硕士学位论文

深度学习对抗攻击防御策略的研究与实现

THE RESEARCH AND IMPLEMENTATION OF THE DEFENSE STRATEGY OF DEEP LEARNING AGAINST ADVERSARIAL ATTACK

张成

哈尔滨工业大学

2020 年 6 月

国内图书分类号：　　学校代码：10213
国际图书分类号：　　密级：公开

硕士学位论文

深度学习对抗攻击防御策略的研究与实现

硕士研究生：张成
导　　师：王玲教授
申请学位：工程硕士
学　　科：计算机技术
所在单位：计算机科学与技术学院
答辩日期：2020 年 6 月
授予学位单位：哈尔滨工业大学

Classified Index: TP315
U.D.C: 681.3

Dissertation for the Master Degree

THE RESEARCH AND IMPLEMENTATION OF THE DEFENSE STRATEGY OF DEEP LEARNING AGAINST ADVERSARIAL ATTACK

Candidate:	Zhang Cheng
Supervisor:	Prof. Wang Ling
Academic Degree Applied for:	Master of Engineering
Speciality:	Computer Technology
Affiliation:	School of Computer Science and Technology
Date of Defence:	June, 2020
Degree-Conferring-Institution:	Harbin Institute of Technology

摘　要

近年来，深度学习技术在多个领域取得十分优异的成就，如计算机视觉、自然语言处理和语音处理等，越来越多的相关应用出现，这给人们的生活带来了很多便利。然而，深度学习在表现优异的同时，还存在易受攻击的缺陷。攻击方法通过在输入样本中增加一些微小的扰动，便可以让深度学习模型做出错误判断。这给深度学习应用带来了巨大的安全问题。当前，已有很多关于防御对抗攻击的研究成果出现。然而，多数防御策略只能应对特定的攻击方法，普适性比较低，另外这些防御策略多为全局过滤扰动，浪费了大量算力。因此，本文针对上述缺点，试图提出一种高效且普适性高的防御策略。本文主要从以下三个方面展开详细研究：

(1) 首先详细分析了多种经典对抗攻击方法，并对这些攻击方法按照攻击范围分类。从分类中抽象出不同点与共同点，发现对抗扰动中存在敏感点，敏感点的波动影响深度学习模型的分类。受此启发，我们提出一种仅过滤对抗样本中敏感点的防御策略，避免了对非敏感点的处理，减少了计算，提高了效率。

(2) 详细研究了寻找敏感点与过滤敏感点的方法。为了提高防御策略的普适性，我们提出了一种基于差分进化算法寻找敏感点的方法，采用黑盒的方式去寻找对抗样本中的敏感点，避免了对深度学习模型的结构、参数等的依赖，可以适用于未知模型细节时的防御，增加了普适性。我们提出新的过滤敏感点方法是将敏感点周围相邻非敏感点像素值的平均值作为敏感点新值，这种过滤方法计算简单且高效。我们通过将寻找敏感点和过滤敏感点两部分相结合构成高效且普适性较好的防御策略。

(3) 为了验证我们所提出的防御策略的有效性与普适性，我们分别实验了FGSM、BIM、PGD三种攻击方法在ResNet、AlexNet、LeNet三种深度学习模型和Cifar-10、MNIST两种数据集上的防御效果。实验结果表明，我们所设计的防御方法可以有效地防御住不同深度学习模型和不同数据集上的不同攻击方法的攻击，这表明我们所提出的防御策略具有一定的普适性。

关键词：深度学习；对抗攻击；防御策略；敏感点

Abstract

In recent years, deep learning technology has made excellent achievements in many fields, such as computer vision, natural language processing and speech processing, etc., and more and more related applications appear, which brings a lot of convenience to people's life. However, deep learning is not only excellent, but also vulnerable to attack. The attack method can make the deep learning model make false prediction by adding some subtle perturbations to the original samples. This brings a huge security problem to the application of deep learning. However, most of the current defense strategies waste a lot of computing power for global filtering disturbance. In addition, most of the defense strategies can only deal with specific attack methods, with low universality. Therefore, in view of these shortcomings, this paper attempts to propose an efficient and universal defense strategy. In this paper, the following three aspects are studied in detail:

(1) In this paper, we first analyze a variety of classic adversarial attack methods, and classify them according to the attack range. The common characteristics are abstracted from the classification, that is, there are sensitive points in the adversarial perturbations, and the fluctuation of sensitive points affects the classification of deep learning model. Inspired by this, we propose a defense strategy that only filters the sensitive points in the adversarial samples, avoiding the processing of the non-sensitive points, reducing the calculation and improving the efficiency.

(2) This paper studies the methods of finding and filtering sensitive points in detail. In order to improve the universality of the defense strategy, we propose a method to find sensitive points based on differential evolution algorithm. We use black box method to find sensitive points in the adversarial samples, avoiding the dependence on the structure and parameters of the deep learning model, which can be applied to the defense of the unknown model details and increase the universality. We propose a new method of filtering sensitive points, which takes the average value of the adjacent non-sensitive points around the sensitive points as the new value of sensitive points. This method is simple and efficient. We combine the two parts of finding and filtering sensitive points to form an efficient and universal defense

strategy.

(3) In order to verify the effectiveness and universality of the defense strategy proposed by us, we respectively tested the defense effects of FGSM, BIM and PGD attack methods on three deep learning models of ResNet, AlexNet and LeNet and two data sets of Cifar-10 and MNIST data. The experimental results show that the defense method we designed can effectively defend against different attack methods on different deep learning models and different data sets, which shows that the defense strategy we proposed has certain universality.

Keywords: deep learning, adversarial attack, defense strategy, sensitive point

目 录

第1章 绪论

1.1 研究背景和意义

随着硬件计算能力的提升，深度学习的发展成为可能。深度学习因其在计算机视觉等领域中的出色表现，越来越多的深度学习应用应运而生。然而深度学习表现优异的同时还存在天生的弱势。2014 年 Szegedy 等人[1]发现深度学习极易受到对抗样本的攻击。攻击者通过对输入样本增加一些轻微的扰动，增加扰动后的样本使人的肉眼很难察觉，即人仍能做出正确分类，但是深度学习模型却不能做出正确判断。这使得大量基于深度学习的应用面临着严重的安全问题。Eykholt 等人[2]发现对真实物理世界中的交通标识添加一些扰动，便可以成功攻击广泛应用于无人驾驶领域的深度学习系统，使得“STOP”标志被识别为其他标志。Sitawarin 等人[3]发现对抗样本可以使得深度学习对一些非交通标志做出误分类，比如把 KFC 图标分类为“STOP”标志，这大大扩展了对抗样本的威胁范围。对抗样本除了能够欺骗基于 CNN 的决策器外，循环深度学习[4]、深度强化学习[5,6,7]、语义分割和目标检测[8-13]等网络也未能幸免于难。

深度学习的易被攻击性，使得大量基于深度学习的应用面临严重安全威胁。比如基于深度学习的无人驾驶汽车会通过识别道路上的标志做出决策。如果对这些输入信息进行攻击，即在交通标志上增加一些极小的噪声，虽然肉眼依旧能够判断出标志的正确类别，但是无人车却会做出错误判断，这将有可能导致交通事故得发生。

为提高深度学习应用的安全性和鲁棒性，对对抗攻击的防御策略的研究显得很重要。自 Szegedy 等人发现深度学习易受攻击以来，各种针对深度学习的对抗攻击方法层出不穷。针对对抗攻击的防御方法也在不断出现。每当有新的防御方法出现，就会有新的使防御方法无效的对抗攻击方法出现；每当有新的对抗方法出现，又会研究出新的应对对抗攻击方法攻击的防御方法。为使得基于深度学习的应用能够安全运行，必须要加快防御策略的研究并且要让防御策略尽可能的能够抵御多种对抗攻击。

1.2 研究现状

为了应对深度学习中不同的对抗攻击方法的攻击，研究人员研究出很多对抗防御方法。防御方法是一种提高深度学习安全性的方法，与对抗攻击方法相对立。当前的防御方法可以分为两大类——数据层面的防御方法和模型层面的防御方法。

1.2.1 数据层面防御

数据层面的防御方法又可分为在训练阶段修改模型参数，比如常用的对抗训练[27,28]，和在测试阶段修改输入数据，如数据压缩。

对抗训练是指通过对抗攻击方法生成大量对抗样本，然后纠正对抗样本的类别后作为训练样本，传入深度学习，构建鲁棒性更好的模型。对抗训练之所以可以起到对抗防御的效果，是因为将对抗样本传入深度学习后，模型学到了对抗样本中的特征，当测试阶段再次见到相似特征的对抗样本后，便不再把其当作对抗样本，起到了防御作用。Goodfellow 等人[14]和 Huang 等人[15]采用对抗训练的方式在 MNIST 上进行试验，试验结果表明这种将原始样本与生成的对抗样本放在一起进行训练的深度学习具有更强的抵御对抗攻击的能力。然而 Kuraki 等人[16]通过在 ImageNet 数据集上实验发现，对抗训练的抵御攻击能力是有限的，它只能抵御住 FGSM 方法[14]产生的对抗样本，而对迭代类攻击方法不起作用。Moosavidezfooli[7] 指出,对抗训练只能使模型对训练集中对抗样本具有很好的鲁棒性，但是理论上讲对抗样本具有无穷多个，因此不管增加多少对抗样本，依然存在新的对抗样本使得深度学习做出错误判断。这是因为，对抗训练在训练过程中加入的对抗样本只能是某种或某些已知的攻击方法产生的对抗样本，因此对抗训练方法通常不能对其他新兴或者未知的攻击方法起到防御作用。并且，对抗训练在训练阶段需要生成很多的对抗样本，计算成本较大，这就导致对抗训练很难适用于大规模数据集。

测试阶段修改输入数据的防御方法通常可以分为输入变换和数据压缩。输入变换方法不需要改变模型结构，而是对输入的预测样本进行各种变换以期减少样本中的噪音，然后再将处理后的样本传入深度学习分类。Luo 等人[17]通过平移输入样本成功防御了 L-BFGS[1]和 FGSM[14]生成的对抗样本。但这种方法只能抵御部分弱攻击，对更强大的攻击失效。Bhagoji 等人[11]提出利用主

成分分析对输入数据进行压缩，以获得对抗的鲁棒性。然而，Xu 等人[12]指出，这种压缩也会破坏图像的空间结构，因此常常会对分类性能产生负面影响。Guo 等人[18]对输入预测样本执行了多种图像变换方法，如随机裁剪与缩放，提高了模型的鲁棒性。然而实验表明，输入变换防御方法具有较高的误报率和漏报率。数据压缩技术是指对输入样本进行压缩以期减少扰动对输出结果的影响。Das 等人[19]采用 JPEG 压缩方法，缺点是这种数据压缩方法不能防御迭代式、扰动细微型的攻击，即适用范围有限，并且实验结果表明，只有在对抗样本中的扰动较少时数据压缩才会起到一定的防御作用。

1.2.2 模型层面防御

模型层面的防御方法可以分为两种，一种为修改网络；另一种为使用附加网络，即在保持原深度学习模型不变的情况下，在该模型之前加一层网络用来防御对抗攻击。

修改网络的防御方法有更改损失函数或激活函数、防御蒸馏和正则化等。更改损失函数或激活函数，使得该函数对一个小的对抗性扰动不太可能大幅改变训练模型的输出,结果表明，该方法与蛮力对抗训练相结合，对 FGSM 和 JSMA[20]等攻击具有很好的鲁棒性[21]。蒸馏是把复杂模型的知识提取至简单模型中，该方法最早由 Hinton[22]提出，该方法可以抵御微小扰动的攻击，另外该方法不能应对黑盒攻击。正则化防御是指在训练阶段在目标函数上惩罚模型输出随着输入变化的情况，通过惩罚使得微小扰动在一定程度上不会影响模型的输出类别。Lyu 等人[23]用正则化方法来实验正则化防御方法对 FGSM 对抗攻击方法的防御效果，实验表明较小的对抗扰动不会对模型的判断做出误导。Moosavidezfooli 等人[24]提出曲率正则化，该方法能够最小化损失面的曲率。这种正则防御方法可以提高深度学习模型的抗攻击能力，但也存在缺点，即该正则化可能会导致深度学习模型的准确率下降，也就是说模型鲁棒性的提高是以牺牲模型的准确率为代价的。

使用附加网络的防御方法有防御网络和基于 GAN 防御。防御网络是指在原始深度学习模型之前附加一层神经网络，该附加的神经网络作用是学习对抗样本特征然后校正对抗样本，目的是使得原始深度学习模型对对抗样本的分类与原未加扰动时的预测一致。看以看出，这种对抗防御方法不需要修改原始深

度学习的参数。Akhtar 等人[25]提出了一种在原始网络前增加一个防御网络的方法，通过训练防御网络让其学会对扰动后的图像校正，从而使分类器对对抗样本预测与未经扰动的样本预测一致。原始网络是学习分类，该防御网络是学习矫正对抗扰动。Lee 等人[26]通过 GAN 来训练一个能够抵御 FGSM 对抗攻击的网络。GAN 是一个附加网络，在训练阶段会不断矫正对原始输入和对抗样本的分类，使之正确。可以发现，基于 GAN 的防御方法并不会对原始网络产生影响，即不会修改原始网络的参数。基于 GAN 的防御方法可以有效防御对抗攻击，然而如果在训练阶段训练不够深入，那么防御效果会较差，而要训练的彻底一些需要更大的运算和时间开销。这种附加网络类的防御方法因其抽象与分层，把预测与防御抽离开来，因此不会对原始网络造成任何影响，这为迁移学习带来了巨大便利，这促使研究员们研究一些计算代价较低的附加网络类防御方法。

通过比较各个防御方法发现，每一种防御方法都有各自的适用范围，并且针对某一种对抗攻击方法设计的防御方法，往往不能有效的防御其他对抗攻击。防御效果越好的防御模型需要更大的计算量。基于这些观点，我们想要提出一种可以具有普适性并且计算代价较少的防御方法，并且要求此种防御方法不会更改原深度学习模型。要想实现这种防御模型，需要解决两方面的问题：

（1） 提高普适性

我们对防御方法的普适性定义是：防御方法可以抵御未曾见过的攻击方法的攻击，即具有较好的泛化性，可以防御的攻击方法越多普适性就越好。

我们知道当前存在的各种防御方法之所以具有较差的普适性就是因为这些防御方法都是白盒防御，也就是设计之初便已经知道要去防御哪些攻击方法。如果假设我们不知道将来要面对的攻击方法，然后去防御这些攻击，这便是黑盒防御。要想实现这种防御方法，需要找到一种可以抽离出对抗样本中的附加的扰动的算法。我们采用差分进化算法去寻找这些扰动，差分进化算法不需要关心生成对抗样本的具体攻击方法，只需要知道深度学习模型的输出和输入样本的标签即可。

（2）处理扰动

通过以上对抗防御方法的比较可以发现，不同的防御方法对扰动的处理各不相同。有些是全局处理，如数据压缩，有些是处理敏感的像素点，如正则

化。为了减少防御模型的计算代价我们也采用处理部分扰动的方法。这种防御方法要求所需过滤的部分扰动必须对对抗样本的输出产生较大影响，以至于扰动的变动使得深度学习模型的预测发生变化。

1.3 主要研究内容与组织结构

由于深度学习模型在计算机视觉、自然语言处理等领域具有极好的表现，进而在这些领域的应用越来越多。但是深度学习模型表现优异的同时也面临易受攻击的缺点，给深度学习应用带来了安全问题。为提高深度学习模型安全性我们提出了一种新的对抗防御方法。由于当前针对深度学习模型的对抗攻击层出不穷，往往出现一种新的对抗攻击方法，以往的对抗防御方法便不再有效。并且，目前多数防御模型都需要大量的计算代价，比如对抗训练，想要起到很好的防御效果，则需要在训练阶段生成大量的对抗样本，对算力的要求很高。

针对当前对抗防御存在的这些问题，本课题试图提出一种高效的具有普适性的对抗防御模型。试图找到一种只需过滤掉对抗样本中少许几个扰动像素点便可以抵御对抗样本攻击的防御方法。本文主要研究内容如下：

