计算机科学
COMPUTER SCIENCE

基于动量的映射式梯度下降算法

吴子斌，闫巧

引用本文

吴子斌，闫巧. 基于动量的映射式梯度下降算法[J]. 计算机科学, 2022, 49(6A): 178-183.

WU Zi-bin, YAN Qiao. Projected Gradient Descent Algorithm with Momentum[J]. Computer Science, 2022, 49(6A): 178-183.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.210500039

基于动量的映射式梯度下降算法

吴子斌　闫　巧
深圳大学计算机与软件学院　广东 深圳 518060
(695193423@qq.com)

摘　要　近年来,深度学习已被广泛应用于计算机视觉问题中,并取得了卓越的成功。但研究人员发现神经网络容易受到添加微弱扰动的原始样本的干扰,导致模型给出一个错误的输出,这类输入样本称为“对抗样本”。目前已有一系列生成对抗样本的算法被提出。针对已有的对抗样本生成算法——映射式梯度下降算法(Projected Gradient Descent),提出了结合动量并采用新的损失函数的改进方法 $MPGD_{CW}$ 算法,以确保更新方向的稳定且避免不良局部最大值的出现,同时避免交叉熵损失函数可能出现的梯度消失情况。通过与包含3种架构4个鲁棒模型的实验,证实了所提 $MPGD_{CW}$ 算法具有更优的攻击效果和更强的攻击迁移性。

关键词:深度学习;卷积神经网络;图像对抗样本;对抗攻击
中图法分类号　TP391.41;TP18

Projected Gradient Descent Algorithm with Momentum

WU Zi-bin and YAN Qiao
College of Computer Science & Software Engineering,Shenzhen University,Shenzhen,Guangdong 518060,China

Abstract　In recent years,deep learning is widely used in the field of computer vision and has achieved outstanding success. However,the researchers found that the neural network is easily disturbed by adding subtle perturbations in the dataset,that can cause the model to give incorrect outputs. Such input examples are called “adversarial examples”. At present,a series of algorithms for generating adversarial examples have emerged. Based on the existing adversarial sample generation algorithm—projected gradient descent(PGD),this paper proposes an improved method—$MPGD_{CW}$ algorithm,which combines momentum and adopts a new loss function to ensure the stability of the update direction and avoid bad local maximums. At the same time,it can avoid the disappearance of the gradient by replacing the cross-entropy loss function. Experiments on 4 robust models containing 3 architectures confirm that the proposed $MPGD_{CW}$ algorithm has better attack effect and stronger transfer attack capacity.

Keywords　Deep learning,Convolutional neural network,Image adversarial examples,Adversarial attacks

1　引言

深度神经网络是人工智能领域的重要分支。随着近几年深度学习技术的飞速发展,人工智能在计算机视觉[1]、自然语言处理[2]、语音识别[3]等领域获得了巨大的成功,在某些领域已经达到甚至超过了人类的水平[4]。尤其是计算机视觉领域的各项技术日益成熟,如今已逐步部署到实际系统中,如基于深度神经网络的图像识别[5]被应用到移动设备的 FaceID、支付宝的刷脸支付,基于目标检测[6]的体温监测也在疫情期间发挥了重要作用,基于图像语义分割[7]的无人驾驶技术成为当下讨论的热点等。

以图像分类为例,深度学习中的一个重要应用是利用卷积神经网络进行图像分类。在图像分类任务中,深度神经网络能够通过组合图像不同的低阶特征形成更加抽象的高阶图像特征,网络在经过训练后能给出这些高阶特征组成各个目标对象的概率,即分类结果。相较于传统神经网络,卷积神经网络的特征提取是通过卷积运算实现的,卷积运算不仅能让神经网络接受多维图像数据的输入而无需经过其他处理,而且卷积核拥有更大的视野,相比于传统的特征提取方式能更有效地提取局部特征,同时显著减少了网络参数。卷积神经网络的另一个值得称道的优点是权值共享机制,一个卷积核权值不变地扫描全图,从视觉系统的角度理解则是以一种特定的视角观察整个图像,确保每一个卷积核仅提取一种特征,进一步减少了需要训练的权值。此外,dropout 操作令参数更稀疏,一定程度上避免了过拟合等。参数和权值的大幅减少提高了训练速度,使卷积神经网络获得了比传统神经网络更优秀的性能,因此被广泛应用在图像处理方面。

卷积神经网络进行图像处理时具有优异的性能,因此被广泛地应用在图像分类任务中。然而,早在2014年,Szegedy等[8]发现深度神经网络学习的输入到输出映射在很大程度上是相当不连续的,容易受到添加微弱扰动的原始样本的干扰,这些扰动人眼通常无法察觉,却能造成深度神经网络输出错误的结果。这类添加微弱扰动的输入样本被称为对抗样本。由于对抗样本的存在,人工智能安全领域面临着前所未有的巨大威胁,其中蕴含的风险可能导致基于深度神经网络的识别系统出现混乱,形成误判或导致系统崩溃甚至被劫持,造成

