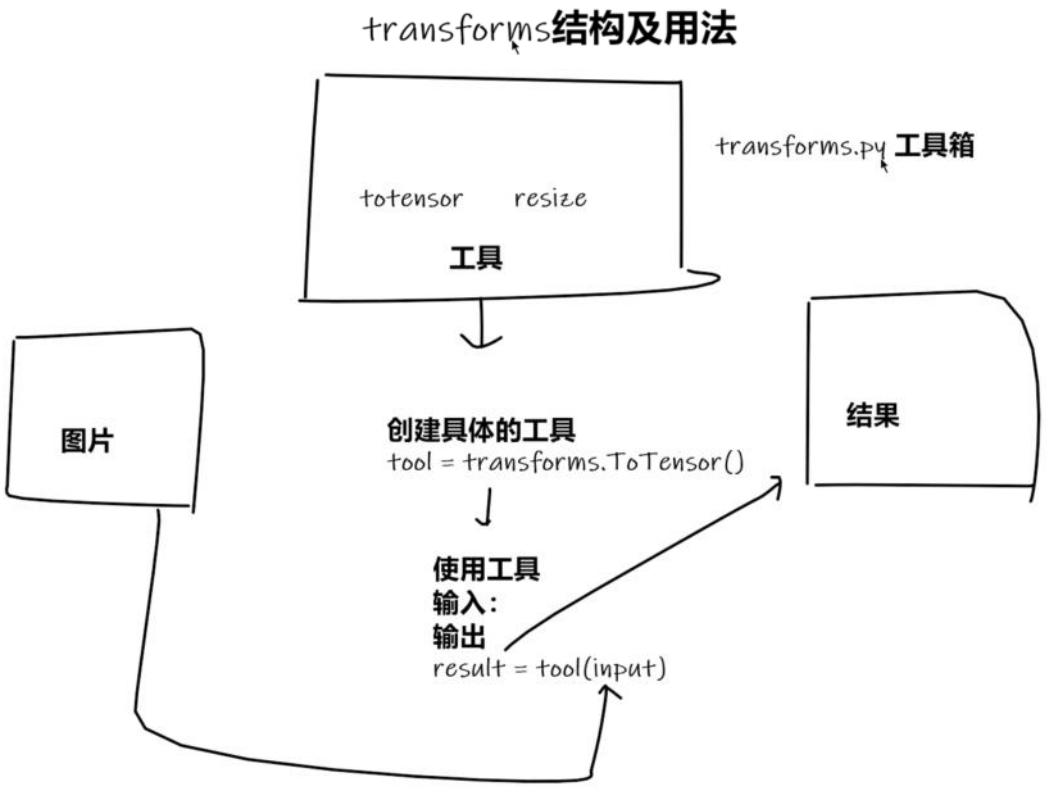
transform

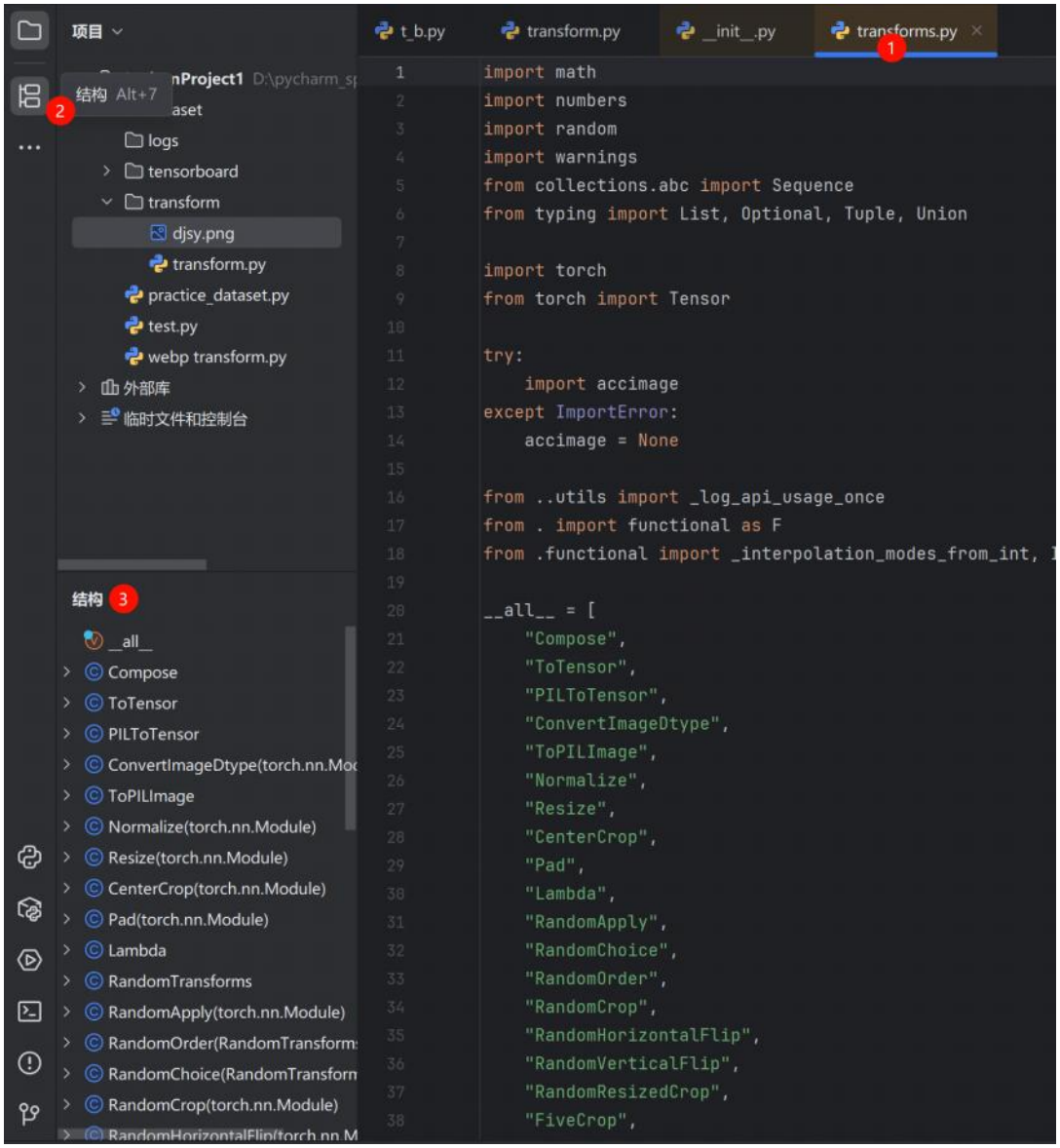
2025年11月15日 19:26



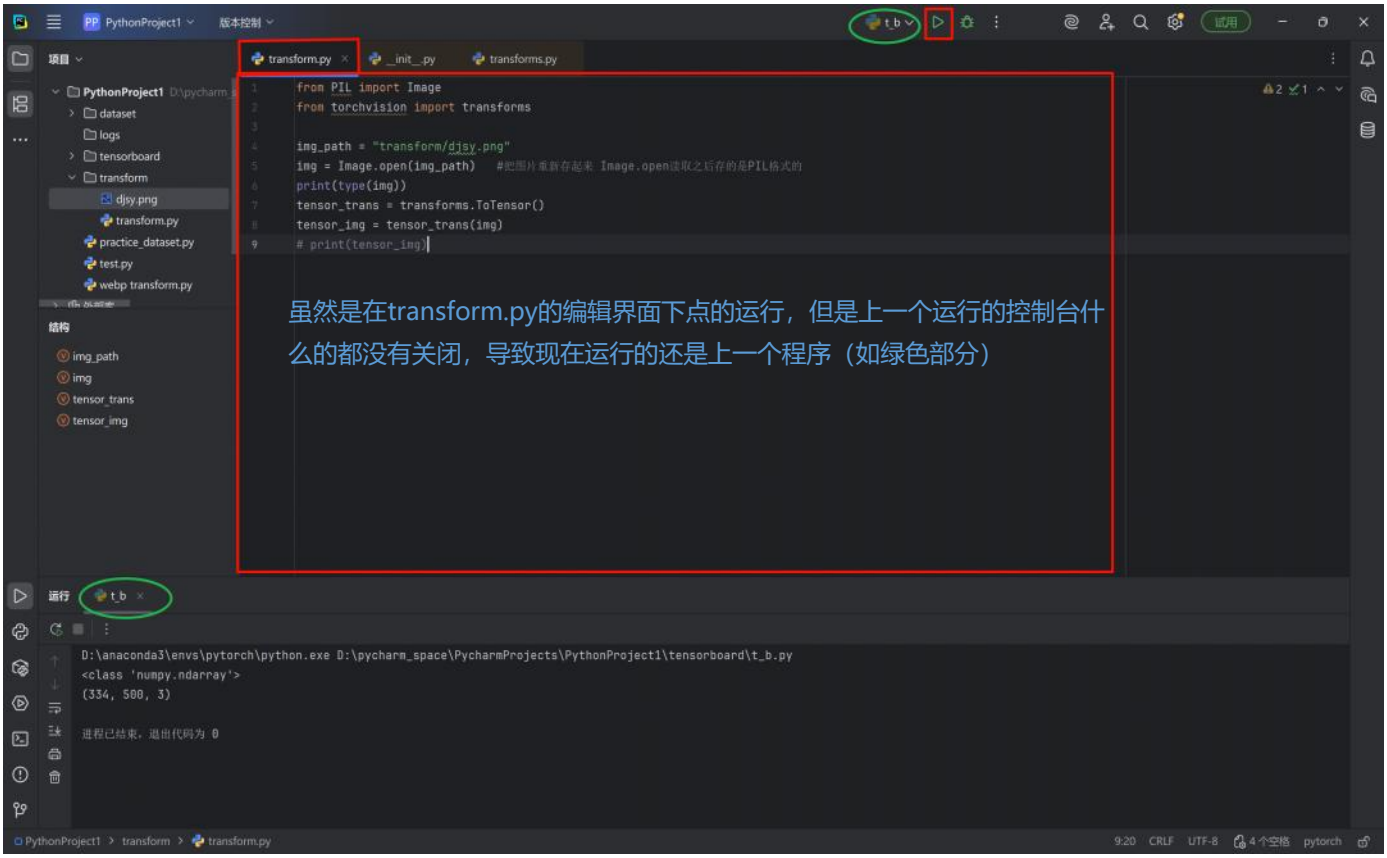
为什么需要Tensor数据类型

包含了深度学习所要用到的一些数据，方便后面进行一些训练

