基于迭代自编码器的深度学习对抗样本防御方案

杨浚宇 [1,2,3]

[1] 上海微系统与信息技术研究所, 上海 中国 200050
[2] 上海科技大学信息学院, 上海 中国 201210
[3] 中国科学院大学, 北京 中国 10002

摘要 近年来, 深度学习在计算机视觉领域表现出优异的性能, 然而研究者们却发现深度学习系统并不具备良好的鲁棒性, 对深度学习系统的输入添加少许的人类无法察觉的干扰就能导致深度学习模型失效, 这些使模型失效的样本被研究者们称为对抗样本。我们提出迭代自编码器, 一种全新的防御对抗样本方案, 其原理是把远离流形的对抗样本推回到流形周围。我们先把输入送给迭代自编码器, 然后将重构后的输出送给分类器分类。在正常样本上, 经过迭代自编码器的样本分类准确率和正常样本分类准确率类似, 不会显著降低深度学习模型的性能; 对于对抗样本, 我们的实验表明, 即使使用最先进的攻击方案, 我们的防御方案仍然拥有较高的分类准确率和较低的攻击成功率。

关键词 对抗样本; 自编码器; 深度学习; 图像分类
中图法分类号 TP309 **DOI 号** 10.19363/J.cnki.cn10-1380/tn.2019.11.03

IDAE: Iterative Denoising Autoencoder based Deep Learning Model Enhancement Mechanism against Adversarial Examples

YANG Junyu[1,2,3]

[1] Shanghai Institute of Microsystem and Information Technology, Shanghai 200050, China
[2] School of Information Science and Technology of ShanghaiTech University, Shanghai 201210, China
[3] University of Chinese Academy of Sciences, Beijing 100029, China

Abstract Nowadays, in computer vision area, deep learning has shown impressive performance. However, researchers found that deep learning systems are not robust enough. Deep learning models will fail when attackers add some specially crafted perturbations, which are imperceptible to humans. These examples that cause the model fail to work are named adversarial examples by researches. We propose a new defense mechanism named iterative denoising autoencoder(IDAE). The intuition behand IDAE is that we iteratively push examples that far away from manifold onto the manifold. We apply IDAE to test examples and then send the reconstructed examples to the classifier. We show that, for normal examples, the reconstructed examples after IDAE have classification accuracy comparable to their original versions, suggesting that the reconstructed examples are on the manifold and will not decrease the performance of model. For adversarial examples, we show that this defense achieved high classification accuracy and low attack success rate on the state of the art attacks in both grey-box and white-box attacks.

Key words adversarial examples; autoencoder; deep learning; image classification

1 引言

近几年来, 随着人工智能应用的火热发展, 深度学习应用在越来越多的领域中, 例如图像分类[1]、自然语言处理[2]等。随之而来的人工智能安全问题也引起了越来越多人的关注, 在安全要求性较高的应用场景中, 例如无人自动驾驶汽车, 深度学习需要具备足够的鲁棒性来应对各种实际场景, 从而保障行驶安全。然而, 最近研究者们发现深度学习系统并不具有良好的鲁棒性, 攻击者可以通过对正常输入

通讯作者: 杨浚宇, 硕士, Email: yangjy@shanghaitech.edu.cn。
收稿日期: 2019-01-06; 修改日期: 2019-03-20; 定稿日期: 2019-11-04

增加微小到人为不可见的噪声使得深度学习模型失效, 这些使得模型失效的样本被称之为对抗样本。对抗样本给深度学习系统带来了很大的安全隐患, 例如 Kevin[3]等人攻击现有的无人驾驶汽车产生对抗样本, 使得无人驾驶汽车错误地把停止路标识别为限速 45 路标; 特斯拉将大货车错误地识别为蓝天导致驾驶员身亡。随着人工智能应用的普及, 深度学习的安全性显的愈发重要。

目前深度学习安全研究的主要关注点是对抗样本, 研究者们通过使用不同方法产生[4-6]和防御[7]对抗样本, 以此来验证深度学习的安全性。在本文中, 我们提出迭代自编码器, 一种全新的基于流形(manifold)学习理论的防御深度学习中对抗样本的方法。流形学习理论认为我们观测到的高维空间数据实际上是由低维空间映射而来, 稀疏的高纬空间观测数据实际上密集地分布在产生数据的低维空间中(流形上)。研究者们通常使用自编码器来获得训练数据的流形[8-9], 他们的研究表明一个最优的去噪自编码器或者正则化自编码器可以把输入往流形中密度最高的方向移动[8-13], 使得重构后的输出更加靠近流形。

研究者们把自编码器应用在各个领域中, 例如图像去噪[14],图像恢复[15],图像增强[16], 防御对抗样本[7]等等。MagNet[7]是目前最先进的对抗样本防御方案, 也是第一个基于自编码器的防御方案, 他们把输入送给自编码器, 然后把重构后的样本送给分类器分类。其背后的原理是: 分类器是通过密集分布在流形上的数据训练得到的, 因此分类器对于在流形上的输入拥有较好的分类效果, 对于在流形外的样本分类效果较差。然而, 该方案的缺陷是经过一次自编码器重构后的样本可能并未到达流形上, 即使自编码器可以将输入往流形上移动。MagNet 的作者也意识到了该问题, 他们在论文中表明随着对抗干扰幅度增加, MagNet 防御效果会随之降低。从流形学习的角度可以理解为, 随着对抗干扰幅度的增加, 输入样本离流形越来越远。

MagNet 提出的解决方案是通过输入样本的重构误差来判断一个样本是否离流形过远, 如果样本离流形过远, 就拒绝该样本, 但是该方案很容易导致 DOS(deny of service)攻击。为了克服上述缺陷, 我们想要尽可能地把远离流形的输入拉回到流形上, 因此我们提出了迭代自编码器(IDAE), 迭代自编码器迭代地把样本往流形上移动, 使其更靠近流形, 以此来达到更好的防御效果。为了验证迭代自编码器防御的有效性, 我们使用几种不同攻击手段攻击被迭代自编码器保护的分类器, 实验表明, 我们的防御方案对于对抗样本有较好的防御效果。并且, 我们的防御方案可以和其他防御方案结合, 例如对抗样本训练, 随机集成模型等, 以此获得更好的防御效果。

2 背景知识和相关工作

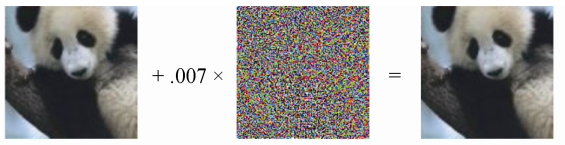
2.1 自编码器

一个自编码器$ae = decode(encode(x))$是一个复合函数, 由编码函数$encode$和解码函数$decode$组成, 是一种特殊的无监督神经网络。编码函数负责把输入空间压缩映射到一个具有简单结构的隐藏空间H中, 解码函数负责将压缩后的隐藏空间H转换回输入空间。假设自编码器的输入是x,则输出标签也是x, 为了防止自编码器学到简单的自我拷贝, 研究者们通常会采用在隐藏空间H中使用各种正则化手段来保证隐藏空间拥有良好的表达[13,17]。还有一种常用的自编码器是去噪自编码器, 去噪自编码器训练的目标是从包含噪声的训练样本中恢复正常清洁的样本。例如去噪自编码器的输入是$x + \varepsilon$,其中ε是随机产生的噪声或者对抗干扰, 输出标签是x。

