软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2024,35(6):3013−3035 [doi: 10.13328/j.cnki.jos.006894]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于自引导进化策略的高效自动化数据增强算法*

朱光辉, 陈文忠, 朱振南, 袁春风, 黄宜华

(计算机软件新技术国家重点实验室 (南京大学), 江苏 南京 210023)

通信作者: 朱光辉, E-mail: zgh@nju.edu.cn; 黄宜华, E-mail: yhuang@nju.edu.cn

摘　要: 深度学习在图像、文本、语音等媒体数据的分析任务上取得了优异的性能. 数据增强可以非常有效地提升训练数据的规模以及多样性, 从而提高模型的泛化性. 但是, 对于给定数据集, 设计优异的数据增强策略大量依赖专家经验和领域知识, 而且需要反复尝试, 费时费力. 近年来, 自动化数据增强通过机器自动设计数据增强策略, 已引起了学界和业界的广泛关注. 为了解决现有自动化数据增强算法尚无法在预测准确率和搜索效率之间取得良好平衡的问题, 提出一种基于自引导进化策略的自动化数据增强算法 SGES AA. 首先, 设计一种有效的数据增强策略连续化向量表示方法, 并将自动化数据增强问题转换为连续化策略向量的搜索问题. 其次, 提出一种基于自引导进化策略的策略向量搜索方法, 通过引入历史估计梯度信息指导探索点的采样与更新, 在能够有效避免陷入局部最优解的同时, 可提升搜索过程的收敛速度. 在图像、文本以及语音数据集上的大量实验结果表明, 所提算法在不显著增加搜索耗时的情况下, 预测准确率优于或者匹配目前最优的自动化数据增强方法.

关键词: 深度学习; 数据增强; 自动化机器学习; 自引导进化策略

中图法分类号: TP391

中文引用格式: 朱光辉, 陈文忠, 朱振南, 袁春风, 黄宜华. 基于自引导进化策略的高效自动化数据增强算法. 软件学报, 2024, 35(6): 3013–3035. http://www.jos.org.cn/1000-9825/6894.htm

英文引用格式: Zhu GH, Chen WZ, Zhu ZN, Yuan CF, Huang YH. Efficient Automated Data Augmentation Algorithm Based on Self-guided Evolution Strategy. Ruan Jian Xue Bao/Journal of Software, 2024, 35(6): 3013−3035 (in Chinese). http://www.jos.org.cn/1000-9825/6894.htm

Efficient Automated Data Augmentation Algorithm Based on Self-guided Evolution Strategy

ZHU Guang-Hui, CHEN Wen-Zhong, ZHU Zhen-Nan, YUAN Chun-Feng, HUANG Yi-Hua

(State Key Laboratory for Novel Software Technology (Nanjing University), Nanjing 210023, China)

Abstract: Deep learning has achieved great success in image classification, natural language processing, and speech recognition. Data augmentation can effectively increase the scale and diversity of training data, thereby improving the generalization of deep learning models. However, for a given dataset, a well-designed data augmentation strategy relies heavily on expert experience and domain knowledge and requires repeated attempts, which is time-consuming and labor-intensive. In recent years, automated data augmentation has attracted widespread attention from the academic community and the industry through the automated design of data augmentation strategies. To solve the problem that existing automated data augmentation algorithms cannot strike a good balance between prediction accuracy and search efficiency, this study proposes an efficient automated data augmentation algorithm SGES AA based on a self-guided evolution strategy. First, an effective continuous vector representation method is designed for the data augmentation strategy, and then the automated data augmentation problem is converted into a search problem of continuous strategy vectors. Second, a strategy vector search method based on the self-guided evolution strategy is presented. By introducing historical estimation gradient information to guide the sampling and updating of exploration points, it can effectively avoid the local optimal solution while improving the convergence of the search process. The results of extensive experiments on image, text, and speech datasets show that the proposed algorithm is superior to or

* 基金项目: 国家自然科学基金 (62102177, U1811461); 江苏省自然科学基金 (BK20210181); 江苏省重点研发计划 (BE2021729)
收稿时间: 2022-06-27; 修改时间: 2022-09-12; 采用时间: 2023-01-05; jos 在线出版时间: 2023-05-24
CNKI 网络首发时间: 2023-05-26

matches the current optimal automated data augmentation methods without significantly increasing the time consumption of searches.
Key words: deep learning; data augmentation; automated machine learning; self-guided evolution strategy

近年来, 深度学习技术已广泛应用于图像处理、语音识别以及文本分析等领域, 并且取得了巨大的成功. 然而, 在实际应用场景中, 由于数据获取成本高、缺少大规模带标签数据, 导致深度学习模型的训练容易产生过拟合等问题. 数据增强 (data augmentation, DA) 技术[1](如图像处理领域中的图像翻转、裁剪、色彩调整等) 是提升训练数据规模和多样性的有效途径, 其能够从原始数据中加工出具有更强表示能力的数据, 从而提升深度学习模型的泛化能力. 与超参数优化类似, 对于给定的数据集及应用场景, 设计良好的数据增强策略往往需要大量的专家经验和领域知识. 除了选择合适的数据增强操作外, 还需要考虑增强操作的组合以及每一种增强操作的幅度 (如图像旋转角度). 因此, 数据增强策略设计是一种复杂的组合优化问题, 需要反复尝试, 费时费力.

为了降低机器学习门槛, 提升 AI 建模效率, 近年来, 自动化机器学习 (automated machine learning, AutoML) 技术[2]应运而生. AutoML 的本质是以 AI 设计 AI, 将机器学习模型选择以及架构设计等问题抽象为搜索问题, 通过设计良好的搜索空间和搜索方法, 实现机器学习模型的自动化设计. 基于 AutoML 技术的核心思想, Google 的研究者首次提出了基于强化学习的自动化数据增强 (automated data augmentation, AutoAugment) 技术[3], 即从给定训练数据中自动去学习到一个最优的数据增强策略. 凭借着巨大的应用价值, 自动化数据增强技术已引起了国内外学术界和工业界的广泛关注.

已有的自动化数据增强方法主要聚焦在提升模型准确率以及搜索效率两个方面. Google AA 虽然能够达到目前最优的模型准确率, 但是其在增强策略搜索过程中, 需要评估每个采样策略, 导致搜索开销巨大, 如在 CIFAR-10 数据集上搜索耗时高达 5 000 个 GPU 小时. 近年来, 为了提升增强策略的搜索效率, 基于种群并行训练的 PBA (population based augmentation)[4]、基于密度匹配的 Fast AutoAugment (Fast AA)[5]、基于可微分机制的 Faster AutoAugment (Faster AA)[6]及 DADA (differentiable automated data augmentation)[7]、基于随机数据增强的 Rand AA[8]等方法陆续被提出, 可将增强策略的搜索效率提升 1 000 倍以上, 大大降低了增强策略的搜索耗时. 然而, 如图 1 所示, 这些方法通过策略近似评估、梯度近似计算等手段提升搜索效率的同时, 对模型的准确率造成了一定的负面影响, 导致模型的准确率不如 Google AA 算法. 因此, 需要设计一种在模型准确率和搜索耗时之间能够实现更好平衡的自动化数据增强方法与算法, 在不降低模型准确率的前提下, 将搜索耗时降低至可接受的水平.

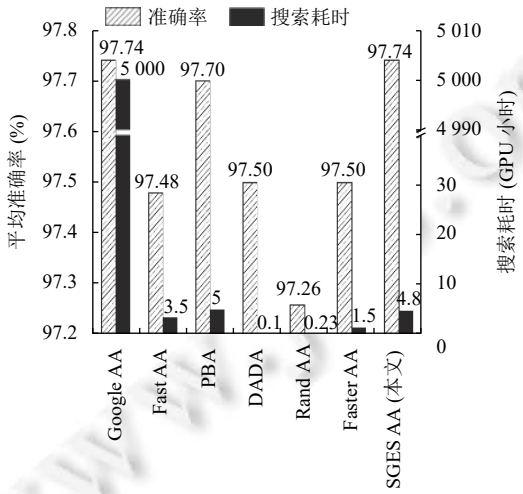


图 1 不同自动化数据增强算法在 CIFAR-10 数据集下的平均准确率与搜索耗时

为此, 本文研究提出了一种基于自引导进化策略[9]的自动化数据增强算法 SGES AA (self-guided evolution strategy for auto-augment). 如图 1 所示, SGES AA 可得到与目前最优的 Google AA 相匹配的准确率, 而且其搜索

耗时仅为 4.8 GPU 小时, 远远低于 Google AA, 而且和其他 AutoAugment 算法的耗时差距并不明显.

具体地, 针对现有增强策略搜索空间存在的离散不连续及搜索范围过大的问题, 本文首先提出一种有效的数据增强策略连续化向量表示方法, 将自动化数据增强问题转换为连续化策略向量的搜索问题, 同时通过约束策略向量的取值范围, 解决搜索空间过大的问题. 其次, 提出一种基于自引导进化策略的策略向量搜索方法. 自引导进化策略利用历史估计梯度构建梯度矩阵并分解该梯度矩阵获取梯度子空间, 在探索点 (即策略向量) 的搜索方向采样时使用该梯度子空间进行引导采样. 这种引入历史估计梯度信息指导探索点更新的方法, 不仅能够有效避免搜索过程陷入局部最优解, 而且能够加速算法收敛速度.

另外, SGES AA 算法高度适合并行化. 每一轮进化迭代过程中, 将产生多个相互无关的探索方向. 本文采用分布式并行计算框架 Ray[10]对多个探索方向对应的策略向量进行并行化评估, 从而进一步提升搜索效率. 除此之外, 已有的自动化数据增强算法大都图像聚焦于分类任务, 本文工作基于统一 SGES AA 算法框架可支持图像分类、语音分类和文本分类等多种媒体数据的智能化分析任务.

综上所述, 本文的研究内容和贡献点主要包含以下 3 个方面.

(1) 设计一种有效的增强策略连续化向量表示方法, 并在此基础上将离散的数据增强策略选择问题抽象为连续化策略向量的搜索问题.

(2) 提出一种基于自引导进化策略的自动化策略向量搜索方法. 通过引入历史估计梯度信息引导搜索方向的采样, 能够在避免陷入局部最优解的同时, 提升搜索效率.

(3) 在图像分类、语音分类以及文本分类等任务数据集上的大量实验结果表明, SGES AA 算法在不显著增加搜索耗时的情况下, 准确率优于已有大多数自动化数据增强算法.

1 相关工作

自动化数据增强的概念由 Google 的研究者最先提出[3], 其通过利用 AI 技术实现增强策略的自动化设计. 一般来讲, 数据增强策略由多个子策略构成, 每个子策略包含若干个串联的数据增强操作. 每个增强函数操作对应两个参数: 幅度 (magnitude) 和应用概率 (probability). 近年来, 学界研究提出的自动化数据增强算法可具体划分为以下 5 类.

1) 基于强化学习的算法: Google AA[3]采用强化学习技术[11]对数据增强策略进行搜索, 该算法从基于循环神经网络的控制器中采样增强策略, 并将增强策略的预测准确率作为回报值, 优化更新控制器的参数. 该算法定义的数据增强策略由 5 个子策略组成, 其中每个子策略包含了 2 个增强操作. 定义的搜索空间中包含 16 种增强操作. 由于强化学习需要在离散的状态空间和动作空间上进行训练, 因此该算法将幅度参数均分为 10 个区间, 应用概率均分为 11 个区间, 最终的搜索空间复杂度高达 $(16\times10\times11)^{10}\approx2.9\times10^{32}$. 而且, 策略的评估需要从头开始训练一个神经网络代理模型, 从而导致巨大搜索耗时开销. Google AA 在 CIFAR-10 数据集[12]上需要5 000 个 GPU 小时才能完成策略搜索工作. 尽管 Google AA 能够达到目前最优的准确率, 但是实用性不高.

2) 基于种群并行训练的算法: 为了解决 Google AA 在评估阶段需要重复地从头开始训练一个代理模型而导致搜索效率低下的问题, PBA 算法[4]则从一边训练一边观察不同数据增强策略的增强效果的角度出发, 采用基于种群训练[13]的超参数优化思想, 构建了由 16 个子模型组成的种群, 其中的子模型可并行训练. PBA 算法可以实现子模型之间的权重共享, 并且在训练的不同阶段使用不同的增强超参数. 尽管 PBA 算法最终的预测性能比 Google AA 算法稍差, 但是其在 CIFAR-10 数据集上的搜索耗时仅为 5 个 GPU 小时. 另外, PBA 算法仍然需要配备大量的 GPU 设备和计算节点保证种群中子模型的并行训练, 应用成本仍然较高.

3) 基于密度匹配的算法: 自动化数据增强一个假设是: 让未增强数据和增强后数据的密度尽量匹配从而保证模型的学习能力. 基于此假设, 研究人员提出了基于密度匹配的算法 Fast AA[5], 其通过在 D_{train} 上不使用数据增强进行模型的训练, 之后在使用数据增强后的 D_{valid} 上进行预测, 最后使用验证集的预测准确率评估一个数据集和另一个数据集的匹配程度. 这种方式有效避免了重复训练模型带来的巨大时间开销. 然而, 研究者对于“密度匹配”思想的原理没有给出理论解释, 只能从数据分布一致性的角度理解该思想. Fast AA 同样能够大幅降低搜索耗时, 但

其在测试集上的预测性能表现一般.

