南京大学

研 究 生 毕 业 论 文

（申请硕士学位）

论 文 题 目　基于像素自动删减的稀疏对抗攻击及其在数据增强中的应用

作 者 姓 名　李金桥

专 业 名 称　计算机科学与技术

研 究 方 向　人工智能

指 导 教 师　申富饶　教授

2022 年 05 月 20 日

学　　号：MG1933035

论文答辩日期：2022 年 05 月 17 日

指 导 教 师：  　　（签字）

Sparse Adversarial Attack Based on Automatic Pixel Reduction and its Application in Data Augmentation

by

Jinqiao Li

Supervised by

Professor Furao Shen

A dissertation submitted to
the graduate school of Nanjing University
in partial fulfilment of the requirements for the degree of
MASTER
in
Computer Science and Technology

Department of Computer Science and Technology
Nanjing University

May 20, 2022

南京大学研究生毕业论文中文摘要首页用纸

毕业论文题目：基于像素自动删减的稀疏对抗攻击及其在数据增强中的应用

计算机科学与技术 专业 2019 级硕士生姓名：李金桥

指导教师（姓名、职称）：申富饶 教授

摘　要

近年来，越来越多的领域都通过深度学习取得了更好的效果，深度学习技术也已经广泛应用到高安全需求的场景中，如自动驾驶、人脸身份识别和恶意软件检测等。但深度学习自身却面临严峻的安全问题，这其中对抗攻击（Adversarial Attack）通过对输入样本进行微妙扰动，从而导致深度学习模型的预测出现错误。研究对抗攻击一方面可以深入理解神经网络的缺陷和行为，进一步帮助构建鲁棒的模型；另一方面也可以基于对抗攻击发掘更多的诸如隐私保护和数据增强这类正面的应用。因此对抗攻击逐渐成为深度学习安全领域的热门研究。

本文主要研究白盒场景下图像分类任务中的稀疏对抗攻击方法，并挖掘了其在模型训练中的正面应用。在稀疏对抗攻击的研究中，我们跳出了人为设计像素点重要性指标和贪心修改像素值的通用过程，使得哪些像素点被扰动完全由模型自动决定，从而提升了对抗扰动的稀疏性。此外，我们进一步发掘了稀疏对抗攻击的正面应用，将所提方法结合到数据增强中，通过迫使模型关注那些不敏感但是非常重要的区域，大大提升了模型的泛化性。本文的主要研究内容与贡献如下：

本文从神经网络剪枝任务中获得灵感，创新性地将自动剪枝技术与对抗攻击技术相结合，提出了基于像素自动删减的稀疏对抗攻击方法，简称AutoAdversary。本方法在一般的对抗攻击过程中添加一个包含可训练的编码器网络和近似二值化模块的分支，用于生成01-mask，进而在构建对抗样本的过程中利用mask来自动确定哪些像素点需要被扰动，最终在不降低攻击效果的前提下大大减小了扰动量。经实验验证，本方法在多个数据集上都达到了最优的效果。除此以外，本方法具备灵活性和通用性，在大多数一般对抗攻击方法的基础上都能进一步减小对抗扰动，因此可以视作通用的扰动删减框架。

本文进一步发掘了稀疏对抗攻击在模型训练中的正面应用，提出了一种基于稀疏对抗攻击的图像数据增强方法，简称 AdvMask。我们首先利用 AutoAdversary 来攻击分类模型，进而发现图像中最为敏感的像素点，通过在训练中按照一定的策略适当删除敏感像素周围的随机区域，迫使模型关注不敏感的重要特征以及全图上下文信息，大大提升了模型的泛化性。经实验验证，相较于现有的数据增强方法，AdvMask 在多个数据集和多个网络结构上都取得了更好的效果。

本文将所提方法应用于实际的环境中，搭建了一个准确率高、实时性强的在线花卉识别系统。该系统包括花卉实时识别、花卉知识科普以及趣味知识学习等功能，具有一定的科普和实用价值。相较于使用其他图像数据增强方法，模型的泛化性得到了进一步的提升，满足了实际系统的需求，也佐证了我们所提算法的实用性以及有较大的实际应用价值。

关键词： 对抗攻击，稀疏优化，图像数据增强，深度学习

南京大学研究生毕业论文英文摘要首页用纸

THESIS: Sparse Adversarial Attack Based on Automatic Pixel Reduction and its Application in Data Augmentation

SPECIALIZATION: Computer Science and Technology

POSTGRADUATE: Jinqiao Li

MENTOR: Professor Furao Shen

Abstract

In recent years, more and more fields have achieved better results through deep learning, and the technology has been widely used in scenarios with high security requirements, such as autonomous driving, face recognition and malware detection. However, deep learning faces serious security problems of its own, including adversarial attacks that cause the predictions of deep learning models to be wrong through subtle perturbations of input examples. On the one hand, the study of adversarial attacks can deeply understand the defects and behaviors of neural networks and further help to build robust models. On the other hand, more positive applications such as privacy protection and data augmentation can be explored based on adversarial attacks. Therefore, adversarial attacks have gradually become a hot research in deep learning security field.

This paper mainly studies the sparse adversarial attack method in image classification task in the white-box scene, and explores its positive application in model training. In the research of sparse adversarial attack, we jump out of the general process of artificially designing pixel importance evaluation metric and greedily modifying pixel value, so that which pixels are perturbed are automatically determined by the model, so as to improve the sparsity of adversarial perturbations. In addition, we further explore the positive application of sparse adversarial attack, combine the proposed method with data augmentation, and greatly improve the generalization of the model by forcing the model to pay attention to those insensitive but very important areas. The main research contents and contributions of this paper are as follows:

Inspired by the task of neural network pruning, this paper innovatively combined automatic pruning technology with adversarial attack, and proposed a sparse adversar-

ial attack method based on automatic pixel reduction, referred to as AutoAdversary. This method adds a branch including a trainable encoder network and an approximate binarization module in the general adversarial attack process to generate a 0-1 mask, and then uses the mask to automatically determine which pixels need to be perturbed. Finally, the perturbation is greatly reduced without reducing the attack effect. Experiments show that this method achieves the best effect on multiple datasets. In addition, this method has flexibility and versatility, and can further reduce the adversarial perturbations on the basis of most general adversarial attack methods. Therefore, it can be regarded as a general framework for reducing perturbations.

This paper further explores the positive application of sparse adversarial attack in model training, and proposes an image data augmentation method based on sparse adversarial attack, abbreviated as AdvMask. We first attack the classification model by using AutoAdversary, and then find the most sensitive pixels in the image. By appropriately deleting the random area around the sensitive pixels according to certain strategies in the training, we force the model to pay attention to the insensitive important features and the whole image context information, which greatly improves the generalization of the model. Experiments show that compared with the existing data enhancement methods, AdvMask has achieved better results in many datasets and network structures.

In this paper, the proposed method is applied to the actual environment, and an online flower recognition system with high accuracy and strong real-time is built. The system includes the functions of flower real-time recognition, flower knowledge popularization and interesting knowledge learning, which has certain popular science and practical value. Compared with other image data augmentation methods, the generalization of the model has been further improved, which not only meets the needs of the actual system, but also proves the practicability and great practical application value of our proposed algorithm.

keywords: Adversarial Attack, Sparse Optimization, Image Data Augmentation, Deep Learning

目　　录

插图清单

附表清单

第一章　绪论

1.1　研究背景与意义

近年来，随着各类数据的海量积累以及计算能力的大幅提升，机器学习特别是深度学习技术得到了快速的发展并且在人们的生产生活中得到了广泛的应用，例如计算机视觉[2],[3],[4]、自然语言处理[5]、语音识别[6]、个性化推荐[7]和企业风险控制[8]等。

特别的，深度学习技术也已经广泛应用到高安全需求的场景中，如自动驾驶[9],[10]、人脸身份识别[11]、恶意软件检测[12],[13]等。但深度学习自身却面临严峻的安全问题，比如：药饵攻击[14]在训练样本中掺杂少量恶意样本从而很大程度上降低了模型的准确率、后门攻击[15]在模型中嵌入后门从而使得模型对特殊输入的判断受到攻击方的控制、隐私窃取攻击[16]通过查询分析推断出训练数据以及模型参数。

除此以外，对抗样本攻击更是目前已知的、威胁最大的安全性问题之一。Szegedy 等人[17]首先在图像分类任务上发现了对抗样本的存在，他们通过在正常图像上添加人眼难以察觉的细微扰动，使得模型对其输出错误的分类结果。不仅如此，研究者在其他任务中也发现了对抗样本，例如：Kos 等人[18]提出了一种针对深度强化学习策略的对抗攻击方法；Xie 等人[19]提出了一种能够愚弄目标检测模型和语义分割模型的对抗攻击方法；在评估阅读理解系统的任务中，Jia 等人[20]成功地愚弄了斯坦福问题回答数据集（SQuAD[21]）上的 16 种模型。更为严重的是，研究者证明在真实物理世界中也存在对抗样本的威胁，比如：Eykholt 等人[22]制作了一款可以打印并粘贴在交通标志牌上的涂鸦贴纸，成功地愚弄了自动驾驶系统；此外，Komkov 等人[23]通过打印特制的矩形贴纸并将其粘贴在帽子上，成功地蒙骗了人脸识别系统。综上，对抗样本攻击成为了威胁人工智能系统安全的重要因素之一。

由于深度学习的理论研究尚未成熟，很难从数学层面精确解释深度学习模型的种种行为，因此直接提出对抗样本的防御方法是较为困难的。现有最有效的对抗训练防御方法[24]利用大批对抗样本来训练深度模型，但是仍存在过拟

合、训练数据利用不充分等问题。因此本文通过研究对抗样本攻击算法，帮助研究者深入了解现有深度模型的缺陷，为今后的对抗样本防御方法提供灵感和参考，进一步促进具有对抗鲁棒性的模型诞生。

并且，虽然对抗攻击表面上看起来是一种破坏性的工作，但是越来越多的研究者也都开始探寻基于对抗攻击的正面应用，比如：Zhu 等人[25]观察到对抗攻击算法可以利用深度模型来隐式编码大量有用信息，进而提出了基于深度学习的数字水印技术；Chen 等人[26]利用对抗样本攻击技术，通过在网络验证码上添加对抗扰动，有效打击了黑灰产的自动验证码识别；Xie 等人[27]将对抗样本攻击作为一种数据增强技术，在没有额外数据的情况下显著提升了模型的泛化性。

综上所述，研究对抗样本攻击算法一方面可以深入理解深度神经网络的缺陷和行为，为日后提出防御方法带来灵感和依据，另一方面也能够进一步发掘对抗样本攻击的正面应用，因此具有重要的研究意义和价值。

1.2 研究现状

本文主要围绕对抗攻击进行研究，并且进一步将所提攻击算法与图像数据增强相结合。因此下面将对这两个方面的研究现状进行介绍。

1.2.1 对抗攻击

对抗样本最初由 Szegedy 等人[17]提出，他们将构建对抗样本表示一个优化问题，即求最小化的扰动使得模型输出发生指定错误，并使用 L-BFGS 优化算法求解该问题。该工作提出后引发了大量的关注，越来越多的研究者开始探究对抗样本引发的安全问题。基于深度模型的线性假设，Goodfellow 等人[28]提出了快速符号梯度法（Fast Gradient Sign Method，FGSM），该算法利用分类损失在图像数据点上的梯度符号信息，仅需对输入图像进行一次修改即可快速构建对抗样本。后续大部分算法都在上述两种方法的基础上进行改进。例如：由于 FGSM 仅对梯度进行一次计算，某些情况下很难攻击成功，因此 Kurakin 等人[29]提出了迭代的快速符号梯度法（Iterative Fast Gradient Sign Method，I-FGSM），攻击成功率得到了极大的提高；进一步的，Dong 等人[30]提出的动量迭代快速符号梯度法（Momentum Iterative Fast Gradient Sign Method，MI-FGSM）在 I-FGSM 的基础上引入动量的思想，使得构建出的对抗

样本跳出局部最优；与 L-BFGS 类似，以 Carlini 与 Wagner 两人命名的 C&W 方法[31]同样将构建对抗样本表示为一个优化问题，通过设计更好的目标函数以及换元优化技巧（change-of-variables）取得了更优的结果。

上述提到的方法都仅适用于白盒场景，即模型结构、参数和训练数据等对于攻击方来说是完全可见和可用的。然而实际情况往往是黑盒场景，攻击方无法像白盒场景下通过计算输入图像的梯度来构建对抗样本。在黑盒情况下，根据攻击方能够获得的信息可以分为 soft-label 和 hard-label 两类场景。soft-label 是指虽然攻击方无法获取模型信息，但是能够得到模型的输出向量。soft-label 场景下最具有代表性的方法是由 Chen 等人[32]提出的零阶优化黑盒攻击（ZOO），该方法旨在通过模型的输出向量利用有限差分来估计输入图像的梯度，进而直接利用估计的梯度修改输入图像。后续提出的自动编码器零阶优化方法（AutoZOOM）[33]极大地降低了所需估计梯度的次数，提高了效率。在信息量更少的 hard-label 场景下，攻击方连模型的输出向量都不再能获得，而仅能知晓模型的决策结果，比如图像分类任务中的分类结果。由于模型的决策结果是离散的，无法再通过有限差分来估计输入数据的梯度信息，因此类似 ZOO 的方法不再有效。为了解决这个问题，Brendel 等人[34]提出的边界攻击（Boundary Attack）通过在模型的决策超平面上使用随机游走，在仅有模型决策结果的情况下成功构建了对抗样本。进一步的，Cheng 等人[35]表明，hard-label 攻击问题可以转化为 soft-label 场景下的一个优化问题，使得目标函数对于输入图像是连续的，从而可以再次基于有限差分法估计梯度来生成对抗样本。

除此以外，还有研究者开始考虑在实际的物理环境中构建对抗样本。实际物理环境与数字环境最大的区别在于，数字环境中的对抗攻击都默认对抗样本可以直接进入到模型的输入层，而物理环境中则不然。比如在人脸识别等视觉任务中，攻击方必须先将对抗样本通过打印粘贴等步骤放置到真实世界中，再由摄像头拍摄画面送入到模型输入层，因此过程中存在颜色、角度以及位置等多重失真，这往往导致对抗攻击难以成功。通过添加打印颜色损失、全变分（Total Variation，TV）损失，Sharif 等人[36]第一次构建出了在真实物理世界中能够愚弄人脸识别系统的对抗眼镜框，攻击者戴上眼镜框后将不能被人脸系统正确识别。进一步的，Athalye 等人[37]将物理世界的变换如光线变化、旋转平移等建模到构建对抗样本的过程中，提出了期望变换算法（Expectation Over Transformation，EOT），使得对抗样本在真实物理世界中具备一定的鲁棒性，

后续大多物理世界的方法都基于 EOT 的思想。

本文主要关注白盒场景下数字图像分类任务中的对抗样本攻击，从而促进人们对深度神经网络的理解，为今后的对抗样本防御方法提供灵感和参考。如前文所述，白盒场景下的对抗样本攻击已经有了大量的研究工作，这类工作的目的是使得对抗扰动尽可能小，即图像的像素修改量要尽量小。但是现有的大部分对抗样本攻击都对整张图像即所有像素点进行扰动，然而参考 Simonyan 等人[38]提出的显著图（Saliency Maps）工作，图像中不同像素点对于模型分类结果的影响是大不相同的。也就是说，图像中某些像素点是无需扰动的，因此大多现有对抗攻击方法所产生的对抗扰动能够在保证攻击效果的前提下进一步被减小。

目前已有研究者提出了仅扰动图像中部分像素点的对抗攻击方法，称为稀疏对抗攻击（Sparse Adversarial Attack）。比如，Papernot 等人[39]就根据上述显著图的思想，提出了基于雅克比矩阵的对抗显著图攻击（Jacobian-based Saliency Map Attack，JSMA），仅选取那些最重要的像素点进行改动，从而产生了稀疏的对抗扰动，然而每一轮迭代中计算显著图的效率过低导致 JSMA 在尺寸较大的图像上几乎无法运行。后续提出的大多稀疏对抗攻击[31],[40],[41]都可归纳为一个通用过程：首先人为设计一个衡量像素点重要性的指标，然后使用贪心的方法，在迭代过程中一边计算重要性一边修改当前最重要的一批像素点，最终达到修改尽可能少的像素点就可以使得模型出错的目的。这些不同的方法都试图设计更好的重要性指标，然而人为设计的重要性指标不一定能够适用于所有的情况，并且贪心的修改方法可能并非最优。如何跳出人为设计重要性指标以及贪心修改的通用过程，转而利用模型本身来自动确定哪些像素点需要被扰动，是一个值得探究的问题。综上所述，稀疏对抗攻击方法在性能和效率上仍然存在较大的改进空间。

1.2.2 图像数据增强

众所周知，深度学习在计算机视觉领域中做出了巨大的贡献，在许多具有挑战性的视觉任务中取得了最好的性能。这些改进一方面是由于卷积神经网络（Convolutional Neural Network，CNN）等优秀网络结构的出现，一方面也是因为模型的容量和参数量变得越来越大，使得其能够学习到复杂的特征表示。现代的神经网络通常包含数千万甚至数亿个需要学习的参数，这能够带来很强的特征表示能力，但是同样也增大了过拟合（Over-fitting）的风险，容易导致

模型的泛化性降低。

一般来说，降低过拟合风险的方法分为正则化（Regularization）和数据增强（Data Augmentation）两类。正则化一直都是训练神经网络模型的常用技术，具有代表性的是 Krizhevsky 等人于 2012 年提出的 Dropout[2]，该方法在训练过程中按照一定的概率随机丢弃隐藏单元（hidden neuron）的输出，可以理解为一种 bagging 集成学习策略[42]，有效提升了模型的泛化性。后续也出现了许多在 Dropout 基础上进一步改进的正则化方法[43],[44]。除此以外，Dropblock[45]和 Droppath[46]等方法在训练过程中根据不同的策略在参数中添加噪声，也提高了模型的泛化能力。

与正则化相对应的，数据增强也能够有效降低模型的过拟合风险，在深度学习模型的训练中同样被广泛应用。最常用的图像数据增强是一类称为空域变换（spatial transformation）的方法，这类方法主要利用图像平移、旋转、翻转、裁剪以及亮度调整等多种变换，从现有的数据中人为地扩大训练数据集，达到模拟真实世界数据的目的。比如，LeCun 等人[47]在训练 LeNet5 进行光学字符识别时就利用了包括平移、缩放、压缩和水平剪切等在内的仿射变换，提高了模型的泛化性能；同样的，Wu 等人[48]使用了更为激进的色彩变换、光晕、镜头失真以及拉伸等数据增强方法，在 ImageNet 数据集上取得了优秀的性能。

除了空域变换，近期提出的一类基于信息删除（information dropping）的数据增强方法如 Cutout[49]、RandomErasing[50]和 GridMask[51]等都取得了更好的性能。这类方法通过某些策略在训练过程中随机删除输入图像上的部分区域，迫使神经网络模型学习到原本不太会关注到的信息，鼓励模型更好地利用图像的全部上下文，而不是依赖于一组特定的、局部的视觉特征，从而显著提高了模型的泛化性。更加重要的是，这类方法与空域变换等其他数据增强方法可以相互结合，因此使用更加广泛。但是目前这类方法都是按照一定的策略在全图中随机产生删除区域，只不过是删除区域的位置、大小以及区域的个数有一些差别。现有方法对于一些复杂的图像仍然无法很好地处理，有可能完全删除了图像中的对象或者仅删除了背景。如何减少这种因为全图随机性带来的低质量样本，通过某些前置方法找到每张图像最需要被删除区域是一个值得探究的问题。综上所述，基于信息删除的图像数据增强方法仍然存在可改进的空间。

1.3 本文研究内容

本文主要研究白盒场景下图像分类任务中的稀疏对抗攻击方法，跳出了人为设计像素点重要性指标和贪心修改像素值的通用过程，使得哪些像素点被扰动完全由模型自动决定。此外，本文进一步发掘了稀疏对抗攻击的正面应用，将所提方法结合到数据增强中，大大提升了模型的泛化性。最后，本文所提方法被应用到实际的系统当中，通过实践验证了其有效性。本文的主要研究内容总结如下：

- 本文创新性地将神经网络自动剪枝技术与对抗攻击技术相结合，提出了基于像素自动删减的稀疏对抗攻击方法，简称 AutoAdversary。本方法在一般的对抗攻击过程中添加一个包含可训练的编码器网络和二值化模块的分支，用于生成 01-mask，进而在构建对抗样本的过程中利用 mask 来自动确定哪些像素点需要被扰动，最终在不降低攻击效果的前提下大大减小了扰动量。CIFAR-10 和 ImageNet 数据集上的实验都表明本文提出的方法具有最优的效果。并且由于该方法具备灵活性和通用性，在大多数一般攻击方法的基础上都能进一步减小扰动，因此可以视作通用的扰动删减框架。
- 本文进一步发掘了稀疏对抗攻击的正面应用，提出了一种基于稀疏对抗攻击的图像数据增强方法，简称 AdvMask。我们首先利用 AutoAdversary 来攻击分类模型以发现图像中最敏感的像素，紧接在训练中按照一定的策略适当删除敏感像素周围的随机区域，迫使模型关注不敏感的重要特征以及全图上下文信息，提升了模型的泛化性。相较于现有的数据增强方法，本方法在多个数据集上都取得了更好的效果。
- 本文将所提方法应用于实际的环境中，搭建了一个准确率高、实时性强的在线花卉识别系统。该系统包括花卉实时识别、花卉知识科普以及趣味知识学习等功能，具有一定的科普和实用价值。相较于使用其他图像数据增强方法，模型的泛化性得到了进一步的提升，满足了实际系统的需求。该系统的实现进一步说明了本文所提方法具有较强的实际应用价值。

1.4 本文结构安排

本文主要研究了白盒场景下图像分类任务中的稀疏对抗攻击，创新性地将神经网络自动剪枝技术与对抗攻击技术相结合，达到了最优的效果。进一步

的，本文将提出的方法结合到图像数据增强中，提升了模型的泛化性，并成功在实际的系统中得到应用。全文共分为六章，第一章为绪论部分，主要介绍了对抗样本攻击的研究背景和意义，同时还介绍了当前研究现状以及存在的问题；第二章为相关工作部分，详细介绍了对抗攻击相关的基本概念和代表工作，以及基于信息删除的图像数据增强的代表方法。第三章介绍了本文提出的基于像素自动删减的稀疏对抗攻击方法，包括研究动机和模型详细设计，最后通过实验验证了所提出方法的有效性；第四章介绍了基于稀疏对抗攻击的图像数据增强方法，包括提出的动机以及算法的详细流程，最后通过实验验证了数据增强的有效性和优越性；第五章介绍了本文所提方法在实际的花卉识别系统中的应用；第六章对全文进行总结回顾，以对未来的工作进行展望。

第二章　相关工作

本章主要介绍本文所涉及的背景知识与相关代表工作。首先介绍了对抗样本的定义和基本概念，包括与之相关的术语与度量指标。然后重点介绍了具有代表性的对抗样本攻击算法，并对其优缺点进行简要分析。并且本文将稀疏对抗攻击结合到了数据增强中，因此也对具有代表性的基于信息删除的数据增强算法进行了介绍，为后续章节中提出的算法提供了理论基础。

2.1　对抗样本

2.1.1　定义及基本概念

近年来，深度学习得到了快速的发展并且在人们的生产生活中得到了广泛的应用。然而大多数研究者都只关注深度学习模型的性能，却不重视模型的鲁棒性和安全性。Szegedy 等人[17]发现，如果在正常图像上添加细微的扰动，模型可能将输出错误的分类结果，并以此提出了对抗样本的概念。对抗样本是一类由攻击方精心设计的特殊样本，这类样本在像素空间上往往与原始样本无多大差别，但是却导致模型无法输出正确的结果。在这之后，关于对抗样本的研究飞速发展，从简单的图像分类任务到复杂的强化学习、自然语言处理、目标检测等任务，从实验性质的数字环境到实际应用的真实物理环境，均出现了对抗攻击方法。特别是在高安全需求的应用中如自动驾驶、人脸识别等，对抗样本的存在限制了深度学习模型在这类场景中的应用。因此越来越多的研究者开始重视深度学习所面临的安全问题，从攻击和防御两个彼此对立而又相互促进的角度来开展对抗样本的研究。

2.1.2　术语介绍

由于大量研究者开始重视深度学习模型的安全性，对抗样本得到了飞速的发展，如今已经有了多种分类。比如根据攻击方所掌握的信息来分，可以将对抗样本攻击分为白盒攻击和黑盒攻击；根据模型出错的类型又可以分为有目标

攻击和无目标攻击等。因此，为了进一步清楚地介绍对抗样本的相关概念和方法，本文详细列举了与对抗样本相关的常用术语：

- **对抗样本（Adversarial Example）**：对抗样本是一种通过在正常干净的输入样本中加入人眼近似不可察觉的扰动而使深度学习模型严重失效的样本，这种扰动不是随机的，而是精心设计的。
- **对抗扰动（Adversarial Perturbation）**：对抗扰动一般简称扰动，是加到干净输入样本上使其成为对抗样本的微小噪声。在绝大多数情况下，扰动是人眼不可察觉的。
- **白盒攻击（White-box Attack）**：白盒攻击是指攻击方知晓与模型相关的一切信息，包括训练数据、网络结构、损失函数、模型参数等。在此种情况下，对抗样本通常通过模型损失相对输入的梯度来构建，就像训练参数一样“训练”对抗样本。基于梯度的攻击的代表方法有 FGSM[28]和 C&W[31]等。
- **黑盒攻击（Black-box Attack）**：黑盒攻击与白盒攻击正好相反。攻击者对网络结构、损失函数和模型参数一无所知，只能利用模型的输出来构建对抗样本。特别的，根据输出的具体类型又可以分为 soft-label[32]和 hard-label[34]两种情况。
- **迁移性（Transferability）**：迁移性是对抗样本的共同特性，即从一个模型中构建的对抗样本可以欺骗不同结构的其他模型，即使这些模型在不同的数据集上训练。因此，攻击方可以在已知的模型上构建对抗样本，转而攻击未知的模型。
- **无目标攻击（Non-targeted Attack）**：无目标攻击要求模型出错即可，不对出错的具体类型做出限制。比如图像分类任务中，模型将对抗样本分类为任意其他类别标签即可；人脸识别任务中，模型将人脸图像识别为任意其他身份即可。
- **有目标攻击（Targeted Attack）**：有目标攻击要求模型产生指定的错误。比如图像分类任务中，攻击者首先指定一个目标类别标签（Target-label），然后模型将对抗样本分类为该特定的类别标签；人脸识别任务中，模型将人脸图像识别为攻击者指定的身份。通常来讲，因为有目标攻击有更严格的限制条件，所以比无目标攻击更加困难。