基金项目:国家自然科学基金面上项目(61976142)
This work was supported by the National Natural Science Foundation of China(61976142).
通信作者:闫巧(yanq@szu.edu.cn)

巨大的安全隐患。因此对抗样本成为了研究人员关注的重点，继而出现了一系列对抗攻击的方法，如 Goodfellow 等[9]提出的快速梯度下降法(Fast Gradient Sign Method，FGSM)实施单步的梯度攻击，以及 Kurakin 等[10]提出的基础迭代法(Basic Iterative Method，BIM)进行迭代的梯度攻击等。除了攻击方向的相关研究，也有防御方向的研究，无论哪个方向的研究，其出发点都是保证人工智能的安全可靠。对抗样本的存在，不仅仅能攻击神经网络，还是鲁棒性更好的神经网络的重要训练样本，所以推动对抗样本生成算法的发展能够促进更有效的防御算法的诞生，具有现实意义。

针对目前已加入相应对抗防御措施的卷积神经网络分类模型在面对对抗样本时仍具备较高的识别率，本文从攻击者的角度，提出一种改进已有 PGD[11]算法的思路，贡献主要如下。

(1)提出一种将动量迭代方法与基本 PGD 算法相结合的基于动量的 PGD 算法(Momentum Projected Gradient Descent，MPGD)，增强了算法的攻击迁移性。

(2)选用 CW 损失函数[12]替换基本 PGD 算法广泛采用的交叉熵损失函数，以避免梯度消失的情况，增强算法的攻击能力。

(3)通过与原有 PGD 算法对比，针对某一鲁棒神经网络模型生成对抗样本，并用这些生成的对抗样本攻击其他网络模型，证实了所提出的 MPGD 算法具有更好的攻击迁移性；通过与原有 PGD 算法及其他对抗样本生成算法对比，针对同一鲁棒模型生成对抗样本，证实所提出的 $MPGD_{CW}$ 算法具有更强的攻击能力。

2 相关工作

2.1 对抗样本生成技术

Szegedy 等[8]于 2014 年提出了深度神经网络一个有趣的特性——其所学习的输入到输出的映射很大程度上是不连续的，当对原样本添加微弱扰动时，模型容易产生错误结果，使图片以很高的置信度被错误分类；并首次将这类添加扰动的输入样本称为对抗样本。

对抗样本示例如图 1 所示，预训练 ResNet-18 架构的卷积神经网络将原始样本以 70.7%的置信概率分类为“拉布拉多猎犬”；但在使用 PGD 算法加入了微小的对抗扰动后，原本的“拉布拉多猎犬”图像以高置信概率(99.9%)错误分类为“惠比特犬”，且因扰动量非常小，原始样本与对抗样本在视觉上的差距人眼几乎不可察觉。

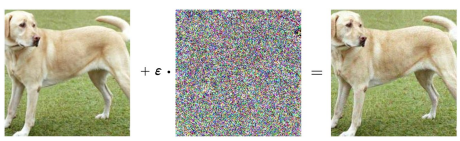

图 1 PGD 算法在 ResNet-18 上生成对抗样本演示

Fig. 1 Demonstration of PGD algorithm generates adversarial examples on ResNet-18

对抗样本的生成算法根据最终的攻击成功标准，可分为只需使原始样本的分类结果为任意错误分类的无目标攻击和分类为某一指定错误分类的有目标攻击。又或者根据攻击者对被攻击神经网络的知悉程度，将得知目标神经网络的具体参数等所有信息的前提下进行的对抗攻击归类为白盒攻击；相对地，在仅可对目标神经网络进行查询而无法得知其他参数的条件下进行的攻击归类为黑盒攻击。如今，各类攻击均出现了一系列算法。

起初，Szegedy 等[8]将对抗样本的存在性归结于神经网络的高度非线性，模型只是过拟合地学习到非对抗样本的特征，而非真正需要学习的泛化特征，导致即便是微小的扰动也容易导致神经网络的分类结果出错。对于如何生成对抗样本，Szegedy 等将其归结为一个优化问题[8]：

$$\text{Minimize } \|r\|_2 \quad \text{s.t. } \begin{cases} f(x+r)=l \\ 2.\ x+r\in[0,1]^m \end{cases} \tag{1}$$

