先进编译实验室
Advanced Compiler

深度学习模型压缩方法（一）
知识蒸馏

嘉宾： 唐文生

先进编译实验室
Advanced Compiler

知识蒸馏

1、知识蒸馏简介

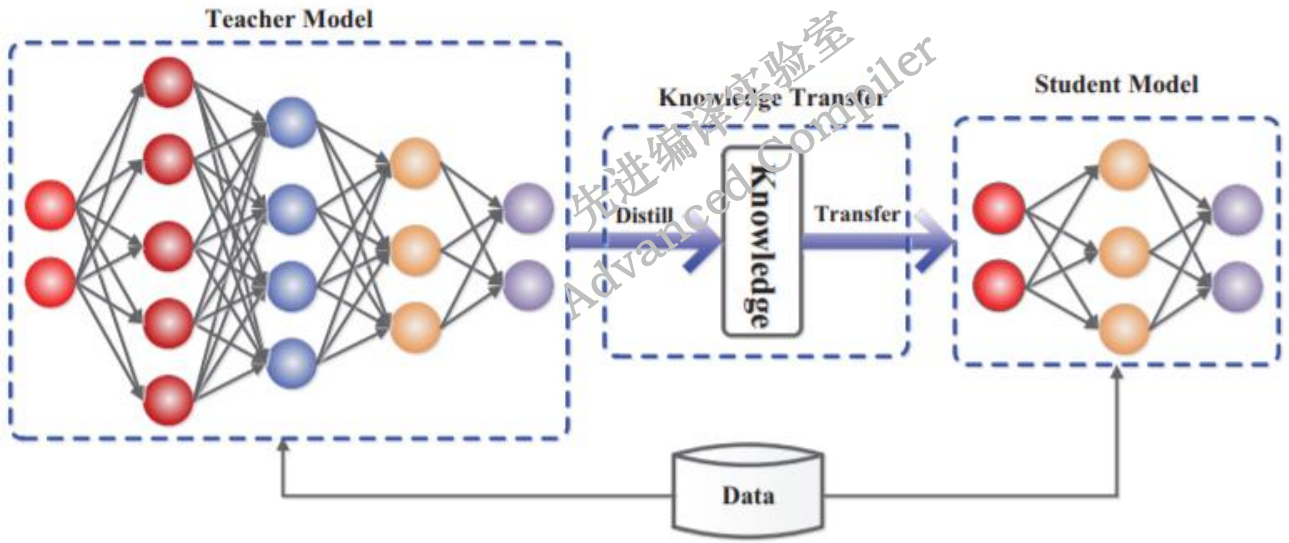
2、知识的种类

3、蒸馏机制

4、师生网络结构

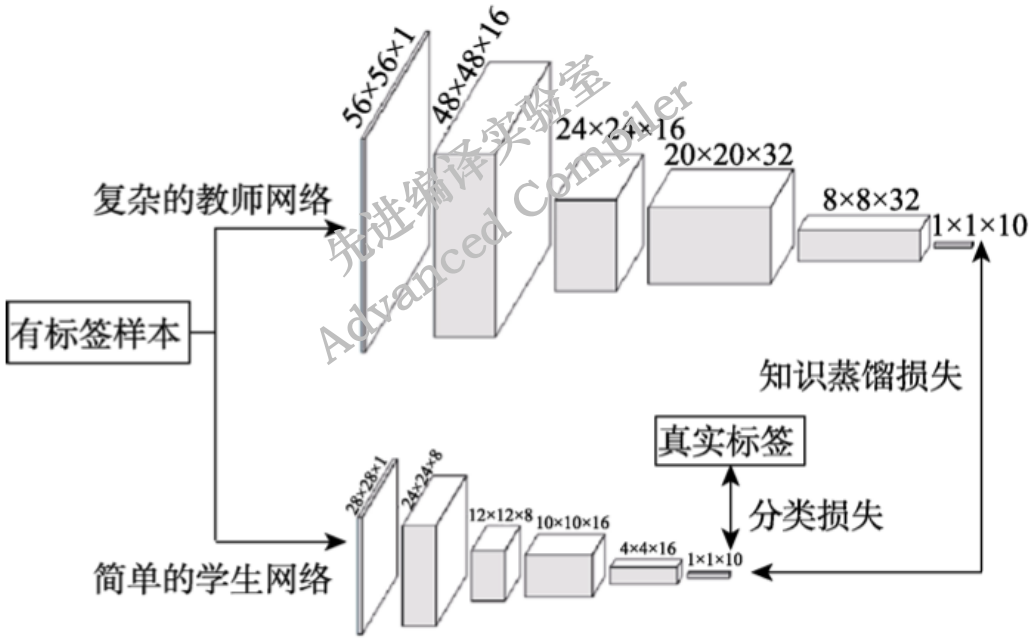
5、蒸馏算法

6、蒸馏方法



1、知识蒸馏简介

知识蒸馏是指通过教师模型指导学生模型训练，通过蒸馏的方式让学生模型学习到教师模型的知识，最终使学生模型达到或媲美老师模型的准确度。



网络

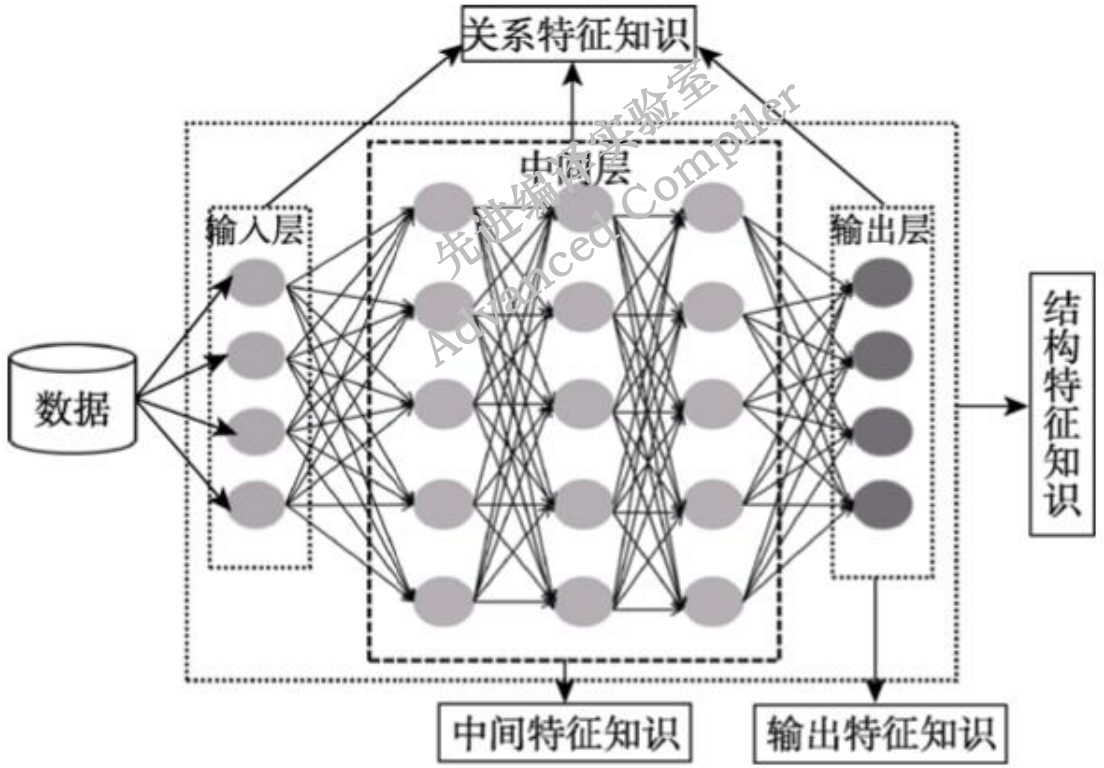
2、知识种类

1、输出特征知识

2、中间特征知识

3、关系特征知识

4、结构特征知识

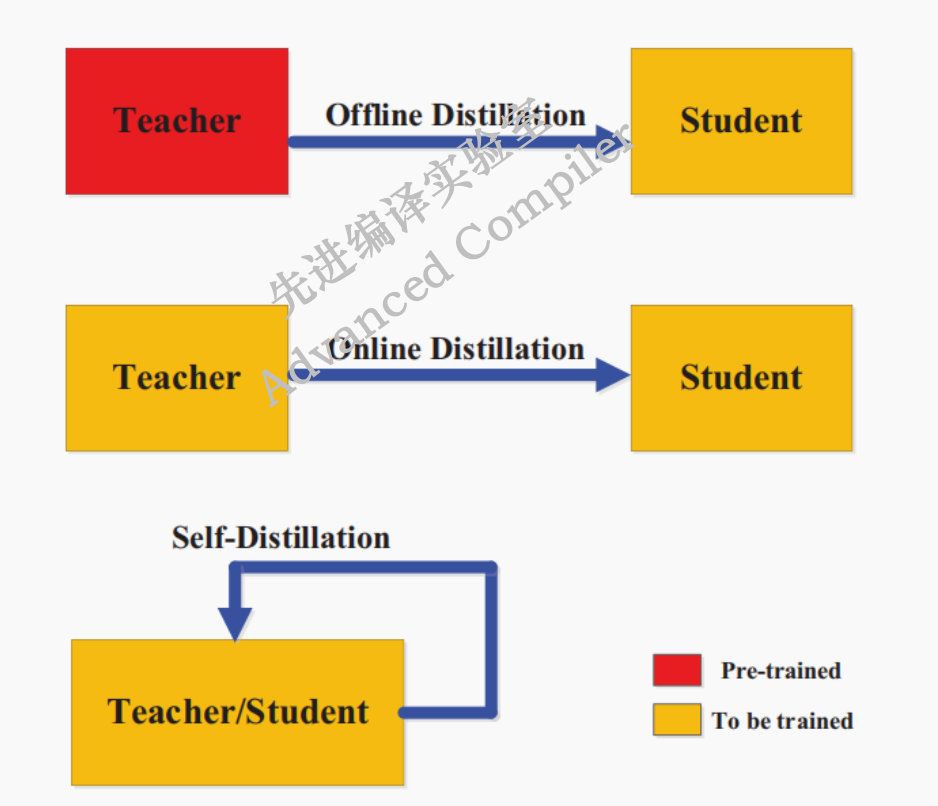


3、蒸馏机制

1、离线蒸馏

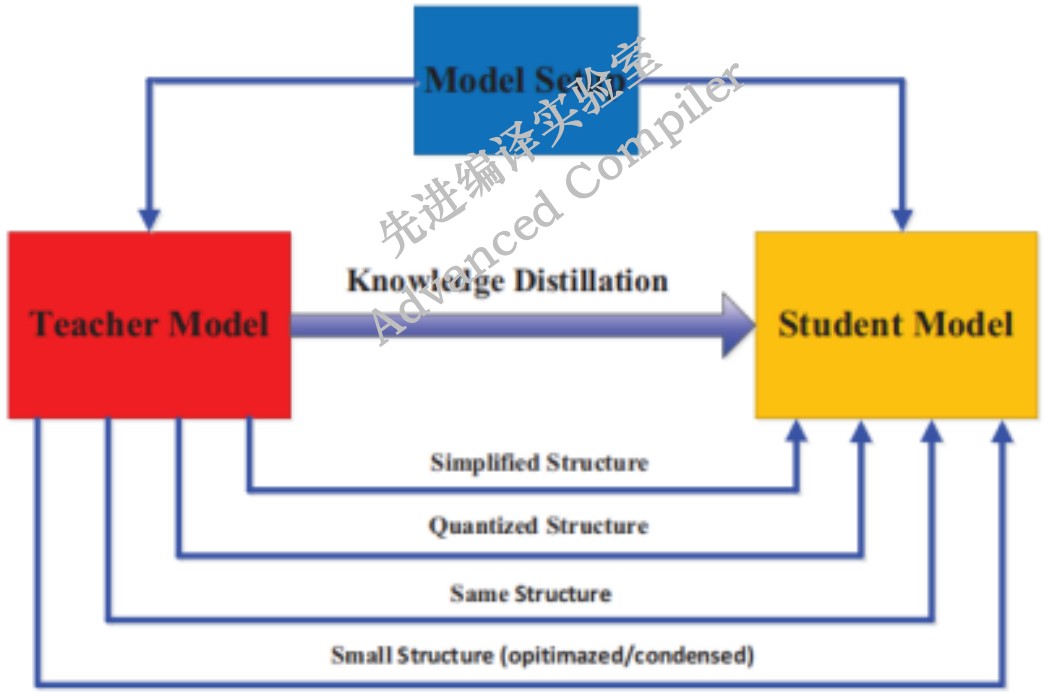
2、在线蒸馏

3、自蒸馏



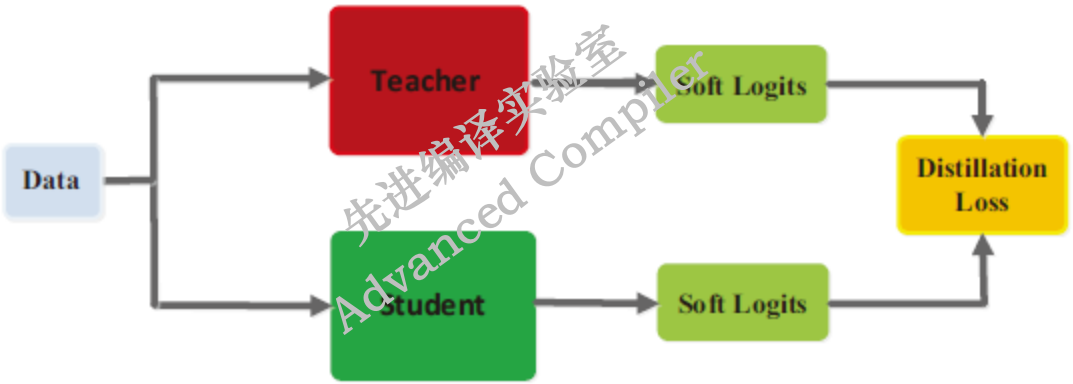
4、师生网络架构

学生网络一般是:
1）教师网络的简化版本，具有较少的层和每层中较少的信道。
2）教师网络的量化版本，其中网络的结构被保留。