2.1.3 性能度量

对抗攻击的目的是构建一个与原始样本在视觉上几乎没有区别的对抗样本，并使得模型出错。所以对抗攻击方法是否有效，需要从对抗样本的攻击能力以及对抗样本的隐蔽性两个角度来衡量。

在攻击能力方面，攻击成功率（Attack Success Rate，ASR）使用最广泛，该指标直接衡量了欺骗模型的能力。简单来讲，ASR 是指在所有构建的对抗样本中，满足预设攻击条件的对抗样本在总样本中的占比。比如在进行无目标攻击时，ASR 就等于模型在所有对抗样本上的分类错误率 ER（Error Rate），即预测分类标签与真实分类标签不同的对抗样本的占比:

$$\mathrm{ER} = \frac{1}{M}\sum_{i=1}^{M}\mathbb{I}(\hat{y}_i \neq y_{\mathrm{true}i}), \tag{2-1}$$

其中 M 是总样本个数，$\hat{y}_i$ 和 $y_{\mathrm{true}i}$ 分别表示第 i 个样本的模型预测标签和真实分类标签。与之对应的，在进行有目标攻击时，使用匹配率 MR（Matching Rate）来表示 ASR，其中匹配率指预测分类标签与目标分类标签（Target-label）相同的对抗样本的占比：

$$\mathrm{MR} = \frac{1}{M}\sum_{i=1}^{M}\mathbb{I}(\hat{y}_i = y_i^*), \tag{2-2}$$

其中 y_i^* 表示第 i 个样本的目标分类标签。总之，ASR 越高则表明对抗攻击方法的攻击能力越强。

在隐蔽性方面，需要使用距离度量来量化原始样本与对抗样本之间的相似性。在图像像素空间内，l_p 范数是人类感知距离的合理近似[31]，因此对抗样本与原始样本之间的距离可以定义为：

$$d(\boldsymbol{x}', \boldsymbol{x}) = \|\boldsymbol{x}' - \boldsymbol{x}\|_p, \tag{2-3}$$

其中 $\boldsymbol{x}$ 和 $\boldsymbol{x}'$ 分别表示原始样本与对抗样本，l_p 范数 $\|\cdot\|_p$ 定义如下：

$$\|\boldsymbol{\delta}\|_p = (\sum_{i=1}^{n}|\delta_i|)^{\frac{1}{p}}, \tag{2-4}$$

其中 $\boldsymbol{\delta} = \boldsymbol{x}' - \boldsymbol{x}$ 表示所添加的对抗扰动（Adversarial Perturbation）。l_0 范数、

l_2 范数和 l_∞ 范数是最常用的三种指标，l_0 范数测量了 $x_i \neq x_i'$ 的个数，即 l_0 范数对应于被改变的像素点的个数；l_2 范数测量了原始样本 $\boldsymbol{x}$ 和对抗样本 $\boldsymbol{x}'$ 之间的欧式距离；l_∞ 范数表示为 $\|\boldsymbol{x}' - \boldsymbol{x}\|_\infty = \max(|x_1' - x_1|, ..., |x_n' - x_n|)$，即任意像素点的最大变化量，也称为最大扰动幅度（Perturbation Magnitude）。除开最常用的 l_p 范数以外，也有研究者从其他角度来度量原始样本与对抗样本的相似性，比如 Wang 等人[52]提出的结构相似性（Structural Similarity，SSIM）以及 Rozsa 等人[53]提出的心理测量感知度评分（Psychometric Perceptual Adversarial Similarity Score，PASS）。目前并没有工作表明哪种相似性度量是最完美的，因此本文主要使用最常用的 l_p 范数来度量对抗样本和原始像本的相似性。

2.2 稠密对抗攻击

本节将介绍对抗样本攻击中最具有代表性的几类方法。由于大多数对抗攻击方法都以 l_2 范数或者 l_∞ 范数作为扰动大小的度量，因此这类方法最终构建的对抗样本将修改整张图像的全部像素点，因此也可以被称为稠密对抗攻击（Dense Attack），一般就简称为对抗攻击。由于稠密对抗攻击是其他类型攻击的基础，因此本文首先介绍最具有代表性的稠密对抗攻击方法。

2.2.1 L-BFGS

Szegedy 等人[17]首先提出了能够欺骗深度神经网络模型的对抗攻击方法。给定原始图像向量 $\boldsymbol{x}$，该方法找到与 $\boldsymbol{x}$ 在 l_2 范数距离上最为接近的对抗样本向量 $\boldsymbol{x}'$，并且使得 $\boldsymbol{x}'$ 被模型分类为目标类别标签 y^*。具体的，他们将该问题表示为：

$$\begin{aligned} \min_{\boldsymbol{\delta}} \quad & \|\boldsymbol{\delta}\|_2 \\ \text{s.t.} \quad & f(\boldsymbol{x} + \boldsymbol{\delta}) = y^* \\ & \boldsymbol{x} + \boldsymbol{\delta} \in [0,1]^n, \end{aligned} \tag{2-5}$$

其中 $\boldsymbol{\delta} = \boldsymbol{x}' - \boldsymbol{x}$ 为对抗扰动，$f(\cdot)$ 是将 n 维图像向量映射到离散分类标签集合的分类器。图像的像素值范围原本在 $[0, 255]$ 区间，但在实际处理时一般都会归一化到 $[0, 1]$ 区间。因此为了保证对抗样本是合法的图像，需要保证对抗样本的像素值处于 $[0, 1]$ 区间。由于 $f(\cdot)$ 是一个非线性且非凸的函数，所以问题 (2-5)

难以直接求解。因此 Szegedy 等人将该问题转化为：

$$\begin{aligned} \min_{\boldsymbol{\delta}} \quad & c \cdot \|\boldsymbol{\delta}\|_2 + J(\boldsymbol{x}+\boldsymbol{\delta}, y^*) \\ \text{s.t.} \quad & \boldsymbol{x}+\boldsymbol{\delta} \in [0,1]^n, \end{aligned} \tag{2-6}$$

其中 $J(\cdot,\cdot)$ 是模型训练时所用的损失函数，一般为交叉熵损失（Cross-entropy Loss）。通过这种转化，即可使用带箱形约束（Box-constrained）的 L-BFGS 算法求解问题 (2-6)，从而得到问题 (2-5) 的近似解。在细节方面，通过对常量 $c>0$ 执行线性搜索，最终得到 l_2 范数距离上最接近原始样本的对抗样本。该方法的提出揭露了深度学习的盲点和缺陷，因此越来越多的人开始关注对抗样本的研究。

2.2.2 FGSM 及其拓展

L-BFGS 是一种构建对抗样本的有效方法，但该方法的计算量太大导致速度太慢。基于模型的线性假设，Goodfellow 等人[28]提出了快速符号梯度法（Fast Gradient Sign Method，FGSM），该算法利用分类损失在图像上的梯度符号信息，仅需对输入图像进行一次修改即可快速构建对抗样本，大大提升了效率。具体来讲，给定原始图像 $\boldsymbol{x}$ 以及真实分类标签 y_{true}，对抗样本 $\boldsymbol{x}'$ 可以被表示为：

$$\boldsymbol{x}' = \boldsymbol{x} + \epsilon \cdot \text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{x}, y_{\text{true}})), \tag{2-7}$$

其中 $J(\cdot,\cdot)$ 是模型训练时所用的损失函数，$\text{sign}(\cdot)$ 表示符号函数，$\epsilon>0$ 是一个用于控制像素最大扰动幅度的常量，也即是扰动的 l_∞ 范数。设置的扰动幅度越大，攻击成功率越高但是越容易被察觉；反之，攻击成功率越低但越不容易被察觉。总之，该方法通过损失函数的梯度符号来确定图像像素点的变化方向，最终所有的像素都会增大或减小相同的量 ϵ。

在这之后，很多研究者基于 FGSM 提出了新的对抗攻击方法。比如，Kurakin 等人[54]提出的 OTCM 即是 FGSM 的一种进行有目标攻击的变体。具体的，该方法将 (2-8) 修改为：

$$\boldsymbol{x}' = \boldsymbol{x} - \epsilon \cdot \text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{x}, y^*)), \tag{2-8}$$

其中 y^* 是指定的目标标签，因此该方法可以构建出被模型分类为指定类别标签

的对抗样本。

2.2.3 I-FGSM 及其拓展

虽然 FGSM 可以快速构建对抗样本，但是由于模型并非完全线性，因此沿 FGSM 确定的方向移动过长可能会偏离梯度方向，所以此时即便是增大设定的扰动幅度，也无法提高攻击成功率。因此 Kurakin 等人[29]尝试在 FGSM 上迭代，提出了迭代的快速符号梯度法（Iterative Fast Gradient Sign Method，I-FGSM），该方法每次迭代时只移动一小步，每次迭代后再调整方向。具体的，I-FGSM 的计算过程如下：

$$\boldsymbol{x}'_0 = \boldsymbol{x}, \tag{2-9}$$

$$\boldsymbol{x}'_{t+1} = \text{Clip}_{\boldsymbol{x},\epsilon}\{\boldsymbol{x}'_t + \alpha \cdot \text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{x}'_t, y_{\text{true}}))\}, \tag{2-10}$$

其中 $\boldsymbol{x}'_t$ 表示第 t 次迭代构建的对抗样本；α 表示每一轮迭代的移动步长；$\text{Clip}_{x,\epsilon}\{\cdot\}$ 是裁剪函数，用于将对抗样本限制以在原始样本 $\boldsymbol{x}$ 为圆心，以 ϵ 为半径的 l_∞ 范数球内，目的是为了使得最终的对抗扰动仍然处于预定的范围内。实验结果显示，在同样的扰动幅度 ϵ 的限制下，I-FGSM 比 FGSM 的攻击成功率更高。

Madry 等人[24]在 I-FGSM 的基础上引入了随机初始化，进而提出了投影梯度下降（projected gradient descent，PGD）方法。具体的，不同于式 (2-9)，该方法的初始点并非原始图像所在点，而是以原始图像为原点、ϵ 为半径的 l_∞ 范数球内的随机一点。I-FGSM 可以看作是 PGD 的一个特殊形式。

上述方法与模型训练中的梯度下降法有很大的相似性，因此与梯度下降法类似，I-FGSM 的缺点之一也是容易陷入局部最优。因此 Dong 等人[30]提出了动量迭代快速符号梯度法（Momentum Iterative Fast Gradient Sign Method，MI-FGSM），在 I-FGSM 的基础上引入动量的思想，使得构建出的对抗样本跳出局部最优。具体的，其构建对抗样本的过程如下：

$$\boldsymbol{g}_{t+1} = \mu \boldsymbol{g}_t + \frac{\nabla_{\boldsymbol{x}} J(\boldsymbol{x}'_t, y_{\text{true}}))}{||\nabla_{\boldsymbol{x}} J(\boldsymbol{x}'_t, y_{\text{true}}))||_1}, \tag{2-11}$$

$$\boldsymbol{x}'_{t+1} = \boldsymbol{x}'_t + \frac{\epsilon}{T}\text{sign}(\boldsymbol{g}_{t+1}), \tag{2-12}$$

其中 $\boldsymbol{g}_t$ 以衰减系数 μ 收集了前 t 轮迭代的梯度信息，$\frac{\epsilon}{T}$ 可以看作每一轮迭代

的移动步长，T 表示最大迭代次数，ϵ 表示设定的最大扰动幅度。通过引入动量，MI-FGSM 可以稳定更新方向，避免陷入局部最优。其实验表明，无论在白盒场景还是黑盒场景下，MI-FGSM 都比 I-FGSM 有更高的攻击成功率。

2.2.4 C&W

与 L-BFGS 类似，以 Carlini 与 Wagner 两人命名的 C&W 方法[31]同样将构建对抗样本表示为一个优化问题，通过设计更好的优化目标函数以及换元优化技巧（change-of-variables）取得了更优的结果。具体的，该方法解决如下的优化问题：

$$\begin{aligned} \min_{\boldsymbol{\delta}} \quad & ||\boldsymbol{\delta}||_2^2 + c \cdot g(\boldsymbol{x}+\boldsymbol{\delta}) \\ \text{s.t.} \quad & \boldsymbol{x}+\boldsymbol{\delta} \in [0,1]^n, \end{aligned} \tag{2-13}$$

其中 $c>0$ 是正则化参数，在优化过程中由二分搜索得到，$g(\cdot)$ 是自定义的目标函数。若 $g(\cdot)$ 为训练模型的损失函数，则 C&W 退化为 L-BFGS 方法。实验表明，以下形式的 $g(\cdot)$ 函数最为有效：

$$g(\boldsymbol{x}') = \max(\max_{i \neq y^*}(Z(\boldsymbol{x}')_i) - Z(\boldsymbol{x}')_{y^*}, -\kappa), \tag{2-14}$$

其中 $Z(\cdot)$ 表示模型 softmax 前一层的输出向量，也称为 logits 层输出向量，y^* 表示目标类别标签。很容易可以看出当 $\max_{i \neq y^*}(Z(\boldsymbol{x}')_i) - Z(\boldsymbol{x}')_{y^*} \leq 0$ 时，表明目标攻击已经成功；当 $\max_{i \neq y^*}(Z(\boldsymbol{x}')_i) - Z(\boldsymbol{x}')_{y^*} > 0$ 时，表明模型对输入的预测标签不为 y^*，即攻击并未成功。此外，可以适当增加 κ 以使得对抗样本在错误分类时具有更大的置信度，提升攻击效果。

除此以外，C&W 还去除了箱形约束（Box-constrained）。2.2.1节中已经提到，必须保证加上扰动后的对抗样本 $\boldsymbol{x}'$ 在合法的图像空间中，即 $\boldsymbol{x}' \in [0,1]^n$。C&W 定义了临时变量 $\boldsymbol{w} \in \mathbb{R}^n$，并使用换元的方式来通过优化无约束的临时变量 $\boldsymbol{w}$ 进而优化有约束的扰动变量 $\boldsymbol{\delta}$，$\boldsymbol{\delta}$ 与 $\boldsymbol{w}$ 的关系如下：

$$\delta_i = \frac{1}{2}(\text{thanh}(w_i)+1) - x_i, \tag{2-15}$$

从上式可以发现 $0 < x_i + \delta_i < 1$，因此构建出的对抗样本 $\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{\delta}$ 已经处于合法的图像空间内。因此，问题 (2-13) 中的箱形约束可以被移除，就可以使用其他不支持箱形约束的优化算法如 Adam[55]，可以更快速和更有效地构建对抗

样本。

C&W 具有较强的攻击能力，许多对抗防御方法如蒸馏[56]都变得无效。所以 C&W 经常被用作测试防御有效性的基准之一。上述 C&W 特指最小化 l_2 范数的攻击，除此以外 C&W 还可以拓展到最小化 l_0 范数的攻击，这将在后续2.3.2节的稀疏对抗攻击中详细介绍。

2.3 稀疏对抗攻击

2.2节中介绍的都是稠密对抗攻击，即构建的对抗样本会修改整张图像的全部像素点。然而参考 Simonyan 等人[4]提出的显著图（Saliency Maps）工作，图像中不同像素点对于模型分类结果的影响是大不相同的。也就是说，图像中某些像素点是无需扰动的，因此也有一些对抗攻击方法仅扰动图像中的部分重要的像素点，这类方法被称为稀疏对抗攻击（Sparse Attack）。下面将首先介绍目前已有的具有代表性的稀疏对抗攻击方法。

2.3.1 JSMA

首先由 Papernot 等人[39]提出了基于雅克比矩阵的对抗显著图攻击（Jacobian-based Saliency Map Attack，JSMA），该方法考虑扰动尽可能少的像素点来生成对抗样本。基于显著图的思想，JSMA 定义了对抗性显著图（Adversarial Saliency Map），然后仅修改对分类结果影响最大的部分像素点。具体的，给定原始图像向量 x 和目标分类标签 y^*，对抗性显著图 $\mathcal{S}(\boldsymbol{x}, y^*)$ 的定义如下：

$$S(\boldsymbol{x}, y^*)[i] = \begin{cases} 0, \text{ if } \frac{\partial Z_{y^*}(\boldsymbol{x})}{\partial x_i} < 0 \text{ or } \sum_{j \neq y^*} \frac{\partial Z_j(\boldsymbol{x})}{\partial x_i} > 0 \\ (\frac{\partial Z_{y^*}(\boldsymbol{x})}{\partial x_i}) | \sum_{j \neq y^*} \frac{\partial Z_j(\boldsymbol{x})}{\partial x_i} |, \quad \text{otherwise}, \end{cases} \tag{2-16}$$

其中 $Z(\cdot)$ 表示模型 logits 层的输出，基于该输出对于输入像素的梯度信息，$\mathcal{S}(\boldsymbol{x}, y^*)[i]$ 可以用来估计原始图像 $\boldsymbol{x}$ 中第 i 个像素点的重要性。因此 $\mathcal{S}(\boldsymbol{x}, y^*)$ 一定程度上表明了攻击者应该扰动哪些像素点，以最有效地影响模型输出的预期变化。JSMA 使用迭代的方式，每一轮迭代中都计算当前图像的对抗性显著图，然后基于对抗性显著图修改当前最重要的少数几个像素点，一直重复这个过程直到当前图像的预测分类标签变为目标标签 y^* 或者达到最大迭代次数。

JSMA 可以在修改少量的像素点的情况下构建对抗样本，但是每一轮迭代都需要计算雅可比矩阵和对抗性显著图，使得该方法的计算量过大，不适用于

类似 ImageNet 数据集中的大尺寸图像。

2.3.2 C&W-l_0

在 2.2.4节中已经介绍了 C&W 方法，该方法的基础版本将 l_2 范数作为距离度量，试图找到与原始样本的 l_2 范数距离最小的对抗样本，这里称为 C&W-l_2。Carlini 与 Wagner[31]仍然将 C&W 拓展到了最小化 l_0 范数的攻击，即最小化扰动像素点的个数，这里称为 C&W-l_0。由于 l_0 范数是不可微的，无法像最小化 l_2 范数那样使用梯度下降法，因此 C&W-l_0 与 JSMA 一样使用贪心迭代的方法，在每次迭代中先找到部分对模型输出没有太大影响的像素点，然后固定这些像素点，并让其值保持与原始图像中的值一致，紧接着在其他未固定的像素点集合上继续迭代。

具体来讲，C&W-l_0 首先将所有像素点放入可修改像素点集合 $\mathcal{A}$ 中。在每一轮迭代中，首先调用基础的 C&W-l_2 攻击方法在集合 $\mathcal{A}$ 上进行攻击，即只允许修改集合 $\mathcal{A}$ 中的像素点；然后令 $\boldsymbol{\delta}$ 表示当前扰动，$\boldsymbol{x}$ 表示原始图像，则 $\boldsymbol{g} = \nabla J(\boldsymbol{x} + \boldsymbol{\delta})$ 表示损失值相对于当前输入图像的梯度。计算需要固定的像素点序号 i：

$$i = \arg\min_i |g_i||\delta_i|, \tag{2-17}$$

紧接着固定第 i 个像素点，将其值与原始图像中保持一致，并将其从集合 $\mathcal{A}$ 中移除。换句话说，C&W-l_0 基于当前扰动绝对大小以及梯度绝对大小来确定像素点的重要性，当前该像素的扰动绝对值越大、梯度绝对值越大则越重要。C&W-l_0 会重复这个迭代过程，直到迭代中无法用 C&W-l_2 在集合 $\mathcal{A}$ 上攻击成功，此时则表明上一轮的集合 $\mathcal{A}$ 已经最小，其中的像素点必须被修改才能使得攻击成功。

实验结果显示 C&W-l_0 在攻击成功率以及速度效率上均优于 JSMA。但 C&W-l_0 在不可察觉性方面仍然存在缺点，由于其使用了最小化 l_2 范数的 C&W-l_2 作为每一轮迭代的基础攻击方法，但是 C&W-l_2 没有考虑单个像素点的最大扰动幅度即 l_∞ 范数，这使得对抗样本中某些像素点的扰动幅度过大，在图像中呈现突兀的色彩从而令人容易察觉。

2.3.3 PGD-$l_0 + l_\infty$

原始的 PGD[24]方法会指定任意像素点的最大扰动幅度但不限制扰动像素点的个数，即仅指定扰动的 l_∞ 范数，可以称为 PGD-l_∞。Croce 等人[40]在 PGD 的基础上，提出了一种将对抗扰动投影到 l_0 范数球内的方法，称为 PGD-$l_0 + l_\infty$，该方法可以同时指定最终扰动的 l_∞ 范数和 l_0 范数。

具体的，PGD-$l_0 + l_\infty$ 是一种迭代的方法，给定原始三通道 RGB 图像 $\boldsymbol{x} \in [0,1]^{n\times 3}$，每一轮迭代中首先使用 PGD 方法得到临时对抗样本 $\boldsymbol{y} \in [0,1]^{n\times 3}$，紧接着投影到 l_0 范数球内的问题可以表示为：

$$\begin{aligned} \min_{z\in\mathbb{R}^{d\times 3}} \quad & \sum_{i=1}^{n}\sum_{j=1}^{3}(y_{ij}-z_{ij})^2 \\ \text{s.t.} \quad & l_{ij} < z_{ij} < u_{ij}, \quad i = 1,2,...,n, \quad j = 1,2,3 \\ & \sum_{i=1}^{n}(\max_{j=1,2,3}\mathbb{I}(|z_{ij}-x_{ij}|>0)) \le k, \end{aligned} \tag{2-18}$$

其中 l_{ij} 和 u_{ij} 表示限定的最小像素值和最大像素值，用以控制扰动的 l_∞ 范数，其中 k 表示限定的扰动像素点个数，用以控制扰动的 l_0 范数。也就是说，上述问题是找到距离临时对抗样本 $\boldsymbol{y}$ 最近的样本 $\boldsymbol{z}$，使得样本 $\boldsymbol{z}$ 的扰动满足 l_∞ 范数和 l_0 范数的限制。

由于问题 (2-18) 有组合约束，难以直接求解，Croce 等人仍然使用贪心的方法来求解。具体来讲，首先忽略组合约束，则该问题有闭式解 $z^*_{ij} = \max\{l_{ij}, \min\{y_{ij}, u_{ij}\}\}$。紧接着 Croce 等人自定义了每个像素点的重要性：

$$\phi_i := \sum_{j=1}^{3}(y_{ij}-x_{ij})^2 - \sum_{j=1}^{3}(y_{ij}-z^*_{ij})^2, \tag{2-19}$$

上式的重要性可以理解为：当前所需要的扰动幅度越大、投影到限定范围内的距离越小，则像素点越重要。接下来 PGD-$l_0 + l_\infty$ 仅修改最重要的 k 个像素点，并保持其他像素点的值与原始图像中的像素值一致，完成一轮迭代。

PGD-$l_0 + l_\infty$ 可以同时在 l_0 和 l_∞ 范数的限制下构建对抗样本。在具体实现中，不同于其他稀疏对抗攻击，PGD-$l_0 + l_\infty$ 与 PGD 紧密结合，实现了批量化处理，因此可以快速构建大量稀疏对抗样本，但是攻击成功率相对较低。

2.3.4 GreedyFool

Dong 等人[41]提出了一个基于贪心思想的两阶段的稀疏对抗攻击方法，称为 GreedyFool。与其他稀疏对抗攻击方法类似，该方法也自定义了像素重要性指标。在第一个阶段，该方法每轮迭代中都贪心地选择当前最重要的若干像素点进行扰动，直到攻击成功则停止迭代。由于贪心的方法容易陷入局部最优，因此 GreedyFool 在第二个阶段再次进行贪心搜索，在已经修改的像素点集中找到冗余的、不必要的像素点，并将其恢复至原始像素值。

特别的，GreedyFool 从两个方面来考虑像素点的重要性。首先是攻击效果方面，GreedyFool 同样利用了损失值相对于输入的梯度绝对大小，该值越大则说明该像素点越重要。具体的，GreedyFool 参考了 C&W[31]的损失函数来计算梯度，用 $Z(\cdot)$ 表示模型 logits 层输出向量，y^* 表示目标类别标签，则梯度 $\boldsymbol{g}$ 计算为：

$$\begin{aligned} L(\boldsymbol{x}, y^*) &= \max(\max_{i \neq y^*}(Z(\boldsymbol{x}')_i) - Z(\boldsymbol{x}')_{y^*}, -\kappa), \\ \boldsymbol{g} &= \nabla_{\boldsymbol{x}} L(\boldsymbol{x}, y^*). \end{aligned} \tag{2-20}$$

其次为了实现对抗样本的不可察觉性，GreedyFool 引入了基于生成式对抗网络（Generator Adversarial Networks，GAN）得到的失真图（Distortion Map），其中像素的失真值代表了像素修改后的可见性，失真值越高，表明像素修改后越容易被观察。令失真图表示为 $\boldsymbol{\varrho} \in [0,1]^n$，为了平衡攻击效果与不可察觉性，该方法基于失真图 $\boldsymbol{\varrho}$ 计算了图像每个像素点的权重 $\boldsymbol{p}$，并基于权重 $\boldsymbol{p}$ 和梯度 $\boldsymbol{g}$ 计算像素点的重要性 $\boldsymbol{v}$：

$$p_i = \begin{cases} 0, & \varrho_i \geq t_1 \\ \frac{t_1 - \varrho_i}{t_1 - t_2}, & t_2 < \varrho_i < t_1 \\ 1, & \varrho_i < t_2 \end{cases}, \tag{2-21}$$

$$\boldsymbol{v} = \boldsymbol{p} \cdot \boldsymbol{g} \cdot (1 - \boldsymbol{m}), \tag{2-22}$$

其中 t_1 和 t_2 为预定义的阈值，$\boldsymbol{m} \in \{0,1\}^n$ 为 01-mask，表示对应像素点是否已经被选择。换句话说，GreedyFool 在每一轮迭代中，都会基于梯度以及失真值计算像素点的重要性，并选择 k 个还未被选择的像素点加入到允许修改像素集合中，然后该轮只修改允许修改集合中的像素。随着不断迭代，允许修改像素集合会从空集逐渐变大，直到攻击成功。由于这个过程是贪心的过程，构建出

的对抗样本可能并非最优，因此 GreedyFool 还会进行第二阶段来筛除部分冗余的扰动。简单来说，第二阶段该方法仍然使用迭代的贪心搜索，每一轮迭代时尝试抹除掉幅度最小的扰动，若抹除后仍然能够攻击成功则正式抹除，若抹除后无法攻击成功，则不再抹除并将该像素加入到不可选择的集合中。

与 PGD-$l_0 + l_\infty$ 一样，GreedyFool 也能够同时预先指定对抗扰动的 l_0 和 l_∞ 范数，并且引入的失真图能够一定程度上更加避免被人眼所察觉。但是 GreedyFool 仍然未能摆脱贪心的迭代策略，虽然通过第二阶段抹除了部分冗余的扰动，该方法仍然容易陷入局部最优。