其中，$f:R^m\to\{1\cdots k\}$ 表示将图像像素向量映射到离散标签集的分类器，$x\in R^m$ 表示原始图像，$l\in\{1,\cdots,k\}$ 表示目标标签(与图像原始标签不同)，而 r 则是添加的扰动。该优化问题可概括为找到最小化扰动 r，使得分类器 f 在接受添加扰动的原始样本 $x+r$ 时将其分类到新分类 $l(f(x)\neq l)$，且在添加扰动后的 $x+r$ 仍在合理数值范围内。该优化问题非凸，不容易求解结果，因此作者采用了有界约束的 L-BFGS 算法寻找近似解，即原问题被转化为[8]：

$$\text{Minimize } c|r|+loss_f(x+r,l) \quad \text{s.t. } x+r\in[0,1]^m \tag{2}$$

其中，$loss_f$ 表示分类器的损失函数。对于 $D(x,l)$ 任意选择的最小化器，作者通过线性搜索找到最小值 $c>0$ 来找到 $D(x,l)$ 的近似值。此外，文中还提出了对抗样本具有迁移性，即针对某一神经网络生成的对抗样本同样可以欺骗其他网络。

对于神经网络而言，训练的目的是找到合适的权重 W 和偏置项 b，使得神经网络的效果最好。而判断神经网络效果的好坏通常是定义一个代价函数(损失函数)，衡量期望输出与神经网络预测输出的差距。为了具体说明梯度下降的原理，这里使用 Nielsen[13]举的一个例子，首先定义损失函数为二次损失函数(均方误差，Mean Squared Error，MSE)[13]：

$$C(w,b)\equiv\frac{1}{2n}\sum_x\|y(x)-a\|^2 \tag{3}$$

假设 C 是一个具有 m 个变量 $v_1,v_2,v_3,\cdots,v_m$ 的函数，C 中自变量的变化量 $\Delta v=(\Delta v_1,\Delta v_2,\Delta v_3,\cdots,\Delta v_m)^T$，$C$ 的梯度向量为 $\nabla C\equiv\left(\frac{\partial C}{\partial v_1},\cdots,\frac{\partial C}{\partial v_m}\right)$，相对应地，$C$ 的变化量应该是 $\Delta C\approx\nabla C\cdot\Delta v$。如果希望神经网络的效果更好，则需要降低损失，即 $\Delta C<0$，那么需要自变量的变化量取如下形式：$\Delta v=-\eta\nabla C$，则 $\Delta C\approx-\eta\nabla C\cdot\nabla C=-\eta\|\nabla C\|^2\leqslant 0$，其中 η 为一个很小的正数，被称为学习率。变化量 Δv 取该形式意味着训练过程中只需沿着当前梯度的反方向不断更新变化量函数，函数即可达到最小值或局部最小值，这一更新规则被称为梯度下降法。在神经网络的反向传播当中，通常采用梯度下降法在训练过程中沿着梯度方向更新权值 W 和偏置项 b，使得网络向 $Loss$ 减小的方向收敛[13]：

$$W_k'=W_k-\eta\frac{\partial C}{\partial W_k};b_l'=b_l-\eta\frac{\partial C}{\partial b_l} \tag{4}$$

相对地，由于神经网络的参数是固定不变的，如果要针对该神经网络生成对抗样本，则应该使添加的扰动在合理的范围内对神经网络造成尽可能大的影响，也就是使网络的 $Loss$ 尽可能大。Goodfellow 等[9]的快速梯度符号算法(Fast Gradient Sign Method，FGSM)正是根据这一思路提出的。假设

输入的原始样本为 x，先求得模型对 x 的导数以得到梯度，然后用符号函数 $sign(\cdot)$ 得到具体的梯度方向，再在该方向乘以扰动幅度即可得到添加的扰动，最后将扰动加上原来的输入即可得到对抗样本。该过程可概括为如下公式[9]：

$$\eta=\varepsilon\, sign(\nabla_x J(\theta,x,y)) \tag{5}$$

其中，θ 为网络参数；x 为模型输入；y 为模型输出；$J(\theta,x,y)$ 表示神经网络的损失函数；$\nabla_x J(\theta,x,y)$ 计算了在当前网络 θ 参数下的损失函数梯度；$sign(\cdot)$ 是数学符号函数，为所添加扰动的幅度系数；η 则表示最终生成对抗样本需要加入的扰动。

如图 2(a)所示，假设扰动约束 ε 使用 L_∞ 范式，则扰动空间就成为一个方形，原始样本与每条边的距离都是 ε。如图 2(b)所示，计算出原始样本 x^0 的梯度，正常训练时的梯度下降方式是向着梯度的反方向移动使模型收敛，即图中的 x^1。而 FGSM 算法则是希望扰动造成的影响尽可能大，所以在梯度方向添加扰动，使用 $\varepsilon * sign(\cdot)$ 得到最终的扰动，即图中的 x^*。

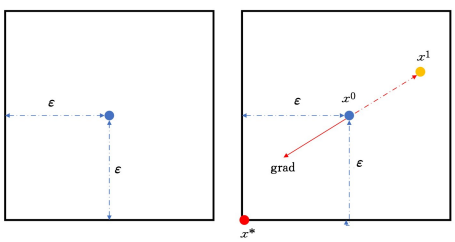

图 2 快速梯度符号法示意图

Fig. 2 Schematic diagram of fast gradient sign method

FGSM 算法只添加一次扰动，具有极高的效率，因此也常被用作新的对抗防御算法的基准评估方法。其后大部分基于梯度的攻击方法都遵循这一基本思路并不断进行改进，如进行多次迭代、加入裁剪操作、引入动量等。

