高技术通讯 2023 年 第 33 卷 第 4 期:339-351

doi:10.3772/j.issn.1002-0470.2023.04.001

基于 DSSIM 非范数约束增强的对抗训练方法①

王保利②*** 范鑫鑫** 景全亮*** 毕经平③**

(*中国科学院大学计算机科学与技术学院 北京 100049)

(**中国科学院计算技术研究所 北京 100190)

摘 要 针对当前对抗训练(AT)中存在的鲁棒过拟合问题,即在对抗训练超过一定轮次后,网络模型对抗防御能力出现不升反降的现象,本文提出了一种基于结构相异性非范数约束增强的对抗训练方法(DSSIM-AT)。该方法将非范数约束引入到对抗训练过程中用于对抗样本生成,根据样本间的结构相异度剔除对抗样本中的无语义特征,使得生成的对抗样本更适合于对抗训练。该方法进一步设计了梯度异步更新机制,优化对抗样本生成与模型参数更新耗时问题。实验结果表明,该方法可有效缓解对抗训练鲁棒过拟合情况,相比于已有对抗训练方法,可以将 CIFAR-10 数据集上的干净样本识别准确率提高约 3%,同时对抗样本识别准确率提高约 4% ~8%。

关键词 对抗攻击;对抗防御;对抗训练(AT);非范数约束

0 引 言

深度学习(deep learning,DL)作为当前跨学科人工智能复兴的核心,在各种计算机辅助任务中均展现出强大的优势,如自动驾驶、安全监控、人脸/物体识别、恶意软件分析和检测等应用场景[1]。对于涉及金融、安全、隐私等场景的应用,深度神经网络模型的鲁棒性显得尤为重要。近些年的诸多研究表明深度神经网络模型面临多种多样的安全威胁,由此吸引了许多对深度神经网络模型鲁棒性相关的研究工作[2-3]。

深度神经网络模型在遭受恶意攻击时,性能急剧下降,各类恶意攻击中,表现最为突出的一种攻击方式被称作对抗攻击(adversarial attack)[4]。攻击者通过在图像中加入人为构造的特定细微扰动,能够在人类视觉无法觉察的情况下,使得模型出现错误预测结果,这种攻击生成的样本被称为对抗样本(adversarial example)。为提升深度神经网络模型应对此类对抗攻击的能力,一系列对抗防御方法相继提出,包括图像降噪[5]、图像压缩[6]、模型蒸馏[7]、梯度随机化[8]、图像生成[9]、对抗训练[4,10]等方法。近年来,一些新提出的更强大的对抗攻击方法[11-13],通过梯度近似的手段,使得大量已有的对抗防御方法失效,唯独投影梯度下降对抗训练(projection gradient descent adversarial training, PGD-AT)[14]依旧能够展现出一定程度的防御能力,自此之后对抗训练方法成为了最为有效的经验型对抗防御策略。

然而,对抗训练并非已经彻底地解决了对抗防御问题。首先,深度神经网络模型经过对抗训练后,虽然一定程度上提升了自身对抗防御能力,但其在对抗样本数据集上的识别准确率,依旧远低于其在干净样本数据集上的识别准确率;其次,对抗训练自身还额外地引入了一些问题,如鲁棒过拟合(robust overfitting)、对抗训练耗时(time consuming)、方法原

① 国家自然科学基金(62077044, 61702470, 62002343)资助项目。

② 男,1989 年生,博士生;研究方向:AI 安全,对抗攻防;E-mail:wangbaoli@ict.ac.cn。

③ 通信作者,E-mail:bjp@ict.ac.cn。

(收稿日期:2021-12-17)

理可解释性(interpretability)等[15]。本文聚焦于对抗训练的鲁棒过拟合问题,该问题指的是:随着训练轮次的迭代增加,在干净样本数据集上,网络模型在测试集上与训练集上的损失值呈现同步减小的趋势;然而在对抗样本数据集上,网络模型在测试集上与训练集上的损失值却并未表现一致,训练集上损失值虽然持续减小,但测试集上却出现了损失值先减小再上升的现象。这种在训练超过某个时刻后出现的训练集鲁棒损失较小而对应测试集鲁棒损失增加的现象,被称作对抗训练过程中的鲁棒过拟合[16]。

当前针对如何缓解对抗训练中鲁棒过拟合问题的研究工作主要可分为两大类:一类工作从模型训练损失函数的角度出发,如对抗 logit 配对(adversarial logit pairing,ALP)[17]、结合三元组损失的对抗训练(adversarial training with triplet loss,AT^2L)[18]、三元组损失对抗训练(triplet loss adversarial training,TLA)[19]等,这些方法主要借助度量函数来约束样本特征距离;另一类工作从对抗训练使用的对抗样本生成角度出发,如联邦对抗训练(federated adversarial training,FAT)[20]、定制化对抗训练(customized adversarial training,CAT)[21]、早停-验证(early stop-validation, ES-VAL)[16]等,这些方法主要是基于提早停止(early stop,ES)策略,调整控制对抗样本强度,以达到提升对抗训练效果的目的。第 1 类方法虽然也能在一定程度上提升网络模型在对抗样本上的识别准确率,但往往会牺牲网络模型在干净样本上的识别准确率;第 2 类方法在控制对抗训练过程中对抗样本强度时,严重依赖对抗样本迭代生成时的步长超参选择,小步长虽然能够更细粒度控制样本强度,但会大幅增加迭代轮次,从而引入巨大的时间开销;大步长虽然能够快速生成对抗样本,但难以细粒度控制样本强度。且上述两类方法都是在范数层面优化训练损失函数或约束样本强度,单一使用范数值来表征对抗样本相对原始样本之间的差异有其局限性,生成的对抗样本也更容易产生与原标签类别语义无关的扰动噪声。

针对已有对抗训练方法仅在范数层面约束样本强度的局限性,本文将非范数约束引入对抗训练,通过进一步细粒度地调节控制样本强度,从而缓解对抗训练中出现的过拟合情况,提升网络模型的鲁棒性。

本文的主要贡献总结如下。

(1) 分析并验证已有对抗训练方法研究的缺陷与不足,指出单纯依赖范数约束(norm constraint)无法有效缓解对抗训练过程中产生的鲁棒过拟合问题。

(2) 提出了一种新的基于结构相异性(structural dissimilarity,DSSIM)非范数约束增强对抗训练方法 DSSIM-AT,该方法将非范数约束引入到对抗训练过程中来,根据样本间的结构相异度调整对抗样本强度,剔除无语义特征,并设计了异步模型参数更新机制,从而优化对抗训练流程,增强对抗训练效果,提升网络模型的对抗鲁棒性。

(3) 在广泛使用的 MNIST 与 CIFAR-10 真实图像数据集上进行了大量实验与对抗训练效果评估分析。实验结果表明,相较于当前主流的对抗训练方法,本文方法能够有效缓解鲁棒过拟合问题,提升深度神经网络模型的鲁棒性,包括在干净样本和对抗样本上的识别准确率。

本文剩余部分总结如下。第 1 节介绍了与神经网络模型鲁棒性相关的研究进展,包括对抗攻击与对抗训练方法,范数与非范数约束。第 2 节先介绍了对抗训练的定义和相关概念,然后详细介绍了本文提出的 DSSIM-AT 方法,并对算法运行机理、时间与空间复杂度进行了分析讨论。第 3 节通过实验对本文所提对抗训练方法进行了有效的验证,并分析实验结果。最后,第 4 节对本文的研究工作进行了总结并展望该技术的未来发展方向与前景。

