软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2024,35(5):2543−2565 [doi: 10.13328/j.cnki.jos.006893]

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

基于龙格库塔法的对抗攻击方法*

万 晨[1], 黄方军[2,3]

[1](中山大学 计算机学院, 广东 广州 510006)

[2](中山大学 网络空间安全学院, 广东 深圳 518107)

[3](郑州信大先进技术研究院, 河南 郑州 450001)

通信作者: 黄方军, E-mail: huangfj@mail.sysu.edu.cn

摘 要: 深度神经网络在许多领域中取得了显著的成果, 但相关研究结果表明, 深度神经网络很容易受到对抗样本的影响. 基于梯度的攻击是一种流行的对抗攻击, 引起了人们的广泛关注. 研究基于梯度的对抗攻击与常微分方程数值解法之间的关系, 并提出一种新的基于常微分方程数值解法-龙格库塔法的对抗攻击方法. 根据龙格库塔法中的预测思想, 首先在原始样本中添加扰动构建预测样本, 然后将损失函数对于原始输入样本和预测样本的梯度信息进行线性组合, 以确定生成对抗样本中需要添加的扰动. 不同于已有的方法, 所提出的方法借助于龙格库塔法中的预测思想来获取未来的梯度信息 (即损失函数对于预测样本的梯度), 并将其用于确定所要添加的对抗扰动. 该对抗攻击具有良好的可扩展性, 可以非常容易地集成到现有的所有基于梯度的攻击方法. 大量的实验结果表明, 相比于现有的先进方法, 所提出的方法可以达到更高的攻击成功率和更好的迁移性.

关键词: 对抗样本; 黑盒攻击; 龙格库塔法; 迁移性

中图法分类号: TP309

中文引用格式: 万晨, 黄方军. 基于龙格库塔法的对抗攻击方法. 软件学报, 2024, 35(5): 2543–2565. http://www.jos.org.cn/1000-9825/6893.htm

英文引用格式: Wan C, Huang FJ. Adversarial Attack Based on Runge-Kutta Method. Ruan Jian Xue Bao/Journal of Software, 2024, 35(5): 2543–2565 (in Chinese). http://www.jos.org.cn/1000-9825/6893.htm

Adversarial Attack Based on Runge-Kutta Method

WAN Chen[1], HUANG Fang-Jun[2,3]

[1](School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China)

[2](School of Cyber Science and Technology, Sun Yat-sen University, Shenzhen 518107, China)

[3](Zhengzhou Xinda Institute of Advanced Technology, Zhengzhou 450001, China)

Abstract: Deep neural networks (DNNs) have made remarkable achievements in many fields, but related studies show that they are vulnerable to adversarial examples. The gradient-based attack is a popular adversarial attack and has attracted wide attention. This study investigates the relationship between gradient-based adversarial attacks and numerical methods for solving ordinary differential equations (ODEs). In addition, it proposes a new adversarial attack based on Runge-Kutta (RK) method, a numerical method for solving ODEs. According to the prediction idea in the RK method, perturbations are added to the original examples first to construct predicted examples, and then the gradients of the loss functions with respect to the original and predicted examples are linearly combined to determine the perturbations to be added for the generation of adversarial examples. Different from the existing adversarial attacks, the proposed adversarial attack employs the prediction idea of the RK method to obtain the future gradient information (i.e., the gradient of the loss function with respect to the predicted examples) and uses it to determine the adversarial perturbations to be added. The proposed attack

* 基金项目: 国家自然科学基金 (62072481); 广州市科技计划 (202201011587); 河南省网络空间态势感知重点实验室开放课题 (HNTS 2022014)
收稿时间: 2022-05-14; 修改时间: 2022-07-14, 2022-09-26; 采用时间: 2023-01-05; jos 在线出版时间: 2023-06-28
CNKI 网络首发时间: 2023-06-29

features good extensibility and can be easily applied to all available gradient-based attacks. Extensive experiments demonstrate that in contrast to the state-of-the-art gradient-based attacks, the proposed RK-based attack boasts higher success rates and better transferability.

Key words: adversarial example; black-box attack; Runge-Kutta method; transferability

近年来, 深度神经网络 (deep neural network, DNN) 在图像处理[1,2]、自然语言处理[3]、语音识别[4]等诸多领域都取得了显著的成果, 甚至已经超越了人类. 然而, 最近的研究表明, 几乎所有的深度学习模型都存在着安全隐患[5−7]. 在原始样本中添加一些微小的扰动可以得到对抗样本, 添加扰动后的对抗样本与原始样本在观察者看来具有相同的类别或属性, 但会误导神经网络模型产生错误的输出[8]. 这种在原始样本中添加微弱噪声误导分类器的操作称之为对抗攻击, 其核心思想是最大化网络模型的损失函数, 生成可以误导神经网络的对抗样本[9]. 对抗样本的存在给深度神经网络的实际应用带来了严重的挑战, 研究对抗样本有助于分析基于深度学习系统存在的安全漏洞, 并建立相应的防范机制.

迄今为止, 研究者提出了许多对抗攻击算法[8−11], 如基于梯度的攻击[8]、基于优化的攻击[10]和基于生成对抗网络的攻击[11]等. 在这些方法中, 基于梯度的攻击具有较低的计算成本和较好的性能, 受到学术界和产业界的共同关注[12−15]. 根据攻击者对目标模型的了解程度, 现有的对抗攻击通常划分为白盒攻击和黑盒攻击两类[9]. 在白盒攻击中, 攻击者可以访问目标模型的所有信息, 包括模型的结构、参数和梯度等, 并充分使用这些信息来制作对抗样本[8]. 然而, 在现实场景中, 网络结构和相关数据都是严格保密的, 攻击者很难知道模型的所有信息, 需要在不知道模型的详细信息的条件下攻击模型, 即黑盒攻击[16,17], 这比白盒攻击更加困难. 目前, 常用的两种黑盒攻击方法如下所示.

● 基于查询的攻击. 基于查询的攻击通过对目标模型多次交互查询来观察模型输出结果的变化, 推测出有利于攻击模型的信息, 用于近似估计网络模型的损失函数梯度方向[18]或直接使用贪婪策略搜索对抗扰动[19], 进而实施对抗攻击. 这类方法通常需要多次查询访问目标模型, 耗费大量的计算资源, 导致生成对抗样本的效率较低, 难以应用在实际场景中.

● 基于迁移性的攻击. 基于迁移性的攻击不需要对目标模型进行任何的输入输出查询, 直接利用对抗样本的迁移性误导目标模型产生错误的输出[20]. 对抗样本的迁移性是指攻击一个给定模型生成的对抗样本仍然可以以很高的概率欺骗另一个模型, 这使得在不了解模型的情况下攻击模型成为可能. 然而, 在面对经过对抗训练的防御模型的时候, 现有的方法生成的对抗样本的迁移性通常会大幅度地降低.

最近, 对抗样本的迁移性吸引了学者们的广泛研究[20−27]. 2017 年, Liu 等人[20]首次在大型模型和大规模数据集上研究了对抗样本的迁移性, 指出对抗样本的迁移性是由于不同模型具有相似的决策边界造成的. Wu 等人[21]从模型的结构角度讨论了对抗样本的迁移性, 研究表明攻击一个给定模型产生的对抗样本可以以很高的成功率迁移到与给定模型具有相似结构的目标模型中. 此外, 为了提高对抗样本的迁移性, 学者们提出了各种技术, 例如高级梯度计算[22−24]、模型融合攻击[22]、数据增广策略[25−28]和模型结构调整[29]. 在这些方法中, 高级梯度计算和数据增广策略是两种常用的方法. 基于高级梯度计算的攻击通过引入一个新的高级梯度项, 尽可能地预防生成的对抗样本陷入较差的局部极值. 数据增广策略是在梯度计算过程中对输入样本执行一系列转换操作, 以防止生成的对抗样本过度拟合模型. 在这些方法中, 所添加的对抗扰动主要是通过损失函数对于样本的梯度决定的, 这也就意味着梯度对于攻击算法至关重要. 然而, 现有的方法中, 对抗扰动主要是由过去和当前的梯度决定, 即损失函数关于过去输入样本和当前输入样本的梯度, 很少考虑到未来的梯度信息. 在实践中, 如果考虑更全面的梯度信息 (即过去、当前和未来的梯度信息), 通常情况下可以进一步地提高算法的攻击成功率.

在本文中, 我们研究了对抗攻击方法与常微分方程的数值解法之间的密切关系. 数值方法的基本原理是通过给定的初值和微分系数来求解精确解的近似值[30,31]. 在我们看来, 对抗样本的生成过程可以近似看作常微分方程中近似值的求解过程, 二者都是通过梯度或导数来确定求得的对抗样本或近似解, 从而达到彼此的目标. 受常微分方程的数值解法中龙格库塔法的思想的启发, 本文提出了一种新的基于龙格库塔法的攻击方法. 在我们的方法中, 添加到输入样本中的对抗扰动是由一个原始梯度和多个预测梯度的线性组合决定, 其中原始梯度和预测梯度分别表示损失函数对于原始输入样本和预测样本的梯度. 也就是说, 本文所提出的方法使用了未来的一些梯度信息 (即

预测梯度) 来确定对抗扰动, 其核心步骤是利用已有的信息预测未来可能生成的对抗样本, 即构建预测样本. 与现有的基于梯度的攻击相比, 本文的主要贡献如下.

1) 研究了基于梯度的对抗攻击与常微分方程的数值解法之间的关系, 在对抗攻击与数值解法之间建立了紧密的联系.

2) 基于上述发现, 提出了一系列新的基于龙格库塔法的对抗攻击, 该方法在白盒攻击和黑盒攻击中都可以达到较高的攻击成功率, 且生成的对抗样本具有较好的迁移性.

3) 所提出的基于龙格库塔法的对抗攻击具有很好的可扩展性, 可以非常容易地集成到现有的所有基于梯度的攻击方法.

本文第 1 节简要回顾相关工作. 在第 2 节中, 首先研究对抗攻击与常微分方程的数值解法之间的密切关系. 基于此发现, 提出一种新的基于龙格库塔法的对抗攻击方法. 最后, 研究所提出算法的可扩展性, 并将其融入到现有的方法中, 形成一系列新的对抗攻击方法. 第 3 节通过大量地实验验证本文所提出的方法的有效性. 第 4 节是本文工作的总结.

1 相关工作

将 x 定义为干净的图像, 其对应的真实标签用 y 表示. 给定一个分类器 F, 输出 $\hat{y}=F(x)$ 作为输入图像 x 的预测标签. 对抗样本 x^{adv} 是在 x 中添加微小的扰动而得到的, 它可以误导分类器, 即 $F(x^{\text{adv}})\neq y$. 我们采用 $L(x,y)$ 表示预测标签与真实标签之间的损失函数. 在无目标攻击中, 对抗样本生成的目标是使损失函数 $L(x^{\text{adv}},y)$ 最大化. 添加的对抗扰动通常情况需要小于给定的约束值 ε, 即 $\left\|x^{\text{adv}}-x\right\|_p\leqslant\varepsilon$, 其中 p 表示用于测量扰动幅值的模, 包括 l_0, l_1, l_2, l_∞ 范数. 在本文中, 我们用 l_∞ 范数来测量失真.

1.1 基于梯度的对抗攻击

在所有基于梯度的对抗攻击中, 快速梯度符号方法 (fast gradient sign method, FGSM)[8]是最经典的一种. 现有的基于梯度的攻击一般都是基于 FGSM 的改进, 如 I-FGSM (iterative FGSM)[12]、MI-FGSM (momentum iterative FGSM)[22]、NI-FGSM (Nesterov iterative FGSM)[23]均是基于 FGSM 的迭代攻击.

快速梯度符号法 (FGSM)[8]通过计算损失函数的梯度来生成对抗样本 x^{adv}, 对 x 进行一步更新.

$$x^{\text{adv}}=x+\varepsilon\cdot sign(\nabla_x L(x,y)) \tag{1}$$

其中, ε 表示 l_∞ 约束下生成的对抗样本的最大扰动量, $sign(\cdot)$ 为符号函数, $\nabla_x L$ 表示损失函数 L 对输入样本 x 的梯度.

基于迭代的快速梯度符号法 (I-FGSM)[12]通过在 FGSM 中引入迭代的思想, 通过多次迭代的方式, 以较小的步长 α 逐步添加扰动生成对抗样本.

$$x_{t+1}^{\text{adv}}=Clip_x^{\varepsilon}\left\{x_t^{\text{adv}}+\alpha\cdot sign\left(\nabla_{x_t^{\text{adv}}}L\left(x_t^{\text{adv}},y\right)\right)\right\} \tag{2}$$

其中, $x_0^{\text{adv}}=x$, T 和 α 分别表示迭代次数和步长, 通常情况下 $\alpha=\varepsilon/T$, $Clip_x^{\varepsilon}\{\cdot\}$ 为裁剪函数, 表示将得到的对抗样本裁剪在以原始样本 x 为球心, 半径为 ε 的球内.

基于动量迭代的快速梯度符号法 (MI-FGSM)[22]将动量项引入到 I-FGSM 中, 用于稳定扰动的更新方向, 其更新过程如下.

$$g_{t+1}=\mu\cdot g_t+\frac{\nabla_{x_t^{\text{adv}}}L(x_t^{\text{adv}},y)}{\left\|\nabla_{x_t^{\text{adv}}}L(x_t^{\text{adv}},y)\right\|_1} \tag{3}$$

$$x_{t+1}^{\text{adv}}=Clip_x^{\varepsilon}\left\{x_t^{\text{adv}}+\alpha\cdot sign(g_{t+1})\right\} \tag{4}$$

其中, $x_0^{\text{adv}}=x$, $g_0=0$, g_t 表示前 t 次迭代中归一化梯度的累加, μ 表示衰减因子.

基于 Nesterov 动量迭代的快速梯度符号法 (NI-FGSM)[23]将 Nesterov 加速梯度策略[32]集成到基于梯度的迭代攻击中, 以提升对抗样本的迁移性, 具体过程如下.

$$x_t^{\text{nes}}=x_t^{\text{adv}}+\alpha\cdot\mu\cdot g_t \tag{5}$$

$$g_{t+1}=\mu\cdot g_t+\frac{\nabla_{x_t^{\text{nes}}}L(x_t^{\text{nes}},y)}{\left\|\nabla_{x_t^{\text{nes}}}L(x_t^{\text{nes}},y)\right\|_1} \tag{6}$$

$$x_{t+1}^{\text{adv}}=Clip_x^{\varepsilon}\left\{x_t^{\text{adv}}+\alpha\cdot sign(g_{t+1})\right\} \tag{7}$$

其中, $x_0^{\text{adv}}=x$, $g_0=0$, x_t^{nes} 是通过对抗样本 x_t^{adv} 和前 t 次迭代中归一化梯度的累加 g_t 得到的样本.

1.2 数据增广方法

数据增广方法的核心是在梯度计算中引入各种输入变换, 以防止生成的对抗样本过拟合模型. 在基于梯度的攻击 (如 FGSM[8]、I-FGSM[12]、MI-FGSM[22]、NI-FGSM[23]) 中引入数据增广方法, 可以提升对抗样本的迁移性. 几种常用的数据增广方法如下.

