上周主要工作进展

1. 汇编代码正在调试resnet1x1部分的程序正确性，目前把bug点定位到了DMA部分，周一与工程师交互后继续推进；
2. 周一到周五每日都去921医院做理疗，现在勉强可以持续坐15分钟不出现症状；
3. 对2S-AGCN图卷积程序进行了一些实验。实验结果和总结见后。

下周工作计划：

1. 抓紧完成汇编程序正确性调试；
2. 汇编工作按许老师安排推进；
3. 将量化或卷积应用到图卷积上跑一下；

2S-AGCN图卷积姿态识别算法工作小结

1. 算法背景介绍

2018年，港科大SiJie Yuan, et al. 在CVPR上发表Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition (ST-GCN)，成为图卷积姿态识别领域的开山之作。针对视频流中的人体姿态识别问题，学术界传统的方法是基于2D/3D计算机视觉+深度学习或LTSM+深度学习的方法，但在实际效果上一直未取得突破。ST-GCN首次提出将图卷积算法应用于姿态识别问题中，相比传统算法取得了较大的突破，并在当时达到了SOTA的水准。ST-GCN的工作证明了图卷积算法对视频流姿态识别问题的有效性和优越性。

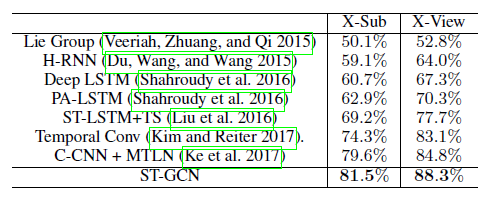


图1. ST-GCN与其他算法在NTU-RGB+D数据集上的精度对比

2019年，中科院自动化所Lei Shi, et al发表 Two-Stream Adaptive Graph Convolutional Networks for Skeleton-Based Action Recognition (2S-AGCN). 2S-AGCN算法相较ST-GCN和其他图卷积算法又在精度上取得了较大的提高，但该模型增加了网络的层数和其他结构，计算量大幅上升，计算速度也随之下降，在2080Ti上的吞吐率为28fps，存在较大的优化空间。

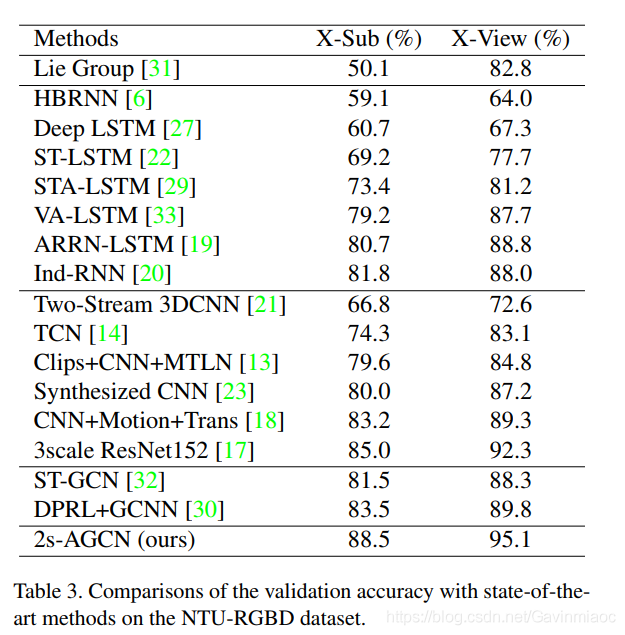


图2. 2S-AGCN与其他算法在NTU-RGB+D数据集上的精度对比

1. 姿态识别算法中的图卷积和数据输入

ST-GCN和2S-AGCN算法都采用人体骨骼图数据作为输入。Openpose(CMU)和alphapose(SJTU)等姿态识别框架在视频流中绘出人体骨骼点和骨架连接，每帧视频都会生成如图3所示的人体骨骼图，2S-AGCN等算法将该人体骨骼图进行预处理后生成实际的输入数据。在数据集NTU-RGB+D中，每个人体骨骼图中都被标注了25个关键关节点，点与点之间

