2020年12月14日 2s-AGCN姿态识别图卷积算法加速进展

已有工作进展：

1. 去掉了C模块并对网络进行了重训练，得到了比预想中好得多的精度效果；
2. 针对姿态识别类图卷积算法（计算范式为）提出了一种通用的等效计算加速方法；
3. 针对（2）中的方法提出了一种专用剪枝方法。
4. C模块介绍

C模块是原网络计算输入数据自相似度的模块，其主要操作为对输入feature X进行的高维矩阵乘操作，在进行矩阵乘操作之前，还要执行多次1x1卷积运算，运算量极大，极其耗时。

基于此，我们把C模块整个去掉并对网络进行了重训练，得到了可以接受的精度：



表1 基础精度

我们把重训练后得到的woC参数模型（93.98%）作为我们后续工作的基础。

1. woC模型的基本计算流图

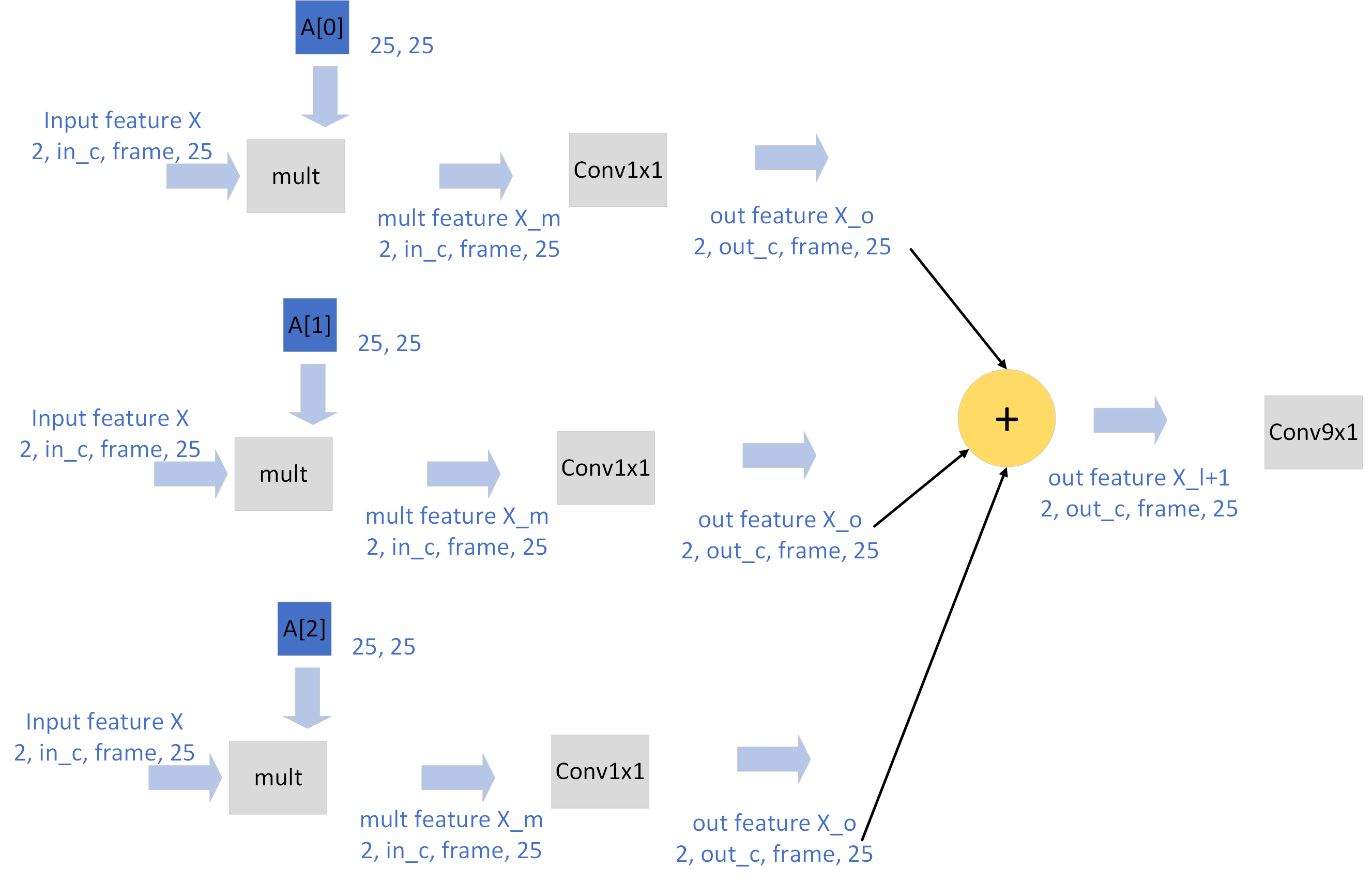


图1. woC模块基本计算流图

以第2卷积层为例：

Input X.dim = (2, 64, 300, 25) 与矩阵A需要做3\*64\*300\*2\*25\*25=72M次乘法；

与conv1x1需要做184M次乘法运算；

与conv9x1需要做 1.66G次乘法运算。

看起来conv1x1和conv9x1的运算量占主要部分，但实际上存在的问题是：

1. A矩阵并不稀疏（实际上没有任何稀疏性），如果对其进行稀疏处理会极大的影响精度，因此AX实际上是一个非常稠密的矩阵运算，优化空间很小；而conv1x1和conv9x1都可以通过剪枝进行稀疏化处理跳过大量计算；
2. 在的计算范式下，部分的稠密计算和中间结果会给整个模型带来很大的计算延迟和存储资源消耗。
3. 高维矩阵乘与conv1x1的协同优化

