ISSN 1004-9037,CODEN SCYCE4
Journal of Data Acquisition and Processing Vol. 39,No. 3,May2024,pp. 659－667
DOI:10. 16337/j. 1004-9037. 2024. 03. 013

http://sjcj. nuaa. edu. cn
E-mail:sjcj@nuaa. edu. cn
Tel/Fax:＋86-025-84892742

一种基于稀疏优化和Nesterov动量策略的模型剪枝算法

周　强,陈　军,鲍　蕾,陶　卿

(陆军炮兵防空兵学院信息工程系,合肥 230031)

摘　要: 随着深度学习快速发展,模型的参数量和计算复杂度爆炸式增长,在移动终端上部署面临挑战,模型剪枝成为深度学习模型落地应用的关键。目前,基于正则化的剪枝方法通常采用L2正则化并结合基于数量级的重要性标准,是一种经验性的方法,缺乏理论依据,精度难以保证。受Proximal梯度方法求解稀疏优化问题的启发,本文提出一种能够在深度神经网络上直接产生稀疏解的Prox-NAG优化方法,并设计了与之配套的迭代剪枝算法。该方法基于L1正则化,利用Nesterov动量求解优化问题,克服了原有正则化剪枝方法对L2正则化和数量级标准的依赖,是稀疏优化从传统机器学习向深度学习的自然推广。在CIFAR10数据集上对ResNet系列模型进行剪枝实验,实验结果证明Prox-NAG剪枝算法较原有剪枝算法性能有所提升。

关键词: 稀疏;优化;剪枝算法;Proximal梯度方法;Nesterov加速梯度(Nesterov accelerated gradient, NAG)

中图分类号: TP311　　**文献标志码:** A

Model Pruning Algorithm Based on Sparse Optimization and Nesterov Momentum Strategy

ZHOU Qiang, CHEN Jun, BAO Lei, TAO Qing

(Department of Information Engineering, PLA Army Academy of Artillery and Air Defense, Hefei 230031, China)

Abstract: With the rapid development of deep learning, the number of parameters and computational complexity of models have exploded, which pose challenges for deployment on mobile terminals. Model pruning has become the key to the implementation and application of deep learning models. At present, the pruning method based on regularization usually adopts L2 regularization combined with the importance standard based on the order of magnitude. It is an empirical method lacking theoretical basis, and its accuracy is difficult to guarantee. Inspired by the Proximal gradient method for solving sparse optimization problems, we propose a Prox-NAG optimization method that can directly generate sparse solutions on deep neural networks and a corresponding iterative pruning algorithm is designed. This method is based on L1 regularization and uses Nesterov momentum to solve the optimization problem. It overcomes the dependence of the original regularization pruning method on L2 regularization and order of magnitude standards, and is a natural extension of sparse optimization from traditional machine learning to deep learning. Pruning experiments are conducted on the ResNet series models on the CIFAR10 dataset, and

基金项目: 国家自然科学基金(62076252)。
收稿日期: 2023-06-29;**修订日期:** 2023-09-20

the results show that the Prox-NAG pruning algorithm has improved its performance compared to the original pruning algorithm.

Key words: sparse; optimization; pruning algorithm; Proximal gradient method; Nesterov accelerated gradient (NAG)

引　　言

近年来,深度学习快速发展,在图像分类、目标识别、自然语言处理、医学图像判读以及语音识别等领域取得了令人瞩目的成功。但不可避免的是深度学习模型在存储、计算、能耗等方面的巨大开销日益凸显。用于手写字判别的LeNet[1]有6万多个参数,ResNet152[2]有约6 100万个参数,目前的热点应用ChatGPT有超过1 750亿个参数。在物联网时代(Internet of things, IoT),AI算法最重要的部署对象是物联网终端设备,这就要求AI算法必须部署到相应的IoT设备上去。因此,如何对模型进行轻量化处理以适应这些计算资源有限设备变得至关重要。

剪枝是对深度学习模型进行轻量化处理,去除不重要的、冗余的联接,从而获得一个高效的、稀疏的网络,同时保证其精度损失尽可能小,如图1所示。目前,多数剪枝算法采用基于数量级的剪枝方法,即用参数的绝对值大小来度量参数的重要性,剪除绝对值小的参数而保留绝对值大的参数。即便是基于正则化的剪枝,大多也在正则化训练后,以参数的数量级作为重要性评判依据。这种方法经验性地认为参数的重要性等价于数量级的大小,忽略了目标函数的形态,没有充分利用训练数据蕴含的潜在信息。如图2所示,基于数量级剪枝忽略了目标函数的形态,会得到$(\boldsymbol{w}_1^*,0)$的错误解,而$(0,\boldsymbol{w}_2^*)$显然是更优解。

