软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2023,34(9):4003−4017 [doi: 10.13328/j.cnki.jos.006878]

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

基于稀疏扰动的对抗样本生成方法*

吉顺慧[1,2], 胡黎明[1,2], 张鹏程[1,2], 戚荣志[1,2]

[1](水利部水利大数据重点实验室 (河海大学), 江苏 南京 211100)

[2](河海大学 计算机与信息学院, 江苏 南京 211100)

通信作者: 张鹏程, E-mail: pchzhang@hhu.edu.cn

摘　要: 近年来, 深度神经网络 (deep neural network, DNN) 在图像领域取得了巨大的进展. 然而研究表明, DNN 容易受到对抗样本的干扰, 表现出较差的鲁棒性. 通过生成对抗样本攻击 DNN, 可以对 DNN 的鲁棒性进行评估, 进而采取相应的防御方法提高 DNN 的鲁棒性. 现有对抗样本生成方法依旧存在生成扰动稀疏性不足、扰动幅度过大等缺陷. 提出一种基于稀疏扰动的对抗样本生成方法——SparseAG (sparse perturbation based adversarial example generation), 该方法针对图像样本能够生成较为稀疏并且幅度较小的扰动. 具体来讲, SparseAG 方法首先基于损失函数关于输入图像的梯度值迭代地选择扰动点来生成初始对抗样本, 每一次迭代按照梯度值由大到小的顺序确定新增扰动点的候选集, 选择使损失函数值最小的扰动添加到图像中. 其次, 针对初始扰动方案, 通过一种扰动优化策略来提高对抗样本的稀疏性和真实性, 基于每个扰动的重要性来改进扰动以跳出局部最优, 并进一步减少冗余扰动以及冗余扰动幅度. 选取 CIFAR-10 数据集以及 ImageNet 数据集, 在目标攻击以及非目标攻击两种场景下对该方法进行评估. 实验结果表明, SparseAG 方法在不同的数据集以及不同的攻击场景下均能够达到 100% 的攻击成功率, 且生成扰动的稀疏性和整体扰动幅度都优于对比方法.

关键词: 深度神经网络; 对抗样本生成; 稀疏扰动; 图像识别; 目标攻击; 非目标攻击

中图法分类号: TP391

中文引用格式: 吉顺慧, 胡黎明, 张鹏程, 戚荣志. 基于稀疏扰动的对抗样本生成方法. 软件学报, 2023, 34(9): 4003–4017. http://www.jos.org.cn/1000-9825/6878.htm

英文引用格式: Ji SH, Hu LM, Zhang PC, Qi RZ. Adversarial Example Generation Method Based on Sparse Perturbation. Ruan Jian Xue Bao/Journal of Software, 2023, 34(9): 4003–4017 (in Chinese). http://www.jos.org.cn/1000-9825/6878.htm

Adversarial Example Generation Method Based on Sparse Perturbation

JI Shun-Hui[1,2], HU Li-Ming[1,2], ZHANG Peng-Cheng[1,2], QI Rong-Zhi[1,2]

[1](Key Laboratory of Water Big Data Technology of Ministry of Water Resources (Hohai University), Nanjing 211100, China)

[2](College of Computer and Information, Hohai University, Nanjing 211100, China)

Abstract: In recent years, deep neural network (DNN) has made great progress in the field of image. However, studies show that DNN is susceptible to the interference of adversarial examples and exhibits poor robustness. By generating adversarial examples to attack DNN, the robustness of DNN can be evaluated, and then corresponding defense methods can be adopted to improve the robustness of DNN. The existing adversarial example generation methods still have some defects, such as insufficient sparsity of generated perturbations, and excessive perturbation magnitude. This study proposes an adversarial example generation method based on sparse perturbation, sparse perturbation based adversarial example generation (SparseAG), which can generate relatively sparse and small-magnitude perturbations for

* 基金项目: 国家自然科学基金 (U21B2016, 61702159); 中央高校基本科研业务费专项资金 (B220202072, B210202075); 江苏省自然科学基金 (BK20191297, BK20170893)
本文由“AI 软件系统工程化技术与规范”专题特约编辑张贺教授、夏鑫博士、蒋振鸣副教授、祝立明教授和李宣东教授推荐.
收稿时间: 2022-09-04; 修改时间: 2022-10-13; 采用时间: 2022-12-14; jos 在线出版时间: 2023-01-13
CNKI 网络首发时间: 2023-07-05

image examples. Specifically, SparseAG first selects the perturbation points iteratively based on the gradient value of the loss function for the input image to generate the initial adversarial example. In each iteration, the candidate set of the new perturbation points is determined in the order of gradient value from large to small values, and the perturbation which makes the value of loss function value smallest is added to the image. Secondly, a perturbation optimization strategy is employed in the initial perturbation scheme to improve the sparsity and authenticity of the adversarial example. The perturbations are improved based on the importance of each perturbation for jumping out of the local optimum, and the redundant perturbation and the redundant perturbation magnitude are further reduced. This study selects the CIFAR-10 dataset and the ImageNet dataset to evaluate the method in the target attack and non-target attack scenarios. The experimental results show that SparseAG can achieve a 100% attack success rate in different datasets and different attack scenarios, and the sparsity and the overall perturbation magnitude of the generated perturbations are better than those of the comparison methods.

Key words: deep neural network (DNN); adversarial example generation; sparse perturbation; image recognition; target attack; non-target attack

近年来, 深度神经网络 (deep neural network, DNN) 在图像领域取得了巨大的进展, 被广泛地应用于目标检测[1]、图像语义分割[2]以及图像分类[3]等任务. 随着 DNN 结构愈加复杂, 其对图像的特征提取能力不断加强, 在一些安全性需求较高的领域, 如人脸识别[4]、自动驾驶[5], 也不断取得新的突破.

尽管 DNN 在不同的任务中具有较高的识别准确率, 在面对一些经过特殊修改的样本时, DNN 可能会给出截然相反的结论, 表现出较差的鲁棒性. Szegedy 等人[6]提出, 攻击者对图像添加一些人眼难以察觉到的微小扰动, 构造对抗样本, DNN 就可能对这种图像产生错误的判断. 这在安全性需求较高的领域是一个十分致命的缺陷, 例如在自动驾驶中, 攻击者通过对路边的广告牌添加一些人为构造的微小扰动, 自动驾驶系统可能会对广告牌产生错误的识别结果, 进而做出错误的决策, 导致自动驾驶汽车发生严重的交通事故. 为了保障相关应用的安全性, 有必要研究对抗样本生成方法, 借助对抗样本对 DNN 的鲁棒性进行评估.

当前面向图像的对抗样本生成方法可以分成两大类: 一类是密集对抗样本生成方法[7−15], 这类方法不会对扰动的像素点个数进行限制, 仅对添加到图像当中的整体扰动幅度大小进行优化, 通常情况下会添加 l_2 或者 l_∞ 范数约束, 对图像中所有像素点都进行不同程度的修改; 另一类是稀疏对抗样本生成方法[16], 这类方法对扰动的像素个数进行限制, 添加的扰动通常会受到 l_0 范数约束, 力求尽可能少地修改图像像素点来达到攻击成功的目的. 当前大部分研究主要集中在密集对抗样本生成这一方法[7−15], 然而对图像添加稀疏扰动能够使得添加的扰动更微小, 使对抗样本更真实, 因此稀疏对抗样本生成方法同样值得我们去关注和研究.

近些年来已经有一些稀疏对抗样本生成方法被提出[17−25], 但是这些方法依然存在一些局限性.

(1) 一些稀疏对抗样本生成方法通过选择损失函数关于输入图像的梯度值中最大或最小的像素点来搜索扰动位置, 然而这种搜索策略无法使损失函数值快速变化, 最终导致生成扰动的稀疏性不足.

(2) 大部分稀疏对抗样本生成方法没有考虑局部最优的问题, 导致方法可能陷入局部最优, 进而生成的扰动稀疏性较差.

(3) 一些稀疏对抗样本生成方法只关注生成扰动的稀疏性, 没有对扰动的幅度进行优化, 导致像素添加的扰动过大, 和周围像素的差异较为明显, 构造出的对抗样本不够真实.