2.2 对抗样本

对于一个分类任务来说, 正常样本是那些从正常数据分布中采集出来的样本, 对抗样本则是在正常样本基础上经过精心设计让分类器分类错误的样本, 精心设计指的是给定正常样本和对应的对抗样本, 人无法区分对抗样本和正常样本, 图一展示了正常样本和使用快速梯度方向法产生的对抗样本, 最左边的图片表示正常样本, 中间图片表示通过快速梯度方向法产生的噪声, 最右边的图片表示最后产生的对抗样本。正常样本以 57.7%的置信值被正确分类为熊猫, 而对抗样本以 99.3%的置信值被错误分类为长臂猿。对于计算机来说, 需要特定的度量标准来描述对于原图的干扰大小, 研究者们通常使用如下三种度量标准[4]来模拟人的感官: L^0,L^2和L^∞,上述三种度量标准其实是L^p范数的特殊形式:

$$\left|\left|x\right|\right|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}} \# \qquad (1)$$



x
"panda"
57.7% confidence

$\text{sign}(\nabla_x J(\theta, x, y))$
"nematode"
8.2% confidence

$x + \varepsilon\text{sign}(\nabla_x J(\theta, x, y))$
"gibbon"
99.3% confidence

图 1 使用快速梯度方向法产生的对抗样本

Figure 1 Adversarial examples generated by Fast Gradient Sign Method

2.3 现有攻击方法

Goodfellow 等人[5]最早发现深度学习中存在对抗样本, 之后其他研究者提出各种不同的方法来产生对抗样本, 对于某个固定输入, 大部分的攻击方法使用基于梯度优化的方法产生对抗样本[4,18]。甚至文献[19]针对特定的白盒模型找到了通用的对抗干扰, 把该干扰加到任意输入图像上可以使得大部分输入分类错误。Pin-yu 等[20]使用黑盒攻击的方法, 在没有获得梯度的情况下生成对抗样本。我们实验了如下三种攻击方法, 来说明迭代自编码器防御的有效性:

2.3.1 快速梯度方向法

快速梯度方向法[5](Fast Gradient Sign Method,FGSM)希望在原输入的L^∞球中找到对抗样本来欺骗分类器。快速梯度方向法定义了损失函数$Loss(x, l_x)$来表述输入x被分类器分类成真实标签l_x的损失, 然后通过最大化这个损失函数来产生对抗样本。快速梯度方向法求得损失函数对于输入的梯度, 然后对于输入的每个像素加上梯度方向ε大小的干扰, 可以用如下公式表示:

$$x' = x + \varepsilon \cdot sign\left(\nabla_x Loss(x, l_x)\right) \# \quad (2)$$

随着ε的增加, 攻击样本x'的干扰大小增加, 攻击成功率增加。快速梯度方向法的优点是可以快速产生攻击样本, 但是却不能保证产生的攻击样本成功欺骗分类器。

2.3.2 映射梯度下降法

映射梯度下降法[6] (Projected Gradient Decent, PGD)也是在L^∞球中寻找对抗样本欺骗分类器的方法。映射梯度下降法可以认为是多步快速梯度方向下降法的一种变形。其基本思想是多次进行快速梯度方向法攻击, 在每次攻击之后把攻击样本映射回原样本的L^∞球内。可以用如下公式表示:

$$x^{t+1} = \prod_{x,\varepsilon}\left(x^t + \lambda \cdot sign\left(\nabla_{x^t} L(x^t, l_x)\right)\right) \# \quad (3)$$

其中, ε表示最大允许更改的对抗样本干扰大小, $\prod_{x,\varepsilon}$ 表示把攻击样本映射回原样本x的ε球内, λ表示每次攻击的修改幅度。与快速梯度方向法类似, 映射梯度下降法也不能保证生成的攻击样本可以成功欺骗分类器, 但是相较于快速梯度方向法成功率更高, 并且修改幅度较小。

2.3.3 C&W 攻击方法

Carlini 和 Wagner[4]攻击方法是目前最先进的攻击方法, 该方法可以生成所有三种度量单位的对抗样本, 并且可以指定生成后的对抗样本被分类到哪一个类别中。我们用不生成指定类别的L^2攻击方法来说明该方法的主要思路。该攻击方法主要优化如下优化目标:

$$minimize_\delta \ \left|\left|\delta\right|\right|_2 + c \cdot f(x+\delta)$$
$$s.t \quad x + \delta \in [0,1]^n \# \quad (4)$$

对于输入样本x,该攻击方法希望找到一个对抗干扰δ尽可能小, 并且原样本加该干扰得到的对抗样本可以使分类器分类错误。c是一个超参数, 用于控制两个目标函数之间的平衡, 同时对抗样本需要是一张合理的图像(在 0—1 范围内)。$f(\cdot)$是一个人工设计的函数, 该函数小于等于零当且仅当分类器的分类结果和真实标签不同时, 即分类器分类错误, 他们推荐使用如下函数:

$$f(x) = \max\left(Z(x)_{l_x} - \max\{Z(x)_i : i \neq l_x\}, -k\right) \# \quad (5)$$

其中, $Z(x)$是 softmax 函数前一层的输出结果, l_x表示输入样本x的真实标签, k是一个称之为置信值的超参数, 随着置信值的增加, 生成的对抗样本会被更可信的分类错误, 与此同时, 对抗干扰的幅度也会随之增加。

与前两种攻击方法不同的是, C&W 攻击方法可以保证生成的对抗样本一定被分类错误, 并且至少以高出其他类别k值的分类错误, 该方案的缺点是计算量巨大导致对抗样本生成非常缓慢。

2.4 现有防御方法

现有对抗样本防御方法基本可以分为如下两类:

2.4.1 对抗样本训练

一个非常直观的想法是把对抗样本加入训练集训练分类器, 训练的时候使用正确的标签, 以此来让分类器学会分类对抗样本[6, 21], 该方法也可以认为是一种数据加强的手段。

对抗样本训练是符合直觉的, 但是该方法的缺陷也非常明显: 应该使用哪种攻击方法生成的对抗样本来训练分类器。实验表明, 用一种攻击方法生成的对抗样本进行训练只能防御该攻击方法, 对于其他攻击方法生成的对抗样本防御会失效。目前做的最好的是 Madry 的对抗样本训练方法, 他们使用映射梯度下降法生成的对抗样本训练分类器。然而, sharma[22]表明 Madry 防御有效的前提是把对抗干扰控制在L^∞的某个小范围球内, 他们提出了一种弹性攻击方法可以生成L^1的攻击样本, 并且成功攻破了 Madry 的防御方法。

对抗样本训练和我们提出的防御方案 IDAE 有很大的不同。对抗样本训练需要修改被保护分类器的参数, 修改后的分类器认为具有更强的鲁棒性。而 IDAE 和被保护的分类器是两个独立的部分, 我们可

