知能情報実験 1 : レポート課題 4

205713B 朝比奈 太郎

2021 年 5 月 29 日

目次

1　目的

Python 科学ライブラリ Numpy の add 関数を使用した場合と Python の標準ライブラリの加算演算子 (+) ををを使用して要素ごとに加算した場合の処理時間を比較し、それぞれの使用用途を的確にするため。また、加算演算子があるのにもかかわらず Python 科学ライブラリ Numpy の add 関数が存在しているということは、add 関数の方が計算スピードにおいて優秀であるという自分の考えを確かめるため。

2　方法

2.1　Numpy の add 関数を使用する場合

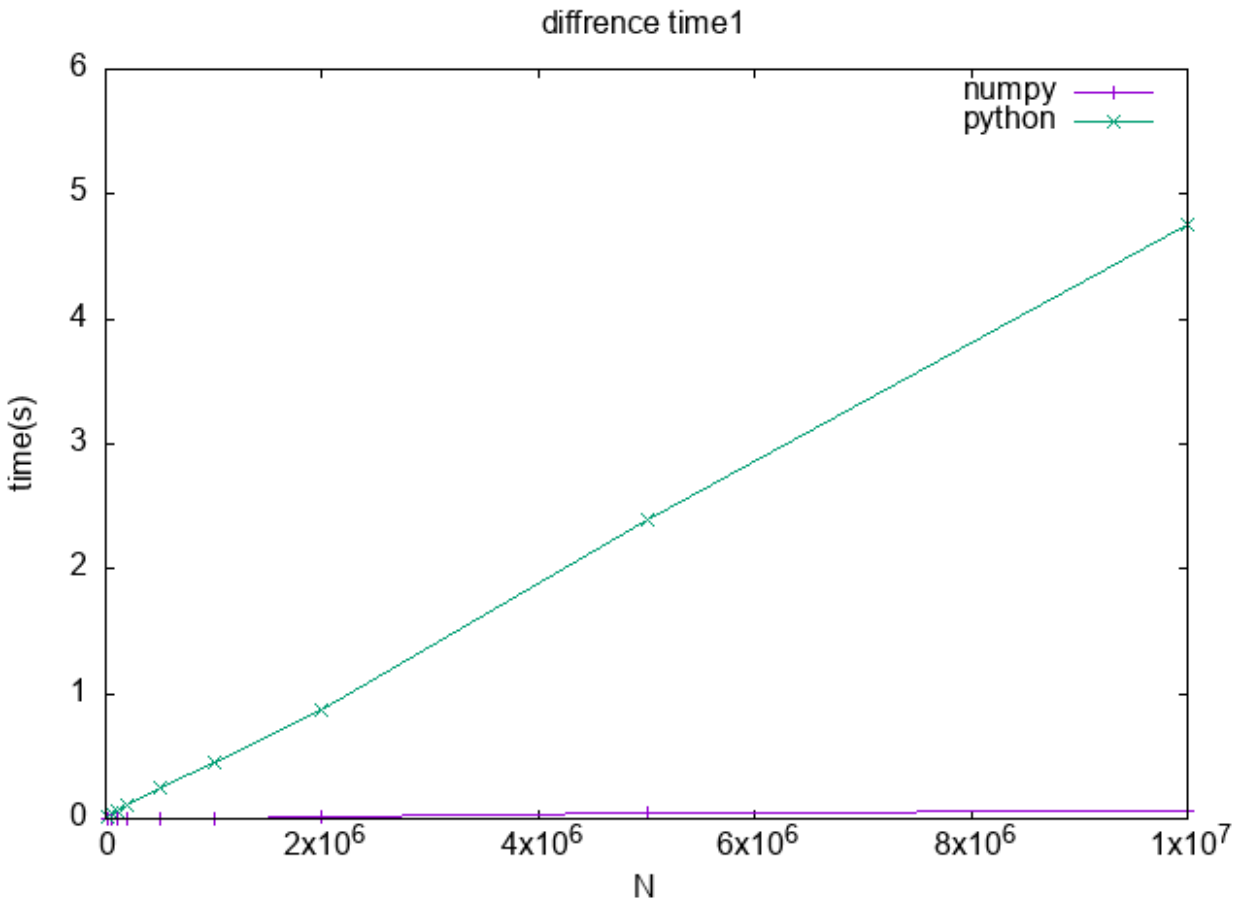
import した time の t0 = time.time() を add 関数の前の行に入れ、add 関数の後ろの行に t1 = time.time() を入れ、t1 - t0 で add 関数における行列演算にかかった時間を求める。

2.2　加算演算子 (+) を使用する場合

import した time の t0 = time.time() を加算演算子を行う for 文の前の行に入れ、加算演算子を行う for 文を抜けた次の行に t1 = time.time() を入れ、t1 - t0 で加算演算子 (+) における行列演算にかかった時間を求める。

