软件学报 ISSN 1000-9825, CODEN RUXUEW
2026,37(4):1472−1491 [doi: 10.13328/j.cnki.jos.007522] [CSTR: 32375.14.jos.007522]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于自适应策略优化的鲁棒精度权衡学习*

翟浩杰[1], 王 冉[2,5], 邬文慧[3,5], 贾育衡[4]

[1](深圳大学 数学科学学院, 广东 深圳 518060)
[2](深圳大学 人工智能学院, 广东 深圳 518060)
[3](深圳大学 电子与信息工程学院, 广东 深圳 518060)
[4](东南大学 计算机科学与工程学院, 江苏 南京 211189)
[5](广东省智能信息处理重点实验室 (深圳大学), 广东 深圳 518060)
通信作者: 王冉, E-mail: wangran@szu.edu.cn

摘 要: 对抗训练被视为提升深度模型鲁棒性的核心防御手段, 但其固有缺陷严重制约了实际应用效果. 传统对抗训练方法依赖固定攻击模式生成对抗样本, 导致训练过程中样本多样性不足、模型泛化能力受限, 且在鲁棒性与干净准确率间难以达成有效平衡. 更为关键的是, 现有对抗训练框架缺乏对训练过程的自适应控制, 容易引发鲁棒过拟合现象. 针对上述挑战, 利用演化优化提出一个自适应对抗训练框架, 称为基于自适应策略优化的鲁棒精度权衡学习, 简称 TRA^{2}SO. 该方法将遗传算法引入对抗训练过程, 通过动态调整不同训练阶段的对抗攻击策略, 实现对抗样本生成模式的渐进式复杂化. 这种层级递进的对抗机制不仅增强了样本多样性, 还可通过策略优化记录实现训练早停, 有效抑制过拟合风险. 在 CIFAR 系列数据集上的实验表明, 相较于传统对抗训练方法, 所提框架在维持基础分类性能的同时, 提升了模型面对多种攻击范式的防御能力, 且加快了训练收敛速度. 为对抗训练中鲁棒性-准确性的权衡提供了新思路, 对构建可信深度学习系统具有重要实践价值.

关键词: 鲁棒性; 自适应策略优化; 权衡学习; 对抗训练; 演化优化
中图法分类号: TP18

中文引用格式: 翟浩杰, 王冉, 邬文慧, 贾育衡. 基于自适应策略优化的鲁棒精度权衡学习. 软件学报, 2026, 37(4): 1472–1491. http://www.jos.org.cn/1000-9825/7522.htm
英文引用格式: Zhai HJ, Wang R, Wu WH, Jia YH. Trade-off Learning for Robustness and Accuracy Based on Adaptive Strategy Optimization. Ruan Jian Xue Bao/Journal of Software, 2026, 37(4): 1472–1491 (in Chinese). http://www.jos.org.cn/1000-9825/7522.htm

Trade-off Learning for Robustness and Accuracy Based on Adaptive Strategy Optimization

ZHAI Hao-Jie[1], WANG Ran[2,5], WU Wen-Hui[3,5], JIA Yu-Heng[4]

[1](School of Mathematical Sciences, Shenzhen University, Shenzhen 518060, China)
[2](School of Artificial Intelligence, Shenzhen University, Shenzhen 518060, China)
[3](College of Electronic and Information Engineering, Shenzhen University, Shenzhen 518060, China)
[4](School of Computer Science and Engineering, Southeast University, Nanjing 211189, China)
[5](Guangdong Key Laboratory of Intelligent Information Processing (Shenzhen University), Shenzhen 518060, China)

Abstract: Adversarial training is regarded as a core defense mechanism for enhancing the robustness of deep models, yet its inherent limitations significantly constrain its effectiveness in practical applications. Traditional adversarial training methods rely on fixed attack

* 基金项目: 国家自然科学基金 (62176160, U24A20322, 62376162); 广东省基础与应用基础研究基金自然科学基金 (2024B1515020109); 广东省智能信息处理重点实验室 (2023B1212060076)
本文由“高可信自适应机器学习”专题特约编辑王熙照教授、张敏灵教授、曹付元教授推荐.
收稿时间: 2025-04-15; 修改时间: 2025-06-30, 2025-08-15; 采用时间: 2025-08-20; jos 在线出版时间: 2025-09-02
CNKI 网络首发时间: 2025-11-27

patterns to generate adversarial examples (AEs), leading to insufficient sample diversity, limited generalization capabilities, and difficulties in achieving an effective balance between robustness and clean accuracy. More crucially, existing adversarial training frameworks lack adaptive control over the training process, resulting in the robust overfitting phenomenon. To address these challenges, an evolutionary optimization-based adaptive adversarial training framework is proposed, named trade-off robustness and accuracy via adaptive strategy optimization (TRA^2SO). It innovatively integrates a genetic algorithm into adversarial training and achieves progressive complexity escalation in AE generation through dynamic adjustment of attack strategies across different training phases. This mechanism not only enhances sample diversity but also effectively suppresses overfitting risks through early stopping enabled by strategy optimization records. Experiments on CIFAR series datasets demonstrate that, compared with traditional adversarial training methods, the proposed TRA^2SO framework maintains baseline classification performance while improving robustness against multiple attack paradigms and accelerating training convergence. This study provides new insights into the robustness-accuracy trade-off in adversarial training, offering significant practical value for building trustworthy deep learning systems.

Key words: robustness; adaptive strategy optimization; trade-off learning; adversarial training; evolutionary optimization

近年来, 深度学习技术在计算机视觉[1]、自然语言处理[2,3]、生成式人工智能[4]等领域取得了突破性进展, 并在众多任务中展现出远超越人类水平的性能. 然而, 深度学习的实际部署面临一个根本性挑战: 模型面对对抗攻击的脆弱性.

研究表明[5], 在输入空间对样本添加人类视觉系统难以察觉的细微扰动, 会使深度神经网络以高置信度产生误分类, 这种扰动后的样本被称为对抗样本, 不仅阻碍了深度学习的实际部署与规模化应用, 更对系统可信性构成了严峻挑战. 在安全关键型应用场景中, 此类漏洞可能引发严重后果. 例如, 在人脸识别系统中, 攻击者可通过生成视觉不可区分的对抗样本冒充授权用户以非法获取敏感信息或实施财产欺诈; 在自动驾驶场景中, 针对车载感知模块的对抗攻击可能引发交通标志误识别或障碍物漏检, 进而造成严重交通事故. 因此, 提升深度模型的对抗鲁棒性具有重要意义, 已成为可信人工智能领域的核心研究课题之一. 其目标在于实现双重优化: 一方面维持模型在原始数据分布上的泛化性能, 即对干净样本保持高分类准确率; 另一方面增强模型对对抗扰动的免疫能力, 即在对抗攻击下仍能保持稳定决策.

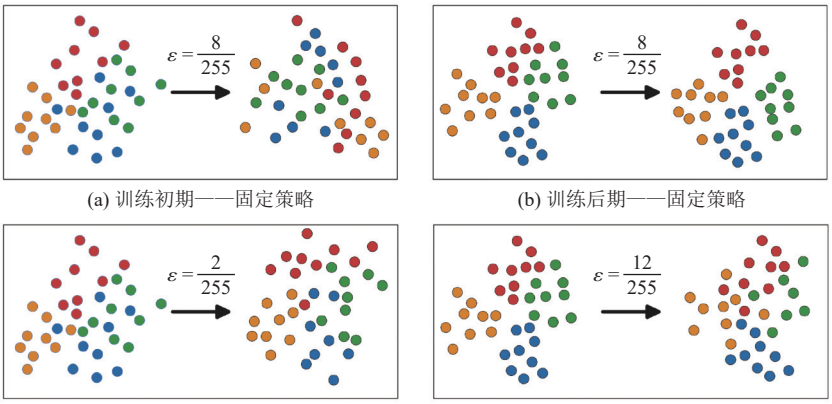
对抗训练[6−8]是提升深度模型对抗鲁棒性最行之有效的方式, 其核心思想是利用各种对抗攻击产生对抗样本, 将对抗样本添加到原始数据集中, 一起或者单独使用进行模型训练. Madry 等人[6]将对抗训练形式化地表述为一个双层优化问题, 准确描述为以下极小极大化公式:

$$\min_{\mathbf{w}} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\max_{\delta\in\Omega} \mathcal{L}(f(\mathbf{x}+\boldsymbol{\delta},\mathbf{w}),y)] \tag{1}$$

其中, $\mathcal{L}$ 是损失函数, $\boldsymbol{\delta}$ 是添加在样本 $\mathbf{x}$ 上的扰动, y 是样本 $\mathbf{x}$ 的真实标记, Ω 是扰动范围, $\mathbf{w}$ 是神经网络的参数. 数学上, 这是一个关于 $\mathbf{w}$ 和 $\boldsymbol{\delta}$ 的鞍点问题, 并且 Madry 等人[6]证明了可通过内外层问题的交替优化寻找到鞍点. 然而, 实际应用表明, 对抗训练在提升模型鲁棒性的同时会显著降低其在干净样本上的泛化性能, 当模型训练到一定程度时, 鲁棒性与干净精度似乎会出现不可避免的冲突, 此问题称 accuracy-robustness problem. 为缓解此问题, 对抗训练涌现出多种改进策略以平衡模型在干净样本和对抗样本上的性能, 这些研究主要围绕自适应扰动调整和训练效率优化展开. 关于自适应扰动调整, Wu 等人[9]提出了一种无需搜索的自适应扰动半径框架, 通过理论分析证明了其对自然泛化性能和鲁棒性的双重提升; 类似地, Yang 等人[10]引入数据自适应扰动大小 (DAAT), 利用校准网络动态调整扰动范围, 减少对自然准确率的负面影响. 在训练效率与过拟合控制方面, Yu 等人[11]提出了基于损失平稳条件 (LSC) 的权重扰动策略, 避免对抗训练中的鲁棒过拟合; Wang 等人[12]进一步提出可持续自进化对抗训练 (SSEAT), 通过对抗数据回放和一致性正则化缓解持续学习中的灾难性遗忘等.

尽管上述方法在特定任务中表现优异, 但仍存在明显的局限性. 例如, 自适应扰动策略的计算开销较高, 且部分方法需要设计额外的群组划分规则, 增加了实现复杂度; 相应的优化策略是针对特定数据集而制定的, 泛化能力存疑等. 与此同时, 多数方法在应对鲁棒过拟合问题上效果仍旧不理想, 即: 在对抗训练中后期, 模型在干净样本上的准确率继续提升或保持稳定, 但在对抗样本上的准确率却开始下降. 此外, 对抗样本的生成质量严重依赖攻击策略的选择, 然而, 现有对抗训练方法大多采用固定的攻击策略, 限制了训练的灵活性和自适应调节能力. 图 1(a) 和图 1(b) 揭示了传统对抗训练采用固定攻击策略在训练初期和后期所面临的局限性. 具体而言, 在训练初期, 由于

样本分类边界尚未清晰, 若采取固定的攻击策略可能导致攻击强度过强, 从而进一步加剧样本分布的模糊性, 显著增加学习的难度. 而在训练后期, 当分类边界已趋于清晰时, 固定的攻击策略可能导致攻击强度过弱, 无法生成更具挑战性的对抗样本, 进而限制了模型鲁棒性的进一步提升.



(a) 训练初期——固定策略 (b) 训练后期——固定策略

(c) 训练初期——自适应策略 (d) 训练后期——自适应策略

图 1 不同训练阶段适合的攻击策略不同

受上述现象启发, 在本文中, 我们提出了一种自适应对抗训练方式, 称为基于自适应策略优化的鲁棒精度权衡学习, 简称 TRA^{2}SO (trade-off robustness and accuracy via adaptive strategy optimization). 图 1(c) 和图 1(d) 展示了 TRA^{2}SO 方法相较于传统对抗训练方法在克服上述局限性方面的优势. 通过阶段性评估模型状态, TRA^{2}SO 可以把握模型在不同训练阶段鲁棒性与泛化性的提升潜力, 利用遗传算法自适应地调整攻击策略. 具体而言, 在训练初期, TRA^{2}SO 倾向于选择较弱的攻击策略, 促进模型对数据分布的初步理解, 以减轻学习的难度. 而在训练后期, 随着模型能力的逐步增强, TRA^{2}SO 倾向于选择较强的攻击策略, 探索更具挑战性的样本分布, 实现鲁棒性的进一步提升. 本文的贡献可总结如下.

1) 系统性探索了演化优化在对抗防御中的应用范式, 将演化计算嵌入对抗训练框架, 通过设计种群驱动的策略搜寻机制, 突破传统方法使用静态扰动限制生成对抗样本的局限性, 构建了动态自适应的对抗训练框架 TRA^{2}SO.