以用某数据集独立训练 IDAE, 该 IDAE 可以保护任意在该数据集上训练的分类器, 是一种更加通用的防御方案。

2.4.2 对抗样本检测和恢复

另外一种防御思路是检测出对抗样本, 拒绝该样本或者对该样本进行恢复使其分类正确。MagNet 就属于该方法, MagNet 是目前最先进的防御手段, 它由检测函数和恢复函数两部分组成。检测函数检测一个样本离流形是否过远, 如果是的话拒绝该样本。否则送给恢复函数, 恢复函数是一个自编码器, 把样本送给自编码器恢复后再送给分类器分类。我们的迭代自编码器就是在 MagNet 的基础上改进得到的, 我们希望尽可能地把远离流形的(被 MagNet 拒绝的)对抗样本也恢复正确。当然, 我们的防御手段也可以结合一个检测函数, 将会得到更好的结果。

3 迭代自编码器防御对抗样本

3.1 防御原理

根据之前的工作[7],对抗样本之所以被分类错误, 是因为对抗样本虽然在输入空间某种度量单位下离原样本很近, 但是在流形学习假设中, 却远离流形或者在流形周围,因为分类器泛化能力较差导致分类错误。

根据上述猜测, 寻找对抗样本的过程可以认为是把一个正常在流形上的样本, 移动到到远离流形并且保持寻找到的样本和原样本在输入空间中某种度量单位下距离很小的一个过程, 根据 2.3 章节的介绍, 研究者大多使用输入和真实标签之间损失函数的梯度方法来实现该过程, 使用基于梯度的方法可以看成是深度学习训练的逆过程, 只是学习的权重变成了输入而不是神经网络需要学习的参数。

既然对抗样本是通过将在流形上的样本通过梯度方向, 一步步推到远离流形的地方, 一个非常直观的防御思路是: 实现该过程的逆过程, 把流形外的样本一步步推回到流形上或者周围。因为流形外的样本容易分类错误, 将流形外的样本推回到流形周围就实现了防御的目的。我们提出迭代自编码器来实现上述过程,图二用二维平面的流形来描述迭代自编码器是如何工作的,其中橙色部分表示数据分布密集的流形, 箭尾表示数据样本所在流形外围位置, 迭代自编码器根据箭头的方向把在流形外围的数据样本移动回流形上使其可以被正确分类。

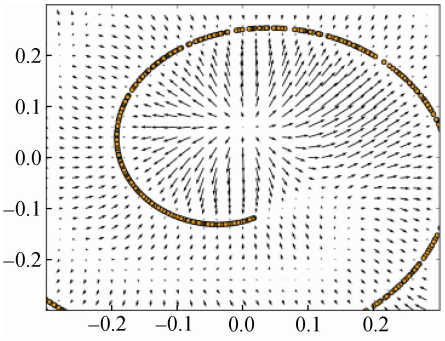


图 2 迭代自编码器工作原理

Figure 2 The principle of IDAE

3.2 防御实现和数学解释

结合对抗样本: 1)视觉上相似(在输入空间某种度量单位下和原样本距离近); 2)离流形较远的特性。我们提出一种新的防御手段, 称之为迭代自编码器, 迭代自编码器由两部分函数组成:

1. 目标: 尽可能把对抗样本移动回流形周围;
2. 正则化: 在输入空间中恢复后的样本和原样本之间的度量距离尽可能小。

正则化项实际上是对抗样本的必要条件, 这是我们第一次反过来利用该条件来防御对抗样本。

下面我们从数学角度来解释这两个部分。假设所有用来训练分类器的图像都是从$prior(x)$真实数据分布中独立采样得到, 但是我们没有这个分布函数, 我们可以利用这些图像来逼近这个分布得到近似分布$\widehat{prior}(x)$, 把输入样本移动回学到的流形上的数学解释为max $\widehat{prior}(x)$,即我们希望恢复后的样本尽可能从真实数据分布中采样得到。对于正则化项, 我们用条件概率表示: $p(\hat{x}|x)$, 其中$\hat{x}$表示恢复后的样本, x表示输入样本, 其含义是在观测到输入样本的时候, 恢复后的样本有多大概率被观测到, 随着$\hat{x}$和x度量距离的增加, 条件概率越小。于是, 迭代自编码器实际上的优化目标函数可写成:

$$\max_{\hat{x}} \widehat{prior}(\hat{x}) + \alpha\, p(\hat{x}|x) \# \tag{6}$$

其中, α是一个超参数, 用来表示正则项的力度。

优化上述目标函数, 需要知道目标函数的形式, 然后通过求导得到梯度来优化。然而我们并不知道$\widehat{prior}(\cdot)$函数的形式, 但是我们却可以通过去噪自编码器知道该函数的梯度。文献[8-9]已经表明最优去噪自编码器可以把输入往流形的方向上推, 数学表示就是:

$$\nabla_x \log prior(x) \propto ae(x) - x \ \# \tag{7}$$

因此, 我们可以使用去噪自编码器恢复后的样本和输入样本之间的方向来表示$\widehat{prior}(\cdot)$的梯度方向。

对于正则化项, 我们希望恢复后的样本和原样本之间度量距离尽可能近, 因此我们直接用度量距离的负数来表示条件概率: $p(\hat{x}|x) = -\frac{1}{2}||\hat{x}-x||_2^2$ 随着恢复后的图像和原图像在度量距离上越远, 条件概率变得越小,该形式的优点是导数非常容易求解:$\nabla_{\hat{x}} p(\hat{x}|x) = x - \hat{x}$。

有了上述定义, 我们可以先把公式 6 转化为最小化问题:

$$\min_{\hat{x}} \frac{1}{2}\alpha||\hat{x}-x||_2^2 - \widehat{prior}(x)\# \tag{8}$$

对于公式(8), 结合公式(7)我们可以求解每一步的梯度方向:

$$-(ae(\hat{x}) - x) + \alpha(\hat{x} - x)\# \tag{9}$$

于是就可以使用梯度下降法求解公式(8)的优化目标函数, 得到的解就是希望寻找的离流形更近并且和原样本度量距离小的恢复后样本。图 3 详细描述了迭代自编码器防御的方案。

综上所述, 恢复后的样本有离流形更近并且和输入样本在L^2度量单位内距离小的特点。实际上, 对于条件概率$p(\hat{x}|x)$, 可以完全不局限于L^2度量单位, 如果我们知道攻击者是通过哪种度量单位产生的攻击样本, 就可以使用相应的度量单位来做正则化项, 只是优化方法会有所不同。我们使用 2 范数的原因是导数容易求解, 并且我们的危险模型是防御方案不知道攻击者产生对抗样本的方法和度量单位。在实验中, 我们表明即使使用 2 范数做正则化项, 我们仍然可以防御基于L^∞度量单位产生的攻击样本, 图四展示了对于某个对抗样本, 迭代自编码器是如何将该样本一步步转换并且被分类正确的, 第一行的最左边图片是正常的图片, 第二行和第三行最左边的图片表示用 C&W 攻击方法产生的L^2对抗样本和用 PGD 攻击方法产生的L^∞攻击样本。最上面的列表示迭代自编码器的迭代次数, 三行经过 1000 次迭代之后最后恢复样本都被分类器正确分类。并且, 假设我们知道攻击者是基于某种度量单位产生的对抗样本, 使用相应的度量单位作为正则化项, 我们实验发现迭代自编码器能取得更好的防御效果。