针对上述方法存在的局限性, 本文提出了一种基于稀疏扰动的对抗样本生成方法 SparseAG (sparse perturbation based adversarial example generation), 基于损失函数关于输入图像的梯度值迭代地选择扰动点来生成初始对抗样本, 通过一种扰动优化策略来提高对抗样本的稀疏性和真实性. 其中, 本文选取 C&W[24]中使用的损失函数, 该损失函数值越小, 表示当前模型对扰动图像的分类偏离正确标签的程度越大. 本文的主要贡献包括 4 个方面.

(1) 添加扰动的每一次迭代中, 按照梯度值由大到小的顺序确定新增扰动点的候选集, 选择使损失函数值最小的扰动添加到图像中, 以确保扰动的稀疏性.

(2) 针对初始扰动方案, 通过计算每个扰动相对于整体扰动的重要性, 去除冗余扰动以及重要性最小的前 n 个扰动, 并对图像重新添加扰动, 以跳出局部最优, 进而提高扰动的稀疏性.

(3) 为了减少扰动的可见性, 基于扰动重要性对扰动方案进行进一步优化, 在保证攻击成功的情况下不断减少冗余扰动和冗余扰动幅度, 提高最终生成对抗样本的真实性.

(4) 本文将 SparseAG 方法分别应用于 CIFAR-10 数据集[26]和 C&W[24]中使用的 DNN 模型、ImageNet 数据集[27]和 Inception-v3 模型[28]来生成非目标攻击和目标攻击两种场景下的对抗样本, 并与已有方法 GreedyFool[21]、Homotopy-attack[25], SparseFool[23]和 PGD l_0+l_∞[20] 进行实验对比. 实验结果表明, SparseAG 方法在不同的数据集以及不同的攻击场景下均能够达到 100% 的攻击成功率, 且生成扰动的稀疏性和整体扰动幅度都优于对比方法. 以 ImageNet 数据集为例, 与效果较好的 Homotopy-attack 方法相比, 在非目标攻击下, SparseAG 方法的扰动稀疏性提升了 45.60%, 用于衡量整体扰动幅度的 l_1, l_2 范数分别减少了 39.56% 和 17.18%; 在目标攻击下, SparseAG 方法的扰动稀疏性提升了 42.04%, l_1, l_2 范数分别减少了 34.99% 和 16.03%. 此外, 本文还通过消融实验评估 SparseAG 方法关键步骤的效果.

本文第 1 节对现有针对图像的对抗样本生成方法进行总结. 第 2 节对 SparseAG 方法进行详细介绍. 第 3 节介绍评估实验的设置, 包括使用的图像数据集、图像分类模型、方法衡量指标、对比方法. 第 4 节对实验结果进行详细分析. 第 5 节对全文进行总结, 并对未来工作进行阐述.

1 相关工作

目前已经有许多稀疏对抗样本生成方法被相继提出, 大多是基于启发式搜索或者梯度下降法来生成稀疏对抗样本.

Karmon 等人[17]提出一种基于扰动块的稀疏对抗样本生成方法 LaVAN. 该方法将扰动位置限定在一个闭合矩阵区域内, 通过梯度下降法不断优化损失函数值, 生成基于扰动块的稀疏扰动, 使图像逐步被分类为目标攻击标签, 扰动添加的位置并不会遮挡图像任何的关键区域. 该方法通过梯度下降法不断搜索损失函数的最优值. 方法生成的对抗样本攻击成功率较高, 并且由于只对图像的某一块区域进行扰动, 修改的像素点个数只占总像素点个数约 2%, 生成的扰动较为稀疏, 然而该方法需要攻击者手动确定图像中扰动块的位置, 无法根据不同图片以及不同特征来自适应地决定扰动块位置, 灵活性较差.

Croce 等人[18]提出基于随机搜索策略 (random search, RS) 的对抗样本生成方法 Sparse-RS. 在第 i 次迭代下, RS 构造对抗样本的公式如下:

$$\delta \sim D(x^{(i)}),\ x^{(i+1)} = \arg\min_{y\in\{x^{(i)},x^{(i)}+\delta\}} L(y) \tag{1}$$

其中, δ 为添加的扰动, D 为某种抽样分布. 在 Sparse-RS 中, 为了生成稀疏扰动, 方法首先随机初始化像素点集合 M 和相应位置的随机扰动幅度 Δ, 之后不断更新 M 以及 Δ 来构造对抗样本. 具体来讲, 在第 i 次迭代过程中, 方法随机选取像素点构造集合 $A\subset M$ 以及 $B\subset U\backslash M$, 其中 $|A|=|B|=\alpha^{(i)}$, 其中 U 为图像的所有像素点集合. 方法基于集合 A 和 B 构造集合 $M'=(M\backslash A)\cup B$, 并对集合 B 随机生成扰动幅度 Δ', 如果集合 M' 对应的扰动能够使损失函数下降, 则令 $M=M',\Delta=\Delta'$, 否则不修改 M 和 Δ. 不断重复上述过程直到图像成功攻击 DNN 为止. 该方法属于黑盒攻击, 对 DNN 模型的访问次数较少, 攻击成功率较高. 但是无法动态地控制扰动稀疏性, 扰动的像素点个数需要攻击者在方法一开始确定好, 灵活性较差.

Papernot 等人[19]提出面向目标攻击的基于显著图的方法 JSMA. 方法基于前向导数构建显著图, 以反映修改哪些像素点对图像的影响比较大. 通过挑选显著图中像素值最大的像素点作为扰动点, 对其添加固定幅度的扰动, 不断重复该扰动添加过程, 直到扰动后的图像能够成功攻击 DNN 为止. 该方法在构造对抗样本时扰动的像素点个数较多, 并且时间复杂度较高, 在大分辨率数据集上花费的时间较长.

Croce 等人[20] 针对目标攻击, 提出一种基于启发式搜索的黑盒对抗样本生成方法 CornerSearch. 方法依次对图像的每一个像素点添加一系列相应的固定幅度的扰动, 并从中挑选目标攻击标签概率和正确标签概率相差最大的像素点以及相应的扰动幅度作为图像的扰动点. 不断通过启发式搜索寻找扰动点直到图像成功攻击 DNN 为止. 由于需要对图像中所有的像素点都添加扰动进行处理, 因此算法的时间复杂度非常高, 并且该方法可能会陷入局部最优.

Dong 等人[21]提出一种基于贪婪策略的两阶段稀疏对抗样本生成方法 GreedyFool. 方法在第 1 阶段计算损失函数关于输入图像的梯度, 并基于畸变图和梯度相乘的结果, 不断地选取相乘结果最大值对应的像素点作为扰动点, 并基于该像素点对应的梯度值添加扰动幅度, 直到 DNN 被攻击成功为止. 在第 2 阶段中, 使用贪婪策略, 按照

每一个扰动点的扰动幅度由小到大的顺序逐个减少冗余的扰动点. 其中, 本文基于生成对抗网络生成的畸变图用来指示图像不同区域的像素敏感值, 敏感值高的地方, 修改此处容易被人眼察觉到, 反之则不容易被察觉到. 该方法最终生成的稀疏扰动能够达到较高的稀疏性, 以及较低的扰动幅度, 并且该方法迭代速度较快. 由于 GreedyFool 方法使用贪婪策略来搜索扰动点, 因此该方法极易陷入局部最优.

Su 等人[22]提出一种基于差分进化算法的对抗样本生成方法 one-pixel attack, 该方法关注稀疏扰动的一种极端场景——只对图像扰动一个像素点. 该方法使用差分进化算法, 随机初始化种群为不同位置, 不同攻击幅度的扰动点, 然后经过交叉、变异、选择操作, 最终选取能够使得损失函数最小的点作为图像当前的扰动点. 该方法为黑盒攻击方法, 无需计算梯度, 添加的扰动点个数较少, 然而由于该方法只对图片扰动一个像素点, 因此该方法攻击成功率较低.