1 相关工作

本节介绍当前与对抗训练相关的研究工作进展以及范数/非范数约束。

1.1 对抗样本

对抗样本的概念在 2014 年由文献[4]提出。对抗样本生成是对抗训练的重要环节与步骤,根据所使用的技术原理分为 3 类,其一为基于梯度的对

抗样本生成方法,如快速梯度符号法(fast gradient sign method, FGSM)[10]、随机快速梯度符号法(random fast gradient sign method,RFGSM)[22]、基础迭代法(basic iteration method,BIM)[23]和 PGD[14]等,该类方法主要借助梯度符号化、多步迭代、随机启动等技术来生成对抗样本;其二为基于优化的对抗攻击方法,如 L-BFGS (limited-memory Broyden-Fletcher-Goldfarb-Shanno)[4]和 CW(Carlini-Wagner)[11]等,该类方法主要是将对抗样本作为优化的参数,利用不同损失函数设计,更新迭代获得对抗样本;其三为基于对抗生成网络(generative adversarial network,GAN)的对抗样本生成方法,如 AdvGAN[24]、AdvGAN ++[25]和 AdvCGAN[26]等,该类方法主要借助 GAN 的生成能力构造对抗扰动。上述方法中,PGD 对抗样本是目前对抗训练普遍采用的训练样本。

1.2 对抗训练

对抗训练发展初期,文献[4]指出可将对抗样本与干净样本混合一起作为训练样本用于模型训练以提升模型的鲁棒性。文献[10]提出将通过单步迭代生成的 FGSM 对抗样本用于对抗训练,对应网络模型能应对单步对抗样本攻击,但依然无法防御基于多步迭代生成的 BIM 对抗样本的攻击,而且该方法会导致模型过拟合于对抗样本。文献[23]提出使用多步迭代生成的 BIM 对抗样本进行对抗训练。文献[14]提出对抗训练优化范式 PGD-AT,并使用 PGD 对抗样本作为训练样本进行对抗训练,该方法显著提升了模型鲁棒性,其定义的优化范式成为对抗训练发展的里程碑,后续多数对抗训练研究工作均是对该优化范式下进行的优化与改进。

ALP[17]在 PGD-AT 原训练损失函数的基础上,额外引入了评价损失(logit loss)作为度量来约束原始干净样本和对抗样本的特征距离。AT^2L[18]、TLA[19]则将度量学习(matrix learning)中的三元组损失(triplet loss)加入到对抗训练损失函数中,用以约束样本特征之间的距离。FAT[20]在 PGD 对抗样本生成过程中会额外地加入对当前样本分类预测标签的判断操作,对于仍未使模型出现错误预测的样本沿用 PGD-AT 中解决内部最大化优化的策略,而对已经能够使模型错误分类的样本,则将损失函数替换为最小化优化策略。CAT[16]在使用 PGD 对抗样本做对抗训练过程中,限制 PGD 对抗样本的迭代次数,在训练开始使用较小的迭代次数,而随着迭代轮次的增加逐渐增多对抗样本的迭代次数,以达到控制对抗样本强度的目的来缓解对抗训练过程中的过拟合。ES-VAL[16]本质上并未对 PGD-AT 对抗训练逻辑本身做相应调整或优化,而是预先从训练集中额外采样出一些样本与标签数据作为验证集,在对抗训练过程中,用模型在验证集上的损失作为评估标准来确定最佳的模型参数。

1.3 范数与非范数约束

范数约束是一种常见的缓解过拟合的技术手段。具体到图像视觉领域,范数主要指空间范数中的 p -范数,常用的 p -范数包括 4 种:0 范数、1 范数、2 范数和无穷范数。其中 1 范数也被称为绝对值范数,指向量元素(图像像素值或特征值)的绝对值之和,对应于平面上一个正方形正则区域;2 范数也被称为欧几里得范数,指图像特征向量的模长,对应于平面上一个圆形的正则区域;无穷范数指图像所有像素统计的绝对值的最大值。现有对抗训练方法所使用的对抗样本均是在 p- 范数约束下构建生成。

与范数约束对应的,非范数约束(non-norm constraint)也是图像领域重要的评价指标,最常用的非范数主要指结构相似性(structural similarity, SSIM)[27],该指标主要用于衡量 2 幅图像相似度,由德州大学奥斯丁分校的图像和视频工程实验室提出。结构相似度指数从图像组成的角度将结构信息定义为独立于亮度、对比度的反映场景中物体结构的属性,并将失真建模为亮度、对比度和结构 3 个不同因素的组合。用均值作为亮度的估计,标准差作为对比度的估计,协方差作为结构相似程度的度量。本文展示了借助非范数约束能够更好地构造更具“语义性”的对抗样本,在引入了非范数约束后,生成的对抗样本剔除了更多无关噪声,保留了跟原样本对应标签更一致的对抗噪声。

2 基于非范数约束增强的对抗训练

本节首先介绍对抗训练问题定义和相关概念,

然后详细介绍如何使用非范数约束增强来构造用于对抗训练的对抗样本,以及构造完成后如何用于模型参数更新。最后对算法进行运行机理及复杂度分析。

2.1　对抗训练问题定义与相关概念

图 1 展示了对抗训练的整体架构与训练流程。对抗训练核心包含 2 个阶段:内部最大化优化阶段和外部最小化优化阶段。在内部最大化优化阶段主要任务是利用初始网络模型和训练集数据,采用一定的方法生成对抗样本;在外部最小化优化阶段主要任务是利用生成的对抗样本,用于训练网络模型,更新模型参数。经过上述 2 个阶段,最终使得对抗训练后的网络模型具备更好地抵御对抗攻击的能力,即模型对抗鲁棒性。为了更好地描述对抗训练算法,与对抗训练相关的基本概念总结如下。

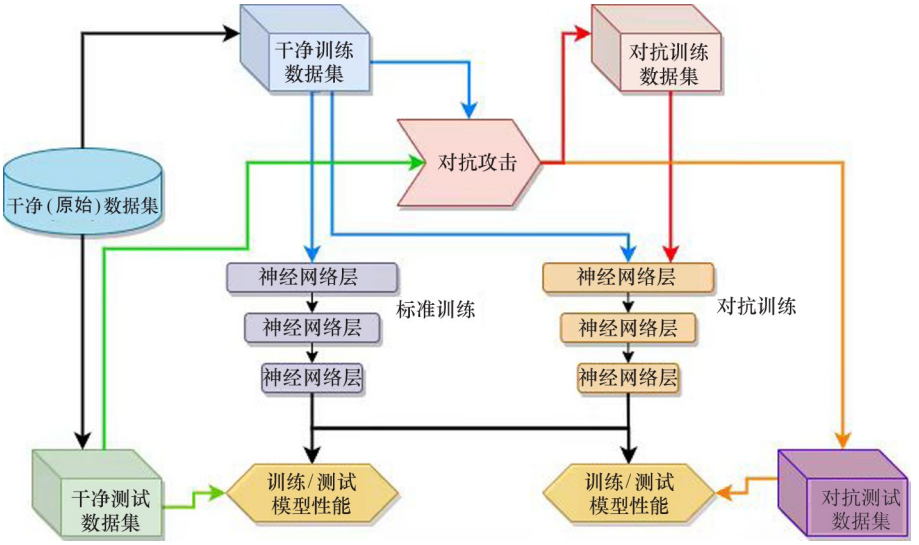

图 1　对抗训练整体流程示意图

对抗样本(adversarial example)[4,10]　指在图片中通过加入特定构造的肉眼不可见的细微扰动所形成的样本,这类样本会导致训练好的网络模型以高置信度给出与原样本不同的分类输出。与之对应的未被扰动的样本被称为干净/原始样本。