多样性输入法 (diverse inputs method, DIM)[27]首先以固定的概率对输入样本进行随机调整大小和填充, 然后将转换后的样本作为分类器的输入. 其转换过程如下所示.

$$T\left(x_t^{\text{adv}};p\right)=\begin{cases}T(x_t^{\text{adv}}), & \text{以概率}p\text{执行}\\ x_t^{\text{adv}}, & \text{以概率}1-p\text{执行}\end{cases} \tag{8}$$

其中, $T(\cdot)$ 表示图像的变换操作, p 表示执行变换的概率.

平移不变方法 (translation invariant method, TIM)[28]提出使用一组平移图像来计算梯度. 为了提高执行效率, 这可以近似等价于将未平移图像的梯度与预先定义的核进行卷积来近似计算梯度, 其过程如下所示.

$$\nabla^{*}_{x_t^{\text{adv}}}L\left(x_t^{\text{adv}},y\right)\approx W*\nabla_{x_t^{\text{adv}}}L\left(x_t^{\text{adv}},y\right) \tag{9}$$

其中, W 为一个预定义的核矩阵.

尺度不变方法 (scale invariant method, SIM)[23]提出在每次迭代中对输入样本按一定的比例多次压缩得到多个比例副本, 用于梯度的计算, 其核心思想是在多个比例副本中优化对抗扰动, 如公式 (10) 所示.

$$\arg\max_{x_t^{\text{adv}}}\frac{1}{m}\sum_{i=1}^{m}L\left(S_i\left(x_t^{\text{adv}}\right),y\right) \tag{10}$$

其中, m 表示比例副本的数量, $S_i(x)=x/2^i$ 表示输入样本 x 的比例副本 (比例因子为 $1/2^i$, $0\leqslant i\leqslant m-1$).

2 基于龙格库塔法的对抗攻击方法

在本节中, 我们首先研究基于梯度的对抗攻击与常微分方程的数值解法之间的相似性. 受常微分方程的数值解法中龙格库塔法的思想的启发, 提出一种新的基于龙格库塔法的对抗攻击方法. 最后, 研究本文所提出的算法的可扩展性.

2.1 对抗攻击与数值解法之间的联系

科学研究和工程技术中的许多实际问题都需要求解常微分方程. 然而现有的解析方法只能用于求解一些特殊类型的问题, 实际上许多很有价值的常微分方程不能用初等函数来表示, 通常情况下需要求其数值解. 常微分方程的数值解法是设法将常微分方程离散化, 建立差分方程, 给出解在一些离散点上的近似值[31]. 设 $f(t,u)$ 在区域 $G: a\leqslant t\leqslant b$, $|u|<\infty$ 上连续, 求 $u=u(t)$ 满足方程 (11) 的近似值 u_n.

$$\begin{cases}\dfrac{du}{dt}=f(t,u)\\ u(t_0)=u_0\end{cases} \tag{11}$$

其中, u_0 为给定的初值. 假设常微分方程 (11) 的解 $u=u(t)$ 唯一存在且足够平滑, 求解区域 $[a,b]$ 可以等分为 N 个子区间, 如下所示.

$$a=t_0<t_1<t_2<\ldots<t_n<\ldots<t_N=b \tag{12}$$

其中, $t_n=a+nh$ $(n=0,1,\ldots,N)$, h 称为步长. 常微分方程的数值解法就是求精确解 $u(t)$ 在节点 t_n 上的近似值 u_n (即 $u_n\approx u(t_n)$, $n=0,1,\ldots,N$). 为了得到近似解, 在区间 $[t_n,t_{n+1}]$ 上对方程 (11) 采用数值积分方法来建立差分公式,

则有:

$$u(t_{n+1})-u(t_n)=\int_{t_n}^{t_{n+1}} f(t,u(t))\mathrm{d}t \tag{13}$$

对公式 (13) 中的右侧积分 $\int_{t_n}^{t_{n+1}} f(t,u(t))\mathrm{d}t$ 应用矩形公式, 可以得到如下所示的欧拉法.

$$u(t_{n+1})-u(t_n)=h\cdot f(t_n,u_n) \tag{14}$$

其中, $u(t_0)=u_0$, $n=1,2,\ldots,N-1$. 根据公式 (14) 可以逐步得到各节点上解的近似值 $u_{t_{n+1}}$. 欧拉法的基本思想是采用差商近似导数, 由欧拉法 (即公式 (14)) 可得, $\frac{u(t_{n+1})-u(t_n)}{h}=f(t_n,u_n)$, 其中 $h=t_{n+1}-t_n$. 当且仅当 $h\to 0$ 时, 等式 $\frac{u(t_{n+1})-u(t_n)}{h}=\frac{du_n}{dt_n}$ 成立, 此时求得的近似解与精确解之间的误差为 0.

如前所述, 常微分方程的数值解法是根据初始值 u_0 和导数 $f(t,u)$ 求得满足一定精度的近似解 $u_{t_{n+1}}$ ($n=1,2,\ldots,N-1$). 对于对抗攻击而言, 根据损失函数对于输入样本的梯度 $\nabla_x L(x,y)$ 在原始样本 x 中添加细微的扰动可以得到对抗样本 x^{adv}. 接下来, 我们以 I-FGSM 为例研究对抗样本的生成过程与常微分方程中近似解的求解过程之间的密切关系.

I-FGSM[12]是一种经典的基于梯度的迭代攻击方法, 其核心步骤如公式 (2) 所示, 算法中采用了两个非线性函数, 即 $sign(\cdot)$ 和 $Clip_x^{\varepsilon}\{\cdot\}$, 其中 $sign(\cdot)$ 函数试图在 l_∞ 范数失真的情况下在输入样本中添加更多的扰动, $Clip_x^{\varepsilon}\{\cdot\}$ 函数应用于生成的对抗样本, 以确保所有的像素值都在 [0,255] 的范围内, 避免像素值溢出. 虽然这两个非线性函数不可避免地会影响攻击成功率, 但添加到输入样本中的对抗扰动主要由损失函数相对于输入样本的梯度决定. 为了便于解释, 我们忽略了公式 (2) 中的两个非线性函数 (即 $sign(\cdot)$ 和 $Clip_x^{\varepsilon}\{\cdot\}$). 此时 I-FGSM 可以简化为:

$$\begin{cases}\dfrac{x_{t+1}^{\mathrm{adv}}-x_t^{\mathrm{adv}}}{\alpha}=\nabla_{x_t^{\mathrm{adv}}}L\left(x_t^{\mathrm{adv}},y\right)\\ x_0^{\mathrm{adv}}=x\end{cases} \tag{15}$$

如公式 (15) 所示, $(x_{t+1}^{\mathrm{adv}}-x_t^{\mathrm{adv}})$ 表示相邻两次迭代得到的对抗样本之间的差值, α 表示每次迭代所增加的扰动的幅度. 当 $\alpha\to 0$ 时, 公式 (15) 可以改写为:

$$\begin{cases}\dfrac{\mathrm{d}x_t^{\mathrm{adv}}}{\mathrm{d}t}=\nabla_{x_t^{\mathrm{adv}}}L\left(x_t^{\mathrm{adv}},y\right)\\ x_0^{\mathrm{adv}}=x\end{cases} \tag{16}$$

将公式 (16) 与公式 (11) 进行比较, 可以发现两式具有相同的形式. 从这个角度来看, 对抗样本的生成过程可以看成常微分方程中近似解的求解过程, 二者都是通过梯度或导数来确定求得的对抗样本或近似解, 从而达到彼此的目标. 对抗攻击的目标是生成最优的对抗样本误导模型产生错误的预测, 数值解法的目标是提升近似值的精度, 减少近似值与精确解之间的误差. 因此, 我们将对抗攻击与数值解法联系起来是合理的.

根据公式 (1) 和公式 (14) 也可以看出, 利用 FGSM 生成的对抗样本与利用欧拉法求解常微分方程的近似解的过程几乎相同. 这意味着现有的基于梯度的攻击可以看作是常微分方程中的欧拉方法. 忽略非线性函数 (即 $sign(\cdot)$ 和 $Clip_x^{\varepsilon}\{\cdot\}$), 我们可以把公式 (1) 改写为:

$$x^{\mathrm{adv}}=x+\varepsilon\cdot\nabla_x L(x,y) \tag{17}$$

如果将对抗攻击中的原始样本 x 看成常微分方程中的初始值 u_0, 公式 (17) 中的梯度 $\nabla_x L(x,y)$ 和扰动幅度 ε 分别看作公式 (14) 中的导数 $f(t_0,u_0)$ 和步长 h, 那么 FGSM 生成的对抗样本 x^{adv} 可以视为欧拉法求得的近似解 u_1.

2.2 基于龙格库塔法的对抗攻击

2.2.1 龙格库塔算法

龙格库塔 (Runge-Kutta, RK) 方法是一种在工程上应用广泛的高精度算法. 理论上讲, 只要函数 $u=u(t)$ 在区间 $[a,b]$ 上足够光滑, 那么它的各阶导数值 $u^{(k)}(t_n)$ 与函数 $u(t)$ 在区间 $[a,b]$ 上某些点的值就相互联系. 也就是说, 函数值可以用各阶导数值近似表示, 反之各阶导数值也可以用函数一些点上值的线性组合近似表示. 龙格库塔算法的主要思想是在函数 $u=u(t)$ 中 t 点的附近选取一些特定的点, 然后把这些点的导数值进行线性组合, 构造一类算

法使其按泰勒公式展开后与常微分方程解的泰勒展开式有尽可能多的项相同.

在龙格库塔算法中, 首先计算函数 $u=u(t)$ 在点 $t=u_n$ 处附近的 γ ($\gamma \geqslant 1$) 个点上的导数值, 然后对这些点的导数值进行线性组合, 构造一组近似计算公式, 最后将近似公式与常微分方程解的泰勒展开式进行比较, 使得近似公式前面的若干项与泰勒展开式前面的若干项尽可能地相同, 进而得到一定精度的数值计算公式. 龙格库塔算法一般形式如下.

$$u(t_{n+1})=u(t_n)+h\cdot\sum_{i=1}^{\gamma}\omega_i k_i \tag{18}$$

其中,

$$\begin{cases} k_1=f(t_n,u_n), & i=1 \\ k_i=f\left(t_n+b_i h,u_n+h\sum\limits_{j=1}^{i-1}a_{ij}k_j\right), & i=2,3,\ldots,\gamma \end{cases} \tag{19}$$

其中, h 表示步长, ω_i,b_i,a_{ij} 均为待定系数, 并且 $\sum\limits_{j=1}^{i-1}a_{ij}k_j=b_i$. 适当地选取这些系数, 可以减少求得的近似解与精确解之间的误差. 公式 (18) 称为 γ 级龙格库塔法, 通常情况下点数 γ 越多, 得到的近似解的精度越高. 注意, 在公式 (18) 中, 当 $\gamma=1$ 时, 龙格库塔算法就是欧拉法 (如公式 (14) 所示).

在实际的常微分方程的数值求解中, 为了提升效率, 通常采用三阶或四阶龙格库塔方法计算近似解. 将公式 (18) 中的 γ 设置为 3, ω_1, ω_2 和 ω_3 设置为 1/6, 2/3 和 1/6; $b_2=1/2$, $b_3=1$, $a_{32}=2$, 可以得到如公式 (20) 所示的三阶龙格库塔方法.

$$\begin{cases} u_{n+1}=u_n+h(k_1+4k_2+k_3)/6 \\ k_1=f(t_n,u_n) \\ k_2=f(t_n+h/2,u_n+hk_1/2) \\ k_3=f(t_n+h,u_n-hk_1+2hk_2) \end{cases} \tag{20}$$

其中, h 表示步长. 公式 (20) 是一种常用的三阶龙格库塔方法, 近似解 u_{n+1} ($n=1,2,\ldots,N-1$) 主要是通过将 k_1, k_2 和 k_3 进行线性组合得到的. 根据公式 (20) 可以逐步得到各节点上解的近似值 $u_{t_{n+1}}$.

类似地, 将公式 (18) 中的 γ 设置为 4, ω_1, ω_2, ω_3 和 ω_4 设置为 1/6, 1/3, 1/3 和 1/6; b_2, b_3 和 b_4 设置为 1/2, 1/2 和 1; $a_{32}=1/2$, $a_{43}=1$, 可以得到如下所示的四阶龙格库塔方法.

$$\begin{cases} u_{n+1}=u_n+h(k_1+2k_2+2k_3+k_4)/6 \\ k_1=f(t_n,u_n) \\ k_2=f(t_n+h/2,u_n+hk_1/2) \\ k_3=f(t_n+h/2,u_n+hk_2/2) \\ k_4=f(t_n+h,u_n+hk_3) \end{cases} \tag{21}$$

其中, h 表示步长. 公式 (21) 是一种常用的四阶龙格库塔方法, 近似解 u_{n+1} ($n=1,2,\ldots,N-1$) 主要是通过将 k_1, k_2, k_3 和 k_4 进行线性组合得到的. 根据公式 (21) 可以逐步得到各节点上解的近似值 $u_{t_{n+1}}$.

2.2.2 所提出的方法

如前所述, FGSM 生成的对抗样本可以视为常微分方程数值解法中欧拉法得到的近似解. 然而在常微分方程的数值解法中, 欧拉法求得的近似解的精度较低, 与精确解之间存在较大的误差. 为了提升求得近似解的精度, 通常采用龙格库塔法来求解常微分方程的近似解. 基于此, 受龙格库塔算法中预测思想的启发, 我们将三阶龙格库塔算法的思想引入到 FGSM, 可以得到一种新的基于三阶龙格库塔法的快速梯度符号方法 (即 RK3-FGSM), 其核心过程如下所示.

$$k_1=\nabla_x L(x,y) \tag{22}$$

$$x^{\text{pre\_1}}=x+\varepsilon\cdot sign(k_1/2) \tag{23}$$

$$k_2 = \nabla_{x^{\text{pre\_1}}} L\left(x^{\text{pre\_1}}, y\right) \tag{24}$$

$$x^{\text{pre\_2}} = x + \varepsilon \cdot sign(-k_1 + 2k_2) \tag{25}$$

$$k_3 = \nabla_{x^{\text{pre\_2}}} L\left(x^{\text{pre\_2}}, y\right) \tag{26}$$

$$x^{\text{adv}} = x + \varepsilon \cdot sign((k_1 + 4k_2 + k_3)/6) \tag{27}$$

其中, k_1 表示损失函数对于原始输入样本 x 的梯度, 称为原始梯度, k_2 和 k_3 分别表示损失函数对于预测样本 $x^{\text{pre\_1}}$ 和 $x^{\text{pre\_2}}$ 的梯度, 称为预测梯度. 公式 (23) 中 k_1 的系数为 1/2 , 这与公式 (20) 中第 3 行 k_1 的系数 1/2 相同. 公式 (25) 中 k_1 和 k_2 的系数分别为 –1 和 2, 这与公式 (20) 中第 4 行 k_1 和 k_2 的系数相同. 公式 (27) 中 k_1 , k_2 和 k_3 的系数分别为 1/6 , 4/6 和 1/6 , 这与公式 (20) 第 1 行中 k_1 , k_2 和 k_3 的系数 1/6 , 4/6 和 1/6 相同. 分别化简公式 (23) 和公式 (27), 可得:

$$x^{\text{pre\_1}} = x + \varepsilon \cdot sign(k_1) \tag{28}$$

$$x^{\text{adv}} = x + \varepsilon \cdot sign(k_1 + 4k_2 + k_3) \tag{29}$$

预测样本 (即 $x^{\text{pre\_1}}$, $x^{\text{pre\_2}}$) 和对抗样本 x^{adv} 都是通过在原始样本 x 中添加步长为 ε 的扰动得到的. 具体来说, 预测样本 $x^{\text{pre\_1}}$ 中添加的扰动由一个梯度 (原始梯度 k_1) 决定, 预测样本 $x^{\text{pre\_2}}$ 中添加的扰动由两个梯度 (原始梯度 k_1 和预测梯度 k_2) 确定, 对抗样本 x^{adv} 中添加的扰动由 3 个梯度 (原始梯度 k_1 , 预测梯度 k_2 和 k_3) 确定, 如公式 (25), 公式 (28) 和公式 (29) 所示.

