

week15实验记录

zxp

December 30, 2023

1 environment

cpu:Inter i5-12400f (2.5 GHz)

System:Ubuntu 22.04.1

Compiler:gcc 12.3

2 code

现在统一使用size_t。之前的tensor1d和tensor4d的NCHW是int64_t，但wetensor使用的是size_t，为了测试这些数据结构，所以在直接卷积的七层循环中有很多int64_t和size_t互相转化。现在抛弃了tensor1d和4d，使用tensor_1d，于是统一使用size_t。size_t没有负数和实际情况相符，使用size_t更加合理。

增加了FTensor_1D(tensor_1d的float版本)ITensor_1D(tensor_1d的int版本)STensor_1D(tensor_1d的short版本) 新增tensor_1d_v，将数据类型动态数组改成vector数组

```
第1批次
第1通道
1.40686 d 1.36709 d
7.48089 d 8.58368 d
数据类型:d 字节数:8
第1批次
第1通道
9.25185 f 1.49745 f
5.05165 f 8.21513 f
数据类型:f 字节数:4
第1批次
第1通道
1 i 9 i
7 i 6 i
数据类型:i 字节数:4
第1批次
第1通道
4 s 9 s
8 s 2 s
数据类型:s 字节数:2
```

Figure 1: 验证正确性

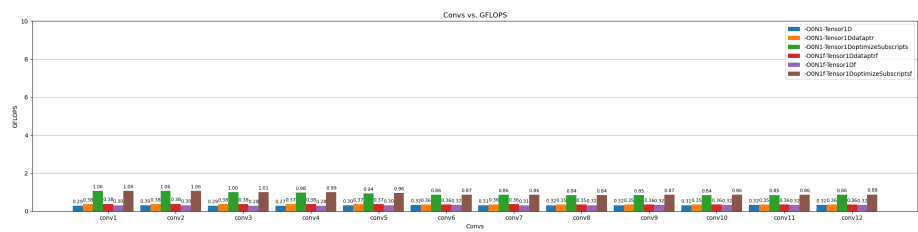


Figure 2: O0

3 Experiment

对比float版本和double版本的tensor_1d。

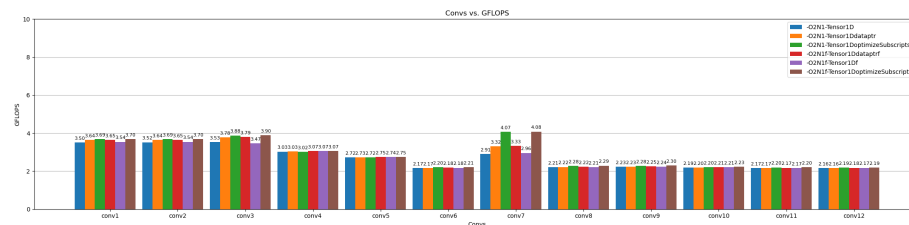


Figure 3: O2

