【AI不惑境】模型剪枝技术原理及其发展现状和展望

言有三

公众号《有三AI》号主，书籍作者，AI/摄影/羽毛球/电影

已关注

大家好，这是专栏《AI不惑境》的第九篇文章，讲述模型剪枝相关的内容。

进入到不惑境界，就是向高手迈进的开始了，在这个境界需要自己独立思考。如果说学习是一个从模仿，到追随，到创造的过程，那么到这个阶段，应该跃过了模仿和追随的阶段，进入了创造的阶段。从这个境界开始，讲述的问题可能不再有答案，更多的是激发大家一起来思考。

作者&编辑 | 言有三

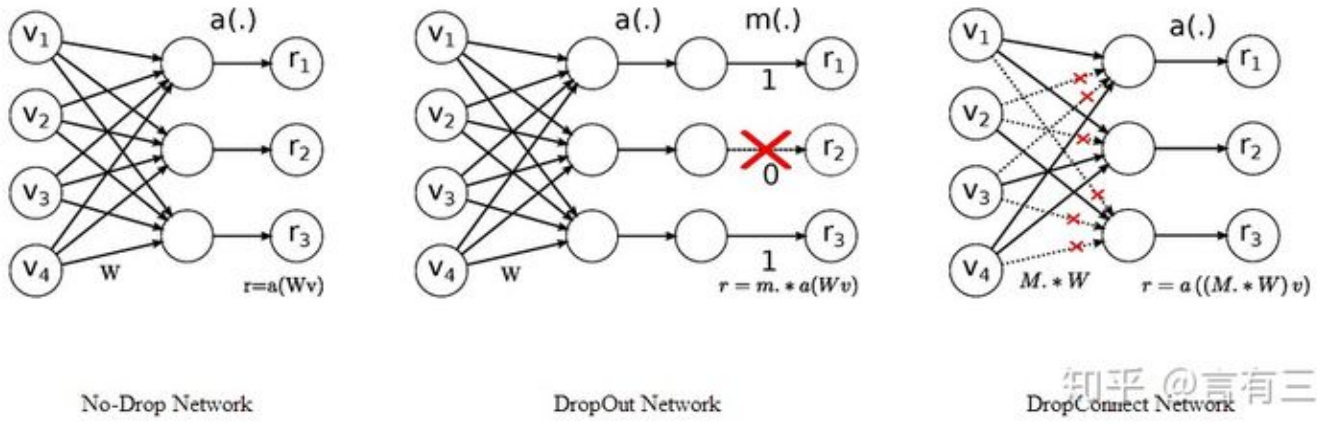
模型剪枝作为一项历史悠久的模型压缩技术，当前已经有了比较大的进步和发展，本文给大家梳理模型剪枝的核心技术，发展现状，未来展望以及学习资源推荐。

1 模型剪枝基础

1.1 什么是模型剪枝

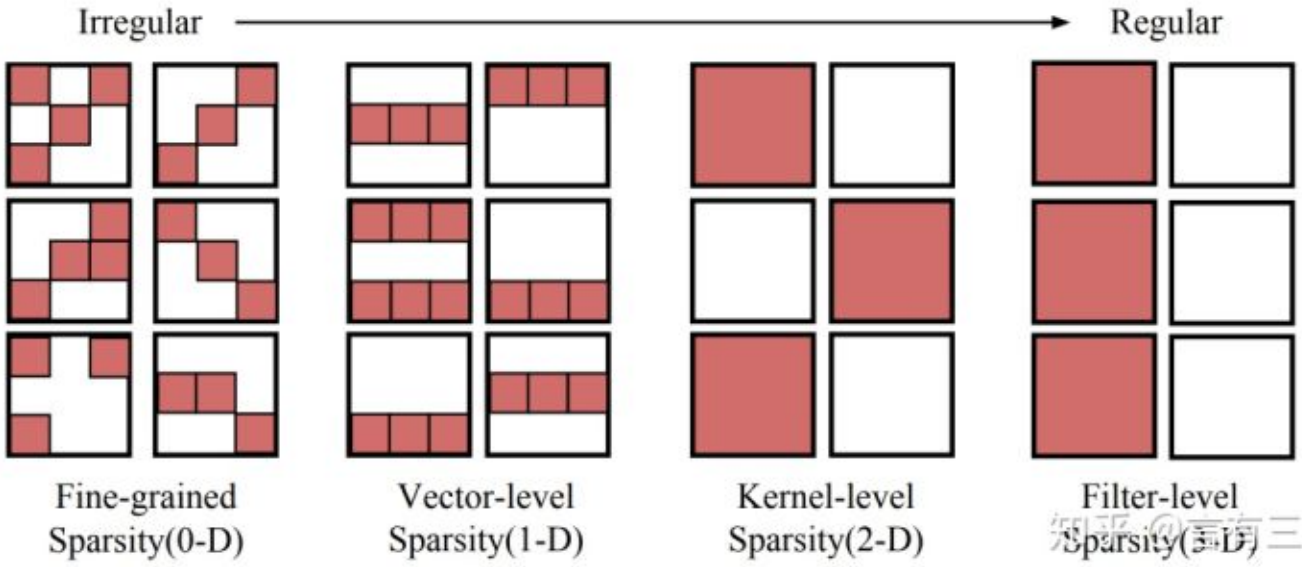
深度学习网络模型从卷积层到全连接层存在着大量冗余的参数，大量神经元激活值趋近于0，将这些神经元去除后可以表现出同样的模型表达能力，这种情况被称为过参数化，而对应的技术则被称为模型剪枝。

模型剪枝是一个新概念吗？并不是，其实我们从学习深度学习的第一天起就接触过，Dropout和DropConnect代表着非常经典的模型剪枝技术，看下图。



Dropout中随机的将一些神经元的输出置零，这就是神经元剪枝。DropConnect则随机的将一些神经元之间的连接置零，使得权重连接矩阵变得稀疏，这便是权重连接剪枝。它们就是最细粒度的剪枝技术，只是这个操作仅仅发生在训练中，对最终的模型不产生影响，因此没有被称为模型剪枝技术。

当然，模型剪枝不仅仅只有对神经元的剪枝和对权重连接的剪枝，根据粒度的不同，至少可以粗分为4个粒度。



细粒度剪枝(fine-grained)：即对连接或者神经元进行剪枝，它是粒度最小的剪枝。

向量剪枝(vector-level)：它相对于细粒度剪枝粒度更大，属于对卷积核内部(intra-kernel)的剪枝。

核剪枝(kernel-level)：即去除某个卷积核，它将丢弃对输入通道中对应计算通道的响应。

滤波器剪枝(Filter-level)：对整个卷积核组进行剪枝，会造成推理过程中输出特征通道数的改变。

细粒度剪枝(fine-grained)，向量剪枝(vector-level)，核剪枝(kernel-level)方法在参数量与模型性能之间取得了一定的平衡，但是网络的拓扑结构本身发生了变化，需要专门的算法设计来支持这种稀疏的运算，被称之为非结构化剪枝。

而滤波器剪枝(Filter-level)只改变了网络中的滤波器组和特征通道数目，所获得的模型不需要专门的算法设计就能够运行，被称为结构化剪枝。除此之外还有对整个网络层的剪枝，它可以被看作是滤波器剪枝(Filter-level)的变种，即所有的滤波器都丢弃。

1.2 模型剪枝的必要性

既然冗余性是存在的，那么剪枝自然有它的必要性，下面以Google的研究来说明这个问题。

Google在《To prune, or not to prune: exploring the efficacy of pruning for model compression》[1]中探讨了具有同等参数量的稀疏大模型和稠密小模型的性能对比，在图像和语音任务上表明稀疏大模型普遍有更好的性能。

