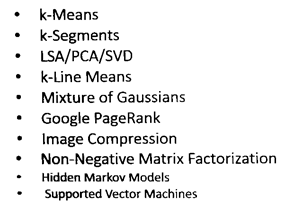
Coreset解决的问题：

压缩数据集，从D到C，使得算法的结果一致。（数据及压缩）

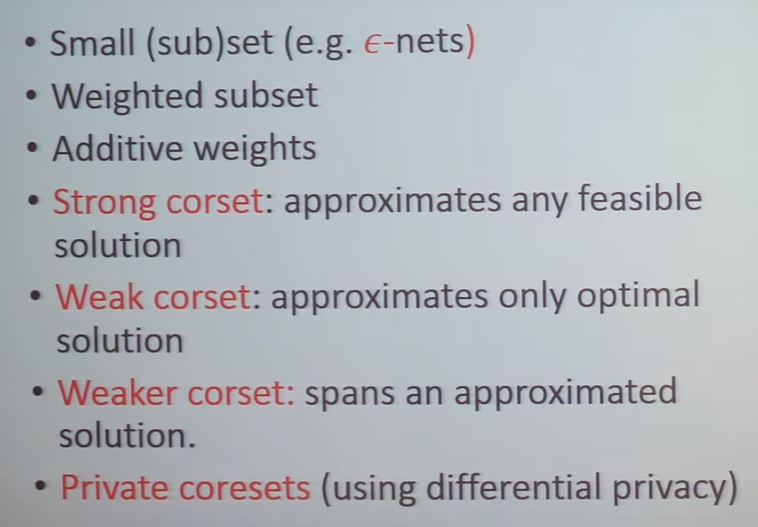
Coreset例子：

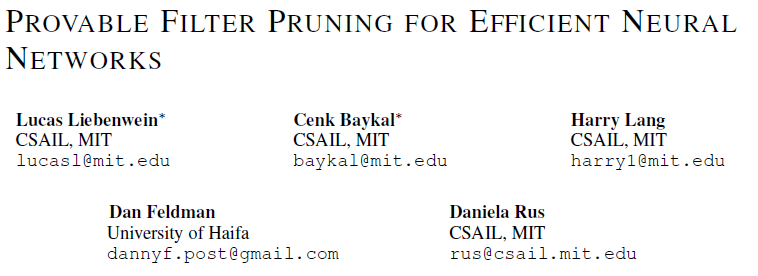


大数据的特征：

Streaming、parallel、simple computation

Coreset分类



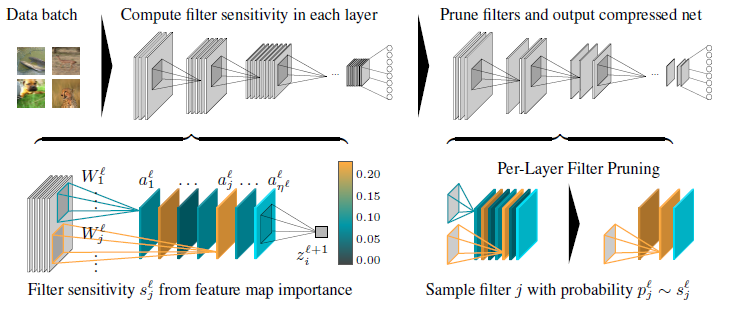


Liebenwein L , Baykal C , Lang H , et al. Provable Filter Pruning for Efficient Neural Networks. ICLR 2020

BG: Weight-based/neuron/filter pruning

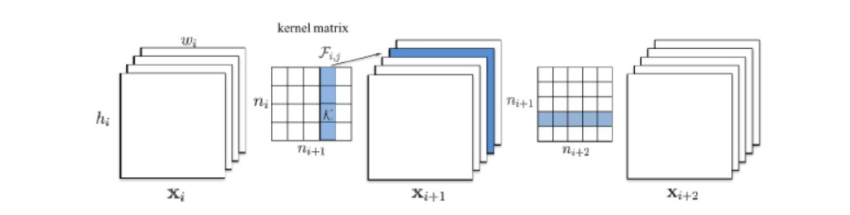
本文目标：基于采样的方法，从过参数化的神经网络中，识别和移除多余的filters。

过程：



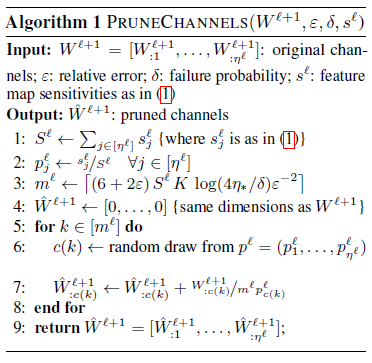
（基本概念：表示第l层神经网络的第j个filter-四元组，a是激活后，z是激活前，是第l层第j个filter生成的特征图，是整体，神经网络共L层）

