华中科技大学

2023

卷积神经网络模型压缩综述报告

计算机视觉结课报告

专　　业：计算机科学与技术

班　　级：计科 2008 班

学　　号：U202015550

姓　　名：刘方兴

完成日期：2023 年 5 月 3 日

目 录

一、引言

卷积神经网络（CNN）由于具有良好的特征提取能力和泛化能力，在图像处理、目标跟踪与检测、自然语言处理、场景分类、人脸识别、音频检索、医疗诊断等诸多领域都取得了巨大的成功[1]。然而，随着网络深度的增加和参数数量的增长，CNN 模型的复杂性也在增加。这导致了计算资源和存储需求的增加、推理速度的降低以及模型过拟合等问题。为了解决这些问题，研究人员已经提出了许多模型压缩技术，包括权重剪枝、低秩分解、知识蒸馏、量化[2-3]以及混合型技术[4]等。本综述报告的目的是梳理这些技术、总结它们各自的优缺点并分析它们在 CNN 模型压缩中的应用和发展趋势。

二、未经过模型压缩的模型存在的问题

随着卷积神经网络在各种计算机视觉任务中取得优异表现，模型的规模也随之增大。这导致了以下几个问题[3,5,6]：

- 计算资源和存储需求增大：大型神经网络模型通常具有大量的参数和层，这导致模型会占用大量的存储空间且具有较高的计算复杂性。因此，大型 CNN 模型需要大量的计算资源和存储空间，这限制了它们在资源受限的设备（如移动设备和嵌入式系统）上的应用。
- 推理速度慢：大型 CNN 模型的推理速度较慢，在对实时性有要求的应用场景中，这可能导致模型性能无法满足应用的需要。
- 能耗提高：由于计算复杂性和存储需求的增加，大型 CNN 模型会消耗更多的能量。这可能导致模型无法应用在功耗受限的设备和应用场景上，如某些移动设备和物联网设备上。
- 过拟合问题：当模型具有大量参数时，很容易发生过拟合现象。

因此，在许多情况下，采用模型压缩技术对神经网络模型进行优化是非常有必要的。

三、卷积神经网络模型压缩技术梳理

1. 网络剪枝与稀疏化

网络剪枝与稀疏化是一种可以稳定地优化并调整网络结构并以较小精度损失的代价压缩网络规模的网络结构优化方法。根据卷积神经网络训练阶段的不同，网络剪枝与稀疏化方法主要包含训练中稀疏约束与训练后剪枝两个大类。通过在优化函数添加稀疏性约束，可以诱导网络结构趋于稀疏，同时可以使得网络稀疏化、精简化。而网络剪枝方法则使得精简后的小型网络继承了原始网络的有用知识，并且具有接近甚至超过原有网络的性能表现，此类方法目前已取得了一系列卓有成效的成果[1]。

[7]中指出，我们有可能获得剩余权重小于 10% 的高度稀疏模型。该模型与之前的与密集模型相比，其准确性不会降低，偏差也不会显著增加。然而，在更高的稀疏度下，修剪后的模型也会在其输出中表现出更高的不确定性。网络剪枝与稀疏化方法能够获得较大的压缩比，同时对于网络精度的影响较小，在需要模型稳定运行的场景下较为适用[1]。其中，权重剪枝是通过移除模型中较不重要的权重参数来降低模型复杂度的技术。结构剪枝则是删除整个神经元或层来实现模型压缩[7]。

以下我将分析几种常见的剪枝方法，并结合论文中的数据描述具体效果：

- **阈值剪枝**

该方法包括三个步骤[8]：首先，训练网络以学习所有连接的初始权重；其次，删除所有权重低于某个阈值的连接，从而得到一个稀疏网络；最后，重新训练剩余的稀疏网络以学习重要连接的最终权重。

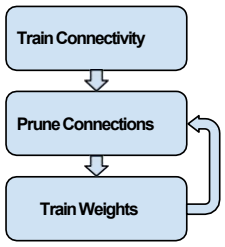

图 1 阈值剪枝三步训练流程

这种方法简单易实现，但可能削减掉一些重要的参数。优点是实施速度快且易于理解。缺点是剪枝粒度较粗，可能会导致模型性能下降。

[8]中的实验结果表明，使用该方法可以将神经网络中的连接数减少 9 倍至 13 倍，而不影响其准确性。作者还在 AlexNet 和 VGGNet 上进行了实验，结果表明了完全连接层和卷积层都是可以被修剪的。实验结果表明，通过删除冗余连接并微调重要连接的方法，可以在保持其准确性的同时显著降低神经网络的存储和计算要求。

