

week17实验记录

zxp

January 13, 2024

1 environment

cpu:Inter i5-12400f (2.5 GHz)

System:Ubuntu 22.04.1

Compiler:gcc 12.3

2 code

更改了Tensor1D（解决了内存泄漏问题，重写给元素赋值随机数和子类的构造函数），新增了一个resize方法（改变数组大小，步骤为释放之前的申请的内存然后申请新的）用于Im2win变换的时候使用。

实现了使用Tensor1D做数据结构的Im2winConvolution。

将之前卢帅师兄写的使用WeTensor做数据结构的Im2winConvolution跑起来了（有一个Convolution类，Im2winConvolution也在里面，注释掉了使用了cblas.h的代码）。

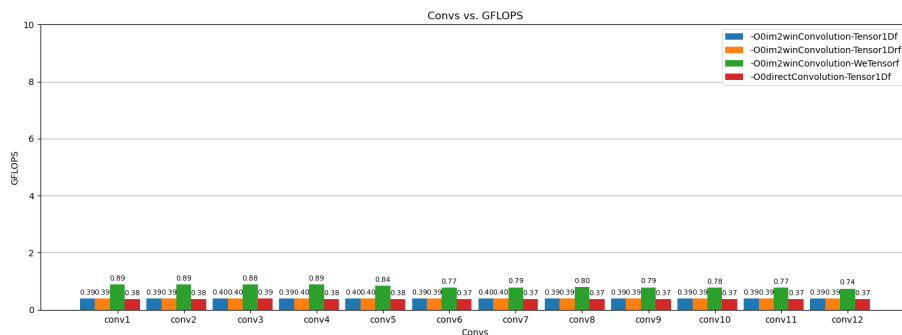


Figure 1: O0

3 Experiment

这周主要是实现了最朴素的Im2win。实现了使用Tensor1D存数据的im2win。然后和最朴素的使用Tensor1D存数据的直接卷积对比，用的接近的循环顺序（将filter的width放在最里面，filter的height作为内二层）。因为数据布局的问题（im2win变化后的张量在filter的height上连续的，直接卷积在filter的width上是连续的），这样对im2win不是很公平，于是又测了调换了历遍filter的height和width的循环顺序的版本。将这三个加上WeTensor做数据结构的Im2winConvolution放在一起对比（直接使用了卢帅师兄的代码，这个代码的实现的im2win并不是完全没有优化，循环顺序不一样，做了index hoisting，还有openmp[测的时候未启用openmp]）。

