

深度神经网络学习率策略研究进展

刘云飞, 张俊然[†]

(四川大学 电气工程学院, 成都 610065)

摘要: 学习率 (learning rate, LR) 是深度神经网络 (deep neural networks, DNNs) 能够进行有效训练的重要超参数。然而, 学习率的调整在 DNNs 训练过程中仍存在诸多困难与挑战, 即使以恒定的学习率选择为目标, 为训练 DNNs 选择一个最优的恒定初始学习率也非易事。动态学习率涉及到训练过程的不同阶段, 需对学习率进行多步调整以达到高精度度和快速收敛的目的: 调整过程中学习率过小可能会导致模型收敛缓慢或陷入局部最优值; 而学习率过大则会阻碍收敛, 造成震荡发散。对此, 综述了近年来基于深度学习算法的学习率研究进展, 并对分段衰减学习率、平滑衰减学习率、循环学习率、具有热启动的学习率 4 种类型的学习率簇在几个常见数据集上的性能表现进行测试分析和对比研究, 包括收敛速度、鲁棒性和均值方差等。最后总结全文, 并对该领域仍存在的问题以及未来的研究趋势进行展望。

关键词: 卷积神经网络; 学习率; 深度学习; 模型训练

中图分类号: TP273

文献标志码: A

DOI: 10.13195/j.kzyjc.2022.0147

引用格式: 刘云飞, 张俊然. 深度神经网络学习率策略研究进展[J]. 控制与决策, 2023, 38(9): 2444-2460.

Research advances in deep neural networks learning rate strategies

LIU Yun-fei, ZHANG Jun-ran[†]

(College of Electrical Engineering, Sichuan University, Chengdu 610065, China)

Abstract: Learning rate (LR) is an important hyperparameter for effective training of deep neural networks (DNNs). However, there are still many difficulties and challenges in tuning the learning rate during the training of DNNs, and it is not easy to choose an optimal constant initial learning rate for training DNNs even with the goal of constant learning rate selection. The dynamic learning rate involves multi-step adjustment of the learning rate at different stages of the training process to achieve high accuracy and fast convergence: too small a learning rate in the adjustment process may cause the model to converge slowly or fall into a local optimum; while too large a learning rate may hinder convergence and cause oscillation and scattering. Therefore, we summarize the progress of learning rate research based on deep learning algorithms in recent years, and test and compare the performance of four types of learning rate clusters, including segmented decay learning rate, smooth decay learning rate, cyclic learning rate, and learning rate with hot start, on several common data sets, including convergence speed, robustness, and mean variance, etc. Finally, we summarize the full paper and discuss the remaining problems and future research trends in this field. Finally, we conclude the paper and give an outlook on the remaining problems and future research trends in this field.

Keywords: convolutional neural network; learning rate; deep learning; model training

0 引言

在 DNNs 训练过程中, 随机梯度下降优化器 (stochastic gradient descent, SGD)^[1] 通常会陷入局部极小值区域, 或在附近徘徊, 而不是继续向全局极小值收敛。虽然一些研究建议使用局部极小值, 因为它们较大的网络中容易出现^[2] 且网络输出精度在一些情形下是可以接受的, 但是, 寻求整个网络的

全局最优以及减少收敛时间始终是学界的目标。学习率是避免陷入这种局部极小值的关键和必要因素, 在训练过程中自适应学习率调整策略在许多问题上显示出良好的效果, 自适应学习率调度算法的研究已成为近年来深度学习领域的研究热点。

本文对近年来有重要影响力的学习率调度算法进行全面分析和讨论, 并对各簇学习率算法特性进行

收稿日期: 2022-01-21; 录用日期: 2022-05-31.

基金项目: 智能电网四川省重点实验室应急重点项目 (020IEPG-KL-20YJ01); 德阳科技 (揭榜) 项目 (2021JBZ007); 四川大学华西医院 1-3-5 优秀学科项目 (ZYJC21041); 四川省科技计划项目 (2022YFS0178).

[†]通讯作者. E-mail: junranzhang@scu.edu.cn.

*本文附带电子附录文件, 可登录本刊官网该文“资源附件”区自行下载阅览.

梳理和分类. 本文第1节介绍学习率超参数的概念及其在DNNs训练过程中的作用;第2节总结了学习率调度算法的现有工作并分析其应用统计趋势,以及模型训练过程中面临的问题与挑战;第3节详细介绍分段衰减学习率、平滑衰减学习率、循环学习率、具有热启动的学习率等4种类型的学习率簇,重点内容是各种学习率调度算法在模型训练周期中的演化趋势;第4节首先在三维函数上分析不同学习率机制对模型性能的影响,然后在不同的数据集和DNNs模型上通过实验分析学习率机制对神经网络训练性能的影响;第5节对现有学习率调度算法存在的问题进行分析讨论,并对未来的研究方向和发展趋势作出展望.

1 学习率概述

深度神经网络广泛应用于图像识别^[3]、物体检测^[4]、语音识别^[5]、面部识别^[6]、机器翻译^[7]和无人驾驶^[8]等领域. 评价DNNs优劣的最重要指标之一是训练一个能够达到高测试精度的深度学习模型. 深度神经网络的训练是一个全局优化问题. 在每次迭代中,可以使用损失函数(L_θ)来度量预测值与真实值之间的偏差,并更新模型参数(θ). 梯度下降(gradient descent, GD)是一类流行的非线性优化算法,它通过最小化损失函数迭代更新模型参数^[9]. 梯度下降算法包括SGD^[11]、Momentum^[10]、Adam^[11]等,本文以SGD优化器对DNNs的更新为例. 考虑

$$\theta_k = \theta_{k-1} - \eta_k \cdot \nabla_{\theta_{k-1}} L(\theta_{k-1}). \quad (1)$$

其中: L_θ 是损失函数; ∇L_θ 表示梯度; k 是当前迭代步; η_k 是学习率,其控制模型在迭代步 k 时参数 θ 的更新范围. 在每个迭代步 k 选择一个合适的学习速率是极其困难的. 学习率过小会导致收敛缓慢;而学习率过大会阻碍收敛,导致损失函数震荡,陷入局部最小值,甚至发散^[12]. 鉴于确定通用LR策略的困难性,常值LR通常作为深度学习框架(TensorFlow^[13]、Pytorch^[14])中训练DNNs的默认基线策略. 通过引入动态多步学习率调整策略,研究人员试图通过使用某种类型的退火机制,在DNNs训练的不同阶段动态调整学习率. 然而,良好的LR调度需要适应不同的数据集或不同的神经网络模型的特性^[15]. 经验方法在实践中往往通过试错法找到更好的LR初始值和LR更新时间表. 由于缺乏相关的系统性研究和分析,LR参数搜索空间大,导致手动调整过程成本巨大,极大地影响了DNNs训练的效率和性能. 学习率在DNNs训练过程中起着至关重要的作用,在几乎所有梯度下降算法中,初始学习率的选择以及学习率退火机制始终是效率的核心. Bengio^[16]曾断言,LR“通常是最重要

的超参数”,必须对其进行合理地调整. 这是因为选择跟随梯度信号而不是正确的量,无论是过多还是过少,都会对整个下降过程达到特定目标值的水平面的速度造成很大损失.

本文首先对相关文献的学习率策略进行分析和总结;然后在不同数据集和任务下,全面测试和验证4簇学习率算法的相关指标. 本文选用LR的有效性、鲁棒性、模型训练时间成本作为LR有效性的评价指标,并使用不同的神经网络模型(Multilayer Perceptron^[17]、WideResNet^[18]、EfficientNet^[19]),在三维函数(Rosenbrock function)、CIFAR-10、CIFAR-100、ImageNet数据集上进行全面的实验,以测试不同LR策略对模型精度的影响,以及数据集和DNNs模型自身特性对LR调优的影响. 同时,在本文中使用SGD优化器来探讨不同学习率调度方法在不同数据集上的性能表现,以期更加深刻地理解学习率调度机制的研究与应用.

2 学习率统计分析及应用

2.1 DNNs优化过程

减小步长(学习率衰减)对于SGD的收敛是必不可少的^[1]. 良好的学习率策略会随着时间推移动态改变学习率大小,从而显著加快模型收敛速度,并使损失函数实现最优收敛. 图1中的一维例子可以说明理想化的学习率是如何提高训练过程的,详细过程描述如下.

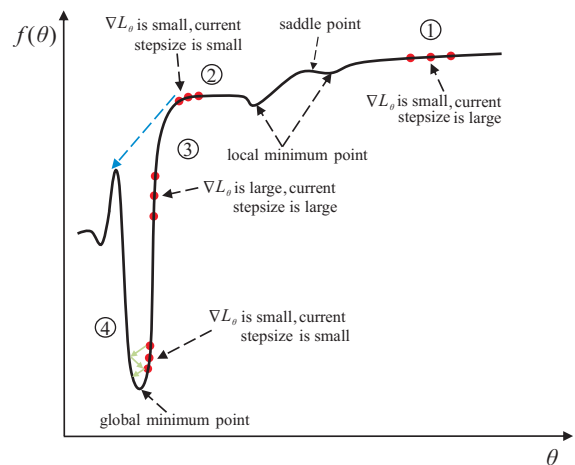


图1 考虑损失函数局部曲率的最优优化过程
(其中 f 表示具有模型 L 的优化任务)

1) 区域①: 损失函数曲面较平坦,梯度接近于0. 此时,SGD应采取较大的更新步长,即采用更大的学习率. 需要注意的是,优化过程中存在大量的局部最小值以及鞍点,过早的衰减学习率将导致搜索过程陷入局部最小值区域并在该处停止,或者导致优化器长期在鞍点附近徘徊,无法摆脱鞍点束缚^[20].

2) 区域②: 损失函数由平向陡, 梯度急剧变化. 在拐点附近, SGD 应采取更为审慎的学习率更新策略, 即采取更小的学习率, 否则极易错过全局最优解, 如图1蓝色虚线所示.

3) 区域③: 损失函数具有大梯度、小曲率. 理想化的优化过程应增加更新步长, 或维持学习率恒定.

4) 区域④: 损失函数处于“狭窄而陡峭”的山谷^[21]中, 恒定的学习率将导致损失函数在最小值附近震荡. 此时, 应逐步衰减更新步长, 使得模型收敛于全局最小值.

2.2 学习率应用统计分析

大规模随机优化可驱动各种机器学习任务. 选择恰当的学习率策略并合理地调整其超参数会大大提高模型的训练速度和最终性能, 这对于相关研究人员来说是一项重要的挑战. 学习率策略的研究已成为近年来深度学习的研究重点 (参见图2), 越来越多的任务列表选择特定的学习率算法控制训练进程. 机器学习研究人员可以从数十种学习率调度策略中选择合适的学习率衰减策略训练模型, 每种方法都有特定的一组可调超参数.