对抗攻击(adversarial attack)[4,10]　对抗样本生成的重要手段,即对抗样本中被加入的轻微扰动。一般来说,对抗扰动具备 2 个特性:欺骗性与隐蔽性。对抗攻击根据攻击者是否掌握模型、训练集等信息可细分为白盒对抗攻击与黑盒对抗攻击。

对抗鲁棒性(adversarial robustness)[16]　指网络模型抵御对抗攻击的能力,可以用模型的图像分类准确率来衡量。

干净数据集(clean dataset)　指未经过对抗攻击的原始训练/测试样本构成的数据集,通常也称原始数据集。

对抗数据集(adversarial dataset)　指基于原始训练/测试数据集样本,使用一定的对抗攻击方法生成对抗样本构成的对抗训练/测试数据集,该数据集可用于对抗训练或模型鲁棒性测试。

对抗训练根据其优化步骤和目的可被定义为一个 min－max 形式的“鞍点优化”问题[14],其公式化可表述为

$$\min_{\theta}\sum_{i}\max_{\delta\in\Delta} l(f_{\theta}(x_i+\delta),\ y_i) \tag{1}$$

其中,x_i 代表原始干净样本,δ 表示对抗扰动(或对抗噪声),y_i 代表 x_i 对应的标签信息,l 表示目标网络模型训练时采用的损失函数(如常用的交叉熵损失函数)。Δ 用来约束解决内部最大化优化问题时的扰动强度,通常限制样本强度在 p -范数约束内。

2.2　基于 DSSIM 非范数约束的对抗样本生成

图 2 展示了对抗训练过程中,针对第一步样本生成环节,如何引入非范数约束来更好地构造用于对抗训练的对抗样本。图中实线展示了基于范数约束构造样本,虚线展示了基于非范数约束增强的对

抗样本构造过程。x_0 对应矩形为训练集原始样本,y 对应矩形为训练集原始标签信息,$x_1 \sim x_n$ 对应矩形为每轮迭代后对应的对抗样本,梯形框为最终需要训练的目标网络模型。首先,原始样本 x_0 与其对应标签 y 会一同送入目标网络模型并计算交叉熵损失,沿着使交叉熵损失增大的方向可以获得梯度 $Grad_{ce}$;将其符号化后乘以固定的步长可以获得初步的扰动,将扰动叠加在样本 x_0 上即得到初步的样本 x'_1;此时对 x'_1 与 x 计算两者的结构相异度损失,并获取对应的梯度值 $Grad_{\text{dssim}}$,即进一步更为细粒度的扰动,将其叠加在样本 x'_1 上,从而获得本轮迭代后的样本 x_1。而后根据样本迭代次数反复进行上述操作,交替使用交叉熵与结构相异度损失来优化对抗样本。

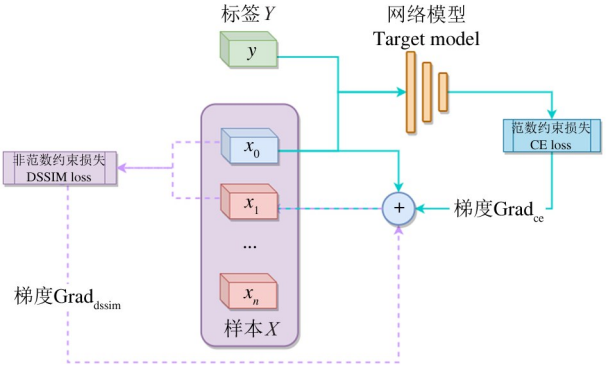

图 2　基于非范数约束的对抗样本生成示意图

SSIM 损失函数　给定 2 个图像,根据结构相似性定义,其 SSIM 损失函数公式化表述如下:

$$SSIM(\boldsymbol{x}, \boldsymbol{y}) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{2}$$

其中,x、y 为输入图像样本,μ_x 和 μ_y 分别为 x 和 y 的均值,σ_x 和 σ_y 分别为 x 和 y 的标准差,σ_{xy} 为 x 和 y 的共变异数,C_1 和 C_2 为常数。

DSSIM 损失函数　鉴于深度神经网络的训练目标为损失值最小化,而 SSIM 函数自身为值域在 $[-1,1]$ 的单调增函数,即相似度随 SSIM 值增大而增大,无法很好地适应深度神经网络标准训练目标。本文对 SSIM 做了额外的改进处理,使其值域由 $[-1,1]$ 转换为 $[0,1]$ 的同时由单调增函数变为单调减函数,以更好地适用于深度神经网络模型的训练与参数更新,改进后的函数称为结构相异性(DSSIM)损失函数,其公式化表述如下:

$$DSSIM = 0.5 - \frac{SSIM(x_1, x_2)}{2} \tag{3}$$

基于二阶段损失函数的对抗样本生成　在引入 DSSIM 非范数约束后,原本的 PGD 对抗样本迭代生成过程中 $t+1$ 轮对抗样本 x^{t+1} 的计算生成,除了对上一轮对抗样本 x^t 和当前交叉熵损失 l_{ce} 的梯度依赖,还额外地引入了第三项依赖,即对抗样本 x' 和对应原始样本 x 的结构相异度依赖用以进一步微调样本强度。整体流程可看作两步,首先计算当前样本与标签的交叉熵损失对于样本的梯度值并更新样本;然后计算当前样本与原始样本的相异度损失,更新调整样本得到新的对抗样本。每次更新样本时也均会做一次投影操作以防止像素值越界,其损失函数公式化表示为

$$x' = \prod_{x+S}(x^t + \alpha \cdot \text{sign}(\nabla_{x^t} l_{ce}(\theta, x^t, y))) \tag{4}$$

$$x^{t+1} = \prod_{x+S}(x' + \beta \cdot (\nabla_{x'} l_{\text{dssim}}(x, x'))) \tag{5}$$

其中,$\nabla_{x^t} l_{ce}$ 表示通过计算样本 x^t 与其对应标签 y 的交叉熵损失值对于样本 x^t 的梯度信息,$\nabla_{x'} l_{\text{dssim}}$ 则表示当前对抗样本 x' 与其对应的原始样本 x 计算出的 DSSIM 值对于样本 x' 的梯度信息,sign 为符号化处理函数,α 为交叉熵损失对应的梯度的权重,β 为 DSSIM 相似度损失对应的梯度权重。

基于联合损失函数的对抗样本生成　为了进一步优化算法时间开销及提升算法超参调控的灵活性,本文提出了基于联合损失函数的对抗样本生成方法。其核心思路是将式(5)中的参数 x' 直接替换为该轮迭代的起始样本 x^t,即可将式(4)与式(5)进行合并优化,联合损失函数公式化表示为式(6)。

$$x^{t+1} = \prod_{x+S}(x^t + \gamma \cdot \Delta + (1-\gamma) \cdot \Delta) \tag{6}$$

其中,Δ 为交叉熵与 DSSIM 损失值之和符号化后与更新步长乘积对应的结果,公式化表示如式(7)所示。

$$\Delta = \delta \cdot \text{sign}(\nabla_{x^t}(l_{ce}(\theta, x^t, y) + l_{\text{dssim}}(x, x^t))) \tag{7}$$

其中,δ 表示单步迭代步长,sign 为符号化处理函

数，θ 为模型参数，l 为损失函数。

2.3　基于 DSSIM 生成样本的模型参数更新

在基于 DSSIM 非范数约束的对抗样本生成之后，即可用生成的样本数据更新网络模型参数。如图 3 所示，借助计算图共享的机制，根据对抗样本生成和模型参数更新的频次关系，本文设计了同步更新和异步更新 2 种模型参数更新方式。

