中国矿业大学

CHINA UNIVERSITY OF MINING AND TECHNOLOGY

本科生毕业设计(论文)

基于自监督学习的对抗样本攻击防御技术研究

Research on Defense against Adversarial Attack Based on Self-supervised Learning

作　者：袁孝健

导　师：张立江

中国矿业大学

2021 年 6 月

中国矿业大学

本科生毕业设计(论文)

基于自监督学习的对抗样本攻击防御技术研究

Research on Defense against Adversarial Attack Based on Self-supervised Learning

作　　者 袁孝健　　学　　号 06172151

导　　师 张立江　　职　　称 讲师

学　　院 计算机科学与技术学院　　专　　业 信息安全

二〇二一年六月

致谢

时光荏苒，白驹过隙。四年的求学生涯即将迈入终点，回顾伊始，当时定下的目标如今多数能够达成，虽路途曲折，但好在结局还算圆满。

感谢我的指导老师——张立江老师。在这几个月里，张老师认真负责的作风不断激励着我，也正是在张老师的指导与督促下，论文才得以顺利完成。张老师也是我引领我进入信息安全领域的指路人，在我的学习、竞赛道路上提供了诸多资源与机会。

感谢研究生课题组的陈可江师兄。此次毕业设计是我第一次接触人工智能安全领域，从选题到研究思路，师兄都悉心的给予我指导，也使我能够更好的适应即将到来的研究生生涯。

感谢 BXS 网络安全社团。从第一次的校赛开始接触 CTF，便被其中的魅力所吸引。随后有幸进入 BXS 社团，与队友们一起学习安全知识、参加安全竞赛，这段经历使我收获了很多、也成长了很多。

感谢即将成为母校的中国矿业大学，提供了优质的平台与良好的环境，使我能够没有顾虑的向着理想前进。

感谢我的家人一直以来的支持与关心。漫漫求学路，经历过挫折与困难，也有过彷徨与无措，而家人是我最后的港湾，给予我鼓励和重新上路的勇气。

感谢自己一直以来的坚持与努力。学习本无底，前进莫彷徨。

摘　要

深度学习的快速发展使其在许多领域都取得了巨大的成功，而对抗样本的存在又很大程度上威胁着深度学习系统在实际应用场景中的安全性。另一方面，随着数据量的不断增多，人工注释标签将会花费高昂的成本，因此自监督学习在近年来成为了一种流行的新范式，并有望成为人工智能未来发展的新方向。

截至目前很少有工作考虑自监督学习与对抗样本防御之间的关联，因此本文将自监督学习与对抗训练的思想相结合，研究基于自监督学习的对抗样本防御方法。首先，本文再现了基于 SimCLR 的对抗样本防御方法，然后分析了 MoCo 框架相对于 SimCLR 存在的优势，并提出了一种基于 MoCo 的对抗样本防御方法 AdvMoCo。在 MoCo 的基础上，AdvMoCo 将对抗扰动看成一次数据增强，从而获得正样本的对抗性视图。除此之外，还设计了两个动态存储队列分别用于存储先前样本干净的表征和对抗性的表征，并将它们作为负样本。通过在 CIFAR-10 数据集上的实验证明，本文提出的方法在干净样本的准确率和对抗鲁棒性两方面均优于传统的对抗防御方法。更重要的是，由于该方法自监督的特性，在实际应用中可以使用任意多的无标签数据进行预训练，从而使模型获得更高的鲁棒性。

最后，本文以交通标志识别为目标任务，开发了一个基于 Flask 的 Web 演示系统，并展示了深度学习系统的脆弱性以及提出方法的有效性。

该论文有图 31 幅，表 11 个，参考文献 44 篇。

关键词：对抗攻击；对抗防御；自监督学习；深度学习；交通标志识别

Abstract

The rapid development of deep learning has made it a great success in many fields, but the existence of adversarial examples greatly threatens the security of deep learning systems in practical application scenarios. On the other hand, as the amount of data continues to increase, manual annotation of labels will be costly. Therefore, self-supervised learning has become a popular new paradigm in recent years and is expected to become a new direction for the development of artificial intelligence in the future.

So far, few works have considered the connection between self-supervised learning and adversarial defense. Therefore, this paper combines the ideas of self-supervised learning and adversarial training to study the method of adversarial defense based on self-supervised learning. First, this paper reproduces the adversarial defense based on SimCLR, then analyzes the advantages of the MoCo framework over SimCLR. And inspired by this, this paper proposes a novel adversarial defense method based on MoCo called Adversarial MoCo(AdvMoCo), which regards adversarial perturbations as data augmentations to obtain adversarial positive samples. Besides, AdvMoCo have two dynamic memory queues to maintain the historical clean and adversarial representations respectively, and negative samples will be taken from the queue. Experiments on CIFAR-10 dataset have proved that the proposed method is superior to traditional adversarial defense methods in terms of the accuracy of clean samples and the adversarial robustness. More importantly, due to the self-supervised features of this method, any amount of unlabeled data can be used for pre-training in practical applications, so that the model can achieve higher robustness.

Finally, this paper also takes the traffic signs recognition as the target task, and develops a Web demonstration system based on Flask to demonstrate the fragility of the deep learning system and the effectiveness of the proposed method.

The thesis has 31 figures, 11 tables, and 44 references.

Keywords: adversarial attack; adversarial defense; self-supervised learning; deep learning; traffic sign recognition

目 录

Contents

1 绪论

1 Introduction

1.1 研究背景及意义(Research Background and Significance)

深度学习是一种基于神经网络架构，对数据进行表征学习的算法。近年来，深度学习技术以日新月异的速度不断发展着，并在计算机视觉、自然语言处理等诸多领域都取得了令人惊讶的成果。基于深度学习的各项技术已经在实际工程中得到了广泛的应用，例如自动驾驶[1]、疾病诊断[2]、人脸识别[3]等，但这些领域对深度学习系统的安全性和鲁棒性也有着更高的要求。然而，自从 Szegedy 等人首次提出了对抗样本(Adversarial Example)的概念[4]之后，许多研究[4-9]证明了深度学习模型在对抗攻击(Adversarial Attack)面前的脆弱性。

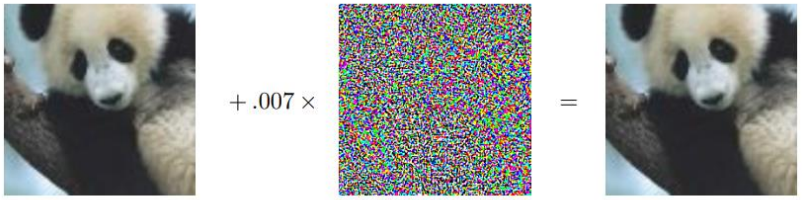

图 1-1 对抗样本[5]

Figure 1-1 Adversarial Example[5]

对抗攻击是目前限制深度学习系统在安全性要求较高的领域使用的主要原因之一，其最早出现于 Szegedy 等人发表在 2014 年国际表征学习大会的论文中[4]。攻击者通过恶意优化算法生成精心设计的微小对抗扰动(Adversarial Perturbation)，并将其加入原始输入样本中来构造对抗样本。这些对抗样本很难被人类视觉系统察觉，但是却能对深度学习模型的输出造成很强的干扰，使模型做出与正常情况完全不同的决策，从而达到欺骗人工智能系统的目的。如图 1-1 所示[5]，左图中的原始样本会被深度学习模型以 56.7%的置信度分类为“panda”，而将中间精心设计的对抗扰动与原始样本相加来生成右图中的对抗样本时，模型将会以 99.3% 的高置信度将其分类为“gibbon”，而以人类视觉的角度却很难分辨原始样本与对抗样本之间的差异。除此之外，构造出的对抗样本已被证明在类似的神经网络结构以及不同的数据集之间具有一定的迁移性[10]。对抗训练[15](Adversarial Training)是一种有效的防御方式，通过将对抗样本加到训练数据中，从而使监督模型在训练时学习到较为鲁棒的表征来抵御对抗攻击，目前还没有攻击方法能将其完全攻破。

自监督学习(Self-supervised Learning)介于监督学习和无监督学习之间，是一种具有监督形式的无监督学习新范式，它利用数据本身的特点生成监督信息，旨在从大量无标记数据中学习到高效、鲁棒、具有语义的表征。基于对比学习的自监督学习方法在最近取得了十分瞩目的进展，随着 MoCo[11]、SimCLR[12]、MoCo v2[13]、BYOD[14]等方法的相继提出，自监督学习获得的表征已经能在下游 ImageNet 分类任务上取得与监督学习相当的性能，并且整个过程不需要任何人工注释的标签。

在构建包括神经网络在内的人工智能模型时，标签效率和模型鲁棒性是两个理想化的特性，也是将深度学习模型大规模应用于实际工程中的两个重要前提。自监督学习的发展有望解决第一个问题，缓解人工注释标签所带来的高昂成本。图灵奖得主 Yann LeCun 在 2020 年的 AAAI 和 ICLR 两大人工智能顶级会议上都表达了对于自监督学习的看好，他认为“自监督学习是人工智能的未来，可能使其产生类人的推理能力”。然而，这些最新的自监督方法却并没有考虑模型的鲁棒性，依然很容易受到对抗攻击的威胁。标签数据稀缺的问题在训练鲁棒的深度学习模型以抵御对抗攻击时显得更加重要[16]，和其他深度学习方法类似，鲁棒性的研究也是自监督学习发展道路上所不可避免要攻克的难题。

因此，本设计将自监督学习与对抗训练的思想相结合，研究如何提升自监督学习模型防御对抗攻击的能力，并且利用自监督学习不需要标签数据的特点来缓解对抗训练对人工注释标签的依赖。

1.2 国内外研究现状(Research Status at Home and Abroad)

1.2.1 对抗样本攻击研究现状

通常来说，对抗攻击可以从几个不同的角度来进行分类。首先，从攻击者的目的来看，可以分为无目标攻击和有目标攻击。在无目标攻击中，攻击者所构造的对抗样本仅仅需要使模型产生错误的结果即可，而不限定具体的错误类别。在有目标攻击中，攻击者则希望构造的对抗样本能够使模型产生预期中的错误类别。其次，从攻击者对目标模型的了解程度来看，可以分为白盒攻击和黑盒攻击。在白盒攻击中，攻击者不仅可以知道目标模型所使用的算法，还能获取其神经网络的结构、参数、梯度等信息。攻击者可以利用目标模型来针对性的构造对抗样本，由于公开的信息容易使攻击者了解模型的弱点，因此目前研究者们希望鲁棒的机器学习模型要具有抵御白盒攻击的能力[17]。在黑盒攻击中，攻击者无法得知目标模型的任何信息，只能在一定的限制条件下向模型提供输入信息，然后得到相应的输出结果。因此这种场景下的攻击只能通过观察、分析输入与输出之间的对应关系，根据经验推测目标模型的弱点，并构造对抗样本。黑盒攻击相对于白盒攻

击来说更具有实用性，因为模型的使用者通常不会公开模型的具体信息来保证安全性。从对抗样本与原始样本的距离来看，可以分为无穷范数攻击(ℓ_∞攻击)、二范数攻击(ℓ_2攻击)和零范数攻击(ℓ_0攻击)，分别表示用ℓ_∞、ℓ_2和ℓ_0范数来计算对抗样本和原始样本的距离，用于衡量对抗扰动的强度大小。攻击者通常会对对抗扰动的强度加以限制来保证其对人类视觉系统的不可见性。由于ℓ_0范数的很难进行优化求解(NP-hard)，因此多数情况下使用ℓ_1范数作为它的“最优凸近似”。

传统的对抗样本生成方法主要分为两类，分别是基于模型梯度的攻击和基于优化的攻击。基于模型梯度的攻击以 Goodfellow 等人于 2014 年所提出的快速梯度符号方法(Fast Gradient Sign Method, FGSM)[5]为代表，通过将对抗扰动看成要更新的参数，以增大目标模型的损失函数值为目标进行反向传播，再根据得到的梯度信息对扰动值进行更新，从而构造对抗样本使模型出现错误。通常这类生成方法的速度很快，并且具有良好的迁移性。另一种基于优化的攻击以 Carlini 和 Wagner 提出的 C&W 攻击方法为代表[9]，通过将对抗样本的生成作为一个优化问题进行求解，优化的目标为“对抗样本和原始样本距离尽可能小的同时能够使目标模型分类错误，且错误分类的置信度尽可能高”，以此来不断优化对抗扰动。这类方法虽然攻击速度较慢，但是可以生成扰动幅度很小、与原始样本极为相似的对抗样本，在白盒攻击的场景下性能很好。

1.2.2 对抗样本防御研究现状

对抗性机器学习近年来成为了一个热门的研究领域，研究者们相继提出了许多方法来提高模型抵御对抗样本攻击的能力，主要分为鲁棒性防御和对抗样本检测两类。鲁棒性防御又分为输入预处理和模型的鲁棒性优化，旨在模型遭受对抗样本的攻击时仍然能够做出正确的决策。对输入的预处理也称为基于变换的防御方法，主要是对模型的输入进行一定处理来消除对抗扰动对模型的影响，例如 JPEG 压缩、旋转、去噪等。而模型的鲁棒性优化是指通过修改网络结构、数据集或改进模型的训练过程等方法使模型本身具有一定对对抗样本的鲁棒性。Zantedeschi 等人[18]提出使用一种有界限的 ReLU 激活函数来消除对抗样本对模型的扰动。Papernot 等人提出的防御蒸馏方法[19]，通过对网络结构进行一定的修改以及加入优化项来避免模型过于拟合正常样本从而导致很容易被对抗样本欺骗。梯度混淆是一种特殊的梯度掩膜方法[20]，主要目的是使攻击者不能轻易的获取模型的梯度信息来构造对抗样本。Mardy 等人[15]提出在训练过程中针对模型构造对抗样本，并将其加入训练集中对模型进行对抗训练，从而使模型对特定的对抗攻击鲁棒，这被认为是目前最有效的防御方法之一。由于使模型能够在对抗样本面前正确输出的难度较大，因此不少研究开始将思路转向对抗样本的检测，即只需要判断图片是否为对抗样本，避免将其模型，从而起到防御对抗样本攻击的

作用。Li 等人[21]用 PCA 和级联分类器来检测图片是否为对抗样本。Xu 等人[22]提出一种基于特征压缩的检测框架，通过对比模型对原始样本和对抗样本输出的不同来判断是否为对抗样本。

1.2.3 自监督学习研究现状

深度学习模型的性能在很大程度上取决于神经网络结构的容量以及训练数据的数量。近年来，为了提高神经网络的容量，研究者们提出了许多新的网络结构，如 AlexNet[23]、VGG[24]、GooLeNet[25]、ResNet[26]等。与此同时，人们也收集了大量的数据来构建大规模的数据集用于训练这些愈发复杂的模型，如 ImageNet[27]、OpenImage[28]等。以 ImageNet 为例，这个大规模的数据集总共包含了 130 万张覆盖了 1000 个类别的图片，其中每一张图片的标签都是人工进行标注的。毫无疑问，想要制作这样一个庞大的数据集是需要耗费大量人力、物力的。因此，无监督学习由于其不需要标签的训练方式，可以节省收集数据集所带来的高昂成本，一直以来都是研究的热点方向之一。近年来，自监督学习作为无监督学习的分支发展迅速，在一些任务上已经能够达到媲美监督学习的性能。

自监督学习旨在从大量无标记数据中学习到高效、鲁棒且具有语义信息的表征，通常的方法是人工定义一个前置任务(pretext task)，训练目标是让神经网络能够很好的解决这个前置任务，然后将训练后的网络进行参数冻结，并作为特征提取器用于下游任务(如图像分类)。因此，基于自监督学习的神经网络能否提取出良好的视觉表征很大程度上取决于所定义的前置任务的好坏，其最基本的要求是能够将数据本身的某些特征定义为“标签”从而提供一定的监督信息，使网络通过有监督的方式进行训练。Noroozi 等人[29]对图像进行分块并打乱相对顺序，以原始图像作为标签，训练网络将图像块恢复为正确的顺序。Gidari 等人[30]将图像旋转一定的角度，以旋转角度作为标签训练网络对其进行预测。Zhang 等人[31]通过对图像进行灰度化，并将原图作为标签训练网络完成图像上色任务。Pathak 等人[32]通过随机剪裁掉某一位置上的图像块，然后训练网络对缺失部分进行修补。

近年来，基于对比学习的自监督学习方法变得越来越流行，其主要思想是利用对比损失最大化来自同一张图片不同视图之间的相似性，而最小化来自不同图片的视图之间的相似性。这也可以看作将前置任务定义为：训练网络能够识别出同一张图片的不同视图之间的相似性，并鉴别出不同图片之间的差异性。使用这种思想的自监督学习，已经被多项优秀的工作证明了其在学习丰富的特征表示方面非常有效[11-14]，在 ImageNet 图像分类的下游任务上甚至达到了与全监督模型相当的性能。

1.3 主要研究内容(Main Research Contents)

本课题系统地研究了自监督学习与对抗样本攻击防御技术之间的关联，旨在设计一种基于自监督学习的对抗样本防御方法，从而在提高自监督模型鲁棒性的同时，摆脱对抗训练对数据标签的过分依赖。研究的主要内容包括：

(1) 对 Kim 等人[44]的工作进行梳理，实现了一种基于 SimCLR 的对抗样本防御方法。在此过程中，探究了自监督学习场景下的对抗样本生成方法，并分析了其有效的原因。