Fan 等人[16]提出一种基于 ADMM (alternating direction method of multipliers) 算法的对抗样本生成方法 SAPF (sparse adversarial attack via perturbation factorization). 在该方法中, SAPF 将添加的稀疏扰动分解为扰动幅度大小 δ 和二值掩码 G 两个影响因素, 即整体扰动可以表示为: $\varepsilon=\delta\odot G$. 作者将生成稀疏扰动的问题转换为混合整数问题, 使用 ADMM 算法迭代对 δ 和 G 进行优化, 最终生成稀疏扰动. 该方法生成的稀疏扰动整体扰动幅度较低, 然而扰动稀疏性较差, 并且由于 ADMM 算法本身的特性, 该方法在构造稀疏扰动的时候花费的时间较长, 并且 ADMM 算法可能会无法收敛, 导致方法无法正常结束.

Modas 等人[23]基于 DNN 分类图像的分类边界提出 SparseFool 方法, 该方法通过将 l_0 范数优化问题近似的转换成 l_1 范数优化问题. 作者首先使用 DeepFool[9]对图像添加基于 l_2 范数的扰动 r, 目的是将图像移动到分类边界附近, 并基于分类边界的法线向量的坐标值, 不断的优化扰动 r 来提高扰动的稀疏性, 近似地求得攻击 DNN 所需要的稀疏解. 该方法生成扰动速度较快, 然而最后求得的稀疏扰动的稀疏性不足.

Zhu 等人[25]提出一种基于非单调加速近端梯度下降算法 (nonmonotone accelerated proximal gradient method, nmAPG) 和同伦算法的稀疏对抗样本生成方法 Homotopy-Attack. 作者使用 nmAPG 算法来解决基于 l_0 正则化的损失函数不可导, 无法利用梯度下降法优化的问题, 通过 nmAPG 算法生成稀疏扰动, 并通过同伦算法动态地调整参数以保证生成的扰动既能够成功攻击 DNN, 又能够保证足够的稀疏性. 作者在生成稀疏扰动的基础上提出两种优化方法提高扰动的稀疏性: 通过稀疏性控制和可选攻击扰动对扰动的稀疏性进行进一步提升. 稀疏性控制用来控制添加到每一个扰动点的扰动幅度, 而可选扰动攻击用来帮助方法跳出局部最优. Homotopy-Attack 方法相较于其他稀疏对抗样本生成方法而言, 生成的扰动较为稀疏, 攻击成功率较高, 稀疏扰动的整体扰动幅度较低.

2 对抗样本生成方法

2.1 问题描述

对抗样本生成通过对图像添加一些人眼难以察觉到的微小扰动形成样本, 使 DNN 产生错误的判断. 令 x 为原始图像, y 为原始图像对应的正确标签, Z 为被攻击的 DNN 模型, $Z_r(x)$ 为 DNN 关于第 r 种标签的预测概率, 其中 $1\leqslant r\leqslant K$. 在非目标攻击场景下, 针对原始图像 x 添加扰动 δ 形成对抗样本, 要使得模型 Z 针对该对抗样本给出与 y 不同的分类输出, 且添加的扰动需要足够小以保证对抗样本的真实性. 通常情况下, 对抗样本生成方法会通过 l_p 范数 ($p=0,1,2,\infty$) 衡量对抗样本[7−15]. 综上, 对抗样本生成问题一般描述为如下目标函数:

$$\min_{\delta}\|\delta\|_p \quad \text{s.t.} \quad \arg\max_{r=1,\ldots,K} Z_r(x+\delta)\neq y \tag{2}$$

本文着重关注生成对抗样本的稀疏性, 而稀疏性一般通过 l_0 范数进行约束[17−25]. 此外, 为了保证对抗样本的真实性, 在生成扰动的过程中, 需要为每一个扰动点添加的扰动设置扰动阈值 ε, 即为扰动 δ 的 l_∞ 范数设置阈值, 从而在指定的最大扰动幅度下扰动尽可能少的像素点. 因此, 在非目标攻击场景下, 本文的对抗样本生成问题描述为如下目标函数.

$$\min_{\delta}\|\delta\|_0 \ \text{ s.t. } \ \arg\max_{r=1,\ldots,K} Z_r(x+\delta)\neq y, \|\delta\|_\infty\leqslant\varepsilon \tag{3}$$

相应地, 在目标攻击场景下, 针对原始图像 x 添加扰动 δ 形成对抗样本, 要使得模型 Z 针对该对抗样本给出特定的分类输出 y', 则对抗样本生成问题描述为如下目标函数:

$$\min_{\delta}\|\delta\|_0 \quad \text{s.t.} \quad \arg\max_{r=1,\ldots,K} Z_r(x+\delta)=y',\ \|\delta\|_\infty \leqslant \varepsilon \tag{4}$$

由于以上目标函数实际求解比较困难, 本文基于损失函数关于输入图像的梯度值迭代地选择扰动点来生成初始对抗样本, 每一次迭代选择使损失函数值最小的扰动添加到图像中. 基于每个扰动的重要性来改进扰动以跳出局部最优, 并进一步减少冗余扰动以及冗余扰动幅度, 使得最终生成的对抗样本具有较高的稀疏性和真实性.

2.2 方法总体框架

本文提出 SparseAG 方法, 为针对图像分类的 DNN 生成扰动稀疏且真实的对抗样本, 方法框架如图 1 所示, 主要分为 3 个部分.

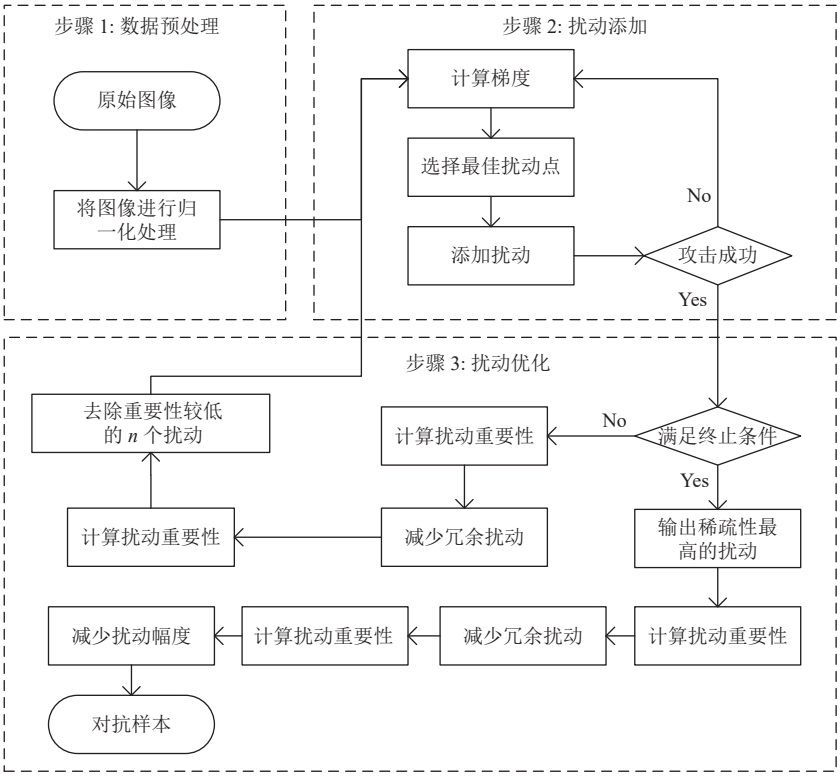


图 1 方法总体框架

(1) 数据预处理: 针对原始图像数据集中的图像样本进行归一化处理, 将图像中每一个 [0, 255] 的像素值归一化到 [0, 1] 区间内.

(2) 扰动添加: 基于损失函数关于输入图像的梯度值迭代地选择扰动点来生成初始对抗样本, 每一次迭代按照梯度值由大到小的顺序确定新增扰动点的候选集, 选择使损失函数值最小的扰动添加到图像中, 直到被扰动的图像能够成功攻击 DNN.

(3) 扰动优化: 针对初始扰动方案, 通过一种扰动优化策略来提高对抗样本的稀疏性和真实性. 为了跳出局部最优, 通过计算每个扰动相对于整体扰动的重要性, 去除冗余扰动以及重要性最小的前 n 个扰动, 并对图像重新添加扰动直到攻击成功, 重复该过程直到满足终止条件, 或者达到最大迭代次数 m, 或者生成扰动的稀疏性连续 p 次迭代不再提高, 选择其中稀疏性最高的一种作为扰动方案. 为了使得生成的对抗样本更加真实, 在保证攻击成功的情况下, 通过减少冗余扰动和剩余扰动点的冗余扰动幅度, 进一步优化扰动方案得到最终的对抗样本.

