## 视频课程

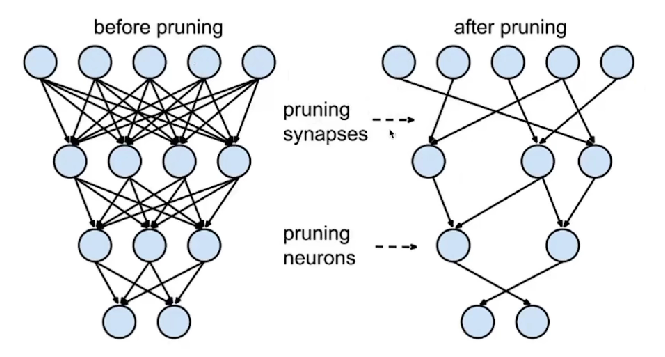
模型压缩

①什么是模型压缩：模型压缩是指在保持模型性能的前提下，减少机器学习模型的参数数量或复杂度，以减少模型的存储空间、加快推理速度、降低内存使用等目的。在正常使用模型过程中，部署到本地的模型过大会导致响应延迟变高，因此一般要进行模型压缩。

②模型压缩之所以可行，是因为许多机器学习模型，尤其是深度神经网络，往往存在冗余和过度参数化现象。在训练过程中，为了保证模型的泛化能力，通常会引入一些额外的复杂性。经过训练后，这些复杂性往往可以被简化，而不影响模型的推理能力。大型模型中的很多参数对最终的预测结果贡献很小。

③常见的模型压缩方法;

剪枝：在训练结束后，删除不重要的权重或神经元，通常基于一些重要性度量，如小权重的阈值。



量化：减少模型中参数和计算的精度，以降低存储空间、加快计算速度，并减少能耗。量化通常在不显著降低模型性能的前提下，将浮点数权重和激活值转换为低精度的整数表示。

知识蒸馏：通过将一个较大、性能较好的模型（通常称为“教师模型”）的知识传递给一个较小的模型（通常称为“学生模型”），使得学生模型在保持较高性能的同时，具有更小的参数量和计算复杂度。这个方法广泛应用于在需要高效推理的场景中，比如移动设备或嵌入式系统。

低秩分解：通过将神经网络中的权重矩阵分解为多个低秩矩阵的乘积，以减少模型的参数数量和计算复杂度。低秩分解的原理基于这样一个事实：在实际应用中，许多高维矩阵实际上是低秩的或接近低秩的。这意味着矩阵中的信息可以通过较少的线性组合来表示，从而减少计算量。

轻量化网络：在资源受限的环境中部署，设计的高效且小型的神经网络模型。这些网络通过减少参数量、计算复杂度和内存占用，在保持相对较好性能的前提下，提高推理速度和效率。

网络结构搜索：自动化设计神经网络架构的方法，通过使用搜索算法在庞大的候选网络架构空间中寻找最优架构。NAS 的目的是取代人工设计网络结构的繁琐过程，找到性能更佳、计算更高效的模型。