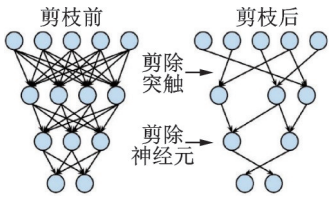

图1　剪枝示意图

Fig.1　Schematic diagram of pruning

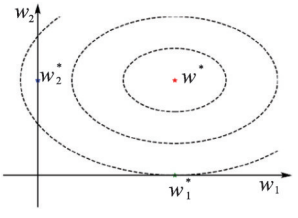

图2　基于数量级剪枝的缺点

Fig.2　Disadvantages of order of magnitude pruning

剪枝本质上可以看成一种稀疏优化。在稀疏优化领域,人们最先采用的是内点法等批处理优化方法,其高精度使得简单的基于数量级的方法就能够产生理想的稀疏效果,并在信号处理等领域获得广泛应用。随着机器学习进入大规模数据时代,以随机梯度下降(Stochastic gradient descent, SGD)为代表的一阶随机梯度方法成为主流,但其内在的低精度使得基于数量级的方法可靠性降低,虽然人们提出了软阈值、截断等算法[3-4],但效果仍然不能同批处理方法相比。2009年,Xiao[5]提出RDA算法;2010年,Duchi等[6]提出COMID(Composite objective mirror descent)算法,它们的共同点是:在优化过程中将正则化项和损失函数分别看待,对损失函数进行线性展开近似,而保持正则化项不变。这种方法被称为Proximal方法,有效兼顾解的稀疏性和精确性。从稀疏优化的角度看,目前主流的基于数量级的正则化剪枝方法类似于截断等算法。受Proximal方法在稀疏优化领域应用的启发,本文在L1正则化剪枝的基础上,引入Proximal梯度方法,使优化器直接产生稀疏解,提高了剪枝精度。

NAG (Nesterov accelerated gradient)[7][$O(1/t^2)$]是一种经典的机器学习优化方法,它采用Nesterov动量策略,较Heavy-ball[8]型动量表现更为出色。在求解光滑凸问题时,它能够将SGD算法的收敛速率从$O(1/t)$加速至最优的$O(1/t^2)$,其中t为迭代次数。文献[9-10]指出,在求解非光滑凸问题时,

Nesterov动量同样具有良好的加速效果。稀疏优化问题是典型的非光滑问题,Nesterov动量策略对非光滑问题能够加速收敛过程,得到更稳定的稀疏解。因此,本文把Nesterov动量策略与Proximal方法相结合,设计了能够快速产生稀疏解的优化算法,不仅适用于剪枝,也适用于其他正则化优化场景。

1　模型剪枝相关工作

1.1　剪枝方法分类

根据剪枝范式,主流的剪枝方法可以分为基于数量级的剪枝、基于正则化的剪枝、基于二阶近似的剪枝3大类:

(1)基于数量级的剪枝。该方法认为参数数量级越大越重要,反之数量级越小越次要,剪除数量级小的参数,配合迭代的调优训练,从而达到较好的训练效果。采用该方法的代表性工作参见文献[11-13]。

(2)基于正则化的剪枝。由于L0和L1正则化具有内在的稀疏属性,使其成为剪枝的重要方法,文献[14-16]采用这些正则化方式。L2正则化不具有稀疏属性,但可以使参数产生不等比例的衰减,重要的参数衰减的少,次要的参数衰减的多;而后,采用基于数量级的剪枝方法,减去数量级小的部分[17-18]。

(3)基于二阶近似的剪枝。该方法通常基于经典的OBD/OBS(Optimal brain damage/Optimal brain surgeon)框架[19],在最优解(待剪枝模型参数)附近对损失函数二阶泰勒展开近似,利用二阶信息可以近似估计显著性。但是对于高维模型来说,海森矩阵的估计代价高昂,人们提出了一些近似的方法,如:K-FAC Fisher[20]、EigenDamage[21]和WoodFisher[22]等。

1.2　剪枝问题描述

剪枝问题可以描述如下:给定一个待剪枝模型$\boldsymbol{w}$,输入剪枝器$P(*)$,得到小模型$\boldsymbol{w}_1$,而后输入调优器$F(*)$,得到最终输出$\boldsymbol{w}_2$。其中,调优器$F(*)$为普通的优化训练;剪枝问题关注的重点是剪枝器$P(*)$,其过程可以细分成两部分:(1)获取一个0/1符号向量$\boldsymbol{M}$,来确定哪些参数被剪除,哪些将被保留;(2)对保留的参数值进行调优。公式表达为

$$\boldsymbol{w}_1=P(\boldsymbol{w})=P_1(\boldsymbol{w})\odot P_2(\boldsymbol{w})=\boldsymbol{M}\odot P_2(\boldsymbol{w}) \tag{1}$$