2) TRA^{2}SO 方法可利用遗传算法自适应地调整对抗训练期间的内层优化策略, 增强了对抗样本的多样性, 确保模型在训练过程中, 生成的对抗样本对于鲁棒性的提升持续有效; 同时, 通过记录和观察适应度函数值的变化引导模型实现训练早停, 防止鲁棒过拟合的发生, 从而提升模型的可解释性和可信性.

3) 在 CIFAR-10 和 CIFAR-100 数据集上做了大量实验, 使用 L_2 和 L_∞ 范数限制扰动的评估方式, 验证了 TRA^{2}SO 方法可有效提高模型鲁棒性, 实现鲁棒性与准确性的权衡, 同时和标准对抗训练相比, 可以使模型更快收敛且避免鲁棒过拟合.

本文第 1 节系统性阐述对抗样本的概念, 综述现有对抗攻击与对抗防御方法的技术脉络, 重点剖析对抗训练的技术原理及研究进展, 并探讨演化优化在本领域的创新应用. 第 2 节介绍 TRA^{2}SO 方法涉及的基础概念与核心理论框架, 为后续算法设计提供支撑. 第 3 节详细介绍 TRA^{2}SO 方法的技术细节, 包括多阶段优化的数学建模、方法实现及算法伪代码. 第 4 节基于公开基准数据集开展系统性对比实验, 通过多维度指标的综合评价与模型收敛趋势的可视化, 验证 TRA^{2}SO 在鲁棒性提升与泛化性平衡方面的有效性及技术优势. 第 5 节总结本文的贡献及其在鲁棒精度权衡学习方面的研究潜力.

1 相关工作

Szegedy 等人[5]发现, 在原始空间中对样本添加某些精心设计的、人类肉眼不可察觉的扰动后, 神经网络会以极高的置信度对原本正确识别的样本产生错误分类, 这种样本称之为对抗样本, 使得神经网络的可信性面临极大

挑战. 通常, 这种扰动的不可察觉性可转化为对扰动大小 L_p 范数的 ε 上界限制. 令 $\mathbf{x}$ 表示原样本, y 表示其真实标签, $\boldsymbol{\delta}$ 表示添加到 $\mathbf{x}$ 上的扰动, $f(\cdot,\mathbf{w})$ 表示神经网络模型, $\mathbf{w}$ 表示网络参数, $\mathcal{L}$ 为损失函数, 于是针对样本 $\mathbf{x}$ 的对抗样本集合可以表示为:

$$A(\mathbf{x}) = \{\mathbf{x}+\boldsymbol{\delta} | f(\mathbf{x}+\boldsymbol{\delta},\mathbf{w}) \neq f(\mathbf{x},\mathbf{w}), \|\boldsymbol{\delta}\|_p \leqslant \varepsilon\} \tag{2}$$

1.1 对抗攻击

一般情况下, 获取对抗样本的任何方法均可称为对抗攻击. 结合上述对抗样本和扰动限制的概念, 对抗攻击从数学上可以被建模为一个优化问题:

$$\max_{\boldsymbol{\delta}} \mathcal{L}(f(\mathbf{x}+\boldsymbol{\delta},\mathbf{w}),y),\ \text{s.t.}\ \|\boldsymbol{\delta}\|_p \leqslant \varepsilon \tag{3}$$

即在扰动 $\boldsymbol{\delta}$ 的 L_p 范数的 ε 上界约束下, 最大化网络输出对于真实标签的损失, 记 $\mathbf{x}_{\text{adv}} = \mathbf{x}+\boldsymbol{\delta}$ 为对抗样本. 针对特定分类任务, 扰动 $\boldsymbol{\delta}$ 的 L_p 范数约束会根据 p 的不同取值具有不同的特殊意义, 例如在图像分类中, 样本 $\mathbf{x}$ 的单个通道通常被建模为数学概念上的矩阵, 常使用 L_0、L_2 和 L_∞ 这 3 种特殊的矩阵元范数对其进行约束, 其中 L_0 范数限制图像中可修改的像素数量, L_2 范数限制图像中像素值的平均变化, L_∞ 范数限制图像中所有像素值的最大变化.

经典对抗攻击方法依据其核心策略可分为 3 大类: 基于梯度信息的攻击、基于函数优化的攻击以及自适应集成攻击, 各类方法在扰动生成机制和应用场景上具有显著差异.

1) 基于梯度信息的攻击. 此类方法利用模型对样本的梯度信息直接构造对抗扰动. 以 L_∞ 范数约束下的攻击为例, Goodfellow 等人[13]提出的快速梯度符号法 FGSM (fast gradient sign method) 是奠基性工作, 通过单步计算损失函数对输入样本的梯度符号生成扰动, 具有高效性但攻击精度有限. 后续研究通过迭代优化改进攻击效果, 如基本迭代法 BIM (basic iterative method)[14]将 FGSM 扩展为多步小步长更新; 动量迭代法 MIM (momentum iterative method)[15]引入动量项以加速收敛并规避局部最优; Madry 等人[6]提出的投影梯度下降法 PGD (projected gradient descent) 进一步结合随机初始化和迭代投影机制, 在对抗训练中被广泛用作强基准攻击. 此类方法依赖一阶梯度信息, 计算效率高且在白盒设定下表现优异.

2) 基于函数优化的攻击. 此类方法将对抗样本生成问题建模为带约束的最优化问题, 通过数值优化算法求解最优扰动. 典型代表是 Carlini 等人[16]提出的 C&W 攻击, 通过设计替代损失函数 (如 DLR 损失) 并采用 Adam 等优化器, 在 L_2 范数约束下最小化扰动幅度的同时确保模型对扰动样本分类错误. C&W 攻击能够生成视觉不可察觉的对抗样本, 且对部分防御策略具有强穿透性. 其优化目标可灵活适配 L_0、L_2 和 L_∞ 等多种范数约束, 但因需要多次前向传播和梯度计算, 时间成本较高.

3) 自适应集成攻击. 为全面评估模型鲁棒性, 集成多种攻击策略的自适应方法逐渐成为主流. Croce 等人[17]提出的自动攻击 AA (autoattack) 是此类方法的典范, 整合了 3 种白盒攻击技术与一种黑盒攻击技术. 其中, APGD-CE 和 APGD-DLR 分别基于交叉熵损失和 DLR 损失实现自适应步长调整的白盒 PGD 攻击; 快速适应性边界攻击 FAB (fast adaptive boundary attack)[18]通过逼近决策边界生成跨范数约束的对抗样本; square attack[19]则以黑盒方式利用随机局部搜索探索模型脆弱区域. AA 攻击通过自动调度不同攻击策略, 有效规避单一攻击的盲区, 成为当前鲁棒性评估的标准工具集.

对抗攻击方法从初期的单步梯度优化发展到多策略协同优化, 逐步形成了层次化的技术体系. 基于梯度的方法聚焦效率与白盒场景的强攻击性, 基于优化的方法追求扰动最小化与防御穿透性, 而自适应集成攻击则通过策略融合实现全面的鲁棒性评估.

1.2 对抗防御

主流的对抗防御方法主要分为 4 类: 对抗训练、输入预处理、模型结构优化和对抗样本检测. 其中, 对抗训练通过将对抗样本注入训练过程, 迫使模型在扰动下保持正确预测, 被广泛证明是提升模型鲁棒性最有效的方法, 其核心在于通过极小极大优化实现风险上界的最小化. 相比之下, 输入预处理 (如去噪、随机化) 易被自适应攻击绕过, 模型结构优化 (如防御层、梯度掩码) 存在泛化局限, 而对抗样本检测则面临误判率和攻击演进的挑战. 本文

重点探讨对抗训练的理论机制与优化策略, 分析其在不同攻击场景下的有效性边界.

1.2.1 对抗训练

对抗训练作为提升深度模型鲁棒性的关键性防御手段, 其核心机理在于通过构造对抗样本并将其纳入训练过程, 使模型在面对分布偏移扰动或恶意攻击时保持稳定. 该方法的理论框架可追溯至 Goodfellow 等人[13]提出的 FGSM 对抗样本生成策略, 通过将对抗样本引入训练集, 首次验证了该策略对模型鲁棒性的提升效果. Madry 等人[6]进一步建立了标准对抗训练 PGD-AT 的理论体系, 提出将 PGD 攻击作为对抗样本生成基准, 并形式化定义了公式 (1) 中的双层优化问题. 该鞍点问题的求解范式成为后续对抗训练研究的理论基础, 特别是面对基于一阶梯度信息的攻击方法具有普适性防御效果.

通过对损失函数进行改进, 标准对抗训练进一步衍生出若干更有效的对抗训练方法. 例如, Zhang 等人[7]通过推导对抗样本预测误差上界, 提出 TRADES (tradeoff-inspired adversarial defense via surrogate-loss minimization) 方法. 该方法引入 KL 散度作为损失函数正则项, 建立如下优化问题:

$$\min_{\mathbf{w}} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\mathcal{L}_{\mathrm{CE}}(f(\mathbf{x},\mathbf{w}),y)+\beta\cdot KL(p(\mathbf{x},\mathbf{w})\,\|\,p(\mathbf{x}_{\mathrm{adv}},\mathbf{w}))] \tag{4}$$

其中, $\mathcal{L}_{\mathrm{CE}}$ 表示交叉熵损失, $p(\mathbf{x},\mathbf{w})$ 和 $p(\mathbf{x}_{\mathrm{adv}},\mathbf{w})$ 分别表示干净样本 $\mathbf{x}$ 和对抗样本 $\mathbf{x}_{\mathrm{adv}}$ 在网络 $f(\cdot,\mathbf{w})$ 中的输出概率分布, β 为平衡系数. 该框架在保持干净样本精度的同时, 显式约束对抗样本与原始样本输出分布的差异, 实现鲁棒性与准确性的平衡优化. Wang 等人[8]提出的 MART (misclassification aware adversarial training) 方法在 TRADES 基础上引入误分类感知机制, 通过动态调整正则项权重, 重点增强模型对困难样本 (即原本分类错误的样本) 的防御能力. 其优化问题建立为:

$$\min_{\mathbf{w}} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\mathcal{L}_{\mathrm{BCE}}(f(\mathbf{x}_{\mathrm{adv}},\mathbf{w}),y)+\lambda\cdot KL(p(\mathbf{x},\mathbf{w})\,\|\,p(\mathbf{x}_{\mathrm{adv}},\mathbf{w}))\cdot(1-p_y(\mathbf{x}))] \tag{5}$$

其中, $\mathcal{L}_{\mathrm{BCE}}$ 表示二元交叉熵损失, λ 为调节因子, $p_y(\mathbf{x})$ 表示样本 $\mathbf{x}$ 在类别 y 上的预测概率. 该方法通过 $(1-p_y(\mathbf{x}))$ 项实现样本的自适应加权, 强化对容易误分类样本的鲁棒性约束.

此外, Wu 等人[20]通过对神经网络的权重进行扰动, 提出了一种通用的对抗训练增强方法 AWP.

1.2.2 自适应对抗训练

传统对抗训练通过固定扰动预算提升模型鲁棒性, 但难以平衡自然准确率与抗攻击能力. 近年来, 许多研究探索了自适应对抗训练 AAT (adaptive adversarial training) 方法, 通过灵活调整对抗扰动进一步提高模型的鲁棒性和泛化能力. 核心创新主要从以下几个方面展开.

1) 扰动半径动态调整. 针对固定扰动预算导致的过拟合或欠拟合问题, 学者们提出了数据驱动的动态调整策略. Qian 等人[21]提出分组自适应对抗训练 (GAAT), 根据样本 logit 最大值划分数据组, 通过二分搜索为每组分配最优扰动半径, 实现了细粒度鲁棒性和准确率的权衡. Wang 等人[22]提出特征敏感调整 (WAPAT), 引入特征变化权重动态调节扰动步长, 缩短对抗样本与分类边界的距离. Yu 等人[23]提出强度渐进策略 (SAAT), 设计对抗损失约束机制, 使扰动预算随训练进程动态增长, 逐步提升模型鲁棒性. Yang 等人[10]研究了数据自适应对抗训练 (DAAT), 通过校准网络动态调整每个样本的扰动大小, 以降低对抗训练对干净准确率的影响.

2) 多扰动鲁棒性增强. 为应对多样化攻击类型 (如 L_0、L_2、L_∞), Xiao 等人[24]提出平滑加权方法 (ASW-AT), 通过稳定性分析优化风险边界, 在多种范数约束下平衡扰动效果. Wang 等人[25]提出鲁棒模式连接 (RMC), 通过构建多扰动联合优化框架, 结合自免疫机制 (SRMC) 降低计算复杂度, 显著提升了模型面对跨范数攻击的防御能力.

3) 训练效率优化. 针对对抗训练的高计算成本, Huang 等人[26]提出自适应步长 (ATAS), 通过实例级步长调整加速单步对抗训练, 有效缓解灾难性过拟合. Ma 等人[27]提出参数自由的自适应边界防御 (SMD), 通过梯度感知的边界扩展策略, 无需超参数调优即可实现高效鲁棒训练.