2.3.5 SAPF

不同于之前提出的稀疏对抗攻击，Fan 等人[57]提出的基于扰动分解的稀疏对抗攻击（Sparse Adversarial Attack via Perturbation Factorization，SAPF）不再使用贪心的搜索策略。简单来讲，SAPF 不再人工定义像素点的重要性，而是将每个像素点的扰动分解为扰动幅度与 01-mask 两个变量的乘积。当某个像素点的 mask 对应为 1 时，则表明该像素点被修改；否则，该像素点保持原值。在此基础上，Fan 等人将该问题表述为混合整数规划问题 (Mixed Integer Programming，MIP)，并联合优化所有像素的扰动幅度和 01-mask 变量，并对 01-mask 进行基数约束（Cardinality Constrain），以显式控制扰动的稀疏程度。

形式上来讲，SAPF 将扰动向量 $\boldsymbol{\delta}$ 分解为：

$$\boldsymbol{\delta} = \boldsymbol{\epsilon} \odot \boldsymbol{G}, \tag{2-23}$$

其中 $\boldsymbol{\epsilon} \in \mathbb{R}^n$ 表示扰动幅度，$\boldsymbol{G} \in \{0,1\}^n$ 为 01-mask 代表了扰动的位置，$\odot$ 表示元素对应相乘。因此稀疏对抗攻击可以表示为：

$$\begin{aligned} \min_{\boldsymbol{\epsilon},\boldsymbol{G}} \quad & \|\boldsymbol{\epsilon} \odot \boldsymbol{G}\|_2^2 + \lambda_1 L(f(\boldsymbol{x} + \boldsymbol{\epsilon} \odot \boldsymbol{G}), y^*) \\ \text{s.t.} \quad & \mathbf{1}^T \boldsymbol{G} = k, \boldsymbol{G} \in \{0,1\}^n, \end{aligned} \tag{2-24}$$

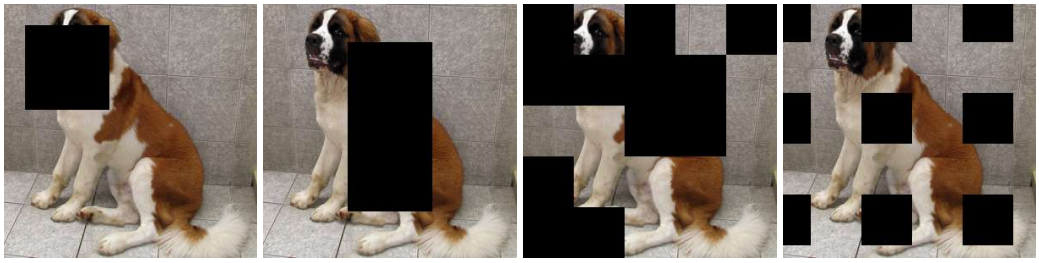
其中 $\lambda_1 > 0$ 是平衡扰动大小和攻击效果的参数，约束条件 $\mathbf{1}^T \boldsymbol{G} = k$ 是为了强制扰动个数为 k。由于 $\boldsymbol{\epsilon}$ 是连续性变量，而 $\boldsymbol{G}$ 是整数变量，因此问题 (2-24) 是一个混合整数规划问题。因此 Fan 等人使用性能优良的整数规划方法 l_p-Box ADMM[58]来迭代地求解该问题。

总之，SAPF 没有再人为定义像素点的重要性，而是将扰动分解为扰动幅

度和 01-mask 选择因子，并使用整数规划方法求解。这使得 SAPF 构建的对抗样本相较于其他方法有更强的稀疏性。但是由于 SAPF 包含较多比较敏感的超参数，在实际实现中需要不断搜索合适的超参数才能得到较优的结果，因此速度特别慢且攻击不稳定。

2.4 图像数据增强

本文除了提出一种稀疏对抗攻击方法以外，还将其与基于信息删除的数据增强方法相结合，提出了一种基于稀疏对抗攻击的图像数据增强方法，因此本节对这类方法进行介绍。基于信息删除的图像数据增强方法应用广泛，具有代表性有 Cutout[49]、RandomErasing[50]、HaS[59]和 GridMask[51]等，图 2-1展示了这几种方法的示例。下面将对这几个方法进行详细介绍。



(a) Cutout (b) RandomEarsing (c) HaS (d) GridMask

图 2-1: 基于信息删除的数据增强方法示例

2.4.1 Cutout

Cutout 是一种实现起来非常简单的图像数据增强方法，该方法在训练过程中随机删除输入图像内的某个正方形区域，模拟图像中物体被遮挡的情况，鼓励网络更好地利用图像的全部上下文特征，可以用来提高卷积神经网络的鲁棒性和整体性能。Cutout 可以看作是正则化方法 Dropout 在输入层的扩展，但是具有两点重要的不同。首先 Cutout 是一种数据增强方法，它仅作用于图像数据上，也就是网络模型的输入层，而不像 Dropout 一样作用在隐藏层；其次，Cutout 删除的是输入图像上的连续像素区域，而不是单个独立的点。这是因为对于 CNN 来讲，卷积运算对去掉一个非常小的区域不敏感[45],[49]，因此点状的删除将无法提升模型鲁棒性。

Cutout 的具体实现流程为：在训练的每一轮中，在每个输入图像的随机位置上应用一个固定大小的 0 值 mask，即删除（置 0）该区域的像素信息。具体来讲，Cutout 首先在图像中随机选择一个像素坐标作为中心点，然后将 0 值 mask 应用到该位置上。这样的操作在某些情况下会使得 mask 的一部分落在图像边界外，使得图像内被删除的区域的面积是变化的，增加了多样性，使得最终的性能更好。总的来讲，每一张图像在每一轮训练时都会重新随机一个删除区域，因此一张图像将会扩展为出多张不同的图像，最终提升了模型的性能。

2.4.2 RandomErasing

RandomErasing（RE）与 Cutout 几乎同时被提出，并且思想比较相近。RE 在训练过程中按照一定的概率随机选择图像中的一个矩形区域，并在区域内填充随机的像素值。也就是说与 Cutout 相比，RE 引入了更多的随机性。

具体来讲，在每一轮的训练中，每张图片都有概率 p 会进行信息删除，有 $1-p$ 的概率保持不变，这样的目的是生成不同遮挡程度的训练图像，提升多样性。对于每个进行信息删除的图像，会随机一个矩形区域 I_e，该矩形区域的位置、面积以及长宽比都是随机确定的。首先假定训练图像的尺寸为 $w \times h$，总面积为 $s = w \times h$，则 RE 首先随机初始化矩形区域 I_e 的面积为 s_e，并保证 $\frac{s_e}{s}$ 处于预设上下限参数 s_l 和 s_h 的范围内；同样的，长宽比 r_e 在预设的上下限参数 r_1 和 r_2 之间随机初始化；因此矩形区域 I_e 的宽和高分别可以表示为 $h_e = \sqrt{s_e \times r_e}$ 和 $w_e = \sqrt{\frac{s_e}{r_e}}$；在这之后，RE 需要指定图像中一个随机位置 $\mathcal{P} = (x_e, y_e)$ 作为矩形区域 I_e 的左上角坐标，此时如果矩形区域 I_e 全部处于图像边界内部则完成矩形区域的选择，否则将重复上述步骤直到完成 I_e 的选择。对于已经选择好的矩形区域，每个像素都会进一步被填充为 $[0, 255]$ 的随机值。总的来说，RE 中删除区域的位置、面积、长宽比以及填充值都是随机的，这大幅度增加了图像删除区域的多样性，迫使网络利用局部未删除的信息进行识别，提高了模型泛化能力。

2.4.3 HaS

HaS 全称为 Hide-and-Seek，同样是基于信息删除的数据增强方法。由于与 RE 与 Cutout 的思想基本一致，因此这里仅介绍其不同的地方。HaS 最大的特点就是不像前两种方法只删除图像某一个连续区域，而是可以删除多个区域。

具体来讲，假定训练图像 I 的尺寸为 $w \times h$，则 HaS 将图像分为若干 $s \times s$ 的区域，对于每个区域以概率 p 进行删除，也即是说每个区域删除与否是独立的。在删除区域并填充值时，HaS 使用整个数据集的平均值而不是直接使用 0 值，这样能够一定程度上保持训练数据和测试数据的数值分布相同。

由于 HaS 同样有一定的概率产生与 RE 一样的随机矩形删除区域，因此 HaS 也可以看作更广义的 RE。而 HaS 的优势在于：多个不连续的删除区域提供了更多的变化，例如：图像中犬类的头部和腿部的区域被删除，但身体部位可见，这在 RE 这种单一删除区域下是无法办到的。

2.4.4 GridMask

通过总结分析 Cutout、RE 和 HaS 等数据增强方法的优缺点，Chen 等人提出观点认为基于信息删除的数据增强方法的核心要求是：需要避免图像中连续区域的过度删除或者过度保留，需要在删除和保留之间找到合理的平衡。简单来说，过度删除一个或者多个区域可能会导致图像中的主体对象及其上下文被完全删除，余下的信息不足以被模型正确分类，可以被认为是噪声；而过度保留可能会使一些图像中的对象完全不受影响，而这些琐碎的图像可能会导致模型的泛化性降低。比如 Cutout 和 RE 都仅删除图像中的单个连续区域，因此很难做到平衡，有一定可能完全删除了对象区域或者仅删除了背景区域。而对于删除多个区域的 RE 同样存在此问题，图像中的物体同样有一定几率被完全删除从而称为噪声图像。

因此，Chen 等人提出了名为 GridMask 的方法来解决上述两个问题。简单来说，GridMask 既不删除像 Cutout 那样删除单个连续的大区域，也不像 HaS 那样随机删除多个正方形区域。如图 2-1所示，GridMask 删除的区域是一组在空间中均匀分布的正方形，每个正方形之间的间隔是相同的。在这个结构中，通过控制每个小区域的宽度以及区域之间的间隔，GridMask 在统计上相较于上述几个方法更有机会达到两种条件之间的良好平衡，也就是既不会完全删除图像中的主体对象也不会仅删除掉了背景区域。CIFAR-10 和 ImageNet 数据集上的实验表明，GridMask 确实比上述其他方法的性能更好。

2.5 本章小结

本章主要介绍了本文所涉及的背景知识与相关代表工作。首先介绍了对抗样本的基本定义、术语以及性能度量方式。然后介绍了最具有代表性的稠密对抗攻击方法如 FGSM、PGD 以及 C&W 等。与之对应的，我们也介绍了具有代表性的稀疏对抗攻击方法，这些方法在保证攻击成功率的情况下减少了不必要的扰动。但是现有的稀疏对抗攻击仍然存在较大的问题，现有的大多数方法都使用贪心策略逐次修改重要的像素点，其中自定义的重要性度量指标可能不适用于所有情况，并且贪心的策略很容易陷入局部最优。其次，部分稀疏对抗攻击方法仅仅是尽可能减少扰动像素点的个数，但是却忽略了扰动幅度从而丧失了不可察觉性。因此如何跳出这样一个固定的模式，并保持稀疏对抗攻击有较高的攻击成功率和较强的不可察觉性就是本文接下来要研究的问题之一。除此以外，本章也介绍了具有代表性的基于信息删除的数据增强方法如 Cutout、RE 以及 GridMask 等。这些方法通过某些策略在训练过程中随机删除输入图像上的部分区域，显著提高了模型的泛化性。但是目前这类方法都是按照一定的策略在全图中随机产生删除区域，对于一些复杂的图像仍然无法很好地处理。因此如何减少这种因为全图随机性带来的低质量样本，通过某些前置方法找到每张图像最需要被删除区域也是本文接下来要研究的问题之一。

第三章　基于像素自动删减的稀疏对抗攻击框架

本章首先对问题进行分析，整理现有方法的特点和存在的问题，进而根据神经网络剪枝与稀疏对抗攻击的内在相似性，设计了基于像素自动删减的稀疏对抗攻击框架（AutoAdversary），旨在跳出固定的贪心策略，并最终通过实验验证了本方法的有效性。

3.1　问题分析

3.1.1　问题形式化

如2.1.2节所述，对抗攻击可以被分为无目标攻击和有目标攻击。有目标攻击要求模型产生指定的错误，因此通常来讲比无目标攻击更加困难，所以本文主要关注有目标攻击。令 $f:[0,1]^{w\times h\times c}\to\mathbb{R}^k$ 表示分类模型，其中 w、h 和 c 分别表示输入图像的宽、高和通道数。对于归一化后的原始图像 $\boldsymbol{x}\in[0,1]^{w\times h\times c}$，$f(\boldsymbol{x})\in\mathbb{R}^k$ 为模型的 logits 层输出，k 个维度分别对应到 k 个类别标签。此时目标对抗攻击通常可以形式化为：

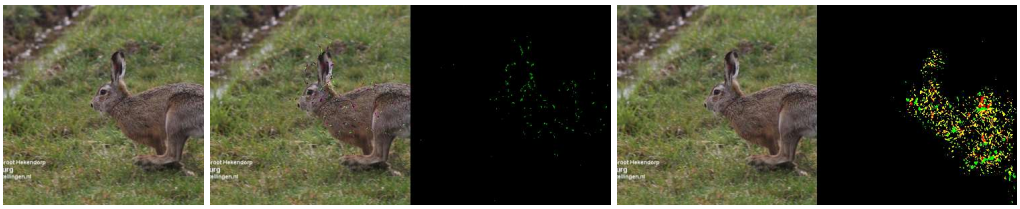
$$\begin{aligned}\min_{\boldsymbol{\delta}}\quad & \mathcal{D}(\boldsymbol{\delta})\\ \text{s.t.}\quad & \arg\max_{i=1,\dots,k} f_i(\boldsymbol{x}+\boldsymbol{\delta})=y^*,\end{aligned}\tag{3-1}$$

其中 $\mathcal{D}(\cdot)$ 是距离函数，$\boldsymbol{\delta}\in\mathbb{R}^{w\times h\times c}$ 表示对抗扰动，y^* 表示目标分类标签，$f_i(\cdot)$ 表示的是模型 logits 层输出向量第 i 维的值，所以上述问题的约束条件要求模型的预测标签与目标标签相同。至于距离函数 $\mathcal{D}(\cdot)$，大多数具有代表性的对抗攻击方法都使用 l_p 范数，最常用的是利用 l_2 范数来限制对抗样本和原始样本的欧氏距离，或者利用 l_∞ 范数来限制任意像素的最大扰动幅。这类方法均对全图的所有像素点进行扰动，因此也被称为稠密对抗攻击。

但是并非整张图像的所有像素点均有必要扰动，根据许多神经网络可视化研究指出，图像中不同像素点对模型决策的影响不尽相同，并且影响最大的像素往往集中在若干区域内。因此，现有的经典对抗攻击方法仍然具有改进的空间，即可以通过进一步减少扰动像素点的个数去除掉冗余的扰动，从而降低整体的扰动量。也就是说本文的目标函数为最小化扰动 $\boldsymbol{\delta}$ 的 l_0 范数。但是需要注意的是，仅仅达成这一目标并不能得到足够好的对抗样本，如图3-1所示，3-1(b)和3-1(c)中的扰动像素点个数的占比分别为 0.15% 和 4.36%，但是人眼能够轻松察觉到3-1(b)和原图3-1(a)的不同，即便是其扰动的像素点更少。换句话说，当允许的最大扰动幅度 ϵ 也即是 l_∞ 范数较大时，修改极少数的像素点就能够攻击成功，但是这些扰动很容易被人眼察觉。因此本文提出，在限制扰动的 l_∞ 范数处于预定范围的情况下，可以进一步优化扰动的 l_0 范数即扰动像素点的个数，该问题可以形式如下：

$$\begin{aligned} \min_{\boldsymbol{\delta}} \quad & \|\boldsymbol{\delta}\|_0 \\ \text{s.t.} \quad & \arg\max_{i=1,\ldots,k} f_i(\boldsymbol{x}+\boldsymbol{\delta}) = y^* \\ & \boldsymbol{\delta} \in [-\epsilon, \epsilon]^{w\times h\times c}, \end{aligned} \tag{3-2}$$

其中 ϵ 表示允许的任意像素最大扰动幅度，即扰动的 l_∞ 范数。问题 (3-2) 可以理解为，稠密对抗攻击存在冗余的扰动，在保证扰动幅度处于一定范围内的情况下，仍可以进一步最小化扰动像素点的个数，完成稀疏对抗攻击。



(a) 原始图像 (b) $\epsilon = 128/255$ (c) $\epsilon = 4/255$

图 3-1: 稀疏对抗攻击在不同扰动幅度限制下的示意

3.1.2 现有稀疏对抗攻击的缺陷

稀疏对抗攻击需要最小化扰动的 l_0 范数，这使得该问题变为 NP 难问题[41]。大多数现有的方法都可以都描述为一个二阶段的流程：首先，这些方法人为定义一个像素点重要性的度量指标，其中重要性往往都是基于梯度以及扰

动幅度等信息确定的；然后根据每个像素点的重要性，这些方法使用贪心的策略选出最重要的像素点进行攻击，并按照这个流程一直循环迭代直到攻击成功。如2.3节所述：JSMA 基于对抗性显著图使用贪心策略在每次迭代中选择某几个像素点进行扰动；C&W-l_0 在每次迭代中先在 l_2 范数的限制下进行攻击，紧接着基于扰动幅度和梯度来固定最不重要的部分像素点；PGD-l_0+l_∞ 在每轮迭代中将 PGD 产生的扰动投影到 l_0 范数球内，其中投影的方法是根据扰动大小以及自定义的投影损失来固定那些没有必要扰动的点；基于梯度信息和失真图，GreedyFool 在每次迭代中选取一些像素点添加到可修改像素集中，然后使用贪心的方法去掉尽可能多的不重要像素点以获得更好的稀疏性。

在上述方法中，人为定义的重要性度量指标起着至关重要的作用，然而自定义的指标可能并不适合所有的情况。此外，像素点的选取和攻击是该流程中两个独立的步骤，即这些方法均使用贪心策略先筛选一部分像素点再进行修改，然而贪心策略很容易陷入局部最优。而且，部分方法仅仅是尽可能减少扰动像素点的个数，但是却忽略了扰动幅度等其他限制从而丧失了不可察觉性。

3.1.3 稀疏对抗攻击与神经网络剪枝的联系

通过前文的问题分析可知，稀疏对抗攻击需要跳出现有的固定模式，即避免人为直接定义像素点的重要性度量指标和使用贪心的策略。因此一个有趣的问题出现了：能否将像素点的选取和攻击结合起来，利用攻击来直接引导像素点的选择？换句话说，我们是否可以使用自动的方法来选择需要被扰动的像素点，而不依赖于人为直接定义的规则？研究人员在神经网络剪枝任务中回答了一个类似的问题。本文从神经网络剪枝任务的发展历程中得到了灵感，最终解决了稀疏对抗攻击中存在的问题。因此下面首先对神经网络剪枝进行简要的介绍。

虽然深度神经网络在众多领域都取得了令人惊异的效果，但是其猛增的参数量使得计算效率越来越低、存储开销越来越大。在某些实时性要求较高的任务中或者资源受限的边缘设备上，部署这种笨重的神经网络模型是不可能的。神经网络剪枝通过删除掉不重要的神经元来减小模型，从而降低计算开销。经过多年的发展，神经网络剪枝可分为连接级（connection level）、卷积核级（filter level）和层级（layer level）三类方法。其中连接级方法是最先出现的剪枝方法[60]，该类方法基于权重大小来判断神经元的重要性，并删除不重要的神经元来减小模型。但是连接级这类非结构性剪枝（non-structured pruning）会

产生不规则的卷积操作，需要专用的硬件才能高效地完成推理过程，因此在实际中很难降低计算开销。因此神经网络剪枝领域内逐渐都开始关注结构性剪枝（structured pruning）。结构性剪枝又包括卷积核级剪枝和层级剪枝，这两类方法分别将整个卷积核或者整个网络层看做一个整体，剪枝之后的网络结构仍然是规则的，不会拖慢推理速度。

经典的剪枝流程为：首先人为定义一个足够好的重要性度量指标，并基于此删除最不重要的部分卷积核或层，然后再在新的网络上微调以恢复模型的精度。比如，Luo 等人[61]根据下一层的统计量来判断当前层卷积核的重要性，He 等人[62]通过基于 LASSO 回归的方法来选择不重要的通道，Liu 等人[63]通过引入通道比例因子（channel scaling factors）来表示每一层的重要性。上述的方法都是力求找到更好的重要性度量指标，然而自定义的重要性度量指标并不一定适用于所有情况。

为了解决经典剪枝方法的问题，自动剪枝方法被提出。这类方法不再人为定义卷积核的重要性评估方式，而是在模型微调的过程中自动地将需要的卷积核保留，将不需要的卷积核删掉。其中具有代表性的一个方法是 Luo 等人[1]提出的端到端（end-to-end）可训练的剪枝方法 AutoPruner。如图 3-2所示，该方法可以看作是添加了一个新的网络分支，它以前一层激活值（activation）为输入，生成一个二进制编码，将二进制编码与该层的激活值相乘后再输入到后一层中。因此二进制编码中的 0 值意味着对应的卷积核的激活值总是 0，因此可以删除掉该卷积核。最终通过联合训练原网络中的权重以及分支网络中的权重，模型在微调的过程中自动完成了卷积核的选择。总之，AutoPruner 使用微调来指导卷积核的选择，消除了人为制定重要性评价标准的需要，也取得了更好的实验结果。

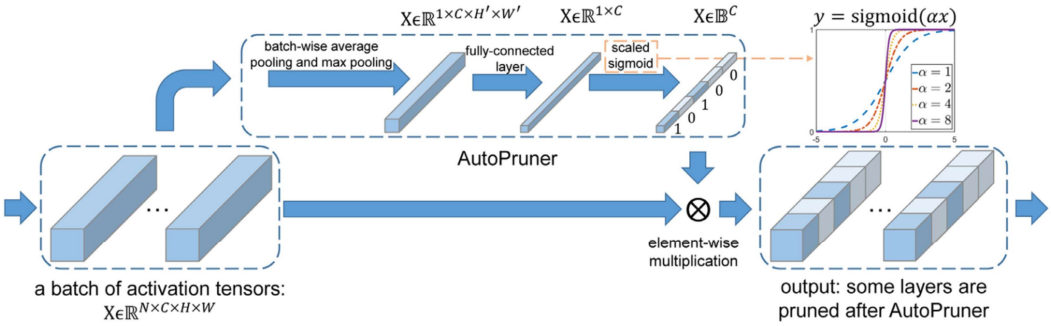


图 3-2: 神经网络自动剪枝 AutoPruner 的示意图[1]

稀疏对抗攻击与神经网络剪枝有着很强的内在相似性。首先，对抗攻击的

计算过程几乎等价于模型训练过程，模型的一般训练过程是迭代更新网络中的权值，使分类损失尽可能小。相反，对抗攻击过程中网络的权值是固定的，转而迭代更新输入图像，使对抗性损失尽可能小。因此可以从模型训练的角度来看待对抗攻击，也就是说将网络中的权值看作固定的输入，而将输入图像看作待更新的权值。因此稀疏对抗攻击问题可以看作是一个神经网络剪枝问题，即需要删除部分的不重要的权值。从另外一个角度来讲，神经网络剪枝可以看作一个 l_0 范数的优化问题，要求保留的卷积核越少越好、模型的精度越高越好；稀疏对抗攻击也是一个 l_0 范数的优化问题，要求扰动的像素点越少越好、攻击成功率越高越好。综上所述，根据神经网络剪枝的思想，也可以对扰动进行删减，从而减小扰动像素点的个数。

如前文所述，AutoPruner 是一种神经网络自动剪枝方法。它在通常的模型训练过程中添加一个分支，该分支输出一个二进制编码来自动选择卷积核，并通过联合训练原始网络中的权重以及分支网络中的权重，最终完成了卷积核的剪枝。参考这个想法，本文在通常的对抗攻击过程中增加了一个分支网络，该分支也输出一个 01-mask 来自动选择图像像素点，并通过联合优化扰动变量以及分支网络中的权重，使得大量像素点在 mask 中的对应值为 0，最终完成了稀疏对抗攻击。

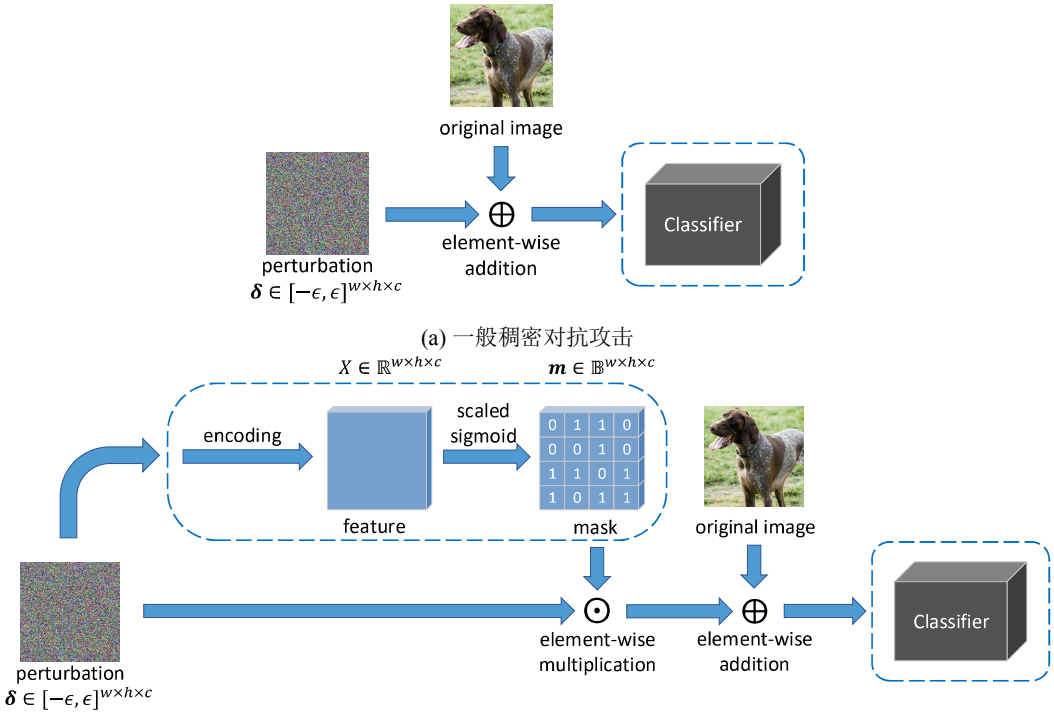
3.2 基于像素自动删减的稀疏对抗攻击

前文对问题进行了分析，并整理了现有稀疏对抗攻击方法的特点和存在的问题，并总结了神经网络剪枝与稀疏对抗攻击的内在相似性。本节将基于此提出一种端到端的、基于像素自动删减的稀疏对抗攻击方法，称为 AutoAdversary。下面对本方法进行详细的介绍。