然后测试了一下im2win转换的时间，将这个时间和在整个im2win卷积所用的时间做了对比

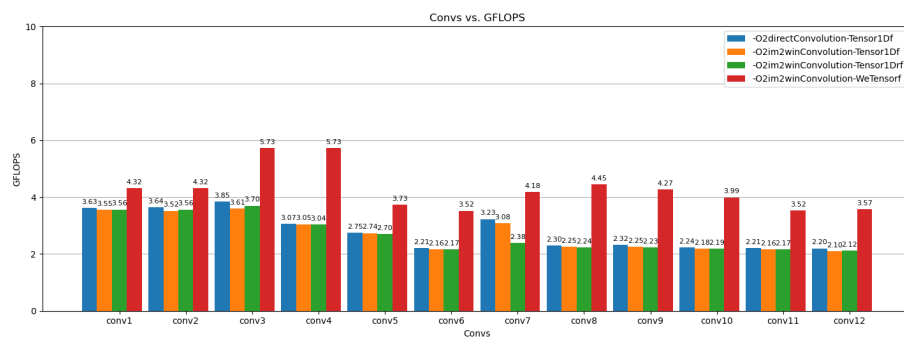


Figure 2: O2

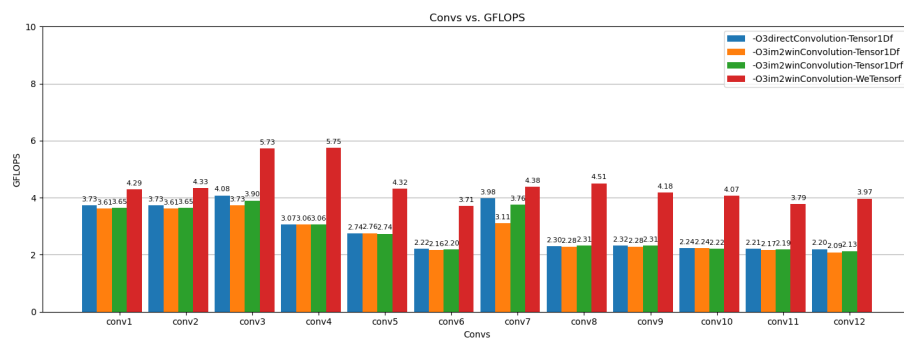


Figure 3: O3

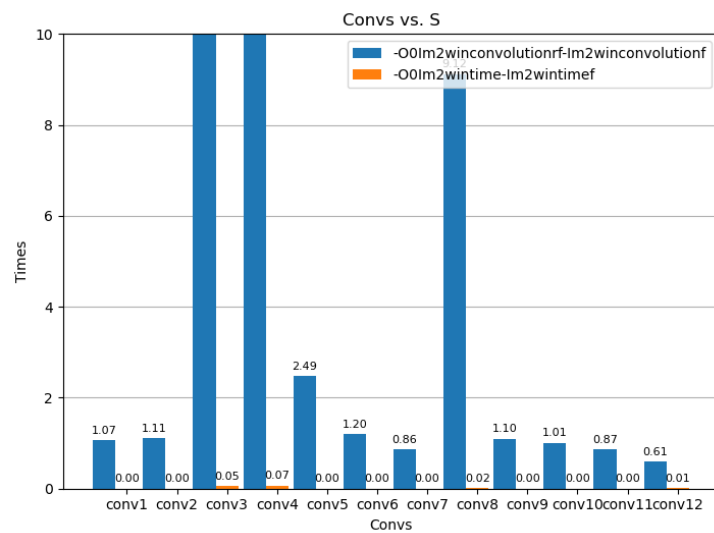


Figure 4: O0情况下的时间

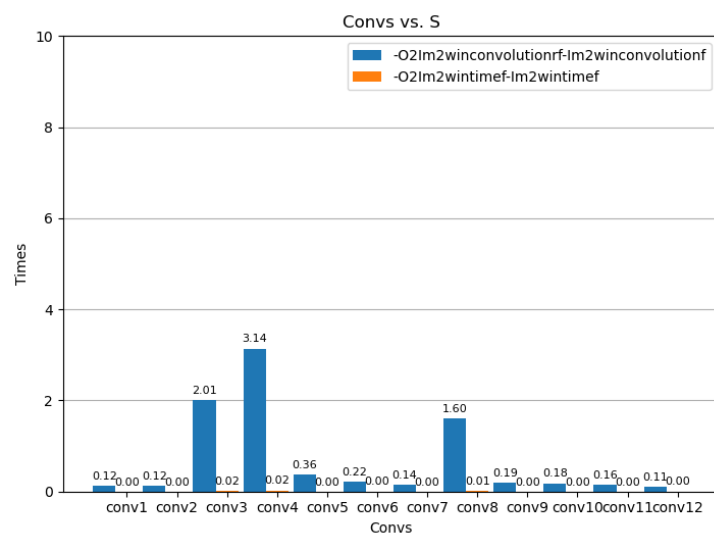


Figure 5: O2情况下的时间

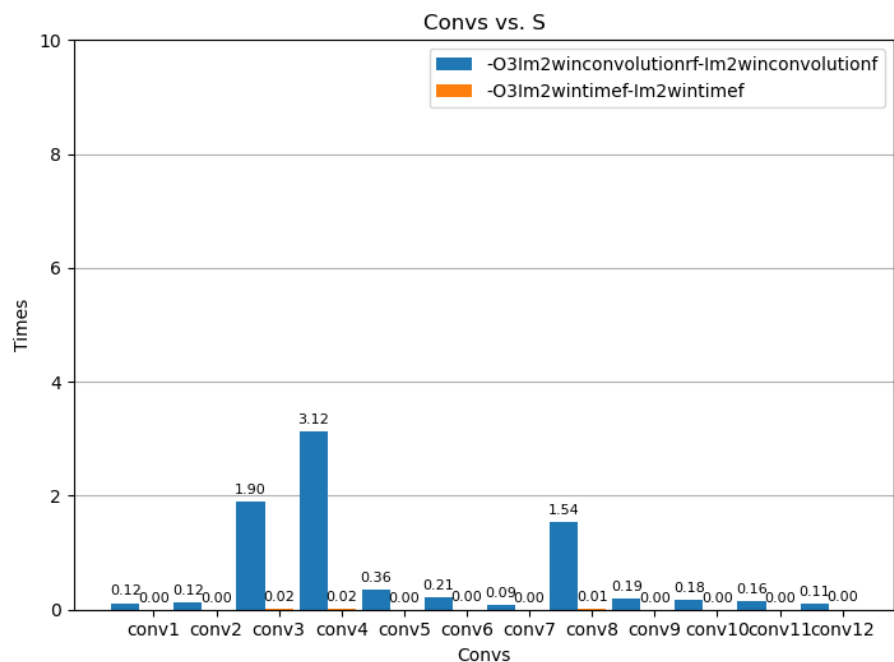


Figure 6: O3情况下的时间

3.1 Analysis

没有任何优化的im2win卷积表现和直接卷积十分接近，在O0的时候im2win卷积要快一点点，在O2O3im2win卷积要慢一点点，但是十分接近。

更换了循环顺序后，im2win变化后的张量数据连续性好一点，表现也要好一点，也十分接近，除了在O2的时候有conv4、5、7、8、9反而更换后慢一点（conv4、5、8、9十分接近，保留两位小数用的四舍五入，实际上的差距0.01左右，不敢确定是不是偶然），conv7尤为明显。WeTensor表现一直要比Tensor1D好很多，因为有更多优化。

im2win转换的时间占整个im2win卷积所用的时间很小一部分，因为只取两位小数，所以在图中甚至是0，只有很小一点点柱子。比较具体的来说，在O0优化下conv1的情况下，im2win卷积平均需要1.07055秒，im2win转换只需要0.00268702秒，只占0.25%左右的时间。

需要做优化才能体现im2win的优势。