它们对Inception V3模型进行了实验，在参数的稀疏性分别为0%，50%，75%，87.5%时，模型中非零参数分别是原始模型的1，0.5，0.25，0.128倍进行了实验。实验结果表明在稀疏性为50%时，Inception V3模型的性能几乎不变。稀疏性为87.5%时，在ImageNet上的分类指标下降为2%。

Table 1: Model size and accuracy tradeoff for sparse-InceptionV3

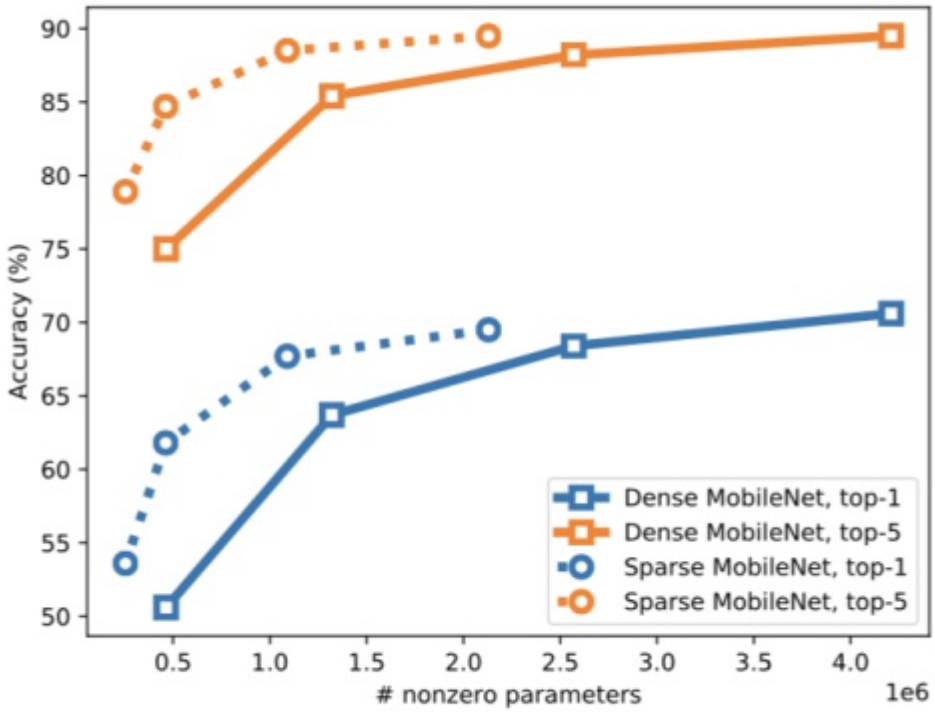
Sparsity	NNZ params	Top-1 acc.	Top-5 acc.
0%	27.1M	78.1%	94.3%
50%	13.6M	78.0%	94.2%
75%	6.8M	76.1%	93.2%
87.5%	3.3M	74.6%	92.5%

除了在大模型上的实验结果，还对小模型MobileNet也进行了实验，分别在同样大小参数量的情况下，比较了更窄的MobileNet和更加稀疏的MobileNet的分类指标，发现稀疏的MobileNet模型性能明显优于非稀疏的MobileNet模型。

Table 2: MobileNets sparse vs dense results

Width	Sparsity	NNZ params	Top-1 acc.	Top-5 acc.
0.25	0%	0.46M	50.6%	75.0%
0.5	0%	1.32M	63.7%	85.4%
0.75	0%	2.57M	68.4%	88.2%
1.0	0%	4.21M	70.6%	89.5%
	50%	2.13M	69.5%	89.5%
	75%	1.09M	67.7%	88.5%
	90%	0.46M	61.8%	84.7%
	95%	0.25M	53.6%	78.9%

具体来说，稀疏率为75%的模型比宽度为原始MobileNet的0.5倍的模型在ImageNet分类任务的top-1指标上高出了4%，而且模型的体积更小。稀疏率为90%的模型比宽度为原始MobileNet的0.25倍的模型在ImageNet分类任务的top-1指标上高出了10%，而两者的模型大小相当。



(a)

因此，我们完全可以相信，模型剪枝是有效的而且是必要的，剩下的问题就是怎么去找到冗余的参数进行剪枝。

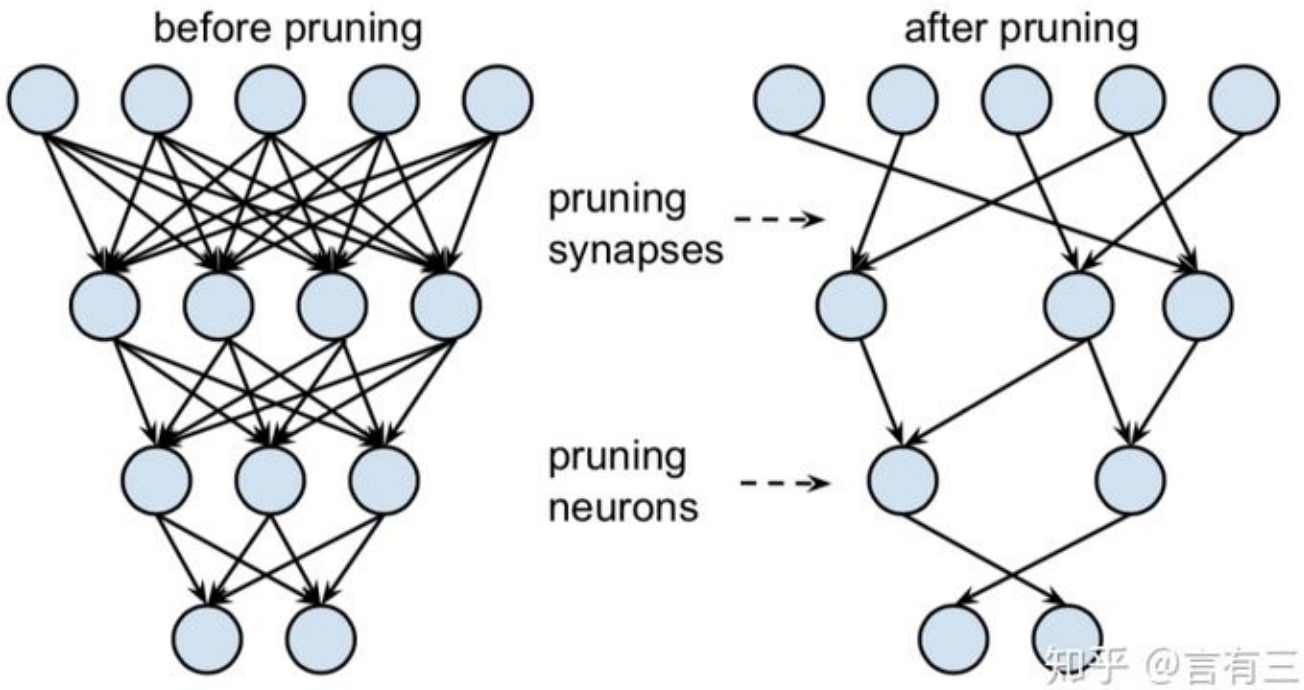
2 模型剪枝核心算法

模型剪枝算法根据剪枝的处理策略来说，可以分为对模型进行稀疏约束然后进行训练后的剪枝，在模型的训练过程中进行剪枝，以及在模型训练之前就进行剪枝。而根据粒度的不同，流行的剪枝算法是细粒度的权重连接剪枝和粗粒度的通道/滤波器剪枝。