可以注意到, 生成对抗样本 x^{adv} 之前, 我们构建了两个预测样本 $x^{\text{pre\_1}}$ 和 $x^{\text{pre\_2}}$. 由原始梯度 k_1 生成的预测样本 $x^{\text{pre\_1}}$ 实际上就是直接采用 FGSM 得到的对抗样本. 在我们的方法中, 首先将 FGSM 生成的对抗样本作为预测样本 $x^{\text{pre\_1}}$, 并计算损失函数对于预测样本 $x^{\text{pre\_1}}$ 的梯度 k_2 . 然后, 结合原始梯度 k_1 与预测梯度 k_2 , 构建预测样本 $x^{\text{pre\_2}}$. 最后, 计算损失函数对于预测样本 $x^{\text{pre\_2}}$ 的梯度 k_3 , 并将其与预测梯度 k_2 和原始梯度 k_1 进行线性组合, 以确定对抗样本 x^{adv} 中需要添加的扰动. 在 RK3-FGSM 中, 生成对抗样本 x^{adv} 时计算了 3 次梯度, 即 k_1 , k_2 和 k_3 .

实际上, 构建的两个预测样本 $x^{\text{pre\_1}}$ 和 $x^{\text{pre\_2}}$ 是生成的对抗样本 x^{adv} 中附近的两个样本, 因为它们都是在原始样本 x 中添加步长为 ε 的扰动得到的. 预测样本 $x^{\text{pre\_1}}$ 和 $x^{\text{pre\_2}}$ 用于预测未来可能生成的对抗样本 x^{adv} , 损失函数对于预测样本 $x^{\text{pre\_1}}$ 和 $x^{\text{pre\_2}}$ 的梯度 k_2 和 k_3 是对未来的梯度的预测. 也就是说, 在我们的方案中, 采用了未来的一些梯度信息来确定原始输入样本中需要添加的扰动.

类似地, 将四阶龙格库塔算法的思想引入 FGSM 中, 可以得到基于四阶龙格库塔法的 FGSM (即 RK4-FGSM), 其核心过程如下所示.

$$k_1 = \nabla_x L(x, y) \tag{30}$$

$$x^{\text{pre\_1}} = x + \varepsilon \cdot sign(k_1/2) \tag{31}$$

$$k_2 = \nabla_{x^{\text{pre\_1}}} L\left(x^{\text{pre\_1}}, y\right) \tag{32}$$

$$x^{\text{pre\_2}} = x + \varepsilon \cdot sign(k_2/2) \tag{33}$$

$$k_3 = \nabla_{x^{\text{pre\_2}}} L\left(x^{\text{pre\_2}}, y\right) \tag{34}$$

$$x^{\text{pre\_3}} = x + \varepsilon \cdot sign(k_3) \tag{35}$$

$$k_4 = \nabla_{x^{\text{pre\_3}}} L\left(x^{\text{pre\_3}}, y\right) \tag{36}$$

$$x^{\text{adv}} = x + \varepsilon \cdot sign((k_1 + 2k_2 + 2k_3 + k_4)/6) \tag{37}$$

其中, k_1 表示损失函数对于原始输入样本 x 的梯度, 称为原始梯度, k_2 , k_3 和 k_4 分别表示损失函数对于预测样本 $x^{\text{pre\_1}}$, $x^{\text{pre\_2}}$ 和 $x^{\text{pre\_3}}$ 的梯度, 称为预测梯度. 公式 (31) 中 k_1 和公式 (33) 中 k_2 的系数均为 1/2 , 这与公式 (21) 中第 3 行 k_1 和第 4 行 k_2 的系数 1/2 相同. 公式 (35) 中 k_3 的系数为 1, 这与公式 (21) 中第 5 行 k_3 的系数 1 相同. 在公式 (37) 中, k_1 , k_2 , k_3 和 k_4 的系数分别为 1/6 , 2/6 , 2/6 和 1/6 , 这与公式 (21) 第 1 行中 k_1 , k_2 , k_3 和 k_4 的系数 1/6 , 2/6 , 2/6 和 1/6 相同. 对公式 (31), 公式 (33) 和公式 (37) 进行化简, 可得:

$$x^{\text{pre\_1}} = x + \varepsilon \cdot sign(k_1) \tag{38}$$

$$x^{\text{pre\_2}} = x + \varepsilon \cdot sign(k_2) \tag{39}$$

$$x^{\text{adv}} = x + \varepsilon \cdot sign(k_1 + 2k_2 + 2k_3 + k_4) \tag{40}$$

预测样本 (即 $x^{\text{pre\_1}}$, $x^{\text{pre\_2}}$ 和 $x^{\text{pre\_3}}$) 和对抗样本 x^{adv} 都是通过在原始样本 x 中添加步长为 ε 的扰动得到的. 具体来说, 预测样本 $x^{\text{pre\_1}}$ 中添加的扰动由原始梯度 k_1 确定, 预测样本 $x^{\text{pre\_2}}$ 中添加的扰动由预测梯度 k_2 确定, 预测样本 $x^{\text{pre\_3}}$ 中添加的扰动由预测梯度 k_3 确定. 如公式 (40) 所示, 生成的对抗样本 x^{adv} 中所添加的扰动主要是由 4 个梯度 (1 个原始梯度 k_1 和 3 个预测梯度 k_2, k_3, k_4) 进行线性组合得到的. 在基于四阶龙格库塔法的攻击 RK4-FGSM 中, 生成对抗样本 x^{adv} 需要计算 4 次梯度, 即 k_1, k_2, k_3 和 k_4.

最后, 我们对上述 3 种算法 (即 FGSM、RK3-FGSM 和 RK4-FGSM) 的运行时间和存储空间进行分析.

● 运行时间. 对于基于梯度的攻击而言, 通常情况下梯度的计算次数通常是衡量算法运行时间的主要因素, 假设计算一次梯度所需要的时间为 t. 在 FGSM 中, 如公式 (1) 所示, 算法只需要计算一次梯度, 所需时间为 t. 在 RK3-FGSM 中, 如公式 (22)–公式 (27) 所示, 共需计算 3 次梯度, 所需时间为 $3t$. 在 RK4-FGSM 中, 如公式 (30)–公式 (37) 所示, 共需计算 4 次梯度, 所需时间为 $4t$. 可以发现, RK3-FGSM 和 RK4-FGSM 计算梯度所需的运行时间分别为 FGSM 的 3 倍和 4 倍.

● 存储空间. 在 FGSM 生成对抗样本 x^{adv} 的过程中, 除了模型参数外, 主要需要存储两个变量, 即图像矩阵 x 和梯度矩阵 $\nabla_x L(x,y)$, 二者尺寸大小相同, 假设二者所需要的存储空间均为 p, 则总的存储空间为 $2p$. 在 RK3-FGSM 中, 生成对抗样本 x^{adv} 时主要需要存储 4 个变量, 即 x, k_1, k_2 和 k_3, 所需的存储空间为 $4p$. 在 RK4-FGSM 中, 生成对抗样本 x^{adv} 时主要需要存储 5 个空间变量, 即 x, k_1, k_2, k_3 和 k_4, 故 RK4-FGSM 所需的存储空间为 $5p$. 可以发现, 程序运行过程中, RK3-FGSM 和 RK4-FGSM 存储相关变量所需的空间分别为 FGSM 的 2 倍和 5/2 倍.

2.3 可扩展性

如前文所述, 本文提出了两种新的基于龙格库塔法的对抗攻击方法, RK3-FGSM 和 RK4-FGSM. 龙格库塔算法的核心思想可以自然地引入到其他基于 FGSM 的对抗攻击中, 形成一系列新的基于龙格库塔法的对抗攻击. 例如, 将三阶和四阶龙格库塔法与 I-FGSM[12]相结合, 可以得到 RK3-I-FGSM 和 RK4-I-FGSM, 其中 RK4-I-FGSM 的详细过程如算法 1 所示.

算法 1. RK4-I-FGSM.

输入: 真实标签为 y 的原始样本 x, 损失函数为 L 的分类器 F, 扰动量大小 ε, 迭代次数 T;
输出: 对抗样本 x^{adv}.

1. 初始化参数 $\alpha = T/\varepsilon$, $x_0^{\text{adv}} = x$
2. **for** $t = 0$ to $T-1$ **do**
3. 计算损失函数对于原始输入样本 x_t^{adv} 的梯度 $k_t^1 = \nabla_{x_t^{\text{adv}}} L(x_t^{\text{adv}}, y)$
4. 构建预测样本 $x_t^{\text{pre\_1}} = x_t^{\text{adv}} + \alpha \cdot sign(k_t^1)$
5. 计算损失函数对于预测样本 $x_t^{\text{pre\_1}}$ 的梯度 $k_t^2 = \nabla_{x_t^{\text{pre\_1}}} L\left(x_t^{\text{pre\_1}}, y\right)$
6. 构建预测样本 $x_t^{\text{pre\_2}} = x_t^{\text{adv}} + \alpha \cdot sign(k_t^2)$
7. 计算损失函数对于预测样本 $x_t^{\text{pre\_2}}$ 的梯度 $k_t^3 = \nabla_{x_t^{\text{pre\_2}}} L\left(x_t^{\text{pre\_2}}, y\right)$
8. 构建预测样本 $x_t^{\text{pre\_3}} = x_t^{\text{adv}} + \alpha \cdot sign(k_t^3)$
9. 计算损失函数对于预测样本 $x_t^{\text{pre\_3}}$ 的梯度 $k_t^4 = \nabla_{x_t^{\text{pre\_3}}} L\left(x_t^{\text{pre\_3}}, y\right)$
10. 更新对抗样本 $x_{t+1}^{\text{adv}} = Clip_x^{\varepsilon}\{x_t^{\text{adv}} + \alpha \cdot sign(k_t^1 + 2k_t^2 + 2k_t^3 + k_t^4)\}$
11. **end for**
12. **return** $x^{\text{adv}} = x_T^{\text{adv}}$

此外, 本文提出的三阶和四阶龙格库塔策略可以分别与 MI-FGSM[22]和 NI-FGSM[23]结合, 形成 4 种新的对抗攻击方法, 包括两种基于三阶龙格库塔法的攻击 (RK3-MI-FGSM、RK3-NI-FGSM) 和两种基于四阶龙格库塔法的攻击 (RK4-MI-FGSM、RK4-NI-FGSM). 同时, 我们可以在这些攻击中引入数据增广策略 (例如 DIM[27]、TIM[28]和 SIM[23]), 以提升对抗样本的迁移性. 请注意在每次迭代生成对抗样本时, 基于三阶龙格库塔法的对抗攻击有 3 个样本 (即两个预测样本和一个当前样本), 基于四阶龙格库塔法的对抗攻击有 4 个样本 (即 3 个预测样本和 1 个当前样本). 对于每个样本, 都可以获得相应的损失函数, 并根据获得的损失函数计算梯度. 在基于梯度的对抗攻击中, 梯度的计算次数 N 是衡量攻击算法复杂度和运行时间的主要因素. 当攻击算法迭代 T 次生成对抗样本 x_T^{adv} 时, 基于三阶龙格库塔法的对抗攻击中梯度的计算次数为 $N=3T$, 基于四阶龙格库塔法的对抗攻击中梯度的计算次数为 $N=4T$. 在基于龙格库塔法的攻击方法中, 需要保存一些预测样本并计算这些预测样本的梯度, 这可能会增加了算法的运行时间. 然而, 正是由于所提出的算法使用了一些预测样本来确定对抗扰动, 对抗样本的迁移性可以得到进一步的提高.

数据增广策略可以直接应用于上述提到的预测样本和当前样本. 例如, 上述 3 种数据增广策略分别引入基于三阶龙格库塔法的攻击 RK3-NI-FGSM 中, 可以形成 RK3-DI-NI-FGSM、RK3-TI-NI-FGSM 和 RK3-SI-NI-FGSM. 3 种数据增广策略分别引入基于四阶龙格库塔法的攻击 RK4-NI-FGSM 中, 可以得到 RK4-DI-NI-FGSM、RK4-TI-NI-FGSM 和 RK4-SI-NI-FGSM. 融合上述提到的 3 种数据增广策略, 一起引入 RK3-NI-FGSM 和 RK4-NI-FGSM 中, 则可以获得两种强大的对抗攻击方法, 命名为 RK3-STD-MI-FGSM 和 RK4-STD-MI-FGSM.

3 实验分析

3.1 实验设置

本文的所有实验都是在 TensorFlow DNN 计算框架[33]上进行的, 并在 NVIDIA GeForce GTX 1080Ti GPU 上运行. 在实验中, 所有攻击方法采用的损失函数均为交叉熵损失函数. 数据集、模型、基线方法和超参数的设置如下所示.

数据集: 测试数据集由从 ImageNet 验证集[34]中随机选择的 10 000 张图像组成, 这些图像都被预先调整为 299×299×3, 并且几乎所有的图像都可以被下面描述的模型正确地分类.

模型: 在实验中, 我们选择了 17 种模型进行测试, 包括 3 种基于 Transformer 的网络模型, 4 种正常训练的基于卷积神经网络的模型和 10 种防御模型.

- 3 种 Transformer 的模型, ViT-B/8[35], ViT-B/16[35]和 ViT-L/16[35].
- 4 种正常训练的基于卷积神经网络的模型, Inception-v3 (Inc-v3)[36], Inception-v4 (Inc-v4)[37], Inception-ResNet-v2 (IncRes-v2)[37]和 ResNet-v2-152 (Res-152)[38].
- 10 种防御模型包括:
  - 3 种对抗训练的模型, Inc-v3$_{ens3}$[39], Inc-v3$_{ens4}$[39]和 IncRes-v2$_{ens}$[39].
  - NIPS 2017 对抗大赛中排名前 3 的防御模型, 高级表示引导降噪 (HGD 排名第 1)[40], 随机调整大小和填充 (R&P 排名第 2)[41]和排名第 3 的模型 (NIPS-r3) (https://github.com/anlthms/nips-2017/tree/master/mmd).
  - 4 种最近提出来的防御模型, 图像压缩和重构 (ComDefend)[42]、随机平滑 (RS)[43]、特征蒸馏 (FD$_1$)[44]和特征去噪 (FD$_2$)[45].