1. 通过分析目前为止存在的经典对抗攻击方法，从原理入手，把对抗攻击方法分为两大类，一类为全局攻击，另一类为局部攻击。通过抽象出两类攻击的优化目标公式并予以比较，发现在全局攻击时如果只有某些分量的数值变化比较大，其他点变化比较小，那么全局攻击的约束条件近似为局部攻击的约束条件。故而得出结论：对抗样本中的扰动是有关键分量的，也就是存在敏感点，仅仅在这些敏感点上添加扰动便足以让深度学习模型对输入样本做出错误预测。这给我们设计防御方法提供了启发：在把输入样本传入深度学习模型做预测之前，我们可以通过演化算法来找到输入样本的敏感像素点，然后对该敏感点执行过滤操作使其变为非敏感点，以此实现对抗防御功能。

2. 我们以 Cifar-10 数据集与 ResNet 深度学习模型为例进行实验，验证了过滤对抗样本少量敏感像素点起到防御作用的可行性。首先利用 FGSM（快速梯度符号法）对抗攻击方法攻击数据集生成对抗样本。当前主流防御方法基本都是采用全局过滤扰动的方式，也就是不论扰动对深度学习模型输出的影响如何，都要去过滤处理，这在无形中增加了防御方法的计算代价。我们用演化算法找出扰动中的关键分量进行过滤处理，忽略其他位置的扰动，这将大大减少

防御方法的计算量。为了加快寻找敏感像素点的速度，我们选择差分进化算法作为寻找敏感点的演化算法，差分进化算法曾被证明是速度最快的进化算法，基于此将大大提升我们所提出的防御方法的执行效率。在此基础上，随机选择样本生成对抗样本验证防御方法的效果。实验表明，我们所提出的防御方法针对 FGSM 攻击方法的攻击具有良好的防御作用，具有可行性。

3. 将基于差分进化算法寻找敏感点与过滤敏感点相结合，设计对抗防御方法。因为我们提出的对抗防御方法是一种黑盒式防御方法，不需要关心深度学习模型的参数、结构、激活函数和损失函数等信息，理论上应该具有较好的普适性，可以应对针对各种深度学习模型的攻击。也就是说我们所提出的防御方法所具有的普适性是有两方面含义的，一是可以防御多种攻击方法，二是可以适用于多种深度学习模型。即该防御方法应该可以防御住多种攻击方法对多种深度学习模型的攻击。我们通过实验验证该防御方法对于不同的对抗攻击方法（FGSM、BIM 和 PGD）在不同深度学习预测模型（ResNet、AlexNet 和 LeNet）和不同数据集（Cifar-10 和 MNIST）上的防御效果，并详细分析了实验结果。实验结果表明，我们所设计的防御方法应对 FGSM 攻击时表现较好，面对 BIM 和 PGD 攻击时想要提高防御效果需要进一步增加敏感点的数量，防御方法在三种深度学习模型上的表现基本一致。

本文的组织结构如下：

第 1 章绪论，阐述了深度学习在表现优异的同时还伴随着天生容易被攻击的特性，这给深度学习相关应用的安全带来了威胁。有必要针对深度学习的安全性做出进一步研究。重点分析了国内外关于对抗防御的研究历史和现状，确立了本课题的研究方向。

第 2 章背景知识介绍，简述了本课题展开研究所需具备的相关理论知识。一是深度学习理论，介绍了本课题所需的三种深度学习模型，二是对抗攻击相关理论，详细介绍了本课题所用到的三种对抗攻击方法。

第 3 章基于对抗攻击的防御策略，详细分析了经典的对抗攻击方法的实现原理，并对攻击方法按照攻击范围进行分类，通过对比提出针对深度学习对抗攻击的防御策略。

第 4 章防御策略的设计与实现，针对第四章提出的防御策略设计实现算法。实现了基于差分进化算法寻找敏感点与过滤敏感点的算法，并设置实验验证了防御策略在不同数据集上的防御效果。

第2章 背景知识介绍

2.1 深度学习

深度学习是一种深层次的人工神经网络[29,30]。深度学习通过学习数据的特征，实现对数据的分类，其通常由输入层、隐藏层和输出层三部分组成，如图 2-1 所示。图中每一个圆圈代表一个神经元，每个神经元是一个感知器，神经元之间的连线表示模型参数，神经元内的激活函数负责激活输入值，激活后的数值又是下一层神经元的输入，直至到达输出层实现分类。隐藏层通常有多层，按功能分又可以分为卷积层和池化层。卷积层主要用来提取特征和参数共享，池化层主要用来减少参数数量和避免过拟合。

深度学习的优异的特征提取能力使其在计算机视觉等许多领域取得了巨大成功。深度学习解决的很多复杂问题已经达到甚至超越人类的水平。但研究表明，深度学习技术也存在严重的安全问题，比如很容易被对抗样本所攻击使其产生错误类别。

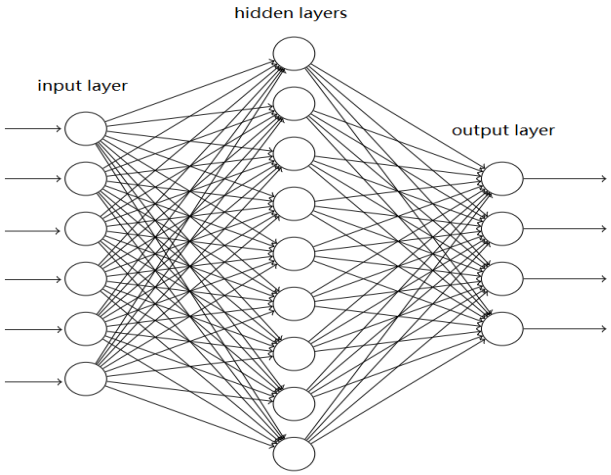

图 2-1 深度学习框架

本课题是针对计算机视觉领域的对抗攻击防御策略的研究，因此本小节将重点介绍卷积神经网络以及本课题实验中用到的三种经典深度学习模型。

2.1.1 卷积神经网络简介

卷积神经网络（Convolutional neural network, CNN）[31,32]是深度学习的一种，其灵感来自于人类对大脑认知原理的研究，人类对物体的认知是通过逐层分级来完成的。原始信号摄入大脑皮层，接着大脑皮层相关细胞发现方向与边缘，然后大脑由边缘与方向抽象出物体的形状，最后大脑进一步抽象出物体的类别。卷积神经网络便是模仿这种人类对物体的认知方式，构造多层神经网络，低层的网络用来提取图像特征，然后底层的特征构成上一层特征，上一层神经网络进一步提取特征，最终在顶层神经网络做出分类。

卷积神经网络最早是由 Yann Lecun 于 1994 年提出，用于 MNIST 数据集的分类，在当时取得了极其优秀的成绩，从此打开了研究者们对卷积深度学习研究的大门，各种卷积深度学习模型和各种优化方法层出不穷。在卷积深度学习出现之前就存在很多深度学习模型，但是这些模型往往是全连接网络，模型参数众多，训练模型的代价很高，在当时的硬件发展条件下，训练模型几乎是很难实现的一件事。卷积深度学习与之前的深度学习不同，它采用卷积核与池化层相结合的方式大大减少了模型参数，并且不但没有减少模型的准确率，甚至还提升了一个层次。

一个典型的卷积神经网络通常由卷积层、池化层和全连接层组成，其结构图如图 2-2 所示[33]。其中池化层紧着卷积层后面构成多个卷积组，逐层提出特征并传递特征，最终通过全连接层完成分类。我们用$\Theta = \{W^l, b^l\}$表示卷积深度学习模型的参数，其中W^l和b^l分别表示卷积核在l层上的权值和偏差，那么模型的前馈操作如下式所示：

$$A^l(X) = f_p\left(f_a(W^l * A^{l-1}(X) + b^l)\right) \qquad (2-1)$$

其中，符号$*$表示卷积操作，f_a表示激活函数，主要有 ReLu, pRelu 等。f_p表示池化操作，通常有最大池化与平均池化。池化操作可以用来减维，它不仅降低了计算复杂度，而且提高了模型的泛化能力。A^l表示卷积层与池化层执行后的特征图。当特征图被传入全连接层时，所有的特征图都需要被扁平化并转换为向量。最后，由全连接层做出预测。当预测不正确时，我们使用交叉熵损失函数来计算预测类别与实际标签之间的距离，然后通过反向传播算法更新模型的连接权重。反向传播算法有利于模型降低损失函数的值，加快模型的收敛。模型的优化如下式所示：

$$\Theta^* = \arg min J(\Theta; D) \qquad (2-2)$$

其中，$J(\Theta)$代表经验损失函数,$\Theta*$代表模型收敛后的参数。训练后的模型将会在新的数据集上表现良好，具有良好的泛化能力。

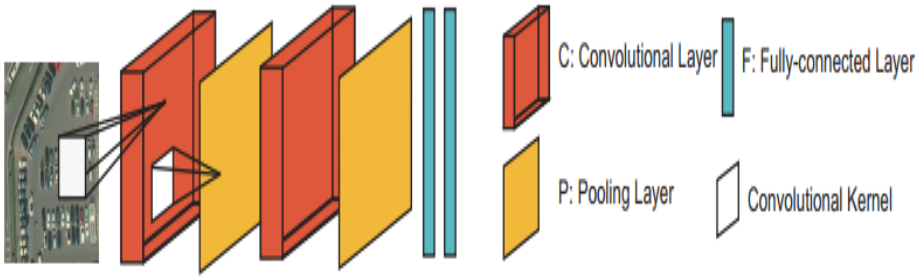

图 2-2 卷积深度学习结构

接下来将重点讲解卷积层、池化层和全连接层。

(1) 卷积层

卷积层是卷积神经网络中最核心的一部分。每一个卷积层都由一组卷积核组成。卷积的计算过程可以看成是二维的滑动窗口，每个卷积核从左到右、从上到下滑动，每一次滑动计算图像对应位置的值与卷积核对应数值的乘积之和作为特征图的对应位置新值，每一次滑动的步长是一个超参数，由人为指定。当滑动步长为 1 时，卷积过程如图 2-3 所示。每一个卷积核都会计算出一个对应的特征图。

卷积过程中有两个超参数-步长（strides）与零填充（padding）决定每次卷积结果。当步长为 1 时，则每次滑动一个像素的距离；当步长为 2 时，则每次移动 2 个像素的距离，以此类推。通常以步长为 1 最为常见，移动步长不同则卷积后的特征图大小也会不同。零填充主要用来控制卷积核的输出结果。零填充是指在原图像边界填充指定个数的零值，这可以用来控制卷积结果的大小。比如如果没有零填充，则每次卷积后特征图的大小小于卷积之前的大小，如果加上了零填充则有可能使得卷积后的特征图大小大于卷积前大小。由于是零填充故而不会影响卷积的计算。在边界处添加零有利于模型提取边界的特征，这也是零填充的一大原因。

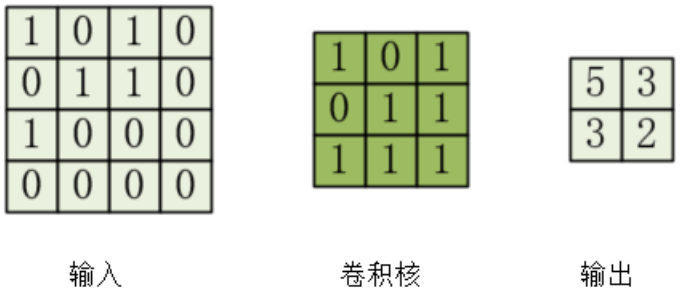

图 2-3 步长为 1 时的卷积过程

(2) 池化层

在卷积深度学习中，通常池化层与卷积层同时出现，池化层用来池化卷积层得到的特征图。池化层的主要作用是降维，即减少参数个数，这将大大减少训练网络的计算代价，同时引入池化后增加了随机操作可以起到避免过拟合的作用。当前最常见的池化有最大池化和平均池化。最大池化是指每次滑动窗口时都是选择池化 filter 覆盖范围内的最大值作为结果值；平均池化是指每次滑动窗口时会计算 filter 覆盖范围内所有像素值的平均值作为结果值。可以发现池化操作其实就是下采样，每一次池化都会大大减少参数数量。其中最大池化最为常见。这是由于最大池化是选择 filter 覆盖范围内的最大值作为选择特征，也就是选择目标范围内最显著的特征作为结果，这与我们想要取得最显著特征相符。另外由于最大池化只涉及到比较操作，而平均池化会涉及到求和与除法操作，所以最大池化会比平均池化计算代价少，这也是大家选择最大池化的一个原因。最大池化与平均池化的具体计算如图 2-4 所示。

池化层也有两个超参数-步长与零填充。池化层步长与卷积层步长作用相同，也是用来控制每次池化时移动的距离。零填充就是在池化层得到的特征图边界处填充零，由于零值对最大池化影响不大，故而池化操作时通常不会使用零填充。

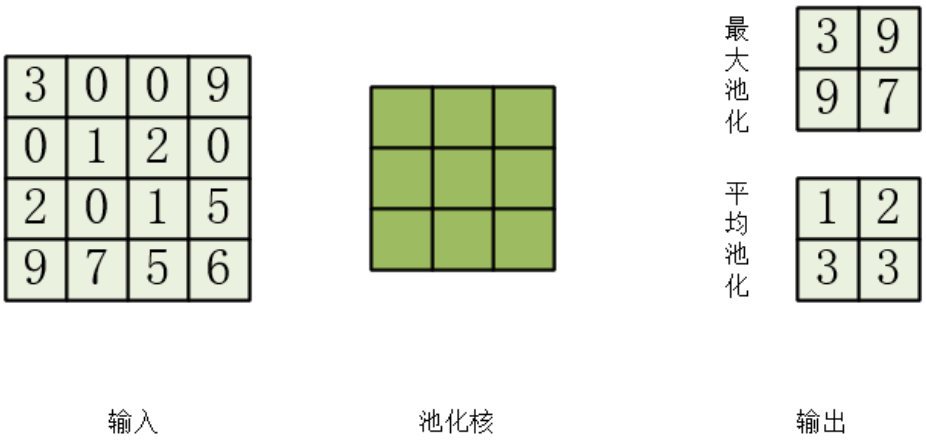

图 2-4 步长为 1 时的最大池化与平均池化

(3) 全连接层

全连接层通常在卷积神经网络模型的最后一层或最后多层，全连接层每个神经元都有一个激活函数，用来处理全连接后的数值，最后一层全连接层的输出传入输出层进行分类，输出层通常为 softmax 逻辑回归。因为其全连接的特性，这会使得连接层参数众多，需要较大的计算代价。随着硬件技术的发展，可以使用 GPU 来加速全连接层的计算。

2.1.2 深度学习经典模型

近年来，随着算力的提升出现了很多经典深度学习模型，大有越来越深之趋势。由于本课题将用到 LeNet[31]、AlexNet[34]和 ResNet[35]三种深度学习模型，故而本小节将重点介绍这三种深度学习模型。

(1) LeNet

LeNet 是由 Yann Lecun 于 1994 年提出，用于 MNIST 数据集分类，在当时取得了极其优秀的成绩。LeNet 网络结构包含输入层在内共有 8 层，分别为输入层、C1 卷积层、S2 池化层、C3 卷积层、S4 池化层、C5 卷积层，F6 全连接层和输出层。具体结构如图 2-5 所示：

第一层：该层为输入层，输入层用来接收图像的输入，由于 LeNet 是用来处理手写数字识别，数据集大小为 32*32，故而输入层大小也为 32*32。

第二层：该层为卷积层，卷积核大小为(5,5)，共 6 个，所以根据卷积计算方式可以得到 6 组 28*28=784 的卷积特征图。该层共有 6 个通道，每个通道有

784 个权值，所以该层共有 6*784=4704 个神经元，由于卷积核共享权重，因此需要训练的参数有 6*25+6=156 个，其中加 6 是加的偏差的个数，连接数为 784*156=122304 个。

第三层：该层为池化层，并且为平均池化。池化窗口为(2,2)，因此神经元个数为 6*14*14=1176 个，连接数为 6*196*（4+1）=5880 个。

第四层：该层为卷积层，卷积核大小为(5,5)，共 60 个，所以根据卷积计算方式可以得到 16 组 10*10=100 的卷积特征图。该层共有 16 个通道，每个通道有 100 个神经元，所以该层共有 16*100=1600 个神经元，需要训练的参数有 16*25+16=416 个，连接数为 100*1516=151600 个。