4) 基于可微分机制的算法: 受 AutoML 领域中的可微神经网络架构搜索算法 DARTS[14]的启发, 先后有研究者提出基于可微分机制的自动化数据增强算法 Faster AA[6]和 DADA[7] . 由于增强操作的选择以及增强操作的应用幅度是离散化的, 故 Faster AA 为这两种离散化的参数引入了近似梯度, 并采用对抗网络最小化增强数据与原始数据的分布距离. 因此整个搜索过程可以看作是端对端的可微分学习. DADA 则将数据增强策略形式化为子策略范畴分布 (categorical distribution) 的采样问题, 将子策略中每个操作被应用的概率选择转为伯努利分布采样, 然后将上述分布的参数优化问题通过 Gumbel Softmax[15]松弛为可微分的参数优化问题. 上述两个算法均基于可微分机制的思想, 使用随机梯度下降算法实现参数优化, 从而在搜索效率上实现了数量级的提升, 在 CIFAR-10 数据集上搜索耗时仅为 0.23 和 0.1 个 GPU 小时. 虽然这两种算法在搜索耗时方面是最优的, 然而使用可微分估计梯度和真实梯度之间的间隙难以评估, 最终算法在测试集上的预测性能没有明显竞争力.

5) 基于随机数据增强的算法: 针对 Google AA 算法和数据集的强关联关系而导致算法的迁移能力差的缺陷, Google 研究团队提出了一种基于随机数据增强的算法 Rand AA[8]. 该算法不再采用应用概率参数决定是否使用某种子策略, 而是所有的子策略都会以同样的概率被选中应用. Rand AA 算法只定义了两个整数参数 N 和 M, 分别代表增强操作的数量和增强操作的变换幅度, 搜索空间可降低至 M^N (N 一般为 2, 因此搜索空间一般在 10^2 级别左右数量级). 因此, 使用朴素的网格搜索就可以寻找到一组较优的组合解. 然而, 实验发现该算法在测试集的预测准确率并不理想, 而且需要探索更多的参数组合而导致 N 和 M 增加时, 搜索空间复杂度呈指数级上升, 简单的网格搜索将不再适用.

综上所述, 已有自动化数据增强算法在准确率和搜索效率上存在“只顾一头”的问题. 为此, 本文将提出基于自引导进化策略的自动化数据增强算法 SGES AA, 在准确率和搜索效率之间能够实现更好的平衡.

2 背景知识

本节将重点介绍自引导进化策略的相关背景知识. 进化策略是一种无梯度随机优化算法, 针对一般性全局优化问题, 目标函数 F 的梯度 ∇F 通常是无法获取的. 为了估计函数 F 在 θ 处的梯度, 一种常见的方法是使用高斯平滑技术[16]使函数 F 平滑, 如公式 (1) 所示:

$$F_v(\theta)=\mathbb{E}_{\delta\in\mathcal{N}(0,I_n)}[F(\theta+v\delta)]=(2\pi)^{-\frac{n}{2}}\int_{\mathbb{R}^n}F(\theta+v\delta)\mathrm{e}^{-\frac{1}{2}\|\delta\|_2^2}\mathrm{d}\delta \tag{1}$$

其中, $v>0$ 是平滑参数, δ 是均值为 0、维度为 n 的高斯向量, 根据文献 [17] 可以推得函数 F_v 在 θ 的梯度为:

$$\nabla F_v(\theta)=\frac{1}{v}\mathbb{E}_{\delta\in\mathcal{N}(0,I_n)}[F(\theta+v\delta)\delta] \tag{2}$$

然而实际上该梯度仍难以计算, 通常会用蒙特卡洛估计器进行估计, 最常用的两种估计器是一般的进化策略[17]梯度估计:

$$\hat{\nabla}F_v(\theta)=\frac{1}{vN_s}\sum_{i=1}^{N_s}F(\theta+v\delta_i)\delta_i \tag{3}$$

和对偶进化策略梯度[18]估计:

$$\hat{\nabla}F_v(\theta)=\frac{1}{2vN_s}\sum_{i=1}^{N_s}(F(\theta+v\delta_i)-F(\theta-v\delta_i))\delta_i \tag{4}$$

其中, $\{\delta_1, \delta_2, \ldots, \delta_{N_s}\}$称为搜索方向, 根据分布 $\mathcal{N}(0,I_n)$ 进行采样, N_s 为搜索方向数量. 一般来说, 第 2 种策略梯度估计方法比第 1 种拥有更小的方差[18].

为了解决进化策略算法在高维优化问题中固有的高方差问题[16], 自引导进化策略算法[9]应运而生. 自引导进化策略算法在迭代的过程中, 收集了最近 k 次的估计梯度, 并且构建了两个子空间. 假设 $G_t\in\mathbb{R}^{n\times k}$ 定义为在迭代 t 时刻的梯度矩阵, 包含了从迭代 $t-k$ 时刻到 $t-1$ 时刻的估计梯度, 则自引导进化策略算法通过分解矩阵 G 来生成两个子空间: 梯度子空间 $\mathcal{L}_G$ 和其对应的正交补空间 $\mathcal{L}_G^{\perp}$. 直观上来说, $\mathcal{L}_G$ 是一个信息量很大的子空间, 包含了

优化问题的内在结构. 因此, 在采样搜索方向的过程中利用 $\mathcal{L}_G$ 可以提高梯度估计器的质量 (如减少方差等), 尤其在高维优化问题中非常有效.

在采样搜索方向过程中, 自引导进化策略[9]利用了这两个非纠缠子空间对混合概率分布进行了编码, 如公式 (5) 所示:

$$P=\begin{cases}\delta\sim\mathcal{N}(0,I_{\mathcal{L}_G}), & \text{以概率}\alpha\\ \delta\sim\mathcal{N}(0,I_{\mathcal{L}_G}^{\perp}), & \text{以概率}1-\alpha\end{cases} \tag{5}$$

这种采样方式起到了利用和探索 (exploitation & exploration) 的作用, 其中 $\alpha\in(0,1)$ 是优化过程中探索与利用的权衡参数, 即利用作用是搜索方向有 α 的概率是从协方差矩阵为梯度子空间 $\mathcal{L}_G$ 中所获取的多元高斯分布采样而来, 而探索作用则是搜索方向有 $1-\alpha$ 的概率是从协方差矩阵为所述正交补空间 $\mathcal{L}_G^{\perp}$ (或者是从整个空间) 中所获取的多元高斯分布中采样而来.

当从混合概率分布中采样搜索方向时, 使用固定的概率参数 α 可能会导致搜索效率低下, 无法为不同的优化阶段调进行动态调整. 更理想的做法应该是在远离全局最优时应该贪婪地从梯度子空间中采样搜索方向, 而陷入局部最优时应该进行均匀采样, 因此自引导进化策略使用了自适应采样技术, 即根据不同的搜索阶段动态调整 α 值.

3 基于自引导进化策略的自动化数据增强

3.1 问题定义

自动化数据增强算法是一种自动地搜索最优数据增强策略的算法. 自动化数据增强算法首先需要设计一个搜索空间 $\mathbb{O}$, 搜索空间 $\mathbb{O}$ 是数据增强操作 $O:\mathcal{X}\to\mathcal{X}$ 组成的集合, 其中 $\mathcal{X}$ 为输入空间. 每个数据增强操作 $O\in\mathbb{O}$ 包含两个参数: 被调用概率 p (probability) 和数据增强幅度 λ (magnitude), p 表示该增强操作是否会被应用的概率, λ 表示该增强操作对原始数据造成“扭曲”的幅度, 如图像旋转的度数、被裁切的面积等. 令 $\mathcal{S}$ 表示子策略 (sub-policy) 的集合, 其中一个子策略 $\pi\in\mathcal{S}$ 则是由 L 个串联的数据增强操作 $\{\overline{O}_i^{(\pi)}(x;p_i^{(\pi)},\lambda_i^{(\pi)}),i=1,\ldots,L\}$ 组成, 每个操作会有相应的概率被应用到输入的原始数据 x 上, 如公式 (6) 所示:

$$\overline{O}(x:p:\lambda)=\begin{cases}O(x:\lambda), & \text{以概率}p\\ x, & \text{以概率}1-p\end{cases} \tag{6}$$

因此, 子策略 $\pi(x)$ 可被描述为一系列操作的组合, 如公式 (7) 所示:

$$\widetilde{x}_n=\overline{O}_n^{(\pi)}(\widetilde{x}_{(n-1)}),\ n=1,\ldots,L \tag{7}$$

其中, $\widetilde{x}_{(0)}=x,\widetilde{x}_{(L)}=\pi(x)$. 图 2 为在子策略 $\pi(x)$ 下增强图像数据的一个示例. 由于每个子策略 π 是一个随机的依赖于参数 p 和 λ 的增强操作变换序列, 所以一个子策略可以产生大量不同的数据增强效果.

因此, 自动化数据增强算法给出的最终数据增强策略 Π 是 N_π 个子策略的集合, 即对于每个子策略 $\pi\in\Pi$, $\Pi(D_{\text{train}})$ 表示在训练数据集上应用数据增强策略 Π 后的增强数据集合:

$$\Pi(D_{\text{train}})=\bigcup\nolimits_{\pi\in\Pi}\{(\pi(x),y):(x,y)\in D_{\text{train}}\} \tag{8}$$

假设自动化数据增强算法表示为 $\mathcal{A}$, 原始数据集表示为 D (其中训练集为 D_{train}), 则自动化数据增强算法 $\mathcal{A}$ 在数据集 D_{train} 上进行增强后得到优化后的增强策略 Π, 而原始训练数据集 D_{train} 在数据增强策略 Π 下进行变换得到增强数据 $\Pi(D_{\text{train}})$. 假设 $\mathcal{M}$ 代表固定架构的模型, $F(\mathcal{M},D_{\text{train}},D_{\text{test}})$ 代表在模型 $\mathcal{M}$ 上使用训练数据集 D_{train} 进行训练, 在测试数据集 D_{test} 进行评估得到的预测评估值, 则自动化数据增强算法 $\mathcal{A}$ 的优化目标可表示为公式 (9) 所示:

$$\max_{\Pi}\mathbb{E}[F_\xi(\mathcal{M},\Pi(D_{\text{train}}),D_{\text{test}})] \tag{9}$$

其中, 随机变量 ξ 代表了模型 $\mathcal{M}$ 评估增强数据 $\Pi(D_{\text{train}})$ 得到预测结果的随机性. 为了方便表述, 下文对评估函数 F 的使用省略了参数 $\mathcal{M}$ 和 ξ.