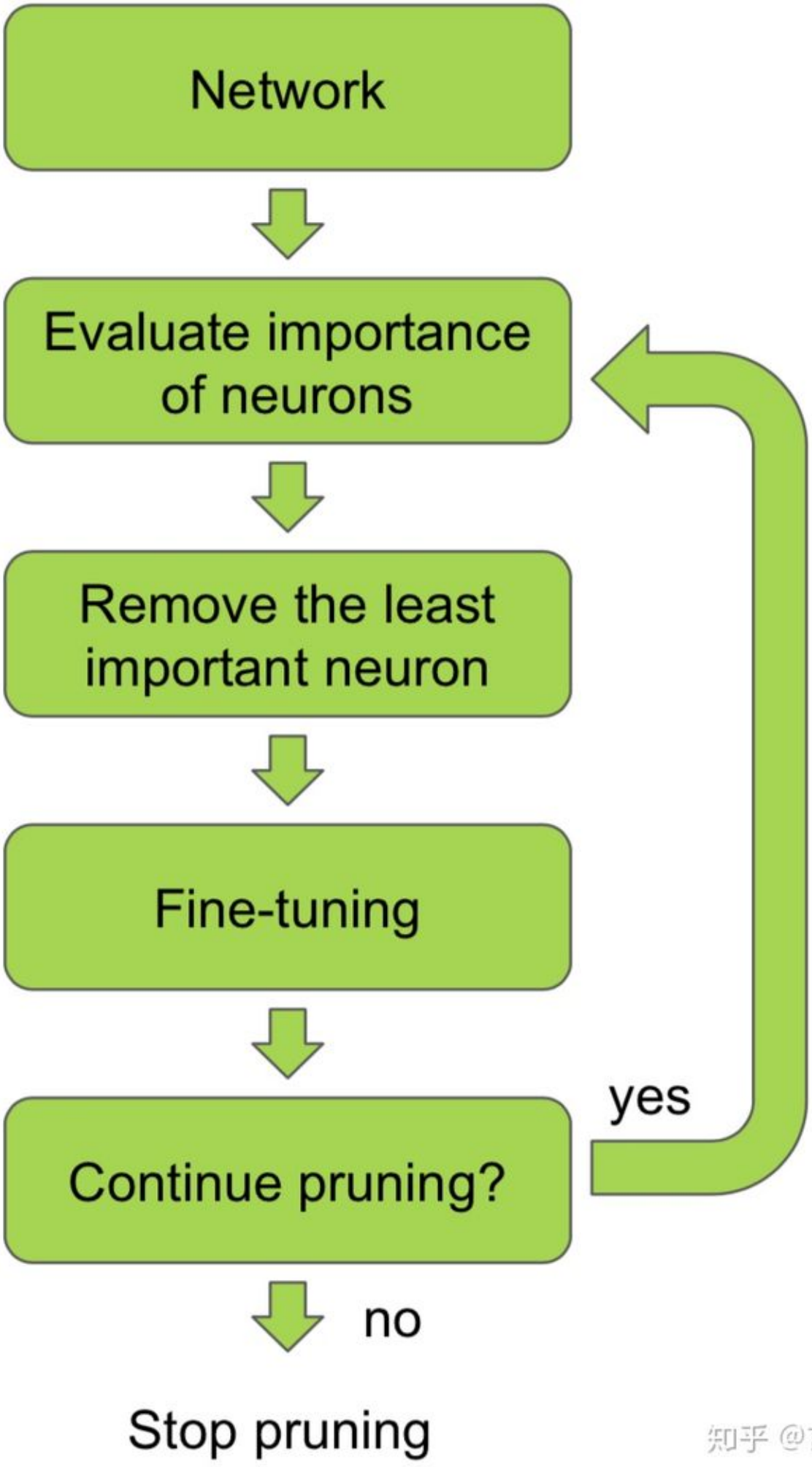
这些方法各自有交叉，无法完全分开，下面我们就基于两大不同的粒度来介绍一些训练中剪枝的代表性方法，而不再单独介绍稀疏约束以及训练前剪枝方法，相关内容感兴趣的读者可以去有三AI知识星球中阅读。

2.1 细粒度剪枝核心技术(连接剪枝)

对权重连接和神经元进行剪枝是最简单，也是最早期的剪枝技术，下图展示的就是一个剪枝前后对比，剪枝内容包括了连接和神经元。



这一类技术的整体步骤如下：



其中重点在于两个，一个是如何评估一个连接的重要性，另一个是如何在剪枝后恢复模型的性能。

对于评估连接的重要性，这里我们介绍两个最典型的方法代表，其一是基于连接幅度的方法[2]，其二是基于损失函数的方法[3]。

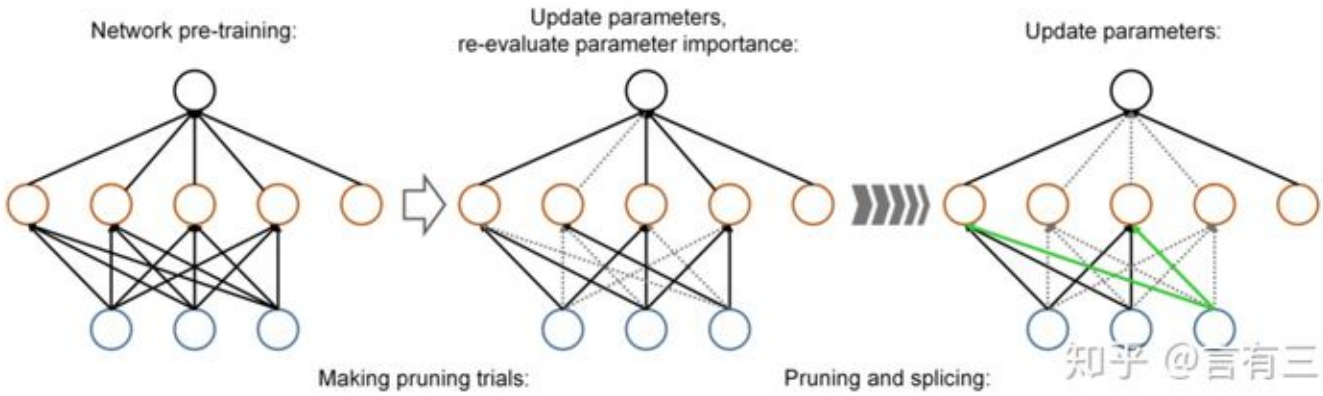
由于特征的输出是由输入与权重相乘后进行加权，权重的幅度越小，对输出的贡献越小，因此一种最直观的连接剪枝方法就是基于权重的幅度，如L1/L2范数的大小。这样的方法只需要三个步骤就能完成剪枝：

第一步：训练一个基准模型。

第二步：对权重值的幅度进行排序，去掉低于一个预设阈值的连接，得到剪枝后的网络。

第三步：对剪枝后网络进行微调以恢复损失的性能，然后继续进行第二步，依次交替，直到满足终止条件，比如精度下降在一定范围内。

这一类方法原理简单，前述提到的Google的方法也属于这一类。当然这类框架还有可以改进之处，比如Dynamic network surgery框架[4]观察到一些在当前轮迭代中虽然作用很小，但是在其他轮迭代中又可能重要，便在剪枝的基础上增加了一饿spliciing操作，即对一些被剪掉的权重进行恢复，如下：



基于权重幅度的方法原理简单，但这是比较主观的经验，即认为权重大就重要性高，事实上未必如此。而另一种经典的连接剪枝方法就是基于优化目标，根据剪枝对优化目标的影响来对其重要性进行判断，以最优脑损伤(Optimal Brain Damage, OBD)[3]方法为代表，这已经是上世纪90年代的技术了。