此外，Goodfellow 等[9]还发现即便构建一个线性的模型并加入对抗干扰，只要模型的输入拥有足够的维度就可以具有对抗样本，反驳了对抗样本的存在是因为模型的高度非线性的解释，同时还指出深度神经网络模型对对抗样本的脆弱性正是源于网络高维空间中的线性。

基于对抗样本的线性解释，越来越多的对抗样本生成算法相继出现。

2.2 PGD 算法

PGD 算法的核心是迭代的 FGSM 算法，而最先提出基于 FGSM 算法进行迭代的是 Kurakin 等提出的基础迭代方法(Basic Iterative Method，BIM)[10]：

$$\begin{aligned} &X_0^{adv}=X \\ &X_{N+1}^{adv}=Clip_{X,\varepsilon}\{X_N^{adv}+\alpha\, sign(\nabla_x J(X_N^{adv},y_{true}))\} \end{aligned} \tag{6}$$

其中，X_N^{adv} 表示迭代 N 次后的对抗样本；α 表示步长，即单次扰动的幅度；y_{true} 表示原图像样本的正确分类标签；而 $Clip_{X,\varepsilon}\{X'\}$ 则是裁剪函数，作用是在每一步迭代后剪切中间结果的像素值，确保结果位于原始图像的值域中，具体定义如下[10]：

$$\begin{aligned} Clip_{X,\varepsilon}\{X'\}=&\min\{255,X(x,y,z)+\varepsilon,\max\{0,(x,y,z)- \\ &\varepsilon\},X'(x,y,z)\} \end{aligned} \tag{7}$$

其中，z 指的是图像样本的通道数，而 ε 则表示最大的扰动幅度。BIM 算法对 FGSM 算法进行了扩展，以较小的步长进行了多次迭代，同时引入了 Clip 方法对每次迭代的结果进行裁剪。实验表明，BIM 算法在自然图像上比 FGSM 算法更有效。

PGD 算法在进行梯度迭代前后的操作与 BIM 算法有所不同，Madry 等[11]认为，对抗鲁棒性工作中的鞍点问题可以视为内部最大化问题和外部最小化问题的组合，内部最大化问题旨在找到对抗样本实现高损失，内部最小化问题则是使攻击问题给定的对抗损失最小化。为了解决鞍点问题，作者给出了两个方案，其一是进行小步长的梯度迭代。PGD 算法的迭代过程可概括为[11]：

$$x^{t+1}=\prod_{x+S}(x^t+\alpha\, sign(\nabla_x L(\theta,x,y))) \tag{8}$$

其中，$\prod$ 表示 Projection 操作，L 表示损失函数。Pojection 的实际作用与 BIM 算法中的 Clip 函数的实际作用无异，但在实现上 BIM 算法的裁剪是通过最大最小值运算的形式，而 PGD 在实现时则使用 numpy 的 clip 方法或 torch 的 clamp 方法；此外，PGD 不同于 BIM 算法还体现在作者的第二个方案，即在进行迭代前尝试加入随机噪声进行初始化，以便在线性化模型损失之前逃避数据点的非平滑区域。这些特点使得 PGD 算法得到了较强的一阶攻击效果。

3 基于动量的 PGD 算法

3.1 算法的改进

本文基于动量的 PGD 算法，将动量迭代与 PGD 算法相结合，确保更新方向的稳定且避免不良局部最大值的出现，同时替换了新的损失函数，避免交叉熵损失函数可能出现的梯度消失情况。改进算法具备较强的攻击能力和攻击迁移性。

3.1.1 引入动量迭代思想

PGD 已被证明拥有较优的一阶攻击效果，但其核心本质依然是 FGSM 算法，这就意味着 PGD 在每次迭代中都会将对抗样本沿着梯度符号的方向贪婪地移动，导致对抗样本很容易掉入不良的局部最大值并“过度拟合”模型，提高了 PGD 的白盒攻击能力但削弱了其可传递性。

Polyak[14]最早将动量法用于改进随机梯度下降算法(Stochastic Gradient Descent，SGD)，提出带动量的 SGD(SGD with Momentum，SGDM)。动量的引入让后续的梯度更新保留了之前的梯度信息，使网络能更优、更稳定地收敛，减少了震荡过程，效果如图 3 所示。

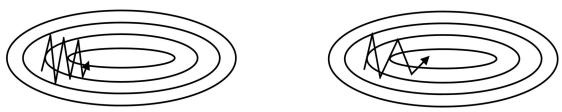

图 3 SGD 引入动量前后的效果对比[15]

Fig. 3 Effect comparision of SGD before and after momentum introduction[15]