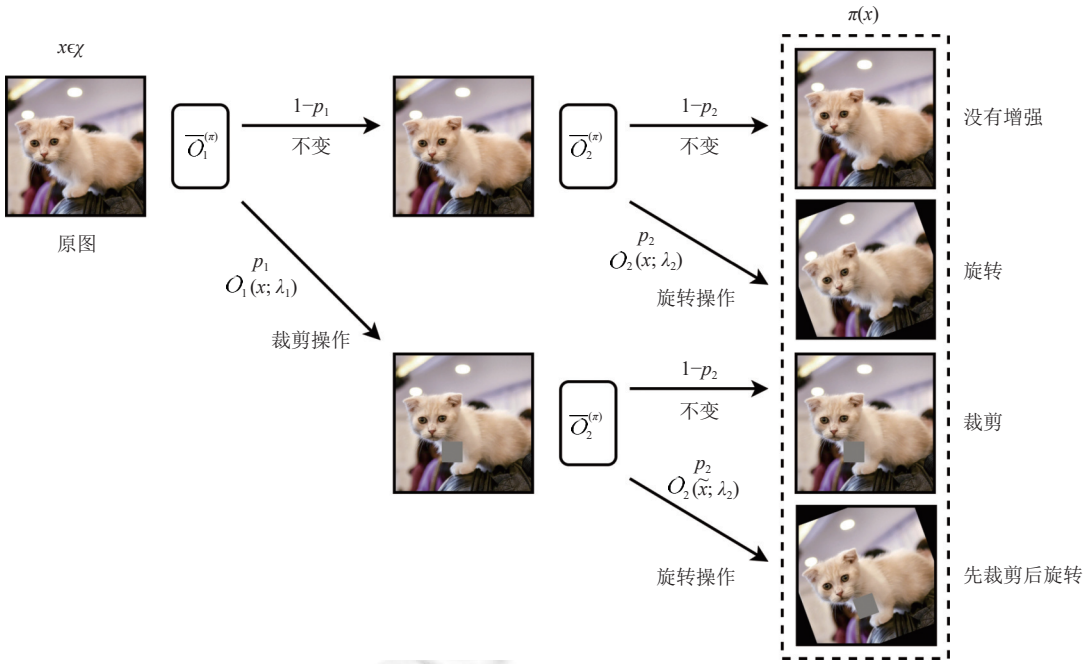


图 2 采用单个子策略 Π 增强图像的示例

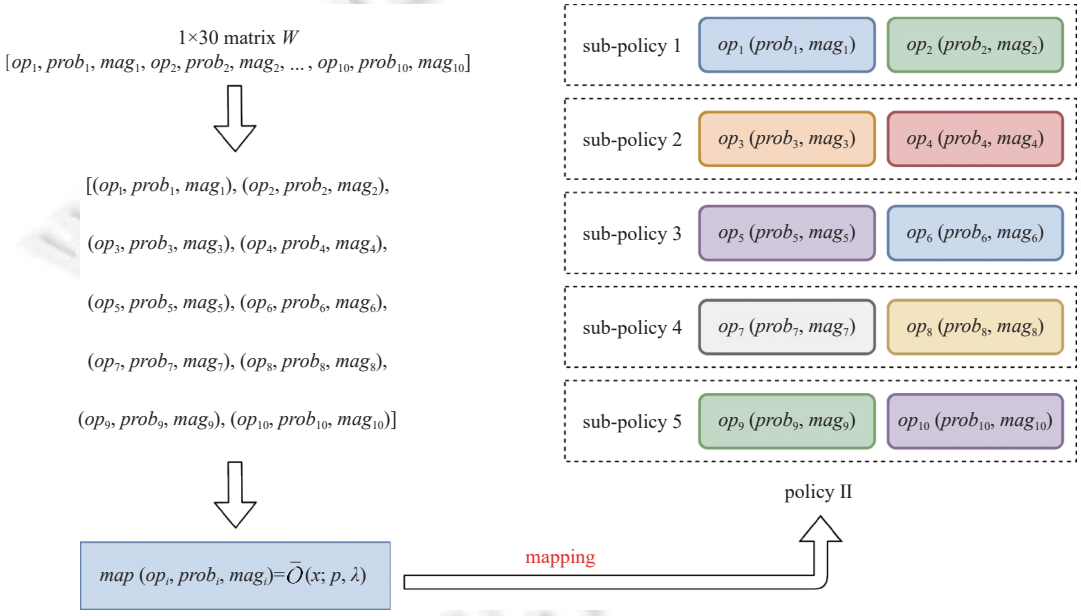


图 3 数据增强策略生成示意图

3.2 数据增强策略的连续化向量表示

本文设计的数据增强策略是由 N_π 个子策略所组成, 一个子策略 $\pi \in \mathcal{S}$ 由 L 个串联的操作 $\{\overline{O}_i^{(\pi)}(x; p_i^{(\pi)}, \lambda_i^{(\pi)}), i = 1,\ldots,L\}$ 所组成, 针对数据增强子策略 π 的定义, 本文将子策略个数 N_π 设置为 5, L 设置为 2. 由于每个操作 O 可以表示为一个三元组<增强函数, 应用概率, 应用幅度>, 因此可将数据增强策略 Π_W 的参数 W 定义为策略向量 $W \in \mathbb{R}^{r\times s}$ 的矩阵, 假设当取 r=1, s=3×2×5=30 时, 那么 W 为一个 1×30 的策略向量. 将策略向量 W 按照每 3 个元素

划分为 10 个 3 维的向量 v, 则向量 v 中的分量分别表示为:

v[0]: 数据增强函数的类型.

v[1]: 数据增强函数的应用概率.

v[2]: 数据增强函数的应用幅度.

为了实现数据增强策略的连续化向量表示, 同时尽可能降低搜索空间的大小, 本文将策略向量 W 中的每一个元素的取值范围均限制在 [0, 1], 并且设计了一个映射函数 $map()$, 函数定义如公式 (10) 所示:

$$map(op, prob, mag) = \overline{O}(x : p, \lambda) \tag{10}$$

其中, op, $prob$, mag 分别为向量 v 的 3 个元素, $\overline{O}(x : p, \lambda)$ 由公式 (6) 所定义. 在该映射转换中, 针对增强函数的类型, 本文将规整后的区间 [0, 1] 进行了均匀划分, 如图像分类任务划分为 15 个均匀的子区间, 分别代表了 15 种不同的数据增强操作, 以此类推语音分类和文本分类分别划分为 6 个均匀的子区间; 针对参数 p 则不需要做较大改动, 因为 p 的取值范围默认为 [0, 1]; 针对参数 λ, 需要根据幅度定义区间的上界 $\lambda_{\max}$ 和下界 $\lambda_{\min}$ 做反归一化, 如公式 (11) 所示:

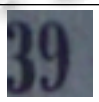
$$\widetilde{\lambda} = \lambda \times (\lambda_{\max} - \lambda_{\min}) + \lambda_{\min} \tag{11}$$

最后, 针对策略向量 W, 本文需要将该权重转换为具体的数据增强策略, 如图 3 所示, 将策略向量 W 按顺序以 3 个元素为一组组成了操作三元组<增强函数, 应用概率, 应用幅度>, 2 个串联的增强操作组成了一个子策略, 5 个子策略组成了最终的数据增强策略 Π.

3.3 搜索空间设计

搜索空间 $\mathbb{O}$ 是数据增强操作 $O: \mathcal{X} \to \mathcal{X}$ 组成的集合, 其中 $\mathcal{X}$ 为输入空间, $x \in \mathcal{X}$ 是一个样本. 每个数据增强操作 O 包含两个参数: 被调用概率 p 和数据增强幅度 λ. 如表 1 所示, 子策略 3 表示一张图像最开始有 32% 的概率应用 Solarize 增强操作, 应用的幅度为 111.87, 接着有 37% 的概率应用 Invert 操作, 然而 Invert 操作不需要幅度信息, 因此搜索出的幅度 0.35 没有实际意义. 根据表 1 可以看出, 本文在数据增强的具体实施方面, 主要是针对每个 batch 的训练数据, 从 5 个子策略中随机抽取 1 个子策略用于该 batch 的每个训练样本, 故不同 epoch 轮次之间相同的 batch 抽取到的子策略可能是不同的, 同时也由于概率参数 p 的存在, 在相同的 batch 中同一训练样本也存在不同的数据增强效果.

表 1 采用 5 个子策略在 SVHN 图像上的增强示例

批次	原始图像	子策略1 Solarize, 0.42, 104.45 Equalize, 0.48, 0.46	子策略2 Equalize, 0.37, 0.57 AutoContrast, 0.54, 0.48	子策略3 Solarize, 0.32, 111.87 Invert, 0.37, 0.35	子策略4 Equalize, 0.61, 0.53 Posterize, 0.24, 6.40	子策略5 Invert, 0.40, 0.47 Color, 0.23, 1.28
Batch 1						
Batch 2						
Batch 3						

针对图像、语音和文本这 3 种媒体数据集, 本文分别设计了对应的数据增强函数方法并定义了本文算法要搜索的策略集合. 针对图像数据, 本文主要使用 AutoAugment[3]提出的 13 种图像增强方法 (去除了 Sample Pairing 方法[19]), 主要包括对图像的平移、旋转、剪切和颜色调整等变换; 针对语音数据集, 主要使用了 6 种语音增强方法, 包括音量调整、音量归一化、语音前后移动和声道调整等; 针对文本数据集, 主要使用了 6 种常见的文本增强方法, 包括错词替换、同义词替换、反义词替换和单词删除等. 增强方法详见附录 A.

3.4 自引导进化策略与自动化数据增强的结合

3.4.1 策略评估与更新

SGES AA 算法在搜索过程中, 需要对相应的连续化策略向量进行映射转换, 得到离散的数据增强策略, 然后再进行策略评估. 策略评估的过程可以视为评价该数据增强策略在具体的模型训练中所表现出来的模型泛化性能. 具体流程为在模型训练的每一个 epoch 期间, 每一批次的训练数据都会经过具体的数据增强子策略 $\pi^{(e,b)}$ 进行数据增强, 而 $\pi^{(e,b)}$ 的选择则是从数据增强策略 Π 中进行随机抽取. 经过一定训练轮数后, 计算模型在验证集上的预测准确率, 并以此判断数据增强策略的增强效果. 通过将预测准确率作为评估值反馈给搜索算法, 进行下一步策略向量的更新.

3.4.2 策略选择

在算法更新迭代过程中, 如果 $F(\Pi_{t,i,+}) > F(\Pi_{t,i,-})$ 时, 则更新步骤会将策略向量 W_t 推向 δ_i 方向, 反之则推向 $-\delta_i$ 方向. 然而, 由于评估函数对策略的评估是一个含噪声的评估, 具有不确定性和随机性, 因此可能会存在即使 δ_i 方向在真实的情况下更好, 然而算法却将 W_t 推向 $-\delta_i$ 方向, 尤其是当 $F(\Pi_{t,i,+})$ 和 $F(\Pi_{t,i,-})$ 都非常小的时候. 因此当评估值 $F(\Pi_{t,i,+})$ 和 $F(\Pi_{t,i,-})$ 都比其他对偶评估值小时, 表明在 δ_i 和 $-\delta_i$ 方向上移动 W_t 会降低平均收益. 为了解决这个问题, 通过对 $\max\{F(\Pi_{t,i,+}), F(\Pi_{t,i,-})\}$ 进行递减排序, 然后取前 $b<N_s$ 个观测点样本的评估值来更新策略向量 W_t.

3.4.3 策略收集与验证

在自引导进化策略算法完成搜索后, 本文根据搜索迭代的历史记录, 从中选取出表现能力最佳的 N_H 个数据增强策略组成数据增强策略集合 H_Π, 其好处是多个数据增强策略集合能够在一定程度上发挥出多种数据增强策略的效果. 在验证阶段, 选择一个最终的验证模型开始从头训练, 与策略评估阶段的子模型训练过程类似, 在每一个 epoch 期间, 每一批次的训练数据都会经过具体的数据增强子策略 $\pi^{(e,b)}$ 来进行数据增强, 而 $\pi^{(e,b)}$ 的选择则是从策略集合 H_Π 中随机抽取选择, 在训练固定轮数后, 模型需要在最终的测试集数据上进行预测 (测试集数据无需任何改变), 最终测试集的模型准确率即可作为自动化数据增强算法的最终评价指标, 以此评估搜索算法的好坏.

3.4.4 并行化设计

自引导进化策略每次迭代搜索将产生 N_s 个搜索方向以及相应方向上的 $2N_s$ 个探索点, 这些探索点对应的策略评估工作也是相互无关的, 故算法可以进行高度并行化, 并行化后的算法主要存在以下两个优点.

1) 每个工作节点只负责对自己当前被分配到的策略进行评估, 相互之间不影响.

2) 每个工作节点除了需要被评估的策略外, 仅需要同步一个随机种子, 这样在评估之前每个工作节点可以感知到其他工作节点使用的扰动噪声值, 因此每个工作节点之间仅需要通信一个标量.

3.5 自动化数据增强算法总体流程

图 4 为本文设计的基于自引导进化策略的自动化数据增强算法总体流程图, 算法 1 给出了详细流程. 首先, 增强随机搜索算法通过在搜索空间 $\mathbb{O}$ 中的初始探索点进行多个方向上的扰动探索, 探索的过程本质是在当前时刻观测样本点 W_t 附近进行参数的噪声扰动后, 进行随机采样.

算法 1. 基于自引导进化策略搜索的自动化数据增强算法 SGES AA.

输入: 更新步长 η, 每次迭代的更新方向数量 N_s, 平滑参数 ν, 迭代索引 t, 迭代次数上限 T, 预热迭代次数 $T_w \geqslant k$, 大小为 k 的队列 Q, 精英数量 $b \leqslant N_s$, 更新间隔 T_U, 容量为 N_H 的最大堆容器 H, 容器 H 中存放的数据增强策略集合 H_Π, 工作节点数量 $m \leqslant N_s$;

输出: 容器 H 中存放的数据增强策略集合 H_Π.