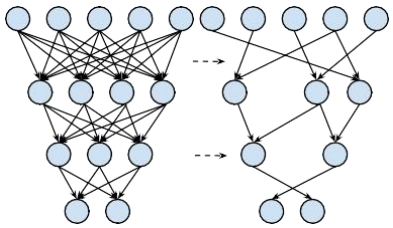

图 2 修剪前后的突触和神经元

这种方法几乎可以应用于任何类型的神经网络结构，并且可以在嵌入式系统中实现实时图像处理。通过减少神经网络中的连接数，可以降低图像处理所需的内存容量和带宽要求，从而使其更易于部署在移动系统上。此外，该方法还可应用于其他具有有限计算资源的嵌入式系统，例如可穿戴设备和物联网设备等。

- **迭代剪枝**

该方法通过 LASSO 回归和最小二乘重构的迭代两步算法有效地剪枝每一层[9]

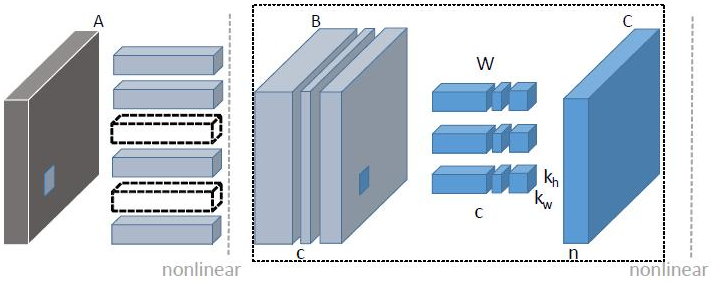

图 3 加速卷积层的通道修剪方法

（c；n：特征图 B 和 C 的通道数，kh * kw：卷积核大小）

其中，剪枝的目标是在减少特征图 B 的通道数的同时，最小化特征图 C 的重建误差。采取的优化算法在虚线框内执行，不涉及非线性。此外，虽然论文中采用的算法没有直接优化前一层中的相应滤波器（由虚线滤波器标记），但它们也可以被删除。在训练过程中，通过不断比较各个参数对模型性能的影响，移除最

不重要的参数。这种方法它保持了原有的模型架构，没有引入额外的层，可以在一定程度上保留重要参数，但计算量较大。

因此，这种方法的优点是能够较好地保留重要参数，从而减少性能损失。缺点是计算量较大，可能需要更长的训练时间。根据[9]中的实验结果，他们的修剪后的 VGG-16 模型在实现了 5 倍加速比的同时仅增加了 0.3%的误差。

- **Lottery Ticket 假设**

[10]中提到的这种方法基于实际观察到的现象，即在大型神经网络中存在一个较小的子网络，其权重具有良好的初始值，能够在较短的训练时间内达到与原始网络相似的性能。这种方法需要在训练过程中找到这个“幸运子网络”。

其优点是在找到合适的子网络后，训练时间和计算资源消耗相对较少。缺点是寻找合适的子网络可能需要较复杂的搜索过程。根据[10]中的实验结果，Lottery Ticket 方法在 VGG-19 网络上保持模型在测试集上的准确率几乎没有发生变化的同时实现了约 4 倍的压缩比。

- **系统性偏差分析**

[7]中还介绍了一项关于稀疏神经网络的系统性偏差分析的研究。研究表明，尽管稀疏模型在精度和推理速度方面可以超越密集模型，但它们也可能存在系统性偏差，这可能会损害其在实际应用中的可用性和公正性。为了测量这些偏差，文中提出了一组度量标准，包括“偏置放大”、“阈值校准偏差”、“不确定性和校准”以及“标签相互关系”。

文中在 CelebA、iWildCam 和 Animals with Attributes2 等数据集数据集上的评估结果表明，论文中提出的标准是有效的。这种标准可以帮助研究人员更好地理解修剪对模型性能和偏差的影响，并采取相应的措施来减轻这些影响。同时，[7]中的实验结果也表明了稀疏模型往往会放大数据集中存在的偏差，并且它们的预测结果可能会对受保护群体进行歧视。因此，我们在开发稀疏模型时需要更加谨慎地进行训练和评估，以确保其在实际应用中能够公正和准确地执行任务。

2. 低秩分解

卷积操作在卷积神经网络中占据了大部分计算资源，因此减少卷积层可以提高压缩率和整体速度。卷积核可以看作是一个 3D 张量。基于张量分解的思想源于 3D 张量中存在的结构稀疏性。对于全连接层，它可以被视为一个 2D 矩阵（或

