目录

[1 绪论 2](#_Toc37018727)

[1.1研究背景及意义 2](#_Toc37018728)

[1.2国内外研究现状 3](#_Toc37018729)

[1.2.1对抗样本攻击研究现状 4](#_Toc37018730)

[1.2.2 对抗样本防御研究现状 5](#_Toc37018731)

[1.3研究内容及章节安排 7](#_Toc37018732)

[2 相关理论基础 9](#_Toc37018733)

[2.1对抗样本的基本概念 9](#_Toc37018734)

[2.2 对抗样本攻击算法 11](#_Toc37018735)

[2.2.1 FGSM算法 11](#_Toc37018736)

[2.2.2 BIM算法 12](#_Toc37018737)

[2.2.3 Deepfool算法 12](#_Toc37018738)

[2.2.4 C&W算法 13](#_Toc37018739)

[2.3对抗样本防御方法 13](#_Toc37018740)

[2.3.1 对抗训练 14](#_Toc37018741)

[2.3.2 蒸馏防御法 14](#_Toc37018742)

[**2.3.3 图像压缩重建** 15](#_Toc37018743)

[2.4 VAE概述 15](#_Toc37018744)

[2.5 GAN概述 18](#_Toc37018745)

[2.6 本章小结 20](#_Toc37018746)

[3基于VAE-GAN的对抗样本防御方法 21](#_Toc37018747)

[3.1 VAE-GAN防御模型设计 21](#_Toc37018748)

[3.2 VAE-GAN模型网络结构 23](#_Toc37018749)

[3.2.1 编码器网络结构 23](#_Toc37018750)

[3.2.2 生成器网络结构 25](#_Toc37018751)

[3.2.3 判别器网络结构 26](#_Toc37018752)

[3.3 VAE-GAN模型优化目标及训练流程 27](#_Toc37018753)

[3.3.1 模型优化目标 27](#_Toc37018754)

[3.3.2模型训练流程 28](#_Toc37018755)

[3.4 本章小结 30](#_Toc37018756)

[4 实验及结果分析 31](#_Toc37018757)

[4.1实验环境 31](#_Toc37018758)

[4.1.1实验数据集 31](#_Toc37018759)

[4.1.2攻防目标模型 32](#_Toc37018760)

[4.2 白盒实验结果及分析 36](#_Toc37018761)

[4.3 黑盒实验结果及分析 36](#_Toc37018762)

[4.4 去噪实验结果及分析 36](#_Toc37018763)

[4.5 本章小结 36](#_Toc37018764)

# 1 绪论

## 1.1研究背景及意义

近年来深度学习研究取得了巨大进展，使得深度学习在图像识别、自然语言处理、语音识别等领域取得了巨大的成就，尤其在图像识别、语音识别等模式领域，深度学习的分类准确率已经超越了人类。尽管深度学习在解决很多复杂问题时表现优异，但Szegedy等人发现深度神经网络（Deep Neural Networks，DNN）极易受对抗样本的攻击，这些对抗样本只是在原有的图像样本添加轻微的扰动，人类视觉系统几乎无法察觉这些扰动，就可导致基于深度学习的图像分类系统输出错误的分类结果，对抗样本攻击深度学习模型过程如图1-1所示。Kdnuggets等人还指出对抗样本不仅仅存在于计算机视觉领域，也普遍存在于语音识别、恶意软件检测、文本处理等深度学习领域。对抗样本的出现引发人们对深度学习的工作原理的质疑，也使得越来越多的研究人员开始对对抗样本的生成和防御进行研究。随着研究的不断深入，研究者们发现同一对抗样本可攻击不同网络结构的深度学习模型，这使得攻击者无需知道深度学习模型的具体网络结构就可对该模型进行攻击。此外，Sharif等人还发现将对抗样本打印、拍照处理后，对抗样本仍然可以欺骗深度学习图像识别系统，作者的发现证明了物理世界也存在对抗样本。

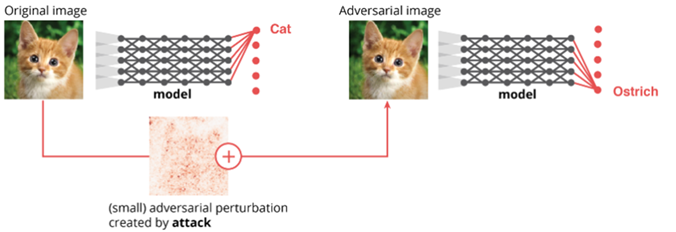


图1-1 对抗样本攻击举例

目前很多深度学习解决方案已经进入人们的日常生活，如人脸识别、自动驾驶、视频检测等，对抗样本的存在给这些解决方案的使用带来了巨大的风险。比如，在人脸识别系统中，恐怖分子可以利用对抗样本冒充他人身份，然后进行恐怖袭击，将造成不可估量的损失。又或者，在自动驾驶过程中，使用对抗样本覆盖真实的路标，车辆自动驾驶系统将无法对该路标做出正确的决策，从而引发严重的交通事故。因此设计安全性要求严苛的深度学习解决方案时，必须考虑如何防御对抗样本的攻击。

对抗样本的存在极大的限制了深度学习解决方案的使用，因此研究如何有效的防御对抗样本攻击具有非凡的现实意义。目前针对对抗样本的攻击，研究人员提出许多不同类型的防御方案，并取得了不错的防御效果，但随着对抗样本的不断升级，防御方案很容易被进化后的对抗样本攻破。另外，现有的防御方案大多是被动式防御，基本上只能防御一种对抗样本。鉴于目前对抗样本防御领域存在的问题，本文提出一种新的防御策略，结合变分自动编码器(Variational Auto-Encoder，VAE)和生成式对抗网络(Generative Adversarial Networks，GAN)，设计并实现了一种专门针对对抗样本去噪模型VAE-GAN，通过去噪模型去除对抗样本中的扰动，以达到防御对抗样本的目的。

## 1.2国内外研究现状

自Szegedy等人提出对抗样本概念以来，研究人员们为了解释对抗样本存在原因提出了许多不同的假设和推理，如模型正则化不足、输入高纬度等。根据这些假设和推理，研究者人员设计并实现多种不同的生成对抗样本算法，并尝试攻击不同的深度学习场景。在大量生成对抗样本算法提出的同时，深度学习应用领域也迎来了巨大的安全挑战，这也使得研究人员根据对抗样本的特点提出了多种不同的对抗样本防御方案。本节接下了将分别介绍目前对抗样本的攻击和防御国内外研究现状。

### 1.2.1对抗样本攻击研究现状

对抗样本概念最早由Szegedy等人于2013年提出，作者发现通过向图像添加经过设计的微小的扰动就可导致DNN输出错的分类结果。作者首先尝试求解让深度学习模型做出误分类的最小扰动方程，但发现求解该问题的算法复杂度太高，于是作者将问题转化为寻找最小损失函数添加项方程，并提出L-BFGS优化算法求解该问题。此外，作者还证明同一对抗样本可以攻击相同训练数据集的不同模型，这说明对抗样本具有迁移性。

为生成能使分类器产生误识别的最小对抗样本扰动，Moosavi-Dezfooli等人提出了一个与前人不同的生成对抗样本算法——Deepfool。该生成对抗样本算法的原理是通过迭代的方式将求解线性分类最小扰动的方法推广到非线性分类问题中。Deepfool产生的对抗样本扰动要小于Szegedy提出的使用L-BFGS生成的对抗样本扰动，攻击性和L-BFGS的攻击性相当，缺点是生成对抗样本的迭代次数要多于L-BFGS算法。

2015年Goodfellow 等人就对抗样本存在原因提出线性假设，作者认为由于目前的深度神经网络大量使用线性激活函数，使得整个模型趋于线性化。作者指出对抗样本的扰动对神经网络的影响会随层数的增加而变大，并称这种现象为扰动累加效应。基于以上两个特性，作者提出一种快速生成对抗样本的算法——快速梯度法(FastGradient Sign Method，FGSM)，该算法的最大特点是可以一步生成对抗样本。

2016年Kurakin 等人对FGSM算法进行改进，将原有的单步生成对抗样本改进为以迭代方式生成对抗样本，提出了基础迭代法(Basic Iterative Method，BIM)、最小可能类法 (Leastlikely Class Method，LCM)、迭代最小可能类法 (Iterative least-likely class method，ILCM) 三种对抗样本生成方法，这些算法产生的对抗样本扰动比FGSM算法更小且攻击性更强，同时这些算法的迭代次数要远少于L-BFGS算法和Deepfool算法。

