

“启明” PyTorch 手册

编写时间：2025 年 1 月

目录

一、 介绍	3
1.1. 简介	3
二、 安装软件	3
2.1. 配置anaconda	3
2.2. 加载南科大镜像源	3
2.3. 创建虚拟环境	4
2.4. 安装pytorch	4
三、 作业样例	5
3.1. 提交脚本	5
3.2. 提交并查看作业	6
3.3. 作业测试结果	6
四、 注意事项	6
4.1. 关于测试样例	6
4.2. 联系方式	6

一、介绍

1.1. 简介

PyTorch 是一个基于软件的开源深度学习框架，用于构建神经网络，将 Torch 的机器学习 (ML) 库与基于 Python 的高级 API 相结合。它的灵活性和易用性以及其它优点使其成为学术和研究界领先的机器学习框架。

PyTorch 支持多种神经网络架构，从简单的线性回归算法到复杂的卷积神经网络和用于计算机视觉和自然语言处理 (NLP) 等任务的生成式转换器模型。PyTorch 基于广为人知的 Python 编程语言构建，并提供广泛的预配置（甚至预训练）模型库，使数据科学家能够构建和运行复杂的深度学习网络，同时最大限度地减少在代码和数学结构上花费的时间和精力。

PyTorch 最初由 Facebook AI Research（现为 Meta）开发，于 2017 年配置为开源语言，自 2022 年起由 PyTorch Foundation（隶属于大型 Linux Foundation）管理。该基金会为深度学习社区提供一个中立空间，以合作并进一步开发 PyTorch 生态系统。

更多内容，请参考官方网站：<https://pytorch.org/get-started/locally/>

二、安装软件

2.1. 配置anaconda

我们建议使用anaconda来管理环境，目前启明集群已安装好anaconda3/2022.10版本，直接加载使用即可，如有其它版本需求可以自行安装。

加载环境：

```
module load anaconda3
```

```
source activate
```

```
[ccse-yangyc@b04u17l ~]$ module load python/anaconda3/2022.10  
[ccse-yangyc@b04u17l ~]$ source activate  
(base) [ccse-yangyc@b04u17l ~]$
```

2.2. 加载南科大镜像源

把以下镜像地址写入 ~/.condarc 文件

channels:

- defaults

show_channel_urls: true

default_channels:

- <https://mirrors.sustech.edu.cn/anaconda/pkg/main>
- <https://mirrors.sustech.edu.cn/anaconda/pkg/free>
- <https://mirrors.sustech.edu.cn/anaconda/pkg/r>

- <https://mirrors.sustech.edu.cn/anaconda/pkg/pro>
- <https://mirrors.sustech.edu.cn/anaconda/pkg/msys2>

custom_channels:

conda-forge: <https://mirrors.sustech.edu.cn/anaconda/cloud>
msys2: <https://mirrors.sustech.edu.cn/anaconda/cloud>
bioconda: <https://mirrors.sustech.edu.cn/anaconda/cloud>
menpo: <https://mirrors.sustech.edu.cn/anaconda/cloud>
pytorch: <https://mirrors.sustech.edu.cn/anaconda/cloud>
simpleitk: <https://mirrors.sustech.edu.cn/anaconda/cloud>
nvidia: <https://mirrors.sustech.edu.cn/anaconda-extra/cloud>

2.3. 创建虚拟环境

在登录节点创建环境安装python，虚拟环境可以帮助您隔离不同项目的依赖项，避免不同项目之间的冲突。以下是创建虚拟环境的步骤：

1. 创建虚拟环境：运行以下命令来创建一个新的虚拟环境。您可以将<env_name>替换为您喜欢的环境名称，例如“pytorch_env”。

例如，要创建一个名为pytorch_2.5.0-cuda11.8的虚拟环境，其中Python版本为3.10，可以运行：

```
conda create -n pytorch_2.5.0-cuda11.8 python=3.10 -y
```

2. 激活虚拟环境：运行以下命令来激活刚刚创建的虚拟环境

```
conda activate pytorch_2.5.0-cuda11.8
```

```
(base) [ccse-yangyc@b04u17l ~]$ conda activate pytorch_2.5.0-cuda11.8  
(pytorch_2.5.0-cuda11.8) [ccse-yangyc@b04u17l ~]$
```