3D 张量），低秩性也有助于压缩。低秩分解是一种降低张量（例如卷积核）秩的技术，通过将高维张量分解成较低维度的张量积来实现模型压缩[2]。随着卷积神经网络逐渐发展为更深、更大的模型，卷积操作过程中的计算资源和存储需求已成为限制模型小型化和快速化的瓶颈。例如，ResNet-152 网络中卷积层的参数数量占全部参数的 92%，而卷积层的计算量占总计算量的 97%[1]。研究表明，卷积神经网络仅需很少一部分参数即可准确地预测结果，说明卷积核中存在大量的冗余信息。张量分解是一种有效去除冗余信息、加速卷积计算的方法，可以压缩网络规模并提升网络运行速度，有益于深度神经网络在移动嵌入式环境下的高效运行。

一般来说，向量称为一维张量，矩阵称为二维张量，而卷积神经网络中的卷积核可以被视为四维张量。张量分解的目标是将原始张量分解为若干低秩张量，从而减少卷积操作数量并加速网络运行过程。常见的张量分解方法有 CP 分解和 Tucker 分解等。Tucker 分解将卷积核分解为一个核张量与若干因子矩阵，是一种高阶的主成分分析方法。CP 分解是 Tucker 分解的一种特殊形式，其分解过程更为简单，但选取秩的问题较为困难，且可能涉及分解稳定性问题。

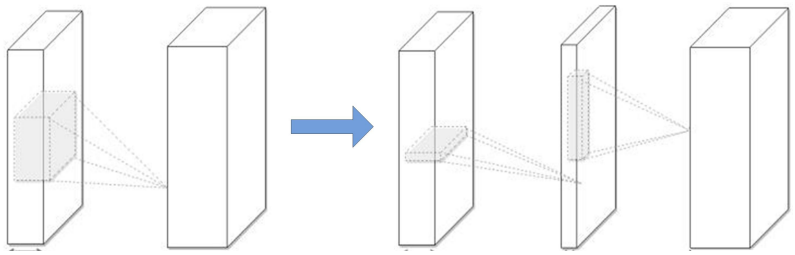

图 4 低秩正则化方法的典型框架。左侧是原始的卷积层，右侧是秩为 K 的低秩约束卷积层。

全连接层可以视为二维张量，因此可以利用矩阵奇异值分解（SVD）去除全连接层的冗余信息。Denton[11]等认为卷积神经网络的绝大部分冗余参数都位于全连接层，因此主要针对全连接层展开奇异值分解，分解后的网络网络参数最多减少 13 倍，同时其运行速度可提升 2-3 倍.

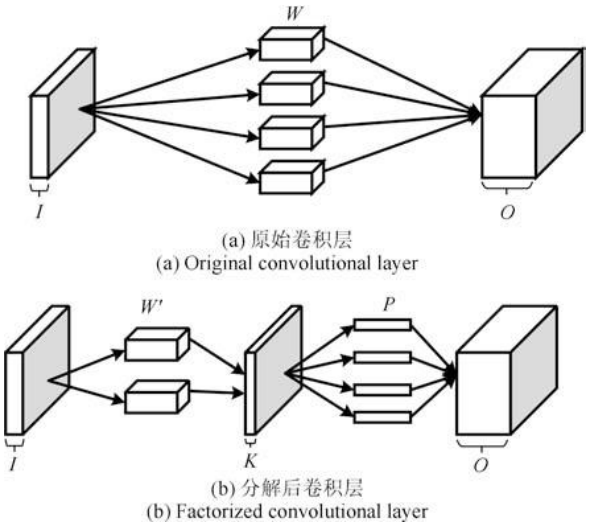

图 5 张量分解过程

基于张量分解的方法虽然能取得一定效果，但大多数方法仅压缩和加速一层或几层网络，缺乏对网络整体的考虑。有研究者据此提出了一些整体网络加速的方法。例如：非对称张量分解方法（将高维卷积核分解为较低维的张量）、基于 PCA 累积能量的低秩选择方法和具有非线性的重构误差优化方法等[13]。

[11]中提出了几种针对卷积核的低秩近似和聚类方案。它们在分类精度下降 1%的情况下实现了 2 倍的单层卷积加速。[12]中提出了使用不同的张量分解方案，用于文本识别，实现了 4.5 倍的加速和 1%的精度降低。

3. 知识蒸馏

知识蒸馏是一种将大型模型或模型集成中的知识压缩到单个较小模型的技术[14]。知识蒸馏可以显著提高机器学习算法的性能，降低部署大型神经网络的计算成本，并通过减少过拟合来提高泛化能力。

实现知识蒸馏的方法通常是训练一个小型的学生模型以模仿大型模型或模型集成的行为。在训练过程中，学生模型学习预测大型模型或模型集成的输出。这样，学生模型可以学习到大型模型或模型集成中的知识，从而实现对知识的压缩。