1. 初始化数据增强策略 Π 的策略向量 $W_0 = 0 \in \mathbb{R}^{r \times s}$, $t = 0$

2. **while** $t < T$ **do**

3. **if** $t < T_w$ **then**
4. 从分布 $\mathcal{N}(0, I_n)$ 中独立采样搜索方向$\{\delta_1, \delta_2, \ldots, \delta_{Ns}\}$
5. **else**
6. 从队列 Q 中读取梯度矩阵 $G \in \mathbb{R}^{n\times k}$, 并将矩阵进行 QR 分解生成梯度子空间 $\mathcal{L}_G$ 和正交补空间 $\mathcal{L}_G^{\perp}$
7. 根据公式 (5) 采样搜索方向$\{\delta_1, \delta_2, \ldots, \delta_{Ns}\}$并且归一化
8. **end if**
9. **for** each worker i=1, 2, 3, …, m do
10. 将搜索方向 $\delta_1, \delta_2, \ldots, \delta_{Ns}$ 分配给 m 个工作节点
11. 计算本次迭代所有搜索方向的策略评估值, 其中一次对偶实验使用的数据增强策略分别为:

$$\begin{cases}\Pi_{t,i,+} = \mathcal{M}(W_t + v\delta_i)\\ \Pi_{t,i,-} = \mathcal{M}(W_t - v\delta_i)\end{cases},$$

其中, $i \in \{1,2,\ldots,N_s\}$

12. 使用固定架构的子模型 $\mathcal{M}_{\text{sub}}$ 在使用数据增强策略 $\Pi_{t,i,+}$ 和 $\Pi_{t,i,-}$ 的增强数据上分别进行训练, 然后得到验证集准确率 $F(\Pi_{t,i,+})$ 和 $F(\Pi_{t,i,-})$ 作为评估值
13. **end for**
14. 收集 m 个节点的所有评估值 $F(\Pi_{t,i,+})$ 和 $F(\Pi_{t,i,-})$, 并使用 $\max\{F(\Pi_{t,i,+}), F(\Pi_{t,i,-})\}$ 对方向 δ_i 进行降序排序, 其中 δ_i 表示第 i 个最大的收益方向, $i \in \{1,2,\ldots,b\}$
15. 通过公式 (12) 估计梯度 $\hat{\nabla}F_v(\prod_t)$
16. 通过梯度上升法来更新下一次策略向量 W_{t+1}:

$$W_{t+1} = W_t + \eta\hat{\nabla}F_v(\Pi_t).$$

17. 将估计梯度 $\hat{\nabla}F_v(\prod_t)$ 压入队列 Q
18. 如果 $t \geq T_w$, 需要根据自引导进化策略调整公式 (5) 中的 α 值
19. **if** t % T_U == 0 **then**
20. 将当前的策略向量值 W_{t+1} 转换成数据增强策略 Π_{t+1}
21. **for** each worker i=1, 2, 3, …, m **do**
22. 在子模型 $\mathcal{M}_{\text{sub}}$ 上评估数据增强策略 Π_{t+1} 获得评估值 $F^{(i)}(\prod_{t+1})$
23. **end for**
24. 收集 m 个工作节点的评估值, 计算评估均值:

$$F(\Pi_{t+1}) = \frac{1}{m}\sum_{i=1}^{m} F^{(i)}(\Pi_{t+1}).$$

25. 将 $F(\Pi_{t+1})$ 和策略 Π_{t+1} 更新到堆容器 H 中
26. $t = t+1$
27. **end while**
28. **return** H_{Π}

然后, 算法将策略向量 W_t 转换成为最终的数据增强策略 Π_t, 随后使用该数据增强策略 Π_t 在原始训练数据集 D_{train} 上进行数据增强得到增强数据集 $\Pi_t(D_{\text{train}})$. 策略评估阶段需要使用子模型 $\mathcal{M}_{\text{sub}}$ 对增强的数据集进行训练并评估, 一般采取验证集准确率作为评估值返回给增强随机搜索算法. 最终, 增强随机搜索算法根据多个噪声样本点的评估值对算法的步长和观测样本点策略向量 W_t 进行更新.

在每一次迭代过程时, 算法需要将估计的梯度 $\hat{\nabla}F_v(\prod_t)$ 存入到一个容量为 k 的队列容器 Q, 经过固定的预热迭代次数 T_w 后, 算法需要从队列 Q 中取出 k 个最近的历史评估梯度, 形成一个梯度矩阵 $G_t \in \mathbb{R}^{n\times k}$, 然后生成梯度子空间 $\mathcal{L}_G$ 和其对应的正交补空间 $\mathcal{L}_G^{\perp}$, 根据公式 (5) 去采样搜索方向$\{\delta_1, \delta_2, \ldots, \delta_{Ns}\}$并且归一化. 评估完每个搜索

方向的数据增强策略后, 对得到的所有评估值进行降序排列, 得到前 b 个精英数据增强策略以及相应的评估值, 然后根据公式 (12) 来估计梯度 $\hat{\nabla}F_v(\prod_t)$, 根据估计的梯度使用梯度上升法来更新策略向量 W_{t+1}, 作为下次迭代所需要探索点的策略向量.

$$\hat{\nabla}F_v(\Pi_t)=\frac{1}{2vN_s}\sum_{i=1}^{N_s}(F(W_t+v\delta_i)-F(W_t-v\delta_i))\delta_i \tag{12}$$

最终, 根据预先设计的搜索空间和策略评估方式, 使用自引导进化策略算法在搜索空间中搜索最优的数据增强策略. 由于算法迭代结束时, 并不一定收敛到最优解, 而且策略评估返回的评估值也会存在一定的随机性, 故算法在迭代过程中每经历 T_U 次搜索后都会执行一次观测样本点的策略评估工作, 对应的评估值和数据增强策略将会被更新到一个容量为 N_H 的最大堆容器 H 中. 算法结束时, 只需要将容器 H 内的策略作为最终输出的数据增强策略即可.

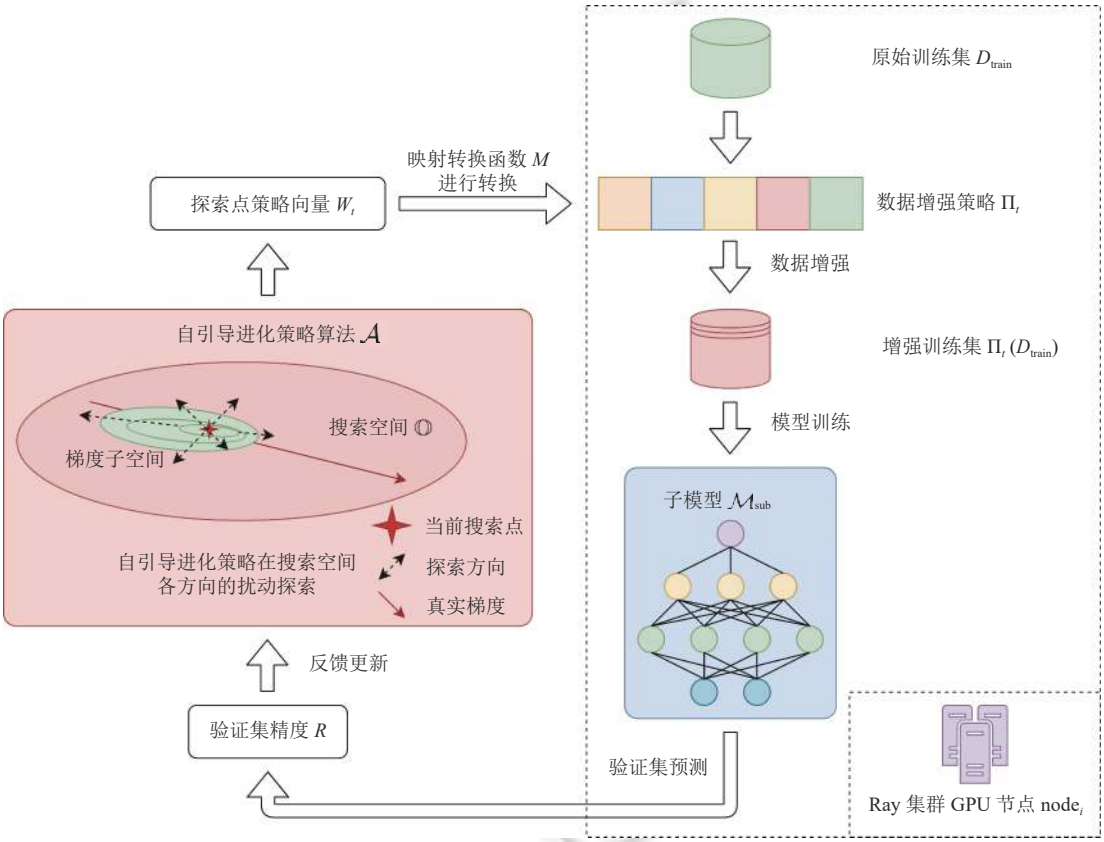


图 4 算法总体流程图

4 实验评估与分析

4.1 实验设置与数据集

本文所有实验环境硬件配置信息如表 2 所示, 工作节点的操作系统为 Ubuntu 18.04, 主要的依赖软件和版本: Python (3.6.8), Torch (1.7.1)、Torchvision (0.8.2)、Torchaudio (0.6.0), Ray (1.2.0), Torch-audiomentations (0.6.0)、nltk (3.5)、nlpaug (1.1.3) 和 librosa (0.8.0) 等.

(1) 图像数据集

实验所用的图像数据集主要是 CIFAR-10[12]、CIFAR-100[12]和 SVHN[20]数据集. 其中 CIFAR-10 数据集为 10 类 32×32 的彩色图像, 一共包含 60 000 张图像, 每一类包含 6 000 张图像, 其中 50 000 张图像作为训练集, 10 000 张图像作为测试集. SVHN 数据集是来自 Google 街景图像中的门牌号, 一共包含 10 个类别, 共包含 73 257 个训练集, 26 032 个测试集.

(2) 语音数据集

实验所用的语音数据集主要是 ESC-50[21]和 GTZAN[22]. 其中 ESC-50 数据集包含了 2 000 个 5 s 环境语音片段, 共分为 50 个类别, 每个类别有 40 个样本, 训练集和测试集按照 4:1 的比例进行划分, 即训练样本 1 600 个, 测试样本 400 个. GTZAN 数据集包含了 1 000 个音频文件, 共分为 10 个流派 (类别), 每个音频文件长 30 s, 训练集和测试集按照 3:1 的比例进行划分, 即训练样本 750 个, 测试样本 250 个.

(3) 文本数据集

实验所用的文本数据集主要是 AGNews[23]和 DBpedia[24]数据集. 其中 AGNews 数据集是学术新闻搜索引擎 ComeToMyHead 从 2 000 多个新闻源收集的新闻文章的集合, 该数据集共有 4 个类别, 包括 120 000 个训练样本和 7 600 个测试样本. DBpedia 数据集是大规模的多语言知识库, 根据 Wikipedia 中最常用的信息框创建的. DBpedia 最受欢迎的版本包含 560 000 个训练样本和 70 000 个测试样本, 每个样本都带有 14 类标签.

表 3 介绍了图像、语音和文本类型数据的基本统计信息, 包括训练样本数目、测试样本数目和标签数目.

表 2 实验环境硬件配置信息

属性	配置描述(单个工作节点)
CPU	20×Intel(R) Xeon(R) Gold 6248 CPU @ 2.50 GHz
GPU	4×Tesla V100 SXM2 32 GB
内存	240 GB (15×16 GB)
硬盘	1 TB HDD
网络	1 Gb/s Ethernet

表 3 图像、语音和文本类型数据的统计信息

类别	数据集	训练样本数目	测试样本数目	标签数目
图像	CIFAR-10	50 000	10 000	10
	CIFAR-100	50 000	10 000	100
	SVHN	73 257	26 032	10
语音	ESC-50	1 600	400	50
	GTZAN	750	250	10
文本	AGNews	120 000	7 600	4
	DBpedia	560 000	70 000	14

4.2 超参数设置

根据算法 1 的设计, 本文对 SGES AA 算法中所需要的超参数进行了设置, 具体设置如表 4 所示.

(1) 图像分类实验超参数设置

针对图像分类任务, CIFAR-10、CIFAR-100 和 SVHN 需要在不同的模型上进行训练, 模型的超参数设置如表 5–表 7 所示. 其中, CIFAR-100 数据集没有单独搜索, 因与 CIFAR-10 数据集比较相似, 故在 CIFAR-100 上的增强策略可通过在 CIFAR-10 上搜索获取.

(2) 语音分类实验超参数设置

针对语音分类任务, ESC-50 和 GTZAN 需要在不同的模型上进行训练, 模型的超参数设置如表 8 所示.

表 4 SGES AA 算法超参数设置

参数项	设定值	参数项	设定值
更新步长 η	0.2	更新间隔 T_U	5
搜索方向数量 N_s	8	容器 H 的容量 N_H	5
平滑参数 v	0.05	工作节点数量 m	8
迭代上限 T	60	预热次数 T_w	16
精英数量 b	6	队列 Q 的容量 k	20

表 5 CIFAR-10 图像分类实验超参数设置

阶段	模型	epochs	learning rate & type	batch size	optimizer
搜索阶段	ResNet-18[25]	60	0.1, cosine	128	SGD
验证阶段	WRN[26] 40×2	200	0.1, cosine	128	SGD
	WRN 28×10	200	0.1, cosine	128	SGD
	Shake-Shake (26, 2×32d)[27]	1 800	0.01, cosine	128	SGD
	Shake-Shake (26, 2×96d)	1 800	0.01, cosine	128	SGD
	Shake-Shake (26, 2×112d)	1 800	0.01, cosine	128	SGD

表 6 CIFAR-100 图像分类实验超参数设置

阶段	模型	epochs	learning rate & type	batch size	optimizer
验证阶段	WRN 40×2	200	0.1, cosine	128	SGD
	WRN 28×10	200	0.1, cosine	128	SGD
	Shake-Shake (26, 2×96d)	1 800	0.01, cosine	128	SGD

表 7 SVHN 图像分类实验超参数设置

阶段	模型	epochs	learning rate & type	batch size	optimizer
搜索阶段	ResNet-18	60	0.01, cosine	128	SGD
验证阶段	WRN 28×10	200	0.01, cosine	128	SGD
	Shake-Shake (26, 2×96d)	200	0.01, cosine	128	SGD

表 8 ESC-50 和 GTZAN 语音分类实验超参数设置

阶段	模型	epochs	learning rate	batch size	optimizer	pretrained
搜索阶段	Inception[28]	5	0.000 1	32	Adam[29]	True
验证阶段	ResNet	70	0.000 1	32	Adam[29]	True
	Inception	70	0.000 1	32	Adam[29]	True
	DenseNet[30]	70	0.000 1	32	Adam[29]	True

(3) 文本分类实验超参数设置

针对文本分类任务, AGNews 和 DBpedia 需要在不同的模型上进行训练, 模型的超参数设置如表 9 所示.

