arXiv:2304.04120v1 [cs.NE] 8 Apr 2023

代理拉格朗日松弛：迈向无需重新训练的深度神经网络剪枝

Shanglin Zhou，美国康涅狄格大学Mikhail A. Bragin，美国康涅狄格大学Lynn Pepin，美国康涅狄格大学Deniz Gurevin，美国康涅狄格大学
Fei Miao，美国康涅狄格大学Caiwen Ding，美国康涅狄格大学

网络剪枝是一种广泛采用的技术，可降低深度神经网络的计算成本和模型规模。然而，典型的三阶段流程，即训练、剪枝和重新训练（微调），会显著增加总训练时间。本文提出一种基于代理拉格朗日松弛（SLR）的系统化权重剪枝优化方法，旨在克服权重剪枝问题因其离散性质而带来的困难。我们进一步证明，该方法能够确保模型压缩问题快速收敛，并通过使用二次惩罚项加速 SLR 的收敛。与其他最先进方法相比，SLR 在训练阶段得到的模型参数更接近其最优值。我们使用 CIFAR-10 和 ImageNet 评估该方法在图像分类任务上的表现，所用模型包括最先进的多层感知机（MLP）网络（如 MLP-Mixer）、基于注意力的网络（如 Swin Transformer），以及基于卷积神经网络的模型，如 VGG-16、ResNet-18、ResNet-50、ResNet-110 和 MobileNetV2。我们还使用多种模型，在 COCO、KITTI 基准数据集和 TuSimple 车道线检测数据集上评估目标检测和分割任务。实验结果表明，在准确率要求相同的情况下，基于 SLR 的权重剪枝优化方法比最先进的方法实现了更高的压缩率；在压缩率要求相同的情况下，也能达到更高的准确率。在分类任务中，我们的 SLR 方法在两个数据集上都能更快地达到目标准确率 3× 。在目标检测和分割任务中，SLR 也能更快地达到目标准确率 2× 。此外，即使在硬剪枝阶段，无需重新训练，SLR 也能实现较高的模型准确率，从而将传统的三阶段剪枝流程简化为两阶段流程。在再训练轮数预算有限的情况下，我们的方法能够快速恢复模型的准确率。

CCS 概念： •**计算机系统组织** → **实时系统架构**。

附加关键词和短语：代理拉格朗日松弛、模型压缩、权重剪枝、图像分类、目标检测和分割

ACM参考文献格式：
Shanglin Zhou， Mikhail A. Bragin、Lynn Pepin、Deniz Gurevin、Fei Miao 和 Caiwen Ding。2018 年。代理拉格朗日松弛：一通往重新训练之路-免费深度神经网络剪枝。在。ACM，纽约，纽约州，美国，17 页。 https://doi.org/XXXXXXX.XXXXXXX

允许为个人或课堂使用而免费制作本作品的全部或部分数字副本或纸质副本，前提是这些副本不得用于或分发以获取利润或商业利益，并且副本首页须附有本声明及完整引文。对于本作品中由 ACM 以外的其他方拥有版权的组成部分，必须尊重其版权。注明出处后，允许摘录。若要以其他方式复制、再版、发布到服务器或分发至列表，须事先获得特定许可和/或支付费用。请通过 permissions@acm.org 申请许可。
© 2018 Association for Computing Machinery。稿件已提交至 ACM

1 引言

基于深度神经网络（DNN）的统计模型对计算和存储资源的需求日益增加，其成本与模型大小（即模型中的参数数量）成正比。这种资源消耗对于嵌入式设备或物联网设备尤其成问题 [16, 35]。通过减小模型大小，可以在评估模型时同时降低存储成本和计算成本。在保持模型性能的同时减小模型大小存在多种技术，例如权重剪枝、稀疏正则化、量化和聚类。这些技术统称为模型压缩 [6, 7, 10, 14, 27, 28, 30, 37, 40]。

这些工作利用了这样一个观察：从头训练一个紧凑模型比重新训练剪枝后的模型更加困难且效果更差 [9, 25]。因此，通常采用三阶段流程：训练（大型模型）、剪枝和重新训练（也称为“微调”）。剪枝过程包括将冗余权重设为零，同时保留重要权重以维持性能。由于硬剪枝后模型准确率会显著下降，因此必须进行重新训练。然而，这种三阶段权重剪枝方法大幅增加了总体训练成本。例如，尽管当前最先进的权重剪枝方法在许多 DNN 架构上能够在保持预测准确率的同时实现非常高的压缩率，但重新训练过程需要更多时间，例如，在 ImageNet 上对 ResNet-18 进行 80 个训练轮次，这相当于使用交替方向乘子法（ADMM）时原始训练轮数的 70% [33, 42]。

鉴于当前权重剪枝方法的优缺点，本文旨在回答以下问题：是否存在一种优化方法，即使在硬剪枝阶段也能达到较高的模型准确率，并显著减少再训练轮数？在再训练轮数预算有限的情况下，是否存在一种优化方法能够快速恢复模型准确率（速度远快于最先进的方法）？

解决这些问题的主要障碍在于模型压缩问题具有离散性，这是由“基数”约束造成的；这类约束确保一定比例的权重被剪枝。本文提出了一种基于近期代理拉格朗日松弛（SLR） [3], 的权重剪枝优化方法，该方法克服了标准拉格朗日松弛面临的所有主要收敛难题。在 SLR 方法中，拉格朗日乘子收敛到最优值的速度远快于其他方法（例如交替方向乘子法）。

我们的贡献/发现总结如下：

- 我们调整了基于 SLR 的方法，以克服权重剪枝问题离散性带来的困难，同时确保快速收敛。
- 我们采用二次惩罚进一步加快 SLR 的收敛。该方法具有良好的收敛性质，这得益于二次惩罚能够迅速减少约束违反；二次惩罚最终也能带来更快的收敛。此外，与 ADMM 等以往方法不同，SLR 能够保证收敛，因此其性能优于其他方法。与其他最先进方法得到的模型参数相比，SLR 得到的参数更接近其最优值。
- 我们为 SLR 方法在权重剪枝问题中的收敛性提供了证明。现有的基于协调的权重剪枝方法在求解非凸问题时无法收敛。其他协调技术（例如 ADMM）并非为处理离散变量及其他类型的非凸性而设计。
- 我们提出的基于SLR的模型压缩方法通过采用基于SLR的权重剪枝优化方法，即使在硬剪枝阶段也能实现较高的模型准确率；在再训练轮数预算有限的情况下，我们的方法能够快速恢复模型准确率。

我们在各种任务和数据集上进行了全面实验，以进一步证明我们提出的基于 SLR的模型压缩方法的有效性。对于分类任务，我们不仅在 VGG-16、ResNet-18、ResNet-50、ResNet-110和 MobileNetV2 等基于卷积神经网络（CNN）的模型上测试了我们的方法，还在非 CNN 模型上进行了测试，例如基于多层感知机（MLPs）的网络 MLP-Mixer，以及基于注意力的网络 Swin Transformer。我们还在分割和检测任务上将我们的 SLR方法与其他最先进水平（SoTA）的剪枝方法进行了测试和比较。我们的实验涉及多个数据集基准测试，包括 CIFAR-10、ImageNet、COCO、KITTI 和 TuSimple。结果表明，我们提出的 SLR方法优于最先进水平的压缩方法。在基于 CNN 和非 CNN 的分类任务中，对于CIFAR-10和ImageNet 数据集，我们的方法都能更快地收敛 3× 至所需准确率；在 COCO目标检测任务中，也能更快地收敛 2× 。此外，在分类任务的硬剪枝阶段，SLR 与最先进水平（SoTA）之间的准确率差距最高可达 6%，而在目标检测和分割任务中，准确率差距可达 44%。

2 相关研究

2.1 模型压缩

鉴于深度神经网络（DNNs）日益增长的计算和存储需求，在现实世界中实现高效的深度学习应用时，模型压缩变得越来越重要。常见的压缩技术有两种：权重剪枝和权重量化。许多研究人员发现，神经网络中的部分权重具有冗余性，因此权重剪枝旨在移除这些不太重要的系数值，并在保持与未压缩模型相近性能的同时实现模型压缩。结构化权重剪枝和非结构化（不规则）权重剪枝是两种主流方法。权重量化是另一种通过减少表示权重所需的比特数来降低权重存储需求的技术。