[14]中提到的知识蒸馏具体方法是将一个大型模型（如一个神经网络集合）中的知识压缩到一个小型模型中。这个小型模型可以比原来的大型模型更容易地部署和使用，并且可以显著减少计算成本。在这个方法中，我们使用大型模型的

输出作为训练数据，然后用这些输出来训练小型模型。在训练过程中，我们尝试最小化小型模型的输出与大型模型的输出之间的差异。通过这种方式，小型模型可以学习到大型模型中的知识，并且可以在测试时产生类似于大型模型的输出。

System	Test Frame Accuracy	WER
Baseline	58.9%	10.9%
10xEnsemble	61.1%	10.7%
Distilled Single model	60.8%	10.7%

图 6 帧分类准确率和 WER 显示，经蒸馏的单个模型的表现与用于创建软目标的 10 个模型的平均预测表现相当。

[4]中使用知识蒸馏时观察到了一个有趣的现象：当主干模型的参数较少时，知识蒸馏会带来更大的收益。例如，当修剪不是很激进时，知识蒸馏可以使 PEEL 在 ResNet-50、ResNet-18 和 MobileNetV2 上分别提高约 0.3%、0.5%和 1.0%的准确率。作者猜测，这些增益归因于小型模型在自己的标签知识方面的不足，因此它们更渴望原始模型的指导。

总之，知识蒸馏可以在显著降低计算成本的同时提高模型泛化能力。与此同时，知识蒸馏对于部署大型神经网络到大量用户具有重要的实际意义。部署模型集成可能非常繁琐且计算成本高昂，特别是当模型集成由大型神经网络组成时。通过将模型集成中的知识压缩到单个模型中，可以更容易地进行部署，并显著降低计算成本。这种方法还有助于提高迁移学习性能，在有限的标签数据任务中具有更好的性能。

4. 量化

量化是一种流行的模型压缩方法，它通过减少神经网络中的参数精度来实现压缩。在量化中，原始的浮点数参数被舍入为定点数，以减少所需的位数。通过降低参数的精度，量化方法可以显著减小神经网络的大小，并提高推理速度和功耗效率[15-16]。

通常使用的几种经典量化技术包括 Ternary weight networks (TWN)、Binary Neural Networks (BNN)和 XNOR-net[15]，它们通过减少权重精度来实现神经网络的压缩，从而实现快速和高效的推理。

其中：

- **Ternary weight networks（TWN）：**

Ternary weight networks（TWN）通过将权重量化为-1、0 和 1 的三值网络来实现神经网络的压缩和加速。在 TWN 中，网络中的每个权重将被近似为-1、0 或 1，这可以将存储和计算要求降低到原始网络的 1/32。该方法通过使用三值权重和常规卷积操作实现了与高精度网络相当的分类性能[17]。

$$\tilde{\mathbf{W}}_i = f(\mathbf{W}_i|\Delta) = \begin{cases} +1 & \text{if} \quad \mathbf{W}_i > \Delta \\ 0 & \text{if} \quad |\mathbf{W}_i| \leq \Delta \\ -1 & \text{if} \quad \mathbf{W}_i < -\Delta \end{cases} \tag{3}$$

图 7 TWN 核心原理

- **Binary Neural Networks（BNN）：**

Binary Neural Networks（BNN）通过将权重量化为-1 和 1 的二值网络来实现神经网络的压缩和加速。BNN 中的权重仅使用-1 和 1 两个值表示，并且在训练过程中，网络只使用二进制权重进行前向传递和反向传播。这样可以大大减少计算和存储开销，并实现高效的推理。虽然二值网络的准确度通常会受到一定程度的影响，但是 BNN 使用的技术可以在分类准确度和网络压缩率之间取得良好的平衡[18]。

$$x^b = \text{Sign}(x) = \begin{cases} +1 & \text{if } x \geq 0, \\ -1 & \text{otherwise,} \end{cases} \tag{1}$$

where x^b is the binarized variable (weight or activation) and x the real-valued variable. It is very straightforward to implement and works quite well in practice. Our second binarization function is stochastic:

$$x^b = \begin{cases} +1 & \text{with probability } p = \sigma(x), \\ -1 & \text{with probability } 1 - p, \end{cases} \tag{2}$$

where σ is the *"hard sigmoid"* function:

图 8 BNN 核心原理

- **XNOR-net：**

XNOR-net 是一种将二值权重和二值激活值结合起来的方法。它是基于 BNN 的扩展，可以在实现快速和高效的神经网络推理的同时具有相对较高的分类精度。在 XNOR-net 中，权重和激活值都被二值化，这使得计算变得更加高效。通过使用 XNOR 和 popcount 操作（即计算一个二进制数字中 1 的个数），可以将神