第五层：该层为池化层，并且为平均池化。池化窗口为(2,2)，因此神经元个数为 16*5*5=400 个，连接数为 16*25*（4+1）=2000 个。

第六层：该层为卷积层，卷积核大小为(5,5)，共 1920 个，神经元数量为 120 个，连接数为 48120 个。

第七层：该层为全连接层，神经元个数为 84，连接数个数为 10164 个。

第八层：该层为输出层，利用径向基函数作为输出单元，一共有 10 个径向基函数。

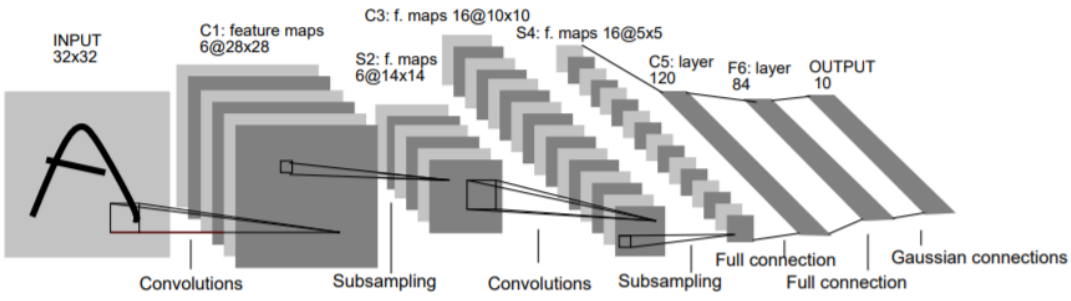

图 2-5 LeNet 网络模型结构

(2) AlexNet

AlexNet 是由 Alex Krizhevsky 等人于 2012 年提出，用于图像识别，其在 ImageNet 2012 图像识别挑战赛中获得了冠军，从而一举打破了计算机视觉研究的前状，吸引大批研究者进军深度学习。相比于 LeNet，AlexNet 网络的参数更多，规模更大。LeNet 网络使用的激活函数为 sigmoid，AlexNet 中的激活函数改为了 ReLu。AlexNet 模型更换激活函数出于两种考虑，其一 ReLu 的计算更加简单，因为其并没有 sigmoid 激活函数中的求幂运算，其二 ReLu 作为

激活函数比 sigmoid 更不易梯度消息，因为其在正区间的梯度恒为 1。ReLu 函数和其导数示意图如图 2-6 所示[42]。

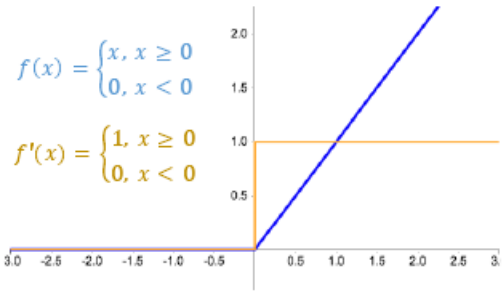

图 2-6 ReLu 函数和其导数

AlexNet 模型网络结构共有 11 层组成，分别为卷积层、池化层、卷积层、池化层、卷积层、卷积层、卷积层、池化层、全连接层、全连接层、全连接层。具体网络结构如图 2-7 所示。

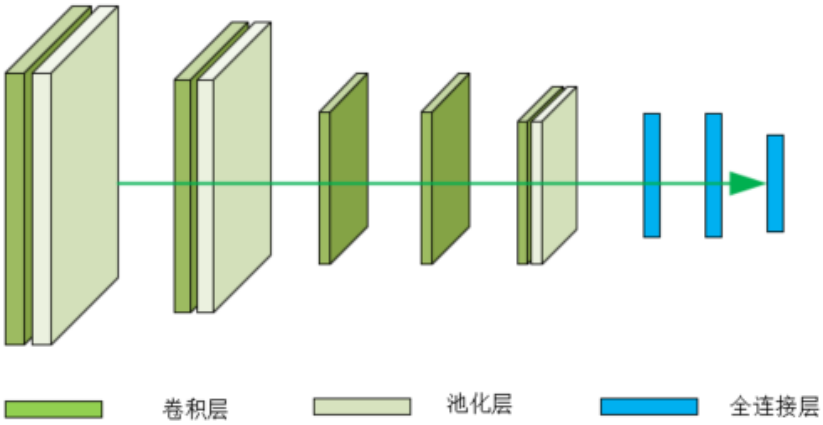

图 2-7 AlexNet 网络模型结构

我们把 AlexNet 网络结构分成卷积层、池化层和全连接层三组进行详细结构介绍。可见整个深度学习模型包含卷积层 5 个，池化层 3 个和全连接层 3 个。

卷积层：第一个卷积层共有 96 个卷积核，每个卷积核的大小为(11,11)，每次移动步长为 4；第二个卷积层共有 256 个卷积核，每个卷积核的大小为(5,5)，每次移动步长为 1；第三个与第四个卷积层的卷积核个数、步长、卷积核大小均相同，都是有 384 个卷积核，每个卷积核大小为(3,3)，每次移动步长

为 1；第五个卷积层共有 256 个卷积核，每个卷积核的大小为(3,3)，移动步长为 1。

池化层：AlexNet 的三个池化层都是采用的最大池化，也就是选择池化 filter 覆盖范围内的最大值作为卷积层的特征图。

全连接层：AlexNet 模型最后三层均为全连接层，第一个全连接层共有 4096 个神经元，紧着这第二个全连接层也是 4096 个神经元，最后一层全连接层神经元个数为 1000，这是因为 AlexNet 是用来处理 ImageNet 图形识别，该数据集共有 1000 种类别。

(3) ResNet

自 AlexNet 取得 ImageNet 2012 图像识别挑战赛的冠军以来，绝大多数研究者们认为深度学习模型的深度越深，参数越多，模型越复杂，其学习到特征就越多，表现也就越优异。然而随着人们不断加深神经网路，却发现添加过多的层之后模型表现反而不升反降。针对这一问题，何恺明等人提出了 ResNet。该模型成功解决了该问题，并在 ImageNet 2015 图像识别挑战赛中一举夺冠[36]。他们的解决方案是用残差块代替传统的直连卷积，如图 2-8 左图所示。残差块通过一个跨层连接实现的，这个模块的输入和输出也是是通过跨层链接方式添加的，在不增加多余参数的情况下，可以快速提高模型的训练速度。在残差块中数据可以跨层向前传播，避免了逐层传播加快了训练速度。ResNet 模型通过改变学习目标，即变为学习残差，简化了学习的难度，同时解决了传统卷积层在进行逐层传递时存在的数据丢失和消耗等问题，通过直接将特征从输入直接连接到输出，一定程度上保护了特征的完整性。

ResNet 包含多个残差模块，这些模块有助于解决梯度消失和梯度爆炸的问题，进而支持更加深层次的卷积深度学习，其网络模型结构如图 2-8 右图所示。ResNet 模型将深度发挥到了极致，第一次达到了 152 层的深度。ResNet50 是 ResNet 的 50 层版本。为了训练更深层次的结构，ResNet 引入了一个残差模块，如图 2-8 左图所示，它通过跨层连接实现了层与层之间的信息传递。在不需要增加参数的情况下，直接将浅层特征放入上层，大大提高了模型的训练速度。ResNet 在 ImageNet 大赛上取得了里程碑式的成绩，它是第一个在 ImageNet 分类任务上超越人类识别的模型。此外，ResNet 作为检测模型的主要骨干，在 VOC Pascal[37]、COCO[38]数据集等检测任务中取得了良好的效果

[38]。由于 ResNet 的优异表现，使得该模型在计算机视觉等领域广为流行。针对 ResNet 的研究也越来越多，出现了很多变种，性能不断提升，模型深度也在不断加大。

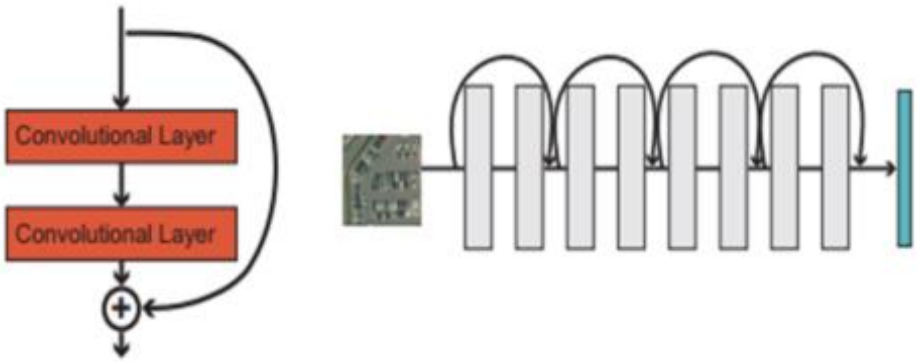

图 2-8 左图为残差块，右图为 ResNet 网络模型结构概图

2.2 对抗攻击

2.2.1 对抗攻击相关概念

对抗攻击是指通过一些方法对原始样本人为的添加一些微小噪音，保证人的肉眼几乎察觉不到样本的变化，也就是人眼仍能做出正确决策，然而深度学习模型对该样本做出错误预测。

对抗攻击相关术语：

对抗样本：对抗攻击生成的可以使深度学习模型产生错误预测的样本叫做对抗样本。

对抗扰动：添加在原始输入样本之上的噪音。

黑盒攻击：对抗攻击方法产生对抗样本时不需要利用被攻击网络模型的结构、参数、激活函数和损失函数等信息，只需要知道模型的输入对应的输出信息便可以完成攻击。

白盒攻击：对抗攻击方法要想完成攻击效果则需要知道被攻击网络模型的结构和参数等信息。

目标攻击：对抗攻击方法可以指定针对输入样本攻击后的类别。这种攻击方法因要指定攻击后的具体类别难度更大。

非目标攻击：只要攻击方法产生的对抗样本的预测结果不是正确类别即可，具体是哪一类别并不关心。

模型的鲁棒性：深度学习模型防御对抗攻击方法的攻击的能力。

2.2.2 距离度量

在设计对抗攻击方法时，优化函数约束条件通常会涉及到距离度量，即度量原始样本与对抗样本的距离，即：

$$L_p = \| X - X^{adv} \|_p = \left(\sum_{i=1}^{n} \left| X_i - X_i^{adv} \right|^p \right)^{\frac{1}{p}} \quad (2-3)$$

其中，X表示原始样本，X^{adv}表示对抗样本.

在对抗攻击领域常用的距离度量范数有无穷范数L_∞与零范数L_0。

L_∞ 表示所有向量中元素的最大值，也就是对抗样本与原始样本对应位置相减后的绝对值的最大值，用来限制对抗攻击方法添加的扰动的大小：

$$\| X - X^{adv} \|_\infty = max\left(\left| X_1 - X_1^{adv} \right|, \cdots, \left| X_n - X_n^{adv} \right| \right) \quad (2-4)$$

L_0 表示所有向量中改变元素的数量，也就是对抗样本与原始样本对应位置相减后绝对值大于零的元素个数，用来限制对抗攻击所改变的像素点的个数。

2.2.3 对抗攻击方法介绍

近年来，由于深度学习的火爆和深度学习模型易受攻击等特点，大量研究者涌入研究深度学习对抗攻击的浪潮，不断有新的对抗攻击方法出现。由于本课题将用到 FGSM、BIM 和 PGD 三种经典对抗攻击方法，故而本小节将重点介绍这三种对抗攻击方法。

(1) FGSM

Goodfellow 等人发现之前人们对对抗样本存在的解释归因于网络的非线性和过拟合，而他们通实验得出恰好相反的结论—对抗样本的存在是因为网络的线性特征[14]。并且基于此结论他们提出了一种既简单又可快速生成对抗样本的方法进而可以对抗训练。他们称这种生成对抗样本的方法叫快速梯度符号方法（即 FGSM）。具体来讲：设 x 表示输入数据，y 表示输入数据对应的类别标签，θ 表示模型参数，那么把损失函数相对于 x 的梯度值的符号函数值加上

x，然后传入模型便可产生错误类别生成对抗样本，即生成对抗样本x^{adv}的公式为：

$$x^{adv} = x + \varepsilon \text{sign}\left(\nabla_x J(\theta, x, y)\right) \qquad (2-5)$$

其中$J(\theta, x, y)$表示损失函数，ε表示较小的正常数（如 0.1 等），用来控制扰动幅度。FGSM 攻击效果如图 2-9 所示，对原始样本为熊猫的图片加上 FGSM 方法生成的扰动后，在传入深度学习做预测时类别变为长臂猿。

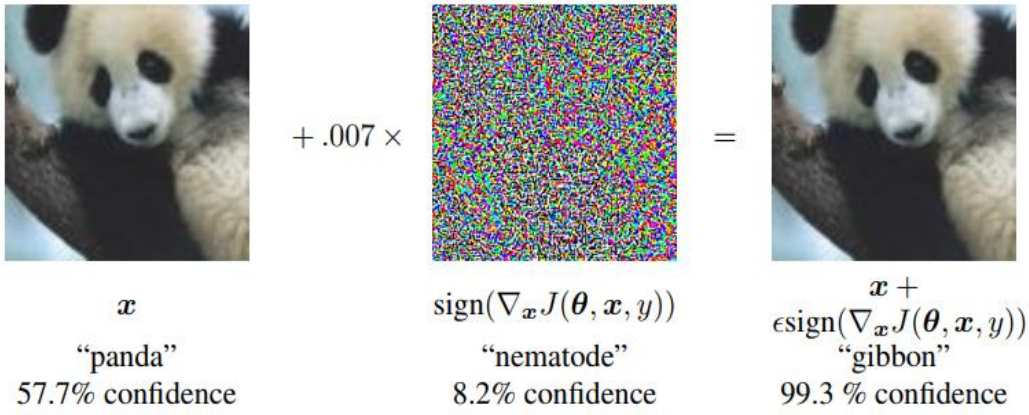

图 2-9 FGSM 攻击效果图

(2) BIM

BIM 攻击方法由 Kurakin 等人[39]于 2016 年提出，BIM 方法是对 FGSM 的改进，方法是通过迭代的方式每次一小步修改像素值，并且修剪生成的像素值使之确保在原始图片 X 的给定区间内。公式为：

$$X_0^{adv} = X,$$

$$X_{N+1}^{adv} = Clip_{X,\varepsilon}\left\{X_N^{adv} + \alpha\, sign\left(\nabla_X J\left(X_N^{adv}, y_{true}\right)\right)\right\} \qquad (2-6)$$

其中X_N^{adv}表示第 N 次迭代时的扰动图像，$Clip$表示剪辑图像的像素值，α表示较小的正常数（如 0.1 等），用来控制扰动幅度。这种对抗攻击方法，相比 FGSM 能够生成更加不易察觉的对抗扰动。

(3) PGD

PGD [28]攻击是一种迭代攻击，可以看作是 FGSM 的迭代版本。我们知道 BIM 也是 FGSM 的迭代版本，这两者的区别是，PGD 的迭代次数更多而且加了随机化操作，使得 PGD 的攻击效果比 FGSM 要好，相比 FGSM 能够生成更加不易察觉的对抗扰动。其攻击原理与 BIM 基本一致，即通过迭代的方式每

次一小步修改像素值，并且修剪生成的像素值使之确保在原始图片 X 的给定区间内。

2.3 本章小结

本章主要介绍了本课题需要用到的一些理论知识。重点介绍了在后面章节需要涉及的 ResNet、AlexNet 和 LeNet 三种深度学习模型与 FGSM、BIM 和 PGD 三种对抗攻击方法。同时详细讲解了卷积神经网络中卷积层和池化层的计算，简述了对抗攻击涉及到的度量函数和相关术语，为后面章节服务。

第3章 基于对抗攻击的防御策略

目前为止，研究者们已经提出了多种对抗攻击方法，攻击原理各不相同。新的对抗攻击方法的提出会促进新的防御方法的出现；新的防御方法的出现又会促进新的防御方法的提出。攻击与防御相互对立又相互促进。正所谓知己知彼，方能百战不殆。我们从攻击原理入手分析以期能够从攻击方法中得出防御策略。依据攻击范围的不同，我们把攻击方法分为两大类，一类为全局攻击，另一类为局部攻击。

3.1 全局攻击

全局攻击是指攻击方法对整张输入样本添加扰动，区别在于有些像素点的扰动范围较大，有些像素点扰动范围较小，如图 3-1 所示。

图 3-1 全局扰动

FGSM 作为经典且常用的对抗样本生成方法，其计算扰动的方式为通过计算损失函数相对输入样本的梯度的符号信息所得，具体公式为：

$$\rho = -\varepsilon \cdot \mathrm{sign}\left(\nabla_x J(\theta, x, y)\right) \tag{3-1}$$

其中，θ表示深度学习模型的参数，$J(\theta, x, y)$为损失函数，sign(·)为符号函数，作用是用来限制梯度的大小，避免加上的扰动过大使得肉眼易于察觉，ε是用来控制扰动变化的超参数。假设输入样本x所对应的真实类别为 $y=l$，若存在