4) 特定领域的权衡方案. Guo 等人[28]结合历史梯度攻击 (HGAA) 与域适应训练 (DAAT), 在工业软传感器场景中针对鲁棒精度与自然精度实现协调. 此外, Hardy 等人[29]通过理论分析证明了自适应训练不会显著增加系统补救成本, 为其在关键领域的部署提供了依据.

然而, 值得注意的是, 上述方法无法解决鲁棒性与自然准确率的权衡困境, 其核心难点在于, 当模型训练到一

定程度时, 鲁棒性与自然准确率似乎会出现不可避免的冲突现象, 难以继续同时优化. 尽管 GAAT[21]和 DAAT[10]通过分组或校准试图平衡鲁棒性与自然准确率, 但这种权衡高度依赖启发式设计, 难以在复杂任务 (如高分辨率图像分类) 中取得最优解. 其次, 动态扰动预算存在敏感性问题, 如 SAAT[23]和 ATAS[26]等方法通过动态调整扰动强度以缓解过拟合, 但可能因扰动预算的初始设置或调整策略不当导致训练不稳定.

1.3 基于演化优化的对抗攻防

基于演化优化的对抗攻防研究当前呈现显著的非对称性. 在攻击侧, 演化计算因其无梯度信息依赖特性, 已成为黑盒攻击的核心方法之一. 该类方法通过模拟生物进化机制 (如遗传算法中的选择-交叉-变异操作、差分进化中的种群扰动策略), 在模型结构及梯度信息未知的情况下, 高效搜索满足 L_p 范数约束的对抗样本. 代表性工作包括 Su 等人[30]提出的基于差分进化的单像素攻击, 通过 L_0 稀疏扰动验证了演化算法在高维空间搜索的有效性; 白祉旭等人[31]提出的关键区域约束进化策略, 通过类激活热力图定位决策敏感区域, 将扰动搜索空间压缩了 63% 以上; 以及 Williams 等人[32]构建的多目标优化框架, 利用 Pareto 前沿动态平衡扰动稀疏性与隐蔽性, 显著提升了对抗样本的视觉自然性和迁移攻击成功率.

然而, 在防御侧, 利用演化优化提升模型鲁棒性的工作还十分稀少. 现有的鲁棒性增强方法, 特别是对抗训练方法, 大多依赖固定的对抗样本生成策略, 未能有效利用演化计算方法在全局搜索、策略优化、自适应调整等方面的优势. 以遗传算法为例, 将其应用于对抗训练主要存在以下两个难点.

1) 计算成本过高. 对抗训练通常涉及复杂的神经网络结构和高维空间运算, 而遗传算法需要通过大量迭代和种群评估来搜索最优解, 在高维大规模问题上的计算成本很高. 此外, 对抗训练需持续生成对抗样本用于模型更新, 每次种群评估都需要调用目标模型进行前向传播, 进一步增加了时间开销.

2) 局部最优问题. 遗传算法虽然具有较强的全局搜索能力, 但在某些情况下仍可能陷入局部最优, 尤其是在面对非凸、非连续的目标函数时. 同时, 算法初始种群的选择至关重要, 若初始种群分布不均或多样性不足, 可能导致搜索过程难以有效探索参数空间, 从而影响对抗样本的生成质量和攻击效果.

这种攻防方法论的非对称性导致防御策略难以适应复杂多变的攻击模式, 亟待建立基于演化优化的防御方法.

2 基础知识

本节就 TRA^{2}SO 方法涉及的基础知识予以介绍, 包括演化计算中的若干概念, 以及在此概念框架下对攻击策略的形式化描述.

2.1 演化计算

演化计算方法是一类利用自然界进化机制 (如自然选择、变异、交叉等) 来解决复杂问题的全局优化算法. 它模拟了生物种群的进化过程, 借助适应度评估、选择、变异和交叉等操作逐步改善解的质量, 通过迭代优化来找到问题的全局最优解或全局近似最优解. 常见的演化计算方法包括遗传算法 GA (genetic algorithm)、遗传规划 GP (genetic programming)、演化策略 ES (evolutionary strategy)、差分进化 DE (differential evolution) 等. 其中, 遗传算法是演化计算中最为经典和应用最为广泛的算法之一, 它模拟了达尔文的自然选择理论, 通过模拟基因的交叉、变异和选择等生物进化过程来优化问题的解. 这一类方法在求解问题时对于目标函数几乎没有要求, 不需要其梯度信息, 对于问题的约束也没有严格限制, 同时对先验知识的依赖性低, 且由于种群的性质天然支持并行计算, 使得其对于问题的适应性很强, 被广泛用于各类复杂问题的求解. 总的来讲, 遗传算法涉及以下核心概念.

1) 种群 (population): 由 n 个候选解 $\mathbf{a}$ 构成的集合 $\mathcal{P} \subseteq \mathcal{S}$, 其中 $\mathcal{S}$ 为解空间, 每个个体 $\mathbf{a}$ 称为染色体, 常用二进制串或实数向量编码进行表示.

2) 适应度函数 (fitness function): 映射 $\mathcal{F}: \mathcal{S} \to \mathbb{R}^+$, 用于量化个体 $\mathbf{a} \in \mathcal{P}$ 的质量, 值越大表示越优.

3) 选择操作 (selection): 基于适应度值的概率筛选机制, 常用方法包括轮盘赌选择、锦标赛选择、排序选择等.

4) 交叉操作 (crossover): 模拟基因重组, 将两个父代个体的部分基因交换以生成子代, 典型策略包括单点交叉、

均匀交叉等.

5) 变异操作 (mutation): 以低概率随机修改个体基因 (如翻转二进制位或扰动实数值), 防止种群陷入局部最优.

6) 替换策略 (replacement): 确定新种群的构成方式, 分为代际替换 (完全由子代取代父代) 和稳态替换 (保留部分父代个体).

7) 终止条件 (termination): 算法停止的判定标准, 如达到最大迭代次数 T 或当前最优解 $\mathbf{a}^*$ 满足阈值判断 $\mathcal{F}(\mathbf{a}^*) \geqslant \theta$.

基于上述相关概念, 算法 1 给出了遗传算法的总体伪代码.

算法 1. 遗传算法.

输入: 进化代数 T, 种群大小 n, 适应度函数 $\mathcal{F}$, 交叉概率 p_c, 变异概率 p_m;
输出: 最优个体 $\mathbf{a}^*$.

1. 随机初始化第 0 代种群 $\mathcal{P}_0 = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\}$;
2. 初始化迭代次数 $t = 0$;
3. While $t \leqslant T$ do
4. $\quad \mathcal{P} = \mathcal{P}_t$;
5. $\quad$ For each $\mathbf{a} \in \mathcal{P}$ do
6. $\quad\quad$ 计算 $\mathcal{F}(\mathbf{a})$;
7. $\quad$ End For
8. $\quad$ 通过选择操作根据适应度值选取成对的父代个体;
9. $\quad$ 对每对父代个体以概率 p_c 执行交叉操作, 生成子代集合 $\mathcal{C}$;
10. $\quad$ 对每个子代 $\mathbf{c} \in \mathcal{C}$ 以概率 p_m 执行变异操作;
11. $\quad t = t + 1$;
12. $\quad \mathcal{P}_t \leftarrow$ 根据替换策略, 合并父代和子代生成新种群, 并根据适应度大小排序选出前 n 个策略;
13. End While;
14. $\mathbf{a}^* = \underset{\mathbf{a} \in \mathcal{P}_t}{\operatorname{argmax}} \mathcal{F}(\mathbf{a})$.

2.2 攻击策略和攻击策略空间

攻击策略: 攻击策略定义为对干净样本执行对抗攻击的方法和此方法下的一组具体参数, 不同的攻击方法其参数个数和参数取值范围各不相同. 对于含有 m 个参数的对抗攻击方法, 记 $\mathbf{a} = (\mathrm{a}^1, \mathrm{a}^2, \ldots, \mathrm{a}^m)$ 构成的 m 元组为该攻击方法对应的攻击策略. 特别地, 以经典的 PGD 攻击为例, 其在数学上的表述如下:

$$\mathbf{x}_{t+1} = \prod_{\mathbf{x}+\Omega} \mathbf{x}_t + \alpha \cdot \operatorname{sign}(\nabla_{\mathbf{x}_t} \mathcal{L}(f(\mathbf{x}_t + \delta, \mathbf{w}), y)), \quad t = 0, \ldots, I-1 \tag{6}$$

其中, $\mathbf{x}_{t+1}$ 是第 t 步添加扰动截断后的样本, I 为最大步数, α 为步长, $\Omega = \{\delta \mid \|\delta\|_p \leqslant \varepsilon\}$ 为扰动范围, $\prod$ 表示投影操作, 在每一步迭代时对超出范围的扰动值进行截断使其不超过 ε, 当达到最大步数时, 得到最终的对抗样本 $\mathbf{x}_{\mathrm{adv}} = \mathbf{x}_I$. 从公式 (6) 中可以看出, PGD 攻击有 3 个参数, 扰动强度 ε, 扰动步长 α, 扰动步数 I, 由这 3 个参数构成的三元组 (ε, α, I) 称为一个攻击策略, 不同的参数值组合代表不同的攻击策略, 利用不同的攻击策略对干净样本进行扰动会产生不同的对抗样本.

攻击策略空间: 策略空间指在确定一种攻击方法后, 所有可能的攻击策略的集合, 用 $\mathcal{A} = \{\mathbf{a}_1, \mathbf{a}_2, \ldots\}$ 表示. 一般地, 对于含有 m 个参数的对抗攻击方法, 每个参数的可用选择构成一个集合 S_{a^m}, 可以记 $\mathcal{A} = S_{\mathrm{a}^1} \times S_{\mathrm{a}^2} \times \ldots \times S_{\mathrm{a}^m}$, 是这 m 个集合的笛卡尔积. 以 PGD 攻击为例, 假设扰动强度 ε 的区间为 $S_\varepsilon = [2/255, 16/255]$, 扰动步长 α 的区间为 $S_\alpha = [1/255, 4/255]$, 扰动步数 I 的区间为 $S_I = [4, 20]$ (在实际运行时步数作取整操作), 那么 $\mathcal{A} = S_\varepsilon \times S_\alpha \times S_I$ 就构成了 PGD 的攻击策略空间.

3 基于自适应策略优化的鲁棒精度权衡学习

演化计算在对抗攻击领域、特别是黑盒攻击领域应用广泛, 但鲜有工作将其应用于对抗防御领域、特别是对抗训练的优化中. 受益于遗传算法对于无约束优化问题的解空间的强大启发式搜索能力, 在本节中, 我们将使用遗传算法来寻找公式 (1) 中内层优化的最优攻击策略, 并在此基础上提出基于自适应策略优化的鲁棒精度权衡学习框架 TRA^{2}SO.

3.1 标准对抗训练

标准对抗训练 PGD-AT 利用最小-最大优化框架提升模型鲁棒性, 其核心思想是: 在训练过程中, 主动生成对当前模型最具威胁性的对抗样本, 并基于这些样本更新模型参数以增强其抗干扰能力. 具体实现时, 给定数据集 $\mathcal{D}=\{(\mathbf{x}_i,y_i)\}_{i=1}^N$, 损失函数使用交叉熵损失 $\mathcal{L}_{\rm CE}$, 扰动约束采用 L_∞ 范数, 则将公式 (1) 中的损失函数替换为代理损失后的优化问题可表示为:

$$\min_{\mathbf{w}}\frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_{\rm CE}(f(\mathbf{x}_{i,\rm adv}^{\mathbf{a}^+},\mathbf{w}),y_i) \tag{7}$$

该最小化问题可采用内外层交替优化来求解: 首先, 使用固定的攻击策略 $\mathbf{a}^+=(8/255,2/255,10)$ 通过梯度上升法进行内层最大化, 对干净样本 $\mathbf{x}_i$ 进行扰动生成对抗样本 $\mathbf{x}_{i,\rm adv}^{\mathbf{a}^+}$; 随后, 再通过梯度下降法利用生成的对抗样本 $\mathbf{x}_{i,\rm adv}^{\mathbf{a}^+}$ 对神经网络进行更新, 进行外层的损失最小化.

标准对抗训练利用固定攻击策略生成对抗样本, 其本质是通过最大化对抗损失来提升模型鲁棒性. 然而, 这一范式隐含一个关键矛盾: 模型的鲁棒性与干净样本分类精度之间存在动态权衡关系, 采用固定攻击策略无法感知模型训练状态的变化, 导致以下问题.

1) 训练动态不匹配: a) 早期训练阶段, 模型参数随机初始化, 防御能力薄弱, 强攻击 (过大的 ε、α、I) 易导致梯度爆炸或训练不稳定; b) 后期训练阶段, 模型逐渐收敛, 固定强度的攻击可能不足以进一步暴露模型脆弱性, 导致鲁棒性提升停滞.