模型压缩

①知识类型：

输出特征知识：将教师模型在数据集上的输出传递给学生模型。这种知识的传递主要依赖于最终的分类结果（如软标签使用比较多），而帮助学生模型学习到更有效的表示方式。

中间特征知识：将教师模型在中间层生成的特征信息传递给学生模型。相比于仅传递最终的输出结果，中间特征知识能够让学生模型学习到更多层次的表示信息，提升其整体性能。

关系特征知识：教师模型中关于输入样本之间、或者输出类别之间关系的信息。这种知识关注的是输入数据样本之间的相互关联性，以及这些关联性如何影响模型的决策。

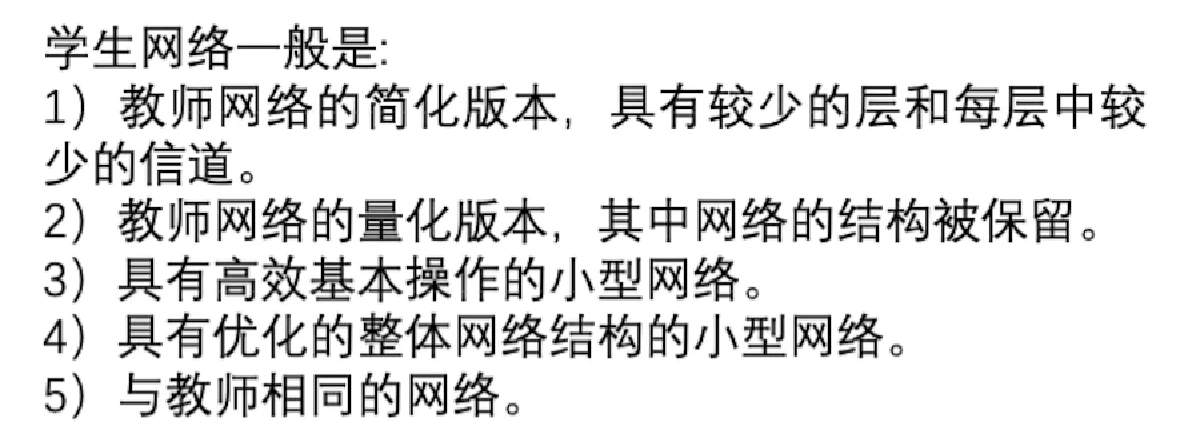
②蒸馏机制：

离线蒸馏：教师模型通常是一个已经经过充分训练的大规模模型，它在某个数据集或任务上表现优异。这个教师模型在蒸馏过程开始之前已经完全训练好，并且不会在学生模型的训练过程中更新。

在线蒸馏：教师模型和学生模型同时进行训练的知识蒸馏方法。与离线蒸馏不同，在线蒸馏过程中，教师模型的知识是动态更新的，这使得蒸馏过程更加灵活和适应性强。在线蒸馏的目标是在训练过程中实时地将教师模型的知识传递给学生模型，以提升学生模型的性能。