在早期工作中 [12], 研究人员提出了一种迭代式非规则权重剪枝方法，其中大部分压缩都在全连接层中实现，而卷积层中实现的压缩几乎无法在 GPU上实现显著加速。对于权重存储，该方法减少了 9× AlexNet 中的参数数量，并减少了 13× VGGNet-16 中的参数数量。为了解决非规则权重剪枝的局限性， [38]提出了结构化权重剪枝方法。该方法研究了滤波器、通道和滤波器形状层面的结构化稀疏性。然而，与非结构化剪枝相比，结构化剪枝的整体压缩率受到限制。在准确率不下降的情况下，AlexNet 卷积层中的平均权重剪枝率仅为 1.4×。近期工作 [15] 实现了 2× 通道剪枝，在 VGGNet-16 上的1%准确率下降。随后， [26] 提出了一个用于 L_0 神经网络范数正则化的框架，旨在通过选择权重并将其精确置零，在训练期间对网络进行剪枝。 [9] 提出了彩票假说，该假说指出，随机初始化网络中的一个子网络可以在保持相同性能的情况下替代原始网络。

在这项工作中，我们关注非结构化剪枝。由于它在选择权重方面具有灵活性，因此与结构化剪枝相比，能够实现高得多的准确率 [38] 。

2.2 交替方向乘子法

交替方向乘子法是一种优化算法，它将优化问题分解为更小的子问题，然后对每个子问题进行迭代求解，从而更容易地得到解决。对交替方向乘子法的早期研究可以追溯到20世纪70年代，并且人们讨论了各种可以利用交替方向乘子法高效求解的统计学和机器学习问题 [2]。最近，通过整合强大的

交替方向乘子法。采用交替方向乘子法的成功应用通过对所有目标权重施加动态惩罚，优于先前的方法。该算法可应用于非结构化剪枝和结构化剪枝的各种方案。 [42] 是首个在深度神经网络权重剪枝上实现基于交替方向乘子法框架的工作，实现了 21×在 AlexNet中几乎不损失准确率的不规则权重剪枝。有人提出了一种基于模式的权重剪枝方法，专为移动设备设计和优化，具有很高的效率 [29], 该方法探索了细粒度稀疏性，以最大限度地提高资源受限设备的利用率。 [22] 通过采用基于软约束的公式，改进了之前基于交替方向乘子法的结构化权重剪枝框架，从而实现更高的压缩率并调节更少的超参数。

3 使用替代拉格朗日松弛（SLR）进行权重剪枝

包含 N层，层索引为 $n \in 1, ..., N$，其中第 n层的权重表示为$\mathbf{W}_n$。目标是最小化一个损失函数

$$\min_{\mathbf{W}_n} \{f(\mathbf{W}_n)\} \tag{1}$$

满足各层 n中权重数量的约束，其中非零权重的数量应小于或等于预定义的数量 l_n。该约束可以使用指示函数 $g_n(\cdot)$ 表示为：

$$g_n(\mathbf{W}_n) = \begin{cases} 0 & \text{if } \mathrm{card}(\mathbf{W}_n) \le l_n,\ n = 1, \ldots, N \\ +\infty & \text{otherwise} \end{cases} \tag{2}$$

就问题整体而言，既无法通过解析方法求解，也无法使用随机梯度下降求解。为将其分解为规模较小、易于处理的子问题，引入重复变量，并将问题等价地改写为：

$$\min_{\mathbf{W}_n, \mathbf{Z}_n} \left\{ f(\mathbf{W}_n) + \sum_{n=1}^{N} g_n(\mathbf{Z}_n) \right\}, \quad \text{subject to } \mathbf{W}_n = \mathbf{Z}_n,\ n = 1, \ldots, N \tag{3}$$

这里，第一项是非线性光滑损失函数，第二项是不可微的“基数”惩罚项 [42]。为了解决该问题，首先通过引入拉格朗日乘子放松约束，将由此得到的问题分解为易于处理的子问题，并由乘子协调这些子问题。还会使用二次惩罚项对约束违反进行惩罚，以加快收敛速度。因此，上述优化问题的增广 拉格朗日函数 [2, 42]为：

$$L_\rho(\mathbf{W}_n, \mathbf{Z}_n, \mathbf{\Lambda}_n) = f(\mathbf{W}_n) + \sum_{n=1}^{N} g_n(\mathbf{Z}_n) + \sum_{n=1}^{N} \mathrm{tr}[\mathbf{\Lambda}_n^T(\mathbf{W}_n - \mathbf{Z}_n)] + \sum_{n=1}^{N} \frac{\rho}{2} \|\mathbf{W}_n - \mathbf{Z}_n\|_F^2 \tag{4}$$

其中， $\mathbf{\Lambda}_n$ 是与约束 $\mathbf{W}_n = \mathbf{Z}_n$相对应的拉格朗日乘子矩阵，其维度与$\mathbf{W}_n$ 相同。正标量 ρ 是惩罚系数，$\mathrm{tr}(\cdot)$表示迹， $\|\cdot\|_F^2$ 表示 Frobenius 范数。

接下来，SLR [3], 所带来的可分解性为我们提供了启发。SLR 克服了标准拉格朗日松弛的所有主要难题，显著减少了之字形振荡并确保收敛。松弛后的问题将被分解为两个易于处理的子问题，随后由拉格朗日乘子协调这两个子问题。

步骤 1：求解“损失函数”子问题 $\mathbf{W}_n$，使用随机梯度下降。 在迭代 k中，对于给定的乘子 $\mathbf{\Lambda}_n^k$，第一个“损失函数”子问题是在保持的同时最小化拉格朗日函数。

$\mathbf{Z}_n$ 在先前获得的值$\mathbf{Z}_n^{k-1}$下

$$\min_{\mathbf{W}_n} L_\rho(\mathbf{W}_n, \mathbf{Z}_n^{k-1}, \mathbf{\Lambda}_n^k) \tag{5}$$

由于正则项是一个可微二次函数，且损失函数可微，因此可以通过随机梯度下降（SGD）求解子问题 [1]。为确保乘子更新方向"适当"，需要满足以下"替代"最优性条件，具体如下 [3, 第 179 页，公式（12）]:

$$L_\rho(\mathbf{W}_n^k, \mathbf{Z}_n^{k-1}, \mathbf{\Lambda}_n^k) < L_\rho(\mathbf{W}_n^{k-1}, \mathbf{Z}_n^{k-1}, \mathbf{\Lambda}_n^k) \tag{6}$$

如果满足（6），则按照以下方式更新乘子 [3, 第 179 页，公式（15）] :

$$\mathbf{\Lambda'}_n^{k+1} := \mathbf{\Lambda}_n^k + s'^k(\mathbf{W}_n^k - \mathbf{Z}_n^{k-1}) \tag{7}$$

其中，步长更新为 [3, 第 180 页，公式（20）]，即

$$s'^k = \alpha^k \frac{s^{k-1}||\mathbf{W}^{k-1} - \mathbf{Z}^{k-1}||}{||\mathbf{W}^k - \mathbf{Z}^{k-1}||} \tag{8}$$

步骤 2：通过使用离散子空间投影进行剪枝，求解 Z 的"基数"子问题。第二个"基数"子问题针对以下变量求解$\mathbf{Z}_n$，同时将其他变量固定为以下值 $\mathbf{W}_n^k$:

$$\min_{\mathbf{Z}_n} L_\rho(\mathbf{W}_n^k, \mathbf{Z}_n, \mathbf{\Lambda'}_n^{k+1}) \tag{9}$$

由于 $g_n(\cdot)$ 是指示函数，该问题的全局最优解可显式推导为 [2] :

$$\mathbf{Z}_n^k = \Pi_{\mathbf{S}_n}(\mathbf{W}_n^k + \frac{\mathbf{\Lambda'}_n^{k+1}}{\rho}) \tag{10}$$

其中，$\Pi_{\mathbf{S}_n}(\cdot)$ 表示到集合 $\mathbf{S}_n = \{\mathbf{W}_n \,|\, \mathrm{card}(\mathbf{W}_n) \le l_n\}$ $n = 1, \ldots, N$ 的欧几里得投影。为确保乘子更新方向是"适当的"，需要满足以下"替代"最优性条件:

$$L_\rho(\mathbf{W}_n^k, \mathbf{Z}_n^k, \mathbf{\Lambda'}_n^{k+1}) < L_\rho(\mathbf{W}_n^k, \mathbf{Z}_n^{k-1}, \mathbf{\Lambda'}_n^{k+1}) \tag{11}$$