基于这一思路，Dong 等[16]尝试将动量法的思想加入到对抗样本的生成算法中[16]：

$$g_{t+1}=\mu g_t+\frac{\nabla_x J(x_t^*,y)}{\|\nabla_x J(x_t^*,y)\|_1} \tag{9}$$

$$x_{t+1}^*=x_t^*+\alpha\, sign(g_{t+1}) \tag{10}$$

以当前损失函数的梯度方向作为速度矢量的方向，并在迭代过程中不断积累速度矢量，确保更新方向的稳定，有效防止样本掉入不良的局部最大值，避免了算法“过度拟合”模型，使算法在不削弱白盒攻击能力的情况下仍保持足够的可传递性。

本文将 MI-FGSM 算法的动量迭代方法与 PGD 算法

相结合，得到基于动量的 PGD(见算法 1)。动量的引入使得 PGD 能够在迭代时中记录以前的梯度信息，保证梯度维持一个稳定的更新方向，而动量和随机噪声初始化将有助于 PGD 穿越狭窄的山谷、小的驼峰和较差的局部最小值或最大值，最终提高了整个 PGD 算法的攻击迁移性。

算法 1 MPGD

输入：分类器 f 的损失函数 J；输入样本 x 和正确标签 y；扰动幅度 ε；步长 α；迭代次数 T 和扰动因子 μ

输出：对抗样本 x^*，满足 $\|x^*-x\|_\infty\leqslant\varepsilon$

1. 初始化动量 $g_0=0$；初始化对抗样本 $x_0^*=x+\alpha\ \mathrm{sign}(\nabla_x J(x,y))$

2. for t=0 to T−1 do

3. 输入 x_t^* 到分类器 f，获得梯度 $\nabla_x J(x_t^*,y)$；

4. 在梯度方向更新动量 g_{t+1}：$g_{t+1}=\mu g_t+\dfrac{\nabla_x J(x_t^*,y)}{\|\nabla_x J(x_t^*,y)\|_1}$

5. 使用梯度符号更新对抗样本 $x_{t+1}^*=\mathrm{Proj}\{x_t^*+\alpha\ \mathrm{sign}(g_{t+1})\}$

6. end for

7. return $x^*=x_T^*$

3.1.2 替换损失函数

在日常训练深度神经网络时，通常采用交叉熵作为损失函数，表示如下[17]：

$$CE(x,y)=-\log p_y=-z_y+\log\left(\sum_{j=1}^{K}e^{z_j}\right)\tag{11}$$

其中，z_i 为神经网络输出中对应某一分类的置信概率，$p_i=e^{z_i}/\sum_{j=1}^{K}e^{z_j}$，$i=1,\cdots,K$，求 x 的偏导得到梯度[17]：

$$\nabla_x CE(x,y)=(-1+p_y)\nabla_x z_y+\sum_{i\neq y}p_i\nabla_x z_i\tag{12}$$

可以看到在式(12)中，如果 $p_y\approx1$，则 $p_i\approx0(i\neq y)$，此时的梯度 $\nabla_x CE(x,y)\approx0$，即梯度消失，此时 PGD 等基于梯度算法如果直接使用交叉熵损失函数计算当前梯度将难以攻击成功。为了避免生成对抗样本时梯度消失的情况，Carlini 和 Wagner[12] 尝试对比了 7 种损失函数的攻击效果，最终得到一个攻击效果最好的损失函数(下称 CW Loss)[12]：

$$CW(x,y)=-z_y+\max_{i\neq y}z_i\tag{13}$$

Croce 等[17] 则在 CW Loss 的基础上提出了 Difference of Logits Ratio(DLR)损失函数[17]：

$$DLR(x,y)=-\frac{z_y-\max\limits_{i\neq y}z_i}{z_{\pi_1}-z_{\pi_3}}\tag{14}$$

其中，π 代表的是 z 的分量按降序顺序，保证了损失函数的平移不变性。DLR 损失函数在 CW Loss 作为分子的基础上，再加入两个对数的差作为母，使得当最大化 DLR Loss 时便可轻易地找到神经网络不将输入样本分类为 y 的点，且此时分类为 y 的可能性最小。当模型预测正确($\pi_1\equiv y$)时，$DLR(x,y)=-\dfrac{z_y-z_{\pi_2}}{z_y-z_{\pi_3}}$，其中 $DLR(x,y)\in[-1,0]$。归一化 $z_{\pi_1}-z_{\pi_3}$ 的作用是将 z_{π_2} 推到 $z_y=z_{\pi_1}$。由于它倾向于 $z_y\approx z_{\pi_2}>z_{\pi_3}$ 的点，因此更倾向于改变决策[17]。

这里尝试了基于 CW 和 DLR 两种损失函数，分别实现了 $\mathrm{PGD_{CW}}$ 和 $\mathrm{PGD_{DLR}}$ 算法。后续通过来实验比对两个损失函数的实际效果，选取表现最好的 CW 损失函数来替换 MPGD 算法中的损失函数。