3）具有高效基本操作的小型网络。
4）具有优化的整体网络结构的小型网络。
5）与教师相同的网络。



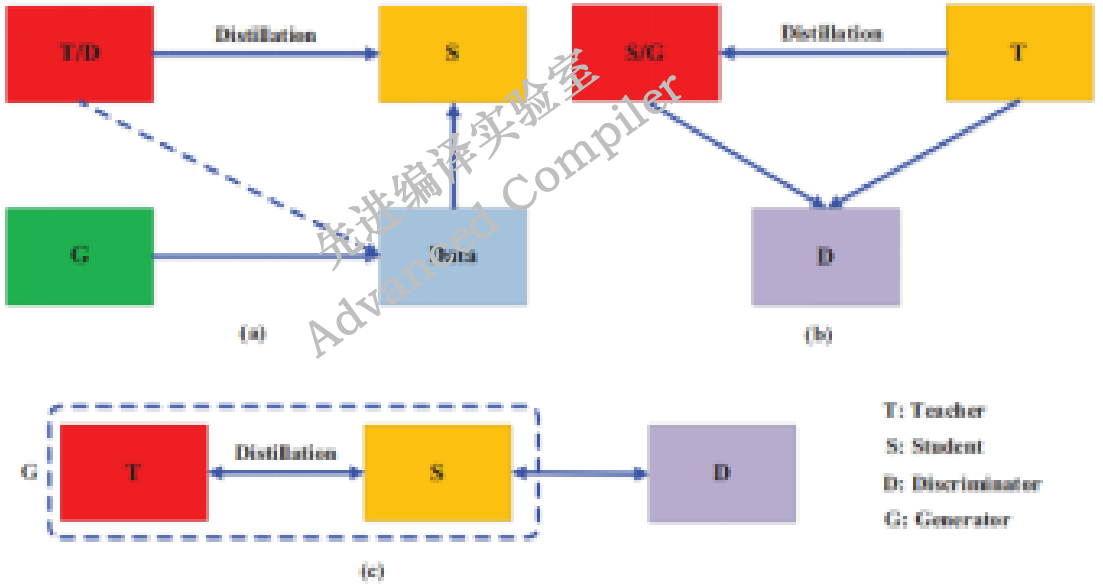
5、蒸馏算法

为了改进在更复杂的环境中传递知识的过程，已经出现了许多不同的知识蒸馏算法。下面，我们一起回顾知识蒸馏领域中最近提出的几种典型的蒸馏方法。



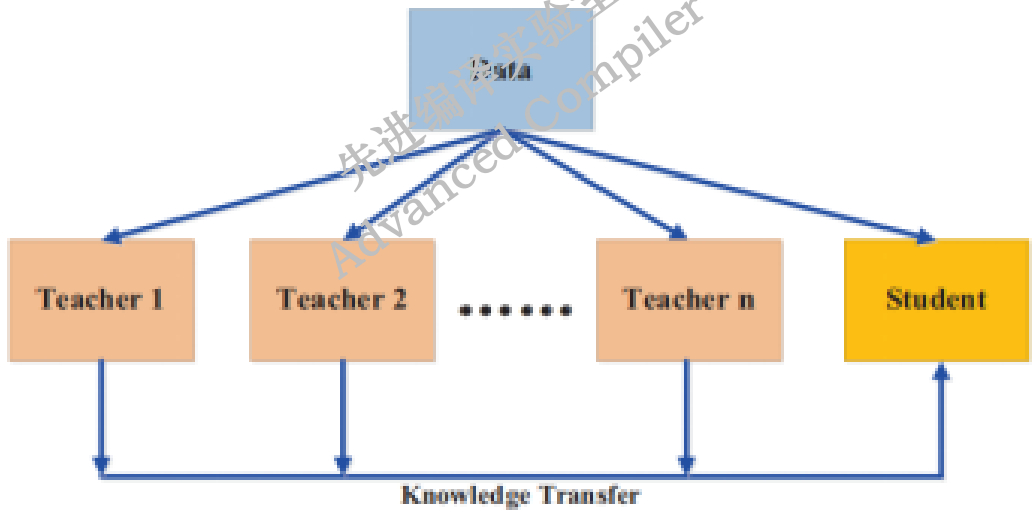
5.1、对抗蒸馏

在对抗性学习中，对抗网络中的鉴别器用来估计样本来自训练数据分布的概率，而生成器试图使用生成的数据样本来欺骗鉴别器。受此启发，已经出现了许多基于对抗的知识蒸馏方法，以使教师和学生网络能够更好地理解真实的数据分布。



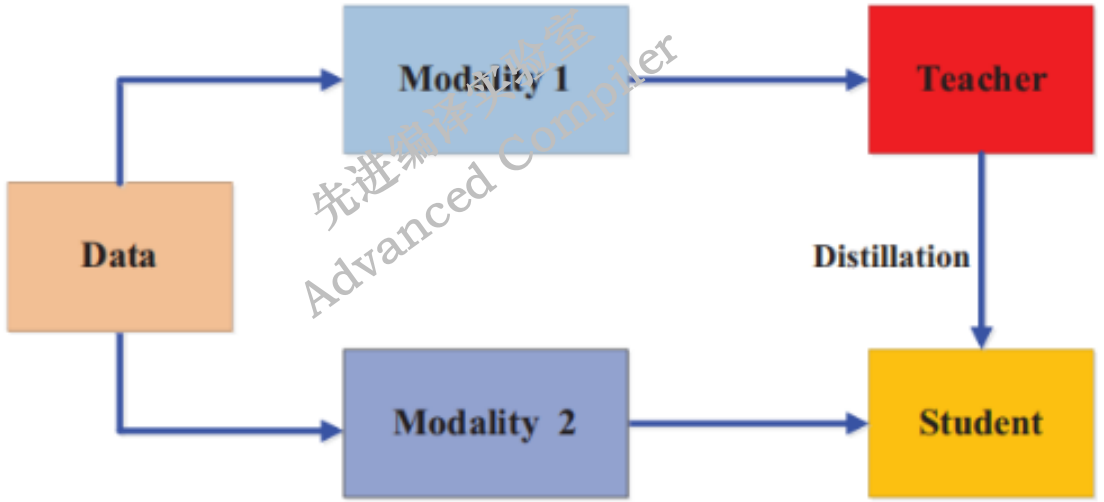
5.2、多教师蒸馏

不同的教师架构可以为学生网络提供不同有用的知识。在训练学生网络期间，多个教师网络可以单独地，也可以整体地用于蒸馏。为了传递来自多个教师的知识，最简单的方法是使用来自所有教师的平均响应作为监督信号。



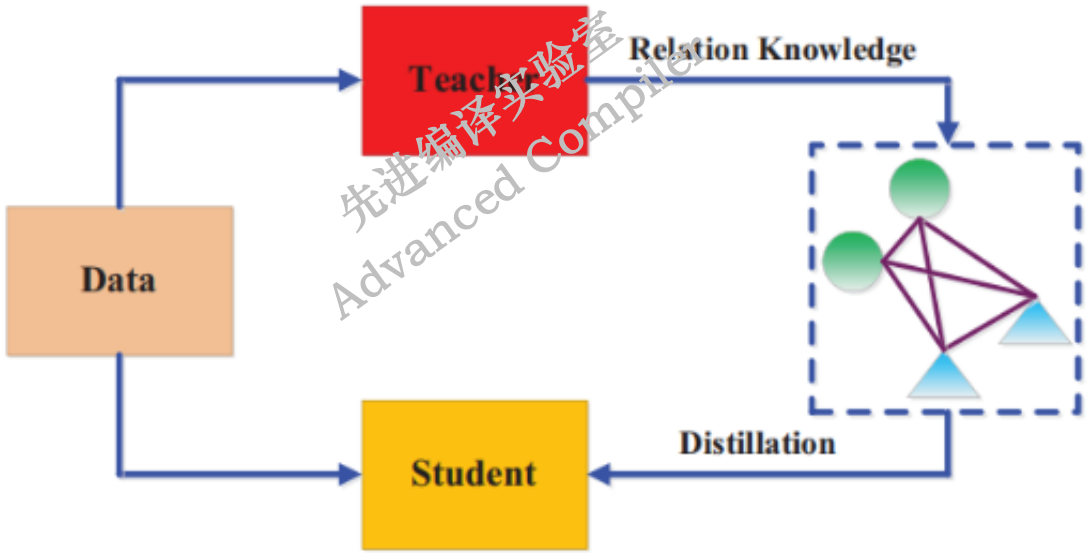
5.3、交叉模式蒸馏

在训练或测试期间，某些数据或标签可能不可用。因此，在不同的模型之间传递知识是很重要的。然而，当模型存在差异时，跨模型知识蒸馏是一项具有挑战性的研究，例如，当不同模式之间缺乏配对的样本时。