Optimal Brain Damage首先建立了一个误差函数的局部模型来预测扰动参数向量对优化目标造成的影响。具体来说用泰勒级数来近似目标函数E，参数向量U的扰动对目标函数的改变使用泰勒展开后如下：

$$\delta E = \sum_i g_i \delta u_i + \frac{1}{2} \sum_i h_{ii} \delta u_i^2 + \frac{1}{2} \sum_{i \neq j} h_{ij} \delta u_i \delta u_j + O(||R^-||^3)$$

$$g_i = \frac{\partial E}{\partial u_i} \quad \text{and} \quad h_{ij} = \frac{\partial^2 E}{\partial u_i \partial u_j}$$

其中gi是优化目标对参数u的梯度，而h是优化目标对参数u的海森矩阵。对模型剪枝的过程是希望找到一个参数集合，使得删除掉这个参数集合之后损失函数E的增加最小，由于上面的式子需要求解损失函数的海森矩阵H，这是一个维度为参数量平方的矩阵，几乎无法进行求解，为此需要对问题进行简化，这建立在几个基本假设的前提上：

(1) 参数独立。即删除多个参数所引起的损失的改变，等于单独删除每个参数所引起的损失改变的和，因此上式第三项可以去除。

(2) 局部极值。即剪枝是发生在模型已经收敛的情况下，因此第一项可以去除，并且hii都是正数，即剪枝一定会带来优化目标函数的增加，或者说带来性能的损失。

(3) 二次近似假定。即上式关系为二次项，最后一项可以去除。

经过简化后只剩下了第二项，只需要计算H矩阵的对角项。它可以基于优化目标对连接权重的导数进行计算，复杂度就与梯度计算相同了，如下：

$$h_{kk} = \sum_{(i,j) \in V_k} \frac{\partial^2 E}{\partial w_{ij}^2}$$

计算完之后就可以得到连接对优化目标改变的贡献，这就是它的重要性，因此可以进行剪枝，整个流程如下：

1. Choose a reasonable network architecture
2. Train the network until a reasonable solution is obtained
3. Compute the second derivatives h_{kk} for each parameter
4. Compute the saliencies for each parameter: $s_k = h_{kk} u_k^2/2$
5. Sort the parameters by saliency and delete some low-saliency parameters
6. Iterate to step 2

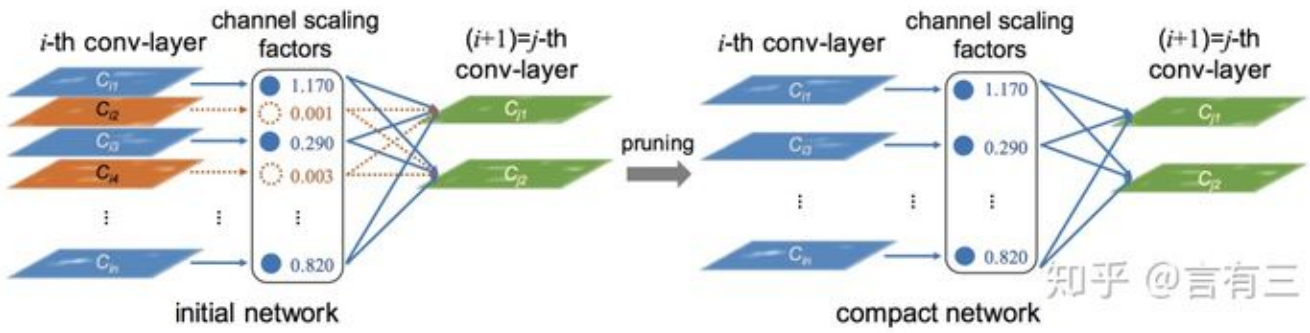
基于优化目标的权重剪枝方法还有很多，感兴趣的可以移步有三AI知识星球中的模型压缩板块。

2.2 粗粒度剪枝核心技术(通道剪枝)

相对于连接权重剪枝，粗粒度剪枝其实更加有用，它可以得到不需要专门的算法支持的精简小模型。对滤波器进行剪枝和对特征通道进行剪枝最终的结果是相同的，篇幅有限我们这里仅介绍特征通道的剪枝算法代表。

通道剪枝算法有三个经典思路。第一个是基于重要性因子，即评估一个通道的有效性，再配合约束一些通道使得模型结构本身具有稀疏性，从而基于此进行剪枝。第二个是利用重建误差来指导剪枝，间接衡量一个通道对输出的影响。第三个是基于优化目标的变化来衡量通道的敏感性。下面我们重点介绍前两种。

Network Trimming[5]通过激活的稀疏性来判断一个通道的重要性，认为拥有更高稀疏性的通道更应该被去除。它使用batch normalization中的缩放因子γ来对不重要的通道进行裁剪，如下图：

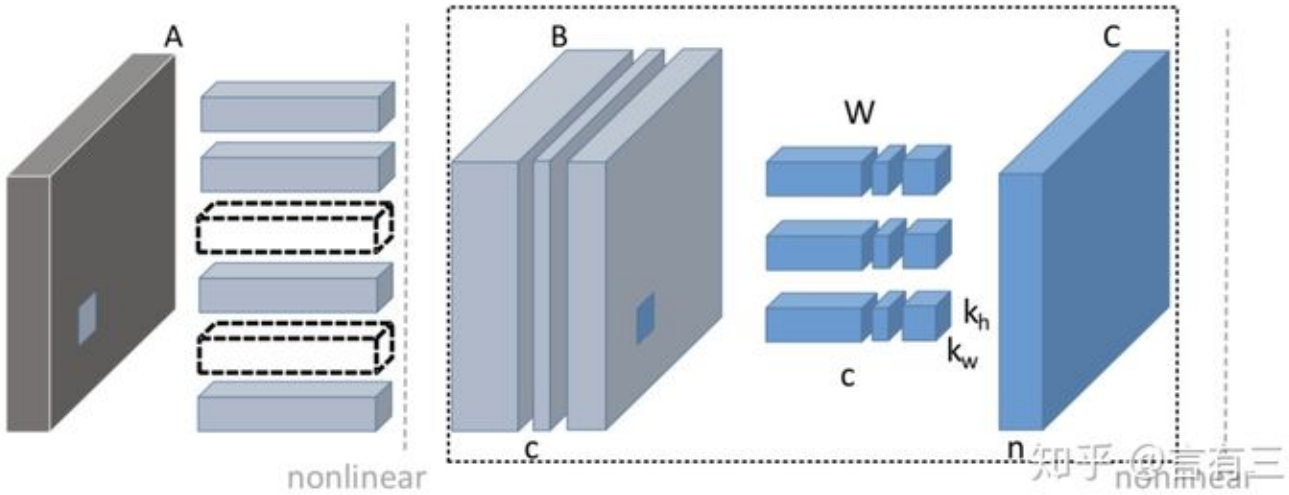


具体实现起来，就是在目标方程中增加一个关于γ的正则项，从而约束某些通道的重要性。

$$L = \sum_{(x,y)} l(f(x, W), y) + \lambda \sum_{\gamma \in \Gamma} g(\gamma)$$

类似的框架还有《Rethinking the smaller-norm-less-informative assumption in channel pruning of convolution layers》[6]，《Data-Driven Sparse Structure Selection》[7]，读者感兴趣可以自己学习或者移步有三AI知识星球。

与基于权重幅度的方法来进行连接剪枝一样，基于重要性因子的方法主观性太强，而另一种思路就是基于输出重建误差的通道剪枝算法[8]，它们根据输入特征图的各个通道对输出特征图的贡献大小来完成剪枝过程，可以直接反映剪枝前后特征的损失情况。