针对的计算范式，我们提出了一种等效计算的协同优化方法。

符号说明：

我们先只考虑的部分

令：，原式即为

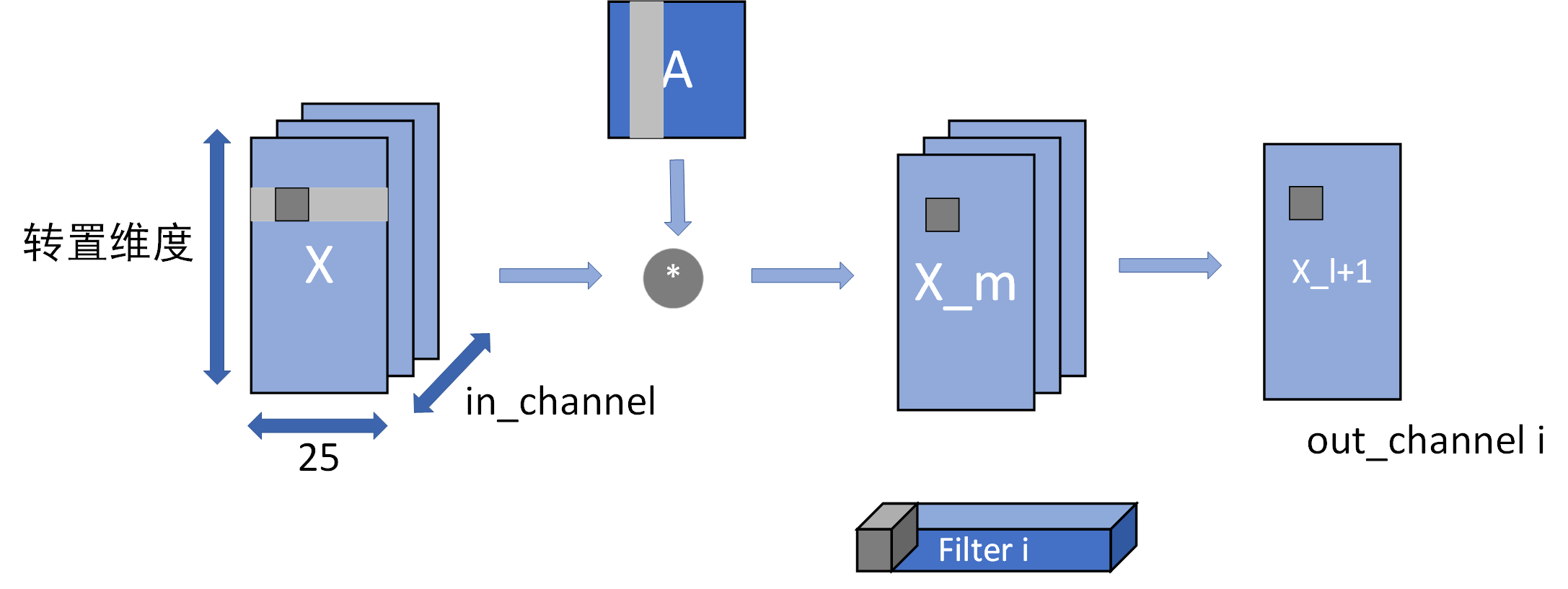


图2. 等价计算示意图

再不改变计算结果正确性的情况下（已完成数学证明），把三者的计算结合起来，只要中存在一个0，即可跳过对应的一组稠密向量乘。

图3中表示了等价计算推广到多输出channel时的情况。通过剪枝可以构造我们需要的0元素和稀疏性。该剪枝方法的实质对所有的filter的p通道都被剪枝，即抛弃input feature在某一input channel上的所有数据。