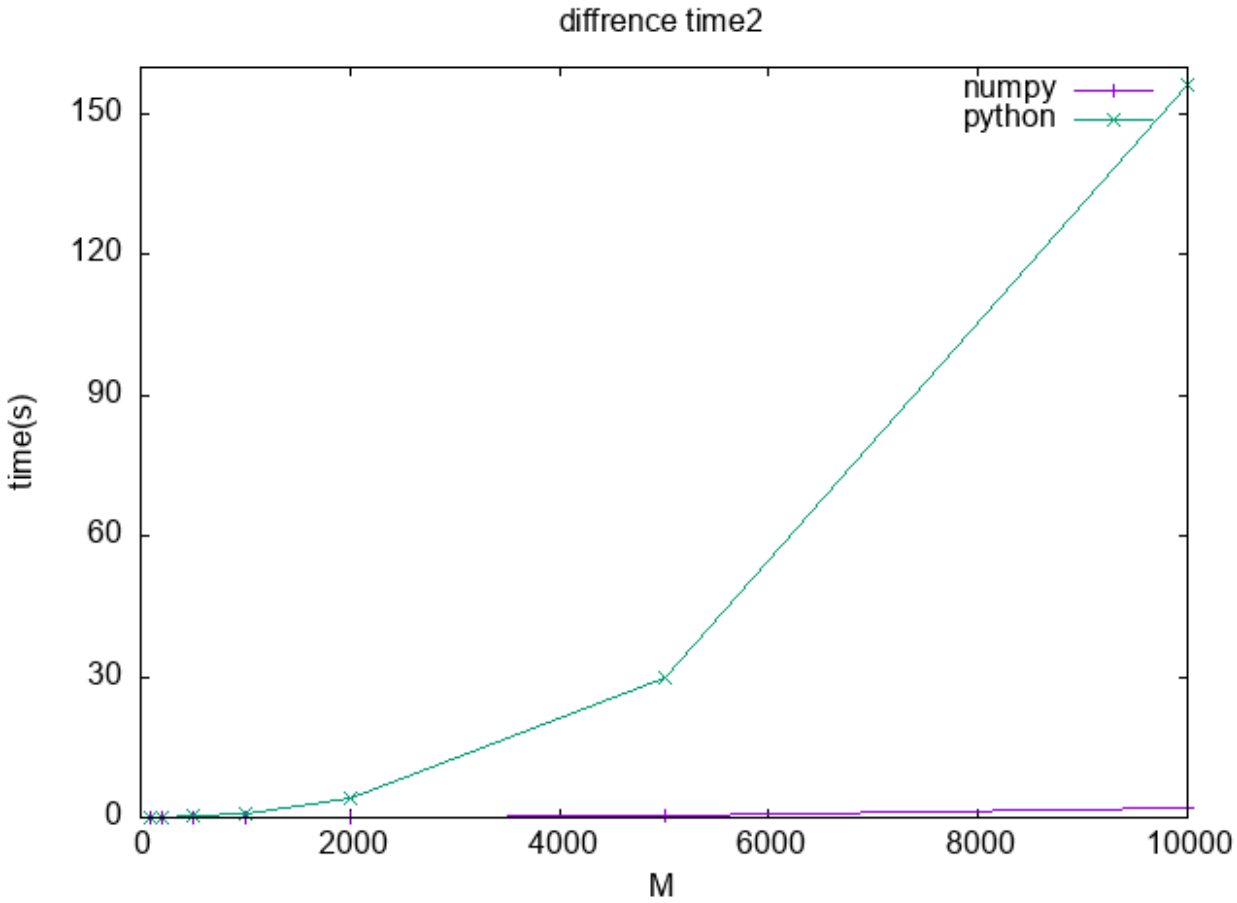
3　結果

3.1　(1,N) の 1 次元行列



上図 (diffrence time1) は,x 軸が N, y 軸が time(s) となっている。numpy,python を指すグラフは共に 1 次関数となっており、python のグラフの傾きが numpy のグラフの傾きよりも大きいことから、numoy(add 関数) よりも、python(加算演算子) の方が計算にかかる時間が長いといえる。従って、1 次元行列の計算をする際には Python 科学ライブラリ Numpy の add 関数を使用する方が計算速度が早くて良いと言える。

3.2　(M,M) の 2 次元行列



上図 (diffrence time2) は,x 軸が N, y 軸が time(s) となっている。numpy のグラフは、diffrence time1 と同様に 1 次関数になっているが、python のグラフは、 2 次関数のような形をとっている。M が 0 から 1000 までの間は numpy のグラフと python のグラフの time(s) に関する値に目視できるほどの差はないが、M が 1000 付近になると numpy と python のグラフの time(s) をとる値に差が出始め、それ以降 python のグラフが numpy のグラフに time(s) において大きく差が出る。従って、正方行列の計算をする際には M が 1000 未満の際は加算演算子 (+) と Numpy の ad 関数のどちらを用いても良いが、それ以上になる際には、Python 科学ライブラリ Numpy の add 関数を使用する方が計算速度が早くて良いと言える。

3.3 ソースコード

Listing 1　ソースコード

```
import numpy as np
import time


N = np.power(10,4)
A = np.random.rand(N, N)
B = np.random.rand(N, N)

print("A␣=␣{}".format(A))
print("B␣=␣{}".format(B))

X = np.zeros((N,N), np.float64)
t0 = time.time()
X = np.add(A, B)
t1 = time.time()
tx = t1 - t0
print("X␣=␣{}".format(X))
print("tx␣=␣{}".format(tx))

Y = np.zeros((N,N), np.float64)
t0 = time.time()
for i in range(N):
    for k in range(N):
        Y[i][k] = A[i][k] + B[i][k]
t1 = time.time()
ty = t1 - t0
print("Y␣=␣{}".format(Y))
print("ty␣=␣{}".format(ty))
```

4　考察

上図 (diffrence time1,diffrenve time2) より、計算する量 (N,M) が大きくなるにつれ add 関数が加算演算子より早く計算できることがわかった。計算する量が少なくても、add 関数と加算演算子ではかすかに add 関数の方が計算スピードが早いので、NumPy を利用するべきだと言える。

参考文献

[1] 國田 樹, 2021_StuLab1_理工系のレポート作成技術 , 2021/05/29.

[2] Latex 入門/図表, https://texwiki.texjp.org/?LaTeX, 2021/05/29