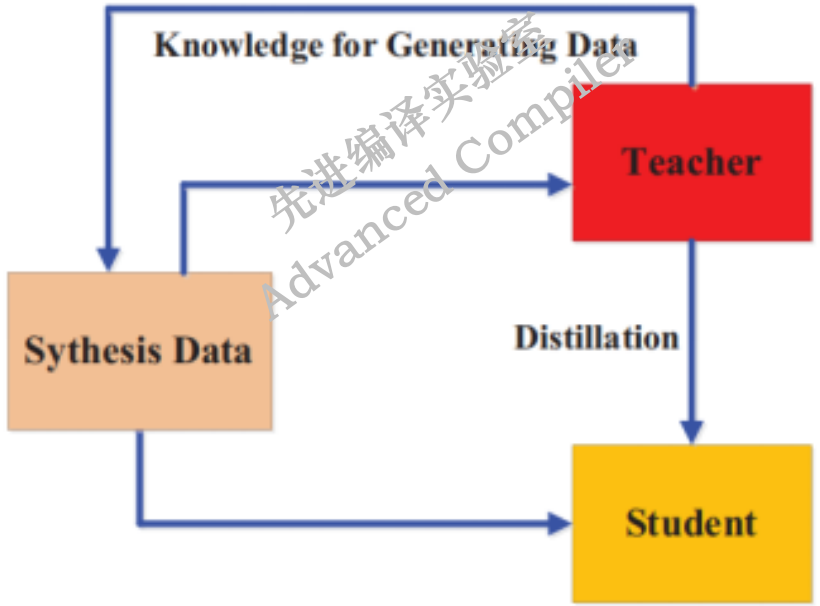
5.4、基于图形的蒸馏

基于图的蒸馏方法的主要思想是：
1、用图作为教师知识的载体；
2、用图来控制教师知识的信息传递。



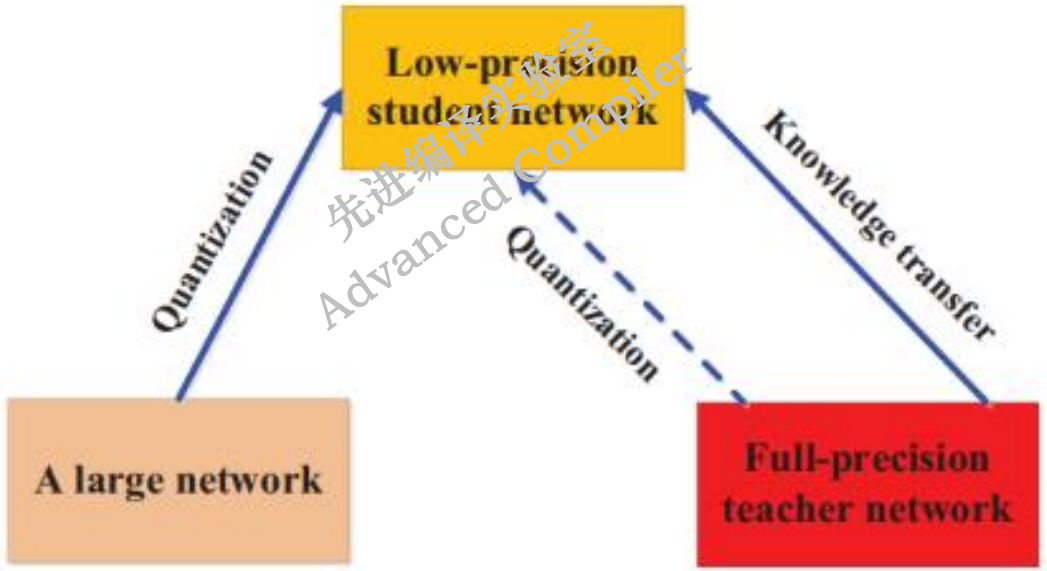
5.5、无数据蒸馏

为了克服由隐私、合法性、安全性和保密性问题等原因引起的不可用数据的问题，出现了一些无数据知识蒸馏的方法。无数据蒸馏中的合成数据通常是从预训练教师模型的特征表示中生成的



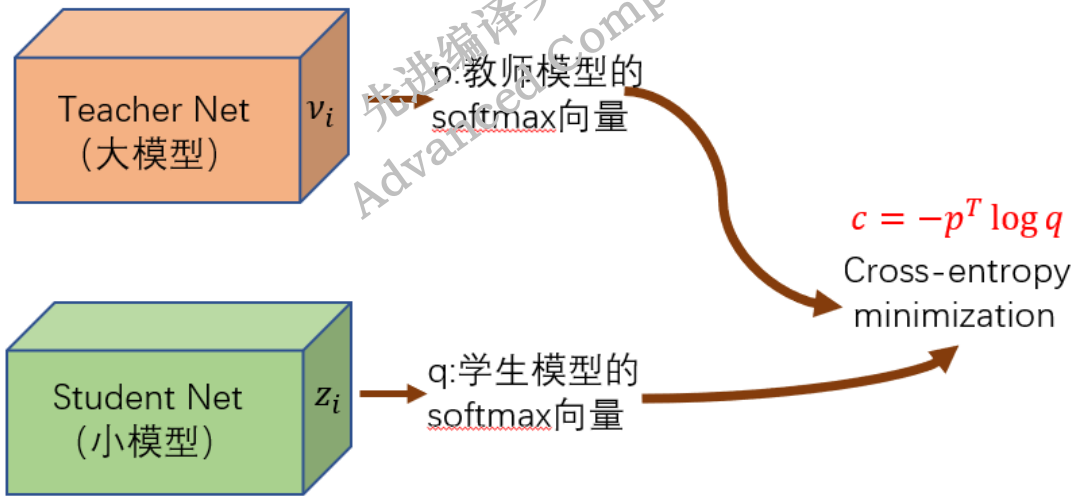
5.6、量化蒸馏

一个大的高精度的教师网络将知识传递给一个小的低精度的学生网络。为了确保小的学生网络精确地模仿大的教师网络，首先在特征图上量化教师网络，然后将知识从量化的教师转移到量化的学生网络。



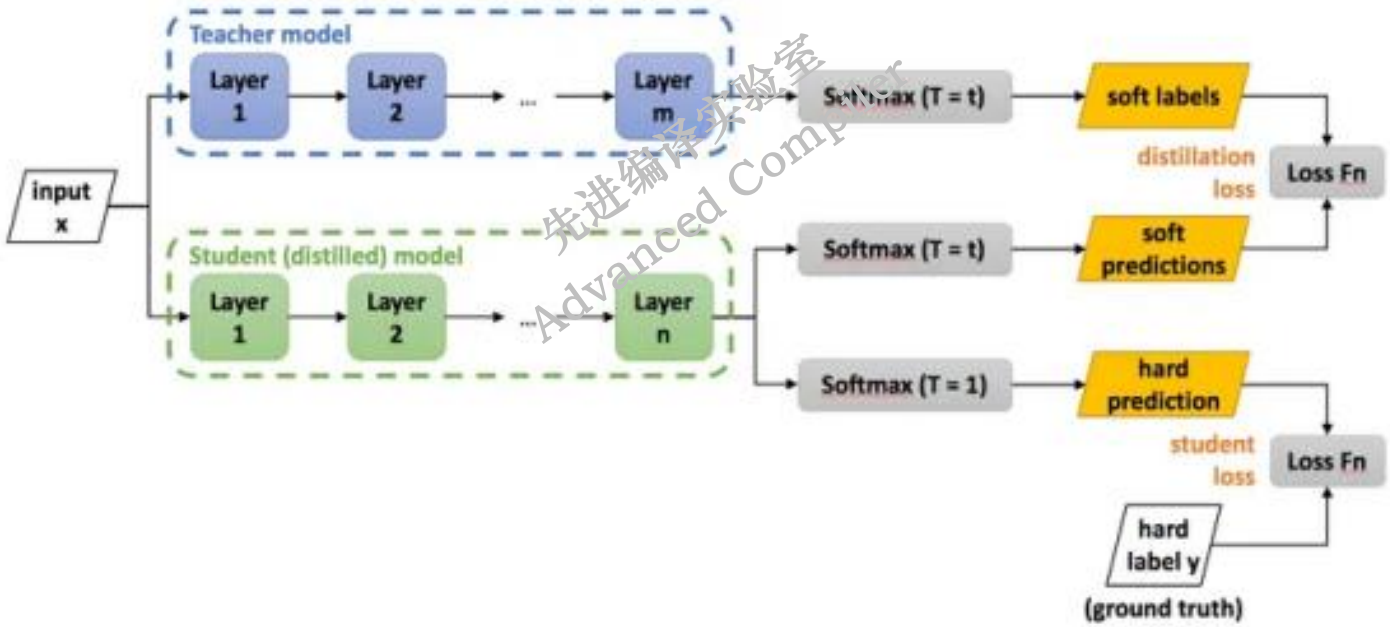
5.7、其他知识蒸馏算法

基于注意力的蒸馏，终身蒸馏，NAS蒸馏。



6、蒸馏流程

step1：训练 Teacher 模型。
step2：利用高温T产生Soft-target，用T=1产生Hard-target。
step3：利用{高温T，Soft-target}和{T=1，Hard-target} 同时训练 Student 模型。
step4：设置T=1，Student模型线上做推理。



6、蒸馏流程

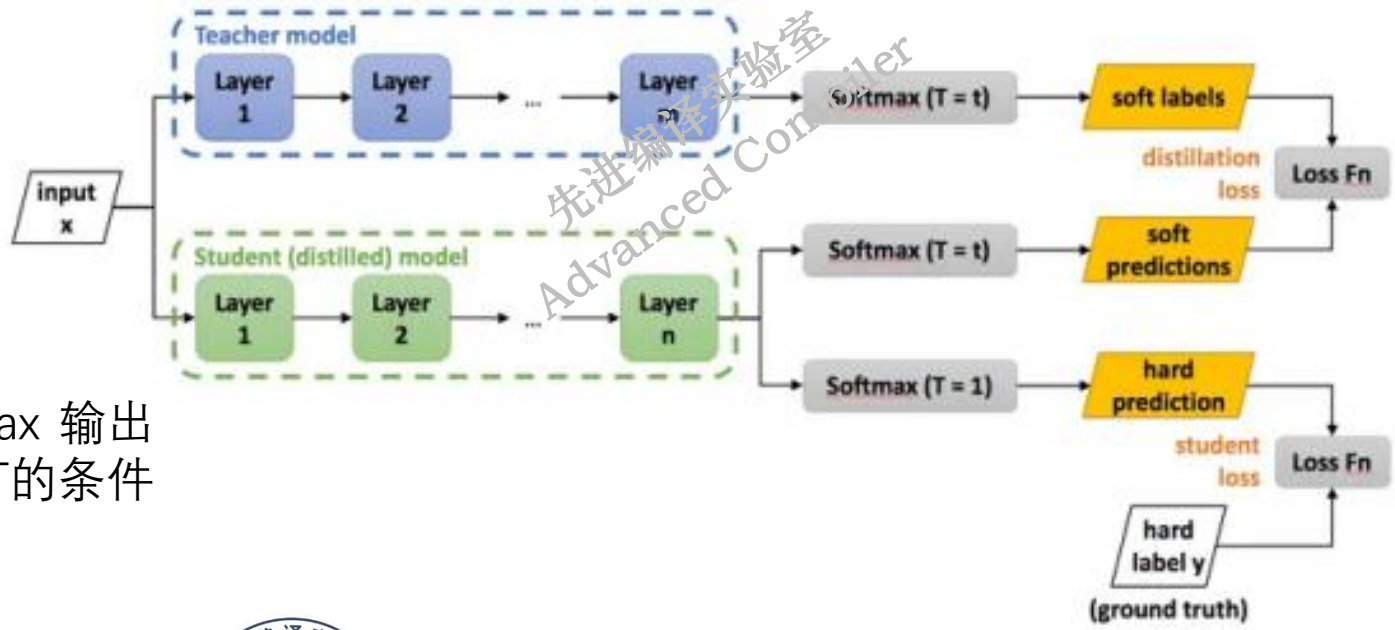
我们把步骤2和步骤3统一称为：高温蒸馏的过程。

损失函数：$L = \alpha L_{soft} + \beta L_{hard}$

其中：$L_{soft} = -\sum_i^N p_i^T \log(q_i^T)$　　$L_{hard} = -\sum_i^N c_i \log(q_i^1)$

p_i^T是 Teacher 模型在温度等于T的条件下 softmax 输出在第i类上的值，q_i^T是Student 模型在温度等于T的条件下 softmax 输出在第i类上的值。

$$p_i^T = \frac{exp(v_i/T)}{\sum_k^N exp(v_k/T)} \qquad q_i^T = \frac{exp(z_i/T)}{\sum_k^N exp(z_j/T)}$$