表 9 AGNews 和 DBpedia 文本分类实验超参数设置

阶段	模型	epochs	learning rate	batch size	optimizer
搜索阶段	fastText[31]	2	0.001	64	Adam
验证阶段	fastText	10	0.001	64	Adam
	TextCNN[32]	5	0.001	64	Adam
	Bi-LSTM+Attention[33]	5	0.001	64	Adam
	Transformer[34]	10	0.001	64	Adam

4.3 实验结果与分析

实验主要对比分析了 SGES AA 与现有的其他自动化数据增强算法, 包括 Google AA、Fast AA、PBA、Faster AA、DADA、Rand AA 等. 同时, 本文还分别实现了 DADA 和 Rand AA 两种算法在语音和文本数据集的实验验证, 这两种算法均为近两年代表性的开源算法, DADA 是可微分自动化数据增强的代表性算法, Rand AA 是随机搜索的代表性算法, 此外这两种算法的扩展性较强, 通过少量的修改源代码即可灵活地支持针对语音分类

和文本分类任务的自动化数据增强.

另外, 为了验证自引导进化策略的有效性, 本文也实现了基于增强随机搜索[35]的算法 ARS AA (augmented random search for auto-augment), ARS AA 同样采用了本文设计的搜索空间. 由于搜索耗时的对比已在图 1 显示, 本节将重点关注增强策略在测试集上准确率指标. 所有实验均运行 3 次, 并统计准确率均值和标准差.

4.3.1 图像分类实验

表 10–表 12 为不同模型和算法在 CIFAR-10、CIFAR-100 和 SVHN 这 3 个图像数据集上的测试准确率结果对比. 实验结果表明, SGES AA 算法在图像数据集上取得了优于或者匹配目前最优算法的性能. 其中, 在 CIFAR-10 数据集上平均排名第 1 (参与排名的算法共 10 个). 在 CIFAR-100 和 SVHN 数据集上平均排名为第 2.

表 10 SGES AA 与其他算法在 CIFAR-10 数据集上不同模型的测试准确率 (%)

算法	WRN 40×2	WRN 28×10	Shake-Shake (26, 2×32d)	Shake-Shake (26, 2×96d)	Shake-Shake (26, 2×112d)	Average rank
Baseline*	94.70	96.10	96.40	97.10	97.20	10/10
Cutout*	95.90	96.90	97.00	97.40	97.40	9/10
Google AA	96.40	97.32	**97.50**	**98.04**	**98.11**	2/10
Fast AA*	96.31	97.26	97.24	97.55	97.78	5/10
PBA	—	**97.42**	97.46	97.97	97.97	3/10
DADA*	96.14	97.11	97.32	97.80	97.83	7/10
Faster AA*	96.26	97.23	97.18	97.78	97.82	6/10
Rand AA*	95.67	96.63	96.93	97.54	97.88	8/10
ARS AA	96.46±0.008	97.30±0.010	97.43±0.012	97.87±0.008	97.95±0.006	4/10
SGES AA	**96.63**±0.003	97.39±0.006	97.45±0.010	**98.04**±0.006	98.09±0.004	**1/10**
Rank	1/9	2/10	3/10	1/10	2/10	—

注: 右上角标*表示该结果是使用该算法官方代码或第三方开源代码运行而来, 未标*则表示取自原论文实验结果. 加粗表示实验结果在同组内最好. Rank代表SGES AA算法在所有自动化数据增强算法中的排名

表 11 SGES AA 算法和其他算法在 CIFAR-100 数据集上不同模型的测试准确率 (%)

算法	WRN 40×2	WRN 28×10	Shake-Shake (26, 2×96d)	Average rank
Baseline*	74.21	81.22	82.89	10/10
Cutout*	74.83	81.58	84.07	9/10
Google AA	**79.28**	82.88	**85.71**	**1/10**
Fast AA*	79.02	81.78	84.65	6/10
PBA	—	**83.27**	84.69	3/10
DADA*	78.87	81.88	84.67	5/10
Faster AA*	77.90	81.96	84.42	8/10
Rand AA*	78.36	82.28	84.30	7/10
ARS AA	79.10±0.015	82.68±0.011	84.75±0.013	4/10
SGES AA	79.07±0.016	83.11±0.008	84.80±0.010	2/10
Rank	3/9	2/10	2/10	—

注: 右上角标*表示该结果是使用该算法官方代码或第三方开源代码运行而来, 未标*则表示取自原论文实验结果. 加粗表示实验结果在同组内最好. Rank代表SGES AA算法在所有自动化数据增强算法中的排名

表 12 SGES AA 算法和其他算法在 SVHN 数据集上不同模型的测试准确率 (%)

算法	WRN 28×10	Shake-Shake (26, 2×96d)	Average rank
Baseline*	98.47	98.56	10/10
Cutout*	98.66	98.67	9/10
Google AA	**98.87**	**98.96**	**1/10**
Fast AA*	98.72	98.74	6/10
PBA	98.82	98.87	3/10
DADA*	98.70	98.74	5/10
Faster AA*	98.68	98.71	7/10
Rand AA*	98.60	98.28	8/10
ARS AA	98.77±0.008	98.75±0.005	4/10
SGES AA	**98.87±0.007**	98.94±0.005	2/10
Rank	1/10	2/10	—

注: 右上角标*表示该结果是使用该算法官方代码或第三方开源代码运行而来, 未标*则表示取自原论文实验结果. 加粗表示实验结果在同组内最好. Rank代表SGES AA算法在所有自动化数据增强算法中的排名

4.3.2 语音分类实验

表 13 为不同的模型与算法在 ESC-50 和 GTZAN 数据集上的测试准确率结果, 其中 Random 代表数据增强策

略参数随机选择. 实验结果表明, SGES AA 算法在语音分类任务的 ESC-50 和 GTZAN 上均达到了最佳的数据增强效果.

表 13 SGES AA 算法和其他算法在语音数据集上的准确率对比 (%)

数据集	算法	ResNet	Inception	DenseNet
ESC-50	Baseline	84.75±0.13	81.25±0.14	87.15±.012
	Random	86.20±0.18	81.68±0.21	88.29±0.17
	DADA	89.45±0.26	83.25±0.44	89.75±0.76
	Rand AA	88.58±0.31	83.75±0.54	89.08±0.82
	ARS AA	88.74±0.11	83.85±0.20	89.92±0.33
	SGES AA	**90.06±0.11**	**84.46±0.18**	**90.23±0.20**
GTZAN	Baseline	91.60±0.09	85.21±0.14	92.40±0.10
	Random	91.55±0.21	86.20±0.18	92.93±0.17
	DADA	91.75±0.26	86.80±0.40	92.42±0.28
	Rand AA	91.50±0.52	85.87±1.54	92.80±0.32
	ARS AA	91.83±0.14	87.44±0.11	92.88±0.10
	SGES AA	**92.23±0.07**	**88.40±0.10**	**93.62±0.09**

4.3.3 文本分类实验

表 14 为不同的模型与算法在 AGNews 和 DBpedia 数据集上的测试准确率结果, 其中 Random 代表数据增强策略参数随机选择. 实验结果表明, SGES AA 算法在文本分类任务的 AGNews 上达到了最佳的数据增强效果, 除了 DBpedia 的 TextCNN 模型比使用增强随机搜索的 ARS AA 算法稍差之外, 其他模型均达到了最佳的数据增强效果. 由于分类的准确率已经较高, 相对于图像分类和语音分类, 数据增强对于文本分类任务的作用效果比较微弱, 而且从随机的数据增强策略也可以发现, 坏的数据增强策略反而会降低文本分类模型的预测能力和泛化能力.

表 14 SGES AA 算法和其他算法在文本数据集上的准确率对比 (%)

数据集	算法	fastText	TextCNN	Bi-LSTM+Attention	Transformer
AGNews	Baseline	90.59±0.06	91.94±0.06	91.34±0.10	90.93±0.04
	Random	90.67±0.06	91.87±0.05	90.89±0.09	91.02±0.06
	DADA	91.57±0.08	91.75±0.13	90.98±0.28	91.64±0.29
	Rand AA	91.04±0.12	90.92±0.17	90.41±0.06	91.57±0.14
	ARS AA	91.49±0.08	92.11±0.06	91.10±0.08	92.06±0.05
	SGES AA	**91.66±0.05**	**92.60±0.07**	**91.56±0.08**	**92.12±0.04**
DBpedia	Baseline	97.71±0.02	98.61±0.05	98.70±0.08	98.27±0.06
	Random	97.88±0.04	98.10±0.05	98.26±0.10	97.95±0.08
	DADA	97.74±0.08	98.56±0.07	98.78±0.13	98.34±0.10
	Rand AA	97.83±0.12	98.49±0.04	98.74±0.10	98.35±0.09
	ARS AA	97.92±0.04	**98.76±0.06**	98.82±0.10	98.44±0.04
	SGES AA	**98.04±0.05**	98.72±0.06	**98.94±0.07**	**98.62±0.09**

4.4 策略评估对实验准确率的影响分析

在 SGES AA 算法的策略评估流程中, 算法需要将策略向量转换为具体的数据增强策略, 然后再对该策略进行评估. 不同的评估网络架构以及评估网络架构训练轮次均会对策略的评估产生不同的评估值, 从而最终影响到算法输出的数据增强策略在测试集上的准确率. 本文分别在 CIFAR-10、ESC-50 和 AGNews 数据集上, 针对不同的评估网络架构和不同的网络架构训练轮次这两个因素进行了相关实验.

由表 15 可以看出, 在固定评估网络训练轮数的情况下, 针对两种不同的评估网络架构, 即 ResNet-18 (网络模型结构相对更简单) 与 WRN 40×2 (网络模型结构相对更复杂), SGES AA 得到的数据增强策略对于最终训练模型的测试集准确率影响不显著. 但是, 由于简单的网络架构模型在训练时更加快速, 导致最终自动化数据增强算法所

需的搜索耗时差异较大, 如在训练轮数固定为 60 轮时, ResNet-18 耗时 4.8 h, 远远小于 WRN 40×2 所需的 21.85 h, 但是两者最终输出的数据增强策略在 WRN 40×2 和 WRN 28×10 上的测试集准确率表现差异比较小. 同时, 在固定评估网络架构的情况下, 随着评估网络训练轮数的增加, 算法输出的数据增强策略在最终的训练模型上表现均有提升, 但是 60 轮至 90 轮的准确率提升较小, 90 轮至 120 轮的准确率几乎不再变化.

表 15 在 CIFAR-10 上评估网络架构和训练轮数对 SGES AA 准确率的影响对比

评估网络架构	评估网络训练轮数	搜索耗时 (h)	最终训练模型的测试集准确率 (%)	
			WRN 40×2	WRN 28×10
ResNet-18	30	2.56	96.52±0.009	97.29±0.014
	60	4.80	96.63±0.003	97.39±0.006
	90	7.48	96.63±0.006	97.38±0.010
	120	9.98	96.64±0.005	97.39±0.008
WRN 402	30	11.14	96.54±0.015	97.30±0.013
	60	21.85	96.64±0.008	97.39±0.011
	90	33.08	96.64±0.007	97.39±0.010
	120	44.24	96.65±0.009	97.38±0.008

表 16 和表 17 说明了 SGES AA 算法在 ESC-50 和 AGNews 数据集上使用不同的评估网络架构以及不同的训练轮数对最终测试网络架构准确率的影响对比结果. 与 CIFAR-10 数据集上的结果类似, 即在固定训练轮数时, 评估网络的架构越复杂, 算法输出的数据增强策略在最终的训练模型上会有小的提升, 但是复杂的评估网络模型需要耗费更多的搜索时间. 在固定评估网络架构时, 随着评估网络训练轮数的增加, 算法输出的数据增强策略在最终的训练模型上表现持平或有小幅提升. 因此, 本文针对 SGES AA 算法在数据集 CIFAR-10 上所选择的评估网络架构模型为 ResNet-18, 并将训练轮数固定为 60 轮; 在数据集 ESC-50 上所选择的评估网络架构模型为 Inception, 并将训练轮数固定为 5 轮; 在数据集 AGNews 上所选择的评估网络架构模型为 fastText, 并将训练轮数固定为 2 轮, 这样的选择可以在兼顾算法表现能力的同时缩减算法所需的耗时.