3.2.1 整体结构

图3-3展示了一般稠密对抗攻击的结构和 AutoAdversary 的整体结构。AutoAdversary 的结构可以看作是在通常的稠密对抗攻击的过程中添加了一个包含编码和二值化两种操作的分支。具体的，该分支以待优化的对抗扰动变量作为输入，首先通过一个可训练的神经网络对其进行编码并产生一个相同尺寸的张量（tensor），然后使用带缩放因子的 sigmoid 函数来生成一个近似二值的 mask，将分支网络输出的 mask 与对抗扰动变量进行对应元素相乘后，就产

生了删减后的扰动。此时将删减后的扰动添加到原始图像上，并将其作为对抗样本输入到分类模型中。根据攻击效果以及扰动的稀疏性，同时对扰动变量进行优化以及对分支中的编码器网络进行训练。在训练过程中，通过逐步增大 sigmoid 函数中的缩放因子，强制其输出绝对二值的 mask，最终使得一些像素点在 mask 中的对应值为 0，也就是说该点的扰动可以被删除，此时进行像素删减就产生了稀疏的对抗扰动。因此，AutoAdversary 可以在攻击过程中端到端地自动选择最需要被扰动的一批像素点，无需人为定义像素点重要性评价指标，并且不包含贪心的选取策略。



(a) 一般稠密对抗攻击

(b) AutoAdversary

图 3-3: AutoAdversary 整体结构图

3.2.2 关键模块

本小节对 AutoAdversary 的关键模块进行详细的介绍。

3.2.2.1 编码器

3.2.1节中提到 AutoAdversary 添加了一个包含编码和二值化这两个过程的分支，其编码过程由一个可训练的神经网络来负责。令 $\mathcal{H}:\mathbb{R}^{w\times h\times c}\to\mathbb{R}^{w\times h\times c}$ 表示一个可训练的神经网络，用 $\boldsymbol{\delta}\in\mathbb{R}^{w\times h\times c}$ 表示待优化的对抗扰动变量，则 $\mathcal{H}(\boldsymbol{\delta})\in\mathbb{R}^{w\times h\times c}$ 表示用编码器 $\mathcal{H}$ 编码变量 $\boldsymbol{\delta}$ 后的张量。使用编码器的目的是提取扰动变量 $\boldsymbol{\delta}$ 的特征，并便于后续的二值化操作。若不通过编码就直接将 $\boldsymbol{\delta}$ 输入到二值化模块中，则输出的 mask 将受到 $\boldsymbol{\delta}$ 的数值的直接影响。比如当二值化模块为 sigmoid 函数时，$\boldsymbol{\delta}$ 的负数元素所对应的 mask 值都小于 0.5，将会在最终的处理中被置 0，即永远无法产生负向的扰动，因此会缩减对抗空间，不利于构建对抗样本。后续的3.3.4节的消融实验也证明了编码器的必要性。

当图像尺寸较小时，本文直接使用全连接层（fully-connected layer）作为编码器，其权重变量可以表示为 $\mathcal{W}\in\mathbb{R}^{(w\times h\times c)\times(w\times h\times c)}$。具体来讲，在小尺寸图像如 CIFAR-10 数据集上，以全连接层作为编码器所需要的参数量为 9.44×10^6，完全在可接受的范围内。但全连接层的参数量与图像尺寸相关，当图像的尺寸较大时，使用全连接层作为编码器需要训练巨量的参数。如图3–3(b)所示，考虑到编码器的输入和输出的尺寸需要维持一致，本文使用经典的图像分割网络 U-net[64]作为编码器，其参数量为 3.10×10^7，这种小级别的参数量使得 AutoAdversary 仍然能够在 ImageNet 这样的大尺寸图像数据集上保持较高的效率。

需要注意的是，在本文以及其他稀疏对抗攻击方法[57]中都将 RGB 图像一个位置上 3 个通道当作独立的 3 个像素来看待，也就是说可以独立地选择其中一个或者多个进行扰动。然而在其他一些方法中，RGB 图像的 3 通道被当作一个整体即一个像素来看待，此时这 3 个通道要么都被扰动要么都不被扰动。这里想要说明的是，通过改变编码器 $\mathcal{H}$ 的输出尺寸，AutoAdversary 同样可以做到将 3 通道视作一个整体。具体的，我们可以把 $\mathcal{H}$ 的输出尺寸设置为 $w\times h\times 1$，此时 mask 的尺寸也为 $w\times h\times 1$，那么图像中相同位置的 3 通道将共用一个 mask 元素值，最终使得 3 个通道将同时被扰动或者同时保持不变。

3.2.2.2 二值化

编码后的张量 $\mathcal{H}(\boldsymbol{\delta})$ 的每个元素都处于实数域内，但是作为二元选择的 mask 的元素应当是二值的。为了保持连续和可导的性质以便通过梯度下降进行优化，AutoAdversary 使用带有缩放因子的 sigmoid 函数作为二值化函数，产生

近似二值的 mask：

$$\boldsymbol{m} = \text{sigmoid}(\alpha \cdot \mathcal{H}(\boldsymbol{\delta})), \tag{3-3}$$

其中 $\boldsymbol{m} \in [0,1]^{w \times h \times c}$ 表示近似二值的 mask，α 表示控制二值化程度的缩放因子。

图3-4展示了 α 的变化对二值化程度的影响。若将 α 作为固定的常量，则当 α 太小时，式 (3-3) 会导致最终 mask 中的大量元素值都维持在 0.5 附近，不足以产生二值的 mask。当 α 太大时，可以产生二值的 mask，但是 sigmoid 函数的特性是两端的导数几乎为 0，过大的 α 使得绝大部分梯度无法回传，导致编码器网络 $\mathcal{H}$ 得不到有效的训练。所以如果一开始就使用较大的 α 将导致像素点的选择仅跟随机初始化相关，也即是方法退化为随机选择像素点。

因此参照神经网络剪枝方法 AutoPruner[1]中的处理方式，缩放因子 α 并非一个固定的常量，而是会在优化扰动变量 $\boldsymbol{\delta}$ 以及训练编码器网络 $\mathcal{H}$ 的过程中逐渐增大。也就是说，在训练过程中，我们将 α 从 α_{start} 逐渐增大到 α_{end}，以保证最终 mask 的元素收敛到二值，并且避免方法退化为随机选择像素点。

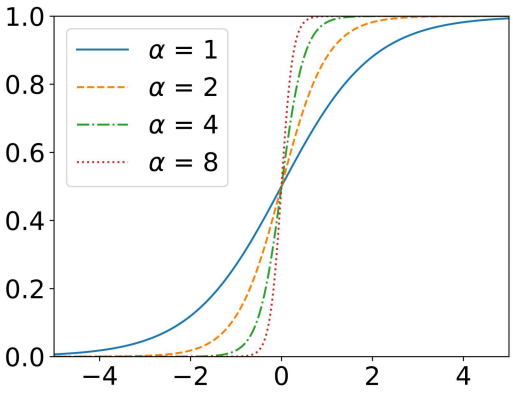


图 3-4: 缩放因子对 sigmoid 函数二值化程度的影响

3.2.2.3 损失函数

前文介绍了 AutoAdversary 的整体框架特别是所添加分支的详细结构。本文使用梯度下降来同时优化扰动变量 $\boldsymbol{\delta}$ 以及训练分支中的编码器网络，因此需要定义一个损失函数来更新这中间存在的权重参数。根据前文所述，当前的目标是令删减后的扰动尽可能稀疏也就是 mask 尽可能稀疏，以及添加扰动后的对抗样本欺骗目标模型的能力尽可能强。最常用的稀疏正则化方法是最小化变

量的 l_1 范数，因此 AutoAdversary 的损失函数定义如下：

$$\mathcal{L} = \mathcal{L}_{\text{adv}}(f(\boldsymbol{x} + \boldsymbol{\delta} \odot \boldsymbol{m}), y^*) + \lambda \frac{\|\boldsymbol{m}\|_1}{N}, \tag{3-4}$$

其中 $\odot$ 代表按元素相乘，$\mathcal{L}_{\text{adv}}(\cdot,\cdot)$ 是对抗损失，通常是交叉熵损失（cross-entropy loss）。$N = w \times h \times c$ 表示像素点的个数，$\boldsymbol{m}$ 代表近似二值化的 mask，因此 $\|\boldsymbol{m}\|_1$ 近似等于扰动像素点的个数，所以式 (3-4) 的第二项能够一定程度上代表 mask 的稀疏程度。特别的，λ 是一个自动动态调整的参数，用于平衡攻击效果和稀疏程度，其定义为：

$$\lambda = C + \frac{\gamma}{N} \sum_{i=1}^{N} \mathbb{I}(m_i > 0.5), \tag{3-5}$$

其中 $C > 0$ 是超参数，理解为 λ 的最小值，$\mathbb{I}(\cdot)$ 为指示函数，m_i 是 mask 的第 i 个元素，$\gamma > 0$ 是超参数。由于最终迭代结束后需要以 0.5 为界将 mask 的元素进行绝对二值化，所以这里 $\sum_{i=1}^{N} \mathbb{I}(m_i > 0.5)$ 就可以代表当前 mask 的稀疏程度。可以容易看出 λ 与当前 mask 的稀疏程度有很强的相关性。如果当前 mask 的不够稀疏即需要扰动的像素点过多，则 λ 会相对比较大，此时 AutoAdversary 可以更加聚焦于使得 mask 更稀疏；如果当前 mask 足够稀疏即扰动的像素点比较少，则 λ 会相对比较小，此时 AutoAdversary 可以更加聚焦于使得扰动更具有攻击性。更加细节的地方在于，C 和 γ 是预先定义的超参数，C 和 γ 越大，则在同样的 mask 下将会得到越大的 λ，也即是针对稀疏不足的惩罚越大；反之若 C 和 γ 越小，则针对稀疏不足的惩罚越小。

这里需要强调的是，将 $\|\boldsymbol{m}\|_1$ 作为正则项与直接将 $\|\boldsymbol{\delta}\|_1$ 作为正则项是全然不同的，换言之直接将扰动变量 $\boldsymbol{\delta}$ 的 l_1 范数 $\|\boldsymbol{\delta}\|_1$ 作为正则项无法得到稀疏的对抗扰动。这是因为虽然常用的稀疏正则化方法是最小化变量的 l_1 范数，但是由于神经网络的训练是非凸优化，并且由于计算机的精度限制，变量并不能准确优化到 0 值，最终会导致大量的像素点拥有较小的扰动幅度但是又无法忽略。由于 AutoAdversary 所添加的分支结构中包含二值化模块，得到的 $\boldsymbol{m}$ 是近似二值化的，并且如3.2.2.2节所述，随着缩放因子 α 逐步增大，$\boldsymbol{m}$ 二值化的程度越来越高，这就使得 $\boldsymbol{m}$ 中的大量元素被优化到 0 值或者非常接近 0，进而产生了稀疏的对抗扰动。后续的消融实验也显示了直接将 $\|\boldsymbol{\delta}\|_1$ 添加到正则项中无法产生稀疏的扰动，证明了 AutoAdversary 所添加的分支结构是有效的。

3.2.3 算法流程

前文介绍了 AutoAdversary 的整体结构和关键的几个模块，本节将给出其详细的算法流程，主要包括扰动变量的更新和编码器网络的训练，最终得到稀疏对抗样本。AutoAdversary 能够端到端地自动确定需要扰动的像素点，所以无需使用贪心策略修改像素点，而是可以直接通过梯度下降来优化得到最终的对抗扰动，因此可以以更快的速度完成对抗攻击。

更加重要的是，AutoAdversary 可以看作是在通常的稠密对抗攻击的过程中添加了一个包含编码和二值化的分支网络，并且与具体的稠密对抗攻击无关。也就是说，AutoAdversary 可以作为一种通用的像素删减框架，能够在大多数现有稠密对抗攻击方法的基础上，通过像素删减以减少扰动像素点的个数，在满足原本约束条件的情况下进一步减小扰动量，完成稀疏对抗攻击。按照3.1.1节中问题 (3-2) 的约束条件，稀疏对抗攻击时仍然也需要考虑 l_∞ 范数，因此 AutoAdversary 特别贴合那些原本就是在预设 l_∞ 范数限制下进行稠密对抗攻击的方法，可以直接在其基础上进行像素删减以达成稀疏对抗攻击。下面将着重以几个经典的稠密对抗攻击为基础，介绍不同版本的 AutoAdversary。

3.2.3.1 基于 I-FGSM 进行删减

本文在2.2.3节已经详细介绍了 I-FGSM，该方法预先定义任意像素的最大扰动幅度即扰动 $\boldsymbol{\delta}$ 的 l_∞ 范数，然后在此限制条件下通过符号梯度来优化得到对抗样本。AutoAdversary 可以轻松将 I-FGSM 作为基础的稠密对抗攻击方法，进而产生稀疏对抗样本，并且满足其原本的 l_∞ 范数约束条件。形式化的，I-FGSM 将交叉熵损失作为对抗损失，因此在这里式 (3-4) 可以进一步写为：

$$\mathcal{L} = \mathcal{L}_{\mathrm{CE}}(f(\boldsymbol{x} + \boldsymbol{\delta} \odot \boldsymbol{m}), y^*) + \lambda \frac{\|\boldsymbol{m}\|_1}{N}, \tag{3-6}$$

其中 $\mathcal{L}_{\mathrm{CE}}(\cdot,\cdot)$ 表示交叉熵损失函数。在此基础上，同样按照 I-FGSM 原本的方式来迭代更新扰动变量 $\boldsymbol{\delta}$，只不过需要改为目标攻击：

$$\boldsymbol{\delta}_0 = \mathbf{0}, \tag{3-7}$$

$$\boldsymbol{\delta}_{t+1} = \mathrm{Clip}_\epsilon\{\boldsymbol{\delta}_t - \beta \cdot \mathrm{sign}(\nabla_{\boldsymbol{\delta}}\mathcal{L})\}, \tag{3-8}$$

其中 $\mathcal{L}$ 为式 (3-6) 所定义的损失，β 是每次迭代中移动的步长，$\text{Clip}_\epsilon\{\cdot\}$ 用于将第 t 次迭代中的扰动变量 $\boldsymbol{\delta}_t$ 投影到半径为 ϵ 的 l_∞ 范数球内。具体的，对于扰动变量 $\boldsymbol{\delta}$ 的每个元素 δ_i，$\text{Clip}_\epsilon\{\delta_i\}$ 的具体实现如下：

$$\text{Clip}_\epsilon\{\delta_i\} = \max(-\epsilon, \min(\epsilon, \delta_i)). \tag{3-9}$$

除了优化扰动变量 $\boldsymbol{\delta}$ 以外，AutoAdversary 的关键还在于同时训练分支结构中的编码器网络，以及按照3.2.2.2节所述的逐步增大二值化模块中的缩放因子 α。具体来讲，在训练编码器方面，本文直接使用带动量的随机梯度下降法（Stochastic Gradient Descent，SGD）来更新编码器。在逐步调整 α 方面，在不同的数据集上需要确定初始值 α_{start} 和结束值 α_{end}。首先需要找到能产生绝对二值 mask 的 α_{end}，仅需在两到三张图像上尝试几个不同的值，就能确定一个足够好的 α_{end}。不同的 α_{start} 对于结果的影响不大，因此本文直接令 $\alpha_{\text{start}} = 0.1$。形式化的，更新 α 的方法如下：

$$\begin{aligned} \alpha_0 &= \alpha_{\text{start}}, \\ \alpha_{\text{step}} &= \frac{\alpha_{\text{end}} - \alpha_{\text{start}}}{T}, \\ \alpha_{t+1} &= \alpha_t + \alpha_{\text{step}}, \end{aligned} \tag{3-10}$$

其中 T 表示最大迭代次数，可以看出正常情况下 α 在每一轮迭代中以固定的步长 α_{step} 增大。更为细节的是，需要在迭代后期检查 mask 的二值化程度，具体方法为：

$$\text{Check}(\boldsymbol{m}) = \frac{1}{N}(\sum_{i=1}^{N} \mathbb{I}(m_i < v_l) + \sum_{i=1}^{N} \mathbb{I}(m_i > v_h)), \tag{3-11}$$

其中 $\mathbb{I}$ 代表指示函数，N 表示 mask 的尺寸即像素点个数，当 $\boldsymbol{m}$ 的第 i 个元素 m_i 小于阈值 v_l（本文取 0.01）时则认为已经收敛到 0 值，当 m_i 大于阈值 v_h（本文取 0.99）时则认为已经收敛到 1 值。$\text{Check}(\boldsymbol{m})$ 就表示 mask 的二值化百分比，越接近 1 则二值化程度越高。如果二值化程度不足则需要以更大的步长来增大 α 以确保 mask 的元素在迭代结束时收敛到二值。迭代结束后，则将 mask 中收敛到 0 值的像素所对应的扰动剪去，保留收敛到 1 值的像素所对应的

算法 3.1 AutoAdversary

输入: 原始图像 $\boldsymbol{x}$，目标类别标签 y^*，目标分类模型 f，最大迭代次数 T，l_∞ 范数阈值 ϵ，平衡参数 γ 和 C，缩放因子 α_{start} 和 α_{end}，更新步长 β
输出: 稀疏对抗样本 $\boldsymbol{x}^{\text{adv}}$

$\boldsymbol{\delta}_0 \leftarrow \mathbf{0}$;
随机初始化编码器网络 $\mathcal{H}_0$;
$\alpha_0 \leftarrow \alpha_{\text{start}}$;
$\alpha_{\text{step}} \leftarrow \frac{\alpha_{\text{end}}-\alpha_{\text{start}}}{T}$;
for $t = 0, 1, ..., T-1$ **do**
  $\boldsymbol{m}_t \leftarrow \text{sigmoid}(\alpha_t \cdot \mathcal{H}_t(\boldsymbol{\delta}_t))$;
  根据 $\boldsymbol{m}_t$ 通过式 (3-5) 计算动态平衡参数 λ_t;
  $\mathcal{L} \leftarrow \mathcal{L}_{\text{CE}}(f(\boldsymbol{x} + \boldsymbol{\delta}_t \odot \boldsymbol{m}_t), y^*) + \lambda \frac{\|\boldsymbol{m}_t\|_1}{N}$;
  $\boldsymbol{\delta}_{t+1} \leftarrow \text{Clip}_\epsilon\{\boldsymbol{\delta}_t - \beta \cdot \text{sign}(\nabla_{\boldsymbol{\delta}}\mathcal{L})\}$;
  基于损失 $\mathcal{L}$ 利用带动量的梯度下降更新一次编码器得到 $\mathcal{H}_{t+1}$;
  if $\frac{t}{T} > 0.9$ **and** $\text{Check}(\boldsymbol{m}_t) < 0.99$ **then**
    $\alpha_{t+1} \leftarrow \alpha_t + 10 \cdot \alpha_{\text{step}}$;
  else
    $\alpha_{t+1} \leftarrow \alpha_t + \alpha_{\text{step}}$;
  end if
end for
$\boldsymbol{m}_T \leftarrow \text{sigmoid}(\alpha_T \cdot \mathcal{H}_T(\boldsymbol{\delta}_T))$;
$\boldsymbol{m}_T \leftarrow \text{Binary}(\boldsymbol{m}_T)$;
$\boldsymbol{x}^{\text{adv}} \leftarrow \boldsymbol{x} + \boldsymbol{\delta}_T \odot \boldsymbol{m}_T$;
return $\boldsymbol{x}^{\text{adv}}$.

扰动。具体的，将 mask 的每个元素按照下式进行绝对二值化：

$$\text{Binary}(m_i) = \begin{cases} 1, & \text{if } m_i \geq 0.5 \\ 0, & \text{otherwise} \end{cases} . \tag{3-12}$$

然后再将绝对二值化后的 mask 与扰动变量按照元素相乘即可得到稀疏对抗扰动。算法 3.1详细总结了以 I-FGSM 为基础攻击的 AutoAdversary 算法流程。

3.2.3.2 基于 PGD 进行删减

本文在2.2.3节也介绍了 PGD，该方法与 I-FGSM 非常类似，同样也需要预先指定任意像素的最大扰动幅度即扰动 $\boldsymbol{\delta}$ 的 l_∞ 范数，不同点在于该方法的 $\boldsymbol{\delta}$ 初始值并非为 $\mathbf{0}$，而是以 ϵ 为半径的 l_∞ 范数球内的随机一点。AutoAdversary 同样可以轻松以 PGD 方法作为基础的稠密对抗攻击，进而产生稀疏对抗样本，并且满足其原本的 l_∞ 范数约束条件。具体的，与算法 3.1唯一的不同在于 $\boldsymbol{\delta}$ 的

初始化方式：

$$\boldsymbol{\delta}_0 = U(-\epsilon, \epsilon), \tag{3-13}$$

其中 ϵ 是预设的 l_∞ 范数阈值，$U(-\epsilon, \epsilon)$ 表示在 $[-\epsilon, \epsilon]$ 的均匀分布中抽样，并用该值填充 $\boldsymbol{\delta}_0$ 的每个元素。除此以外，算法流程中的损失计算、编码器网络的更新以及二值化缩放因子的更新都与算法3.1保持一致。

3.2.3.3 基于 MI-FGSM 进行删减

本文在2.2.3节也介绍了 MI-FGSM，该方法在 I-FGSM 的基础上引入了动量，使得构建出的对抗样本跳出局部最优。AutoAdversary 同样可以将 MI-FGSM 方法作为基础的稠密对抗攻击，进而产生稀疏对抗样本，并且满足其原本的 l_∞ 范数约束条件。具体的，$\boldsymbol{\delta}$ 的更新过程需要引入动量 $\boldsymbol{g}$：

$$\begin{aligned} \boldsymbol{g}_0 &= \mathbf{0}, \\ \boldsymbol{\delta}_0 &= \mathbf{0}, \\ \boldsymbol{g}_{t+1} &= \mu \cdot \boldsymbol{g}_t + \frac{\nabla_{\boldsymbol{x}} \mathcal{L}}{\|\nabla_{\boldsymbol{x}} \mathcal{L}\|_1}, \\ \boldsymbol{\delta}_{t+1} &= \mathrm{Clip}_{\epsilon}\{\boldsymbol{\delta}_t - \beta \cdot \mathrm{sign}(\boldsymbol{g}_{t+1})\}, \end{aligned} \tag{3-14}$$

其中 $\boldsymbol{g}_t$ 是第 t 次迭代中的动量，μ 是动量衰减因子。除此以外，算法流程中的损失计算、编码器网络的更新以及二值化缩放因子的更新都与算法3.1保持一致。

3.3 实验与分析

本节将会在不同数据集、不同网络模型上进行实验，以展示 AutoAdversary 的性能。本节主要分为 4 个部分，第一部分将详细介绍实验的设置包括实验环境、实验数据集、评价指标以及部分实现细节；第二部分将通过删减前后的对比实验说明利用 AutoAdversary 能够减小扰动像素点的个数，体现其有效性；第三部分将会与其他稀疏对抗攻击进行比较，以展现 AutoAdversary 的优越性；第四部分将会对 AutoAdversary 的各个模块进行消融实验，进而说明各个模块的作用及其必要性。

3.3.1 实验设置

3.3.1.1 实验环境

本文中的所有实验都在以下软硬件环境中进行。

(1) CPU：Intel(R) Xeon(R) CPU E5-2678 v3 @ 2.50GHz

(2) 内存：125GB

(3) GPU：GeForce RTX 2080Ti

(4) 操作系统：Ubuntu 16.04.6 LTS

(5) 编程语言和编程框架：Python 3.6，PyTorch 1.4.0

3.3.1.2 数据集设置与目标分类模型

本文在两个广泛使用的图像分类数据集 CIFAR-10[65]和 ImageNet[66]上进行了实验。CIFAR-10 数据集包含 10 个类别共 60000 张图片，其中训练集包含 50000 张图片，测试集包含 10000 张图片。本文使用的是 ImageNet 数据集的常用子集 ISLVRC2012，其类别个数为 1000，训练集包含 100 多万张图像，并且有 50000 张图像用于验证。由于对抗攻击的特殊性，本文首先在训练集上训练目标分类模型，然后仅在验证集或者测试集中选取图像进行对抗攻击。

在 CIFAR-10 数据集上，本文首先训练了一个输入尺寸为 $32 \times 32 \times 3$ 的 ResNet-18[67]模型。对于训练数据的预处理，本文使用了随机填充裁剪、随机水平翻转、归一化以及标准化。归一化操作即是将图像像素范围从 $[0, 255]$ 归一化为 $[0, 1]$，紧接着的标准化参数为：

$$\begin{aligned} \boldsymbol{\mu}_{\text{CIFAR-10}} &= [0.4914, 0.4822, 0.4465], \\ \boldsymbol{\sigma}_{\text{CIFAR-10}} &= [0.2023, 0.1994, 0.2010], \end{aligned} \tag{3-15}$$

其中 $\boldsymbol{\mu}_{\text{CIFAR-10}}$ 和 $\boldsymbol{\sigma}_{\text{CIFAR-10}}$ 分别表示 CIFAR-10 数据集上 RGB 图像 3 通道的均值和标准差。对于测试图像的预处理，本文仅使用归一化，并在图像输入模型前进行标准化，标准化参数与训练时保持一致。最终得到的 ResNet-18 模型在 CIFAR-10 的完整测试集上的 top-1 分类准确率为 95.460%。紧接着本文取出 CIFAR-10 的测试集的前 1000 张图像，由于在模型本已分类错误的图像上进行对抗攻击是没有意义的，因此本文仅使用其中被正确分类的共 959 张图像。因为本文是进行目标攻击，需要指定出错的具体类别，因此对于每一张图像，本

文随机选取一个不正确的类别作为目标类别标签。

在 ImageNet 数据集上，本文使用 PyTorch 提供的预训练（pre-trained）Inception-v3[68]模型，因此无需再在训练集上进行训练。对于验证集图像的预处理，本文首先将图像的尺寸缩放为 $300 \times 300 \times 3$，再进行归一化，并在图像输入到模型前进行标准化，标准化的参数为：

$$\begin{aligned} \boldsymbol{\mu}_{\text{ImageNet}} &= [0.485, 0.456, 0.406], \\ \boldsymbol{\sigma}_{\text{ImageNet}} &= [0.229, 0.224, 0.225], \end{aligned} \tag{3-16}$$

其中 $\boldsymbol{\mu}_{\text{ImageNet}}$ 和 $\boldsymbol{\sigma}_{\text{ImageNet}}$ 分别表示 ImageNet 数据集上 RGB 图像 3 通道的均值和标准差。最终发现该 Inception-v3 模型在 ImageNet 完整验证集上的 top-1 分类准确率为 77.132%。同样的，本文仅在 ImageNet 验证集上选取图像进行对抗攻击。首先取出 ImageNet 验证集的随机 100 张图像，由于在模型本已分类错误的图像上进行对抗攻击是没有意义的，因此本文仅使用其中被正确分类的共 81 张图像。与 CIFAR-10 上的处理一样，ImageNet 数据集上的目标类别标签也是随机选取的。