（1）同步参数更新方式。即在对抗训练过程中的每一轮中，均进行样本生成以及模型参数更新，首先根据模型和原始训练数据（或上一轮生成数据）生成针对当前模型的对抗样本，然后通过当前对抗样本及标签数据，更新网络模型参数。在这种更新方式下，对抗样本生成与模型参数更新共享同一张计算图，而每张计算图也仅用于一批次的对抗样本生成以及一轮次的模型参数更新。根据生成的对抗样本计算得到的损失并更新网络模型的参数可公式化表示为

$$\theta_f^{t+1} = \theta_f^t - \nabla_\theta L_{ce}(\boldsymbol{x}_{\text{adv}}, y) \tag{8}$$

其中 $\boldsymbol{x}_{\text{adv}}$ 为 3.2 中生成的对抗样本，$\nabla_\theta(*)$ 表示损失 $*$ 对于模型参数 θ 的梯度信息。

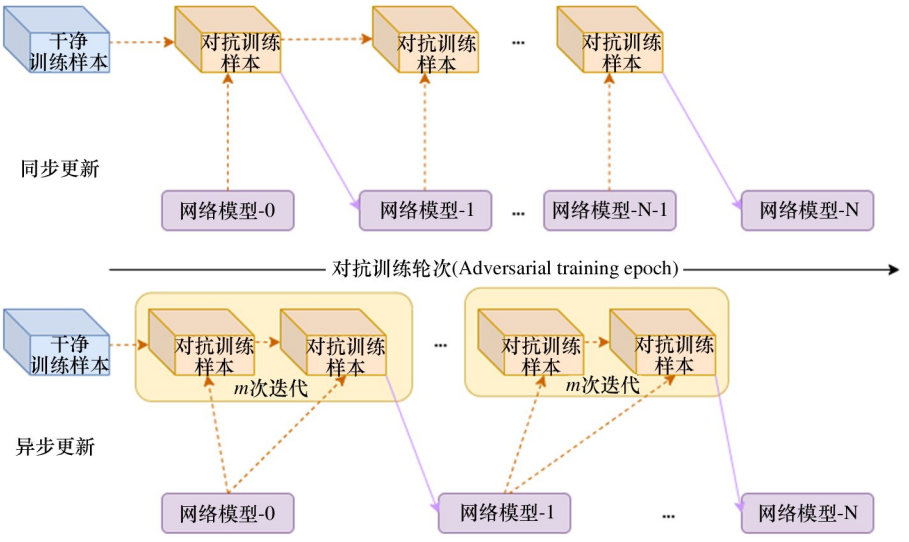

图 3　网络模型参数更新方式示意图

（2）异步参数更新方式。与同步参数更新方式相对应，为了提升对抗训练效率，也可以使用异步参数更新的方式，即在对抗训练的每一轮均生成对抗样本，而网络模型参数在 m 轮对抗训练后进行更新，如此可使得网络模型参数得到多次复用，从而降低模型更新次数以及由此带来的相关时间与空间开销。需要注意的是，对于异步参数更新通常需要更小的学习率的优化器，以防止学习率过大导致网络模型对抗训练时损失在局部最优点附近震荡而难以收敛，更新过程可表示为

$$\theta_f^{t+1} = \theta_f^t - \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta l_{\text{ce}}(x_{\text{adv}}^{(i)}, y^{(i)}) \tag{9}$$

其中 m 为模型参数更新间隔次数，θ 为模型参数。

DSSIM-AT 对抗训练整体算法总结如算法 1 所示，该算法流程详细展示了在使用联合损失函数和异步更新的情况下 DSSIM-AT 对抗训练过程。

算法 1　DSSIM-AT 对抗训练算法

算法输入：

训练集原始样本 $\boldsymbol{X}$：$\{\boldsymbol{x}^{(1)},\cdots,\boldsymbol{x}^{(n)}\}$，共包含 n 个样本实例

训练集原始样本标签 $\boldsymbol{Y}$：$\{\boldsymbol{y}^{(1)},\cdots,\boldsymbol{y}^{(n)}\}$，与 X 中的样本逐个对应

算法输出：

网络模型 f 的参数：θ_f^T

#随机初始化网络模型参数

Randomly initialize network $\boldsymbol{\theta}_f^0$.

#对抗训练，epochs 为训练总轮次 T

For epoch in epochs **Do**：

#从训练集中采样 batch 的原始训练样本与标签

Sample a mini-batch of m examples from training set：

$\boldsymbol{X} = \{x^{(1)},\cdots,x^{(m)}\}$，$\boldsymbol{Y} = \{y^{(1)},\cdots,y^{(m)}\}$

#迭代构造对抗样本,iterations 为迭代轮次

For iter in iterations **Do**:

#根据 3.2 节对抗样本生成式(6),构造对抗样本

Generate a mini-batch of m adversarial examples with CrossEntropy and DSSIM Loss:

$$\boldsymbol{X}_{\text{adv}} = \{x_{\text{adv}}^{(0)}, \cdots, x_{\text{adv}}^{(m)}\}$$

End For

使用梯度下降的方法更新目标网络模型参数

Update the target network's parameters f_θ^t by the stochastic gradient of f_θ^{t-1}:

$$\nabla_\theta f = \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta l_{\text{ce}}(x_{\text{adv}}^{(i)}, y^{(i)})$$

$$\theta_f^t = \theta_f^{t-1} - \nabla_\theta f$$

End For

2.4 对抗训练算法机理分析

为了更进一步地阐明本文所提 DSSIM-AT 对抗训练方法的运作机理,可以从特征降噪(feature denoising)及表征一致性(consistency representation)的角度进行类比分析 DSSIM 在对抗样本生成以及对抗训练过程中所起到的作用。

图 4 和图 5 分别展示了对于对抗样本使用 DSSIM 和降噪函数处理后的注意力示意图。从图中可以看到,相比于附带较多“噪声”的对抗样本,经过 DSSIM 约束处理后的对抗样本,成功降低了噪声的数量与干扰程度。这也使得网络模型更容易学习到真正有用的图像特征,有效减少因对抗攻击产生的无语义激活点而导致模型出现错误分类预测的概率。此外,董胤蓬等人[28]观察到语义概念与神经元学习到的特征存在不一致性现象,并认为加入特征表示一致性损失可提升对抗训练可解释性,DSSIM 约束则恰好起到了加强表征一致性的作用。因此本文将非范数约束引入到对抗训练的对抗样本生成过程中,从更多维度上优化样本,使其更适合于对抗训练,从而缓解对抗训练的过拟合情况,提升网络模型的鲁棒性。

