

实验室相关

一、环境配置

1.连接服务器

以下以psdz这台服务器为例。

该服务器的IP地址是59.77.5.64，使用SSH端口为10001，账号是haohan (yingying)，密码是Vcg@xmu2023student，使用XFtp或者使用XShell进行连接即可。

2.pytorch环境配置

(1) 显卡版本检查

这里以深度学习的课程作业为例，采用3.9的Python版本，Pytorch版本是1.12.0。

首先用nvidia-smi查看显卡cuda driver API的版本：

```
haohan@psdz:~$ nvidia-smi
Sun Nov 12 13:35:07 2023
```

NVIDIA-SMI 470.182.03 Driver Version: 470.182.03 CUDA Version: 11.4									
GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile	Uncorr.	ECC		
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage	GPU-Util	Compute	M.		
						MIG	M.		
0	NVIDIA GeForce ...	Off	00000000:08:00.0	Off			N/A		
53%	52C	P2	181W / 350W	6577MiB / 24268MiB	45%	Default	N/A		
1	NVIDIA GeForce ...	Off	00000000:09:00.0	Off			N/A		
71%	64C	P2	190W / 350W	9175MiB / 24268MiB	30%	Default	N/A		

也可以用 nvcc --version 查看Cuda Runtime API的版本：

```
haohan@psdz:~$ nvcc --version
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2022 NVIDIA Corporation
Built on Tue_Mar__8_18:18:20_PST_2022
Cuda compilation tools, release 11.6, V11.6.124
Build cuda_11.6.r11.6/compiler.31057947_0
```

补充知识：nvidia-smi和nvcc显示的cuda版本号不一致是什么原因？

[【CUDA】nvcc和nvidia-smi显示的版本不一致？ - 简书\(jianshu.com\)](https://jianshu.com/p/111111111111)

(2) Conda环境创建，安装torch

(a) 首先，新建一个环境，使用Python 3.9环境，指令如下：

```
conda create -n chris_torch python=3.9
```

注：提示调用了别人之前下载的缓存，但是没有读别人创建的文件权限怎么办，具体报错信息如下：

```
NotWritableError: The current user does not have write permissions to a required path.
path: /home/psdz/miniconda3/pkgs/pytorch-1.8.1-py3.7_cuda10.2_cudnn7.6.5_0/info/repodata_record.json
uid: 1015
gid: 1016

If you feel that permissions on this path are set incorrectly, you can manually
change them by executing

$ sudo chown 1015:1016 /home/psdz/miniconda3/pkgs/pytorch-1.8.1-py3.7_cuda10.2_cudnn7.6.5_0/info/repodata_record.json

In general, it's not advisable to use 'sudo conda'.
```

解决方案：

```
sudo chmod -R 777 /home/psdz/miniconda3/
```

关于chmod 777的介绍如下：[chmod 777. The chmod 777 command is used in... | by The Humble Coder | Medium](#)

简单来说，就是修改用户对指定文件/文件夹的读写权限。

(b) 接下来进入该环境：

```
conda activate chris_torch
```

注：报如下错误：

```
CommandNotFoundError: Your shell has not been properly configured to use 'conda activate'.
To initialize your shell, run

$ conda init <SHELL_NAME>

Currently supported shells are:
- bash
- fish
- tcsh
- xonsh
- zsh
- powershell

See 'conda init --help' for more information and options.

IMPORTANT: You may need to close and restart your shell after running 'conda init'.
```

解决方案为：[conda activate激活虚拟环境时报错: Your shell has not been properly configured to use 'conda activate'的解决方法 想变厉害的大白菜的博客-CSDN博客](#)

输入下面两句：

```
source activate
conda deactivate
```

问题得到了解决。

(c) 安装pytorch

这里参考Pytorch的官方文档: [Previous PyTorch Versions | PyTorch](#), 以及深度学习课程的环境配置指南[DEEP LEARNING \(xmu.edu.cn\)](#)

```
# CUDA 11.6
conda install pytorch==1.12.0 torchvision==0.13.0 torchaudio==0.12.0
                cudatoolkit=11.3 -c pytorch
```

更新的版本可以参考这篇文档: [Start Locally | PyTorch](#)

注: 如果网络不好, 提示 `Solving environment: failed with initial frozen solve. Retrying with flexible solve.`, 可以考虑使用清华源进行下载, 方法如下:

[为Conda添加清华软件源 - 知乎\(zhihu.com\)](#)

等待下载安装即可。

(d) 测试pyorch是否安装成功, 是否是GPU版本, 以及对应的版本。

```
# test.py
import torch
from torch import nn
import numpy as np
import logging
import sys

# set log
logging.basicConfig(level=logging.INFO,
                    format='%(asctime)s %(levelname)s: %(message)s',
                    datefmt='%Y-%m-%d %H:%M:%S',)
logging.info('The version information:')
logging.info(f'Python: {sys.version}')
logging.info(f'PyTorch: {torch.__version__}')
assert torch.cuda.is_available() == True, 'Please finish your GPU develop
environment'
```

将上述代码用XFtp传到服务器对应文件夹下, 然后执行python test.py, 查看输出结果。结果如下:

```
(chris_torch) haohan@psdz:~/AI/DeepLearningHomework/Assignment3$ python test.py
2023-11-12 14:33:38 INFO: The version information:
2023-11-12 14:33:38 INFO: Python: 3.9.18 (main, Sep 11 2023, 13:41:44)
[GCC 11.2.0]
2023-11-12 14:33:38 INFO: PyTorch: 1.12.0
```