一旦（11）得到满足，[1] 乘子更新如下:

$$\mathbf{\Lambda}_n^{k+1} := \mathbf{\Lambda'}_n^{k+1} + s^k(\mathbf{W}_n^k - \mathbf{Z}_n^k) \tag{12}$$

其中，步长和步长设置参数 [3, 第 188 页，公式 (67)] 更新如下:

$$s^k = \alpha^k \frac{s'^k||\mathbf{W}^{k-1} - \mathbf{Z}^{k-1}||}{||\mathbf{W}^k - \mathbf{Z}^k||}; \quad \alpha^k = 1 - \frac{1}{M \times k^{(1-\frac{1}{k^r})}}, \; M > 1, 0 < r < 1 \tag{13}$$

理论结果基于 [4, 5, 11]，总结如下:

定理 1（收敛的充分条件）假设对于任意整数 κ 存在 $k > \kappa$ 使得代理最优性条件（6）和（11）得到满足，那么在步长条件（8）和（13）下，拉格朗日乘子收敛到其最优值 $\mathbf{\Lambda}_n^*$，使以下对偶函数最大化:

$$q(\mathbf{\Lambda}) \equiv \min_{\mathbf{W}_n, \mathbf{Z}_n} L_\rho(\mathbf{W}_n, \mathbf{Z}_n, \mathbf{\Lambda}_n) \tag{14}$$

证明。证明将基于 [3]。原始 SLR方法 [3] 与本文的 SLR方法之间的主要区别在于拉格朗日函数中存在二次项（4）。

[1] 如果不满足条件（11），则使用$\mathbf{W}_n$ 和$\mathbf{Z}_n$的最新可用值重新求解子问题（5）和（9）。

值得一提的是，权重剪枝问题可以等价地改写为通用形式（15），其中 $\mathbf{X}$ 统称决策变量 $\{\mathbf{W}_n, \mathbf{Z}_n\}$

$$\min_{\mathbf{X}} \mathbf{F}(\mathbf{X}), \quad s.t. \quad \mathbf{G}(\mathbf{X}) = \mathbf{0} \tag{15}$$

其中

$$\mathbf{F}(\mathbf{X}) \equiv f(\mathbf{W}_n) + \sum_{n=1}^{N} g_n(\mathbf{Z}_n) + \sum_{n=1}^{N} \frac{\rho}{2} \|\mathbf{W}_n - \mathbf{Z}_n\|_F^2, \quad \mathbf{G}(\mathbf{X}) \equiv \mathbf{W}_n - \mathbf{Z}_n, \quad n = 1, \ldots, N \tag{16}$$

(15) 的可行集等价于原始模型压缩问题。可行性要求 $\mathbf{W}_n = \mathbf{Z}_n$，这使得 (16) 中的项 $\frac{\rho}{2}\|\mathbf{W}_n - \mathbf{Z}_n\|_F^2$ 消失。因此，与 (15) 对应的拉格朗日函数就是原始模型压缩问题的增广拉格朗日函数 (4)。此外，(6) 和 (11) 中的代理最优性条件，是与对应于 (15) 的拉格朗日函数 $\mathbf{F}(\mathbf{X}) + \mathbf{\Lambda}\mathbf{G}(\mathbf{X})$ 相对应的代理最优性条件。因此，由于在原始 SLR 中，[3，命题 2.7，第 188] 页已证明：在满足步长条件 (8) 和 (13) 以及代理最优性条件的情况下可以收敛，而这里假定这些条件均得到满足，所以对于所考虑的模型压缩问题，乘子同样会收敛到其最优值。□

尽管定理 1 中的收敛性证明对于根据收敛性区分 SLR与先前的分解和协调方法（例如，交替方向乘子法）具有价值，但其本身并未揭示解的质量。这一问题在定理 2中得到了解决，其中对 SLR 更快的收敛进行了严格量化。由于交替方向乘子法和 SLR方法都是用于最大化对偶函数的对偶方法（14），因此确定最大化问题的上界是一种常见做法。在所研究问题的背景下，最优对偶值的上界 q^* 将在每种方法中建立，从而能够评估对偶解的质量——具体而言，拉格朗日乘子充当对偶空间中的决策变量。[2]

定理 2（对偶解质量：最佳情况性能）。假设满足定理 1 中所述的"收敛的充分条件"，则与交替方向乘子法相比，代理拉格朗日松弛可提供更高的对偶解质量：具体而言，存在一个迭代 κ，使得对于所有迭代 $k > \kappa$，以下条件均成立：

$$\overline{q}_{\kappa}^{SLR} < \overline{q}_{\kappa}^{ADMM} \tag{17}$$

证明。存在一个最优对偶值的上界 [4, 5],，就当前考虑的问题而言，可表示为：

$$\overline{q}_{k}^{SLR} = \gamma \cdot s^k \cdot \|\mathbf{W}_n^k - \mathbf{Z}_n^k\|_F^2 + L_\rho(\mathbf{W}_n^k, \mathbf{Z}_n^k, \mathbf{\Lambda}_n^k) \tag{18}$$

此处 $\gamma \in [0, 1]$ 是一个参数。由于在代理拉格朗日松弛中，步长趋近于 0，因此 n

$$\overline{q}_{k}^{SLR} \to L_\rho(\mathbf{W}_n^k, \mathbf{Z}_n^k, \mathbf{\Lambda}_n^k) \tag{19}$$

此外，假设满足收敛的充分条件，则拉格朗日对偶值趋近于对偶值（例如，已有证明见 [3]），因此，

$$\overline{q}_{k}^{SLR} \to q(\lambda^*) \tag{20}$$

然而，在 ADMM 中，由于步长/惩罚系数不会趋近于零，因此最优对偶值的高估值 $q(\lambda^*)$与其之间始终存在非零距离：

$$\overline{q}_{k}^{ADMM} = \gamma \cdot \rho \cdot \|\mathbf{W}_n^k - \mathbf{Z}_n^k\|_F^2 + L_\rho(\mathbf{W}_n^k, \mathbf{Z}_n^k, \mathbf{\Lambda}_n^k) \tag{21}$$

[2]原始变量（权重和偏置）的质量更高，这意味着通过拉格朗日乘子进行更优的协调，能够显著提升原始解的质量。尽管这一论断所处的问题背景不同，但已通过实证得到验证 [5]。

因此，由于第一项不会趋近于零，且第二项不会从上方趋近于最优对偶值，故存在 κ 使得（17）成立。□

算法1代理拉格朗日松弛

1：初始化$\mathbf{W}^0{}_n, \mathbf{Z}_n^0, \mathbf{\Lambda}_n^0$ 且 s^0
2：**当**尚未满足停止条件**时执行**
3: **1** 求解子问题 (5),
4: **如果** 替代最优性条件(6) 得到满足 **则**
5: 保持 $\mathbf{W}_n^k$,$\mathbf{Z}_n^k$，并根据 $\mathbf{\Lambda}_n^k$ 式（7）更新乘子，
6: **else**
7: 保持 $\mathbf{W}_n^k$,$\mathbf{Z}_n^k$，不要更新乘子 $\mathbf{\Lambda}_n^k$,
8: **结束 if**
9: **2** 求解子问题 (9),
10: **如果**替代最优性条件（11）得到满足**那么**
11: 保留$\mathbf{W}_n^k$，$\mathbf{Z}_n^k$，并根据（12）更新乘子$\mathbf{\Lambda}_n^k$，
12: **else**
13: 保留$\mathbf{W}_n^k$，$\mathbf{Z}_n^k$，不更新乘子$\mathbf{\Lambda}_n^k$,
14: **结束 if**
15: **结束 while**

所提出的SLR方法的算法总结见算法1。SLR方法受益于高效的子问题求解协调，并在步长（8）和（13）趋于零时保证收敛。若不满足这一要求，乘子（12）将无法收敛。满足替代最优性条件（6）和（11）可确保乘子沿着“良好”方向更新。第4节将通过实证验证，总存在某次迭代 κ 在此之后，“替代”最优性条件得到满足，从而确保在整个迭代过程中，乘子趋近其最优值。

SLR方法还受益于对两个超参数的独立且系统的调整：惩罚系数 ρ 和步长 s^k。相比之下，其他协调方法并非设计用于处理离散变量和其他类型的非凸性。例如，交替方向乘子法在求解非凸问题 [2,p. 73] 时无法收敛，因为步长 ρ 在该方法内不会收敛到零。将交替方向乘子法中的步长降至零也会导致惩罚系数降低，从而导致收敛速度变慢。