感谢大家聆听

本期视频主要参考了《Knowledge Distillation: A Survey.》这篇文章。

先进编译实验室
Advanced Compiler

先进编译实验室
Advanced Compiler

深度学习模型压缩方法（二）
剪枝

嘉宾： 唐文生

先进编译实验室
Advanced Compiler

剪枝

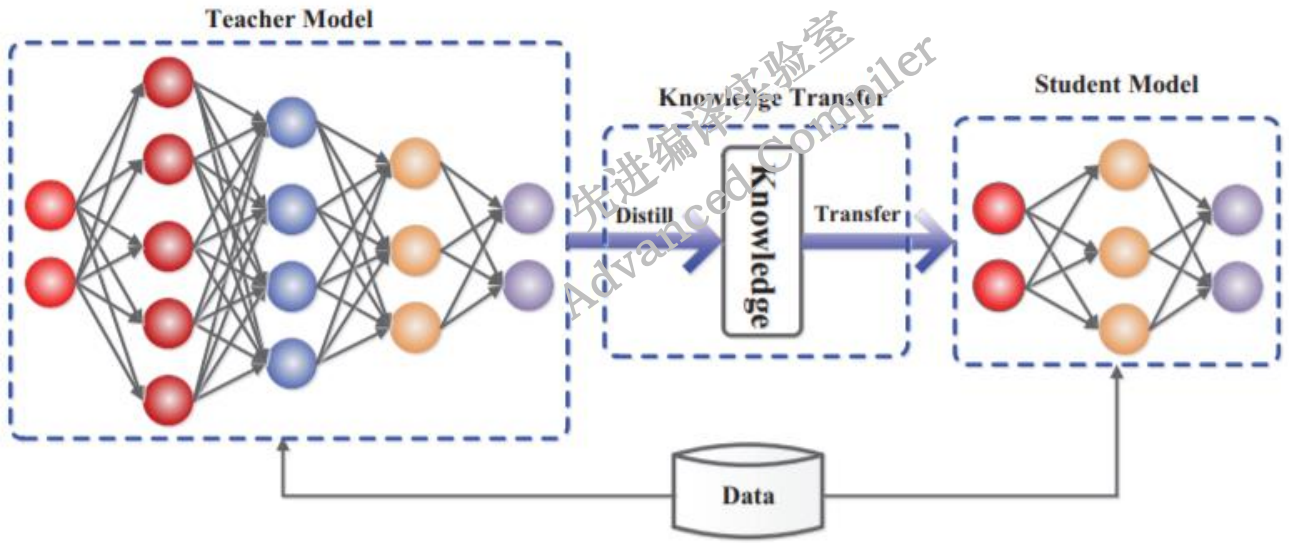
1、剪枝简介

2、剪枝步骤

3、结构化剪枝与非结构化剪枝

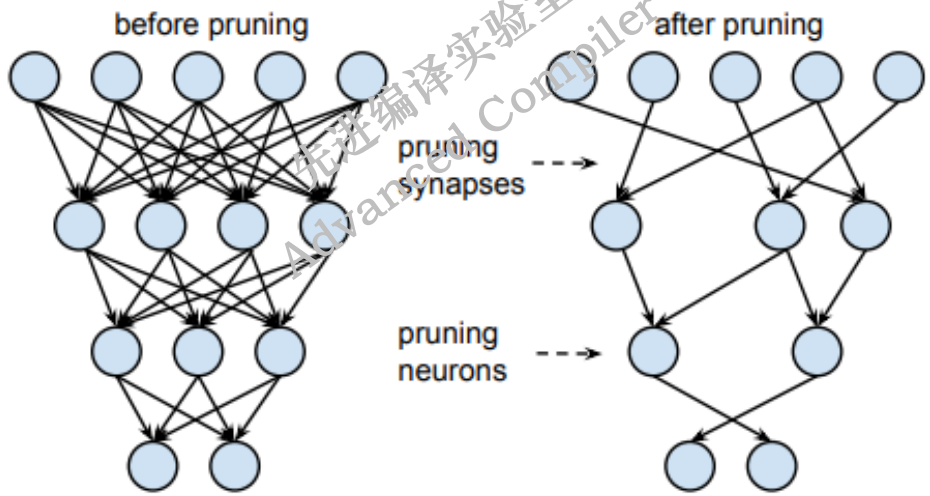
4、静态剪枝与动态剪枝

5、硬剪枝与软剪枝



1、剪枝简介

过参数化主要是指在训练阶段，在数学上需要进行大量的微分求解，去捕捉数据中微小的变化信息，一旦完成迭代式的训练之后，网络模型在推理的时候就不需要这么多参数。而剪枝算法正是基于过参数化的理论基础提出来的。剪枝算法核心思想就是减少网络模型中参数量和计算量，同时尽量保证模型的性能不受影响。