2016年Carlini和 Wagner提出一种根据修改目标函数的生成对抗样本算法——C&W算法，并通过实验证明该算法生成扰动可以攻破大多数基于目标函数的防御方法，该算法主要缺点是生成对抗样本需要大量迭代，生成一个对抗样本要花很多时间。

2017年Madry 等人对BIM算法进行改进提出了投影梯度下降法(Projected Gradient Descent，PGD)，该算法与BIM算法的主要区别是初始化攻击参数的方式不同，PGD算法选择随机初始化攻击参数，BIM选择使用FGSM结果作为初始化攻击参数。通过实验证明PGD生成的对抗样本在黑盒的条件下依旧能成功攻击深度学习模型。

2017年清华大学 TSAIL 团队 Dong 等人提出了动量迭代法(Momentum Iterative Method，MIM)，该方法通过将动量项整合到BIM迭代过程中，极大提高了BIM的黑盒攻击性。

2018年Bose等人提出一种基于GAN攻击人脸识别系统的方法，该算法将人脸识别系统作为GAN的判别器，通过对抗训练的方式生成对抗样本，通过实验证明该算法能在黑盒的条件下攻破人脸识别系统。

2019年Xie在BIM和MIM的基础上提出了一种基于数据增强的迭代快速梯度法（Momentum Diverse Inputs Iterative FGSM，M-DII-FGSM），该算法利用数据增强解决了传统迭代式生成对抗样本存在过拟合的缺点，极大的提高了传统迭代式生成对抗样本的黑盒攻击能。

自Szegedy等人提出对抗样本概念以来，生成对抗样本的算法在不断的改进，生成对抗样本的代价也越来越低，对抗样本攻击能力也越来越强，这也使得防御对抗样本的难度也越来越大，因此目前的深度学习解决方案依然面临着巨大的安全风险。

### 1.2.2 对抗样本防御研究现状

目前针对对抗样本防御方法主要分为四类：对抗训练、梯度掩蔽、输入重构和检测防御。本小节接下来将根据这四个分类详细介绍目前国内外对抗样本防御现状。

(1)对抗训练

Szegedy等人在提出FGSM算法时也提出在训练深度学习模型中，添加对抗样本辅助模型进行对抗训练，可以有效的提高模型防御对抗样本的能力，同时也可以减轻模型的过拟合程度。该防御方法能够有效的防御训练时使用的攻击算法的攻击，缺点是无法防御其他攻击算法生成的对抗样本。为了解决这个问题，Tramer等人提出了一种基于集成式对抗训练防御方法，该方法选择使用多种不同类型的对抗样本辅助模型进行对抗训练，以提高模型对不同对抗样本防御能力。对抗训练类防御方法优点是可以很好的防御已知攻击算法（对抗训练中使用的对抗算法）生成的对抗样本；缺点是训练需要大量的对抗样本，且无法防御未知攻击算法生成的对抗样本。

(2)梯度掩蔽

虽然生成对抗样本算法有很多，但大部分算法是通过模型的梯度信息来生成对抗样本，因此通过隐藏模型的梯度可以很大程度上增加攻击算法生成对抗样本的难度。目前基于梯度掩蔽的防御方法主要有深度压缩网和蒸馏防御法。基于深度压缩网的防御方法是通过引入一个压缩自编码器的平滑项，使模型对输入变化的敏感性降低，进而达到隐藏梯度的目的。蒸馏防御法主要通过知识蒸馏的训练方式，将一个已训练网络的知识迁移到另一个未训练网络中，以达到隐藏未训练网络的梯度信息。梯度掩蔽类防御方法优点是可以很好的阻止白盒攻击；缺点是需该变深度学习模型的网络结构，同时攻击者可以利用网络结构类似的模型梯度信息进行攻击。

(3)输入重构

Song等人提出基于PixelCNN的对抗样本防御方法，该防御方法防御思路是：通过一个能够将对抗样本恢复成普通样本的生成网络PixelCNN，将对抗样本转化为普通样本后，输入到深度学习模型中，来达到防御对抗样本攻击的目的。Liao等人提出利用高层特征训练降噪器（High-Level Representation Guided Denoiser, HGD）的对抗样本防御方法，该防御方法主要利用原始样本和对抗样本在高层网络特征的不同，来训练专门针对于对抗样本的降噪器。Jia 等人提出基于ComDefend防御对抗样本方法，该防御方法使用一个端到端的图片压缩重构模型去除对抗样本中的对抗扰动，以达到防御对抗样本的目的。输入重构类防御方法主要优点是不需要改变深度学习模型的训练集，也不需要改变深度学习模型的结构，理论上对所有类型的对抗样本都有一定的防御效果；缺点是会影响普通样本的识别准确率。

(4)检测防御

Lu等人提出基于SafetyNet对抗样本检测方法，该方法通过观察ReLU函数的输出情况来鉴别深度学习模型输入的是对抗样本还是普通样本。Pang 等人提出通过最小化反向交叉熵（Reverse Cross-Entropy, RCE）训练深度神经网络，可以深度神经网络学习到对抗样本的特征，进而检测出深度学习模型输入是对抗样本还是不是对抗样本。Hinton等人提出使用胶囊网络（capsule network，CapsNet）也可以检测对抗样本，该方法主要是根据对抗样本在胶囊重构图像时重构误差会远大于普通样本重构误差，来检测样本对抗样本。虽然目前的检测防御类防御策略检测出对抗样本准确率很高，但多数检测类防御方法无法对检测出来的对抗样本做进一步的处理。

尽管有些防御方法在防御部分对抗样本时取得了一定的效果，但也存在很多挑战。一方面，大部分的防御方法只能防御有限的对抗样本，同时随着对抗样本的升级，这些防御策略很容易被对抗样本绕过；另一方面，大部分防御算法的防御成本很高，并且有些防御方法会导致普通样本分类准确率下降。总而言之，面对不断进化的对抗样本，目前的防御研究还有很长的路要走。

## 1.3研究内容及章节安排

尽管目前有很多的对抗样本防御方法，但大部分防御方法都被攻破，同时大部分的防御方法只能防御一种对抗样本的攻击。针对目前对抗样本防御存在的问题，本文提出基于VAE-GAN的对抗样本防御方法，该方法能在几乎不损害图像质量的情况，去除对抗样本的对抗扰动，将对抗样本还原成原始样本，以达到防御对抗样本的目前。本文主要研究内容为：

(1)利用GAN的判别器能够区分对抗样本和普通样本的特性，构建针对对抗样本的降噪模型；

(2)通过设计不同的实验，测试基于VAE-GAN对抗样本防御方法对于各种对抗样本的防御性能。

全文章节安排如下：

第一章简要介绍了对抗样本基本概念和对抗样本给深度学习模型带来的威胁，分析了生成对抗样本算法和对抗样本防御方法的国内外研究现状，最后简单概况了本文的主要研究内容和章节安排。

第二章简要介绍了对抗样本基本概念，详细介绍了目前主流的对抗样本生成方法和对抗样本防御方法。

第三章详细的阐述了本文提出的基于VAE-GAN的对抗样本防御方法的各模块网络结构、优化目标和训练流程。

第四章设计对抗样本攻防实验，测试本文提出的防御方法防御性能。

第五章总结了全文主要研究内容，提出研究的改进方向。

# 2 相关理论基础

## 2.1对抗样本的基本概念

2014年Szegedy等人首次提出对抗样本概念，作者发现深度神经网络极易受到对抗样本的攻击，这些对抗样本只是在原始样本中添加轻微扰动，人眼甚至无法察觉这些扰动(人类能够正确的分类)，就能使得分类器输出错误的分类结果。为了更好的描述对抗样本，作者对对抗样本作如下定义：

假设将图像映射为标签的分类器模型为：

 (2-1)

分类器的损失函数为：

 (2-2)

那么对于一个样本且，其对抗样本生成过程可以表示为一个有界优化问题：

 (2-3)

其中表示对抗扰动，表示对抗样本。作者认为公式(2-3)无法直接求解，于是将公式(2-3)条件等价替换为求解，最终将公式(2-3)转为：

 (2-4)

作者提出可以使用L-BFGS算法求解公式(2-4)的近似解。在文章最后，作者对对抗样本存在原因给出了一个可能的解释(如图2-1所示)：现实世界出现对抗样本的概率很低，因此训练集和测试集中几乎不存在对抗样本或者说训练集无法覆盖所有的样本，而深度神经网络的高度非线性，导致过拟合只学习到非对抗样本的特征，并没有学习到泛化性特征。

如图2-1所示，蓝线为样本的真实决策边界，红线为模型训练得到的决策边界，蓝线和红线围成的区域表示对抗样本存在的空间。可以从图中发现，由于模型的训练集无法覆盖所有的样本，导致模型训练得到的决策边界与样本真实的边界相差较远，最终模型无法学习到区域变成的对抗样本存在的区域。