通过骨骼相连，形成了连通关系进而形成了一张图。

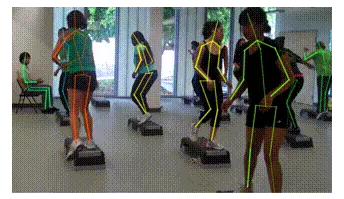


图3-1. Alphapose处理得到的视频帧中的标定人体骨架图

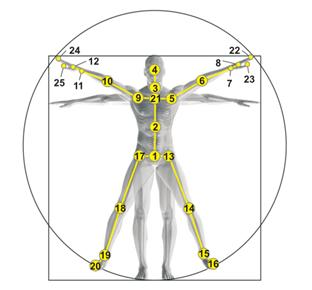


图3-2. NTU-RGB+D数据集中关节点和骨架的连接关系

Alphapose等框架不仅可以绘制视频流中的人体骨骼图，还同时给出了每个骨骼点的相对坐标。2S-AGCN算法使用这些结果进行对1个完整的视频进行数据预处理，得到一个维度为[2, 3, 300, 25]的张量，分别对应[视频中的人体数, 输入通道数, 帧数, 图节点数]，三个通道中的数据由点的相对坐标和点之间的连接关系生成。

姿态识别中的图卷积算法不含复杂的搜索点等图操作，其基本计算模式为

其中X为feature，A为图连通矩阵，W为权值。AX代表矩阵乘法，代表卷积操作。从公式上可以看出，姿态识别图卷积和传统的CNN更为接近，需要解决的新问题是如何处理好左乘的稀疏矩阵A。

轻量化的openpose和alphapose算法已经可以在普通笔记本上跑到30fps的性能，一些更激进的改进版本甚至可以跑到70fps，而后端的2S-AGCN在GPU上28fps的速度显得非常沉重，存在加速的需求。另一方面，在实际应用中也存在需求，把前端的openpose/alphapose与后端的2S-AGCN相结合，实现端到端的实时姿态检测加速系统。