网络

2、剪枝步骤

对模型进行剪枝三种常见做法：

1）训练一个模型 ——> 对模型进行剪枝 ——> 对剪枝后模型进行微调

2）在模型训练过程中进行剪枝 ——> 对剪枝后模型进行微调

3）进行剪枝——> 从头训练剪枝后模型

剪枝步骤

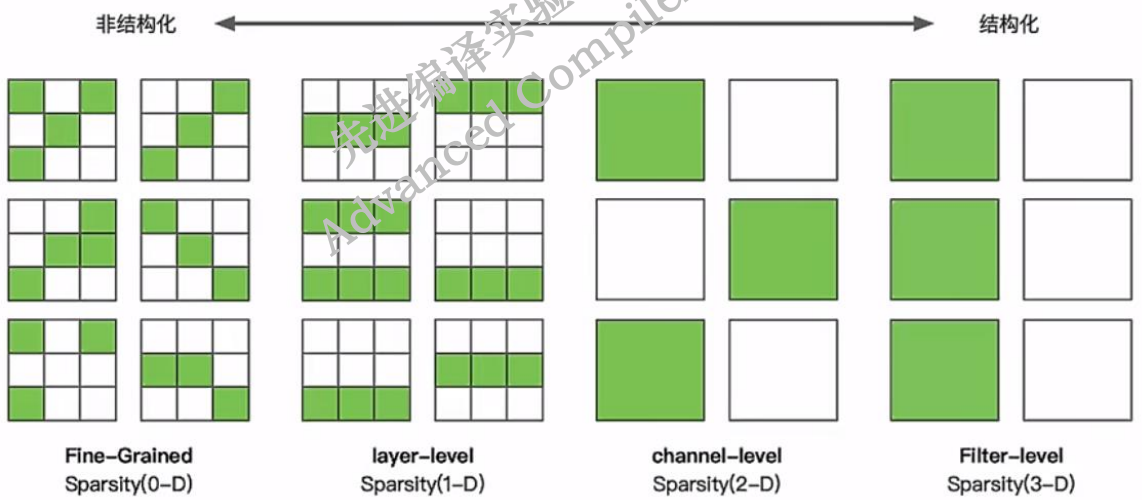
训练 Training：是对网络模型进行训练。

剪枝 Pruning：在这里面可以进行如细粒度剪枝、向量剪枝、核剪枝、滤波器剪枝等各种不同的剪枝算法。

微调 Finetune：微调是恢复被剪枝操作影响的模型表达能力的必要步骤。

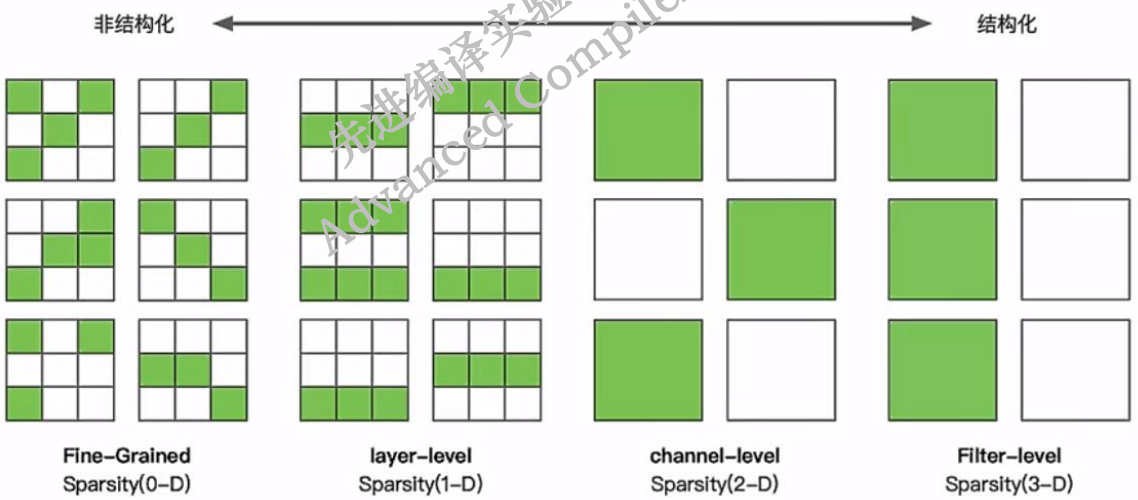
3、结构化剪枝与非结构化剪枝

最左边的非结构化剪枝粒度最小，右边结构化剪枝中的层级、通道级、滤波器级剪枝粒度依次增大。



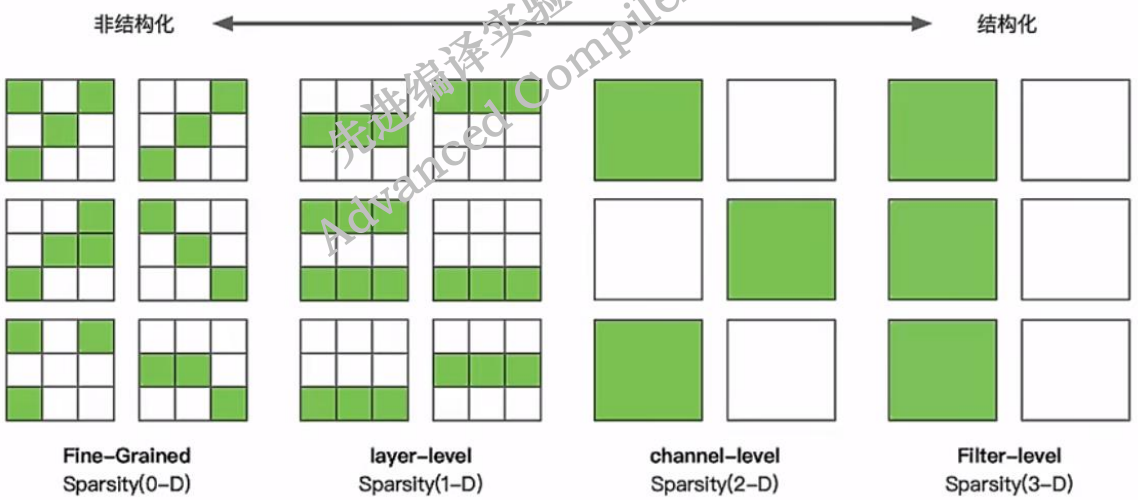
3.1、非结构化剪枝

非结构化剪枝主要是对一些独立的权重或者神经元在或者一些神经元的连接接进行剪枝，就是随机的剪，是粒度最小的剪枝。



3.1、非结构化剪枝

最简单的方法是预定义一个阈值，低于这个阈值的权重被剪去，高于的被保留。



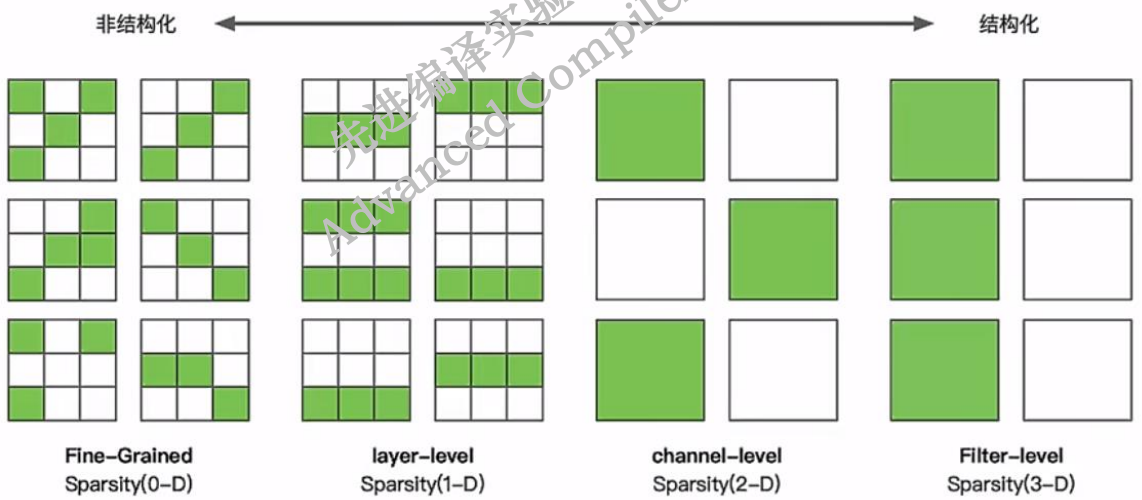
3.1、非结构化剪枝

还有一种方法是使用一个拼接函数来屏蔽权重。

$$\Delta w = -\eta \frac{\partial \mathcal{L}}{\partial (h(w)w)}$$

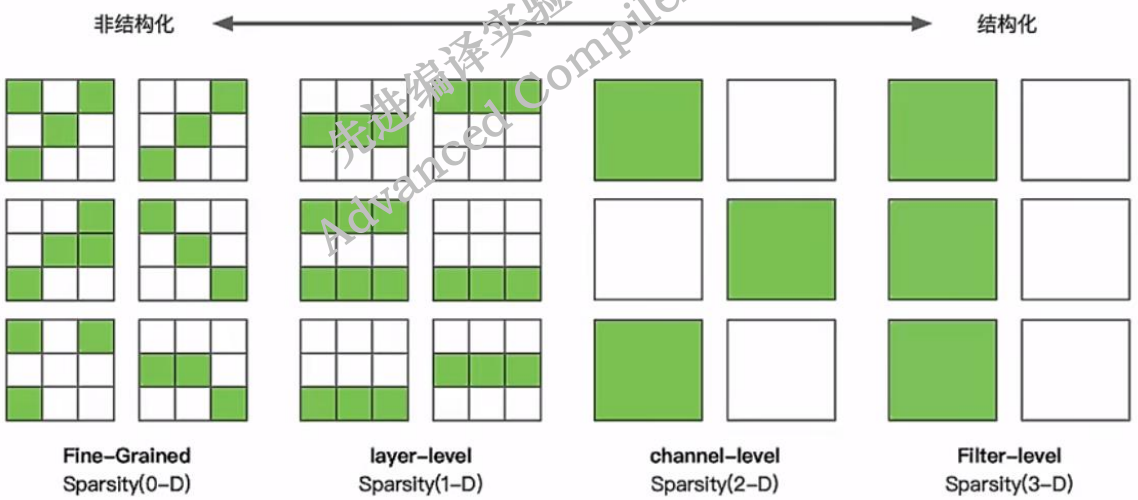
里面h（w）逐渐将不必要的权重较少到0

$$h(w) = \begin{cases} 0, & \text{if } a > |w|, \\ T, & \text{if } a \leq |w| < b, \\ 1, & \text{if } b \leq |w|. \end{cases}$$



3.2、结构化剪枝

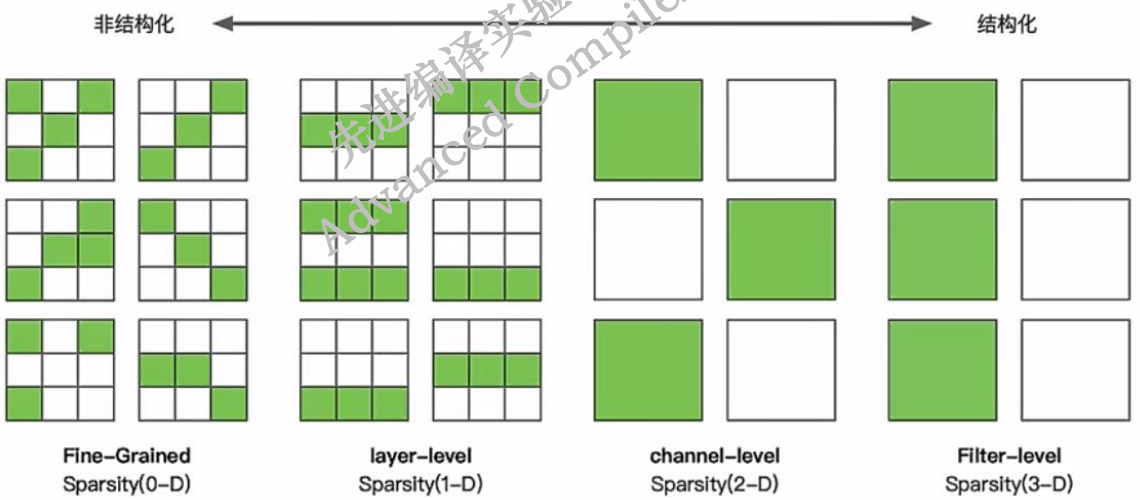
右边的这三个图是结构化剪枝，结构化的剪枝是有规律、有顺序的。对神经网络，或者计算图进行剪枝，几个比较经典的就是对layer进行剪枝，对channel进行剪枝，对Filter进行剪枝，剪枝粒度依次增大。



3.2、结构化剪枝

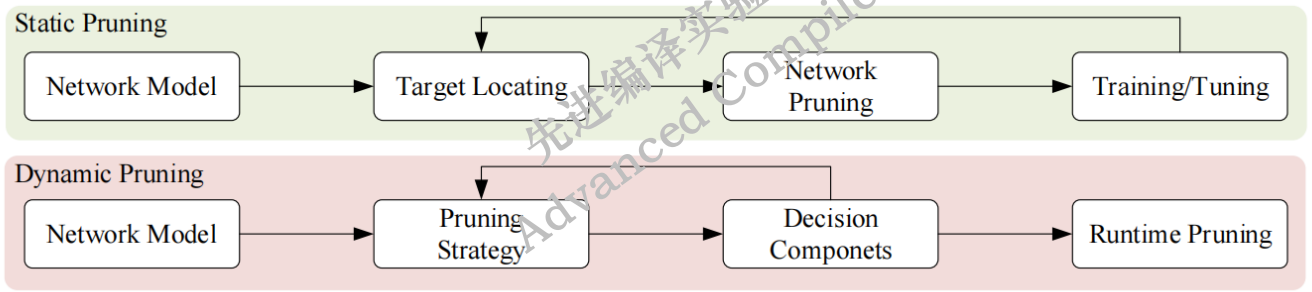
在滤波器剪枝中，有一种方法是使用滤波器的Lp范数（p为1则使用L1范数，p为2则使用L2范数）来评估每个滤波器的重要性。

$$\|\mathcal{F}_{i,j}\|_p = \sqrt[p]{\sum_{n=1}^{N_i}\sum_{k_1=1}^{K}\sum_{k_2=1}^{K}|\mathcal{F}_{i,j}(n,k_1,k_2)|^p},$$



4、静态剪枝与动态剪枝

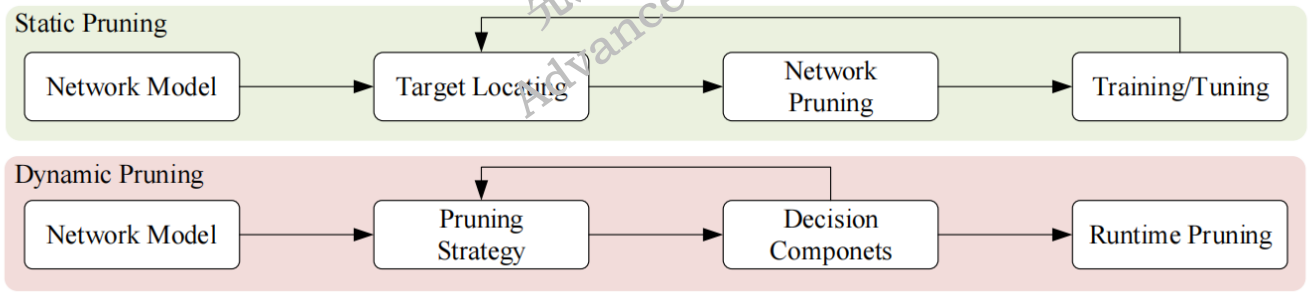
这张图显示了静态剪枝和动态剪枝之间的差异。静态剪枝在推断之前离线执行所有剪枝步骤，而动态剪枝在运行时执行。



4.1、静态剪枝

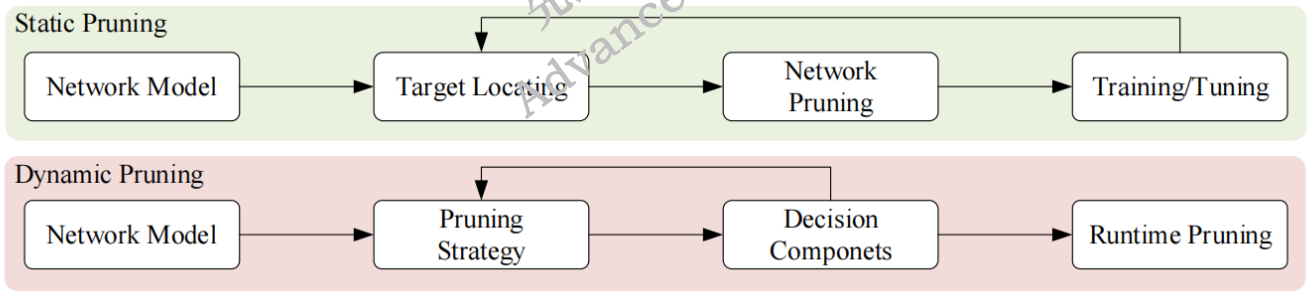
静态剪枝在训练后和推理前进行剪枝。在推理过程中，不需要对网络进行额外的剪枝。

静态剪枝通常包括三个部分:
1 剪枝参数的选择;
2 剪枝的方法;
3 选择性微调或再训练。



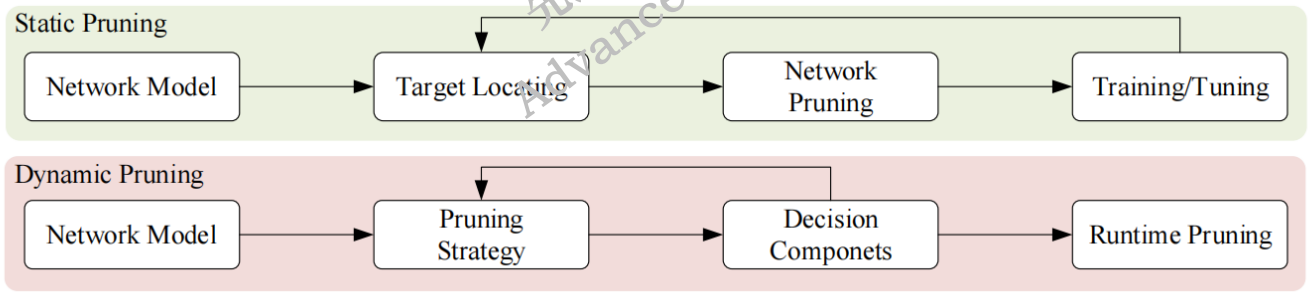
4.2、动态剪枝

网络中有一些奇怪的权重，他们在某些迭代中作用不大，但在其他的迭代却很重要。动态剪枝就是通过动态的恢复权重来得到更好的网络性能。动态剪枝在运行时才决定哪些层、通道或神经元不会参与进一步的活动。