3.3.1.3 实验评价指标

本文主要使用攻击成功率（Attack Success Rate，ASR）和所有扰动的平均 l_p 范数（$p = 0, 1, 2, \infty$）来评估各个方法的性能。如2.1.3节所述，ASR 主要衡量了方法的攻击效果，指的是在所有构建的对抗样本中，满足预设攻击条件的在总样本中的占比。本文进行有目标攻击，因此这里 ASR 指所有图像中被目标分类模型预测为目标类别标签的对抗样本所占的比例，即：

$$\text{ASR} = \frac{1}{M} \sum_{i=1}^{M} \mathbb{I}(y_i^* = \hat{y}_i), \tag{3-17}$$

这里 M 为总样本数，y_i^* 和 $\hat{y}_i$ 分别表示第 i 个样本的目标分类标签以及模型预测的分类标签，ASR 越高则方法的攻击能力越强。各个对抗样本的扰动的平均 l_p 范数度量了原始样本与对抗样本之间的相似性，主要用于评价方法的攻击隐蔽性。这里用 $\boldsymbol{\delta}_i$ 表示第 i 个样本的对抗扰动，则所有扰动的平均 l_p 范数表

示为：

$$l_p = \frac{1}{M}\sum_{i=1}^{M}\|\boldsymbol{\delta}_i\|_p\,, \tag{3-18}$$

2.1.3节详细介绍了不同 l_p 范数背后的意义，扰动的范数越小则对抗样本与原始样本的区别越难以被人眼察觉。这里需要注意的是，所有的 l_p 范数均在图像归一化到 $[0,1]$ 区间之后进行计算。

3.3.1.4 实现细节

本节主要描述了实现的细节，主要包括算法中一些超参数的设置方法。AutoAdversary 使用带有缩放因子的 sigmoid 函数作为二值化函数，并且在训练过程中逐步将缩放因子 α 从 α_{start} 逐渐增大到 α_{end}，以保证最终 mask 的元素收敛到二值，并且避免方法退化为随机选择像素点。这里首先需要找到能产生绝对二值 mask 的 α_{end}，实践表明仅需在两到三张图像上尝试几个不同的值，就能确定一个足够好的 α_{end}。具体的，我们分别在 CIFAR-10 和 ImageNet 数据集上将 α_{end} 设置为 100 和 10。实践表明不同的 α_{start} 对于结果的影响不大，因此本文在所有数据集上直接令 $\alpha_{\text{start}} = 0.1$。

式 (3-4) 是 AutoAdversary 的损失函数，其中 λ 是自动动态调整的平衡参数，λ 的定义如式 (3-5) 所示。特别需要注意的是，其中 C 和 γ 是预先定义的超参数，C 和 γ 越大，则在同样的 mask 下将会得到越大的 λ，也即是针对稀疏不足的惩罚越大；反之若 C 和 γ 越小，则针对稀疏不足的惩罚越小。具体的，我们在 CIFAR-10 和 ImageNet 数据集上都将 C 设置为 1，也即是 λ 的最小值被设置为 1。γ 的设置与具体的数据集相关，我们分别在 CIFAR-10 和 ImageNet 上将 γ 设置为 10 和 100。并且，可能有些图像上的 γ 设置过大而导致过于关注稀疏程度而忽略攻击效果使得攻击失败，我们会在攻击失败后自动减小 γ 直到攻击成功。实践表明需要自动减小 γ 的情况仅在极个别图像上发生，因此并不会拖慢算法的速度。

实验的随机部分包括图像的随机选择、目标类别的随机选择、编码器网络的随机初始化和对抗扰动的随机初始化。为了使得实验结果可复现，所有随机函数的随机种子均设置为 10。

3.3.2 有效性实验

AutoAdversary 作为一种通用的像素删减框架，能够在大多数现有稠密对抗攻击方法的基础上减少扰动像素点的个数，在满足原本约束条件的情况下进一步减小扰动量。本节主要给出 3.2.3节中提到的 3 个版本的 AutoAdversary 的在 CIFAR-10 和 ImageNet 数据集上的实验结果，以说明利用 AutoAdversary 能够进一步减小扰动像素点的个数，体现其有效性。

3.3.2.1 CIFAR-10 上的有效性实验

本节首先在小尺寸的 CIFAR-10 数据集上进行实验，其图像尺寸为 $32 \times 32 \times 3$，因此每张图像的总像素点个数为 3072。I-FGSM、PGD 和 MI-FGSM 都需要预先设定最大扰动幅度 ϵ 也即是扰动的 l_∞ 范数，因此本文在两种不同最大扰动幅度的情况下进行了实验。

表3-1展示了当 $\epsilon = 8/255$ 和 $\epsilon = 16/255$ 时，CIFAR-10 数据集上各个方法的攻击成功率以及平均 l_p 范数。当 $\epsilon = 8/255$ 时，虽然 I-FGSM、PGD 和 MI-FGSM 三种稠密对抗攻击方法分别取得了 99.8%、100.0% 和 100.0% 的 ASR，但是其 l_0 范数均非常高，也就是说被修改的像素点几乎就是全部的像素点。此时分别在这三种方法的基础上，我们利用 AutoAdversary 进行像素删减，进一步减小扰动。可以看到在删减后，ASR 均并未下降也即是删减并不会影响攻击效果，并且 l_0 范数也即是扰动像素点的个数得到了大幅度的下降，比如 MI-FGSM 在删减前后的平均扰动像素点个数从 2968.3（96.62% 的像素点）减少到了 172.3（5.61% 的像素点）。与此同时，l_1 范数与 l_2 范数也有大幅度的降低。需要注意的是，l_∞ 范数代表了任意像素的最大扰动幅度，可以看到删减后的 l_∞ 范数仍然保持不变，不会因为扰动像素点的减少而导致扰动幅度的增大，这更加说明删减过程中去掉的确实是冗余的扰动。当最大扰动幅度限制为 $\epsilon = 16/255$ 时，也有类似的实验结果。

因此本实验说明，在 CIFAR-10 数据集上利用 AutoAdversary 像素删减框架可以在一些稠密对抗攻击方法的基础上进一步减少扰动像素点的个数从而降低整体的扰动量，并且在删减过程中不会降低攻击成功率，还能够满足最大扰动幅度等约束条件。

表 3-1: CIFAR-10 上不同最大扰动幅度下的像素删减结果

Threshold	Method	ASR(%)	l_0	l_1	l_2	l_∞
$\epsilon = 8/255$	I-FGSM	99.8	2896.6	67.475	1.329	0.031
	I-FGSM*	**99.8**	**175.2**	**5.437**	**0.395**	**0.031**
	PGD	100.0	2896.9	67.440	1.329	0.031
	PGD*	**100.0**	**180.9**	**5.594**	**0.398**	**0.031**
	MI-FGSM	100.0	2968.3	82.600	1.564	0.031
	MI-FGSM*	**100.0**	**172.3**	**5.382**	**0.393**	**0.031**
$\epsilon = 16/255$	I-FGSM	100.0	2965.7	129.715	2.565	0.063
	I-FGSM*	**100.0**	**88.9**	**5.478**	**0.558**	**0.063**
	PGD	100.0	2976.3	129.646	2.563	0.063
	PGD*	**100.0**	**90.6**	**5.575**	**0.562**	**0.063**
	MI-FGSM	100.0	2995.3	152.279	2.940	0.063
	MI-FGSM*	**100.0**	**86.3**	**5.337**	**0.552**	**0.063**

* 表示以该方法作为基础对抗攻击，并利用 AutoAdversary 进行像素删减减小扰动。

3.3.2.2 ImageNet 上的有效性实验

上一小节的实验已经说明了 AutoAdversary 在小尺寸图像上具有有效性，本节将在大尺寸图像 ImageNet 上进行实验。本文将 ImageNet 上的图像的尺寸统一缩放为 $300 \times 300 \times 3$，因此每张图像的像素点总个数为 270000。与 CIFAR-10 上的实验设置一样，这里我们也以 I-FGSM、PGD 和 MI-FGSM 这三种稠密对抗攻击为基础，并用 AutoAdversary 进行像素删减，进一步减小扰动。

表 3-2展示了当 $\epsilon = 8/255$ 和 $\epsilon = 4/255$ 时，ImageNet 数据集上各个方法的攻击成功率以及平均 l_p 范数。当 $\epsilon = 8/255$ 时，虽然 I-FGSM、PGD 和 MI-FGSM 三种稠密对抗攻击方法分均取得了 100.0% 的 ASR，但是其 l_0 范数都非常高。可以看到在删减后，ASR 均未下降，并且 l_0 范数得到了大幅度的降低，比如 MI-FGSM 在删减前后的平均扰动像素点个数从 254325.8（94.19% 的像素点）减少到了 5442.9（2.02% 的像素点）。与此同时，l_1 范数与 l_2 范数也有大幅度的降低。与 CIFAR-10 上的结果一样，删减后的 l_∞ 范数仍然保持不变，不会因为扰动像素点的减少而导致扰动幅度的增大。当最大扰动幅度限制为

表 3-2: ImageNet 上不同最大扰动幅度下的像素删减结果

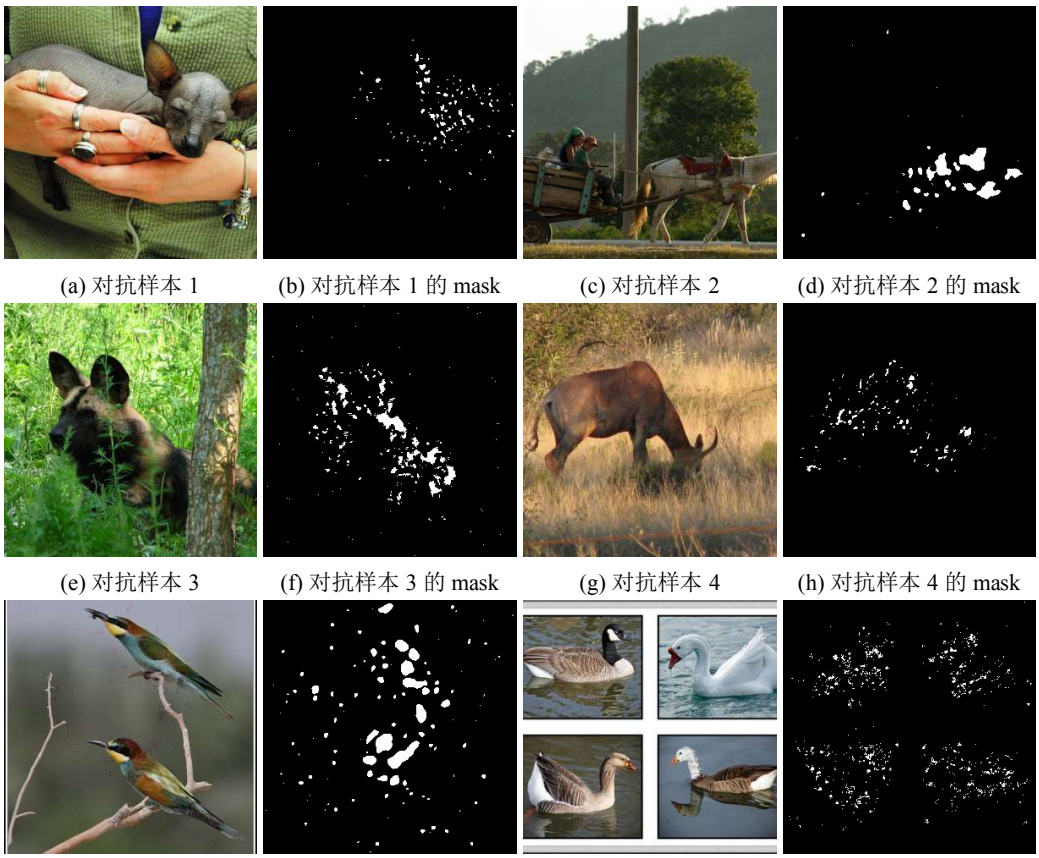
Threshold	Method	ASR(%)	l_0	l_1	l_2	l_∞
$\epsilon = 8/255$	I-FGSM	100.0	240118.5	4744.600	10.595	0.031
	I-FGSM*	**100.0**	**10592.4**	**273.389**	**2.705**	**0.031**
	PGD	100.0	241981.2	4775.905	10.629	0.031
	PGD*	**100.0**	**8340.8**	**229.587**	**2.513**	**0.031**
	MI-FGSM	100.0	254325.8	6610.318	13.753	0.031
	MI-FGSM*	**100.0**	**5442.9**	**164.851**	**2.183**	**0.031**
$\epsilon = 4/255$	I-FGSM	100.0	219698.3	2616.715	5.880	0.016
	I-FGSM*	**100.0**	**26547.7**	**351.169**	**2.187**	**0.016**
	PGD	100.0	221070.6	2624.191	5.885	0.016
	PGD*	**100.0**	**27758.1**	**372.702**	**2.261**	**0.016**
	MI-FGSM	100.0	248838.5	3566.601	7.303	0.016
	MI-FGSM*	**100.0**	**12900.6**	**198.117**	**1.696**	**0.016**

* 表示以该方法作为基础对抗攻击，并利用 AutoAdversary 进行像素删减减小扰动。

$\epsilon = 4/255$ 时，也有类似的实验结果。

CIFAR-10 和 ImageNet 数据集上的实验结果都说明了，一般的稠密对抗攻击所产生的对抗扰动中包含大量的冗余，而 AutoAdversary 可以以这些方法为基础进行自动的像素删减从而减少冗余。更重要的是，AutoAdversary 在删减过程中不会降低攻击成功率，也不会增大任意像素的扰动幅度。

图 3-5展示了利用 AutoAdversary 在 ImageNet 上进行稀疏对抗攻击得到的对抗样本以及对应的扰动区域 mask。这些图像均为添加扰动后的对抗样本，我们可以发现这些图像对于人眼来说都是正常的。更为重要的是，扰动区域 mask 中白色表示有扰动而黑色表示没有任何扰动，观察后可以发现扰动区域与图像中的主体对象有大量的重叠，这是因为图像中的主体对象对分类结果起到了关键性的作用，进一步从可视化的角度说明了 AutoAdversary 能够找到图像中最重要和最敏感的像素点，在扰动极少像素的情况下成功完成攻击。



(a) 对抗样本 1 (b) 对抗样本 1 的 mask (c) 对抗样本 2 (d) 对抗样本 2 的 mask

(e) 对抗样本 3 (f) 对抗样本 3 的 mask (g) 对抗样本 4 (h) 对抗样本 4 的 mask

(i) 对抗样本 5 (j) 对抗样本 5 的 mask (k) 对抗样本 6 (l) 对抗样本 6 的 mask

图 3-5: 可视化稀疏对抗攻击的结果

3.3.3 对比实验

上一节的有效性实验说明了 AutoAdversary 可以进行像素删减从而进一步减小扰动像素点的个数。本文在2.3节介绍了现有的一些稀疏对抗攻击方法，这些方法也立足于扰动图像中尽可能少的点，因此还需要将 AutoAdversary 与现有的稀疏对抗攻击方法进行比较，进而说明 AutoAdversary 的优越性。

需要注意的是，3.2.3.2节和3.2.3.3节介绍的是 AutoAdversary 的两种不同的版本，这两种版本分别基于 PGD 和 MI-FGSM。PGD 改进了扰动变量的初始化方式，MI-FGSM 改进了扰动变量的更新方式，这两种方法并不冲突因此可以结合起来。所以在本实验中，AutoAdversary 将 PGD 和 MI-FGSM 结合起来共同作为基础的稠密对抗攻击，从而取得更好的性能。

正如3.1.1节所提到的，当允许的任意像素最大扰动幅度即 l_∞ 范数很大时，往往扰动非常少的像素点就可以成功攻击；而当允许的 l_∞ 范数比较小时，

一般就需要扰动更多的像素点才能成功攻击。因此稀疏对抗攻击的难度随着 l_∞ 范数限制的不同将有很大的不同。由于不同方法考虑的约束有些许差别，一些稀疏对抗攻击方法能够控制最终的稀疏程度（l_0 范数），但是无法确保最大扰动幅度（l_∞ 范数）处于预设的范围内；相反的另一些方法能够确保扰动的 l_∞ 范数处于预设的范围内，但是无法控制最终的 l_0 范数。因此为了公平的比较，对于那些与 AutoAdversary 一样能够控制 l_∞ 范数的方法，则都在相同的 l_∞ 范数约束下进行攻击；对于那些无法控制 l_∞ 范数但是能够控制 l_0 范数的方法，则先在某个 l_0 范数的约束下进行攻击，进一步由 AutoAdversary 参考其实验结果的平均 l_∞ 范数来进行攻击。

3.3.3.1 CIFAR-10 上的对比实验

我们同样先在 CIFAR-10 这类小尺寸图像数据集上进行实验。表3–3和表3–4展示了不同的最大扰动幅度下，不同稀疏对抗攻击方法的 ASR 和平均 l_p 范数。比如当 $\epsilon = 8/255$ 时，我们提出的 AutoAdversary 达到了 100% 的 ASR，并且其 l_0 范数仅为 173.3（5.64% 的像素点）；当 $\epsilon = 16/255$，我们提出的 AutoAdversary 同样达到了 100% 的 ASR，并且其 l_0 范数仅为 87.2（2.83% 的像素点）。JSMA、PGD-$l_0 + l_\infty$ 和 GreedyFool 能够预先指定 l_∞ 范数，因此这三种方法与我们提出的 AutoAdversary 一样都在同样的 ϵ 限制下进行攻击。可以看到无论是 $\epsilon = 8/255$ 还是 $\epsilon = 16/255$，这三种方法的 ASR 都不高于我们，特别是 JSMA 和 PGD-$l_0 + l_\infty$ 的 ASR 甚至不能达到 100%，并且其平均 l_0 范数、l_1 范数、l_2 范数均显著大于我们。

C&W-l_0 和 SAPF 不能预先指定 l_∞ 范数但是可以保证 l_0 范数在一定范围内。为了公平的比较，本实验令这两种方法先在某个 l_0 范数条件下进行攻击，转而我们的 AutoAdversary 参考其结果中的平均 l_∞ 来攻击，确保两者的平均 l_∞ 范数保持一致。结果如表3–4所示，可以看到在多个不同的 l_∞ 条件下，我们的 l_0 范数都是远远小于这两种方法的，也就是说在同样的攻击难度下，我们的方法能够找到图像中最需要被扰动的像素点，从而取得更强的稀疏性。一个有趣的现象在于，同样的最大扰动幅度下，我们的扰动像素点个数更少，但是扰动的 l_2 范数和 l_1 范数却大于 C&W-l_0 和 SAPF，也就是说 C&W-l_0 和 SAPF 产生的扰动中，很多像素点的扰动幅度很小，这意味着这些像素点可能并不是必须要扰动的像素点，这进一步说明了我们的方法更能够找到图像中需要被扰动的像素点。因此上述实验结果均显示了 AutoAdversary 的优越性。

表 3-3: CIFAR-10 上各稀疏对抗攻击的对比（预设最大扰动幅度）

Threshold	Method	ASR(%)	l_0	l_1	l_2	l_∞
$\epsilon = 8/255$	JSMA	96.5	327.7	10.212	0.517	0.031
	PGD-$l_0 + l_\infty$	74.8	1499.5	45.825	1.193	0.031
	GreedyFool	100.0	420.4	8.637	0.429	0.031
	AutoAdversary	**100.0**	**173.3**	**5.412**	**0.393**	**0.031**
$\epsilon = 16/255$	JSMA	98.9	182.1	11.232	0.769	0.063
	PGD-$l_0 + l_\infty$	84.7	1199.7	71.782	2.104	0.063
	GreedyFool	100.0	274.8	9.727	0.611	0.063
	AutoAdversary	**100.0**	**87.2**	**5.396**	**0.555**	**0.063**

表 3-4: CIFAR-10 上各稀疏对抗攻击的对比（参考平均最大扰动幅度）

Threshold	Method	ASR(%)	l_0	l_1	l_2	l_∞
$\epsilon = 0.045$	C&W-l_0	100.0	383.1	**5.988**	**0.345**	0.045
	AutoAdversary	**100.0**	**119.3**	5.323	0.467	**0.045**
$\epsilon = 0.070$	C&W-l_0	100.0	194.2	**5.036**	**0.406**	0.070
	AutoAdversary	**100.0**	**78.6**	5.411	0.587	**0.070**
$\epsilon = 0.055$	SAPF	100.0	400.1	**4.720**	**0.292**	0.055
	AutoAdversary	**100.0**	**98.9**	5.379	0.518	**0.055**
$\epsilon = 0.088$	SAPF	100.0	201.8	**4.495**	**0.390**	0.088
	AutoAdversary	**100.0**	**64.9**	5.585	0.668	**0.088**

3.3.3.2 ImageNet 上的对比

本节将在 ImageNet 这样的大尺寸图像数据集上进行实验。需要注意的是，由于 JSMA 在大尺寸图像上的计算复杂度太高[39]，因此本实验不与 JSMA 进行比较。同时 SAPF 提供的正式代码实现[57]无法在 ImageNet 数据上收敛，因此我们也不与 SAPF 进行比较。表 3-5和表 3-6展示了不同的最大扰动幅度下，不同稀疏对抗攻击方法的 ASR 和平均 l_p 范数。比如当 $\epsilon = 8/255$ 时，可以看到我们提出的 AutoAdversary 达到了 100% 的 ASR，并且其 l_0 范数仅为 5591.5

表 3-5: ImageNet 上各稀疏对抗攻击的对比（预设最大扰动幅度）

Threshold	Method	ASR(%)	l_0	l_1	l_2	l_∞
$\epsilon = 8/255$	PGD-$l_0 + l_\infty$	50.0	11989.0	307.416	2.957	0.031
	GreedyFool	100.0	12454.5	293.080	2.663	0.031
	AutoAdversary	**100.0**	**5591.5**	**171.299**	**2.223**	**0.031**
$\epsilon = 4/255$	PGD-$l_0 + l_\infty$	62.2	22980.9	308.048	2.116	0.016
	GreedyFool	100.0	26107.7	338.756	2.083	0.016
	AutoAdversary	**100.0**	**12855.0**	**199.003**	**1.694**	**0.016**

（2.08% 的像素点）；当 $\epsilon = 4/255$，我们提出的 AutoAdversary 同样达到了 100% 的 ASR，并且其 l_0 范数仅为 12855.0（4.76% 的像素点）。PGD-$l_0 + l_\infty$ 和 GreedyFool 能够预先指定 l_∞ 范数，因此这两种方法与 AutoAdversary 都在同样的 ϵ 条件下进行攻击。可以看到 PGD-$l_0 + l_\infty$ 的攻击成功率非常低，并且在两种 ϵ 的情况下，PGD-$l_0 + l_\infty$ 和 GreedyFool 的平均 l_0 范数、l_1 范数、l_2 范数均显著大于我们。

与3.3.3.1中的实验设置一样，本实验也令 C&W-l_0 先在某个 l_0 范数条件下进行攻击，而后 AutoAdversary 参考其结果中的平均 l_∞ 范数来攻击。结果如表3-6所示, 与 CIFAR-10 上的现象基本相同，在多个不同的 l_∞ 条件下，我们的 l_0 范数都是远远小于 C&W-l_0 的。并且，由于 C&W-l_0 产生的扰动的 l_2 范数和 l_1 范数更小，可以分析出来其中有大量的像素点的扰动幅度很小，而这些像素点可能并不是都是必须要被扰动的，这进一步说明了我们的方法更能够找到图像中需要被扰动的像素点。

通过分析 CIFAR-10 和 ImageNet 上的实验结果可知，在相同的最大扰动幅度的限制下，我们提出的 AutoAdversary 能够更加有效地找到那些最需要被扰动地像素点，进而产生最稀疏的对抗扰动。

3.3.4 消融实验

3.3.2节和3.3.3.1节的实验说明了我们提出的 AutoAdversary 的有用性和优越性。本节将进一步研究分析 AutoAdversary 各个模块的作用及其必要性。在本节的实验中，本文用 “Dense” 来表示上述实验中 AutoAdversary 所使用的基础稠密对抗攻击，也即是 PGD 与 MI-FGSM 的结合；用 “Random” 表示随

表 3-6: ImageNet 上各稀疏对抗攻击的对比（参考平均最大扰动幅度）

Threshold	Method	ASR(%)	l_0	l_1	l_2	l_∞
$\epsilon = 0.068$	C&W-l_0	100.0	5697.4	**81.230**	**1.257**	0.068
	AutoAdversary	**100.0**	**2098.8**	137.733	2.926	**0.068**
$\epsilon = 0.051$	C&W-l_0	100.0	13028.2	**102.081**	**1.084**	0.051
	AutoAdversary	**100.0**	**3011.0**	148.621	2.647	**0.051**

机选取像素点进行扰动，而不是通过生成 01-mask 来选择需要扰动的像素点；用 “l_1-$\boldsymbol{\delta}$” 表示绕过 AutoAdversary 的分支结构，直接在损失函数中添加 $\|\boldsymbol{\delta}\|_1$ 以试图得到稀疏的扰动；用 “l_1-$\boldsymbol{m}$” 表示遵循 AutoAdversary 的流程，使用分支生成 mask 后在损失函数中添加 mask 的 l_1 范数以间接得到稀疏的扰动；用 “Binarization” 表示分支结构中带有缩放因子的 sigmoid 函数，用以进行二值化操作；用 “Encoder” 表示分支结构中对扰动变量进行编码的可训练神经网络；用 “Attention” 表示使用注意力机制来试图得到稀疏扰动。

本节在 CIFAR-10 数据集上进行了消融实验并设置最大扰动幅度 $\epsilon = 8/255$，实验结果如表 3-7所示。可以看到 “Dense” 能够达到 100% 的 ASR，但是其 l_0 范数为 2965.0（96.5% 的像素点），也就是说本文中 AutoAdversary 所使用的稠密对抗攻击本身具有较强的攻击能力但是需要扰动几乎全部的像素点。

“Dense + Random” 表示为每张图像生成一个静态的随机的 01-mask，然后同样利用 “Dense” 攻击方法来修改那些允许被扰动的像素点。可以看到随机选择需要被扰动的像素点时，ASR 仅为 6.2%，这说明 AutoAdversary 并非随机选择像素点，而是根据稀疏程度和攻击效果来优化得到的。