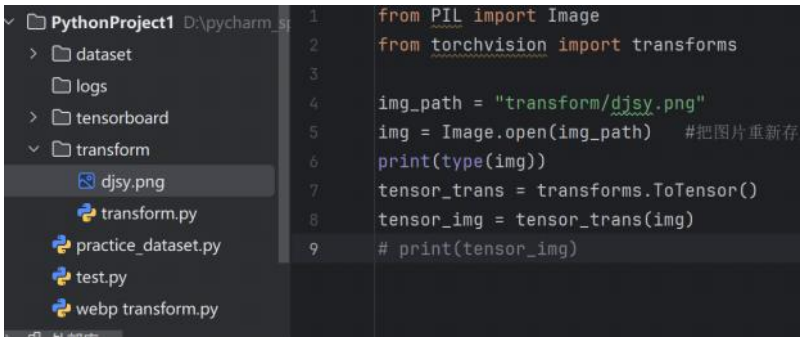
学会查看官方文件

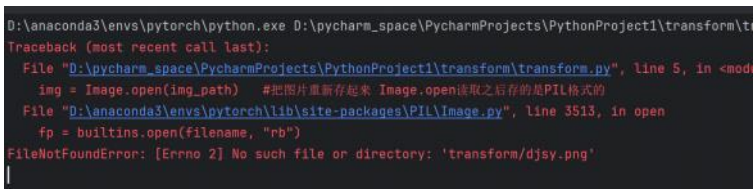


忽视的小细节，导致运行错误



路径的错误



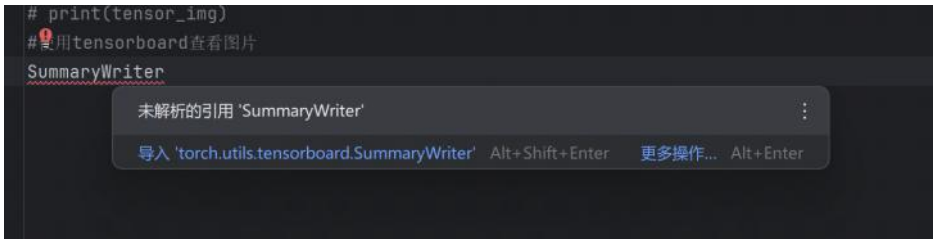


报错中显示程序在找 'transform/djsy.png'，这说明当前工作目录是 **PythonProject1**（项目根目录），而脚本误以为图片在 PythonProject1/transform/djsy.png。但实际情况是：

如果图片和脚本在同一个 transform 文件夹下，正确的相对路径应该是 **djsy.png**（脚本所在目录为基准），而非 transform/djsy.png。