关于Kernel matrix四元组：



看蓝色部分，放入xi中，生成1个结果层，共ni+1个卷积，所以xi+1是ni+1层。（本质上还是普通卷积，但是表示法不同）

裁剪掉蓝色的卷积核，会同时裁掉后面的蓝色卷积行。所以运算次数会减少两个部分的乘积运算。

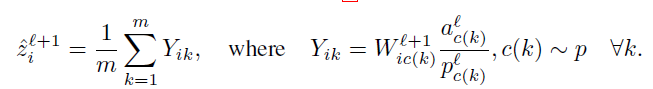


算法目的：

根据敏感性挑选，算出概率。随机采样其中的k个，然后更新权重。

Q：s的计算，W的更新（第7行），m的意义

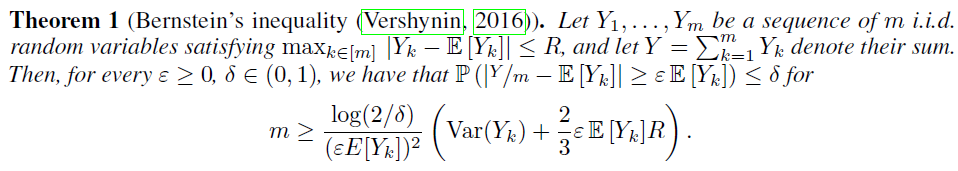
A1：W的更新



可以看到根据期望的线性性，可以得到：



采样复杂度：



Uniform Distribution：

特征：



代入公式：



A2：s的计算

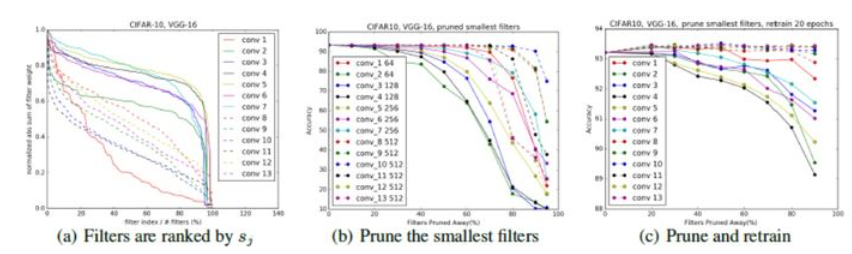
CNN剪枝的参考方法：

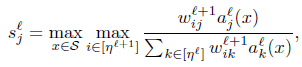
通过计算重要性。

图：CIFAR-10数据集的VGG-16网络中每一个卷积层中每个卷积核权重绝对值之和的分布

结论1：剪掉值最小的卷积核比随机（或剪最大）的效果要好。

结论2:L1比其他的剪枝（比如基于激活值的feature map剪枝）效果好。





计算在l层第j个filter过滤后，到l+1层时运算后得到的最大的卷积运算结果（的概率）。

取样本S中的最大概率。（也就是说，敏感度是指最大卷积运算结果）

注意到ES利用了固有随机性，进行采样。相关分析如下：

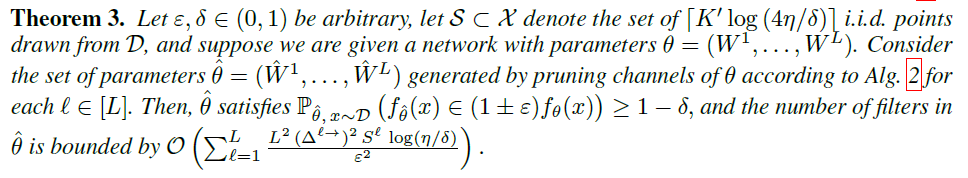


采样的复杂度：

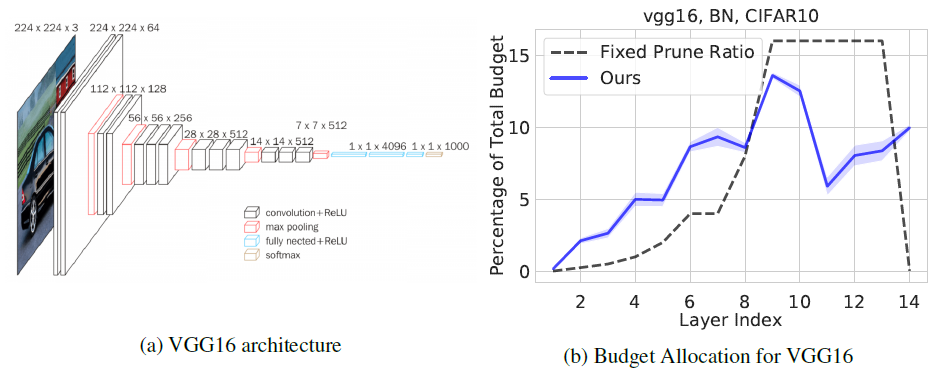


ES不需要先验数据分布，从验证集中随机获取样本集S，跑一遍神经网络即可。

Filter的个数上界：



Q3：实验-m的意义



在前面的部分中，我们建立了我们的过滤器修剪方案对于任何指定和的采样复杂度。然而，在实践中，实践者更常见的做法是指定所需的修剪比率，该比率指定修剪模型的结果大小。在这种抽样预算的基础上，如何优化网络各层的抽样预算，以最大限度地减少修剪模型的误差，是一个实际问题。

