中图分类号：TP391 论文编号：1028716 20-S015
学科分类号：081200

硕士学位论文

卷积神经网络的鲁棒性在图像分类中的研究与应用

研究生姓名 谢烟平
学科、专业 计算机科学与技术
研 究 方 向 人工智能
指 导 教 师 谭晓阳 教授

南京航空航天大学

研究生院 计算机科学与技术学院

二O二O年三月

Nanjing University of Aeronautics and Astronautics
The Graduate School
College of Computer Science and Technology

Research and Application of the Robustness of Convolutional Neural Network in Image Classification

A Thesis in

Computer science and technology

By
Xie Yanping
Advised by
Prof. Tan Xiaoyang

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

Master of Engineering

March, 2020

承诺书

本人声明所呈交的硕士学位论文是本人在导师指导下进行的研究工作及取得的研究成果。除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得南京航空航天大学或其他教育机构的学位或证书而使用过的材料。

本人授权南京航空航天大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后适用本承诺书）

作者签名：_____

日　　期：_____

摘 要

深度学习已经得到飞速发展，在图像识别领域，卷积神经网络的鲁棒性保证显得尤其重要。由于卷积神经网络的分布式表示特点，会对输入图像产生众多特征表示，这些特征表示中存在大量噪声信息，会严重影响网络的性能，此外由于训练数据的局限性，会影响神经网络区分非正常输入的能力。为了提高卷积神经网络的鲁棒性，我们在噪声通道选择、噪声特征过滤和防御对抗攻击三个方面分别提出了三个方法。

针对噪声通道的存在而影响神经网络鲁棒性的问题，我们提出了基于通道选择的鲁棒性提高方法，通过重新分配通道权重来减少噪声通道的影响，其中我们提出了两阶段通道选择方法，利用空间聚合与信息融合来产生包含局部信息的通道描述向量。针对残差网络族的噪声特征问题，我们提出了基于特征过滤的鲁棒性提高方法，即一个即插即用的轻量级信息过滤模块，我们把它称为去噪编解码器，将它插入到跨层连接中，以特征过滤的方式来提高跨层连接所传递信息的质量。针对对抗样本普遍存在的问题，我们提出了基于对抗训练的鲁棒性提高方法，即一种基于生成式网络的对抗样本生成方法，用于模拟训练样本的邻域信息，以一种数据增强的方式，来提高神经网络的鲁棒性。

关键词：深度学习，分布式表示，鲁棒性，特征过滤，数据增强

ABSTRACT

Deep learning has developed rapidly. In the field of image recognition, the robustness guarantee of convolutional neural networks is particularly important. Due to the distributed representation characteristics of the convolutional neural network, the neural network will generate many feature representations of the input image. There is a lot of noise information in these feature representations, which will seriously affect the performance of the network. In addition, due to the limitations of training data, it will affect the ability of neural networks to distinguish abnormal inputs. In order to improve the robustness of the convolutional neural network, we propose three methods in three aspects: noise channel selection, noise feature filtering, and defense adversarial attacks.

Aiming at the problem that the existence of noisy channels affects the robustness of neural networks, we propose a method for improving the robustness based on channel selection, by reassigning channel weights to reduce the impact of noisy channels. We propose a two-stage channel selection method that uses spatial aggregation and information fusion to generate a channel description vector containing local information. To solve the problem of noise feature of residual network family, we propose a robust improvement method based on feature filtering, that is, a lightweight plug and play information filtering module, which we call denoising encoder-decoder, and insert it into the cross layer connection to improve the quality of information transmitted by the cross layer connection by feature filtering. In view of the common problems of the adversarial samples, we propose a method to improve the robustness of the neural network based on the adversarial training, that is, a method to generate adversarial samples based on the generative network, which is used to simulate the neighborhood information of the training samples and to improve the robustness of the neural network by a means of data augmentation.