Require: $\boldsymbol{x_0}$: original example. ae: autoencoder. α: regularization strength
function IDAE($\boldsymbol{x_0}, ae, \alpha$)
  $\hat{\boldsymbol{x}} \leftarrow \boldsymbol{x_0}$
  $gradient \leftarrow \mathbf{0}$
  $i \leftarrow 0$
  for $i \in [1, \ldots, max\_iteration]$ **do**
    $g_0 \leftarrow ae(\hat{\boldsymbol{x}}) - \hat{\boldsymbol{x}}$
    $g_1 \leftarrow \hat{\boldsymbol{x}} - \boldsymbol{x_0}$
    $direction \leftarrow \alpha * g_1 - g_0$
    $gradient \leftarrow \mu * gradient + \lambda * direction$ ▷ μ: momentum hyperparameter. λ: step size
    $\hat{\boldsymbol{x}} \leftarrow \hat{\boldsymbol{x}} - gradient$
    $\hat{\boldsymbol{x}} \leftarrow clip(\hat{\boldsymbol{x}}, 0, 1)$
  return $\hat{\boldsymbol{x}}$

图 3　迭代自编码器防御对抗样本方案
Figure 3　IDEA's defense code

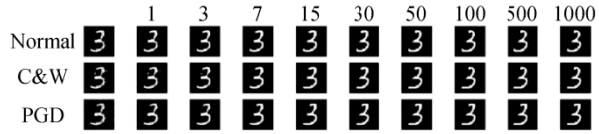


图 4　迭代自编码器迭代过程中的图像
Figure 4　An illustration of how IDAE transforms normal and adversarial examples

4　攻击迭代自编码器

假设攻击者知道迭代自编码器的存在和迭代自编码器的所有参数, 攻击者能否带着迭代自编码器和分类器一起攻击产生对抗样本？为了验证迭代自编码器的有效性, 我们提出一种白盒的攻击方案来攻击整个防御模型。

2.3 章节中已经提到攻击者大多使用基于梯度的攻击方法攻击模型, 其难点也就是如何连带着迭代自编码器和模型计算相对于输入的梯度。我们根据链式法则来计算分类器的输出相对于输入的梯度。在后续关于迭代自编码器攻击的章节中, 我们不考虑基于 momentum 的优化方法, 迭代自编码器在防御的时候也不使用该优化方法。我们把迭代自编码器和分类器看成一个整体, 该整体的计算图可表示为 $x^0 \rightarrow x^1 \rightarrow x^2 \rightarrow \cdots \rightarrow x^n \rightarrow Classifier \rightarrow Output$. 其中由公式 9 可得:

$$x^{t+1} = x^t - \lambda * \left(\alpha * (x^t - x^0) - (ae(x^t) - x^t)\right)\# \tag{10}$$

其中, λ是每次梯度下降的步伐大小, α表示正则项的力度大小, x^0是原始输入样本。我们先考虑迭代一次的情况, 迭代多次的梯度计算公式可以从迭代一次梯度计算公式中推导出来。定义$x^* = ae(x^0)$,为了计算损失函数相对于x^0的梯度g_0, 我们首先计算损失函数相对于x^*的梯度g_1:

$$g_1 = \begin{bmatrix} \frac{\partial loss}{\partial x_1^*} \\ \vdots \\ \frac{\partial loss}{\partial x_n^*} \end{bmatrix}\# \tag{11}$$

其中, n表示输入样本的维度数量, 然后计算x^*相对于x^0的 Jacobian 矩阵$\boldsymbol{J}$:

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial x_1^*}{\partial x_1^0} & \cdots & \frac{\partial x_1^*}{\partial x_n^0} \\ \vdots & \ddots & \vdots \\ \frac{\partial x_n^*}{\partial x_1^0} & \cdots & \frac{\partial x_n^*}{\partial x_n^*} \end{bmatrix}\# \tag{12}$$

最后使用链式法则$\frac{\partial loss}{\partial x_j} = \sum_i^n \frac{\partial loss}{\partial x_i^*}\frac{\partial x_i^*}{\partial x_j}$得到梯度$g_0$:

$$g_0 = \boldsymbol{J}^T g_1\# \tag{13}$$

如果迭代自编码器迭代很多次, 我们可以根据

公式 13, 乘以多次x^{t+1}相对于x^t的 Jacobian 矩阵得到最后的梯度。对于公式 10 的 Jacobian 矩阵求解, 其中自编码器的 Jacobian 矩阵可以通过 Tensorflow 求解得到, 其他项的 Jacobian 矩阵都非常容易求解, 我们就可以得到每次迭代的 Jacobian 矩阵。对于损失函数相对于分类器的输入梯度g_1, 可以用 Tensorflow 直接求解, 所以公式 13 在每一步都可以求解, 图五展示了如何求解多次迭代的梯度方法。

攻击者想要进行白盒攻击, 需要记录迭代自编码器的每一步中间状态, 然后使用图 5 的方法计算得到输出相对于输入的梯度。有了梯度之后, 攻击者就可以使用基于梯度攻击方法,例如快速梯度方向法和映射梯度下降法构造包含迭代自编码器在内的攻击样本。

Require: $classifier$, ae: denoising autoencoder, $inputs$:a list containing each optimization step's input. λ: gradient decent step size, $label$: ground truth label, x^*: IDAE's output, α: regularization strength.
function COMPUTE GRADIENT($classifier, ae, inputs, \lambda, \alpha, label, x^*$)
  $grad \leftarrow \nabla loss(classifier(x^*), label)$
  for $i \in [max\_iteration, \ldots, 1]$ **do**
    $J = Jacobian(inputs[i], \alpha, \lambda)$
    $grad = J^T grad$
  return $grad$

图 5 白盒攻击包含迭代自编码器的分类器的攻击方法
Figure 5 White-box attack that includes the IDAE

5 实验结果和分析

我们在本章节使用大量实验来验迭代自编码器防御的有效性。对于每个输入样本x , 我们都先让它经过足够迭代次数的迭代自编码器恢复后得到新样本x^*,然后再把x^*送给分类器分类。对于所有的输入样本, 迭代自编码器都不知道输入样本是否是对抗样本, 所以我们希望对于正常样本, 增加迭代自编码器之后, 正常样本仍然应该被正确分类, 对于原分类器分类错误的对抗样本, 经过迭代自编码器之后, 应该被正确分类, 以此达到防御的目的。