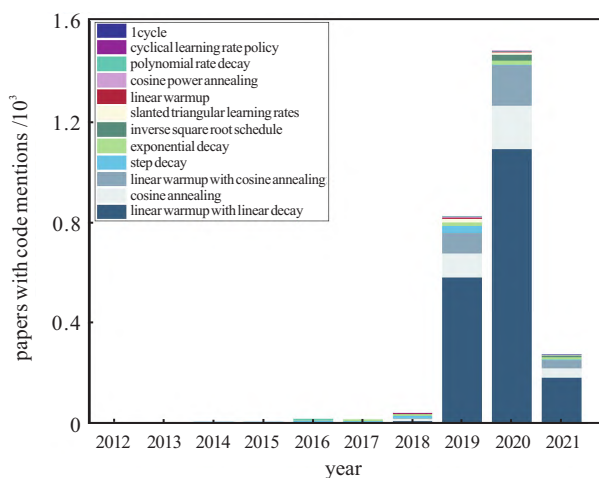


图2 paper with code (<https://paperswithcode.com>) 标题和摘要提及学习率算法的次数统计

对学习率调度机制的有限理论分析显然会偏爱其中一种选择^[22]. 一些研究人员对少量几种流行学习率算法进行了实证比较^[23-24], 但是, 对于大多数学习率算法而言, 引入相关方法的原始工作仅提供了正式的经验评估. 在很多情况下, 对各种新的学习率算法在公开数据集下进行测试和验证能够较快理解该算法的性能表现和应用场景. 进行客观基准测试的关键在于如何通过理论和实验分析各种学习率调度算法对优化器的潜在影响, 在不同任务范式下测试各种学习率调度算法的实际性能表现, 以及调整每种方法的参数并重复每个实验的高资源和时间成

本. Wu等^[25]以ResNet和LeNet为基础模型, 探讨了不同参数下部分分段衰减 (piecewise decay)、平滑衰减 (smooth decay) 和循环衰减 (cyclical decay) 学习率策略在CIFAR-10和MNIST数据集下的性能表现. Retsinas等^[26]提出了类似于在线搜索的可训练的学习率调整策略, 通过考虑神经网络的损失函数结构, 其学习率在一个额外的梯度下降步骤进行优化. Senior等^[24]在语义识别数据集上使用不同优化器, 对指数衰减学习率 (exponential decay^[27-28])、常值学习率 (constant)、多项式衰减 (polynomial decay^[29]) 和损失衰减学习率 (loss decay^[30-31]) 这4种学习率策略进行了简单综述. 已有研究综述缺乏相关的系统研究和分析, 在类似应用上虽有一定借鉴作用, 但场景变化后LR参数搜索空间大, 往往导致手工调优过程成本巨大, 严重影响了DNNs训练的效率和性能.

本文侧重于更全面的学习率策略研究比较, 结合已有学习率调度算法在不同数据集下的最优方案, 在不同数据集和模型下对不同学习率算法进行综合分析.

2.3 学习率调度算法的应用挑战

基于梯度的学习率优化算法不能保证良好的收敛性, 存在以下需要解决的挑战:

1) 选择合适的学习速率比较困难. 学习率太小会导致收敛缓慢, 而学习率太大会阻碍收敛, 并使损失函数在 (局部) 最小值附近震荡甚至发散.

2) 域适应的学习速率表^[1]定义困难. 学习速率表需要根据预定义阈值对学习率进行动态调节^[32], 存在经验规则预定义困难以及无法适应数据集特征的情况^[30].

3) 适应参数特性的差异化学习率设置困难. 数据稀疏性或模型权重更新频率的差异对学习率更新步长和频率的需求存在明显不同.

4) 如何避免陷入次优局部最小值. 对于广泛存在的局部最小值^[2], 通过启发式学习率机制可能收敛到更优解. 同时, Dauphin等^[33]认为鞍点的存在将导致优化算法很难逃脱鞍点的束缚, 如图3所示.

3 常见学习率调度算法

学习率作为一个全局超参数, 其确定了优化器沿损失函数表面梯度方向直到 (局部/全局) 最小值时采取的步长大小. 较小的学习率将减慢训练速度, 当损失值持续恶化或剧烈波动时更可取, 但较小的学习率也可能导致训练陷入局部最小值, 从而阻止模型的参数更新, 导致准确性降低. 与之相对, 较大的学习率将加快训练进度, 并且在损失函数持续降低的多次连续

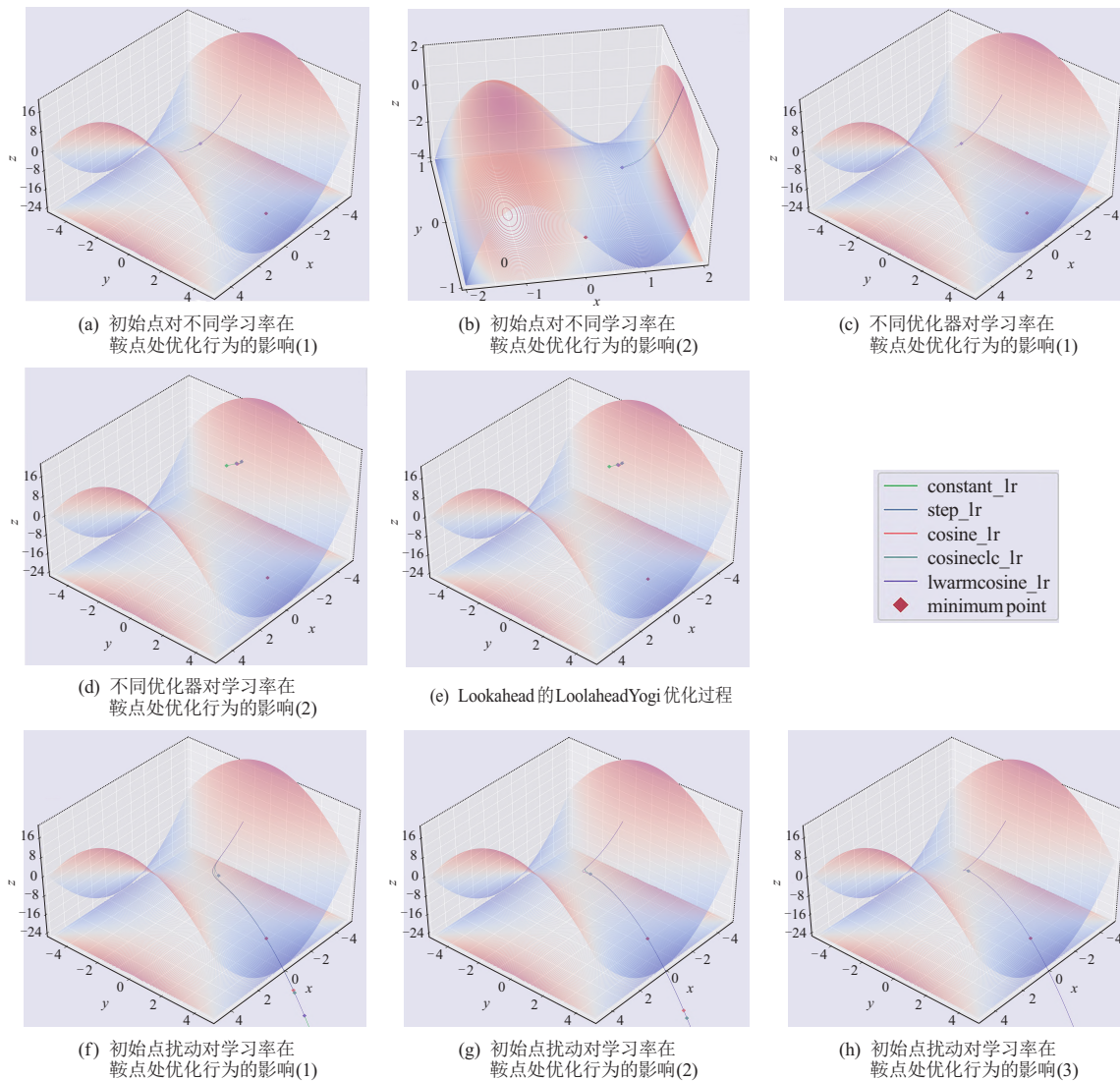


图3 SGD在鞍点处的优化行为

迭代中非常有用,但较大的学习率也可能导致训练无法收敛.理想情况下,研究人员希望有一个合适的学习率策略,该策略逐步优化模型并在训练结束时达到较高的准确性.然而找到最佳学习率策略是极其困难的,通常需要反复试错以找到最佳方案.学习率调度算法经历了人工经验调整到策略调整的发展过程,经验调整通常根据设置的初始学习率来观察网络初始阶段的损失函数变化情况,策略调整包括固定策略和自适应策略.文献[24-25,34-35]在不同任务范式上对少量学习率策略的选择进行了详细研究.在下文中,本文将概述在深度学习领域广泛使用的一些学习率机制以应对上述挑战,同时将SGD作为本文的基准优化器.从理论上讲,进度表可以应用于优化算法的所有超参数,但就本文而言,仅将进度表应用于学习率.

下面给出本文的符号说明.

η_0 : 初始化学习率;

Δt : epoch (iteration) 间隔;

α : 衰减因子;

T : 模型训练时长;

k, t : 第 k, t 个 epoch (或 iteration);

δ : 损失函数在 Δt 个 epoch (iteration) 中累计变化量阈值;

λ : 局部曲率统计信息;

β : 服从 0 均值正态分布噪声方差系数;

$\lfloor \cdot \rfloor$: 向下取整.