表 16 在 ESC-50 上使用 Inception 评估网络架构和训练轮数对 SGES AA 准确率的影响对比

评估网络架构	评估网络训练轮数	搜索耗时 (h)	最终训练模型的测试集准确率 (%)	
			Inception	DenseNet
Inception	3	1.60	83.70±0.22	89.74±0.16
	5	3.13	84.46±0.18	90.23±0.20
	7	4.41	84.46±0.14	90.23±0.19
	9	5.80	84.48±0.13	90.25±0.18
ResNet	3	1.82	83.73±0.18	89.41±0.10
	5	3.64	84.49±0.12	90.25±0.08
	7	5.37	84.50±0.13	90.26±0.06
	9	6.78	84.49±0.10	90.27±0.09

表 17 在 AGNews 上评估网络架构和训练轮数对 SGES AA 准确率的影响对比

评估网络架构	评估网络训练轮数	搜索耗时 (h)	最终训练模型的测试集准确率 (%)	
			fastText	Transformer
fastText	1	0.38	91.35±0.14	91.90±0.10
	2	0.72	91.66±0.05	92.12±0.04
	3	1.28	91.68±0.09	92.12±0.05
	4	1.81	91.68±0.11	92.13±0.04
Transformer	1	2.20	91.37±0.12	91.90±0.14
	2	4.53	91.66±0.06	92.13±0.09
	3	6.87	91.67±0.04	92.14±0.16
	4	9.05	91.68±0.10	92.14±0.11

4.5 子策略对实验准确率的影响分析

4.5.1 策略个数对实验准确率的影响分析

在 SGES AA 算法设计中, 策略个数 N_H 作为算法超参数影响着最终输出数据增强策略的表现效果. 由算法 1 可知, 最终算法返回的容器 H 中存放的是最优表现的数据增强策略集合. 图 5–图 10 分别为不同策略个数在 CIFAR-10、ESC-50 和 AGNews 数据集下所得到的测试集准确率. 图中阴影是误差带阴影, 每个实验运行 3 次, 统计均值 (黑点) 和标准差. 可以看出, 随着策略个数的增加, SGES AA 算法的准确率也会逐渐提升, 而在 5 个数据增强策略时保持了最好的准确率, 继续增加策略个数反而会降低准确率. 上述结果表明在某个范围内增加策略个数

有助于增强数据的丰富性, 而过多的数据增强策略在训练阶段由于随机选择的特点导致引入更多的数据噪声, 致使最终训练得到的模型预测性能下降.

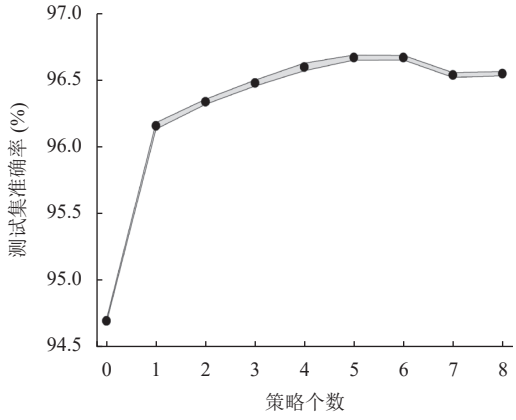


图 5 不同策略个数在 CIFAR-10 数据集使用 WRN 40×2 模型下得到的测试集准确率对比

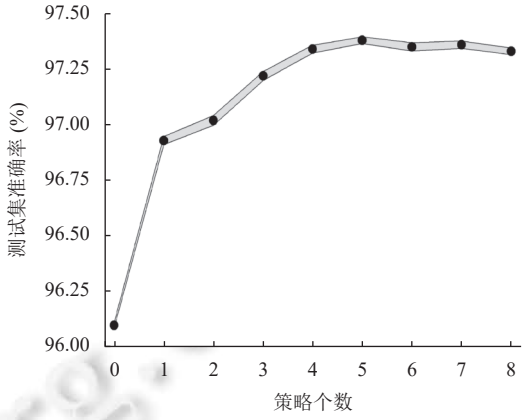


图 6 不同策略个数在 CIFAR-10 数据集使用 WRN 28×10 模型下得到的测试集准确率对比

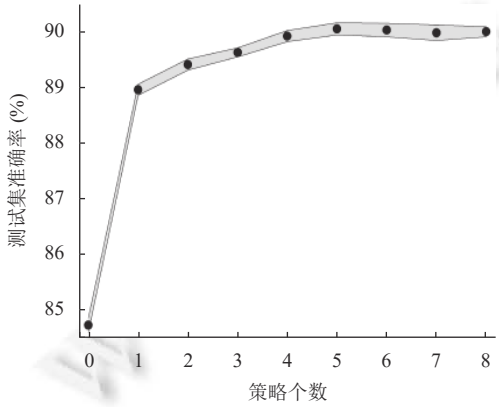


图 7 不同策略个数在 ESC-50 数据集使用 ResNet 模型下得到的测试集准确率对比

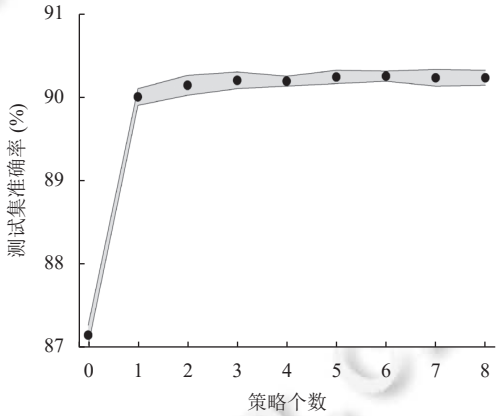


图 8 不同策略个数在 ESC-50 数据集使用 DenseNet 模型下得到的测试集准确率对比

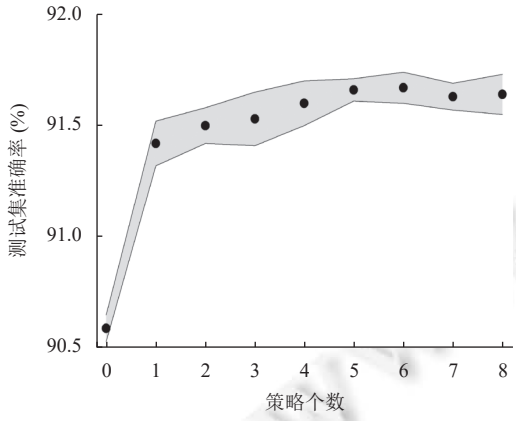


图 9 不同策略个数在 AGNews 数据集使用 fastText 模型下得到的测试集准确率对比

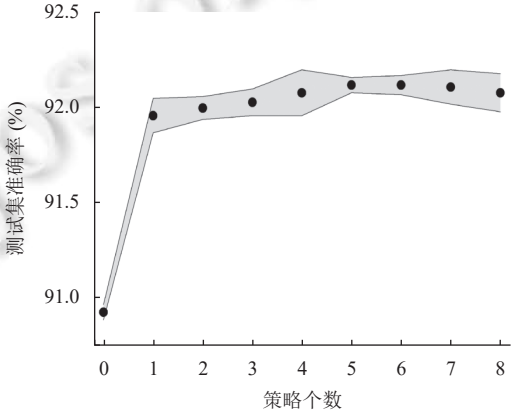


图 10 不同策略个数在 AGNews 数据集使用 Transformer 模型下得到的测试集准确率对比

4.5.2 子策略中操作函数个数对实验准确率的影响分析

在 SGES AA 算法设计中, 一个子策略包含了 L 个数据增强操作. 图 11–图 16 分别为不同子策略操作个数在 CIFAR-10、ESC-50 和 AGNews 数据集使用两种模型所得到的测试集准确率. 可以看出, 一个子策略中包含 2 个操作函数均为最合适的选择, 过少的数据操作函数导致数据增强影响不足, 而过多的数据增强操作函数导致对数据的修改幅度过大, 容易引起更大的数据噪声甚至修改数据的原有特征.

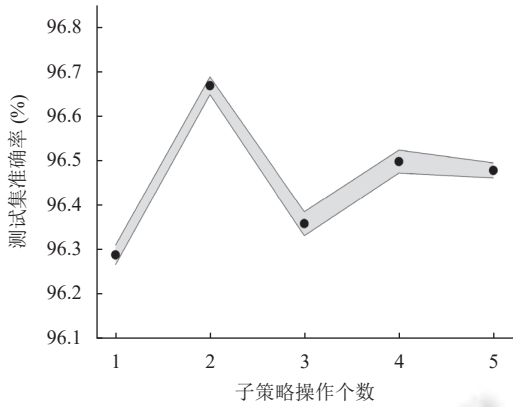


图 11 子策略操作个数在 CIFAR-10 数据集使用 WRN 40×2 模型下得到的测试集准确率对比

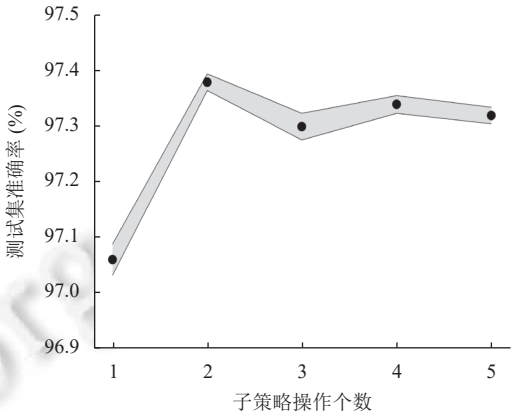


图 12 子策略操作个数在 CIFAR-10 数据集使用 WRN 28×10 模型下得到的测试集准确率对比

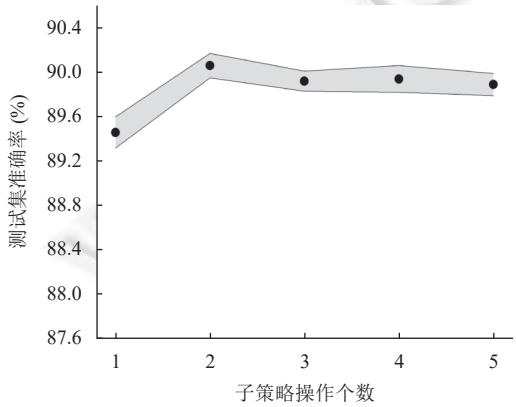


图 13 不同子策略操作个数在 ESC-50 数据集使用 ResNet 模型下得到的测试集准确率对比

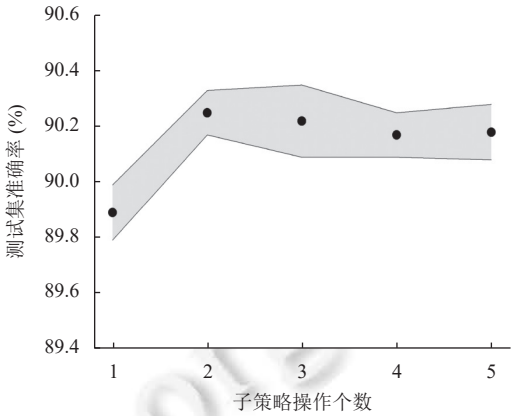


图 14 不同子策略操作个数在 ESC-50 数据集使用 DenseNet 模型下得到的测试集准确率对比

4.5.3 策略向量构造方式对实验准确率的影响分析

本文提出的 SGES AA 算法首先将数据增强策略表示成连续化的向量, 然后基于自引导进化策略实现策略向量的搜索. 在策略向量构造过程中, 对于增强函数类型, SGES AA 将区间 [0, 1] 均匀划分为与增强函数数量相同的份数, 并将特定区间上的实数值映射到某个增强函数. 为了验证不同策略向量构造方式对准确率的影响, 本文进一步实现了基于 one-hot 的策略向量构造方法 (SGES AA-one hot), 采用 one-hot 向量对增强函数类型进行表示. 令 N 为增强函数的数量, 在 SGES AA-one hot 中, <增强函数, 应用概率, 应用幅度> 三元组中的增强函数维度从 R 变为 R^N. 表 18 显示了不同策略向量构造方式在 CIFAR-10 数据集上的准确率.

从表 18 中可以看到, SGES AA-one hot 在多个模型上的表现不如 SGES AA, 基于区间表示的方法优于基于 one-hot 的表示方法. 对于 SGES AA-one hot, 其策略向量大小约是原策略向量的 N 倍 (本文 $N=15$), 大大增加了需要优化的策略参数数量, 导致搜索过程的不稳定以及难以收敛. 另外, 每一次搜索迭代结束后, 还需要对 one-hot 向量进行

离散化, 选择 one-hot 中最大值对应的增强函数作为实际使用的增强函数, 离散化过程也将导致一定的性能偏差. 实际上, 已有的自动化数据增强算法, 例如 Google AA[3]、PBA[4]等均采用了区间表示增强函数的方法, 并取得了不错的效果.