3.2 实验对比和结果分析

实验所采用的数据集为 CIFAR-10 彩色图像数据集，预训练的鲁棒模型使用的是 Croce 等[18] 提出的标准化对抗鲁棒性基准 Robustbench 中的部分预训练模型。共选取 4 个鲁棒模型，包含 3 种架构，包括 Gowal2020Uncovering_28_10_extra(简记为 Gowal2020)，Wu2020Adversarial_extra(简记为 Wu2020)，Engstrom2019Robustness(简记为 Engstrom2019)，Wong2020Fast(简记为 Wong2020)，具体参数表 1 所列。

表 1 Robustbench 中部分 Linf 鲁棒模型的架构及准确率[19]

Table 1 Architecture and accuracy of some Linf robust models in Robustbench

(单位：%)

Model ID	架构	准确率	鲁棒准确率
Gowal2020	WideResNet-28-10	89.48	62.76
Wu2020	WideResNet-28-10	88.25	60.04
Engstrom2019	ResNet-50	87.03	49.25
Wong2020	ResNet-18	83.34	43.21

首先，通过实验寻找更有效的损失函数，在攻击参数相近的情况下(迭代次数 *step* 均为 16 次，扰动幅度 ε 均为 8/255，PGD 的步长 α 设置为 2/255，为了降低实验的偶然性，这里不使用 PGD 的随机初始化)，从 CIFAR-10 数据集中随机抽取 2 000 张样本图像生成对抗样本，使用各个鲁棒模型测试准确率。各个攻击算法在不同鲁棒模型下的测试结果如表 2 所列。

表 2 不同鲁棒模型下使用不同损失函数 PGD 算法的准确率

Table 2 Accuracy of PGD algorithm using different loss function under different robust models

(单位：%)

攻击方法	Gowal2020	Wu2020	Engstrom2019	Wong2020
Clear	88.80	87.80	86.40	82.9
PGD	65.10	62.00	52.80	46.45
$\mathrm{PGD_{CW}}$	62.90	59.40	52.55	46.30
$\mathrm{PGD_{DLR}}$	63.15	59.60	54.10	48.10

表 2 中的准确率越低，说明该对抗样本生成算法的攻击效果越强。可以看出，$\mathrm{PGD_{DLR}}$ 算法的攻击效果在 WideResNet-28-10 架构的神经网络效果优于使用交叉熵损失函数的基础版本 PGD，但在其他网络架构(ResNet-18，ResNet-50)上甚至不如普通的 PGD，而 $\mathrm{PGD_{CW}}$ 算法则展现出较强的攻击效果，其在各种网络架构模型上的表现都优于 PGD 和 $\mathrm{PGD_{DLR}}$，因此尝试用 CW Loss 替换 MPGD 算法中的交叉熵损失函数。

而在引入动量的各个算法效果对比中，同样采用近似的攻击参数(动量的衰减因子 μ 设置为 1.0，其余参数、测试样本与前文一致)测试 MI-FGSM，MPGD 和 $\mathrm{MPGD_{CW}}$ 的攻击效果，最终测试结果如表 3 所列。

表 3 不同鲁棒模型下各个基于动量算法的准确率

Table 3 Accuracy of each momentum-based algorithm under different robust models

(单位：%)

攻击方法	Gowal2020	Wu2020	Engstrom2019	Wong2020
Clear	88.80	87.80	86.40	82.9
MI-FGSM	67.65	64.65	57.65	51.55
MPGD	66.00	62.55	53.75	47.25
$\mathrm{MPGD_{CW}}$	63.50	59.85	53.40	47.10

可以看到，MPGD 算法的攻击效果优于 MI-FGSM 算法，在不使用 PGD 的随机初始化的情况下，这样的差异是由于 MI-FGSM 的步长是通过 $\varepsilon/step$ 得出而 MPGD 的步长大于 MI-FGSM 而导致的；且在替换了损失函数后，$\mathrm{MPGD_{CW}}$ 算法的攻击效果进一步增强，明显优于其他两种算法。

接下来验证引入动量迭代后，改进的 PGD 算法是否具备

更强的攻击迁移性。这里选取两种不同架构的鲁棒模型：Gowal2020 和 Wong2020。具体测试方法为：

(1)使用算法针对 Gowal2020 模型生成对抗样本；

(2)将生成的对抗样本使用 Wong2020 进行分类；

(3)比较对抗样本经过 Wong2020 分类的准确率。

为了有一个相对统一的标准比较各个算法的性能差异，这里定义攻击效果为经过对抗攻击后，原样本分类准确率的下降比例(数值越大表示效果越好)。具体计算方式如下：

$$\text{攻击效果}=\frac{\text{模型准确率}-\text{对抗样本准确率}}{\text{模型准确率}}=1-\frac{\text{对抗样本准确率}}{\text{模型准确率}} \tag{15}$$

实验分为两组，PGD 与 MPGD 为一组，PGD_{CW} 与 $MPGD_{CW}$ 为一组，分别测试迁移攻击效果，相关参数与前文保持一致，测试结果如表 4、表 5 所列。