4.2、动态剪枝

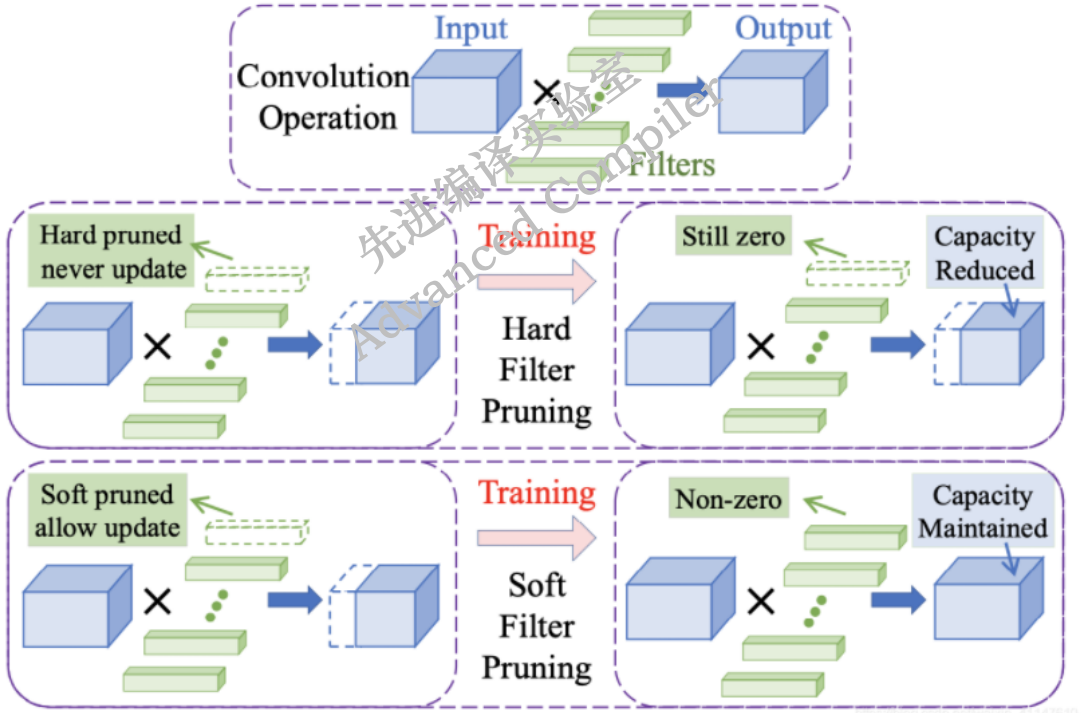
动态剪枝也存在一些问题：
1：之前有方法通过强化学习来实现动态剪枝，但在训练过程中要消耗非常多的运算资源。
2：很多动态剪枝的方法都是通过强化学习的方式来实现的，但是"阀门的开关"，是不可微的，也就是说，梯度下降法在这里是用不了的。
3：存储成本高，不适用于资源有限的边缘设备。



5.1、硬剪枝

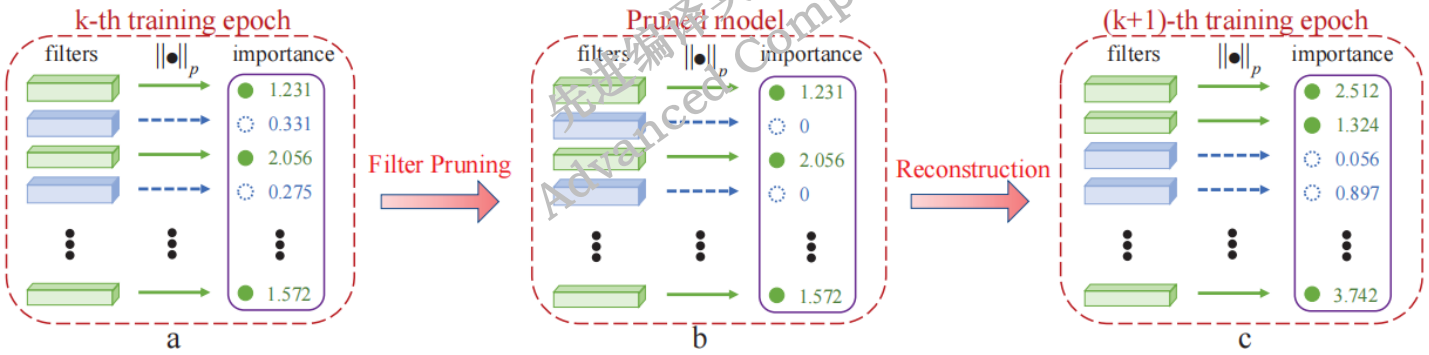
在每个epoch后会将卷积核直接剪掉，被剪掉的卷积核在下一个epoch中不会再出现。
存在的问题：

1）模型性能降低；
2）依赖预先训练的模型。



5.2、软剪枝

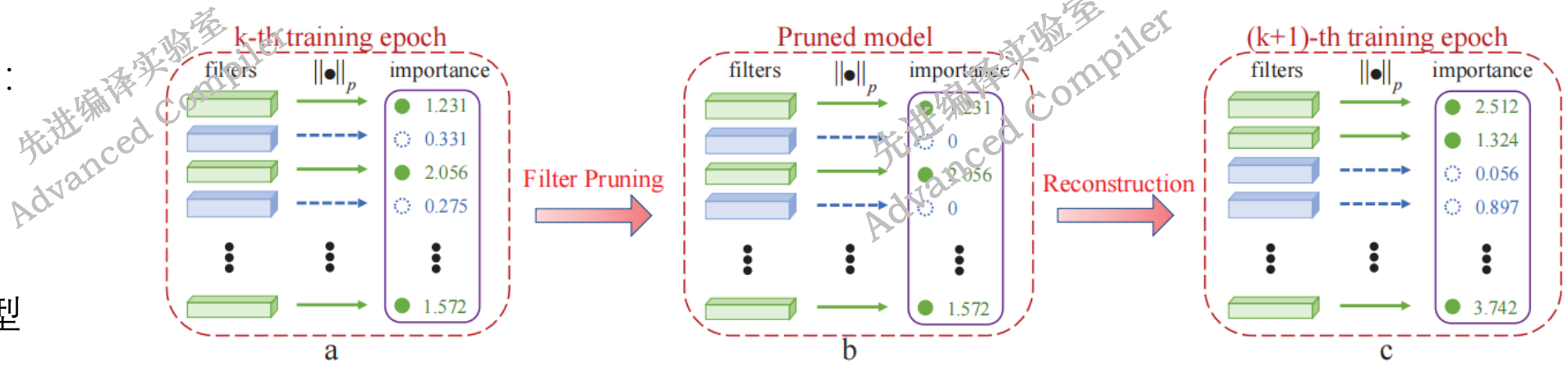
相比较硬剪枝，软剪枝剪枝后进行训练时，上一个epoch中被剪掉的卷积核在当前epoch训练时仍参与迭代，只是将其参数置为0，因此那些卷积核不会被直接丢弃，在所有epoch循环结束后进行权重修剪。



5.2、软剪枝

一般有四个步骤：

1：滤波器选择
2：滤波器剪枝
3：重建
4：获得紧凑模型



感谢大家聆听

参考：

《Soft Filter Pruning for Accelerating Deep Convolutional Neural Networks.》

https://www.zhihu.com/zvideo/1601174809257484288

深度学习模型压缩方法（三）量化

嘉宾：魏铭康

目录

- 量化概念
- 量化方式
- 校准方法

量化概念

模型量化是指将神经网络模型中的连续取值的权重或激活值近似为有限多个离散值的过程。

优势：

- 压缩参数
- 提升速度
- 降低内存占用

劣势：

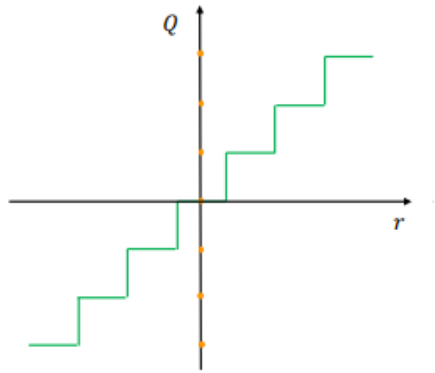
- 模型精度下降

r：浮点值

Q：整型值

量化：$Q = clamp(round\left(\frac{r}{s}\right))$

反量化：$r = s * Q$



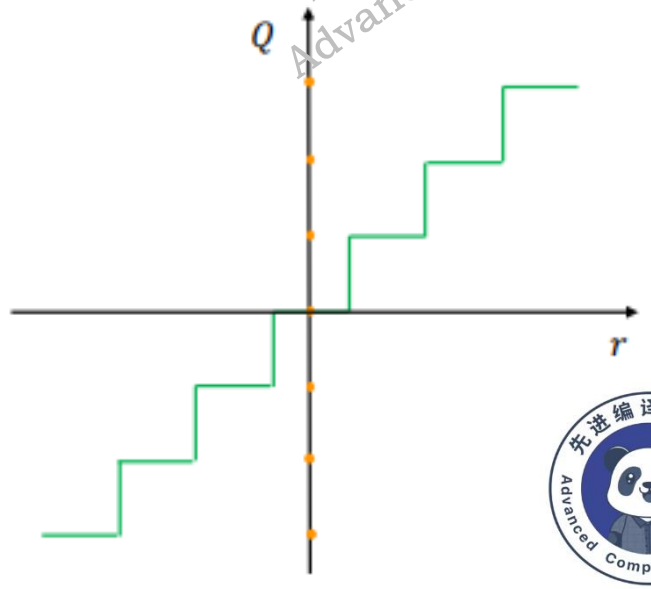
量化概念-量化分类

线性量化与非线性量化

线性量化

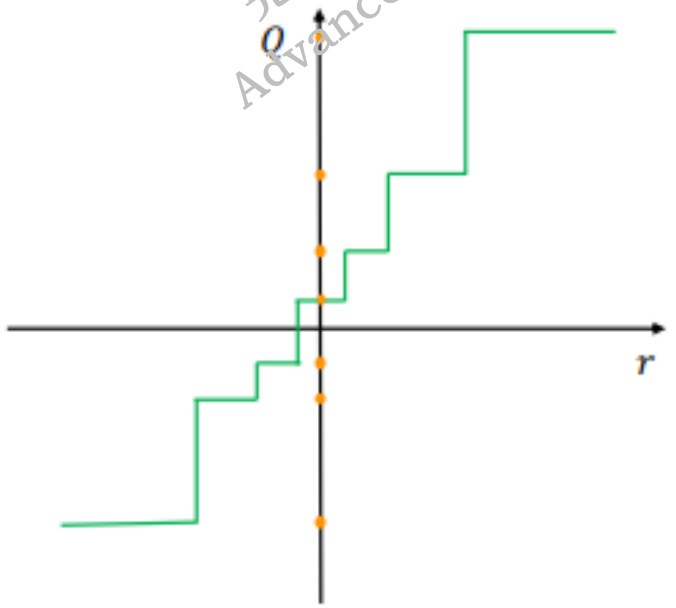
量化：$Q = clamp(round\left(\frac{r}{s}\right))$