式中⊙表示对应元素相乘。

基于正则化的剪枝,可以归纳为一个约束优化问题[23],其数学表达式为

$$\min_{\boldsymbol{w}} f(\boldsymbol{w};\boldsymbol{x},y)=L(\boldsymbol{w};\boldsymbol{x},y)+\lambda R(\boldsymbol{w}) \tag{2}$$

式中:f表示组合目标函数;L表示损失函数;R表示正则化约束;λ表示正则化系数;$\boldsymbol{w}$表示模型参数;$\boldsymbol{x}$表示特征向量;y表示标记。如图3所示,对$\boldsymbol{w}$施加稀疏约束($\boldsymbol{w}$的非零分量尽可能少),最自然的是使用L0范数,但L0范数不连续,难以优化求解,因此常使用L1范数来凸松弛近似。

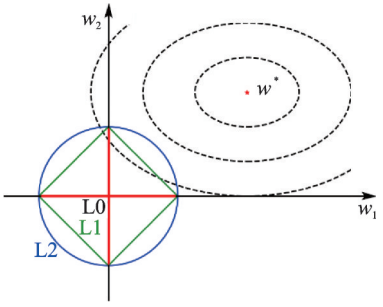

图3　不同正则化约束的形态

Fig.3　Morphology of different regularization constraints

2　Prox-NAG优化算法

对于式(2),本文选用L1正则化,并通过Proximal方法与NAG算法结合,设计了Prox-NAG优化算法,利用优化器直接生成稀疏解。

2.1　Proximal梯度方法

对于上述正则化优化问题,本文用Proximal梯度方法进行求解。把组合目标函数分成两部分:损失函数L形态未知,对其进行泰

勒展开，并添加近端项；正则化项R形态已知，不作近似。在$\boldsymbol{w}_t$点作近似逼近得

$$f(\boldsymbol{w})\cong L(\boldsymbol{w}_t)+\left\langle\nabla L(\boldsymbol{w}_t),\boldsymbol{w}-\boldsymbol{w}_t\right\rangle+\frac{1}{2\alpha_t}\|\boldsymbol{w}-\boldsymbol{w}_t\|_2^2+\lambda_1\|\boldsymbol{w}\|_1 \tag{3}$$

则

$$\boldsymbol{w}_{t+1}=\underset{\boldsymbol{w}}{\operatorname{argmin}}\left\{L(\boldsymbol{w}_t)+\left\langle\nabla L(\boldsymbol{w}_t),\boldsymbol{w}-\boldsymbol{w}_t\right\rangle+\frac{1}{2\alpha_t}\|\boldsymbol{w}-\boldsymbol{w}_t\|_2^2+\lambda_1\|\boldsymbol{w}\|_1\right\} \tag{4}$$

对括号内求导，令导数等于零，求极值点如下：$\nabla L(\boldsymbol{w}_t)\alpha_t+(\boldsymbol{w}-\boldsymbol{w}_t)+\partial\left(\|\boldsymbol{w}\|_1\right)\alpha_t\lambda_1=0$。其中$\partial\left(\|\boldsymbol{w}\|_1\right)$表示$\boldsymbol{w}$的次梯度。为方便描述，引入以下符号：$w_{t,i}$表示第$t$轮迭代时$\boldsymbol{w}$的第$i$个参数分量，其中，$i=1,2,\cdots,d$。由于上式中各维度相互独立，可以将向量转化为标量，对向量$\boldsymbol{w}_{t+1}$的第i维度$w_{t+1,i}$，分类讨论求其闭式解。

当$w_{t+1,i}>0$时，$\partial\left(\|w_{t+1,i}\|_1\right)=1$，上式化简为：$w_{t+1,i}=w_{t,i}-\nabla L\left(w_{t,i}\right)\alpha_t-\alpha_t\lambda_1\left(w_{t,i}>\nabla L\left(w_{t,i}\right)\alpha_t+\alpha_t\lambda_1\right)$；当$w_{t+1,i}<0$时，$\partial\left(\|w_{t+1,i}\|_1\right)=-1$，上式化简为：$w_{t+1,i}=w_{t,i}-\nabla L\left(w_{t,i}\right)\alpha_t+\alpha_t\lambda_1\left(w_{t,i}<\nabla L\left(w_{t,i}\right)\alpha_t-\alpha_t\lambda_1\right)$；当$w_{t+1,i}=0$时，$\partial\left(\|w_{t+1,i}\|_1\right)\in[-1,1]$，上式化简为：$w_{t+1,i}=0$，$\nabla L\left(w_{t,i}\right)\alpha_t-\alpha_t\lambda_1\leqslant w_{t,i}\leqslant\nabla L\left(w_{t,i}\right)\alpha_t+\alpha_t\lambda_1$。其中，$\nabla L\left(w_{t,i}\right)$表示$L$对$w_{t,i}$的偏导数。综上所述，有

$$w_{t+1,i}=\begin{cases}w_{t,i}-\nabla L\left(w_{t,i}\right)\alpha_t-\alpha_t\lambda_1 & w_{t,i}-\nabla L\left(w_{t,i}\right)\alpha_t>\alpha_t\lambda_1\\ 0 & -\alpha_t\lambda_1<w_{t,i}-\nabla L\left(w_{t,i}\right)\alpha_t<\alpha_t\lambda_1\\ w_{t,i}-\nabla L\left(w_{t,i}\right)\alpha_t+\alpha_t\lambda_1 & w_{t,i}-\nabla L\left(w_{t,i}\right)\alpha_t<-\alpha_t\lambda_1\end{cases} \tag{5}$$