如上图，基于重建误差的剪枝算法，就是在剪掉当前层B的若干通道后，重建其输出特征图C使得损失信息最小。假如我们要将B的通道从c剪枝到c'，要求解的就是下面的问题，第一项是重建误差，第二项是正则项。

$$\arg\min_{\boldsymbol{\beta},\mathrm{W}} \frac{1}{2N}\left\|\mathrm{Y}-\sum_{i=1}^{c}\beta_i \mathrm{X_i}\mathrm{W_i}^{\top}\right\|_F^2+\lambda\left\|\boldsymbol{\beta}\right\|_1$$

$$\text{subject to } \left\|\boldsymbol{\beta}\right\|_0 \leq c', \forall i \left\|\mathrm{W_i}\right\|_F = 1$$

该问题可以分两步进行求解。

第一步：选择候选的裁剪通道。

我们可以对输入特征图按照卷积核的感受野进行多次随机采样，获得输入矩阵X，权重矩阵W，输出Y。然后将W用训练好的模型初始化，逐渐增大正则因子，每一次改变都进行若干次迭代，直到beta稳定，这是一个经典的LASSO回归问题求解。

第二步：固定beta求解W，完成最小化重建误差，需要更新使得下式最小。

$$\arg\min_{\mathrm{W}'}\left\|\mathrm{Y}-\mathrm{X}'(\mathrm{W}')^{\top}\right\|_F^2$$

以上两个步骤交替进行优化，最后迭代完剪枝后，就可以得到新的权重。类似的框架还有ThiNet[9]等，更多就移步有三AI知识星球吧。

以上就是粗粒度剪枝和细粒度剪枝中最主流的方法的一些介绍，当然这还只是一小部分。

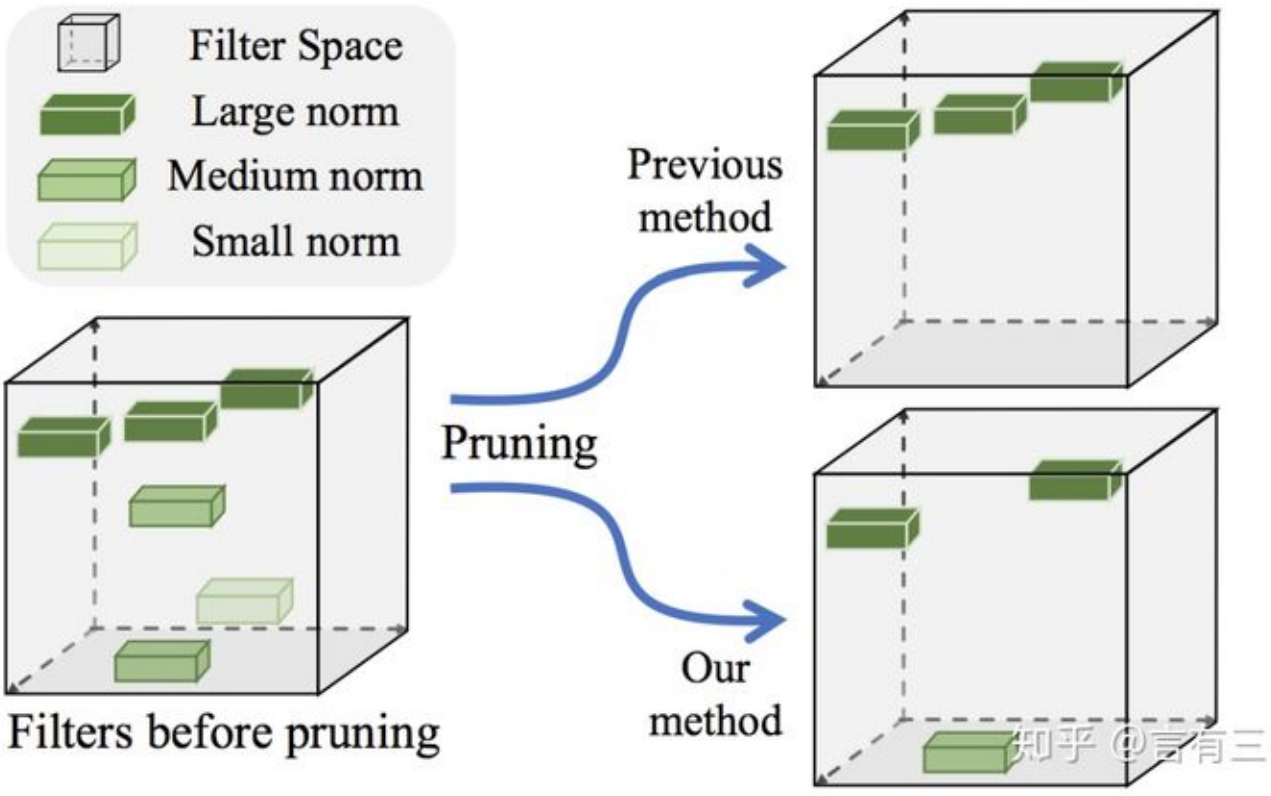
3 剪枝算法的展望

上一节我们对一些代表性的方法的核心思想进行了总结，它们只能代表模型剪枝算法的一小部分，而模型剪枝仍然是当前学术界和工业界的研究热点，至少还有几个方向值得关注。

3.1 重要性因子选择

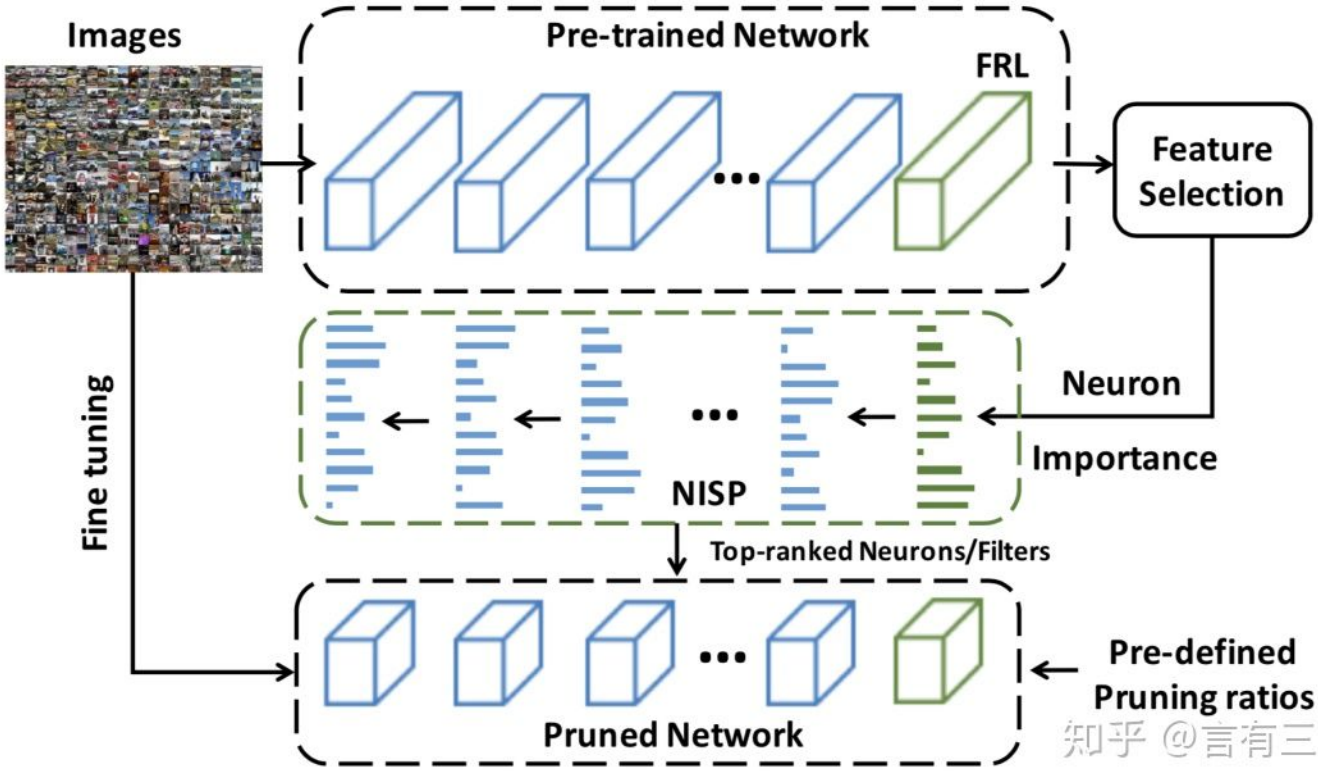
不管是连接剪枝还是通道剪枝，前面都提到了重要性因子，即我们通过某种准则来判断一个连接或者通道是否重要，比如范数。这是非常直观的思想，因为它们影响了输出的大小。但这类方法的假设前提条件太强，需要权重和激活值本身满足一定的分布。