$y=m$ 且 $m\neq l$,使得$J(\theta,x^*,m)<J(\theta,x,l)$,那么深度学习模型会将生成样本$x^*$分类为$m$，从而成为对抗样本。

$FG-\ell_2$对抗样本生成方法是通过ℓ_2对$\nabla_x J(\theta,x,y)$归一化后的数值作为扰动，即：

$$\rho=-\varepsilon\frac{\nabla_x J(\theta,x,t)}{\parallel\nabla_x J(\theta,x,t)\parallel_2} \tag{3-2}$$

其中，$\parallel\nabla_x J(\theta,x,t)\parallel_2$为$L_2$范数。可以看出$FG-\ell_2$方法也是利用了损失函数相对于输入样本的梯度信息。

BIM 是在 FGSM 基础之上进行了改进。通过迭代的方式来控制扰动的大小。其迭代方式如下所示：

$$x_{n+1}^*=\mathrm{Clip}(x_n^*+\rho_n),x_0=x \tag{3-3}$$

其中，x_n^*和ρ_n分别表示第 n 次迭代的输入样本和生成的扰动，且ρ_n与 FGSM 的生成扰动公式一样。

PGD 与 BIM 类似，也是一种迭代式对抗样本生成方法，其生成对抗样本的公式如下：

$$x_{n+1}=\prod_{x+s}(x_n+\alpha\cdot\mathrm{sign}(\nabla_x J(x_n,y))) \tag{3-4}$$

PGD 和 BIM 的区别在于 PGD 增加迭代次数，并且还加了一层随机化操作，这使得 PGD 比 BIM 攻击能力更强。

可以发现，上述几种对抗攻击方法都涉及到了损失函数相对于输入样本的梯度信息，梯度的计算是全局的，其计算会影响到每个像素点的值。上述对抗攻击方法可以概括为如下数学公式：

$$\begin{aligned}&\text{Find } x'\\&\text{s.t. } f(x')\neq f(x),\left|\left|x'-x\right|\right|_\infty\leq\epsilon\end{aligned} \tag{3-5}$$

其中，f表示深度学习模型，x表示原始输入样本，x'是添加了扰动的对抗样本，ϵ作用是用来限制扰动大小。

3.2 局部攻击

局部攻击是指攻击方法对整张输入样本中某一个或多个像素点添加扰动，像素点的个数或百分比往往是个超参数，可以人为指定。如图 3-2 所示。

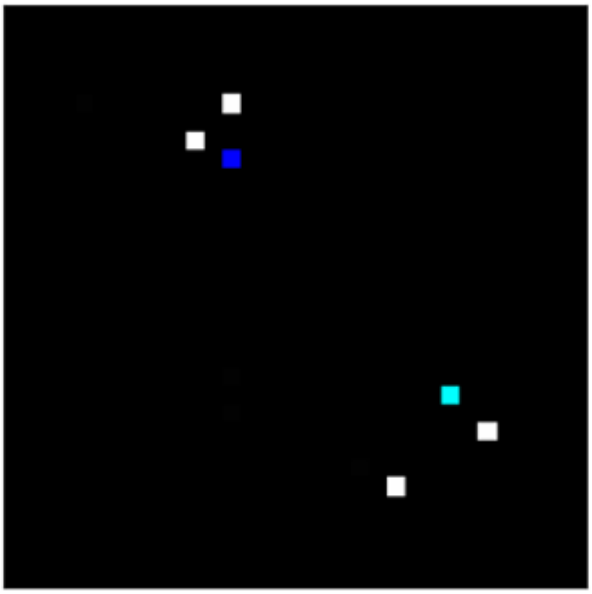

图 3-2 局部扰动

该类攻击方法往往利用演化算法求出对深度学习模型输出影响较大的关键分量，然后在关键分量上添加微小的扰动生成对抗样本。这种攻击方法的优点是，不需要了解深度学习模型的参数等信息，对一些不可微分的网络模型也可以成功攻击。基于演化算法的对抗攻击方法主要有 One-Pixel Attack 和 LocSearchAdv。

One-Pixel Attack 顾名思义该方法利用差分进化算法可以仅仅在一个关键分量上添加扰动从而生成对抗样本[40]。该方法通过建立并求解如下所示的目标函数来求得相应的关键分量坐标和扰动大小：

$$\arg\min_{j,\varepsilon} f^{(l)}\big(M(x,j,\varepsilon)\big) \tag{3-6}$$

其中，$f(x)$为深度学习模型的输出，$M(x,j,\varepsilon)$为：

$$M(x,j,\varepsilon)=\begin{cases} x^{(i)}+\varepsilon, i=j \\ x^{(i)}, i\neq j \end{cases}$$

表示对输入样本x的第j个分量添加扰动ε所得的向量。由公式可以发现该对抗样本生成方法不仅仅可以做到单点像素攻击，通过改变$\boldsymbol{j}$值是可以变为多点像素攻击的。One-Pixel Attack 攻击方法可以在数据集为 Cifar-10 测试集和 ImageNet 验证集的情况下，仅仅改变一个像素点便可以分别实现 68.36%和 41.22%的攻击成功率。

受 One-Pixel Attack 的启发，LocSearchAdv 对抗攻击方法利用贪心搜索算法，在输入样本x中搜索出多个使得$f^{(l)}(x)$下降最明显的几个关键分量，并在每一个关键分量上添加对应的扰动得到对抗样本，以实现对抗攻击。

可以发现，上述两种对抗攻击方法都是非全局范围内添加扰动，也就是说仅仅在整张图片中对某些像素点添加扰动。上述对抗攻击方法可以概括为如下数学公式：

$$\begin{aligned} &\text{Find } x' \\ &\text{s.t. } f(x') \neq f(x), \left||x' - x|\right|_0 \leq d \end{aligned} \tag{3-7}$$

其中，d表示是欲求的关键分量的个数，也就是想要更改的像素点的个数。可以发现扰动的变化是受L_0范数约束的，这是与全局攻击的显著区别。

3.3 防御策略

从对抗攻击方法原理入手，对比全局攻击与局部攻击。

虽然对抗攻击方法多种多样，但从攻击范围考虑可以分为两大类。这两大类的攻击成功率都相当高。只是全局攻击通过求梯度的方式把所有像素点都考虑在内，而局部攻击仅仅找一些敏感的点去添加扰动。

$$\begin{aligned} &\text{Find } x' \\ &\text{s.t. } \begin{cases} f(x') \neq f(x), \left||x' - x|\right|_\infty \leq \epsilon & if\ global\ attack \\ f(x') \neq f(x), \left||x' - x|\right|_0 \leq d & otherwise \end{cases} \end{aligned} \tag{3-8}$$

将两类攻击方法抽象出的公式合并在一起，可以发现，在全局攻击时如果只有某些分量的数值变化比较大，其他点变化比较小以至于可以忽略，那么全局攻击的约束条件近似为局部攻击的约束条件。这就告诉我们，两类对抗样本生成方法本质上都在找一些关键分量，然后在这些关键分量之上加一些扰动，并且在关键分量之上所加的扰动通常会明显一些，目的就是达到欺骗深度学习模型的目的。

受此启发，我们可以通过演化算法来找到关键分量，然后限制攻击方法对该关键分量的修改，便可以实现防御效果。如果我们的深度学习模型是计算机视觉领域的决策器，那么关键分量可以理解为敏感像素点，也就是给这部分像素点添加扰动后会对决策器的输出产生较大的影响，以至于最终的分类类别发生变化，实现攻击效果。因此，我们可以制定针对对抗样本的防御策略，主要有以下两步组成：

Step1：利用演化算法找出敏感点

利用演化算法寻找敏感点可以实现黑盒防御，也就是不涉及对损失函数求梯度等信息，对深度学习模型的认知程度要求较低，不需要关心模型的参数、激活函数和损失函数等信息，只需要知道模型的输入和输出便可以运用演化算法来找出敏感点。这在一定程度上扩展了该防御方法的适用范围，因为往往实际防御攻击时我们是不知道模型的具体参数等信息的。

我们选择差分进化算法作为寻找敏感点的演化算法。差分进化算法是 1995 年在遗传算法的基础上提出的，本质是一种自适应多目标优化算法，用于求解多维空间的最优解。差分进化算法具有结构简单、收敛速度快等特点,曾在 1996 年被证明是速度最快的进化算法[43,44,45]。因此我们选择差分进化算法作为寻找敏感点的演化算法。详细步骤会在第四章进行介绍。

Step2：处理敏感点

在步骤(1)找到敏感像素点后，我们需要对敏感点进行过滤操作，目的是使得深度学习模型对对抗样本得判定类别更改为该样本的真实类别。由于对抗样本中的扰动既可能是全局扰动也可能是局部扰动，然而我们只需要过滤对应的敏感像素点处的扰动，不需要去处理大量的非敏感像素点处的扰动，这在一定程度上减少了防御方法的计算代价。

过滤敏感点的策略是把敏感点周围的相邻非敏感点的像素值求和取平均值作为敏感点的新值。此策略的理论依据是：敏感点之所以会成为敏感点是因为该点与周围像素点的像素值具有较大不同，如果我们用非敏感点的均值作为该点的新值，那么敏感点的像素值也会变成非敏感点的像素值，成为非敏感点。如果我们把步骤(1)找到的所有敏感点都执行该操作，那么就有可能把所有敏感点都变为非敏感点，把对抗样本中的关键扰动过滤掉，起到防御作用。详细步骤会在第四章进行介绍。

3.4 本章小结

本章，我们重点分析了几种经典的对抗攻击方法，把这些攻击方法按照攻击范围分类为全局攻击与局部攻击，对每一类抽象出共同的优化公式，然后通过对比优化公式，得出对抗样本中存在关键分量的结论，受此启发提出一种新的防御策略：第一步通过差分进化算法寻找对抗样本中的关键像素点，第二步过滤关键像素点，方法是让其新值等于周围相邻非敏感点的均值。

第4章 防御策略的设计与实现

我们所提出的防御策略如图4-1所示，即输入样本在传入预测模型之前会进行敏感点过滤，然后把过滤完的输入数据传入预测模型做出预测。这种类型的防御方法可以尽最大可能的减少对预测模型的结构、参数等的更改，这对迁移学习的应用有积极作用，研究人员只需要把其他研究人员预训练好的模型拿来使用，可以不必或很少出于鲁棒性考虑而调整模型结构和参数，把鲁棒性的提升放在预测模型之前完成。

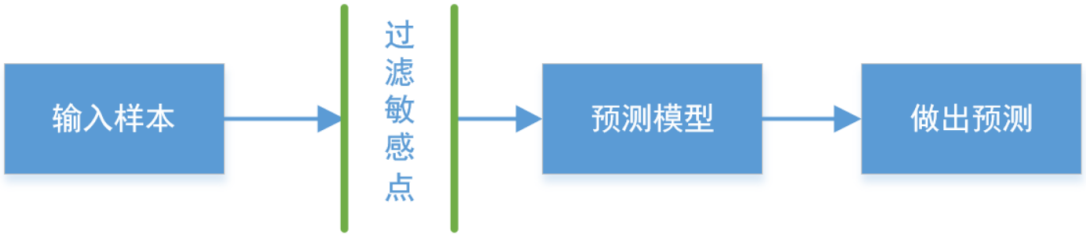

图 4-1 防御策略

4.1 基于差分进化算法寻找敏感点

4.1.1 引言

差分进化算法是进化算法的一种，其主要步骤与遗传算法基本一致，也包括变异，交叉和选择三种操作。

差分进化算法通过差分方法实现个体变异，常见的差分方法是从种群中随机选择两个不相同的个体，将其向量做差后乘以一个缩放因子然后与待变异的个体执行向量加法，即

$$v_i(g+1) = x_{r1}(g) + F \cdot \left(x_{r2}(g) - x_{r3}(g)\right) \quad i \neq r_1 \neq r_2 \neq r_3 \qquad (4-1)$$

其中，$v_i(g+1)$为变异的中间体，$r1, r2$和$r3$是三个随机数，$x_i(g)$表示第g代种群中第i个个体，F为缩放因子。本实验采取此差分方法，在进化过程中，为了保证解得合理性，需要保证“染色体”中各“基因”满足实际约束，一旦不能满足约束条件将会随机重新生成“基因”，使其合理。

交叉操作是指第g代种群与其变异的中间个体之间进行随机交叉，因交叉

操作具有随机性，故需引入概率。交叉操作的具体方法如下：

$$u_{j,i}(g+1)=\begin{cases} v_{j,i}(g+1), & \text{if rand}(0,1)\leqslant CR \text{ or } j=j_{\text{rand}} \\ x_{j,i}(g), & \text{otherwise} \end{cases} \quad (4-2)$$

其中，CR是一个概率值，j_{rand}为随机生成的正整数，$v_{j,i}(g+1)$为中间体$v_i(g+1)$的第j个位置的元素值，其值取决于CR。

差分进化算法使用贪心思想来选择进入下一代种群的个体，具体方法如下：

$$x_i(g+1)=\begin{cases} u_i(g+1), & \text{if } f\left(u_i(g+1)\right)\leqslant f\left(x_i(g)\right) \\ x_i(g), & \text{otherwise} \end{cases} \quad (4-3)$$

其中，$f(.)$即为需要优化的问题。

4.1.2 问题描述

敏感点是指输入样本中那些像素点的值轻微变动就会使得深度学习预测产生较大波动的点。比如一张 32×32 大小的 RGB 彩色图片，其类别为小狗，改变(11,12)处像素点的值，发现把改变后的图片传入深度学习f产生的预测类别为小猫，那么(11,12)处像素点便为敏感点。

一张图片的类别是由所有像素点共同确定的，把这些像素点可以看成高维向量，每一维的变动都会使得图片在高维空间中的坐标位置发生变动。又因为决策边界是已经训练好的，由权重和偏差等共同确定，固定不变。如果该图片所在的高维坐标距离决策边界很近，那么可能坐标轻微变动便使得该图片的新位置越过决策边界，成为其他类别。

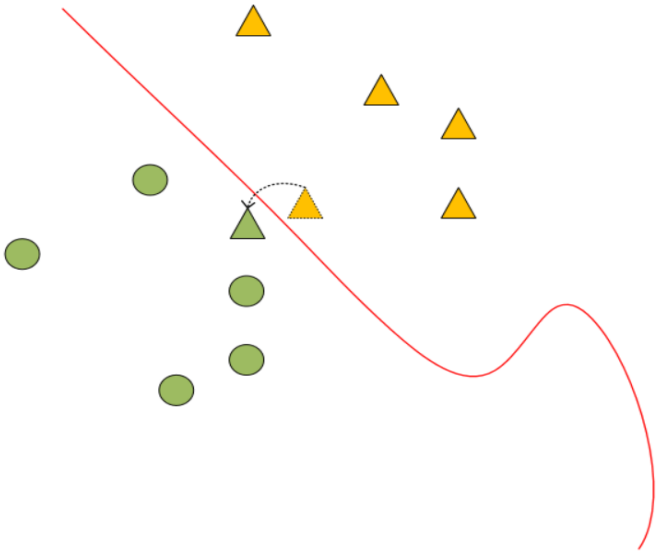

图 4-2 敏感点示意图

如图 4-2 所示，红色线表示决策边界，绿色和黄色分别表示不同的类别。在决策边界附近的点很容易通过平移变换越过红色决策边界，变成另一个类别。

我们可以把图片看成一个高维向量，其中高维向量的每一维的值由原图片展开所得。即：彩色图片由 RGB 三种颜色组成，表示为$p_i = (r, g, b)$,那么高维向量可以由图片从上到下、从左到右展开，即$P = (p_1, p_2, p_3, \dots, p_m)$，其中 m 为图片的像素点的个数，即长乘宽。把$p_i$带入$P$，可得高维向量

$$P = (r_1, g_1, b_1, r_2, g_2, b_2, r_3, g_3, b_3, \dots, r_m, g_m, b_m) \quad (4-4)$$

那么，图片的平移体现在P向量上的表现，便是r_i, g_j或b_k值的变化。显然对任意一个输入图片而言，当沿着某一维度或者多个维度一起移动一小段距离后存在越过决策边界的可能性，那么这些维度所在的像素点便为敏感点。

需要注意的是，敏感点并非只能由一个像素点组成，敏感点是一个可以使深度学习f预测值产生较大波动的像素点的集合，可以有多个像素点。敏感点的寻找可以形式化为带约束的优化问题，即在改变尽可能少的像素的情况下，使得深度学习f的预测置信度尽可能低。我们把输入图像展开成为一个向量，用X表示，向量中的每个元素表示一个像素点，用符号x_i表示,则$X = (x_1, \dots, x_n)$。