(2) 通过对自监督学习和对抗样本攻防领域的理解，给出了针对自监督学习模型鲁棒性的实验评估指标。

(3) 分析了 MoCo 框架相对于 SimCLR 的优势，并设计了一种基于 MoCo 的对抗样本防御方法，并通过实验的对比和分析，证明了提出方法的优越性。

(4) 为了更好的展示深度学习模型在实际场景中的脆弱性，将本设计中的防御方法与交通标志识别任务相结合，设计并开发了一个基于 Flask 的对抗攻防演示系统。

1.4 论文组织结构(The Organizational Structure of the Thesis)

本文围绕基于自监督学习的对抗样本攻击防御技术展开，论文共分为 6 章，各章节主要内容如下：

第 1 章：绪论。本章主要介绍了本设计的研究背景和研究意义，并调研了国内相关共工作的研究现状。

第 2 章：背景知识。本章从神经网络的基本概念入手，引出了深度学习系统的脆弱性，并由此介绍了几种主流的对抗样本攻击和防御技术，最后阐述了自监督学习的基本概念与核心思想。

第 3 章：基于 SimCLR 的对抗样本防御方法。本章首先介绍了自监督学习框架 SimCLR，然后根据 Kim 等人[44]的工作重现了基于 SimCLR 的对抗样本防御方法，并进行了一定的实验与分析。

第 4 章：基于 MoCo 的对抗样本防御方法。本章首先介绍了自监督学习框架 MoCo，并分析了其相对于 SimCLR 的优势。然后，受到第 3 章工作的启发，设计了一种基于 MoCo 的对抗样本防御方法，并通过实验证明了方法的有效性。

第 5 章：防御方法在交通标志识别系统中的应用。本章首先介绍了交通标志数据集 GTSRB 并对前面方法的迁移性进行了一定实验分析，最后将该防御方法与交通标志识别相结合，开发了一个基于 Flask 的 Web 演示系统。

第 6 章：总结与展望。本章对研究的主要内容、完成的工作以及创新点进行了总结，并展望了未来进一步的研究。

2 背景知识

2 Background Knowledge

本章首先介绍了神经网络的基本概念，由此引出了深度学习系统在对抗样本面前的脆弱性，然后介绍了常用的五种对抗样本攻击技术以及四种较为有效的对抗样本防御方法。最后对自监督学习的主要思想进行了阐述，并介绍了目前主流的三种自监督学习方法，为后续章节提供了理论基础。

2.1 神经网络(Neural Network)

深度学习主要指的是基于神经网络架构的一种表征学习算法，而神经网络，又称人工神经网络，则是模拟生物神经网络的一种数学计算模型。神经网络也可看作为一个通用的函数逼近器，理论上来说，在拥有足够多的神经元时，神经网络模型可以拟合任意函数。

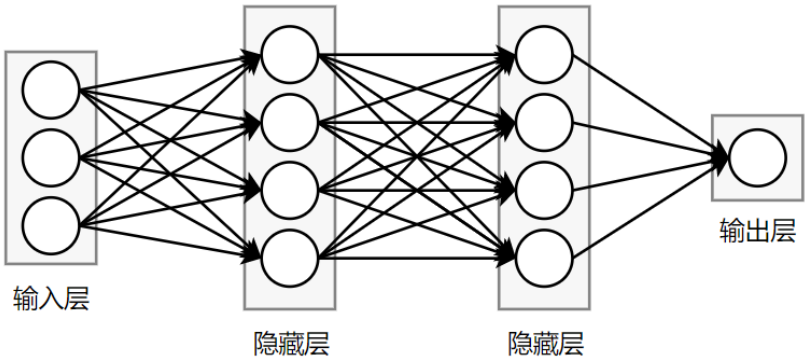

图 2-1 人工神经网络

Figure 2-1 Artificial neural network

前馈神经网络也叫做多层感知机，是人工神经网络中一种最基本的结构，由输入层、隐藏层、输出层三部分组成。图 2-1 展示了一个具有两层隐藏层的人工神经网络结构，通常情况下一个神经网络可以含有若干个隐藏层，这也是“深度学习”的名称来源，神经网络的隐藏层数量越多，其“深度”也就越深，具有多个隐藏层的神经网络也被叫作深度神经网络。随着神经网络“深度”的增加，其可以学习到的函数也越复杂，但是也同时容易产生过拟合的现象。

神经网络中的隐藏层通常由多个神经元组成(其结构见图 2-2)，其为神经网络中最基本的计算单元，$x \in \mathbb{R}$为神经元的输入，$y \in \mathbb{R}^m$为神经元的输出，计算方法如公式(2-1)：

$$\boldsymbol{y} = f(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}) \tag{2-1}$$

其中，$\boldsymbol{W} \in \mathbb{R}^{m \times n}$为权重矩阵，$\boldsymbol{b} \in \mathbb{R}^m$为偏置，$f(\cdot)$为激活函数，常用的激活函数包括 tanh、sigmoid、ReLU 等[33]。神经网络的训练过程实际上是一个参数拟合

的过程，模型的参数也就对应着每个神经元的$\boldsymbol{W}$和$\boldsymbol{b}$，通常将其看作一个优化问题，使用梯度下降或其变种方法来求解深度学习模型的一组最佳参数，具体的求解过程则需要用到链式法则和反向传播算法。

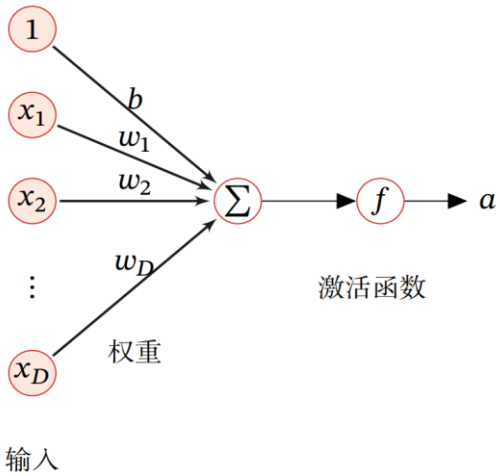

图 2-2 单个神经元的结构

Figure 2-2 The structure of a single neuron

卷积神经网络是指在前馈神经网络的至少一个层中用卷积运算代替矩阵乘法运算，是目前计算机视觉领域的主流神经网络架构，在多个任务上都取得了优异的表现，著名的 AlexNet、VGG、ResNet 等神经网络模型均是基于卷积神经网络的。图 2-3 给出了卷积神经网络的一般结构，主要包括输入层、卷积层、池化层(下采样)、全连接层和输出层。

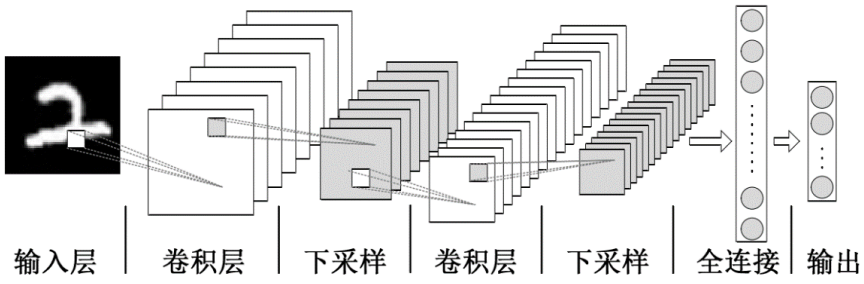

图 2-3 卷积神经网络架构图

Figure 2-3 Convolutional neural network architecture diagram

2.2 对抗样本攻击技术(Adversarial Attack Technologies)

2.2.1 FGSM 攻击

Goodfellow 等人[5]提出的快速梯度符号法(FGSM)旨在仅仅用一步迭代快速而简单的生成对抗样本，是一种基于梯度的白盒攻击算法，具体计算如下：

$$\text{无目标攻击：}\ x' = x + \epsilon \cdot sign\left(\nabla_x \mathcal{L}(\theta, x, y)\right) \qquad (2-2)$$

$$\text{有目标攻击：}\ x' = x - \epsilon \cdot sign\left(\nabla_x \mathcal{L}(\theta, x, t)\right) \qquad (2-3)$$

其中，x为原始样本；x'为生成的对抗样本；ϵ为步长，用于限制扰动的幅度大小以保证攻击的不可见性；$sign(\cdot)$为符号函数，用来得到梯度的方向；$\nabla_x \mathcal{L}(\theta, x, y)$则表示训练过程中使用的损失函数$\mathcal{L}$对于样本$x$的梯度。

在无目标攻击中，可以将(2-2)式看作梯度上升的一步，目的是构造出使模型损失值增大的对抗扰动，从而使模型在遇到对抗扰动时出错的概率增大。而在有目标攻击中，则可以将公式(2-3)看作梯度下降的一步，其与正常的训练过程类似，但是使用攻击的目标类别t来替代样本的真实类别y，目的增大模型将样本预测为目标类别的概率，而不是单纯的使模型出错。图 1-1 即为使用 FGSM 攻击在 ImageNet 数据集上生成对抗样本的示例。

2.2.2 PGD 攻击

FGSM 攻击虽然能够快速的生成对抗样本，但是对于神经网络这样复杂的非线性模型来说，极小的范围内也有可能产生剧烈的变化，仅仅通过一次梯度下降得到的更新方向不一定是完全符合预期的，因此 Mardy 等人[15]提出了投影梯度下降法(Projected Gradient Descent, PGD)。PGD 攻击是可以看作 FGSM 的迭代版本，它的主要思想是通过多次迭代，但是每次迭代只更新较小的值，若超过最大扰动范围即进行剪裁，最终经过设置的迭代次数之后得到对抗样本，具体计算方法(2-4)式所示：

$$\begin{aligned} x'_0 &= x \\ x'_{t+1} &= \mathrm{Clip}_{\mathrm{x},\epsilon}\left(x + \alpha\,\mathrm{sign}\left(\nabla_x L(x'_t, y)\right)\right) \end{aligned} \tag{2-4}$$

其中，x'_t表示第t步迭代时的对抗样本，在初始时通常将一个随机初始化(如高斯分布、正态分布等)的对抗扰动加在原始样本上进行后续的迭代；每个中间迭代步骤与 FGSM 类似，但是会将步长α设置成一个相对较小的值；$Clip$操作将每一步得到的对抗样本x'投影到半径为ϵ的球型邻域 $\boldsymbol{B}_\varepsilon(x)\colon\{x'\colon \|x' - x\|_\infty \leqslant \epsilon\}$ 表面。

2.2.3 DeepFool 攻击

与 FGSM 和 PGD 不同，Moosavi-Dezfooli 等人[8]提出的 DeepFool 攻击方法利用神经网络的几何结构来构造对抗扰动，研究分类器F对样本x的决策边界，尝试找出一条让x越过决策边界的路径。DeepFool 在每次迭代中，使对抗样本朝着垂直于决策边界的方向进行更新，直到越过决策边界时停止，旨在找到使分类模型出错的最小对抗扰动。作者通过实验结果表明了，对一些常见神经网络分类器，大部分样本都非常接近决策边界，因此即使是小扰动，也足以使分类器出错。

然而大多是神经网络模型并不是线性的，因此这种攻击方法具有一定的局限性。图 2-4 展示了分别通过 DeepFool 和 FGSM 生成的对抗样本，可以看出

DeepFool 生成对抗扰动确实具有更好的不可见性。

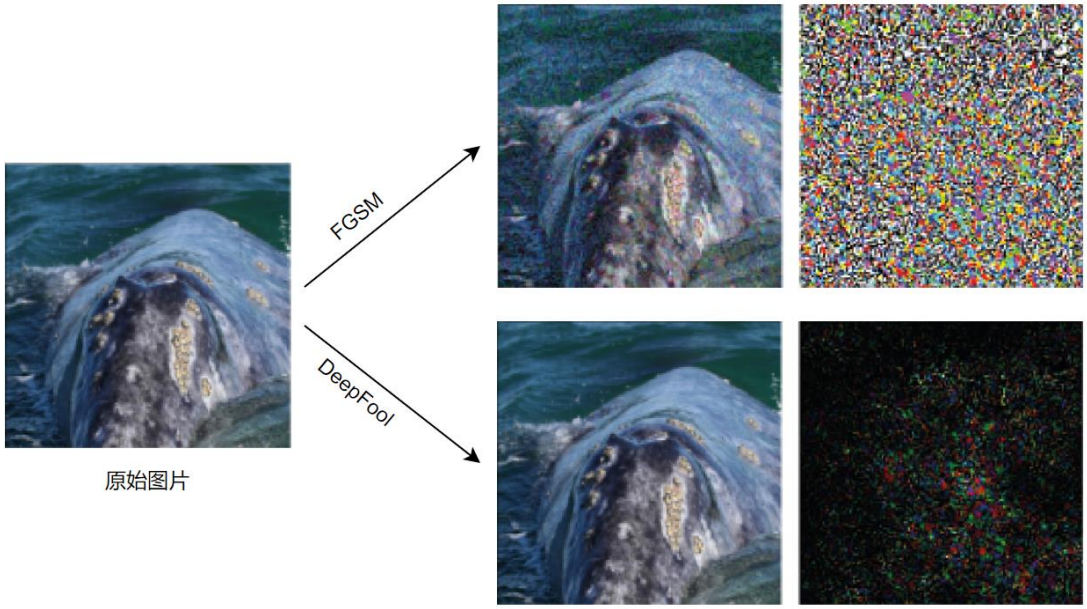

图 2-4 FGSM 和 DeepFool 生成的对抗样本

Figure 2-4 adversarial examples generated by FGSM and DeepFool

2.2.4 C&W 攻击

Carlini 和 Wagner 两人提出的 C&W 攻击[9]，是一种基于优化的对抗攻击方法，在白盒场景下具有很强的攻击能力，攻破了能够防御 FGSM 的模型蒸馏防御方法[19]。C&W 攻击旨在同时保证低对抗扰动和高攻击成功率，其对应的具体优化问题如下：

$$\begin{aligned} &\text{minimize } \mathcal{D}(x, x') + c \cdot f(x', t) \\ &\text{such that } x' \in [0,1]^n \end{aligned} \qquad (2-5)$$

在上式中，x表示原始样本；x'表示对抗样本；$D(x, x')$表示原始样本和对抗样本之间的距离，通常用L_p范数来计算；f通常定义为$f(x', t) = (\max_{i \neq t} \mathrm{F(x')_i} - \mathrm{F(x')_t})^+$，其中$F$表示模型输出类别的概率，$\max_{i \neq t} \mathrm{F(x')_i}$即为模型将对抗样本分类到正确类别$i$的概率，而$F(x')_t$则为模型将对抗样本分类的指定类别$t$的概率，因此最小化$f(x', t)$也就达到了使模型分类错误，且错误类别的置信度尽可能高的优化目标；c是用来两个优化目标权重的参数，可以用二分法来确定其具体的数值。

(2-5)式中的$f(x, y)$也被称为边际损失函数，旨在惩罚其他类别i的得分$F(x)_i$大于真实类别等分$F(x)_y$的情况，后续的许多工作也证明了使用边界损失生成的对抗样本比使用交叉熵损失具有更强的攻击性。

C&W 攻击被认为是最强的攻击方法之一，也攻破了许多对抗防御策略，因此可以作为检验神经网络分类器的安全性或防御策略的基准。

2.2.5 AutoAttack 攻击

除了上面这些经典的对抗攻击方法之外，最近研究者们也提出了一些新的攻击方法用来评估深度学习模型的安全性及鲁棒性，如白盒场景下的快速自适应边界攻击(FAB)[34]以及基于高效查询的黑盒攻击方法 Square Attack[35]等。

由于 PGD 攻击被认为是最强的一阶梯度白盒对抗攻击方法，所以经常被用来评价模型的鲁棒性或新型防御策略的效果。然而，Croce 等人[36]认为目前的 PGD 攻击因为使用了交叉熵损失以及固定步长的迭代策略，会导致其攻击效果并没有我们所认为的那么好。为了解决这些问题，他们首先提出了 Auto-PGD，旨在寻求一种在迭代过程中更加合适的步长更新策略，主要从步长的动量更新、步长的自适应选取以及重新启动三个方面进行考虑。其次，他们提出了一种新的 DLR 损失用来替代交叉熵损失，如(2-6)式所示：

$$\mathrm{DLR}(x,y) = -\frac{z_y - \underset{i \neq y}{max} z_i}{z_{\pi_1} - z_{\pi_3}} \qquad (2-6)$$

作者将使用交叉熵损失和 DLR 损失的 Auto-PGD 分别称为 APGD-CE 和 APGD-DLR，并将它们与 FAB 和 Square Attack 结合形成了一个对抗攻击的集成方法 AutoAttack(AA)，其被证明是目前位置最先进的对抗样本攻击方法之一。由于 AA 将多种互补的对抗攻击相结合，在相对较低的计算成本下的同时，不需要用户调整额外的超参数，因此可以很好的作为测试深度模型对抗鲁棒性的基准方法。

2.3 对抗样本防御技术(Adversarial Defense Technologies)

2.3.1 防御蒸馏

“蒸馏”技术是由 Hinton 等人提出用于减少深度神经网络结构和参数规模的一种模型压缩训练技术，通过将训练集真实标签作为“硬标签”训练复杂网络(教师模型)，再将复杂网络对训练集输出的概率分布作为“软标签”用于训练简单网络(学生模型)。