我们在两个常用的数据集: MNIST 和 CIFAR-10 上进行实验。对于 MNIST 数据集, 我们训练了一个分类准确率达到 99.4%的深度卷机神经网络(结构如表一所示)来做为被攻击的分类器。与此同时, 我们训练了一个表二结构的去噪自编码器来作为迭代自编码器的基础结构。对于 MNIST 数据集, 我们不使用正则化项(通过把公式 8 中的α设置成 0), 在求解公式 8 的时候, 我们使用 momentum 优化方法, 其中 momentum 大小$\mu = 0.9$, 每一次梯度下降的步伐大小$\lambda = 0.1$, 迭代次数为 50 次。经过对正常样本使用迭代自编码器之后, 正常样本的分类正确率稍微下降到了 97.9%。

对于 CIFAR-10 数据集, 我们训练了一个 DenseNet 分类器[23], 在正常样本上达到了 92.5%的分类正确率, 我们把该分类器作为被攻击的分类器。与此同时, 我们训练了一个去噪自编码器, 其基本结构如文献[24]所示, 我们在每层卷积函数之后增加了一层 batch-normalization。和 MNIST 数据集不同的是, 我们使用正则化项, 其中$\alpha = 0.01$,使用带 momentum 的优化方法, momentum 大小$\mu = 0.95$,每次梯度下降的步伐大小$\lambda = 0.1$, 一共迭代 25 次。经过对正常样本使用迭代自编码器之后, 正常样本的准确率下降到了 84.5%, 虽然下降了 8%正确率, 但是和其他模型相比, 还是在可接受范围之内。

表 1 MNIST 分类器卷机神经网络结构
Table 1 Architecture of the MNIST classifier

层结构	参数
Conv.ReLU	3*3*32
Conv.ReLU	3*3*32
Max Pooling	2*2
Conv.ReLU	3*3*64
Conv.ReLU	3*3*64
Max Pooling	2*2
Dense.ReLU	200
Dense.ReLU	200
Softmax	10

表 2 MNIST 自编码器结构
Table 2 Architecture of the Autoencoder

层结构	参数
Conv. ReLU(stride = 2)	8*8*30
UpSampling	2*2
Conv.Sigmoid	3*3*1

我们以下实验的所有危险模型都是攻击者知道被保护的分类器的所有参数, 例如被保护分类器的结构, 学习到的每一层神经元的权重, 训练时候使用的参数等等, 攻击者除了不能修改分类器, 他可以对被保护分类器做任何事, 例如求导得到输入的梯度, 给分类器输入样本, 得倒每一层的输出结果。这种危险模型, 在之前的文献中[10,12]被称为白盒模型。在接下来的实验中, 我们根据攻击者是否知道迭代自编码器模型的参数来对攻击模型分为如下两类:

- 灰盒模型: 攻击者不知道迭代自编码器的任何参数, 也没有意识到迭代自编码器的存在, 直接在被保护分类器上产生对抗样本。
- 白盒模型: 攻击者知道迭代自编码器的所有

参数, 把迭代自编码器和被保护分类器当成一个整体产生对抗样本。

可以看到, 我们的危险模型对于攻击者是非常有利的, 并且迭代自编码器防御方案是不知道攻击手段的一个通用的防御方案, 我们通过使用目前最先进的几种攻击方式, 来说明 IDAE 防御的有效应。

5.1 灰盒攻击模型

在这种攻击模型中, 攻击者知道被保护分类器的任何参数, 不知道迭代自编码器的存在, 在被攻击分类器中产生对抗样本, 迭代自编码器对产生的对抗样本进行恢复, 然后送给受害分类器分类。我们实验了三种常用的攻击方法: 快速梯度方向法、映射梯度下降法和 C&W 基于 2 范数的攻击方法, 三种攻击方法包含了两种度量单位: L^∞ 和 L^2, 在不知道攻击方式的前提下, 我们都使用 2 范数作为防御的正则项, 说明迭代自编码器可以防御根据多个度量单位产生的对抗样本。对于 MNIST 和 CIFAR-10 两个数据集的测试集, 我们随机挑选了 2000 张被分类器正确分类的图片作为测试迭代自编码器防御性能的测试集。并且, 我们把我们的结果和 MagNet 做对比, 因为 MagNet 包含两个结构: 检测函数和恢复函数, 但是检测函数和恢复函数是相互独立的两个部分, 我们也可以在迭代自编码器中加上检测函数, 因此我们实验中只和 MagNet 的恢复函数做对比。MagNet 和迭代自编码器都不是分类器, 只是防御手段, 在下文中使用 MagNet 和迭代自编码器来表示被 MagNet 和迭代自编码器保护的分类器的分类准确率。和文献[17]中的实验一致, 对于 MNIST 数据集, 我们把图片归一化到 0~1 之间, 因此对抗样本干扰大小ε也在 0~1 之间。对于 CIFAR-10 数据集, 我们不进行归一化, 对抗干扰是 0~255 的整数。

图 6a)和图 6b)分别描述了在 MNIST 数据集上迭代自编码器防御快速梯度方向法和映射梯度下降法生成的对抗样本的鲁棒性。这两种方法都是基于L^∞度量单位产生的攻击样本, 都使用ε来描述可修改的L^∞球的大小, 随着ε的增加, 对抗干扰大小增加, 分类器越难正确分类对抗样本, 防御也越来越不容易成功。从图中可以看出, 对于这两种攻击方式, 当ε小于 0.3 的时候, 对于对抗样本, 迭代自编码器都拥有 80%以上的分类正确率。和 MagNet 相比, 在$\varepsilon = 0.3$的时候, MagNet 对于梯度映射下降法产生的对抗样本的分类正确率低于 20%。并且在$0.4 \leqslant \varepsilon \leqslant 0.5$的情况下, 迭代自编码器仍然拥有超过 60%的分类正确率。综上所述在 MNIST 数据集使用L^∞度量单位产生的对抗样本上, 迭代自编码器的恢复效果明显要好于 MagNet 的恢复效果。

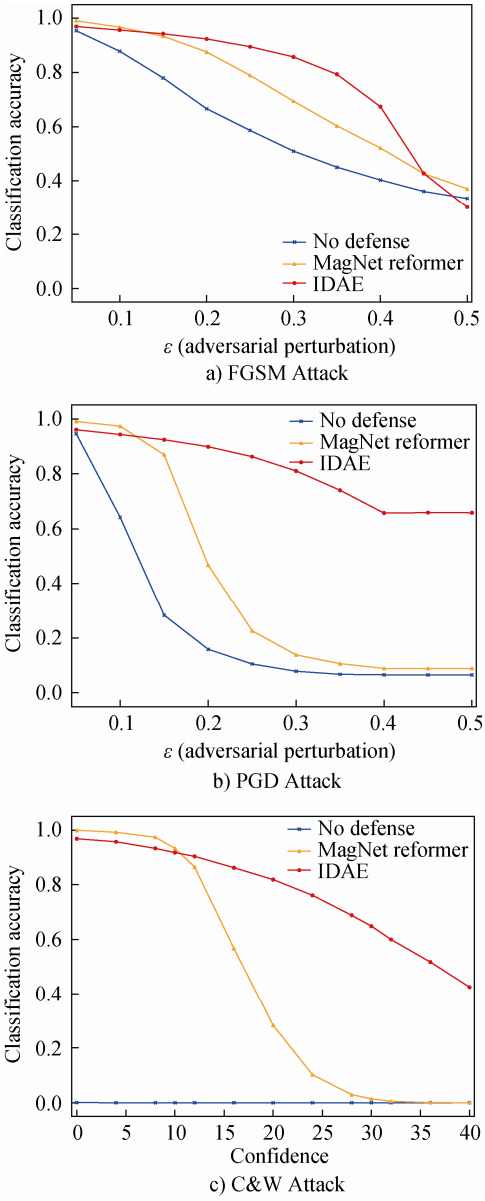


图 6 在 MNIST 数据集上分类器在基于灰盒攻击生成的对抗样本上的分类准确率

Figure 6 Classification accuracy on adversarial examples generated in gray-box attacks on MNIST