2.4. 安装pytorch

在启明集群上，我们强烈建议安装支持cuda11.8的pytorch来兼容集群GPU驱动的使用，安装之前到官网查看对应cuda的版本：<https://pytorch.org/get-started/previous-versions/> 使用pip进行安装，这里安装pytorch-2.5.0为例：

```
pip install torch==2.5.0 torchvision==0.20.0 torchaudio==2.5.0 --index-url  
https://download.pytorch.org/whl/cu118
```

```
Collecting MarkupSafe>=2.0 (from Jinja2->torch==2.5.0)  
  Downloading https://download.pytorch.org/whl/MarkupSafe-2.1.5-cp310-cp310-manylinux_2_17_x86_64_manylinux2014_x86_64.whl (25 kB)  
Installing collected packages: mpmath, typing-extensions, sympy, pillow, nvidia-nvtx-cu11, nvidia-nccl-cu11, nvidia-cuspars-cu11, nvidia-curand-cu11, nvidia-cufft-cu11, nvidia-cuda-runtime-cu11, nvidia-cuda-nvrtc-cu11, nvidia-cuda-cupti-cu11, nvidia-cublas-cu11, numpy, networkx, MarkupSafe, fsspec, filelock, triton, nvidia-cusolver-cu11, nvidia-cudnn-cu11, Jinja2, torch, torchvision, torchaudio  
Successfully installed MarkupSafe-2.1.5 filelock-3.13.1 fsspec-2024.2.0 Jinja2-3.1.3 mpmath-1.3.0 networkx-3.2.1 numpy-1.26.3 nvidia-cublas-cu11-11.11.3.6 nvidia-cuda-cupti-cu11-11.8.87 nvidia-cuda-nvrtc-cu11-11.8.89 nvidia-cuda-runtime-cu11-11.8.89 nvidia-cudnn-cu11-9.1.0.70 nvidia-cufft-cu11-10.9.0.58 nvidia-curand-cu11-10.3.0.86 nvidia-cusolver-cu11-11.4.1.48 nvidia-cuspars-cu11-11.7.5.86 nvidia-nccl-cu11-2.21.5 nvidia-nvtx-cu11-11.8.86 pillow-10.2.0 sympy-1.13.1 torch-2.5.0+cu118 torchaudio-2.5.0+cu118 torchvision-0.20.0+cu118 triton-3.1.0 typing-extensions-4.9.0  
(pytorch_2.5.0-cuda11.8) [ccse-yangyc@b04u17l ~]$
```

```
# ROCM 6.1 (Linux only)
pip install torch==2.5.0 torchvision==0.20.0 torchaudio==2.5.0 --index-url https://download.pytorch.org/whl,
# ROCM 6.2 (Linux only)
pip install torch==2.5.0 torchvision==0.20.0 torchaudio==2.5.0 --index-url https://download.pytorch.org/whl,
# CUDA 11.8
pip install torch==2.5.0 torchvision==0.20.0 torchaudio==2.5.0 --index-url https://download.pytorch.org/whl,
# CUDA 12.1
pip install torch==2.5.0 torchvision==0.20.0 torchaudio==2.5.0 --index-url https://download.pytorch.org/whl,
# CUDA 12.4
pip install torch==2.5.0 torchvision==0.20.0 torchaudio==2.5.0 --index-url https://download.pytorch.org/whl,
# CPU only
pip install torch==2.5.0 torchvision==0.20.0 torchaudio==2.5.0 --index-url https://download.pytorch.org/whl,
```

三、 作业样例

3.1. 提交脚本

提交脚本文件为pytorch.lsf，测试torch是否可用gpu，test.py 脚本来测试作业分配的gpu卡是否能检测到

1. 测试脚本内容：

```
import torch
x = torch.rand(5, 3)
print(x)

print("gpu status:", torch.cuda.is_available())
```

2. 作业脚本内容：

```
#!/bin/bash
#BSUB -J torch      #作业名称，可以自定义
#BSUB -q 4a100-40    #提交队列
#BSUB -n 1           #申请1个核，和申请的-gpu卡数一致
#BSUB -e %J.err      #定义错误输出文件
#BSUB -o %J.out      #定义正常输出日志文件
#BSUB -R "span[ptile=1]" #申明一个节点核数，这个跟-n 申请的核数一致
#BSUB -gpu "num=1"    #申请1张gpu卡

#加载安装pytorch环境
module load python/anaconda3/2022.10
source activate
conda activate pytorch_2.5.0-cuda11.8
```

```
#运行python脚本  
python test.py > $LSB_JOBID.log 2>&1
```

3.2. 提交并查看作业

```
[ccse-yangyc@b04u17l pytorch]$ bsub < test.lsf  
Job <312013> is submitted to queue <4a100-40>.  
[ccse-yangyc@b04u17l pytorch]$ bjobs  
JOBID  USER  STAT  QUEUE  FROM_HOST  EXEC_HOST  JOB_NAME  SUBMIT_TIME  
312013 ccse-ya RUN   4a100-40  b04u17l    be1u03g    torch     Jan 19 16:21
```

3.3. 作业测试结果

测试结果文件为311937.log

```
[ccse-yangyc@b04u17l pytorch]$ cat 312013.log  
tensor([[0.8541, 0.6915, 0.9733],  
        [0.6412, 0.3411, 0.5252],  
        [0.4550, 0.1467, 0.3675],  
        [0.7121, 0.2538, 0.9524],  
        [0.7227, 0.4786, 0.9022]])  
gpu status: True
```

四、 注意事项

4.1. 关于测试样例

- 本文档测试样例，只是验证改软件能正常运行。安装torch之后，还得根据自己使用的其它模块在登录节点安装进行调试使用。
- 欢迎使用该软件的用户，提供更加好的python脚本样例，以便提高该软件的更多测试。

4.2. 联系方式

- 本文档由科学与工程计算中心编写，联系邮箱：hpc@sustech.edu.cn, 联系人：谢作扬，电话：0755-88015831