2.3 扰动添加

为了生成稀疏扰动, SparseAG 方法基于损失函数关于输入图像的梯度值迭代地选择扰动点并添加相应幅度的扰动, 直到被扰动的图像能够成功攻击 DNN, 形成初始扰动方案.

为了记录扰动的像素点位置, 根据图像大小使用矩阵 v 来进行标识, 其中添加扰动的像素位置元素标为 1, 否则标为 0. 在每次迭代中, 首先计算 DNN 的损失函数关于输入图像的梯度 g, 再基于 g 根据扰动点搜索策略确定最佳扰动点, 并确定在该扰动点添加扰动的幅度. 本文使用 C&W[24]中提出的损失函数, 针对非目标攻击的损失函数为:

$$f(x,y)=\max(Z(x)_y-\max\{Z(x)_i:i\neq y\},-h) \tag{5}$$

针对目标攻击的损失函数为:

$$f(x,y')=\max(\max\{Z(x)_i:i\neq y'\}-Z(x)_{y'},-h) \tag{6}$$

其中, x,y 分别为图像样本以及其相应的真实标签, y' 为目标攻击中特定的分类输出, $Z(x)_i$ 为 DNN 针对图像 x 关于标签 i 的预测概率, h 为置信度参数, 默认为 0, h 的值越大则最终生成的对抗样本在黑盒攻击场景下迁移性越强[29].

在选择最佳扰动点时, 已有方法往往选择 g 中梯度值最大的像素点作为图像的扰动点[18,21], 然而实现中发现这种扰动点搜索策略不能保证损失函数下降最大, 从而导致对抗样本生成方法需要更多的迭代次数来搜索损失函数的最小值, 使得生成的扰动稀疏性较差. 为了能使损失函数快速下降, 提高添加扰动的稀疏性, 本文将梯度与损失函数相结合来搜索最佳扰动点. 具体来讲, 在 t+1 次迭代中, 首先针对图像中前 t 次迭代确定的扰动点, 在扰动阈值的限制下, 基于本次迭代计算出的梯度值继续添加扰动, 其次按照损失函数下降最大的搜索策略确定新增扰动点. 在图像剩余未被扰动的像素点中, 按照梯度值由大到小的顺序将前 k 个像素点 $p_1,p_2,\ldots,p_k$ 作为候选扰动点集, 分别对其中 1 个像素点添加扰动, 若在 p_i 添加的扰动导致损失函数值最小, 则将 p_i 确定为第 t+1 次迭代新增的扰动点, 并将其扰动添加到图像中, 得到第 t+1 次迭代后的扰动图像 x_{t+1}^{adv}, 相应地修改 v 中 p_i 相应像素位置的值为 1. 其中, 在选定前 k 个候选扰动点后, 可以同时将生成的 k 张扰动图像输入进 DNN 中, 并行计算 DNN 预测对每张图像的损失函数值, 有效减少程序运行的时间开销. 不断重复上述扰动点搜索过程, 直到被扰动的图像能够使 DNN 分类为错误标签或者目标攻击标签为止. 向图像迭代添加扰动的具体公式如下:

$$x_{t+1}^{adv}=Clip_x^{\varepsilon}(x_t^{adv}+\alpha\cdot(g\cdot v)) \tag{7}$$

其中, $g\cdot v$ 表示两个矩阵对应位置元素值的逻辑乘, 即若 v_{ij} 为 1, 则在对应像素点上添加值为 g_{ij} 的扰动; α 是方法设定的固定值参数, 用于控制添加扰动的幅度; $Clip_x^{\varepsilon}(x)$ 将图像 x 中每一个像素添加的扰动限制在扰动阈值 ε 内. 由于部分像素点添加的扰动可能会超过设定的扰动阈值, 导致整体的扰动幅度偏大, 扰动真实性降低, 因此需要对图像中添加的扰动进行限制操作. 具体来讲, 如果当前扰动点的扰动大小超过了阈值, 则将扰动大小限制到和阈值大小相等, 反之不对该扰动进行任何操作. 添加扰动的具体流程如算法 1 所示.

算法 1. *PertAdd*().

输入: 原始图像样本 x, 样本标签 y;
输出: 对抗样本 x^{adv}.

1. 初始化: t=0, $x_t^{adv}=x$, $v'=v$;
2. **while** $\arg\max_{r=1,2,\ldots,K} Z_r(x_t^{adv})==y$ **do**
3. 初始化 $minLoss=\infty$;
4. 计算损失函数关于 x_t^{adv} 的梯度 g;
5. 挑选 g 中未被扰动的像素点中梯度最大的前 k 个像素点 P;
6. **for** $p_i\in P$
7. $v'_{p_i}=1$; // v' 中 p_i 对应元素值设为 1
8. $x_{t+1}^{adv}=Clip_x^{\varepsilon}(x_t^{adv}+\alpha\cdot(g\cdot v'))$;

9. **if** $f(x_{t+1}^{adv}, y) < minLoss$
10. $minLoss = f(x_{t+1}^{adv}, y)$;
11. $selectedpixel = p_i$;
12. **end if**
13. $v' = v$
14. **end for**
15. $v_{selectedpixel} = 1$;
16. $x_{t+1}^{adv} = Clip_x^{\varepsilon}(x_t^{adv} + \alpha \cdot (g \cdot v))$;
17. $t \leftarrow t+1$;
18. **end while**
19. $x^{adv} = x_t^{adv}$;
20. **return** x^{adv};

2.4 扰动优化

在添加初始扰动之后, 此时构造的对抗样本 x^{adv} 虽然能够成功攻击 DNN, 使其错误分类, 但添加的扰动容易落入局部最优, 扰动的稀疏性仍然具有较大的优化空间. 为了跳出局部最优, 进一步提高扰动的稀疏性和真实性, 本文提出了一种扰动优化策略, 分为以下两个阶段.

(1) 提高扰动稀疏性: 基于扰动重要性去除冗余扰动以及重要性较低的 n 个扰动, 并按扰动添加方法重新添加扰动直到攻击成功, 重复该过程来选择稀疏性最高的扰动方案.

(2) 提高扰动真实性: 在保证攻击成功的情况下, 通过减少冗余扰动以及剩余扰动点的扰动幅度, 来降低对抗样本的整体扰动幅度.

在提高扰动稀疏性阶段, 首先针对初始扰动 δ 中各个扰动计算其重要性. 假设对抗样本 x^{adv} 的扰动由 q 个像素点的扰动 δ_j ($j = 1,2,3,\ldots,q$) 组成, 对扰动 δ_j 的重要性评估方式为: 从原对抗样本 x^{adv} 中去除扰动 δ_j, 得到新的扰动图像 x^{adv*}, 计算 x^{adv*} 的损失函数值 $f(x^{adv*}, y)$, δ_j 的扰动重要性通过 x^{adv*} 相比原对抗样本 x^{adv} 的损失函数变化量来衡量, 计算公式如下:

$$\sigma_j = f(x^{adv*}, y) - f(x^{adv}, y) \tag{8}$$

通过扰动重要性可以评估哪些扰动对 DNN 而言攻击性较弱, 最弱的扰动有可能是冗余的. 方法首先计算扰动重要性, 并按照扰动重要性由小到大的顺序逐个判断其是否为冗余扰动, 如果去掉该扰动后的样本仍然能够使 DNN 产生错误判断, 则该扰动为冗余扰动. 去除冗余扰动点后由于整体扰动发生变化, 每个扰动的重要性也可能发生变化, 因此需再次计算扰动重要性, 根据扰动重要性排序去除对抗样本中重要性较小的前 n 个扰动, 并按照初始扰动添加阶段的扰动添加方法迭代地选择扰动点并添加相应幅度的扰动, 直到被扰动的图像成功攻击 DNN 为止. 不断重复该过程, 当迭代次数超过 m 次或者生成的扰动稀疏性连续 p 次不再下降时, 迭代结束, 选择其中稀疏性最高的一种作为扰动方案. 通过去除重要性较低的扰动, 在不过多影响对抗样本攻击性的情况下, 使得对抗样本生成方法以一个新的起始状态重新对图像添加扰动, 进而帮助方法跳出局部最优, 生成稀疏性更高的扰动.