Geometric Median[10]方法就利用了几何中位数对范数进行替换，那是否有更多更好的指标呢？这非常值得关注。



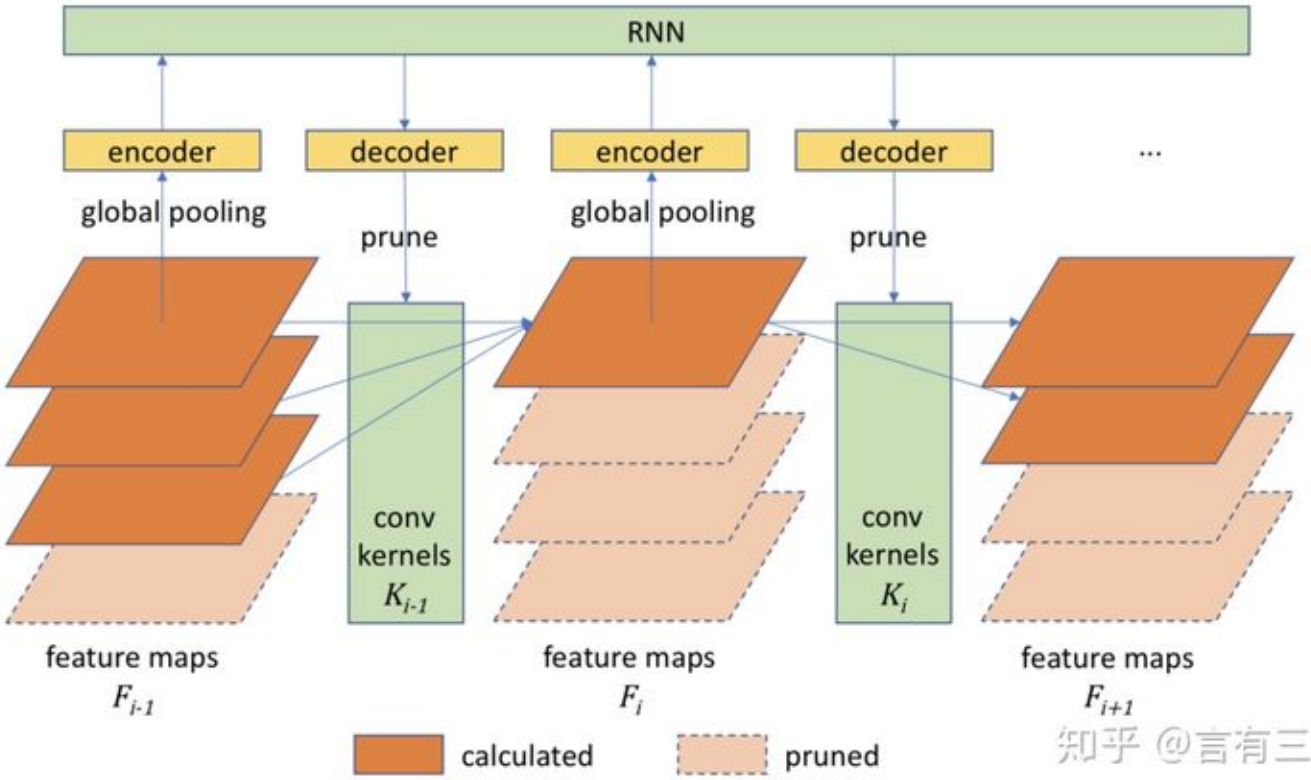
3.2 剪枝流程优化

当前大部分框架都是逐层进行剪枝，而没有让各层之间进行联动，这其实是有问题的。因为在当前阶段冗余的模块，并不意味着对其他阶段也是冗余的。以NISP[11]为代表的方法就通过反向传播来直接对整个网络神经元的重要性进行打分，一次性完成整个模型的剪枝。



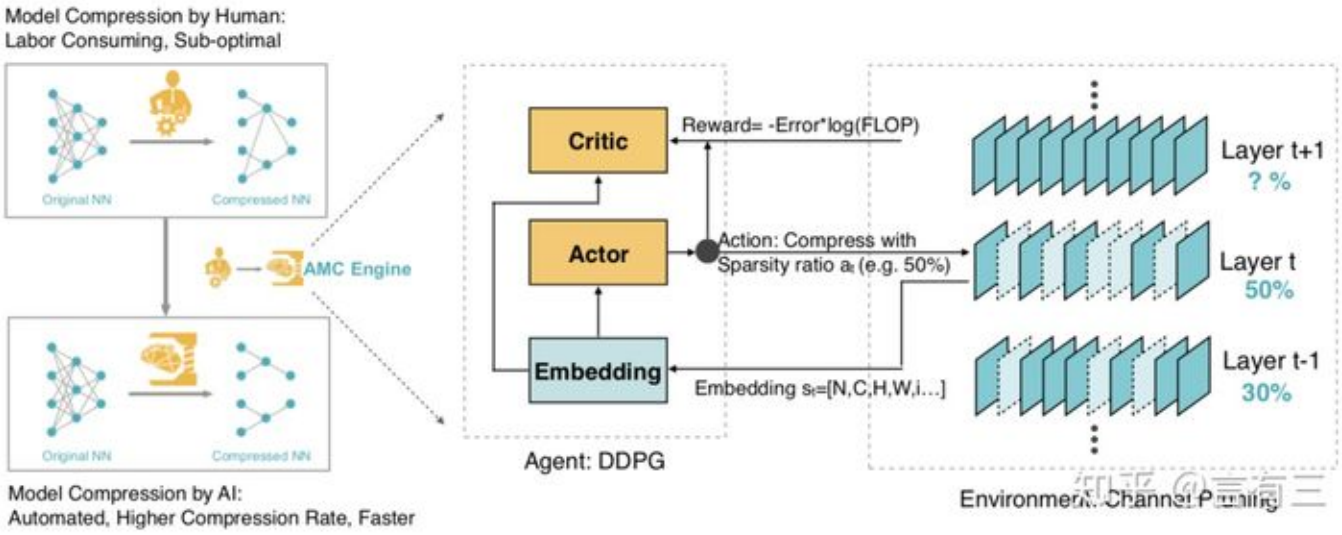
3.3 个性化剪枝

通常来说，模型在剪枝完后进行推理时不会发生变化，即对于所有的输入图片来说都是一样的计算量，但是有的样本简单，有的样本复杂，以前我们给大家介绍过动态推理框架，它们可以对不同的输入样本图配置不同的计算量，剪枝框架也可以采用这样的思路，以Runtime Neural Pruning [12]为代表。