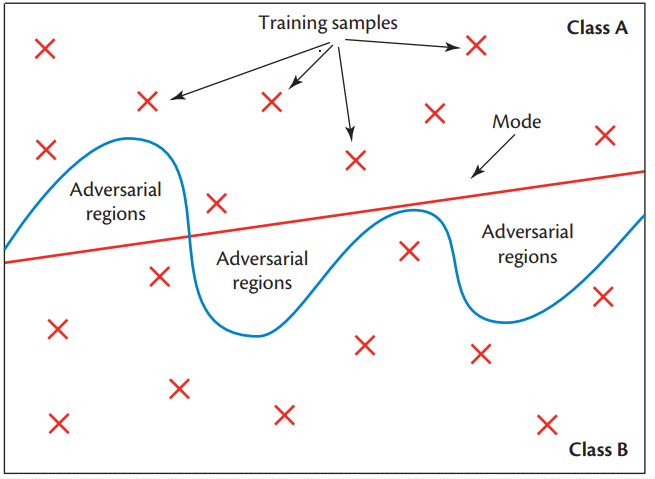


图2-1 对抗样本存在原因

除了考虑对抗样本能否成功误导深度学习模型之外，攻击算法还需要考虑生成扰动的大小，扰动过大可能会覆盖图像的主要特征，导致人类也无法识别该样本。目前研究人员主要使用范数来度量对抗样本中扰动的大小，范数定义如下：

 (2-5)

其中、和三种范数使用较为广泛。范数主要度量对抗样本相对于原始样本改变了多少像素，范数主要度量对抗样本与原始样本的欧式距离，范数主要度量对抗样本与原始样本像素的最大差值。一般情况下基于范数的攻击算法攻击性最强，基于范数的攻击算法生成对抗样本扰动最小。

目前对抗样本攻击分类方式主要有两种，一种是根据攻击者是否拥有深度学习模型的详细信息，分为白盒攻击和黑盒攻击；另一种是根据对抗样本能否使深度学习模型将其误识别为指定分类，分为目标攻击和无目标攻击。白盒攻击是指攻击者知道深度学习模型的全部信息，包括网络结构和网络参数等；黑盒攻击是指攻击者不知道模型的详细信息，只知道模型的输入输出。一般情况下，黑盒攻击可以通过对抗样本的迁移性实现，即先找到一个与攻击目标类似的模型，进行白盒攻击生成对抗样本，然后使用这些对抗样本攻击目标模型。目标攻击是指攻击算法生成的对抗样本，能被深度学习模型误识别为指定分类；无目标攻击只要求攻击算法生成的对抗样本，能够被深度学习模型误识别。

## 2.2 对抗样本攻击算法

自Szegedy等人提出对抗样本概念以来，对抗样本存在的原因一直是一个巨大的争议话题，为诠释对抗样本存在的根本原因，研究人员提出了多种不同的推理和假设。根据不同的推理和假设，研究人员设计并实现了多种不同的对抗样本攻击算法。本节接下来将详细介绍目前主流的四种对抗样本攻击算法。

### 2.2.1 FGSM算法

2015年Goodfellow等人对对抗样本存在原因提出线性假设，作者认为目前深度神经网络大量使用ReLU、LReLU等线性激活函数，导致整个深度神经网络趋向于线性。同时作者发现对抗样本对深度神经网络的影响，会像滚雪球一样，随着网络层数的增加越来越大。基于上述假设和发现，作者提出了一种基于梯度的生成对抗样本算法——FGSM。FGSM算法既可以进行目标攻击，也可进行无目标攻击，其生成无目标攻击对抗样本公式如下：

 (2-6)

其生成目标攻击对抗样本公式如下：

 (2-7)

其中表示模型输入，和表示结果标签，表示损失函数，表示符号函数，表示控制扰动大小的的自定义参数，表示生成的对抗样本扰动。FGSM算法思想是:通过让扰动方向与梯度方向一致，使损失函数值变化最大，进而使分类器分类结果变化最大。

FGSM算法优点是只需一步就能生成对抗样本，并且可以通过控制参数生成任意尺度的对抗样本；缺点是扰动自身抗干扰能力不强，容易受到其他噪声的影响，另外，模型损失函数与模型输入并不是完全线性，这说明该算法生成的对抗样本扰动不是最优扰动。

### 2.2.2 BIM算法

针对FGSM算法存在的问题，Kurakin 等人在FGSM算法基础上提出了一种以迭代的方式生成对抗样本的方法BIM。与FGSM算法类似BIM算法也可以进行目标攻击和无目标攻击，其生成无目标攻击对抗样本公式如下：

 (2-8)

其生成目标攻击对抗样本公式如下：

 (2-9)

其中截断函数为：

 (2-10)

截断函数的目的是保证的每一个像素都在的临域内且每个像素都有意义。BIM每次迭代的含义为：每次迭代在上一步生成的对抗样本基础上每个像素增加(也可能减少)，然后对新生成的对抗样本进行裁剪，保证新的对抗样本每个像素都在的各像素临域内。BIM生成的对抗样本攻击效果一般要优于FGSM生成的对抗样本，最差的情况生成的对抗样本攻击效果与FGSM相同。

### 2.2.3 Deepfool算法

为生成能够使深度学习模型产生误识别的最小扰动，Moosavi-Dezfooli 等人提出了Deepfool算法。Deepfool算法和使用L-BFGS生成对抗样本算法类似，主要区别是：Deepfool算法每次迭代都计算当前样本距离各决策边界的距离，然后选择向最近的决策边界迭代生成扰动。Deepfool 算法生成最小扰动公式如下：

 (2-11)

其中表示分类器模型，表示分类器对的分类类别，表示当前迭代轮次样本到各决策边界的最小距离，表示分类器权重矩阵的第列。

Deepfool算法能生成、和三种范数距离的对抗样本，其生成的对抗样本扰动要小于BIM生成的对抗样本扰动，其攻击效果与FGSM算法生成的对抗样本攻击效果相当。

### 2.2.4 C&W算法

C&W算法是一种基于优化的对抗样本攻击算法，其主要思想是：假设对抗样本是一个变量，那么要使其成功攻击深度学习模型，必须满足两个条件：一是其与真实样本的距离要尽可能的小，二是其能够误导深度学习模型对其进行错误分类。C&W算法生成对抗样本扰动公式如下：

 (2-12)

其中表示分类器模型；是一种激活函数；表示原始样本，表示将样本输入到分类器模型，分类器第层网络的输出，不包含softmax层；和是模型训练时使用的超参，作用是平衡与对整个损失函数的影响，表示分类器将对抗样分类为类的置信度。

C&W算法能生成、和三种范数距离的对抗样本，其攻击效果与BIM生成的对抗样本攻击效果相当。

## 2.3对抗样本防御方法

对抗样本的存在严重影响了深度学习解决方案的应用，为防御对抗样本的攻击，研究人员根据对抗样本的特性设计并实现了多种对抗样本防御方法。本节接下来将详细介绍目前防御性能较好的三种防御方法。

### 2.3.1 对抗训练

Goodfellow在提出FGSM算法的时，也提出通过对抗训练可以显著的提高分类器防御对抗样本攻击的能力。对抗训练是指在分类器模型损失函数中添加一个正则项，该正则项是基于对抗样本引发损失的建模。分类器模型使用这种新的损失函数进行训练,可以显著的提高其对对抗样本的防御能力。作者提出的基于FGSM对抗训练分类器损失函数如下：

 (2-13)

其中为模型原始损失函数，表示对抗样本引发的损失。在实际的对抗训练过程中，一般不采取直接优化公式(2-13),而是选择将FGSM算法生成的对抗样本，添加到分类器模型的训练集中，参与分类器模型的训练。对抗训练的优点在于可以通过添加不同种类的对抗样本，使分类器模型防御能力不断提升；缺点是需要大量生成对抗样本和反复训练模型。

### 2.3.2 蒸馏防御法

2015年Papernot提出了一种基于知识蒸馏的防御对抗样本方法——蒸馏防御法。作者指出对抗样本攻击算法一般是利用模型分类判定区域的对抗方向发起对抗攻击，使用知识蒸馏训练模型可以让这些方向更加平滑，从而达到抵御对抗样本攻击的目的。如图2-2所示，蒸馏防御法主要的防御流程为：首先准备两个相同结构的网络模型，预先训练其中一个模型，然后利用预先训练模型的输出训练另外一个模型。通过这种知识蒸馏的方式，可以很大程度的提高第二模型抵御对抗样本攻击的能力。