基线方法: 4 种基于梯度的对抗攻击, FGSM[8]、I-FGSM[12]、MI-FGSM[22]和 NI-FGSM[23], 被选择为基线攻击. 此外, 本文提出的攻击和基线攻击分别与 3 种数据增广方法结合起来进行比较, 包括 DIM[27]、TIM[28]和 SIM[23].

超参数: 每个像素的最大扰动、迭代次数、步长和衰减因子分别设置为 $\varepsilon=16$, $T=10$, $\alpha=\varepsilon/T=1.6$ 和 $\mu=1.0$. 对于 3 种数据增广方法而言, DIM[27]中执行变换的概率 p 设置为 0.5, TIM[28]的核矩阵 W 设置为 15×15 的高斯核, SIM[23]中比例副本的数量 $m=5$.

3.2 单个模型的攻击

在本节中, 我们对单个模型进行攻击以验证本文所提出的方法的有效性, 具体包括基于梯度的攻击方法的对比和基于梯度与数据增广策略组合的攻击方法的对比.

3.2.1 基于梯度的攻击方法的对比

4 种基线方法 (FGSM、I-FGSM、MI-FGSM 和 NI-FGSM) 与本文提出的 8 种基于龙格库塔法的攻击方法进行对比, 包括 4 种基于三阶龙格库塔法的攻击方法 (RK3-FGSM、RK3-I-FGSM、RK3-MI-FGSM 和 RK3-NI-FGSM) 和 4 种基于四阶龙格库塔法的攻击方法 (RK4-FGSM、RK4-I-FGSM、RK4-MI-FGSM 和 RK4-NI-FGSM). 我们采用提到的 12 种方法分别攻击 Inc-v3、Inc-v4、IncRes-v2 和 Res-152 模型, 可以得到 48 组对抗样本. 得到的这些对抗样本对于 7 种模型的攻击成功率如表 1 所示, 包括 4 种正常训练的模型 (Inc-v3、Inc-v4、IncRes-v2 和 Res-152) 和 3 种对抗训练的模型 (Inc-v3$_{ens3}$、Inc-v3$_{ens4}$ 和 IncRes-v2$_{ens}$). 可以观察到, 在白盒攻击中, 基于龙格库塔算法的攻击成功率始终优于基线攻击. 在黑盒攻击正常训练的模型中, 通常情况下, 本文所提出的基于三阶龙格库塔方法的攻击成功率比基线方法的攻击成功率高约 2.59%–27.20%, 基于四阶龙格库塔方法的攻击成功率比基线方法的攻击成功率高约 4.59%–39.80%. 对于黑盒攻击对抗训练的模型 (即防御模型), 本文所提出的基于三阶龙格库塔方法的攻击成功率通常比基线方法的攻击成功率高约 0.13%–15.73%, 基于四阶龙格库塔方法的攻击成功率通常比基线方法的攻击成功率高约 0.01%–20.19%. 实验结果验证了本文所提出的方法的有效性, 并表明该方法可以作为一种有效的策略来提高生成的对抗样本的迁移性.

表 1 基于梯度的攻击方法生成的对抗样本对于 7 个模型的攻击成功率 (%)

模型	攻击方法	Inc-v3	Inc-v4	IncRes-v2	Res-152	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$
Inc-v3	FGSM	65.17*	27.17	25.87	25.52	9.12	8.53	4.03
	RK3-FGSM (Ours)	94.36*	47.02	45.23	35.96	18.52	18.56	3.53
	RK4-FGSM (Ours)	**97.84***	**63.65**	**60.85**	**49.59**	**24.94**	**24.08**	**4.04**
	I-FGSM	99.91*	24.66	19.60	14.76	4.93	3.85	2.35
	RK3-I-FGSM (Ours)	**100.00***	29.72	23.85	17.83	10.33	9.11	2.48
	RK4-I-FGSM (Ours)	99.91*	**34.65**	**27.94**	**20.81**	**11.18**	**9.92**	**2.56**
	MI-FGSM	99.89*	47.87	43.55	36.04	13.07	12.11	6.27
	RK3-MI-FGSM (Ours)	**100.00***	55.24	51.32	40.55	22.91	20.91	6.04
	RK4-MI-FGSM (Ours)	99.92*	**60.29**	**55.45**	**43.66**	**24.74**	**22.59**	**6.63**
	NI-FGSM	99.90*	55.02	51.15	39.71	13.02	11.58	5.94
	RK3-NI-FGSM (Ours)	**100.00***	59.12	54.59	42.76	22.49	20.66	5.88
	RK4-NI-FGSM (Ours)	99.92*	**62.62**	**57.69**	**45.74**	**24.29**	**22.31**	**6.63**
Inc-v4	FGSM	30.65	51.49*	23.74	24.80	9.96	8.93	4.56
	RK3-FGSM (Ours)	57.64	87.45*	47.77	41.78	22.78	20.75	3.99
	RK4-FGSM (Ours)	**70.45**	**94.31***	**61.00**	**54.10**	**29.51**	**27.45**	**5.08**
	I-FGSM	34.94	99.83*	22.84	18.62	5.25	4.62	2.89
	RK3-I-FGSM (Ours)	40.06	**99.98***	26.85	21.71	11.71	**10.66**	2.80
	RK4-I-FGSM (Ours)	**44.19**	99.96*	**30.08**	**23.21**	**12.36**	10.63	**3.14**
	MI-FGSM	60.49	99.82*	49.22	42.58	16.88	14.23	7.79
	RK3-MI-FGSM (Ours)	70.24	**99.96***	58.57	49.18	27.78	24.40	8.14
	RK4-MI-FGSM (Ours)	**73.11**	99.94*	**61.63**	**51.96**	**29.24**	**25.86**	**8.69**
	NI-FGSM	67.16	99.92*	55.09	45.46	15.11	13.40	7.04
	RK3-NI-FGSM (Ours)	72.94	**99.99***	62.06	52.11	28.59	24.87	8.02
	RK4-NI-FGSM (Ours)	**74.67**	99.95*	**63.30**	**52.67**	**30.02**	**26.01**	**8.53**

表 1 基于梯度的攻击方法生成的对抗样本对于 7 个模型的攻击成功率 (%)(续)

模型	攻击方法	Inc-v3	Inc-v4	IncRes-v2	Res-152	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$
IncRes-v2	FGSM	28.06	22.67	41.06*	22.78	10.14	9.10	5.44
	RK3-FGSM (Ours)	55.26	49.27	79.27*	42.79	23.67	21.97	5.57
	RK4-FGSM (Ours)	**65.53**	**60.34**	**87.43***	**52.33**	**30.33**	**26.47**	**6.63**
	I-FGSM	34.82	28.50	97.44*	20.03	6.14	5.16	4.11
	RK3-I-FGSM (Ours)	40.72	33.92	98.86*	22.62	13.45	10.86	4.26
	RK4-I-FGSM (Ours)	**42.16**	**35.06**	**99.28***	**24.93**	**13.57**	**11.03**	**4.45**
	MI-FGSM	60.95	54.09	97.20*	45.15	20.78	17.06	12.29
	RK3-MI-FGSM (Ours)	70.71	63.89	98.56*	52.92	32.63	26.36	13.71
	RK4-MI-FGSM (Ours)	**72.73**	**66.23**	**98.84***	**54.86**	**33.54**	**26.83**	**14.23**
	NI-FGSM	64.46	56.56	98.75*	44.71	18.14	14.71	10.73
	RK3-NI-FGSM (Ours)	74.60	68.38	98.75*	57.00	33.87	27.45	13.98
	RK4-NI-FGSM (Ours)	**75.30**	**69.65**	**99.03***	**57.56**	**34.08**	**27.51**	**14.53**
Res-152	FGSM	37.42	31.59	30.69	75.00*	14.62	12.18	6.81
	RK3-FGSM (Ours)	59.96	54.10	53.06	97.65*	26.09	24.29	5.55
	RK4-FGSM (Ours)	**72.16**	**67.63**	**66.65**	**98.43***	**33.50**	**30.98**	**7.00**
	I-FGSM	32.73	27.70	24.21	98.60*	7.71	6.70	4.27
	RK3-I-FGSM (Ours)	37.16	32.55	27.97	**99.96***	14.84	13.00	4.63
	RK4-I-FGSM (Ours)	**42.10**	**38.34**	**32.82**	98.69*	**16.37**	**14.08**	**5.21**
	MI-FGSM	58.24	53.97	50.61	98.56*	23.86	20.34	13.06
	RK3-MI-FGSM (Ours)	63.95	59.75	56.43	**99.96***	33.70	29.34	12.58
	RK4-MI-FGSM (Ours)	**67.89**	**64.37**	**60.60**	98.68*	**35.92**	**31.12**	**13.85**
	NI-FGSM	65.10	61.01	58.30	98.80*	23.33	19.29	12.70
	RK3-NI-FGSM (Ours)	68.56	64.52	60.98	**99.96***	34.51	29.67	12.93
	RK4-NI-FGSM (Ours)	**70.24**	**66.83**	**63.34**	98.89*	**36.52**	**31.51**	**14.15**

注: *表示白盒攻击

3.2.2 基于梯度与数据增广策略组合的攻击方法的对比

上述提到的 12 种基于梯度的攻击分别与 3 种常用的数据增广策略 (DIM、TIM 和 SIM) 进行组合, 可以形成 36 种攻击方法. 具体来说, 如图 1 所示, 3 种数据增广策略 DIM、TIM 和 SIM 分别引入 4 种基线攻击 (FGSM、I-FGSM、MI-FGSM 和 NI-FGSM) 中, 可以得到 12 种基线攻击方法, 即 DI-FGSM、DI-I-FGSM、DI-MI-FGSM、DI-NI-FGSM、TI-FGSM、TI-I-FGSM、TI-MI-FGSM、TI-NI-FGSM、SI-FGSM、SI-I-FGSM、SI-MI-FGSM 和 SI-NI-FGSM. 如图 2 所示, 3 种数据增广策略 DIM、TIM 和 SIM 分别引入 4 种基于三阶龙格库塔法的攻击方法 (RK3-FGSM、RK3-I-FGSM、RK3-MI-FGSM 和 RK3-NI-FGSM) 中, 可以组成 RK3-DI-FGSM、RK3-DI-I-FGSM、RK3-DI-MI-FGSM、RK3-DI-NI-FGSM、RK3-TI-FGSM、RK3-TI-I-FGSM、RK3-TI-MI-FGSM、RK3-TI-NI-FGSM、RK3-SI-FGSM、RK3-SI-I-FGSM、RK3-SI-MI-FGSM 和 RK3-SI-NI-FGSM 这 12 种攻击方法. 如图 3 所示, 3 种数据增广策略 DIM、TIM 和 SIM 分别引入 4 种基于四阶龙格库塔法的攻击方法 (RK4-FGSM、RK4-I-FGSM、RK4-MI-FGSM 和 RK4-NI-FGSM) 中, 可以形成 12 种攻击方法, 包括 RK4-DI-FGSM、RK4-DI-I-FGSM、RK4-DI-MI-FGSM、RK4-DI-NI-FGSM、RK4-TI-FGSM、RK4-TI-I-FGSM、RK4-TI-MI-FGSM、RK4-TI-NI-FGSM、RK4-SI-FGSM、RK4-SI-I-FGSM、RK4-SI-MI-FGSM 和 RK4-SI-NI-FGSM.

我们分别采用上述基于梯度与数据增广策略组合得到的 36 种方法攻击 Inc-v3 模型生成对抗样本. 得到的对抗样本对 Inc-v3、Inc-v4、IncRes-v2、Res-152、Inc-v3$_{ens3}$、Inc-v3$_{ens4}$ 和 IncRes-v2$_{ens}$ 这 7 种模型的攻击成功率如表 2 所示. 可以观察到, 在白盒攻击中, 相比于基线方法, 本文所提出的方法的攻击成功率更接近 100%. 在黑盒攻击中, 对于正常训练的模型 (Inc-v4、IncRes-v2 和 Res-152), 本文所提出的基于三阶龙格库塔方法的攻击成功率比

基线方法的攻击成功率高约 1.60%–20.29%, 基于四阶龙格库塔方法的攻击成功率比基线方法的攻击成功率高约 1.49%–30.98%; 对于防御模型 (Inc-v3$_{ens3}$、Inc-v3$_{ens4}$ 和 IncRes-v2$_{ens}$), 本文所提出的基于三阶龙格库塔方法的攻击成功率通常比基线方法的攻击成功率高约 0.32%–19.52%, 基于四阶龙格库塔方法的攻击成功率通常比基线方法的攻击成功率高约 0.50%–22.64%, 这验证了本文所提出的方法的有效性.