4 评估

在本节中，我们将讨论使用基于 CNN 的模型和非基于 CNN 的模型进行图像分类任务的实验结果。我们还会评估该方法在目标检测和图像分割任务中的表现。

4.1 实验设置

我们的所有代码，包括图像分类任务、目标检测和分割任务，均使用 Python 3.6 和 PyTorch 1.6.0 实现。对于 COCO 2014 数据集上的实验，我们使用了 pycocotools v2.0 软件包。对于 TuSimple 车道线检测基准数据集[3]上的实验，我们使用了 SpConvv1.2 软件包。我们在Ubuntu 18.04 上使用配备24 GB GPU 显存的 Nvidia Quadro RTX 6000 GPU 进行了实验。

我们首先通过 SLR训练对预训练模型进行剪枝。随后，我们对模型执行硬剪枝，从而完成压缩阶段。我们报告总体压缩率（或剩余权重的百分比）和预测准确率。

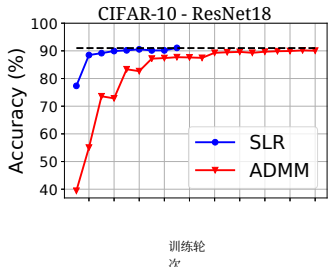
[3]https://github.com/TuSimple/tusimple-benchmark

4.2 使用基于 CNN 的模型进行图像分类任务评估

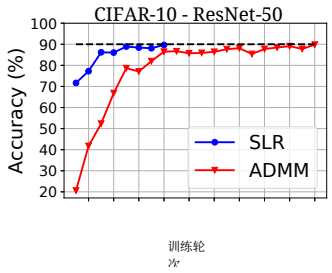
模型和数据集。 我们使用 ResNet-18、ResNet-50、ResNet-56、ResNet-110 [13] 和 VGG-16 [35] 在 CIFAR-10 上。在 ImageNet ILSVRC 2012 基准测试中，我们使用 ResNet-18、ResNet-50 [13] 和 MobileNetV2 [34]。我们使用 Torchvision 的"models"子包中在 ImageNet 上预训练的 ResNet 模型。

表 1。 SLR 和 交替方向乘子法在 CIFAR-10 和ImageNet上的比较。ImageNet 结果显示 Top-1 / Top-5 准确率 y.

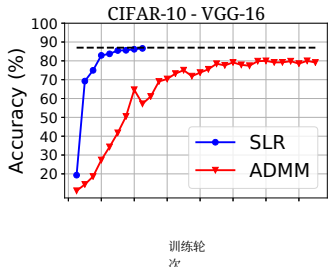
	基线 (%)	训练轮次	交替方向乘子法 (%)	SLR (%)	压缩率
CIFAR-10					
ResNet-18	93.33	40	72.84	*89.93*	8.71×
ResNet-50	93.86	50	78.63	*88.91*	6.57×
VGG-16	93.27	110	69.05	*87.31*	12×
ResNet-56	93.39	30	90.5	*92.3*	6.5×
ResNet-110	93.68	30	89.71	*92.31*	9.7×
ImageNet					
ResNet-18	69.7 / 89.0	40	58.9 / 81.7	*60.9 / 84.4*	6.5×
ResNet-50	76.1 / 92.8	30	64.8 / 85.1	*65.9 / 87.5*	3.89×
MobileNetV2	71.8 / 91.0	60	61.8 / 84.3	*63.2 / 85.5*	1.76×



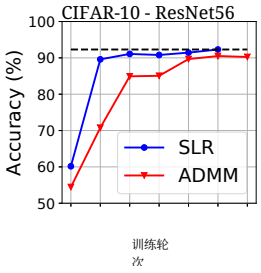
(a) CIFAR-10 上的 ResNet-18。



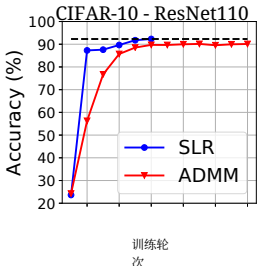
(b) CIFAR-10 上的 ResNet-50。



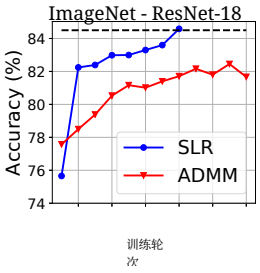
(c) CIFAR-10 上的 VGG-16。



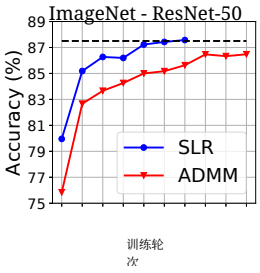
(d) Res Net-56 在 CIFAR-10 上。



(e) ResNet-110 在 CIFAR-10 上。



(f) ResNet-18 在 ImageNet 上。



(g) ResNet-50 在 ImageNe t.

图 1. 在 CIFAR-10 和 ImageNet 基准测试上经过 SLR 和交替方向乘子法训练后的硬剪枝准确率。准确率会定期报告，并在达到目标准确率时停止训练。

训练设置。在所有实验中，我们设置 $\rho = 0.1$。在 CIFAR-10 实验中，我们使用 0.01 的学习率、128 的批量大小和 ADAM优化器。在 ImageNet 实验中，我们使用 10^{-4} 的学习率、256 的批量大小和 SGD 优化器。为公平比较 SLR 和交替方向乘子法，我们在实验中对两种方法使用相同的训练轮数和稀疏度配置。

表1 展示了我们在 CIFAR-10 和 ImageNet 基准测试上对 SLR 和交替方向乘子法的比较。对于这两组实验，SLR 参数设为 $M = 300$，$r = 0.1$，以及 $s_0 = 10^{-2}$。SLR 和交替方向乘子法训练完成后，我们执行硬剪枝，并在训练轮数预算有限的情况下，不进行额外重新训练，报告硬剪枝准确率。结果表明，在相同压缩率下，SLR 的准确率优于交替方向乘子法。在压缩率相同的情况下，经过相同训练轮数后，SLR 的分类准确率始终高于交替方向乘子法。随着压缩率提高，不同网络架构上的结果显示，SLR 与交替方向乘子法之间的准确率差距更加显著。这表明，在有限的训练预算下，SLR 收敛更快，并能在硬剪枝阶段迅速恢复模型准确率，从而无需重新训练。

图 1 展示了表1所对应的 CIFAR-10 和 ImageNet 上 SLR 与交替方向乘子法的硬剪枝准确率。在训练过程中，会定期检查硬剪枝准确率。如果硬剪枝准确率达到准确率标准，训练便会停止。如图 1所示，SLR 收敛迅速，并且在 CIFAR-10 上几乎比交替方向乘子法快 3× 倍达到目标准确率。此外，在图 1c中，VGG-16 经过 300 个训练轮次后，交替方向乘子法仍未达到目标准确率，而 SLR 在 80 个训练轮次时便完成训练。同样，如图 1f 和 1g所示，ImageNet 上经过 60 和 50 个训练轮次后，交替方向乘子法仍无法达到目标准确率，而 SLR 更快达到阈值。

表 2. SLR 与 VGG-16、ResNet-18、ResNet-50 和 ResNet-56 在 CIFAR-10 上的性能比较。

模型	方法	准确率(%)	参数 剪枝 (%)
VGG-16	**SLR**	91.2	90
	AMC [14]	91.0	
	L0 [26]	80.0	
	SLR	93.1	60
	一次性剪枝[25]	92.4	
	SLR	93.2	50
	迭代剪枝 [12]	92.2	
ResNet-18	**SLR**（在 *20k* 次迭代时）	89.9	88.6
	迭代剪枝 [9]	75.0	
ResNet-50	**SLR**	93.6	60
	AMC [14]	93.5	
ResNet-56	**SLR**	92.3	84.4
	GSM [8]	94.1	85.0
	组稀疏性 [20]	92.65	79.2
	[43]	92.26	20.49
	GAL-0.6 [23]	93.38	11.8
	[18]	93.06	13.7
	NISP [41]	93.01	42.6
	KSE [21]	93.23	54.73
	DHP-50 [19]	93.58	41.58