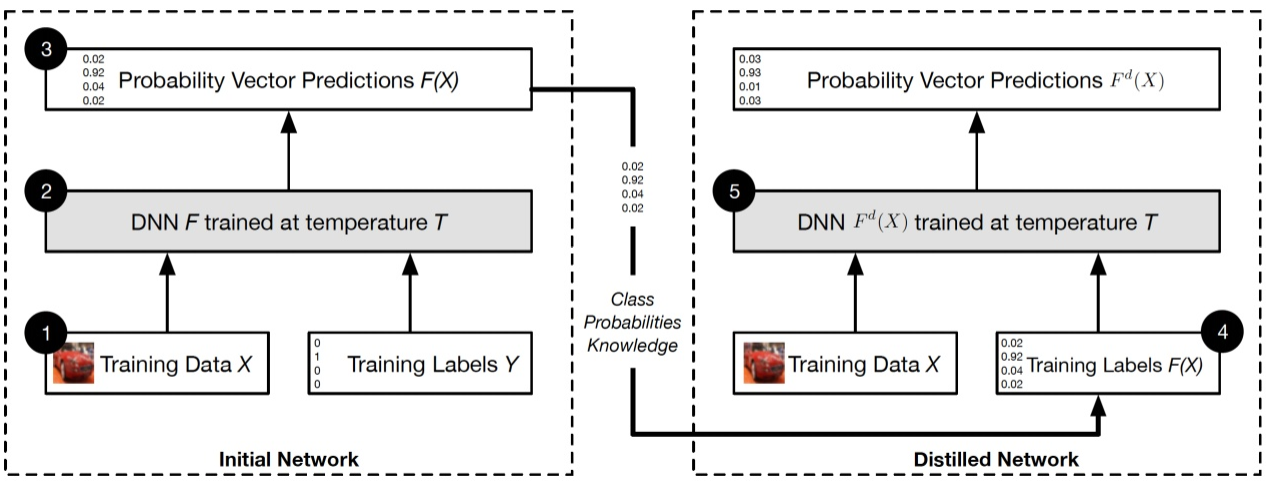


图2-2 蒸馏网络整体流程

蒸馏防御法可以防御大部分的白盒情景下的攻击，同时其防御对抗样本的泛化能力要好于对抗训练。缺点是很难抵御黑盒攻击，同时每防御一个模型需要训练两个模型，防御成本较高。

**2.3.3 图像压缩重建**

2018年jia等人提出一种基于图像压缩重建的对抗样本防御方法——ComDefend。作者认为对抗样本中的扰动相对原始图像是冗余信息，使用图像压缩重建可以在保留对抗样本显著性信息的同时去除对抗样本中的扰动。ComDefend主要包含两个部分图像压缩模块和图像重建模块，图像压缩模块功能是在保留图像主要信息的同时去除冗余信息，重建模型功能是将压缩模块压缩的图像高质量的还原成原始图像。ComDefend整体框架如图2-2所示。

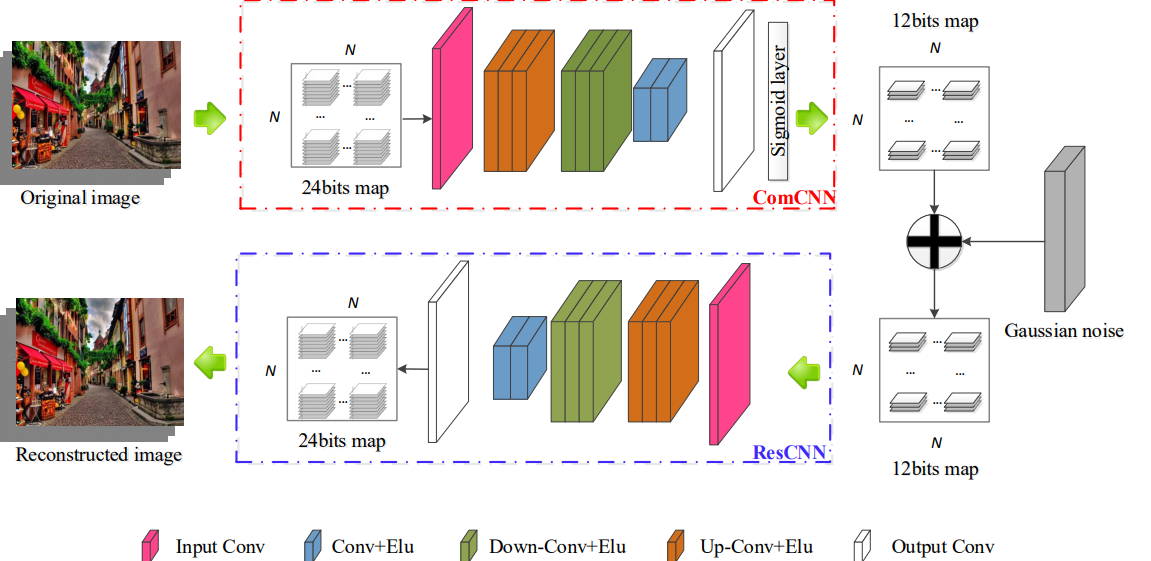


图2-3 ComDefend整体框架

ComDefend防御方法的主要的优点是：不需要专门训练对抗样本，也不需要重新训练分类器模型，并且对于大多数对抗样本都有一定的防御能力；主要缺点是：图像在经过压缩重建后图像质量会下降，这会导致分类器对普通图像分类准确率会下降。

## 2.4 VAE概述

2012年kingma等人基于变分贝叶斯推断提出了变分自动编码器(Variational Auto-Encoder,VAE)，VAE是一种典型的生成式模型，主要运用于数据压缩、图像降噪和图像重构。VAE模型一共包含编码器和解码器两个部分：编码器作用是对输入数据进行变分推断，生成隐变量的近似后验概率分布；解码器的作用是将隐变量恢复成输入数据的近似概率分布。VAE的整体框架如图2-4所示。

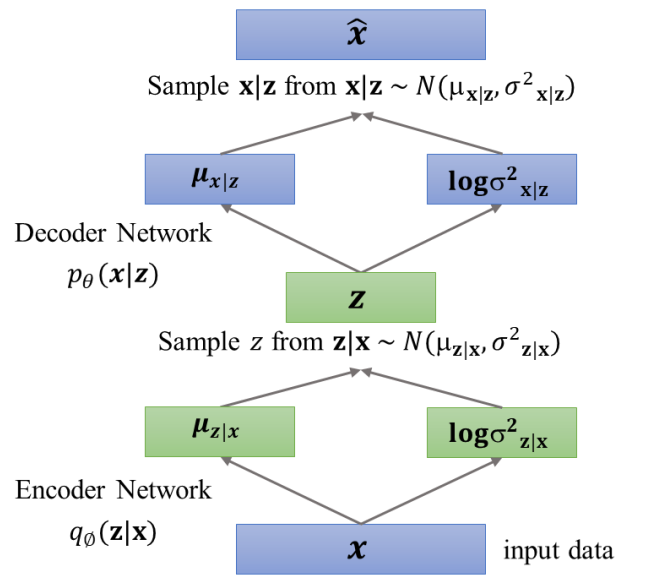


图2-4 VAE的整体框架

在图2-4中，表示真实样本，表示隐变量，表示解码器的输出，表示编码器中网络的参数，表示解码器网络中的参数。其中是一个高维空间可观测的随机向量，是一个相对低维的不可观测随机向量。VAE生成样本可分为两个过程：编码器通过推断得到隐变量的近似分布过程，解码器将隐藏变量还原成近似于输入样本概率分布的过程。

对于编码器，由于隐变量的概率分布无法被观察，于是VAE引入了一个学习模型来代替隐藏变量的真实后验分布，并且假设服从普通正态分布，同时为了方便计算也假设隐变量先验概率分布服从标准正态分布。编码器的优化目标是使尽可能与接近，VAE选择使用KL散度来评价这两个分布的相似度，因此编码器优化目标可以表示为：

 (2-14)

其中表示的边缘概率分布。假设：

 (2-15)

结合公式(2-14)和公式(2-15)可以得到：

 (2-16)

由于恒成立，所以恒成立，因此也被称为VAE的变分下界函数。结合公式(2-14)和公式(2-16)，编码器的优化目标可以转换为：

 (2-17)

对于编码器来说是一个常数，因此编码器的优化目标可以转化为。

对于解码器，VAE希望其能将隐变量恢复为原始输入样本的近似分布,解码器将隐变量恢复为输入样本近似分布可以表示为：

 (2-18)