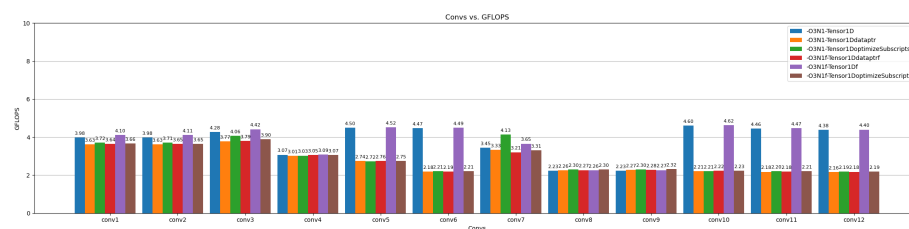


Figure 4: O3

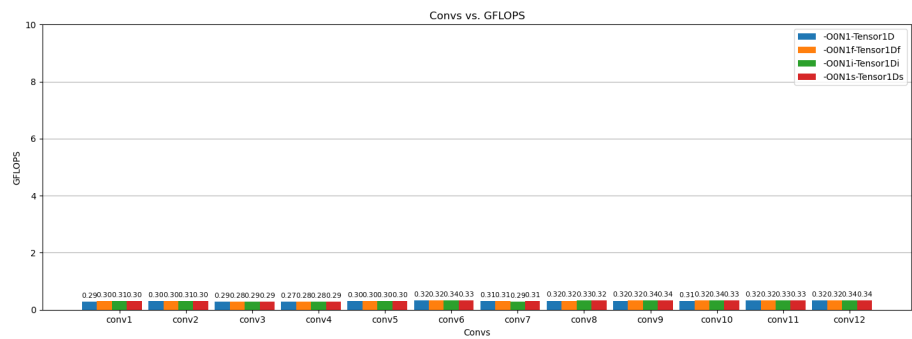


Figure 5: O0

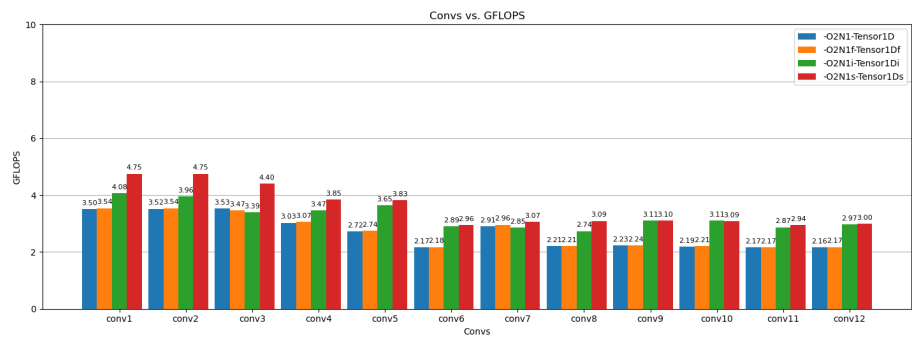


Figure 6: O2

然后增加对比int版本和short版本

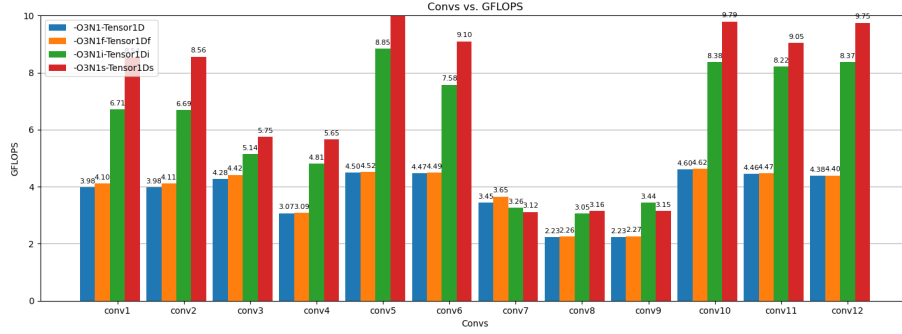


Figure 7: O3

3.1 Analysis

使用float并没有和预期那样比double快很多，大部分情况下就几乎可以忽略不计的快一点点。而int和short在O2和O3下的表现就好很多，short也明显比int快，但也没有出现因为位数减少一半而翻倍的情况。我认为这是因为CPU并没有因为因为数据结构的位数减半而在在搬运数据的时候多搬运几个进寄存器。但如果使用SIMD，在数据位数减半的时候能多搬运一倍，我认为如果使用SIMD指令能达到预期效果。

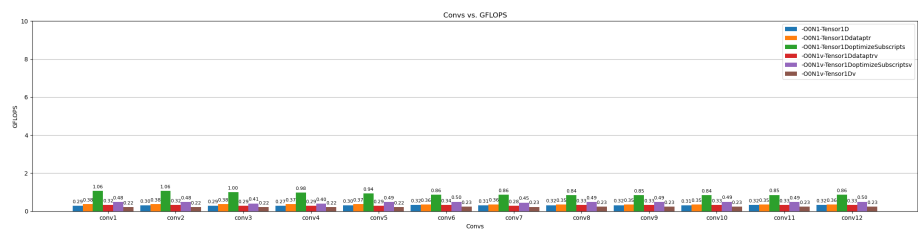


Figure 8: O0

4 Experiment2

对比动态数组和vector数组

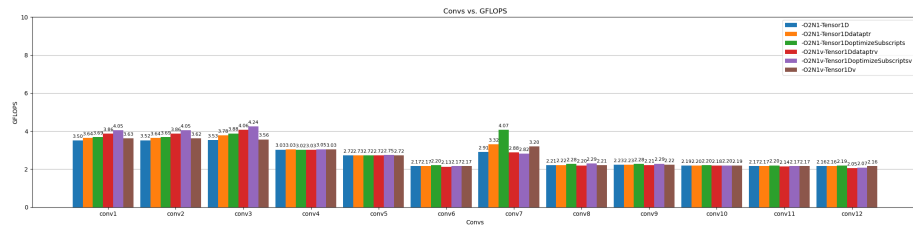


Figure 9: O2

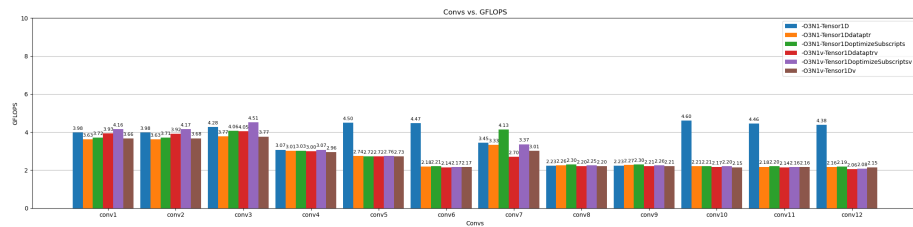


Figure 10: O3

4.1 Analysis

vector数组的性能要比动态数组差。虽然vector数组底层也是动态数组，但vector数组是安全的数组包装了很多东西，所有性能有所下降。

5 Experiment3

矩阵的规律，conv3和conv5为什么在循环顺序改变后表现好和conv9和conv10为什么在循环顺序改变后表现差，还没有思路。