3.1 常值学习率

在常值学习率 (constant LR) 中,只有一个初始学习率超参数 η_0 ,其值通常是 (0, 1) 范围内的常数.使用非常小的常值学习率 (Mnist 数据集: $\eta_0 = 0.01$, CIFAR-10 数据集: $\eta_0 = 0.001$) 将是不错的默认策略,如图4所示.尽管使用较小的常值学习率是避免糟糕学习率策略的保守方法,但是,它具有收敛速度慢、训练时间长以及无法保证高精度的明显缺陷^[25].另

表2 分段学习率

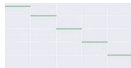
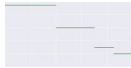


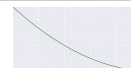
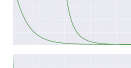




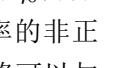
category	ref.	illustration	schedules
piecewise decay	step ^[40]		$\eta_k = \eta_0 \cdot \alpha^{\lfloor \frac{k}{\Delta t} \rfloor}, k \in [0, T]$
	multi-step ^[41]		$\eta_k = \begin{cases} \eta_0, & k \in [0, t_0) \\ \eta_0 \cdot \alpha_1, & k \in [t_0, t_1) \\ \vdots \\ \eta_0 \cdot \alpha_1 \cdots \alpha_{n-1}, & k \in [t_{n-1}, T] \end{cases}$
	linear ^[42-43]		$\eta_k = \begin{cases} \left(1 - \frac{k}{t_1}\right) \cdot \eta_0 + \frac{k}{t_1} \cdot \eta_1, & k \in [0, t_1) \\ \eta_1, & k \in [t_1, T] \end{cases}$
	loss ^[30-31]		$\eta_k = \begin{cases} \eta_0, & \sum_{\Delta t} L(\cdot) > \delta \\ \eta_0 \cdot \alpha, & \text{else} \\ \vdots \end{cases}$

表3 平滑学习率

category	ref.	illustration	schedules
smooth decay	polynomial ^[29]		$\eta_k = \eta_0 \cdot \left(1 - \frac{k}{T}\right)^\alpha$
	exponential ^[27-28]		$\eta_k = \begin{cases} \eta_0 \cdot \alpha^{\lfloor \frac{k}{\Delta t} \rfloor}, & \text{or} \\ \eta_0 \cdot \alpha^{\lfloor \frac{\max(k, t_1) - k}{\Delta t} \rfloor} \end{cases}$
	inverse time ^[38]		$\eta_k = \eta_0 (1 + \alpha \cdot k)^{-1}$
	linear cosine(noisy) ^[49]		$\eta_k = \begin{cases} \text{ld} \cdot \text{cd}, & \text{without noise} \\ (\text{ld} + N(\cdot)) \cdot \text{cd} + 0.001 \end{cases}$
	cosine ^[23]		$\eta_k = \eta_{\min} + \frac{1}{2}(\eta_{\max} - \eta_{\min}) \cdot \left(1 + \cos\left(\pi \frac{k}{T}\right)\right)$
	cosine power ^[50]		$\eta_k = \eta_{\min} + (\eta_{\max} - \eta_{\min})p(\cdot)$
	HTD ^[35]		$\eta_k = \eta_{\min} + \frac{1}{2}(\eta_{\max} - \eta_{\min}) \cdot (1 - \tanh(\cdot))$

过高的初始学习率,确保学习率从预定义的 η_0 开始逐渐减小. 当在模型中加入具有较小学习率的非正则化偏置项时, inverse time decay 学习率策略可以与不可微铰链(hinge)损失函数很好地工作^[37].

相对而言,较大的初始化学学习率可以允许使用更大的批处理量^[46-47],实现更快的收敛和提高模型泛化能力^[48]. Bello等^[49]提出了线性余弦衰减(linear cosine decay)学习率策略(具有噪声(with noise)),在训练DNNs时,可以允许更大的初始学习率. 其中:线性衰减 $\text{ld} = 1 - k/T$,余弦衰减 $\text{cd} = \eta_0 \cdot \left(1 + \cos\left(2\pi n \frac{k}{T}\right)\right)$, $N\left(0, \frac{1}{(1+k)^{0.55}}\right)$ 表示高斯函数. Loshchilov^[23]指出,在大型数据集上训练DNNs是主要的计算瓶颈,即便在高性能GPU上其过程通常需要几天的时间,并且任何提速都将具有实质性的价值. Loshchilov等^[23]使用SGDR学习率策略,与当前使用的学习率进度表方案相比,通常需要减少2~4个epochs的训练周期,并可以达到可比甚至更好的实

验结果. 然而采用余弦机制时,学习率在训练早期衰减相当缓慢,在训练末期进行相对快速的衰减. 对于占据绝大多数时间的模型训练中期而言,这将造成学习率衰减的相对不平衡. Hundt等^[50]将exponential decay引入余弦衰减,以改善训练中期学习率在余弦衰减中变换缓慢的问题,其exponential decay部分可以表示为

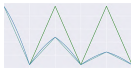
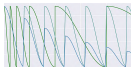
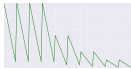
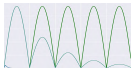
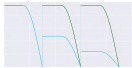
$$p(\cdot) = \frac{\alpha^{\frac{1}{2} \cdot (1 + \cos(\pi \cdot \frac{k}{T})) + 1} - \alpha}{\alpha^2 - \alpha}. \quad (2)$$

Hsueh等^[35]引入上下界边界调控因子(L, U),以改善余弦策略在学习率收敛边界上设置的不灵活性,有

$$\tanh(\cdot) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, x = L + (U - L) \cdot \frac{k}{T}. \quad (3)$$

分段衰减策略通过谨慎地改变学习速率和阶段周期来实现良好的性能,但需要反复实验才能找到一个可以接受的步进衰减调度程序. 如表3所示,平滑衰减的学习率策略在分段衰减的学习率策略的基础

表4 循环学习率

category	ref.	illustration	schedules
	triangular ^[22]		$\eta_k = \eta_0 + (\eta_{\max} - \eta_0) \cdot \max(0, (1 - x)) \cdot f(\cdot)$
	SGDR ^[23]		$\eta_k = \frac{\eta_0}{2} \left(\cos \left(\frac{\pi \cdot \text{mod}(k, \lfloor T/M \rfloor)}{\lfloor T/M \rfloor} \right) + 1 \right)$
cyclical decay	triangle ^[54]		$\eta_k = \eta_{\max} - \text{mod}(k, T) \cdot \eta_{\min}$
	sine ^[25]		$\eta_k = \eta_0 \cdot \left \sin \left(\pi \cdot \frac{k}{2 \cdot T_{\text{stepsize}}} \right) \right (\text{SIN})$
	Ssgdr ^[56]		$\eta_k = \frac{\eta_0}{2} \left(1 + \cos \frac{\pi \cdot k}{T_{\text{stepsize}}} \right)$

上应用连续函数的形式简化参数配置,具有比分段衰减学习率策略更少的超参数,在每次迭代时将学习率降低一个给定的因子,在图像分类等应用中取得了不错的效果^[35],但也存在模型收敛速度缓慢等问题。

3.4 循环学习率

循环学习率(cyclical LR)在每个预定义更新节点上周期性地动态改变LR值,如表4所示.文献[51]表明,循环学习率可以加速DNNs达到一定精确度阈值的进程.本研究主要考虑以下几种循环学习率算法:基于正余弦、基于三角函数及其变体的循环学习率算法.学习率作为调整深度神经网络最重要的超参数,文献[22]提出3种基于三角函数的循环学习率策略(TRI、TRI2、TRIEXP,如表4中triangular^[22]所示).该方法消除了通过实验找到最佳值的需要,从而方便制定适合不同任务范式和模型大小的学习率衰减速率.在训练周期内,学习率不再单调降低,而是使学习率在合理的边界值之间循环变化.triangular^[22]循环学习率函数为

$$f(\cdot) = \begin{cases} 1, & \text{TRI}; \\ \frac{1}{2^{x-1}}, & \text{TRI2}; \\ \alpha^x, & \text{TRIEXP}. \end{cases} \quad (4)$$

其中

$$x = \left\lfloor \frac{k}{\Delta t} - 2 \cdot \left\lfloor 1 + \frac{k}{2 \cdot \Delta t} \right\rfloor + 1 \right\rfloor.$$

Smith^[22,52]引入基于余弦的SGDR学习率重启技术以加速梯度流动,提高模型应对高维数据中常见的漏斗状多峰函数以及病态条件问题的能力^[23].通过定期模拟SGD的热重启,在每次重启中,学习率均初始化为某个阈值并逐渐衰减(重启并不是从头开始训练,而是通过提高学习率来模拟,这种增量学习可以捕获先前获取的信息).对于一类非光滑非强凸优化问题,具有重启的SGD可以实现线性收敛速

度^[53].表4展示了该学习率衰减算法的两种变体,即没有初始学习率衰减的SGDR和具有初始学习率衰减的SGDR方案.与当前已有的学习率衰减机制相比,具有热重启的SGD收敛更为迅速,且具有更高的验证精度^[23].使用具有衰减初始学习率的SGDR时需要乘以衰减因子 $\alpha \left(\left\lfloor \frac{k}{\lfloor T/M \rfloor} \right\rfloor \right)$,其中 M 为循环周期数,控制周期宽度.在三角函数型循环学习率策略的基础上,文献[54]和文献[55]分别提出了基于斜三角和梯形的循环学习率策略;文献[25]提出了基于正弦函数(SIN)的循环学习率策略(SIN2、SINEXP),可视为余弦学习率策略的变体^[23].其中正弦函数的循环学习率公式如下:

$$\begin{aligned} \text{SIN2} &= \frac{1}{2^{\left\lfloor \frac{k}{2 \cdot T_{\text{stepsize}}} \right\rfloor}} \cdot \text{SIN}, \\ \text{SINEXP} &= \alpha^k \text{SIN}. \end{aligned} \quad (5)$$

Yang等^[56]提出了结合step decay的SGDR学习率策略.其中step decay与SGDR周期步长相等(T_{stepsize}).

当学习率足够大时,固有随机扰动会使优化器倾向于沿优化路径收敛于损失函数尖锐的局部最小值;当学习率很小时,模型倾向于收敛到最近的局部最小值.尽管一些研究建议使用局部最小值,因为他们易于在较大的网络中找到^[2],但SGD的两种截然不同的行为在优化阶段都有使用^[30].最初需要保持大的学习率才能进入局部最小值区域的大致范围,当搜索完成时,触发学习率下降,最终收敛到局部最小值.使用周期性学习率进行训练可以提高模型分类精度,加快模型收敛速度^[57-58].通过在每个epochs周期内线性增加学习率,可以估算出适合特定任务和模型的学习率合理界限余量,从而方便地设定循环学习率的上下界.与自适应学习率(依靠局部适应性学习率代替全局学习率,将产生大量的计算成本)不同,循环学习率不需要额外的计算负担,消除了学习率调整

的需要,达到了接近最佳的分类精度. 循环学习率策略的优势在于改进了平滑衰减学习率策略存在的模型收敛缓慢等问题,在大规模图像分类实验中取得了不错的效果,但其需要动态调整循环周期和学习率上下界,存在超参数调整量大、模型收敛不稳定等问题.

3.5 热启动学习率

梯度下降算法广泛应用于深度神经网络. 对于卷积神经网络和递归神经网络,通常在训练开始时设置相对较大的初始学习率,然后随优化过程而衰减^[59]. 事实表明,在某些特定问题上(例如大批次训练时,或训练深层神经网络时),使用学习率预热策略是必不可少的(Warmup 学习率)^[30]. Warmup: 优化过程从使用极小的学习率开始($1e^{-7}$),然后以预定的迭代次数将其逐渐增加到预定的最大值(η_0),之后采用上述提到的学习率衰减策略. Warmup 学习率策略包括以下两种:

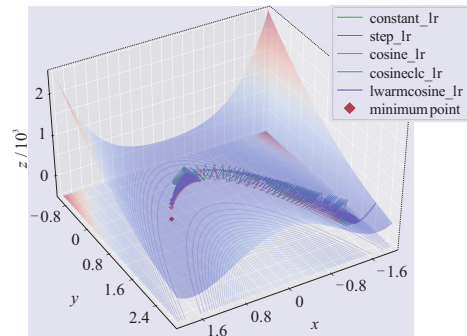
1) 常值热启动(constant warmup). He 等^[30]首次在 ResNet 中使用这一训练技巧,在训练的前几个迭代周期使用较低的恒定学习率(0.01),直到训练误差低于 80% 时(≈ 400 iterations),将学习率切换为初始学习率(0.1). 常值热启动适用于目标检测和实例分割方法,该方法可以同时微调预训练层和新初始化层^[60-63]. 然而,对于大型分布式训练,常值热启动不足以解决这类优化问题,并且从低学习率过渡到初始学习率时会导致训练误差震荡.