Papernot 等人[19]借鉴了“蒸馏”的思想来训练神经网络模型抵御对抗样本攻击，该防御方法的具体架构见图 2-5：

(1) 首先，利用利用训练数据X和对应的“硬标签”训练教师模型F，并设置 softmax 函数的温度参数为T，训练结束后输出概率分布$F(X)$。

(2) 然后，将第一步的$F(X)$作为数据的“软标签”训练另一个与教师模型结构相同的学生模型(或称蒸馏模型)F^d，并且设置相同的温度参数T。

(3) 最后，将 softmax 函数的温度参数设为$T = 1$后，使用蒸馏模型F^d在测试

数据(包含对抗样本)上进行分类。

防御蒸馏方法使模型的决策边界更加平滑，在保证干净样本准确率的同时，能够有效的防御 FGSM 等基于梯度攻击。然而，C&W 攻击已经被证明可以成功的攻破防御蒸馏。

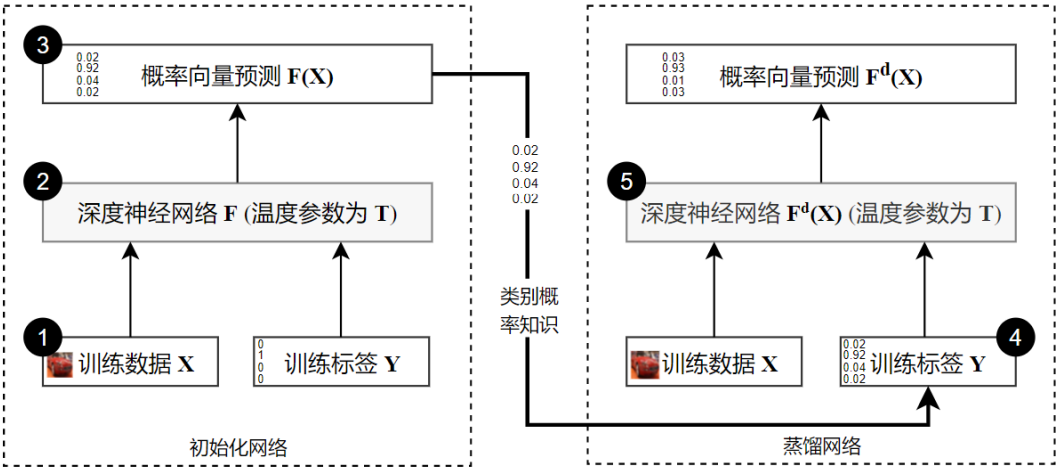

图 2-5 防御蒸馏架构图

Figure 2-5 Distillation defense architecture diagram

2.3.2 对抗训练

对抗训练是目前为止最常用的一种对抗防御策略，主要思想是在训练阶段将生成的对抗样本加入到训练数据中，并用真实的标签训练模型尽可能的成功分类对抗样本。这样对抗训练得到的模型将具有一定的鲁棒性，从而在使用时不易被对抗样本欺骗。

Goodfellow 等人[5]利用提出的快速符号梯度(FGSM)攻击方法生成对抗样本，然后在干净样本和对抗样本组成新的数据集上训练模型，从而使训练后的模型变得更加鲁棒，可以在使用时抵御快速符号梯度攻击。

Kurakin 等人[37]为了将对抗训练扩展到 ImageNet 这样的更大的数据上，引入了批标准化的训练策略，实验证明它们的方法能够有效的提高对抗训练的效率。

通过上述方法进行对抗训练的模型虽然对快速符号梯度法具有一定的鲁棒性，但是很容易被使用迭代攻击生成的对抗样本所影响。为此 Madry 等人[15]提出了使用投影梯度下降法(PGD)代替单步攻击来生成对抗样本用于对抗训练，并且模型只在数据集的对抗样本上进行训练，而并不使用任何的干净样本。这种对抗训练方法在对不仅对迭代攻击鲁棒，也同样能够抵御单步攻击，是目前较为流行的一种对抗防御策略。然而，由于此方法需要在训练过程中加入一个内部迭代进行对抗样本的生成，因此时间复杂度与正常训练相比会成倍的增加(取决于对抗攻击迭代的次数)。所以此对抗训练策略目前主要展示了在 CIFAR10、MNIST 等小规模数据集上的效果，很难扩展到如 ImageNet 这样庞大的数据集上。

Tramèr 等人[38]提出了一种新的对抗训练策略，可以有效地使模型抵御单步攻击，同时也能够扩展到大型数据集上。其主要思想是将对抗样本的生成与目标的模型的训练分离，利用单步攻击生成的对抗样本迁移性较好的特性对训练集进行扩充。具体的，假设想要得到鲁棒的目标模型M，则首先使用不同的超参数预训练另外三个模型M_1、M_2和M_3。然后对于数据集中的每个干净样本x，使用单步攻击(FGSM)以M_1、M_2和M_3为目标模型分别构造对抗样本x_1'、x_2'和x_3'。由于单步攻击良好的迁移性，对抗样本x_1'、x_2'和x_3'对目标模型M也具有较好的攻击性，因此可以将这些对抗样本同样加入到数据集中对目标模型M进行训练，从而增强模型的对抗鲁棒性。实验表明，这种集成对抗训练的方法可以扩展的 ImageNet 数据集上以提升深度学习模型对单步攻击的鲁棒性。

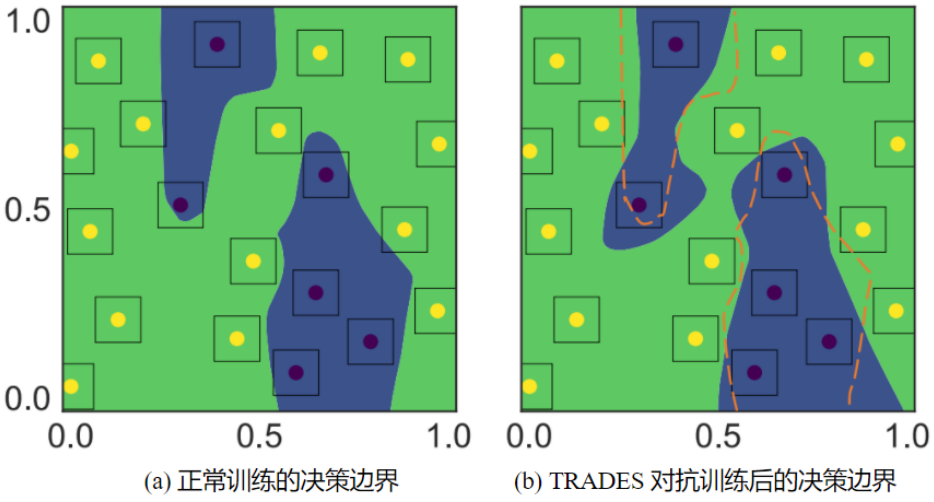

图 2-6 正常训练与 TRADES 训练后的决策边界

Figure 2-6 Decision boundary after normal training and TRADES

近年许多工作证明了对抗训练的有效性，因此这也成为了目前主流的防御策略，许多研究者们在此思想上提出了许多对抗训练的改进版本。Zhang 等人[39]提出了 TRADES，使用了一种新的损失函数用于对抗训练，可以通过超参数的设定来权衡目标模型的准确率和鲁棒性，其具体形式见(2-7)式：

$$\min_f \mathbb{E}\left\{\mathcal{L}(f(X),Y)+\beta \max_{X'\in\mathbb{B}(X,\epsilon)} \mathcal{L}\big(f(X),f(X')\big)\right\} \qquad (2-7)$$

上式中，第一项通过最小化模型预测$f(X)$和标签Y之间的差异来保证准确率，可以看作正常训练时的损失项；第二项通过最小化模型对干净样本X和对抗样本X'的预测之间的差异来保证鲁棒性；β是一个用于权衡这两项的超参数，选择较大的β值可以提高模型鲁棒性，但同时也会降低其在干净样本上的准确率。TRADES 可以使模型的输出更加平滑，并将各个样本尽可能的推离决策边界(见图 2-6，左图为正常训练后的决策边界，右图为使用 TRADES 方法训练后的决策边界)，可以有效的避免模型被轻微的对抗扰动所欺骗。

2.3.3 高斯数据增强

高斯噪声是指从高斯分布: $f(x)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$中随机取样的噪声，也是图像处理领域常见一种噪声，其

Zantedeschi 等人[18]通过观察，发现大部分的对抗样本相当于在原始图像上叠加一定的噪声，并且认为理想情况下使可以直接用高斯噪声来模拟对抗扰动。这样的好处是可以在训练过程中以数据增强的方式构造对抗样本对模型进行训练，而不需要用使用额外的攻击算法。实验表明，通过此种方法提高鲁棒性的模型，能够将对抗攻击的成功率从 53.6%降低到 36.2%。虽然直接进行高斯数据增强非常的简单、快速，但是这种模拟对抗样本的方式只是理论上，在实际环境下依然比较容易收到对抗攻击的影响。

2.3.4 去噪

许多研究者认为对抗扰动是一种噪声，因此可以利用一些图像处理技术对输入样本进行去噪。Osadchy 等人[40]利用滤波的思想，考虑使用滤波器处理对抗样本来进行去噪。因为剩余的小扰动也会随着特征层的传递而放大，传统的基于像素级别的去噪器仍然容易使模型将分类错误。为了解决这样的问题，Liao 等人[41]提出了使用高级表征为引导的去噪器(High-level representation Guided Denoise, HGD)，主要思想是将损失函数定义在网络最后几层输出的高级表征上，而不再基于像素之间的差异。实验表明，HGD 能够在特征的层面抑制对抗扰动，能够有效的提高模型针对白盒以及黑盒攻击的鲁棒性。

但是，这种基于去噪的防御策略并不能解决对抗攻击所带来的威胁，若攻击中已知 HGD 的存在，那么仍然可以将其攻破，通常可能需要配合其他的防御策略共同使用。

2.4 自监督学习(Self Supervised Learning)

2.4.1 基于生成的自监督学习

基于生成的自监督学习主要是通过训练模型生成数据来学习其中的视觉表征。这种类型的方法通过将某个生成任务定义为自监督学习的前置任务来训练特征提取器，常用的生成任务包括图像上色[31]、图像修补[32]、图像超分[42]等。Zhang 等人[31]通过对图像进行灰度化，将灰度图片作为输入、原图作为标签训练网络正确的为其进行上色，此过程中就需要网络学到相应的视觉特征。Pathak 等人[32]通过使图像中的某一块区域随机缺失，然后训练网络对缺失部分进行预测、修补，只有网络能够理解图片代表的含义并学习到相应的颜色、结构等视觉特征，才能正确的完成该任务。Ledig 等人[42]通过将低分辨率的图像作为输入，训练网络生

成对应的高分辨率图像，来确保网络学习到了其中的视觉特征。

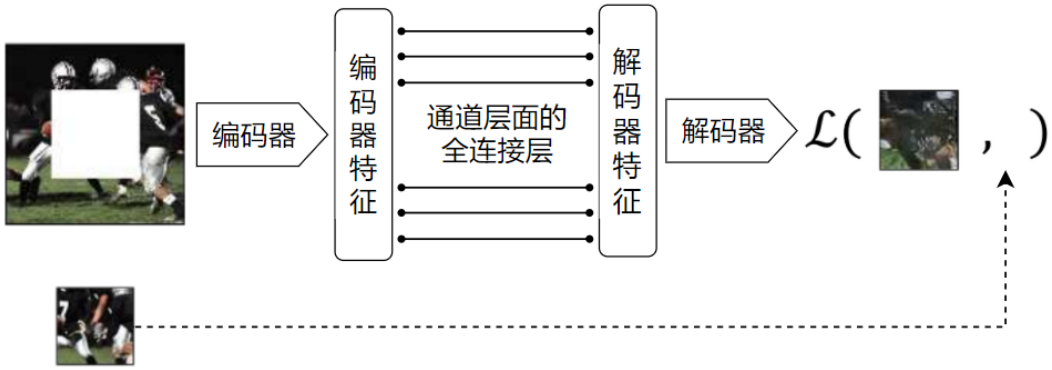

图 2-7 图像修补

Figure 2-7 Image Inpainting

2.4.2 基于上下文的自监督学习

基于上下文的自监督学习是指基于数据上下文来制定前置任务，图像领域主要为空间上下文结构，其他领域中也可为时间上下文结构等。这类方法主要将图像蕴含的丰富空间结构含义作为监督信息，训练网络学习到相应的视觉特征，其关键在于前置任务既不能过于简单(网络无法学到有用的信息)，也不能过于困难(网络的训练过程难以收敛)。Noroozi 等人[29]首先将图像进行切块，并将每一块的相对顺序打乱，然后将打乱前的图像作为标签，训练网络能将图像块恢复成正确的顺序。Gidari 等人[30]提出了另一个简单有效的方法，他们首先将图像旋转某一个的角度，然后将这个旋转角度作为标签，训练模型对其进行预测。

图 2-8 图像拼图[29]

Figure 2-8 Image Jigsaw Puzzle[29]

2.4.3 基于对比的自监督学习

基于对比的自监督学习方法(又称对比学习)，是最近非常流行的一种表征学习方法。对比学习的主要思想是通过“正样本”和“负样本”之间的对比，使网络学习到数据之间的相似性与差异性。其中，“正样本”通常定义为来自同一张图片的多个不同数据增强视图，“负样本”则定义为其他完全不同的图片，而对比的思想体现在希望正样本之间的相似度要远大于其与负样本之间的相似度：

$$\text{score}\big(f(x), f(x^+)\big) \gg \text{score}\big(f(x), f(x^-)\big) \tag{2-8}$$

一般是通过在训练过程中使用对比损失来实现，目前主要是基于 Oord 等人[43]提出的 InfoNCE 损失：

$$\mathcal{L}_N = -\mathbb{E}_X\left[\log\frac{\exp\big(f(x)^T f(x^+)\big)}{\exp\big(f(x)^T f(x^+)\big) + \sum_{j=1}^{N-1}\exp\Big(f(x)^T f\big(x_j\big)\Big)}\right] \tag{2-9}$$

上式中，分子表示正样本之间的相似度，分母表示正负样本之间的相似度，后续许多优秀的对比学习工作大多是基于此进行的。

He 等人[11]提出的 MoCo 将对比学习看作一个构建动态字典的过程，引入了动量更新和队列存储的思想，增加了编码的一致性并使负样本的数量不再局限于训练批次的大小。Chen 等人[12]提出了一个基于对比的自监督学习框架 SimCLR，他们通过大量的实验确定了一组最优的数据增强组合，并提出了在对比损失之前使用一个多层感知机作为映射器等技巧。Chen 等人[13]将 SimCLR 中的一些技巧用于 MoCo 中提出了 MoCo v2，显著的提高了其性能。Grill 等人[14]不再显式地使用负样本进行对比，提出将图像的不同视图分别输入到在线网络和目标网络中，并训练在线网络预测目标网络所输出的特征表示。

本文的主要工作将使用 SimCLR 和 MoCo v2 作为自监督学习的主干框架，因此在后续章节中将会对它们做进一步的介绍。

2.5 本章小结(Chapter Summary)

本章首先介绍了神经网络的基本概念，随后进一步引出其在面对对抗样本攻击时的脆弱性；然后从对抗样本攻击和对抗样本防御两方面进行阐述，分别介绍了当前主流的几种攻击方法和防御方法；最后介绍了自监督学习的相关方法以及一些常见的前置任务，为后续章节设计基于自监督学习的对抗样本防御框架提供了相应的理论支持。

3 基于 SimCLR 的对抗样本防御方法

3 Adversarial Defense Based on SimCLR

本章首先介绍了自监督学习框架 SimCLR，然后在 Kim 等人[44]的基础上实现了基于 SimCLR 的自监督学习对抗样本防御方法，称其为 AdvSimCLR。最后介绍了相关的评估指标并在 CIFAR-10 数据集上进行实验，探究该方法在面对多种对抗攻击时的鲁棒性与准确率，为第四章中方案的设计提供了基础。

3.1 SimCLR 介绍(The Introduction of SimCLR)

SimCLR 是由 Chen 等人[13]提出的一个基于对比的自监督学习框架，他们对最近的对比学习方法进行了总结和简化，然后通过大量实验分析了不同部分对自监督学习性能的影响，并最终给出了这个优秀的框架，这也成为后续许多工作的基础之一，其简化的架构图如 3-1 所示：

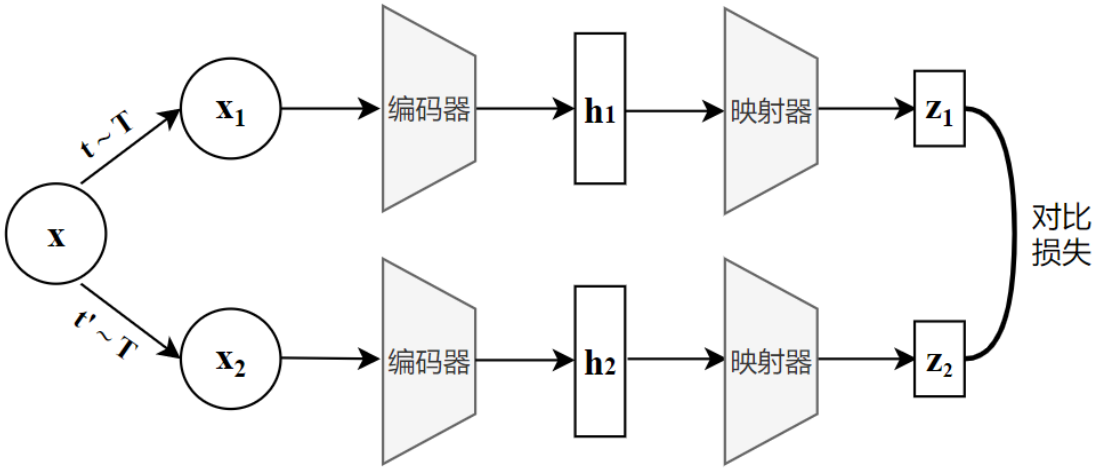

图 3-1 SimCLR 架构图

Figure 3-1SimCLRarchitecture diagram

SimCLR 的主要思想是在隐空间中最大化同一样本的不同视图之间的相似性，最小化不同样本之间的相似性。具体的，首先将原始图像x经过两次随机的数据增强$t(x)$和$t'(x)$，其中包含随机剪裁、随机翻转、随机颜色变化，得到原始图像的两个不同视图x_1和x_2作为正样本对，用$\{x_{pos}\}$统一表示，而同一个 batch 中其他不同图像则作为负样本，用$\{x_{neg}\}$统一表示；然后再将两个视图的图像分别输入到编码器(Encoder)中，得到的隐向量h_1和h_2，编码器可以使用常见的神经网络架构，如 ResNet 等。传统的对比学习会将这里得到的h_1和h_2直接输入到对比损失中进行计算，而 SimCLR 则提出在编码器和对比损失之间加上一个可学习的多层感知机作为映射器(Projector)，再将输出得到的 128 维隐向量z_1和z_2输入到对比损失中会有更好的性能。

