模型轻量化

--知识蒸馏与模型剪枝

模型轻量化

动机

工业应用中，既要求模型有好的精度，又要求模型的开支（算力要求）小。需满足：

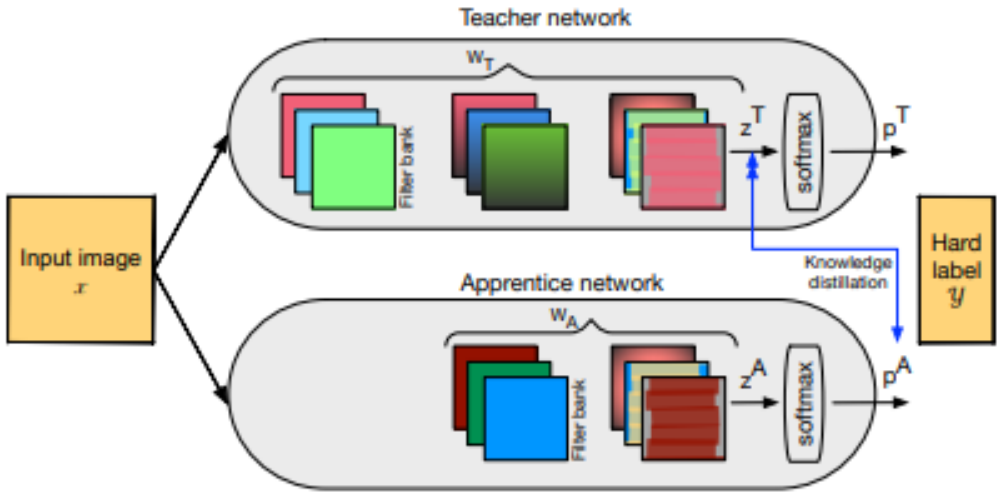
- 减小模型大小
- 减小运行时内存占用
- 在不影响精度的同时，降低计算操作数

两大方法

- 知识蒸馏
- 模型剪枝

知识蒸馏

- 有两个模型，小模型理解为学生模型，大模型或ensemble模型理解为老师模型，希望通过老师模型来指导学生模型学习，让学生模型的分布匹配老师模型，获得老师模型的泛化能力



经过训练后的原模型，其softmax分布包含有一定的知识——真实标签只能告诉我们，某个图像样本是一辆宝马，不是一辆垃圾车，也不是一颗萝卜；而经过训练的softmax可能会告诉我们，它最可能是一辆宝马，不大可能是一辆垃圾车，但绝不可能是一颗萝卜

知识蒸馏

为什么叫知识蒸馏?

$$q_i = \frac{\exp(z_i)}{\sum_j \exp(z_i)} \Longrightarrow q_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_i/T)}$$

T是温度，当温度T趋向于无穷时，softmax的输出更软。在训练新模型的时候，较高的温度使得softmax的分布足够软，这样学生模型的softmax输出近似于老师模型，从而将老师模型的知识提取出来，因此将其称之为蒸馏

待拟合分布

真实分布

```
KD_loss = nn.KLDivLoss()(F.log_softmax(outputs/T, dim=1), F.softmax(teacher_outputs/T, dim=1)) * (alpha * T * T) + \
          F.cross_entropy(outputs, labels) * (1. - alpha)
```

知识蒸馏

实现步骤

1. 训练一个容量较大的或者ensemble模型为老师模型
2. 搭建一个容量小的模型作为学生模型
3. 在训练阶段，老师模型对输入的数据推理产生teacher_output，学生模型对输入的数据推理产生output，由teacher_output和targets共同监督指导对学生模型的训练

知识蒸馏

实验结果

cifar数据集

网络结构	Test Acc
Baseline ResNet-18	94.175%
+ KD WideResNet-28-10	94.333%
+ KD PreResNet-110	94.531%
+ KD DenseNet-100	94.729%
+ KD ResNext-29-8	**94.788%**

烟火数据集

网络结构	Test Acc
Baseline mbv3	95.22%
+ KD resnext50_32x4d	96.37%

实验结果表明，采用知识蒸馏的方式，能够实现小模型精度的提升，从而达到轻量级部署的要求

知识蒸馏

注意事项

1. teacher和student要合适。不合适的student可能训练过程不稳定，并且有可能loss比teacher还高。
2. lr_scheduler要合适。尝试过其他的lr_scheduler，可能会导致实验结论发生改变，或者有些kd算法失效。
3. kd loss的参数要合适。kd算法对于超参敏感，不同的超参设置可能导致截然不同的结果。

