



进阶实验篇第1章

Python矩阵乘法

1

矩阵乘法的Python实现

2

计时方法：函数调用

3

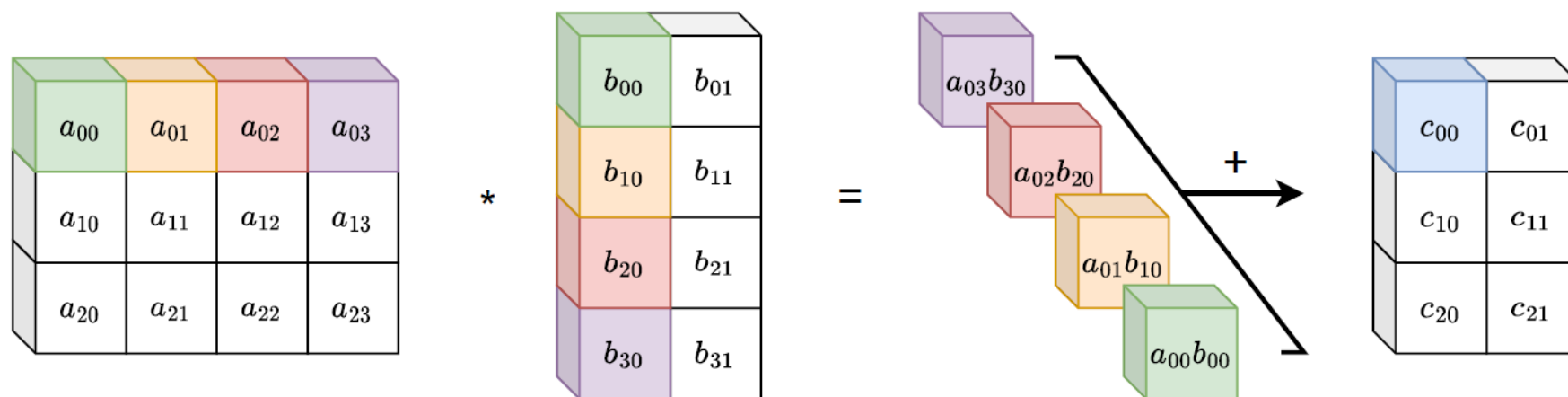
对比表格：简单数据分析的工具

4

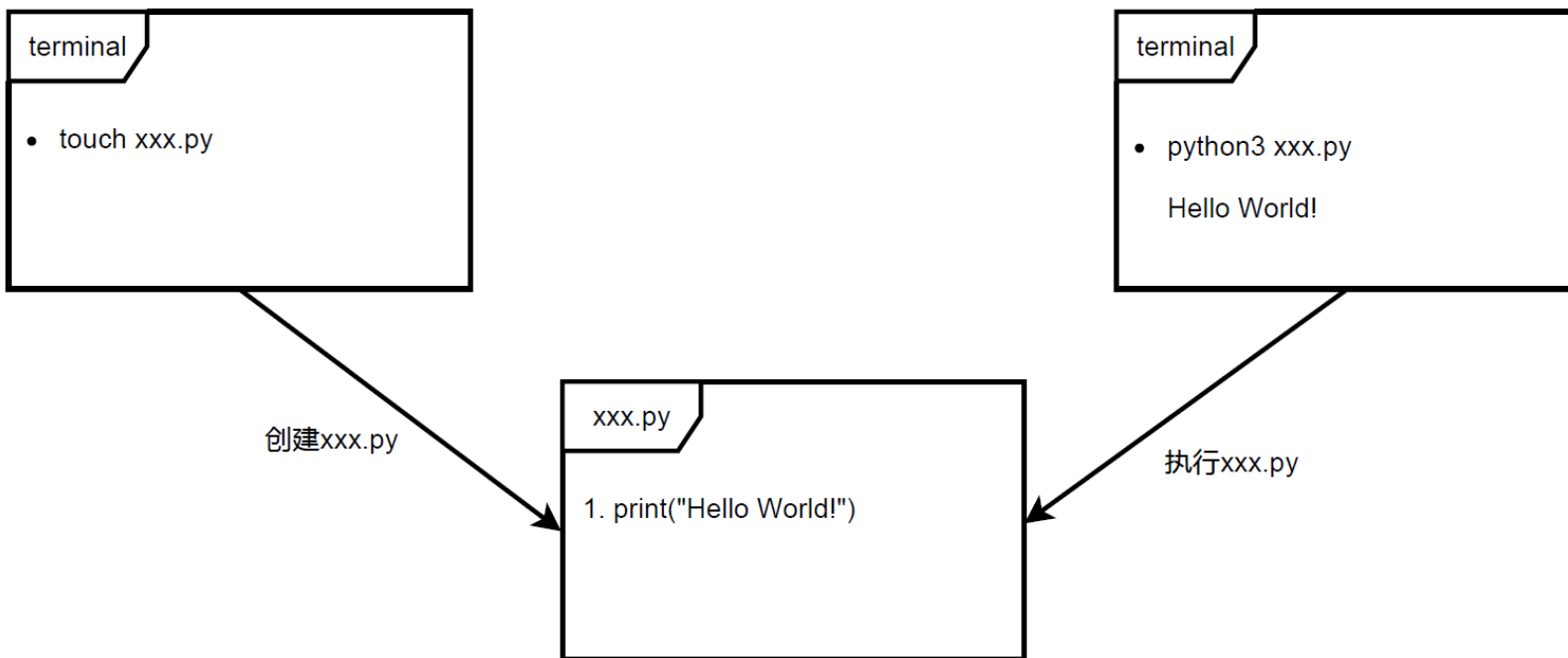
性能分析：Python的解释执行机制



矩阵乘法的Python实现



- 矩阵乘法是神经网络的核心算子，其性能决定人工智能的效率。
- Python作为一种主流编程语言，可以很简便地实现矩阵乘法。



- 在终端中使用touch命令可以创建文件。
- 使用python3 + 文件名可以执行Python文件。

```
1. for i in range(rows of matrix A):  
2.     for j in range(cols of matrix B):  
3.         for k in range(cols of matrix A or cols of matrix B):  
4.             C[i][j] += A[i][k] * B[k][j]
```

- 使用for循环和range对数据进行遍历，for i in range(3)则i会依次赋值为0, 1, 2。
- 使用列表list用于存储数据，A[i][k]表示第i行第k个数据。

计时方法：函数调用

xxx.py

```
1. import time
2. begin = time.perf_counter()
3. mm_mul(A, B)
4. end = time.perf_counter()
5. result = end - begin
