week6实验记录

张新鹏

October 28, 2023

1 实验环境

cpu：Inter i5-12400f

操作系统：windows 10 专业版64位

编译器：gcc version 8.1.0(x86_64-posix-seh-rev0, Built by MinGW-W64 project)

2 结果

2.1 使用不同大小的Tensor1D

Table 1: input and filter tensors of size 1×100×1000×1000,the time unit is ms

	-0	-O2	-O3	-Ofast -march=native
run time	937.278	326.906	314.968	331.516
GFLOPS	0.213384	0.611797	0.634985	0.603289

Table 2: input and filter tensors of size 1×500×1000×1000,the time unit is ms

	-0	-O2	-O3	-Ofast -march=native
run time	4758.94	1860.55	1743.91	1899.73
GFLOPS	0.210131	0.537475	0.573424	0.526391

Table 3: input and filter tensors of size 1×1000×1000×1000,the time unit is ms

	-0	-O2	-O3	-Ofast -march=native
run time	18691.1	17361	14800.1	13388.9
GFLOPS	0.107003	0.115201	0.135134	0.149377

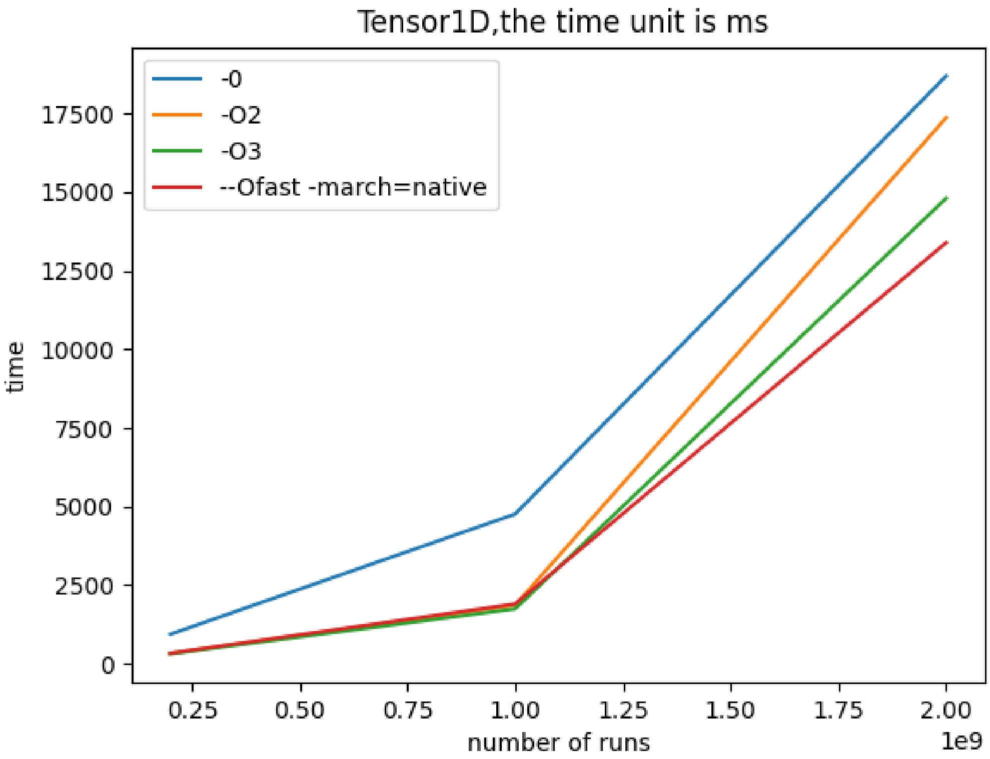

Figure 1: Tensor1D,the time unit is ms

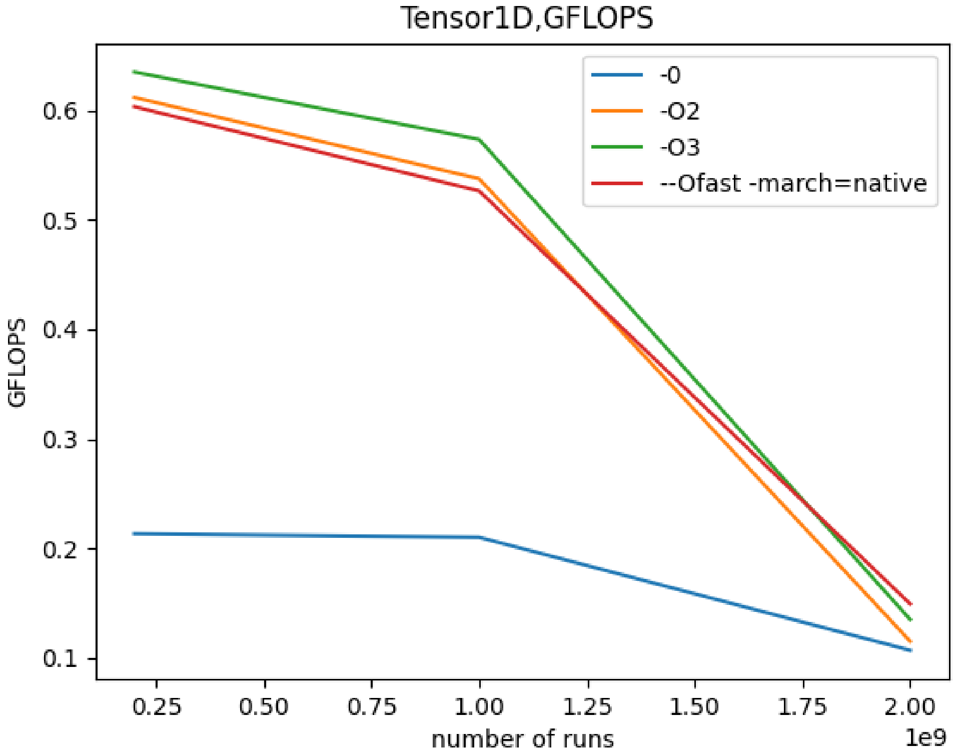

Figure 2: Tensor1D,GFLOPS

2.2 使用不同大小的Tensor4D

Table 4: input and filter tensors of size 1×100×1000×1000,the time unit is ms

	-0	-O2	-O3	-Ofast -march=native
run time	1441.01	339.863	351.695	342.796
GFLOPS	0.138792	0.588472	0.568675	0.583437

Table 5: input and filter tensors of size 1×500×1000×1000,the time unit is ms

	-0	-O2	-O3	-Ofast -march=native
run time	7240.47	1897.1	1996.94	1853.26
GFLOPS	0.138113	0.52712	0.500766	0.53959

Table 6: input and filter tensors of size 1×1000×1000×1000,the time unit is ms

	-0	-O2	-O3	-Ofast -march=native
run time	29123.3	17802.9	15473.4	10340.9
GFLOPS	0.0686735	0.112341	0.129254	0.193407