对于公式(2-18)，VAE希望越大越好。结合公式(2-15)，解码器的优化目标也可以转化为。由于编码器和解码器的损失函数都是，所VAE整体优化目标为最大化其变分下界函数。

由于一开始VAE假设服从普通正态分布，同时也假设服从标准正态分布，因此变分下界函数(2-14)的前半部分可以进行转化：

 (2-19)

对于变分下界的右半部分，该项是关于的后验分布，该项不能通过解析的方式解出，只能通过采样的方式对该项取近似值，因此该项可以转化为：

 (2-20)

需要注意的是不是直接从分布中直接采样，而是使用了一种重参数技巧。该技巧将从直接采样，转变为先从采样得到，然后通过计算得到的采样。最终采样得到的一般是高斯分布，所以该项最可以转化为：

 (2-21)

其中表示解码器网络的输出，也表示分布中的均值，表示分布的方差，与解码器输出无关。

结合公式(2-15)、公式(2-21)和公式(2-22)，VAE的优化目标最终可以表示为：

 (2-22)

在实际训练过程中损失函数的第一项与解码器网络参数更新无关，解码器的参数更新只受损失函数第二项的影响，因此该项也被称为解码器的重构误差。

VAE优点是其通过编码和解码的方式可以使重构样本分布逼近其输入样本的分布，缺点是其重构样本时只考虑了重构样本和输入样本内容误差，这导致使用VAE生成的图像样本通常比较模糊。

## 2.5 GAN概述

2014年Goodfellow等人基于零和博弈提出了生成式对抗网络(Generative Adversarial Networks，GAN)。GAN包含两个网络结构，一个生成网络(Generator)和一个判别网络(Discriminatior)。GAN的生成网络的输入为一个随机噪声，输出为。判别网络的输入为和真实样本，输出为输入样本是真实样本的概率。生成网络和判别网络都会根据判别网络的输出不断的优化自己，生成网络希望生成的能够尽可能的欺骗判别网络，而判别网络希望通过不断的学习尽可能的分辨出输入样本是真实样本还是。GAN的整体框架如图2-5所示。

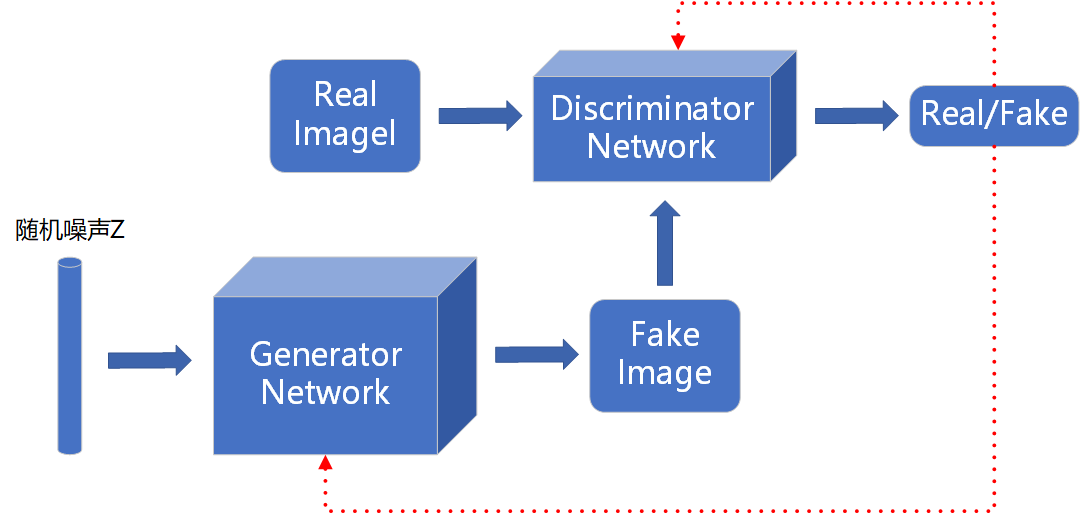


图2-5 GAN的整体框架

与传统深度学习模型训练不同，GAN训练过程中两个网络结构交替训练。训练判别网络时，固定生成网络，相反在训练生成网络时，固定判别网络。经过反复的对抗训练，生成网络最终生成的样本与真实样本的分布一致，判别网络最终无法辨别输入的样本是真实样本还是生成的样本。GAN的优化目标定义如下：

 (2-23)

其中D代表判别网络，G代表生成网络，代表生成网络的输入随机噪声，服从分布，表示真实图像样本服从分布，表示期望。对于判别网络D，其目标是最大化优化目标函数，使得接近并且接近0。对于生成网络，其目标是使优化目标函数尽可能的小，也就是要尽可能的接近1。

GAN生成的图像不是单纯的记忆数据集中的图像样本，而是通过特征学习生成了新图像。GAN对抗训练有很多优点，但也存在一个比较大的缺点：初始的判别网络训练的不好，将会导致生成网络的损失函数值很小，导致生成网络无法学习。

## 2.6 本章小结

本章首先介绍了对抗样本的相关概念，包括对抗样本的定义、度量对抗样扰动大小的标准以及目前对抗攻击的分类；然后介绍了目前主流的四种对抗样本攻击算法：FGSM、BIM、Deepfool和C&W；紧接着又介绍了目前防御性能较好的三种对抗样本防御方法：对抗训练、蒸馏防御法和图像压缩重建；最后介绍了VAE模型和GAN模型的相关概念。

# 3基于VAE-GAN的对抗样本防御方法设计与实现

## 3.1 VAE-GAN防御模型设计

对抗样本是在原始自然图像中加入特定噪声生成的，这部分噪声相对整个图片来说很小，但并不是完全不可见，通过将对抗样本局部放大，我们可以发现对抗样本和原始图像还是有很大的区别。因此，最直接的想法就是可以通过去除对抗样本中的噪声，来抵御对抗样本的攻击。目前比较常用的图像降噪方法分为两种：一种是传统的图像降噪方法，如中值滤波(median filter，MF)或者3D块匹配（Block-Matching 3D，BM3D）降噪；另一种是使用深度神经网络进行图像降噪。前一种方法已经证明无法去除对抗样本扰动，主要是因为对抗样本中的扰动是攻击者专门设计的，与日常生活中的噪声区别很大。而后一种使用深度神经网络降噪，研究人员发现只能去除部分攻击性较弱的对抗样本，攻击性强的对抗样本即使被去除了绝对部分的扰动，依旧能够攻击分类器模型。关于为什么去除了对抗样本中绝大部分噪声，依旧无法防御该样本对分类的攻击，存在多种解释，有一种观点认为普通样本与对抗样本属于两个种不同的分布，大部分降噪方法只考虑了图像的内容特征，忽视了图像的分布特征，导致即使去除大部分对抗样本中的扰动，也无法提高分类器对对抗样本分类准确率。针对上述问题，本章设计并实现了专门针对于对抗样本降噪的防御对抗样本方法——基于VAE-GAN的对抗样本防御方法，该方法主要思路是在图像进入分类器之前，使用VAE-GAN模型去除对抗样本中的扰动，然后将降噪后的图像送入分类器模型，以提高分类器模型分类器对对抗样本的分类准确率。本章接下来将详细介绍VAE-GAN防御模型的设计、VAE-GAN防御模型网络结构的设计以及VAE-GAN模型优化目标及训练流程。

