深度学习模型的压缩原理

黄京

Nov 29, 2026

深度学习模型在图像分类、目标检测和自然语言处理等任务上取得了显著的成功，但这些模型往往包含数亿甚至数十亿参数。端侧设备如智能手机、嵌入式系统以及边缘计算节点普遍面临算力、内存和功耗的限制，无法直接承载大型模型。实时性要求模型在毫秒级完成推理，隐私性要求数据在本地处理而非上传云端，部署成本则直接影响产品落地可行性。这些因素共同催生了对模型压缩技术的迫切需求。

模型压缩并非孤立存在，而是与知识蒸馏、量化、剪枝以及神经架构搜索等效率优化手段相互交织。知识蒸馏通过教师模型指导学生模型学习，量化将浮点权重映射为低比特表示，剪枝删除冗余连接，神经架构搜索则在搜索空间内自动发现高效结构。本文聚焦于模型压缩的核心原理与实践路径，依次展开剪枝、量化、知识蒸馏、低秩分解与混合策略的理论基础和工程细节。

1　模型压缩的四大核心思路

模型中存在大量参数冗余，主要源于过参数化现象。研究表明，深度网络的权重矩阵往往呈现低秩特性，而激活值在训练后也表现出显著的稀疏性。这些特性为压缩提供了理论依据。压缩的数学目标可以表述为在满足精度损失不超过给定阈值的前提下最小化模型存储空间，即最小化模型尺寸，同时约束准确率下降不超过预设的容忍范围。

四大方法在实践中并非完全独立，而是形成互补关系。剪枝直接移除不重要的权重，量化降低每个权重的比特位宽，知识蒸馏利用软标签传递知识，低秩分解则通过矩阵近似降低参数量。理解它们之间的协同效应，有助于设计更高效的压缩流水线。

2　剪枝

非结构化剪枝以单个权重为粒度进行删除，判据通常包括幅值大小、梯度信息或海森矩阵对角元素。幅值剪枝简单直接，假设较小权重的贡献可以忽略；基于梯度的判据考虑了权重对损失函数的影响；海森矩阵方法则试图估计剪枝后的损失变化。细粒度稀疏矩阵常用 CSR 或 CSC 格式存储，这些格式通过记录非零元素的值、行索引和列指针来节省空间，但实际推理时仍需专用稀疏计算库才能发挥加速效果。

结构化剪枝则以通道、滤波器或组为单位进行裁剪，更适合通用硬件加速。BN 缩放因子 L1 正则化方法在训练过程中对 BatchNorm 层的缩放参数施加 L1 惩罚，促使部分通道的缩放因子趋近于零，从而在不显著损失精度的前提下安全移除对应通道。Taylor 展开方法通过一阶近似评估通道对损失的贡献，避免了直接计算海森矩阵的高昂代价。

动态或迭代剪枝在训练过程中周期性地评估并移除权重，同时通过恢复训练让模型重新适应稀疏结构。硬件感知剪枝则在判据中引入 MACs 计数和访存模式分析，确保剪枝后的模型在目标硬件上真正获得加速，而非仅在理

论 FLOPs 上降低。

3 量化

定点量化与浮点表示的核心差异在于线性映射与零点偏移。浮点数通过指数位动态调整范围，而定点数则固定小数点位置。为了将浮点值映射到整数域，需要确定缩放因子和零点，使得浮点零值能够精确映射到整数零值，避免累积舍入误差。

训练后量化直接将已训练好的浮点模型转换为定点表示，无需重新训练，但可能因量化误差导致精度下降。量化感知训练则在训练过程中模拟量化操作，使模型能够学习适应量化噪声，通常能获得更高的最终精度。低比特极致方案包括二值网络和三值网络，二值网络将权重限制为正负一，三值网络则允许零值存在。对称量化假设权重分布关于零点对称，非对称量化则通过零点偏移处理偏态分布。

混合精度量化搜索旨在为不同层自动分配最优比特位宽，HAQ 和 EdMIPS 等方法利用强化学习或可微分优化在精度与模型尺寸之间寻找帕累托最优解。量化本身还具有正则化效应，低比特约束可以抑制过拟合，实践中常配合知识蒸馏或微调策略来补偿可能的精度损失。

4 知识蒸馏

经典响应蒸馏使用教师模型输出的软标签来训练学生模型，软标签携带了类别间的相似性信息，比硬标签提供更丰富的监督信号。中间层特征蒸馏则试图让学生网络的特征图与教师网络对应层保持一致，通常在选定的中间层添加 L2 或注意力损失。

关系蒸馏关注样本间的成对距离或角度关系，试图在学生模型中重构教师模型学到的样本相似性结构。对抗蒸馏引入判别器网络，迫使学生分布逼近教师分布，在线蒸馏则允许教师和学生在同一训练过程中相互提升。蒸馏在压缩流水线中通常作为后处理步骤，在剪枝或量化之后进行微调，以恢复因压缩而损失的精度。

5 其他轻量化与混合策略

低秩分解通过矩阵或张量近似降低参数量，CP 分解将高阶张量表示为若干秩一张量的和，Tucker 分解引入核心张量实现更灵活的近似，Tensor-Train 格式则适合处理高维参数。轻量化架构如 MobileNet 使用深度可分离卷积将标准卷积分解为逐深度卷积和逐点卷积，ShuffleNet 通过通道混洗实现跨组信息流动，EfficientNet 则通过复合缩放统一调整网络深度、宽度和分辨率。

神经架构搜索在预定义搜索空间内自动发现高效结构，Once-for-All 系列通过一次训练得到可弹性伸缩的超级网络，支持后续按需子网络提取。实际部署中，多方法协同往往能取得更好效果，例如先进行结构化剪枝降低 FLOPs，再应用量化降低位宽，最后通过蒸馏恢复精度。

6 评估、工具链与展望

评估指标不仅包括 Top-1 准确率，还需关注 FLOPs、实际延迟和能耗等硬件相关指标。移动端 benchmark 通常在特定芯片上测量单次推理耗时和功耗，以反映真实部署性能。主流框架如 PyTorch 的 Prune 模块提供结构化和非结构化剪枝接口，TensorFlow Lite 和 TensorRT 支持 INT8 量化推理，ONNX Runtime 提供跨平台优化执行，MQBench 则专注于量化感知训练的可复现实现。

未来方向包括可微分压缩，将剪枝掩码和量化参数纳入训练图进行端到端优化；动态推理根据输入难度自适应调整计算路径；联邦场景下的压缩则需在保护隐私的前提下实现模型协同更新。模型压缩本质上是在精度、速度与能耗之间进行权衡，没有一种方法能够同时在所有指标上达到最优，工程实践需要根据具体应用场景和硬件约束做出合理选择。