我们用$s(X) = (s_1, \cdots, s_n)$表示相对于输入图像原始像素值的变化。那么相对于深度学习f的敏感点的寻找可表示为：求解最优化$e(X)^*$，即：

$$\underset{s(x)^*}{minimize} \quad f\big(X + s(X)\big) \quad (4-5)$$

$$Subject\ to \quad \|s(X)\|_0 \le d$$

其中，d表示敏感点的个数，如当$d = 1$表示在整张输入图像中在改变一个点的像素的情况下，将会导致深度学习f的预测值发生较大变化，那么这个点便是敏感点。

针对此优化问题的求解算法有很多，相对于梯度下降和贪婪搜索算法，差分进化算法是一种相对较少受局部极小值影响的启发式算法，另外差分进化算法不使用梯度信息进行优化，因此不要求目标函数是可微的或已知的，这对于本问题的求解是至关重要的，因为有些深度学习是不可微的或者在很多情况下计算其梯度是不现实的，因此，与基于梯度的方法相比，它可以用于更广泛的优化问题。此外，本文所考虑的问题有严格的约束(敏感点的个数相对于整个

输入图像像素点数目极少)，这使得问题的求解变得相对困难，其他方法很难胜任，而差分进化算法求解此问题只需知道深度学习的标签概率值就足够了，与深度学习的类别、结构、参数等无关。因此，我们选择运用差分进化算法求解此优化问题。

4.1.3 问题求解

为了便于定位像素位置，我们把每个像素点表示为：

$$x_i = (x, y, r, g, b) \tag{4-6}$$

其中，x,y表示像素点的坐标，x为像素点相对于原点左上角在竖直方向上的距离，y为像素点相对于左上角在水平方向上的距离。r, g, b分别为像素点的三原色红、绿、蓝，其取值为 0 到 255。

因差分进化算法要求输入值为向量，为方便运用此算法，我们把输入图像$X = (x_1, \dots, x_n)$转化为：

$$X = (x_1, y_1, r_1, g_1, b_1, x_2, y_2, r_2, g_2, b_2, \dots) \tag{4-7}$$

同样，我们也把$s(X)$转换成上述形式。

首先，我们随机生成 m 个$s(X)$作为初始群体，然后运用下述公式产生 m 个变异体：

$$v_i(g+1) = x_{r1}(g) + F \cdot \left(x_{r2}(g) - x_{r3}(g)\right) \quad r_1 \neq r_2 \neq r_3 \tag{4-8}$$

也就是，每个变异体是通过随机选择 3 个上一代个体，然后相互结合生成下一代个体。下一代个体在以一定的概率交叉操作生成新的个体。如果新个体V_i相比上一代个体X_i能够使得深度学习f的预测概率值更小，则淘汰上一代个体。重复上述过程多次，直至找到对应数量的敏感点可以使得深度学习f的预测结果错误。算法具体流程如算法 4-1 所示。

Algorithm 1 Find Sensitive Points

Input: Initialize the following parameters.
(1) the initial number of population: $popSize = 400$
(2) the scale parameter: $F = 0.5$
(3) the probability of crossover: $CR = 0.8$
(4) the maximum number of iteration: $maxIter = 100$
(5) the number of sensitive point: $d = 10$
(6) the input image: $\mathbf{X}$

Output: the coordinates of d sensitive points.

Randomly generate m individuals x_i to form the initial population, initialize current iterations: $g = 1$
for $g = 1, ..., maxIter$ **do**
  for $i = 1, ..., popSize$ **do**
    Pick 3 random individuals from the previous generation and recombine them to make a new candidate solution:
      $v_i(g+1) \leftarrow x_{r1}(g) + F \cdot (x_{r2}(g) - x_{r3}(g))$
    Cross transformation with probability CR:
    if $rand(0,1) \leq CR$ **or** $j = j_{rand}$ **then**
      $u_{j,i}(g+1) \leftarrow v_{j,i}(g+1)$
    else
      $u_{j,i}(g+1) \leftarrow x_{j,i}(g)$
    end if
    Each candidate solution compete with their corresponding father according to the index of the population and the winner survive for next iteration:
    if $f(u_i(g+1)) \leq f(x_i(g))$ **then**
      $x_i(g+1) \leftarrow u_i(g+1)$
    else
      $x_i(g+1) \leftarrow x_i(g)$
    end if
    if the prediction lable of neural network $f(\mathbf{x} + x_i(g+1))$ changes **then**
      return $x_i(g+1)$
    end if
  end for
end for

算法 4-1 寻找敏感点算法

由上述伪代码可知，寻找敏感点算法时间复杂度由迭代次数 maxIter 和种群大小 popSize 共同决定，其最坏时间复杂度为：O(maxIter*popSize)。因算法会在迭代过程中满足结束条件时便终止运行，所以平均时间复杂度会小于最坏时间复杂度。

4.1.4 实验结果

我们以 Cifar-10 数据集与 ResNet 网络模型为例进行实验。Cifar-10 数据集共有 50000 张训练图片和 10000 张测试图片，每张是尺寸为 32×32 RGB 彩色图片，共有 10 种类别，分别为：飞机、鸟、鹿、青蛙、船、汽车、猫、狗、马和卡车。我们从 Cifar-10 数据集中随机每个类别选一张图片作为示例，如图

4-3 所示。我们设计的 ResNet 模型共有三种残差块构成，每种残差块依次有 Batch Normalization、Activation、Conv2D、Batch Normalization、Activation、Conv2D 六部分组成，每个残差块的输入与下一个残差块的输入相连，作用是减缓梯度消失，三种残差块的每部分具体 Output Size 如表 4-1 所示。模型训练超参数如表 4-2 所示。从图 4-4 可以看出 ResNet 模型最终达到了在训练数据集上损失函数 0.1324,准确率 0.9965,在验证数据集上损失函数 0.4967，准确率 0.9115，已经属于一个性能优异的模型。我们将针对此模型寻找敏感点。

图 4-3 Cifar-10 数据集示例

表 4-1 三种残差块的 Output Size

Layer	Residual Block1	Residual Block2	Residual Block3
Batch Normalization	32×32×16	16×16×32	8×8×64
Activation	32×32×16	16×16×32	8×8×64
Conv2D	32×32×16	16×16×32	8×8×64
Batch Normalization	32×32×16	16×16×32	8×8×64
Activation	32×32×16	16×16×32	8×8×64
Conv2D	32×32×16	16×16×32	8×8×64

表 4-2 模型超参数

Hyper Parameters	Value
Learning Rate	0.05
Batch Size	128
Epochs	200
Momentum	0.8
Weight Decay	0.0001

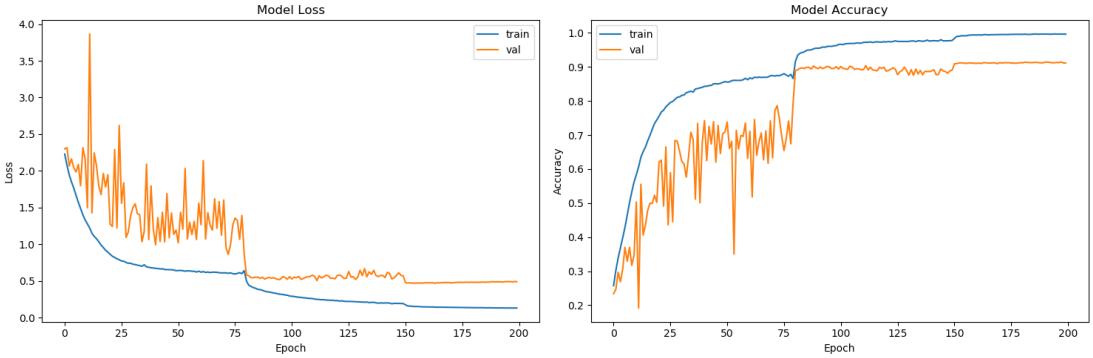

图 4-4 ResNet 模型在对应超参数下的损失函数和准确率相对训练次数的变化情况。

Goodfellow 等人提出的 FGSM 方法是一种被广泛使用的对抗样本生成方法。Goodfellow 等人认为神经网路模型的易受攻击性是因为模型的高维线性特征。由于目前大多数深度学习使用的模型或激活函数都非常接近线性或分段线性，如 ReLU 作为激活函数。因此 Goodfellow 等人基于线性的扰动可以成功攻击深度学习这一假设，提出了 FGSM 方法来生成对抗样本$\bar{x}$：

$$\bar{x} = x + \varepsilon sign\left(\nabla_x J(\theta, x, y)\right) \qquad (4-9)$$

其中x表示输入数据，y表示输入样本对应的类别，θ 表示模型参数，$J(\theta, x, y)$是模型的损失函数，ε正常数用来约束对输入图像的扰动规模。由公式可知对抗攻击方法要想能够生成对抗样本，深度学习模型必须可导。由于我们设计的深度学习模型可微分满足公式成立的条件，我们将以此对抗样本生成方法为例进行实验。

通过 FGSM 产生的对抗样本如图 4-5 右图所示，与左图原始输入图片对比可以发现，FGSM 产生的扰动很难用人类的肉眼察觉出来，也就是说人眼来分类的话，两幅图片的类别都是船，然而深度学习模型却把左图识别为船，右图识别为飞机。这也放映了深度学习模型对细微的变动是很敏感的。

图 4-5 左图为原始样本：船，右图为对抗样本：飞机

针对图 4-5 左图原始样本运用 FGSM 攻击方法生成的扰动如图 4-6 左图所示，可以发现 FGSM 是针对整张图片所有像素点产生细微扰动。如果我们能够找出方法可以过滤掉这些扰动的像素点，那么原对抗样本的分类将变为船，即成为正确的类别，也就是成功防御住了 FGSM 攻击方法的对抗攻击。随着深度学习模型应用越来越多的落地，人们对深度学习安全性也越来越重视，目前有很多研究人员研究针对深度学习对抗攻击的防御方法。防御方法总体可分为两大类，第一类为检测，也就是只负责发现输入图片是否遭受攻击方法的攻击，如果判断为遭受攻击则丢掉这张输入图片，不在传入深度学习做预测，这是一种懒汉行为，适用范围有限；第二类为过滤，也就是当发现输入图片遭受攻击方法攻击时会设计算法减少扰动对深度学习预测的影响，比如压缩图片。不同的过滤方法对输入样本的影响程度是不一样的，压缩图片会较大程度破坏原始输入样本。我们提出的基于差分进化的敏感点的寻找方法，可以最低程度对输入样本造成破坏，因为我们只会找出整张图片占比很少的几个像素点作为过滤点。

我们运用算法 4-1 寻找的敏感点如图 4-6 右图所示。我们通过敏感点寻找算法找出了六个像素点，这六个像素点是对深度学习预测值产生较大波动的点。也就是我们只需要针对图 4-5 右图所示对抗样本过滤掉对应的图 4-6 右图中的六个像素点，便可以让深度学习对过滤掉六个点的对抗样本做出正确分类，即识别为船。通过图 4-6 的左右两图的对比可以发现，我们提出的基于差

分进化方法寻找敏感点的算法，可以比过滤大多数扰动像素点的方法对原始样本的破坏更小。

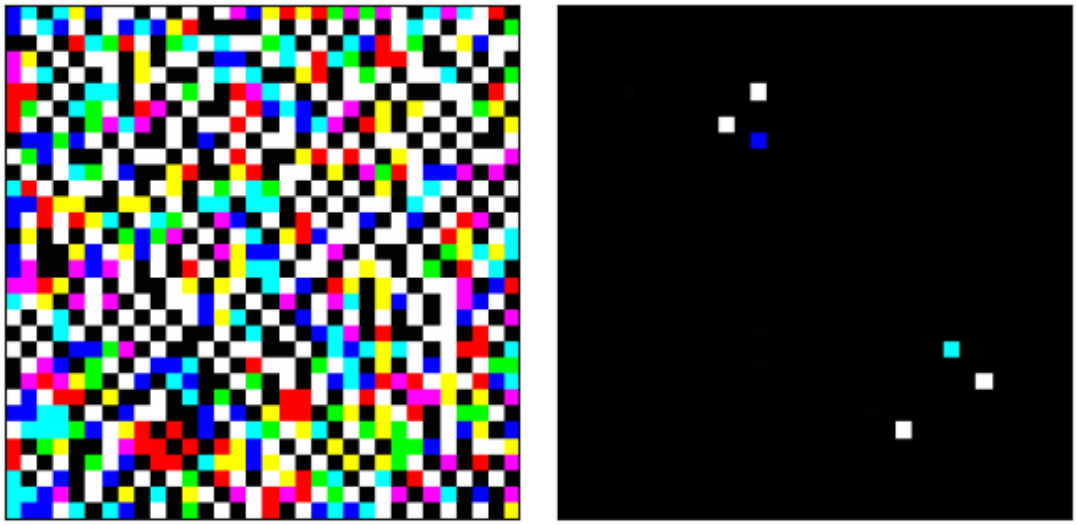

图 4-6 左图为 FGSM 产生的扰动，右图为算法 4-1 找到的敏感点

4.2 过滤敏感点

由于在上一节已经设计出了寻找敏感点的算法，接下来只需设计算法把这些对预测模型的输出能够产生较大波动的敏感点过滤掉，以起到防御的效果。

我们提出的过滤方法是，取每个敏感点的四周共计8个像素点的平均值作为该敏感点的过滤后的像素值，以起到平滑敏感点的作用。这8个点分别为敏感点的左上方、正上方、右上方、正左方、正右方、左下方、正下方、右下方。我们知道图片像素点的坐标是以左上方第一个点为原点的，从该点往下是x轴正方向，x轴坐标值递增，往右是y轴正方向，y轴坐标值递增。那么设敏感点的坐标为(x，y)则四周的8个点相对此敏感点的坐标为(x-1，y-1)，(x-1，y)，(x-1，y+1)，(x，y-1)，(x，y+1)，(x+1，y-1)，(x+1，y)，(x+1，y+1)。相对于敏感点(x，y)的坐标如图4-7所示。那么过滤后的坐标(x，y)处的数值为：

$$\overline{Pixel}(x,y)=\frac{1}{m}\left\{\sum_{i=-1}^{1}\sum_{j=-1}^{1}Pixel(x+i,y+j)-Pixel(x,y)\right\} \qquad (4-10)$$
$$s.t. \quad 0\le x+i\le width,$$
$$0\le y+j\le length.$$

其中，$width$表示输入图像的宽度，$length$表示输入图像的长度，$Pixel$(a,b)表示坐标(a,b)处的RGB像素值,$\overline{Pixel}(x,y)$表示坐标为(x,y)的敏感点被过滤之后

的RGB像素值，m表示敏感点(x,y)四周能够满足约束条件的像素点的个数。需要特别说明的是，必须要有上式中的两个约束条件，因为每个敏感点四周像素点的数量不一定为8个，比如当敏感点的坐标为第一行、最后一行、第一排或最后一排时，周围像素点的数量会小于8个。

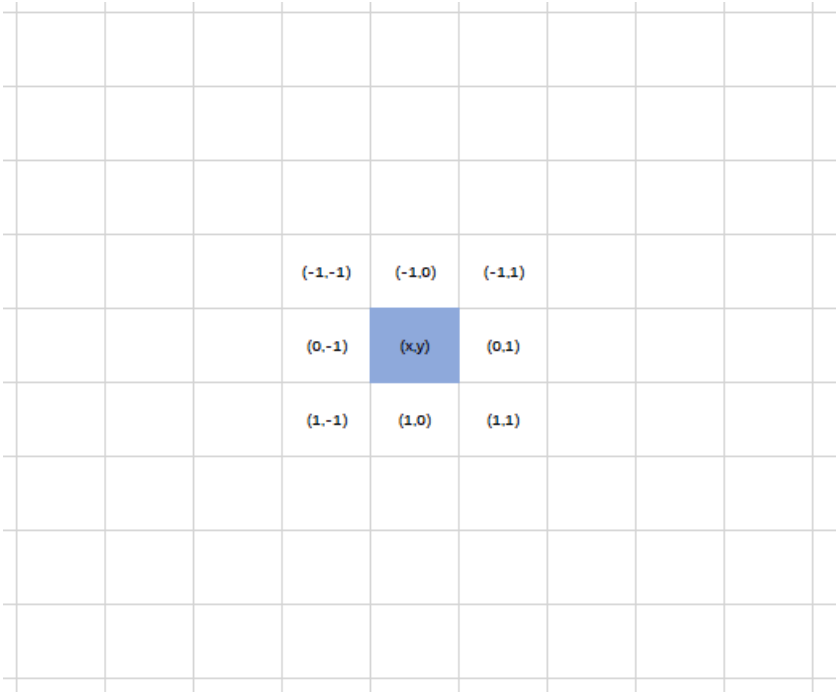

图 4-7 敏感点(x, y)四周 8 个点的相对坐标表示

为了方便计算，我们采用方向数组的形式来表示敏感点四周的像素点的坐标，即令dx = [-1,-1,-1,0,0,1,1,1]，dy = [-1,0,1,-1,1,-1,0,1]分别表示x轴和y轴的坐标相对变换。也就是在寻找相对于中心坐标(x,y)的四周的像素点时，寻找顺序依次为左上方、正上方、右上方、正左方、正右方、左下方、正下方和右下方，当发现坐标超出图片的长度或宽度时，则跳过该点，继续往下寻找。举例来说，当要寻找左上方的像素点时，只需令x = x + dx[0]，y = y + dy[0]，寻找右下方的像素点时，只需令x = x + dx[7]，y = y + dy[7]。每找到一个合法的像素点则像素点的个数m加一。遍历完方向数组，也就是寻找完周围的像素点，则把找到的像素点的RGB像素相加，求平均值，然后把平均值赋值为敏感点的新值。我们之所以这么做，是因为我们对敏感点的定义是该点像素值的波动会对深度学习预测模型的输出产生较大波动，既然该像素点有这种影响整个深度学习预测的能力，那么该点的值对于周围非敏感点的像素值而言，必定数值差异较大，故而我们用周围非敏感点的像素值的平均值来平滑该点的像素值，使得该点得

值接近周围非敏感点得值，进而起到过滤敏感点，减少对深度学习预测模型得干扰，起到防御作用。过滤敏感点的算法流程如算法4-2所示。

Algorithm 2 Filter Sensitive Points

Input: Initialize the following parameters.
(1) the input image: $\mathbf{X}$
(2) the width of $\mathbf{X}$: $width$
(3) the length of $\mathbf{X}$: $length$
(4) the coordinates of sensitive points: $\mathbf{SPC}$

Output: the filtered image.