经网络中的大部分乘法和加法操作转换为更简单的位运算。XNOR-net 能够在保持高精度的同时，将存储和计算要求降低到原始网络的 1/32[19]。

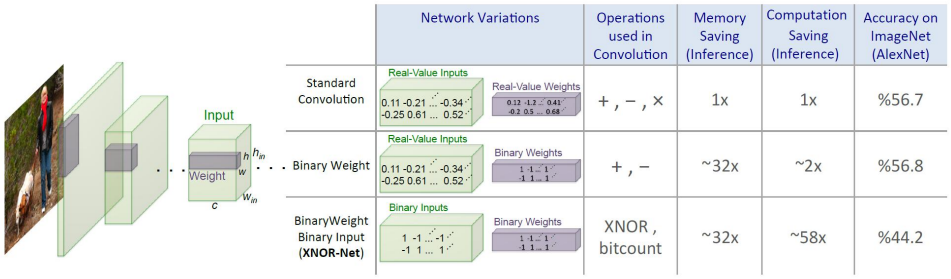

图 9 XNOR-net 优化效果

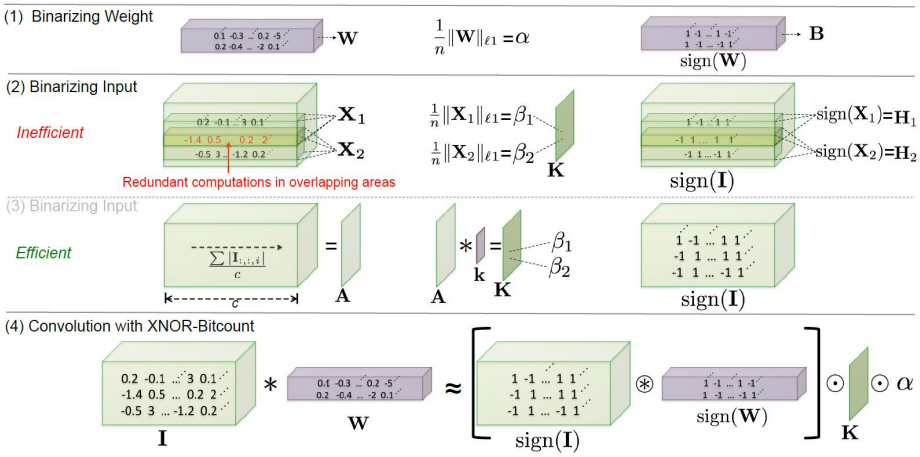

图 10 XNOR-net 核心原理

[15]中介绍了一种新型的量化方案，允许使用仅整数运算进行推理，可以大幅提高模型推理的效率。该方案通过将 CNN 的权重和激活从 32 位浮点数量化为 8 位整数来实现。联合训练的方法被用于同时优化量化后的模型和原始模型，以保留端到端模型精度。

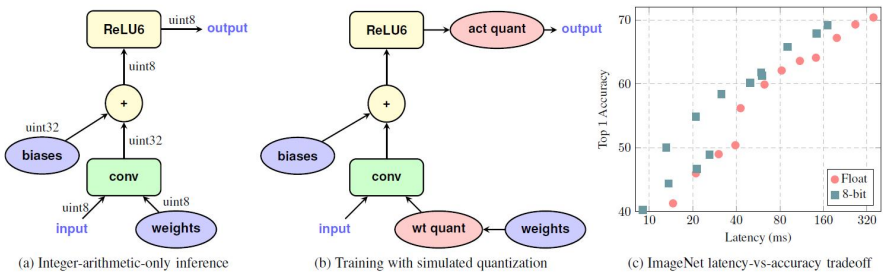

图 11 [15]量化核心原理

具体而言，该方案包括了最大值最小值映射、前向传递、反向传递和距离项等步骤。该方案的优点是在保持较高精度的同时大幅减少模型大小、加速推理速度，并且可以在 ARM CPU 上实现高效计算。根据[15]中的实验结果，在 CIFAR-10 数据集上，使用该方案量化 ResNet-20 模型，可以将模型大小减少 4 倍，并将推理时间缩短 3 倍，同时准确率只下降 0.2%。在 ImageNet 数据集上，使用该方案量化 ResNet-18 模型，可以将模型大小减少 4 倍，并将推理时间缩短 2 倍，同时准确率只下降 1.1%。此外，该优化方案在 MobileNet-V1、VGG-16 等多个模型上也取得了类似的效果。

在[20]中，作者提出了一种名为 Trained Ternary Quantization (TTQ) 的新型神经网络量化方法。这种方法将网络权重压缩为三值。这种方法对于部署在功耗有限的移动设备上的大型神经网络模型具有显著优势。