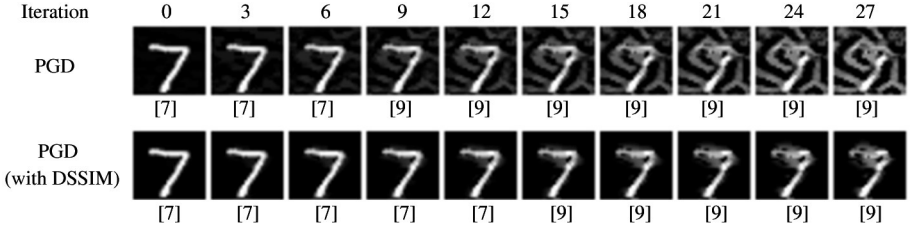

图 4 对抗攻击中范数-非范数约束对比示意图

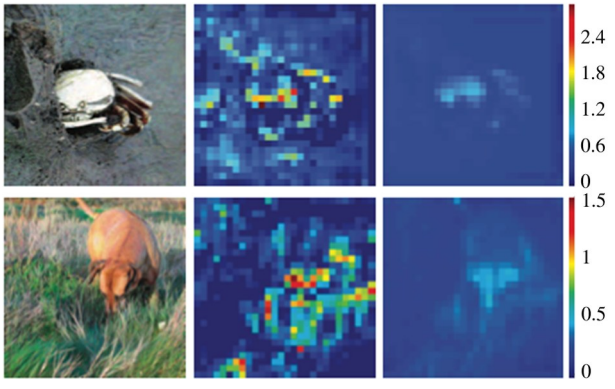

图 5 神经网络中特征降噪效果示意图[5]

2.5 算法复杂度分析

DSSIM-AT 对抗训练方法是否能够进行实际应用的关键在于算法的时间复杂度,在此对该算法的时间及空间复杂度进行详细分析。

空间复杂度方面,在对抗训练初始化阶段,针对每个批次的 K 个训练数据,对抗训练主要的空间开销在于存储模型结构参数 M、批次包含的样本 X 及对应标签 Y,对应的空间复杂度 $S(DSSIM-AT)_{\text{init}}$ 为 $O(|M|+|X|+|Y|)$,其中 $|*|$ 表示 $*$ 所占用的存储空间。在对抗训练的对抗样本生成及模型参数更新阶段,原始样本 X 通过对抗攻击算法变成了对抗样本 X',标准对抗训练方法的空间开销变为 $O(|M|+|X'|+|Y|)$,与初始化阶段保持一致,而本文所提的 DSSIM-AT 对抗训练算法需要额外空间用于存储原始样本,其空间开销变为 $O(|M|+|X|+|X'|+|Y|)$。可以看出,DSSIM-AT 虽然需要额外空间存储记录原始样本,但其空间复杂度仍保持线性级别 $O(N)$,其中 N 为模型、原始样本、对抗样本、

样本标签的空间存储占用总量。

时间复杂度方面，在对抗训练过程中，主要的时间开销包括内部最大化优化时生成对抗样本以及外部最小化优化时模型参数更新 2 个阶段，2 个阶段均又包括前向计算损失以及反向传播计算梯度信息。由于共享相同的计算图，总传播时间为单次传播时间的 2 倍。针对每个批次的训练数据，在对抗样本生成阶段，前向传播与反向传播时间复杂度均为 $O(|\varnothing_{\text{backbone}}|+|\varnothing_{\text{cls}}|+|\varnothing_{\text{dssim}}|)$，这里 $|\varnothing_{\text{backbone}}|$、$|\varnothing_{\text{cls}}|$ 和 $|\varnothing_{\text{dssim}}|$ 分别指神经网络模型特征抽取、范数约束损失以及非范数约束损失对应的时间开销。其中 $|\varnothing_{\text{backbone}}|$ 主要取决于神经网络模型的规模与结构设计，其开销远大于约束损失计算所占用的时间，$|\varnothing_{\text{cls}}|$ 主要取决于输入数据的维度大小，时间开销为 $O(d_{\text{feat}}^2)$，其中 d_{feat} 为单个样本特征的维度大小与批次内样本数量的乘积。对于本文所使用的 10 分类图像数据集，单个样本特征的维度大小为神经网络模型的 *logit* 输出向量。$|\varnothing_{\text{dssim}}|$ 主要取决于输入样本维度大小，时间开销为 $O(d_{\text{img}}^2)$，其中 d_{img} 为输入图像大小。此外对于当使用二阶段损失函数生成对抗样本时，范数与非范数损失需顺次串行计算，时间开销约为标准对抗训练中约束损失计算时间的 2 倍；使用联合损失函数生成对抗样本时，约束损失计算时间与标准对抗训练时间基本保持一致。在模型参数更新阶段，仅需要使用分类损失计算模型参数梯度，前向传播与反向传播时间复杂度均为 $O(|\varnothing_{\text{backbone}}|+|\varnothing_{\text{cls}}|)$。可以看出，相对于标准对抗训练，DSSIM-AT 在时间开销上仅在样本生成阶段增加了用于计算范数约束的 $|\varnothing_{\text{dssim}}|$ 部分，并未对对抗训练整体流程构成负担，这也说明该对抗训练方法可应用于复杂模型的对抗训练中。

3　实验评估与结果分析

本节对 DSSIM-AT 对抗训练方法进行全面对比实验与结果分析。首先介绍了实验所使用的数据集、网络模型及超参数设置，然后介绍深度神经网络模型鲁棒性测试所采用的对抗攻击方法，以及实验对比所采用的对抗训练基准方法与对比方法，并对实验结果进行分析，验证本文所提出的对抗训练方法的有效性，最后从算法运行时间及模型占用空间复杂度方面对比分析了 DSSIM-AT 算法的运行效率。

3.1　数据集

当前，针对图像分类任务，用于评估深度神经网络模型鲁棒性的数据集主要有 2 个：MNIST 手写体识别数据集和 CIFAR-10 图像分类数据集，2 个数据集的图片样例如图 6 所示。下面从数据集大小、类别数和数据特点等方面分别介绍这 2 个数据集。

MNIST 数据集共包含 70 000 张手写体数字图片，其中 60 000 张图片用于训练，10 000 张图片用于测试。其共包含 10 个类别，分别为数字 0 ~ 9。每张图片均为单通道的灰度图，分辨率大小为 28 × 28。

CIFAR-10 数据集共包含 60 000 张图片，其中 50 000 张图片用于训练，10 000 张图片用于测试。其共包含 10 个类别。每张图片均为红绿蓝（red green blue，RGB）3 通道的彩色图，分辨率大小为 32 ×32。

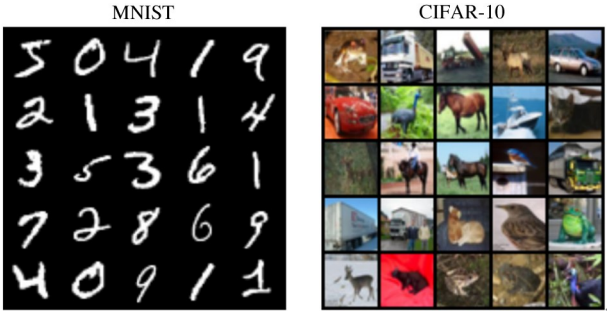

图 6　数据集样例图

3.2　网络模型

对于 MNIST 数据集的实验采用 LeNet 作为目标网络模型，对于 CIFAR-10 数据集的实验采用 ResNet18 作为目标网络模型。

3.3　参数设置

本文所有实验均采用 2.3 节中算法 1 的对抗训练策略，软件环境为 PyTorch v1.8.0 编程实现，硬件环境为 128 GB 内存、24 核 Intel CPU 和 4 张 2080Ti GPU 显卡。针对 MNIST 数据集，对抗训练总轮次设置为 40 轮，学习率为恒定值 0.1，批大小（batch

size)在训练阶段设置为 512,测试阶段为 1024,对抗扰动阈值为无穷范数下 0-1 归一化后的 0.3,PGD 对抗样本迭代轮次为 40 轮,每轮均包含随机重启;针对 CIFAR-10 数据集,对抗训练总轮次为 110 轮,学习率初始值为 0.1,分别在 100 轮和 105 轮调整为 0.01 和 0.001,训练时的批大小设置为 512,测试时的批大小为 1024,对抗扰动阈值统一设置为无穷范数下 0-1 归一化后的 0.03,PGD 对抗样本迭代轮次为 10 轮,每轮次均包含随机重启。