自蒸馏：学生模型通过学习自身的一些知识进行蒸馏，而无需外部的教师模型。可以看作是知识蒸馏的简化版本，通常无需增加额外的计算资源或模型复杂度。

③教师网络和学生网络：



④多教师蒸馏：一个学生模型从多个教师模型中学习知识。

交叉模式蒸馏：将知识从一种模式（如图像、文本、音频等）传递到另一种模式的知识蒸馏技术。这种方法特别适用于多模态学习任务。

基于图形的蒸馏：一种利用图结构来传递知识的蒸馏方法，广泛应用于图神经网络中。这种方法通过图结构捕捉节点之间的关系和连接模式，以此来进行知识的传递和学习。

剪枝

①过参数化：指模型的参数数量远多于训练数据的样本数量或任务实际所需的参数量。因此可以进行剪枝。

②剪枝的方法：细粒度剪枝、向量剪枝、核剪枝、滤波器剪枝等。

③非结构化剪枝：一种细粒度的剪枝方法，它直接移除神经网络中的单个权重或连接，而不考虑它们在网络结构中的位置。

结构化剪枝：一种粗粒度的剪枝方法，删除整个神经网络的结构性组件，如卷积核、神经元、通道或整个滤波器。剪枝后，模型的结构保持规则且易于优化。

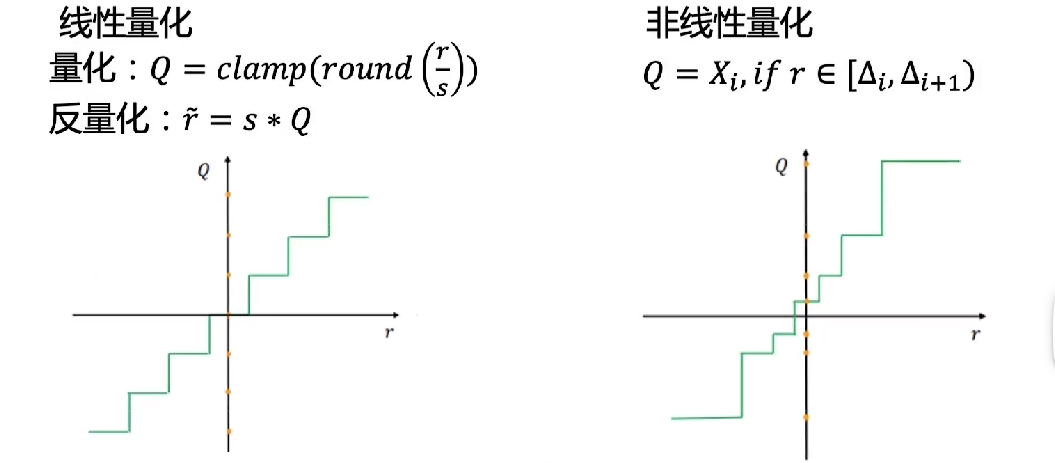
④在结构化剪枝中，分硬剪枝和软剪枝。

硬剪枝是一种直接且不可逆的剪枝方法。在剪枝过程中，一旦某些权重或结构被判定为不重要，它们就会被永久性地删除（置零或移除），而不会被恢复。

软剪枝在剪枝过程中，权重或结构并不会被永久删除，而是暂时减弱（如置零）或者降低其影响力。在后续的训练中，这些权重或结构仍然有机会恢复。

量化

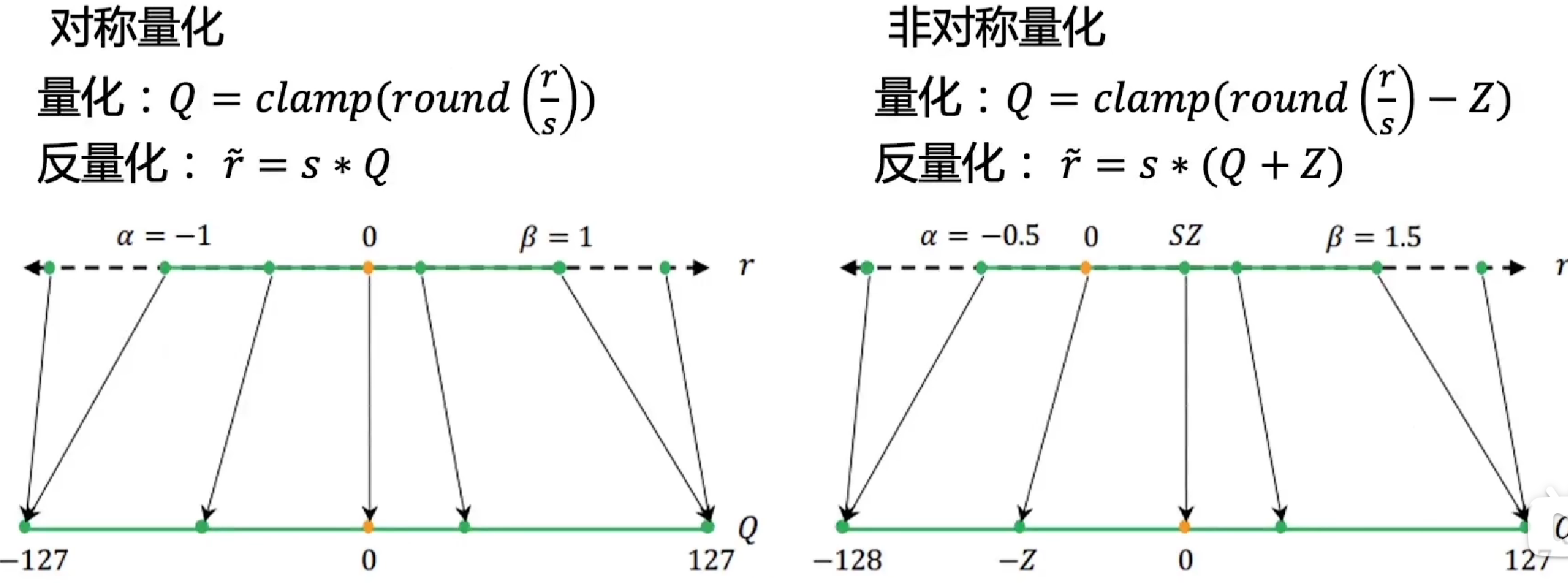
①线性量化和非线性量化：



线性：将浮点数均匀地映射到固定范围内的低比特表示；

非线性：以非均匀步长对数据进行量化的方法。即，数值范围内的量化区间不再是等间隔的，而是根据某种策略动态调整。

②对称量化和非对称量化：

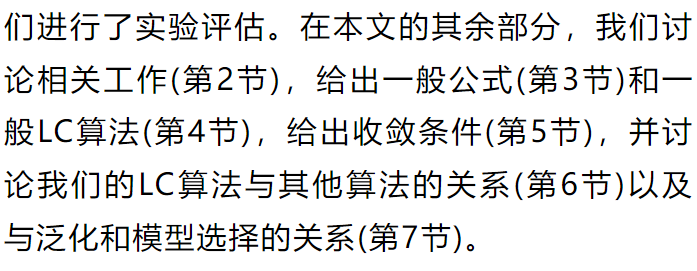


③训练后量化：是一种在模型训练完成后，对模型的权重和激活值进行量化，以减少模型大小和计算复杂度的技术。它不需要重新训练模型，因此应用方便，常用于模型部署阶段，特别是在资源受限的设备上。