2) 线性热启动(linear warmup). 为使模型参数适应任务需求,研究人员希望模型在训练初期对参数分布进行全局感知,然后逐渐完善参数空间建模. 在整个模型训练过程中使用相同学习率或学习率退火策略并不是实现此行为的最佳方案. Goyal 等^[46]提出了常值热启动的替代方法,可以将学习率从小数值逐步提高到初始学习率. 线性热启动避免了学习率的突变,从而允许模型在训练过程中平稳收敛,减少训练初期的波动性. 文献[64]提出将渐变热启动与 inverse square root decay^[28]相结合的学习率策略,在训练初期线性增加学习率,达到初始学习率(η_0)后,按迭代步平方根的反比衰减,其中 $d_{\text{model}} = 512$. Howard 等^[65]提出了斜三角学习率衰减(slanted triangular, ST),首先线性增加学习率,然后根据以下时间表线性衰减学习率:

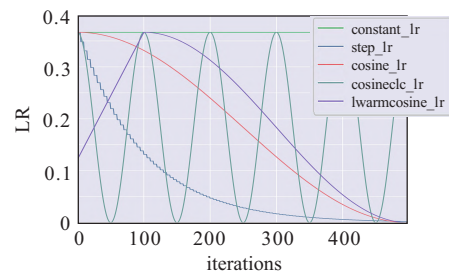
$$\text{cut} = \lfloor T \cdot \text{cut_frac} \rfloor;$$

$$p = \begin{cases} k/\text{cut}, & k < \text{cut}; \\ 1 - \frac{k - \text{cut}}{\text{cut} \cdot (\text{ratio} - 1)}, & \text{otherwise.} \end{cases} \quad (6)$$

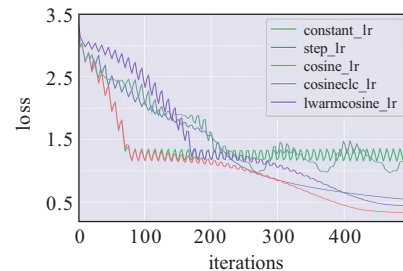
其中: cut_frac 表示线性热启动比例因子(cut_frac = 0.1), ratio 是最低学习率与 η_0 的衰减倍数(ratio = 32). ST 具有短暂的线性学习率增加和较长的衰减周期,在文本分类任务中的表现优于 SGDR^[23]. Xing 等^[66]通过对连续迭代的参数之间的损失面进行插值,并跟踪训练过程中的各项指标,用以研究沿 SGD 轨迹的 DNNs 损失面. 根据 DNNs 损失面的经验观察,每个训练迭代更新前后参数之间的损失插值大致是凸的,推断 SGD 在损失函数谷底以上,通过在峡谷壁之间的跳跃进行参数更新移动,如图 5 所示. 对于这种“跳跃更新机制”,学习率控制距谷底的高度. 由于谷底存在许多障碍(局部最小值),大学习率可以保证 SGD 在谷底之上进行探索,从而快速远离初始点,有利于找到更平坦的区域(泛化更好的局部最小值),这对模型泛化至关重要^[67-70]. 文献[66]使用一种梯形学习率策略(long trapezoid),使 SGD 在小批量和大学习率的情况下更易发现这样的局部最小值^[67,71-73]. Smith 等^[51]针对小样本标签数据集提出了一种超收(super-convergence)训练策略(1cycle),与常规方法相比,利用超收可以更快地训练神经网络,迭



(a) SGD 在 rosenbrock function 中的优化行为



(b) 优化过程中的学习率更新



(c) 优化过程中的损失值变化

图 5 不同学习率策略的优化过程可视化

代次数减少一个数量级,并且最终的测试准确性更高.实现超收的关键要素是训练过程中存在一个学习率周期和一个最大学习率峰值(使用 Hessian-free 估计学习率).较大的学习率可以正则化训练过程,同时需要减小其他正则化措施,以保持最佳的正则化平衡. 1cycle 学习率参数为

$$\begin{cases} \text{ratio}_1 = \frac{k}{\text{step_len}}, \\ \text{ratio}_2 = 1 - \frac{k - \text{step_len}}{\text{step_len}}, \\ \text{ratio}_3 = \frac{k - 2 \cdot \text{step_len}}{T - 2 \cdot \text{step_len}}. \end{cases} \quad (7)$$

其中: $\text{step_len} = \left\lfloor \frac{T}{2} \cdot \left(1 - \frac{\text{prcnt}}{100}\right) \right\rfloor$, prcnt 、 div 为比例因子(均默认为10). Pham 等^[74]提出了结合线性热启动和余弦退火的学习率策略,加速模型收敛. 其中余弦项为

$$\cos(\cdot) = \cos\left(\frac{\pi}{4} \cdot \frac{k - \text{step}}{(T - \text{step})}\right). \quad (8)$$

虽然 Warmup 在各项计算机视觉任务中应用非常广泛,但至今 Warmup 尚未得到理论证明(Liu 等^[75]的研究表明,学习率预热阶段的有效性来源于自适应优化器 Adam^[11]的学习率方差),目前只能从已有文献上得到如下推测:1) Warmup 有助于减缓模型在训练初期对小批次数据(mini-batch dataset)的提前过拟合现象,保持数据分布平稳^[46];2)有助于维持模型深层权重的稳定性^[76](Warmup 可以增加模型最后几层的相似性,避免 FC 层的剧烈变化^[77]). 大批次训练可能会对模型的准确性产生负面影响^[71,78-80],补偿方法通常是提高学习率来执行更大的更新步长. 然而,使用较大的 LR 会使优化更加困难,尤其是在训练初始阶段(可能导致网络发散)^[76],使用 Warmup 和线性缩放 LR^[46]成为大批次模型并行训练的最优选择. 对 LR 进行线性缩放的关键性假设为 $\nabla L(x, \theta_t) \approx \nabla L(x, \theta_{t+j})$. 然而,在至少两种情况下不满足此式:1) 开始训练阶段,模型权重变化迅速. 训练初期模型对数据分布感知为零,模型会很快地进行数据分布修正,如果此时学习率很大,则极有可能导致模型在训练初期对该数据分布过拟合建模. 训练后期随着 LR 的衰减,模型很难通过挖掘数据潜在信息修正模型的先验权重. Warmup 在模型训练早期阶段使用较小的 LR,允许模型对数据分布进行先验探索,避免训练初期较大的学习率破坏模型数据分布感知能力. 2) mini-batch 样本较小时,样本间方差变化较大. 训练过程中,如果 mini-batch 的数据分布方差变化较大,则导致模型权重剧烈波动,这在训练初期最为明显. 训练

后期,随着模型对数据空间分布感知的逐渐完备将得到部分缓解.

热启动学习率策略通过在模型训练初期维持较低的学习率,从而引导模型对数据分布进行感知,改进了分段衰减学习率、平滑衰减学习率和循环学习率模型训练前期存在的因参数更新剧烈变化而极易导致模型训练不稳定,模型权重剧烈波动等问题,在目标检测、实例分割等应用中取得了良好的效果,是 open-mmlab^[81-82]等算法库中优先考虑采用的策略,其主要不足在于最佳热启动周期确定困难,缺乏理论证明^[77,83-84].

4 实验结果与分析

为了系统性地研究学习率调整策略在低维数据中面对局部最小值、鞍点等情形时的行为,以及数据集、任务和 DNNs 模型对各种 LR 策略的效用、成本和鲁棒性等指标的影响,在仅更改 LR 的默认设置,保留其他超参数默认值不变的情况下,本文选用不同的数据集(CIFAR、ImageNet)和网络模型(ResNet^[30]、EfficientNet^[19]),对上述 4 簇学习率算法进行测试和对比. 特别需要注意的是,对于学习率的选择,使用 Hyperopt^[85]工具箱进行随机搜索,选择当前采样频率和范围内的最优初始学习率. 所有实验均在配有 2 块 GeForce RTX 1080 的 Intel(R) Core(TM) i9-9900K CPU @ 3.60 GHz 的服务器上进行,该服务器的环境配置为 Ubuntu 20.04, CUDA 10.1, CUDNN 7.6.5.

4.1 3D 函数优化过程可视化

本文在常值学习率(constant, 表示为 constant_lr), 分段衰减学习率(选择 step decay, 表示为 step_lr), 平滑衰减学习率(选择 cosine decay, 表示为 cosine_lr), 循环学习率(选择 SGDR, 表示为 cosineclr_lr), 热启动(选择 linear warmup cosine decay, 表示为 lwarmcosine_lr)簇中各选一类典型 LR 策略进行测试,并利用三维可视化图形来直观说明学习率策略对优化器行为的直观影响.

4.1.1 峡谷

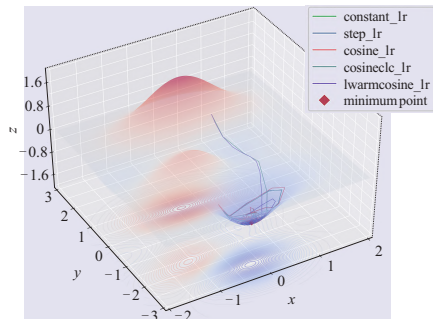
在相同的初始学习率下(η_0),图5分别可视化了 SGD(without momentum)在不同的 LR 策略下穿越形如狭长山谷的损失表面(rosenbrock function—https://en.wikipedia.org/wiki/Test_functions_for_optimization)的跳跃优化行为^[48]. constant_lr 极易造成 SGD 卡在峡谷两侧从而导致优化失败,表现为 loss 值来回震荡,如图 5(c) 所示. step_lr 策略在大约 300 次迭代时能达到与 cosine_lr 相似的优化点,然而收敛速度比 cosine_lr 慢. 同时也应该注意到,在 300 次迭代

时, `cosine_lr` 学习率远比 `step_lr` 大(如图 5(b) 所示), 在训练后期, 由于 `step_lr` 学习率过小, 进一步优化将十分困难. `lwarmcosine_lr` 存在相反的现象, 在大约 200 次迭代时, `lwarmcosine_lr` 收敛到与 `cosine_lr` 相似的区域, 但 `lwarmcosine_lr` 学习率高于 `cosine_lr`, 过高的学习率在短期内快速衰减会导致优化效果不佳. `cosineclr` 在训练初期更新步长最大, 其优化路径距谷底最高, 但其优化结果只稍微好于 `constant_lr`, 推测是由于学习率在一个方向上衰减过快使得优化器更新不及时所导致的(如图 5(a) 所示). 余弦策略在精度和速度上均达到最优. 针对峡谷优化的情形, 5 类学习率可归纳出以下特点: 1) 学习率衰减对于模型优化至关重要, 如 `step_lr`、`cosine_lr` 和 `lwarmcosine_lr` 相较于常值学习率通常会收敛到最优点附近; 2) 模型优化过程中, 不宜过快衰减学习速率, 如 `step_lr` 因过快衰减学习率而导致过早结束优化过程; 3) 应用循环学习率时, 不易将学习率循环周期设置过小, 在模型训练后期可使用较大的循环周期, 以避免学习率在一个方向上过快衰减使得优化器更新不及时而导致优化失败.