$dx \leftarrow [-1,-1,-1,0,0,1,1,1]$
$dy \leftarrow [-1,0,1,-1,1,-1,0,1]$
for $c = 0, ..., len(\mathbf{SPC})$ **do**
  $x \leftarrow \mathbf{SPC}[c][0]$
  $y \leftarrow \mathbf{SPC}[c][1]$
  $sum \leftarrow 0$
  $m \leftarrow 0$
  for $d = 0, ..., 7$ **do**
    $i \leftarrow dx[d]$
    $j \leftarrow dy[d]$
    $x\_i \leftarrow x + i$
    $y\_j \leftarrow y + j$
    if $0 \leq x\_i \leq width$ **and** $0 \leq y\_j \leq length$ **then**
      $sum \leftarrow sum + \mathbf{X}[x\_i][y\_j]$
      $m \leftarrow m + 1$
    end if
  end for
  $\mathbf{X}[x][y] \leftarrow sum/m$
end for
return X

算法 4-2 过滤敏感点

由上述伪代码可知，过滤敏感点算法的时间复杂度主要由敏感点数决定，而敏感点数是一个超参数，其数值占整张图片像素点总数的比值很小，如在32×32大小的图片中，敏感点数取100时，仅占比约9.8%。因防御策略是先运用差分进化算法找到敏感点，然后依次过滤每一个敏感点，故总的时间复杂度为O(d* maxIter*popSize)。

在实验开始之前需要验证我们所设计的防御策略对非对抗样本是否由破坏，以至于导致非对抗样本在防御后传入模型分类错误。由于防御策略中的敏感点数d是个超参数，我们以d=10为例进行实验。我们随机从数据集中选择100张可以被正确分类的样本，然后计算防御后类别与真实类别不相同的样本数，结果如表4-3所示。

表 4-3 防御后类别与真实类别不相同的样本数

Model	Cifar-10	MNIST
ResNet	2	1
AlexNet	0	4
LeNet	2	0

由表4-3可以发现，防御后类别与真实类别不相同的样本数很少，也就是说我们所设计的防御策略对非对抗样本的破坏是很小的，这主要是因为我们设计的过滤敏感点的操作是取敏感点周围非敏感点的均值作为敏感点的新值，而非对抗样本并没有添加扰动，由图片的局部相似性特征可知，这时所取均值与原值基本一致，故而对模型的输出影响不大。

4.3 实现防御策略

实验流程如下如图所示：

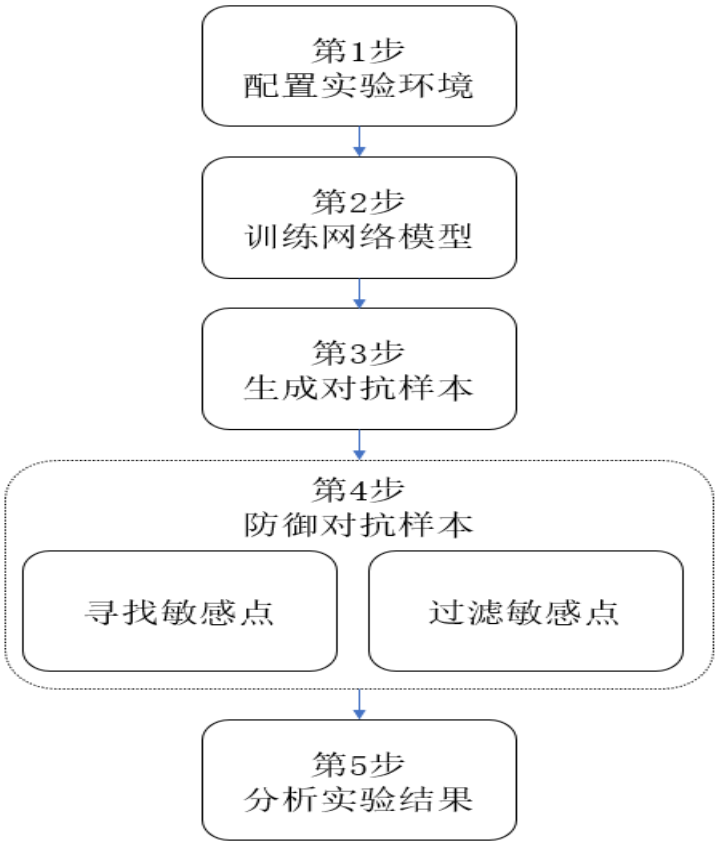

图 4-8 实验流程

4.3.1 配置实验环境

本课题全部实验是在本人个人电脑上完成的，具体实验环境配置如表 4-4 所示。

表 4-4 实验环境规格表

环境	规格
GPU	GTX 1050 Ti
内存	8GB
硬盘	500GB
操作系统	Windows 10
编程语言	Python 3
深度学习框架	PyTorch,TensorFlow,Keras

我们知道当前的深度学习框架都已经支持GPU运算，而深度学习模型往往需要大量的运算，如果使用GPU并行加速将大大节约运行时间。所以我们实验中所有深度学习模型训练与测试部分都放在GPU中去计算，加快了实验进程。我所使用的GPU为1050 Ti，专用GPU内存为4GB，为了方便本地电脑实验，所用深度学习模型深度、结构等较为简单。

4.3.2 训练网络模型

由于本课题需要实验防御方法在 ResNet、AlexNet 和 LeNet 上的防御效果，所以我们需要设计这三种经典深度学习模型。我们所实现的深度学习模型针对的数据集为 Cifar-10 和 MNIST。Cifar－10 数据集类别为飞机、汽车、卡车、船、青蛙、鸟、马、猫、狗和鹿，每张图片大小为 32×32，RGB 彩色通道为 3，MNIST 的类别为 0 到 9，每张图片大小为 28×28，黑白色通道为 1。即两种数据集都是 10 分类问题。所以本课题实验所需的对这两种数据集分类的深度学习模型结构基本一致，区别在于输入层的大小不同，为了实验方便，受迁移学习启发，我们可以先设计好在 Cifar-10 数据集上的深度学习模型，其输入层为 32×32×3，复用其网络结构然后在输入层之前再加一层成为新的输入层大小为 28×28×1，并使该层的输出为 32×32×3。这样便可以不需要再重新设计针对 MNIST 分类的深度学习模型了。

我们在 Cifar-10 数据集和 MNIST 上训练的 LeNet、AlexNet 深度学习模型结构如表 4-5 和表 4-6 所示。由于实验中为了得到更高的分类准确率，我们使用的 ResNet 模型为了 ResNet50,其残差块共有三种类型，具体输出大小如表 4-7 所示。

表 4-5 LeNet 模型网络结构

类型	结构
卷积层	卷积核大小：(5,5)；卷积核数量：6
池化层	池化窗口：(2,2)；最大池化
卷积层	卷积核大小：(5,5)；卷积核数量：60
池化层	池化窗口：(2,2)；最大池化
卷积层	卷积核大小：(5,5)；卷积核数量：1920
全连接层	参数：121*84
输出层	神经元数量：10 ；Softmax

表 4-6 AlexNet 模型网络结构

类型	结构
卷积层	卷积核大小：(3,3)；卷积核数量：24
池化层	池化窗口：(2,2)；最大池化
卷积层	卷积核大小：(3,3)；卷积核数量：96
池化层	池化窗口：(2,2)；最大池化
卷积层	卷积核大小：(3,3)；卷积核数量：192
卷积层	卷积核大小：(3,3)；卷积核数量：192
卷积层	卷积核大小：(3,3)；卷积核数量：96
池化层	池化窗口：(3,3)；最大池化
全连接层	参数：98340×1024
全连接层	参数：1024×1024
输出层	神经元数量：10 ；Softmax

表 4-7 ResNet 模型残差块结构

类型	输出大小
残差块 1	32×32×16
残差块 2	16×16×32
残差块 3	8×8×64

对上述三种模型进行训练，最终在 Cifar-10 与 MNIST 数据集上的验证集分类准确率如表 4-8 所示。我们将以此准确率的模型作为后续生成对抗样本和验证防御效果的所用模型。

表 4-8 模型在验证数据集上的分类准确率

Model	Cifar-10	MNIST
ResNet	91.15%	97.25%
AlexNet	78.46%	90.32%
LeNet	72.69%	97.92%

需要说明的是，如果一个原始样本（未添加任何扰动）的真实类别与模型的预测类别不一致，那么这个样本不能叫做对抗样本，这是模型本身没有训练好，并不是攻击者造成的。对抗攻击是对本来被预测正确的样本进行攻击使其被预测错误，所以 AlexNet 与 LeNet 在 Cifar-10 上准确率较低并不会明显影响我们后续实验，唯一的影响是可能可以用来产生对抗样本的原始样本较少而已，但是我们可以通过对同一个原始样本生成多张对抗样本来解决这个问题，以保证最后对抗样本的总数。

4.3.3 生成对抗样本

本课题用到了三种对抗攻击方法，分别为 FGSM（快速梯度符号法）、BIM（基础迭代法）和 PGD（投影梯度下降法）。这三种对抗攻击方法都是当前验证对抗防御常用的经典方法。我们将使用这三种对抗攻击方法生成对抗样本，用于下一步测试对抗防御效果。

用 FGSM 攻击方法生成对抗样本时，变量ε用来控制所加扰动的幅度，当该变量值较小时，生成的对抗样本很难用人眼察觉到与原始样本有区别，这种不易察觉扰动的样本生成是特别困难的，实验发现当$\varepsilon = 0.01$ 时，验证集能够生成对抗样本的比例为 1.30%,大约 10000 张验证集中只有 130 张图片可以生成

对抗样本，然而随着ε值得不断增大，可以生成对抗样本得比例会有显著提升，如当$\varepsilon = 0.85$ 时，验证集中可以生成对抗样本得图片达到 99.21%。然而，随着ε值得增大，生成得样本与原始样本的区别会特别明显，甚至人类肉眼都再难以辨认图片得类别，我们不把这种样本认为是对抗样本，因为这种样本不仅仅欺骗了深度学习还欺骗了人类的眼睛。所以我们两者取中，选择一个ε值既可以保证攻击成功比例又能不让加在原始样本中的扰动过于明显，实验发现ε取 0.55 比较合适，此时验证集能够生成对抗样本的比例为 82.79%，此时，在 Cifar-10 数据集上生成的对抗样本如图 4-10 所示。

BIM 攻击算法是 FGSM 的迭代版本，所以 BIM 也包含变量ε，并且还包含迭代次数 iterations。BIM 与 FGSM 的主要区别是，BIM 是每次迭代只更新一小步，然后一步一步的累加。如果在达到设置的迭代最大次数前就已经可以欺骗深度学习，那么就停止；如果达到设置的最大次数时依然不能欺骗深度学习，那么不再继续迭代下去，这种情况意味着该原始样本不能被 BIM 攻击方法所攻击。迭代次数是一个超参数，实验发现迭代次数并不是越高越好，有时在十余次迭代时就已经可以发现该原始样本不能被 BIM 攻击了，如果还是继续迭代下去就会做无用功，浪费算力。在固定迭代次数 iterations=50 的情况下，不同的ε值带来的 BIM 攻击 Cifar-10 数据集的攻击成功率如表 4-9 所示。

表 4-9 BIM 在不同ε下模型的攻击成功率

深度学习模型	ε		
	0.01	0.03	0.05
ResNet	76.52%	93.15%	99.36%
AlexNet	83.29%	95.02%	99.18%
LeNet	90.63%	93.69%	99.73%

由表 4-9 可以发现ε的值要比 FGSM 攻击时小很多，这是因为 BIM 是迭代式攻击每次更新时只添加一点扰动，FGSM 是一次性加入所有扰动，所以 FGSM 为了使一次性攻击成功的可能性大一些需要ε的值要比 BIM 大。另外可以发现，随着ε值得不断增大，BIM 对三种模型得攻击成功率都有明显的提升，但是与 FGSM 相同，随着ε值的增大生成的对抗样本所加的扰动会特别明

显，对原始样本破坏较大，所以我们取 0.03 作为ε的值，此时三种模型下的攻击成功率都大于 90%。

PGD 攻击方法也是 FGSM 的迭代版本，与 BIM 攻击方法的区别在于 PGD 迭代次数更多，每次迭代添加的扰动更加微小，并且引入了随机性，使得生成的对抗样本更加不易人眼察觉。要想实现 PGD 攻击，需要确定扰动幅度ε和迭代次数 iterations，由于 PGD 支持更多轮次的迭代，所以我们初始迭代次数为 BIM 的最终确定迭代次数 50，然后以 10 为增量，不断增大迭代次数，找到一个合适的迭代次数，以加快生成对抗样本的速度。至于扰动幅度ε，我们也是以 BIM 的最终确定值 0.03 为初始值，不断以-0.001 为增量，不断减少扰动幅度，争取找到最佳值使得 PGD 攻击方法既能保证生成对抗样本的质量又能提高攻击成功率，加快生成大批量对抗样本。

我们以 Cifar-10 数据集为例，分别实现上述三种对抗攻击方法对同一张输入样本的攻击，如图 4-9 所示。

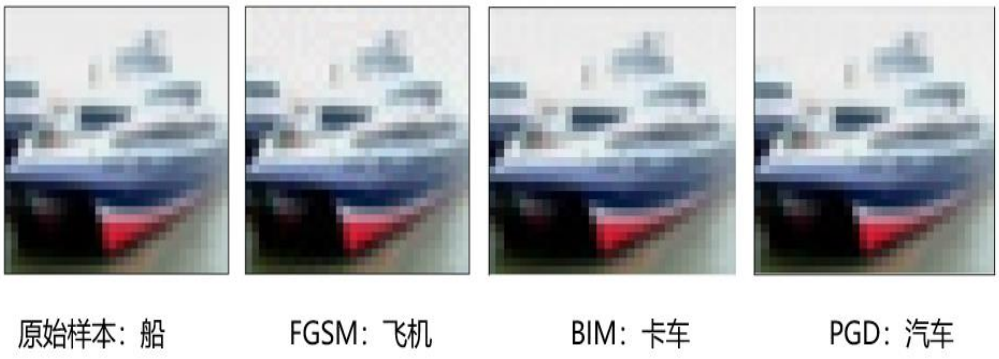

图 4-9　FGSM、BIM 和 PGD 三种攻击方法对比图

通过上图，可以发现 FGSM 攻击方法会把真实类别为船的原始样本分类为飞机，BIM 会把船分类为卡车，PGD 会把船分类为汽车。并且通过与第一张图对比，可以看出第二张图所加扰动较为明显，BIM 与 PGD 所加扰动较难用肉眼发现。