反量化：$\tilde{r} = s * Q$



非线性量化

$Q = X_i, if\ r \in [\Delta_i, \Delta_{i+1})$



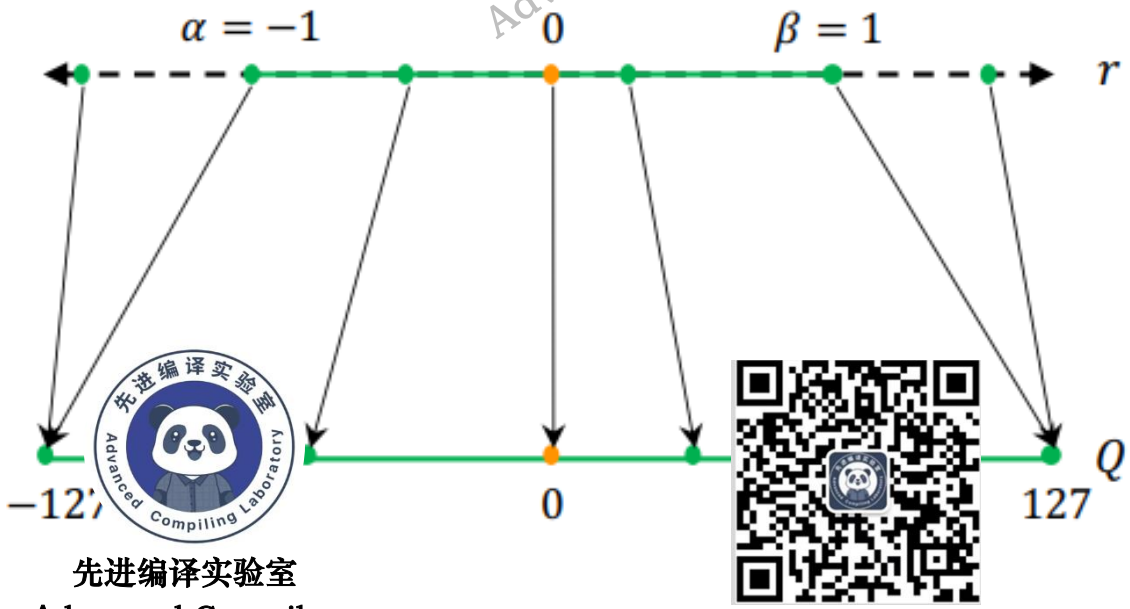
量化概念-量化分类

对称量化与非对称量化

对称量化

量化：$Q = clamp(round\left(\frac{r}{s}\right))$

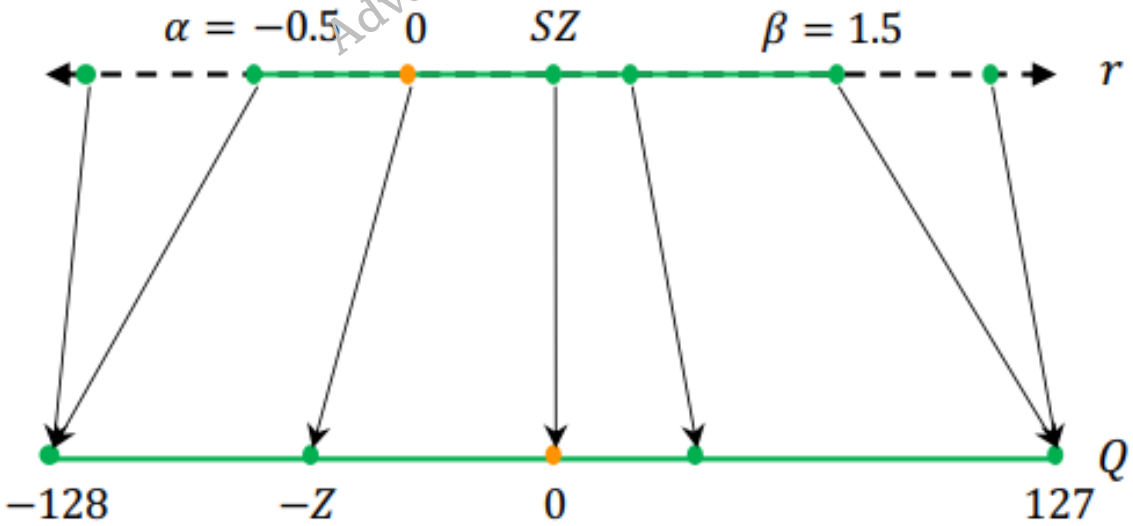
反量化：$\tilde{r} = s * Q$



非对称量化

量化：$Q = clamp(round\left(\frac{r}{s}\right) - Z)$

反量化：$\tilde{r} = s * (Q + Z)$



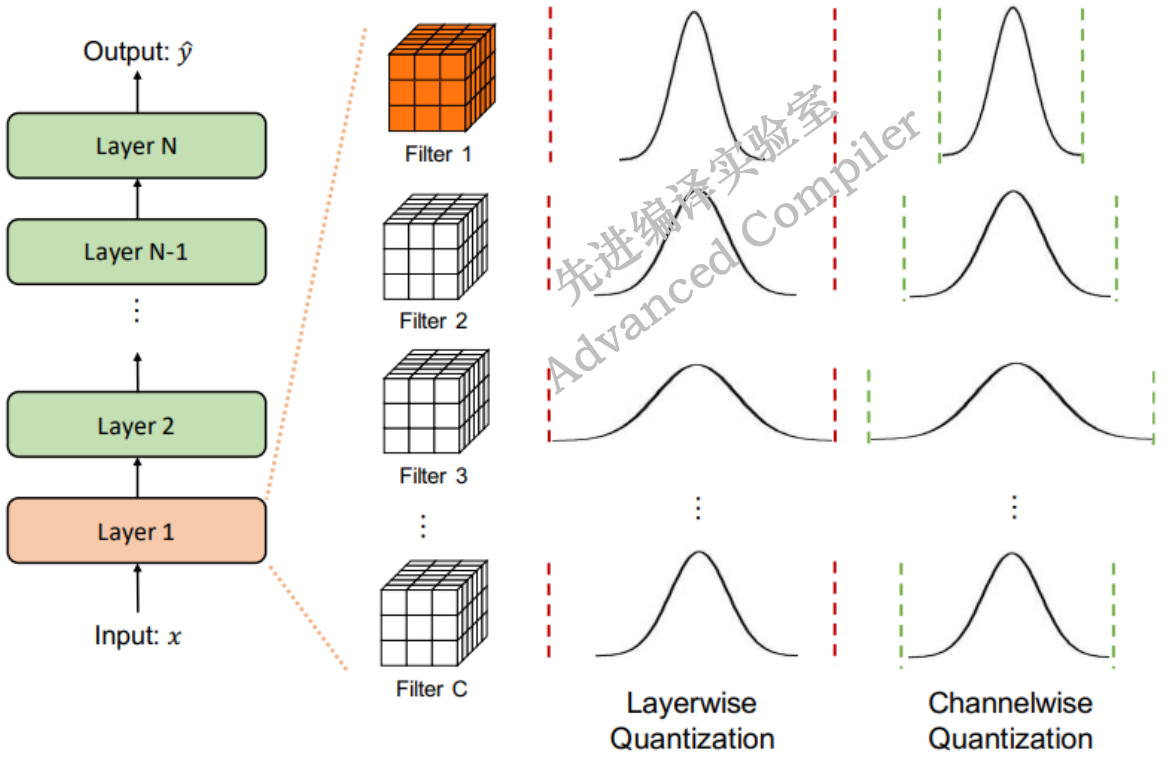
量化概念-量化分类

量化粒度

- 逐层量化
- 逐通道量化

量化位宽

- 统一精度
- 混合精度



先进编译实验室
Advanced Compiler

量化方式-训练后量化

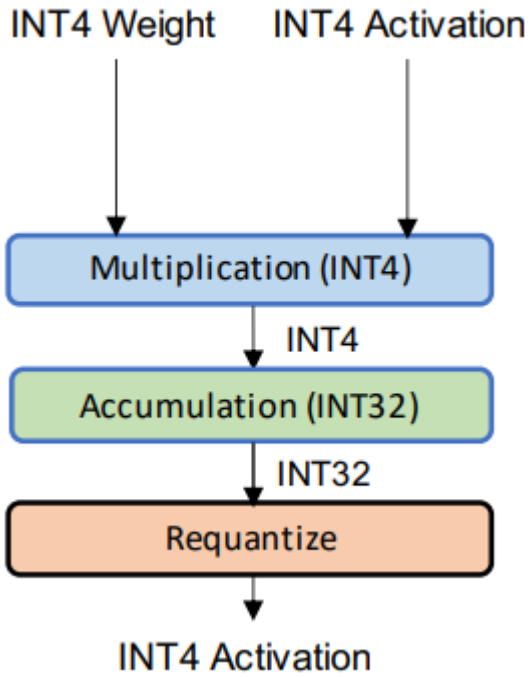
权重量化

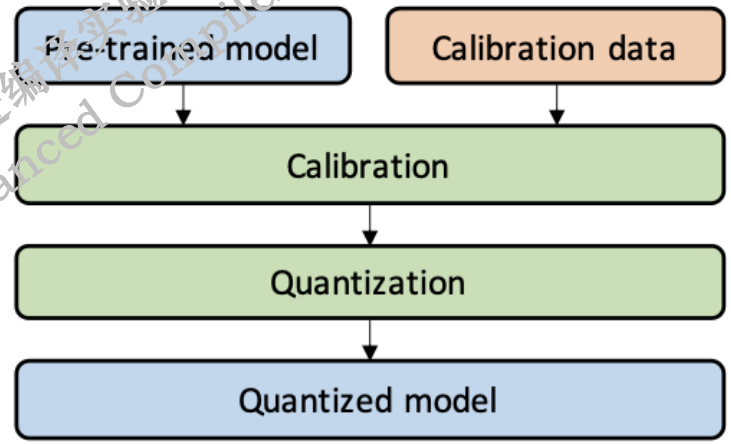
量化模型的权重，仅压缩模型大小，推理时先将权重反量化为浮点值

全量化

静态量化：离线计算权重与激活的量化参数

动态量化：推理时动态计算激活的量化参数

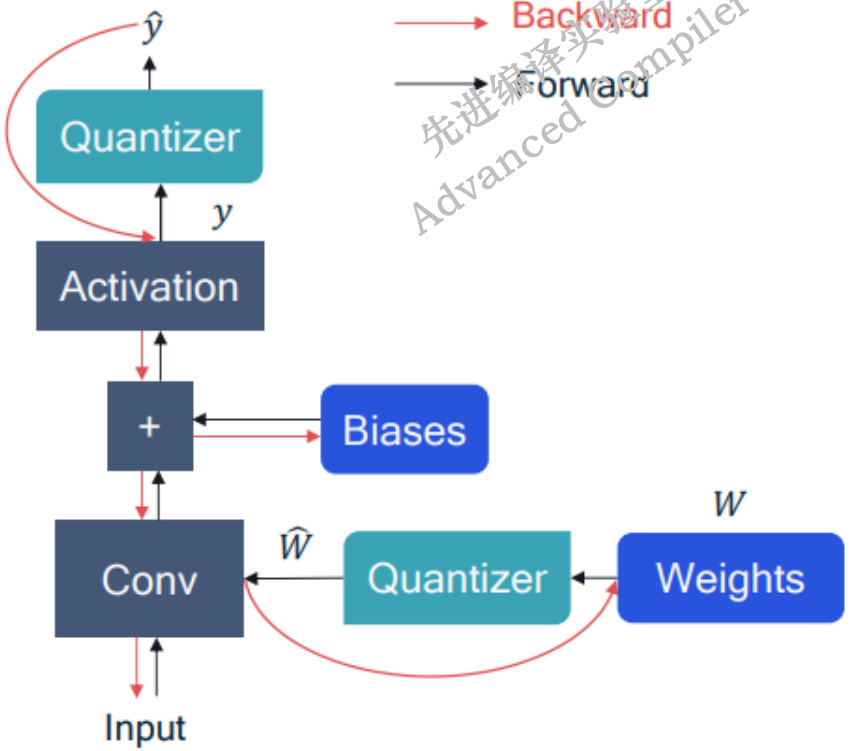


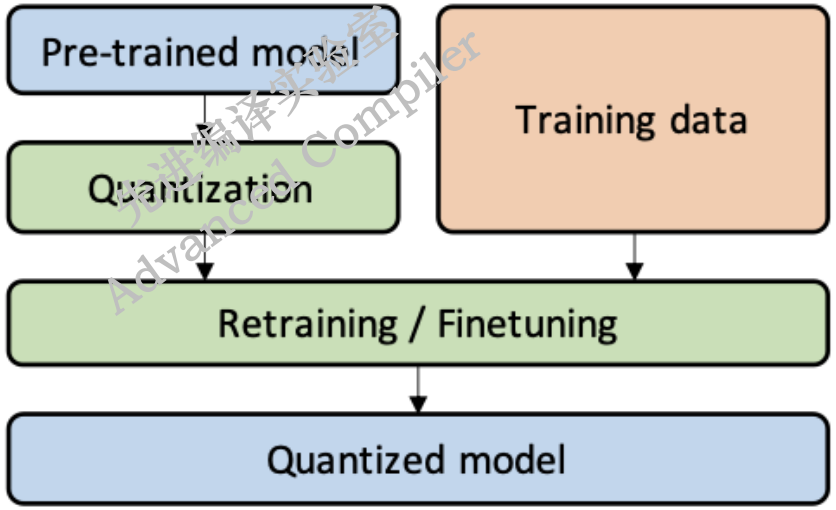


量化方式-量化感知训练

通过训练调整量化参数

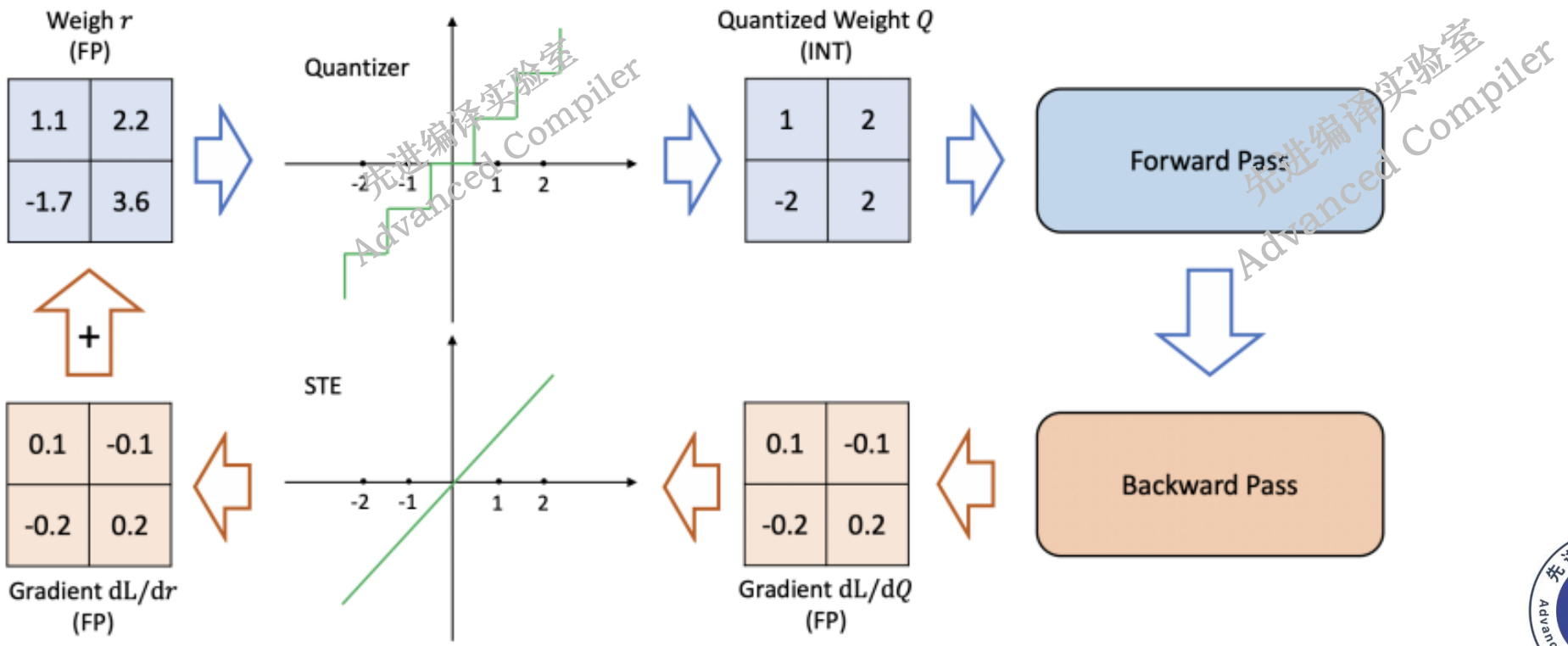
Quantizer: $\tilde{r} = s * clamp(round\left(\frac{r}{s}\right))$





量化方式-量化感知训练

straight-through estimator（STE）



先进编译实验室
Advanced Compiler

校准方法

$$Q = clamp(round\left(\frac{r}{s}\right)) \qquad s = \frac{threshold}{2^{b-1}-1}$$

- global
- max
- percentile
- mse
- KL-divergence

校准方法

$$Q = clamp(round\left(\frac{r}{s}\right)) \qquad s = \frac{threshold}{2^{b-1}-1}$$

- ✓ global
- max
- percentile
- mse
- KL-divergence

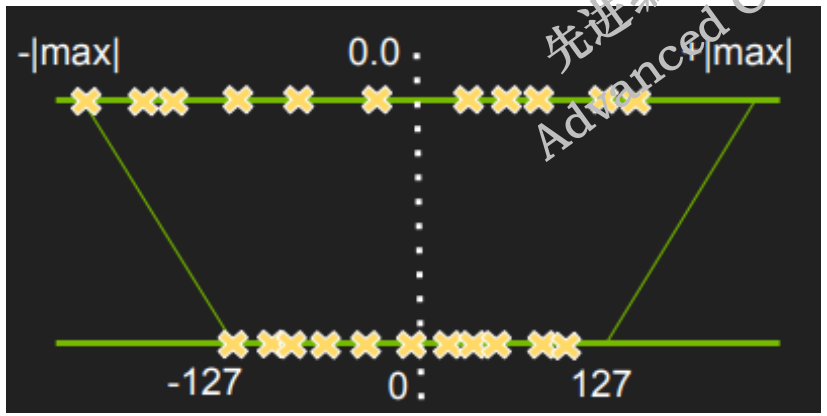
指定全局的threshold

校准方法

$$Q = clamp(round\left(\frac{r}{s}\right)) \qquad s = \frac{threshold}{2^{b-1}-1}$$

- global
- ✓ max
- percentile
- mse
- KL-divergence

$$threshold = \max(|r|)$$



先进编译实验室