代码红了

按 alt + enter



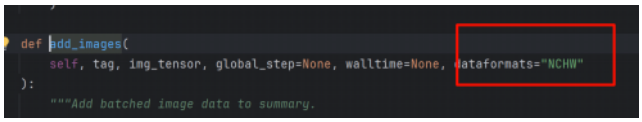
就可直接导入需要的库

查看需要传入的数据

直接按ctrl +P

使用tensorboard时要注意图片的格式

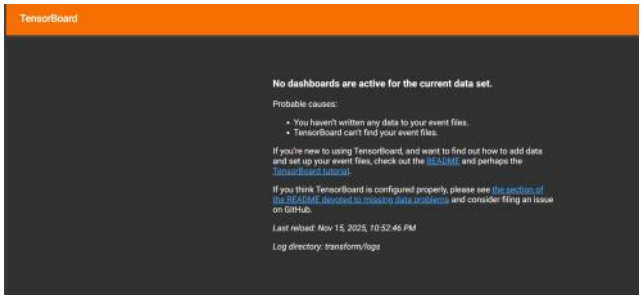
但是



官方描述文件里是这个

所以

别忘记加writer.close()



我没有加writer.close()这个是干嘛的

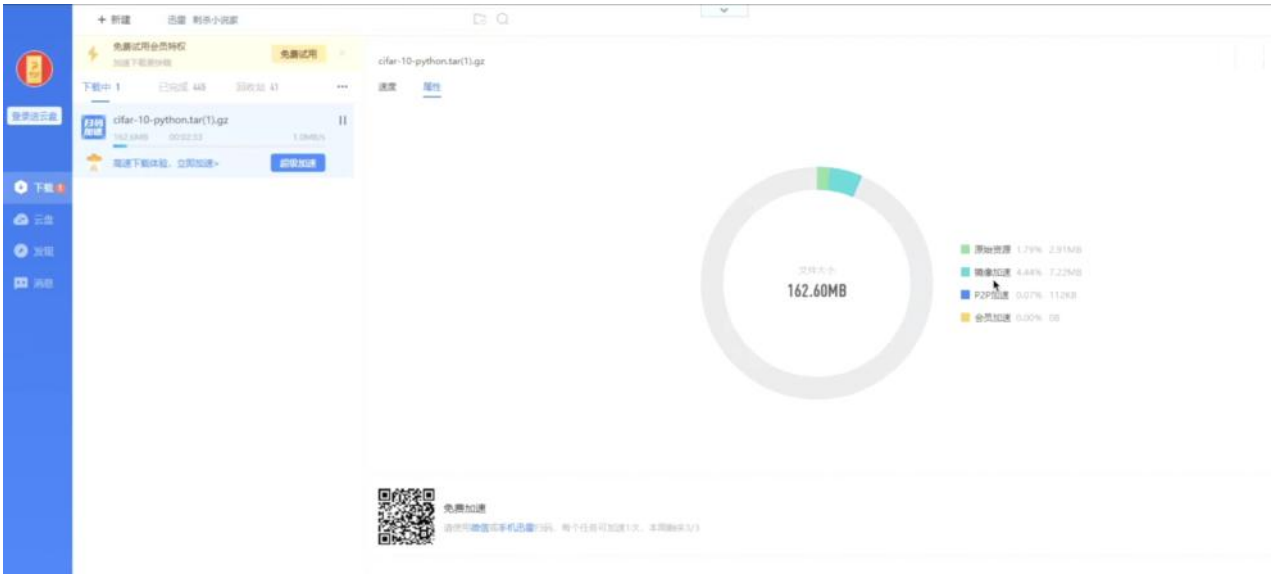
`writer.close()` 是 `SummaryWriter`（TensorBoard 写入工具）的一个重要方法，作用是**确保所有缓存的数据被强制写入到日志文件中，并释放相关资源**。具体来说：