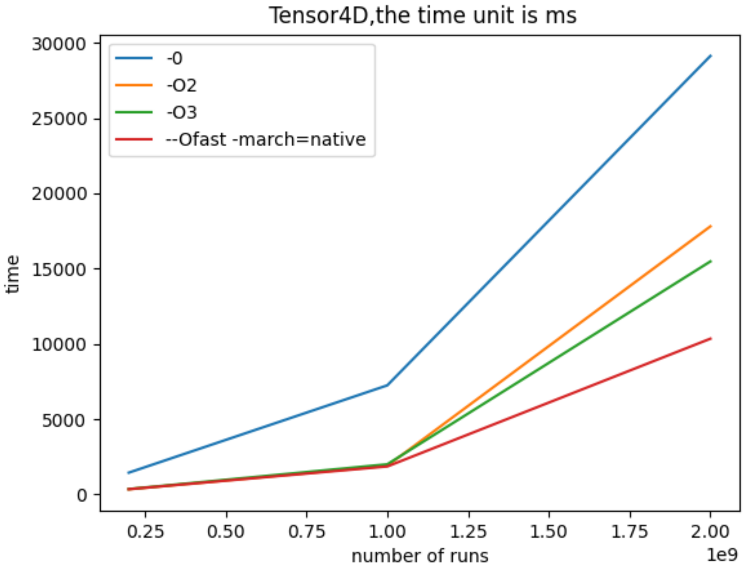

Figure 3: Tensor4D,the time unit is ms

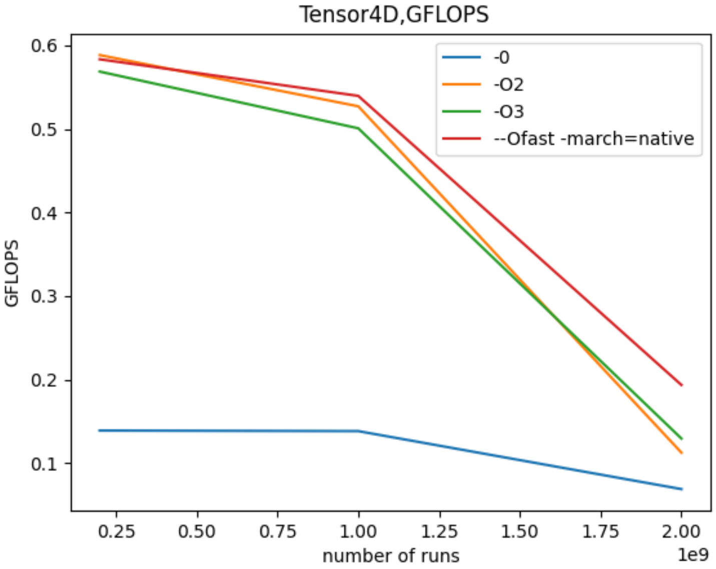

Figure 4: Tensor4D,GFLOPS

2.3 比较Tensor1D和Tensor4D

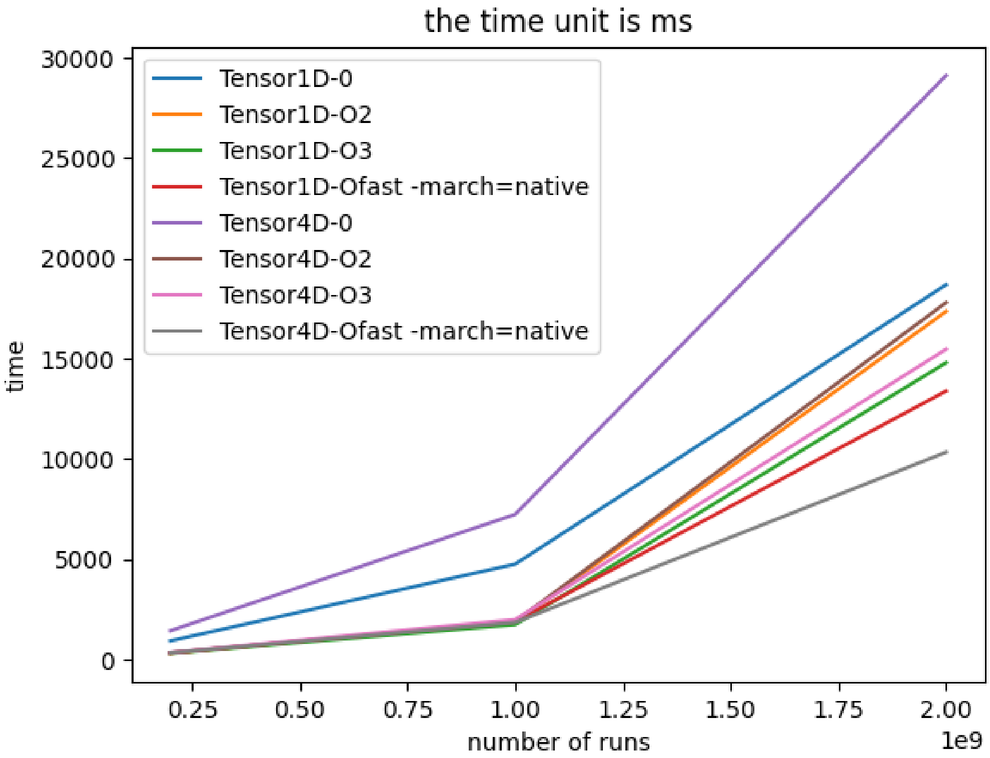

Figure 5: the time unit is ms