“Dense + Attention” 尝试使用注意力机制得到稀疏扰动。这里参照具有代表性的 SENet[69]的注意力方法，同样是添加分支网络并利用 sigmoid 函数生成元素值在 [0,1] 范围的 mask。与我们方法的不同在于，基本的注意力方法并不会在优化过程中改变 sigmoid 函数的缩放因子，因此其最终并不能产生绝对稀疏的扰动。实验结果也显示其 l_0 范数为 3046.7（99.17% 的像素点）。换句话说，我们的方法也可以看作是 “注意力机制” 与 “逐渐绝对二值化” 的结合，一般的注意力方法因为缺乏绝对二值化，最终得到的扰动并不稀疏。

如式 (3-4) 所示，AutoAdversary 将 mask 的 l_1 范数加入到了损失函数中，意在使得生成的 mask 尽可能稀疏。作为对比，“Dense + l_1-$\boldsymbol{\delta}$” 表示绕开 mask 的

表 3-7: AutoAdversary 各模块消融实验

Method	ASR(%)	l_0	l_1	l_2	l_∞
Dense	100.0	2965.0	82.215	1.559	0.031
Dense + Random	6.2	782.8	23.059	0.839	0.031
Dense + Attention	99.8	3046.7	91.058	1.668	0.031
Dense + l_1-$\boldsymbol{\delta}$	99.5	2422.6	29.641	0.703	0.031
Dense + l_1-$\boldsymbol{m}$ + Binarization	98.4	256.8	7.987	0.478	0.031
Dense + l_1-$\boldsymbol{m}$ + Binarization + Encoder	**100.0**	**173.3**	**5.412**	**0.393**	**0.031**

分支转而直接生成稀疏的扰动，也就是将扰动变量 $\boldsymbol{\delta}$ 的 l_1 范数加入到损失函数中。可以看到该方法的 ASR 不能达到 100%，并且其 l_0 范数为 2422.6（78.86% 的像素点）。正如 3.2.2.3节所述，直接将扰动变量 $\boldsymbol{\delta}$ 的 l_1 范数 $\|\boldsymbol{\delta}\|_1$ 作为正则项无法得到稀疏的对抗扰动。而之所以 AutoAdversary 能够产生稀疏的扰动，是因为其添加的分支结构中包含二值化模块，得到的 $\boldsymbol{m}$ 是近似二值化的，随着缩放因子 α 逐步增大，$\boldsymbol{m}$ 二值化的程度越来越高，这就使得 $\boldsymbol{m}$ 中的大量元素被优化到 0 值或者非常接近 0，进而产生了稀疏的对抗扰动。这个实验也证明了 AutoAdversary 所添加的分支结构是有效的。

验证完整个分支结构的作用后，需要进一步验证分支结构中编码器网络的必要性，这里直接将扰动变量 $\boldsymbol{\delta}$ 绕过编码器网络转而直接输入到二值化模块中生成 mask，并表示为 “Dense + l_1-$\boldsymbol{m}$ + Binarization”。可以看到没有编码器网络后，虽然可以产生较为稀疏的对抗扰动，但是 ASR 下降较多。最直观的原因是分支中的二值化模块是 sigmoid 函数，如果不通过编码就直接将 $\boldsymbol{\delta}$ 输入到二值化模块中，则所有负向的扰动都会迫使 mask 的对应值为小于 0.5，最终通过式 (3-12) 被置 0，即永远无法产生负向的扰动，因此会缩减对抗空间，不利于构建对抗样本。因此编码器网络可以增大对抗攻击的解空间。

“Dense + l_1-$\boldsymbol{m}$ + Binarization + Encoder”就是 AutoAdversary 的完整结构，可以看到其无论在 ASR 还是平均 l_p 范数的指标上都是最优的。因此本消融实验证实了 AutoAdversary 各个模块的作用和必要性。

3.4 本章小结

本章首先对稀疏对抗攻击这个任务进行了分析和形式化，并简述了现有方法的特点和存在的问题，进而根据神经网络自动剪枝与稀疏对抗攻击的内在相似性，设计了基于像素自动删减的稀疏对抗攻击框架（AutoAdversary）。具体的，AutoAdversary 在通常的对抗攻击过程中增加了一个包含编码和二值化两种操作的分支，该分支输出一个 01-mask 来自动选择图像像素点，并通过联合优化扰动变量以及分支网络中的权重，使得大量像素点在 mask 中的对应值为 0，最终完成了稀疏对抗攻击。因此，AutoAdversary 可以在攻击过程中端到端地自动选择最需要被扰动的一批像素点，无需人为定义像素点重要性评价指标，并且不包含贪心的选取策略。紧接着本章在 CIFAR-10 和 ImageNet 数据集上进行了充分的实验，首先通过删减前后的对比实验说明利用 AutoAdversary 能够大量减少扰动像素点的个数，体现了本方法的有效性；然后通过与其他现有稀疏对抗攻击进行比较，说明了本方法的优越性；最后通过拆解 AutoAdversary 的各个模块进行消融实验，阐述了各个关键模块的作用以及必要性。

第四章　基于稀疏对抗攻击的图像数据增强

上一章我们提出了一个基于像素自动删减的稀疏对抗攻击算法。尽管利用该算法构建的对抗样本具有强大的攻击能力和较高的隐蔽性，并且实验结果显示相对于其他算法也具有优越性，但是这仍然无法直接为深度学习模型带来正面的影响。为此，本章从挖掘对抗攻击的正面应用出发，提出了一种基于稀疏对抗攻击的图像数据增强算法，旨在利用稀疏对抗攻击寻找图像中脆弱且重要的像素点，并结合“信息删除”数据增强模式，在训练过程中帮助模型避免过拟合，提升模型的泛化性。最终的实验结果也展现了本数据增强方法的有效性和优越性。

4.1　稀疏对抗攻击与数据增强的联系

2.4节对图像数据增强进行了简介，并着重介绍了其中应用广泛的基于信息删除的一类方法。可以总结发现这类方法旨在通过某些策略在训练过程中删除输入图像上的部分信息，迫使神经网络模型不依赖于局部的、特定的视觉特征，而是更好地利用图像的全部上下文，进而显著提高了模型的泛化性。但是如前文所述，现有的方法都是按照一定的策略在全图中随机产生删除区域，对于一些复杂的图像仍然无法很好地平衡区域的删除和保留，也即是可能完全删除了图像中的对象或者仅删除了背景。即便 GridMask 在统计上能够一定程度避免这个问题，但是也无法保证这样的问题不会发生。

为了减少这种因为全图随机性带来的低质量样本，我们认为需要为每张图像单独定制个性化的删除区域，因此我们提出了名为 AdvMask 的方法，将稀疏对抗攻击结合到基于信息删除的图像数据增强中。如第 3 章所述，我们提出的稀疏对抗攻击方法 AutoAdversary 能够找到图像中对于分类来讲最敏感的区域，正常训练出来的模型往往过于关注这些敏感的区域而忽略了全图的上下文信息从而丧失了一定的泛化能力。因此我们首先通过 AutoAdversary 找到每个

图像中的若干敏感区域，然后基于敏感区域再生成需要删除的区域，从而能够更加有效地进行数据增强。

总的来说，我们提出的基于稀疏对抗攻击的数据增强相较于其他方法有以下优势：首先，AdvMask 产生的删除区域是基于对抗敏感区域的，因此删除区域不限于某一个连续的区域而是多个区域，并且形状是多样的；其次，本方法产生的删除区域对于每一张图像来讲都是个性化的而不是完全随机的，这包括了删除区域的位置、大小以及区域个数等，因此对于任一张图像来讲都能够有效增强；更重要的是，本方法利用稀疏对抗攻击找到了图像中最敏感的区域，而不是仅仅关注图像中主体对象所在区域，因此更能够针对模型的缺陷和弱点来进行数据增强，鼓励模型更多关注那些不太敏感的重要特征。

4.2 基于稀疏对抗攻击的图像数据增强

4.2.1 算法整体流程

前面小节分析了现有数据增强方法存在的问题并引出了我们所提出的名为 AdvMask 的方法。该方法利用我们在第 3 章提出的稀疏对抗攻击寻找图像中最敏感的区域，并在此基础上基于信息删除的方法来进行数据增强。如图4-1所示，AdvMask 分为 3 个大的步骤，首先需要以常规方式训练一个基准模型作为对抗攻击的目标分类模型；然后利用我们提出的 AutoAdversary 针对基准模型在训练集上进行稀疏对抗攻击，进而得到每张图像的扰动区域 mask；紧接着开始训练新的分类模型，在每一轮的训练中，利用4.2.3节的方法对扰动区域 mask 进行适当的随机变换以得到删除区域 mask，将图像中的对应区域删除后就得到了增强图像，进而通过增强后图像来训练模型。

4.2.2 稀疏对抗攻击生成敏感区域 mask

我们在第 3 章提出了稀疏对抗攻击 AutoAdversary，该方法能够找到对分类模型来讲最为重要和最敏感的像素点，从而在仅扰动极少部分像素点的情况下使得模型出错。形式上的，该方法通过一个包含编码和二值化的分支结构来生成 01-mask，进而确定需要被扰动的像素点，那么这里的 01-mask 就可以作为敏感区域 mask。因此我们就直接使用 AutoAdversary 在攻击过程中产生的 mask 来找到图像中最为敏感的一批像素点。

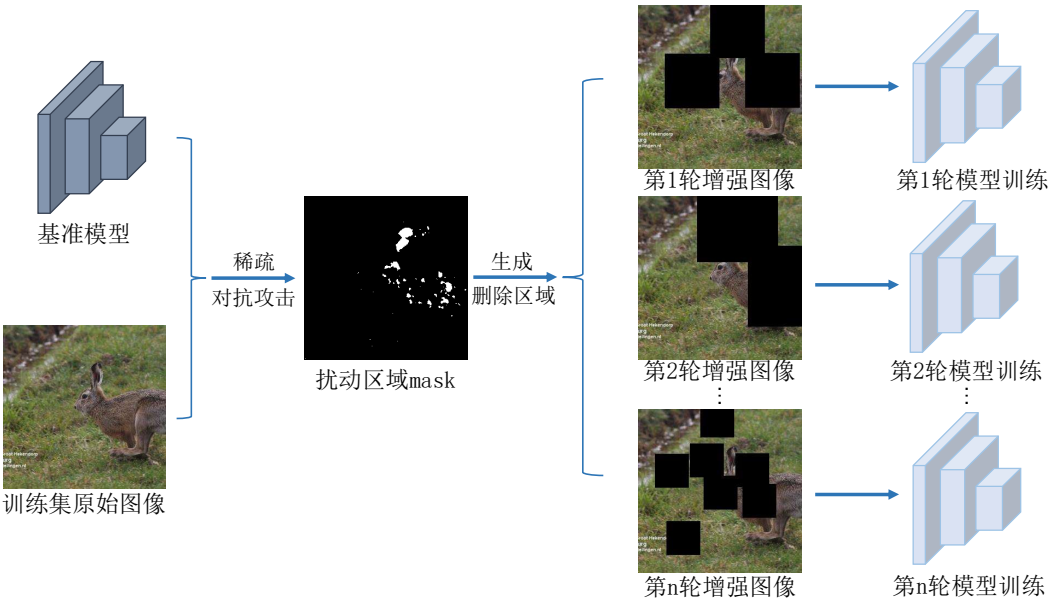


图 4-1: AdvMask 流程示意图

这里在使用 AutoAdversary 的时候有两点需要特别注意的地方。我们在第 3 章中都将 RGB 图像的 3 通道当作独立的 3 个像素点看待，可以独立地选择其中一个或者多个进行扰动，而这里我们的最终目的是删除图像上的一些区域以进行数据增强，所以需要把 RGB 图像 3 通道当作一个整体即一个像素来看待，此时这 3 个通道要么都被扰动要么都不被扰动。而正如3.2.2.1节所述，我们修改图3-3(b)中分支结构内编码器的输出尺寸为 $w \times h \times 1$ 即可，这样就可将 3 通道作为整体来进行稀疏对抗攻击。

还有一点需要注意的是，第 3 章中的对抗攻击都是在有目标攻击的情况下进行的，也就是说需要指定目标分类标签。但是这里我们的目的不局限于攻击模型，而是找到图像中的敏感区域，因此如果使用目标攻击可能会因为目标分类标签的不同而影响到敏感区域的确定，一个效率较低的方法是选择多个分类标签并多次进行目标攻击，并将取每次敏感区域的交集作为最终的敏感区域。这里为了更高效地生成敏感区域，我们进一步将 AutoAdversary 拓展为无目标攻击，也即是不再指定目标分类标签，只需保证模型无法将图像正确分类即可，这样产生的敏感区域只与图像的真实类别相关，只需进行一次攻击就可以得到可靠的敏感区域。具体来讲，我们最开始修改 AutoAdversary 的损失函数

$\mathcal{L}$ 如下：

$$\mathcal{L} = -\mathcal{L}_{\mathbf{CE}}(f(\boldsymbol{x} + \boldsymbol{\delta} \odot \boldsymbol{m}), y_{\text{true}}) + \lambda \frac{\|\boldsymbol{m}\|_1}{N}, \tag{4-1}$$

其中 $\mathcal{L}_{\text{CE}}$ 表示交叉熵损失函数，y_{true} 表示真实分类标签，可以看到与第 3 章中有目标攻击的损失函数非常类似，只是第一项改为了交叉熵损失的相反数，这是因为无目标攻击的目标是使得图像无法被分类为真实标签 y_{true}。但是式 (4-1) 中前后两项存在数值差异，因为交叉熵损失的相反数是没有下限的，而第二项的最小值为 0，导致优化过程中总是过度关注是否能够使分类错误，进而导致扰动的稀疏程度无法达到最佳。因此为了平衡两项的数值差异，我们首先在不考虑稀疏度的情况下对图像进行了一次无目标的稠密对抗攻击，当攻击成功时记录了当前的交叉熵损失的数值为 L_{init}，然后再修改 AutoAdversary 的损失函数 $\mathcal{L}$ 如下：

$$\begin{aligned} \mathcal{L} &= \max(\mathcal{L}_{\text{classify}}, \eta \cdot \mathcal{L}_{\text{classify}}) + \lambda \frac{\|\boldsymbol{m}\|_1}{N}, \\ \mathcal{L}_{\text{classify}} &= -\frac{\mathcal{L}_{\mathbf{CE}}(f(\boldsymbol{x} + \boldsymbol{\delta} \odot \boldsymbol{m}), y_{\text{true}})}{L_{\text{init}}} + 1, \end{aligned} \tag{4-2}$$

其中 $\mathcal{L}_{\text{classify}}$ 是放缩后的对抗分类损失函数，可以看到当 $\mathcal{L}_{\text{classify}} = 0$ 时，交叉熵损失 $\mathcal{L}_{\text{CE}}$ 与 $\mathcal{L}_{\text{init}}$ 相等，也即是说此时的图像已经能够使得模型预测出错，无需再进一步增大交叉熵损失 $\mathcal{L}_{\text{CE}}$ 了，因此我们利用 $\max(\mathcal{L}_{\text{classify}}, \eta \cdot \mathcal{L}_{\text{classify}})$ 来对 $\mathcal{L}_{\text{classify}}$ 函数曲线中小于 0 的部分进行线性衰减，衰减参数为 η（本文中取 0.05）。这样一来，就一定程度上保证了优化过程分类错误与稀疏扰动两项需求之间的平衡。

总的来说，在正常训练得到基准模型后，我们在此基准模型上运用我们提出的稀疏对抗攻击来得到图像的敏感区域 mask。除了编码器输出尺寸以及损失函数不同以外，其他步骤都与第 3 章中提出的算法保持一致。

4.2.3 利用敏感区域 mask 进行数据增强

在使用上一小节的方法生成敏感区域 mask 后，这一节将利用这些 mask 来删除图像的部分区域以达到数据增强的效果。需要注意的是，对于卷积神经网络来讲删除一些尺寸非常小的区域是不起作用的，而当前的敏感区域 mask 是呈散点状的，包含大量孤立的像素点，从图3-5也可以发现这一问题。因此我们需要进一步将其从一个个孤立的像素点扩充为尺寸相对较大的连续区域。

具体的，令 $\boldsymbol{m}_{\text{adv}}$ 表示稀疏对抗攻击产生的 mask，其中值为 1 的位置表示该像素点受到了扰动，也即是敏感像素。假设图像中有 n 个敏感像素，则敏感像素坐标集合可以表示为：$\mathcal{P} = \{(p_{x_1}, p_{y_1}), (p_{x_2}, p_{y_2}), ..., (p_{x_n}, p_{y_n})\}$。需要注意的是，对于那些对抗攻击不成功的图像，我们令其敏感像素坐标集合 $\mathcal{P}$ 为空集。将删除区域 mask 表示为 $\boldsymbol{m}_{\text{drop}}$，其尺寸为 $w \times h \times 1$，全部元素值都初始化为 1，并在过程中不断将某些连续区域的元素值置为 0。为了提升数据增强的多样性，我们从 $[d_l, d_h]$ 中随机选择每个连续区域的边长 d:

$$d = \text{Random}(d_l, d_h). \tag{4-3}$$

进一步的，我们迭代地在 $\mathcal{P}$ 中随机选点，开始将一个个像素点扩充为一个个连续区域。对于当前选择的像素点 (p_x, p_y)，我们以该点为连续区域的中心点，以随机选择的 d 为连续区域的边长，表示为 $R = (p_x, p_y, d)$，并在 $\boldsymbol{m}_{\text{drop}}$ 中将这个连续区域的所对应的元素值都置 0。需要注意的是，部分敏感像素点之间的位置可能比较接近，导致各自扩充为连续区域后存在较多的重叠，从而可能会导致图像中某些重要的区域被完全删除，因此这里我们预设最大重叠阈值 γ_{cover}，如果当前区域 R 与现有的连续区域的重叠度超过 γ_{cover}，则跳过当前选择的像素点，这样能够避免图像中重要对象被完全删除。在这之后为了进一步提升增强的多样性，我们还会将当前得到的 $\boldsymbol{m}_{\text{drop}}$ 进行小范围的随机平移。这里有还有一些细节需要注意。首先，如果删除的总面积过大，那么图像可能损失过多的信息，因此我们会确保删除的面积在适当的范围内。具体的，每个选择的像素点在扩充完成之后，我们会判断当前删除的面积是否超过阈值 S_h，如果超过则停止像素点的选择。除此以外，对于对抗攻击不成功的图像，也就是敏感像素坐标集合 $\mathcal{P}$ 为空集的图像，我们直接利用其他方法如 GridMask 或 RE 等对其进行数据增强。

最终，$\boldsymbol{m}_{\text{drop}}$ 中 0 值代表需要删除的像素，1 值代表需要保留的像素，则删除的具体操作为：

$$\boldsymbol{x}' = \boldsymbol{x} \odot \boldsymbol{m}_{\text{drop}}, \tag{4-4}$$

其中 $\boldsymbol{x}$ 表示原图，$\odot$ 表示按元素相乘，$\boldsymbol{x}'$ 表示删除部分连续区域之后的图像，由于这里的图像是经过标准化的，因此实际的填充值是数据集的均值。除此以外，此方法也可以与其他基于空间变换的数据增强相结合，比如可以在删除部分区域后继续对 $\boldsymbol{x}'$ 进行缩放、平移、裁剪等操作。

算法 4.1 AdvMask

输入: 原始训练集 $D = \{(\boldsymbol{x}^{(1)}, y^{(1)}), (\boldsymbol{x}^{(2)}, y^{(2)}), ..., (\boldsymbol{x}^{(N)}, y^{(N)})\}$，最大迭代次数 T，边长阈值 d_l 和 d_h，面积阈值 S_h，重叠阈值 γ_{cover}

输出: 训练完成后的模型 f_{aug}

利用原始训练集 D 训练一个基准模型 f_{base};
for $i = 1, 2, ..., N$ **do**
  针对样本 $(\boldsymbol{x}^{(i)}, y^{(i)})$ 调用 AutoAdversary 攻击 f_{base} 得到 $\boldsymbol{m}_{\text{adv}}^{(i)}$;
  $\mathcal{P}^{(i)} \leftarrow \boldsymbol{m}_{\text{adv}}^{(i)}$ 中值为 1 的元素坐标集合;
end for
随机初始化模型 f_{aug};
for $t = 0, 1, ...T-1$ **do**
  for $i = 1, 2, ..., N$ **do**
    if $|\mathcal{P}^{(i)}| = 0$ **then**
      $\boldsymbol{x}^{(i)} \leftarrow \text{GridMask}(\boldsymbol{x}^{(i)})$;
    else
      $\boldsymbol{m}_{\text{drop}} \leftarrow \{1\}^{w \times h \times 1}$;
      $d \leftarrow \text{Random}(d_l, d_h)$;
      $S \leftarrow 0$;
      for $j = 1, 2, ..., |\mathcal{P}^{(i)}|$ **and** $S < S_h$ **do**
        $(p_x, p_y) \leftarrow \mathcal{P}^{(i)}$ 中的一个随机未选择过的坐标点;
        $R \leftarrow (p_x, p_y, d)$;
        $S_{\text{cover}} \leftarrow d^2 - \text{Sum}(\boldsymbol{m}_{\text{drop}}(R))$
        if $S_{\text{cover}} < \gamma_{\text{cover}}$ **then**
          $\boldsymbol{m}_{\text{drop}}(R) \leftarrow 0$;
          $S \leftarrow w \times h - \text{Sum}(\boldsymbol{m}_{\text{drop}})$;
        end if
      end for
      $\boldsymbol{m}_{\text{drop}} \leftarrow \text{RandomTranslate}(\boldsymbol{m}_{\text{drop}})$;
      $\boldsymbol{x}^{(i)} \leftarrow \boldsymbol{x}^{(i)} \odot \boldsymbol{m}_{\text{drop}}$;
    end if
  end for
  利用更新后的训练集 D 训练一轮 f_{aug};
end for
return f_{aug}.

算法4.1详细总结了我们提出的 AdvMask 的算法流程。这里需要强调的是，虽然 AdvMask 中也有随机因素存在，但并不是全图范围的完全随机，这与其他方法有根本性的不同。总之，AdvMask 通过稀疏对抗攻击找到图像中对于分类来讲最敏感的像素，并基于敏感像素生成需要删除的区域，能够更加有效地进行数据增强，使得模型摆脱对这些少部分敏感信息的依赖，从而更多地关注全局上下文信息，最终提升模型的泛化能力。

4.3 实验与分析

本节将会在不同数据集、不同网络结构上进行实验，以展示 AdvMask 的性能。本节主要分为 2 个部分，第一部分将会与其他数据增强方法进行比较，以展现 AdvMask 的优越性；第二部分将会对 AdvMask 进行消融实验，进而说明利用稀疏对抗攻击找到的敏感区域与其他方法找到的区域有本质上的差别。

4.3.1 对比实验

4.3.1.1 CIFAR-10 和 CIFAR-100

CIFAR-10 数据集有 10 个类别标签，由 50000 张训练图像和 10000 张测试图像组成，每张图像的尺寸都为 $32 \times 32 \times 3$。需要注意的一点是，第 3 章的实验中我们使用测试集图像来生成对抗样本，而这里的测试集图像仅用于最后测量模型的准确率，我们仅在训练集图像上进行稀疏对抗攻击来生成敏感区域 mask。

基于信息删除的数据增强方法通常都可以与基于空域变换的数据增强方法相结合，现有的 Cutout、GridMask 等方法都在训练时还使用了基线增强如随机裁剪、随机翻转等。为了公平的比较，我们的方法与这些方法使用相同基线增强。具体的，除了运用 AdvMask 进行数据增强，我们还在这之后将图像尺寸填充为 $40 \times 40 \times 3$，然后随机裁剪一个 $32 \times 32 \times 3$ 的区域作为新的图像，并紧接着进行概率为 0.5 的随机水平翻转。这里需要注意的一点是，由于我们使用对抗攻击找到的敏感区域是图像进行基线增强以前的，所以这里需要在使用了 AdvMask 之后再进行基线增强，而不像其他方法在输入图像后首先进行基线增强。

为了实验的充分性，我们在多个网络结构如 ResNet-18、ResNet-44、ResNet-50、WideResNet-28-10 以及 ShakeShake-26-32 上进行了实验。具体的，这里使用带动量的梯度下降法来优化模型权重，其中动量参数为 0.9，权重衰减参数为 5×10^{-4}。在训练过程中，学习率初值为 0.1，并分别在 60、120 以及 160 轮的时候衰减 10 倍，总训练轮数为 200。关于 AdvMask 的几个关键超参数，连续区域的边长 d 的最大值 d_h 和最小值 d_l 分别设置为 20 和 2，面积阈值 S_h 设置为图像大小的 0.03 倍，重叠阈值 γ_{cover} 设置为连续区域面积 d^2 的 0.1 倍。

表 4-1: CIFAR-10 上应用不同数据增强方法得到的模型准确率

Model	Base	Cutout	RE	HaS	GridMask	AdvMask
ResNet-18	95.28	96.01	95.69	96.10	96.34	**96.44**
ResNet-44	94.10	94.78	94.87	94.97	95.02	**95.49**
ResNet-50	95.66	95.81	95.82	95.94	96.15	**96.69**
WideResNet-28-10	95.52	96.92	96.92	96.94	**97.23**	97.03
ShakeShake-26-32	94.90	96.96	96.46	96.89	96.91	**97.02**

表 4-1展示了 CIFAR-10 上应用不同数据增强方法，在不同网络结构上得到的模型准确率。需要注意的是，为了避免随机数的影响，表格中的结果都是多次实验结果的平均值。可以看到，我们提出的 AdvMask 在 ResNet-18、ResNet-44、ResNet-50、WideResNet-28-10 和 ShakeShake-26-32 上的准确率相对 Base 方法分别增加了 1.16%、1.39%、1.03%、1.51% 和 2.12%。与其他数据增强方法相比，AdvMask 在 ResNet-18、ResNet-44、ResNet-50 和 ShakeShake-26-32 取得了最优的结果，在 WideResNet-28-10 上取得了次优的结果。

除此以外，我们还在 CIFAR-100 数据集上进行了实验。CIFAR-100 数据集与 CIFAR-10 数据集基本一样，不同的是 CIFAR-100 将类别个数从 10 个扩充到了 100 个，增加了分类的难度。因为图像的尺寸都是 $32 \times 32 \times 3$，因此我们保持实验设置和实验参数不变。表 4-2展示了 CIFAR-100 上应用不同数据增强方法，在不同网络结构上得到的模型准确率，表格中的结果也都是多次实验的平均值。可以看到，我们提出的 AdvMask 在 ResNet-18、ResNet-44、ResNet-50、WideResNet-28-10 和 ShakeShake-26-32 上的准确率相对 Base 方法分别增加了 0.75%、1.64%、1.58%、1.74% 和 3.31%。与其他数据增强方法相比，AdvMask 在 WideResNet-28-10 上与 RE 取得了同样好的结果，在其他 4 个网络结构上取得了最优的结果，这说明本方法具有一定的优越性。