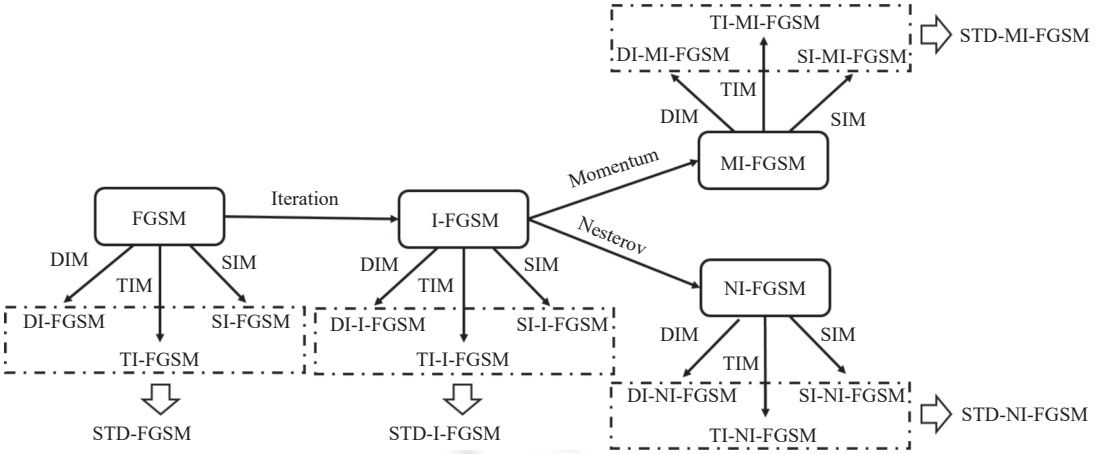


图 1　4 种基线方法与数据增广策略的组合

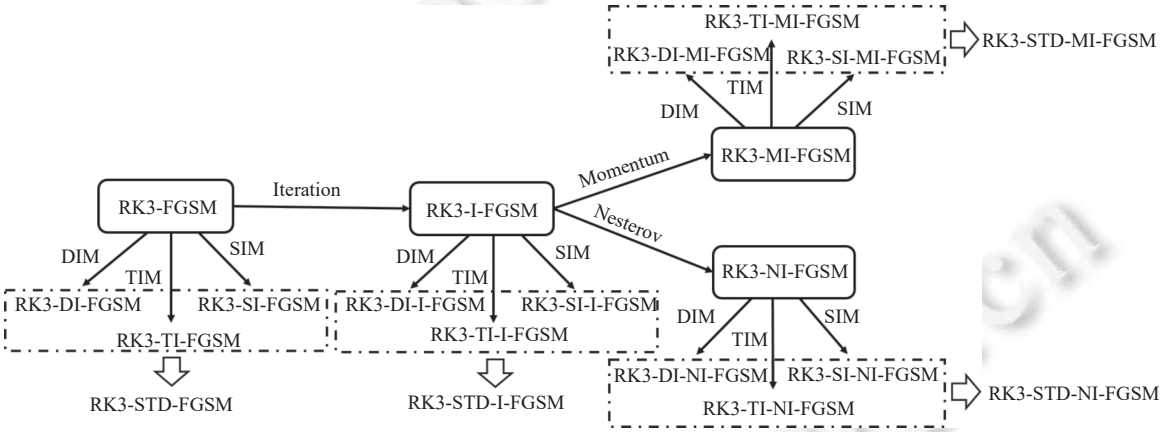


图 2　4 种基于三阶龙格库塔法的攻击方法与数据增广策略的组合

接下来, 将 3 种数据增广方法 DIM、TIM 和 SIM 结合起来, 一起引入 12 种对抗攻击 (FGSM、I-FGSM、MI-FGSM、NI-FGSM、RK3-FGSM、RK3-I-FGSM、RK3-MI-FGSM、RK3-NI-FGSM、RK4-FGSM、RK4-I-FGSM、RK4-MI-FGSM 和 RK4-NI-FGSM), 可以组成 STD-FGSM、STD-I-FGSM、STD-MI-FGSM、STD-NI-FGSM、RK3-STD-FGSM、RK3-STD-I-FGSM、RK3-STD-MI-FGSM、RK3-STD-NI-FGSM、RK4-STD-FGSM、RK4-STD-I-FGSM、RK4-STD-MI-FGSM 和 RK4-STD-NI-FGSM 这 12 种强大的攻击方法, 如图 1–图 3 所示. 采用这 12 种方法分别攻击 Inc-v3, 可以得到 12 组对抗样本. 得到的对抗样本对 Inc-v3、Inc-v4、IncRes-v2、Res-152、Inc-v3$_{ens3}$、Inc-v3$_{ens4}$ 和 IncRes-v2$_{ens}$ 的攻击成功率如表 3 所示. 从表 3 中可以观察到, 在白盒攻击中, 相比于基线攻击 (STD-FGSM、STD-I-FGSM、STD-MI-FGSM 和 STD-NI-FGSM), 本文所提出的方法的攻击成功率更接近 100%. 在黑盒攻击中, 对于正常训练的模型 (Inc-v4、IncRes-v2 和 Res-152), 本文所提出的基于三阶龙格库塔方法 (RK3-STD-FGSM、RK3-STD-I-FGSM、RK3-STD-MI-FGSM 和 RK3-STD-NI-FGSM) 的攻击成功率比基线方法的攻击成功率高约 3.21%–13.17%, 基于四阶龙格库塔方法 (RK4-STD-FGSM、RK4-STD-I-FGSM、RK4-STD-MI-FGSM 和 RK4-STD-NI-FGSM) 的攻击成功率比基线方法的攻击成功率高约 6.30%–23.20%; 对于防御模型 (Inc-

$v3_{ens3}$、Inc-$v3_{ens4}$ 和 IncRes-$v2_{ens}$), 本文所提出的基于三阶龙格库塔方法的攻击成功率比基线方法的攻击成功率高约 3.02%–13.81%, 基于四阶龙格库塔方法的攻击成功率比基线方法的攻击成功率高约 6.10%–17.30%.

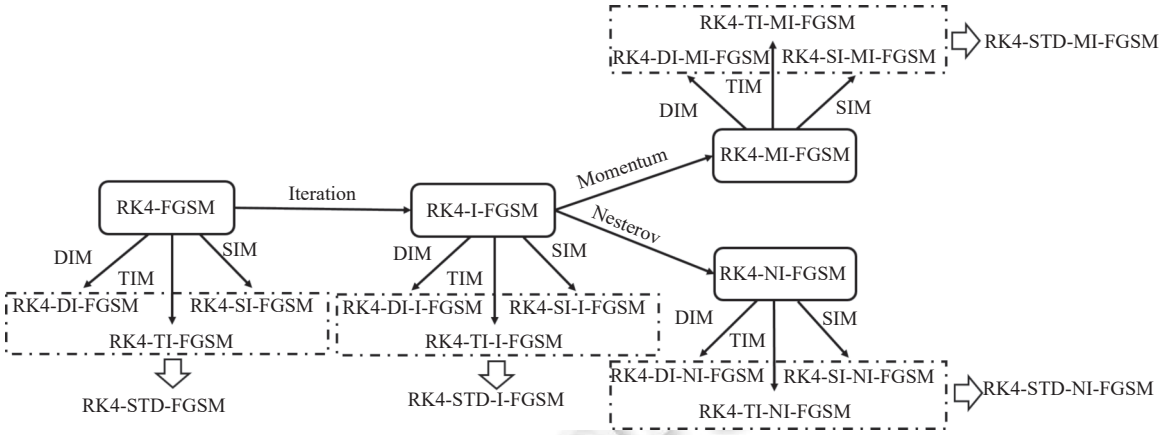


图 3　4 种基于四阶龙格库塔法的攻击方法与数据增广策略的组合

表 2　基于梯度与数据增广策略组合的方法生成的对抗样本对于 7 个模型的攻击成功率 (%)

组合方式	攻击方法	Inc-v3	Inc-v4	IncRes-v2	Res-152	Inc-$v3_{ens3}$	Inc-$v3_{ens4}$	IncRes-$v2_{ens}$
与数据增广DIM的组合	DI-FGSM	56.89*	25.33	23.86	24.73	17.51	17.64	3.91
	RK3-DI-FGSM (Ours)	88.63*	45.62	42.60	36.48	19.10	18.95	3.71
	RK4-DI-FGSM (Ours)	**95.04***	**63.31**	**60.15**	**50.57**	**27.75**	**26.20**	**4.41**
	DI-I-FGSM	99.80*	43.13	33.66	25.50	13.25	12.02	3.54
	RK3-DI-I-FGSM (Ours)	**99.98***	52.27	42.15	30.89	14.79	13.55	3.86
	RK4-DI-I-FGSM (Ours)	99.96*	**57.73**	**46.81**	**35.22**	**15.96**	**14.27**	**4.31**
	DI-MI-FGSM	99.73*	67.99	62.97	51.80	17.39	16.32	8.17
	RK3-DI-MI-FGSM (Ours)	**100.00***	76.73	71.66	58.90	33.61	30.26	8.66
	RK4-DI-MI-FGSM (Ours)	99.91*	**80.07**	**74.87**	**62.27**	**35.63**	**32.54**	**9.80**
	DI-NI-FGSM	99.92*	64.90	59.80	46.88	13.72	13.01	6.53
	RK3-DI-NI-FGSM (Ours)	**99.98***	79.05	73.58	61.15	33.24	30.72	8.39
	RK4-DI-NI-FGSM (Ours)	99.92*	**82.17**	**77.12**	**64.19**	**36.36**	**33.25**	**9.69**
与数据增广TIM的组合	TI-FGSM	54.20*	29.88	23.98	28.33	22.58	22.63	15.75
	RK3-TI-FGSM (Ours)	80.58*	39.24	31.14	34.47	28.85	29.63	21.93
	RK4-TI-FGSM (Ours)	**89.98***	**47.56**	**39.28**	**40.91**	**37.21**	**37.73**	**28.56**
	TI-I-FGSM	99.73*	19.89	13.84	12.25	13.56	13.21	5.43
	RK3-TI-I-FGSM (Ours)	**99.95***	24.06	16.73	14.68	15.07	14.95	7.07
	RK4-TI-I-FGSM (Ours)	99.91*	**30.85**	**21.90**	**18.65**	**18.01**	**17.48**	**8.75**
	TI-MI-FGSM	99.72*	40.24	33.58	31.49	29.42	28.61	20.78
	RK3-TI-MI-FGSM (Ours)	99.85*	45.34	37.73	34.05	32.32	32.92	22.39
	RK4-TI-MI-FGSM (Ours)	**99.89***	**54.82**	**46.15**	**40.59**	**39.35**	**39.42**	**27.86**
	TI-NI-FGSM	99.84*	47.13	38.49	35.31	32.32	30.99	23.09
	RK3-TI-NI-FGSM (Ours)	**99.98***	54.55	45.08	39.84	37.50	37.25	26.80
	RK4-TI-NI-FGSM (Ours)	99.91*	**56.75**	**47.25**	**42.05**	**39.71**	**39.71**	**28.39**

表 2 基于梯度与数据增广策略组合的方法生成的对抗样本对于 7 个模型的攻击成功率 (%)(续)

组合方式	攻击方法	Inc-v3	Inc-v4	IncRes-v2	Res-152	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$
与数据增广SIM的组合	SI-FGSM	80.73*	49.84	46.67	48.19	35.69	35.99	**11.24**
	RK3-SI-FGSM (Ours)	97.92*	68.74	65.54	59.79	30.44	30.85	6.53
	RK4-SI-FGSM (Ours)	**99.48***	**87.50**	**84.45**	**77.62**	**42.28**	**42.73**	9.33
	SI-I-FGSM	99.95*	38.28	33.41	29.29	17.63	16.13	5.54
	RK3-SI-I-FGSM (Ours)	**100.00***	45.73	40.46	34.42	19.70	18.30	6.76
	RK4-SI-I-FGSM (Ours)	99.99*	**52.61**	**46.53**	**39.03**	**21.33**	**20.00**	**7.69**
	SI-MI-FGSM	99.95*	70.34	66.17	59.95	40.27	38.07	17.45
	RK3-SI-MI-FGSM (Ours)	**99.99***	74.36	69.83	62.55	41.15	38.80	17.43
	RK4-SI-MI-FGSM (Ours)	**99.99***	**76.85**	**72.29**	**64.50**	**41.78**	**39.60**	**17.86**
	SI-NI-FGSM	99.99*	77.64	73.03	67.09	39.76	37.99	17.92
	RK3-SI-NI-FGSM (Ours)	**100.00***	80.65	76.16	**68.69**	44.08	42.30	18.81
	RK4-SI-NI-FGSM (Ours)	99.99*	**81.04**	**77.09**	68.58	**44.96**	**42.47**	**19.04**

注: *表示白盒攻击

表 3 生成的对抗样本对于 7 个模型的攻击成功率 (%)

攻击方法	Inc-v3	Inc-v4	IncRes-v2	Res-152	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$
STD-FGSM	63.46*	40.80	34.29	39.05	36.09	36.45	28.92
RK3-STD-FGSM (Ours)	83.34*	50.10	41.83	45.09	39.41	40.70	31.46
RK4-STD-FGSM (Ours)	**94.36***	**63.95**	**54.36**	**55.66**	**52.32**	**53.74**	**44.26**
STD-I-FGSM	99.17*	45.65	33.09	30.90	31.92	32.19	19.60
RK3-STD-I-FGSM (Ours)	99.75*	50.63	37.57	34.68	35.55	35.42	23.20
RK4-STD-I-FGSM (Ours)	**99.92***	**59.28**	**44.90**	**40.58**	**40.86**	**41.12**	**27.75**
STD-MI-FGSM	98.79*	69.08	59.63	56.58	58.50	59.24	47.56
RK3-STD-MI-FGSM (Ours)	99.70*	73.63	63.83	59.79	62.19	63.12	50.58
RK4-STD-MI-FGSM (Ours)	**99.82***	**76.62**	**67.12**	**62.83**	**65.00**	**65.33**	**53.71**
STD-NI-FGSM	99.77*	65.80	55.06	53.22	52.27	52.45	43.04
RK3-STD-NI-FGSM (Ours)	99.79*	77.84	68.23	63.63	65.57	66.26	54.99
RK4-STD-NI-FGSM (Ours)	**99.88***	**78.86**	**69.37**	**64.40**	**66.13**	**67.00**	**55.54**

注: 对抗样本是通过攻击Inc-v3模型得到的, *表示白盒攻击

3.3 融合模型的攻击

遵循 Dong 等人[22]提出的攻击融合模型方法, 我们分别使用 STD-FGSM、STD-I-FGSM、STD-MI-FGSM、STD-NI-FGSM、RK3-STD-FGSM、RK3-STD-I-FGSM、RK3-STD-MI-FGSM、RK3-STD-NI-FGSM、RK4-STD-FGSM、RK4-STD-I-FGSM、RK4-STD-MI-FGSM 和 RK4-STD-NI-FGSM 方法攻击融合模型, 该融合模型是由具有相同权重的 Inc-v3 和 Inc-v4 组成. 然后, 我们可以得到 12 组对抗样本. 生成的对抗样本对 Inc-v3、Inc-v4、IncRes-v2、Res-152、Inc-v3$_{ens3}$、Inc-v3$_{ens4}$ 和 IncRes-v2$_{ens}$ 的攻击成功率如表 4 所示. 可以观察到, 在白盒攻击中, 即对模型 (Inc-v3 和 Inc-v4) 的攻击中, 相比于基线攻击 (STD-FGSM、STD-I-FGSM、STD-MI-FGSM 和 STD-NI-FGSM), 我们的方法可以达到接近 100% 的攻击成功率. 在黑盒攻击中, 对于正常训练的模型 (IncRes-v2 和 Res-152), 本文所提出的基于三阶龙格库塔方法 (RK3-STD-FGSM、RK3-STD-I-FGSM、RK3-STD-MI-FGSM 和 RK3-STD-NI-FGSM) 的攻击成功率比基线方法的攻击成功率高约 3.69%–16.04%, 基于四阶龙格库塔方法 (RK4-STD-FGSM、RK4-STD-I-FGSM、RK4-STD-MI-FGSM 和 RK4-STD-NI-FGSM) 的攻击成功率比基线方法的攻击成功率高约 7.40%–30.60%; 对于防御模型 (Inc-v3$_{ens3}$、Inc-v3$_{ens4}$ 和 IncRes-v2$_{ens}$), 本文所提出的基于三阶龙格库塔方法的攻击成功率比基线方法的攻击成功率高约 3.30%–14.04%, 基于四阶龙格库塔方法的攻击成功率比基线方法的攻击成功率高约 6.50%–24.90%.