提高扰动稀疏性阶段结束之后得到的对抗样本依然可能存在冗余的扰动以及冗余的扰动幅度, 为了进一步提高扰动真实性, 按照扰动重要性排序去除对抗样本中的冗余扰动, 并重新计算扰动重要性, 对剩余扰动点优化扰动幅度. 按照扰动重要性排序, 减少的扰动幅度应该遵循: 重要性较高的扰动点, 减小的扰动幅度较小, 以保留较高的攻击性; 反之, 重要性较低的扰动点, 减少的扰动幅度较大. 针对扰动 δ_j, 在相应扰动点减小的扰动幅度计算公式如下:

$$\delta_j^{\text{reduce}} = \delta_j \cdot \lambda \cdot \frac{1}{|\sigma_j|} \tag{9}$$

其中, 参数 λ 为固定值参数, 用于控制减少的扰动幅度. 若在 δ_j 对应扰动点上减少值为 δ_j^{reduce} 的扰动后得到的样本仍然能成功攻击 DNN, 则保留该扰动优化操作; 否则, 撤销该扰动幅度的调整.

扰动优化的具体流程如算法 2 所示, 其中 $PertImportance(x, x_i^{adv})$ 用于计算对抗样本 x_i^{adv} 中各扰动的重要性.

算法 2. *PertOptimization*().

输入: 原始图像样本 x, 样本标签 y, 对抗样本 x^{adv};
输出: 对抗样本 x^{adv}.

1. 初始化: i=0, $x_i^{adv} = x^{adv}$;
2. **while** $i<m$ or 稀疏性连续 p 次不再提升 **do**
3. $\sigma = PertImportance(x, x_i^{adv})$;
4. $x_i^{adv} \leftarrow x_i^{adv}$ 去除冗余扰动的样本;
5. $\sigma = PerImportance(x, x_i^{adv})$;
6. x_i^{adv} 去除重要性较小的 n 个扰动;
7. $x_{i+1}^{adv} = PertAdd(x_i^{adv}, y)$;
8. $i \leftarrow i+1$;
9. **end while**
10. $x^{adv} \leftarrow$ 迭代过程中扰动稀疏性最高的对抗样本;
11. $x^{adv} \leftarrow$ 去除冗余扰动点和冗余扰动幅度后的对抗样本;
12. **return** x^{adv};

3 实验设置

本节对实验数据集、针对的图像分类模型、衡量指标和对比方法进行介绍, 用于评估本文提出的对抗样本生成方法实际效果.

3.1 实验数据集

本文使用 CIFAR-10[26]数据集以及 ImageNet[27]数据集作为实验数据集对对抗样本生成方法进行评估. 其中 CIFAR-10 数据集包含 50 000 张图像的训练数据集以及 10 000 张图像的测试数据集, 数据集共包含 10 个类别, 图像分辨率为 $32\times32\times3$. 由于 ImageNet 数据集包含的图片以及类别过多, 实验选取 ImageNet 数据集的一个子数据集——ILSVRC2012 数据集, 该数据集包含 128 万张图像的训练数据集、50 000 张图像的验证数据集以及 10 万张图像的测试数据集, 数据集共包含 1 000 个类别, 图像分辨率为 $299\times299\times3$. 在非目标攻击实验中, 从 CIFAR-10 测试数据集中随机抽取 5 000 张图像, 从 ILSVRC2012 验证数据集中随机抽取 1 000 张图像, 作为对抗样本生成方法中待扰动的图像数据集. 在目标攻击实验中, 从 CIFAR-10 测试数据集中随机抽取 1 000 张图像, 从 ILSVRC2012 验证数据集中随机抽取 50 张图像, 作为待扰动的图像数据集, 每一张图像采用 9 个目标攻击标签分别生成对抗样本.

3.2 图像分类模型

在 CIFAR-10 数据集上, 实验采用 C&W[24]中使用的 DNN 模型作为图像分类模型, 该 DNN 模型包含 4 个卷积层, 3 个全连接层以及 2 个池化层, 其输入图像尺寸为 $32\times32\times3$, 基于 CIFAR-10 训练数据集对该模型进行训练, 能够达到 80.65% 的分类准确率. 在 ImageNet 数据集上, 本文使用预训练好的 Inception-v3 模型[28]作为图像分类模型, 能够达到 77.45% 的分类准确率.

3.3 衡量指标

针对对抗样本生成方法在目标攻击以及非目标攻击两种场景下构造的对抗样本, 实验通过扰动的 l_0, l_1, l_2, l_∞

范数、对抗样本的攻击成功率 ASR (attack success rates)、生成对抗样本所需时间 T 这几种指标来评估对抗样本生成方法的效果.

l_0 范数: l_0 范数指扰动矩阵中非零元素的个数, 在对抗样本生成方法中表示图像中被扰动的像素点个数, 用来评估生成扰动的稀疏性. l_0 范数的值越小, 表示图像中被扰动的像素点个数越少, 也就意味着对抗样本生成方法生成的扰动稀疏性更高, 能够通过扰动更少的像素点成功攻击 DNN.

l_1, l_2 范数: l_1 范数指扰动矩阵中各元素绝对值的和, l_2 范数指扰动矩阵中各元素平方和的平方根. l_1, l_2 用来衡量扰动的整体幅度大小. l_1, l_2 范数的值越小, 表示添加在图像中的扰动幅度越小.

l_∞ 范数: l_∞ 范数指扰动矩阵中各个元素绝对值的最大值, 用于衡量添加扰动的最大幅度. l_∞ 范数的值越小, 表示添加在图像中的最大扰动幅度越小.

攻击成功率 ASR: 攻击成功率用于衡量对抗样本生成方法的有效性. 假设对抗样本生成方法共生成 m 个对抗样本, 其中 n 个样本攻击 DNN 模型成功, 则攻击成功率为:

$$ASR = \frac{n}{m} \times 100\% \tag{10}$$

生成对抗样本所需时间 T: 主要用于衡量对抗样本生成方法的性能, 令 t_{start} 表示图像输入对抗样本生成方法的时间, t_{end} 表示方法构造出对抗样本的时间, 公式如下:

$$T = t_{\text{end}} - t_{\text{start}} \tag{11}$$

3.4 对比方法

为了评估 SparseAG 方法的具体效果, 实验在目标攻击以及非目标攻击场景下分别选取不同的对抗样本生成方法进行对比. 在非目标攻击场景下, 实验选取 SparseFool[23]、GreedyFool[21]、Homotopy-attack[25]以及 PGD $l_0 + l_\infty$ [20] 这 4 种对抗样本生成方法进行对比, 上述 4 种方法在生成对抗样本的时间开销、生成扰动的稀疏性以及扰动的真实性等方面, 都要优于其他对抗样本生成方法, 因此实验选取上述 4 种方法作为非目标攻击场景下的对比方法. 在目标攻击场景下, 由于 SparseFool[23]无法应用到目标攻击场景, PGD $l_0 + l_\infty$ [20]生成的扰动稀疏性较低, 扰动幅度较大, 因此实验只选取 GreedyFool[21]以及 Homotopy-attack[25]作为目标攻击的对比方法.

4 实验结果与分析

实验针对选定的两种 DNN 模型, 在非目标攻击以及目标攻击场景中, 基于 CIFAR-10 以及 ImageNet 数据集对不同的对抗样本生成方法进行评估. 在参数选择上, SparseAG 方法在 CIFAR-10 数据集上, 选择的候选扰动点个数 k 设置为 150, 在 ImageNet 数据集上 k 设置为 50. 此外, 为了减少添加在图像当中的扰动幅度, 提高对抗样本的真实性, 对各种对抗样本生成方法的最大扰动幅度 ε 参考 Homotopy-attack 方法[25]设置为 0.05.

4.1 非目标攻击实验结果

在非目标攻击场景下, DNN 模型只要将对抗样本分类为非正确标签即可认为攻击成功. 实验使用上述 5 种方法对原始图像添加扰动构造对抗样本, 实验结果如表 1 所示, 其中第 3 列表示各方法所生成对抗样本的攻击成功率, 第 4–7 列分别表示对抗样本的平均 l_0, l_1, l_2, l_∞ 范数.

