南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

SUSTech
Southern University of Science and Technology

机器学习场景下的Python及其常用工具包

Zhiyuan Wang

wangzy2020@mail.sustech.edu.cn

Numba

- 开源的JIT(Just-in-time)编译工具
- Python为什么慢?
  - 解释运行，无法基于上下文对二进制机器码进行优化
- Numba做了什么?
  - 在正式运行代码前，将一个函数整体进行编译，从而加速函数的执行
- 哪里适合用Numba?
  - 高计算量，高调用频率的函数
- 参考资料：https://numba.readthedocs.io/en/stable/index.html

Numba

矩阵规模	Python	Numba	NumPy
100	0.073s	0.634s	0.001s
200	0.619s	0.059s	0.005s
500	11.87s	0.815s	0.028s
1000	114.5s	8.571s	0.111s

- NumPy能解决的，优先用NumPy
- 计算开销小的代码，用Numba会增加编译开销
- Numba的使用对代码中使用的数据类型有严格要求

```python
import time
from typing import List
import random
from numba import jit
Mat = List[List[float]]

def python_method(a: Mat, b: Mat):
    result: Mat = [[0 for _ in range(len(a))] for _ in range(len(b[0]))]
    py_time = time.time()
    for r in range(len(a)):
        for c in range(len(a)):
            result[r][c] = sum([a[r][i] * b[i][c] for i in range(len(b))])
    print("原生Python时间：", time.time() - py_time)

@jit # 加上这个注解函数就会被编译执行
def numba_method(a: np.ndarray, b: np.ndarray):
    result: Mat = [[0 for _ in range(len(a))] for _ in range(len(b[0]))]
    for r in range(len(a)):
        for c in range(len(a)):
            result[r][c] = sum([a[r][i] * b[i][c] for i in range(len(b))])
if __name__ == '__main__':
    for size in [100, 200, 500, 1000]:
        a: Mat = [[random.random() for _ in range(size)] for _ in range(size)]
        b: Mat = [[random.random() for _ in range(size)] for _ in range(size)]
        python_method(a, b)
        numba_time = time.time()
        print("Numba时间：", time.time() - numba_time)
```