图 6c)描述了迭代自编码器在 MNIST 数据集上防御 C&W 攻击方法基于L^2度量单位产生的对抗样本的鲁棒性。C&W 攻击方法使用置信值参数k来描述对抗干扰的大小, 通常来说置信值参数k越大, 对抗干扰越大, 分类器越难正确分类产生的对抗样本。从图中可以看出, 当置信值等于 20 的时候, MagNet 的准确率只有 30%左右, 而迭代自编码器仍然拥有超过 80%的分类准确率。并且迭代自编码器对于对抗样本的分类准确率下降非常缓慢, 没有在某个置

信值之后出现突然的准确率下降。

图 7a)和图 7b)分别描述了在CIFAR-10 数据集上迭代自编码器防御快速梯度方向法和映射梯度下降法生成的对抗样本的鲁棒性。从图中可以看到随着ε的增长, MagNet 的防御效果迅速下降, 而迭代自编码器的防御效果则下降缓慢很多。例如对于映射梯度下降法产生的对抗样本在$\varepsilon = 5.0$的时候, MagNet 的分类准确率已经低于 10%, 而迭代自编码器的分类准确率仍然有 70%左右。所以在 CIFAR-10 数据集上使用L^{∞}度量单位产生的对抗样本, 迭代自编码器的恢复效果要明显好于 MagNet。

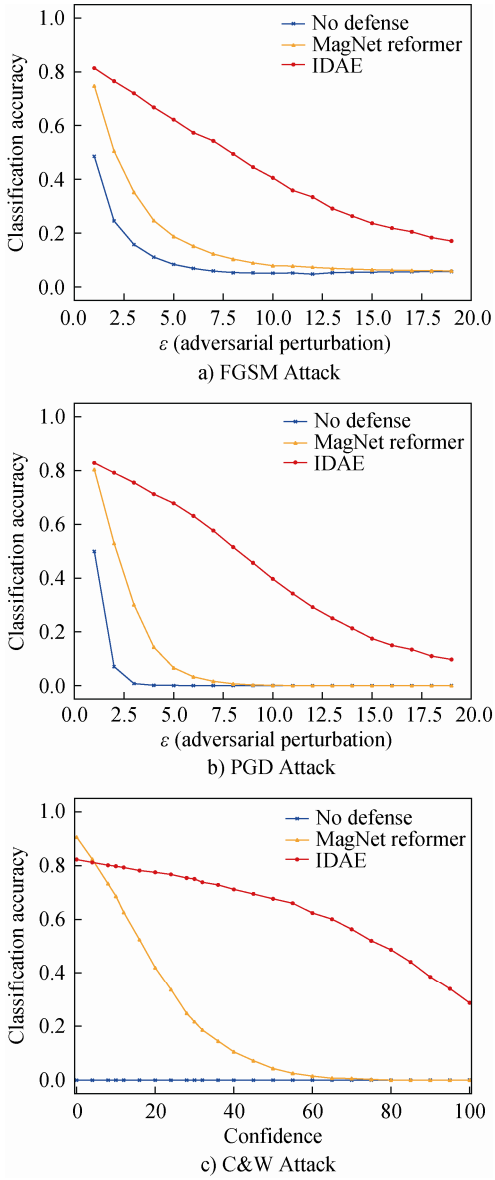


图 7　在 CIFAR-10 数据集上分类器在基于灰盒攻击生成的对抗样本上的分类准确率

Figure 7　Classification accuracy on adversarial examples generated in gray-box attacks on CIFAR-10

图 7c)描述了迭代自编码器在 CIFAR-10 数据集上防御 C&W 攻击方法基于L^2度量单位产生的对抗样本的鲁棒性。从图中可以看到当置信值$k = 60$的时候, MagNet 的分类准确率已经下降到了 0%, 而迭代自编码器的分类准确率仍然高于 60%。

表 3 和表 4 从数值上描述了 IDAE 在 MNIST 和 CIFAR-10 上的有效性。从表 3 可以发现 IDAE 对于对抗样本有良好的防御能力, 对于越强大的攻击方法和越大的对抗干扰样本, IDAE 的防御效果比 MagNet 要强很多。值得注意的是, IDAE 达到这样高的分类准确率, 没有针对任何特定方法产生的对抗样本进行训练, 是个通用的防御方案。

表 3　IDAE 和 Magnet 在 MNST 数据集上的分类准确率

Table 3　Accuracy of IDEA and Magnet in MNIST dataset

攻击方法	参数	无防御	MagNet	IDAE
FGSM	$\varepsilon = 0.1$	87.85%	96.1%	95.65%
FGSM	$\varepsilon = 0.3$	51%	71.2%	85.75%
PGD	$\varepsilon = 0.1$	64.15%	95.5%	94.3%
PGD	$\varepsilon = 0.3$	7.85%	14.5%	81.5%
C&W	$k = 0$	0%	98.35%	92.9%
C&W	$k = 30$	0%	1.45%	71.35%

表 4　IDAE 在 CIFAR-10 数据集上的分类准确率

Table 4　Accuracy of IDEA and Magnet in CIFAR-10 dataset

攻击方法	参数	无防御	MagNet	IDAE
FGSM	$\varepsilon = 1.0$	48.6%	76.4%	83.05%
FGSM	$\varepsilon = 8.0$	5.3%	11.6%	47.8%
PGD	$\varepsilon = 1.0$	50%	80.7%	82.2%
PGD	$\varepsilon = 8.0$	0%	0.8%	48.95%
C&W	$k = 0$	0%	90.3%	82%
C&W	$k = 60$	0%	1.5%	62.5%

在灰盒攻击模型下, IDAE 在对抗样本干扰较小的情况下, 有些时候防御效果不如 MagNet, 我们猜测主要原因是去噪自编码器不是最优造成的。文献[8-9]表明一个理论最优的去噪自编码器可以准确学习到输入样本的流形, 但是在现实实验中, 因为模型复杂度、模型结构、计算精度等原因, 无法准确学习到输入样本的流形, 只能尽可能去拟合样本的真实流形, 导致学到的流形和真实的流形之间存在偏差(文献[8]中的图 4 展示了自编码器模型不同参数导致学到的流形的偏差)。干扰较小的样本, 本身就在流形的周围, 因为学到的流形偏差, 导致样本被移动到拟合的流形周围, 导致分类错误, 这也是