从表 1 可以看出, SparseAG 方法能够在两种数据集上都达到 100% 的攻击成功率, 并且在 l_0, l_1, l_2 范数上均表现最优, 即扰动稀疏性和整体扰动幅度都表现最优. 具体来讲, 在 CIFAR-10 数据集上, SparseAG 方法平均只需要扰动原始图像 2.2% 的像素点个数即可攻击成功, 其扰动点个数与 Homotopy-attack 方法相比减少 43.33%. SparseFool 和 PGD $l_0 + l_\infty$ 方法实验结果较差, 攻击成功率分别只能达到 97.2% 和 85.5%, 并且生成扰动的稀疏性较低. 在生成扰动的幅度方面, SparseAG 方法的 l_∞ 范数要略高于 Homotopy-attack 方法, 虽然生成扰动的最大幅度不是最优, 但其生成扰动的整体幅度最优, l_1, l_2 范数与 Homotopy-attack 方法相比分别减少 36.63% 和 14.39%. 在 ImageNet 数据集上, SparseAG 方法平均只需扰动原始图像 0.1% 的像素点, 其扰动点个数与 Homotopy-attack 方法相比减少 45.60%. SparseFool 和 PGD $l_0 + l_\infty$ 表现较差, 攻击成功率分别只能达到 82% 和 53.8%, 并且生成的扰动

稀疏性较低. 在生成扰动的幅度方面, 虽然 SparseAG 方法的最大扰动幅度不是最优, 但其生成扰动的整体扰动幅度最优, l_1, l_2 范数与 Homotopy-attack 方法相比分别减少 39.56% 和 17.18%. 实验表明 SparseAG 方法在具有复杂特征的 ImageNet 数据集上依然能够生成稀疏性较高且扰动幅度较低的扰动, 构造出较真实的对抗样本.

表 1 非目标攻击实验结果比较

数据集	方法	*ASR* (%)	l_0	l_1	l_2	l_∞
CIFAR-10	SparseAG	**100**	**68**	**3.393**	**0.369**	0.049
	SparseFool	97.2	445	21.90	0.837	0.050
	GreedyFool	100	155	7.667	0.541	0.050
	Homotopy-attack	100	120	5.354	0.431	**0.043**
	PGD l_0+l_∞	85.5	209	10.47	0.723	0.050
ImageNet	SparseAG	**100**	**303**	**15.08**	**0.757**	0.050
	SparseFool	82	1 839	90.20	1.434	0.050
	GreedyFool	100	1 788	58.68	1.304	0.049
	Homotopy-attack	100	557	24.95	0.914	**0.045**
	PGD l_0+l_∞	53.8	2 095	88.03	2.019	0.050

4.2 目标攻击实验结果

在目标攻击场景下, DNN 模型需要将对抗样本分类为攻击者指定的标签才能攻击成功, 目标攻击相比非目标攻击更加困难. 实验使用 3 种对抗样本生成方法对原始图像添加扰动构造对抗样本, 实验结果如表 2 所示. 其中, 第 3 列 Best case 表示在每张图像的 9 个目标标签中, 稀疏性最高的目标对抗样本的平均统计, 第 4 列 Average case 表示所有目标对抗样本的平均统计, 第 5 列 Worst case 表示 9 个目标标签中稀疏性最低的目标对抗样本的平均统计.

表 2 目标攻击实验结果比较

数据集	方法	Best case					Average case					Worst case				
		ASR (%)	l_0	l_1	l_2	l_∞	*ASR* (%)	l_0	l_1	l_2	l_∞	*ASR* (%)	l_0	l_1	l_2	l_∞
CIFAR-10	SparseAG	**100**	**63**	**3.10**	**0.35**	0.049	**100**	**163**	**8.05**	**0.59**	0.050	**100**	**269**	**13.29**	**0.78**	0.050
	GreedyFool	100	114	5.69	0.47	0.049	100	377	18.65	0.88	0.050	100	728	35.26	1.28	0.050
	Homotopy-attack	100	124	5.63	0.45	**0.042**	100	260	11.58	0.68	**0.045**	100	433	18.01	0.85	**0.043**
ImageNet	SparseAG	**100**	**951**	**46.47**	**1.39**	0.050	**100**	**1 977**	**93.30**	**1.99**	0.050	**100**	**3 477**	**158.05**	**2.62**	0.050
	GreedyFool	100	9 560	378.49	3.84	0.050	100	21 008	416.02	3.38	**0.047**	100	30 334	406.89	2.89	**0.046**
	Homotopy-attack	100	1 698	73.70	1.73	**0.049**	100	3 411	143.51	2.37	0.049	100	5 646	231.86	3.02	0.049

从表 2 可以看出, 在两种数据集下, SparseAG 方法在 Best case、Average case 和 Worst case 下都能够实现 100% 的攻击成功率, 并且 l_0, l_1, l_2 范数均能够达到最优, 即扰动稀疏性和整体扰动幅度都表现最优. 具体来讲, 在 CIFAR-10 数据集上, SparseAG 方法在 Average case 下平均只需要扰动原始图像 5.31% 的像素点即攻击成功, 其扰动点个数与 Homotopy-attack 方法相比减少 37.31%. 在生成扰动的幅度方面, 虽然 SparseAG 方法的 l_∞ 范数在 3 种统计场景下均要略高于 Homotopy-attack 方法, 但是在 l_1, l_2 范数方面均能够达到最优, 在 Average case 下和 Homotopy-attack 相比分别减少 30.48% 和 13.24%. 在 ImageNet 数据集上, SparseAG 方法在 Average case 下平均只需要扰动原始图像 0.7% 的像素点即攻击成功, 其扰动点个数与 Homotopy-attack 方法相比减少 42.04%. 在生成扰动的幅度方面, 虽然 SparseAG 方法的最大扰动幅度不是最优, 但整体扰动幅度最优, l_1, l_2 范数与 Homotopy-attack 方法相比分别减少 34.99% 和 16.03%. 实验表明 SparseAG 方法在目标攻击场景下依然能够生成较为稀疏并且更为真实的扰动.

4.3 生成对抗样本时间

实验统计了在目标攻击以及非目标攻击下, 几种方法生成对抗样本所需的平均时间.

表 3 展示了 5 种方法非目标对抗样本生成的平均时间. 结果表明, SparseAG 方法生成对抗样本所需的时间并非最优. 具体来讲, 在 CIFAR-10 数据集上, SparseAG 方法的消耗时间比 SparseFool, GreedyFool 以及 PGD l_0+l_∞ 这 3 种方法高, 但与攻击成功率、扰动稀疏性以及整体扰动幅度表现更优的 Homotopy-attack 方法相比更低. 在 ImageNet 数据集上, 同样地, SparseAG 方法的消耗时间仅低于 Homotopy-attack 方法. 对比 SparseAG 方法在两个数据集上的时间消耗, 可以看出, 随着图像分辨率的增大, 图像中包含的特征更为复杂, SparseAG 方法需要更多的迭代来搜寻扰动点, 其中每一次迭代都需要花费时间来确定当前扰动点, 且需要更多的时间来优化扰动以提升稀疏性, 因此在 ImageNet 数据集上花费的时间相对较长.

表 3 非目标对抗样本生成的平均时间 (s)

数据集	SparseAG	SparseFool	GreedyFool	Homotopy-attack	PGD l_0+l_∞
CIFAR-10	6.32	3.02	**0.46**	42.73	5.14
ImageNet	879.93	**19.93**	37.08	893.63	82.59

表 4 展示了 3 种方法目标对抗样本生成的平均时间. 结果表明, SparseAG 方法生成对抗样本所需的时间并非最优. 具体来讲, 在 CIFAR-10 数据集上, SparseAG 方法的消耗时间比 GreedyFool 高, 但与 Average case 下攻击成功率、扰动稀疏性以及整体扰动幅度表现更优的 Homotopy-attack 方法相比更低. 在 ImageNet 数据集上, 对于大尺寸的图像样本, SparseAG 方法的消耗时间相比其他两种方法较高. 对比 SparseAG 方法在两个数据集上的时间消耗, 可以看出, 由于 ImageNet 数据集图像分辨率大且包含更加复杂的特征, SparseAG 方法在该数据集下花费的时间较长. 对比 SparseAG 方法在目标对抗样本生成和非目标对抗样本生成的时间消耗, 可以看出, 由于目标攻击相比非目标攻击更加困难, 在两个数据集上目标对抗样本生成所需时间均更高.