Advanced Compiler

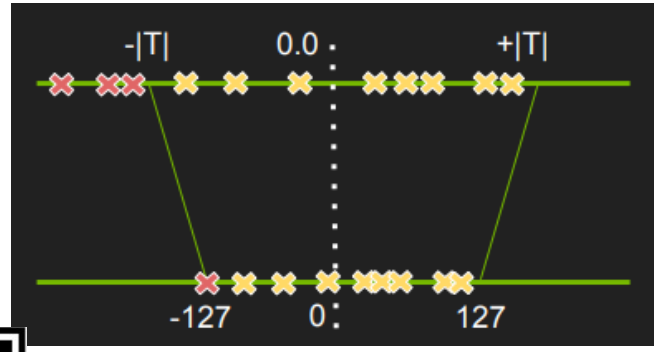
校准方法

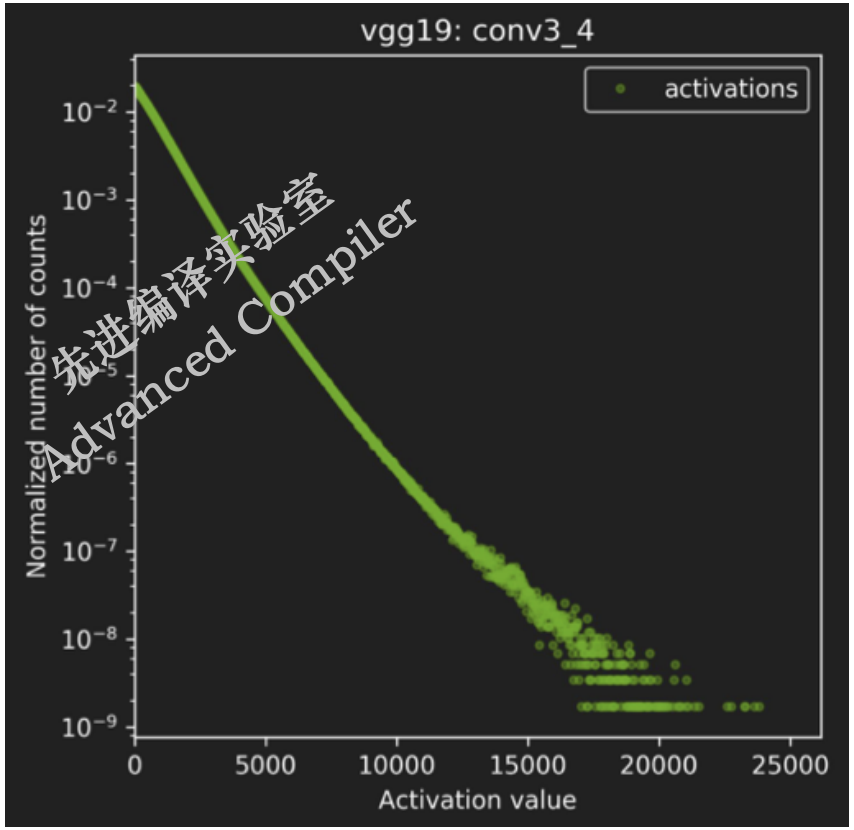
$$Q = clamp(round\left(\frac{r}{s}\right))$$

$$s = \frac{threshold}{2^{b-1} - 1}$$

- global
- max
- ✓ percentile
- mse
- KL-divergence

以分位数作为threshold





校准方法

$$Q = clamp(round\left(\frac{r}{s}\right)) \qquad s = \frac{threshold}{2^{b-1}-1}$$

- global
- max
- percentile
- ✓ mse
- KL-divergence

测试不同threshold下模拟量化值与原始值之间的均方误差，选择使均方误差最小的值作为最终threshold

校准方法

$$Q = clamp(round\left(\frac{r}{s}\right)) \qquad s = \frac{threshold}{2^{b-1}-1}$$

- global
- max
- percentile
- mse
- ✓ KL-divergence

测试不同threshold下模拟量化值的分布与原始值分布之间的KL散度，选择使KL散度最小的值作为最终threshold

参考内容

[1] Gholami A, Kim S, Dong Z, et al. A survey of quantization methods for efficient neural network inference[J]. arXiv preprint arXiv:2103.13630, 2021.
[2] Nagel M, Fournarakis M, Amjad R A, et al. A white paper on neural network quantization[J]. arXiv preprint arXiv:2106.08295, 2021.
[3] Krishnamoorthi R. Quantizing deep convolutional networks for efficient inference: A whitepaper[J]. arXiv preprint arXiv:1806.08342, 2018.
[4] Jacob B, Kligys S, Chen B, et al. Quantization and training of neural networks for efficient integer-arithmetic-only inference[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 2704-2713.
[5] https://on-demand.gputechconf.com/gtc/2017/presentation/s7310-8-bit-inference-with-tensorrt.pdf
[6] https://zhuanlan.zhihu.com/p/548174416
[7] https://www.bilibili.com/video/BV1fB4y1m7fJ