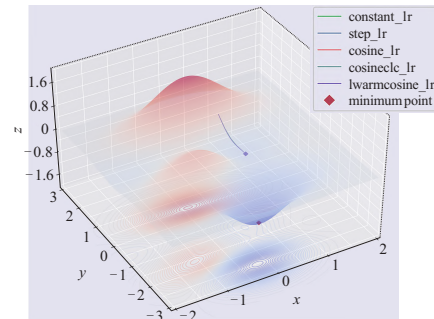
4.1.2 局部极小值

在 Peaks 函数下, 图 6 研究了初始点、学习率策略、`momentum` 对 SGD 行为的影响. 首先注意到

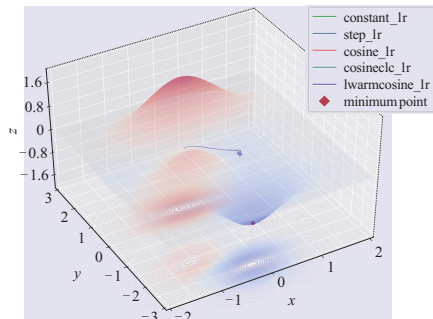
`momentum` 参数对 SGD 至关重要(见图 6(a)), 没有 `momentum` 的 SGD 无法摆脱局部极小值(见图 6(b)), 与学习率策略的选择无关. 其次, 能够发现当 `momentum` 对 SGD 的影响处于临界值时, `constant_lr` 和 `cosine_lr` 学习率策略更易摆脱局部极小值, 相反, `step_lr`、`cosineclr` 和 `lwarmcosine_lr` 往往无法逃离局部极小值(见图 6(d)). 最后, 还应注意到初始点同样对 SGD 有很大影响(见图 6(c)), 当全局最小值与 `momentum` 方向不一致时, `momentum` 和学习率策略均失效. 对于广泛存在的局部极小值问题, 通常可以先使用自适应优化器寻优^[86-92](<https://github.com/jettify/pytorch-optimizer>), 快速收敛到全局收敛域附近, 再切换回 SGD^[93]. 针对局部极小值的情况, 5 类学习率策略可以归纳出以下特点: 对于具有临界 `momentum` 值的 SGD, 维持较大的学习速率(如 `constant_lr` 和 `cosine_lr`), 可以帮助模型快速逃离局部最小值区域; 相反, 快速衰减学习率(`step_lr`), 在局部极小值附近的线性热启动(`lwarmcosine_lr`) 以及具有快速周期变化的循环学习率策略更容易导致优化器收敛于局部最小值区域, 推测是由于 `momentum` 和较小学习率的加性作用不足以使模型逃脱局部最小值区域的束缚.



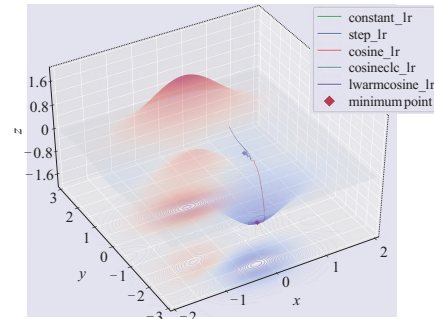
(a) 具有 `momentum` 参数



(b) 没有 `momentum` 参数



(c) 初始点对不同学习率优化过程的影响 (1)



(d) 初始点对不同学习率优化过程的影响 (2)

图 6 局部极小值附近的 SGD 优化行为

4.1.3 鞍点

图 3 展示了优化器在鞍点(一个维度具有正斜率, 而另一个维度具有负斜率的点)处的行为, 鞍

点的存在可能给优化过程带来困难^[33]. 在 saddle 1 ($\text{saddle 1: } f(x, y) = x^2 - y^2$) 和 saddle 2 ($\text{saddle 2: } f(x, y) = x^3 - 3x - 2y^2$) 函数中, 当初始点 ($x = -3, y = 0$)

或 $x = 2, y = 0$)与鞍点处于同一纬度时,SGD很难摆脱鞍点的束缚,最终会收敛到鞍点同平面附近,如图3(a)、(b)所示. 本文进一步测试了目前已知的所有基于SGD和基于Adam^[11]的自适应优化算法,除了Yogi^[89]外,均无法摆脱鞍点,图3(c)、(d)分别可视化了Adam和Yogi由相同初始点($x = -3, y = 0$)出发的优化行为. 限于篇幅,这里仅仅展示了Adam在鞍点附近的优化过程. 图3(e)展示了基于Lookahead^[94]的LookaheadYogi优化过程,可以发现将自适应算法与Yogi结合同样能够逃离鞍点区域. 最后在 y 方向上给初始点施加小扰动,观察SGD的优化过程. 如图3(f)、(g)、(h)所示,对应的初始点分别为 $y = 0.1, 0.01, 0.001$. 无论施加多微小的干扰,各种学习率策略都将成功逃离鞍点区域,但是,step_lr学习率策略逃离速度明显慢于其他方式. 尽管鞍点可能给优化过程带来困扰,但在深度卷积神经网络中,Panageas等^[95]从理论上证明了一阶优化器不会收敛到鞍点(可能在鞍点附近停留相当长时间^[96]). 良好的权重参数初始化方法(如Xavier^[97](各层的激活值和梯度方差在传播过程中保持一致),Kaiming^[98]),适宜的mini-batch引入的噪声^[73],以及在训练过程中注入各向同性的梯度噪声^[20,99]等措施都将加速优化器逃离鞍点区域. 针对鞍点的情形,5类学习率策略可以归纳出以下特点:在良好的初始化、梯度噪声和随机扰动下,过快的衰减学习率将造成参数更新缓慢(参数更新项 $\eta_k \cdot \nabla_{\theta_{k-1}} L(\theta_{k-1})$ 急剧减小),导致模型在鞍点附

近停留相当长时间,不利于模型快速逃离鞍点区域,这将严重滞后于模型收敛速度,如step_lr学习率策略.

4.1.4 病态条件

前馈神经网络具有病态条件的Hessians,并且这种条件相当普遍^[100]. 病态条件的存在导致优化器不能高效地解决神经网络的训练问题. 本节着重介绍不同学习率策略应对神经网络中广泛存在的平坦区域和断崖(cliffs)区域时的优化行为. 当初始点处于cliffs平坦区域时,训练初期梯度非常小,优化器将在此区域停留很长时间,不恰当的学习率衰减策略将导致优化器卡在平坦区域,如图7(d)、(f)所示. 可以看出在cliffs平坦区域,constant_lr策略能够实现最佳的收敛速度和精度,如图7(a)、(b)所示(图7(b)、(e)基于2D sigmoid函数实现). 与尖锐的局部最小值相比,SGD更倾向于收敛到平坦的局部最小值^[101]. 恰当的学习率策略能在训练后期推动优化器快速通过平坦区域,如图7(c)、(f)所示. 综上所述,cliffs区域的存在并不会妨碍优化过程,影响SGD优化的主要因素应被认为是大量存在的相对平坦区域(如图7(f)所示),同时使用constant_lr学习率可以快速通过这样的区域. 针对病态条件的情况,5类学习率策略可以归纳出以下特点:1)快速衰减学习率不利于模型逃离平坦区域,这将导致参数更新项($\eta_k \cdot \nabla_{\theta_{k-1}} L(\theta_{k-1})$)急剧减小,造成模型训练后期无法进行有效的参数更新,如step_lr学习率策略;2)在非峡谷区域,模型收敛速度的快慢似乎与学习率是否具有循环周期无关,相反与学

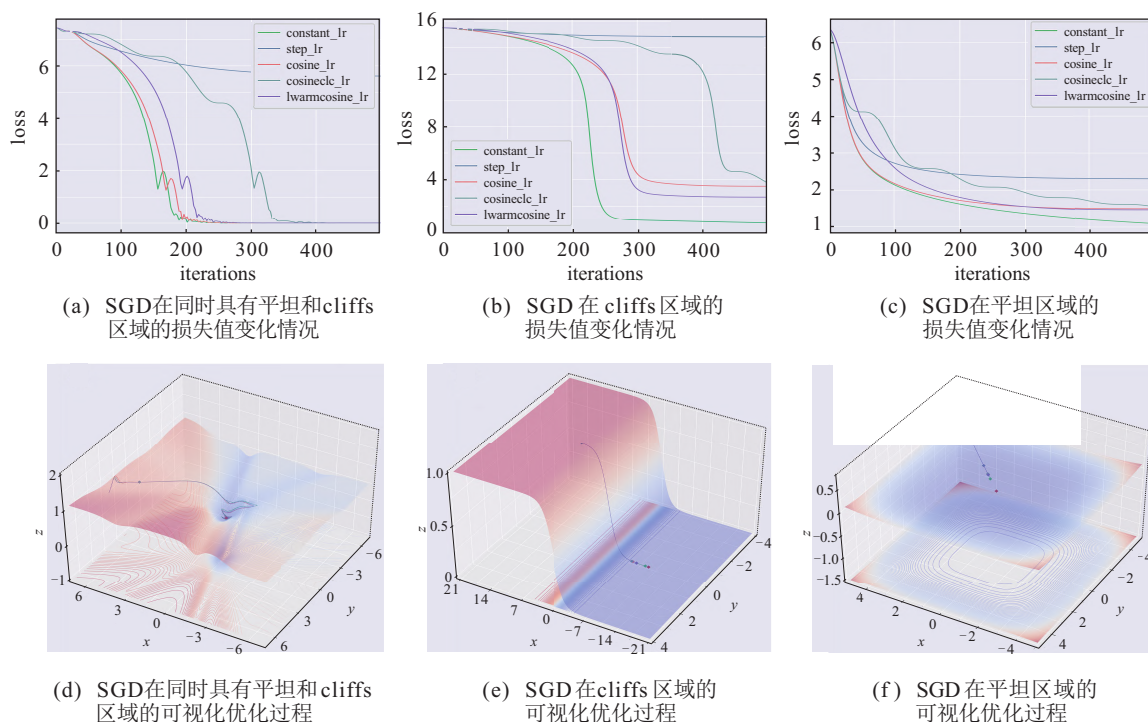


图7 SGD在平坦区域和cliffs区域的优化行为

习率大小正相关,如7(a)、(b)、(c)所示,不具有循环周期的学习率策略(constant_lr, lwarmcosine_lr, cosine_lr)的收敛效果好于具有循环周期的学习率策略(cosineclr_lr),推测是由于在单个循环周期中存在更小的学习率,使得参数更新步长减小所导致的。