1. 为什么需要 `writer.close()`？

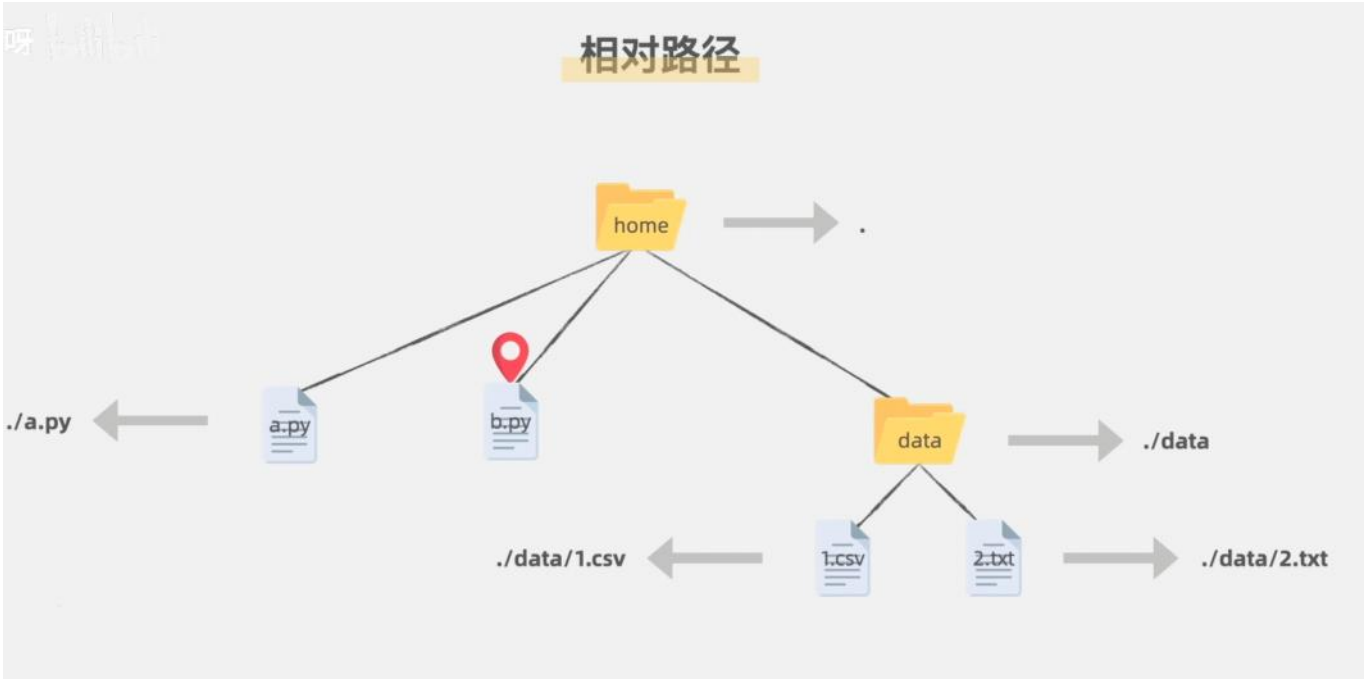
`SummaryWriter` 在写入数据（如 `add_scalar`、`add_image` 等）时，并不会立即把所有数据直接写入磁盘，而是先缓存在内存中，积累到一定量后再批量写入（类似“缓冲区”机制，提升效率）。

如果代码中没有调用 `writer.close()`：

迅雷还能这样用

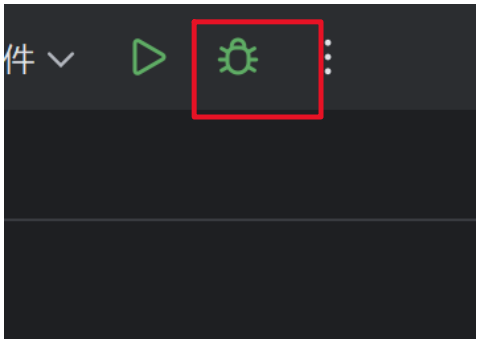


相对路径



debug（调试）

打上断点



再按这个小虫

- **步入（Step Into）**：

  当程序执行到某一行代码时，如果这行代码包含函数调用，执行“步入”操作会**进入该函数内部**，并停在函数的第一行代码处，方便调试函数内部的逻辑。

  例如：若当前行是 `result = add(1, 2)`，步入后会进入 `add` 函数的定义体（如 `def add(a, b):` 后的第一行）。

- **步出（Step Out）**：

  当程序正在函数内部执行时，执行“步出”操作会**继续执行当前函数的剩余代码**，直到函数执行完毕并返回调用处，然后停在函数调用语句的下一行。

  例如：在 `add` 函数内部调试时，步出后会执行完 `add` 函数，回到 `result = add(1, 2)` 的下一行代码。

1. **步过（Step Over）**

   当执行到包含函数调用的代码行时，“步过”会**执行整个函数但不进入函数内部**，直接停在该函数调用语句的下一行。

   例如：当前行是 `result = add(1, 2)`，步过后会计算完 `add` 函数的结果，直接停在 `result = ...` 的下一行，适合确认函数整体返回结果是否正确，无需关注函数内部细节。