实验报告

姓名：孔维凯　　学号：23020007051

2024 年 9 月 14 日

目录

1 实例

1.1 实例 1

1.1.1 练习内容

python 调试

1.1.2 结果

python 的调试器是 pdb，需要引入 pdb 库，并在要调试的地方加上 pdb.set_trace() 函数添加断点，这样当程序运行到这时，程序会停止，听从用户指令一步步运行，找出错误。

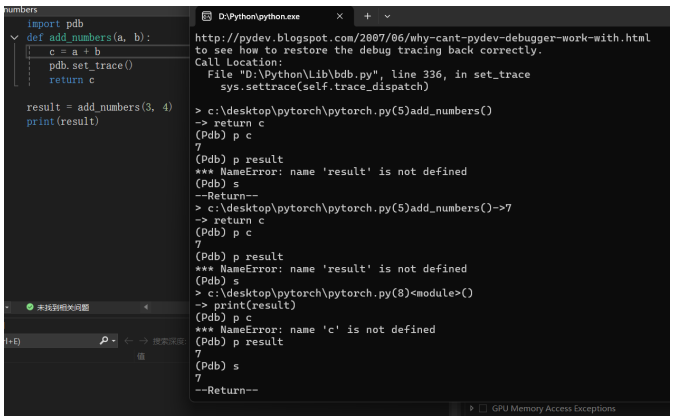

图 1: 例图 1

1.1.3 解题感悟

这个操作能直观地向用户显示出代码错误在哪里，便于用户修改。

1.2 实例 2

1.2.1 练习内容

python 通过时间进行性能分析

1.2.2 结果

第一种方法是通过记录整个程序的时间来分析其性能。首先引入 time 库，然后使用 time() 函数来记录运行时间。

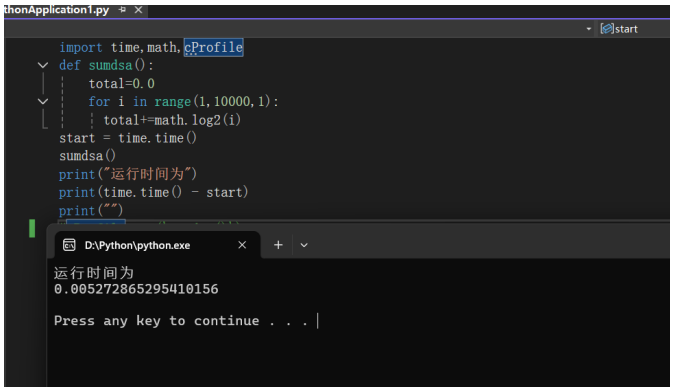

图 2: 例图 2

1.2.3 解题感悟

这种方法可以记录整个程序的时间来分析其性能。

1.3 实例 3

1.3.1 练习内容

用 time() 函数记录时间容易受其他因素干扰，如何更准确地记录时间。

1.3.2 结果

首先引入 cProfile 库，然后使用 cProfile.run(' 函数名') 函数来记录函数运行时间。

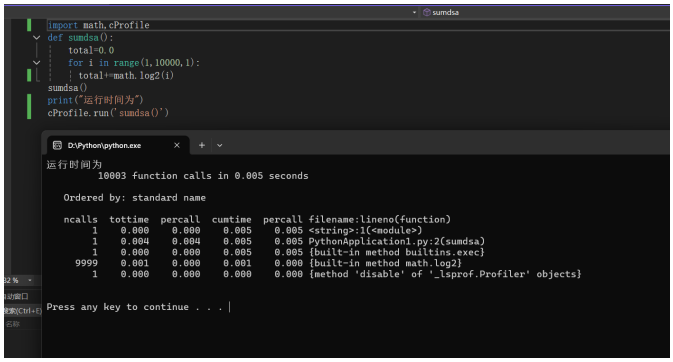

图 3: 例图 3

1.3.3 解题感悟

使用这种方法可以准确地得到函数运行时间来分析其性能。

1.4 实例 4

1.4.1 练习内容

pytorch 想要进行数据类型转换

1.4.2 结果

使用 data.type() 函数即可，type 为想要转化的类型，转化为 double 就将 type 换成 double，() 中为想要转化类型的变量。

1.4.3 解题感悟

通过这个函数，用户可以进行数据类型转换，防止系统报错。

1.5 实例 5

1.5.1 练习内容

如何进行代码调试

1.5.2 结果

最简单的方法就是每执行一步操作就输出一行文字，比如执行完某个函数就 cout«” 运行正常”，这样当程序出错时就可以找到出错的位置进行修改。

1.5.3 解题感悟

这样做可以快捷找到出错位置，在 C 语言学习中得到广泛使用，其他编码软件也同样适用。

1.6 实例 6

1.6.1 练习内容

如何创建一个零矩阵

1.6.2 结果

使用 torch.zeros() 可以创建一个零矩阵，() 中输入矩阵的长和宽。

1.6.3 解题感悟

这个函数能快速创建出一个符合用户要求的零矩阵。

1.7 实例 7

1.7.1 练习内容

如何创建一个随机初始化的矩阵

1.7.2 结果

使用 torch.rand() 函数可以创建一个矩阵，矩阵中的数据为随机生成，() 中输入矩阵的长和宽。

1.7.3 解题感悟

这个函数能使用户创建一个随机初始化的矩阵，满足用户需求。

1.8 实例 8

1.8.1 练习内容

如何进行张量加法和乘法。

1.8.2 结果

torch.add() 可以进行张量加法，() 中填入相加的两个元素，torch.matmul() 可以进行张量乘法，() 中填入相乘的两个元素。

1.8.3 解题感悟

通过这两个函数，用户可以进行一些复杂的操作。

1.9 实例 9

1.9.1 练习内容

想要转置张量即改变张量的维度顺序

1.9.2 结果

使用 torch.transpose() 函数进行张量的转置，() 中为要转置的张量。

1.9.3 解题感悟

这个操作帮助用户进行张量的转置，在用户处理高维数据时极为重要。

1.10 实例 10

1.10.1 练习内容

如何对新创建的张量求导

1.10.2 结果

首先创建一个张量，然后修改张量的 requires_grad 属性为 True，使用 torch.tensor(,requires_grad=True) 来自动求导。

1.10.3 解题感悟

使用这个函数形式，用户可以对创建的这个张量进行求导。

1.11 实例 11

1.11.1 练习内容

如何获取张量的元素