4.2 CIFAR实验结果与分析

为了进一步验证在不同模型和不同数据集条件下上述学习率的实际效果,试图找到不同模型和数据集条件下的最佳学习率设置(在常值学习率策略下寻找最佳的初始学习率)。基准测试中的所有LR策略都将其作为可调超参数,为了对比,本文的相关学习率策略超参数的选择在同一数据集下尽量忠实于原文献。同时考虑了两种不同的可调预算,即只使用一个随机种子进行调整,然后使用不同的种子重复最佳设置3次,这样可以得到测试结果均值和标准偏差,从而评估学习率策略的稳定性。区别于单一种子多次运行再求平均值的方法,这样的设置更能反映现实问题,避免因随机种子的选择而造成的干扰。首先在CIFAR-10和CIFAR-100数据集上训练宽残差网络(wide residual neural networks(WRNs), WRN-28-10architecture)^[18]。CIFAR-10和CIFAR-100^[102]数据集分别由10类和100类 32×32 彩色图像构成,划分为50000个训练图像和10000个测试图像。相关学习率超参数的选择和数据增强及预处理方式遵循Loshchilov等^[23]和Zagoruyko等^[18]的实验设置。

图4展示了使用WRN-28-10网络在CIFAR-10数据集上的实验结果。需要注意的是,对于每一类学习率策略,如SGDR(包括变步长SGDR以及带衰减的SGDR),结果复现了原文献中该类学习率策略的最佳参数设置,并且选择其中top-1 accuracy的最大值作为该类学习率策略的测试基准。通过对图4的分析可以得到3个学习率设置的启示:1)学习率衰减是必须的,constant学习率在3次随机种子运行条件下实现了平均91.92%的测试精度,任何具有衰减机制的学习率策略在测试过程中的表现均优于constant。2)选择合适的学习率衰减策略对模型的学习进程至关重要,如HTD衰减学习率策略可以提高模型精度达4.65%。3)对于具有衰减机制的每一簇学习率(piecewise decay, smooth decay, cyclical decay, warmup decay),绝大部分学习率策略在CIFAR-10上对模型的性能影响有限,其测试精度之间的差距大约在0.869%之间。

为了更进一步测试不同学习率策略对模型训练速度的影响,本文在每一簇学习率中选择在CIFAR-

10测试集上表现最佳的前两者,在CIFAR-100数据集上进行进一步验证,如图8所示。可以观察到:1)除constant学习率外,其余学习率策略在CIFAR-100数据集上的性能差距较小($\approx 1.14\%$)。2)同CIFAR-10上得出的启示1)相似,具有衰减的学习率策略对模型至关重要,具有衰减的学习率策略有助于模型训练后期收敛到较优的局部最优解,有利于模型泛化并提高模型测试性能(10.97% ~ 12.11%)。同时应该注意到,在更具有挑战性的数据集上,学习率衰减策略对模型的影响更为严重(CIFAR-10: 3.78% ~ 4.65% vs CIFAR-100: 10.97% ~ 12.11%)。3)从图8中可以看到,Constant学习率策略能以最快的训练速度达到90%的最终测试精度,收敛速度优于其他任何学习率策略。将constant(或linear warmup decay)与cosine decay学习率结合可以在速度和精度上取得较好的权衡(如Ssgdr和linear warmup cosine),其训练时长大约仅为multi-step decay的一半。这些性能表现与文献[25, 51, 56]一致,可以归纳出以下特点:1)模型训练前中期的损失地景通常是一些平坦区域,维持较大的常值学习率可以加速DNNs训练(参数更新项 $\eta_k \cdot \nabla_{\theta_{k-1}} L(\theta_{k-1})$ 较大);2)CIFAR-10和CIFAR-100数据集测试的结果表明,具有余弦衰减的学习率策略具有较高的实用性,其达到最佳精度的训练周期相较于HTD学习率策略大约减少一倍,精度仅减少0.49%。

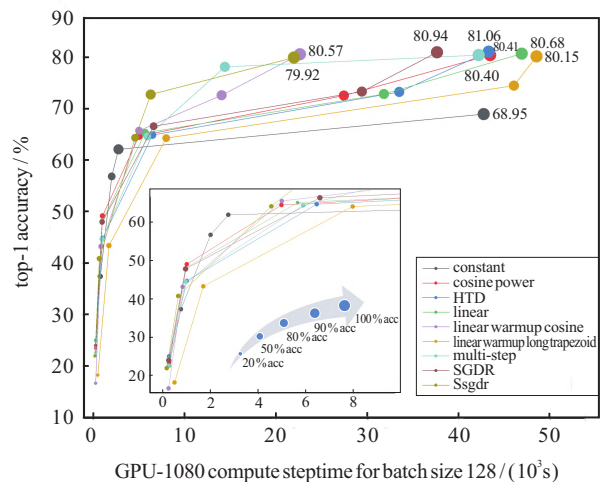


图8 采用WRN-28-10网络在数据集CIFAR-100的实验结果

4.3 ImageNet实验结果与分析

基于ImageNet-2012数据集,本文针对不同的学习率策略使用相似的实验设置来训练EfficientNet-B0^[19],以评估不同学习率机制对模型训练的影响。该数据集包含1000个类别,模型在128万张训练图像上进行训练,并在5万张验证图像上进行评估。本文采用top-1和top-5错误率作为评价指标。如图9

所示,HTD学习率策略在ImageNet数据中取得最低top-1错误率(25.95%),略优于余弦学习率策略。HTD比step decay需要更少的超参数,比余弦学习率更灵活^[35]。SGDR学习率策略取得了与HTD相似的实验结果(仅相差0.07%)。SGDR通过调度学习速率来模拟热重启,可以加速DNNs的训练,如图9所示,使用SGDR训练EfficientNet-B0相当长时间,其收敛速度

均快于HTD。与HTD和SGDR相比,具有线性热启动的余弦学习率策略(lwarmcosine_lr)在训练低参数量模型时,实验结果最差,分别达到0.36%和0.43%。推测可能是因为在低参数量模型训练中,学习率热启动不是必须的,相反可能退化模型学习能力,同时在模型训练的前中期维持较高的学习率有助于模型对数据分布的探索。

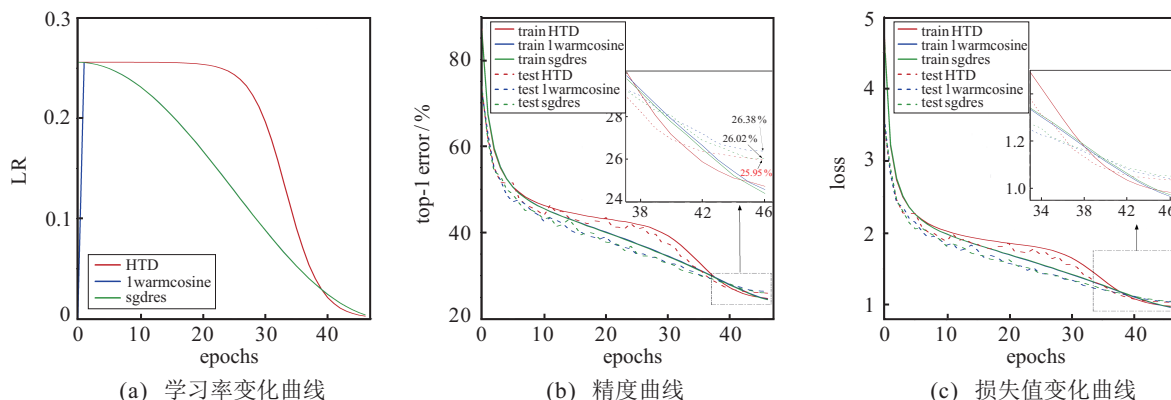


图9 采用EfficientNet-B0网络在数据集ImageNet的实验结果

针对ImageNet的情况,3类学习率策略可以归纳出以下特点:1)模型前中期(大约前30 epochs)维持较高的学习率有助于模型对数据分布空间的探索,提高模型的性能,如HTD学习率策略,其准确度提高0.07%,泛化能力更强;2)具有线性热启动的余弦学习率策略取得了稍弱的测试精度,其在测试集上的泛化性能较差,然而其模型的训练和测试集损失值却较低,推测具有热启动的学习率策略在某种程度上可以帮助模型训练。

5 总结与展望

学习率是深度神经网络最重要的超参数,近年来取得了很多可喜的研究成果,如Wu等^[25]以ResNet和LeNet为骨干模型,探讨了不同参数下部分分段衰减、平滑衰减和循环衰减学习率策略在CIFAR-10和MNIST数据集下的性能表现。但是,对于大多数学习率算法的研究而言,引入相关方法的原始工作仅针对特定数据集进行了评估。在很多情况下,对各种新的学习率算法在公开数据集下进行测试和验证能够较快理解该算法的性能表现和应用场景。本文针对学习率研究策略中模型收敛速度和收敛精度两个方面近5年来取得的成果以及未来可能的研究方向进行了综述。

首先介绍了学习率在深度神经网络模型优化中的作用,总结如下:学习率控制网络优化过程中的参数更新步长,学习率过小会导致收敛缓慢,而学习率

过大会阻碍收敛,导致损失函数震荡,甚至发散。然后,以考虑函数局部曲率的1D优化例子展示了最优优化过程。本文的学习率策略应用统计分析显示,在深度神经网络的训练部署中,较多的研究工作投入大量资源关注模型结构的设计,而忽视了学习率在模型训练中的影响。目前,主流的学习率策略包括余弦衰减学习率,具有线性热启动的余弦衰减学习率以及具有线性热启动的线性衰减学习率等。如图5、图3、图7所示,余弦衰减学习率和具有线性热启动的余弦衰减学习率在不同2D曲面上的优化通常具有可接受的优化结果,从侧面印证了余弦衰减学习率和具有线性热启动的余弦衰减学习率流行的部分原因。但是,对于在神经网络优化过程中广泛存在的平坦区域,从模型收敛速度和精度来看,常值学习率具有提高模型训练前中期性能的潜力,如图6、图7、图4、图8所示。具有循环衰减的学习率在收敛精度上略差,但其收敛速度在本文所比较的策略中 fastest,如图5、图8所示。循环衰减学习率在应对网络训练中存在的梯度消失、资源开销大甚至算法不收敛等问题上还有进一步探讨和挖掘的空间。