```

- 使用import time导入time库。
- 在调用待测函数mm_mul前调用perf_counter记录调用前时间。
- 在调用测试函数后再次计时。
- 最终计算时间差得到测试函数运行之间。



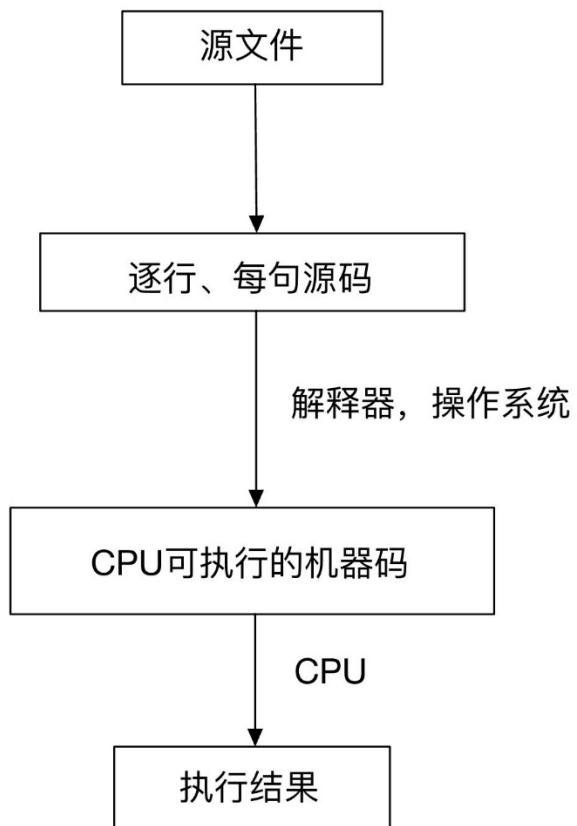
对比表格：简单数据分析的工具

矩阵大小(n*n)	32	64	128	256
预计周期数	32850930	282575930	2507124780	22770312830
预期耗时/s	0.0037683200	0.0301465600	0.2411724800	1.9293798400
实际耗时/s	0.0152830110	0.0854666360	0.5584707610	5.0275840600

- 表格可以较为清晰地用于简单的数据对比。
- 通过上表的数据对比得知，使用Python实现的矩阵乘法耗时比理论预期要长很多。



性能分析：Python的解释执行机制



- Python程序效率低的一个重要原因是其解释执行的机制。
- 解释器逐行逐句翻译源代码，翻译后的源码由操作系统翻译为目标CPU可执行的二进制机器码。
- CPU执行二进制机器码。
- 计时记录的时间不仅是CPU执行机器码的时间，还包括了解释器解释源码的时间。



本章作业

程序设计

利用python语言编程实现矩阵乘法

1. 利用python中的循环语句（不要使用库函数）实现矩阵乘法
2. 测量不同尺寸的矩阵的乘法运算时间，并使用数据表格呈现结果

观察并完成实验报告

1. 比对numpy等库函数中提供的矩阵乘法与你自己实现的矩阵乘法的性能差异



感谢阅读