2) 次优收敛: 固定策略只针对特定扰动模式对网络进行优化, 可能使模型过早陷入局部最优.

3) 鲁棒性与干净精度失衡: 固定策略倾向于生成高破坏性的对抗样本, 迫使模型过度拟合对抗样本特征, 牺牲对干净样本的分类性能, 难以在训练全程维持二者的均衡提升.

4) 参数敏感性: ε、α、I 的取值需人工调优, 且最优组合因数据集、模型结构而异, 缺乏普适性指导原则.

3.2 自适应策略优化

在标准对抗训练框架中, 我们将内层优化的攻击策略搜索进一步建模为无约束最优化问题, 在对抗训练的不同阶段, 通过改进的遗传算法动态调整攻击参数, 以实现模型鲁棒性与干净精度的协同提升. 该算法以阶段式搜索为核心, 在对抗训练的特定周期触发一次策略优化, 通过种群进化机制生成适合当前训练阶段的最优攻击策略, 最终与模型参数更新形成内外层交替优化.

3.2.1 攻击策略的参数编码和种群初始化

根据遗传算法的相关概念, 我们将攻击策略 $\mathbf{a}$ 视为种群中的个体, 多个攻击策略构成种群 $\mathcal{P}$, $\mathcal{P}\subseteq\mathcal{A}$ 是攻击策略空间 (解空间) 的子集. 攻击策略由 PGD 方法的核心参数构成, 即扰动强度 ε、扰动步长 α 和扰动步数 I, 编码为三元组 $\mathbf{a}=(\varepsilon,\alpha,I)$, 其中 ε 和 α 在实数范围内取值, 而 I 在整数范围内取值. 为了方便编码和计算, 将 3 个参数均进行实数编码, 当具体生成对抗样本时, 将 I 向下取整即可. 种群初始化时, 在预设参数范围内采用拉丁超立方抽样 (LHS) 生成初始个体, 确保参数空间的高覆盖性与多样性, 避免初始解聚集导致的搜索偏差.

3.2.2 选择操作的精英保留机制

选择操作采用混合策略: 每代保留适应度最高的前 M 个个体 (精英), 剩余个体通过无放回锦标赛选择产生. 具体而言, 从种群中随机选取若干个候选个体, 保留其中适应度最优者, 其余放回, 重复此过程直至补全种群规模为 n. 精英保留机制可防止高适应度策略在进化过程中丢失, 而锦标赛选择则通过竞争压力维持种群多样性, 缓解

早熟收敛问题.

3.2.3 参数重组与局部扰动

在遗传算法的进化过程中, 参数重组与局部扰动是驱动种群多样性与优良基因传递的核心机制. 针对攻击策略参数实数编码的特性, 采用模拟二进制交叉 (SBX) 对参数进行重组, 生成子代参数. 同时设置较大的交叉概率 p_c 以促进优良基因传播, 可在继承高适应度策略的同时, 维持较高的新个体生成率, 有效防止种群早熟收敛.

3.2.4 适应度函数的设计

种群进化需要合适的适应度函数作为引导, 受文献 [33] 启发, 我们设计个体 $\mathbf{a} = (\varepsilon, \alpha, I)$ 的适应度函数为:

$$\mathcal{F}(\mathbf{a}) = \alpha\mathcal{F}_1(\mathbf{a}) + \beta\mathcal{F}_2(\mathbf{a}) \tag{8}$$

其中, α和β 为平衡系数. 此适应度函数用于综合评估攻击策略的双重目标.

1) 鲁棒性提升: 一个优秀的攻击策略应该使得模型在此策略下生成的对抗样本能够进一步提升模型的鲁棒性, 反映当前策略生成的对抗样本用于模型 $f(\cdot, \mathbf{w})$ 更新后对标准攻击策略的抗干扰能力, 形式上表述为更新后的模型 $f(\cdot, \mathbf{w}')$ 对于标准攻击策略下生成的对抗样本的损失:

$$\mathcal{F}_1(\mathbf{a}) = -\sum\nolimits_{\mathbf{x}\in\mathcal{D}'} \mathcal{L}(f(\mathbf{x}_{\text{adv}}^{\mathbf{a}^+}, \mathbf{w}'), y)/|\mathcal{D}'| \tag{9}$$

其中, $\mathbf{a}^+ = (8/255, 2/255, 10)$ 是标准对抗训练使用的攻击策略, $\mathcal{D}'$ 是评估数据集.

2) 干净样本精度提升: 一个优秀的攻击策略应该同时使得模型在此策略下维持或提升其对于干净样本的识别准确率, 反映模型 $f(\cdot, \mathbf{w})$ 更新后在干净样本上的性能, 形式上表述为更新后的模型 $f(\cdot, \mathbf{w}')$ 对于干净样本的损失:

$$\mathcal{F}_2(\mathbf{a}) = -\sum\nolimits_{\mathbf{x}\in\mathcal{D}'} \mathcal{L}(f(\mathbf{x}, \mathbf{w}'), y)/|\mathcal{D}'| \tag{10}$$

其中, $\mathcal{D}'$ 是与公式 (9) 中一致的评估数据集.

3.2.5 自适应策略优化算法

将上述设计集成到对抗训练中, 阶段式触发, 每隔 K 个训练周期触发一次, 得到当前最优攻击策略 $\mathbf{a}^*$ 用于后续 K 个训练周期的对抗样本生成, 直至下一轮策略搜索触发, 形成“进化-训练”交替循环. 这种周期性的自适应策略更新机制, 使得模型在动态对抗环境中持续优化, 兼顾鲁棒性与泛化性. 算法 2 给出了对抗训练过程中内层优化的攻击策略搜索过程.

算法 2. 内层最大化问题的自适应策略优化 (adaptive strategy optimization, ASO).

输入: 进化代数 T, 种群大小 n, 攻击策略空间 $\mathcal{A} = S_\varepsilon \times S_\alpha \times S_I$, 适应度函数 $\mathcal{F}$, 精英保留数 M, 神经网络 $f(\cdot, \mathbf{w})$, 损失函数 $\mathcal{L}$, 学习率 η, 平衡系数 α 和 β, 评估数据 $\mathcal{D}' = \{(\mathbf{x}, y)\}$;

输出: 最优攻击策略 $\mathbf{a}^*$, 最优个体适应度 $\mathcal{F}(\mathbf{a}^*)$.

1. 利用拉丁超立方抽样从攻击策略空间中获得 n 个策略用于初始化第 0 代种群 $\mathcal{P}_0 = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\}$;
2. 初始化代数 $t = 0$;
3. While $t \leqslant T$ do;
4. $\quad \mathcal{P} = \mathcal{P}_t$;
   //计算种群 $\mathcal{P}$ 中每个攻击策略的适应度
5. $\quad$ For each $\mathbf{a} \in \mathcal{P}$ do
6. $\quad\quad$ For each $\mathbf{x} \in \mathcal{D}'$ do
7. $\quad\quad\quad \mathbf{x}_{\text{adv}}^{\mathbf{a}} = \text{PGD}(f(\cdot, \mathbf{w}), \mathbf{x}, \mathbf{a})$;
8. $\quad\quad$ End For
9. $\quad\quad loss = \sum\nolimits_{\mathbf{x}\in\mathcal{D}'} \mathcal{L}(f(\mathbf{x}_{\text{adv}}^{\mathbf{a}}, \mathbf{w}), y)/|\mathcal{D}'|$;
10. $\quad\quad \mathbf{w}' = \mathbf{w} - \eta\nabla_{\mathbf{w}} loss$;
11. $\quad\quad \mathbf{a}^+ = (8/255, 2/255, 10)$;

12. $\mathcal{F}_1(\mathbf{a}) = -\sum_{\mathbf{x}\in\mathcal{D}'} \mathcal{L}(f(\mathbf{x}_{\text{adv}}^{\mathbf{a}^*}, \mathbf{w}'), y)/|\mathcal{D}'|$;
13. $\mathcal{F}_2(\mathbf{a}) = -\sum_{\mathbf{x}\in\mathcal{D}'} \mathcal{L}(f(\mathbf{x}, \mathbf{w}'), y)/|\mathcal{D}'|$;
14. $\mathcal{F}(\mathbf{a}) = \alpha\mathcal{F}_1(\mathbf{a}) + \beta\mathcal{F}_2(\mathbf{a})$;
15. End For
16. $\mathcal{P}' \leftarrow$ 根据适应度 $\mathcal{F}(\mathbf{a})$ 大小排序选出前 M 个策略, 并通过锦标赛选择补全种群规模 n;
17. $\mathcal{P}'' \leftarrow$ 采用交叉和变异算子生成新的策略并添加到 $\mathcal{P}'$, 扩大种群规模为 $2n$;
18. For each $\mathbf{a} \in \mathcal{P}''$ do
19. $\mathcal{F}(\mathbf{a}) \leftarrow$ 依据步骤 6–14 计算适应度;
20. End For
21. $t = t + 1$;
22. $\mathcal{P}_t \leftarrow$ 根据适应度 $\mathcal{F}(\mathbf{a})$ 大小排序选出 $\mathcal{P}''$ 中前 n 个策略;
23. End While
24. $\mathbf{a}^* = \underset{\mathbf{a}\in\mathcal{P}_t}{\operatorname{argmax}}\, \mathcal{F}(\mathbf{a})$.

3.3 TRA^{2}SO 算法描述

3.3.1 训练框架

我们沿用标准对抗训练范式, 采用内外层交替优化来设计 TRA^{2}SO 算法, 创新点在于对内层优化问题的阶段性自适应策略搜索, 即每隔 K 个训练周期利用算法 2 寻找适合于现阶段的最优攻击策略 $\mathbf{a}^*$, 使得现阶段的神经网络 $f(\cdot,\mathbf{w})$ 根据干净样本生成的对抗样本, 对于其鲁棒性和干净精度的进一步提升仍具有效性. 本质上, 将公式 (1) 的损失函数替换为代理损失, 优化问题表示为:

$$\min_{\mathbf{w}} \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_{\text{CE}}(f(\mathbf{x}_{i,\text{adv}}^{\mathbf{a}^*}, \mathbf{w}), y_i) \tag{11}$$

其中, $\mathbf{x}_{i,\text{adv}}^{\mathbf{a}^*} = \mathbf{x}_i + \delta(\mathbf{x}_i, \mathbf{a}^*)$ 表示利用攻击策略 $\mathbf{a}^*$ 攻击样本 $\mathbf{x}_i$ 产生的对抗样本, $\delta(\cdot)$ 表示一个产生扰动的操作. TRA^{2}SO 的伪代码如算法 3.

算法 3. 基于自适应策略优化的鲁棒精度权衡学习 (TRA^{2}SO).

输入: 训练数据 $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, 损失函数 $\mathcal{L}$, 训练周期 E, 学习率 η, 批次大小 B, 进化代数 T, 种群大小 n, 攻击策略空间 $\mathcal{A} = S_\varepsilon \times S_\alpha \times S_I$, 适应度函数 $\mathcal{F}$, 精英保留数 M, 交替优化频次 K, 平衡系数 α 和 β, 适应度阈值 l;
输出: 神经网络 $f(\cdot,\mathbf{w})$.

1. 随机初始化神经网络 $f(\cdot,\mathbf{w})$;
2. 从训练数据 $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ 中随机采样用于评估的数据 $\mathcal{D}' = \{(\mathbf{x}, y)\}$;
3. $\mathcal{F}(\mathbf{a}) = 0$;
4. For *epoch* = 0 to E–1 do
   //阶段性自适应策略优化
5. If *epoch* mod K == 0
6. $\mathbf{a}^*, \mathcal{F}(\mathbf{a}^*) = \text{ASO}(T, n, \mathcal{A}, \mathcal{F}, M, f(\cdot,\mathbf{w}), \mathcal{L}, \eta, \alpha, \beta, \mathcal{D}')$;
7. If $|\mathcal{F}(\mathbf{a}^*) - \mathcal{F}(\mathbf{a})| < l$
8. Early stop;
9. End If
10. $\mathcal{F}(\mathbf{a}) = \mathcal{F}(\mathbf{a}^*)$;

11. End If
//外层优化
12. For batch = 1 to num_batch do
13. 从训练集 $\mathcal{D}=\{(\mathbf{x}_i,y_i)\}_{i=1}^{N}$ 中随机采样一个批次数据 $\mathcal{B}=\{(\mathbf{x}_i,y_i)\}_{i=1}^{B}$;
//内层优化
14. For $i=1$ to B do
15. $\mathbf{x}_{i,\mathrm{adv}}^{\mathbf{a}^*}=\mathrm{PGD}(f(\cdot,\mathbf{w}),\mathbf{x}_i,\mathbf{a}^*)$;
16. End For
17. $loss=\sum_{i=1}^{B}\mathcal{L}(f(\mathbf{x}_{i,\mathrm{adv}}^{\mathbf{a}^*},\mathbf{w}),y_i)/B$;
18. $\mathbf{w}=\mathbf{w}-\eta\nabla_{\mathbf{w}}loss$;
19. End For
20. End For

3.3.2 训练早停和过拟合预防

TRA^{2}SO 方法对于神经网络 $f(\cdot,\mathbf{w})$ 的训练模式是交替优化, 在训练的不同阶段寻找一种合适的攻击策略, 使得模型在此攻击策略下生成的对抗样本对于外层优化保持有效性. 此过程中, 算法记录了每次进化得到的最优个体的适应度函数值, 当相邻两次进化得到的最优个体的适应度值变化小于阈值 l 时, 可判定在策略空间中已无法找到更有效的攻击策略, 此时触发早停终止训练, 以避免鲁棒过拟合现象的发生 (算法 3 步骤 7–9).