由式(5)可得Proximal梯度方法可分两步实现：

第1步：梯度下降

$$\hat{w}_{t+1,i}=w_{t,i}-\nabla L\left(w_{t,i}\right)\alpha_t \tag{6}$$

第2步：计算近端算子

$$w_{t+1,i}=\operatorname{sign}\left(\hat{w}_{t+1,i}\right)*\max\left\{\left|\hat{w}_{t+1,i}\right|-\alpha_t\lambda_1,0\right\} \tag{7}$$

近端算子的函数图形如图4(a)所示，每步迭代都要对参数进行衰减，当$\hat{w}_{t+1,i}$落入阈值范围内时，它能够把阈值范围内的参数直接归零，这也是Prox-NAG算法能够直接产生稀疏解的根本原因。而传统的“黑箱”方法，如图4(b)所示，把损失函数和正则化项不加区别对待，丧失了目标函数本身的稀疏性。

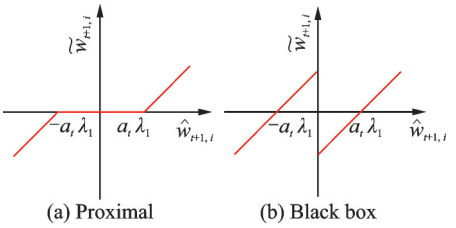

图4 Proximal方法与“黑箱”方法权重衰减对比

Fig.4 Comparison of weight decay between the Proximal and the Black box methods

2.2 NAG方法改进

经典的NAG方法表达如下

$$\tilde{\boldsymbol{w}}_t=\boldsymbol{w}_t+\beta_t(\boldsymbol{w}_t-\boldsymbol{w}_{t-1}) \tag{8}$$

$$\boldsymbol{w}_{t+1}=\underset{\boldsymbol{w}}{\operatorname{argmin}}\left\{L(\tilde{\boldsymbol{w}}_t)+\left\langle\nabla L(\tilde{\boldsymbol{w}}_t),\boldsymbol{w}-\tilde{\boldsymbol{w}}_t\right\rangle+\frac{1}{2\alpha_t}\|\boldsymbol{w}-\tilde{\boldsymbol{w}}_t\|_2^2+\lambda_1\|\boldsymbol{w}\|_1\right\} \tag{9}$$

式中$t=1,2,\cdots;\boldsymbol{w}_0=\boldsymbol{w}_1$。由于在BP算法中，$\nabla L(\tilde{\boldsymbol{w}}_t)$无法计算，不妨调换式(8,9)顺序，并交换$\boldsymbol{w}$、$\tilde{\boldsymbol{w}}$的表示符号，得到

$$\tilde{\boldsymbol{w}}_{t+1}=\underset{\boldsymbol{w}}{\operatorname{argmin}}\left\{L(\boldsymbol{w}_t)+\left\langle\nabla L(\boldsymbol{w}_t),\boldsymbol{w}-\boldsymbol{w}_t\right\rangle+\frac{1}{2\alpha_t}\|\boldsymbol{w}-\boldsymbol{w}_t\|_2^2+\lambda_1\|\boldsymbol{w}\|_1\right\} \tag{10}$$

$$\boldsymbol{w}_{t+1}=\tilde{\boldsymbol{w}}_{t+1}+\beta_t(\tilde{\boldsymbol{w}}_{t+1}-\tilde{\boldsymbol{w}}_t) \tag{11}$$

式中$t=0,1,2,\cdots;\tilde{\boldsymbol{w}}_0=\boldsymbol{w}_1$。式(10)表示进行近端梯度优化,在$\boldsymbol{w}_t$点能够利用BP算法快速求导,并且利用式(5)能够得到稀疏解;式(11)表示进行动量加速。上述两式构成了Prox-NAG的基本公式。需要注意的是,$\tilde{\boldsymbol{w}}_{t+1}$为稀疏解,$\boldsymbol{w}_{t+1}$为普通解。剪枝需要稀疏解,则输出中间量$\tilde{\boldsymbol{w}}_{t+1}$。