至此, Pytorch环境就配置完成了。

3.Jupyter notebook连接远程服务器

参考链接: [Jupyter notebook远程连接服务器 - 知乎\(zhihu.com\)](#)

首先, 在服务器端安装jupyter notebook:

```
pip install jupyter notebook
```

(1) 在服务器端的配置:

step 1:

连接好服务器后，在终端运行以下命令来生成配置文件：

```
jupyter notebook --generate-config
```

服务器的输出如下：

```
Writing default config to: /home/haohan/.jupyter/jupyter_notebook_config.py
```

step 2:

- 接下来键盘输入 `ipython`
- 输入 `from notebook.auth import passwd`
 - 注：如果报错没有 `notebook.auth`，则改为 `from jupyter_server.auth import passwd`
- 输入 `passwd()`，Enter password: 输入你要设定的密码，Verify password: 验证密码

如下图：

```
In [1]: from notebook.auth import passwd
-----
ModuleNotFoundError                                Traceback (most recent call last)
Cell In[1], line 1
----> 1 from notebook.auth import passwd

ModuleNotFoundError: No module named 'notebook.auth'

In [2]: from jupyter_server.auth import passwd
In [3]: passwd()
Enter password:
Verify password:
Out[3]: 'argon2:$argon2id$v=19$m=10240,t=10,p=8$LfPY8evx48RbbirfJzJfg$3FN3GsU1H+PwTXN1AjhFJCtSdSYqIHmW4TmlsqK0izU'
```

这里只需你输入密码确认，然后他会自动帮你把生成含有密码的hash码输入到 `jupyter_notebook_config.json` 文件。

step3:

- 修改相应的配置文件
- 输入以下的命令 `vim ~/.jupyter/jupyter_notebook_config.py`
- 在文件末尾添加以下代码：

```
c.NotebookApp.ip = '*' # 允许访问此服务器的 IP，星号表示任意 IP
c.NotebookApp.password =
'argon2:$argon2id$v=19$m=10240,t=10,p=8$LfPY8evx48RbbirfJzJfg$3FN3GsU1H+PwTXN1AjhFJCtSdSYqIHmW4TmlsqK0izU' # 之前生成的密码 hash 字串，粘贴进去
c.NotebookApp.open_browser = False # 运行时不打开本机浏览器
c.NotebookApp.port = 8890 # 使用的端口，随意设置，但是要记得你设定的这个端口
c.NotebookApp.enable_mathjax = True # 启用 MathJax
c.NotebookApp.allow_remote_access = True # 允许远程访问
c.NotebookApp.allow_root = True
```

补充1：注意：这里 `c.NotebookApp.password` 是你刚才设定密码的hash码，修改成自己的hash码。`c.NotebookApp.port` 是ssh转发的端口号，随意设定一个即可，后面要记住你设定的这个转发的端口号。其他的都不需要修改。

补充2：

```
# Vim 命令帮助
# 跳转到文件末尾的方法：按esc键，然后输入:$ 即可跳转到文件末尾

# SHIFT + i 进行插入

# ESC 退出，然后输入 :wq! 进行保存
```

(2) 在服务器端运行jupyter notebook

```
jupyter notebook --no-browser --port=8890
```

此时服务器那边的就开启了8890端口（之前在配置文件当中改的8890端口），用来运行jupyter notebook。

4.在Windows上远程连接服务器

如果上面的流程都走完之后，且也在服务器端运行 `jupyter notebook` 之后，需要在本地上通过SSH连接远程服务器并连接到jupyter。

cmd打开dos命令窗口，敲入此句命令，将本地端口与服务器端相映射。

```
ssh -L [本地端口]:localhost:[远程端口] [远程用户名]@[远程IP] -p [ssh连接端口]
# 我的情况: ssh -L 8890:localhost:8890 haohan@59.77.5.64 -p 10001
```

注意：这里的远程端口是上面配置的端口号（例如我配置的是8890，这里就需要写8890），本地端口随便写一个就可以，我这里写的是8890。

输入上面的命令之后，会提示输入远程连接的密码，输入密码不报错即连接成功。

对应示意图如下：

```
C:\Users\15969>ssh -L 8890:localhost:8890 haohan@59.77.5.64 -p 10001
haohan@59.77.5.64's password:
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-120-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

Expanded Security Maintenance for Infrastructure is not enabled.

87 updates can be applied immediately.
70 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

115 additional security updates can be applied with ESM Infra.
Learn more about enabling ESM Infra service for Ubuntu 18.04 at
https://ubuntu.com/18-04

New release '20.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2023.
Last login: Sun Nov 12 15:15:30 2023 from 10.32.53.31
```

接下来在本地cmd当中输入 `jupyter notebook`（好像不输入也没事），然后在浏览器中输入 `localhost:8890`（根据前面自己设定的本地端口） 它会自动映射到服务器的jupyter服务，出现jupyter窗口，输入你设定的jupyter密码，即成功进入。

接下来就可以在远程服务器上跑本地jupyter notebook上的代码了。

二、一些遇到的问题

1.连接断开

如果每次跑一会就断开连接，可以在XShell中修改持续连接的时间等。

Linux指令（常使用）

1.修改文件权限

`chmod -R` ([chmod 777. The chmod 777 command is used in... | by The Humble Coder | Medium](#))：举例：

```
sudo chmod -R 777 /home/psdz/miniconda3/
```

2.查看进程相关的信息

[使用ps命令根据内存使用情况对输出进行排序之路教程 \(onitroad.com\)](#)