1.11.2 结果

通过使用 tensor[] 函数可以获取函数，注意标号从 0 开始，tensor[3,4] 获取第 4 行第 5 列的元素，而 tensor[:,2] 获取所有行第 3 列的元素，通过 print 函数显示出来。

1.11.3 解题感悟

这些函数方便用户快速获取张量的元素。

1.12 实例 12

1.12.1 练习内容

如何进行矩阵运算

1.12.2 结果

通过使用 tensor.sum() 函数可以对所有元素求和，通过 print 函数将其显示出来。

1.12.3 解题感悟

这个函数极大方便了用户对矩阵的运算。

1.13 实例 13

1.13.1 练习内容

如何进行广播机制也就是将张量的每个元素都加 1。

1.13.2 结果

使用 tensor+1 可以进行广播机制，快捷地将张量的每个元素都加一。

1.13.3 解题感悟

这个操作能快捷地帮助用户对张量的每个元素进行加 1 操作，方便快捷。

1.14 实例 14

1.14.1 练习内容

如何修改键位映射

1.14.2 结果

首先按下 windows+R 组合键，在对话框中输入 regedit 打开注册表，
进入目录 HKEY_LOCAL_MACHINE SYSTEM CurrentControlSet Control Keyboard Layout, 右键点击新建二进制值，名称为 Scancode Map，打开后按需求输入 8 列数据修改键位即可。

1.14.3 解题感悟

通过这些操作，用户可以将按键修改成自己习惯的形式。

1.15 实例 15

1.15.1 练习内容

常见命令行标志参数及模式

1.15.2 结果

想要使用 rm 删除一个名为-r 的文件需要使用–符号，即使用 rm – -r 指令才能删除一个名为-r 的文件。

1.15.3 解题感悟

有些文件不能使用常规的方法删除，需要添加特殊符号来辅助删除。

1.16 实例 16

1.16.1 练习内容

如何用命令行进行文件的复制

1.16.2 结果

使用 cp 文件 1 文件 2 指令可以将文件 1 的内容复制到文件 2 上。

1.16.3 解题感悟

使用这种方式复制文件是对命令行的简单技巧，但方便快捷。

1.17 实例 17

1.17.1 练习内容

如何使用特殊参数

1.17.2 结果

$0 用来表示当前命令的名称，$+ 数字（除了 0）用来表示位置参数，$ 用来表示位置参数的数目，$? 用来显示前一条指令的结果。

1.17.3 解题感悟

使用这些特殊参数能为用户提供更多便捷。

1.18 实例 18

1.18.1 练习内容

如何在 github 上创建一个个人仓库

1.18.2 结果

首先登录 github，然后点击我的仓库，输入仓库名称，可以选择仓库的协议来达成用户的目的，然后创建仓库，可以通过拖拽将要上传的文件传到仓库中，如果是公共仓库，其他人可以通过搜索查看你的仓库，使用你的代码。

1.18.3 解题感悟

github 是一个非常开源的软件，许多人都在上面分享代码，用户通过这种方式也可以向其他人分享自己的代码。

1.19 实例 19

1.19.1 练习内容

pytorch 如何保存模型并读取

1.19.2 结果

使用 torch.save（模型名，' 文件名'）函数可以将整个模型保存到硬盘上，然后可以使用 torch.load() 函数读取，保存分为两种模式一种就是上文说的，另一种就是使用模型名.state_dict() 函数替换模型名就可以只保存模型的参数了。

1.19.3 解题感悟

使用这几个函数能完成模型的保存和读取操作，方便用户随时使用。

1.20 实例 20

1.20.1 练习内容

LaTeX、Markdown 中的标记失效

1.20.2 结果

Markdown 使用标记语言时，需要注意一些用法，比如有些标记后需要加上空格才能生效（标题标记 ，后面需要加上空格才能正常对标题进行标记）。

LaTeX 使用 _ 时需要在前面加上斜杠才能正常使用，再比如 LaTeX 中两个斜杠表示换行。

1.20.3 解题感悟

使用 Markdown 和 LaTeX 时需要注意一些细节，否则无法达到预期效果。

2 github 链接