表 4 目标对抗样本生成的平均时间 (s)

数据集	SparseAG	GreedyFool	Homotopy-attack
CIFAR-10	13.86	**2.49**	64.41
ImageNet	2 526.16	**273.42**	1 447.45

综合表 3、表 4 可以看出, SparseAG 方法在两种场景下的时间开销与部分对比方法有差距, 但是 SparseAG 方法在生成对抗样本的攻击成功率、生成扰动的稀疏性以及整体扰动幅度方面都表现最优, 这也正是本文的主要改进目标. 此外, SparseAG 方法在非目标攻击场景的两个数据集以及目标攻击场景的 CIFAR-10 数据集上时间开销并非最差, 低于对比方法中攻击成功率、扰动稀疏性以及整体扰动幅度表现更优的 Homotopy-attack 方法, 且在两种场景的 CIFAR-10 数据集上与其他方法相比差距较小. SparseAG 方法仅在目标攻击场景的 ImageNet 数据集上时间开销较大, 这是由于目标攻击较为困难以及 ImageNet 图像较为复杂两方面所导致. 总的来说, 本文认为 SparseAG 方法在时间开销上与对比方法的差距是可以接受的.

4.4 消融实验

为了更进一步地理解 SparseAG 方法关键步骤所起到的作用, 实验在非目标攻击场景下, 在 CIFAR-10 以及 ImageNet 数据集上进行消融实验. 其中, 将 GreedyFool 方法只进行扰动添加, 不进行冗余扰动去除的实验命名为 M1; 将 SparseAG 方法中只进行初始扰动添加的实验命名为 M2; 将 SparseAG 方法中只进行初始扰动添加以及冗余扰动和冗余扰动幅度减少的实验命名为 M3; 将 SparseAG 方法中只进行初始扰动添加以及跳出局部最优的实验命名为 M4; 将完整的 SparseAG 方法实验命名为 M5.

表 5 展示了非目标攻击下的消融实验结果, 从表中可以看出, SparseAG 的扰动添加策略以及扰动优化策略能够有效地提高扰动的稀疏性和真实性. 具体来讲, 在 CIFAR-10 数据集下, M2 与 M1 相比, l_0 范数下降明显, 即生

成扰动的稀疏性大幅提高, 这表明基于损失函数值最小的扰动添加策略能够使得损失函数值快速下降, 从而扰动更少的像素点; 并且l_1, l_2范数下降明显, 即生成扰动的整体幅度更小, 由于扰动的像素点少了, 整体幅度自然更小. M3 与 M2 相比, l_0, l_1, l_2范数都有提升, 表明减少冗余扰动和冗余扰动幅度能进一步提高生成扰动的稀疏性和整体幅度. M4 与 M2 相比, l_0, l_1, l_2范数提升更多, 表明跳出局部最优能够有效地提高扰动的稀疏性和真实性, 并且优化的效果要优于 M3. M5 与 M3 相比, 生成扰动的稀疏性和整体幅度均有较大提升, 表明跳出局部最优的重要性; 与 M4 相比, 生成扰动的稀疏性没有提升, 整体幅度有所提升, 由于 M4 使用的跳出局部最优的优化策略已经能够使得扰动达到较高的稀疏性, 因此 M5 相比较 M4 提升的效果有限, 但是冗余扰动和冗余扰动幅度减少的优化策略依然具有一定的优化效果. 在 ImageNet 数据集下, 由于图像尺寸变大, 包含的特征更加复杂, 因此 SparseAG 每一部分的优化效果更明显, M2 与 M1 相比, M3、M4 与 M2 相比, M5 与 M3 相比, 生成扰动的稀疏性和整体幅度都大幅度提升, 并且 M4 的优化效果比 M3 更优, M5 与 M4 相比的提升效果有限, 但是依然具有一定的优化效果.

表 5　消融实验结果

数据集	方法	*ASR* (%)	l_0	l_1	l_2	l_∞
CIFAR-10	M1	100	215	10.580	0.635	0.050
	M2	100	74	3.725	0.384	0.050
	M3	100	72	3.622	0.378	0.050
	M4	100	68	3.397	0.370	0.050
	M5	100	68	3.379	0.367	0.050
ImageNet	M1	100	2 763	131.793	2.042	0.050
	M2	100	1 008	47.858	1.150	0.050
	M3	100	633	30.180	0.949	0.050
	M4	100	304	15.181	0.761	0.050
	M5	100	303	15.086	0.757	0.050

4.5 稀疏扰动可视化

SparseAG 方法添加到图像的扰动位置会因图片的不同而发生改变. 为了能够更好地观察图像中扰动的添加位置以理解图像相对 DNN 的敏感像素点, 并且能够更加直观地观察生成扰动的稀疏性和对抗样本的真实性, 分别选取了 2 个图像应用 SparseAG 方法为其生成非目标对抗样本和目标对抗样本, 为对抗样本中所添加的扰动进行可视化.

图 2 为基于 SparseAG 方法进行非目标攻击的扰动可视化, 其中 3 列分别对应原始图像、对抗样本和稀疏扰动可视化图像. 可以看出, SparseAG 方法仅扰动图像的关键识别区域, 如第 1 行图中扰动主要添加在头部位置, 并且扰动的像素点个数较少, 添加的扰动具有较高的稀疏性, 生成的对抗样本与原始图像基本没有肉眼区分度.

图 3 为目标攻击下的扰动可视化, 最左边一列表示原始图像, 中间一列为针对 3 种不同目标标签生成的对抗样本, 第 3 列为对抗样本稀疏扰动可视化图像, 最后一列为不同目标标签下对图像添加稀疏扰动的公共区域. 可以看出, SparseAG 方法在目标攻击下, 尽管选取不同的目标标签, 图像中添加的扰动依旧和图像的关键识别区域有着高度相关性, 并且生成的对抗样本较为真实, 与原始样本基本没有肉眼区分度.

通过图 2、图 3 的扰动可视化图可以发现, 与密集对抗样本生成方法相比, SparseAG 方法能够提供更为精确的扰动位置, 通过扰动位置能够为对抗样本攻击 DNN 提供更好的解释性, 即 DNN 对图像中哪些像素点的鲁棒性较差.

4.6 实验结果分析

本文从对抗样本生成方法的攻击成功率、生成扰动的l_p范数$(p=0,1,2,\infty)$以及构造对抗样本所需时间这几个方面对 SparseAG 方法以及选择的几个对比方法进行实验评估. 实验结果表明, SparseAG 方法在非目标攻击和目标攻击场景下以及 CIFAR-10 和 ImageNet 数据集下均能够达到 100% 的攻击成功率, 并且生成扰动的稀疏性

(l_0 范数) 和整体扰动幅度 (l_1 和 l_2 范数) 都优于对比方法, 然而最大扰动幅度 (l_∞ 范数) 略高于 Homotopy-attack 以及 GreedyFool 方法. 在构造对抗样本所需时间方面, 在非目标攻击和目标攻击场景下, SparseAG 方法在 CIFAR-10 数据集上花费的时间较少, 但是在 ImageNet 数据集上花费的时间较长.