表 4 生成的对抗样本对于 7 个模型的攻击成功率 (%)

攻击方法	Inc-v3	Inc-v4	IncRes-v2	Res-152	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$
STD-FGSM	63.75*	55.77*	40.54	44.32	42.70	43.79	36.66
RK3-STD-FGSM (Ours)	85.67*	83.91*	56.58	56.20	54.28	55.62	47.22
RK4-STD-FGSM (Ours)	**94.14***	**93.74***	**71.17**	**68.15**	**67.64**	**68.47**	**61.50**
STD-I-FGSM	98.68*	96.86*	61.12	52.62	53.27	53.45	40.43
RK3-STD-I-FGSM (Ours)	99.64*	99.32*	64.83	56.31	57.21	56.75	44.42
RK4-STD-I-FGSM (Ours)	**99.88***	**99.69***	**72.05**	**62.71**	**61.73**	**62.04**	**50.99**
STD-MI-FGSM	97.88*	95.54*	78.13	71.87	74.80	75.01	68.37
RK3-STD-MI-FGSM (Ours)	99.42*	98.76*	83.37	76.23	80.08	79.09	72.18
RK4-STD-MI-FGSM (Ours)	**99.59***	**99.33***	**86.19**	**79.28**	**82.34**	**82.26**	**74.84**
STD-NI-FGSM	99.49*	99.18*	75.50	69.77	69.70	69.55	62.73
RK3-STD-NI-FGSM (Ours)	99.68*	99.37*	87.18	80.46	83.01	82.30	76.77
RK4-STD-NI-FGSM (Ours)	**99.78***	**99.63***	**88.11**	**81.28**	**83.79**	**83.30**	**77.31**

注: 对抗样本是通过攻击由Inc-v3和Inc-v4模型组成的融合模型得到的, *表示白盒攻击

然后, 我们采用上述提到的 12 种方法生成的对抗样本在 7 种先进的防御模型中进行测试, 包括 HGD、R&P、NIPS-r3、ComDefend、RS、FD_1 和 FD_2. 实验结果如表 5 所示, 可以观察到, 本文所提出的基于三阶龙格库塔方法 (RK3-STD-FGSM、RK3-STD-I-FGSM、RK3-STD-MI-FGSM 和 RK3-STD-NI-FGSM) 的攻击成功率比基线方法 (STD-FGSM、STD-I-FGSM、STD-MI-FGSM 和 STD-NI-FGSM) 的攻击成功率高约 0.01%–14.09%, 基于四阶龙格库塔方法 (RK4-STD-FGSM、RK4-STD-I-FGSM、RK4-STD-MI-FGSM 和 RK4-STD-NI-FGSM) 的攻击成功率比基线方法的攻击成功率高约 0.20%–26.50%. 实验结果表明, 本文所提出的方法在防御模型中可以达到较高的攻击成功率.

表 5 生成的对抗样本对于 7 个先进的防御模型的攻击成功率 (%)

攻击方法	HGD	R&P	NIPS-r3	ComDefend	RS	FD1	FD2
STD-FGSM	33.49	35.14	37.78	48.43	52.85	49.78	19.64
RK3-STD-FGSM (Ours)	44.91	46.30	50.25	60.70	63.19	63.87	19.69
RK4-STD-FGSM (Ours)	**60.00**	**60.60**	**64.90**	**71.62**	**70.44**	**75.58**	**21.05**
STD-I-FGSM	44.80	37.98	43.22	53.81	33.47	63.09	17.03
RK3-STD-I-FGSM (Ours)	49.44	41.94	47.47	57.19	35.48	66.33	17.04
RK4-STD-I-FGSM (Ours)	**57.10**	**48.55**	**53.99**	**61.59**	**37.83**	**70.55**	**17.20**
STD-MI-FGSM	71.00	66.04	69.72	75.37	60.87	79.86	19.99
RK3-STD-MI-FGSM (Ours)	75.75	70.29	74.09	80.09	64.11	84.44	**20.04**
RK4-STD-MI-FGSM (Ours)	**79.03**	**73.26**	**77.09**	**82.13**	**64.45**	**86.20**	19.83
STD-NI-FGSM	66.48	60.80	64.68	71.12	57.21	76.40	19.20
RK3-STD-NI-FGSM (Ours)	80.04	74.48	78.21	82.98	66.41	86.91	**20.19**
RK4-STD-NI-FGSM (Ours)	**81.58**	**75.56**	**79.31**	**83.64**	**66.55**	**87.74**	19.94

注: 对抗样本是通过攻击由Inc-v3和Inc-v4模型组成的融合模型得到的

最后, 我们将上述 12 种方法生成的对抗样本在 3 种 Transformer 模型中进行测试, 包括 ViT-B/8, ViT-B/16 和 ViT-L/16. 实验结果如表 6 所示, 可以观察到, 本文所提出的基于三阶龙格库塔方法 (RK3-STD-FGSM、RK3-STD-I-FGSM、RK3-STD-MI-FGSM 和 RK3-STD-NI-FGSM) 的攻击成功率比基线方法 (STD-FGSM、STD-I-FGSM、STD-MI-FGSM 和 STD-NI-FGSM) 的攻击成功率高约 0.26%–9.54%, 基于四阶龙格库塔方法 (RK4-STD-FGSM、RK4-STD-I-FGSM、RK4-STD-MI-FGSM 和 RK4-STD-NI-FGSM) 的攻击成功率比基线方法的攻击成功率高约

2.33%–11.56%. 实验结果表明, 相比于基线方法, 本文所提出的方法生成的对抗样本在 Transformer 模型中可以达到较高的攻击成功率.

表 6　生成的对抗样本对于 3 种 Transformer 模型的攻击成功率 (%)

攻击方法	ViT-B/8	ViT-B/16	ViT-L/16
STD-FGSM	12.49	17.23	13.55
RK3-STD-FGSM (Ours)	13.11	20.89	14.56
RK4-STD-FGSM (Ours)	**18.13**	**28.79**	**21.34**
STD-I-FGSM	14.56	20.38	14.02
RK3-STD-I-FGSM (Ours)	15.92	22.64	15.70
RK4-STD-I-FGSM (Ours)	**18.45**	**25.91**	**18.15**
STD-MI-FGSM	26.71	36.39	27.93
RK3-STD-MI-FGSM (Ours)	28.03	39.35	29.44
RK4-STD-MI-FGSM (Ours)	**29.04**	**40.47**	**30.71**
STD-NI-FGSM	23.48	32.35	24.31
RK3-STD-NI-FGSM (Ours)	29.53	41.89	31.67
RK4-STD-NI-FGSM (Ours)	**30.26**	**41.97**	**31.80**

注: 对抗样本是通过攻击由Inc-v3和Inc-v4模型组成的融合模型得到的

3.4 讨　论

在本节中, 我们首先评估生成的对抗样本的视觉质量, 然后讨论迭代次数对算法攻击成功率的影响. 最后, 在相同的梯度计算次数下, 综合比较基线方法 (I-FGSM、MI-FGSM 和 NI-FGSM) 和本文所提出方法 (RK3-I-FGSM、RK3-MI-FGSM、RK3-NI-FGSM、RK4-I-FGSM、RK4-MI-FGSM 和 RK4-NI-FGSM) 的运行时间和攻击成功率.

3.4.1　视觉质量的评估

在本节中, 我们比较 4 种基线方法 (FGSM、I-FGSM、MI-FGSM 和 NI-FGSM) 和本文所提出的 8 种攻击方法生成的对抗样本的视觉质量, 其中本文所提出的 8 种攻击方法包括 4 种基于三阶龙格库塔法的攻击方法 (RK3-FGSM、RK3-I-FGSM、RK3-MI-FGSM 和 RK3-NI-FGSM) 和 4 种基于四阶龙格库塔法的攻击方法 (RK4-FGSM、RK4-I-FGSM、RK4-MI-FGSM 和 RK4-NI-FGSM). 两种图像质量评估指标, 峰值信噪比 (peak signal to noise ratio, PSNR) 和结构相似性 (structural similarity, SSIM) 被用来评估对抗样本的视觉质量. 每个像素的最大扰动、迭代次数、步长和衰减因子分别设置为 $\varepsilon = 16$, $T = 10$, $\alpha = \varepsilon/T = 1.6$ 和 $\mu = 1.0$. 采用上述 12 种方法分别攻击 Inc-v3 模型可以得到 12 组对抗样本. 得到的对抗样本与它们对应的原始样本之间的 PSNR 和 SSIM 的平均值如后文表 7 所示. 从表 7 中可以发现, 本文所提出的基于龙格库塔法的对抗攻击的 PSNR 和 SSIM 值与基线攻击的 PSNR 和 SSIM 值非常相近. 此外, 我们随机选择了一些通过 FGSM、RK3-FGSM 和 RK4-FGSM 生成的对抗样本进行可视化. 6 对随机选择的原始样本和对应的对抗样本如后文图 4 所示. 在图 4 中, 第 1 行表示原始样本, 第 2–4 行分别表示通过 FGSM、RK3-FGSM 和 RK4-FGSM 得到的对抗样本, 其中对抗样本下方的数字表示对抗样本和相应的原始样本之间的 PSNR 和 SSIM 值. 从图 4 中可以观察到, 对抗样本中所添加的扰动很难被人眼察觉.

3.4.2　迭代次数

在相同的迭代次数下, 我们将 3 种迭代方法 (I-FGSM、MI-FGSM 和 NI-FGSM) 和本文所提出的 6 种迭代方法 (RK3-I-FGSM、RK3-MI-FGSM、RK3-NI-FGSM、RK4-I-FGSM、RK4-MI-FGSM 和 RK4-NI-FGSM) 的攻击成功率进行比较. 每个像素的最大扰动和步长设置为 $\varepsilon = 16$, $\alpha = \varepsilon/T$.

首先, 在相同的迭代次数下, 将基线方法 I-FGSM 与所提出的方法 (即 RK3-I-FGSM 和 RK4-I-FGSM) 的攻击成功率进行对比. 迭代次数设置为 $T = 1, 2, \ldots, 10$, 分别采用 I-FGSM、RK3-I-FGSM 和 RK4-I-FGSM 攻击 Inc-v3 模型, 生成对抗样本. 生成的对抗样本对于 4 个正常训练的模型 (Inc-v3、Inc-v4、IncRes-v2 和 Res-152) 的攻击成

功率如图 5 所示. 在图 5 中, 横轴和纵轴分别表示迭代次数和攻击成功率, 标签“模型 vs. 方法”表示对抗样本 (由“方法”生成的) 对于“模型”的攻击成功率. 可以观察到, 在白盒模型 Inc-v3 中, I-FGSM、RK3-I-FGSM 和 RK4-I-FGSM 始终保持较高的攻击成功率. 在黑盒模型 (即 Inc-v4、IncRes-v2 和 Res-152) 中, 当迭代次数约为 2 时, I-FGSM 可以达到最高的黑盒攻击成功率; 当迭代次数约为 1 时, RK3-I-FGSM 和 RK4-I-FGSM 可以达到最高的黑盒攻击成功率. 随着迭代次数的增加, I-FGSM 的黑盒攻击成功率先上升后下降, RK3-I-FGSM 和 RK4-I-FGSM 的黑盒攻击成功率逐渐下降. 实验结果表明, 较大的迭代次数并不一定可以提升基线方法 I-FGSM 与所提出的方法 (即 RK3-I-FGSM 和 RK4-I-FGSM) 的黑盒攻击成功率. 此外, 从图 5 中可以发现, 在相同的迭代次数下, 本文所提出的 RK3-I-FGSM 和 RK4-I-FGSM 的攻击成功率通常高于 I-FGSM 的攻击成功率.

表 7　生成的对抗样本的视觉质量评估

攻击方法	PSNR (dB)	SSIM
FGSM	**24.17**	**0.54**
RK3-FGSM (Ours)	**24.17**	**0.54**
RK4-FGSM (Ours)	**24.17**	**0.54**
I-FGSM	**33.68**	**0.88**
RK3-I-FGSM (Ours)	33.36	0.87
RK4-I-FGSM (Ours)	33.11	0.87
MI-FGSM	26.92	**0.66**
RK3-MI-FGSM (Ours)	26.95	**0.66**
RK4-MI-FGSM (Ours)	**27.00**	**0.66**
NI-FGSM	26.91	**0.66**
RK3-NI-FGSM (Ours)	26.93	**0.66**
RK4-NI-FGSM (Ours)	**26.95**	**0.66**

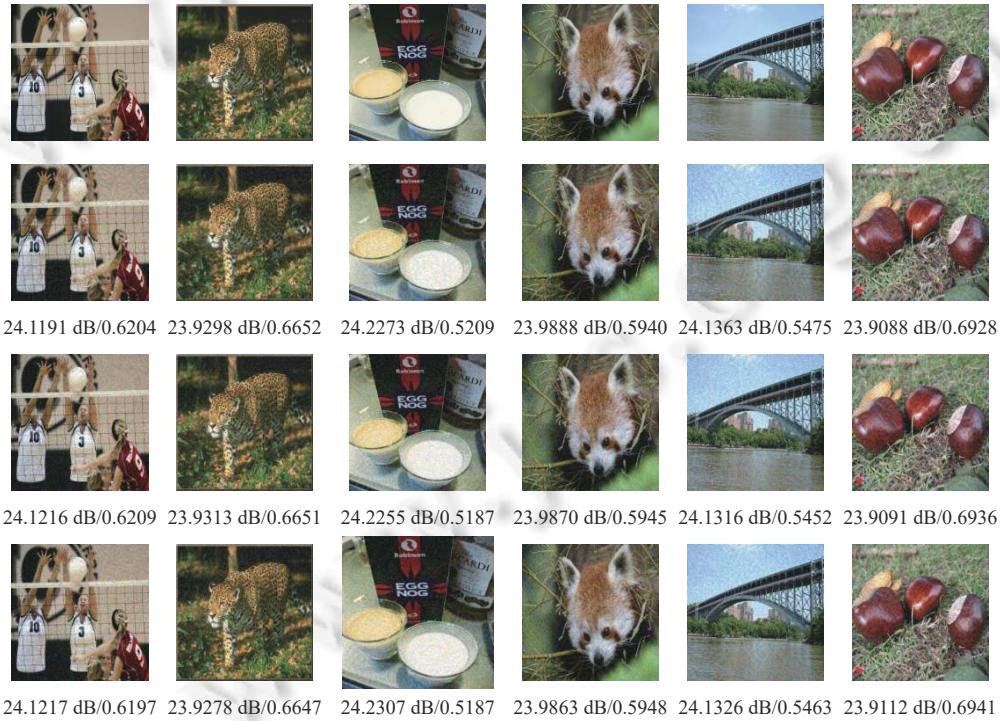
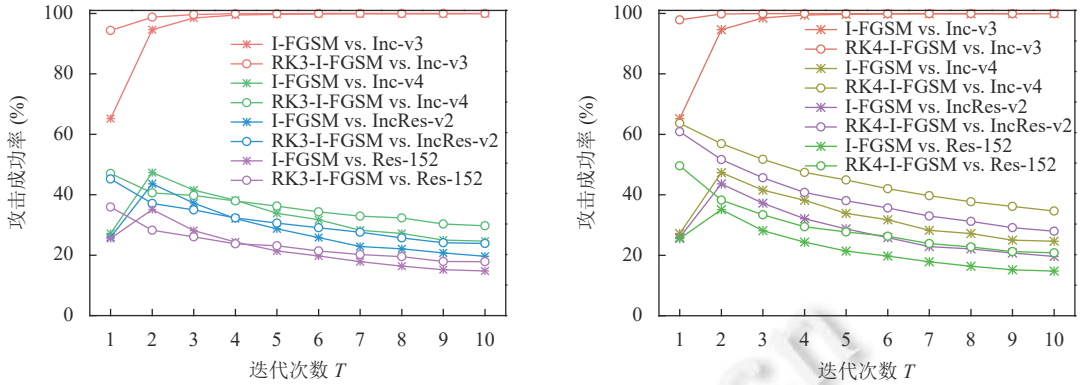