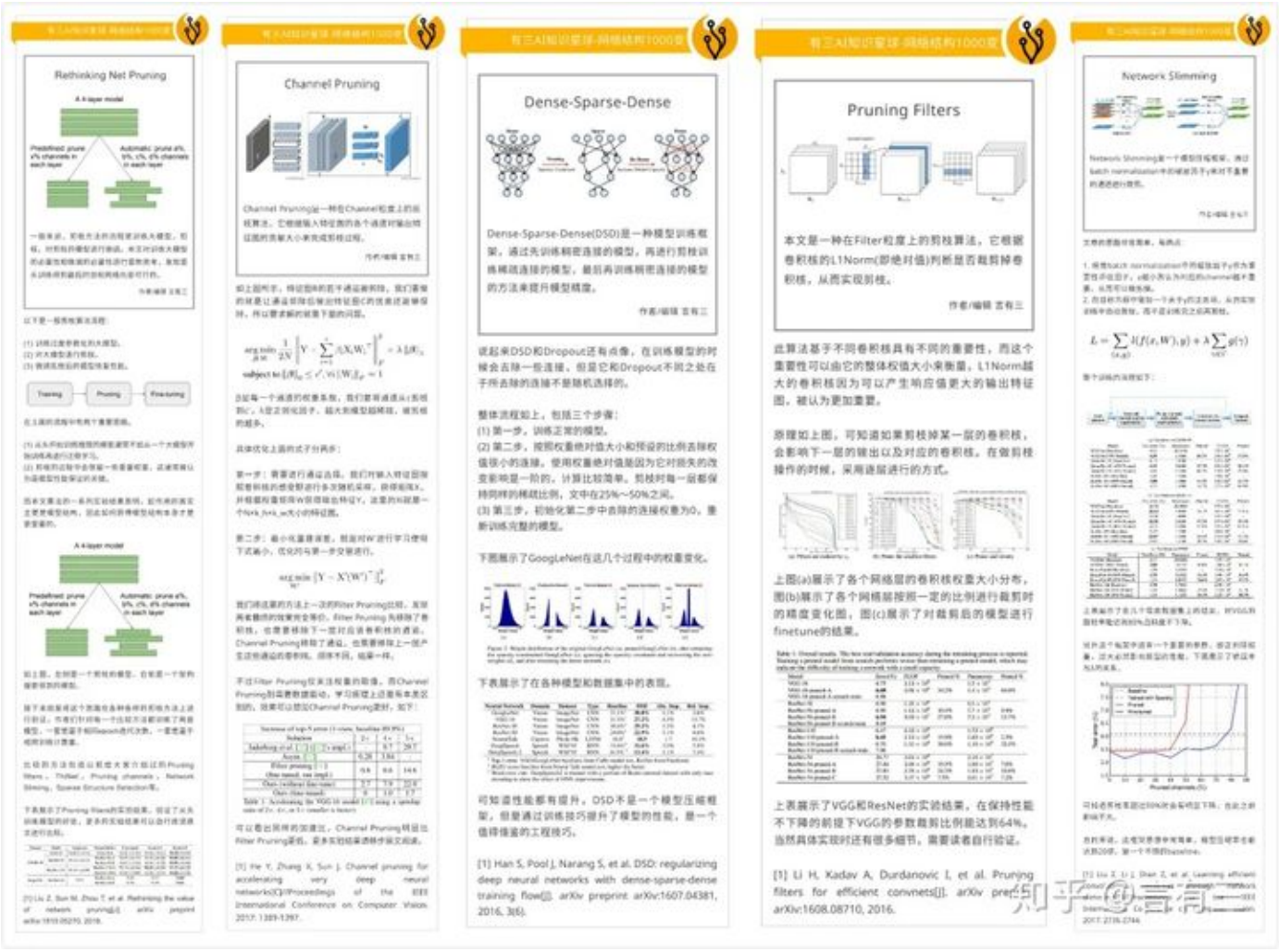


3.4 自动化剪枝

剪枝中我们通常遵循一些基本策略：比如在提取低级特征的参数较少的第一层中剪掉更少的参数，对冗余性更高的FC层剪掉更多的参数。然而，由于深度神经网络中的层不是孤立的，这些基于规则的剪枝策略并不是最优的，也不能从一个模型迁移到另一个模型，因此AutoML方法的应用也是非常自然的，AutoML for Model Compression(AMC)是其中的代表[13]，我们以前也做过介绍。



除此之外，还有训练前剪枝，注意力机制增强等等许多方向，我们不再一一介绍，对模型剪枝感兴趣的同学，欢迎到**有三AI知识星球的网络结构1000变-模型压缩-模型剪枝板块**进行学习，数十期内容定能满足你的求知欲。



扫码即可加入，了解有三AI知识星球详情请阅读以下文章。

【杂谈】万万没想到，有三还有个保密的‘朋友圈’，那里面都在弄啥！

mp.weixin.qq.com

【杂谈】有三AI知识星球一周年了！为什么公众号+星球才是完整的？

mp.weixin.qq.com

参考文献

[1] Zhu M, Gupta S. To prune, or not to prune: exploring the efficacy of pruning for model compression[J]. arXiv: Machine Learning, 2017.

[2] Han S, Pool J, Tran J, et al. Learning both weights and connections for efficient neural network[C]//Advances in neural information processing systems. 2015: 1135-1143.

[3] LeCun Y, Denker J S, Solla S A. Optimal brain damage[C]//Advances in neural information processing systems. 1990: 598-605.

[4] Guo Y, Yao A, Chen Y. Dynamic network surgery for efficient dnns[C]//Advances In Neural Information Processing Systems. 2016: 1379-1387.

[5] Liu Z, Li J, Shen Z, et al. Learning efficient convolutional networks through network slimming[C]//Proceedings of the IEEE International Conference on Computer Vision. 2017: 2736-2744.

[6] Ye J, Lu X, Lin Z, et al. Rethinking the smaller-norm-less-informative assumption in channel pruning of convolution layers[J]. arXiv preprint arXiv:1802.00124, 2018.

[7] Huang Z, Wang N. Data-driven sparse structure selection for deep neural networks[C]//Proceedings of the European Conference on Computer Vision (ECCV). 2018: 304-320.

[8] He Y, Zhang X, Sun J. Channel pruning for accelerating very deep neural networks[C]//Proceedings of the IEEE International Conference on Computer Vision. 2017: 1389-1397.

[9] Luo J H, Zhang H, Zhou H Y, et al. Thinet: pruning cnn filters for a thinner net[J]. IEEE transactions on pattern analysis and machine intelligence, 2018.

[10] He Y , Liu P , Wang Z , et al. Filter Pruning via Geometric Median for Deep Convolutional Neural Networks Acceleration[J]. 2018.