3.3.3 优势总结

对比其他方法, 演化算法在提升深度模型鲁棒性方面的核心优势体现在以下两方面.

1) 演化算法对神经网络评估与优化问题的求解适应性强. 神经网络的训练过程需对模型进行合理的评估, 以发现鲁棒性的提升方向. 相较于其他方法, 演化算法以其种群驱动的启发式优化方式, 在不需要任何梯度信息的情况下即可对评估问题进行求解. 此外, 模型评估与模型更新交替进行, 以实现渐进式优化, 并通过超参数控制实现鲁棒性和干净准确率的权衡.

2) 演化优化信息对模型训练进程的决策具有指导作用. 演化算法在进行模型评估与模型优化时能够记录相应种群中的个体信息, 反映了当前神经网络在某些攻击策略下鲁棒性的提升状态. 可借助这部分信息对模型的训练状态进行动态分析, 从而对训练早停等决策提供指导.

3.4 计算复杂度分析

与标准对抗训练相比, TRA^{2}SO 利用算法 2 搜索当前最优攻击策略时有额外的时间耗费. 这部分时间复杂度主要取决于种群大小 n 和进化代数 T. 此外, 在整个数据集上评估所有攻击策略的复杂度过高, 时间成本上是不现实的, 因此 TRA^{2}SO 采样出了一些数据 $\mathcal{D}'=\{(\mathbf{x},y)\}$ 用于策略评估. 最后, 考虑 PGD 攻击下扰动步数 I 的优化空间 S_I, 并假设单个样本在网络中经过一次前向和反向传播的时间成本为 $\mathrm{O}(W)$, 则 TRA^{2}SO 相较于标准对抗训练的额外时间复杂度介于 $\mathrm{O}(n\times T\times|\mathcal{D}'|\times\min\{S_I\}\times W)$ 和 $\mathrm{O}(n\times T\times|\mathcal{D}'|\times\max\{S_I\}\times W)$ 之间. 然而, 值得注意的是, TRA^{2}SO 可结合早停策略来平衡额外的时间成本. 通过记录每次进化时最优个体的适应度值, 当两次相邻进化中最优个体适应度值变化不大时, 说明模型已无法再寻找到一个更优的攻击策略使得鲁棒性和干净精度同时提升, 此时可停止训练, 不再需要剩余的时间开销.

4 实验设计与比较

4.1 实验设置

4.1.1 实验数据

为了验证本文所提出的 TRA^{2}SO 方法的有效性, 我们在 CIFAR-10 和 CIFAR-100 两个广泛用于图像分类研

究的数据集上进行实验. CIFAR-10 数据集包含 60 000 张 32×32×3 的彩色图像, 其中 50 000 张用于训练, 10 000 张用于测试, 共计 10 个类别, 每个类别有 6 000 张图像, 这些类别包括飞机、汽车、鸟类、猫、鹿、狗、蛙类、马、船和卡车. CIFAR-100 是 CIFAR-10 的扩展版本, 包含 100 个细分类别, 每个类别有 600 张图像, 总计 60 000 张图像. 由于 CIFAR-100 的 100 个类别又可以被分为 20 个超类, 故每个图像都有一个细粒度标签和一个粗粒度标签. 这两个数据集的图像虽小但类别丰富, 且每个类别的图像具有一定多样性与重叠性, 为分类任务增加了挑战性, 故常用于深度模型的训练与评估, 尤其适合针对卷积神经网络等架构的算法验证. 在模型训练过程中, 为了增加数据集的多样性和复杂性, 同时有效防止模型过拟合, 我们进行了数据增强处理, 具体采用了水平翻转和随机裁切两种增强方法.

4.1.2 参数设置

我们采用 ResNet-18 作为骨干网络架构, 并利用随机梯度下降法 SGD (stochastic gradient descent) 对其进行优化, 其中训练周期 (*epoch*) 设置为 200, 初始学习率为 0.1, 动量参数 (momentum) 为 0.9, 权重衰减系数 (weight decay) 为 0.000 5. 此外, 为了在训练后期对网络参数进行精细调整, 配合两种学习率衰减策略: 第 1 种衰减策略 (lr1) 每经过 50 个训练周期后, 将学习率锐减至其先前值的 0.1 倍; 第 2 种衰减策略 (lr2) 在训练周期达到 100 和 150 时, 分别将学习率降至其当前值的 0.1 倍. 遗传算法使用 Geatpy 库实现, 种群大小 n 设置为 30, 进化代数 T 设置为 20, 精英保留数 M 设置为 5, 其余参数均采用默认设置. 策略空间 $\mathcal{A} = S_\epsilon \times S_\alpha \times S_I = [2/255, 16/255]\times[1/255, 4/255]\times[4,20]$, 适应度函数阈值 l 设置为 0.000 1, 评估数据集 $\mathcal{D}'$ 从训练集中随机采样获得, 大小设置为 200, 交替优化频率 K 设置为 30. 我们每隔 10 个训练周期进行一次模型保存, 并利用 3 种方式对模型进行选择, 第 1 种是选择训练结束后的最终模型 (final), 第 2 种是选择训练过程中对抗精度最优的模型 (adv), 第 3 种是选择训练过程中干净精度和对抗精度之和最优的模型 (sum).

4.1.3 模型评估

为了全面评估模型的鲁棒性, 我们选用了 3 类攻击方法: 基于梯度的攻击、基于优化的攻击和集成攻击, 具体包括 L_2和L_∞ 范数约束. 对于梯度攻击, 我们使用 L_∞ 范数约束的 FGSM 和 PGD; 对于优化攻击, 我们使用 L_2 范数约束的 C&W; 对于集成攻击, 则使用 L_∞ 范数约束的 AA. 攻击强度 ε 统一设置为 8/255; PGD 迭代攻击的步长设置为 2/255, 步数分别取 10、20、50 从而构成 PGD-10、PGD-20、PGD-50 这 3 种具体的攻击; C&W 攻击的初始常数设置为 1, confidence 设置为 0, 迭代次数设置为 50, 学习率设置为 0.01. FGSM、PGD 和 C&W 统一由 torchattacks 库实现, 而 AA 使用原文给出的源码实现并选取其标准模式.

4.2 结果分析

4.2.1 与定制策略调整方法的性能对比

我们将本文所提出的 TRA^{2}SO 方法与固定攻击策略下的标准对抗训练方法 PGD-AT 进行比较; 此外, 为了验证 TRA^{2}SO 相较于人为制定策略的优势, 我们还设计若干线性增长攻击强度的方式 PGD-AT-linear. 针对 TRA^{2}SO, 其平衡系数 α和β 均设置为 2, 交替优化的频率设定为 30 个训练周期, 且学习率调整方式运用 lr1; 针对标准对抗训练 PGD-AT, 采用固定参数的 PGD 攻击 $\mathbf{a}^+ = (8/255, 2/255, 10)$ 作为内层优化的手段, 学习率调整方式则同时运用 lr1 和 lr2; 针对线性增长策略 PGD-AT-linear, 其步长 α 和步数 I 固定为标准值 2/255 和 10, 而强度 ε 依据训练周期线性增加, 具体采用 3 种最大强度, 取 $\varepsilon_{\max}$ 分别为 8/255、12/255、16/255, 则:

$$\varepsilon_{epoch} = epoch \times \varepsilon_{\max}/200 \tag{12}$$

学习率调整方式也同时运用 lr1 和 lr2. 以上方法在几种通用的对抗攻击下进行评估, 其在 CIFAR-10 和 CIFAR-100 上的结果分别如表 1–表 3 和表 4–表 6 所示.

表 1–表 6 中, final 指代选择最终模型进行评估; adv 表示选择 PGD-10 攻击下鲁棒准确率最高的模型进行评估; sum 指代选择鲁棒准确率和干净准确率二者之和最高的模型进行评估. 从表中可以看出, 不论是选择哪个阶段的模型, TRA^{2}SO 都可以在测试集上获得最高的鲁棒精度, 特别是在干净精度的牺牲程度可接受的情况下, 仍然得到了最高的鲁棒精度. 值得指出的是, TRA^{2}SO 所获得的 final 性能明显优于标准对抗训练 PGD-AT, 说明其确实可以实现攻击策略的自适应调整, 从而避免模型出现鲁棒过拟合或严重的鲁棒过拟合, 使得最终的性能表现最优.

表 1 CIFAR-10 数据集上各种策略调整方法最终模型 (final) 的测试准确率对比 (%)

方法	lr	Clean	FGSM	PGD-10	PGD-20	PGD-50	AA	C&W
PGD-AT	(lr1)	83.77	52.29	43.43	41.86	41.41	40.07	6.90
	(lr2)	84.42	52.13	43.63	42.36	42.07	40.95	7.74
PGD-AT-linear (ε_{max} = 8/255)	(lr1)	**89.30**	50.85	37.58	35.00	34.26	32.26	1.76
	(lr2)	89.10	53.46	41.15	39.19	38.74	37.00	5.82
PGD-AT-linear (ε_{max} = 12/255)	(lr1)	86.62	55.33	44.66	42.39	41.67	38.29	8.51
	(lr2)	86.00	55.99	46.68	44.67	44.16	42.22	7.77
PGD-AT-linear (ε_{max} = 16/255)	(lr1)	84.82	**56.61**	47.16	45.08	44.55	40.54	**15.45**
	(lr2)	84.18	56.01	47.65	46.14	45.62	43.18	11.60
TRA^2SO	(lr1)	82.04	54.63	**47.88**	**46.41**	**45.95**	**43.66**	13.12

表 2 CIFAR-10 数据集上各种策略调整方法鲁棒模型 (adv) 的测试准确率对比 (%)

方法	lr	Clean	FGSM	PGD-10	PGD-20	PGD-50	AA	C&W
PGD-AT	(lr1)	82.12	56.61	51.17	50.28	50.00	46.63	24.65
	(lr2)	84.54	56.20	50.24	49.16	48.82	46.35	16.30
PGD-AT-linear (ε_{max} = 8/255)	(lr1)	**89.35**	51.19	37.58	35.08	34.41	32.23	1.67
	(lr2)	89.22	53.03	41.39	39.35	38.77	36.95	6.44
PGD-AT-linear (ε_{max} = 12/255)	(lr1)	86.58	55.37	44.68	42.65	41.92	38.52	8.66
	(lr2)	85.95	56.20	47.01	45.21	44.70	42.49	8.06
PGD-AT-linear (ε_{max} = 16/255)	(lr1)	84.80	56.78	47.13	45.18	44.48	40.05	15.78
	(lr2)	84.49	56.63	47.99	46.46	45.83	43.33	11.50
TRA^2SO	(lr1)	78.04	**57.55**	**54.35**	**53.66**	**53.58**	**48.89**	**39.88**

表 3 CIFAR-10 数据集上各种策略调整方法综合模型 (sum) 的测试准确率对比 (%)

方法	lr	Clean	FGSM	PGD-10	PGD-20	PGD-50	AA	C&W
PGD-AT	(lr1)	83.15	56.35	50.48	49.54	49.25	46.37	21.24
	(lr2)	84.90	56.80	50.14	48.83	48.49	46.17	13.16
PGD-AT-linear (ε_{max} = 8/255)	(lr1)	**89.35**	51.19	37.53	35.18	34.28	32.25	1.67
	(lr2)	89.22	53.03	41.35	39.31	38.78	36.94	6.45
PGD-AT-linear (ε_{max} = 12/255)	(lr1)	86.58	55.37	44.73	42.57	41.86	38.53	8.67
	(lr2)	88.63	55.55	45.33	43.29	42.56	40.60	3.81
PGD-AT-linear (ε_{max} = 16/255)	(lr1)	84.80	56.78	47.13	45.18	44.48	40.05	15.78
	(lr2)	87.68	56.10	47.38	45.48	45.01	42.73	6.92
TRA^2SO	(lr1)	82.71	**57.67**	**52.73**	**51.75**	**51.56**	**47.96**	**28.43**

表 4 CIFAR-100 数据集上各种策略调整方法最终模型 (final) 的测试准确率对比 (%)