1. 2S-AGCN算法的数据流图

2S-AGCN算法由10层卷积模块和一层全连接层组成。每个卷积模块内部都有相同的计算结构。以第一个卷积模块为例：

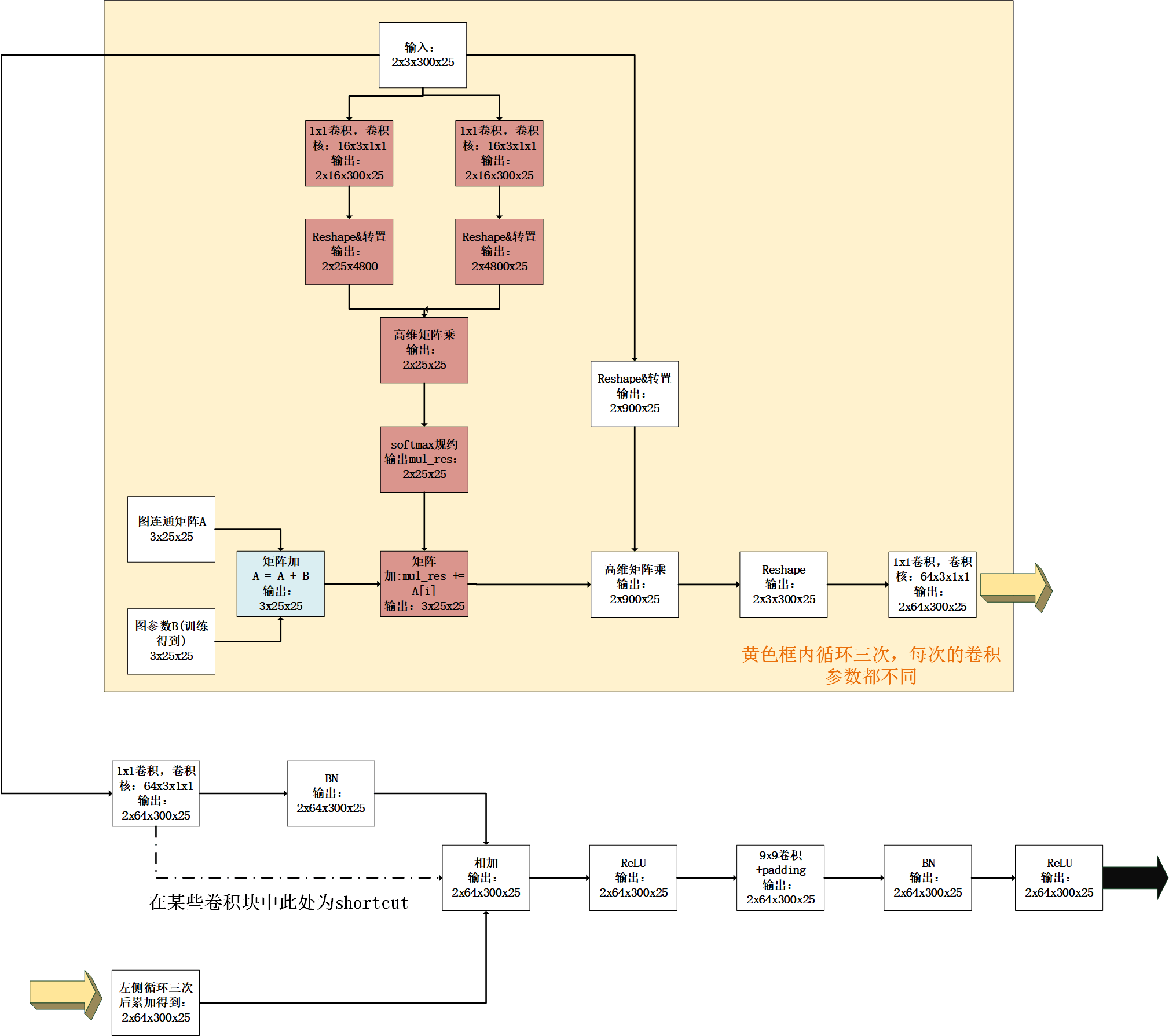


图4. 网络基本卷积块

值得注意的是，黄色框内存在多次1x1卷积，黄色框每个卷积块都会循环执行三次，且三次中每次的卷积核参数都是不同的。上述卷积块在模型中重复了10次，当然数据维度和卷积核维度都会发生变化。当input\_channel = out\_channel时，上图中的shortcut生效并跳过BN层进行相加，不然则执行BN后再与循环累加得到的数据求和。该模型每层的输出feature维度如图5所示：

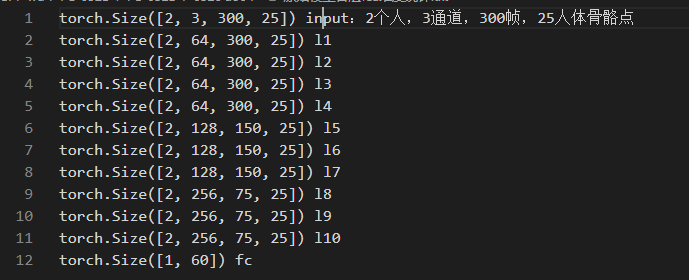


图5. 原始输入数据与每层输出feature的维度信息

图4中红色框标出的计算是对应2S-AGCN中的Ck参数模块，论文中给出的信息显示Ck能够提供0.4%的精度提高，然而在实验中，如果去掉Ck部分的计算，吞吐率将从28fps提升到50fps。Ck部分的高维矩阵乘会较明显的拖慢模型在GPU上（FPGA上）的运行速度，因此计划后续实验将把Ck模块去掉后进行重训练，损失的0.4%精度也并不是不可以接收。

1. 帧跳过实验

如前所述，一个数据样本包含300帧数据，对于较短的视频流来说300帧数据是过于冗余的，我在上周采用了多种方法进行帧跳过实验，以研究该方案对精度的影响。需要注意的是，在CPU平台上的帧跳过并不是真正意义上的帧跳过，考虑到维度保持，我只是做了简单的数据复制，而在FPGA平台上，帧跳过将转化成实在的加速比。

方法1：欧式距离相似度跳过

将高维的输入数据按照帧的维度拉成一个向量，设置阈值和初始关键帧。与关键帧相似度小于阈值的帧用关键帧的数据替代，若新帧的相似度大于关键帧，则保留新帧并将新帧设置为新的关键帧。实验结果如下：



方法2：cos相似度跳过

将高维的输入数据按照帧的维度拉成一个向量，设置阈值和初始关键帧。与关键帧相似度小于阈值的帧用关键帧的数据替代，若新帧的相似度大于关键帧，则保留新帧并将新帧设置为新的关键帧。实验结果如下：



方法3: 分组融合

将300帧数据每n帧为一组，组内的n个数据累加并求均值，得到1帧融合过的数据。计算该帧数据以该组n帧数据的计算。



方法4：随机裁剪跳过



实验得出结论，随机跳过效果是最好的，分组融合次之，cos相似度和欧式相似度效果是最差的。该实验结果有点违背直觉，可能的原因是输入高维数据包含了骨骼点的相对坐标和连接关系，两种数据混在一起，并不能很好的用几何相似度来衡量。后期有时间会补足实验论证原因。