

本周主要在训练网络, 对于网络中的一些细节部分还是考虑的不够到位导致自己训练出来的网络的精度和作者相比要差很多, 在这之中尝试了各种方案, 包括尝试各种数据增强以及训练时参数的调整但是效果都不太好。另外考察了在验证集上运行速度的问题, 因为在验证集上面需要对每一帧都进行提取 feature map, 所以存在大量重复计算, 本周尝试了多种方案企图解决这个问题, 以希望能有一定的提速效果。

先上作者训练好的参数在各个数据集上的测试结果:

Dataset	modality	Map
UCF101	Rgb	0.6410
	Flow	0.5870
	Rgb+flow	0.7019
UCFSports	Rgb	0.8249
	Flow	0.7890
	Rgb+flow	0.8764

刚开始训练自己网络的时候没有加入数据增强, 分别在两个数据集上都训练了一次, 但是结果并不十分理想, 和作者的 map 相比有不小的差距, 同时由于超参数选择不合理也导致了刚开始在 UCF101 上的训练时间很长, 因为 ucfsports 数据集比较小训练快, 所以决定之后的所有测试将先在 UCFSports 上进行验证。

Dataset	modality	map	Epoch
UCF101	Rgb	0.5602	750
		0.5710	950
		0.5725	1250
UCFSports	Rgb	0.6397	105
		0.6469	150
		0.6559	350

下面的实验在 UCFSports 数据集上进行了测试, 跑完上面的结果之后分析结果的巨大差距考虑训练过程中的问题, 开始逐步尝试各种训练技巧企图提高测试精度, 下面是部分数据记录。

方案 (基于 UCFSports)	Map
在正式训练之前加入 lr warm up, 没有加入数据增强	最高 0.6603
加入随机裁剪 (没有保持原图的比例) + 随机翻转, 在训练集上的 loss 最小只能达到 1.1	最高 0.5369
加入随机裁剪 (没有保持原图的比例) + 随机翻转, 尝试在最初的训练中 freeze 前面提取 feature map 的层, 然后单独训练后面, 再回头训练完整的, 即使这样也还是只能把 loss 降低到 1.0	在训练集上的 map 只能达到 0.7463 验证集上只有 0.5039
分析没有加入数据增强为什么精度会比加入了数据增强的精度高, 观察了随机裁剪之后的图像部分比例确实有点失调, 之后调整	最高 map 可以达到 0.6618

随机裁剪的比例，在对原图进行裁剪时强行和原图的比例保持一致。这样就和刚开始训练的时候直接 resize 的比例相同，但是数据集上 ground truth 的位置和大小发生了改变。	
增加对图片的 normalize 和光度变换，对 rgb 三个通道不再采用作者代码中的减去三个参数，而是选择采用 normalize RGB 三个通道，即 sub (0.485, 0.456, 0.406) and div (0.229, 0.224, 0.225)	训练的速度明显加快，但是只能达到 0.3544

上面的各个方案的测试结果都并不理想，很多时候一些细节没有考虑到就直接开始跑程序，跑完了发现结果不对才想起来有些地方其实不是很对，浪费了一定的时间但是又没有办法避免。其中的数据增强的代码最开始全部自己根据需要进行编写，其中多次反复验证程序的正确性，往往转了一大圈回来发现自己新加入的程序没有问题或者是一些很弱智的问题花费了很多时间。不过对于随机 crop 这一块我倒是突然觉得在 crop 的时候尽量保持原图的比例应该对提高精度有一定的作用的，因为人的比例基本上是一个固定的，所以如果在 resize 过程中变形过大确实也挺为难网络。不过不太大的适当的形变也应该有，不然网络的泛化能力差。

下面对验证过程中速度太慢尝试了多种方案并做了实验，这其中提速最麻烦的一个原因是网络在提取 feature map 的时候在 conv6 这一层对于一段 6 帧采用了 dilation，这也就是说每一次提取 feature，其实都是相对独立的，因为对同一帧 6 次提取 feature 都应用了不同的 dilation 系数。但是好在刚开始的时候的流程图提供了部分帮助，因为 dilation 的系数每次都是 6 的指数次幂，所以在一段的第二帧提取 feature map 的时候这个膨胀系数为 36 的 3*3 卷积核就退化成了 1*1，这样考虑的话就只需要对每一帧提取两次 feature，之后进行适当的组合可以减少很多计算，基于此找到了一种速度勉强还能接受的方案：

方案	Modality	Fps
每次选取一段，也就是 6 帧，但是会使用多个这样的段组成 batch 送入计算，这样会有大量重复计算	RGB	9.9
一次性提取一个视频所有 feature map，包括 dilation 为 6 和 36，但是这样 feature map 在 gpu 中容易炸显存	RGB（提取出的 feature map 放在 gpu 中）	27.3
	RGB（提取出的 feature map 放在 cpu 中）	16.7
	RGB+FLOW（提取出的 feature map 放在 gpu 中）	24.3
	RGB+FLOW（提取出的 feature map 放在 cpu 中）	7.3
每次只处理 6 帧，batch 为 1，重复帧的 feature map 不重复计算	RGB	23.0
	RGB+FLOW	11.7