④在模型训练过程中引入量化操作的技术，它允许模型在训练期间“感知”到量化所带来的影响，从而学习到在量化后的低精度环境中仍然能保持高性能的参数。这种方法通过将量化引入训练流程，显著减少了量化对模型精度的影响，是在精度敏感的应用中使用较为广泛的量化技术。

⑤量化阈值：在模型量化过程中用于确定浮点数数据映射到离散整数值范围的一个关键参数。它主要决定了在量化时哪些值将被截断或被映射到固定范围内。这一阈值的设置直接影响到量化模型的精度和性能。

论文



①获得小模型的方法：

1.直接学习：不管参考是什么，找到损耗最好的小模型。即直接训练小模型使损失最小化。

2.直接压缩：找到与参考模型参数最接近的近似值。这迫使h和f是同一类型的模型。直接压缩可以简单地通过w的(无损或有损)压缩来完成，但它通常不会是最优的损失。

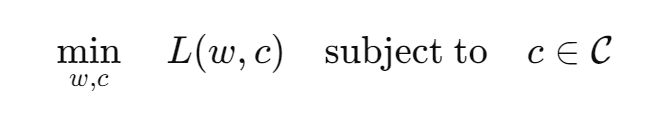
3.模型压缩作为约束优化:论文提出的方法。通过约束由低维参数化w =∆(Θ)构造的权重w，迫使h和f为同一类型的模型，但h必须优化损失L。

4.师生：参考模型f“教导”学生模型h。我们可以让小模型h成为一个完全不同模型的重新参数化版本。较小的模型可以是与集成成员相同的类型，也可以是完全不同类型的模型(例如神经网络)。基本思想是，在一个标记的训练集上训练了大的集合，使用这个集合(“老师”)来标记一个更大的、未标记的数据集(可能在手头的任务中可用，或者可能通过某种形式的抽样生成)。然后使用这个较大的标记数据集来训练较小的模型(“学生”)。

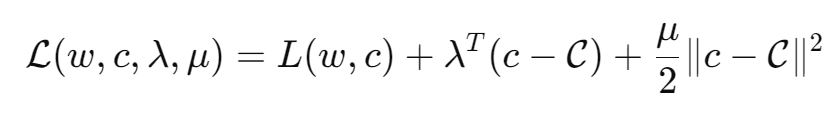
②无损压缩：有多种形式，如霍夫曼编码、算术编码或游程编码。

③**学习压缩(LC)**算法：用于神经网络模型压缩的优化算法，旨在通过交替优化来学习最佳的压缩模型。

增广拉格朗日法的作用是处理压缩过程中带有的约束条件。假设我们的目标是最小化损失函数，同时满足压缩约束条件，这可以表示为一个带有约束的优化问题：



利用增广拉格朗日法，我们构建了以下的拉格朗日函数：



**交替优化**：

* **学习阶段**：固定c和λ，通过最小化 L来优化模型参数w。
* **压缩阶段**：固定w和λ，通过最小化 L来优化压缩参数c，同时更新拉格朗日乘子λ。

LC算法的整体运行时间将由L步控制，就好像我们在长时间训练一个未压缩的模型一样。

在 LC 算法中，参数w和Θ会随着μ的增加而沿着一条路径进行优化，最终在μ→∞时达到约束条件并获得一个最优的压缩模型。但是在路径的起点，即μ→0+时，LC 算法实际上对应的是直接压缩（DC）方法。这是因为在μ很小时，损失函数的惩罚项非常小，优化问题主要由未压缩的模型参数w的损失函数主导。

④LC算法的收敛性：

**前提假设**：假设损失函数L和解压缩映射Δ(Θ)关于其自变量是连续可微的，且损失函数有下界。

**定理陈述**：对于给定的约束优化问题，使用二次罚函数 Q(w,Θ;μ)可以找到近似最小值 (wk,Θk)，在适当条件下，随着迭代次数k趋向无穷大，(wk,Θk)将收敛到问题的KKT点

⑤LC算法的优点：

LC算法通过迭代执行L步和C步来优化模型。L步关注于最小化损失函数，而C步则严格按照压缩技术的约束来执行。这种方法不仅可以确保模型在每次压缩后都朝着最优解方向前进，而且能够避免性能的大幅下降，同时减少了再训练的需要。

⑥压缩模型的一个重要作用是防止过拟合。通过压缩模型，我们可以减少模型的复杂性，同时尽量保持与原始模型相似的损失函数值。常见的策略是先训练一个足够大的参考模型，然后尽可能地压缩它。这不仅有助于防止过拟合，还能有效简化模型选择过程，从而找到最小的、满足应用需求的模型。