《cuDNN: Efficient Primitives for Deep Learning》阅读笔记

zxp

April 27, 2024

1 论文内容

《cuDNN:深度学习的高效原语》这篇论文提出了大名鼎鼎的cuDNN，应该是现在GPU上卷积运算最常用的库，有成熟易用的api可以方便的嵌入各种库中，cuDNN优化了不同大小的常用的卷积，目标是让卷积操作达到矩阵乘法的性能。

1.1 论文中提到的背景

论文发表于2014年，在2014年深度神经网络就已经广泛应用，并且已经大范围在GPU上实现卷积。英伟达为了使神经网络框架社区能够平等地从其api中受益让使用者不需要手动优化卷积或者使用特定的框架，制作了cuDNN。cuDNN支持前向后向传播，支持卷积、池化和激活等深度学习需要用到的操作。
cuDNN要求输入和输出数据驻留在GPU上，理论上将卷积需要用到的所有东西都放在显存上是最快的，但显存是一种十分稀有的资源，所有cuDNN十分强调cuDNN实现卷积需要的内存很小，比起显式的im2col＋cublas来说。
论文还指出矩阵乘法的缺点，由于矩阵乘法是高度优化的，因此将卷积降低为矩阵乘法是有效的。矩阵乘法之所以快速，是因为它每字节传输的数据的浮点运算比率很高。这个比率随着阵变大而增加，这意味着矩阵乘法在小矩阵上的效率较低。即im2col在大矩阵的适合并且十分有效。显式的矩阵乘法需要令人绝望的显存，于是cuDNN使用隐式的。
使用快速傅里叶变换来计算卷积，。FFT可以显著降低卷积的工作复杂度，通过巧妙的工程设计，可以有效地用于深度神经网络。然而，基于FFT的方法使用了大量的临时内存，因为过滤器必须填充到与输入相同的大小。滤波器与图像相比较小时，这尤其昂贵，这种情况还特别常见，通常发生在卷积网络的前几层。
直接计算卷积。这可能非常高效但需要大量专门的实现来处理卷积的11维参数空间中隐含的许多边角情况。（我们的im2win的也会遇到这种情况）需要专门的优化，依赖卷积和硬件的参数（比如批次，依赖批次是比较令人难以忍受

的），不好移植。论文提到了一个优化直接卷积的方法（引用了对应论文），批次大的时候好，小的时候差。

1.2 论文考虑的方式

隐式矩阵乘法，将卷积简化为矩阵乘法，利用优化矩阵乘法的方式优化，计算额外的索引需要通过启动时间常数除数重复计算整数除法和求模操作。利用[21]中提出的整数除法和取模算法，将这些代价高昂的操作转化为整数乘法和移位，从而减少了所需的索引开销。

2 心得

早期的cuDNN，对各种方法做出了优缺点介绍，指出GPU的特性，内存十分稀有，矩阵乘法快，将卷积简化为矩阵乘法。