3.4 对抗攻击方法

本文实验选取了多种对抗攻击方法来全面评估各对抗训练方法的训练效果与质量。这些方法包括以下几种。

FGSM[10] 是较早提出的对抗攻击方法之一,FGSM 对抗攻击方法采用单步迭代攻击策略,能够非常快速地构造对抗样本。

BIM[23] 是基于 FGSM 对抗攻击方法的改进,核心思想是采用多步迭代的方式,每次迭代使用更小的步长(或扰动),从而提升对抗攻击效果。

结合交叉熵损失的投影梯度下降法(projected gradient descent with cross entropy loss,PGD-CE)[14] 是 PGD 对抗攻击方法作为基于梯度的对抗攻击方法代表,采用多步迭代攻击以及随机重启策略,也是目前最强的一阶对抗攻击方法。比 FGSM 方法攻击能力更强,相应地耗时也更长。

结合 CW 损失的投影梯度下降法(projected gradient descent with CW loss,PGD-CW)[13] 是 PGD-CE 对抗攻击方法的变体,主要区别在于将原有的交叉熵 CE 损失函数替换为 CW 损失函数,在一些攻击场景下效果优于 PGD-CE。

CW[11] 对抗攻击方法作为基于优化的对抗攻击方法的代表,其在白盒攻击场景下往往可以达到非常高的攻击成功率,但它的耗时也会随着迭代次数的增加而急剧增加。

结合交叉熵损失的自动投影梯度下降法(auto PGD-CE,APGD-CE)[13] 为一种定向对抗攻击方法,也是自动攻击(auto attack,AA)的标准形式,是对 PGD 对抗攻击的优化,其生成的样本攻击成功率往往高于 PGD 对抗攻击方法。

自动投影梯度下降定向法(auto projected gradient descent targeted,APGD-T)[13] 为一种定向对抗攻击方法,是 AA 的一种变体版本实现。

快速自适应边界定向法(fast adaptive boundary targeted,FAB-T)[29] 为一种定向对抗攻击方法,目的是在攻击成功的同时,尽可能地最小化对抗扰动范数值。

Square Attack[22] 为一种黑盒对抗攻击方法,基于对目标网络模型查询得到的信息构建对抗样本。

3.5 对抗训练对比方法

本文实验使用 PGD-AT 作为基准对抗训练对比方法,同时也横向对比了多个针对优化、缓解对抗训练鲁棒过拟合问题的相关对抗训练方法。这些方法包括以下几种。

PGD-AT[14] 是基准对抗训练方法,使用 PGD 对抗样本作为训练集(不含原始训练集样本)训练模型。

FAT[20] 是对比对抗训练方法之一,使用样本强度控制策略,PGD 对抗样本生成时每次迭代前均额外作一次判断,对已经使模型产生错误分类的样本进行额外处理。

CAT[16] 是对比对抗训练方法之一,使用缓启动策略,在对抗训练前期使用较少的 PGD 对抗样本迭代次数,而在后续慢慢增强 PGD 对抗样本的样本迭代次数。

ES-VAL[16] 整个对抗训练流程与 PGD 一样,但额外引入了验证集用其损失作为评估模型鲁棒性的标准。

DSSIM-AT 是本文提出的对抗训练方法。通过将 DSSIM 非范数约束引入对抗训练过程中用于对抗样本生成,缓解过拟合情况,提升对抗训练效果。

3.6 评价指标

为了方便评估深度神经网络模型的鲁棒性,本文实验采用分类准确率(ratio of classification accuracy,RCA)作为评价指标,来评估网络模型在干净样本和各种对抗样本测试集上的性能表现,RCA 公式化表示如下。

$$RCA = \frac{N_{\text{correct}}}{N_{\text{total}}} \tag{10}$$

其中，N_{total} 表示测试集的样本总数，$N_{correct}$ 表示测试集中能够被网络模型正确分类识别的样本数量。

3.7　实验结果与分析

3.7.1　对抗训练效果整体评估实验

图 7 更为清晰明了地展示了各种对抗训练方法的整体效果对比，其中横坐标轴为各对抗训练算法对应网络模型在 PGD 对抗样本测试集上的识别准确率，纵坐标轴为网络模型在干净样本数据集上识别准确率，不同线型代表不同的对抗方法对应的网络模型，圆点的大小代表了对抗训练最后一轮对应的模型与最佳模型的对抗样本识别准确率之差。

为了更全面地评估比较各对抗训练方法，对于 PGD-AT、FAT、CAT 以及 DSSIM-AT 各对抗训练方法分别取其对抗训练最后10个轮次对应的网络模型在干净样本、对抗样本数据集上的识别准确率并统计计算其均值，对应图中不同的圆点，同时统计训练过程中最后 10 个轮次对应经过归一化后训练对抗损失值的方差作为各点的尺寸大小。另外，ES-VAL 对应的圆点为 PGD-AT 通过将验证集损失作为标准筛选出的网络模型。从图中可以明显看出，从各方法对应的圆点大小来看，相较于 PGD-AT 对抗训练方法，其余各方法均在不同程度上缓解了鲁棒过拟合情况；另外，PGD-AT 基准对抗训练方法在使用了验证集后，即 ES-VAL 方法在干净样本、对抗样本数据集上识别准确率均有所提升；FAT 相较于 ES-VAL 虽然在干净样本上识别准确率获得了提升，但相应地牺牲了其在对抗样本上的识别准确率；CAT 相较于 ES-VAL 在提升对抗样本识别准确率的同时，在干净样本上识别准确率有所下降；而本文所提出的 DSSIM-AT 对抗训练方法相较于 ES-VAL，则在干净样本和对抗样本数据集上的识别准确率均获得提升。

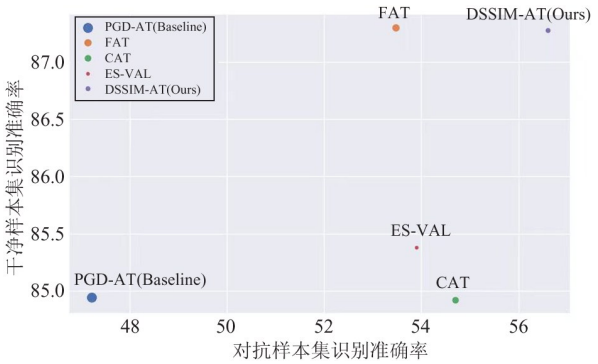

图 7　CIFAR-10 数据集对抗训练效果对比图

3.7.2　对抗训练效果定性评估实验

图 8 展示了整个对抗训练过程中，不同对抗训练方法的损失值变化曲线。从图中可以看出，PGD-AT 对抗训练方法在对抗测试集对抗损失值（三角标记曲线）的整体趋势为先降再升，这也意味着其