在自监督学习中，常用的对比损失函数$\mathcal{L}_{con}$可以定义为(3-1)式：

$$\begin{aligned}&\mathcal{L}_{\text{con},\theta,\pi}\left(x,\{x_{\text{pos}}\},\{x_{\text{neg}}\}\right)\\&:=-\log\frac{\sum_{\{z_{\text{pos}}\}}\exp\left(\text{sim}(z,\{z_{pos}\})/\tau\right)}{\sum_{\{z_{pos}\}}\exp\left(\text{sim}(z,\{z_{pos}\})/\tau\right)+\sum_{\{z_{\text{neg}}\}}\exp\left(\text{sim}(z,\{z_{neg}\})/\tau\right)}\end{aligned}\qquad(3-1)$$

上式中，标准的对比学习框架里$\{x_{pos}\}$只包含一个正样本，即原始样本x的另一个随机增强视图，它们之间组成正样本对；负样本$\{x_{neg}\}$是训练过程中同批次下除了正样本之外的其他所有样本，其数量取决于训练批次的大小(batch size)；$z,\{z_{pos}\},\{z_{neg}\}$分别表示图像$x,\{x_{pos}\},\{x_{neg}\}$经过编码后得到的 128 维隐向量；而 $\text{sim}\ (u,v)=u^{T}v/\|\ u\ \|\|\ v\ \|$ 则表示向量之间的余弦相似度(τ为温度参数)，用来衡量不同样本对之间的相似性。

结合 SimCLR 中所提出的方法，正样本对由同一张图片x经过两次随机数据增强$t(x)$和$t'(x)$后的不同视图组成，公式(3-1)中的隐向量z则由数据增强后的图片分别经过编码器和映射器后得到的，因此可以将 SimCLR 的损失函数写作$\mathcal{L}_{\text{con},\theta,\pi}\left(t(x),\{\text{t}'(\text{x})\},\{t(x)_{\text{neg}}\}\right)$，其中前两项表示正样本对，第三项表示负样本集合。

3.2 方案设计(Scheme Design)

在对抗样本防御方法中，对抗训练被认为是最有效且真正能够提升模型鲁棒性的方法，因此考虑将对抗训练的过程融入 SimCLR 框架中，以达到对抗性的自监督训练过程，本设计中将这个基于 SimCLR 的对抗性自监督学习框架称为 AdvSimCLR。

根据第二章的介绍可知，对抗攻击的主要目标是通过某些方法生成一种能使目标模型损失函数值增大的对抗扰动，而对抗训练直观上就是在训练过程中将针对目标模型生成的对抗样本加入到训练集中，从而使模型在训练时就能将这些可能出现的对抗扰动的损失值降到最低，这通常被定义为一个内部最大化和外部最小化的问题：

$$\underset{\theta}{\text{argmin}}\mathbb{E}_{(x,y)\sim\mathbb{D}}\left[\underset{\delta\in B(x,\epsilon)}{max}\mathcal{L}_{\text{CE}}(\theta,x+\delta,y)\right]\qquad(3-2)$$

在上式中，内部最小化问题可以看成对抗样本的生成过程，外部最大化则表示用对抗样本训练目标模型。可以注意到，在标准的对抗训练框架中，包括 PGD[15] 和 TRADES[39]，通常使用的是交叉熵损失函数$\mathcal{L}_{CE}$，其中需要数据标签y，因此不能直接将传统的对抗训练方法应用于自监督学习(不能使用标签)的过程中。

为了解决上面的问题，可以从对抗攻击和对抗训练的原理入手。所有对抗攻击本质上都是进行一个与目标模型原始训练过程相反的操作，最终目的是为了让目标模型出错，也就需要目标模型的损失函数在输入对抗样本时输出一个较大的

值。在自监督学习过程中，损失函数使用的是(3-1)式中的对比损失，要对针对它进行对抗攻击也就是最大化其损失函数值，将原本“最大化正样本之间相似度”的训练目标替换为“最小化正样本之间相似度”；而对抗训练同样只需要把这样生成的对抗样本加入到训练集中对模型进行对训练即可。具体实现上，这种可以用于自监督学习中的对抗训练定义下：

$$\underset{\theta,\pi}{\operatorname{argmin}}\mathbb{E}_{(x)\sim\mathbb{D}}\left[\underset{\delta\in B(t(x),\epsilon)}{max}\mathcal{L}_{\text{con},\theta,\pi}\left(t(x)+\delta,\{t'(x)\},\{t(x)_{\text{neg}}\}\right)\right] \qquad (3-3)$$

在(3-3)式中，内部最大化问题将(3-2)式中的交叉熵损失替换为了对比损失进行来对抗样本的生成，并是在正样本对的其中一个数据增强视图中加上对抗扰动构成对抗样本；而对抗样本的生成过程则使用 2.2.2 节中介绍的 PGD 攻击的思路进行迭代生成，其迭代过程如(3-4)式，因此在本设计中将此方法称作基于对比的 PGD 攻击(Contrastive PGD)：

$$t(x)^{i+1}=\Pi_{B(t(x),\epsilon)}\left(t(x)^{i}+\alpha\text{sign}\left(\nabla_{t(x)^{i}}\mathcal{L}_{\text{con},\theta,\pi}\left(t(x)^{i},\{t'(x)\},\{t(x)_{\text{neg}}\}\right)\right)\right) \qquad (3-4)$$

在解决了针对自监督学习(对比学习)条件下的对抗样本生成问题后，现在考虑在 SimCLR 框架下对抗性的训练自监督学习模型。根据 SimCLR 的思想，正常训练时的目标为最大化来自同一样本不同视图之间的相似度，而这里的不同视图是通过数据增强来实现的，因此可以将对抗样本也看作另一次的数据增强，由此产生原始样本的一个对抗性新视图。这也就意味着现在的每一个原始样本有与其对应的三种视图，即两次随机数据增强的样本以及在其中一个数据增的强样本上使用 Contrastive PGD 生成的对抗样本，如图 3-2 所示。

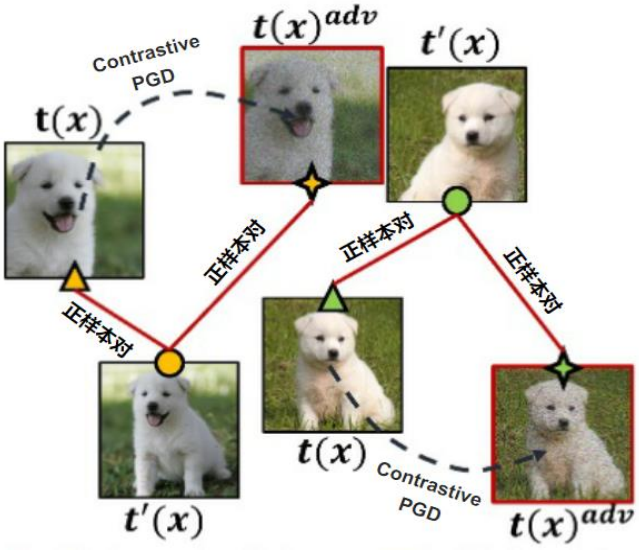

图 3-2 基于对比的 PGD 攻击

Figure 3-2 Contrastive PGD attack

根据上述的思路在图 3-1 的 SimCLR 架构上进行改进，仍然基于对比的思想，最终目的是使图 3-2 中原始样本的三个视图间相似度都尽可能的大，与此同时它

们三者与其他样本的相似度要尽可能的小，改进后的 AdvSimCLR 架构如图 3-3 所示。首先将原始样本x进行两次随机的数据增强得到两个不同的视图$x_1 = t(x)$和$x_2 = t'(x)$；然后在其中一个随机增强视图中使用前面介绍的 Contrastive PGD 方法生成对抗样本。另外，由于这里每次都是先对原始样本进行随机的数据增强，然后在得到的增强视图上生成对抗样本，相较于直接在原始样本上构造，这种方式可以得到更加多样化的对抗样本；在得到原始样本的三个视图之后，将它们依次输入编码器和映射器中得到 128 维的隐向量z_i；最后将上面得到的三个视图看作互为正样本，输入到对比损失中进行计算。

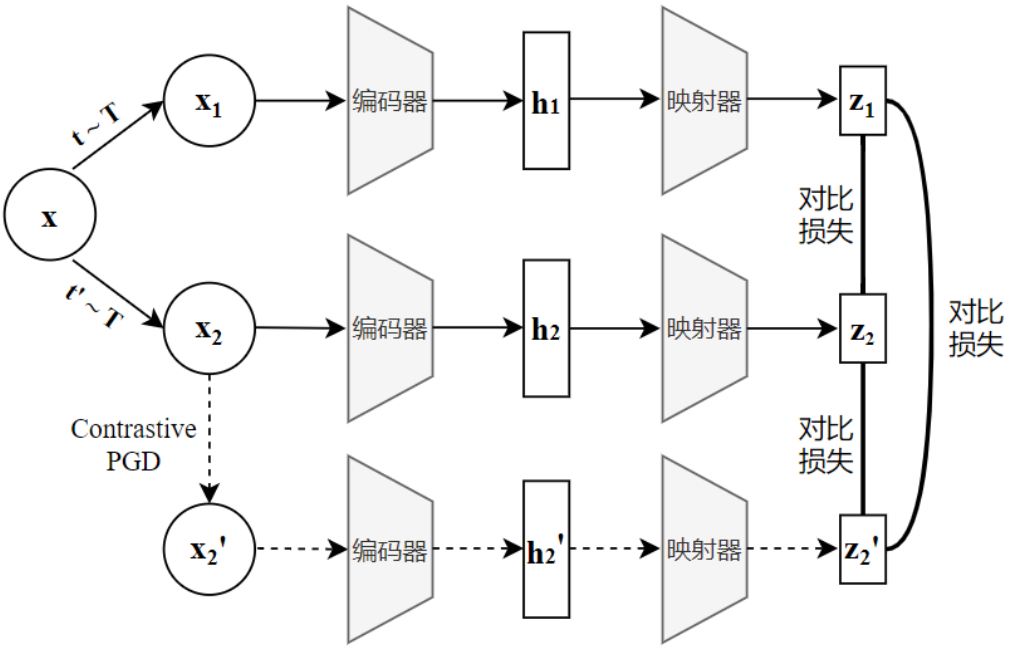

图 3-3 AdvSimCLR 架构图

Figure 3-3 AdvSimCLR architecture diagram

因此，AdvSimCLR 的思路可以总结如下：首先通过 Contrastive PGD 在不需要标签的情况下生成对抗样本，然后根据对比学习的思想最大化干净样本与对抗样本之间的相似性，因为它们均来自于同一个原始样本，且均需要先进行随机的数据增强。具体的，在(3-1)式的基础上可以写出 AdvSimCLR 的损失函数如 3-5 式所示：

$$\begin{aligned}\mathcal{L}_{\text{AdvSimCLR},\theta,\pi} &\coloneqq \mathcal{L}_{\text{con},\theta,\pi}\left(t(x),\{t'(x),t(x)^{adv}\},\{t(x)_{\text{neg}}\}\right)\\ \mathcal{L}_{\text{total}} &\coloneqq \mathcal{L}_{\text{AdvSimCLR},\theta,\pi} + \lambda\mathcal{L}_{\text{con},\theta,\pi}\left(t(x)^{adv},\{t'(x)\},\{t(x)_{\text{neg}}\}\right)\end{aligned} \qquad (3-5)$$

上式中，$t(x)$和$t'(x)$是原始样本x经过两次随机数据增强得到的副本；$t(x)^{adv}$为根据(3-3)式使用 Contrastive PGD 生成的对抗样本；λ为正则化参数。从该损失函数中可以看出其本质就是最大化同一个图片的不同视图间相似度，与图 3-3 相符合。

3.3 实验与分析(Experiment and Analysis)

如何评估深度学习模型所学习到的数据特征的质量，一直是研究们在不断探索的一个问题。而在自监督学习领域，由于模型并没有通过标签进行训练，因此并不能直接将其应用到下游的图像分类等任务中。所以在进行实验前，首先需要介绍一下相关的评估指标，这些评估指标也将同样在第四章中使用。

在自监督学习领域，普遍使用的评估方法称为“线性评估(Linear Evaluate)”，其主要思想是将使用自监督学习的方法训练的编码器(即图 3-1 中编码器)冻结参数并作为一个特征提取器，然后在特征提取器后面直接加上一个全连接层(线性层)，其作用是完成从输出的特征维度到数据类别数的一个映射，最后将这个整体在下游数据集上进行训练，但此时只更新全连接层的参数。这样就可以根据在下游图像分类任务上的准确率来判断此编码器的质量，从而可以判断自监督学习方法的好坏。在本设计中，同样使用了线性评估作为测试方案有效性的评价指标之一。

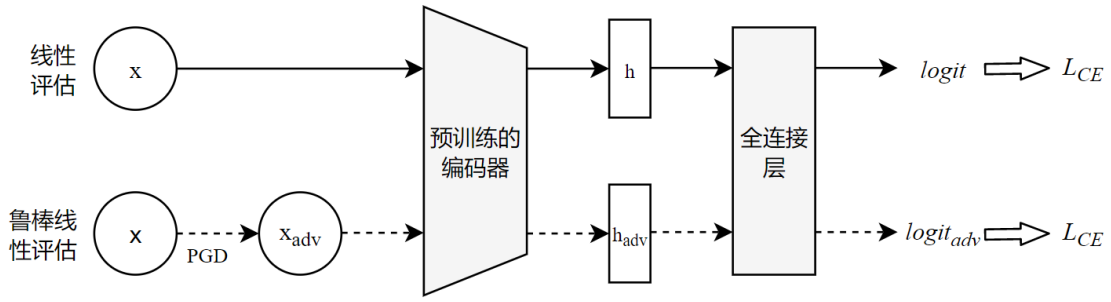

图 3-4 线性评估和鲁棒线性评估

Figure 3-4 Linear Evaluate and Robust Linear Evaluate

由于本设计着重考虑的模型鲁棒性指标，因此除了标准的线性评估之外，同时也使用了另一种改进的评估方法，称为“鲁棒线性评估(Robust Linear Evaluate)”。该方法在线性评估的基础上将自然训练替换成标准的对抗训练，仍然只是更新增加的全连接层的参数，而将编码器的参数冻结。鲁棒线性评估的定义如式(3-6)，其中L_{CE}为交叉熵损失函数，ψ表示全连接线性层的参数。实际上，这种方法不仅可以作为评估方法，同时也可以看作一种对自监督学习的鲁棒性增强方法。

$$\underset{\psi}{\operatorname{argmin}}\mathbb{E}_{(x,y)\sim\mathbb{D}}\left[\underset{\delta\in B(x,\epsilon)}{max}\mathcal{L}_{\mathrm{CE}}(\psi,x+\delta,y)\right] \qquad (3-6)$$

上面的两种评估方法中的特征提取器均将自监督预训练后的编码器直接进行使用且不再进行参数的更新，通常只用于性能的评估。如果要在下游任务上实际应用自监督预训练模型，则还需要进行具体的微调(Fine Tune)。在微调时，仍然像评估时那样在预训练的编码器后直接加一个线性的全连接层，但是在训练具体下游任务(如图像分类)时，不再冻结编码器的参数，而是更新整个网络。要注意的是，虽然这里是更新整个模型的参数，但是模型的主体部分，也就是特征提

取器的参数并不是随机初始化的，而是已经训练过了的，因此微调相较于从头训练只需要迭代很少的 epoch 就能使模型收敛，大大降低了下游任务的计算开销。除此之外，实验中在微调时使用对抗样本替代原始数据集进行对抗性微调，提高模型的鲁棒性。在本设计中也将对抗性微调后的准确率作为评价方案在实际下游任务中性能的指标之一。

在介绍完相关的评价指标后，接下来将进行具体的实验来对 SimCLR 和改进后的 AdvSimCLR 进行评估。具体实验中，将使用 ResNet18 和一个两层的多层感知机分别作为自监督学习模型中的编码器和映射器，输出维度为 128 维；数据集为 CIFAR-10，包含 10 类共 50000 张训练图片和 10000 张测试图片；在 512 的批次大小(batch size)下预训练了 1000 个时期(epoch)；预训练时对抗样本的生成使用 Contrastive PGD，在ℓ_∞限制下最大扰动值为$\epsilon = 0.0314$并进行 10 次迭代。最后对训练后的编码器在分类任务上进行准确率和鲁棒性的评估。