4.3.1.2 Tiny-ImageNet

Tiny-ImageNet 数据集有 200 个类，其中训练集有 100000 张图像，验证集有 10000 张图像。每张图像的尺寸我们都统一缩放为 $64 \times 64 \times 3$。与 4.3.1.1节的实验一样，这里我们也与其他数据增强方法使用同样的基础空域变换，包括随机填充剪裁和随机水平翻转。具体的，我们在使用 AdvMask 进行数据增强

表 4-2: CIFAR-100 上应用不同数据增强方法得到的模型准确率

Model	Base	Cutout	RE	HaS	GridMask	AdvMask
ResNet-18	77.54	78.04	75.97	78.19	75.23	**78.29**
ResNet-44	74.80	74.84	75.71	75.82	76.07	**76.44**
ResNet-50	77.41	78.62	77.79	78.76	78.38	**78.99**
WideResNet-28-10	78.96	79.84	80.70	80.22	80.40	**80.70**
ShakeShake-26-32	76.65	77.37	77.30	76.89	77.28	**79.96**

后，还将图像尺寸填充为 $68 \times 68 \times 3$，然后随机剪裁一个 $64 \times 64 \times 3$ 的区域作为新的图像，并紧接着进行概率为 0.5 的随机水平翻转。

为了实验的充分性，我们在 3 个网络结构 ResNet-18、ResNet-50 以及 WideResNet-50-2 上进行了实验。具体的，我们使用了 ImageNet 数据集上的预训练权重，并在 Tiny-ImageNet 上微调 30 轮。优化方法使用带动量的梯度下降法，其中动量参数为 0.9。在训练过程中，学习率初值为 0.001，每过 7 轮下调 10 倍。关于 AdvMask 的几个关键超参数，连续区域的边长 d 的最大值 d_h 和最小值 d_l 分别设置为 36 和 5，面积阈值 S_h 设置为图像大小的 0.03 倍，重叠阈值 γ_{cover} 设置为连续区域面积 d^2 的 0.1 倍。需要注意的是，由于 Tiny-ImageNet 数据集的特性，我们在实验中发现，现有的方法如果都按照 GridMask 的训练策略，每一轮训练中都有一定的概率不进行数据增强而是保持原数据，都能够取得更好的结果。因此表格中的方法都添加了此策略以取得更好的效果。

表4-3展示了 Tiny-ImageNet 上应用不同数据增强方法，在不同网络结构上得到的模型准确率。这里为了避免随机数的影响，表格中的结果都是多次实验结果的平均值。可以看到，我们提出的 AdvMask 在 ResNet-18、ResNet-50 和 WideResNet-50-2 上的准确率相对 Base 方法分别增加了 3.12%、6.86% 和 1.30%。与其他数据增强方法相比，AdvMask 在这三个模型上都取得了最优的结果，这足以说明本方法的优越性。

可以看到，我们的方法在 Tiny-ImageNet 数据集上的提升比 CIFAR-10/100 数据集上的提升要大。这是因为 Tiny-ImageNet 图像的尺寸更大、主体对象更清晰，方法第一步的稀疏对抗攻击可以更准确地找到那些最敏感的像素点。在实验过程中我们也验证了这一点，我们发现 CIFAR-10 数据集上的敏感像素占 16% 左右，而 Tiny-ImageNet 数据集上的敏感像素只占 6% 左右。由于我们方法

表 4-3: Tiny-ImageNet 上应用不同数据增强方法得到的模型准确率

Model	Base	Cutout	RE	HaS	GridMask	AdvMask
ResNet-18	62.00	63.59	63.80	63.61	63.50	**65.12**
ResNet-50	73.34	77.86	75.08	74.94	77.38	**80.20**
WideResNet-50-2	81.55	81.77	81.89	81.84	81.79	**82.85**

第二步是基于敏感像素区域来删除，因此相较于现有的全图性随机删除的数据增强方法，越少和越准确的敏感像素所能体现的区别就越明显。总之，我们的方法更适用于图像尺寸较大的数据集。

4.3.2 消融实验

前文的实验说明了 AdvMask 与其他数据增强方法相比具有优越性，本节将进一步研究分析 AdvMask 先后两个主要模块的作用以及必要性。如4.2.1节所述，AdvMask 可以看作是包含了两个主要的模块，第一个是利用稀疏对抗攻击 AutoAdversary 来产生扰动区域 mask，第二个是在这个 mask 的基础上生成有一定随机性的连续区域，在本节实验中我们分别将这两个模块表示为 “Adv” 和 “Mask”。除此以外，为了体现这两个模块各自的有用性和必要性，我们还需要引入其他的对照方法，下面用 “Random “表示在全图中随机选择像素点，而不是在扰动区域 mask 中选择敏感像素点；用 “Corner “表示用图像角点检测[70]来得到图像的关键点 mask，用来代替稀疏对抗攻击产生的 mask。

表4-4展示了在 CIFAR-10 数据集上的消融实验结果。“Adv” 表示使用稀疏对抗攻击 AutoAdversary 产生扰动区域 mask，并且直接将此 mask 作为删除区域 mask，而不会像4.2.3节一样做进一步处理。可以看到 “Adv” 在 ResNet-18 和 ResNet-50 上取得的模型准确率分别仅为 92.96% 和 94.52%，这说明直接利用稀疏对抗攻击产生的扰动区域 mask 是无法取得较好的增强效果的，因此说明了我们提出的 AdvMask 算法的第二部分是有作用且有必要的。

“Random + Mask” 表示不在 AutoAdversary 产生扰动区域 mask 中选择像素点，而是在整张图像上随机选点，也就是说敏感像素集合 $\mathcal{P}$ 中包含了全部像素的坐标，其他步骤与算法4.1保持一致。可以看到 “Random + Mask” 在 ResNet-18 和 ResNet-50 上取得的模型准确率分别仅为 95.66% 和 95.61%。与之相对应的，“Corner + Mask” 也不使用稀疏对抗攻击，而是在由图像角点检测

表 4-4: CIFAR-10 上有关 AdvMask 的消融实验

Model	Method	Accuracy(%)
ResNet-18	Base	95.28
	Adv	92.96
	Random + Mask	95.66
	Corner + Mask	95.11
	Adv + Mask	**96.44**
ResNet-50	Base	95.66
	Adv	94.52
	Random + Mask	95.61
	Corner + Mask	95.41
	Adv + Mask	**96.69**

得到的 mask 中选择像素点，也即是敏感像素集合 $\mathcal{P}$ 中是图像中的角点坐标，其他步骤同样与算法 4.1保持一致。可以看到 “Corner + Mask” 在 ResNet-18 和 ResNet-50 上取得的模型准确率分别仅为 95.11% 和 95.41%。这两个对照结果表明了 AdvMask 的第一部分稀疏对抗攻击是有作用的，所得到的扰动区域 mask 并不是简单的随机像素点，确实可以作为敏感区域；并且，简单使用角点检测并不能代替稀疏对抗攻击，因此反映了 AdvMask 所找到的敏感区域并非只考虑到了图像的主体对象或者关键像素点，更多的其实是从模型的弱点和缺陷出发，找到最敏感的区域和特征。

“Adv + Mask” 就是表示我们提出的 AdvMask 的完整结构，相较于其他对照该方法取得了最优的模型准确率。因此本消融实验证实了 AdvMask 先后两个主要模块的作用和必要性。

4.3.3 增强效果可视化

为了分析使用 AdvMask 训练的模型到底学习到了什么，我们在本节使用类激活图（Class Activation Map，CAM[71]）来可视化理解模型的行为。具体的，我们在 CIFAR-100 数据集上分别用不同的增强方法训练了 ResNet-18 网络，并计算各自模型的类激活图。

结果如图 4-2所示。其中第一行表示原始输入图像，后续各行分别表示基

于不同数据增强方法训练得到的模型所产生的类激活图。观察后容易发现，相对于 Base 模型，其他的模型都能够关注图像中更大的区域和更多的信息，这表明成功的数据增强能够迫使模型将注意力放在大而显著的表示上，进而提高泛化能力。更重要的是，图 4-2表明用 AdvMask 训练的模型所关注的信息更加全面。比如，第 3 张图像中包含了两个瓶子，其他大多数模型都倾向于更多地关注左边或者右边某一个瓶子，而用我们提出的 AdvMask 训练的模型能够同时注意到这两个瓶子，因此能够获得更准确的信息。

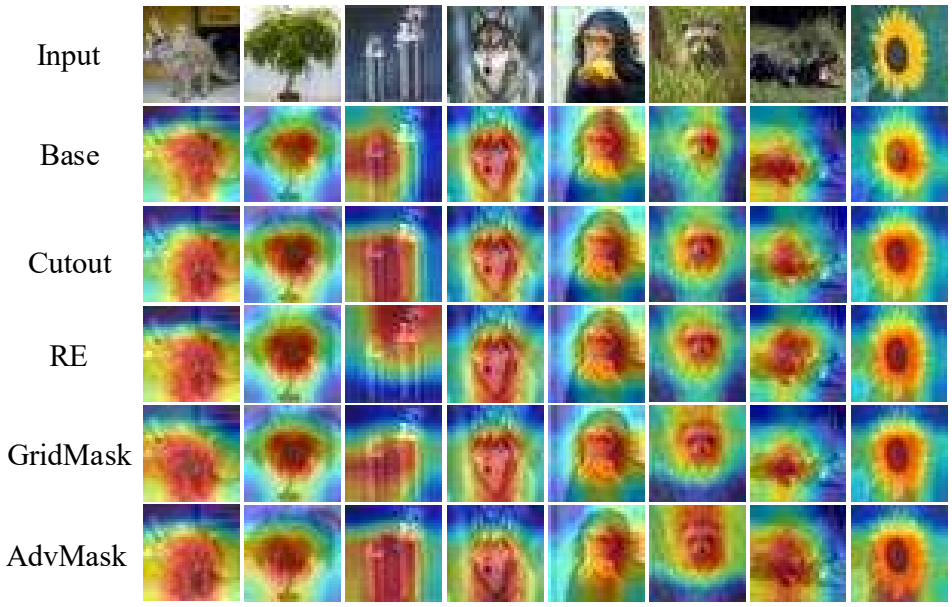


图 4-2: CIFAR-100 数据集上 ResNet-18 模型的类激活图（CAM）可视化

4.4 本章小结

本章首先简述了图像数据增强的基本背景，并整理总结了现有的基于信息删除的数据增强方法的特点和缺陷，进而在第 3 章的稀疏对抗攻击 AutoAdversary 的基础上，提出了一个基于稀疏对抗攻击的数据增强方法 AdvMask。具体的，AdvMask 首先利用拓展的无目标稀疏对抗攻击来找到图像中最敏感的区域，然后基于这些敏感区域再生成需要删除的区域。与其他数据增强相比，AdvMask 为每张图像单独定制了个性化的删除区域，减少了因为全图随机性带来的低质量样本，从而能够更加有效地进行数据增强。CIFAR-10、CIFAR-100 和 Tiny-ImageNet 数据集以及多种网络结构上的实验结果说明了本

方法的优越性。本章最后还进行了消融实验，分析了 AdvMask 先后两个主要模块的有用性和必要性，结果显示将稀疏对抗攻击替换为更直观的随机选点以及角点检测等，将无法取得较好的增强效果，这进一步说明了 AdvMask 利用稀疏对抗攻击找到了图像中最敏感的区域，更能够针对模型的缺陷来进行数据增强，鼓励模型更多关注那些不太敏感的重要特征。

第五章　基于稀疏对抗攻击的数据增强在花卉识别系统中的应用

数据增强作为一个训练神经网络模型的常用技术，在很多实际的场景中都有应用。为了检验本文所提算法在实际场景中的有效性，我们搭建了一个花卉识别系统，并将本文提出的基于稀疏对抗攻击的数据增强算法 AdvMask 应用其中。下文将对围绕该系统的相关背景、主要功能、整体架构以及实际效果等方面进行详细介绍。最终系统的识别效果强有力地证明了我们算法的有效性以及实用性。

5.1　相关背景

随着现代社会的发展，物质生活条件逐渐得到改善，越来越多的人开始追求精神层面和艺术层面的需求。其中花作为观赏价值极高的植物，一直都被人们所喜爱，越来越多的人也开始赏花、栽培花或者从事有关花的商业活动，各种花卉丰富着人们的生产和生活。花的种类繁多，每种花不仅包含各自的生物特征，还有人们所赋予的抽象含义。那么一个易于使用的花卉识别系统，首先可以作为科普工具，帮助普通人随时随地快速了解眼前所见花卉的种类、花语含义以及价值，其次也能够作为科研人员在野外科考时的简易工具，比如通过花卉生长环境反推当地的气候条件，因此花卉识别系统具有一定的科普价值和实用价值。

一个性能优秀的花卉识别系统需要兼具较高的准确性、较强的实时性和较低的成本。花的种类繁多，并且相近品种的花卉在外表特征上可能非常相似，一般训练得到的深度学习模型可能会过于关注图像的局部特征而忽略全局上下文信息，从而误识别那些相似的花，导致识别系统的准确率不够高。这里我们使用了提出的 AdvMask 数据增强方法，在训练过程中删除部分信息，帮助模型

提升泛化能力。并且本方法仅在模型的训练阶段增加计算资源的消耗，因此不会影响模型推断过程的速度，也不会增加其他成本。

5.2 系统需求

花卉识别系统的主要目的有两个，其一是作为在线实时识别花卉的工具，用户通过手机拍照或者上传照片能够实时得到识别结果；其二是作为科普类的应用，为用户查询和学习花卉知识提供一个平台和入口。为了实现上述目的，我们设计的花卉识别系统具有以下需求：

(1) 简易方便的使用条件：考虑到用户进行花卉识别时大都处于即兴的情况，比如用户偶然看到一簇花并且想立即了解其名称和类别，因此承载本系统的平台应该具备简易和方便使用的条件，因此本系统最好应该搭建在移动端而不是 Web 网页端。

(2) 实时准确的花卉识别：本系统的主要功能是进行花卉识别，用户通过在手机上拍照或者上传照片，能够实时得到准确的结果，并且结果应当包含名称、种类、花语以及价值等详尽的信息。

(3) 识别结果反馈：用户可以对识别结果进行反馈从而进一步帮助改进模型。比如在给出识别结果时，系统同时给出该类别花卉的其他图片，用户可以结合图片对照来判断识别结果是否准确，并且进行反馈。如果反馈结果中有识别错误，则可以将数据收集起来进一步改善模型。

(4) 丰富的花卉知识可供查询：本系统作为一个带有科普性质的应用，应当可以作为用户查询花卉知识的平台。因此本系统应当包含详尽的花卉知识，并且适时地为用户推送花卉科普文章或者植物学论文研究报告。

(5) 富含趣味性的花卉知识学习：除了相对严肃的科普和研究报告，本系统应当具备一定的趣味性，比如通过每日趣味问答等方式，在互动中促进用户对花卉知识的学习。

5.3 系统架构

前文已经提到了，为了便于用户及时使用，本系统应该搭建在移动端。然而移动端的计算能力不足，无法通过深度学习模型实时产生识别结果。因此本系统采用客户端/服务器架构，用户在客户端与系统进行交互，而计算和存储则

由服务器负责。

具体来讲，如图5-1所示，从客户端到服务器可以分为页面展示层、业务逻辑层和数据存储层。页面展示层是指用户直接能够看到的前端页面，用户将在这一层与系统进行交互，主要包含了：拍照识别界面、识别结果展示、文章推送界面、文章内容展示和趣味答题界面；业务逻辑层主要负责数据的使用以及系统基本功能的实现，主要包括：花卉信息整理、图像数据预处理、分类模型的训练以及模型推断等；数据存储层负责保存整个系统的数据，主要包括：分类模型的参数、花卉信息如图像和文字介绍等。具体的，页面展示层与业务逻辑层之间通过 HTTP 网络接口进行通信和信息交换，业务逻辑层与数据存储层之间通过关系型数据库以及文件系统实现花卉数据存取和模型参数的存取。

5.4 系统实现

基于5.3节的系统架构设计，本系统使用微信小程序作为客户端，用 Python 来开发服务端代码，并使用 Django 作为 Web 框架，另外数据存储层使用 MySQL 关系型数据库。

5.4.1 整体业务流程

本系统的主体业务流程如图5-2所示。首先需要对花卉数据进行整理和收集，包括大规模的花卉图像和标签数据，以及每种花卉的附带信息如生物特征、花语内涵以及商业价值等；然后我们会对收集来的花卉数据进行整理和存储，主要是将花卉的基本信息录入到关系型数据库中，并将花卉的图像存储在文件系统中；接下来是花卉识别模型的训练，这里主要用到了我们在前文提出的 AdvMask 数据增强方法，训练完成后进一步将模型参数存储在文件系统中；后续的流程主要是由用户发起识别请求，比如拍照或者上传图像，然后系统接受到图像后调用模型进行推断并给出最终的识别结果；如果用户对识别结果的正确性存疑，系统将保存该图像，由专人标注类别后加入到数据集中，可以在未来进一步用于训练以提升模型的准确性。

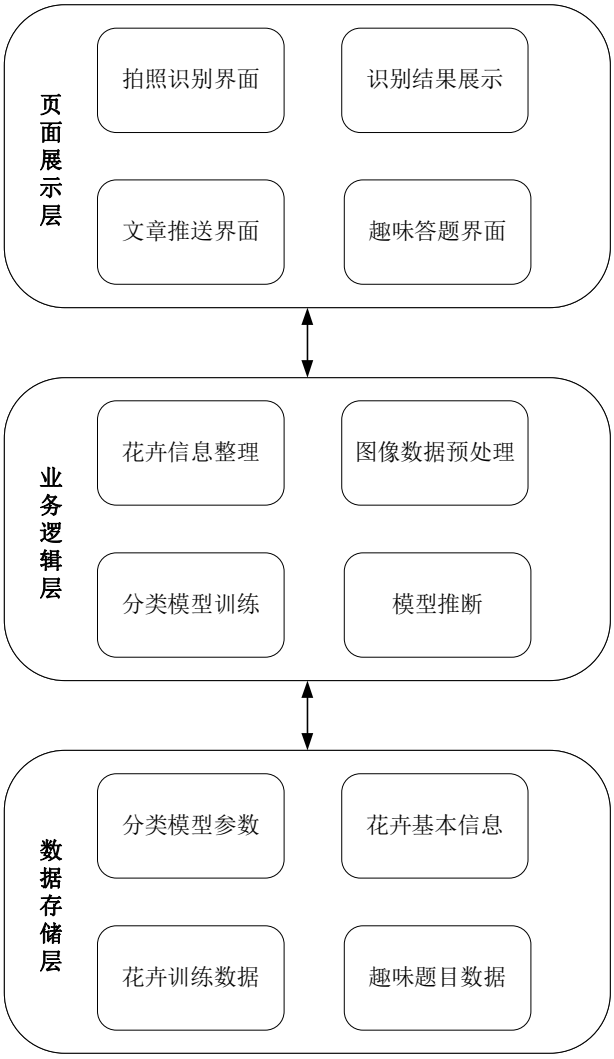


图 5-1: 系统三层架构图

5.4.2 关键模块的实现

前文对本系统的整体架构以及主要业务流程进行了介绍，下面将对其中的关键模块进行更加详细的介绍。

5.4.2.1 数据处理与存取

大多数以深度学习模型为核心的系统，数据都是重中之重。本系统的花卉数据主要来源于公开花卉数据集[72]，该数据集包含了 102 种不同类别的花卉，每个类别包含 40 到 258 张 RGB 彩色图像，总图像数量为 8192 张。特别的，

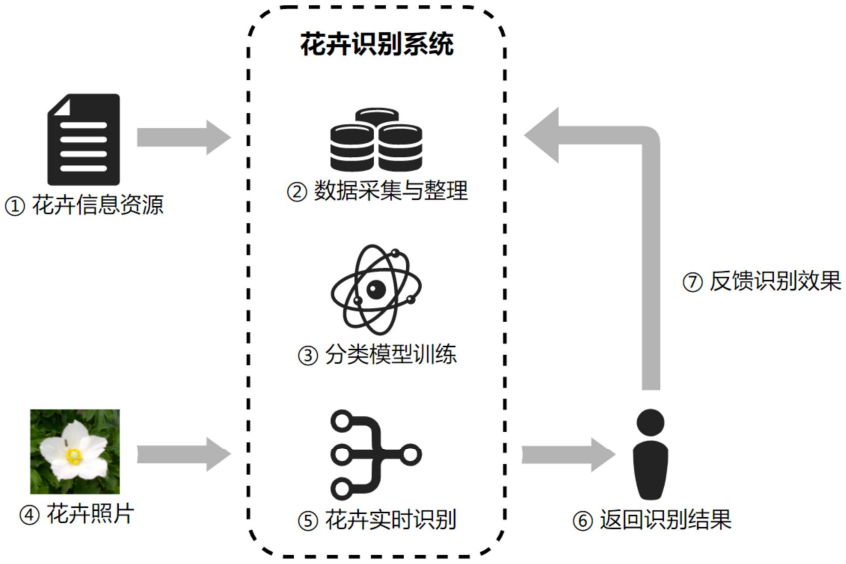


图 5-2: 主体业务流程图

这些图像中具有大量的角度和光线变化，使得一些同类的花卉有较大的差异，并且还有一些非常相近的花卉品种。总之，该数据集质量较高且基本涵盖了常见的花卉类别。有一点需要注意的是，该数据集中的图像尺寸不一，因此我们首先将其尺寸统一缩放为 $224 \times 224 \times 3$。这个尺寸既能够保留大部分的细节信息，又能不至于使得模型的训练和推断的速度太慢，能够保证实时预测。

除了基本的图像和分类信息以外，为了丰富用户的体验以及起到科普的作用，本系统还需要收集每种花卉的附带信息，主要包括：生物特征、花语内涵和商业价值等。以上信息主要靠人工通过查阅书籍以及网络资料获取。除此以外，为了进一步提升某些用户关于花卉专业知识的诉求，还需要收集公开的植物学论文研究以及花卉科普文章链接。

我们利用 MySQL 关系型数据库和文件系统来存储以上信息。由于本系统的功能相对简单，相关操作也较少，因此只需要存储图像信息这一张数据库表即可。该数据库表存储了每一张图像的基本信息包括：图像 ID、图中花卉类别、生物特征、花语内涵、相关文章链接，并以图像 ID 作为唯一标识符。在文件系统中存储了具体的图像文件，每一张图像都以上述数据库表中的图像 ID 命名，因此数据库中的图像信息与这里的图像文件一一对应。通过这样的设计，无论是在训练模型时取得图像，还是在其他需要取得某一类图像的功能中都比较方便。

5.4.2.2 模型训练

完成花卉数据资源的整理后，需要先完成识别模型的训练才能够为用户提供服务。首先对于上述的公开花卉数据集，我们以 6:1:1 的比例将其分为训练集、验证集和测试集，并保持 3 个集合内各个类别的图像数量保持相近，以避免类别不平衡的问题。由于需要为用户提供实时的服务，因此模型容量不能过大，因此本系统选择 ResNet-50 作为分类网络，该网络结构的参数量为 2.56×10^7，既能够拥有较好的分类结果又能够实时地进行推断。模型的训练方式参照本文第 4 章提出的算法4.1，主要就是利用 AdvMask 对花卉数据集进行增强，以提升模型的泛化能力。

具体的，我们使用 PyTorch 提供的在 ImageNet 上预训练好的 ResNet-50 模型作为初始模型，并紧接着在我们的花卉数据集上进行微调。我们首先将每张花卉图像的尺寸缩放为 $224 \times 224 \times 3$，然后将其进行归一化和标准化以便得到更好的训练效果，具体的标准化参数是从整个数据集上计算得到的：

$$\begin{aligned} \boldsymbol{\mu}_{\text{Flower}} &= [0.4752, 0.3941, 0.3080], \\ \boldsymbol{\sigma}_{\text{Flower}} &= [0.2660, 0.2115, 0.2201], \end{aligned} \tag{5-1}$$

其中 $\boldsymbol{\mu}_{\text{Flower}}$ 和 $\boldsymbol{\sigma}_{\text{Flower}}$ 分别表示花卉数据集中 RGB 图像 3 通道的均值和标准差。然后我们使用带动量的梯度下降法来训练模型，其中动量衰减参数设置为 0.9，权重衰减参数为 5×10^{-4}，批量大小为 64，最大轮数为 300，学习率初值为 0.1 并在 100、200 以及 265 轮时衰减 5 倍。关于 AdvMask 的参数，连续区域的边长 d 的最大值 d_h 和最小值 d_l 分别设置为 60 和 20，面积阈值 S_h 设置为图像大小的 0.1 倍，重叠阈值 γ_{cover} 设置为连续区域面积 d^2 的 0.1 倍。并且每一轮在进行数据增强时，我们也会融合其他基础的增强方法如：随机裁剪和随机水平翻转。训练过程中，我们把验证集上表现最好的模型保存下来作为最终的模型，并将该模型在测试集上的准确率作为最终的准确率。

5.4.2.3 花卉识别与反馈

当模型训练完成并成功部署后，就可以为用户提供识别服务了。首先由用户在微信小程序界面发起请求，通过临时拍照或者相册上传的方式将图像发送给服务器。服务器在接受到图像之后，首先进行图像预处理将图像尺寸缩放为 $224 \times 224 \times 3$，然后进行归一化和标准化，标准化参数与式 (5-1) 保持一致。然

后调用模型进行推断，得到模型预测出来的花卉类别，并将其反馈给用户。

与此同时，这里为了进一步提升用户的体验，本系统在返回识别结果时还会将同类别花卉的其他图像一并返回给用户。用户在查看识别结果时，可以将这些图像与之前上传的图像进行比较，以帮助用户确认系统识别是否有误。如果用户反馈识别有误，则系统将收集上传的图像，后续由专人标注后一并加入到训练集中，重新训练模型以提升准确性。

5.5 效果展示

5.5.1 数据增强效果

表5-1展示了本系统使用不同数据增强方法后模型在训练集、验证集和测试集上的准确率。我们在训练集上训练模型，同时保留验证集上准确率最高的模型，最终计算该模型在测试集上的准确率。可以看到基本的 Base 方法训练得到的模型在训练集上的准确率达到了 99.98%，但在测试集上的准确率仅为 80.20%，这说明其严重过拟合。而使用现有的数据增强方法后均能一定程度上降低过拟合。特别的，使用我们提出的 AdvMask 数据增强方法后模型在测试集上的准确率最高，为 91.57%。此对比结果说明了我们的方法在实际场景中能够降低过拟合，有效提升模型泛化性。