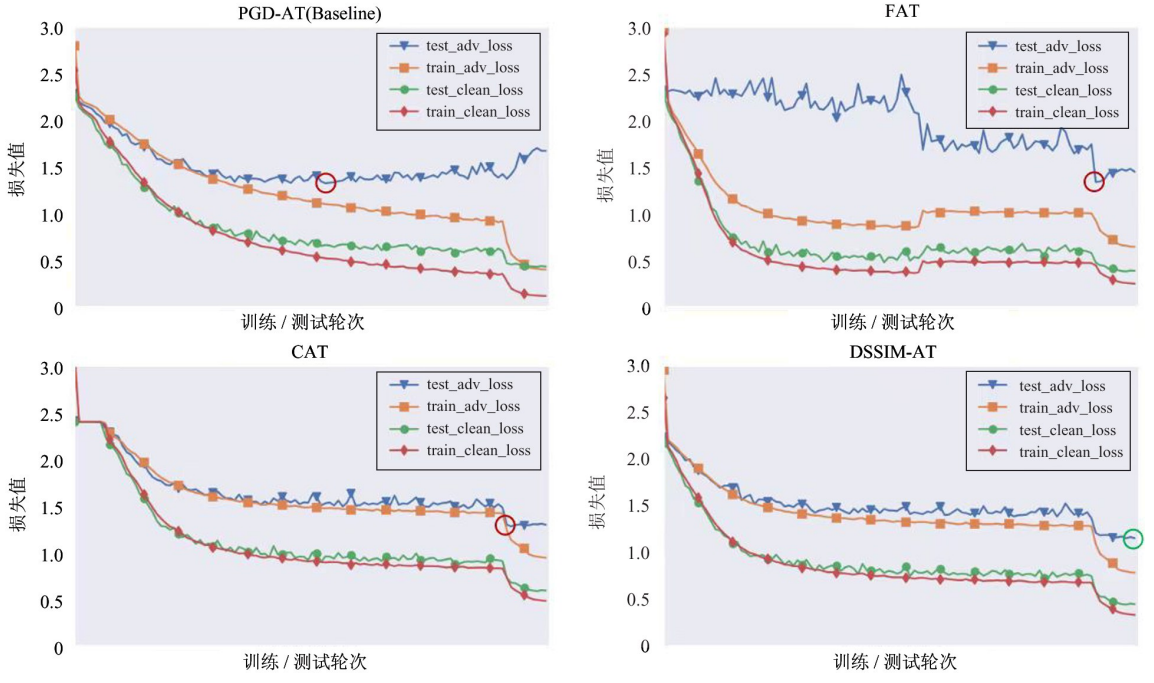

图 8　对抗训练损失值变化对比图

在后期尤其100 次迭代之后陷入了较为严重的过拟合;FAT 对抗训练方法在对抗测试集对抗损失值整体上虽然是逐步下降的趋势,且在干净测试集上损失值(圆点标记曲线)相较于 PGD-AT 更小,但后期依旧存在过拟合的情况;CAT 对抗训练方法在对抗测试集上对抗损失值逐步减小,后期保持平稳,但前期由于样本强度太弱导致损失值前期减小得非常慢;本文所提出的 DSSIM-AT 对抗训练方法则能够在保持较快收敛的同时,使得网络模型在对抗测试集上的损失值持续减小,从而缓解神经网络模型对抗训练中的鲁棒过拟合问题。

3.7.3　对抗训练效果定量评估实验

表 1 的数据结果显示,在 MNIST 数据集上,相较于 PGD-AT 基准对抗训练方法,各对抗训练方法在干净样本和对抗样本测试集上识别准确率均有所提升,其中本文提出的 DSSIM-AT 对抗训练方法在干净样本上识别准确率提升 2%,对对抗样本识别准确率提升 4% ~7%,提升幅度最为显著。这表明 DSSIM-AT 对抗训练方法收敛效率较高。在 CIFAR-10 数据集上,相较于基准 PGD-AT 对抗训练方法,ES-VAL 在干净样本测试集上识别准确率提升约 1% 的同时在对抗样本测试集上识别准确率提升约 1% ~4%;FAT 和 CAT 对抗训练方法相较于 ES-VAL 对抗训练方法,能够在对抗样本和干净样本测试集上识别准确率分别有所提升,但均未能够在 2 个测试集上识别准确率同时提升;本文所提出的 DSSIM-AT 对抗训练方法则能够在干净样本和对抗样本测试集上识别准确率均带来提升,相较于 PGD-AT,在干净样本测试集上识别准确率提升约 3%,同时在对抗样本测试集上识别准确率提升约 4% ~8%。

表 1　不同对抗攻击下各对抗训练方法效果对比测试结果表

对抗训练方法	不同对抗攻击方法下测试集识别准确率/%									
	Clean	FGSM	BIM	PGD-CE	PGD-CW	CW	APGD-CE	APGD-T	FAB-T	Square
MNIST 数据集										
PGD-AT	96.52	86.65	80.52	82.51	82.51	83.20	77.12	75.75	75.64	75.56
FAT	97.39	87.98	83.47	85.42	85.67	83.30	80.57	79.73	79.71	79.60
CAT	97.35	87.91	83.01	84.54	84.63	83.76	78.30	77.09	77.19	76.87
ES-VAL	96.87	88.10	83.94	85.74	85.55	85.37	80.39	79.16	79.17	78.41
DSSIM-AT	98.49	91.28	85.36	88.22	88.94	87.08	83.25	82.35	82.32	82.23
CIFAR-10 数据集										
PGD-AT	84.87	60.56	60.48	47.79	47.06	61.97	48.10	47.17	46.68	46.70
FAT	87.62	61.99	64.03	53.29	50.99	65.57	50.05	48.55	49.51	49.53
CAT	85.25	63.76	65.21	53.85	54.12	67.64	53.87	50.54	50.61	50.43
ES-VAL	85.60	62.84	64.74	53.39	53.87	66.84	53.29	50.39	50.38	50.36
DSSIM-AT	87.53	65.10	66.84	55.55	55.89	69.55	55.00	51.90	51.88	51.87

3.7.4　对抗训练算法性能耗时评估实验

为进一步验证评估各对抗训练算法的性能,在 CIFAR-10 数据集上各对抗训练方法的平均单轮训练耗时见表 2。表 2 的统计结果显示 CAT 最为耗时,主要原因是其使用了多种对抗攻击方法生成样本;FAT 耗时最少,该方法会根据样本类别决定是否用于模型参数更新;DSSIM-AT 耗时约 340 s,即该方法可以用少许时间开销获得较高的模型鲁棒性提升。

表 2　不同对抗训练算法运行耗时统计表

	PGD-AT	FAT	CAT	ES-VAL	DSSIM-AT
对抗样本生成耗时	255 s	245 s	265 s	255 s	260 s
总耗时	300 s	280 s	350 s	320 s	340 s

4 结 论

本文分析了当前对抗训练算法存在鲁棒过拟合问题,指出了现有解决方法仅依赖于范数约束的局限性,并基于此提出了一种基于非范数约束增强的对抗训练方法 DSSIM-AT,在原有范数约束的基础上,引入非范数约束从相异性的角度更为细粒度控制用于对抗训练的对抗样本生成。从定性分析的角度,对比分析了当引入 DSSIM 非范数约束情况下,能够有效剔除对抗样本中的无语义特征噪声,提升对抗样本的质量;从定量分析的角度,在 MNIST 和 CIFAR-10 数据集上的实验结果表明,DSSIM-AT 较现有对抗训练方法在干净样本和对抗样本数据集上识别准确率均获得提升。

在未来的工作中,将进一步深入研究如何通过对抗训练来提升深度神经网络模型的对抗鲁棒性,并将从以下几个方面进行尝试:(1)如何在大规模数据集,如包含 1000 分类的 ImageNet 上有效地进行对抗训练,其中数据集包含的类别数与图像分辨率均会给对抗训练的过程带来困难;(2)近期刚兴起的基于 Transformer 技术的 ViT 在图像分类任务方面取得了和卷积网络模型相当的识别水准,两者在学习图像特征时分别侧重于全局与局部特征,能否借助 ViT 或融合两者的特性,进而设计新的更具鲁棒性的网络架构;(3)借助半监督/无监督技术来提升对抗训练效果。当前对抗训练使用的样本数据均是基于训练集已有数据生成的,未标注的数据目前很少被用于对抗训练。通过结合半监督/无监督技术,如基于对抗生成网络来快速合成样本,使得网络模型能够学习到充足的样本特征,从而提升网络模型的对抗防御能力。

参考文献

[1] BRENDEL W, RAUBER J, KURAKIN A, et al. Adversarial vision challenge[M]. Cham:Springer, 2020: 129-153.