2.3　Prox-NAG方法收敛性分析

参照常见的收敛性分析方法,分析Prox-NAG算法在光滑凸情况下的收敛性。假设损失函数$L(\boldsymbol{w})$的定义域P为闭凸集χ的子集,且$\nabla L(\boldsymbol{w})$满足Lipschitz连续,即

$$\left\|\nabla L(\boldsymbol{x})-\nabla L(\boldsymbol{y})\right\|_*\leqslant L\|\boldsymbol{x}-\boldsymbol{y}\|\quad\forall\boldsymbol{x},\boldsymbol{y}\in\chi \tag{12}$$

式中L为大于0的Lipschitz系数,并且动量系数β_t满足如下条件

$$\beta_t=\theta_{t+1}(\theta_t^{-1}-1) \tag{13}$$

$$\theta_{t+1}=\frac{\sqrt{\theta_t^4+4\theta_t^2}-\theta_t^2}{2} \tag{14}$$

式中:$\theta_0=1$;$t=0,1,2,\cdots$。

定理1　令$\{\tilde{\boldsymbol{w}}_t\}$由Prox-NAG算法式(10~14)产生,对于任意$\boldsymbol{w}\in\mathrm{dom}P$,有

$$f(\tilde{\boldsymbol{w}}_t)\leqslant f(\boldsymbol{w})+\frac{2}{(t+2)^2}LD^2\quad\forall t\geqslant 1,\forall\boldsymbol{w}\in\mathrm{dom}P \tag{15}$$

式中D表示定义域内任意两点间距离的最大值。定理1表明Prox-NAG可以达到$O(1/t^2)$的收敛率,较SGD算法$O(1/t)$的收敛率有跨数量级的提升。本定理的证明使用了文献[24]定理1的证明技巧,关键之处在于式(10)与式(11)位置进行调换,证明过程略。

综上所述,Prox-NAG优化算法如下。

算法1　Prox-NAG算法。

输入:原始模型参数$\boldsymbol{w}_0$,学习率η_t,动量参数β,正则化参数λ_1,损失函数$L(\boldsymbol{w}_t)$

(1) $\tilde{\boldsymbol{w}}_0\leftarrow\boldsymbol{w}_0$

(2) For $t=0$ to T

(3)　$\boldsymbol{g}_t\leftarrow\nabla L_t(\boldsymbol{w}_t)$

(4)　$\hat{\boldsymbol{w}}_{t+1}\leftarrow\boldsymbol{w}_t-\eta_t\odot\boldsymbol{g}_t$

(5)　$\tilde{w}_{t+1,i}\leftarrow\mathrm{sign}\left(\hat{w}_{t+1,i}\right)_t*\max\left\{\left|\hat{w}_{t+1,i}\right|-\alpha_t\lambda_1,0\right\}$

(6)　$\boldsymbol{w}_{t+1}\leftarrow\tilde{\boldsymbol{w}}_{t+1}+\beta(\tilde{\boldsymbol{w}}_{t+1}-\tilde{\boldsymbol{w}}_t)$

(7) End for

输出:$\tilde{\boldsymbol{w}}_{T+1}$

3　剪枝算法设计

Prox-NAG优化器能够直接产生稀疏解,但稀疏度(Sparsity)与正则化系数隐式相关,难以通过预先设定正则化系数得到目标稀疏度。本文设计了一种迭代算法:给定预训练模型,首先进行剪枝训练,

达到一定程度后，进行调优训练；如果稀疏度没有达到预期目标，则增大正则化系数，返回重新进行剪枝训练。循环往复，直至达到目标稀疏度，如图5所示。

3.1 剪枝训练

使用Prox-NAG方法进行剪枝训练有如下特点：随着训练次数增加，稀疏度并非单调上升，而是先上升，到达局部最大值后下降，触底反弹后再次上升。与此同时，目标函数Loss值在稀疏度到达局部最大值前上升缓慢，而后急剧上升，如图6所示。

经实验，在稀疏度的局部最大值处能够取得最优模型。该模型性能优于收敛处模型，且优于同等条件下基于数量级方法。其背后的机理有待研究，一种可能的解释是：当施加L1扰动后，模型参数的L1范数随训练次数增加而减小，稀疏度的局部最大值对应着“最优盆地”边缘；当继续训练时，模型跳出“最优盆地”，参数重新初始化，此时Loss值剧烈上升，稀疏度下降；最后，优化器找到新的“局部最优盆地”，稀疏度重新上升。

设定剪枝训练的终止条件为稀疏度达到局部最大值或Loss值开始剧烈上升。此外，在迭代训练时，需要在剪枝训练前记录模型的符号向量$\boldsymbol{M}$，每次稀疏优化后乘以$\boldsymbol{M}$，以保证每轮“剪枝-调优训练”后稀疏度不降；相当于对上一轮的稀疏模型进一步稀疏优化，从而达到稀疏度更高的模型。