除此以外，与4.3.3节类似，我们使用类激活图来可视化理解模型的行为，结果如图5-3所示。其中第一行表示原始输入图像，后续各行分别表示基于不同数据增强方法训练得到的模型所产生的类激活图。观察后容易发现，相较于使用其他数据增强方法，使用 AdvMask 训练的模型能够关注图像中更加全面的信息。比如，第 3 张图像中包含了两朵球蓟花，其他数模型都倾向于更多地关右边这一朵，而用我们提出的 AdvMask 训练的模型能够同时注意到这两朵，因此能够获得更准确的信息。

5.5.2 其他基础功能

本节将展示本系统的其他基础功能，下面的界面图都来自于微信小程序开发工具中手机运行模拟器的截图。用户在进入本系统后，首先看到的页面如图5-4(a)所示，用户可以点击页面中央的“拍照识别”或者“从相册上传”按钮，来将花卉图像上传到服务器，服务器调用模型完成识别以后将会把结果反

表 5-1: 不同数据增强方法在花卉识别系统中的对比

Method	trainAcc(%)	valAcc(%)	testAcc(%)
Base	99.98	83.92	80.20
Cutout	97.98	87.75	88.53
HaS	92.52	85.98	84.12
RE	97.73	88.73	88.43
GridMask	97.95	90.69	90.19
AdvMask (ours)	98.95	90.78	**91.57**

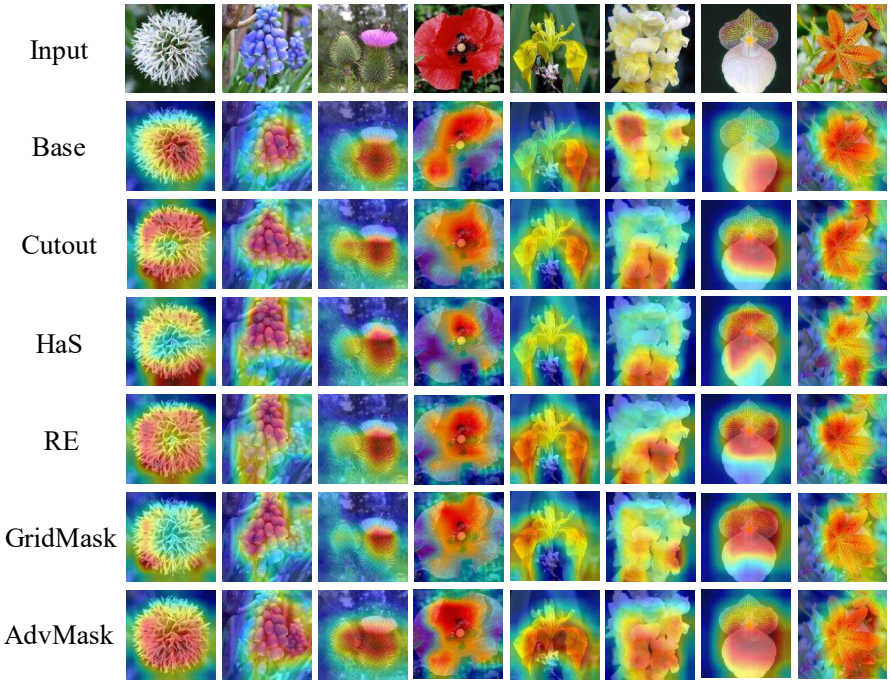
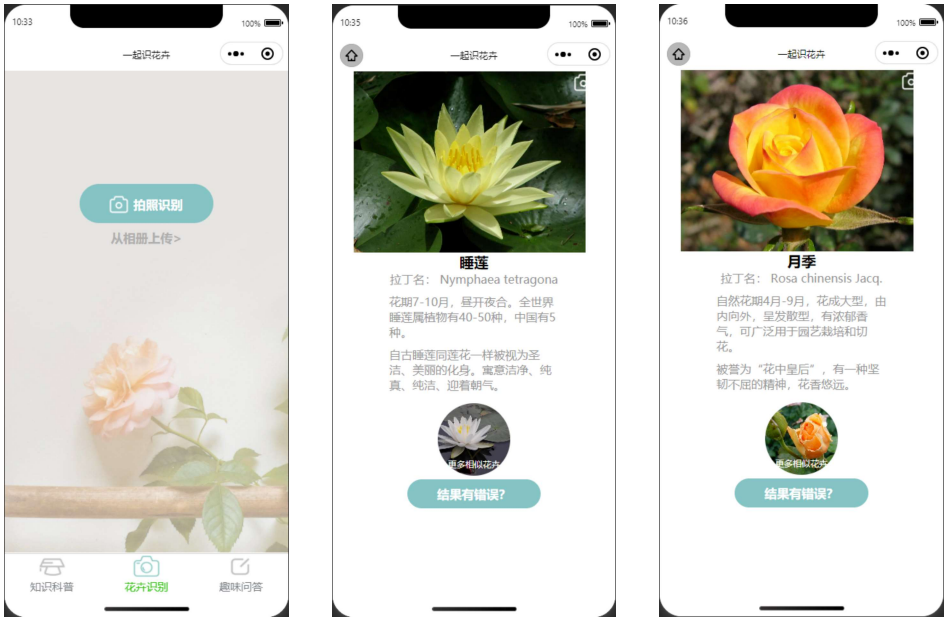


图 5-3: 花卉识别系统上 ResNet-50 模型的类激活图（CAM）可视化

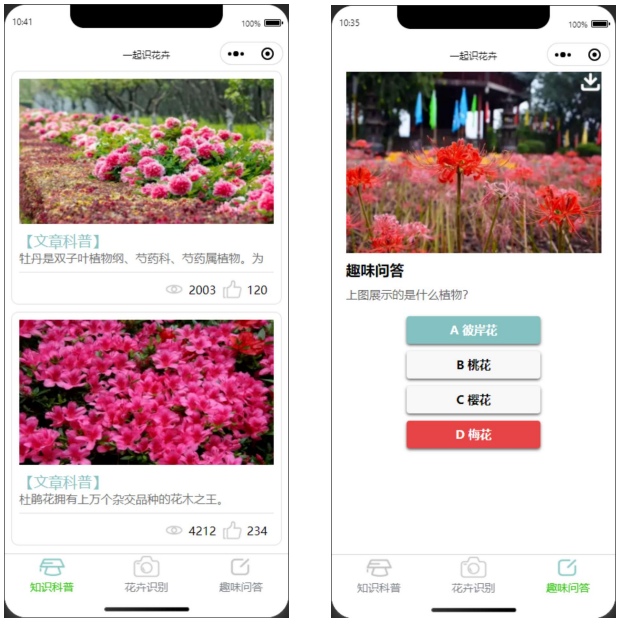
馈给用户。图 5-4(b)和图 5-4(c)分别展示了两张不同的花卉图像的识别结果，可以看到识别结果包括了花卉的类别和该花卉的附加信息。本系统在返回识别结果时会将同类型的花卉图像也返回给用户，让用户可以更加方便地判断识别结果是否可信。如果用户认为识别有误，可以点击页面下方的“结果有错误？”按钮，系统后续将进行进一步的处理。除了识别花卉以外，本系统作为科普类的应用也为用户提供了学习的平台。如图 5-5(a)所示，系统为用户准备了科普性质的文章以及研究报告，用户可以点击学习。此外如图 5-5(b)所示，

系统将会发布趣味问题以供用户答题，提升花卉知识学习的趣味性。



(a) 主界面 (b) 识别结果——睡莲 (c) 识别结果——月季

图 5-4: 花卉识别系统——识别功能展示



(a) 科普文章 (b) 趣味问答

图 5-5: 花卉识别系统——科普功能展示

5.6 本章小结

本章介绍了我们搭建的花卉识别系统。该系统作为在线实时识别花卉种类的工具，用户通过手机拍照能够方便快捷地得到识别结果。同时该系统还作为科普类的应用，为用户学习花卉知识提供一个平台和入口，具有一定的实用价值和科普价值。我们将本文所提的基于稀疏对抗攻击的数据增强算法 AdvMask 应用其中，提升了识别模型的泛化能力，构建了一个准确高效的花卉识别系统。总的来说，实践表明我们所提出的算法能提供良好的数据增强效果，充分体现了本算法的实际应用价值。

第六章　总结与展望

本文主要研究白盒场景下图像分类任务中的稀疏对抗攻击方法，并挖掘了其在模型训练中的正面应用。在稀疏对抗攻击的研究中，我们跳出了人为设计像素点重要性指标和贪心修改像素值的通用过程，使得哪些像素点被扰动完全由模型自动决定，从而提升了对抗扰动的稀疏性。此外，我们进一步发掘了稀疏对抗攻击的正面应用，将所提方法结合到数据增强中，通过迫使模型关注那些不敏感但是非常重要的区域，大大提升了模型的泛化性。本文的主要研究内容与贡献如下：

- 本文从神经网络剪枝任务中获得灵感，创新性地将自动剪枝技术与对抗攻击技术相结合，提出了基于像素自动删减的稀疏对抗攻击方法，简称AutoAdversary。本方法在一般的对抗攻击过程中添加一个包含可训练的编码器网络和近似二值化模块的分支，用于生成01-mask，进而在构建对抗样本的过程中利用mask来自动确定哪些像素点需要被扰动。本方法跳出了人为设计像素点重要性指标和贪心修改像素值的通用过程，使得哪些像素点被扰动完全由模型自动决定，大大提升了对抗扰动的稀疏性。经实验验证，本方法在多个数据集上都达到了最优的效果。除此以外，本方法具备灵活性和通用性，在大多数一般对抗攻击方法的基础上都能进一步减小对抗扰动，因此可以视作通用的扰动删减框架。
- 本文进一步发掘了稀疏对抗攻击在模型训练中的正面应用，提出了一种基于稀疏对抗攻击的图像数据增强方法，简称AdvMask。我们首先利用AutoAdversary来攻击分类模型，进而发现图像中最为敏感的像素点，通过在训练中按照一定的策略适当删除敏感像素周围的随机区域，迫使模型关注不敏感的重要特征以及全图上下文信息，大大提升了模型的泛化性。经实验验证，相较于现有的数据增强方法，AdvMask在多个数据集和多个网络结构上都取得了更好的效果。
- 本文将所提方法应用于实际的环境中，搭建了一个准确率高、实时性强的在线花卉识别系统。相较于使用其他图像数据增强方法，模型的泛化性得到了进一步的提升，满足了实际系统的需求，也佐证了我们所提算法的实

用性以及有较大的实际应用价值。

在本文工作的基础上，还可以继续进行相关的研究工作。对于第 3 章提出的稀疏对抗攻击方法 AutoAdversary，首先可以探究其中编码器网络结构对攻击性和稀疏性的影响，从而设计更好的网络结构来取得更好的性能。其次，由于本方法目前针对每一张图像都有一个编码器模型与之对应，编码器的训练与对图像的攻击同时进行，所以无法批量化地对图像进行攻击导致速度相对偏慢，如果能进一步优化算法的流程或者整体结构使得能够批量化地进行稀疏对抗攻击，将会大大提升本方法的实用性。本文第 4 章提出的基于稀疏对抗攻击的数据增强方法 AdvMask 不仅可以在图像识别任务中使用，也可以进一步推广到目标检测、实例分割、人体姿态识别等视觉任务中，为这些任务带来更好的数据增强效果。

参考文献

[1] LUO J-H, WU J. Autopruner: An end-to-end trainable filter pruning method for efficient deep model inference[J]. Pattern Recognition, 2020, 107 : 107461.

[2] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. Imagenet classification with deep convolutional neural networks[J]. Advances in neural information processing systems, 2012, 25.

[3] SZEGEDY C, LIU W, JIA Y, et al. Going deeper with convolutions[C] // Proceedings of the IEEE conference on computer vision and pattern recognition. 2015 : 1 – 9.

[4] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.

[5] MIKOLOV T, KARAFIÁT M, BURGET L, et al. Recurrent neural network based language model.[C] // Interspeech : Vol 2. 2010 : 1045 – 1048.

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

[7] GUO H, TANG R, YE Y, et al. DeepFM: a factorization-machine based neural network for CTR prediction[J]. arXiv preprint arXiv:1703.04247, 2017.

[8] NAJAFABADI M M, VILLANUSTRE F, KHOSHGOFTAAR T M, et al. Deep learning applications and challenges in big data analytics[J]. Journal of big data, 2015, 2(1) : 1 – 21.

[9] WANG Y, CHAO W-L, GARG D, et al. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving[C] // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019 : 8445 – 8453.

[10] LI P, CHEN X, SHEN S. Stereo r-cnn based 3d object detection for autonomous driving[C] // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019 : 7644 – 7652.

[11] DENG J, GUO J, XUE N, et al. Arcface: Additive angular margin loss for deep face recognition[C] // Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019 : 4690 – 4699.

[12] DAHL G E, STOKES J W, DENG L, et al. Large-scale malware classification using random projections and neural networks[C] // 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. 2013 : 3422 – 3426.

[13] SAXE J, BERLIN K. Deep neural network based malware detection using two dimensional binary program features[C] // 2015 10th international conference on malicious and unwanted software (MALWARE). 2015 : 11 – 20.

[14] JAGIELSKI M, OPREA A, BIGGIO B, et al. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning[C] // 2018 IEEE Symposium on Security and Privacy (SP). 2018 : 19 – 35.

[15] GU T, DOLAN-GAVITT B, GARG S. Badnets: Identifying vulnerabilities in the machine learning model supply chain[J]. arXiv preprint arXiv:1708.06733, 2017.

[16] TRAMÈR F, ZHANG F, JUELS A, et al. Stealing Machine Learning Models via Prediction {APIs}[C] // 25th USENIX security symposium (USENIX Security 16). 2016 : 601 – 618.

[17] SZEGEDY C, ZAREMBA W, SUTSKEVER I, et al. Intriguing properties of neural networks[J]. arXiv preprint arXiv:1312.6199, 2013.

[18] KOS J, SONG D. Delving into adversarial attacks on deep policies[J]. arXiv preprint arXiv:1705.06452, 2017.

[19] XIE C, WANG J, ZHANG Z, et al. Adversarial examples for semantic segmentation and object detection[C] // Proceedings of the IEEE international conference on computer vision. 2017 : 1369 – 1378.

[20] JIA R, LIANG P. Adversarial examples for evaluating reading comprehension systems[J]. arXiv preprint arXiv:1707.07328, 2017.

[21] RAJPURKAR P, ZHANG J, LOPYREV K, et al. Squad: 100,000+ questions for machine comprehension of text[J]. arXiv preprint arXiv:1606.05250, 2016.

[22] EYKHOLT K, EVTIMOV I, FERNANDES E, et al. Robust physical-world attacks on deep learning visual classification[C] // Proceedings of the IEEE conference on computer vision and pattern recognition. 2018 : 1625 – 1634.

[23] KOMKOV S, PETIUSHKO A. Advhat: Real-world adversarial attack on arcface face id system[C] // 2020 25th International Conference on Pattern Recognition (ICPR). 2021 : 819 – 826.

[24] MADRY A, MAKELOV A, SCHMIDT L, et al. Towards Deep Learning Models Resistant to Adversarial Attacks[C] // International Conference on Learning Representations. 2018.

[25] ZHU J, KAPLAN R, JOHNSON J, et al. Hidden: Hiding data with deep networks[C] // Proceedings of the European conference on computer vision (ECCV). 2018 : 657 – 672.

[26] 陈岳峰, 毛潇锋, 李裕宏, et al. AI 安全——对抗样本技术综述与应用 [J]. 信息安全研究, 2019, 5(11) : 1000.

[27] XIE C, TAN M, GONG B, et al. Adversarial examples improve image recognition[C] // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020 : 819 – 828.

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

[29] KURAKIN A, GOODFELLOW I J, BENGIO S. Adversarial examples in the physical world[G] // Artificial intelligence safety and security. [S.l.] : Chapman and Hall/CRC, 2018 : 99 – 112.

[30] DONG Y, LIAO F, PANG T, et al. Boosting adversarial attacks with momentum[C] // Proceedings of the IEEE conference on computer vision and pattern recognition. 2018 : 9185 – 9193.

[31] CARLINI N, WAGNER D. Towards evaluating the robustness of neural networks[C] // 2017 ieee symposium on security and privacy (sp). 2017 : 39 – 57.

[32] CHEN P-Y, ZHANG H, SHARMA Y, et al. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models[C] // Proceedings of the 10th ACM workshop on artificial intelligence and security. 2017 : 15 – 26.

[33] TU C-C, TING P, CHEN P-Y, et al. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks[C] // Proceedings of the AAAI Conference on Artificial Intelligence : Vol 33. 2019 : 742 – 749.

[34] BRENDEL W, RAUBER J, BETHGE M. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models[J]. arXiv preprint arXiv:1712.04248, 2017.

[35] CHENG M, LE T, CHEN P-Y, et al. Query-efficient hard-label black-box attack: An optimization-based approach[J]. arXiv preprint arXiv:1807.04457, 2018.

[36] SHARIF M, BHAGAVATULA S, BAUER L, et al. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition[C] // Proceedings of the 2016 acm sigsac conference on computer and communications security. 2016 : 1528 – 1540.

[37] ATHALYE A, ENGSTROM L, ILYAS A, et al. Synthesizing robust adversarial examples[C] // International conference on machine learning. 2018 : 284 – 293.

[38] SIMONYAN K, VEDALDI A, ZISSERMAN A. Deep inside convolutional networks: Visualising image classification models and saliency maps[J]. arXiv preprint arXiv:1312.6034, 2013.

[39] PAPERNOT N, MCDANIEL P, JHA S, et al. The limitations of deep learning in adversarial settings[C] // 2016 IEEE European symposium on security and privacy (EuroS&P). 2016 : 372 – 387.

[40] CROCE F, HEIN M. Sparse and imperceivable adversarial attacks[C] // Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019 : 4724 – 4732.

[41] DONG X, CHEN D, BAO J, et al. GreedyFool: Distortion-aware sparse adversarial attack[J]. Advances in Neural Information Processing Systems, 2020, 33 : 11226 – 11236.

[42] GOODFELLOW I, BENGIO Y, COURVILLE A. Deep learning[M]. [S.l.] : MIT press, 2016.

[43] WAN L, ZEILER M, ZHANG S, et al. Regularization of neural networks using dropconnect[C] // International conference on machine learning. 2013 : 1058 – 1066.

[44] TOMPSON J, GOROSHIN R, JAIN A, et al. Efficient object localization using convolutional networks[C] // Proceedings of the IEEE conference on computer vision and pattern recognition. 2015 : 648 – 656.

[45] GHIASI G, LIN T-Y, LE Q V. Dropblock: A regularization method for convolutional networks[J]. Advances in neural information processing systems, 2018, 31.

[46] LARSSON G, MAIRE M, SHAKHNAROVICH G. Fractalnet: Ultra-deep neural networks without residuals[J]. arXiv preprint arXiv:1605.07648, 2016.

[47] LECUN Y, BOTTOU L, BENGIO Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11) : 2278 – 2324.

[48] WU R, YAN S, SHAN Y, et al. Deep image: Scaling up image recognition[J]. arXiv preprint arXiv:1501.02876, 2015, 7(8).

[49] DEVRIES T, TAYLOR G W. Improved regularization of convolutional neural networks with cutout[J]. arXiv preprint arXiv:1708.04552, 2017.

[50] ZHONG Z, ZHENG L, KANG G, et al. Random erasing data augmentation[C] // Proceedings of the AAAI conference on artificial intelligence : Vol 34. 2020 : 13001 – 13008.

[51] CHEN P, LIU S, ZHAO H, et al. Gridmask data augmentation[J]. arXiv preprint arXiv:2001.04086, 2020.

[52] WANG Z, BOVIK A C, SHEIKH H R, et al. Image quality assessment: from error visibility to structural similarity[J]. IEEE transactions on image processing, 2004, 13(4): 600–612.

[53] ROZSA A, RUDD E M, BOULT T E. Adversarial diversity and hard positive generation[C] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2016: 25–32.

[54] KURAKIN A, GOODFELLOW I, BENGIO S. Adversarial machine learning at scale[J]. arXiv preprint arXiv:1611.01236, 2016.

[55] KINGMA D P, BA J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.

[56] PAPERNOT N, MCDANIEL P, WU X, et al. Distillation as a defense to adversarial perturbations against deep neural networks[C] // 2016 IEEE symposium on security and privacy (SP). 2016: 582–597.

[57] FAN Y, WU B, LI T, et al. Sparse adversarial attack via perturbation factorization[C] // European conference on computer vision. 2020: 35–50.

[58] WU B, GHANEM B. lp-Box ADMM: A Versatile Framework for Integer Programming[J]. IEEE transactions on pattern analysis and machine intelligence, 2018, 41(7): 1695–1708.

[59] SINGH K K, YU H, SARMASI A, et al. Hide-and-seek: A data augmentation technique for weakly-supervised localization and beyond[J]. arXiv preprint arXiv:1811.02545, 2018.

[60] HAN S, POOL J, TRAN J, et al. Learning both weights and connections for efficient neural network[J]. Advances in neural information processing systems, 2015, 28.

[61] LUO J-H, ZHANG H, ZHOU H-Y, et al. Thinet: pruning cnn filters for a thinner net[J]. IEEE transactions on pattern analysis and machine intelligence, 2018, 41(10): 2525–2538.

[62] HE Y, ZHANG X, SUN J. Channel pruning for accelerating very deep neural networks[C] // Proceedings of the IEEE international conference on computer vision. 2017: 1389–1397.

[63] LIU Z, LI J, SHEN Z, et al. Learning efficient convolutional networks through network slimming[C] // Proceedings of the IEEE international conference on computer vision. 2017: 2736–2744.

[64] RONNEBERGER O, FISCHER P, BROX T. U-net: Convolutional networks for biomedical image segmentation[C] // International Conference on Medical image computing and computer-assisted intervention. 2015: 234–241.

[65] KRIZHEVSKY A, HINTON G, OTHERS. Learning multiple layers of features from tiny images[J], 2009.

[66] DENG J, DONG W, SOCHER R, et al. Imagenet: A large-scale hierarchical image database[C] // 2009 IEEE conference on computer vision and pattern recognition. 2009: 248–255.

[67] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C] // Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770–778.

[68] SZEGEDY C, VANHOUCKE V, IOFFE S, et al. Rethinking the inception architecture for computer vision[C] // Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 2818–2826.

[69] HU J, SHEN L, SUN G. Squeeze-and-excitation networks[C] // Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7132–7141.

[70] CHEN J, ZOU L-H, ZHANG J, et al. The Comparison and Application of Corner Detection Algorithms.[J]. Journal of multimedia, 2009, 4(6).

[71] ZHOU B, KHOSLA A, LAPEDRIZA A, et al. Learning deep features for discriminative localization[C] // Proceedings of the IEEE conference on computer vision and pattern recognition. 2016 : 2921 – 2929.

[72] NILSBACK M-E, ZISSERMAN A. Automated Flower Classification over a Large Number of Classes[C] // Indian Conference on Computer Vision, Graphics and Image Processing. 2008.

致　　谢

光阴似箭，日月如梭，三年的研究生生涯就要结束了。回想起这三年中遇到的人和事，我内心充满了喜悦，也充满了感激，我相信这将成为我一生中都难以忘却的宝贵财富。此时此刻，我由衷地感谢在研究生生涯中给予我关心、支持和帮助的师长与同学。

首先，我要感谢我的导师申富饶老师。申老师治学严谨，对科研有着独到的见解，鼓励我们从问题出发进行独立思考，做对实际生活有价值的研究工作。申老师的科研理念与对待科研的严谨态度对我产生了极大的影响，使我能够逐渐掌握独立进行科研的能力。此外，申老师坚持每周单独与组内每一位同学进行面对面的讨论交流，并组织讨论班分享彼此的研究内容。申老师的辛苦付出，帮助我们在学习与科研中更快地找到不足，并不断取得进步。

其次，我要感谢赵健老师。赵健老师多次为我们分享论文写作的经验，并逐字逐句地帮我们检查修改待投稿的英文论文，对于我们英文论文写作给予了很大帮助。同时，赵老师与我们共同参与讨论班，为我们的研究工作提出许多有价值的意见与建议。

接着，我要感谢陪伴我度过研究生生涯的同学。感谢 RINC 研究组的同学，都热心为我的科研与学习提供了许多帮助。感谢我的室友，在生活中给予我非常多关心与帮助。最后特别感谢曾丽同学，与我分享学习与生活中遇到的各种问题和趣事，付出了许多时间与精力给予我支持，并激励我不断突破自我。

最后，我要感谢我的家人。父母为我提供了最坚强的后盾，良好的生活保障及坚实的精神支柱使我能够将精力更多地投入到科研与学习之中，在面对困难时不畏惧，在面对压力时不崩溃，顺利完成学业。

简历与科研成果

基本信息

李金桥，男，汉族，1997 年 06 月出生，四川省巴中市人。

教育背景

2019 年 9 月 — 2022 年 6 月 南京大学计算机科学与技术系 硕士
2015 年 9 月 — 2019 年 6 月 河海大学计算机科学与技术系 本科

攻读硕士学位期间完成的学术成果

1. Jinqiao Li, Xiaotao Liu, Jian Zhao, Furao Shen, “AutoAdversary: A Pixel Pruning Method for Sparse Adversarial Attack”in arXiv preprint arXiv:2203.09756, 2022

攻读硕士学位期间完成的发明专利

1. 申富饶，李金桥，姜少魁，陆志浩，金祎。一种实时判定摄像头遮挡状态的方法。专利申请号：CN202010736809.6

攻读硕士学位期间参与的科研课题

1. 国家自然科学基金面上项目“基于深度感知增量式联想记忆神经网络的信息融合系统研究，Information fusion system based on deep perception and incremental associative memory neural networks”（课题年限 2019.01 至 2022.12），负责神经网络模型安全相关研究。

《学位论文出版授权书》

本人完全同意《中国优秀博硕士学位论文全文数据库出版章程》（以下简称“章程”），愿意将本人的学位论文提交“中国学术期刊（光盘版）电子杂志社”在《中国博士学位论文全文数据库》、《中国优秀硕士学位论文全文数据库》中全文发表。《中国博士学位论文全文数据库》、《中国优秀硕士学位论文全文数据库》可以以电子、网络及其他数字媒体形式公开出版，并同意编入《中国知识资源总库》，在《中国博硕士学位论文评价数据库》中使用和在互联网上传播，同意按“章程”规定享受相关权益。

作者签名：李金辉

2022 年 5 月 20 日

论文题名	基于像素自动删减的稀疏对抗攻击及其在数据增强中的应用				
研究生学号	MG1933035	所在院系	计算机科学与技术系	学位年度	2022
论文级别	☑硕士 ☐博士	☐硕士专业学位 ☐博士专业学位 （请在方框内画勾）			
作者 Email	lijq@smail.nju.edu.cn				
导师姓名	申富饶 教授				

论文涉密情况：

☑不保密

☐保密，保密期（_____年____月____日 至 _____年____月____日）

注：请将该授权书填写后装订在学位论文最后一页（南大封面）。