本文在几个常见的数据集和模型下,对不同的学习率应用场景进行了详细的测试分析,梳理了学习率调度算法的最新研究进展。总结如下:1)一般来说,较高的初始学习率能提高模型对数据分布的探索能力,加快模型的收敛速度,如使用较高的常值学习率;2)学习率衰减对训练高精度模型至关重要,如

图4所示;3)循环学习率可以在模型训练速度和精度上取得较好的平衡,如图8所示;4)强调分类准确性指标时,HTD在本文分类实验中取得了最佳表现,如图4、图8、图9所示.更进一步,为了验证本文中几篇重要文献的结果,本文对各簇学习率在几个常见数据集和模型下的表现进行了全面的测试分析,希望对后续研究人员系统地评估不同LR策略,并有效地确定最合适的DNNs学习率调度计划能有所帮助.本文实验部分亦存在如下未考虑到的问题:1)限于计算资源不足的问题,本实验并未对单一学习率算法进行全面参数寻优;2)在CIFAR-100和ImageNet数据上采用了CIFAR-10的最优学习率策略进行后续比较,可能存在学习率策略在不同数据集上泛化不一致的问题.在此基础上,本文提出了今后研究中对学习率调度策略的研究展望:

1) 最佳初始学习率的确定.研究表明模型训练过程中存在超收敛^[51],其训练速度比标准训练方法快一个数量级.其中关键难点是简化Hessian Free,以计算最佳学习率的估计值.文献[48]提出一种可求解的神经动力学模型来分析模型在不同学习率下的行为表现,并预测了大而稳定的初始学习率的狭窄范围,为进一步开展初始学习率的研究提供了基础.

2) 学习率调度算法的扩展性.将模型并行部署到具有同步SGD的多个GPU上可以工程化DNNs训练,然而如何有效扩展学习率时间表并非易事.Krizhevsky^[78]建议以 \sqrt{m} 速率扩大学习率,以使梯度期望方差保持恒定.然而,Goyal等^[46]和Smith等^[47]使用 m 比例因子线性缩放学习率,并在实践中取得了良好的实验结果.Jastrzebski等^[67]和Bottou等^[103]对线性缩放策略结果进行了理论解释.文献[99,104-105]在多GPU并行训练中进行了广泛探索,并取得了一些有价值的设置建议,然而在具体实际应用中,还需要考虑相关因素,因此,并行训练中学习率策略的选择仍然是一个开放性问题.

3) 学习率在目标检测、实例分割等任务中的设置.在检测、分割等下游任务中,通常利用ImageNet数据集预训练的模型作为骨干(backbone)网络用于提取相应的特征,随后在检测头或分割头的作用下进行后续任务.Open-mmlab^[81-82]作为流行的计算机视觉工具箱,目标检测常用具有线性热启动的步进衰减学习率,而分割任务常使用多项式衰减的学习率.鲜有文献针对这些下游任务的学习率调度方案进行全面分析比较.因此,本文综述的学习率调度策略在这些任务上的泛化性能还有待进一步探索.

4) 学习率调度算法的超参数搜索.现有的超参数搜索工具Hyperpt^[106]、SMAC^[107]、Optuna^[108]等都使用了经典和通用的超参数搜索算法,如网格搜索、随机搜索、贝叶斯优化等,具有较高的超参数调优代价.目前,缺乏用于存储元数据和历史学习率调用经验的相关工具来为后续任务推荐优化的LR策略,该学习率调度算法工具包应该包括与已知数据集、DNNs模型和任务类型的相似度量,并能结合相似度量与现有的LR策略排名等指标来推荐LR使用方案的最优排序.

参考文献(References)

- [1] Robbins H, Monro S. A stochastic approximation method[J]. The Annals of Mathematical Statistics, 1951, 22(3): 400-407.
- [2] Choromanska A, Henaff M, Mathieu M, et al. The loss surfaces of multilayer networks[C]. Artificial intelligence and statistics. PMLR, 2015: 192-204.
- [3] Pham H, Dai Z H, Xie Q Z, et al. Meta pseudo labels[C]. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Nashville, 2021: 11552-11563.
- [4] Ghiasi G, Cui Y, Srinivas A, et al. Simple copy-paste is a strong data augmentation method for instance segmentation[C]. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Nashville, 2021: 2917-2927.
- [5] Zhang Y, Qin J, Park D S, et al. Pushing the limits of semi-supervised learning for automatic speech recognition[J/OL]. 2020, arXiv: 2010.10504.
- [6] Ming Z H, Xia J S, Luqman M M, et al. Dynamic multi-task learning for face recognition with facial expression[J/OL]. 2019, arXiv: 1911.03281.
- [7] Edunov S, Ott M, Auli M, et al. Understanding back-translation at scale[J/OL]. 2018, arXiv: 1808.09381.
- [8] Bojarski M, Del Testa D, Dworakowski D, et al. End to end learning for self-driving cars[J/OL]. 2016: arXiv: 1604.07316.
- [9] Lecun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.
- [10] Qian N. On the momentum term in gradient descent learning algorithms[J]. Neural Networks, 1999, 12(1): 145-151.
- [11] Kingma D P, Ba J. Adam: A method for stochastic optimization[J/OL]. 2014, arXiv: 1412.6980.
- [12] Bengio Y. Practical recommendations for gradient-based training of deep architectures[C]. Neural Networks: Tricks of the Trade. Berlin: Springer, 2012: 437-478.
- [13] Abadi M, Barham P, Chen J, et al. TensorFlow: A system for large-scale machine learning[C]. The 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16). Savannah, 2016: 265-283.

- [14] Paszke A, Gross S, Massa F, et al. PyTorch: An imperative style, high-performance deep learning library[J/OL]. 2019, arXiv: 1912.01703.
- [15] Coleman C, Narayanan D, Kang D, et al. Dawnbench: An end-to-end deep learning benchmark and competition[J]. Training, 2017, 100(101): 102.
- [16] Bengio Y. Practical recommendations for gradient-based training of deep architectures[C]. Neural Networks: Tricks of the Trade. Berlin, Heidelberg: Springer, 2012: 437-478.
- [17] Rumelhart D E, Hinton G E, Williams R J. Learning representations by back-propagating errors[J]. Nature, 1986, 323(6088): 533-536.
- [18] Zagoruyko S, Komodakis N. Wide residual networks[J/OL]. 2016, arXiv: 1605.07146,
- [19] Tan M, Le Q. Efficientnet: Rethinking model scaling for convolutional neural networks[C]. International Conference on Machine Learning. PMLR, 2019: 6105-6114.
- [20] Ruder S. An overview of gradient descent optimization algorithms[J/OL]. 2016, arXiv: 1609.04747.
- [21] Martens J. Deep learning via hessian-free optimization[C]. International Conference on Machine Learning. Haifa, 2010, 27: 735-742.
- [22] Smith L N. Cyclical learning rates for training neural networks[C]. 2017 IEEE Winter Conference on Applications of Computer Vision. Santa Rosa, 2017: 464-472.
- [23] Loshchilov I, Hutter F. SGDR: Stochastic gradient descent with warm restarts[J/OL]. 2016, arXiv: 1608.03983.
- [24] Senior A, Heigold G, Ranzato M, et al. An empirical study of learning rates in deep neural networks for speech recognition[C]. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. Vancouver, 2013: 6724-6728.
- [25] Wu Y Z, Liu L, Bae J, et al. Demystifying learning rate policies for high accuracy training of deep neural networks[C]. 2019 IEEE International Conference on Big Data (Big Data). Los Angeles, 2019: 1971-1980.
- [26] Retsinas G, Sfikas G, Filntisis P, et al. Trainable learning rate[EB/OL]. [2022-01-21]. <https://openreview.net/forum?id=fHeK814NOMO>.
- [27] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[C]. International Conference on Machine Learning. Lille, 2015: 448-456.
- [28] Raffel C, Shazeer N, Roberts A, et al. Exploring the limits of transfer learning with a unified text-to-text transformer[J/OL]. 2019, arXiv: 1910.10683.
- [29] Chen L C, Papandreou G, Kokkinos I, et al. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, 40(4): 834-848.
- [30] He K M, Zhang X Y, Ren S Q, et al. Deep residual learning for image recognition[C]. 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, 2016: 770-778.
- [31] Pouyanfar S, Chen S C. T-LRA: Trend-based learning rate annealing for deep neural networks[C]. The 3rd IEEE International Conference on Multimedia Big Data (BigMM). Laguna Hills, 2017: 50-57.
- [32] Hu T, Qi H G, Huang Q M, et al. See better before looking closer: Weakly supervised data augmentation network for fine-grained visual classification[J/OL]. 2019, arXiv: 1901.09891.
- [33] Dauphin Y N, Pascanu R, Gulcehre C, et al. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization[J]. Proceedings of the 27th International Conference on Neural Information Processing Systems, 2014, 2: 2933-2941.
- [34] Schmidt R M, Schneider F, Hennig P. Descending through a crowded valley-benchmarking deep learning optimizers[J/OL]. 2020, arXiv: 2007.01547.
- [35] Hsueh B Y, Li W, Wu I C. Stochastic gradient descent with hyperbolic-tangent decay on classification[C]. 2019 IEEE Winter Conference on Applications of Computer Vision. Waikoloa, 2019: 435-442.
- [36] Kuan C M, Hornik K. Convergence of learning algorithms with constant learning rates[J]. IEEE Transactions on Neural Networks, 1991, 2(5): 484-489.
- [37] Shalev-Shwartz S, Singer Y, Srebro N, et al. Pegasos: Primal estimated sub-gradient solver for SVM[J]. Mathematical Programming, 2011, 127(1): 3-30.
- [38] Bottou L. Stochastic gradient descent tricks[C]. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2012: 421-436.
- [39] Murata N. A statistical study of on-line learning[C]. On-Line Learning in Neural Networks. Cambridge: Cambridge University Press, 1999: 63-92.
- [40] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017, 60(6): 84-90.
- [41] Han D, Kim J, Kim J. Deep pyramidal residual networks[C]. 2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, 2017: 6307-6315.
- [42] Goodfellow I, Bengio Y, Courville A. Deep learning[M]. Cambridge: MIT Press, 2016.
- [43] Li M T, Yumer E, Ramanan D. Budgeted training: Rethinking deep neural network training under resource constraints[J/OL]. 2019, arXiv: 1905.04753.
- [44] Sainath T N, Kingsbury B, Ramabhadran B, et al. Making deep belief networks effective for large vocabulary continuous speech recognition[C]. 2011 IEEE Workshop on Automatic Speech Recognition & Understanding. Waikoloa, 2011: 30-35.
- [45] Renals S, Morgan N, Cohen M, et al. Connectionist probability estimation in the DECIPHER speech recognition system[C]. 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing.