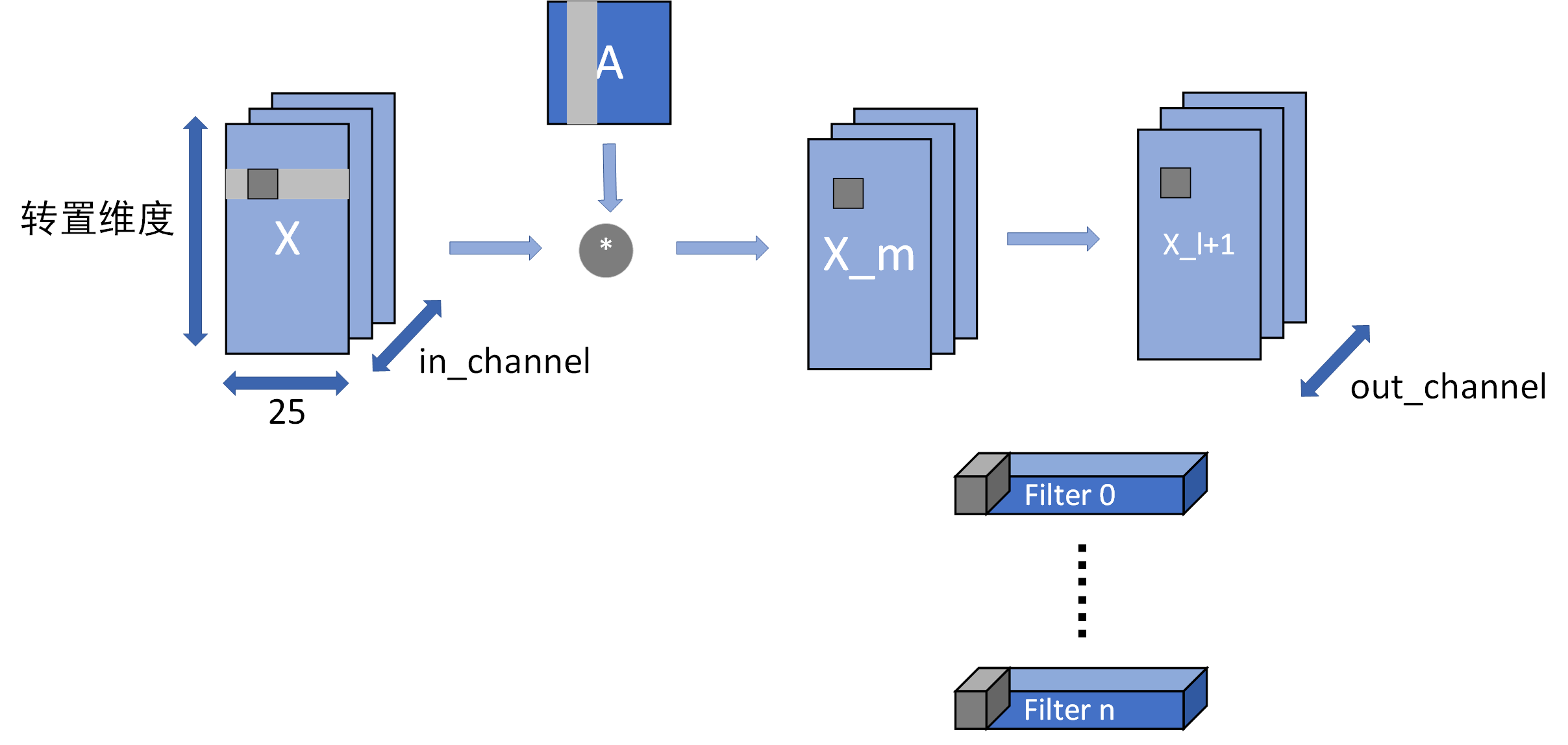


图3. 等价计算推广到多通道

1. 协同的剪枝方法

我们提出了等效跳过计算协同方法，通过对conv1x1权值的剪枝构造稀疏化来跳过高维矩阵乘的计算。该剪枝方法的实质是抛弃Input feature中某些通道的激活值。因此，我们对woC模型的feature数据按通道进进行求平均。

上述三图显示feature数据在通道上的分布极其不均匀，我们完全可以丢弃那些分布数据较小的channel，仅保留那些更有价值的channel信息。Feature的分布可以作为每层卷积权重的剪枝依据，一旦要减掉的channel确定了，稀疏格式也就随之解决。

woC模型中feature的各层稀疏度可以作为我们进行剪枝时的稀疏度参考。对于那些较稀疏的层，我们可以对应的增加权值稀疏度；对于较不稀疏的层，我们可以对应的降低权值稀疏度。

总结——困难与挑战

1. 图卷积计算范式中的高维矩阵乘是图卷积算法中的核心环节，在GPU上，高维转置矩阵乘需要消耗很大的显存和计算资源，占到总计算耗时的40%；在FPGA上，该高维矩阵乘数据稠密、计算延迟高、优化空间小。

提出了一种解决方案：等效跳过计算可以跳过很多稠密矩阵乘，但对剪枝方法提出了新的要求。

1. 第1层的权值信息价值高（权值数量相对较少而且绝对值大），不利于剪枝，因此导致第1层计算的优化空间很小，可能成为瓶颈。
2. Conv9x1的计算量较大，剪枝优化方法还有很多选择空间，目前还没具体确定。

针对第一层优化空间小和第一层conv9x1难以优化的问题，目前使用67.7%和80%随机跳过输入帧的模型已经重训练完毕，精度分别可以达到93.73%和93.0%，最理想的90%随机跳过输入帧的网络正在训练中。

1. 目前我们加速的2s-AGCN模型虽然是领域中相当classic和baseline的模型，但其性能已经不算SOTA(94% vs 96%)，可能会影响工作的说服力或是效果。然而我们提出的优化方法具有一定程度的通用性，可以在以图卷积为基础的姿态识别算法上推广。