图 2　非目标攻击扰动可视化

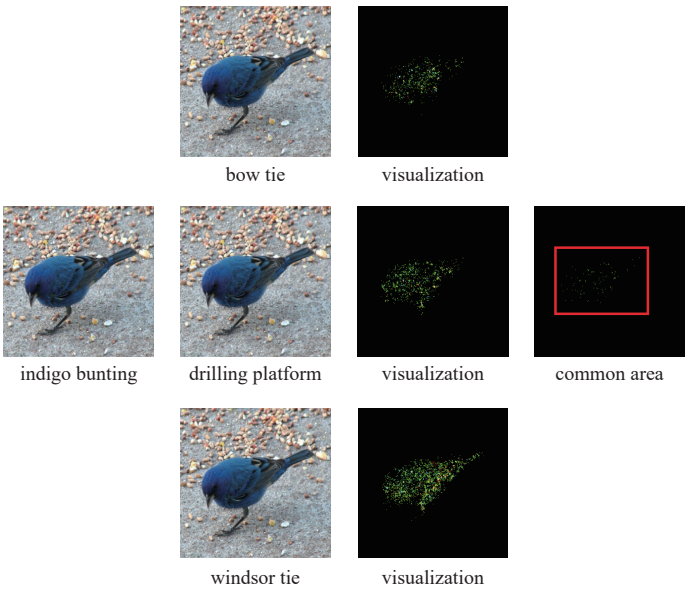


图 3　目标攻击扰动可视化

5 总　结

本文提出了一种基于稀疏扰动的对抗样本生成方法 SparseAG, 对图像样本添加扰动来构造 DNN 的对抗样本. 该方法基于损失函数关于输入图像的梯度值, 迭代地选取能够使得损失函数值最小的扰动添加到图像中, 并且设计了一种基于扰动重要性的扰动优化策略帮助方法跳出局部最优、减少冗余扰动以及冗余扰动幅度, 进一步提升扰动的稀疏性和对抗样本的真实性. 实验结果表明 SparseAG 方法可以有效地生成对抗样本, 且生成的扰动具备较高的稀疏性和较低的整体扰动幅度.

然而, SparseAG 方法依然存在一些不足. 虽然该方法生成扰动的稀疏性和扰动幅度表现较好, 但是构造对抗样本所需时间较长, 因此在未来的工作中, 需要进一步优化方法, 以在保持高稀疏性的同时, 减少对抗样本构造所需要的时间.

References:

[1] Qiu H, Ma YC, Li ZM, Liu ST, Sun J. BorderDet: Border feature for dense object detection. In: Proc. of the 16th European Conf. on Computer Vision. Glasgow: Springer, 2020. 549–564. [doi: 10.1007/978-3-030-58452-8_32]

[2] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: Proc. of the 2015 IEEE Conf. on Computer Vision and Pattern Recognition. Boston: IEEE, 2015. 3431–3440. [doi: 10.1109/CVPR.2015.7298965]

[3] Kim I, Baek W, Kim S. Spatially attentive output layer for image classification. In: Proc. of the 2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020. 9530–9539. [doi: 10.1109/CVPR42600.2020.00955]

[4] Wang F, Chen LR, Li C, Huang SY, Chen YJ, Qian C, Loy CC. The devil of face recognition is in the noise. In: Proc. of the 15th European Conf. on Computer Vision (ECCV). Munich: Springer, 2018. 780–795. [doi: 10.1007/978-3-030-01240-3_47]

[5] Hausler S, Garg S, Xu M, Milford M, Fischer T. Patch-NetVLAD: Multi-scale fusion of locally-global descriptors for place recognition. In: Proc. of the 2021 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Nashville: IEEE, 2021. 14136–14147. [doi: 10.1109/CVPR46437.2021.01392]

[6] Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, Fergus R. Intriguing properties of neural networks. arXiv: 1312.6199, 2014.

[7] Goodfellow IJ, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. arXiv:1412.6572, 2015.

[8] Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A. Towards deep learning models resistant to adversarial attacks. arXiv:1706.06083, 2019.

[9] Moosavi-Dezfooli SM, Fawzi A, Frossard P. DeepFool: A simple and accurate method to fool deep neural networks. In: Proc. of the 2016 IEEE Conf. on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 2574–2582. [doi: 10.1109/CVPR.2016.282]

[10] Cheng SY, Dong YP, Pang TY, Su H, Zhu J. Improving black-box adversarial attacks with a transfer-based prior. arXiv:1906.06919, 2020.

[11] Poursaeed O, Katsman I, Gao B, Belongie S. Generative adversarial perturbations. In: Proc. of the 2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 4422–4431. [doi: 10.1109/CVPR.2018.00465]

[12] Carlini N, Wagner D. Adversarial examples are not easily detected: Bypassing ten detection methods. In: Proc. of the 10th ACM Workshop on Artificial Intelligence and Security. Dallas: ACM, 2017. 3–14. [doi: 10.1145/3128572.3140444]

[13] Croce F, Hein M. A randomized gradient-free attack on ReLU networks. In: Proc. of the 40th German Conf. on Pattern Recognition. Stuttgart: Springer, 2018. 215–227. [doi: 10.1007/978-3-030-12939-2_16]

[14] Hayes J, Danezis G. Learning universal adversarial perturbations with generative models. In: Proc. of the 2018 IEEE Security and Privacy Workshops (SPW). San Francisco: IEEE, 2018. 43–49. [doi: 10.1109/SPW.2018.00015]

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

[16] Fan YB, Wu BY, Li TH, Zhang Y, Li MY, Li ZF, Yang YJ. Sparse adversarial attack via perturbation factorization. In: Proc. of the 16th European Conf. on Computer Vision. Glasgow: Springer, 2020. 35–50. [doi: 10.1007/978-3-030-58542-6_3]

[17] Karmon D, Zoran D, Goldberg Y. LaVAN: Localized and visible adversarial noise. In: Proc. of the 35th Int'l Conf. on Machine Learning. Stockholm: PMLR, 2018. 2507–2515.

[18] Croce F, Andriushchenko M, Singh ND, Flammarion N, Hein M. Sparse-RS: A versatile framework for query-efficient sparse black-box adversarial attacks. arXiv:2006.12834, 2022.

[19] Papernot N, McDaniel P, Jha S, Fredrikson M, Celik ZB, Swami A. The limitations of deep learning in adversarial settings. In: Proc. of the 2016 IEEE European Symp. on Security and Privacy (EuroS&P). Saarbruecken: IEEE, 2016. 372–387. [doi: 10.1109/EuroSP.2016.36]

[20] Croce F, Hein M. Sparse and imperceivable adversarial attacks. In: Proc. of the 2019 IEEE/CVF Int'l Conf. on Computer Vision. Seoul: IEEE, 2019. 4723–4731. [doi: 10.1109/ICCV.2019.00482]

[21] Dong XY, Chen DD, Bao JM, Qin C, Yuan L, Zhang WM, Yu NH, Chen D. GreedyFool: Distortion-aware sparse adversarial attack. In: Proc. of the 34th Int'l Conf. on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2020. 11226–11236.

[22] Su JW, Vargas DV, Sakurai K. One pixel attack for fooling deep neural networks. IEEE Trans. on Evolutionary Computation, 2019, 23(5): 828–841. [doi: 10.1109/TEVC.2019.2890858]

[23] Modas A, Moosavi-Dezfooli SM, Frossard P. SparseFool: A few pixels make a big difference. In: Proc. of the 2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 9079–9088. [doi: 10.1109/CVPR.2019.00930]

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

[25] Zhu MK, Chen TL, Wang ZY. Sparse and imperceptible adversarial attack via a homotopy algorithm. In: Proc. of the 38th Int'l Conf. on Machine Learning. PMLR, 2021. 12868–12877.

[26] Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images. Technical report, Department of Computer Science, University of Toronto, 2009.

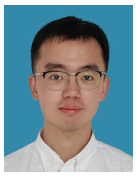
[27] Deng J, Dong W, Socher R, Li LJ, Li K, Li FF. ImageNet: A large-scale hierarchical image database. In: Proc. of the 2009 IEEE Conf. on Computer Vision and Pattern Recognition. Miami: IEEE, 2009. 248–255. [doi: 10.1109/CVPR.2009.5206848]

[28] Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. In: Proc. of the 2016 IEEE Conf. on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 2818–2826. [doi: 10.1109/CVPR.2016.308]

[29] Xu KD, Liu SJ, Zhao P, Chen PY, Zhang H, Fan QF, Erdogmus D, Wang YZ, Lin X. Structured adversarial attack: Towards general implementation and better interpretability. arXiv:1808.01664, 2019.

吉顺慧(1987－), 女, 博士, 副教授, CCF 专业会员, 主要研究领域为软件建模, 测试与验证.

张鹏程(1981－), 男, 博士, 博士生导师, CCF 高级会员, 主要研究领域为人工智能软件测试, 服务计算, 数据科学.



胡黎明(1997－), 男, 硕士生, CCF 学生会员, 主要研究领域为人工智能软件测试.

戚荣志(1980－), 男, 博士, 讲师, CCF 专业会员, 主要研究领域为实体识别, 关系抽取, 智能软件工程.