表 4 PGD 算法结合动量前后迁移攻击效果的对比

Table 4 Transfer attack effect comparison of PGD before and after momentum introduction

(单位：%)

	Clear	PGD	MPGD
Gowal2020 准确率	88.35	66.67	67.15
Wong2020 准确率	82.80	48.10	49.15
Gowal2020 对抗样本攻击 Wong2020 的准确率	82.80	66.65	66.80
Gowal2020 攻击效果	—	24.54	24.00
Wong2020 攻击效果	—	41.91	40.64
迁移后 Wong2020 攻击效果	—	19.50	19.32
迁移攻击效果 / Gowal2020 攻击效果	—	79.49	80.53
迁移攻击效果 / *Wong*2020 直接攻击效果	—	46.54	47.55

表 5 PGD_{CW} 算法结合动量前后迁移攻击效果的对比

Table 5 Transfer attack effect comparison of PGD_{CW} before and after momentum introduction

(单位：%)

	Clear	PGD_{CW}	$MPGD_{CW}$
Gowal2020 准确率	88.35	64.90	65.30
Wong2020 准确率	82.80	47.80	48.60
Gowal2020 对抗样本攻击 Wong2020 的准确率	82.80	65.80	65.90
Gowal2020 攻击效果	—	26.54	26.09
Wong2020 攻击效果	—	42.27	41.30
迁移后 Wong2020 攻击效果	—	20.53	20.41
迁移攻击效果 / Gowal2020 攻击效果	—	77.35	78.23
迁移攻击效果 / Wong2020 直接攻击效果	—	48.57	49.42

可以发现，迁移攻击效果中无论是与直接攻击 Gowal-2020 模型求比值还是与直接攻击 Wong2020 模型求比值，结合动量的算法(MPGD 和 $MPGD_{CW}$)都具有比原来的算法更好的效果，证明了 PGD 算法在结合动量迭代后攻击迁移性有所增强。

最后结合表 2 和表 3，将 PGD，PGD_{CW}，MPGD 和 $MPGD_{CW}$ 4 种算法相比较，$MPGD_{CW}$ 算法的表现优于 MPGD，可见 CW 损失函数的加入使 MPGD 获得了更强的攻击效果。尽管 $MPGD_{CW}$ 的直接攻击效果稍逊色于 PGD_{CW}，但考虑到动量的加入增强了攻击迁移性，具有更强的黑盒攻击能力，$MPGD_{CW}$ 更有实用价值。

结束语 本文针对已有的对抗样本生成算法 PGD，提出了结合动量并采用新的损失函数的改进方法——$MPGD_{CW}$ 算法，确保了更新方向的稳定且避免了不良局部最大值的出现，同时避免了交叉熵损失函数可能导致的梯度消失问题。实验结果表明，本文提出的 $MPGD_{CW}$ 相比原来的 PGD 算法无论是在直接攻击的效果上还是攻击迁移性上都具备更好的效果。

但该算法仍存在一定的局限和不足。首先，是提出的对抗样本生成算法是基于梯度迭代的，意味着在生成对抗样本时需要得到模型的梯度等信息，尽管 $MPGD_{CW}$ 作为白盒攻击算法已具备不错的攻击效果，但在实际情景中攻击者通常难以得到模型的具体信息，因此该算法的黑盒攻击能力仍相对较弱。目前效果较好的算法为了获得更强的黑盒攻击能力，尝试牺牲一定的时间，集成多种攻击算法进行组合攻击。所以在后续工作中，可以考虑结合已有的一些黑盒攻击算法的思路，继续从增强黑盒攻击能力的方向进行优化改进。此外，$MPGD_{CW}$ 算法在迭代过程中攻击参数是不变的，固定的参数无法确保针对任意输入图像生成对抗样本的每次迭代后都能取得很好的效果，因此在迭代过程中动态地计算步长、迭代次数同样是值得考虑的改进方向。

参 考 文 献

[1] SZEGEDY C, LIU W, JIA Y, et al. Going deeper with convolutions[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 1-9.

[2] MIKOLOV T, KARAFIÁT M, BURGET L, et al. Recurrent neural network based language model[C]// Eleventh Annual Conference of the International Speech Communication Association. 2010.

[3] HINTON G, DENG L, YU D, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups[J]. IEEE Signal Processing Magazine, 2012, 29(6): 82-97.

[4] DEVLIN J, CHANG M W, LEE K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv: 1810.04805, 2018.

[5] KRIZHEVSKY A, SUTSKEVER I, HINTONG E. Imagenet classification with deep convolutional neural networks[J]. Advances in Neural Information Processing Systems, 2012, 25: 1097-1105.

[6] REN S, HE K, GIRSHICK R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[J]. arXiv: 1506.01497, 2015.

[7] LoNG J, SHELHAMER E, DARRELL T. Fully convolutional networks for semantic segmentation[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 3431-3440.