IDAE 导致分类器分类准确率下降的原因。而 MagNet 在该场景下防御较好是因为在流形空间只移动了一步的距离, 仍然在真实流形的周围。

5.2 白盒攻击模型

在白盒攻击模型中, 攻击者知道迭代自编码器的存在和迭代自编码器的所有参数, 将被保护的分类器和迭代自编码器当成一个整体产生对抗样本。我们在白盒攻击模型下, 在 MNIST 和 CIFAR-10 数据集上实验了两种攻击方法: 快速梯度方向法和映射梯度下降法。在 MINST 数据集上实验了 C&W 攻击方法, 没有在 CIFAR-10 上实现 C&W 攻击方法是因为 C&W 攻击方法计算量非常巨大, 使用图五的方法计算梯度, 即使在 50 张 CIFAR-10 图片上使用映射梯度下降法, 使用 P40 GPU, 产生一种对抗干扰大小 ε的对抗样本也需要花费 3 天的时间。在我们接下来的实验中, 防御的时候, 迭代自编码器不使用 momentum 优化方法, 在 MNIST 数据集上优化 250 步, 在 CIFAR-10 数据集上优化 200 步, 每次优化的步伐大小为 0.1。也就是说, 使用图五的方法计算梯度的时候, MNIST 数据集需要迭代计算 250 次, CIFAR-10 数据集需要迭代计算 200 次。下文中, 使用迭代自编码器指代迭代自编码器和被其保护的分类器的整体。

图 8 描述了在白盒攻击模型下, 迭代自编码器防御快速梯度方向法产生的对抗样本的有效性。我们在 MNIST 和 CIFAR-10 两个数据集上, 随机选择了 100 个被迭代自编码器分类正确的样本用于产生对抗样本。从图 8a)中可以得出, 在 MNIST 数据集上, 在 $\varepsilon = 0.4$的情况下, 迭代自编码器的分类准确率仍然有 65%, 而单纯的分类器只有 40%左右的正确率。图 8b)中可以看出, 在 CIFAR-10 数据集上, 迭代自编码器在$\varepsilon = 7$的情况下, 迭代自编码器的分类准确率仍然有 51%, 而单纯的分类器只有 0.05%的正确率。

图 9 描述了在白盒攻击模型下, 迭代自编码器防御映射梯度下降法产生的对抗样本的有效性。我们在 MNIST 数据集和 CIFAR-10 数据集上分别采样了 100 个和 50 个迭代自编码器分类正确的样本用于生成对抗样本。从图 9a)中可以得出, 在 MNIST 数据集上, 对于最难的对抗样本$\varepsilon = 0.4$的情况, 迭代自编码器仍然拥有 68%的分类准确率, 单纯的分类器的分类准确率只有 0.065%。从图 9b)中可以得到, 在 CIFAR-10 数据集上, 最难的对抗样本的分类准确率从单纯分类器的 0%提升到了 38%。

在白盒攻击模型下, 我们使用 C&W 攻击成功率来说明迭代自编码器的防御有效性。由于 C&W 攻击

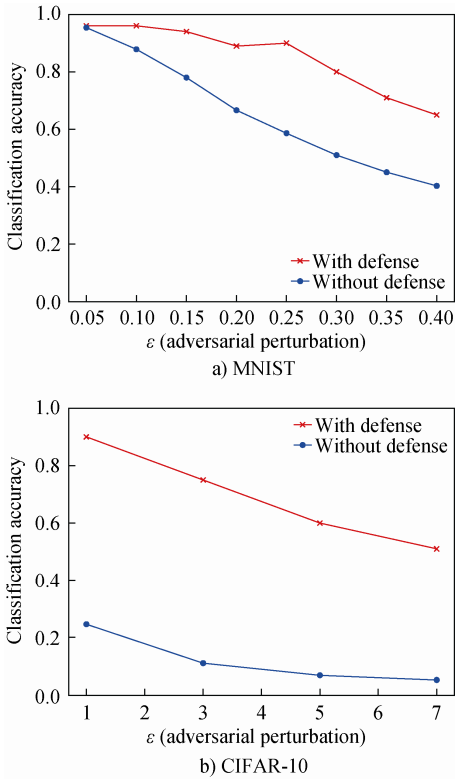


图 8 在白盒攻击模型下, 迭代自编码器对于快速梯度方向法产生的对抗样本的分类准确率

Figure 8 Classification accuracy on adversarial examples generated by FGSM in white-box attack

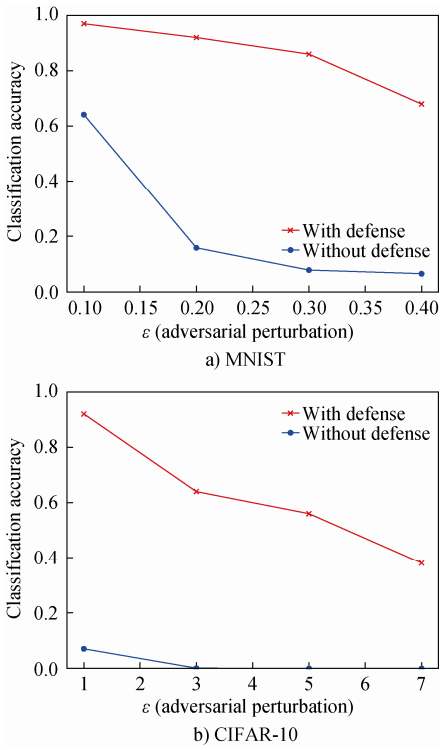


图 9 在白盒攻击模型下, 迭代自编码器对于映射梯度下降法产生的对抗样本的分类正确率

Figure 9 Classification accuracy on adversarial examples generated PGD in white-box attack

需要巨大的计算量, 我们在 MNIST 数据集上随机选择了 100 张被分类正确的图片进行测试。图 10 说明了迭代次数和 C&W 攻击成功率(置信值$k = 0$, 攻击者最有利的情况下)之间的关系, 我们希望通过该图说明随着迭代次数的增加, 最后得到的整体模型拥有更好的鲁棒性。从图 10 中可以看到, 在迭代次数小于 30 次的时候, C&W 有将近 100%的攻击成功率。然而, 随着迭代次数的增加, C&W 攻击成功率 开始下降, 当进行 100 次迭代防御的时候, C&W 作为目前最先进的攻击方案, 只有 15%的攻击成功率。充分说明了迭代自编码器的鲁棒性和迭代防御方案的有效应。