我们为每个深度学习模型在 FGSM、BIM、PGD 三种攻击方法下按照数据集中训练集与测试集的比例生成 10000 张对抗样本用来计算防御成功率，我们知道 Cifar-10 数据集训练集个数/测试集个数为 50000/10000，MNIST 数据集对应的训练集与测试集比值为 60000/10000，每种数据集的对应生成的对抗样本数量如表 4-10 所示。

如表 4-10 所示，我们随机从 Cifar-10 训练集中选择 8333 张、测试集中选择 1667 张用来生成对抗样本，当所选样本发现不能生成对抗样本时，解决办法是再次从数据集中随机选择一张作为替代，重复此操作直至能生成对抗样本为止。对 MNIST 数据集生成对抗样本的操作与此相同。每一种攻击方法都是按照此方法生成 10000 张对抗样本，用来验证防御方法对每一种攻击方法的防御能力。

表 4-10 数据集在三种攻击方法下生成的对抗样本数量

数据集	总数	训练集	测试集
Cifar-10	10000	8333	1667
MNIST	10000	8571	1429

4.3.4 实验结果与分析

本小节我们将通过实验验证我们的防御模型对于不同的对抗攻击方法在不同深度学习预测模型和不同数据集上的防御效果。其中，对抗攻击方法有三种分别为 FGSM、BIM、PGD，深度学习预测模型有三种分别为 ResNet、AlexNet 和 LeNet，数据集有两种，分别为 Cifar-10 和 MNIST。在前面的小节我们已经训练好了深度学习模型，并且用对应的对抗攻击方法在相应的数据集上生成了对抗样本。

防御对抗样本有两个步骤，第一步是找出对抗样本中的敏感像素点，第二步是对每个敏感点过滤使之成为非敏感点。在用差分进化算法寻找敏感点时，敏感点的数量是一个超参数，一张输入图像中可能有一个也可能有多个敏感点。但是并不是敏感点数量越多越好，因为如果我们把所有像素点都认为是敏感点，那么意味着所有像素点都是关键分量，这和我们理论分析部分——部分像素点决定了模型的输出类别——是不符的。另外，设定的敏感点数量越多则计算就越多，这和我们的出发点——提出一种高效且普适性强的防御策略——也不相符。因此，我们分别实验在敏感点数量 d 为 1、10、50 和 100 时，防御方法在不同的深度学习预测模型的基础上对不同的对抗攻击方法的防御能力。接下来，我们将分别通过实验验证我们设计的防御方法在 Cifar-10 与 MNIST 数据集上的防御效果并进行结果分析。

(1) Cifar-10 数据集实验与结果分析

Cifar-10 数据集对应的对抗样本已经在上一小节生成完成，接下来只需要对每张对抗样本寻找敏感点和过滤敏感点，然后传入深度学习模型做预测，判断预测类别是否是该样本的真实类别，如果是则表明对该对抗样本防御成功，否则防御失败。由于运用差分进化算法寻找敏感点时，初始种群的生成与交叉变异等操作涉及到随机值，因此为了合理计算防御成功率，我们对防御模型的防御成功率的计算方式为：分别独立计算三次防御成功率取其平均值作为最终结果。表 4-11、表 4-12、表 4-13 和表 4-14 分别表示 d = 1、d = 10、d = 50、d = 100 时的防御方法对对抗攻击的防御成功率。

表 4-11 d=1 时 防御策略的防御成功率

深度学习模型	对抗攻击方法		
	FGSM	BIM	PGD
ResNet	5.43%	0.26%	0.23%
AlexNet	5.62%	0.73%	0.95%
LeNet	5.94%	1.77%	1.42%

由表 4-11 可以发现，当敏感点数指定为 1 时，防御方法在三种模型上的防御效果都不高。当用 FGSM 方法攻击 ResNet 模型时，防御方法只能防御住 5.43%的对抗样本，当用 BIM 和 PGD 去攻击 ResNet 时防御能力会更低。这主要是因为在 32*32 共 1024 个像素点中，我们用差分进化算法找的敏感点与对抗样本实际存在的敏感点位置相同时才会防御成功，这个概率本身就很低。通过比较可以发现，防御方法在 LeNet 和 AlexNet 上的防御能力要好于 ResNet，这是因为 AlexNet 与 LeNet 两个模型的结构比较简单，相比 ResNet 更容易受到攻击，生成的对抗样本扰动幅度较大，更容易找到敏感点。即防御方法面对越复杂的深度学习模型，越细微的攻击方法，表现会越差。

表 4-12 d=10 时 防御策略的防御成功率

深度学习模型	对抗攻击方法		
	FGSM	BIM	PGD
ResNet	52.23%	2.18%	1.69%
AlexNet	52.89%	3.24%	2.98%
LeNet	53.66%	5.29%	3.41%

表 4-12 是当敏感点数指定为 10 时，对抗方法在三种模型上面对不同攻击方法的防御成功率。可以发现，当 d=10 时防御成功率会比 d=1 时有提升，其中防御 FGSM 攻击方法提升最为明显，同一模型面对 BIM 和 PGD 攻击时防御能力要比 FGSM 差很多。结果表明，找到并过滤的敏感点数量增多，防御成功的可能性会增大。这是因为，在一定范围内，随着指定敏感点数量 d 的增大，演化算法找到的敏感点与对抗样本实际的敏感点相同的可能性就会越高。

表 4-13 d=50 时 防御策略的防御成功率

深度学习模型	对抗攻击方法		
	FGSM	BIM	PGD
ResNet	70.11%	17.18%	14.45%
AlexNet	71.39%	20.34%	18.32%
LeNet	73.08%	20.53%	21.16%

由表 4-13 可以发现，三种模型面对 FGSM 攻击时，防御成功率达到 70%以上，比对 BIM 和 PGD 的防御能力高很多。这主要是因为后两种攻击方法所产生的对抗扰动比较微小，以至于很难找到样本种的实际敏感点。另外，由表可知，三种不同的深度学习模型面对同一种攻击时，我们的防御方法的防御成功率相差不是很大，这也一定程度说明我们所设计的防御策略可以用在多种深度学习模型之上。

表 4-14 d=100 时 防御策略的防御成功率

深度学习模型	对抗攻击方法		
	FGSM	BIM	PGD
ResNet	84.26%	32.25%	21.33%
AlexNet	83.57%	36.68%	21.87%
LeNet	85.13%	38.97%	23.16%

由表 4-14 可以发现，当用 FGSM 方法攻击 ResNet 模型时，防御方法可以防御住 84.26%的对抗样本，这时指定的敏感点数量仅为 100 个。通过上表可以看到三种模型面对 FGSM 攻击时的防御成功率都很高，面对 BIM 攻击时成功率在 35%左右，面对 PGD 攻击时防御成功率在 20%以上。这说明只通过修改对抗样本中的部分像素点起到防御作用的想法是可行的，同时说明我们所设

计的防御方法，可以普适的用在不同的攻击方法和深度学习模型中，可以应对多种对抗攻击方法的攻击。

为了便于观察防御方法在每种攻击方法的防御效果，我们把防御方法在三个模型上的防御成功率绘制成图，如图 4-10、图 4-11 和图 4-12 所示。

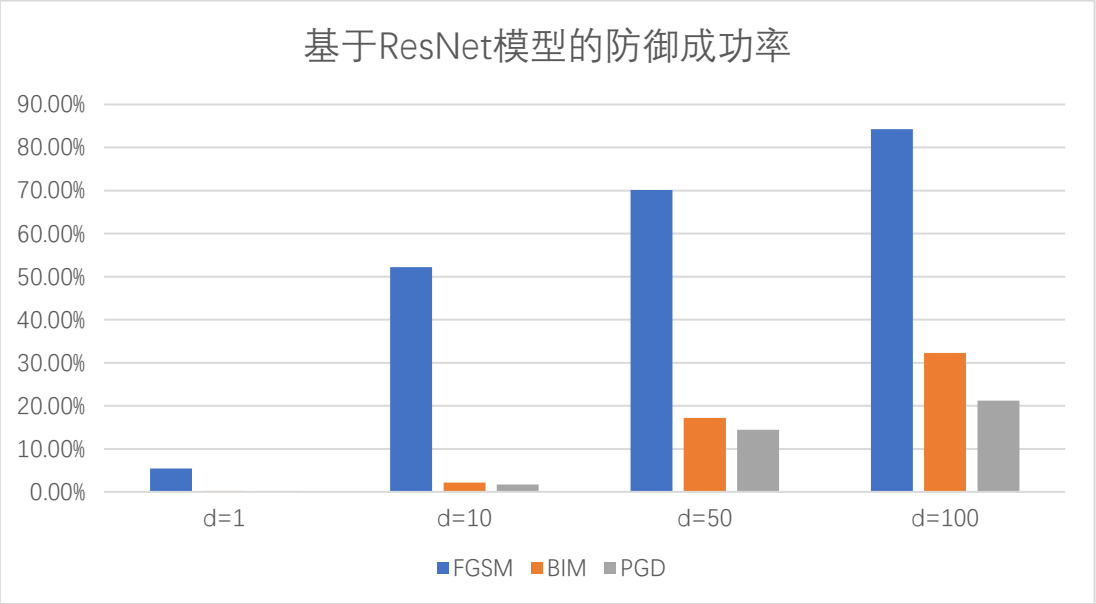

图 4-10 基于 ResNet 模型的不同攻击方法的防御成功率

由上图可知，在 ResNet 模型上的防御成功率会随着敏感点数 d 的增大而提升。我们所设计的防御方法在 ResNet 模型上面对 FGSM、BIM 和 PGD 三种方法的攻击，防御成功率呈递减趋势。其中对 FGSM 的防御能力提升最明显，当 d=100 时已经可以防御 80%以上的对抗样本。对其他两种攻击方法的防御能力也会随着敏感点数目的增多而提升，只是增速不是很高。这是由于后两种攻击方法产生的对抗扰动十分微小导致的。

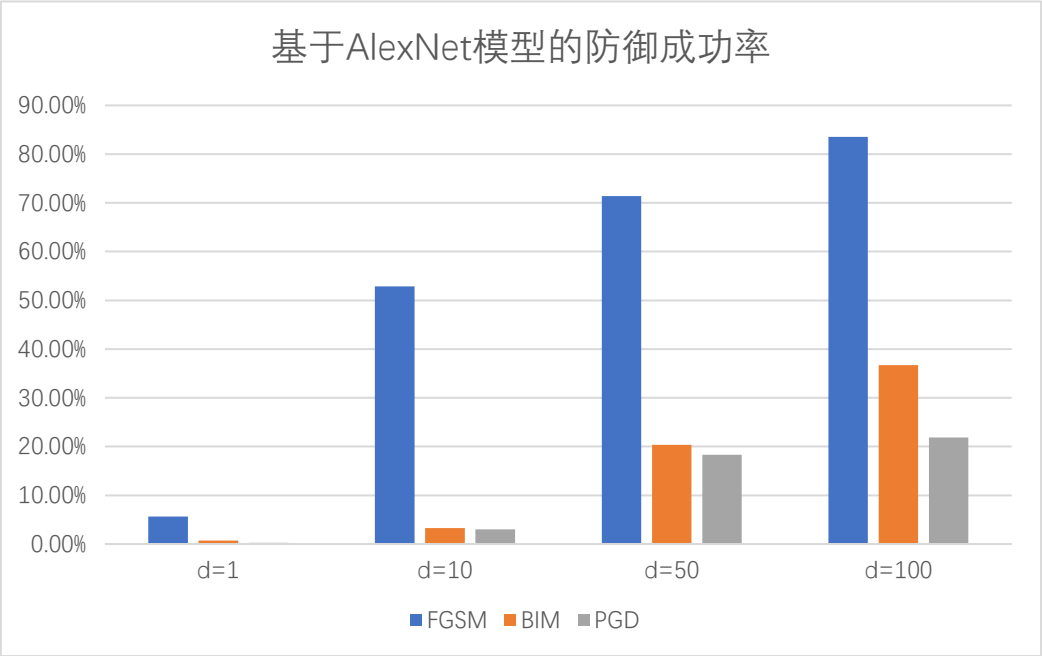

图 4-11 基于 AlexNet 模型的不同攻击方法的防御成功率

由图 4-11 可以发现，我们所实现的防御方法在 AlexNet 上的防御效果基本与 ResNet 模型一致，区别在于当指定相同敏感点 d 时，防御方法在 AlexNet 上的表现会略好于 ResNet。

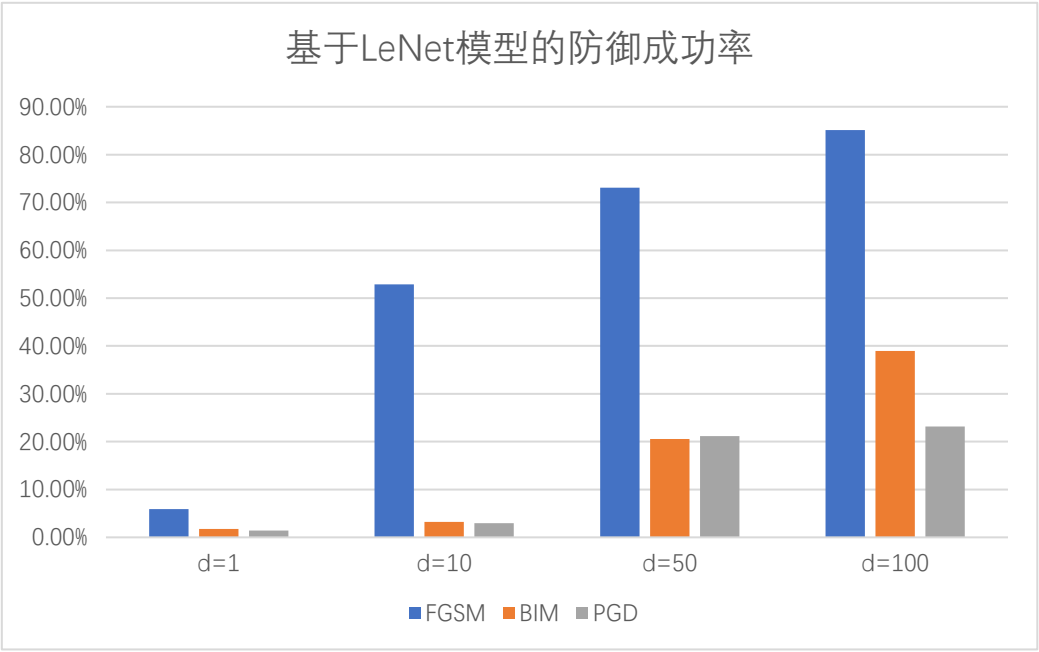

图 4-12 基于 LeNet 模型的不同攻击方法的防御成功率

由图 4-10、图 4-11 和 图 4-12 可以发现：我们设计的对抗防御模型可以抵抗在 ResNet、AlexNet 和 LeNet 上产生的对抗样本的攻击。其中在 LeNet 上的

防御效果最好，在 AlexNet 上的防御效果次之，在 ResNet 的防御效果再次之。但是总的看，防御方法在三个模型上的表现差异不会很大。这说明防御方法对模型结构参数等的差异不是很敏感，具有通用性。

(2) MNIST 数据集实验与结果分析

由于 MNIST 数据集中每张图片是黑白色的，与 Cifar-10 数据集图片都是彩色不同。我们从 MNIST 数据集中随机每个类别选一张图片作为示例，如图 4-13 所示。我们有必要验证我们所提出的防御方法在 MNIST 数据集上的防御效果。MNIST 数据集对应的对抗样本已经在之前生成完毕，接下来只需要对张对抗样本执行防御操作，然后计算防御成功率。其防御成功率的计算方式与 Cifar-10 数据集一样。表 4-15、表 4-16、表 4-17 和表 4-18 分别表示 d = 1、d = 10、d = 50、d = 100 时的防御方法对对抗攻击的防御成功率。