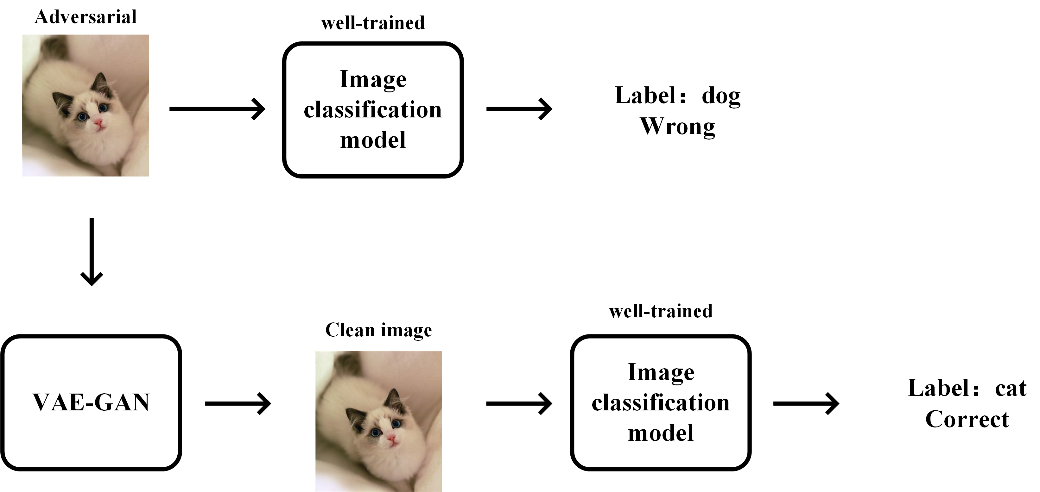


图3-2 VAE-GAN防御模型防御思路

## 3.2 VAE-GAN防御模型网络结构设计

VAE-GAN模型整体框架如图3-2所示，该模型一共包含3个部分：

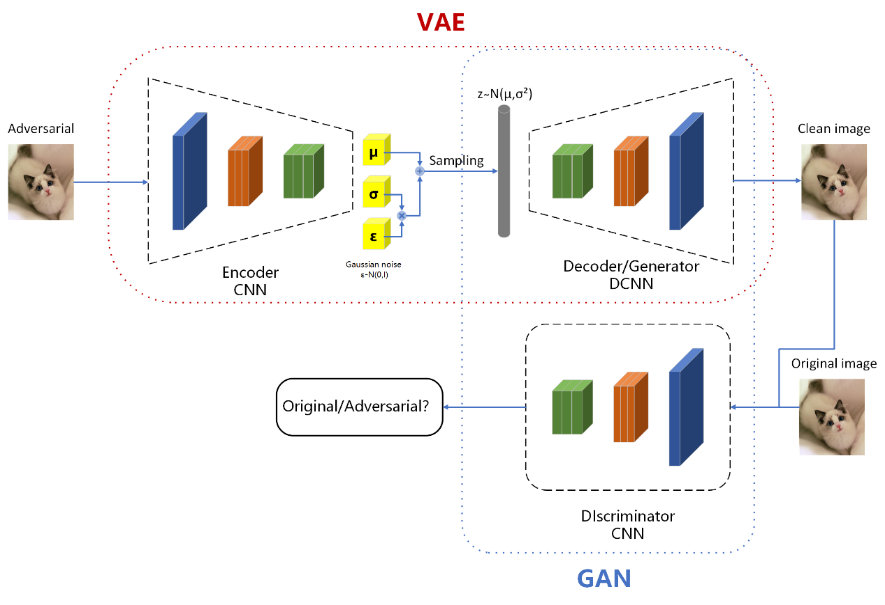


图3-1 VAE-GAN模型整体框架

(1)编码器：编码器主要功能是将输入的图像转化为隐变量的近似后验概率分布，然后对近似后验分布进行采样得到隐变量。编码器的输入为对抗样本或者原始样本，输出是隐变量近似概率分布的均值方差。

(2)生成器：

(3)判别器：GAN的判别器是一个二分类的分类器网络，如果输入的图像是对抗样本，由于对抗样本扰动累加效应判别器的输出会与真实图像的输出不同，这说明在GAN的判别器眼中对抗样本和真实样本属于两种分布，因此通过GAN的生成器与判别器的对抗训练，可以保证VAE解码器生成的图像不是对抗样本。GAN的生成器与DVAE的解码器共用一个网络，所以GAN的生成器输入输出与DVAE的输入输出一致。GAN的判别器的输入是生成器生成的去噪后的图像和该图像的原始图像，输出为输入图像是原始图像的概率。

### 3.2.1 编码器网络结构设计

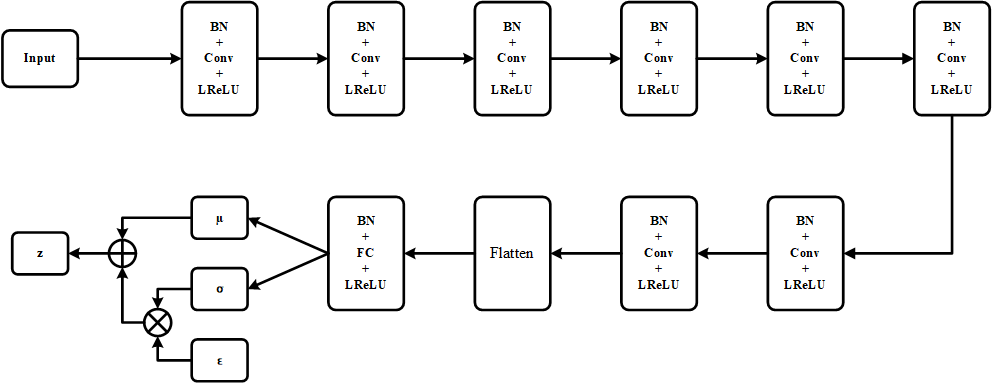


表3-1 VAE-GAN编码器网络结构

|  |  |  |
| --- | --- | --- |
| 各层类型 | | 参数 |
| BN+Conv+LReLU | filters:64, filter\_size:, strides:1, padding: SAME | |
| BN+Conv+LReLU | filters:64, filter\_size:, strides:2, padding: SAME | |
| BN+Conv+LReLU | filters:128, filter\_size:, strides:1, padding: SAME | |
| BN+Conv+LReLU | filters:128, filter\_size:, strides:2, padding: SAME | |
| BN+Conv+LReLU | filters:256, filter\_size:, strides:1, padding: SAME | |
| BN+Conv+LReLU | filters:256, filter\_size:, strides:2, padding: SAME | |
| BN+Conv+LReLU | filters:512, filter\_size:, strides:1, padding: SAME | |
| BN+Conv+LReLU | filters:512, filter\_size:, strides:2, padding: SAME | |
| Flatten | | Min\_ndim:3 |
| FC+LReLU | | 128 units |

如表4.7所示，CIFAR-10数据集DVAE-GAN编码器一共有十层网络，前八层都是卷积层，分别有64、64、128、128、256、256、512、512个卷积核，卷积核大小都是，奇数层的卷积核步长为1，偶数层的卷积核的步长为2，边缘填充模式都是SAME，激活函数激活前都进行归一化处理，激活函数都是LReLU，第九层是扁平化层作用是将输入的多维数据转化为一维数据，最后一层为全连接层作用是输出隐变量的先验分布均值和方差（注意均值和方差是并列连接该全连接层），激活函数为LReLU。

### 3.2.2 生成器网络结构设计

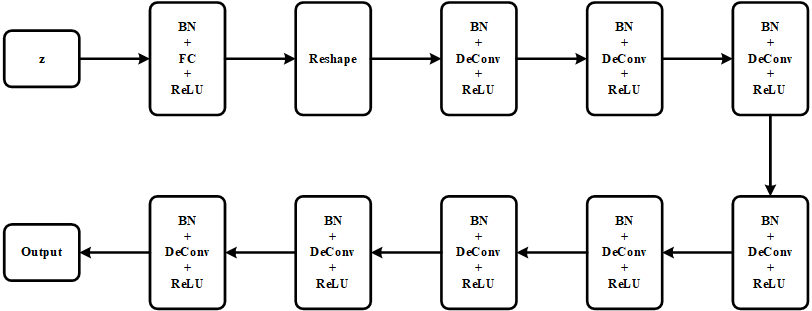


表4-8 VAE-GAN生成器网络结构

|  |  |
| --- | --- |
| 各层类型 | 参数 |
| Reshape+ReLU | reshape\_size: |
| BN+Deconv+ReLU | filters:512, filter\_size:, strides:2, padding:SAME |
| BN+Deconv+ReLU | filters:512, filter\_size:, strides:1, padding:SAME |
| BN+Deconv+ReLU | filters:256, filter\_size:, strides:2, padding:SAME |
| BN+Deconv+ReLU | filters:256, filter\_size:, strides:1, padding:SAME |
| BN+Deconv+ReLU | filters:128, filter\_size:, strides:2, padding:SAME |
| BN+Deconv+ReLU | filters:128, filter\_size:, strides:1, padding:SAME |
| BN+Deconv+ReLU | filters:64, filter\_size:, strides:2, padding:SAME |
| BN+Deconv+ReLU | filters:64, filter\_size:, strides:1, padding:SAME |
| Tanh | output |

如表4-8所示，CIFAR-10数据集上DVAE-GAN模型的生成器（解码器）一共有十层网络，第一层是输入重塑层作用是将输入的隐藏变量形状重塑为，然后将重塑后的结果归一化处理，最后使用ReLU激活函数对归一后的结果进行非线性变换；第二、三、四、五、六、七、八、九层都是反卷积层，分别有512、512、256、256、128、128、64、64个卷积核，卷积核大小都是，偶数层的卷积核步长为2，奇数层的卷积核步长为1，边缘填充模式都是SAME，在激活函数激活前都进行数据的归一化处理，激活函数都是ReLU函数；最后一层是输出层使用的激活函数为Tanh激活函数。