3.2 调优训练

每次剪枝训练结束后，均要对模型进行调优训练。如图7所示，模型每次经过正则化剪枝后，参数数量级衰减严重。调优训练的目的是使重要参数在剪枝后得到充分恢复，特别是迭代训练后期重要参数因正则化约束衰减严重，因此调优训练的次数随迭代过程逐步增加。此外，为保持模型的稀疏度不变，每次调优训练后都乘以$\boldsymbol{M}$。

3.3 正则化系数调整

图8反映的是随着正则化系数λ的增大，稀疏度逐渐增大，而模型精度Acc逐渐降低。在迭代剪枝算法中，如果一轮训练结束后，稀疏度没有达到预期值，则在下一轮训练开始前调整增大正则化系数λ。本文中采用线性增大正则化系数。

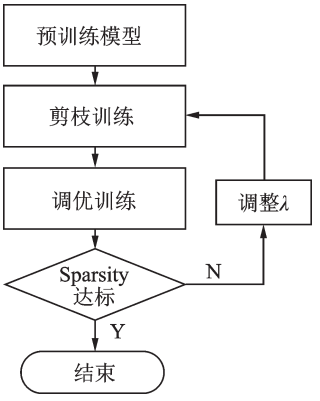

图5 迭代剪枝算法流程图

Fig.5 Flowchart for iterative pruning algorithm

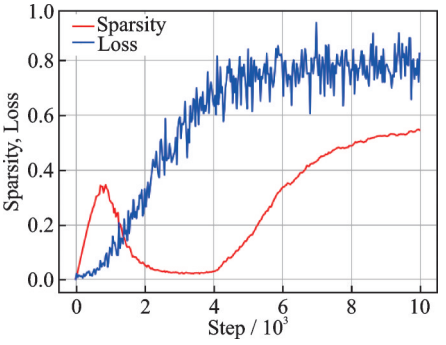

图6 Sparsity和Loss随训练次数增加变化情况

Fig.6 Changes of Sparsity and Loss with training steps

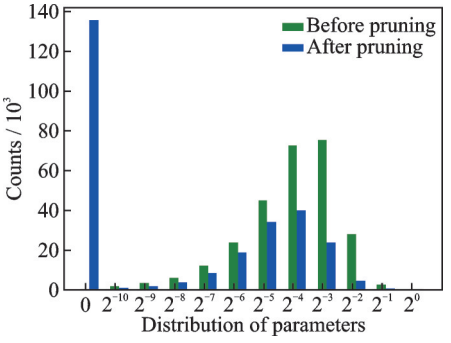

图7 ResNet20剪枝前后参数数量级分布对比图

Fig.7 Comparison of parameter order distribution for ResNet20 before and after pruning

4　实　　验

为了验证了Prox-NAG剪枝算法的效果,本文在CIFAR10数据集上对不同模型进行了实验。和对比算法相比,Prox-NAG剪枝算法性能稳定、表现良好。

4.1　实验设置

实验使用PyTorch平台,硬件使用NVIDIA 3060 GPU,具体设置如下:

(1)数据集。采用CIFAR10数据集[25]进行图像分类任务,该数据集有5万个训练样本,1万个测试样本,样本共分10类。在训练和测试中,实验使用标准的数据增广和预处理技术。

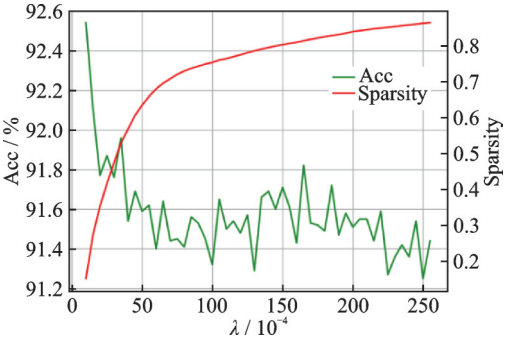

图8　Sparsity和Acc随正则化系数λ变化图

Fig.8　Changes of Sparsity and Acc with λ

(2)模型。为了保证实验的可对比性,本文沿用了与对比方法一致的模型,主要采用ResNet[2]系列模型。Prox-NAG算法可以毫无障碍地扩展到其他模型,如VGG、GoogLenet、Transformer等。本文采用了PyTorch官方[26]的模型训练了一个基准模型,以保证与其他方法比较的公平性。事实上,该方法可以推广到未经训练的随机初始化模型。

(3)对比算法。本文选取了剪枝领域近年来表现突出的几种算法作为比较:Magnitude、L2[17]、SNIP[27]、SM(Sparse momentum)[28]、DSR(Dynamic sparse reparameterization)[29]。