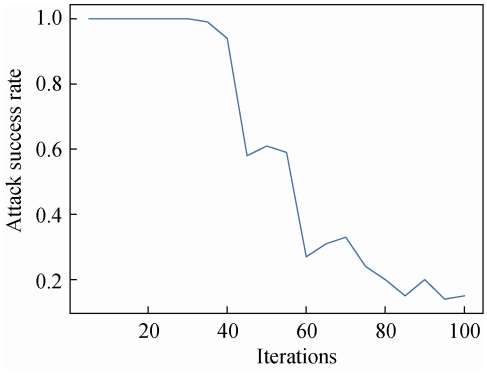


图 10 在白盒攻击模型下, C&W 方法攻击迭代自编码器的攻击成功率和迭代次数之间的关系

Figure 10 Success rate of C&W attack on victim classifier protected by IDAE

ZOO[20]是一种可以在不知道模型参数的情况下产生对抗样本的攻击方案。按照 ZOO 论文中攻击模型的定义, ZOO 应该算黑盒攻击模型。但是在我们的实验中, 我们认为 ZOO 算一种白盒攻击模型, 我们给 ZOO 提供 IDAE 和被保护分类器的所有参数, 只是 ZOO 没有利用这些参数计算梯度, 而是利用一阶优化方法来近似求解梯度。之所以认为 ZOO 是白盒模型, 是因为 ZOO 在攻击的时候知道迭代自编码器的存在, 并且连带着迭代自编码器一起攻击。我们在 MNIST 和 CIFAR-10 数据集上各随机选择了 100 张被分类正确的图片来进行实验, 实验中所使用的所有参数和文献[20]中一样, 并且和 C&W 实验一样选择对攻击者最有利的参数, 把置信值设置为 0。表五展示了我们的实验结果。

从表 5 可以看出, 在没有迭代自编码器防御的情况下, ZOO 在 MNIST 和 CIFAR-10 数据集上都有 100%的攻击成功率。然而, 当我们用迭代自编码器保护分类器的时候, 在 MNIST 数据集上 ZOO 只有 4%的攻击成功率, 在 CIFAR-10 数据集上只有 15%的攻击成功率, 说明了迭代自编码器防御的有效应。

表 5 ZOO 对于迭代自编码器的攻击成功率

Table 5 Success rate of the ZOO attack on the victim classifier

数据集	防御	攻击成功率
MNIST	否	100%
	是	4%
CIFAR-10	否	100%
	是	15%

综上所述, 在白盒攻击模型下, 我们通过各种攻击方法, 说明了迭代自编码器是一个对各种攻击都有效的通用防御方案, 并且具有良好的鲁棒性。

6 结论

随着人工智能应用的火热, 人工智能安全问题越来越受到大家的关注, 特别是在安全性要求较高的场景, 例如自动驾驶。深度学习是人工智能领域中重要的分支, 研究者们会在自动驾驶等各个领域使用深度学习, 随着 Goodfellow 等人提出深度学习中存在对抗样本, 研究者们开始研究各种方法产生和防御对抗样本来验证深度学习的安全性。

本文提出了迭代自编码器, 一种全新的防御对抗样本的方案, 迭代自编码器通过逆向产生对抗样本的过程, 将远离流形的对抗样本一步步推回到流形的周围使其分类正确, 其数学解释为迭代自编码器在最大化样本从真实分布中采样得到的概率。针对迭代自编码器的防御方案, 我们设计了相应的梯度求解算法, 通过灰盒和白盒两种攻击方案, 验证了迭代自编码器防御的有效应。

参考文献

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[2] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. Ask me anything: dynamic memory networks for natural language processing. *In International Conference on Machine Learning*, pp. 1378–1387, 2016.

[3] Eykholt, K. Evtimov, I. Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T. and Song, D. 2018 Robust Physical-World Attacks on Deep Learning Visual Classification. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*(pp. 1625-1634).

[4] Nicholas Carlini and David Wagner. Towards evaluating the ro-

bustness of neural networks. *In IEEE Symposium on Security and Privacy*, 2017.

[5] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. *Intriguing properties of neural networks. In International Conference on Learning Representations (ICLR)*, 2014.

[6] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. "Towards deep learning models resistant to adversarial attacks". *In: arXiv preprint arXiv*:1706.06083 (2017).

[7] Dongyu Meng and Hao Chen. "MagNet: a two-pronged defense against adversarial examples". *In: ACM Conference on Computer and Communications Security (CCS)*. Dallas, TX, Oct. 30–Nov. 3, 2017.

[8] Guillaume Alain and Yoshua Bengio. "What regularized auto-encoders learn from the data-generating distribution". *In: The Journal of Machine Learning Research 15.1 (2014)*, pp. 3563–3593.

[9] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. "Extracting and composing robust features with denoising autoencoders". *In: Proceedings of the 25th international conference on Machine learning. ACM. 2008*, pp. 1096–1103.

[10] Geoffrey E Hinton and Richard S Zemel. "Autoencoders, minimum description length and Helmholtz free energy". *In: Advances in neural information processing systems. 1994,* pp. 3–10.

[11] Christopher Poultney, Sumit Chopra, Yann L Cun, et al. "Efficient learning of sparse representations with an energy-based model". *In: Advances in neural information processing systems. 2007*, pp. 1137–1144.

[12] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. "Domain adaptation for large-scale sentiment classification: A deep learning approach". *In: Proceedings of the 28th international conference on machine learning (ICML-11). 2011*, pp. 513–520.

[13] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and PierreAntoine Manzagol. "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion". *In: Journal of Machine Learning Research 11.Dec (2010)*, pp. 3371–3408.

[14] Junyuan Xie, Linli Xu, and Enhong Chen. "Image denoising and inpainting with deep neural networks". *In: Advances in neural information processing systems. 2012*, pp. 341–349.

[15] Xiao-Jiao Mao, Chunhua Shen, and Yu-Bin Yang. "Image restoration using convolutional auto-encoders with symmetric skip connections". *In: arXiv preprint arXiv:1606.08921 (2016)*.

[16] Kin Gwn Lore, Adedotun Akintayo, and Soumik Sarkar. "LLNet: A deep autoencoder approach to natural low-light image enhancement". *In: Pattern Recognition 61 (2017)*, pp. 650–662.

[17] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. "Contractive auto-encoders: Explicit invariance during feature extraction". *In: Proceedings of the 28th International Conference on International Conference on Machine Learning*. Omnipress. 2011, pp. 833–840.

[18] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *In International Conference on Learning Representations (ICLR)*, 2015.

[19] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. *arXiv preprint arXiv:1610.08401*, 2016.

[20] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models". *In: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security. ACM. 2017*, pp. 15–26.

[21] Uri Shaham, Yutaro Yamada, and Sahand Negahban. Understanding adversarial training: increasing local stability of neural nets through robust optimization. *arXiv preprint arXiv* :1511.05432, 2015.

[22] Yash Sharma and Pin-Yu Chen. "Attacking the Madry Defense Model with L1-based Adversarial Examples". *In: arXiv preprint arXiv*:1710.10733 (2017).

[23] Huang, G., Liu, Z., Van Der Maaten, L. and Weinberger, K.Q., 2017, July. Densely connected convolutional networks. *In CVPR* (Vol. 1, No. 2, pp. 3).

[24] Xiao-Jiao Mao, Chunhua Shen, and Yu-Bin Yang. "Image restoration using convolutional auto-encoders with symmetric skip connections". *In: arXiv preprint arXiv*:1606.08921 (2016)

杨浚宇 于 2016 年在西南交通大学计算机科学与技术专业获得学士学位。现在中国科学院大学上海微系统所攻读计算机科学与技术专业硕士学位。研究领域为机器学习安全，机器学习在计算机安全中的应用。
Email：yangjy@shanghaitech.edu.cn