表 2 展示了 SLR 与其他近期模型压缩方法在 CIFAR-10 基准上的比较。我们报告了 SLR 训练后剪除参数的百分比以及最终准确率。我们开始训练网络时使用

学习率为0。1，并在训练轮次 80 和 100时将学习率降低10 倍。在 ResNet-18 上，我们仅在20k 次迭代时比较结果。对于 VGG-16和 ResNet-50，我们观察到 SLR 能够实现高达 60% 的剪枝，同时准确率下降不到 1%。

4.3 使用先进非 CNN 模型进行图像分类任务评估 s

在本小节中，我们展示将 SLR 权重剪枝方法应用于基于多层感知机（MLP）的架构 MLP-Mixer [36] 以及基于注意力的网络 Swin Transformer 的实验结果 [24]。

模型与数据集。MLP-Mixer 不包含任何卷积层或自注意力模块。该架构完全依赖多层感知机。通过交替使用通道混合 MLP 和 token（图像块）混合 MLP，MLP-Mixer 能够以远少于先进方法的计算资源取得不错的准确率。Swin Transformer 是一种基于层级 Transformer 的架构，利用移位窗口进行表征。其计算复杂度相对于输入图像大小呈线性，并在图像分类、目标检测和语义分割任务上均达到先进水平。我们在 ImageNet ILSVRC 2012 基准数据集上进行实验。我们使用来自 PyTorch Image Models 代码库的预训练模型 [39]。

训练设置。与基于 CNN 的模型实验类似，我们设置 $\rho = 0.1$。我们将学习率设为 0.01，批量大小设为 128，并使用动量为 0.9、权重衰减为 10^{-4} 的 SGD 优化器。为确保公平比较，我们对 SLR 和交替方向乘子法采用相同的训练轮数和稀疏度配置。

SLR 与交替方向乘子法的比较。 表 3 展示了将 SLR 和交替方向乘子法应用于 ImageNet 基准上的 MLP-Mixer 模型时的比较结果。我们使用三种不同的压缩率对这两种方法进行比较。在无需进一步重新训练的情况下，经过 100 个训练轮次后，报告最终的硬剪枝准确率。经过 50 个训练轮次后，还会报告重新训练后的准确率。对于所有 SLR 实验，我们将参数设置为 $M = 300$、 $r = 0.1$以及 $s_0 = 0.01$。如表 3 所示，就硬剪枝而言，当压缩率较低（即 1.96×）时，两种方法的性能相近。随着压缩率提高到 3.16× 和 8.28×，SLR 的表现优于交替方向乘子法，并且随着压缩率的

表 3. 使用 MLP-Mixer 在 ImageNet 上以不同压缩率进行 SLR 剪枝的结果。

基线准确率（%）		压缩 Rate	硬剪枝准确率（%）				重新训练准确率（%）			
			ADMM		SLR		ADMM		SLR	
Top-1	Top-5		Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
76.598	92.228	1.96×	75.332	91.734	75.392	91.752	75.332	91.714	75.392	91.752
		3.16×	71.798	89.64	72.578	90.136	73.362	90.672	73.56	90.750
		8.28×	47.592	71.698	54.834	77.696	70.864	89.38	71.036	89.494

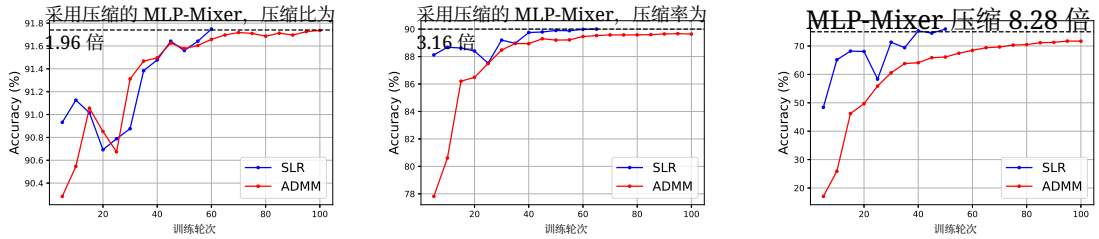
表4.不同压缩率下，使用 Swin Transformer（Tiny）在 ImageNet 上进行 SLR剪枝的结果 .

基线准确率（%）		压缩 Rate	硬剪枝准确率（%）				重新训练准确率（%）			
			ADMM		SLR		ADMM		SLR	
Top-1	Top-5		Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
81.350	95.532	1.95×	78.910	94.288	79.018	94.446	79.146	94.424	79.108	94.456
		3.13×	73.074	91.370	74.322	91.988	75.278	92.55	75.508	92.630
		4.50×	63.892	85.104	67.398	87.494	72.626	90.966	73.046	91.364

压缩率不断提高。这表明，SLR 在硬剪枝阶段的准确率高于交替方向乘子法，因此能更高效地利用训练资源。

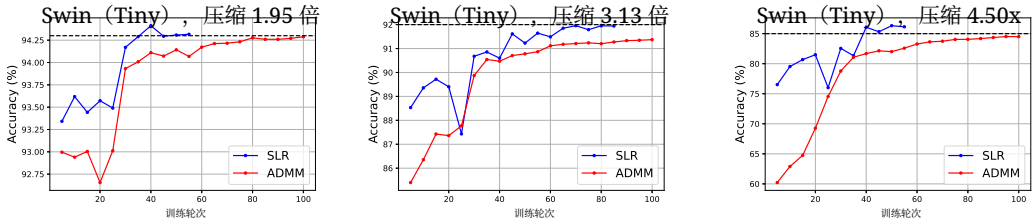
表4展示了在 ImageNet 基准上，SLR 与交替方向乘子法在 Swin Transformer 模型上的比较。在所有实验中，我们都选择 tiny 版本（Swin-T），因为它能很好地平衡模型大小和准确率。对于 SLR 实验，当压缩率为 1.96× 时，我们使用的参数为 $M = 150$、$r = 0.1$ 和 $s_0 = 0.05$。当压缩率为 3.16× 时，我们使用的参数为 $M = 300$、$r = 0.1$ 和 $s_0 = 0.01$。当压缩率为 4.5× 时，我们使用的参数为 $M = 150$、$r = 0.05$ 和 $s_0 = 0.005$。如结果所示，当压缩率为 1.95× 时，在硬剪枝阶段，SLR 的准确率高于交替方向乘子法。经过 50 个训练轮次的重新训练后，交替方向乘子法的准确率提高了 0.2%，更接近 SLR 的准确率，而 SLR 的准确率提高了 0.01%。这表明，我们的 SLR 在硬剪枝阶段能够获得更高的准确率。随着压缩率提高，SLR 与交替方向乘子法之间的准确率差距也随之扩大。例如，当压缩率为 4.5× 时，在硬剪枝阶段，SLR 的准确率比交替方向乘子法高 2.4%。

图 2 展示了在 ImageNet上使用MLP-Mixer时，定期检查 SLR 和交替方向乘子法的硬剪枝准确率的结果。压缩率对应于表 3，遵循第 4.2节所述的相同步骤。如图所示，SLR收敛迅速，且相比交替方向乘子法实现了更高的准确率。当压缩率较低时，例如1.96×，两种方法以相似的速率收敛。然而，随着压缩率的提高，SLR不仅比交替方向乘子法收敛更快，而且达到的准确率也高得多。即使在压缩率为 60 个训练轮次后，交替方向乘子法仍无法达到预期的准确率，而该压缩率为 8.28×。在图 2c中，我们可以看到 SLR 在大约50 个训练轮次时迅速收敛至阈值，并在停止 SLR 训练时，与交替方向乘子法产生了 8.6% 的准确率差距；交替方向乘子法的准确率仍在继续提高，尽管速度缓慢，直到训练轮次100。



(a) 压缩率 = 1.96×。　(b) 压缩率 = 3.16×。　（c）压缩率 = 8.28×。

图2. Top-5 H　经过 SLR 和交替方向乘子法剪枝后，MLP-Mixer 在 ImageNet 上的准确率。准确率按周期报告。　.