模型剪枝

相关剪枝技术

低秩分解：采用低秩近似技术，在全连接层的表现比较好，模型大小压缩3倍，但加速不明显，因为CNN的计算量主要是来自卷积层

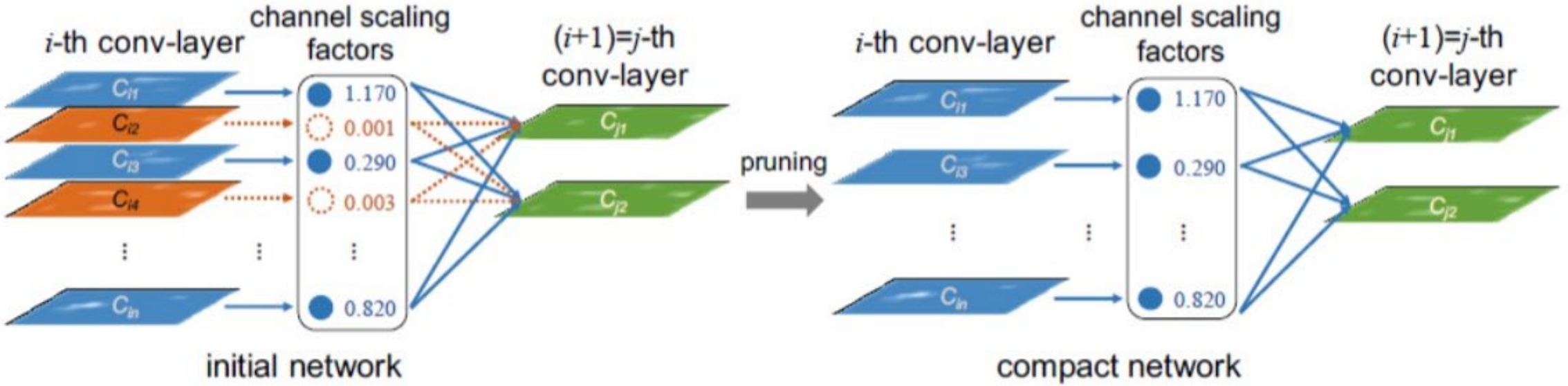
权重量化：HashNet提出了量化网络权重，只有共享的权重值和哈希索引需要被存储，因此可以节省大量的存储空间。然而，这些技术既不能节省运行时的内存，又不能节省推理时间，因为在推理期间，共享的权重需要被恢复到原来的位置（相当于还多了个解码的过程）。

权重剪枝/稀疏化：韩松之前的“Learning both weights and connections for efficient neural network”这篇工作中，提出了在训练网络中剪枝掉不重要的连接关系，这样的话网络中的权重大多数变成了0，可以使用一种稀疏的模式存储模型。然而，这些方法需要专门的稀疏矩阵运算库或硬件来做加速，运行时的内存占用节省非常有限，因为产生的激活值仍然是密集的

上述技术存在一定缺陷，并不能很好的实现轻量化效果

模型剪枝

BN层剪枝技术



将L1正则化施加到BN层的缩放因子上，L1正则化推动BN层的缩放因子趋向于零，这使得我们能够鉴别出不重要的通道或者神经元，因为每一个缩放因子都和一个特定的CNN卷积通道（或者全连接层的一个神经元）相关联。这有助于后续的通道剪枝，另外正则化也很少损伤性能，甚至一些情况下它会导致更高的泛化准确率，剪掉不重要的通道有时候虽然会暂时降低性能，但是通过之后对剪枝网络的微调可以对精度补偿

模型剪枝

BN层剪枝技术

实现逻辑

1. 训练阶段，对bn层加入惩罚项，约束bn层的学习
2. 通过bn层权值的大小，对bn层剪枝，保留权重高的通道，删除权重低的通道以及对应的卷积层，保留bn层、卷积层、池化层以及全连接层的权重。
3. 对裁剪后的模型fine-tune

模型剪枝

实验结果

Cifar10-vgg	Baseline	Prune(70%)	Fine-tune-160
Top1 acc	93.30%	32.54%	93.78%
Parameters	20.04M	2.25M	2.25M

上述数据可以看出，裁剪后模型精度没有下降，反而略有提升，浮点运算量降低了88.8%