### Learning Efficient Convolutional Networks through Network Slimming阅读笔记

首先，通过阅读这篇论文，它的网络瘦身步骤是：训练（L1正则化稀疏）-> 枝剪 -> 微调（也就是拿枝剪后的模型再次训练）。

对于训练这个步骤，L1正则化稀疏是最关键的部分。BN层前的卷积层，有n个特征图，每一个特征图对应一条通道，为每一条通道引入一个缩放因子。本方法直接采用BN层的缩放因子γ作为对应通道的缩放因子，这n个特征图就对应了BN层的n个γ。这样子不会带来额外的负担。在施加L1正则化后，会把γ推向0。我们通常会设置一个全局百分位数阈值，比如70%，将全局缩放因子γ从大到小进行排序。如果有些γ小于这个位于百分之三十位置的γ值，就说明这个γ太小，意味着在整个网络它的贡献微乎其微，那么就是可以丢弃的。所以，L1正则化就是用来筛选出这些没有作用的γ和对应的通道（特征图）。

枝剪就是将L1正则化筛选出的“无用”通道进行移除。而在剪枝后，会得到一个远小于原来的模型，虽然会带来短暂的精度损失，只要将枝剪后模型进行微调（再次训练），精度就会上升，甚至比剪枝前的模型更高。

论文中还提出从单遍学习方案到多遍学习方案，以实现更好的压缩率。即在经过枝剪、微调后的模型再次进行枝剪、微调。在论文的4.5部分，进行了多遍学习方案的实验，对于CIFAR-10数据集，错误率随着迭代次数增加而降低，在第5次（最后一次）达到了最低，且参数和FLOPs明显减少；对于CAFAR-100数据集，第3次迭代后，错误率反而上升。由此可见过度枝剪会导致性能下降，而适度枝剪则能让性能达到最佳，而且也能极大地压缩模型。

本方法同时实现了减小模型大小、减小运行内存、减小计算操作，而论文中提及的其他方法无法同时实现以上优势，且本方法更简单，且生成的模型无需特殊库/硬件就可以进行高效推理。