(a) 压缩率 = 1.95×。　(b) 压缩率 = 3.13×.　(c) 压缩率 = 4.5×。

图3. 在应用 SLR 和交替方向乘子法权重剪枝后，Swin Transformer（Tiny）在 ImageNet 基准测试上的 Hardprune 的 Top-5 准确率。准确率定期报告。

图 3 展示了在ImageNet 上使用SwinTransformer时，SLR 和 ADMM 经定期检查的硬剪枝准确率，其压缩率与表 4中报告的压缩率相对应。在三种压缩率下，与ADMM相比，SLR 始终能以一半的训练轮数收敛到目标准确率。即使压缩率为 3.16×，SLR 在停止训练时与 ADMM 的准确率差距约为 1% 。这进一步表明，在硬剪枝阶段本身，我们的 SLR 能够快速恢复模型准确率。

4.4 目标检测和分割任务上的评估

在本小节中，我们在三个目标检测和分割基准测试上评估了我们基于 SLR 的权重剪枝方法。

模型和数据集。 在第一个实验中，我们在 COCO 2014 基准上测试 YOLOv3 和 YOLOv3-tiny 模型 [32] 。我们使用了公开的 Ultralytics 仓库[4] 中的 YOLOv3 及其预训练模型。第二个实验聚焦于车道线检测。我们在 TuSimple 车道线检测基准数据集上使用来自 Ultra-Fast-Lane-Detection [31] 的预训练模型。第三个实验涉及 3D 点云目标检测。我们按照 OpenPCDet 仓库[5]的设置，在 KITTI 2017 数据集上使用 PointPillars [17] 预训练模型。在该实验中，我们使用 LIDAR 点云数据作为输入。每个点云数据点都存储为大量 3D 高程点的集合，每个点表示为一个包含 x, y, z(3D 坐标)和强度 [44]的向量 $1 * 4$。

训练设置。 在所有实验中，我们使用 $\rho = 0.1$。我们将 SLR 参数设置为 $M = 300$、 $r = 0.1$ 和 $s_0 = 10^{-2}$。我们遵循所用代码库提供的相同训练设置。最后，我们为交替方向乘子法和 SLR 使用相同的训练轮数和稀疏度配置。

表 5. 在 COCO 数据集上，采用不同结构的 YOLOv3 进行 ADMM 和 SLR剪枝的结果。

架构	训练轮次	方法	Hardprune mAP	压缩 Rate
YOLOv3-tiny (平均精度均值 = 37.1)	15	ADMM	35.2	1.19×
		SLR	*36.1*	
	20	ADMM	32.2	2×
		SLR	*36.0*	
	25	ADMM	25.3	3.33×
		SLR	*35.4*	
YOLOv3-SPP (平均精度均值 = 64.4)	15	ADMM	41.2	2×
		SLR	*53.2*	

表6. 在 TuSim- ple 基准上，使用 ResNet-18 通过不同压缩率进行 SLR剪枝的结果。

压缩 Rate	Hardprune 准确率（%）		重新训练准确率（%）	
	ADMM	SLR	ADMM	SLR
1.82×	92.49	94.64	94.28	94.63
2.54×	92.25	94.56	94.04	94.93
4.21×	90.97	94.66	94.18	94.68
12.10×	88.41	94.51	94.45	94.7
16.85×	78.75	94.55	94.23	94.65
22.80×	67.79	94.62	94.08	94.55
35.25×	57.05	93.93	93.63	94.34
77.67×	46.09	89.72	88.33	90.18

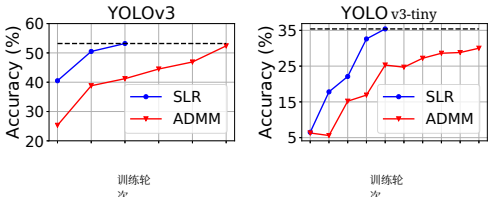
测试设置。 对于 YOLOv3 模型，我们在测试时使用大小为 640 的图像，计算 COCO 的交并比 = 0.50 下的平均精度均值。在车道线检测实验中，评估指标为“准确率”，其计算方式为 $\sum_{\sum} C_{\text{clip}}$每个片段中正确预测的车道线点数 S_{clip}，其中 C_{clip}是每个片段中正确预测的车道线点数， S_{clip}是每个片段中真实标注的总数。

KITTI 数据集分为简单难度、中等难度和困难难度级别。这里，简单难度 表示边界框的最小高度为40像素，图像中的所有物体均完全可见，并且物体的截断比例小于15%；中等难度 表示边界框的最小高度为25像素，图像中的物体

[4]https://github.com/ultralytics/yolov3
[5]https://github.com/open-mmlab/OpenPCDet

部分可见，且物体被截断的比例低于 30%；困难级别 的边界框最小高度为 25 像素，但图像中的某些物体难以辨认，最大截断比例为 50%。在每个难度级别下计算平均精度均值，使用 $IoU = 0.5$。

SLR与交替方向乘子法的比较。。我们在COCO数据集上对SLR方法和交替方向乘子法进行了比较，结果如表 5所示。我们针对YOLOv3-tiny在三种不同压缩率下比较了这两种方法，并使用压缩率为 1.98× 的YOLOv3-SPP预训练模型进行了测试。可以看出，在所有情况下，采用SLR方法剪枝的模型在硬剪枝后都具有更高的准确率。观察YOLOv3-tiny的结果可以发现，随着压缩率提高，SLR的优势更加明显。



(a) YOLOv3。 (b) YOLOv3-tiny。

图 4. YOLOv3 和 YOLOv3-tiny 的硬剪枝准确率。每 5 个训练轮次报告一次准确率，当方法达到准确率阈值时停止训练。

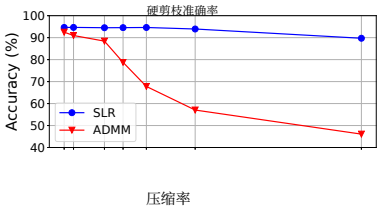
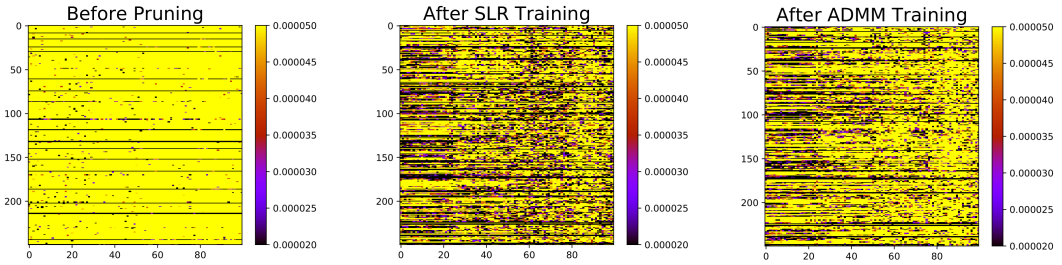
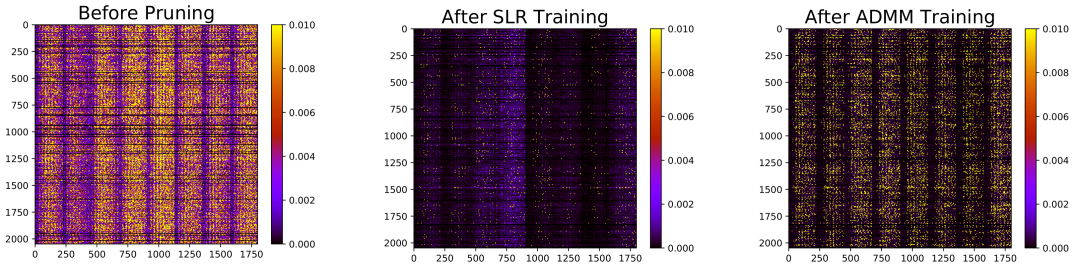


图 5. 在 TuSimple 基准数据集上，使用交替方向乘子法与 SLR 训练时，不同压缩率下的硬剪枝准确率。随着压缩率提高，SLR 相较于交替方向乘子法的优势更为明显。



(a) 在相同压缩率（77.6×）下，采用 SLR（中）和交替方向乘子法（右）剪枝前后的权重。



(b)在相同准确率（89.0%）下，使用 SLR（中）和交替方向乘子法（右）进行剪枝前后的权重。

图 6. 使用 SLR 和交替方向乘子法剪枝前后 ResNet-18 在 TuSimple 基准数据集上的权重热图。与交替方向乘子法相比，SLR 会将更多权重置零。

在压缩率 3.33× 下的 YOLOv3-tiny 中，在给定 25 个训练轮次的限制时，我们可以观察到ADMM与 SLR 之间的差距要大得多，这是由于 SLR收敛更快，如图 4b所示。同样，图 4a展示了在 SLR 和 ADMM 训练期间 YOLOv3 的平均精度均值变化，训练 50 个训练轮次，并采用 2× 压缩率进行剪枝。SLR 直到第 15个训练轮次才达到平均精度均值阈值。