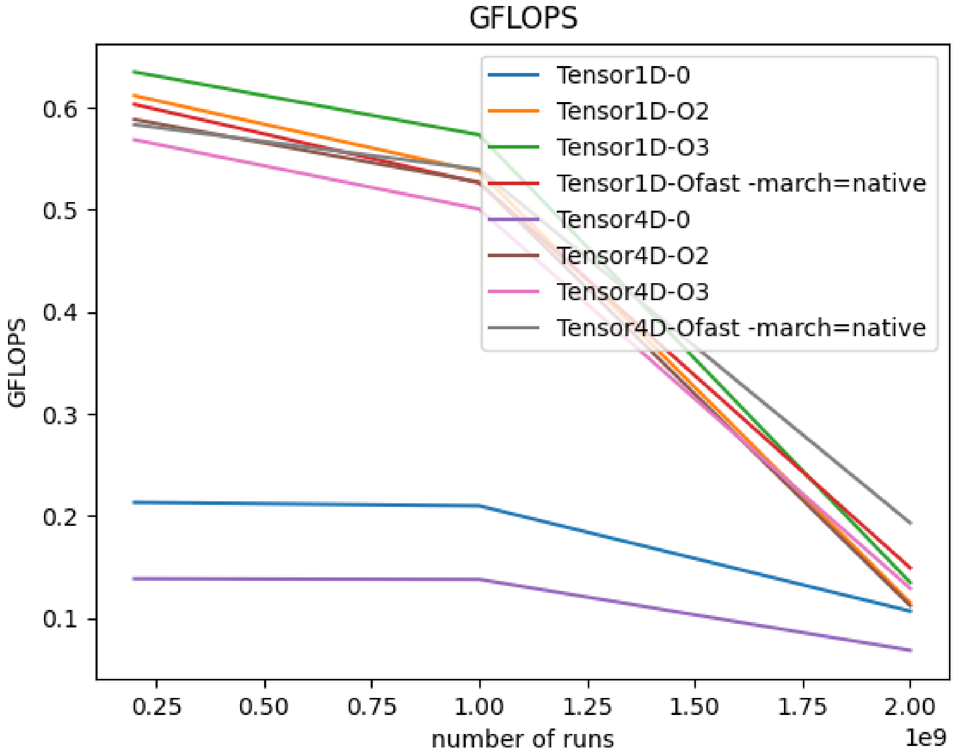

Figure 6: GFLOPS

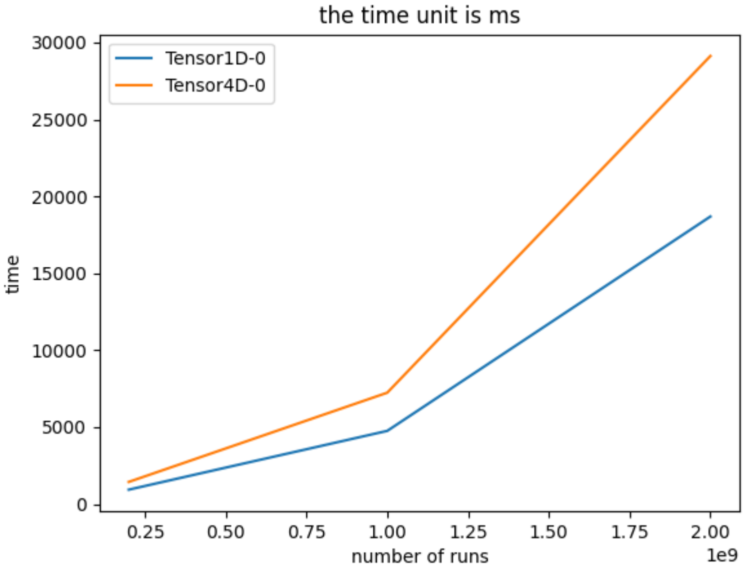

Figure 7: -0,the time unit is ms

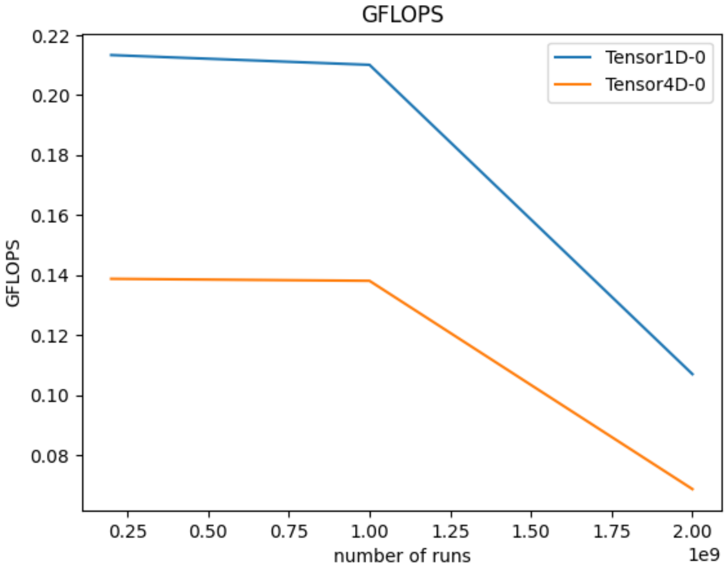

Figure 8: -0,GFLOPS

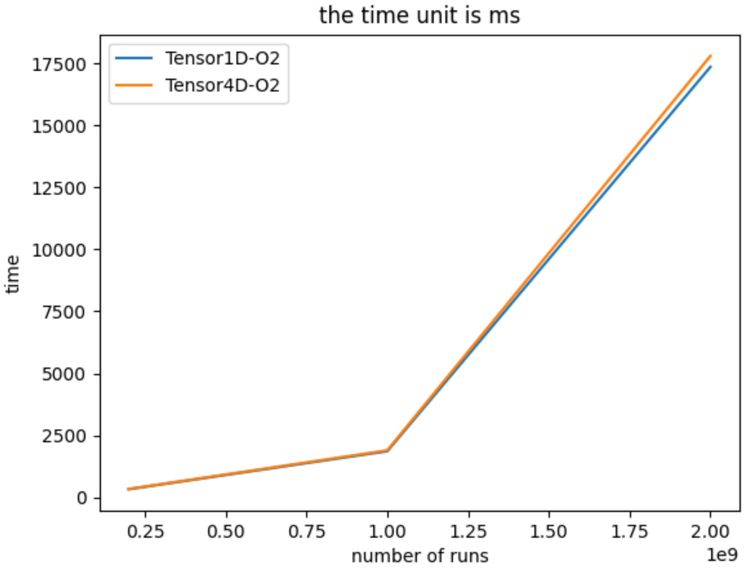