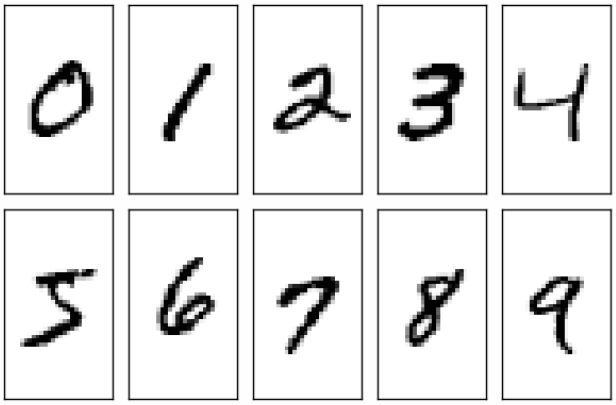

图 4-13 MNIST 数据集示例

表 4-15 d=1 时 防御策略的防御成功率

深度学习模型	对抗攻击方法		
	FGSM	BIM	PGD
ResNet	5.53%	1.17%	0.87%
AlexNet	5.94%	1.24%	1.09%
LeNet	6.35%	2.31%	1.65%

表 4-16 d=10 时 防御策略的防御成功率

深度学习模型	对抗攻击方法		
	FGSM	BIM	PGD
ResNet	54.41%	5.17%	5.20%
AlexNet	54.64%	6.83%	5.54%
LeNet	65.93%	5.89%	6.14%

表 4-17 d=50 时 防御策略的防御成功率

深度学习模型	对抗攻击方法		
	FGSM	BIM	PGD
ResNet	71.06%	20.57%	18.65%
AlexNet	79.20%	23.25%	21.66%
LeNet	80.37%	22.92%	23.84%

表 4-18 d=100 时 防御策略的防御成功率

神经网路模型	对抗攻击方法		
	FGSM	BIM	PGD
ResNet	80.06%	36.25%	31.27%
AlexNet	87.27%	39.74%	34.10%
LeNet	86.18%	40.61%	38.91%

由表 4-15、表 4-16、表 4-17 和表 4-18 可以发现，在 MNIST 数据集上，随着超参数敏感点 d 的数值不断增大，防御效果会越明显。当 d=100 时，防御方法对 FGSM 的防御效果最好，三种模型都在 80%以上，最好可以达到 87.27%。由上述表格可知，我们所设计的防御方法可以在多种深度学习模型之上对不同的攻击方法起到防御作用。

通过与 Cifar-10 数据集结果的比较可以发现，当攻击方法与深度学习模型相同时，在 MNIST 这种结构简单的数据集上的防御成功率会高于结构较复杂的 Cifar-10 数据集。同时也说明了我们所设计的防御方法可以在不同结构的数据集上起到防御效果，对数据集具有一定的普适性。

4.4 本章小结

本章，首先实现了基于差分进化算法寻找敏感点的算法，并实验验证了算法的正确性，然后实现了过滤敏感点的算法。将以上两种算法结合构成防御策略，详细设计了防御策略的实验流程，包含从配置实验环境、训练网络模型、生成对抗样本到防御对抗样本。重点分析了实验结果，表明我们所设计的防御方法，在三种不同结构的深度学习模型中表现基本一致，说明对模型具有一定的普适性，另外发现在敏感点数相同的情况下，我们的防御方法可以防御三种攻击方法，只是对FGSM防御效果较好，对BIM和PGD防御能力欠佳，若想提高防御成功率需进一步增加敏感点数，但这会增加计算量。通过在两种不同数据集上进行防御实验，结果表明我们设计的防御方法可以适应不同类型的数据集。

结　论

针对当前防御方法的缺点和不足，本文试图提出一种高效且普适性较好的防御方法。本文首先详细分析了多种经典对抗攻击方法，并对这些攻击方法按照攻击范围分类，从分类中抽象出共同点与不同点，发现对抗样本中存在敏感点，受次启发提出一种寻找并过滤对抗样本敏感点的防御方法。

本文的主要贡献如下：

(1) 提高了防御方法的普适性。普适性包含两方面。其一是对攻击方法普适，有些防御方法只对特定的攻击方法有效，我们所设计的防御方法是黑盒防御，所有攻击方法的本质都是在原始样本中添加扰动，我们防御时只会关心扰动的幅度和位置，不会关心是用何种攻击方法生成的扰动，使得该方法可以适用于多种攻击方法，解决了白盒防御的缺点。其二是对网络模型普适，有些防御方法要想生效是要涉及损失函数微分等操作导致适用范围有限，我们利用差分进化算法查找敏感点，该算法不会涉及网络模型的内部结构、损失函数等信息，只需要知道模型的输入输出即可，扩大了适用范围。

(2) 提高了防御方法的执行效率。本防御方法主要从两方面提高防御效率。第一是用局部过滤代替全局过滤，当前很多防御方法都是处理所有扰动来达到防御的目的，比如数据压缩，然而我们发现对抗样本中是存在敏感点的，仅仅防御几处敏感点便可以实现防御效果，减少了计算量，同时减轻了对样本的破坏。第二是我们提出的过滤敏感点策略——将敏感点周围的相邻非敏感点像素值的平均值作为敏感点的新值，不涉及微分等操作，计算更加简单。

(3) 为验证我们所提出的防御方法的普适性，分别实验了FGSM、BIM、PGD三种攻击方法在ResNet、AlexNet、LeNet三种深度学习模型和Cifar-10、MNIST两种数据集上的防御效果。首先，我们对三种网络模型进行训练使之能够对Cifar-10或MNIST正确分类，其次我们实现了FGSM、BIM与PGD三种攻击方法，并用这三种方法生成不同的对抗样本，最后我们用我们提出的防御方法验证对对抗样本的防御能力。实验结果表明，我们所设计的防御方法可以有效地防御不同深度学习模型和不同数据集上的不同攻击方法的攻击，表明防御方法具有一定的普适性。

在未来的工作中，我们将本文提出的防御方法应用到更加复杂的深度学习

模型和更大规模的数据集上，并试图进一步提升防御方法寻找敏感点的效率与防御成功率。

参考文献

[1] Szegedy C, Zaremba W, Sutskever I, et al. Intriguing properties of neural networks[C]. international conference on learning representations, 2014.

[2] Eykholt K, Evtimov I, Fernandes E, et al. Robust Physical-World Attacks on Deep Learning Visual Classification[C]. computer vision and pattern recognition, 2018: 1625-1634.

[3] Sitawarin C, Bhagoji A N, Mosenia A, et al. Rogue Signs: Deceiving Traffic Sign Recognition with Malicious Ads and Logos.[J]. arXiv: Cryptography and Security, 2018.

[4] Papernot N, Mcdaniel P, Swami A, et al. Crafting adversarial input sequences for recurrent neural networks[C]. military communications conference, 2016: 49-54.

[5] Lin Y, Hong Z, Liao Y, et al. Tactics of Adversarial Attack on Deep Reinforcement Learning Agents.[C]. international joint conference on artificial intelligence, 2017: 3756-3762.

[6] Huang S H, Papernot N, Goodfellow I, et al. Adversarial Attacks on Neural Network Policies[C]. international conference on learning representations, 2017.

[7] Moosavi-Dezfooli S, Fawzi A, Fawzi O, et al. Universal Adversarial Perturbations[C]. computer vision and pattern recognition, 2017: 86-94.

[8] Metzen J H, Kumar M C, Brox T, et al. Universal Adversarial Perturbations Against Semantic Image Segmentation[C]. international conference on computer vision, 2017: 2774-2783.

[9] Arnab A, Miksik O, Torr P H, et al. On the Robustness of Semantic Segmentation Models to Adversarial Attacks[C]. computer vision and pattern recognition, 2018: 888-897.

[10] Xie C, Wang J, Zhang Z, et al. Adversarial Examples for Semantic Segmentation and Object Detection[C]. international conference on computer vision, 2017: 1378-1387.

[11] Bhagoji A N, Cullina D, Sitawarin C, et al. Enhancing robustness of machine learning systems via data transformations[C]. conference on information sciences and systems, 2018: 1-5.

[12] Xu W, Evans D, Qi Y, et al. Feature Squeezing Mitigates and Detects Carlini/Wagner Adversarial Examples[J]. arXiv: Cryptography and Security, 2017.

[13] Luo Y, Boix X, Roig G, et al. Foveation-based Mechanisms Alleviate Adversarial Examples[J]. arXiv: Learning, 2016.

[14] Goodfellow I, Shlens J, Szegedy C, et al. Explaining and Harnessing Adversarial Examples[C]. international conference on learning representations, 2015.

[15] Huang R, Xu B, Schuurmans D, et al. Learning with a Strong Adversary[J]. arXiv:

Learning, 2015.
[16]Kurakin A, Goodfellow I, Bengio S, et al. Adversarial Machine Learning at Scale[C]. international conference on learning representations, 2017.
[17]Luo Y, Boix X, Roig G, et al. Foveation-based Mechanisms Alleviate Adversarial Examples[J]. arXiv: Learning, 2016.
[18]Guo C, Rana M, Cisse M, et al. Countering Adversarial Images using Input Transformations[C]. international conference on learning representations, 2018.
[19]Das N, Shanbhogue M, Chen S, et al. Keeping the Bad Guys Out: Protecting and Vaccinating Deep Learning with JPEG Compression.[J]. arXiv: Computer Vision and Pattern Recognition, 2017.
[20]Papernot N, Mcdaniel P, Jha S, et al. The Limitations of Deep Learning in Adversarial Settings[C]. ieee european symposium on security and privacy, 2016: 372-387.
[21]Ross A S, Doshivelez F. Improving the Adversarial Robustness and Interpretability of Deep Neural Networks by Regularizing their Input Gradients[C]. national conference on artificial intelligence, 2018: 1660-1669.
[22]Hinton G E, Vinyals O, Dean J, et al. Distilling the Knowledge in a Neural Network[J]. arXiv: Machine Learning, 2015.
[23]Lyu C, Huang K, Liang H, et al. A Unified Gradient Regularization Family for Adversarial Examples[C]. international conference on data mining, 2015: 301-309.
[24]Moosavidezfooli S, Fawzi A, Uesato J, et al. Robustness via Curvature Regularization, and Vice Versa[C]. computer vision and pattern recognition, 2019: 9078-9086.
[25]Akhtar N, Liu J, Mian A, et al. Defense Against Universal Adversarial Perturbations[C]. computer vision and pattern recognition, 2018: 3389-3398.
[26]Lee H, Han S, Lee J, et al. Generative Adversarial Trainer: Defense to Adversarial Perturbations with GAN.[J]. arXiv: Learning, 2017.
[27]Tramer F, Kurakin A, Papernot N, et al. Ensemble Adversarial Training: Attacks and Defenses[C]. international conference on learning representations, 2018.
[28]Madry A, Makelov A, Schmidt L, et al. Towards Deep Learning Models Resistant to Adversarial Attacks[C]. international conference on learning representations, 2018.
[29]Lecun Y, Bengio Y, Hinton G E, et al. Deep learning[J]. Nature, 2015, 521(7553): 436-444.
[30]Schmidhuber J. Deep learning in neural networks[J]. Neural Networks, 2015: 85-117.
[31]LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. proceedings of the IEEE, 1998, 86(11): 2278-2324.

[32]李彦冬，郝宗波，雷航. 卷积神经网络研究综述[J]. 计算机应用, 2016, 36(9): 2508-2515.

[33]Chen L, Zhu G, Li Q, et al. Adversarial Example in Remote Sensing Image Recognition[J]. arXiv: Computer Vision and Pattern Recognition, 2019.

[34]Krizhevsky A, Sutskever I, Hinton G E, et al. ImageNet Classification with Deep Convolutional Neural Networks[C]. neural information processing systems, 2012: 1097-1105.

[35]He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[C]. computer vision and pattern recognition, 2016: 770-778.

[36]Russakovsky O, Deng J, Su H, et al. ImageNet Large Scale Visual Recognition Challenge[J]. International Journal of Computer Vision, 2015, 115(3): 211-252.

[37]Everingham M, Eslami S M, Van Gool L, et al. The Pascal Visual Object Classes Challenge: A Retrospective[J]. International Journal of Computer Vision, 2015, 111(1): 98-136.

[38]Lin T, Maire M, Belongie S, et al. Microsoft COCO: Common Objects in Context[C]. european conference on computer vision, 2014: 740-755.

[39]Kurakin A, Goodfellow I, Bengio S, et al. Adversarial examples in the physical world[C]. international conference on learning representations, 2017.

[40]Su J, Vargas D V, Sakurai K, et al. One Pixel Attack for Fooling Deep Neural Networks[J]. IEEE Transactions on Evolutionary Computation, 2019, 23(5): 828-841.

[41]Narodytska N, Kasiviswanathan S P. Simple Black-Box Adversarial Perturbations for Deep Networks[J]. arXiv: Learning, 2017.

[42]闫明. 基于 DCT 变换的对抗样本防御方法研究[D].哈尔滨工业大学,2018.

[43]Storn R, Price K. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces[J]. Journal of Global Optimization, 1997, 11(4): 341-359.

[44]Civicioglu P, Besdok E. A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms[J]. Artificial Intelligence Review, 2013, 39(4): 315-346.

[45]Das S, Suganthan P N. Differential Evolution: A Survey of the State-of-the-Art[J]. IEEE Transactions on Evolutionary Computation, 2011, 15(1): 4-31.

攻读硕士学位期间发表的论文及其它成果

哈尔滨工业大学学位论文原创性声明和使用权限

学位论文原创性声明

本人郑重声明：此处所提交的学位论文《深度学习对抗攻击防御策略的研究与实现》，是本人在导师指导下，在哈尔滨工业大学攻读学位期间独立进行研究工作所取得的成果，且学位论文中除已标注引用文献的部分外不包含他人完成或已发表的研究成果。对本学位论文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。

作者签名：张成　日期：　2020　年 6　月　22 日

学位论文使用权限

学位论文是研究生在哈尔滨工业大学攻读学位期间完成的成果，知识产权归属哈尔滨工业大学。学位论文的使用权限如下：

（1）学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文，并向国家图书馆报送学位论文；（2）学校可以将学位论文部分或全部内容编入有关数据库进行检索和提供相应阅览服务；（3）研究生毕业后发表与此学位论文研究成果相关的学术论文和其他成果时，应征得导师同意，且第一署名单位为哈尔滨工业大学。

保密论文在保密期内遵守有关保密规定，解密后适用于此使用权限规定。

本人知悉学位论文的使用权限，并将遵守有关规定。

作者签名：张成　日期：　2020　年 6　月　22 日

导师签名：王岭　日期：　2020　年 6　月　22 日

致　谢

研究生阶段求学生涯即将结束，在这两年里我得到很多老师、同学、朋友和亲人的帮助，借此机会我要向你们表示诚挚的谢意。

首先感谢我的指导老师王玲教授，在整个研究生阶段王老师在学术和生活中，给了我很多启发和帮助。从研一开始，王老师就带着我们开组会，实验室同学一起分享一周来读过的文章、做过的实验和课题的进展，每周的组会总能学到很多知识，我的课题便是来源于组会中的讨论。王老师不仅善于启发我们去思考去讨论，她还是个治学态度严谨、了解学术前沿、学识渊博的人，在和老师的交流过程中总能感受到老师对学术的热爱。在我撰写毕业论文的过程中，王老师多次对我的论文进行逐字逐句的修改，付出了很多时间来指导我撰写论文的方法与经验教训，在每一次老师的批改中，她总能给出一些极好的建议，没有王老师的耐心且细致的指导，我是不可能完成的。我感到十分幸运能够在研究生阶段遇到王老师，并成为她的学生。

我还要感谢实验室里的同学们和室友们，他们是吴超、弥林瀚、王佳康、胡鉴、王笑、孙鹏重、吴东阳和李金泽。他们在我的论文书写过程中，给予了很多帮助，有的提出了很多建设性意见和建议，有的在我遇到困难的时候开导我，他们是一群可爱善良的孩子，我们虽然只相处了两年时光，但是我真诚地感谢他们存在于我的研究生生涯中，让我感觉到了温暖与快乐。

感谢我的家人和朋友，是你们给了我完成学业的支持与鼓励。我的父母在我的求学生涯中一直默默的付出默默的支持着我，是他们的支持让我才有机会来到哈工大读书，他们一直深深的爱着我、呵护着我。我的朋友们在我需要倾诉时，总能热情的帮助我开导我，你们对我的关心与关爱，我将铭记于心。

感谢哈工大对我的培养。哈工大校训“规格严格、功夫到家”时刻鞭策我前行，提醒我要在求学过程中严格要求自己、把本领学扎实学到位。今年母校刚好一百周年，趁此机会祝母校开启百年新篇章，越办越好。今年遇到严重疫情，是奋战在一线的医务工作者保障了我们的安全，在此一并表达我对他们的敬重与谢意。

最后，感谢各位专家百忙之中抽出宝贵时间评审我的论文，在此向你们致以最诚挚的谢意！