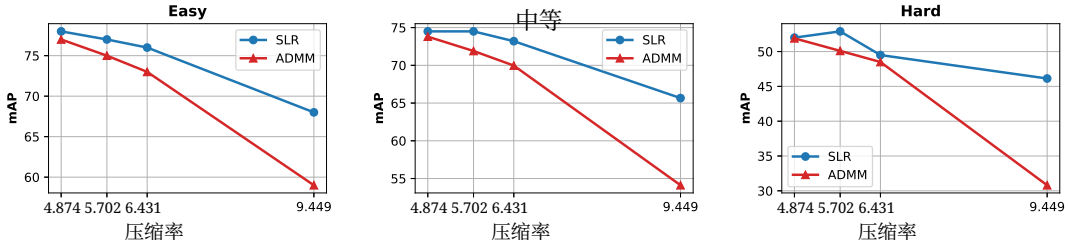
表 6 报告了在 TuSimple 车道线检测基准上进行车道线检测任务的结果，经过 40 轮训练和 5 轮掩码重新训练后获得。我们在 8 种不同的压缩率下进行了实验。图 5展示了随着压缩率增加，硬剪枝后 ADMM 与 SLR 方法之间的准确率差距。

从图 5可以看出，对于较小的压缩率，例如 1.82×，SLR 在硬剪枝准确率方面相较于 ADMM 优势不大。然而，随着压缩率提高，SLR 的表现开始优于 ADMM。例如，SLR 在压缩率为 77.67× 时仍能经受住压缩，准确率仅略有下降，硬剪枝后的准确率达到 89.72%，而 ADMM 的准确率则降至 46.09%。这表明，基于 SLR 的训练方法相较于 ADMM 优势更大，尤其是在压缩率较高时，因为它能够以较小的准确率损失实现压缩，并缩短硬剪枝后重新训练所需的时间。

最后，在图6中，我们展示了使用 SLR 和交替方向乘子法剪枝前后某一层权重的差异。在图6a中，我们展示了初始（未剪枝）权重，以及在相同比率（77×）下使用 SLR 和交替方向乘子法得到的权重稀疏度。初始时，该层的稀疏度较低。经过 SLR 和交替方向乘子法训练后，

表 7. 在不同任务难度等级和压缩率下，交替方向乘子法与 SLR 在 KITTI 上的 PointPillars 模型结果。

等级	原始平均精度均值	压缩	ADMM		SLR	
			硬剪枝后	重新训练后	Hardprune之后	免于重新训练后
Easy	80.7	4.874×	77.0	81.5	77.8	82.2
		5.702×	74.7	74.7	77.3	79.3
		6.431×	72.9	77.5	76.6	75.3
		9.449×	58.6	70.9	68.0	79.9
中等	78.5	4.874×	73.8	77.1	74.5	78.4
		5.702×	71.9	71.0	74.5	75.3
		6.431×	69.9	73.2	73.2	72.7
		9.449×	54.1	66.8	65.6	75.8
Hard	60.7	4.874×	51.9	51.2	52.0	56.4
		5.702×	50.1	50.2	52.9	51.3
		6.431×	48.5	47.8	49.5	50.7
		9.449×	30.8	35.6	46.1	51.4



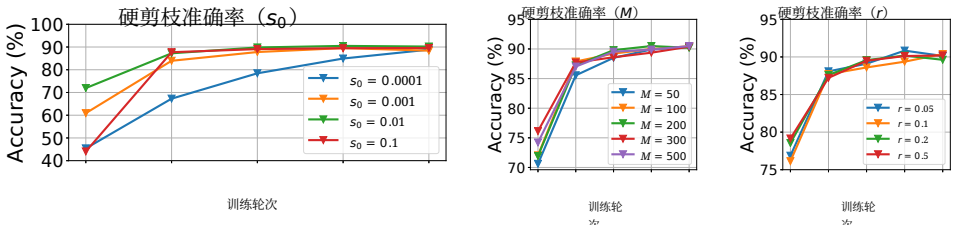
(a) 简单难度任务级别上的剪枝。 (b) 中等任务级别上的剪枝。 （c）在困难任务级别上的剪枝。

图 7。 在不同难度级别下，PointPillars 模型采用不同压缩率进行硬剪枝后的平均精度均值 s.

我们可以看到，权重中被置零的数量有所增加。与交替方向乘子法相比，SLR 更快地趋近目标稀疏度。在图6b中，我们比较了相同准确率（89。0%）下的权重稀疏度。可以观察到，SLR 显著减少了非零权重的数量，而与 SLR 相比，交替方向乘子法仍保留了更多非零权重。

最后，表7 展示了 KITTI 基准上的 3D 点云目标检测模型压缩结果。SLR 和 ADMM 的结果是在经过40 个训练轮次和3 个掩码重新训练轮次后报告的。我们有两个观察结果：首先，随着压缩率的提高，硬剪枝后 SLR 的准确率远高于 ADMM。例如，如图7所示，在 9.44× 压缩率下进行测试时，在“硬”难度层级中，SLR 的平均精度均值比 ADMM 高出超过15% ，如图7c所示。在“简单”和“中等”难度层级中，如图7a 和图7b所示，在 10% 压缩率下，SLR 硬剪枝后的平均精度均值仍比 ADMM 高出超过 9.44× 。其次，我们观察到，由于 SLR 在硬剪枝阶段后的平均精度均值更高，因此重新训练后也能达到显著更高的平均精度均值。在16% 压缩率下，“硬”难度层级中 SLR 和 ADMM 的平均精度均值几乎相差9.44× 。这表明，当重新训练预算有限时，我们的 SLR 方法能够快速恢复模型准确率。

4.5 消融研究



(a) s_0 参数。 (b) M 参数。 (c) r 参数。

图 8. 不同 s_0、 M 和 r取值下，ResNet-18 在 CIFAR-10 上进行 SLR 训练期间的硬剪枝准确率。

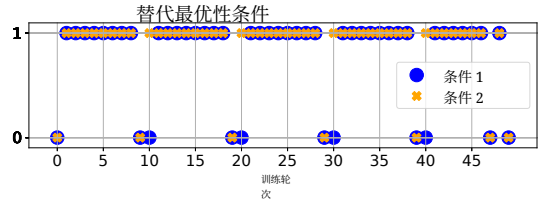


图 9. ResNet-18 在 CIFAR-10 上进行 50 个训练轮次的 SLR 训练期间，代理最优性条件满足情况图（1：满足，0：不满足）。条件会周期性地得到满足。

我们进行了多项实验，以观察 SLR 参数 ρ、 s_0、r以及 M对 ResNet-18 模型（93.33%准确率）在 CIFAR-10 上的影响。我们通过 SLR 训练对模型进行剪枝，训练50 个训练轮次，压缩率为 8.71×，并每隔 10 个训练轮次观察一次硬剪枝准确率。图 8展示了基于 s_0、 M和 r的不同取值进行 SLR 训练时模型的准确率。根据整个训练过程中的硬剪枝准确率可以看出，尽管这些参数对最终结果影响不大，但 s_0 的取值会影响模型的收敛。

从图 8a中，我们可以得出， $s_0 = 10^{-2}$ 具有更高的初始准确率，并且收敛速度很快。图 8b 和图 8c 分别展示了 M 和 r 对硬剪枝准确率的影响。

图 9 表明存在一个迭代 κ （如定理所要求），使得在使用 $s_0 = 10^{-2}$进行训练期间，满足替代最优性条件（SLR方法的高层次收敛准则）， $\rho = 0.1$

这表明总能找到“良好”的乘子更新方向。例如，在第 9 个训练轮次条件被违反后，存在 $\kappa = 10$，使得在第 11 次迭代、经过 $\kappa = 10$ 后，替代条件再次得到满足。

5 结论

本文采用基数函数来引导权重稀疏性，将深度神经网络权重剪枝问题视为非凸优化问题并加以解决。SLR方法将松弛后的权重剪枝问题分解为多个子问题，再通过更新拉格朗日乘子对其进行高效协调，从而实现快速收敛。我们在多个数据集上开展了图像分类、目标检测和分割任务的权重剪枝实验，将SLR方法与交替方向乘子法及其他最先进水平的方法进行了比较。我们发现，在高压缩率下，SLR方法具有显著优势，并且在权重剪枝过程中能够达到更高的准确率。此外，SLR降低了硬剪枝造成的准确率损失，并缩短了重新训练过程。鉴于SLR方法具有有效的优化与协调能力，且在多个实例中展现出明显优势，因此在更广泛的深度神经网络训练应用中具有很大潜力。