Figure 9: -O2,the time unit is ms

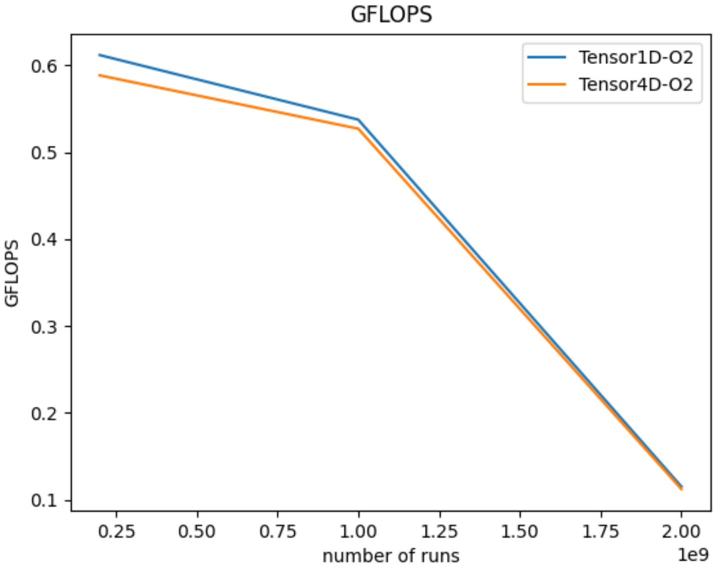

Figure 10: -O2,GFLOPS

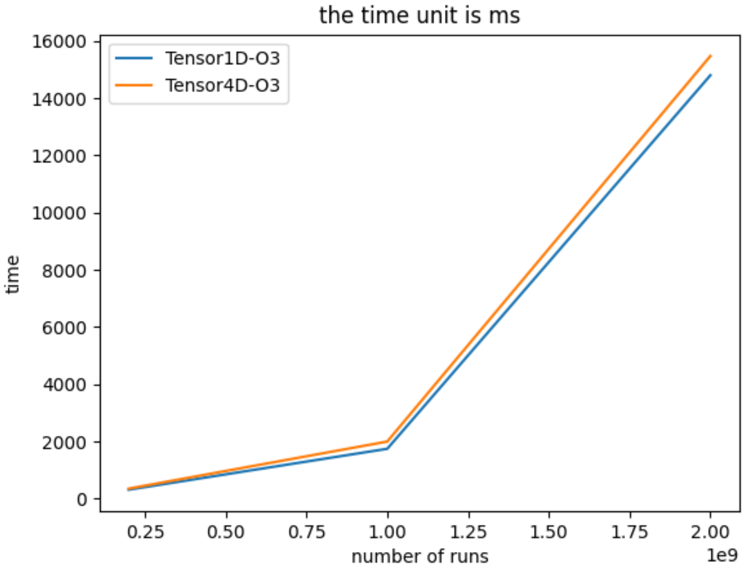

Figure 11: -O3,the time unit is ms

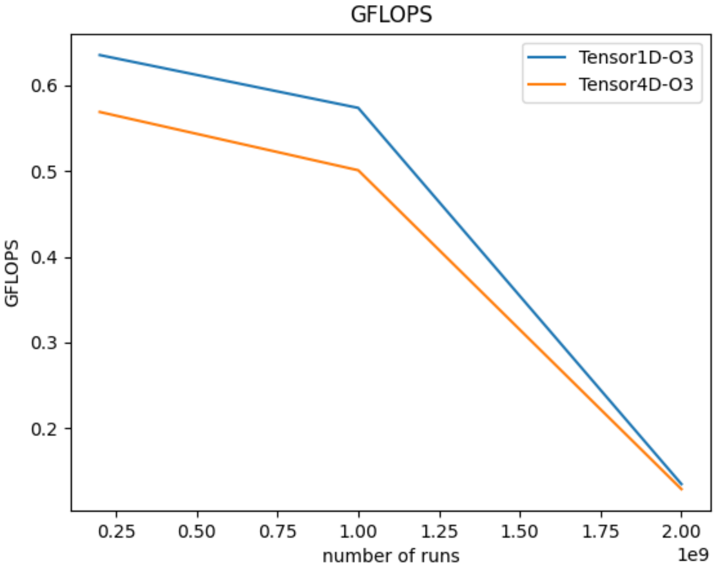