Key Words: deep learning, distributed representation, robustness, feature filtering, data augmentation

目 录

图表清单

注释表

c_{in}	输入通道数	c_{out}	卷积核个数，输出通道数
k	卷积核大小	s	卷积步长大小
h_{int}	输入特征的大小	h_{out}	输出特征的大小
g	卷积分组数	r	空洞卷积抑制比例
p	特征边界填充数	x	输入特征，输入样本
$F(\cdot)$	卷积转换函数	$A(\cdot)$	激活函数
y	标签	w_y	属于第y类的分类向量
m	Softmax 角度约束项	a	Softmax 信号放大项
$f(\cdot)$	深度学习模型函数	x'	对抗样本
l	向量形式标签	δ	对抗扰动
J	损失函数	ϵ	扰动大小缩放因子
$Clip_{x,\xi}\{x'\}$	像素裁剪函数	ξ	裁剪限制因子
$J_f(x)$	样本的雅可比矩阵	$\mathcal{F}$	分类超平面
ϕ_k	神经网络的第k层输出	$L(x_{i,j}, x'_{i,j})$	亮度度量
$C(x_{i,j}, x'_{i,j})$	对比度度量	$S(x_{i,j}, x'_{i,j})$	结构度量
$G(\cdot)$	生成器函数	$I(\cdot)$	逆变器函数
$F_{sq}(\cdot)$	挤压函数	u	输出特征
$F_{ex}(\cdot)$	激励函数	$F_{scale}(\cdot)$	特征通道权重分配函数
$ED(\cdot)$	去噪函数	Δx	加性对抗噪声
$D(\cdot)$	判别器函数	λ	损失权重参数
$B_\epsilon^\infty(x)$	无穷范数球，半径ϵ，中心x	$Project_X(\cdot)$	投影函数

缩略词

缩略词	英文全称
DL	Deep Learning
CNN	Convolutional Neural Networks
FGSM	Fast Gradient Sign Method
OTCM	One-Step Target Class Method
BIM	Basic Iterative Method
ILLC	Iterative Least-Likely Class Method
JSMA	Jacobian-Based Saliency Map Attack
CPPN	Compo-sitional Pattern-Producing Network-encoded
EA	Evolutionary Algorithm
PASS	Psychometric Perceptual Adversarial Similarity Score
SSIM	Structural Similarity index
ZOO	Zeroth Order Optimization
PGD	Projected Gradient Descent
GAN	Generative Adversarial Network
SE	Squeeze-and-Excitation
GAP	Global Average Pooling
GMP	Global Max Pooling
SPSE	Spatial Pyramid pooling Squeeze-and-Excitation block
RGSE	Resolution-Guided pooling Squeeze-and-Excitation block
SGD	Stochastic Gradient Descent
CBAM	Convolutional Block Attention Module
ED	convolutional Encoder-Decoder

第一章 绪论

1.1 研究背景与意义

1.1.1 研究背景

自 2012 年以来，深度卷积神经网络（CNN）在图像分类[1][4][79][6][80]、语义分割[81]和许多其他任务[82][83][85][87][90]上取得了一系列突破。在大规模图像数据集（如 ImageNet[34]和 MS-COCO[26]）上进行监督训练后，深度网络主干部分也可以很好地推广到许多其他不同的任务上。

聚焦于图像识别任务，主干网络的性能是非常重要的，以多目标识别任务为例，一个几乎是成为常识的流水线被默认使用，即首先进行主干网络的预训练，然后在此技术上使用相关的多目标识别算法。一般主干网络的预训练指的是带着全连接层的卷积网络在 ImageNet 数据集上进行单目标识别的训练，主干网络指的是去掉全连接层的训练后的网络。

在目标检测领域，Girshick 等人首次利用深度卷积网络进行滑动窗口形式的目标检测任务[79]，虽然当时提出的方法实时性能差，理由是对于任何一张图片需要提取大量的候选区域，这些被提取并且保留的候选区域都需要经过卷积网络来进行特征提取，但是其首次成功尝试的意义对于目标检测任务是巨大的。后续又有研究者基于此提出了性能更好、检测速度更快的目标检测算法[91][6]。值得提出的是，这些目标检测算法都需要使用预训练的主干卷积网络。