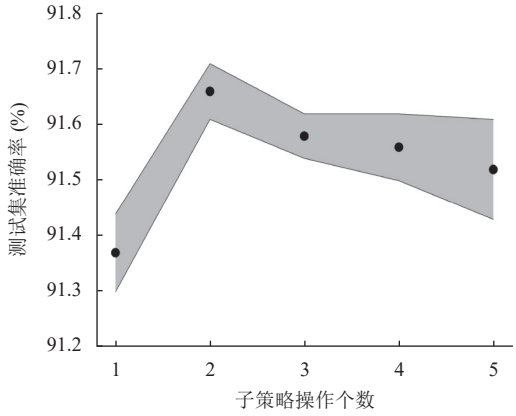


图 15 不同子策略操作个数在 AGNews 数集使用 fastText 模型下得到的测试集准确率对比

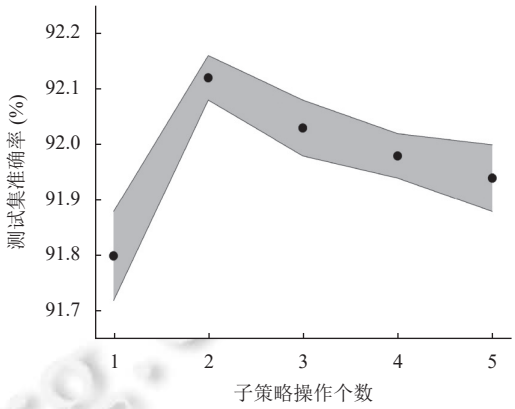


图 16 不同子策略操作个数在 AGNews 数集使用 Transformer 模型下得到的测试集准确率对

表 18 不同策略向量构造方式在 CIFAR-10 上的准确率 (%)

策略向量构造方式	WRN 40×2	WRN 28×10	Shake-Shake (26, 2×32d)	Shake-Shake (26, 2×96d)	Shake-Shake (26, 2×112d)
SGES AA	**96.63**	**97.39**	**97.45**	**98.04**	**98.09**
SGES AA-one hot	96.10	97.21	96.61	97.32	97.38

4.6 算法鲁棒性分析

在 SGES AA 算法设计中, 需要为随机数生成随机种子值, 该随机种子值固定时, 相关随机数生成均保持一致, 可用于算法的复现验证. 为了探索随机种子对实验准确率的影响, 本文进行了相关的实验探索. 表 19 为在 CIFAR-10、ESC-50 和 AGNews 数据集上使用 SGES AA 算法配置不同的随机种子所产生的数据增强策略, 每一个数据增强策略在单独的模型上分别进行了 3 次实验. 本文使用了 3 个不同的随机种子进行了自动化数据增强实验, 随后在两种不同的模型上进行了训练和评估. 从表 19 可以看出, 不同的随机种子对最终实验结果的影响较小, 故侧面反映出 SGES AA 算法具有较强的鲁棒性.

4.7 算法收敛性分析

在自动化数据增强中, 搜索空间、搜索策略和评估方法是 3 个最重要的因素. 为了评估算法的收敛性, 本节对比分析了 SGES AA 和 ARS AA, ARS AA 的搜索空间和评估方法与 SGES AA 一致, 但采用的是增强随机搜索方法, 缺少历史梯度信息的指导. 图 17 为 SGES AA 算法和 ARS AA 算法在搜索过程中, 根据每一次迭代观察到的评估值取均值所绘制的收敛曲线图, 数据集为 CIFAR-10.

从图 17 可以看到 ARS AA 和 SGES AA 算法观察到的评估值总体呈上升的趋势, 说明了两种算法在搜索过程中能够有效对最优策略进行逼近, 最终算法在某个迭代次数过后渐渐趋于收敛, 印证了本文提出的搜索空间和评估策略的合理性. 从迭代次数 40–60 之间的评估值均值可以看出, SGES AA 算法的收敛性和表现能力在一定程度上优于 ARS AA 算法, 验证了自引导进化策略的有效性.

此外, 本文也分析了其他自动化数据增强算法的收敛性, 包括基于可微分机制的 DADA、基于随机数据增强的 Rand AA 以及基于种群训练的 PBA, 3 种算法的收敛性评估结果如图 18 所示. DADA 使用双层优化的方法交替优化数据增强策略选择参数和模型参数, 两种参数耦合度高, 导致其选出次优的增强策略. Rand AA 每次迭代过程从给定数据增强策略池中随机选择数据增强策略, 然后进行模型训练, 其收敛速度只能反映网络模型本身的收敛速度, 无法反映数据增强策略优劣. PBA 在迭代次数为 55 左右时, 其验证集准确率才达到 SGES AA 水平, 表明其收敛性不如 SGES AA 优越.

表 19 随机数种子对准确率的影响 (%)

数据集	测试模型	3次实验准确率	均值与标准差
CIFAR-10	WRN 40×2	96.63, 96.63, 96.64	96.63±0.006
		96.63, 96.64, 96.62	96.63±0.007
		96.64, 96.65, 96.64	96.64±0.006
	WRN 28×10	97.39, 97.37, 97.38	97.38±0.006
		97.40, 97.40, 97.38	97.39±0.010
		97.38, 97.37, 97.39	97.38±0.009
ESC-50	ResNet	89.96, 90.04, 90.18	90.06±0.09
		90.25, 89.95, 89.96	90.05±0.14
		90.19, 90.03, 89.95	90.06±0.10
	DenseNet	90.39, 90.18, 90.18	90.25±0.10
		90.36, 90.14, 90.26	90.25±0.09
		90.40, 90.18, 90.20	90.26±0.10
AGNews	fastText	91.83, 91.54, 91.58	91.65±0.13
		91.51, 91.74, 91.70	91.65±0.10
		91.60, 91.60, 91.81	91.67±0.10
	Transformer	92.13, 92.04, 92.19	92.12±0.06
		91.99, 92.20, 92.20	92.13±0.10
		92.13, 92.18, 92.06	92.12±0.05

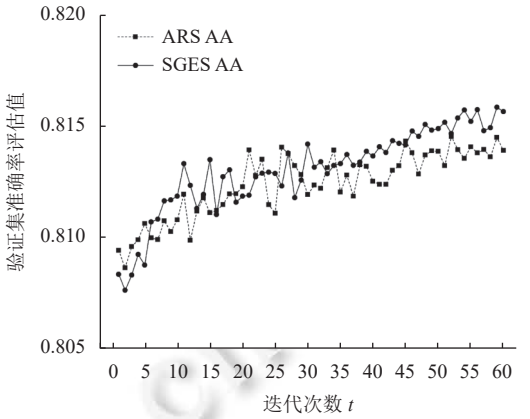


图 17 SGES AA 和 ARS AA 算法在 CIFAR-10 数据集上搜索过程评估均值变化

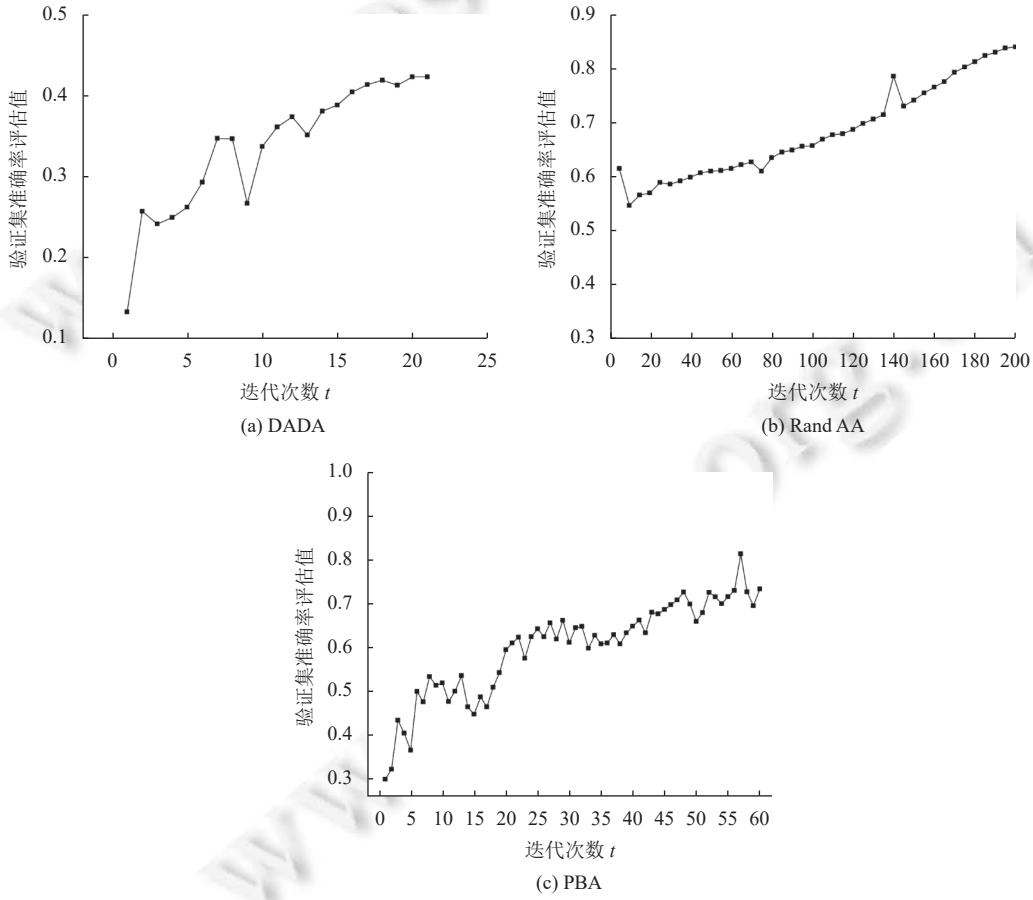


图 18 不同自动化数据增强算法的收敛性

5 讨 论

为了更好地说明本文提出的自动化数据增强算法 SGES AA 能够在搜索耗时和模型准确率方面实现更好的平衡, 本节对不同算法的时间复杂度以及模型准确率进行了综合对比分析. 由于不同自动化数据增强算法的设计原则以及技术细节存在较大差异, 难以准确计算每个算法的时间复杂度. 在自动化数据增强算法的搜索阶段, 搜索策略自身的计算开销往往占比不大, 模型的训练评估开销通常占据了大部分的计算开销. 为此, 本节以模型参数训练次数估计各个算法的时间复杂度. 表 20 对比分析了不同自动化数据增强算法在搜索阶段时间复杂度以及模型准确率, N_{epoch} 和 C_{epoch} 分别代表模型训练的 epoch 数量及每个 epoch 的时间开销.

表 20 不同自动化数据增强算法在搜索阶段时间复杂度以及模型准确率方面的对比分析

自动化数据增强算法	设计原则	时间复杂度分析		模型准确率分析	
		影响因素	时间开销大小	影响因素	准确率
AutoAugment	强化学习	RNN控制器收敛需要T次迭代, 每次迭代需要一次模型训练, 总的时间复杂度为$T\times N_{epoch}\times C_{epoch}$ (T约为15 000, N_{epoch}为120)	时间开销巨大 (5 000 GPU小时)	通过模型训练, 直接评估增强策略的好坏	高
PBA	种群训练	种群数量为P, 总的时间复杂度为$P\times N_{epoch}\times C_{epoch}$ (P为16, N_{epoch}为200)	时间开销一般 (5 GPU 小时)	通过模型训练, 直接评估增强策略的好坏	较高
Fast AA	密度匹配	训练集划分为K份, 总的时间复杂度为$K\times N_{epoch}\times C_{epoch}$ (K为5, N_{epoch}为200)	时间开销低 (3.5 GPU 小时)	在验证集上间接评估增强策略的好坏, 缺少理论解释	一般
Faster AA	可微分	迭代次数为T, 每次迭代需要一次模型训练, 总的时间复杂度为$T\times N_{epoch}\times C_{epoch}$ (T为200, N_{epoch}为20)	时间开销低 (1.5 GPU 小时)	将离散问题近似可微, 然后再将连续的增强策略选择/幅度参数离散化, 导致性能偏差	一般
DADA	可微分+梯度估计	迭代次数为T, 每次迭代需要一次模型训练, 总的时间复杂度为$T\times N_{epoch}\times C_{epoch}$ (T为20, N_{epoch}为15)	时间开销低 (0.1 GPU 小时)	将离散问题近似可微, 然后再将连续的增强策略选择/幅度参数离散化, 导致性能偏差	一般
Rand AA	随机网格搜索	在全量数据集上进行训练, 每个epoch随机选择数据增强策略, 总的时间复杂度为$N_{epoch}\times C_{epoch}$ (N_{epoch}为200)	时间开销低 (0.23 GPU 小时)	随机搜索数据增强方法, 缺乏理论指导	低
SGES AA	自引导进化策略	迭代次数为T, N_s为每次迭代搜索方向数量, 每个方向上对偶执行两次模型训练, 总的时间复杂度为$T\times N_S\times 2\times N_{epoch}\times C_{epoch}$ (N 为8, N_{epoch}为60, 并发度m为8)	时间开销中等 (4.8 GPU 小时)	通过模型训练, 直接评估增强策略的好坏, 并引入历史梯度信息, 指导增强策略的搜索方向	高

另外, 为了能够让自引导进化策略更好地指导数据增强策略的搜索, 本文也从多个层面进一步改进自引导进化策略, 使其能够更好地适应于数据增强策略的搜索. 改进点主要包含以下几个方面.

1) 在策略向量的构造方面, 除了增强策略参数外, 将模型的架构参数以及部分超参数嵌入到策略向量中[36,37], 充分发挥自引导进化策略能够高效处理高维参数向量的优势, 从而实现增强策略参数以及架构参数或模型超参数的联合优化, 进一步提升模型准确率.

2) 在自引导进化策略的优化方面, 可以动态设置历史梯度矩阵的梯度数量, 令 k 为采用的历史梯度的数量. 在增强策略搜索的起始阶段, 由于可利用的信息较少, 可采用较大的 k 值. 当搜索过程逐渐收敛时, 可以采用较小的 k 值, 保证模型的收敛. 通过在搜索过程中动态改变 k 的取值, 提升自引导进化策略的性能. 另外, 也可以在自引导进化策略的基础上, 使用多目标遗传策略[38]同时优化模型准确率以及模型的计算性能.