Figure 12: -O3,GFLOPS

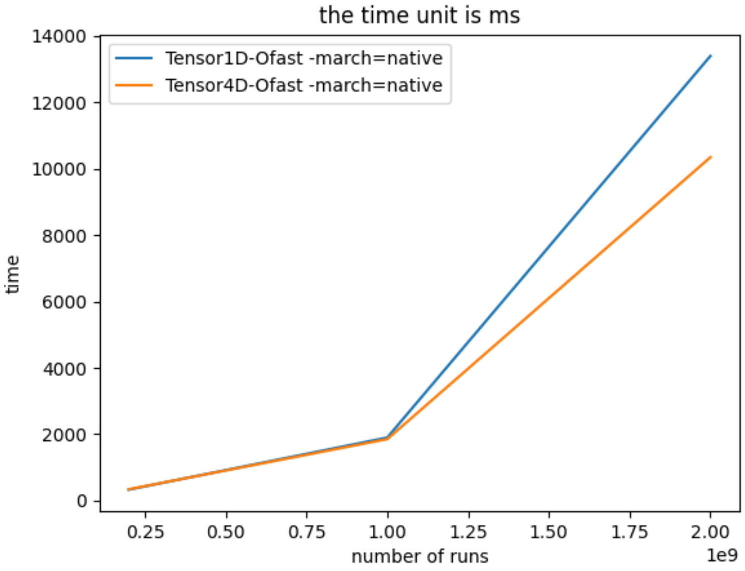

Figure 13: -Ofast -march=native,the time unit is ms

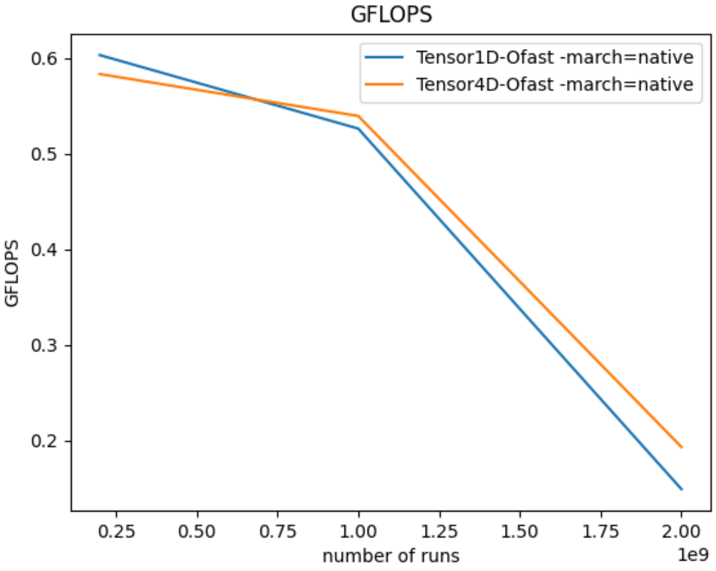

Figure 14: -Ofast -march=native,GFLOPS

3 结论

-Ofast会启用所有的优化选项，march=native编译器自动探测目标架构并生成针对目标架构优化的目标代码。毫无疑问，优化级别越高，优化开的越多，运行速度越快。-0-O2-O3,都是用1维储存的效果好。但是，-Ofast -march=native优化下用1维效果表现比4维差，于是我多跑了几次，在这个优化下使用Tensor4D每次花费的时间差距很大，使用1×1000×1000×1000的输入张量和卷积核，运行三次分别用时5946.86ms，13050ms，10340.9ms。猜测是受到后台运行其他应用影响比较大。