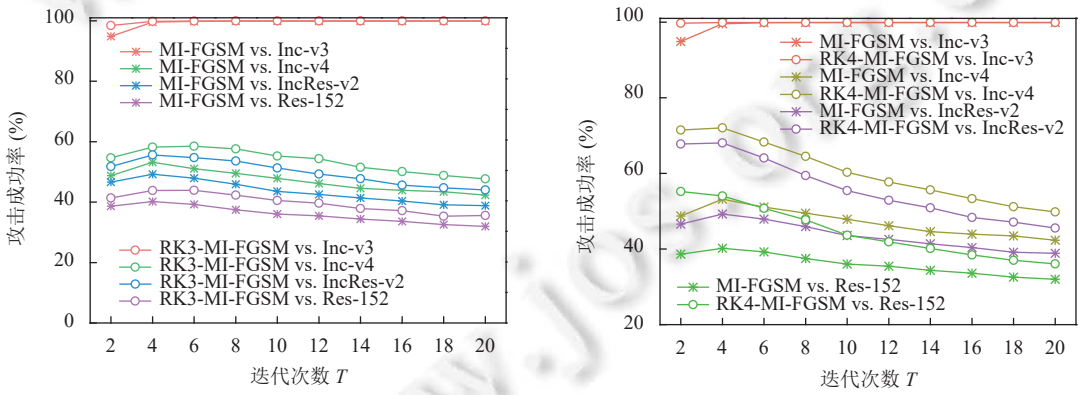
图 4　生成的对抗样本与对应的原始样本的可视化 (PSNR/SSIM)



(a) I-FGSM 和 RK3-I-FGSM 的对比　　(b) I-FGSM 和 RK4-I-FGSM 的对比

图 5　在相同的迭代次数下, I-FGSM、RK3-I-FGSM 和 RK4-I-FGSM 的攻击成功率

然后, 在相同的迭代次数下, 将基线方法 MI-FGSM 与所提出的方法 (即 RK3-MI-FGSM 和 RK4-MI-FGSM) 的攻击成功率进行对比. 使用这 3 种方法攻击 Inc-v3 模型生成对抗样本, 其中迭代次数 T 设置为从 2 到 20, 增量为 2. 生成的对抗样本对于 4 个正常训练的模型 (Inc-v3、Inc-v4、IncRes-v2 和 Res-152) 的攻击成功率如图 6 所示, 其中横轴和纵轴分别表示迭代次数和攻击成功率, 标签“模型 vs. 方法”表示对抗样本 (由“方法”生成的) 对于“模型”的攻击成功率. 可以观察到, 在白盒模型 Inc-v3 中, MI-FGSM、RK3-MI-FGSM 和 RK4-MI-FGSM 始终保持较高的攻击成功率. 在黑盒模型 (即 Inc-v4、IncRes-v2 和 Res-152) 中, 当迭代次数约为 4 时, MI-FGSM、RK3-I-FGSM 和 RK4-I-FGSM 通常可以达到最大的黑盒攻击成功率. 随着迭代次数的增加, MI-FGSM、RK3-I-FGSM 和 RK4-I-FGSM 的黑盒攻击成功率都是先上升后下降, 增加迭代次数并不一定可以提升黑盒攻击成功率. 较大的迭代次数可能会导致攻击 Inc-v3 模型生成的对抗样本对该模型过拟合, 迁移性降低, 进而在黑盒模型中的攻击成功率下降. 此外, 从图 6 中可以发现, 在相同的迭代次数下, 本文所提出的 RK3-MI-FGSM 和 RK4-MI-FGSM 的攻击成功率通常高于 MI-FGSM 的攻击成功率.

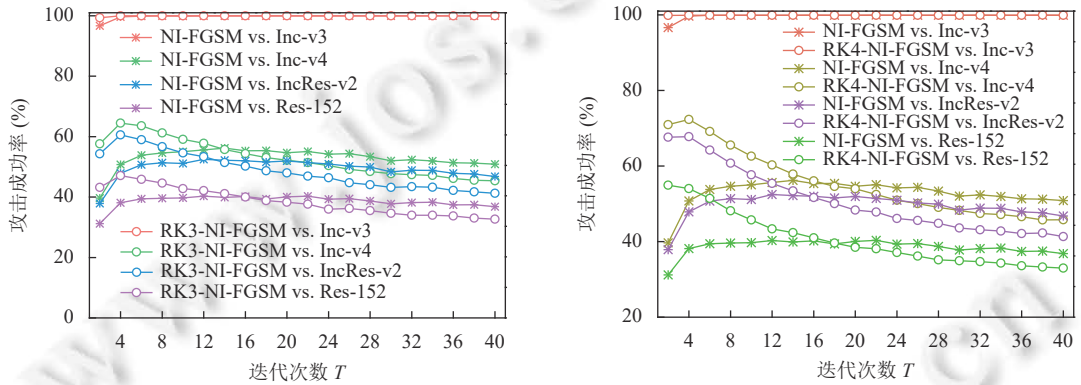


(a) MI-FGSM 和 RK3-MI-FGSM 的对比　　(b) MI-FGSM 和 RK4-MI-FGSM 的对比

图 6　在相同的迭代次数下, MI-FGSM、RK3-MI-FGSM 和 RK4-MI-FGSM 的攻击成功率

最后, 在相同的迭代次数下, 将基线方法 NI-FGSM 与所提出的方法 (即 RK3-NI-FGSM 和 RK3-NI-FGSM) 的攻击成功率进行对比. 分别采用 NI-FGSM、RK3-NI-FGSM 和 RK4-NI-FGSM 攻击 Inc-v3 生成对抗样本, 其中迭代次数 T 从 2 到 40, 增量为 2. 生成的对抗样本对于 4 个正常训练的模型 (Inc-v3、Inc-v4、IncRes-v2 和

Res-152) 的攻击成功率如图 7 所示, 其中横轴和纵轴分别表示迭代次数和攻击成功率, 标签“模型 vs. 方法”表示对抗样本 (由“方法”生成的) 对于“模型”的攻击成功率. 可以观察到, 在白盒模型 Inc-v3 中, NI-FGSM、RK3-NI-FGSM 和 RK4-NI-FGSM 始终保持较高的攻击成功率. 在黑盒模型 (即 Inc-v4、IncRes-v2 和 Res-152) 中, 当迭代次数约为 16 时, NI-FGSM 可以达到最大的黑盒攻击成功率. 当迭代次数约为 4 时, RK3-NI-FGSM 和 RK4-NI-FGSM 可以达到最大的黑盒攻击成功率. 随着迭代次数的增加, NI-FGSM、RK3-NI-FGSM 和 RK4-NI-FGSM 的黑盒攻击成功率都是先上升后下降. 增加迭代次数并不一定可以提升黑盒攻击成功率, 较大的迭代次数会导致生成的对抗样本对白盒模型 Inc-v3 过拟合, 迁移性降低, 进而在黑盒模型中的攻击成功率下降. 此外, 从图 7 中可以发现, 在相同的迭代次数下, 当迭代次数小于 14 时, RK3-NI-FGSM 和 RK4-NI-FGSM 的黑盒攻击成功率始终高于 NI-FGSM 的黑盒攻击成功率, 当迭代次数大于 14 时, RK3-NI-FGSM 和 RK4-NI-FGSM 的黑盒攻击成功率低于 NI-FGSM 的黑盒攻击成功率. 随着迭代次数的变化, 从图 7 中可以观察到, 在黑盒模型中, 基线方法 NI-FGSM 攻击成功率曲线的峰值点 (即迭代次数为 14 时, NI-FGSM 达到的攻击成功率) 始终低于 RK3-NI-FGSM 和 RK4-NI-FGSM 攻击成功率曲线的峰值点 (即迭代次数为 4 时, RK3-NI-FGSM 和 RK4-NI-FGSM 达到的攻击成功率), 这意味着 RK3-NI-FGSM 和 RK4-NI-FGSM 达到的最大攻击成功率要高于 NI-FGSM 所达到的最大攻击成功率.



(a) NI-FGSM 和 RK3-NI-FGSM 的对比　　(b) NI-FGSM 和 RK4-NI-FGSM 的对比

图 7　在相同的迭代次数下, NI-FGSM、RK3-NI-FGSM 和 RK4-NI-FGSM 的攻击成功率

3.4.3　梯度的计算次数

如前所述, 对于基于梯度的攻击, 梯度的计算次数是衡量攻击算法复杂度与运行时间的主要因素. 在每次迭代生成对抗样本时, 基线方法 (即 I-FGSM、MI-FGSM 和 NI-FGSM) 有一个当前样本, 基于三阶龙格库塔法的对抗攻击 (即 RK3-I-FGSM、RK3-MI-FGSM 和 RK3-MI-FGSM) 有 3 个样本 (即两个预测样本和一个当前样本), 基于四阶龙格库塔法的对抗攻击 (即 RK4-I-FGSM、RK4-MI-FGSM 和 RK4-MI-FGSM) 有 4 个样本 (即 3 个预测样本和 1 个当前样本). 对于这些样本, 都可以获得相应的损失函数, 并根据获得的损失函数计算梯度. 当迭代 T 次生成对抗样本 x_T^{adv} 时, 基线方法中梯度的计算次数为 $N=T$, 基于三阶龙格库塔法的对抗攻击方法中梯度的计算次数为 $N=3T$, 基于四阶龙格库塔法的对抗攻击方法中梯度的计算次数为 $N=4T$. 为了讨论本文所提出方法的执行效率, 我们在相同的梯度的计算次数下, 将 3 种迭代方法 (I-FGSM、MI-FGSM 和 NI-FGSM) 和本文所提出的 6 种迭代方法 (RK3-I-FGSM、RK3-MI-FGSM、RK3-NI-FGSM、RK4-I-FGSM、RK4-MI-FGSM 和 RK4-NI-FGSM) 的运行时间和攻击成功率进行对比. 在实验中, 每个像素的最大扰动和步长分别设置为 $\varepsilon=16$, $\alpha=\varepsilon/T$.

首先, 在相同的梯度计算次数下, 比较基线方法 (I-FGSM、MI-FGSM 和 NI-FGSM) 和基于三阶龙格库塔算

法 (RK3-I-FGSM、RK3-MI-FGSM 和 RK3-NI-FGSM) 的运行时间和攻击成功率. 为了在相同的梯度计算次数下进行对比, 我们将基线攻击的迭代次数 T 设置为 3, 6, 9, 对应的梯度的计算次数 N 为 3, 6, 9. 基于三阶龙格库塔算法的迭代次数 T 设置为 1, 2, 3, 对应的梯度的计算次数 N 为 3, 6, 9. 采用上述 6 种方法攻击 Inc-v3 生成对抗样本, 每种方法生成 10 000 幅对抗样本的运行时间和生成的对抗样本对于 7 种模型 (Inc-v3、Inc-v4、IncRes-v2、Res-152、Inc-v3$_{ens3}$、Inc-v3$_{ens4}$ 和 IncRes-v2$_{ens}$) 的攻击成功率如表 8 所示. 可以观察到, 基于三阶龙格库塔算法的运行时间略高于基准攻击的运行时间. 在梯度的计算次数 $N = 3$ 时, 两种基于三阶龙格库塔算法 RK3-MI-FGSM 和 RK3-NI-FGSM 的攻击成功率低于基线方法 MI-FGSM 和 NI-FGSM 的攻击成功率, 这可能是由于我们的方法的迭代次数不足导致的. 在梯度的计算次数 $N = 3$ 时, 基于三阶龙格库塔法的攻击方法只迭代了一次, 而基线方法迭代了 3 次. 需要强调的是, 在梯度的计算次数 N 等于 6 或 9 时, 本文所提出的基于三阶龙格库塔方法的攻击成功率通常远高于基线攻击的攻击成功率.

表 8 在梯度计算次数 N=3, 6, 9 的情况下, 基线方法和基于三阶龙格库塔方法生成 10 000 幅对抗样本的运行时间和对抗样本对于 7 种模型的攻击成功率

N	攻击方法	运行时间 (s)	攻击成功率 (%)						
			Inc-v3	Inc-v4	IncRes-v2	Res-152	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$
3	I-FGSM	**571.30**	**98.48***	41.29	36.81	28.29	16.83	16.39	**5.06**
	RK3-I-FGSM (Ours)	635.75	94.36*	**47.02**	**45.23**	**35.96**	**18.52**	**18.56**	3.53
	MI-FGSM	**596.32**	**98.66***	**52.43**	**49.41**	**40.13**	**24.63**	**22.99**	**6.47**
	RK3-MI-FGSM (Ours)	665.61	94.15*	47.03	45.20	36.23	18.49	18.16	3.61
	NI-FGSM	**596.60**	**99.12***	**47.93**	**45.08**	**36.05**	**20.66**	**19.45**	**5.34**
	RK3-NI-FGSM (Ours)	664.53	94.29*	47.34	44.94	36.04	18.66	18.27	3.57
6	I-FGSM	**882.29**	**99.82***	31.09	25.93	19.74	11.80	11.04	3.36
	RK3-I-FGSM (Ours)	950.79	98.73*	**40.61**	**37.09**	**28.31**	**18.11**	**16.64**	**5.58**
	MI-FGSM	**901.86**	**99.87***	51.45	47.74	39.13	22.90	21.23	**6.47**
	RK3-MI-FGSM (Ours)	972.65	98.37*	**54.80**	**51.51**	**41.75**	**23.74**	**22.06**	5.31
	NI-FGSM	**902.03**	**99.89***	53.51	50.45	39.29	20.93	19.10	**6.01**
	RK3-NI-FGSM (Ours)	975.64	99.38*	**57.57**	**53.57**	**43.29**	**23.68**	**22.32**	5.19
9	I-FGSM	**1 205.09**	**99.91***	24.84	20.16	15.51	9.81	9.03	2.44
	RK3-I-FGSM (Ours)	1 238.50	99.66*	**39.66**	**34.62**	**26.01**	**15.79**	**14.49**	**4.80**
	MI-FGSM	**1 220.70**	**99.89***	48.71	44.97	36.64	22.14	20.88	**6.45**
	RK3-MI-FGSM (Ours)	1 251.62	99.59*	**56.87**	**54.43**	**43.18**	**24.53**	**22.70**	5.92
	NI-FGSM	**1 221.25**	99.91*	54.58	51.75	39.97	21.15	18.90	**6.38**
	RK3-NI-FGSM (Ours)	1 259.82	**99.93***	**62.88**	**59.45**	**46.43**	**24.49**	**22.61**	5.81

注: *表示白盒攻击

最后, 在相同的梯度计算次数下, 我们比较基线方法 (I-FGSM、MI-FGSM 和 NI-FGSM) 的攻击成功率和基于四阶龙格库塔算法 (RK4-I-FGSM、RK4-MI-FGSM 和 RK4-NI-FGSM) 的攻击成功率. 为了在相同的梯度计算次数下进行对比, 基线攻击的迭代次数 T 设置为 4, 8, 12, 基于四阶龙格库塔算法的迭代次数 T 设置为 1, 2, 3. 此时, 基线方法和基于四阶龙格库塔算法的梯度的计算次数 N 为 4, 8, 12. 采用上述 6 种方法攻击 Inc-v3 生成对抗样本, 每种方法生成 10 000 幅对抗样本的运行时间和生成的对抗样本对于 7 种模型 (Inc-v3、Inc-v4、IncRes-v2、Res-152、Inc-v3$_{ens3}$、Inc-v3$_{ens4}$ 和 IncRes-v2$_{ens}$) 的攻击成功率如表 9 所示. 可以观察到, 基于四阶龙格库塔算法的运行时间略微高于基线方法的运行时间, 然而基于四阶龙格库塔算法的攻击成功率通常远高于基线方法的攻击成功率, 尤其是黑盒攻击防御模型.

