代码的调试

VScode调试

点击运行按钮找到调试程序开启调试
随后出现调试界面
调试界面主要有变量，监视，调用控制台，调用堆栈等等

变量

变量中包含局部变量`Locals`和全局变量`Globals`：
其中局部变量是函数内部的变量，作用范围在函数内部，全局变量是整个代码全局的变量，作用范围从定义到代码结束

监视

监视可以输入想要监视的变量名，在逐步调试的过程中观察变量值的变化

调试控制台

向调试控制台中输入变量的名字可以直接得到对应变量的实时输出，也就是此刻该变量的取值

调用堆栈

进入多层嵌套的函数的时候，调用堆栈中会显示对应每一层的函数位置，比如说主函数流程调用了函数a，函数a调用了函数b，通过不断步入调用堆栈就会显示主文件，函数a，函数b对应位置，并且方便进行回溯

调试按键说明

- 继续执行：程序从当前断点开始继续运行，直到遇到下一个断点或者程序结束

- 逐过程：执行当前行代码，如果这一行代码调用了函数，不会进入函数内部，直接将函数当做是一步执行完毕并且进入下一行
- 单步调试：执行当前行代码，如果是函数调用，会进入函数内部，逐步调试函数内部的代码
- 单步跳出：进入某个函数内部之后，快速执行完剩余的代码，并且跳出到调用这个函数的地方

断点设置

在调试前设置断点一般设置在以下位置：

- 异常前一行设置断点
- 判断分支判断出设置断点
- 循环开始前面设置断点
- 关键函数入口设置断点

项目配置

Readme：首先阅读每个项目的`readme`文件，这个里面包含跑通项目的帮助介绍
env：配置好运行环境，一般代码配置在`requirement.txt`上面
config：导入参数的时候一般要设置`default`超参数，这时候运行的时候如果没有设置就会默认运行`default`超参数，如果没有指定则需要使用`--参数名 参数值`
对于需要导入参数的项目文件，需要创建一个`launch.json`文件，并且点击PythonDebugger，选择带有参数的Python文件

在需要debug的项目训练python代码中加上

```
import debugpy
try:
    # 5678 is the default attach port in the VS Code debug configurations. Unless a host and port are specified, host defaults to 127.0.0.1
```

```
    debugpy.listen(("localhost", 9501))
    print("Waiting for debugger attach")
    debugpy.wait_for_client()
except Exception as e:
    pass
```

然后在vscode的launch.json的configuration中加上这个配置

```
{
    "name": "sh_file_debug",
    "type": "debugpy",
    "request": "attach",
    "connect": {
        "host": "localhost",
        "port": 9501
    }
}
```

然后正常启动训练脚本
在需要debug的python文件中打上断点
打印出来的是 `Waiting for debugger attach`
然后在vscode中的debug界面选择 `sh_file_debug` 进行debug
debug结束后把代码中添加的注释删除掉

VScode远程连接

vscode是使用remote_ssh去远程连接服务器，直接对服务器进行操作

- pycharm可以使用jetbrains gateway进行远程连接这个和vscode是一个机制，性能要求可能比较高
- 同步机制：构建本地和运行远程的一个目录映射，构建好映射之后本地文件和代码一旦发生修改就会自动同步到服务器上

- 右键调试或者运行其实就是使用的是远程的虚拟环境，运行的远程的文件和代码

Pycharm远程连接

必须在本地单独创建项目文件，如果在父文件里的子文件连接远程，- PyCharm 会把整个上级文件夹作为项目根，所有路径映射、索引、解释器设置，都是针对**父项目根目录**进行！

- 在本地用户路径下配置本地ssh文件。本地ssh生成一个公私钥，私钥是本地的文件，公钥上传到服务器中
- 配置服务器现有虚拟环境
- 可以远程连接服务器终端，debug的时候观察输出
- 对于有超参数设置的项目，可以使用形参运行，在里面可以选择输入脚本形参然后就能进行正常的debug环节

远程连接终端时候的输出

```
/home/gpu-node/anaconda3/bin/conda run -n DL --no-
capture-output python /home/gpu-
node/.pycharm_helpers/pydev/pydevd.py --multiprocess --
qt-support=auto --client localhost --port 41625 --file
/usr/disk5/jhyang/DL/test.py
```