4.2　实验结果

图9展示了不同的剪枝算法在ResNet20模型上的表现。从图9中可以看出,对于较低的剪枝率(比如40%),Prox-NAG剪枝后模型的性能反而有所上升,这种现象可以由正则化可以减轻过拟合解释;对于较高的剪枝率(比如90%),Prox-NAG算法仍然表现出良好的性能。

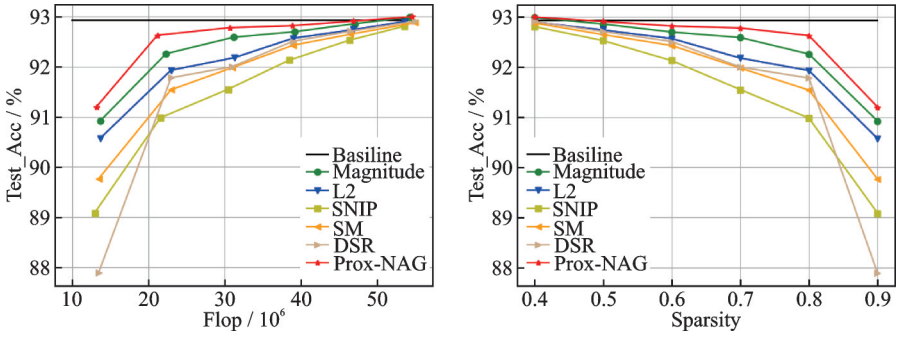

图9　Prox-NAG和常用剪枝算法性能对比

Fig.9　Performance comparison between Prox-NAG and commonly used pruning algorithms

表1展示了对不同的目标剪枝率,Prox-NAG均能达到最优,这表明Prox-NAG算法具有广泛的适用性和良好的性能。

表1 常用方法剪枝后测试精度对比

Table 1 Testing accuracy comparison of common methods after pruning %

剪枝方法	剪枝率40%	剪枝率50%	剪枝率60%	剪枝率70%	剪枝率80%	剪枝率90%
Baesline			92.93			
Magnitude	92.99	92.86	92.70	92.59	92.26	90.92
L2	92.90	92.74	92.57	92.18	91.93	90.57
SNIP	92.81	92.53	92.13	91.55	90.98	89.08
SM	92.88	92.65	92.43	91.98	91.54	89.76
DSR	92.91	92.71	92.51	92.00	91.78	87.88
Prox-NAG	**93.00**	**92.91**	**92.82**	**92.78**	**92.63**	**91.20**

5 结束语

本文提出了一种用于深度学习模型剪枝的Prox-NAG算法,把Proximal梯度方法和Nesterov动量策略引入剪枝领域,实现优化器直接产生稀疏解,提升了剪枝算法的效能。下一步将沿以下两个方向进行探索:(1)由非结构化剪枝向结构化剪枝扩展。非结构化剪枝以单个参数为对象,主要用于模型压缩;结构化剪枝以filter或者channel为对象,主要用于模型加速。下一步,将利用Proximal方法与Group LASSO相结合,实现结构化剪枝的性能提升。(2)利用自适应步长策略提升剪枝算法效能。优化算法的自适应步长策略利用了二阶信息,与Fisher信息阵、Hessian矩阵有着千丝万缕的联系,而利用二阶信息剪枝是一种重要的剪枝范式。未来,拟引入自适应步长策略,把基于正则化的剪枝与基于二阶信息的剪枝有机融合,以提升剪枝效果。

References:

[1] LECUN Y, BERNHARD E B, JOHN S D, et al. Handwritten digit recognition with a back-propagation network[C]// Proceedings of Advance Neural Information Processing System. Vancouver B C, Canada: Chapman Hall/CRC Publishers, 1990: 396-404.

[2] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]//Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas: IEEE Computer Society, 2016.

[3] LANGFORD J, LI L, ZHANG T. Sparse online learning via truncated gradient[J]. Journal of Machine Learning Research, 2009, 10: 777-801.

[4] DUCHI J, ANDSINGER Y. Efficient online and batch learning using forward backward splitting[J]. Journal of Machine Learning Research, 2009, 10: 2873-2898.

[5] XIAO L. Dual averaging methods for regularized stochastic learning and online optimization[J]. Advances in Neural Information Processing Systems, 2010, 11: 2543-2596.

[6] DUCHI J, SHALEV-SHWARTZ S, SINGER Y, et al. Composite objective mirror descent[C]//Proceedings of the 23rd Annual Workshop on Computational Learning Theory. [S.l.]: ACM Press, 2010: 116-128.

[7] NESTEROV Y. A method of solving a convex programming problem with convergence rate $O(1/k_2)$[J]. Soviet Mathematics Doklady, 1983, 27(2): 372-376.

[8] POLYAK B T. Some methods of speeding up the convergence of iteration methods[J]. USSR Computational Mathematics and Mathematical Physics, 1964, 4(5): 1-17.