方法	lr	Clean	FGSM	PGD-10	PGD-20	PGD-50	AA	C&W
PGD-AT	(lr1)	55.93	24.73	20.40	19.67	19.33	18.60	3.08
	(lr2)	55.69	24.51	20.04	19.43	19.27	18.54	2.73
PGD-AT-linear (ε_{max} = 8/255)	(lr1)	**64.63**	22.39	15.24	14.07	13.75	12.76	0.55
	(lr2)	63.07	24.60	18.05	16.87	16.58	15.54	1.17
PGD-AT-linear (ε_{max} = 12/255)	(lr1)	61.37	25.12	19.12	18.21	17.86	16.94	1.31
	(lr2)	59.57	25.83	21.00	20.00	19.77	18.90	2.25
PGD-AT-linear (ε_{max} = 16/255)	(lr1)	58.65	25.30	20.10	19.03	18.82	17.92	2.03
	(lr2)	56.89	26.29	21.88	20.94	20.77	19.79	3.16
TRA^2SO	(lr1)	51.22	**27.69**	**24.56**	**23.98**	**23.92**	**21.84**	**6.20**

表 5 CIFAR-100 数据集上各种策略调整方法鲁棒模型 (adv) 的测试准确率对比 (%)

方法	lr	Clean	FGSM	PGD-10	PGD-20	PGD-50	AA	C&W
PGD-AT	(lr1)	57.04	29.63	26.32	25.75	25.52	22.84	10.24
	(lr2)	56.70	27.74	24.08	23.57	23.38	21.19	6.91
PGD-AT-linear	(lr1)	**64.63**	22.39	15.31	14.05	13.78	12.76	0.55
($\varepsilon_{\max}$ = 8/255)	(lr2)	63.07	24.60	18.12	17.02	16.63	15.65	1.17
PGD-AT-linear	(lr1)	61.37	25.12	19.21	18.24	17.80	16.96	1.30
($\varepsilon_{\max}$ = 12/255)	(lr2)	59.74	25.80	21.01	19.96	19.71	18.93	2.24
PGD-AT-linear	(lr1)	58.94	25.25	20.16	19.40	19.15	18.02	1.97
($\varepsilon_{\max}$ = 16/255)	(lr2)	61.71	27.52	22.68	21.51	21.11	19.48	4.58
TRA^{2}SO	(lr1)	53.78	**30.36**	**28.19**	**28.02**	**27.92**	**24.37**	**14.38**

表 6 CIFAR-100 数据集上各种策略调整方法综合模型 (sum) 的测试准确率对比 (%)

方法	lr	Clean	FGSM	PGD-10	PGD-20	PGD-50	AA	C&W
PGD-AT	(lr1)	57.04	29.63	26.44	25.78	25.61	22.83	10.23
	(lr2)	56.87	27.89	24.17	23.49	23.33	21.45	6.61
PGD-AT-linear	(lr1)	64.69	22.42	15.14	13.86	13.63	12.59	0.58
($\varepsilon_{\max}$ = 8/255)	(lr2)	66.24	23.78	16.06	14.69	14.42	13.33	1.67
PGD-AT-linear	(lr1)	63.30	24.36	17.96	16.64	16.38	15.66	1.17
($\varepsilon_{\max}$ = 12/255)	(lr2)	63.82	26.21	20.39	19.33	18.88	17.42	3.15
PGD-AT-linear	(lr1)	**67.41**	21.83	14.31	12.88	12.31	11.04	1.34
($\varepsilon_{\max}$ = 16/255)	(lr2)	61.71	27.52	22.63	21.58	21.15	19.46	4.58
TRA^{2}SO	(lr1)	54.50	**30.42**	**28.07**	**27.50**	**27.51**	**24.10**	**12.96**

4.2.2 模型收敛趋势分析

为了深入分析模型的收敛趋势并监测过拟合现象, 我们记录了以上方法在 CIFAR-10 测试集上的两项关键指标, 即干净精度和鲁棒精度的变化趋势. 其中, 干净精度反映了模型在未受干扰的原始数据上的性能表现, 而鲁棒精度则通过生成对抗样本来评估模型在对抗环境下的稳定性. 此部分实验中, 鲁棒精度的评估采用 PGD-10 攻击生成对抗样本. 图 2 和图 3 展示了各种训练方式下鲁棒精度和干净精度随训练周期的变化趋势. 图 4 展示了 TRA^{2}SO 针对攻击策略的阶段性自适应优化结果.

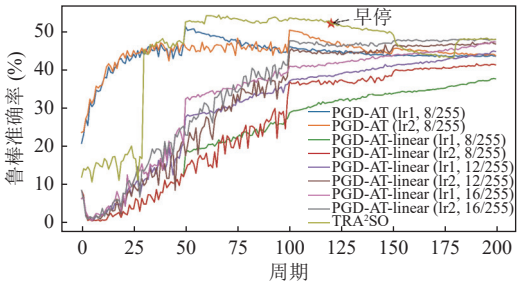


图 2 各策略在 CIFAR-10 上的鲁棒准确率

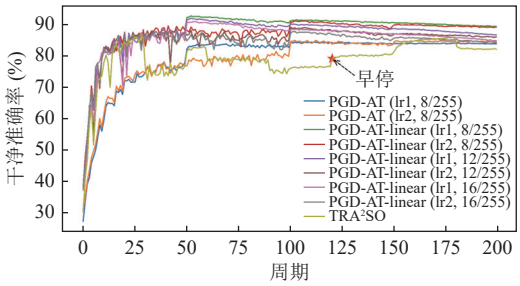
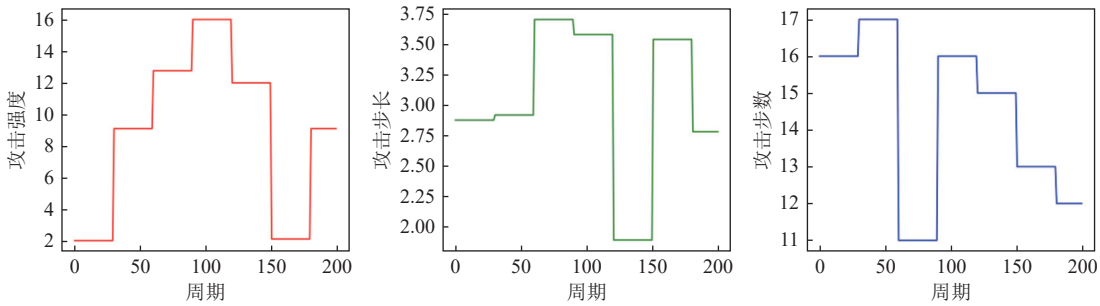


图 3 各策略在 CIFAR-10 上的干净准确率

从图 2 和图 3 中可以看出, 在标准对抗训练 PGD-AT 下, 不论使用哪一种学习率调整方式, 模型在训练后期均会陷入鲁棒过拟合, 并且只有在学习率调整后, 对抗准确率有短暂的上升, 之后仍然出现了过拟合现象. 相比之下, TRA^{2}SO 可以使模型更快更早地获得更高的鲁棒性, 同时可以在相邻两次攻击策略调整时, 依据适应度函数值的变化来判断是否实施早停, 从而避免严重的过拟合现象发生. 由于 TRA^{2}SO 在训练过程中会阶段性地调整策略, 这使得模型即使陷入过拟合, 依然可以借助策略调整从过拟合中解脱出来. 综上所述, TRA^{2}SO 可以在训练初期有效实现对抗精度和干净精度的同步提升, 且当二者在某阶段呈现出相反的趋势时, 依然对鲁棒性提升有效. 该结果

同时证实了模型的鲁棒性与干净精度之间的确存在着动态权衡的关系.

此外, 图 2 和图 3 还展示了 3 种线性增长策略 PGD-AT-linear 在训练过程中模型的鲁棒精度和干净精度的变化. 不同于 TRA^{2}SO 和 PGD-AT, 这 3 种策略在训练期间的模型大多处于鲁棒欠拟合状态, 它们过分提高了干净准确率, 而牺牲了较多的对抗准确率, 降低了模型的可信性. 相比而言, TRA^{2}SO 在可接受的程度下, 尽管损失了部分干净准确率但不会使模型的鲁棒性呈欠拟合状态.



(a) 攻击强度 (b) 攻击步长 (c) 攻击步数

图 4 训练各阶段 TRA^{2}SO 在 CIFAR-10 上的攻击策略优化结果 (攻击强度和攻击步长为乘以 255 后的值)

策略优化的结果方面, 从图 4(a) 中可知, TRA^{2}SO 在训练前期和中期攻击强度呈阶梯状上升, 故图 2 中对抗精度在最初期几乎没有变化, 在执行第一次策略调整后, 随着攻击强度的增大, 对抗精度呈现陡然上升的趋势. 与此同时, 如图 3 所示, 干净精度亦遭受了一定程度的牺牲, 此变化符合我们的直观预期, 为 TRA^{2}SO 提供了可解释性, 亦可证实使用 TRA^{2}SO 训练出的模型具有更高的可信性. 此外, 攻击强度的变化与攻击步数和攻击步长的变化并没有呈现完全的一致性, 体现了对于参数组合进行总体优化的必要性.

4.2.3 适应度函数分析

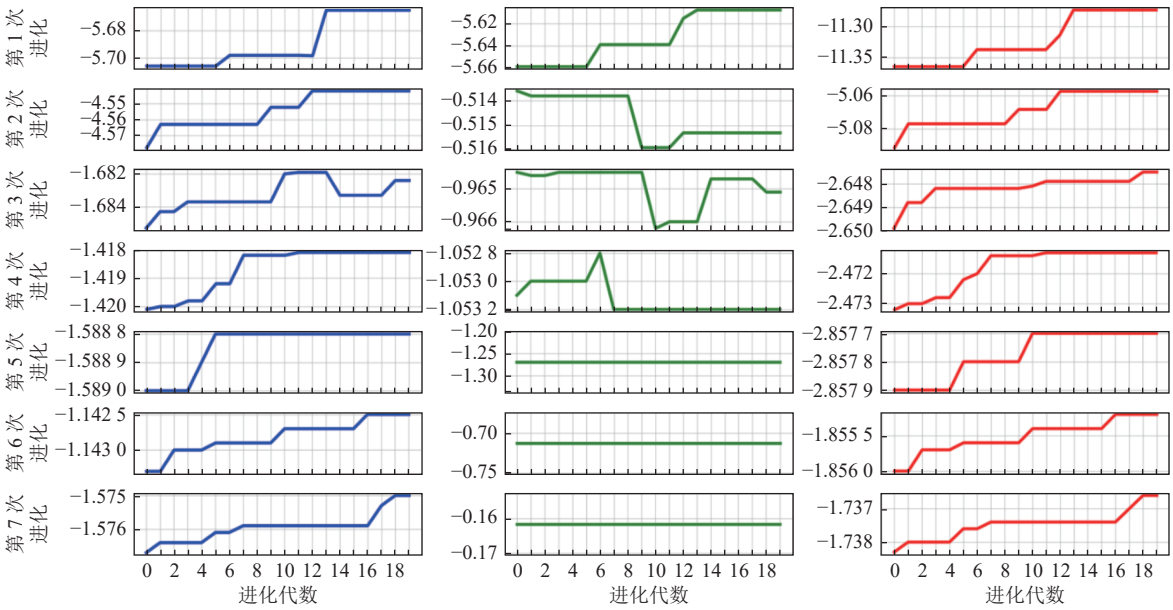
直观来讲, TRA^{2}SO 每次更换攻击策略时, 需要在策略空间中搜索最优解. 鉴于遗传算法的特点, 可以通过感知适应度函数值的变化, 来判断目前策略空间中是否还存在使得模型泛化性和鲁棒性都能进一步提升的解. 因此, 我们记录相关数据以验证适应度函数值的变化与模型泛化性和鲁棒性变化的关联. 训练过程中, 我们记录了 CIFAR-10 数据集每一次进化过程中每一代种群最优个体的适应度函数值 $\mathcal{F}$, 同时记录了适应度函数中两部分各自的值, 即代表鲁棒性提升的 $\mathcal{F}_1$ 和代表干净精度提升的 $\mathcal{F}_2$. 其中, 权衡系数 α 和 β 均设置为 2, 交替优化频率为 30, 学习率调整方式为 lr1. 后文图 5 展示了 $\mathcal{F}$、$\mathcal{F}_1$ 和 $\mathcal{F}_2$ 函数值在每一次进化时分别的收敛情况. 依据图 5 中展示的收敛情况, 如果相邻两次进化的最优适应度函数值差别可以忽略, 或者适应度函数的一部分在进化时已不再变化, 可以认为目前的策略空间中已经找不到一个更好的攻击策略使得模型的鲁棒性和干净准确率进一步提升, 此时认为模型已经收敛, 故停止训练, 从而防止模型陷入过拟合. 结合图 2 得知, 在 CIFAR-10 数据集上执行了 5 次进化后, 模型便提早停止更新, 即图 2 中红色五角星所对应的早停点.