不仅是多目标识别和目标检测，其他各种计算机视觉任务都需要使用预训练主干网络，所以提高卷积神经网络的鲁棒性至关重要。

1.1.2 研究意义

虽然深度学习在众多方面有了飞速发展，但是深度学习的不可解释性、深度卷积神经网络的数据依赖性以及输入敏感性都需要改善。近年来 Bengio 等人对深度卷积神经网络的表示形式进行了研究，发现卷积网络对输入图片是呈现出分布式表示的[3]，这意味着对任一图像，卷积网络的深度表示中存在着大量非前景激活，这些噪声激活会影响后续的结果判别。

另外一方面，Szegedy 等人首次提出了简单攻击深度卷积神经网络的方法[103]，即对抗样本。对抗样本指的是经过微小扰动的图片能够轻易地骗过卷积网络，使得其输出错误的结果，微小的扰动使得图片可以瞒过人类肉眼。这从侧面证明了卷积神经网络的鲁棒性是非常差的。

提高卷积神经网络的鲁棒性不仅能带来性能上的提升，更重要的是带来安全性保障。当今是深度学习技术尝试大量落地的时代，性能的保障至关重要，这决定了深度学习技术对社会的服务品质，安全的保障更为必须，因为这是深度学习技术可以存在的必要条件。

1.2 国内外研究情况

迄今为止，深度学习已经得到飞速发展，在卷积神经网络结构发展方面，众多结构脱颖而出。AlexNet[1]的出现标志着深度学习热潮的再次掀起。两年后 VGG 网络由牛津大学 Visual Geometry Group 提出[2]，并以此组缩写命名，VGG 网络向世人揭示了小卷积核的重要性，小卷积核能够在不严重加深网络计算复杂度的前提下增加网络的深度，提高网络的感受野，后续研究人员的网络设计无不借鉴 VGG 带来的经验。微软亚洲研究院的 He Kaiming 等人继续使用小卷积核，并且使用了恒等映射来减轻深度网络的学习难度，将恒等映射以跨层连接的形式实现，提出了 ResNet[4]，从此深度神经网络的深度可以达到上百层。近几年又有学者提出了各式各样的神经网络模型，有专门为移动设备提出的轻量级网络模型[64][66]，有进行多分支的残差网络模型[43]，也有对于残差模块进行深层讨论的多阶段模型[92]。

在特征选择方面，各种方法也层出不穷。Wang 等人在 ResNet 的基础上加上复杂的注意力机制模块，提出了 Residual Attention Network[93]。Hu Jie 等人通过引入挤压网络和激励网络来重新分配特征通道的重要度[14]，进一步提升了深度网络的性能，获得最后一届 ImageNet 大规模图像分类大赛冠军。在 SE-Net[14]的基础上，韩国学者 Sanghyun Woo 等人通过引入空间维度的注意力机制，进一步提升了网络的性能，称之为 CBAM[18]。

在损失函数领域，大量研究人员通过改善经典的 Softmax Loss，提升了训练性能。Liu Weiyang 等人通过增加类间样本的夹角，提出了基于余弦角约束的 L-Softmax 损失函数[74]，在 L-Softmax 损失函数的基础上，进一步增加了分类向量的模长约束，在人脸识别领域提出了将分类面投影到球面的 SphereFace[77]。Hao Wang 等人又进一步针对特征模长的约束，提出了 CosFace[78]，使得最后的分类只依靠余弦夹角。

1.3 本文主要研究工作

本文针对卷积神经网络的鲁棒性提高问题，在通道选择、特征过滤和防御对抗攻击三个方面分别提出了三个工作。

针对深度神经网络中噪声通道的存在而影响神经网络鲁棒性的问题，我们提出了基于通道选择的鲁棒性提高方法，该方法的目的是通过重新分配特征通道的权重来减少噪声通道的影响，我们希望噪声通道的权重小甚至为零。具体方法是首先通过我们提出的两阶段信息融合过程（见章节 4.1）将深度卷积特征压缩成一个向量来描述此深度卷积特征的信息，我们把该向量称作特征描述符，得到特征描述符后，我们通过一个感知机将描述符映射成可以对各个通道赋予新权重的权重向量，最后通过权重向量与深度卷积特征的相乘来进行对通道权重的重新分配。

针对深度卷积网络中有名的残差网络族的信息噪声问题，我们提出了基于特征过滤的鲁棒性提高方法。由于卷积神经网络对图像的表示是分布式表示[3]的，所以在网络的中间表示中会