[2] 白祉旭,王衡军,郭可翔. 基于深度神经网络的对抗样本技术综述[J]. 计算机工程与应用,2021, 57(23): 61-70.

[3] 刘西蒙, 谢乐辉, 王耀鹏, 等. 深度学习中的对抗攻击与防御[J]. 网络与信息安全学报, 2020, 6(5): 36-53.

[4] SZEGEDY C, ZAREMBA W, SUTSKEVER I, et al. Intriguing properties of neural networks [EB/OL]. (2014-02-19) [2022-03-05]. https: // arxiv. org/pdf/1312.6199. 1312.6199v4. pdf.

[5] XIE C, WU Y, MAATEN L, et al. Feature denoising for improving adversarial robustness[C] // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019: 501-509.

[6] DZIUGAITE G K, GHAHRAMANI Z, ROYD M. A study of the effect of jpg compression on adversarial images [EB/OL]. (2016-08-02) [2022-03-05]. https: // arxiv. org/pdf/1608.00853. pdf.

[7] PAPERNOT N, MCDANIEL P, WU X, et al. Distillation as a defense to adversarial perturbations against deep neural networks[C] // 2016 IEEE Symposium on Security and Privacy (SP). San Jose :IEEE, 2016: 582-597.

[8] COHEN J, ROSENFELD E, KOLTER Z. Certified adversarial robustness via randomized smoothing[EB/OL]. (2019-06-15) [2022-03-05]. https: // arxiv. org/pdf/1902.02918v2. pdf.

[9] 孔锐, 蔡佳纯, 黄钢. 基于生成对抗网络的对抗攻击防御模型[J]. 自动化学报, 2020, 41(x): 1-17.

[10] GOODFELLOW I J, SHLENS J, SZEGEDY C. Explaining and harnessing adversarial examples [EB/OL]. (2014-05-20) [2022-03-05]. https: // arxiv. org/pdf/1412.6572v3. pdf.

[11] CARLINI N, WAGNER D. Towards evaluating the robustness of neural networks[C] // 2017 IEEE Symposium on Security and Privacy. San Jose: IEEE, 2017: 39-57.

[12] ATHALYE A, CARLINI N, WAGNER D. Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples[C] // International Conference on Machine Learning. Macau :PMLR, 2018: 274-283.

[13] CROCE F, HEIN M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks[C] // International Conference on Machine Learning. Shangrila :PMLR, 2020: 2206-2216.

[14] Ma¸DRY A, MAKELOV A, SCHMIDTL, et al. Towards deep learning models resistant to adversarial attacks[EB/OL]. (2019-09-04) [2022-03-05]. https: // arxiv. org/pdf/1805.04807v4. pdf.

[15] BAI T, LUO J, ZHAO J, et al. Recent advances in adversarial training for adversarial robustness [EB/OL]. (2021-04-21) [2022-03-05]. https: // arxiv. org/pdf/2102.01356v5. pdf.

[16] RICE L, WONG E, KOLTER Z. Overfitting in adversari-

ally robust deep learning[C]// International Conference on Machine Learning. Shangri-la: PMLR, 2020: 8093-8104.
[17] KANNAN H, KURAKIN A, GOODFELLOW I. Adversarial logit pairing [EB/OL]. (2018-03-16) [2022-03-05]. https://arxiv.org/pdf/1803.06373.pdf.
[18] LI P, YI J, ZHOU B, et al. Improving the robustness of deep neural networks via adversarial training with triplet loss [EB/OL]. (2019-05-28) [2022-03-05]. https://arxiv.org/pdf/1905.11713.pdf.
[19] MAO C, ZHONG Z, YANG J, et al. Metric learning for adversarial robustness[EB/OL]. (2019-10-28) [2022-03-05]. https://arxiv.org/pdf/1909.00900v2.pdf.
[20] ZHANG J, XU X, HANB, et al. Attacks which do not kill training make adversarial learning stronger[C]// International conference on machine learning. Shangri-la : PMLR, 2020: 11278-11287.
[21] CAI Q Z, DU M, LIU C, et al. Curriculum adversarial training [EB/OL]. (2018-05-13) [2022-03-05]. https://arxiv.org/pdf/1805.04807.pdf.
[22] TRAMÈR F, KURAKIN A, PAPERNOT N, et al. Ensemble adversarial training: Attacks and defenses [EB/OL]. (2017-04-26) [2022-03-05]. https://arxiv.org/pdf/1705.07204v5.pdf.
[23] KURAKIN A, GOODFELLOW I J, BENGIO S. Adversarial examples in the physical world[M]. Chapman and Hall: CRC, 2018: 99-112.
[24] XIAO C, LI B, ZHU J Y, et al. Generating adversarial examples with adversarial networks [EB/OL]. (2018-02-14) [2022-03-05]. https://arxiv.org/pdf/1801.02610.pdf.
[25] JANDIAL S, MANGLA P, VARSHNEY S, et al. AdvGAN++: harnessing latent layers for adversary generation[C]// Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. Seoul: IEEE, 2019.
[26] WANG B, FAN X, JINGQ, et al. AdvCGAN: an elastic and covert adversarial examples generating framework[C]//2021 International Joint Conference on Neural Networks (IJCNN). Shenzhen: IEEE, 2021: 1-8.
[27] WANG Z, BOVIK A C, SHEIKH H R, et al. Image quality assessment: from error visibility to structural similarity[J]. IEEE Transactions on Image Processing, 2004, 13(4): 600-612.
[28] 董胤蓬, 苏航, 朱军. 面向对抗样本的深度神经网络可解释性分析[J]. 自动化学报, 2022, 48(1): 75-86.
[29] CROCE F, HEIN M. Minimally distorted adversarial examples with a fast adaptive boundary attack[C]//International Conference on Machine Learning. Shangri-la: PMLR, 2020: 2196-2205.

Improving adversarial training with DSSIM based non-norm constraint

WANG Baoli[*,**], FAN Xinxin[**], JING Quanliang[*,**], BI Jingping[**]
([*]School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 100049)
([**]Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)

Abstract

Aiming at the robust overfitting problem in the process of adversarial training (AT), i.e., the adversarial defense performance of the network model will not rise gradually but inversely fall to some extent with the increase of adversarial training rounds, this work proposes a novel adversarial training method that leverages a non-norm constraint based on structural dissimilarity, named DSSIM-AT. For the first time, non-norm constraints are introduced to remove non-semantic features of generated adversarial examples from the structural dissimilarity perspective, making them more suitable for AT. The proposed method further designs a gradient asynchronous update mechanism, which optimizes the time-consuming of adversarial examples generation and model parameters update. The experimental results show that DSSIM-AT can effectively alleviate the robust overfitting problem. Compared with the existing baseline methods, the proposed DSSIM-AT can improve the recognition accuracy of clean examples on dataset CIFAR-10 by 3% approximately, while the recognition accuracy for adversarial examples can be improved by 4% −8%.

Key words: adversarial attack, adversarial defense, adversarial training(AT), non-norm constraint