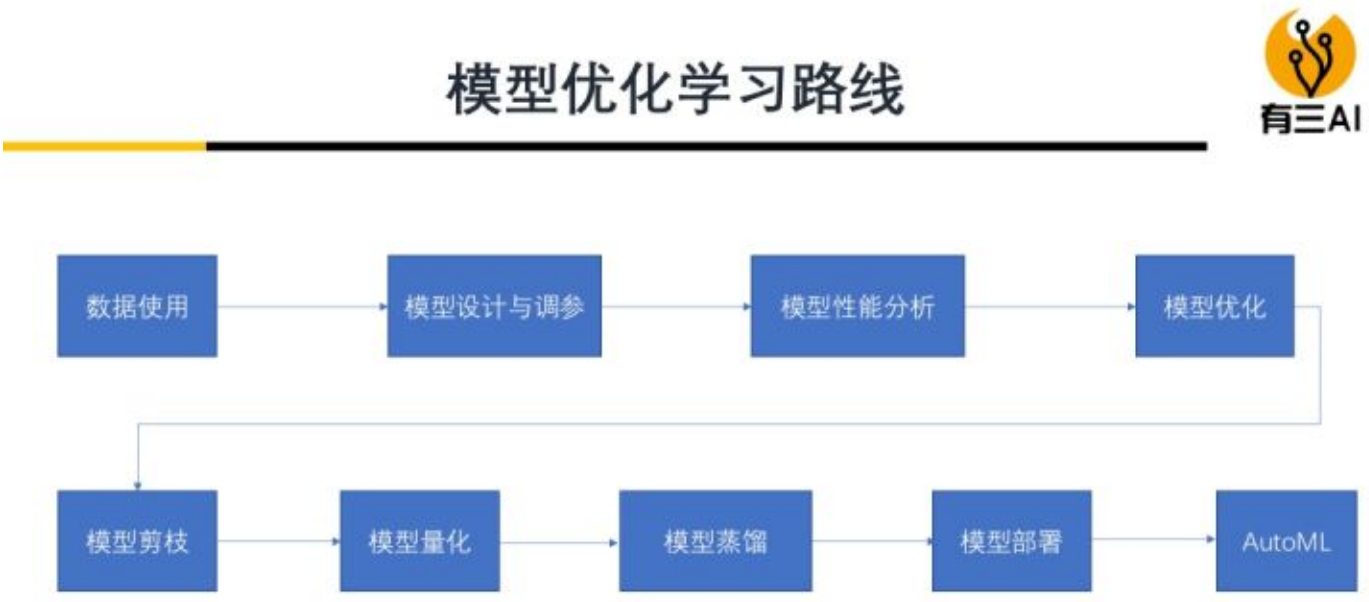
[11] Yu R, Li A, Chen C F, et al. Nisp: Pruning networks using neuron importance score propagation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 9194-9203.

[12] Lin J, Rao Y, Lu J, et al. Runtime Neural Pruning[C]. neural information processing systems, 2017: 2181-2191.

[13] He Y, Lin J, Liu Z, et al. AMC: AutoML for Model Compression and Acceleration on Mobile Devices[C]. european conference on computer vision, 2018: 815-832.

总结

本次我们总结了模型剪枝的核心技术，并对其重要方向进行了展望，推荐了相关的学习资源，下一期我们将介绍量化相关内容。



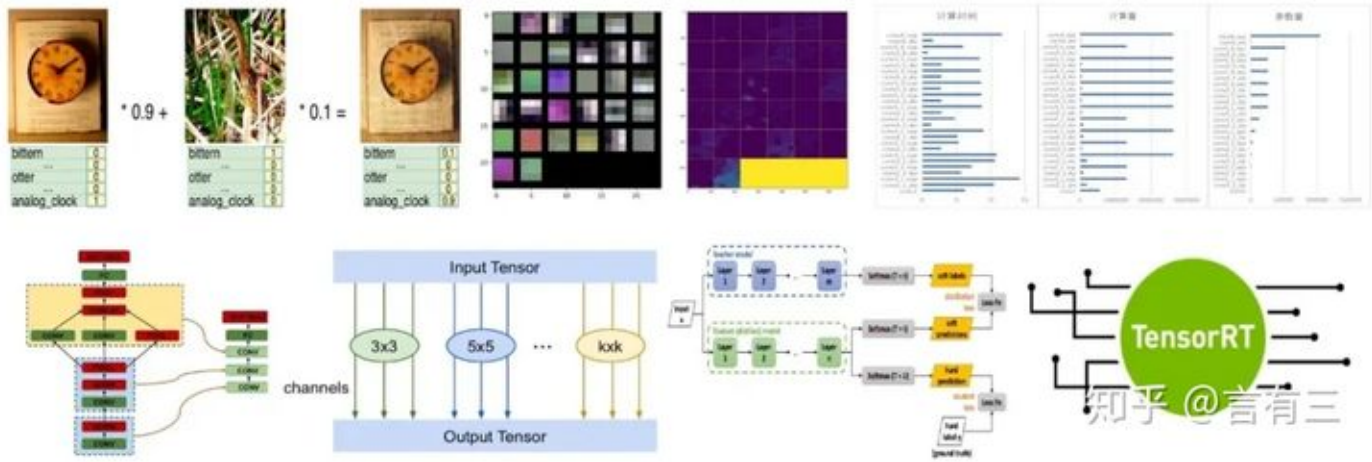
有三AI秋季划-模型优化组

如果你想系统性地学习模型优化相关的理论和实践，并获得持续的指导，欢迎加入有三AI秋季划-模型优化组，系统性地学习数据使用，模型使用和调参，模型性能分析，紧凑模型设计，模型剪枝，模型量化，模型部署，NAS等内容。

模型优化组介绍和往期的一些学习内容总结请参考阅读以下文章：

有三AI秋季划出炉，模型优化，人脸算法，图像质量等24个项目等你来拿
mp.weixin.qq.com

【总结】有三AI秋季划模型优化组3月直播讲了哪些内容，为什么每一个从事深度...
mp.weixin.qq.com



> AI白身境系列完整阅读：

第一期：【AI白身境】深度学习从弃用windows开始

第二期：【AI白身境】Linux干活三板斧，shell、vim和git

第三期：【AI白身境】学AI必备的python基础

第四期：【AI白身境】深度学习必备图像基础

第五期：【AI白身境】搞计算机视觉必备的OpenCV入门基础

第六期：【AI白身境】只会用Python？g++，CMake和Makefile了解一下

第七期：【AI白身境】学深度学习你不得不知的爬虫基础

第八期：【AI白身境】深度学习中的数据可视化

第九期：【AI白身境】入行AI需要什么数学基础：左手矩阵论，右手微积分

第十期：【AI白身境】一文览尽计算机视觉研究方向

第十一期：【AI白身境】AI+，都加在哪些应用领域了

第十二期：【AI白身境】究竟谁是paper之王，全球前10的计算机科学家

> AI初识境系列完整阅读

第一期：【AI初识境】从3次人工智能潮起潮落说起

第二期：【AI初识境】从头理解神经网络-内行与外行的分水岭

第三期：【AI初识境】近20年深度学习在图像领域的重要进展节点

第四期：【AI初识境】激活函数：从人工设计到自动搜索

第五期：【AI初识境】什么是深度学习成功的开始？参数初始化

第六期：【AI初识境】深度学习模型中的Normalization，你懂了多少？

第七期：【AI初识境】为了围剿SGD大家这些年想过的那十几招

第八期：【AI初识境】被Hinton，DeepMind和斯坦福嫌弃的池化，到底是什么？

第九期：【AI初识境】如何增加深度学习模型的泛化能力

第十期：【AI初识境】深度学习模型评估，从图像分类到生成模型

第十一期：【AI初识境】深度学习中常用的损失函数有哪些？

第十二期：【AI初识境】给深度学习新手开始项目时的10条建议

> AI不惑境系列完整阅读：

第一期：【AI不惑境】数据压榨有多狠，人工智能就有多成功

第二期：【AI不惑境】网络深度对深度学习模型性能有什么影响？

第三期：【AI不惑境】网络的宽度如何影响深度学习模型的性能？

第四期：【AI不惑境】学习率和batchsize如何影响模型的性能？

第五期：【AI不惑境】残差网络的前世今生与原理

第六期：【AI不惑境】移动端高效网络，卷积拆分和分组的精髓

第七期：【AI不惑境】深度学习中的多尺度模型设计

第八期：【AI不惑境】计算机视觉中注意力机制原理及其模型发展和应用

第九期：【AI不惑境】模型剪枝技术原理及其发展现状和展望

第十期：【AI不惑境】模型量化技术原理及其发展现状和展望

第十一期：【AI不惑境】模型压缩中知识蒸馏技术原理及其发展现状和展望

第十二期：【AI不惑境】AutoML在深度学习模型设计和优化中有哪些用处？

编辑于 2020-05-25

人工智能　深度学习（Deep Learning）　有三AI