在表 3-1 中，主要进行评估了 SimCLR 在面对白盒无穷范数攻击时的鲁棒性，其中第一列的$\mathcal{A}_{nat}$表示在干净样本上的分类准确率，后面四列表示对使用白盒攻击生成的对抗样本的分类准确率(鲁棒准确率)。这里使用了 PGD 和 DeepFool 进行评估，使用的参数如下：

(1) PGD：无穷范数约束，最大扰动强度ϵ为 8/255 和 16/255，迭代次数 20 步，步长设置为为 0.1 倍的ϵ。

(2) DeepFool：无穷范数约束，最大扰动强度ϵ为 8/255 和 16/255，迭代次数 50 步，步长设置为为 0.02 倍的ϵ。

表 3-1 AdvSimCLR 在白盒对抗攻击(ℓ_∞)下分类准确率(%)

Table 3-1 Classification accuracy of AdvSimCLR under white-box(ℓ_∞) adversarial attack(%)

训练方法	评估方法	$\mathcal{A}_{nat}$	ℓ_∞，$\epsilon = 8/255$		ℓ_∞，$\epsilon = 16/255$	
			PGD	DeepFool	PGD	DeepFool
SimCLR	线性评估	90.37	0.07	0.01	0.01	0.00
AdvSimCLR	线性评估	82.86	40.48	45.14	10.18	20.17
	鲁棒线性评估	79.21	47.66	48.45	16.13	24.52

结果表明，传统的 SimCLR 虽然能够在干净样本上获得不错准确率，但是在对抗样本面前却非常脆弱，即使在扰动大小仅为 8/255 的限制下也几乎完全丧失了准确率。而改进后的 AdvSimCLR 在同样的线性评估条件下，虽然干净样本准确率有所降低，但是在 PGD 和 DeepFool 攻击下鲁棒准确率都有了大幅度的提升。除了标准的线性评估之外，在使用前面介绍的鲁棒线性评估时，AdvSimCLR 的鲁棒性还能进一步的提升，将 PGD 攻击下的鲁棒准确率分别提高了 7.18%和 5.95%，而干净样本的准确率只降低了 3%左右。

除了直接评估编码器的质量之外，还使用 CIFAR-10 数据集在分类任务上对

模型进行了对抗性微调。在对抗性微调时，使用了使用标准的无穷范数 PGD 攻击，最大扰动限制为$\epsilon = 8/255$进行 10 次迭代来生成基于类标签的对抗样本，结果如表 3-2 所示。

表 3-2 AdvSimCLR 对抗性微调后的白盒对抗攻击(ℓ_∞)下分类准确率(%)

Table 3-2 Classification accuracy of AdvSimCLR after adversarial fine-tune under white-box(ℓ_∞) adversarial attack(%)

训练方法	评估方法	$\mathcal{A}_{nat}$	ℓ_∞，$\epsilon = 8/255$		ℓ_∞，$\epsilon = 16/255$	
			PGD	DeepFool	PGD	DeepFool
AdvSimCLR	对抗性微调	81.02	50.27	49.97	19.37	24.96

结果表明，对抗性微调确实能够有效的提高模型的鲁棒性，与鲁棒线性评估相比在 PGD 攻击下准确率分别提高 2.61%和 3.24%，并且在干净样本上的准确率还有所提升；而与线性评估相比，在干净样本准确率几乎没有下降的情况下，大幅提高了模型的鲁棒性。

在评估了 AdvSimCLR 对于ℓ_∞攻击的鲁棒性之后，同样也需要评估其在ℓ_1和ℓ_2攻击面前的鲁棒性。评估方式与评估指标与表 3-1 中相同，但是在鲁棒性测试中将无穷范数攻击的替换如下：

(1) PGD-L_1：一范数约束，最大扰动值为$\epsilon = 12$，迭代次数为 50 步。

(2) PGD-L_2：二范数约束，最大扰动值为$\epsilon = 0.5$，迭代次数为 50 步。

(3) C&W：二范数约束，最大扰动值为$\epsilon = 0.5$，迭代次数为 100 步。

表 3-3 AdvSimCLR 在白盒对抗攻击(ℓ_1/ℓ_2)下分类准确率(%)

Table 3-3 Classification accuracy of AdvSimCLR under white-box(ℓ_1/ℓ_2) adversarial attack(%)

训练方法	评估方法	$\mathcal{A}_{nat}$	ℓ_1，$\epsilon = 12$	ℓ_2，$\epsilon = 0.5$	
			PGD	PGD	C&W
SimCLR	线性评估	90.37	4.83	0.44	43.41
AdvSimCLR	线性评估	82.86	60.19	54.58	74.55
	鲁棒线性评估	79.21	63.18	59.10	71.54
	对抗性微调	81.02	64.22	62.54	70.79

实验结果见表 3-3，对于ℓ_1攻击而言，AdvSimCLR 所体现出来的鲁棒性结果与ℓ_∞类似，均较标准 SimCLR 有了显著提升。对于ℓ_2攻击，使用 PGD 时的结果与前面类似，但使用 C&W 攻击时，线性评估反而是三种评估方式中鲁棒性最好的，使用对抗样本的鲁棒性线性评估和对抗性微调会降低鲁棒性。这可能是因为实验中使用的是ℓ_∞范数的 PGD 攻击来生成对抗样本进行对抗性评估和微调，因此对ℓ_∞攻击的鲁棒性会更好。

AutoAttack 下的鲁棒准确率是一种最近较为流行的模型鲁棒性评估指标，它将多种互补的攻击方法结合来形成了一个集成攻击方法，在 2.2.5 节中进行了详

细的介绍。因此这里也使用了 AutoAttack 对模型的鲁棒性进行了评估，实验结果见表 3-4。

表 3-4 AdvSimCLR 自动攻击(ℓ_∞)下分类准确率(%)

Table 3-4 Classification accuracy of AdvSimCLR under AutoAttack(ℓ_∞) (%)

训练方法	评估方法	ℓ_∞，$\epsilon = 8/255$
		AutoAttack
SimCLR	线性评估	0.00
AdvSimCLR	线性评估	24.57
	鲁棒线性评估	29.71
	对抗性微调	45.12

结果表明，标准的 SimCLR 在 AutoAttack 前没有任何鲁棒性，而使用线性评估和鲁棒线性评估均能获得百分之二十以上的提高；而在使用对抗性微调时，鲁棒性的提升非常显著，由此判断对抗性的训练可能是面对 AutoAttack 时较为有效的对抗防御方法。

3.4 本章小结(Chapter Summary)

本章首先介绍了 SimCLR 的基本概念及主要框架，解释了其中所使用的对比损失函数。然后，介绍了一种针对自监督对比学习的对抗样本生成方法 Contrastive PGD，并基于此引出了基于 SimCLR 的对抗样本防御方法 AdvSimCLR，通过给出的架构图对该防御方法进行详细的说明。最后，介绍了几种评估指标，并在 CIFAR-10 数据集上进行实验探究，验证了此防御方法对提升模型鲁棒性有着显著的效果。

4 基于 MoCo 的对抗样本防御方法

4 Adversarial Defense Based on MoCo

本章首先介绍了自监督学习框架 MoCo，并比较分析了 MoCo 相较于第三章中 SimCLR 的优势，并受此启发设计了一种基于 MoCo 的自监督学习对抗样本攻击防御方法，最后通过在 CIFAR-10 数据集上进行实验，探究其在面对多种对抗攻击时的鲁棒性，证明方法的有效性。

4.1 MoCo 介绍(The Introduction of MoCo)

MoCo 由 He 等人[11]提出的一种基于对比的自监督学习框架，他们重新思考了对比学习的含义，将注意力完全集中于对比损失的理解，提出了一个新的观点。作者认为可以将对比学习的过程看成一个构建动态字典的过程，即训练一个编码器能够将数据集中每个样本映射成一个固定维度的向量，该向量作为样本的 key 组成一个字典。在查询时用该编码器将目标样本进行编码作为 query，此时在字典中和它相同语义的 key 应该能够与之匹配，不同语义的 key 则不能匹配。这样理解后，作者认为对比学习的性能受益于两点：1)更大的字典，即在对比损失中使用更多的负样本。2)编码 key 的编码器需要在过程中保持高度的一致性。

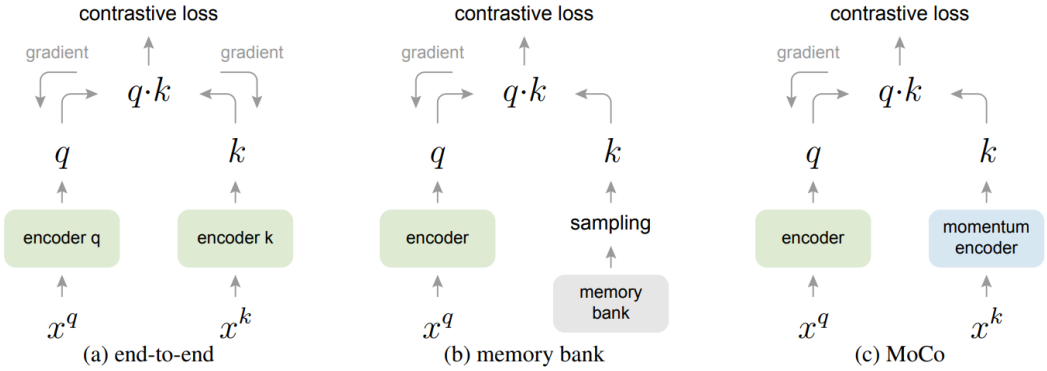

图 4-1 三种对比损失机制的概念比较[11]

Figure 4-1 Conceptual comparison of three contrastive loss mechanisms[11]

作者将标准的对比学习方法用图 4-1 的(a)表示，这种方法直接使用当前批次中的样本作为字典，因此字典中的 key 均是用相同参数的编码器所编码的，能够保证高度一致性，但是字典的大小(负样本的数量)取决于每一个批大小，这样就需要很大的 GPU 计算资源才能达到较好的效果。于是图 4-1 中的(b)是引入了 memory bank 的概念，也就是提前整个数据集中的样本进行编码组成 memory bank，然后每个批次中的字典是从 memory bank 随机取样，不需要反向传播，也就支持更大的字典。但是，memory bank 中的 key 会在最后一次使用时进行更新，而此时的编码器已经更新过参数，无法保证编码的一致性。

为了解决这两个问题，MoCo 提出了两个关键的改进，如图 4-1 中的(c)所示：

(1) 使用队列作为字典：将字典保持为一个先进先出的队列进行动态维护，在训练过程中将最新批次的样本入队，并将最先前批次的样本出队。队列的大小是一个认为指定的超参数，这样可以使字典的大小不再受限于批次大小，使对比的过程可以见到更多的负样本。

(2) 动量更新：使用两个编码器f_q和f_k分别编码 query 和 key，但是只通过反向传播更新f_q的参数，而f_k则以动量的方式使用f_q的参数来进行更新。更新方式见式(4-1)，其中θ_q和θ_k分别表示两个编码器的参数，m表示动量参数，在 MoCo 中设置为 0.999，也就意味着f_k的更新非常缓慢，因此可以尽可能的保持队列中 key 的编码一致性：

$$\theta_k \leftarrow m\theta_k + (1-m)\theta_q \tag{4-1}$$

图 4-1 中使用的对比损失(contrastive loss)本质上与(3-1)式相同，均是基于 InfoNCE[43]，定义如下：

$$\mathcal{L}_{\mathrm{NCE}} = -\log \frac{\exp(q \cdot k^{+}/\tau)}{\exp(q \cdot k^{+}/\tau) + \sum_{k^{-} \in \mathcal{M}} \exp(q \cdot k^{-}/\tau)} \tag{4-2}$$

在上式中，q与k^{+}为正样本对，k^{-}表示从字典队列$\mathcal{M}$中取样的负样本，τ为温度参数，并且是通过点积来计算样本间的相似度。

实际上，MoCo 是比 SimCLR 更早提出的，所以作者将编码器输出的特征直接用于对比损失中，也并没有研究不同的数据增强对模型的影响，导致其在性能上并不如后来的 SimCLR。而在这之后，MoCo 团队借鉴了其中的数据增强组合，并同样在编码器和对比损失直接加入了一个映射器，将改进后的方法称为 MoCo v2，其性能则要优于标准的 SimCLR，并且由于 MoCo 框架解耦了负样本数量和批大小，所以并不需要像 SimCLR 那样使用大量的计算资源。在本设计中，后续使用的 MoCo 框架均指 MoCo v2，其架构如图 4-2 所示：

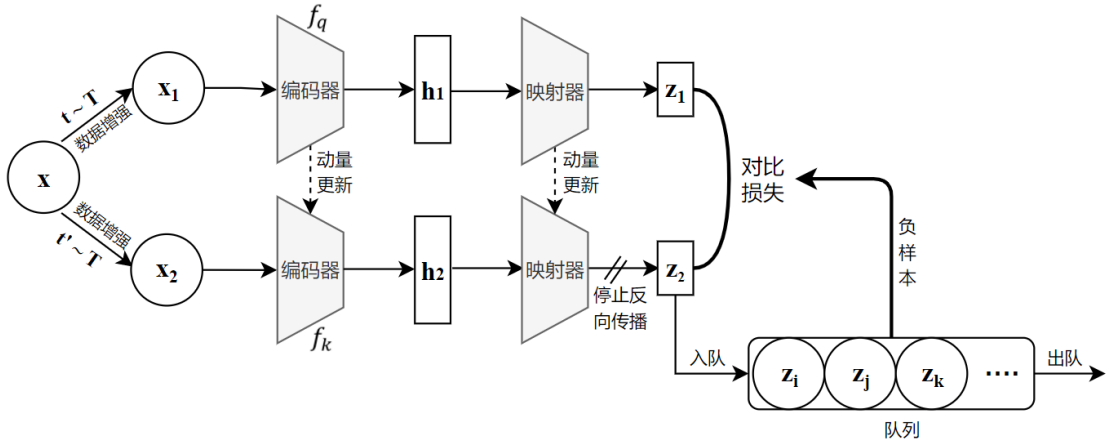

图 4-2 MoCo v2 架构图

Figure 4-2MoCo v2 architecture diagram

4.2 方案设计(Scheme Design)

在 3.2 节中介绍了基于 SimCLR 的对抗样本防御方法 AdvSimCLR，但是受限于 SimCLR 框架本身的缺点，其负样本的数量仍然依赖于训练时的批大小。对比学习的性能好坏很大程度上受益于负样本的数量，因此 AdvSimCLR 要想获得更好的效果，也就需要大量的计算资源。受到这一点的启发，本节开始考虑利用 MoCo 框架的优势设计一种基于 MoCo 的对抗样本防御方法 AdvMoCo。借鉴 AdvSimCLR 中的思想，将对抗样本的生成作为一次数据增强，从而产生原始样本的对抗性视图，对 MoCo 中标准的对比损失进行扩展，其中对抗样本仍然使用 3.2 节中介绍的 Contrastive PGD 方法来生成。

除此之外，MoCo 的两个编码器f_q和f_k并不是完全相同的(动量更新)，并且在训练过程中使用一个全局维护的队列存储 key，然后在对比损失中将整个队列中的样本作为负样本。所以在此基础上又引入了一个对抗性的队列用于存储 Contrastive PGD 生成的对抗性 key。

也就是说，现在在图 4-2 标准框架的基础上，AdvMoCo 防御框架拥有x的四个视图，即两次随机数据增强视图$t(x)_{clean}$、$t'(x)_{clean}$，以及它们再分别经过对抗样本生成后的对抗性视图$t(x)_{adv}$、$t'(x)_{adv}$。除此之外，还动态维护了两个队列分别用于存储对抗性负样本和普通干净负样本，用$\mathcal{M}_{adv}$和$\mathcal{M}_{clean}$来表示。如图 4-3 所示，其中的上分支与图 4-2 相同，表示标准的 MoCo 框架，下分支则是由 Contrastive PGD 产生的对抗性视图构成。图中上下分支的f_q和f_k(包括映射器)实际上为同一个，作图时将其进行了重复。

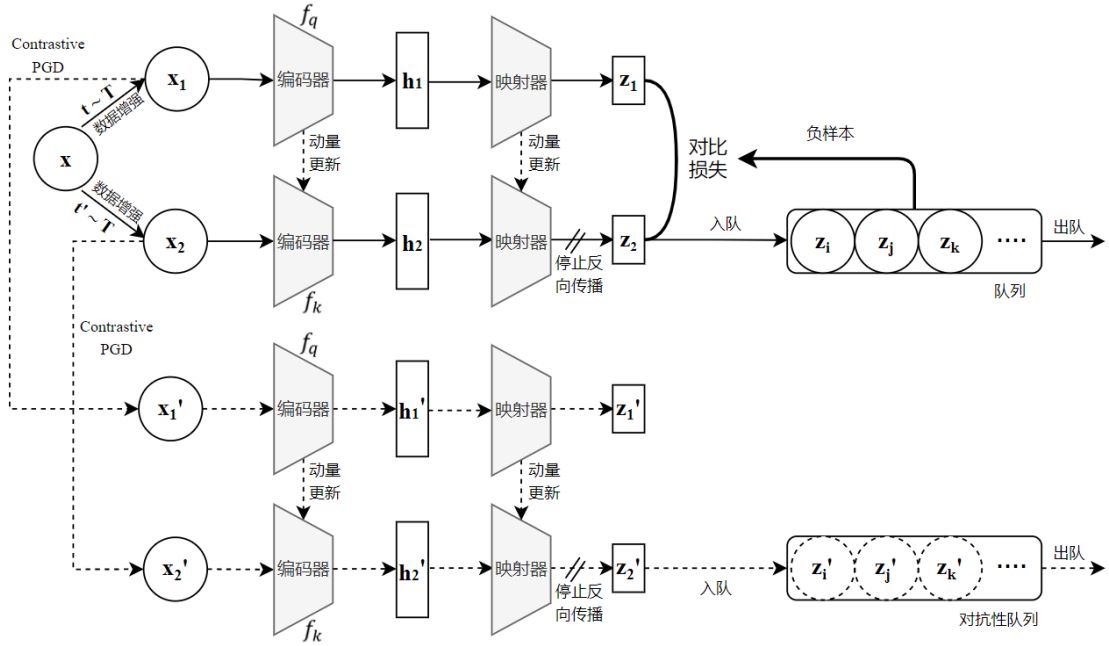

图 4-2 对抗性 MoCo 基础架构图

Figure 4-2 Adversarial MoCo base architecture diagram

另外，为了后续表述方便，可以将标准的 MoCo 损失函数定义为(4-3)式：

$$\mathcal{L}_{MoCo} = \mathcal{L}_{NCE}\left(f_q(t(x)_{clean}), f_k(t'(x)_{clean}), \mathcal{M}_{\text{clean}}\right) \qquad (4-3)$$