4.2.4 消融实验和权衡分析

为了评估攻击策略的优劣程度, 我们将适应度函数设计为两部分, 一部分用于评估攻击策略对于模型鲁棒性的提升, 另一部分用于评估其对模型干净精度的提升. 通过设置两部分的权重, 我们设计了两个消融实验: 第 1 个实验中, 鲁棒性提升权重 α 设置为 0.999, 干净精度提升权重 β 设置为 0.001; 第 2 个实验中, 鲁棒性提升权重 α 设置为 0.001, 干净精度提升权重 β 设置为 0.999. 相应的模型准确率变化和策略变化如图 6 和图 7 所示.

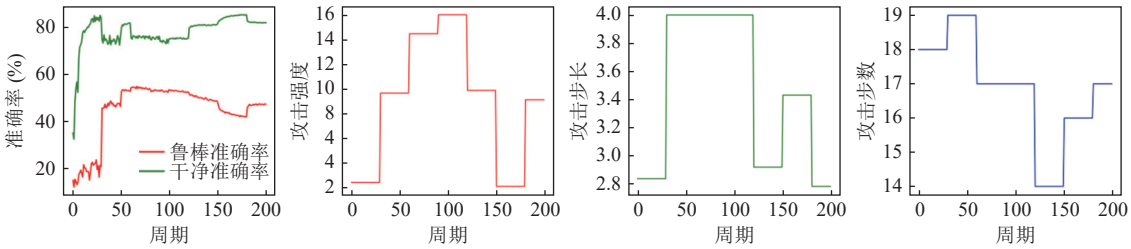
从图 6 和图 7 观察可知, 当优化目标侧重于不同的性能提升时, 攻击策略的变化呈现出显著差异. 当 $\alpha = 0.999$, $\beta = 0.001$ 时, 模型侧重于提升鲁棒性 $\mathcal{F}_1$, 攻击强度在第 1 次优化后相应地提高至 15/255 左右; 而后期当鲁棒性提升不大时, 为了寻找适应度值更高的解, 遗传算法选择提升另一部分 $\mathcal{F}_2$, 使得攻击强度下降到较低的位置, 从而造成对应的鲁棒准确率有所下降, 干净准确率提升. 当 $\alpha = 0.001$, $\beta = 0.999$ 时, 模型侧重于提升干净精度 $\mathcal{F}_2$, 相对应的攻击强度呈现出阶段性提升, 一方面, 为了保证干净精度, 其最大攻击强度没有超过标准值 8/255; 另一方面, 训

练前期算法逐步提升攻击强度时, 可以观察到两种准确率均稳步上升, 并最终达到收敛状态. 综上所述, 我们可以从攻击策略的变化分析出准确率的变化, 这说明 TRA^{2}SO 具有较强的可解释性, 同时所训练出来的模型亦具有较强的可信性.



(a) $\mathcal{F}_1$ (b) $\mathcal{F}_2$ (c) $\mathcal{F}$

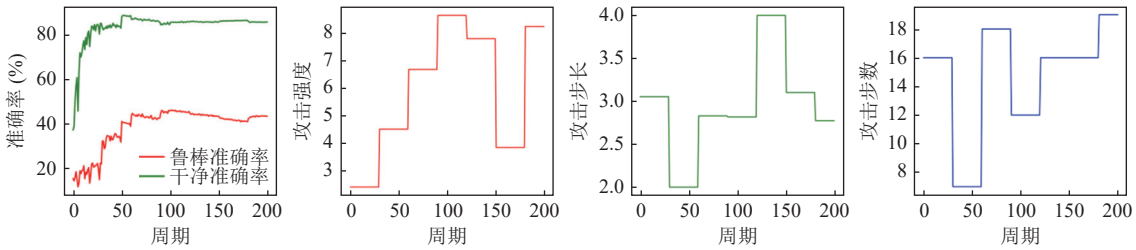
图 5 TRA^{2}SO 在 CIFAR-10 上每次进化的最优个体适应度函数的变化图



(a) 准确率变化 (b) 攻击强度 (c) 攻击步长 (d) 攻击步数

图 6 侧重鲁棒性提升时 TRA^{2}SO 在 CIFAR-10 上的准确率变化图和策略变化图

(攻击强度和攻击步长为×255 后的值)



(a) 准确率变化 (b) 攻击强度 (c) 攻击步长 (d) 攻击步数

图 7 侧重干净精度提升时 TRA^{2}SO 在 CIFAR-10 上的准确率变化图和策略变化图

(攻击强度和攻击步长为×255 后的值)

4.3 与 SOTA 方法的性能对比

为了充分验证 TRA^{2}SO 方法的有效性, 我们选择了多个经典的对抗训练方法进行对比, 包括标准对抗训练 PGD-AT、TRADES[7]、MART[8]、FAT[34]、GAIRAT[35]、MAIL[36]、AT+RiFT[37]. 所有方法的主干网络均采用 ResNet18 或 PreActResNet18, 采用原文所提供的源代码及默认设置进行实现, 包括训练参数设置, 早停设置及最优模型选择. 在 CIFAR-10 和 CIFAR-100 两个数据集上的对比结果如表 7 和表 8 所示, 其中 TRA^{2}SO 方法同时汇报了早停模型 (early) 和鲁棒模型 (adv) 的评估结果.

表 7 各对抗训练方法在 CIFAR-10 数据集上的模型测试准确率对比 (%)

方法	Clean	FGSM	PGD-10	PGD-20	PGD-50	AA	C&W	Time (s)
PGD-AT	84.42	52.13	43.63	42.36	42.07	40.95	7.74	43 857
TRADES	83.42	**58.77**	52.68	51.73	51.49	48.65	30.24	44 057
MART	82.00	56.92	49.76	48.30	47.83	43.36	22.52	29 750
FAT	**86.98**	53.99	43.94	41.87	41.28	39.55	8.96	45 154
GAIRAT	82.33	54.44	50.80	49.65	49.62	32.60	**67.03**	46 276
MAIL	84.03	55.96	51.29	50.64	50.31	46.82	24.19	101 026
AT+RiFT	76.78	44.09	38.57	37.77	37.45	35.19	23.13	127 665
TRA^{2}SO (early)	79.06	56.78	51.94	50.85	50.51	47.25	29.42	47 694
TRA^{2}SO (adv)	78.04	57.55	**54.35**	**53.66**	**53.58**	**48.89**	39.88	47 694

表 8 各对抗训练方法在 CIFAR-100 数据集上的模型测试准确率对比 (%)

方法	Clean	FGSM	PGD-10	PGD-20	PGD-50	AA	C&W	Time (s)
PGD-AT	55.93	24.73	20.40	19.67	19.33	18.60	3.08	44 098
TRADES	55.50	30.56	27.82	27.48	27.27	23.54	14.37	40 519
MART	53.17	29.18	26.52	25.92	25.84	22.60	9.26	29 877
FAT	61.29	25.60	19.73	18.77	18.52	17.75	1.94	41 485
GAIRAT	55.23	24.20	20.12	19.30	19.08	16.36	9.65	42 616
MAIL	**61.33**	**31.29**	28.06	27.44	27.35	22.64	12.56	114 857
AT+RiFT	45.08	20.94	18.77	18.54	18.41	15.38	**17.26**	125 939
TRA^{2}SO (early)	51.41	27.61	24.42	23.82	23.58	21.72	6.52	41 827
TRA^{2}SO (adv)	53.78	30.36	**28.78**	**28.01**	**27.88**	**24.37**	14.38	41 827

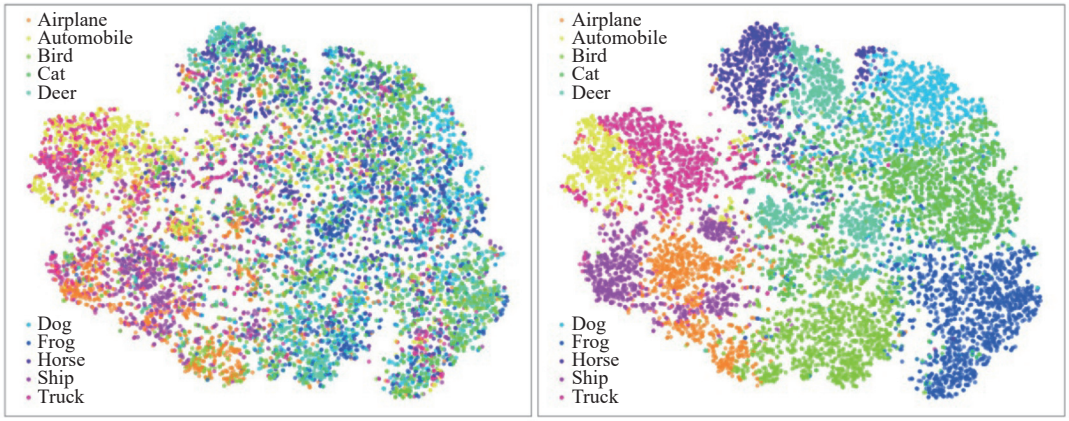
表 7 和表 8 中的最优性能用粗体标识, 第 2 和第 3 好的性能用下划线和波浪线标识. 可以看出, 在 CIFAR-10 数据集上, TRA^{2}SO (adv) 在多种攻击下均表现出第 1 的性能, TRA^{2}SO (early) 在所有攻击下也实现了前 3 的性能, 且从运行时间上分析, 结合早停策略, 计算成本与标准对抗训练 PGD-AT 相比没有明显增加. 在 CIFAR-100 数据集上, TRA^{2}SO 也可在相对较小的时间开销下获得一个不错的模型, 尽管其早停策略下的最终模型性能不是非常理想, 但其鲁棒模型在多种攻击的评估下均表现出第 1 或第 2 的性能, 体现了 TRA^{2}SO 方法整体的有效性.

4.4 可视化分析

由于使用神经网络进行分类时, 最终得到的嵌入特征一般都是高维度的, 难以可视化. 因此, 我们使用 t-SNE 方法[38]对嵌入特征进行降维, 并在二维空间中进行可视化展示. 选取 TRA^{2}SO 方法在训练初期、中期和后期得到的模型, 并利用 PGD-10 生成对抗样本, 其可视化效果分别如图 8–图 10 所示.

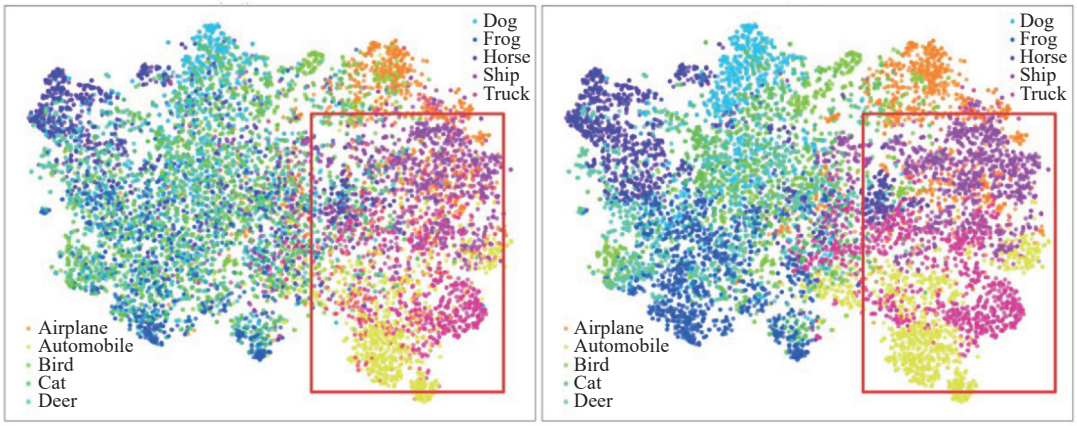
如图 8 所示, 在训练初期, 模型针对对抗样本的识别效果很差, 难以学习到明确的判别特征, 还未建立起清晰的分类边界, 各个类别的样本特征杂糅在一起, 预测标签和真实标签差异巨大. 如图 9 所示, 在训练中期, 模型已能针对部分类别 (红框中) 的样本特征建立起较为清晰的边界, 可以对这些类别的样本正确分类; 同时, 处于边界的部分样本也形成了较为明显的簇, 其分类准确率也大幅度上升. 如图 10 所示, 在训练后期, 模型已经能够将中期分类较

好的样本 (红框中) 分为完整的簇, 对于这些簇的分类几乎完全正确; 对于边界的部分样本, 也形成了较为完整的簇, 分类结果也较好; 对于黑框中难以分类的样本, 虽然没有形成较好的簇, 但其分类结果相对于中期也有所提升.