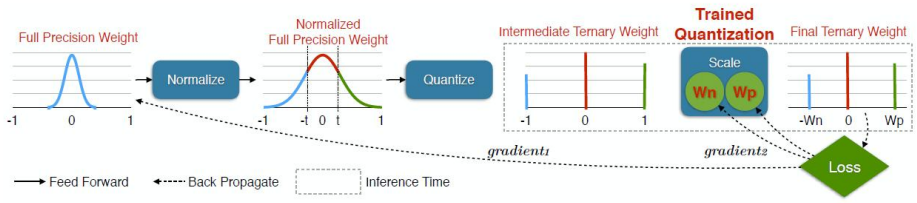

Figure 1: Overview of the trained ternary quantization procedure.

图 12 TTQ 核心原理示意图

TTQ 通过引入两个训练的缩放系数 W1_p 和 W1_n 来将神经网络中的权重精度降低到三值。这些系数针对每一层且使用反向传播进行训练。

$$w_l^t = \begin{cases} W_l^p : \tilde{w}_l > \Delta_l \\ 0 : |\tilde{w}_l| \leq \Delta_l \\ -W_l^n : \tilde{w}_l < -\Delta_l \end{cases}$$

图 13 TTQ 核心原理

在训练过程中，梯度既传播到潜在的全分辨率权重，也传播到缩放系数。我们使用与最大绝对值成比例的逐层阈值来量化权重。因此，在部署三值网络时，我们就只需要三值权重和缩放系数，这可以将参数大小至少减少 16 倍。

根据[20]中的数据，使用 Trained Ternary Quantization (TTQ)方法可以在 CIFAR-10 和 ImageNet 数据集上可以取得达到或甚至超过全精度模型的准确性。在 CIFAR-10 上，通过训练量化方法获得的三元模型比 ResNet-32、44、56 的全

精度模型分别高出 0.04%、0.16%和 0.36%。在 ImageNet 上，这种模型比全精度 AlexNet 模型高出了 0.3%的 Top-1 准确性，并且比之前提到的三元模型（TWN）高出了 3%的准确率[14,20]。与此同时，使用 TTQ 还可以显著减小参数大小，使得在功耗有限的移动设备上部署大型神经网络模型更加容易。

量化方法可以被应用于训练和推理阶段。在训练阶段，量化可以降低计算成本，从而加速训练过程。在推理阶段，量化可以减小模型大小，从而加速推理速度和降低功耗[16]。

虽然量化方法能够显著减小神经网络的大小，但它也可能带来一些性能损失。降低参数精度可能导致模型的准确度下降，并增加推理误差。因此，需要通过适当的权衡，选择合适的量化方法和参数精度，以平衡模型大小和性能损失之间的关系。

5. PEEL 技术（混合技术）

[4]中介绍了一种名为 PEEL 的有效的通道剪枝算法，该算法可以快速生成所需的紧凑模型，并且适用于各种深度学习领域。该算法通过资源重新分配来生成所需的紧凑模型。具体来说，PEEL 技术的实现步骤包括以下五个步骤：

1. 构建过度剪枝的骨干网络和资源池。

2. 用一种基于 Taylor 展开的方法来估计每个层对模型性能的贡献，即估计每个层的重要性。

3. 对层进行分组，以便更好地进行管理资源。

4. 在考虑一些约束条件（例如保持网络结构不变和保持总参数数量不变）的情况下，基于参数的估计重要性将参数重新分配到骨干网络中。

5. 使用知识蒸馏重新训练生成的紧凑模型。

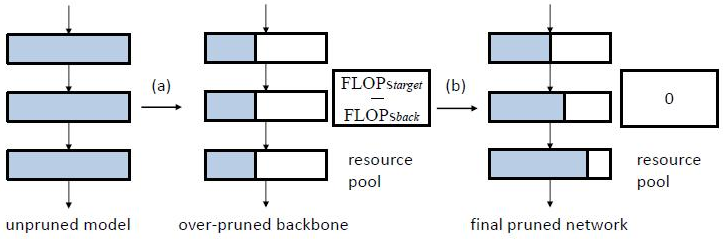

图 14 PEEL 技术实现步骤

通过这些步骤，PEEL 技术可以快速生成所需的紧凑模型，并且适用于各种深度学习领域。

根据论文中的实验结果，当使用现代 CNN 架构，即 ResNet-18，ResNet-50，MobileNetV2，MobileNetV3-small 和 EfficientNet-B0 等进行测试时，PEEL 技术可以在各种剪枝级别下始终可以发现紧凑而准确的架构，展现出与最先进的剪枝算法相当的性能。此外，使用 PEEL 技术可以在模型性能变化小的基础上产生稳定的搜索结果，其搜索成本也显然比现代通道剪枝算法要小。