上式中，$\mathcal{L}_{NCE}$为(4-2)式中的对比损失，$f_q(t(x))$和$f_k(t'(x))$表示正样本对分别经过两个编码器和映射器后得到的隐向量，$\mathcal{M}_{\text{clean}}$表示存储干净负样本的字典队列。最后，就是如何利用图 4-2 这四个视图(正样本)和两个存储队列(负样本)的组合来进行对比学习，通过实验尝试最终确定了四种可行的组合方式，接下来将依次进行介绍。

首先直接根据 AdvSimCLR 中的思路，直接在标准 MoCo 的损失函数中，再加入一项对比损失用来最大化数据增强视图与对抗性视图之间的相似性，可以将这种方法称作 AdvMoCo-ACC(简称 ACC)，如下所示：

$$\mathcal{L}_{ACC} = \mathcal{L}_{MoCo} + \mathcal{L}_{NCE}\left(f_q(t(x)_{adv}), f_k(t'(x)_{clean}), \mathcal{M}_{\text{clean}}\right) \qquad (4-4)$$

也就是这里用到了三个正样本视图，但是负样本均是使用干净样本的存储队列$\mathcal{M}_{clean}$，其架构如图 4-3 所示，其中第一个对比损失为标准的$\mathcal{L}_{MoCo}$，第二个对比损失则为 ACC 所表示的含义。

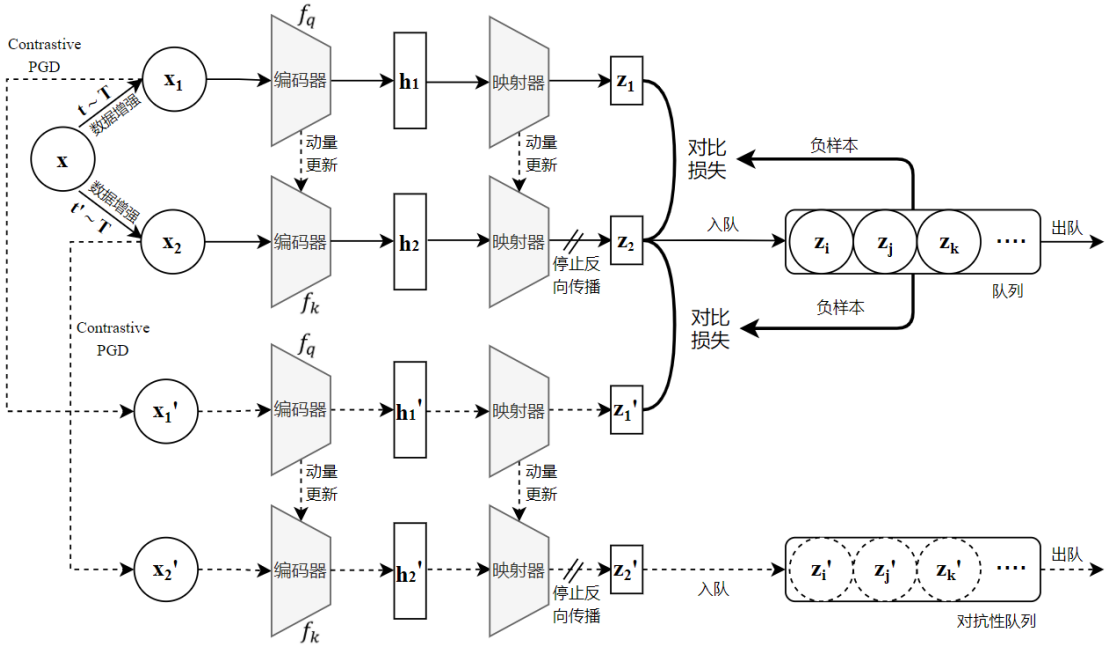

图 4-3 AdvMoCo-ACC 架构图

Figure 4-3 AdvMoCo-ACC architecture diagram

在 ACC 的基础上，考虑将第二个对比损失的$\mathcal{M}_{clean}$替换为$\mathcal{M}_{adv}$，也就是将存储对抗样本的字典队列作为负样本，提出 AdvMoCo-ACA(简称 ACA)，如下所示：

$$\mathcal{L}_{ACA} = \mathcal{L}_{MoCo} + \mathcal{L}_{NCE}\left(f_q(t(x)_{adv}), f_k(t'(x)_{clean}), \mathcal{M}_{\text{adv}}\right) \qquad (4-5)$$

ACA 与 ACC 的区别仅仅是在第二个对比损失中将用对抗性存储队列作为负

样本，通过实验也证明了这样做能够一定程度上提高鲁棒性。

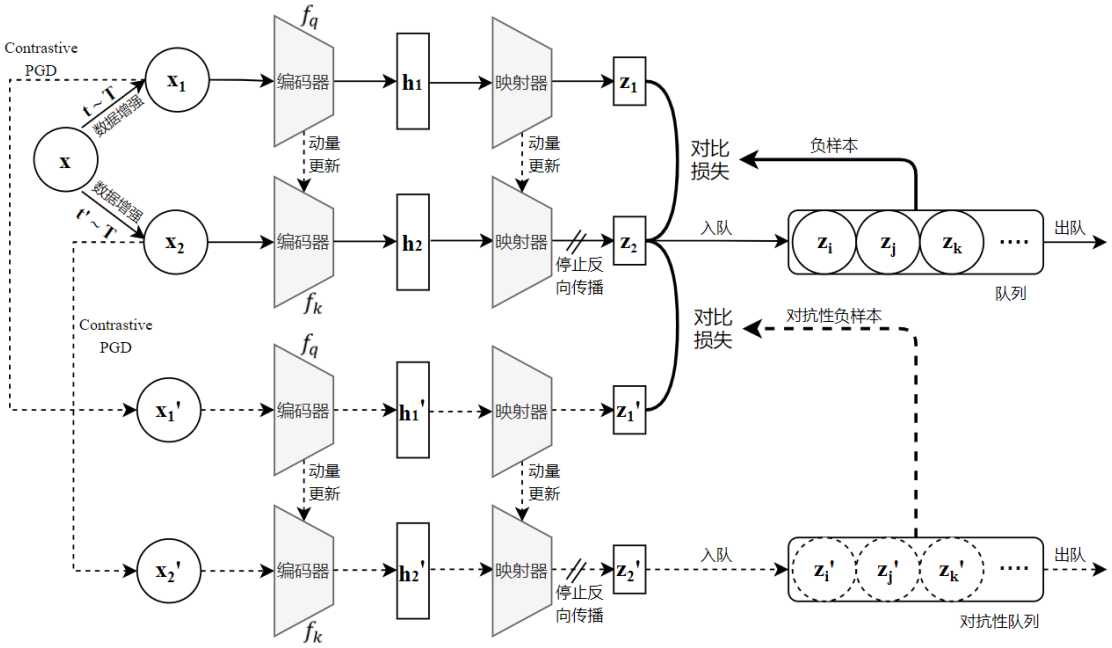

图 4-4 AdvMoCo-ACA 架构图

Figure 4-4 AdvMoCo-ACA architecture diagram

在实验过程中，同样尝试了在第二个对比损失中最大化 key 的对抗性视图与 query 干净视图间的相似性，也就是 CAC 和 CAA，但是发现这样做并不会带来鲁棒性的提升。经过分析，推测是因为f_k并不会反向传播更新参数，而是复制f_q的参数，而且最终也是使用f_q作为下游任务的特征提取器。因此仅仅在 f_k的输入中加入对抗扰动进行训练，并不能得到更加鲁棒的自监督模型。

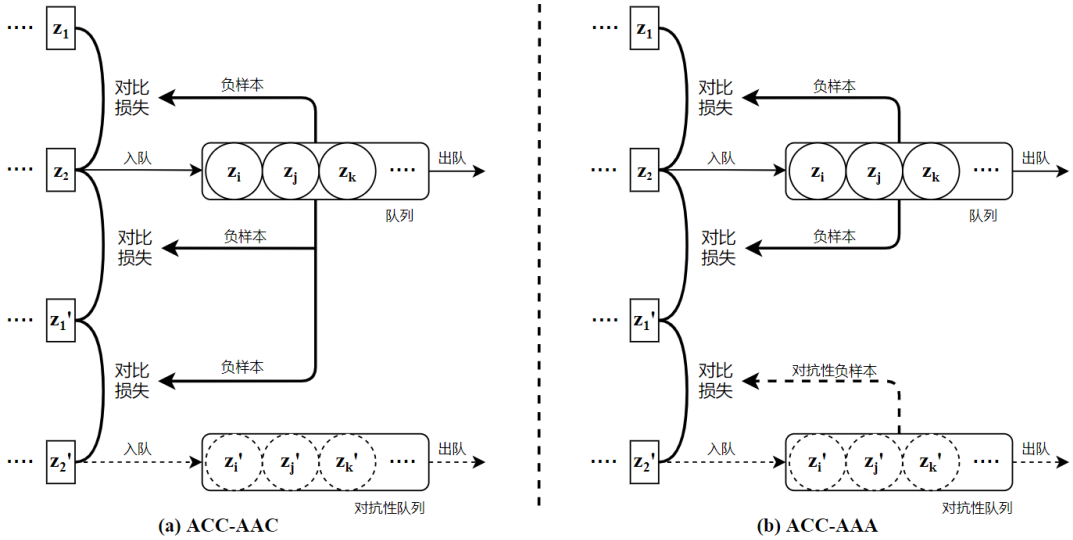

图 4-5 ACC-AAC 与 ACC-AAA 架构图

Figure 4-5 ACC-AAC and ACC-AAA architecture diagram

在 ACC 和 ACA 获得良好的效果之后，进一步拓展该防御方法，在$\mathcal{L}_{ACC}$的基础上，尝试再加入一个对比损失来最大化两个对抗样本视图之间的相似性，该对比损失所使用的负样本队列同样也具有$\mathcal{M}_{clean}$和$\mathcal{M}_{adv}$两种选择，因此设计了

AdvMoCo-ACC-AAC(简称 ACC-AAC)和 AdvMoCo-ACC-AAA(简称 ACC-AAA)，如下：

$$\mathcal{L}_{ACC-AAC} = \mathcal{L}_{ACC} + \mathcal{L}_{NCE}\left(f_q(t(x)_{adv}), f_k(t'(x)_{adv}), \mathcal{M}_{\text{clean}}\right) \quad (4-6)$$

$$\mathcal{L}_{ACC-AAA} = \mathcal{L}_{ACC} + \mathcal{L}_{NCE}\left(f_q(t(x)_{adv}), f_k(t'(x)_{adv}), \mathcal{M}_{\text{adv}}\right) \quad (4-7)$$

这两种方法的架构如图 4-5 所示，其中左半部分与前面类似，主要区别在于最后损失函数的设计部分。如(4-6)式和(4-7)式所示，最终的损失函数包含了三个对比损失，并且正样本的四个视图以及负样本的两个存储队列均进行了使用。

本文也根据同样的思想基于 ACA 进行了改进，即 AdvMoCo-ACA-ACC(简称 ACC-AAC)和 AdvMoCo-ACA-AAA(简称 ACC-AAA)，其架构图如图 4-6 所示，但是实验证明了其整体效果并不如 ACC-AAC，故不再进行赘述。

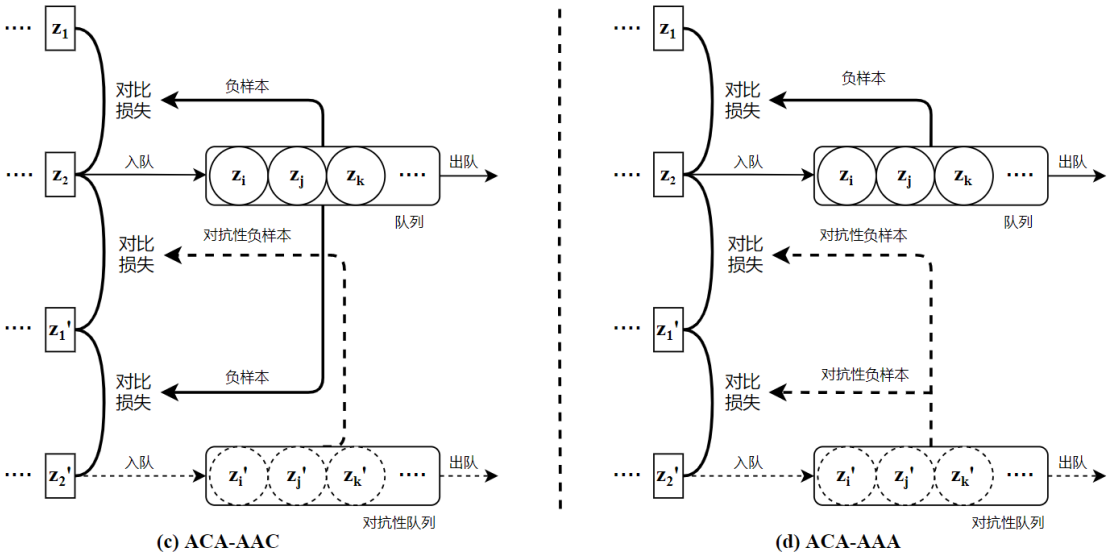

图 4-6 ACA-AAC 与 ACA-AAA 架构图

Figure 4-6 ACA-AAC and ACA-AAA architecture diagram

4.3 实验与分析(Experiment and Analysis)

本节将通过实验来对上一节中提出的 AdvMoCo 防御方法进行评估。数据集仍然使用 CIFAR-10，并将 ResNet18 作为自监督学习框架的主干网络。评估方式与第三章保持一致，主要使用了线性评估、鲁棒线性评估和对抗性微调三种方式(见 3.3 节中介绍)。在预训练过程中，对抗样本使用的是 3.2 节中提到的针对对比损失的 Contrastive PGD，在$\epsilon = 8/255$的扰动限制下，进行了 5 次迭代；而在对抗性评估和对抗性微调时，使用的均为标准的 PGD 攻击，在同样的扰动限制下，进行了 10 次迭代，并且分别训练了 25 和 40 个 epoch。具体的评价指标则包括该方法在各评估方式下的干净样本准确率$\mathcal{A}_{nat}$和面对多种对抗攻击(具体参数见表 4-1 时的鲁棒准确率，因为这里主要考虑模型本身的鲁棒性，所以评估时使

用的均为白盒攻击。

表 4-1 用于评估的 $\ell_\infty,\ell_1,\ell_2$ 对抗攻击设置

Table 4-1 The setup for adversarial attacks in $\ell_\infty,\ell_1,\ell_2$

	ℓ_∞，$\epsilon=8/255$			ℓ_1，$\epsilon=12$	ℓ_2，$\epsilon=0.5$	
	PGD	DeepFool	AutoAttack	PGD	PGD	C&W
迭代次数	20	50	-	50	50	100
步长	0.0031	0.02	-	1.2	0.05	0.1

首先，在白盒场景的ℓ_∞攻击下对上一节中提出的方法进行评估，并与有监督的对抗训练[15]、TRADES[39]等进行比较，结果如表 4-2 所示。传统的监督学习训练和标准的 MoCo 自监督学习所得到的模型均并不具有鲁棒性，而所提出的 AdvMoCo 四种备选方案都能表现出较好的鲁棒性。具体的分析如下：

表 4-2 AdvMoCo 在白盒对抗攻击(ℓ_∞)下分类准确率(%)

Table 4-2 Classification accuracy of AdvMoCo under white-box(ℓ_∞) adversarial attack(%)

训练方法	评估方法	$\mathcal{A}_{nat}$	ℓ_∞，$\epsilon=8/255$		
			PGD	DeepFool	AutoAttack
监督学习	-	94.91	0.00	6.85	0.00
对抗训练	-	85.58	46.64	51.73	42.42
TRADES	-	80.02	50.65	50.29	45.92
MoCo	线性评估	90.96	0.29	3.00	0.00
ACC	线性评估	86.60	44.48	48.38	36.90
ACA		86.21	45.18	48.49	37.87
ACC-AAC		84.85	46.51	49.66	39.73
ACC-AAA		84.76	43.95	47.46	37.61
ACC	鲁棒线性评估	84.46	50.73	50.97	40.43
ACA		84.26	50.82	50.74	41.11
ACC-AAC		82.77	51.51	51.26	42.77
ACC-AAA		81.58	49.75	49.71	41.25
ACC	对抗性微调	83.35	52.67	52.56	47.33
ACA		83.40	52.80	52.84	47.70
ACC-AAC		83.32	52.99	53.11	47.98
ACC-AAA		84.03	**53.29**	**53.34**	**47.99**