[9] TAO W, PAN Z S, WU G W, et al. The strength of Nesterov's extrapolation in the individual convergence of nonsmooth optimization[J]. IEEE Transactions on Neural Networks and Learning Systems, 2020, 31: 2557-2568.

[10] 陶蔚,潘志松,储德军,等.使用Nesterov步长策略投影次梯度方法的个体收敛性[J].计算机学报,2018,41(1): 164-176.
TAO Wei, PAN Zhisong, CHU Dejun, et al. The individual convergence of projected subgradient methods using the Nesterov's step-size strategy[J]. Chinese Journal of Computers, 2018, 41(1): 164-176.

[11] ZHU M, GUPTA S. To prune, or not to prune: Exploring the efficacy of pruning for model compression[EB/OL]. (2017-10-05)[2023-03-02].https://doi.org/10.48550/arXiv.1710.01878.

[12] GALE T, ELSEN E, HOOKER S. The state of sparsity in deep neural networks[EB/OL]. (2019-02-25)[2023-03-02].https://doi.org/10.48550/arXiv.1902.09574.
[13] LIN T, STICH S U, BARBA L, et al. Dynamic model pruning with feedback[C]//Proceedings of International Conference on Learning Representations. Addis Ababa, Ethiopia: ICLR, 2020.
[14] LOUIZOS C, WELLING M, KINGMA D P. Learning sparse neural networks through L0 regularization[C]//Proceedings of ICLR. Toulon, France: ICLR, 2017.
[15] LIU Z, LI J, SHEN Z, et al. Learning efficient convolutional networks through network slimming[C]//Proceedings of ICCV. Venice, Italy: IEEE, 2017.
[16] YE J, LU X, LIN Z, et al. Rethinking the smaller-norm-less-informative assumption in channel pruning of convolution layers [C]//Proceedings of ICLR. Vancouver, Canada: ICLR, 2018.
[17] HAN S, POOL J, TRAN J, et al. Learning both weights and connections for efficient neural network[C]//Proceedings of NeurIPS. Venice, Italy: IEEE, 2015.
[18] HAN S, MAO H, DALLY W J. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding[C]//Proceedings of ICLR. Washington D C: ICLR, 2016.
[19] LECUN Y, DENKER J S, SOLLA S A. Optimal brain damage[C]//Proceedings of Advances in Neural Information Processing Systems 2. Morgan-Kaufmann: [s.n.], 1990: 598-605.
[20] MARTENS J, GROSSE R. Optimizing neural networks with Kronecker-factored approximate curvature[C]//Proceedings of ICML. New York: ICML, 2015.
[21] WANG C, GROSSE R, Fidler S, et al. Eigendamage: Structured pruning in the kronecker-factored eigenbasis[C]//Proceedings of ICML. New York: ICML, 2019.
[22] SINGH S P, ALISTARH D. Woodfisher: Efficient second-order approximations for model compression[C]//Proceedings of NeurIPS. San Diego: NIPS, 2020.
[23] 陶卿,高乾坤,姜纪远,等.稀疏学习优化问题的求解综述[J].软件学报,2013,24(11): 2498-2507.
TAO Qing, GAO Qiankun, JIANG Jiyuan, et al. Survey of solving the optimization problems for sparse learning[J]. Journal of Software, 2013,24(11): 2498-2507.
[24] TSENG P. Approximation accuracy, gradient methods, and error bound for structured convex optimization[J]. Math Program, 2010, 125(2): 263-295.
[25] KRIZHEVSKY A. Learning multiple layers of features from tiny images: 2009-1915[R]. USA: AIAA, 2009.
[26] PASZKE A, GROSS S, Massa F, et al. Pytorch: An imperative style, highperformance deep learning library[C]//Proceedings of NeruIPS. San Diego: NIPS, 2019.
[27] LEE N, AJANTHAN T, TORR P H. SNIP: Single-shot network pruning based on connection sensitivity[C]/Proceedings of International Conference on Learning Representations. Washington D C: ICLR, 2019.
[28] DETTMERS T, ZETTLEMOYER L. Sparse networks from scratch: Faster training without losing performance[EB/OL]. (2019-04-01)[2023-03-02]. https://doi.org/doi:10.48550/arXiv.1907.04840.
[29] MOSTAFA H, WANG Xin. Parameter efficient training of deep convolutional neural networksby dynamic sparse reparameterization[C]//Proceedings of ICML. New York: ICML, 2019.

作者简介:

周强(1990-),男,硕士研究生,研究方向:机器学习、数学优化,E-mail:1071391319@qq.com。

陈军(1990-),男,硕士研究生,研究方向:机器学习、数学优化。

鲍蕾(1987-),女,博士,讲师,研究方向:机器学习和计算机视觉。

陶卿(1965-),**通信作者**,男,博士,教授,博士生导师,研究方向:机器学习、模式识别和应用数学。

(编辑:刘彦东)