一种简单的方法是统一分配采样预算N，以便在每个层中保持相同的滤波器比率。然而，该分配方案隐式地假定网络的每一层对于保持输出具有同等重要性，实际上，在实践中，这一情况从未如此，如图2（a）所示。

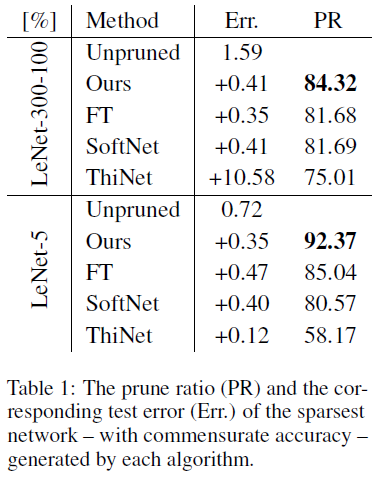
结果表明，在Alg中，我们对每层采样复杂度的分析界是逐层的。根据m的值（采样复杂度）消去不重要的层。

结论：采样复杂度和该层的重要性一样的，因为压缩率和每层重要性是一样的（越重要的，越不容易压缩，采样复杂度反映的是采样的难度，即和压缩率是相关的）（采样复杂度在Theorm1中的描述，根据不等式得到的）

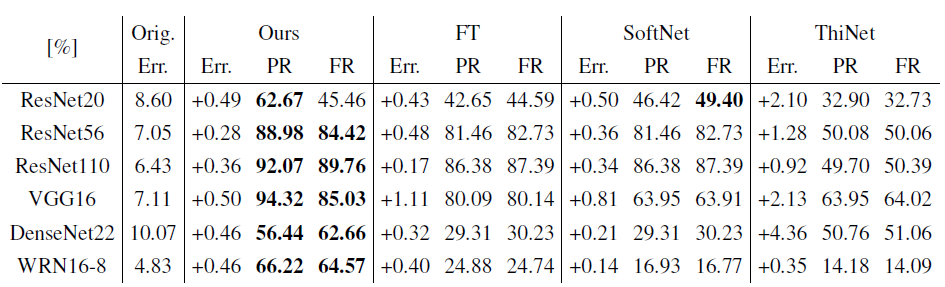
根据N找：（是用户定义的容错区间）

对于采样数量N可以采用以下方法。计算采样复杂度和采样后的神经网络的大小。如果大小=采样数，结束。否则，一直搜索合适的直到=N。

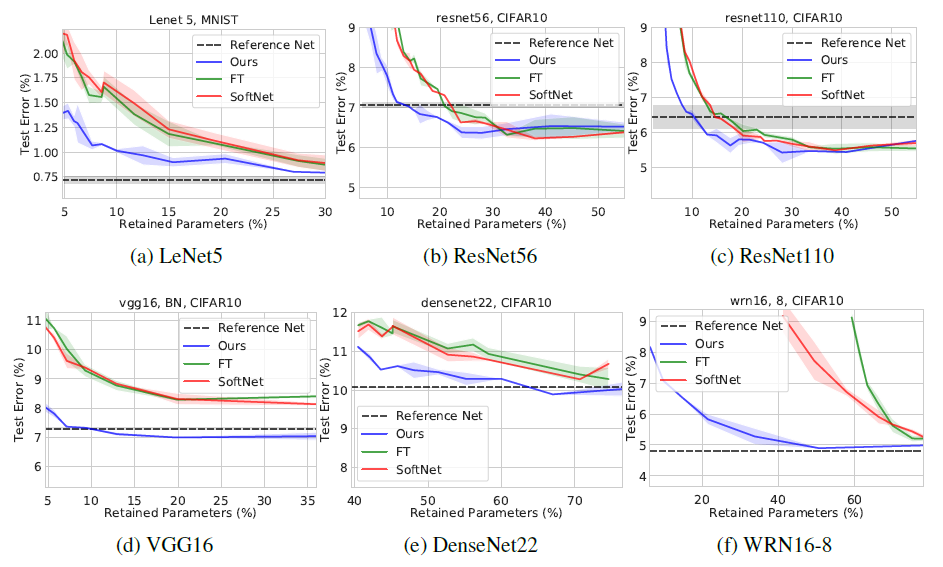
LeNet实验压缩比：



其他网络压缩比：



准确度：



Weight-based/neuron/filter pruning