REFERENCES

[1] Léon Bottou. 2010. 使用随机梯度下降进行大规模机器学习。载于《COMPSTAT'2010论文集：第19届国际计算统计会议，法国巴黎，2010年8月22日至27日，主题演讲、邀请报告与投稿论文》。施普林格，177–186。[2] S. P. Boyd、N. Parikh 等。2011. 通过乘子交替方向法进行分布式优化与统计学习。《机器学习基金会与趋势》3（2011），34–37。[3] M. A. Bragin、P. B. Luh 等。2015. 替代拉格朗日松弛方法的收敛性。《优化理论与应用杂志》164, 1 (2015), 173–201。[4] Mikhail A. Bragin、Peter B. Luh、Bing Yan、Tongxin Zheng、Dane A Schiro、Feng Zhao 和 Jinye Zhao。2022. 机组组合凸包定价的新型质量度量与高效求解。（2022）。[5] Mikhail A Bragin 和 Emily L Tucker。2022. 用于混合整数线性规划的替代“基于水平”的拉格朗日松弛。《科学报告》12, 1 (2022), 22417。[6] X. Dai、H. Yin 和 N. K. Jha。2017. NeST：基于生长与剪枝范式的神经网络综合工具。arXiv 预印本 arXiv:1711.02017 (2017)。[7] C. Ding、S. Liao、Y. Wang、Z. Li 等。2017. Circnn：使用分块循环权重矩阵加速和压缩深度神经网络。载于 MICRO。395–408。[8] X. Ding、G. Ding 等。2019. 用于剪枝超深度神经网络的全局稀疏动量随机梯度下降。载于 NeurIPS。[9] J. Frankle 和 M. Carbin。2018. 彩票假说：寻找稀疏可训练神经网络。载于 ICLR。[10] Y. Guo、A. Yao 和 Y. Chen。2016. 用于高效深度神经网络的动态网络手术。载于 NeurIPS。1379–1387。[11] Deniz Gurevin、Mikhail A. Bragin、Caiwen Ding、Shanglin Zhou、Lynn Pepin、Bingbing Li 和 Fei Miao。2021. 使用代理拉格朗日松弛实现无需重新训练的深度神经网络剪枝。载于国际人工智能联合会议。[12] S. Han、J. Pool 等。2015. 学习权重与连接，以构建高效神经网络。载于 NeurIPS。1135–1143。[13] K. He、X. Zhang 等。2016. 用于图像识别的深度残差学习。载于 CVPR。770–778。[14] Y. He、J. Lin 等。2018. AMC：面向移动设备上模型压缩与加速的自动机器学习。载于 ECCV。[15] Y. He、X. Zhang 和 J. Sun。2017. 用于加速超深度神经网络的通道剪枝。载于 ICCV。电气电子工程师学会，1398–1406。[16] A. Krizhevsky、I. Sutskever 和 G. E. Hinton。2012. 使用深度卷积神经网络进行 ImageNet 分类。载于 NeurIPS。1097–1105。[17] A. H. Lang、S. Vora 等。2019. Pointpillars：用于基于点云进行目标检测的快速编码器。载于 CVPR。12697–12705。[18] H. Li、A. Kadav 等。2016. 剪枝滤波器以构建高效卷积网络。arXiv 预印本 arXiv:1608.08710 (2016)。[19] Y. Li、S. Gu 等。2020. DHP：通过超网络实现可微元剪枝。ArXiv abs/2003.13683 (2020)。[20] Y. Li、S. Gu 等。2020. 组稀疏性：网络压缩中滤波器剪枝与分解之间的桥梁。CVPR (2020)。[21] Y. Li、S. Lin 等。2019. 利用核稀疏性与熵实现可解释卷积神经网络压缩。CVPR (2019)。[22] Z. Li、Y. Gong 等。2020. SS-Auto：一种用于深度神经网络的单次自动结构化权重剪枝框架，效率极高。arXiv 预印本 arXiv:2001.08839 (2020)。[23] S. Lin、R. Ji 等。2019. 通过生成对抗学习实现最优结构化卷积神经网络剪枝。CVPR (2019)。[24] Ze Liu、Yutong Lin、Yue Cao、Han Hu、Yixuan Wei、Zheng Zhang、Stephen Lin 和 Baining Guo。2021. Swin Transformer：使用移位窗口的分层视觉变换器。arXiv 预印本 arXiv:2103.14030 (2021)。[25] Z. Liu、M. Sun 等。2019. 重新思考网络剪枝的价值。载于 ICLR。

[26] C. Louizos、M. Welling 和D.P.Kingma。2018 年。通过 L0 正则化学习稀疏神经网络。载于 ICLR。https://openreview.net/forum?id=H1Y8hhg0b[27] J. Luo、J.Wu 和W.Lin。2017年。ThiNet：一种用于深度神经网络压缩的滤波器级剪枝方法。载于 ICCV。电气电子工程师学会，5068–5076。[28] D. Molchanov、A.Ashukha和D. Vetrov。2017 年。变分丢弃使深度神经网络稀疏化。载于 ICML。2498–2507。[29] W. Niu、X. Ma 等。2020年。Patdnn：通过基于模式的权重剪枝在移动设备上实现实时深度神经网络执行。载于ASPLOS 论文集。907–922。[30] E.Park、J. Ahn和 S. Yoo。2017 年。基于加权熵的深度神经网络量化。载于CVPR。[31] Z. Qin、H.Wang 和 X.Li。2020 年。超快速结构感知深度车道线检测。arXiv预印本 arXiv:2004.11757（2020）。[32] J.Redmon和 A. Farhadi。2018 年。YOLOv3：渐进式改进。CoRRabs/1804.02767（2018）。arXiv:1804.02767 http://arxiv.org/abs/ 1804.02767[33] Ao Ren、TianyunZhang 等。2019年。Admm-nn：使用交替方向乘子法的深度神经网络算法-硬件协同设计框架。载于ASPLOS。925–938。[34] M. Sandler、A.G.Howard 等。2018 年。MobileNetV2：倒置残差与线性瓶颈。CVPR（2018）。[35] K. Simonyan和A. Zisserman。2014年。用于大规模图像识别的超深卷积网络。arXiv预印本 arXiv:1409.1556（2014）。[36] Ilya Tolstikhin、Neil Houlsby、Alexander Kolesnikov、Lucas Beyer、Xiaohua Zhai、Thomas Unterthiner、Jessica Yung、Daniel Keysers、Jakob Uszkoreit、Mario Lucic 等。2021 年。MLP-Mixer：一种用于视觉任务的全 MLP 架构。arXiv预印本 arXiv:2105.01601（2021）。[37] F. Tung、S. Muralidharan 和G. Mori。2017年。Fine-pruning：通过贝叶斯优化联合对卷积网络进行微调和压缩。arXiv预印本 arXiv:1707.09102（2017）。[38] W.Wen、C. Wu 等。2016 年。在深度神经网络中学习结构化稀疏性。载于 NeurIPS。2074–2082。[39] Ross Wightman。2019 年。PyTorch Image Models。https://github.com/rwightman/pytorch-image-models。https://doi.org/10.5281/zenodo.4414861[40] T.Yang、Y.Chen和 V. Sze。2016 年。使用能耗感知剪枝设计节能卷积神经网络。arXiv预印本 arXiv:1611.05128（2016）。[41] R.Yu、A. Li等。2018 年。NISP：使用神经元重要性分数传播进行网络剪枝。CVPR（2018）。[42] T. Zhang、S.Ye等。2018 年。使用交替方向乘子法的系统化深度神经网络权重剪枝框架。载于 ECCV。184–199。[43] C. Zhao、B. Ni 等。2019 年。变分卷积神经网络剪枝。2019 CVPR（2019）。[44] Shanglin Zhou、Mimi Xie、Yufang Jin、Fei Miao 和 Caiwen Ding。2021 年。使用自动驾驶嵌入式 GPU 的端到端多任务目标检测。载于2021 年第22 届国际电子设计质量研讨会（ISQED）。电气电子工程师学会，122–128。