6 总结与展望

近年来, 自动化数据增强技术引起了学界和业界的广泛关注. 然而, 已有的自动化数据增强算法存在增强效果一般或者运行时长难以满足实际应用场景的问题. 针对以上问题, 本文研究提出了一种高效的基于自引导进化策

略的自动化数据增强算法 SGES AA. 首先, 研究设计了一种有效的数据增强策略连续化向量表示方法, 将自动化数据增强问题转换为连续化策略向量的搜索问题. 其次, 研究提出了一种基于自引导进化策略的策略向量搜索方法, 通过引入历史估计梯度信息指导探索点的更新. 大量的实验结果表明, SGES AA 在不显著增加搜索耗时的同时, 预测准确率优于或匹配目前最优的方法. 而且, SGES AA 能够有效支持图像分类、语音分类及文本分类任务的自动化数据增强.

未来工作中, 将尝试改进自引到进化策略, 使其能够更好地适应于数据增强策略的搜索中, 另外将进一步扩展自动化数据增强的应用场景, 探索针对图结构数据的自动化数据增强. 同时, 也将尝试将自动化数据增强与对比学习结合, 通过自动化数据增强提升对比学习性能.

References:

[1] Shorten C, Khoshgoftaar TM. A survey on image data augmentation for deep learning. Journal of Big Data, 2019, 6(1): 60. [doi: 10.1186/s40537-019-0197-0]

[2] He X, Zhao KY, Chu XW. AutoML: A survey of the state-of-the-art. Knowledge-based Systems, 2021, 212: 106622. [doi: 10.1016/j.knosys.2020.106622]

[3] Cubuk ED, Zoph B, Mané D, Vasudevan V, Le QV. AutoAugment: Learning augmentation strategies from data. In: Proc. of the 2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2019. 113−123. [doi: 10.1109/CVPR.2019.00020]

[4] Ho D, Liang E, Chen X, Stoica I, Abbeel P. Population based augmentation: Efficient learning of augmentation policy schedules. In: Proc. of the 36th Int'l Conf. on Machine Learning. Long Beach: PMLR, 2019. 2731–2741.

[5] Lim S, Kim I, Kim T, Kim C, Kim S. Fast AutoAugment. In: Proc. of the 2019 Annual Conf. on Neural Information Processing Systems. Vancouver, 2019. 6662–6672.

[6] Hataya R, Zdenek J, Yoshizoe K, Nakayama H. Faster AutoAugment: Learning augmentation strategies using backpropagation. In: Proc. of the 16th European Conf. on Computer Vision. Glasgow: Springer, 2020. 1–16. [doi: 10.1007/978-3-030-58595-2_1]

[7] Li YG, Hu GS, Wang YT, Hospedales T, Robertson NM, Yang YX. Differentiable automatic data augmentation. In: Proc. of the 16th European Conf. on Computer Vision. Glasgow: Springer, 2020. 580–595. [doi: 10.1007/978-3-030-58542-6_35]

[8] Cubuk ED, Zoph B, Shlens J, Le QV. Randaugment: Practical automated data augmentation with a reduced search space. In: Proc. of the 2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW). Seattle: IEEE, 2020. 3008–3017. [doi: 10.1109/cvprw50498.2020.00359]

[9] Liu FY, Li ZN, Qian C. Self-guided evolution strategies with historical estimated gradients. In: Proc. of the 29th Int'l Joint Conf. on Artificial Intelligence. Yokohama: IJCAI, 2020. 1474–1480. [doi: 10.24963/ijcai.2020/205]

[10] Moritz P, Nishihara R, Wang S, Tumanov A, Liaw R, Liang E, Elibol M, Yang ZH, Paul W, Jordan MI, Stoica I. Ray: A distributed framework for emerging AI applications. In: Proc. of the 13th USENIX Symp. on Operating Systems Design and Implementation. Carlsbad: USENIX Association, 2018. 561–577.

[11] Kaelbling LP, Littman ML, Moore AW. Reinforcement learning: A survey. Journal of Artificial Intelligence Research, 1996, 4: 237–285. [doi: 10.1613/jair.301]

[12] Krizhevsky A. Learning multiple layers of features from tiny images [MS. Thesis]. Toronto: University of Toronto, 2009.

[13] Jaderberg M, Dalibard V, Osindero S, Czarnecki WM, Donahue J, Razavi A, Vinyals O, Green T, Dunning I, Simonyan K, Fernando C, Kavukcuoglu K. Population based training of neural networks. arXiv:1711.09846, 2017.

[14] Liu HX, Simonyan K, Yang YM. DARTS: Differentiable architecture search. In: Proc. of the 7th Int'l Conf. on Learning Representations (ICLR). New Orleans: OpenReview.net, 2019.

[15] Jang E, Gu SX, Poole B. Categorical reparameterization with Gumbel-Softmax. In: Proc. of the 5th Int 'l Conf. on Learning Representations (ICLR). Toulon: OpenReview.net, 2017.

[16] Nesterov Y, Spokoiny V. Random gradient-free minimization of convex functions. Foundations of Computational Mathematics, 2017, 17(2): 527–566. [doi: 10.1007/s10208-015-9296-2]

[17] Wierstra D, Schaul T, Glasmachers T, Sun Y, Peters J, Schmidhuber J. Natural evolution strategies. The Journal of Machine Learning Research, 2014, 15(1): 949–980.

[18] Choromanski K, Rowland M, Sindhwani V, Turner RE, Weller A. Structured evolution with compact architectures for scalable policy optimization. In: Proc. of the 35th Int'l Conf. on Machine Learning. Stockholm: PMLR, 2018. 969–977.

[19] Inoue H. Data augmentation by pairing samples for images classification. arXiv:1801.02929, 2018.
[20] Netzer Y, Wang T, Coates A, Bissacco A, Wu B, Ng AY. Reading digits in natural images with unsupervised feature learning. In: Proc. of the 2011 NIPS Workshop on Deep Learning and Unsupervised Feature Learning. Granada: NIPS, 2011. 1–9.
[21] Piczak KJ. ESC: Dataset for environmental sound classification. In: Proc. of the 23rd ACM Int'l Conf. on Multimedia. Brisbane: ACM, 2015. 1015–1018. [doi: 10.1145/2733373.2806390]
[22] Tzanetakis G, Cook P. Musical genre classification of audio signals. IEEE Trans. on Speech and Audio Processing, 2002, 10(5): 293–302. [doi: 10.1109/tsa.2002.800560]
[23] Zhang X, Zhao JB, LeCun Y. Character-level convolutional networks for text classification. In: Proc. of the 28th Int'l Conf. on Neural Information Processing Systems. Montréal: MIT Press, 2015. 649–657.
[24] Auer S, Bizer C, Kobilarov G, Lehmann J, Cyganiak R, Ives Z. DBpedia: A nucleus for a web of open data. In: Proc. of the 6th Int'l Semantic Web Conf. and the 2nd Asian Semantic Web Conf. Busan: Springer, 2007. 722–735. [doi: 10.1007/978-3-540-76298-0_52]
[25] He KM, Zhang XY, Ren SQ, Sun J. Deep residual learning for image recognition. In: Proc. of the 2016 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR). Las Vegas: IEEE, 2016. 770–778. [doi: 10.1109/cvpr.2016.90]
[26] Zagoruyko S, Komodakis N. Wide residual networks. In: Proc. of the 2016 British Machine Vision Conf. (BMVC). York: BMVA Press, 2016. 87.1–87.12. [doi: 10.5244/c.30.87]
[27] Gastaldi X. Shake-shake regularization. arXiv:1705.07485, 2017.
[28] Szegedy C, Liu W, Jia YQ, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions. In: Proc. of the 2015 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). Boston: IEEE, 2015. 1–9. [doi: 10.1109/cvpr.2015.7298594]
[29] Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv:1412.6980, 2014.
[30] Huang G, Liu Z, van der Maaten L, Weinberger KQ. Densely connected convolutional networks. In: Proc. of the 2017 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). Honolulu: IEEE, 2017. 2261–2269. [doi: 10.1109/cvpr.2017.243]
[31] Joulin A, Grave É, Bojanowski P, Mikolov T. Bag of tricks for efficient text classification. In: Proc. of the 15th Conf. of the European Chapter of the Association for Computational Linguistics, Vol. 2 (Short Papers). Valencia: Association for Computational Linguistics, 2017. 427–431. [doi: 10.18653/v1/e17-2068]
[32] Chen YH. Convolutional neural network for sentence classification [MS. Thesis]. Waterloo: University of Waterloo, 2015.
[33] Zhou P, Shi W, Tian J, Qi ZY, Li BC, Hao HW, Xu B. Attention-based bidirectional long short-term memory networks for relation classification. In: Proc. of the 54th Annual Meeting of the Association for Computational Linguistics, Vol. 2 (Short Papers). Berlin: Association for Computational Linguistics, 2016. 207–212. [doi: 10.18653/v1/p16-2034]
[34] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. In: Proc. of the 31th Int'l Conf. on Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017. 6000–6010.
[35] Mania H, Guy A, Recht B. Simple random search provides a competitive approach to reinforcement learning. arXiv:1803.07055, 2018.
[36] Feurer M, Klein A, Eggensperger K, Springenberg JT, Blum M, Hutter F. Efficient and robust automated machine learning. In: Proc. of the 28th Int'l Conf. on Neural Information Processing Systems. Montréal: MIT Press, 2015. 2755–2763.
[37] Thornton C, Hutter F, Hoos HH, Leyton-Brown K. Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. In: Proc. of the 19th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD). Chicago: Association for Computing Machinery, 2013. 847–855. [doi: 10.1145/2487575.2487629]
[38] Qian C. Multiobjective evolutionary algorithms are still good: Maximizing monotone approximately submodular minus modular functions. Evolutionary Computation, 2021, 29(4): 463–490. [doi: 10.1162/evco_a_00288]

附录 A

表 A1–表 A3 分别为本文采用的图像、语音、文本数据增强策略.

表 A1　13 种图像数据增强函数以及增强幅度取值范围

增强函数	描述	增强幅度取值范围
ShearX (Y)	以某个幅度沿X (Y)轴剪切图像(0.5的概率取反)	[–0.3, 0.3]
TranslateX (Y)	以某个幅度在X (Y)轴方向上平移图像(0.5的概率取反)	[–150, 150]
Rotate	以某个幅度旋转图像(0.5的概率取反)	[–30, 30]

表 A1 13 种图像数据增强函数以及增强幅度取值范围 (续)

增强函数	描述	增强幅度取值范围
AutoContrast	通过将最暗的像素设置为黑色, 将最亮的像素设置为白色, 来最大化图像对比度	—
Invert	反转图像的像素	—
Equalize	均衡图像直方图	—
Solarize	反转所有超过某个幅度的像素	[0, 256]
Posterize	将每个像素的位数减少到某个幅度	[4, 8]
Contrast	控制图像的对比度, 幅度为0时输出灰度图像, 幅度为1时输出原始图像	[0.1, 1.9]
Color	调整图像的色彩平衡, 幅度为0时输出黑白图像, 幅度为1时输出原始图像	[0.1, 1.9]
Brightness	调整图像的亮度, 幅度为0时输出黑色图像, 幅度为1时输出原始图像	[0.1, 1.9]
Sharpness	调整图像的清晰度, 幅度为0时输出模糊的图像, 幅度为1时输出原始图像	[0.1, 1.9]
Cutout	将边长大小为某个幅度的随机正方形色块设置为灰色	[0, 60]

表 A2 6 种语音数据增强函数以及增强幅度取值范围

增强函数	描述	增强幅度取值范围
Gain	以某个概率将音频乘以某个随机幅度因子减小或者增加音量	[0, 1]
ImpulseResponse	以某个概率将音频与脉冲响应音频进行卷积	[0, 1]
PeakNormalization	以某个概率将音频的音量进行归一化	[0, 1]
PolarityInversion	以某个概率反转音频样本	[0, 1]
Shift	以某个概率先前或者向后移动音频	[0, 1]
ShuffleChannels	以某个概率调整音频声道	[0, 1]

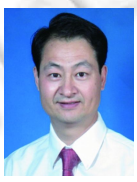
表 A3 6 种文本数据增强函数以及增强幅度取值范围

增强函数	描述	增强幅度取值范围
Spelling	以某个概率使用拼写错误的单词代替单词	[0, 1]
Synonym	以某个概率用WordNet的同义词代替单词	[0, 1]
Antonym	以某个概率用反义词代替单词	[0, 1]
RandomWordSwap	以某个概率随机交换单词	[0, 1]
RandomWordDelete	以某个概率随机删除单词	[0, 1]
RandomWordCrop	以某个概率随机删除一组连续的单词	[0, 1]

朱光辉(1987－), 男, 博士, 助理研究员, CCF 专业会员, 主要研究领域为自动化机器学习, 数据挖掘.

袁春风(1963－), 女, 博士, 教授, CCF 高级会员, 主要研究领域为大数据, 信息检索, 计算机体系结构.

陈文忠(1996－), 男, 硕士生, 主要研究领域为自动化机器学习.



黄宜华(1962－), 男, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为大数据, 分布式与并行计算, 机器学习.

朱振南(1999－), 男, 硕士生, 主要研究领域为数据挖掘, 图机器学习.