[8] SZEGEDY C, ZAREMBA W, SUTSKEVERI, et al. Intriguing properties of neural networks[J]. arXiv: 1312.6199, 2013.

[9] GOODFELLOW I J, SHLENS J, SZEGEDY C. Explaining and harnessing adversarial examples[J]. arXiv: 1412.6572, 2014.

[10] KURAKIN A, GOODFELLOW I, BENGIO S. Adversarial examples in the physical world[J]. arXiv: 1607.02533, 2016.

[11] MADRY A, MAKELOV A, SCHMIDT L, et al. Towards deep learning models resistant to adversarial attacks[J]. arXiv:1706.06083,2017.

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

[13] NIELSEN M A. Neural networks and deep learning(Vol. 25)[M]. San Francisco, CA: Determination Press, 2015.

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

[15] RUDER S. An overview of gradient descent optimization algorithms[J]. arXiv:1609.04747, 2016.

[16] DONG Y, LIAO F, PANG T, et al. Boosting adversarial attacks with momentum[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018:9185-9193.

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

[18] CROCE F, ANDRIUSHCHENKO M, SEHWAG V, et al. RobustBench: a standardized adversa- rial robustness benchmark [J]. arXiv:2010.09670, 2020.

[19] CROCE F, ANDRIUSHCHENKO M, SEHWAG V, et al. RobustBench/robustbench: RobustBench: a standardized adversarial robustness benchmark [EB/OL]. https://github.com/RobustBench/robustbench.

WU Zi-bin, born in 1998. His main research interests include machine learning and so on.

YAN Qiao, born in 1972, Ph.D, professor, Ph.D supervisor, is a member of China Computer Federation. Her main research interests include network security, software-defined networking and machine learning.

(上接第 143 页)

[17] ZHAO S, TSANG E C C, CHEN D. The model of fuzzy variable precision rough sets [J]. IEEE Transactions on Fuzzy Systems, 2009, 17(2): 451-467.

[18] RADZIKOWSKA A M, KERRE E E. Fuzzy rough sets based on residuated lattices [J]. Lecture Notes in Computer Sciences, 2004, 3135: 278-296.

[19] WANG C Y, ZHANG X G, WU Y H. New results on single axioms for L-fuzzy rough approximation operators [J]. Fuzzy Sets and Systems, 2020, 380: 131-149.

[20] SHE Y H, WANG G J. An axiomatic approach of fuzzy rough sets based on residuated lattices [J]. Computers and Mathematics with Applications, 2009, 58(1): 189-201.

[21] YU H, ZHAN W. On the topological properties of generalized rough sets [J]. Information Sciences, 2014, 263: 141-152.

[22] WU H S, LIU G L. The relationships between topologies and generalized rough sets [J]. International Journal of Approximate Reasoning, 2020, 119: 313-324.

[23] YANG L Y, XU L S. Topological properties of generalized approximation spaces [J]. Information Sciences, 2011, 181: 3570-3580.

[24] QIN K Y, YANG J L, PEI Z. Generalized rough sets based on transitive and reflexive relations [J]. Information Sciences, 2008, 178: 4138-4141.

[25] WANG C Y. Topological characterizations of generalized fuzzy rough sets [J]. Fuzzy Sets and Systems, 2017, 312: 109-125.

[26] QIN K Y, ZHENG P. On the topological properties of fuzzy rough sets [J]. Fuzzy Sets and Systems, 2004, 151(3): 601-613.

[27] WANG C Y. Topological structures of L-fuzzy rough sets and similarity sets of L-fuzzy relations [J]. International Journal of Approximate Reasoning, 2017, 83: 160-175.

[28] MA Z M, HU B Q. Topological and lattices structures of L - fuzzy rough sets determined by lower and upper sets [J]. Information Sciences, 2013, 218: 194-204.

[29] HAO J, LI Q G. The relationship between L-fuzzy rough set and L-topology [J]. Fuzzy Sets Systems, 2011, 178(1): 74-83.

[30] BĚLOHLÁVEK R. Some properties of residuated lattices [J]. Czechoslovak Mathematical Journal, 2003, 53(1): 161-171.

[31] PEI D W. The characterization of residuated lattices and regular residuated lattices [J]. Acta-Mathematica Sinica, 2002, 45: 271-278.

[32] GOGUEN J A. L-fuzzy sets [J]. Journal of Mathematical Analysis and Applications, 1967, 18: 145-174.

[33] WANG Z D, WANG Y, TANG K M. Some properties of L-fuzzy approximation spaces based on bounded integral residuated lattices [J]. Information Sciences, 2014, 278: 110-126.

XU Si-yu, born in 1996, postgraduate. Her main research interests include rough set theory and formal concept analysis.

QIN Ke-yun, born in 1962, Ph.D, professor, Ph.D supervisor. His main research interests include rough set theory, formal concept analysis and fuzzy logic.