(1) 在线性评估下，ACC-AAC 在面对 PGD 攻击时的准确率只比对抗训练低了 0.13%，在干净样本上的准确率也只低了 0.73%；而 ACA 在干净样本上的准确率比对抗训练提高了 0.63%了，而 PGD 鲁棒准确率值降低了 1.46%。但是 AutoAttack 的结果表明，对抗性训练对防御该攻击还是具有比较明显的效果的。

除此之外，可以发现对抗训练相对于自然的监督训练的干净样本准确率下降了近10%，而提出的防御方法 ACA 相对标准的 MoCo 在线性评估下只下降了 4.36%，并且随着自监督学习技术的发展，其准确率也会有所增加。

(2) 在鲁棒线性评估下，四种方案的 PGD 鲁棒性均优于对抗训练和 TRADES，ACC-AAC 表现出了最好的防御效果，三种攻击下超过了对抗训练，但是干净样本的准确率下降的较多。

(3) 在对抗性微调下，四种方法与对抗训练和 TRADES 相比，在面对 PGD、DeepFool 和 AutoAttack 攻击时都获得了更高的鲁棒准确率，其中效果最好的 ACC-AAA 比对抗训练分别高了 6.65%、1.61%和 2.07%，而干净样本准确率只比对抗训练降低了 1.54%。除此之外，ACC-AAA 在鲁棒准确率全面高于 TRADES 的情况上，干净样本准确率比其高了 4.01%。

(4) 根据上面的结果，可以表明提出的方法确实能够提高模型面对ℓ_∞攻击时的鲁棒性。综合来看，ACA 和 ACA 在线性评估以及鲁棒线性评估下表现最佳，能以较小的代价获得鲁棒的防御模型，且准确率损失不大。若要应用于下游任务时。ACC-AAA 无论是准确率还是鲁棒性上均要领先于其他方法。

表 4-3 AdvMoCo 在白盒对抗攻击(ℓ_1/ℓ_2)下分类准确率(%)

Table 4-3 Classification accuracy of AdvMoCo under white-box(ℓ_1/ℓ_2) adversarial attack(%)

训练方法	评估方法	$\mathcal{A}_{nat}$	ℓ_1，$\epsilon=12$	ℓ_2，$\epsilon=0.5$	
			PGD	PGD	C&W
监督学习	-	94.91	5.29	0.5	44.06
对抗训练	-	85.58	55.43	56.26	71.98
TRADES	-	80.02	56.62	58.97	68.51
MoCo	线性评估	90.96	25.46	6.33	58.52
ACC	线性评估	86.60	65.08	61.59	74.73
ACA		86.21	65.23	61.97	**74.93**
ACC-AAC		84.85	64.93	62.03	73.46
ACC-AAA		84.76	64.23	60.52	73.73
ACC	鲁棒线性评估	84.46	67.37	64.70	73.05
ACA		84.26	**67.52**	**65.12**	73.60
ACC-AAC		82.77	67.07	64.61	72.39
ACC-AAA		81.58	66.25	63.45	71.31
ACC	对抗性微调	83.35	59.09	61.79	72.49
ACA		83.40	59.05	61.58	72.56
ACC-AAC		83.32	59.36	61.96	72.50
ACC-AAA		84.03	59.50	62.46	73.36

然后，在白盒场景的ℓ_1和ℓ_2攻击下对上一节中提出的方法也进行了相同的评估方式，结果如表 4-3 所示。由于ℓ_1和ℓ_2通常被认为是不可见攻击，因此以降低攻击性为代价换取了更高的鲁棒性。具体的分析如下：

(1) 在线性评估下，可以发现标准的 MoCo 自监督学习在 PGD-L_1和 C&W-L_2攻击下相对于传统监督学习的鲁棒性要有所提高。但是，提出的四种方法在ℓ_1和ℓ_2攻击均获得更高的鲁棒准确率，其中 ACC 和 ACA 比对抗训练的干净样本准确率还要高出百分之一左右。

(2) 在鲁棒性线性评估下，提出方法的鲁棒性相对于线性均又有了一定的提升，其中 ACA 的 PGD-L_1和 PGD-L_2更是达到了 67.52%和 65.12%。但是通过对抗性的训练提高鲁棒性的同时，也损失了一部分的干净样本准确率。

(3) 在对抗性微调下，提出方法的鲁棒性虽然还是优于传统的对抗训练与 TRADES，但是相对于线性评估和鲁棒性线性评估却有所下降，这种趋势与ℓ_∞有很大的区别。

(4) 通过上面实验结果，可以发现在面对ℓ_1和ℓ_2攻击时，适当的对抗性训练能够提高模型的鲁棒性，但是过度的将对抗样本加入训练过程中反而又会使其鲁棒性降低。通过一定分析，推测是因为在实验时，用于训练的对抗扰动均是在ℓ_∞下生成的，而它们之间的防御可能并不是通用的。无论如何，提出的防御方法在模型鲁棒性方面仍然要优于传统方法。

最后，将第三章中的 AdvSimCLR 与第四章中效果较好的 AdvMoCo-ACA 和 AdvMoCo-ACC-AAC 进行对比，主要考虑它们在干净样本的准确率和在 PGD 白盒攻击下的准确率，结果见表 4-4，第四章提出的方法全面优于先前的 AdvSimCLR。

表 4-4 AdvSimCLR 与 AdvMoCo 的分类准确率对比(%)

Table 4-4 Comparison of classification accuracy between AdvSimCLR and AdvMoCo(%)

	AdvSimCLR		AdvMoCo-ACA		AdvMoCo-ACC-AAC	
	$\mathcal{A}_{nat}$	PGD-20	$\mathcal{A}_{nat}$	PGD-20	$\mathcal{A}_{nat}$	PGD-20
线性评估	82.86	40.48	**86.21**	45.18	84.74	**46.51**
鲁棒线性评估	79.21	47.66	**84.26**	50.82	82.77	**51.51**
对抗性微调	81.02	50.27	83.40	52.80	**84.03**	**53.29**

4.4 本章小结(Chapter Summary)

本章首先介绍了自监督学习框架 MoCo 的基本概念和主要思想，并分析了它相对于第三章中 SimCLR 框架的优势所在。然后，受到 AdvSimCLR 的启发，设计了一种基于 MoCo 的对抗样本防御方法 AdvMoCo，同时给出了四种效果较好备选方案 ACC、ACA、ACC-AAC 和 ACC-AAA。最后，在 CIFAR-10 数据集上的进行了大量的实验，并将实验结果与对抗训练、TRADES 等传统方法以及前一章的 AdvSimCLR 进行对比，证明了该防御方法的有效性和优越性。

5 防御方法在交通标志识别系统中的应用

5 The Application of Defense Method in Traffic Sign Recognition System

在前面的章节中，介绍了基于自监督学习的对抗样本防御方法，为了展示防御方法在实际场景中的作用，本章开发了一个融合了对抗样本攻防的交通标志识别演示系统。首先介绍了交通标志识别任务中所使用的数据集，然后对提出方法的迁移性进行了一定的实验，最后展示了系统各模块的设计以及页面效果。

5.1 数据集介绍(The Introduction of Dataset)

无人驾驶一直以来都是人们所极力追求的一项技术，而交通标志识别则是其重要组成部分之一。在这样与人身安全紧密相关的领域，深度学习系统的鲁棒性就显得尤为关键，这也正是对抗样本防御工作的实际意义所在。为了验证之前设计的防御方案在实际场景中的应用，本节使用了交通标志数据集 GTSRB 进行了进一步的探究。该数据集收集于德国真实的道路交通标志，共有 43 类标志，由于数据图片是来自于自然环境下的照片拍摄，所以存在大量分辨率低及尺寸不同的情况，为此需要对该数据集进行了一定的预处理，包括将所有图片归一化到与 CIFAR-10 相同的大小。本章使用的 GTSRB 详细数据见表 5-1 以及图 5-1。

表 5-1GTSRB 数据集信息

Table 5-1GTSRB dataset information

数据集名称	训练集	测试集	数据尺寸	类别数
GTSRB	39209	12630	32 × 32 × 3	43

图 5-1 GTSRB 交通标志

Figure 5-1GTSRB traffic signs

5.2 实验与分析(Experiment and Analysis)

本章在第四章的提出的防御方法上，将 CIFAR-10 替换为了 GTSRB 数据集来来进一步的探究方法的实用性。

首先考虑的自监督学习的应用场景，即在不需要数据标签的情况下预训练深度学习模型，然后再通过微调等方式用于具体下游任务。这样做是因为在实际场景应用中，相关数据集的收集会更加的困难，需要更高的成本。因此为了评估提出的方法迁移到下游任务上时的效果，使用之前在 CIFAR-10 数据集上进行预训练的模型作为特征提取器，然后在 GTSRB 数据上进行 40 个 epoch 的对抗性微调。通过尝试发现使用第四章中的 AdvMoCo-ACC 预训练模型进行微调的效果最好。同时，实验中也直接用 GTSRB 数据在 ResNet18 上进行有监督的自然训练和对抗训练用于对比，它们均进行了 200 个 epoch 的训练。在评估时主要考虑在$\epsilon = 8/255$的无穷范数攻击下的鲁棒性，包括 PGD、DeepFool 和 AutoAtack。实验结果如表 5-2 所示。

表 5-2 对抗攻击下 GTSRB 分类准确率(%)

Table 5-2 Classification accuracy ofGTSRB under adversarial attack(%)

训练方法	$\mathcal{A}_{nat}$	l_∞，$\epsilon = 8/255$		
		PGD	DeepFool	AutoAttack
有监督自然训练(GTSRB)	97.15	14.91	34.50	6.00
有监督对抗训练(GTSRB)	89.52	63.38	67.36	58.27
AdvMoCo 预训练(CIFAR-10) +对抗性微调(GTSRB)	90.95	**66.14**	**69.60**	**61.20**

结果表明，有监督自然训练下的模型，虽然能够获得高达 97.15%的分类准确率，但是鲁棒性并不高，尤其是面对 AutoAttack 时，鲁棒准确率骤降到 6%。传统的有监督对抗训练模型虽然能够大幅的提高模型在面对攻击时的鲁棒性，但是却将干净样本的准确率降低到了 89.52%。先在 CIFAR-10 上进行预训练，经过一定的微调后仍然能够达到 90.82%的准确率，虽然这与 GTSRB 本身数据量较少且数据分布较为简单有关。但是基于自监督学习的防御方法即使在 CIFAR-10 数据集上进行的预训练，在经过 40 个 epoch 的对抗性微调后，干净样本准确率达到了 90.82%，虽然这与 GTSRB 数据集的样本数量较少且分布较为简单有关，但与直接使用 GTSRB 进行对抗训练相比，在鲁棒性上也全部有所提升，还是能够凸显基于自监督学习方法的优越性。经过分析，得出原因是自监督学习的预训练过程不依赖任何人工注释的数据标签，所以模型会更多的学习数据本身的信息，从而使其具有较好的迁移性。

5.3 系统设计及展示(System Design and Presentation)

为了更方便的展示防御方法在实际场景中的应用，以交通标志识别为目标任务搭建了一个 Web 端的演示系统。本系统使用了前后端分离的开发思路，前端页面使用 HTML、CSS、Javascript 的进行搭建，并通过 AJAX(Asynchronous JavaScript and XML)技术与后端 API 进行数据交互。后端为了便于直接使用前面实验部分的模型进行推理，选择了基于 Python 的轻量级 Web 框架 Flask 进行 API 的开发，数据库部分同样使用轻量级的 SQLite，配合 flask_sqlalchemy 库可以快速对数据表进行增、删、改、查等操作，并且可以将数据信息以文件的形式直接保存在当前文件夹中便于后续移植。接下来将对该系统的各个模块及功能进行详细的介绍与页面展示。

5.3.1 用户模块

用户模块主要包括用户的注册及登录，其用户信息表设计表 5-3 所示：

表 5-3 用户信息表

Table 5-3 User information table

名称	类型	字段长度	主键	描述
id	INTEGER	11	是	唯一的 id，自动赋值
username	VARCHAR	40	否	用户名
password	VARCHAR	255	否	用户密码

用户模块的主要作用是对系统的访问权限做出限制，未登录的用户，只能访问系统的首页(见图 5-2)，若要使用其他功能则会自动跳转到登录页面。

图 5-2 首页

Figure 5-2 Home page

登陆界面如图 5-3 所示，用户需要依次输入用户名和密码，通过后端查询数

据库判断用户存在且密码正确后，确认登陆成功，也可选择“注册”按钮跳转至用户注册页面。

图 5-3 登录页面

Figure 5-3 Login page

注册页面如图 5-4 所示，需要依次输入用户名和两次密码，后台会判断用户名是否已被注册以及两次密码输入是否一致，并给出相应提示(见图 5-5)。

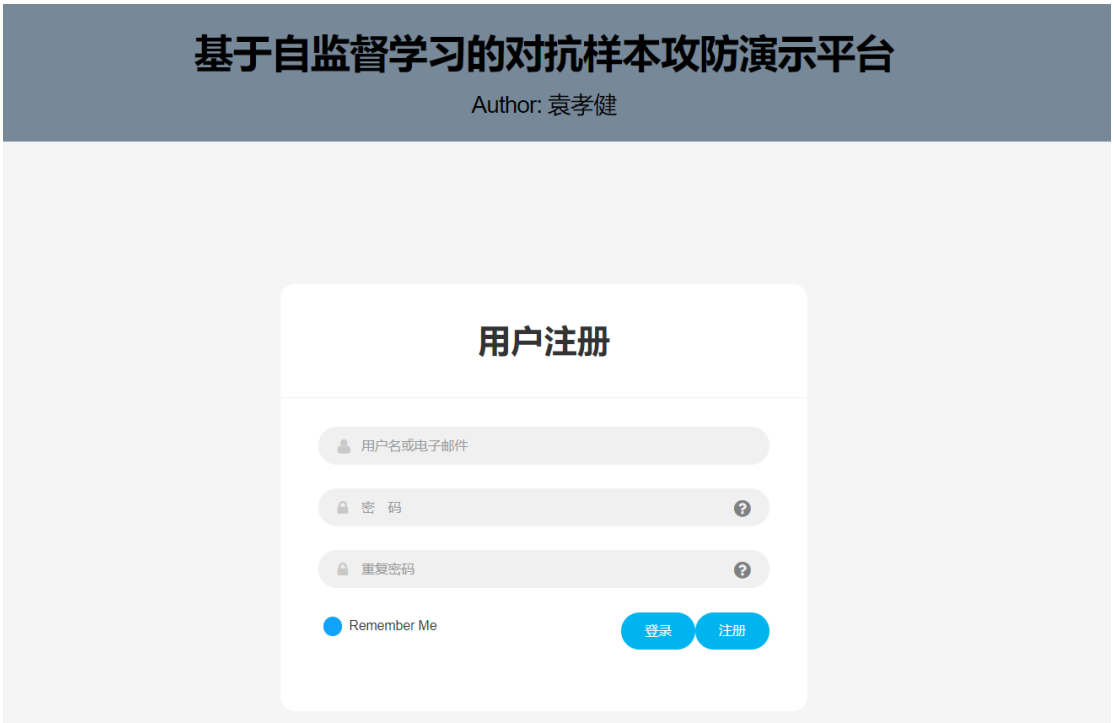

图 5-4 注册页面

Figure 5-4 Register page

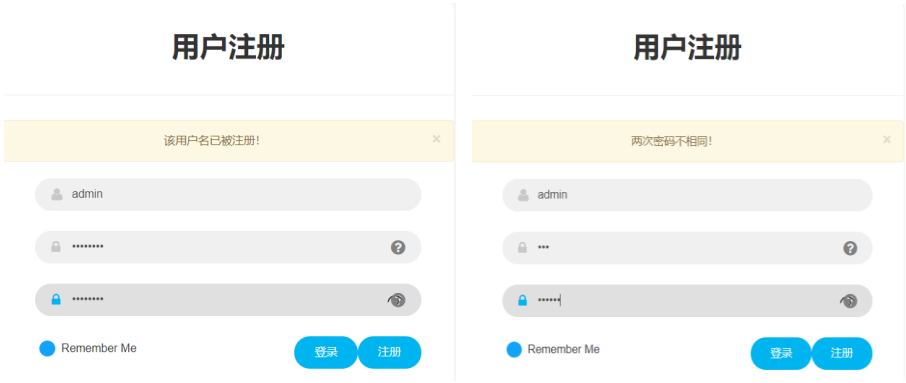

图 5-5 注册提示信息

Figure 5-5 Register prompt message

5.3.2 无目标对抗攻击模块

为了后续更好的演示防御方法在于对抗样本攻击的鲁棒性，该系统中也集成了对抗样本的生成模块，图 5-6 为无目标攻击页面。

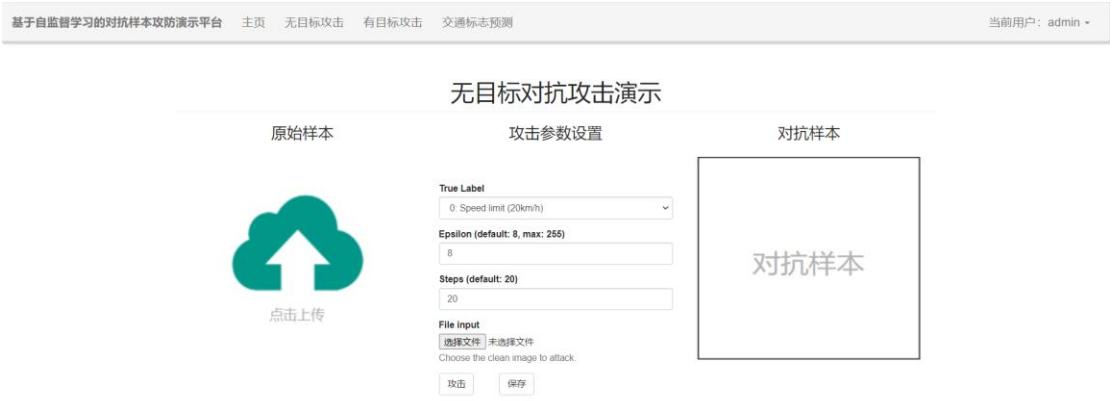

图 5-6 无目标对抗攻击页面

Figure 5-6 Untarget adversarial attack page

图 5-7 生成无目标对抗样本

Figure 5-7 Generate untarget adversarial example

要通过无目标攻击生成对抗样本，首先需要上传一张原始样本，后端通过

PGD 攻击方法在替代模型上构造对抗样本，因此还需要输入必须的攻击参数，包括原始样本的真实标签(从 GTSRB 数据集 43 类中进行选择)、最大扰动幅度(默认为 8)以及迭代的次数(默认为 20)。在上传图片并设置好参数后，点击“攻击”即可在右侧显示生成的对抗样本，通过点击“保存”按钮可以下载该对抗样本来进行后续使用。无目标对抗样本的生成效果如图 5-7 所示。