### 3.2.3 判别器网络结构设计

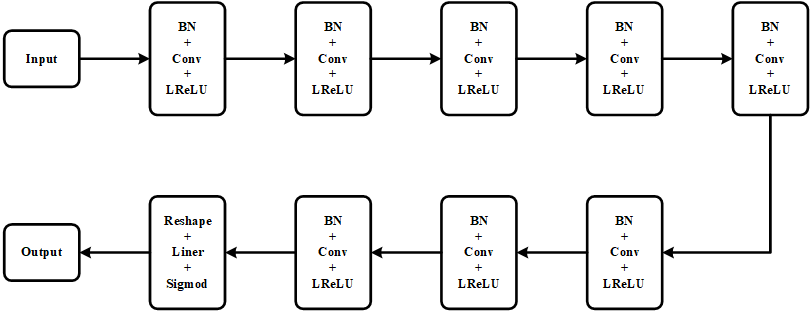


表4-9

|  |  |
| --- | --- |
| 各层类型 | 参数 |
| BN+Conv+LReLU | filters:64, filter\_size:, strides:1, padding:SAME |
| BN+Conv+LReLU | filters:64, filter\_size:, strides:2, padding:SAME |
| BN+Conv+LReLU | filters:128, filter\_size:, strides:1, padding:SAME |
| BN+Conv+LReLU | filters:128, filter\_size:, strides:2, padding:SAME |
| BN+Conv+LReLU | filters:256, filter\_size:, strides:1, padding:SAME |
| BN+Conv+LReLU | filters:256, filter\_size:, strides:2, padding:SAME |
| BN+Conv+LReLU | filters:512, filter\_size:, strides:1, padding:SAME |
| BN+Conv+LReLU | filters:512, filter\_size:, strides:2, padding:SAME |
| Reshape+Liner+sigmod | reshape\_size: |

VAE模块的编码器主要功能是将输入的图像转化为隐变量的近似后验概率分布，然后对近似后验分布进行采样得到稳定的隐变量。编码器的输入为对抗样本或者原始样本，输出是两组维向量：一个组均值向量，另一组是标准差向量，然后我们可以通过这两组维向量构建隐变量的后验近似概率分布。最后我们再从隐变量近似概率分布采样获得隐变量，采样方式为，其中表示从标准正态分布中随机取样的结果。DVAE的解码器主要功能将隐变量恢复为无噪声图像，解码器的输入为隐变量，输出为降噪后的图像。

## 3.3 VAE-GAN模型优化目标及训练流程

### 3.3.1 模型优化目标

基于VAE-GAN的对抗样本防御模型由VAE和GAN两个模型组成，一共有三个网络结构：编码器网络，生成器共用网络和判别器网络。对于VAE的编码器，其优化目标就是最大化VAE的变分下界函数：



对应的VAE编码器的损失函数为：



由于DVAE的解码器和GAN的生成器共用一个网络，因此DVAE解码器和GAN共用的网络的优化目标包含DVAE的变分下界函数和GAN生成器的损失函数：



由于VAE变分下界函数前半部分与DVAE的解码器无关，同时GAN的优化目标的第一项在GAN的生成器训练时是固定值，因此可以将这两项舍弃。然后生成器网络的优化目标可以转化为：



对应的生成器网络可以表示为：



GAN的判别器网络优化目标和普通GAN判别器优化目标一样为：



对应的GAN的判别器网络的损失函数可以表示为：



综上所述，模型整体优化目标为：



对应的模型整体损失函数为：



### 3.3.2模型训练流程

在训练过程中，由于GAN对抗训练的存在，我们无法直接根据模型的整体优化目标直接更新全部网络的参数，最后本文选择分步更新模型中各网络结构中的参数，网络参数更新顺序为：先更新编码器网络参数，然后更新生成器网络参数，最后更新判别器网络参数。VAE-GAN训练流程如表3.2所示。

表3.2 DVAE-GAN训练流程

|  |
| --- |
|  |
|  |

训练具体流程为：

1.初始化DVAE编码器网络，DVAE解码器和GAN生成器共用网络和GAN判别器网络中的参数；

2.如果剩余训练周期大于0并且模型各模块的损失函数不是无穷大执行步骤3，否者训练结束；

3.从训练集中选取一批图像样本；

4.随机向图像样本中添加对抗扰动，得到；

5.通过DVAE的编码器网络计算隐变量后验概率分布的均值和方差和；

6.从隐变量的后验概率分布采样得到隐变量；

7. 通过DVAE解码器和GAN生成器共用网络将隐变量还原成无噪声图像；

8.计算DVAE编码器网络的损失函数；

9.计算DVAE解码器和GAN生成器共用网络的损失函数；

10.计算GAN判别器网络的损失函数；

11.更新DVAE编码器网络中的参数；

12.更新DVAE解码器和GAN生成器共用网络中的参数；

13.更新GAN的判别器网络中参数；

14.训练周期减1，跳转到步骤2。

## 3.4 本章小结

本章针对对抗样本的噪声特性和扰动累加效应，提出了一种基于DVAE-GAN的对抗样本防御策略。首先简单介绍了该防御策略的防御思路和模型的整体框架，然后详细介绍了防御模型中的DVAE模块和GAN模块，并解释了各模块对于防御对抗样本的作用。然后分析了各个模块的优化目标及模型的整体目标。最后详细介绍了整个防御的算法流程。

# 4 实验及结果分析

## 4.1实验环境

### 4.1.1实验数据集

本文使用的实验硬件环境为：英特尔Xeon E-5-2678v3处理器，64G内存，GPU型号为英伟达RTX2080，显存大小为8G。实验使用的软件环境为：Ubuntu16.04操作系统，编程语言为python3，开发环境为PyCharm，使用的机器学习框架为TensorFlow 1.15.1。本文使用的实验数据集为：MNIST数据集和CIFAR-10数据集。

MNIST数据集是计算机视觉领域使用最多的数据集之一，该数据包含有70000张不同的手写数字图片，每张图片包含28\*28个灰度像素点和一个对应的0-9的标签。MNIST数据集一共分为三个部分：训练集，交叉验证集和测试集。训练集包含55000张图片，交叉验证集包含5000张图片，测试集包含10000张图片。

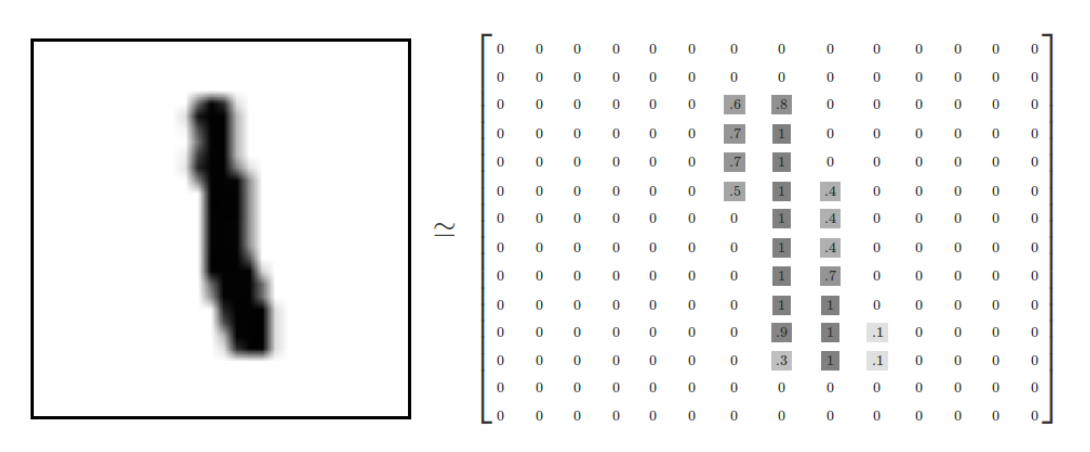


图4.1 MNIST数据集示例



图4.2 CIFAR-10数据集示例

CIFAR-10数据集由6000张彩色图片组成，每张图片包含32\*32个RGB彩色像素点。CIFAR-10数据集包含十个互斥的类别：飞机、鸟、马、青蛙、猫、狗、鹿、卡车、船、汽车。每个分类都包含6000张图片。训练时随机从每类6000张图片中选取1000张图片作为测试集，剩余50000张图片作为训练集。

### 4.1.2攻防目标模型

为了验证本文提出的基于VAE-GAN对抗样本防御策略的有效性，同时检验防御模型是否具有迁移性，本文在MNIST数据集和CIFAR-10数据集中分别实现两个不同分类精度的分类器模型作为对抗样本攻击和防御模型，分类器模型网络结构如表4-1、表4-2、表4-3和表4-4所示。