表 9 在梯度计算次数 $N=4, 8, 12$ 的情况下, 基线方法和基于四阶龙格库塔方法生成 10 000 幅对抗样本的运行时间和对抗样本对于 7 种模型的攻击成功率

N	攻击方法	运行时间 (s)	攻击成功率 (%)						
			Inc-v3	Inc-v4	IncRes-v2	Res-152	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$
4	I-FGSM	**696.40**	**99.50***	37.75	32.46	24.78	14.68	13.88	**4.40**
	RK4-I-FGSM (Ours)	764.22	97.84*	**63.65**	**60.85**	**49.59**	**24.94**	**24.08**	4.04
	MI-FGSM	**709.63**	**99.51***	53.07	49.64	40.36	23.72	22.21	**6.38**
	RK4-MI-FGSM (Ours)	778.16	97.70*	**63.94**	**60.69**	**48.99**	**25.25**	**24.35**	3.79
	NI-FGSM	**711.45**	**99.65***	51.05	48.10	38.39	20.75	19.95	**5.92**
	RK4-NI-FGSM (Ours)	778.79	97.77*	**63.85**	**60.85**	**49.19**	**24.85**	**23.64**	3.87
8	I-FGSM	**1 107.55**	**99.91***	26.98	21.84	16.85	10.72	9.91	2.77
	RK4-I-FGSM (Ours)	1 181.39	99.81*	**57.09**	**51.35**	**38.28**	**22.29**	**20.90**	**6.60**
	MI-FGSM	**1 125.35**	**99.90***	49.68	45.56	37.57	22.63	20.57	6.32
	RK4-MI-FGSM (Ours)	1 191.98	99.75*	**71.19**	**67.37**	**55.03**	**30.71**	**28.66**	**6.74**
	NI-FGSM	**1 127.57**	**99.92***	54.76	51.40	39.64	21.10	19.49	6.23
	RK4-NI-FGSM (Ours)	1 202.32	99.82*	**70.85**	**67.56**	**54.53**	**30.29**	**27.93**	**6.50**
12	I-FGSM	**1 536.22**	**99.92***	21.37	17.39	13.31	8.80	8.38	2.49
	RK4-I-FGSM (Ours)	1 573.95	99.89*	**51.67**	**46.02**	**33.04**	**18.86**	**17.92**	**5.43**
	MI-FGSM	**1 539.66**	**99.90***	46.77	42.17	35.40	20.96	20.06	6.33
	RK4-MI-FGSM (Ours)	1 584.26	**99.90***	**72.84**	**69.41**	**55.44**	**30.98**	**28.57**	**7.12**
	NI-FGSM	**1 548.42**	99.92*	55.90	52.71	40.38	21.03	19.28	5.96
	RK4-NI-FGSM (Ours)	1 587.61	**99.95***	**73.40**	**69.70**	**55.15**	**30.19**	**27.90**	**7.14**

注: *表示白盒攻击

4 总 结

在本文中, 我们首先揭示了基于梯度的攻击与求解常微分方程的数值方法之间的关系, 然后在此基础上, 提出了两种新的基于三阶和四阶龙格库塔法的对抗攻击方法. 所提出方法的主要优点表现在: 1) 基于龙格库塔法的对抗攻击生成的对抗样本可以更有效地攻击白盒模型, 同时在黑盒模型和防御模型中都能取得更高的攻击成功率; 2) 所提出的基于龙格库塔法的对抗攻击具有良好的可扩展性, 可以非常容易地引入几乎所有的基于梯度的攻击方法; 3) 基于龙格库塔法的对抗攻击具有较高的执行效率, 即在相同的迭代次数和相同的梯度计算次数 (或相同的时间约束) 下, 所提出的基于龙格库塔法的攻击通常可以获得较高的攻击成功率.

References:

[1] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. In: Proc. of the 25th Int'l Conf. on Neural Information Processing Systems. Lake Tahoe: Curran Associates Inc., 2012. 1097–1105. [doi: 10.5555/2999134.2999257]

[2] Niu L, Veeraraghavan A, Sabharwal A. Webly supervised learning meets zero-shot learning: A hybrid approach for fine-grained classification. In: Proc. of the 2018 EEE/CVF Conf. on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 7171–7180. [doi: 10.1109/CVPR.2018.00749]

[3] Goyal S, Doddapaneni S, Khapra MM, Ravindran B. A survey in adversarial defences and robustness in NLP. arXiv:2203.06414, 2022.

[4] Xiong W, Droppo J, Huang XD, Seide F, Seltzer ML, Stolcke A, Yu D, Zweig G. Toward human parity in conversational speech recognition. IEEE/ACM Trans. on Audio, Speech, and Language Processing, 2017, 25(12): 2410–2423. [doi: 10.1109/TASLP.2017.2756440]

[5] Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow IJ, Fergus R. Intriguing properties of neural networks. In: Proc. of the 2nd Int'l Conf. on Learning Representations. Banff: ICLR, 2014.

[6] Biggio B, Roli F. Wild patterns: Ten years after the rise of adversarial machine learning. Pattern Recognition, 2018, 84: 317–331. [doi: 10.1016/j.patcog.2018.07.023]

[7] Eykholt K, Evtimov I, Fernandes E, Li B, RahmatiA, Xiao CW, Prakash A, Kohno T, Song D. Robust physical-world attacks on deep learning visual classification. In: Proc. of the 2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 1625–1634. [doi: 10.1109/CVPR.2018.00175]

[8] Goodfellow IJ, Shlens J, Szegedy S. Explaining and harnessing adversarial examples. In: Proc. of the 3rd Int'l Conf. on Learning Representations. San Diego: ICLR, 2015.

[9] Yuan XY, He P, Zhu QL, Li XL. Adversarial examples: Attacks and defenses for deep learning. IEEE Trans. on Neural Networks and Learning Systems, 2019, 30(9): 2805–2824. [doi: 10.1109/TNNLS.2018.2886017]

[10] 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]

[11] Xiao CW, Li B, Zhu JY, He W, Liu MY, Song W. Generating adversarial examples with adversarial networks. In: Proc. of the 27th Int'l Joint Conf. on Artificial Intelligence. Stockholm: AAAI Press, 2018. 3905–3911. [doi: 10.5555/3304222.3304312]

[12] Kurakin A, Goodfellow IJ, Bengio S. Adversarial examples in the physical world. In: Proc. of the 5th Int'l Conf. on Learning Representations. Toulon: OpenReview.net, 2017.

[13] Wan C, Ye BH, Huang FJ. PID-based approach to adversarial attacks. In: Proc. of the 35th AAAI Conf. on Artificial Intelligence. AAAI Press, 2021. 10033–10040. [doi: 10.1609/aaai.v35i11.17204]

[14] Zou JH, Duan YX, Ren CL, Qiu JY, Zhou XY, Pan ZS. Perturbation initialization, Adam-Nesterov and quasi-hyperbolic momentum for adversarial examples. Acta Electronica Sinica, 2022, 50(1): 207–216 (in Chinese with English abstract). [doi: 10.12263/DZXB.20200839]

[15] Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A. Towards deep learning models resistant to adversarial attacks. In: Proc. of the 6th Int'l Conf. on Learning Representations. Vancouver: OpenReview.net, 2018.

[16] Papernot N, McDaniel P, Goodfellow I, Jha s, Celik Z B, Swami A. Practical black-box attacks against machine learning. In: Proc. of the 2017 ACM on Asia Conf. on Computer and Communications Security. Abu Dhabi: ACM, 2017. 506–519. [doi: 10.1145/3052973.3053009]

[17] Huang LF, Zhuang WZ, Liao YX, Liu N. Black-box adversarial attack method based on evolution strategy and attention mechanism. Ruan Jian Xue Bao/Journal of Software, 2021, 32(11): 3512–3529 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6776.htm [doi: 10.13328/j.cnki.jos.006084]

[18] Chen PY, Zhang H, Sharma Y, YI JF, Hsieh CJ. ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In: Proc. of the 10th ACM Workshop on Artificial Intelligence and Security. Dallas: ACM, 2017. 15–26. [doi: 10.1145/3128572.3140448]

[19] Guo C, Gardner J, You YR, Wilson AG, Weinberger K. Simple black-box adversarial attacks. In: Proc. of the 36th Int'l Conf. on Machine Learning. Long Beach: PMLR, 2019. 2484–2493.

[20] Liu YP, Chen XY, Liu C, Song D. Delving into transferable adversarial examples and black-box attacks. In: Proc. of the 5th Int'l Conf. on Learning Representations. Toulon: OpenReview.net, 2017.

[21] Wu L, Zhu ZX, Tai C, E WN. Understanding and enhancing the transferability of adversarial examples. arXiv:1802.09707, 2018.

[22] 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]

[23] Lin JD, Song CB, He K, Wang LW, Hopcroft JE. Nesterov accelerated gradient and scale invariance for adversarial attacks. In: Proc. of the 8th Int'l Conf. on Learning Representations. Addis Ababa: OpenReview.net, 2020.

[24] Wang XS, Lin JD, Hu H, Wang JD, He K. Boosting adversarial transferability through enhanced momentum. In: Proc. of the 32nd British Machine Vision Conf. BMVA Press, 2021. 272.

[25] Wu WB, Su YX, Lyu MR, King I. Improving the transferability of adversarial samples with adversarial transformations. In: Proc. of the 2021 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Nashville: IEEE, 2021. 9020–9029. [doi: 10.1109/CVPR46437.2021.00891]

[26] Zou JH, Pan ZS, Qiu JY, Liu X, Rui T, Li W. Improving the transferability of adversarial examples with resized-diverse-inputs, diversity-ensemble and region fitting. In: Proc. of the 16th European Conf. on Computer Vision. Glasgow: Springer, 2020. 563–579. [doi: 10.1007/978-3-030-58542-6_34]

[27] Xie CH, Zhang ZS, Zhou YY, Bai S, Wang JY, Ren Z, Yuille AL. Improving transferability of adversarial examples with input diversity. In: Proc. of the 2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 2725–2734. [doi: 10.1109/CVPR.2019.00284]

[28] Dong YP, Pang TY, Su H, Zhu J. Evading defenses to transferable adversarial examples by translation-invariant attacks. In: Proc. of the 2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 4307–4316. [doi: 10.1109/CVPR.2019.00444]

[29] Wu DX, Wang YS, Xia ST, Bailey J, Ma XJ. Skip connections matter: On the transferability of adversarial examples generated with

ResNets. In: Proc. of the 8th Int'l Conf. on Learning Representations. Addis Ababa: OpenReview.net, 2020.
[30] Corliss G, Chang YF. Solving ordinary differential equations using Taylor series. ACM Trans. on Mathematical Software, 1982, 8(2): 114–144. [doi: 10.1145/355993.355995]
[31] Butcher JC. Numerical Methods for Ordinary Differential Equations. 3rd ed., Chichester: John Wiley & Sons, Ltd., 2016. 18–20.
[32] Nesterov Y. A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. Doklady AN USSR, 1983, 269: 543–547.
[33] Abadi M, Barham P, Chen JM, *et al*. TensorFlow: A system for large-scale machine learning. In: Proc. of the 12th USENIX Conf. on Operating Systems Design and Implementation. Savannah: USENIX Association, 2016. 265–283. [doi: 10.5555/3026877.3026899]
[34] Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L. ImageNet: A large-scale hierarchical image database. In: Proc. of the 2019 IEEE Conf. on Computer Vision and Pattern Recognition. Miami: IEEE, 2009. 248–255. [doi: 10.1109/CVPR.2009.5206848]
[35] Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai XH, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, Uszkoreit J, Houlsby N. An image is worth 16x16 words: Transformers for image recognition at scale. In: Proc. of the 9th Int'l Conf. on Learning Representations. OpenReview.net, 2021.
[36] Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. In: Proc. of the 2016 Conf. on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 2818–2826. [doi: 10.1109/CVPR.2016.308]
[37] Szegedy C, Ioffe S, Vanhoucke V, Alemi AA. Inception-v4, Inception-ResNet and the impact of residual connections on learning. In: Proc. of the 35th AAAI Conf. on Artificial Intelligence. San Francisco: AAAI Press, 2017. 4278–4284. [doi: 10.5555/3298023.3298188]
[38] He KM, Zhang XY, Ren SQ, Sun J. Identity mappings in deep residual networks. In: Proc. of the 14th European Conf. on Computer Vision. Amsterdam: Springer, 2016. 630–645. [doi: 10.1007/978-3-319-46493-0_38]
[39] Tramèr F, Kurakin A, Papernot N, Goodfellow IJ, Boneh D, McDaniel PD. Ensemble adversarial training: Attacks and defenses. In: Proc. of the 6th Int'l Conf. on Learning Representations. Vancouver: OpenReview.net, 2018.
[40] Liao FZ, Liang M, Dong YP, Pang TY, Hu XL, Zhu J. Defense against adversarial attacks using high-level representation guided denoiser. In: Proc. of the 2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 1778–1787. [doi: 10.1109/CVPR.2018.00191]
[41] Xie CH, Wang JY, Zhang ZS, Ren Z, Yuille AL. Mitigating adversarial effects through randomization. In: Proc. of the 6th Int'l Conf. on Learning Representations. Vancouver: OpenReview.net, 2018. 1–12.
[42] Jia XJ, Wei XX, Cao XC, Foroosh H. ComDefend: An efficient image compression model to defend adversarial examples. In: Proc. of the 2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 6077–6085. [doi: 10.1109/CVPR.2019.00624]
[43] Cohen JM, Rosenfeld E, Kolter Z. Certified adversarial robustness via randomized smoothing. In: Proc. of the 36th Int'l Conf. on Machine Learning. Long Beach: PMLR, 2019. 1310–1320.
[44] Liu ZH, Liu Q, Liu T, Xu N, Lin X, Wang YZ, Wen WJ. Feature distillation: DNN-oriented JPEG compression against adversarial examples. In: Proc. of the 2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 860–868. [doi: 10.1109/CVPR.2019.00095]
[45] Xie CH, Wu YX, van der Maaten L, Yuille AL, He KM. Feature denoising for improving adversarial robustness. In: Proc. of the 2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 501–509. [doi: 10.1109/CVPR.2019.00059]

附中文参考文献:

[14] 邹军华, 段晔鑫, 任传伦, 邱俊洋, 周星宇, 潘志松. 基于噪声初始化、Adam-Nesterov方法和准双曲动量方法的对抗样本生成方法. 电子学报, 2022, 50(1): 207–216. [doi: 10.12263/DZXB.20200839]
[17] 黄立峰, 庄文梓, 廖泳贤, 刘宁. 一种基于进化策略和注意力机制的黑盒对抗攻击算法. 软件学报, 2021, 32(11): 3512–3529. http://www.jos.org.cn/1000-9825/6776.htm [doi: 10.13328/j.cnki.jos.006084]

万晨(1992－), 男, 博士生, 主要研究领域为深度学习的对抗性攻击和防御.

黄方军(1973－), 男, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为 AI 安全, 多媒体内容安全.