4、Caffe的特性与移植

Caffe是一个开源的深度学习框架，主体由贾扬清在伯克利分校读PhD期间完成，目前在开源社区GitHub上由二百多名开发者维护。

本文选择Caffe进行移植并优化的一个原因是其灵活性。Caffe通过Prototxt协议来定义神经网络结构，只需写好配置文件就能实现不同的卷积神经网络。其主体代码由C++写成，也有封装好的Python和MATLAB接口，底层高层接口均比较完备，不仅适合灵活开发比较常规的网络，也可以很方便地加入自定义的操作，开发原创的CNN模型。在Caffe推出后一年内，就有超过1000个fork分支出去的自定义Caffe版本【来源请求】。

本文选择Caffe进行移植的另外一个原因是其性能。在Caffe刚推出时，它在NVIDIA K40 GPU平台上就可以达到每天6000万张图片的处理速度（每张图片前向传播需要1毫秒，训练迭代平均需要4毫秒），其在CPU上的速度同样惊人，而且优化潜力巨大。Intel曾针对Caffe推出了适用于Intel x86架构的CPU版Caffe，相比原版Caffe在纯CPU模式下可以达到13.5倍的速度提升【来源请求】。

这主要得益于其底层运算操作的设计。在卷积神经网络的特殊语境下，卷积的计算过程伪代码逻辑上是一个六重循环（逻辑上六层循环指标w, h, x, y, m, d分别为：输入特征图长，输入特征图宽，卷积核长，卷积核宽，输出频道数，输入频道数）【贾扬清老师的memo https://github.com/Yangqing/caffe/wiki/Convolution-in-Caffe:-a-memo】

for w in 1..W

for h in 1..H

for x in 1..K

for y in 1..K

for m in 1..M

for d in 1..D

output(w, h, m) += input(w+x, h+y, d) \* filter(m, x, y, d)

end

end

end

end

end

end

如果真的按六重循环嵌套的方式实现，优化将根本无从谈起，因为嵌套循环实在太深，无法对任意的输入维度保证良好的性能。而且，由于滤波器需要在输入特征图上滑动，这种实现的空间局部性很差，无法有效利用缓存。

Caffe将难以优化的卷积操作简化为已经高度优化的问题——矩阵相乘。自然科学计算和计算机科学中的很多问题都可以用线性代数表达，基础线性代数库（Basic Linear Algebra Subprograms, BLAS）已经被高度优化，具有良好性能。如果将输入特征图原地展开为列向量，将滤波器权重对应展开为矩阵【im2col】，则卷积层计算可以写成稀疏矩阵与列向量相乘的形式。这就是Caffe底层运算的机制。同理，全连接层也可写作矩阵相乘的形式。目前Caffe支持的BLAS有开源跨平台的ATLAS和OpenBLAS，以及Intel公司为其处理器优化过的MKL。在NVIDIA生产的GPU上，Caffe也支持NVIDIA的cuBLAS。

将滤波器展开为矩阵、输入原地展开为列向量的方式虽然能利用高度优化的BLAS库，获得良好性能，但缺点是对空间要求很大。如果在GPU上并行实现，手机芯片上GPU和CPU间的通信时间将取代计算时间成为性能瓶颈，故本文选择在CPU上使用OpenMP优化Caffe。因MKL闭源，且只能在Intel CPU上使用，在CPU上可用的BLAS只有ATLAS和OpenBLAS两个选项。本文选择使用OpenBLAS【来源请求】