四、技术总结

本综述报告对 CNN 模型压缩技术进行了梳理，这些方法在降低模型复杂度、减少计算资源和存储需求、提高推理速度方面取得了显著的成果。以下是对之前提到的几种模型压缩技术的简要总结：

1. 网络剪枝与稀疏化

- **阈值剪枝：**

  优点：简单易实现，速度快，易于理解。

  缺点：剪枝粒度较粗可能会导致模型性能下降。

- **迭代剪枝：**

  优点：较好地保留重要参数，减少性能损失。

  缺点：计算量较大，可能需要更长的训练时间。

- **Lottery Ticket 假设：**

  优点：在找到合适的子网络后，训练时间和计算资源消耗相对较少。

  缺点：寻找合适的子网络可能需要较复杂的搜索过程。

2. 低秩分解

优点：

1) 可以压缩网络规模并提升网络运行速度，有益于深度神经网络在移动嵌入式环境下的高效运行。

2) 可以针对全连接层展开奇异值分解，降低冗余参数，进一步提高运行速度。

缺点：

1) 大多数方法仅压缩和加速一层或几层网络，缺乏对网络整体的考虑。

2) 部分方法在提升速度的同时，可能会导致精度降低。

3. 知识蒸馏

优点：

1) 显著降低计算成本：通过将大型模型的知识压缩到一个小型模型中，可以降低模型在实际部署时的计算成本。

2) 提高泛化能力：知识蒸馏可以通过减少过拟合来提高模型的泛化能力。

3) 便于部署：小型模型相对于大型模型更容易被部署和使用，降低了模型的实际应用难度。

缺点包括：

1) 学生模型的性能可能受限：尽管学生模型可以学习到大型模型或模型集成中的知识，但其性能可能无法完全达到大型模型的水平。

2) 训练过程可能较为复杂：知识蒸馏的训练过程可能需要设计较复杂的优化策略以求最小化学生模型的输出与大型模型的输出之间的差异。

4. 量化

优点：

1) 减小模型大小：量化可以通过降低神经网络的参数精度的方式减小模型大小。

2) 加速推理速度：降低参数精度可以提高神经网络的推理速度。

3) 降低功耗：在移动设备上部署时，量化方法可以有效降低模型的功耗。

缺点：

1) 降低参数精度可能导致模型的准确度下降。

2) 降低参数精度可能增加推理误差。

具体到不同的量化方法：

- TWN：将权重量化为-1、0 和 1，可以将存储和计算要求降低到原始网络的 1/32，分类性能与高精度网络相当。
- BNN：将权重量化为-1 和 1，大幅减少计算和存储开销，实现高效推理，但准确度可能受到影响。
- XNOR-net：权重和激活值二值化，可以将存储和计算要求降低到原始网络的 1/32 的同时，仍保持较高的分类精度。
- TTQ：将网络权重压缩为三值，参数大小至少减少 16 倍，准确性达到或超过全精度模型。

综合来看，不同的模型压缩技术各有优缺点和适用场景。在选择压缩方法时，需要综合考虑模型性能、计算资源消耗、训练时间等因素，根据具体应用场景和需求进行权衡。总之，在实际应用中，我们需要根据具体需求和场景，灵活选择并结合这些压缩技术，以达到在保证模型性能的同时减小模型大小和计算复杂度的目的。

五、未来展望与研发方向

随着卷积神经网络在各领域的广泛应用，模型压缩技术将继续受到研究者们的关注。然而，尽管近年来深度学习模型压缩方法领域取得了显著的进展，模型压缩在实际应用中仍面临着精度损失、计算量过大等问题。基于这些问题，我们还需要在保持模型性能的前提下进一步优化模型压缩技术。未来的研究方向可能包括：

模型压缩算法的改进：进一步改进现有的模型压缩技术。例如：现有的剪枝策略通常是固定的，这可能会导致在某些情况下剪枝过多或过少。我们可以基于这一现状研究自适应剪枝策略，根据模型在训练过程中的性能自动调整剪枝程度。这将有望进一步提高模型压缩效果，使得模型在保持性能的同时减少计算资源和存储需求。