5.3.3 有目标对抗攻击模块

除了无目标攻击模块外，系统也实现了有目标对抗攻击模块，如图 5-8 所示。

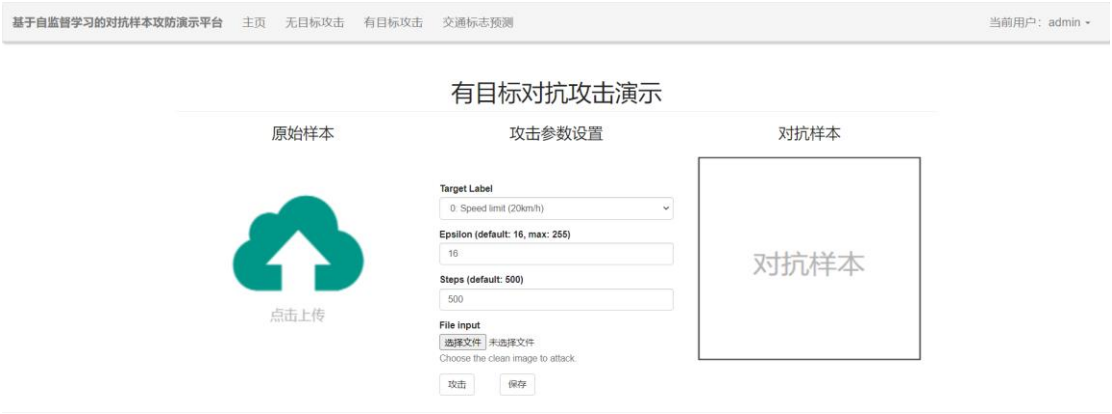

图 5-8 有目标对抗攻击页面

Figure 5-8 Target adversarial attack page

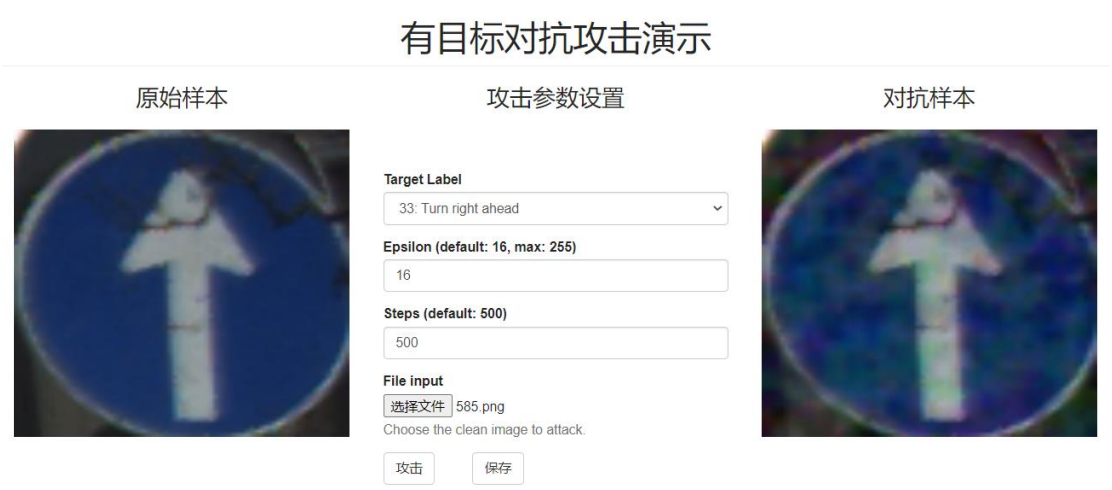

图 5-9 生成有目标对抗样本

Figure 5-9 Generate target adversarial example

在本系统中同样使用 PGD 攻击替代模型来进行有目标对抗样本的生成，如图 5-9 所示。首先需要上传一张原始样本，然后设置相应的攻击参数，由于相同方法下，有目标攻击比无目标攻击要困难许多，因此默认的最大扰动值为 16，默认的迭代次数为 500，并且此时设置的标签并不是原始样本的类别，而是要攻击的目标类别。在图 5-9 中将类别为“限直行(Ahead only)”的样本攻击为“前面右转(Turn right ahead)”目标类别，由于两个类别的样本较为相似，因此有目标攻击

成功率会相对较高，并且可以看到此时生成的对抗样本上已经出现较为明显的对抗扰动痕迹。

5.3.4 交通标志识别模块

该模块是对用户上传的交通标志进行识别，并给出概率最高的五个类别以及对应的置信度，其中包含了自然训练下的“普通路牌识别系统”和根据第四章中提出的方法得到的“鲁棒路牌识别系统”，如图 5-10 所示。

图 5-10 交通标志识别页面

Figure 5-10 Traffic sign recognition page

id	类别	置信度
1	Speed limit (30km/h)	0.9982
2	Speed limit (70km/h)	0.0003
3	Speed limit (60km/h)	0.0002
4	Speed limit (50km/h)	0.0002
5	Speed limit (20km/h)	0.0001

id	类别	置信度
1	Speed limit (20km/h)	0.962
2	Speed limit (30km/h)	0.012
3	Speed limit (60km/h)	0.0058
4	Road narrows on the right	0.0055
5	Slippery road	0.0034

图 5-11 无目标对抗样本分类结果

Figure 5-11 Classification results of untarget adversarial example

我将 5.3.2 节中通过无目标攻击生成的对抗样本保存下来，然后分别上传到

普通路牌识别系统和鲁棒路牌识别系统中，点击“开始识别”即可得到分类结果，如图 5-11 所示。在普通路牌识别系统中，将“限速 20”错误识别成了“限速 30”，并且置信度高达 0.9982，表示对抗攻击成功(该干净样本能够被识别正确)。而在鲁棒路牌识别系统中，该对抗样本仍然被识别为“限速 20”，表示对抗攻击失败，从而证明了防御方法有效性。

5.4 本章小结(Chapter Summary)

本章首先介绍了交通标志识别任务中的常用数据集 GTSRB，然后在该数据集进行实验，证明了本设计中基于自监督学习的对抗样本攻击防御具有良好的迁移性，在准确率、鲁棒性两方面均优于直接使用传统的方法进行训练。最后，为了更好的与实际应用场景相结合，实现了一个以交通标志识别为任务 Web 演示系统，并对用户模块、对抗攻击模块和交通标志识别模块进行了详细的介绍与页面展示，从而也证明了防御方法在实际场景中的有效性与必要性。

6 总结与展望

6 Conclusions and Prospects

6.1 全文总结(Conclusions)

深度学习的快速发展使其在许多领域都取得了巨大的成功，但是对抗样本的发现又指出了深度学习系统的脆弱性问题，这也很大程度上阻碍了其在实际场景中的大规模应用。因此，对抗样本的攻击与防御在近年来成为了一个热点领域，具有很高研究价值。自监督学习是最近越来越流行的一种新的学习范式，旨在利用数据本身的特性作为监督信息，以监督的方式进行无监督学习。其仅需要无标签数据进行训练，就能达到接近监督学习的效果，有望解决海量数据收集所带来的高昂成本问题，也是人工智能未来的发展趋势之一。

但是，通过调研国内外相关工作及研究现状，发现很少有工作考虑了自监督学习与模型鲁棒性之间的关联。SimCLR、MoCo 等表现优秀的自监督学习模型并不具备任何对抗鲁棒性，仍然很容易受到对抗样本的欺骗。与此同时，能否利用自监督学习的思想摆脱传统对抗防御(如对抗训练)对数据标签的依赖也是一个值得思考的问题。

本课题正是基于上述问题进行研究，将自监督学习与对抗训练的思想相结合，研究了基于自监督学习的对抗样本攻击防御技术。最后，本文的主要工作如下：

(1) 调研了对抗样本和自监督学习相关研究，介绍了多种对抗样本攻击、防御技术以及自监督学习最新进展，并详细阐述了自监督学习框架 SimCLR 和 MoCo 的主要思想。

(2) 对 Kim 等人[44]的工作进行重新梳理，再现了基于 SimCLR 的对抗样本防御方法。并在此过程中，探究了在自监督学习场景下的对抗样本生成方法，并分析了其有效的原因。

(3) 通过对自监督学习和对抗样本攻防领域的理解，给出了线性评估、鲁棒线性评估和鲁棒性微调三种方法，用于对本文中基于自监督的对抗样本防御方法进行评估。

(4) 分析了 MoCo 框架相对于 SimCLR 的优势，并受此启发设计了一种基于 MoCo 的对抗样本防御方法，并通过实验的对比和分析，证明了所提出方法在准确率和对抗鲁棒性上均优于传统方法和基于 SimCLR 的方法。

(5) 将本设计中的防御方法与交通标志识别任务相结合，开发了一个基于 Flask 的对抗攻防演示系统。

6.2 未来展望(Prospects)

本文将对抗训练的思想与自监督学习相结合，提出了基于自监督学习的对抗样本防御方法，并取得了很好的效果，但是目前关于这方面的研究还较少，仍然存在许多问题需要解决。这里对本文的改进和可能的发展方向进行简要阐述：

(1) 本文提出的防御方法虽然基于自监督学习，但在实验过程中使用的还是基准数据集 CIFAR-10 进行评估。理论上，在实际应用中可以使用任意多的无标签数据对提出的方法进行预训练，从而进一步获取更高得鲁棒性。

(2) 在第四章设计基于 MoCo 的自监督学习方法时，通过多次的实验和尝试，从结果的角度确定了最佳的组合方式，但不同组合之间差别的理论解释仍然有待探究。

参考文献

[1] 王科俊,赵彦东,邢向磊.深度学习在无人驾驶汽车领域应用的研究进展[J].智能系统学报, 2018, 13(01): 55-69.

[2] 田娟秀,刘国才,谷珊珊,鞠忠建,刘劲光,顾冬冬.医学图像分析深度学习方法研究与挑战[J].自动化学报, 2018, 44(03): 401-424.

[3] 景晨凯,宋涛,庄雷,刘刚,王乐,刘凯伦.基于深度卷积神经网络的人脸识别技术综述[J].计算机应用与软件, 2018, 35(01): 223-231.

[4] Szegedy C, Zaremba W, Sutskever I, et al. Intriguing properties of neural networks[J]. arXiv preprint arXiv:1312.6199, 2013.

[5] Goodfellow I J, Shlens J, Szegedy C. Explaining and harnessing adversarial examples[J]. arXiv preprint arXiv:1412.6572, 2014.

[6] 张思思,左信,刘建伟.深度学习中的对抗样本问题[J].计算机学报, 2019, 42(08): 1886-1904.

[7] Kreuk F, Adi Y, Cisse M, et al. Fooling end-to-end speaker verification with adversarial examples[C]//2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018: 1962-1966.

[8] Moosavi-Dezfooli S M, Fawzi A, Frossard P. Deepfool: a simple and accurate method to fool deep neural networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 2574-2582.

[9] Carlini N, Wagner D. Towards evaluating the robustness of neural networks[C]//2017 ieee symposium on security and privacy (sp). IEEE, 2017: 39-57.

[10] Papernot N, McDaniel P, Goodfellow I. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples[J]. arXiv preprint arXiv:1605.07277, 2016.

[11] He K, Fan H, Wu Y, et al. Momentum contrast for unsupervised visual representation learning[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 9729-9738.

[12] Chen T, Kornblith S, Norouzi M, et al. A simple framework for contrastive learning of visual representations[C]//International conference on machine learning. PMLR, 2020: 1597-1607.

[13] Chen X, Fan H, Girshick R, et al. Improved baselines with momentum contrastive learning[J]. arXiv preprint arXiv:2003.04297, 2020.

[14] Grill J B, Strub F, Altché F, et al. Bootstrap your own latent: A new approach to self-supervised learning[J]. arXiv preprint arXiv:2006.07733, 2020.

[15] Madry A, Makelov A, Schmidt L, et al. Towards deep learning models resistant to adversarial attacks[J]. arXiv preprint arXiv:1706.06083, 2017.

[16] Schmidt L, Santurkar S, Tsipras D, et al. Adversarially robust generalization requires more data[J]. arXiv preprint arXiv:1804.11285, 2018.

[17] Biggio B, Corona I, Maiorca D, et al. Evasion attacks against machine learning at test time[C]//Joint European conference on machine learning and knowledge discovery in databases. Springer, Berlin, Heidelberg, 2013: 387-402.

[18] Zantedeschi V, Nicolae M I, Rawat A. Efficient defenses against adversarial attacks[C]//Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security. 2017: 39-49.

[19] 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). IEEE, 2016: 582-597.

[20] Papernot N, McDaniel P, Goodfellow I, et al. Practical black-box attacks against machine learning[C]//Proceedings of the 2017 ACM on Asia conference on computer and communications security. 2017: 506-519.

[21] Li X, Li F. Adversarial examples detection in deep networks with convolutional filter statistics[C]//Proceedings of the IEEE International Conference on Computer Vision. 2017: 5764-5772.

[22] Xu W, Evans D, Qi Y. Feature squeezing: Detecting adversarial examples in deep neural networks[J]. arXiv preprint arXiv:1704.01155, 2017.

[23] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. Advances in neural information processing systems, 2012, 25: 1097-1105.

[24] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.

[25] 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.

[26] 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.

[27] 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. Ieee, 2009: 248-255.

[28] Kuznetsova A, Rom H, Alldrin N, et al. The open images dataset v4[J]. International Journal of Computer Vision, 2020: 1-26.

[29] Noroozi M, Favaro P. Unsupervised learning of visual representations by solving jigsaw puzzles[C]//European conference on computer vision. Springer, Cham, 2016: 69-84.

[30] Gidaris S, Singh P, Komodakis N. Unsupervised representation learning by predicting image

rotations[J]. arXiv preprint arXiv:1803.07728, 2018.

[31] Zhang R, Isola P, Efros A A. Colorful image colorization[C]//European conference on computer vision. Springer, Cham, 2016: 649-666.

[32] Pathak D, Krahenbuhl P, Donahue J, et al. Context encoders: Feature learning by inpainting[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 2536-2544.

[33] Goodfellow I, Bengio Y, Courville A, et al. Deep learning[M]. Cambridge: MIT press, 2016.

[34] Croce F, Hein M. Minimally distorted adversarial examples with a fast adaptive boundary attack[C]//International Conference on Machine Learning. PMLR, 2020: 2196-2205.

[35] Andriushchenko M, Croce F, Flammarion N, et al. Square attack: a query-efficient black-box adversarial attack via random search[C]//European Conference on Computer Vision. Springer, Cham, 2020: 484-501.

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

[37] Kurakin A, Goodfellow I, Bengio S. Adversarial examples in the physical world[J]. 2016.

[38] Tramèr F, Kurakin A, Papernot N, et al. Ensemble adversarial training: Attacks and defenses[J]. arXiv preprint arXiv:1705.07204, 2017.

[39] Zhang H, Yu Y, Jiao J, et al. Theoretically principled trade-off between robustness and accuracy[C]//International Conference on Machine Learning. PMLR, 2019: 7472-7482.

[40] Osadchy M, Hernandez-Castro J, Gibson S, et al. No bot expects the DeepCAPTCHA! Introducing immutable adversarial examples, with applications to CAPTCHA generation[J]. IEEE Transactions on Information Forensics and Security, 2017, 12(11): 2640-2653.

[41] Liao F, Liang M, Dong Y, et al. Defense against adversarial attacks using high-level representation guided denoiser[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 1778-1787.

[42] Ledig C, Theis L, Huszár F, et al. Photo-realistic single image super-resolution using a generative adversarial network[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 4681-4690.

[43] Oord A, Li Y, Vinyals O. Representation learning with contrastive predictive coding[J]. arXiv preprint arXiv:1807.03748, 2018.

[44] Kim M, Tack J, Hwang S J. Adversarial self-supervised contrastive learning[J]. arXiv preprint arXiv:2006.07589, 2020.