- San Francisco, 1992: 601-604.
- [46] Goyal P, Dollár P, Girshick R, et al. Accurate, large minibatch SGD: Training ImageNet in 1 hour[J/OL]. 2017, arXiv: 1706.02677.
- [47] Smith S L, Kindermans P J, Ying C, et al. Don't decay the learning rate, increase the batch size[J/OL]. 2017, arXiv: 1711.00489.
- [48] Lewkowycz A, Bahri Y, Dyer E, et al. The large learning rate phase of deep learning: The catapult mechanism[J/OL]. 2020, arXiv: 2003.02218.
- [49] Bello I, Zoph B, Vasudevan V, et al. Neural optimizer search with reinforcement learning[C]. International Conference on Machine Learning. Sydney, 2017: 459-468.
- [50] Hundt A, Jain V, Hager G D. Sharpdarts: Faster and more accurate differentiable architecture search[J/OL]. 2019, arXiv: 1903.09900.
- [51] Smith L N, Topin N. Super-convergence: Very fast training of neural networks using large learning rates[C]. Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications. Baltimore, 2019, 11006: 369-386.
- [52] Smith L N. No more pesky learning rate guessing games[J/OL]. 2015, arXiv: 1506.01186.
- [53] Yang T, Lin Q. Stochastic subgradient methods with linear convergence for polyhedral convex optimization[J/OL]. 2016, arXiv: 1510.01444.
- [54] Mehta S, Rastegari M, Shapiro L, et al. ESPNetv2: A light-weight, power efficient, and general purpose convolutional neural network[C]. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, 2019: 9182-9192.
- [55] Li J Q, Yang X D. A cyclical learning rate method in deep learning training[C]. 2020 International Conference on Computer, Information and Telecommunication Systems (CITS). Hangzhou, 2020: 1-5.
- [56] Yang H, Yuan C F, Xing J L, et al. Diversity encouraging ensemble of convolutional networks for high performance action recognition[C]. 2017 IEEE International Conference on Image Processing. Beijing, 2017: 2846-2850.
- [57] Lee C M, Liu J, Peng W. Applying cyclical learning rate to neural machine translation[J/OL]. 2020, arXiv: 2004.02401.
- [58] Gulde R, Tuscher M, Csiszar A, et al. Deep reinforcement learning using cyclical learning rates[C]. The 3rd International Conference on Artificial Intelligence for Industries (AI4I). Irvine, 2020: 32-35.
- [59] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[J/OL]. 2014, arXiv: 1409.3215.
- [60] Girshick R. Fast R-CNN[C]. 2015 IEEE International Conference on Computer Vision. Santiago, 2015: 1440-1448.
- [61] He K M, Gkioxari G, Dollár P, et al. Mask R-CNN[C]. 2017 IEEE International Conference on Computer Vision. Venice, 2017: 2980-2988.
- [62] Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection[C]. 2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, 2017: 936-944.
- [63] Ren S Q, He K M, Girshick R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137-1149.
- [64] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J/OL]. 2017, arXiv:1706.03762.
- [65] Howard J, Ruder S. Universal language model fine-tuning for text classification[J/OL]. 2018, arXiv: 1801.06146.
- [66] Xing C, Arpit D, Tsirigotis C, et al. A walk with sgd[J/OL]. 2018, arXiv: 1802.08770.
- [67] Jastrzbski S, Kenton Z, Arpit D, et al. Three factors influencing minima in sgd[J/OL]. 2017, arXiv: 1711.04623.
- [68] Hochreiter S, Schmidhuber J. Flat minima[J]. Neural Computation, 1997, 9(1): 1-42.
- [69] Keskar N S, Mudigere D, Nocedal J, et al. On large-batch training for deep learning: Generalization gap and sharp minima[J/OL]. 2016, arXiv: 1609.04836.
- [70] Wu L, Zhu Z X. Towards understanding generalization of deep learning: Perspective of loss landscapes[J/OL]. 2017, arXiv: 1706.10239.
- [71] Keskar N S, Mudigere D, Nocedal J, et al. On large-batch training for deep learning: Generalization gap and sharp minima[J/OL]. 2016, arXiv: 1609.04836.
- [72] Chaudhari P, Soatto S. Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks[C]. 2018 Information Theory and Applications Workshop (ITA). San Diego, 2018: 1-10.
- [73] Smith S L, Le Q V. A bayesian perspective on generalization and stochastic gradient descent[J/OL]. 2017, arXiv: 1710.06451.
- [74] Pham Q H, Anh Nguyen V, Doan L B, et al. From universal language model to downstream task: Improving RoBERTa-based Vietnamese hate speech detection[C]. The 12th International Conference on Knowledge and Systems Engineering (KSE). Can Tho, 2020: 37-42.
- [75] Liu L Y, Jiang H M, He P C, et al. On the variance of the adaptive learning rate and beyond[J/OL]. 2019, arXiv: 1908.03265.
- [76] You Y, Gitman I, Ginsburg B. Large batch training of convolutional networks[J/OL]. 2017, arXiv: 1708.03888.
- [77] Gotmare A, Keskar N S, Xiong C M, et al. A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation[J/OL]. 2018, arXiv: 1810.13243.
- [78] Krizhevsky A. One weird trick for parallelizing convolutional neural networks[J/OL]. 2014, arXiv: 1404.5997.
- [79] Li M, Zhang T, Chen Y Q, et al. Efficient mini-batch training for stochastic optimization[C]. KDD'14: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and

- Data Mining. New York, 2014: 661-670.
- [80] Hoffer E, Hubara I, Soudry D. Train longer, generalize better: Closing the generalization gap in large batch training of neural networks[C]. NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach, 2017: 1729-1739.
- [81] Chen K, Wang J Q, Pang J M, et al. MMDetection: Open MMLab detection toolbox and benchmark[J/OL]. 2019, arXiv: 1906.07155.
- [82] Contributors M M S. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark[EB/OL]. [2022-01-21]. <https://github.com/open-mmlab/mms Segmentation>.
- [83] Nguyen T Q, Salazar J. Transformers without tears: Improving the normalization of self-attention[J/OL]. 2019, arXiv: 1910.05895.
- [84] Neelakantan A, Vilnis L, Le Q V, et al. Adding gradient noise improves learning for very deep networks[J/OL]. 2015, arXiv: 1511.06807.
- [85] Bergstra J, Yamins D, Cox D. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures[C]. International Conference on Machine Learning. Atlanta, 2013: 115-123.
- [86] Luo L C, Xiong Y H, Liu Y, et al. Adaptive gradient methods with dynamic bound of learning rate[J/OL]. 2019, arXiv: 1902.09843.
- [87] Yao Z W, Gholami A, Shen S, et al. ADAHESSIAN: An adaptive second order optimizer for machine learning[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2021, 35(12): 10665-10673.
- [88] Kingma D P, Ba J. Adam: A method for stochastic optimization[J/OL]. 2014, arXiv: 1412.6980.
- [89] Zaheer M, Reddi S, Sachan D, et al. Adaptive methods for nonconvex optimization[J]. Advances in Neural Information Processing Systems, 2018: 31.
- [90] Liu L Y, Jiang H M, He P C, et al. On the variance of the adaptive learning rate and beyond[J/OL]. 2019, arXiv: 1908.03265.
- [91] Ma J, Yarats D. Quasi-hyperbolic momentum and Adam for deep learning[J/OL]. 2018, arXiv: 1810.06801.
- [92] Dubey S R, Chakraborty S, Roy S K, et al. DiffGrad: An optimization method for convolutional neural networks[J]. IEEE Transactions on Neural Networks and Learning Systems, 2020, 31(11): 4500-4511.
- [93] Keskar N S, Socher R. Improving generalization performance by switching from Adam to SGD[J/OL]. 2017, arXiv: 1712.07628.
- [94] Zhang M, Lucas J, Ba J, et al. Lookahead optimizer: K steps forward, 1 step back[J/OL]. 2019, arXiv: 1907.08610.
- [95] Panageas I, Piliouras G, Wang X. First-order methods almost always avoid saddle points: The case of vanishing step-sizes[J/OL]. 2019, arXiv: 1906.07772.
- [96] Du S S, Jin C, Lee J D, et al. Gradient descent can take exponential time to escape saddle points[J/OL]. 2017, arXiv: 1705.10412.
- [97] Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks[C]. Proceedings of the 13th International Conference on Artificial Intelligence and Statistics. Sardinia, 2010: 249-256.
- [98] He K M, Zhang X Y, Ren S Q, et al. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification[C]. 2015 IEEE International Conference on Computer Vision. Santiago, 2015: 1026-1034.
- [99] Ge R, Huang F, Jin C, et al. Escaping from saddle points—Online stochastic gradient for tensor decomposition[C]. Conference on Learning Theory. Paris, 2015: 797-842.
- [100] Saariinen S, Bramley R, Cybenko G. Ill-conditioning in neural network training problems[J]. SIAM Journal on Scientific Computing, 1993, 14(3): 693-714.
- [101] Xie Z, Sato I, Sugiyama M. A diffusion theory for deep learning dynamics: Stochastic gradient descent exponentially favors flat minima[J/OL]. 2020, arXiv: 2002.03495.
- [102] Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images[R]. Toronto: University of Toronto, 2009.
- [103] Bottou L, Curtis F E, Nocedal J. Optimization methods for large-scale machine learning[J]. SIAM Review, 2018, 60(2): 223-311.
- [104] You Y, Gitman I, Ginsburg B. Scaling sgd batch size to 32k for imagenet training[J/OL]. 2017, arXiv: 1708.03888.
- [105] Popel M, Bojar O. Training tips for the transformer model[J]. The Prague Bulletin of Mathematical Linguistics, 2018, 110(1): 43-70.
- [106] Bergstra J, Yamins D, Cox D D. Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms[C]. Proceedings of the 12th Python in Science Conference. Austin, 2013, 13: 20.
- [107] Hutter F, Hoos H H, Leyton-Brown K. Sequential model-based optimization for general algorithm configuration[C]. International Conference on Learning and Intelligent Optimization. Berlin: Springer, 2011: 507-523.
- [108] Akiba T, Sano S, Yanase T, et al. Optuna: A next-generation hyperparameter optimization framework[C]. KDD'19: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. Anchorage AK, 2019: 2623-2631.

作者简介

刘云飞 (1995—), 男, 博士生, 从事深度学习、医学图像分割与检测等研究, E-mail: liuyunfei@stu.scu.edu.cn;

张俊然 (1978—), 男, 教授, 博士生导师, 从事类脑人工智能与模式识别、微传感器与智能仪器、医学影像与大数据分析等研究, E-mail: junranzhang@scu.edu.cn.