自动模型压缩：由于不同的应用场景对模型性能、计算资源消耗等方面的要求可能不同。我们可以基于自动机器学习（AutoML）技术，根据具体应用场景优

化模型压缩方法，以实现更好的压缩效果和性能平衡，为特定应用场景自动选择和配置最佳的模型压缩方案[5]。

与可解释性的结合：模型压缩可以通过降低模型复杂度的方式提高模型的可解释性[21]。随着深度学习模型在各个领域取得显著成功，模型的可解释性变得越来越重要。提高模型的可解释性有助于建立人们对模型的信任，使其在敏感领域（如医疗、金融和法律）得到更广泛的应用。模型压缩技术通过降低模型复杂度，为提高模型可解释性提供了可能。

端到端的模型压缩：现有的模型压缩方法通常只关注模型的某一部分（如权重、结构等）。研究端到端的模型压缩方法，从输入数据、网络结构、权重参数等多个方面进行优化，可以实现更全面的、效果更好的模型压缩。

硬件加速与模型压缩的结合：结合硬件加速技术（如专用 AI 芯片）和模型压缩技术，可以构建高性能、低功耗的深度学习应用。

总之，卷积神经网络模型压缩技术将继续发展，为实现高性能、低资源消耗的深度学习应用提供支持。

参考文献

[1] Lin, J., Wu, X., Chai, Y., & Yin, H. (2020). A review of convolutional neural network structure optimization. Acta Automatica Sinica. (林景栋, 吴欣怡, 柴毅, 尹宏鹏. (2020). 卷积神经网络结构优化综述. 自动化学报. 2020,46(01))

[2] Cheng, Y., Wang, D., Zhou, P., & Zhang, T. (2017). A survey of model compression and acceleration for deep neural networks. arXiv preprint arXiv:1710.09282.

[3] Han, S., Mao, H., & Dally, W. J. (2015). Deep compression: Compressing deep neural networks with pruning, trained quantization, and Huffman coding. arXiv preprint arXiv:1510.00149.

[4] Hou, Y., Ma, Z., Liu, C., Wang, Z., & Loy, C. C. (2021). Network pruning via resource reallocation. arXiv preprint arXiv:2103.01847.

[5] Sze, V., Chen, Y. H., Yang, T. J., & Emer, J. S. (2017). Efficient processing of deep neural networks: A tutorial and survey. Proceedings of the IEEE, 105(12), 2295-2329.

[6] Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2016). Understanding deep learning requires rethinking generalization. arXiv preprint arXiv:1611.03530.

[7] Iofinova, E., Peste, A., & Alistarh, D. (2023). Bias in pruned vision models: In-depth analysis and countermeasures. arXiv preprint arXiv:2304.12622.

[8] Han, S., Pool, J., Tran, J., & Dally, W. (2015). Learning both weights and connections for efficient neural network. In Advances in neural information processing systems (pp. 1135-1143).

[9] He, Y., Zhang, X., & Sun, J. (2018). Channel pruning for accelerating very deep neural networks. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 3-19).

[10] Frankle, J., & Carbin, M. (2018). The lottery ticket hypothesis: Finding sparse, trainable neural networks. arXiv preprint arXiv:1803.03635.

[11] Denton, E. L., Zaremba, W., Bruna, J., LeCun, Y., & Fergus, R. (2014). Exploiting linear structure within convolutional networks for efficient evaluation. In Advances in neural information processing systems (pp. 1269-1277).

[12] Jaderberg, M., Vedaldi, A., & Zisserman, A. (2014). Speeding up convolutional neural networks with low rank expansions. In Proceedings of the British Machine Vision Conference. BMVA Press.

[13] Chadha, G. S., & Schwung, A. (2019). Learning the non-linearity in convolutional neural networks. arXiv preprint arXiv:1905.12337.

[14] Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531.

[15] Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., & Bengio, Y. (2016). Quantization and training of neural networks for efficient integer-arithmetic-only inference. arXiv preprint arXiv:1611.00329.

[16] Courbariaux, M., Bengio, Y., & David, J. P. (2014). Training deep neural networks with low precision multiplications. arXiv preprint arXiv:1412.7024.

[17] Li, F., Zhang, B., & Liu, B. (2016). Ternary weight networks. arXiv preprint arXiv:1605.04711.

[18] Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., & Bengio, Y. (2016). Binarized neural networks. In Advances in Neural Information Processing Systems (pp. 4107-4115).

[19] Rastegari, M., Ordonez, V., Redmon, J., & Farhadi, A. (2016). XNOR-Net: ImageNet classification using binary convolutional neural networks. arXiv preprint arXiv:1603.05279.

[20] Zhu, C., Han, S., Mao, H., & Dally, W. J. (2017). Trained ternary quantization. In Proceedings of the International Conference on Learning Representations (ICLR).

[21] Weber, D., Merkle, F., Schöttle, P., & Schlögl, S. (2023). Less is more: The influence of pruning on the explainability of CNNs. arXiv preprint arXiv:2302.08878.