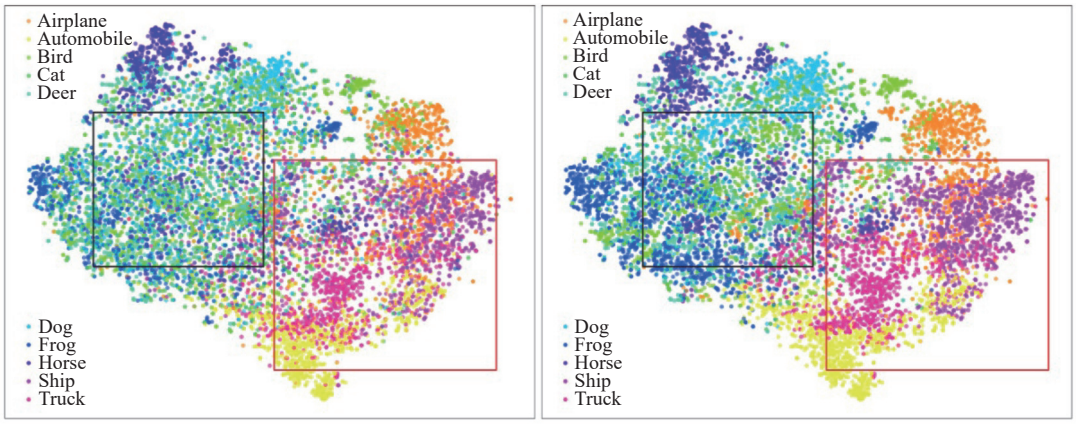
(a) 真实标签 (b) 预测标签

图 8 TRA^{2}SO 训练初期 CIFAR-10 数据集中对抗样本降维特征的真实标签和预测标签可视化



(a) 真实标签 (b) 预测标签

图 9 TRA^{2}SO 训练中期 CIFAR-10 数据集中对抗样本降维特征的真实标签和预测标签可视化



(a) 真实标签 (b) 预测标签

图 10 TRA^{2}SO 训练后期 CIFAR-10 数据集中对抗样本降维特征的真实标签和预测标签可视化

4.5 小 结

通过上述一系列实验分析, 我们观察到, 对抗训练过程中鲁棒性与自然准确率之间存在显著的动态竞争关系. 这一现象在模型训练的特定阶段 (通常发生在中后期) 尤为突出, 表现为鲁棒性的提升伴随自然准确率的下降. 这一现象说明, 在模型容量有限的情况下, 其对于对抗样本的鲁棒性与干净样本的泛化能力也是有限的, 两者不能够同时持续增长, 体现了权衡学习的重要性.

5 总 结

本文提出了一种对抗训练方式 TRA^{2}SO, 借助遗传算法和攻击策略等概念, 通过对双层优化问题的内层进行改进, 在对抗训练的不同阶段自适应地获取最适合当前模型的攻击策略, 使得模型的鲁棒性和泛化性取得了一个良好的平衡. 相较于标准对抗训练, TRA^{2}SO 方法训练得到的模型鲁棒性更强, 收敛速度更快; 同时, 攻击策略的变化情况为模型鲁棒性和泛化性的变化趋势作出了合理解释; 此外, 通过记录适应度函数值的变化, 可以分析模型的收敛程度和收敛效果, 从而为模型的早停作出引导, 避免了其对训练数据的过度拟合. 大量实验证明, TRA^{2}SO 具有自适应性, 同时可增强模型的可信性.

References

[1] He KM, Zhang XY, Ren SQ, Sun J. Deep residual learning for image recognition. In: Proc. of the 2016 IEEE Conf. on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 770–778. [doi: 10.1109/CVPR.2016.90]

[2] Collobert R, Weston J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In: Proc. of the 25th Int'l Conf. on Machine Learning. Helsinki: Association for Computing Machinery, 2008. 160–167. [doi: 10.1145/1390156.1390177]

[3] Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv:1301.3781, 2013.

[4] 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 31st Conf. on Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017. 6000–6010. [doi: 10.5555/3295222.3295349]

[5] Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, Fergus R. Intriguing properties of neural networks. In: Proc. of the 2nd Int'l Conf. on Learning Representations. Banff, 2014. 14–16. [doi: 10.48550/arXiv.1312.6199]

[6] Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A. Towards deep learning models resistant to adversarial attacks. arXiv:1706.06083, 2017.

[7] Zhang HY, Yu YD, Jiao JT, Xing E, El Ghaoui L, Jordan M. Theoretically principled trade-off between robustness and accuracy. In: Proc. of the 36th Int'l Conf. on Machine Learning. Long Beach: PMLR, 2019. 7472–7482. [doi: 10.48550/arXiv.1901.08573]

[8] Wang YS, Zou DF, Yi JF, Bailey J, Ma XJ, Gu QQ. Improving adversarial robustness requires revisiting misclassified examples. arXiv:2103.08307, 2021.

[9] Wu HM, Shi WL, Zhang CK, Gu B. Self-adaptive perturbation radii for adversarial training. In: Proc. of the 29th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining. Long Beach: ACM, 2023. 2570–2581. [doi: 10.1145/3580305.3599495]

[10] Yang S, Xu C. One size does not fit all: Data-adaptive adversarial training. In: Proc. of the 17th European Conf. on Computer Vision. Tel Aviv: Springer, 2022. 70–85. [doi: 10.1007/978-3-031-20065-6_5]

[11] Yu CJ, Han B, Gong MM, Shen L, Ge SM, Du B, Liu TL. Robust weight perturbation for adversarial training. In: Proc. of the 31st Int'l Joint Conf. on Artificial Intelligence. Vienna, 2022. 3688–3694. [doi: 10.24963/ijcai.2022/512]

[12] Wang WX, Wang CL, Qi HH, Ye MH, Qian XL, Wang P, Zhang YN. Sustainable self-evolution adversarial training. In: Proc. of the 32nd ACM Int'l Conf. on Multimedia. Melbourne: ACM, 2024. 9799–9808. [doi: 10.1145/3664647.3681077]

[13] Goodfellow IJ, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. arXiv:1412.6572, 2014.

[14] Kurakin A, Goodfellow IJ, Bengio S. Adversarial examples in the physical world. arXiv:1607.02533, 2016.

[15] Dong YP, Liao FZ, Pang TY, Su H, Zhu J, Hu XL, Li JG. Boosting adversarial attacks with momentum. In: Proc. of the 2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 9185–9193. [doi: 10.1109/CVPR.2018.00957]

[16] Carlini N, Wagner D. Towards evaluating the robustness of neural networks. In: Proc. of the 2017 IEEE Symp. on Security and Privacy. San Jose: IEEE, 2017. 39–57. [doi: 10.1109/SP.2017.49]

[17] Croce F, Hein M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In: Proc. of the 37th Int'l Conf. on Machine Learning. New York: JMLR.org, 2020. 2206–2216. [doi: 10.5555/3524938.3525144]

[18] Croce F, Hein M. Minimally distorted adversarial examples with a fast adaptive boundary attack. In: Proc. of the 37th Int'l Conf. on

Machine Learning. New York: JMLR.org, 2020. 2196–2205. [doi: 10.5555/3524938.3525143]
[19] Andriushchenko M, Croce F, Flammarion N, Hein M. Square attack: A query-efficient black-box adversarial attack via random search. In: Proc. of the 16th European Conf. on Computer Vision. Glasgow: Springer, 2020. 484–501. [doi: 10.1007/978-3-030-58592-1_29]
[20] Wu DX, Xia ST, Wang YS. Adversarial weight perturbation helps robust generalization. In: Proc. of the 34th Conf. on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2020. 2958–2969. [doi: 10.5555/3495724.3495973]
[21] Qian YG, Liang XY, Kang M, Wang B, Gu ZQ, Wang X, Wu CM. GAAT: Group adaptive adversarial training to improve the trade-off between robustness and accuracy. Int'l Journal of Pattern Recognition and Artificial Intelligence, 2022, 36(13): 2251015. [doi: 10.1142/S0218001422510156]
[22] Wang Y, Zhang DM, Zhang HY. Weighted adaptive perturbations adversarial training for improving robustness. In: Proc. of the 19th Pacific Rim Int'l Conf. on Artificial Intelligence. Shanghai: Springer, 2022. 402–415. [doi: 10.1007/978-3-031-20865-2_30]
[23] Yu CJ, Zhou DW, Shen L, Yu J, Han B, Gong MM, Wang NN, Liu TL. Strength-adaptive adversarial training. arXiv:2210.01288, 2022.
[24] Xiao JC, Qin ZY, Fan YB, Wu BY, Wang J, Luo ZQ. Adaptive smoothness-weighted adversarial training for multiple perturbations with its stability analysis. arXiv:2210.00557, 2022.
[25] Wang R, Li YX, Liu SJ. Robust mode connectivity-oriented adversarial defense: Enhancing neural network robustness against diversified l_p attacks. arXiv:2303.10225, 2023.
[26] Huang ZC, Fan YB, Liu C, Zhang WZ, Zhang Y, Salzmann M, Süsstrunk S, Wang J. Fast adversarial training with adaptive step size. IEEE Trans. on Image Processing, 2023, 32: 6102–6114. [doi: 10.1109/TIP.2023.3326398]
[27] Ma LH, Liang L. Improving adversarial robustness of deep neural networks via adaptive margin evolution. Neurocomputing, 2023, 551: 126524. [doi: 10.1016/j.neucom.2023.126524]
[28] Guo RY, Chen QY, Liu H, Wang WQ. Adversarial robustness enhancement for deep learning-based soft sensors: An adversarial training strategy using historical gradients and domain adaptation. Sensors, 2024, 24(12): 3909. [doi: 10.3390/s24123909]
[29] Hardy I, Yetukuri J, Liu Y. Adaptive adversarial training does not increase recourse costs. In: Proc. of the 2023 AAAI/ACM Conf. on AI, Ethics, and Society. Montréal: ACM, 2023. 432–442. [doi: 10.1145/3600211.3604704]
[30] Su JW, Vargas DV, Sakurai K. One pixel attack for fooling deep neural networks. IEEE Trans. on Evolutionary Computation, 2019, 23(5): 828–841. [doi: 10.1109/TEVC.2019.2890858]
[31] Bai ZX, Wang HJ. Adversarial example generation method based on improved genetic algorithm. Computer Engineering, 2023, 49(5): 139–149 (in Chinese with English abstract). [doi: 10.19678/j.issn.1000-3428.0065260]
[32] Williams PN, Li K. Black-box sparse adversarial attack via multi-objective optimisation. In: Proc. of the 2023 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Vancouver: IEEE, 2023. 12291–12301. [doi: 10.1109/CVPR52729.2023.01183]
[33] Jia XJ, Zhang Y, Wu BY, Ma K, Wang J, Cao XC. LAS-AT: Adversarial training with learnable attack strategy. In: Proc. of the 2022 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. New Orleans: IEEE, 2022. 13388–13398. [doi: 10.1109/CVPR52688.2022.01304]
[34] Zhang JF, Xu XL, Han B, Niu G, Cui LZ, Sugiyama M, Kankanhalli M. Attacks which do not kill training make adversarial learning stronger. In: Proc. of the 37th Int'l Conf. on Machine Learning. New York: JMLR.org, 2020. 11278–11287. [doi: 10.5555/3524938.3525984]
[35] Zhang JF, Zhu JN, Niu G, Han B, Sugiyama M, Kankanhalli M. Geometry-aware instance-reweighted adversarial training. arXiv: 2010.01736, 2020.
[36] Wang QZ, Liu F, Han B, Liu TL, Gong C, Niu G, Zhou MY, Sugiyama M. Probabilistic margins for instance reweighting in adversarial training. In: Proc. of the 35th Conf. on Neural Information Proc. Systems. 2021. 23258–23269. [doi: 10.48550/arXiv.2106.07904]
[37] Zhu KJ, Hu XX, Wang JD, Xie X, Yang G. Improving generalization of adversarial training via robust critical fine-tuning. In: Proc. of the 2023 IEEE/CVF Int'l Conf. on Computer Vision. Paris: IEEE, 2023. 4401–4411. [doi: 10.1109/ICCV51070.2023.00408]
[38] van der Maaten L, Hinton G. Visualizing data using t-SNE. Journal of Machine Learning Research, 2008, 9(86): 2579–2605.

附中文参考文献

[31] 白祉旭, 王衡军. 基于改进遗传算法的对抗样本生成方法. 计算机工程, 2023, 49(5): 139–149. [doi: 10.19678/j.issn.1000-3428.0065260]

作者简介

翟浩杰, 硕士生, 主要研究领域为深度学习.
王冉, 博士, 副教授, 主要研究领域为机器学习, 深度学习, 不确定性建模.
邬文慧, 博士, 副教授, 主要研究领域为机器学习, 图像增强, 网络数据聚类.
贾育衡, 博士, 副教授, CCF 高级会员, 主要研究领域为机器学习, 数据表示, 半监督学习.