表4-1 MNIST数据集分类器模型MNIST\_A网络结构

|  |  |
| --- | --- |
| 各层类型 | 具体结构 |
| Conv+ReLU | filters:64, filter\_size:, strides:2, padding:SAME |
| Conv+ReLU | filters:128, filter\_size:, strides:2, padding:SAME |
| Conv+ReLU | filters:128, filter\_size:, strides:1, padding:SAME |
| Flatten | Min\_ndim:3 |
| FC+ReLU | 200 units |
| Softmax | 10 units |

表4-2 MNIST数据集分类器模型MNIST\_B网络结构

|  |  |
| --- | --- |
| 各层类型 | 具体结构 |
| Conv+ReLU | filters:64, filter\_size:, strides:1, padding:SAME |
| Conv+ReLU | filters:64, filter\_size:, strides:1, padding:SAME |
| Max Pooling | pool\_size: |
| Conv+ReLU | filters:128, filter\_size:, strides:1, padding: SAME |
| Conv+ReLU | filters:128, filter\_size:, strides:1, padding: SAME |
| Max Pooling | pool\_size: |
| Flatten | Min\_ndim:3 |
| FC+ReLU | 100 units |
| FC+ReLU | 100 units |
| Softmax | 10 units |

如表4-1所示，MNIST数据集分类器模型MNIST\_A一共有6层网络，第一层是卷积层包含64个卷积核，每个卷积核大小为，卷积核的移动步长为2，激活函数为ReLU函数，边缘填充模式为SAME；第二层也是卷积层包含128个卷积核，每个卷积核大小为，卷积核移动步长为2，激活函数为ReLU函数，边缘填充模式为VALID，第三层也是卷积层包含128个卷积核，每个卷积核大小为，卷积核移动步长为1，激活函数为ReLU函数，边缘填充模式为VALID；第四层是扁平化层用于将多维输入转化为一维输出；第五层是全连接层一共包含100个神经元，激活函数为ReLU函数；最后一是Softmax层作用是输出分类结果。如表4-2所示，MNIST数据集分类器模型MNIST\_B一共有10层网络，第一层和第二层都是卷积层都拥有64卷积核，卷积核大小都是，卷积核的移动步长都是1，边缘填充模式都是SAME；第三层是池化层，池化方法为最大池化法，池化核大小为；第四层和第五层都是卷积层都拥有64卷积核，卷积核大小都是，卷积核的移动步长都是1，边缘填充模式都是VALID；第七层是池化层，池化方法为最大池化法，池化核大小为；第八层和第九层都是全连接层分别包含100个神经元和100个神经元，激活函数都是ReLU函数；最后一层为Softmax层作用是输出分类结果。

表4-3 CIFAR-10数据集分类器模型CIFAR\_A网络结构

|  |  |
| --- | --- |
| 各层类型 | 具体结构 |
| Conv+ReLU | filters:64, filter\_size:, strides:1, padding:SAME |
| Conv+ReLU | filters:64, filter\_size:, strides:1, padding:SAME |
| Max Pooling | pool\_size: |
| Conv+ReLU | filters:128, filter\_size:, strides:1, padding: VALID |
| Conv+ReLU | filters:128, filter\_size:, strides:1, padding: VALID |
| Max Pooling | pool\_size: |
| Flatten | Min\_ndim:3 |
| FC+ReLU | 256 units |
| FC+ReLU | 256 units |
| Softmax | 10 units |

如表4-3所示，CIFAR-10数据集分类器模型CIFAR\_A网络结构与MNIST数据集分类器模型MNIST\_B类似，唯一区别是CIFAR\_A网络的第八层和第九层是256个神经元的全连接层，MNIST\_B网络的第八层和第九层是100个神经元的全连接层。

表4-4 CIFAR-10数据集分类器模型CIFAR\_B网络结构

|  |  |
| --- | --- |
| 各层类型 | 具体结构 |
| Conv+ReLU | filters:64, filter\_size:, strides:1, padding: SAME |
| Conv+ReLU | filters:64, filter\_size:, strides:1, padding: SAME |
| Max Pooling | pool\_size: |
| Conv+ReLU | filters:128, filter\_size:, strides:1, padding: SAME |
| Conv+ReLU | filters:128, filter\_size:, strides:1, padding: SAME |
| Max Pooling | pool\_size: |
| Conv+ReLU | filters:256, filter\_size:, strides:1, padding: SAME |
| Conv+ReLU | filters:256, filter\_size:, strides:1, padding: SAME |
| Conv+ReLU | filters:256, filter\_size:, strides:1, padding: SAME |
| Max Pooling | pool\_size: |
| Conv+ReLU | filters:512, filter\_size:, strides:1, padding: SAME |
| Conv+ReLU | filters:512, filter\_size:, strides:1, padding: SAME |
| Conv+ReLU | filters:512, filter\_size:, strides:1, padding: SAME |
| Max Pooling | pool\_size: |
| Conv+ReLU | filters:512, filter\_size:, strides:, padding:SAME |
| Conv+ReLU | filters:512, filter\_size:, strides:, padding: SAME |
| Conv+ReLU | filters:512, filter\_size:, strides:, padding: SAME |
| Flatten | Min\_ndim:3 |
| FC+ReLU | 4096 units |
| FC+ReLU | 4096 units |
| Softmax | 10 units |

如表4-4所示，CIFAR-10数据集分类器模型CIFAR\_A一共拥有21层网络，前17层除了第三层、第六层、第十层和第十四层之外全是卷积层，除了拥有的卷积核的数量不同，卷积核的大小、激活函数、卷积核的移动步长和边缘填充模式都相同。第三层、第六层、第十层和第十四层都是池化层，池化方式都是最大池化法，池化核大小都是；第十八层是扁平化层用于将多维输入转化为一维输出；第十九层和第二十层都是全连接层分别拥有4096个神经元和10个神经元；最后一层是Sofmax层作用是输出分类结果。

构建完分类器模型后，对各个分类器模型进行训练，训练参数如表4-3所示。对于MNIST\_A和MNIST\_B模型，训练时学习率都设置为0.001，每批次都并行训练100张图片，训练周期分别为15和20。对于CIFAR\_A和CIFAR\_B模型，训练学习率都设置为0.0005，每批次都并行训练100张图片，训练周期分别为50和100。训练完的结果如表4-4所示，MNIST数据集两个分类器模型经过训练后，在测试集的分类准确率分别为96.43%和98.72%；CIFAR-10数据集两个分类器模型经过训练后，在测试集的分类准确率分别为73.92%和85.53%。

表4-3 训练参数表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 训练参数 | MNIST\_A | MNIST\_B | CIFAR\_A | CIFAR\_B |
| Learning Rate | 0.001 | 0.001 | 0.0005 | 0.0005 |
| Batch Size | 100 | 100 | 100 | 100 |
| Epochs | 15 | 20 | 50 | 100 |

表4-4 分类器训练结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | MNIST\_A | MNIST\_B | CIFAR\_A | CIFAR\_A |
| 准确率 | 96.43% | 98.72% | 73.92% | 85.53% |

在构建完两个DVAE-GAN模型网络结构后需要对两个模型进行训练，具体的训练过程为：使用FGSM、BIM和PGD生成对抗样本方法对本文提出的4个攻防分类器模型进行攻击，对应的每个模型生成五组扰动大小为0.02、0.04、0.06、0.08、0.1的对抗样本，每组对抗样本数量为1000个；然后将这15组对抗样本加上35000张普通训练集中的图像作为训练DVAE-GAN的一个训练数据集，一共能生成2个MNIST数据集防御模型的训练集和2个CIFAR-10数据集防御模型的训练集；最后使用这4个训练集对两个防御模型进行训练，训练时使用的参数如表4-10所示，训练完成后能得到2个MNIST数据防御模型和2个CIFAR-10数据集模型。为了更好的区分这四个防御模型，本文将这4个防御模型命名为：DGMA、DGMB、DCMA和DCMB。DGMA表示使用MNIST\_A分类器生成的对抗样本训练成的MNIST数据集防御模型，DGMB表示使用MNIST\_B分类器生成的对抗样本训练成的MNIST数据集防御模型，DCMA表示使用CIFAR\_A分类器生成的对抗样本训练成的CIFAR-10数据集防御模型，DCMB表示使用CIFAR\_B分类器生成的对抗样本训练成的CIFAR-10数据集防御模型。

## 4.2 白盒实验结果及分析

## 4.3 黑盒实验结果及分析

## 4.4 去噪实验结果及分析

## 4.5 本章小结