps = {
03     "opencv-python": ["cv2", "4.11.0"],
04     "numpy": ["numpy", "1.24.3"],
05     "pandas": ["pandas", "2.0.3"],
06     "requests": ["requests", "2.32.3"],
07     "matplotlib": ["matplotlib", "3.7.5"],
08     "scipy": ["scipy", "1.10.1"],
09     "oauthlib": ["oauthlib", "3.0.0"],
10     "tensorboard": ["tensorboard", "1.15.0"],
11     "tqdm": ["tqdm", "4.67.1"],
12     "pyyaml": ["yaml", "6.0.2"],
13     "thop": ["thop", None]
14 }
15 def chk_ver(actual, expected): return actual.split('+')[0] == expected
16 results = []
17 for lib, (imp, exp) in deps.items():
18     try:
19         mod = importlib.import_module(imp)
20         ver = getattr(mod, "__version__", "未知版本")
21         stat = "√" if (exp is None or chk_ver(ver, exp)) else "X"
22     except Exception as e:
23         ver, stat = f"未安装({e})", "X"
24     results.append((lib, ver, stat))
25 print("\n 验证结果： ")
26 print(f'{"库名":<15} | {"实际版本":<15} | {"状态":<5}')
27 print("-"*40)
28 for lib, ver, stat in results:
29     color = "\033[92m" if stat=="√" else "\033[91m"
30     print(f'{"lib":<15} | {"ver":<15} | {color}{stat}\033[0m')
31 if all(s=="√" for _,s in results):
32     print("\n 所有依赖库验证通过！ ")
33 else:
34     print("\n 警告： 部分依赖库未正确安装！ ")
```