存在大量的噪声信息，噪声信息对深度神经网络的危害是巨大的，会大大影响神经网络的鲁棒性，又由于残差网络家族中普遍存在跨层连接，会导致噪声信息被简单地传递到网络的各个层，基于此我们提出了一个即插即用的轻量级信息过滤模块，我们把它称为去噪编解码器，插入到跨层连接中，作为对跨层连接的提高。

针对最近被证明的对抗样本普遍存在的问题[103]，我们提出了基于对抗训练的鲁棒性提高方法。与原始图片相比，对抗样本是人类肉眼无法察觉的，只是在原图的基础上加上微小的扰动，训练良好的神经网络就会输出错误的结果。看似性能良好的卷积神经网络的鲁棒性是远远不够的，我们猜测的一个可能的原因是：对于一个训练样本，现在的技术无法让神经网络学习到该训练样本邻域内的信息。对此我们提出了一种基于生成式的对抗样本产生方法，用于模拟现有训练样本的邻域信息，作为一种数据增强手段，来提高神经网络的鲁棒性。

1.4 本文内容安排

本文第一章针对提高卷积神经网络鲁棒性问题，阐述了问题研究的背景与意义以及当前国内外研究情况，同时也简单总结了本文对提高神经网络鲁棒性提出的三个工作。第二章介绍了卷积神经网络在图像分类中的发展，包括基础的卷积计算、网络的构成、什么是分布式表示、卷积神经网络的结构发展和损失函数发展，作为后续讨论的基础。第三章针对对抗样本问题，介绍了对抗样本的发现以及目前流行的白盒与黑盒攻击方法。第四章介绍了我们提出的基于通道选择的鲁棒性提高方法，用于改善卷积神经网络中的噪声通道问题。第五章介绍了我们提出的基于特征过滤的鲁棒性提高方法，用于改善残差网络族中跨层连接对噪声信息的传递问题。第六章介绍了基于对抗训练的鲁棒性提高方法，我们提出的生成式方法增强了训练数据，以此来提高卷积神经网络的鲁棒性。最后第七章是对本文的总结以及未来的展望。

第二章 卷积神经网络在图像分类中的发展

自从 AlexNet 夺得 ImageNet 大规模图像分类大赛冠军以来[1]，深度学习得到飞速发展。为了提高神经网络的鲁棒性，新的网络结构层出不穷。本章将先介绍神经网络基础，包括计算单元，网络组成以及神经网络对输入的表示形式，再者将会介绍当今流行的几种网络结构，最后会从损失函数的角度介绍最新的技术。

2.1 卷积神经网络基础与特征表示形式

在本节将介绍卷积神经网络的基本组成部件，以及这些部件的工作原理，作为本文所讨论问题的基础。首先将介绍的是卷积计算的方式以及目前为止研究人员提出的有效的其他计算方式，比如组卷积[71][43][65]、空洞卷积[67][68]、按通道分组的卷积[64][66]。然后将会介绍组成网络的各个网络层，包括卷积层、激活层、池化层等，介绍完基本部件之后，将会介绍如何使用这些部件组成一个神经网络。最后，将会介绍神经网络对输入的表示形式，即分布式表示[3]，以及其存在的问题。

2.1.1 卷积计算方式

本节将会介绍卷积的计算方式，会从计算形式以及计算复杂度两个方面进行阐述。计算形式说明了不同卷积形式的工作原理，计算复杂度说明了各个卷积形式的计算方式会带来多大的复杂度开销。

2.1.1.1 普通卷积

普通卷积是使用率最为广泛也是最早提出的卷积计算方式，包含输入通道数c_{in}、卷积核个数c_{out}（也是输出通道数）、卷积核大小k以及卷积移动的步长大小s。假设再给出输入特征的大小为h_{int}，若不考虑填充，据此就能定义出输出特征的大小h_{out}：

$$h_{out} = \frac{h_{in} - k}{s} + 1$$

如图 2.1 所示，卷积核将在输入特征上以滑动窗口的形式进行加权计算，得到输出特征的组成单元，这些组成单元拼接得到输出特征。

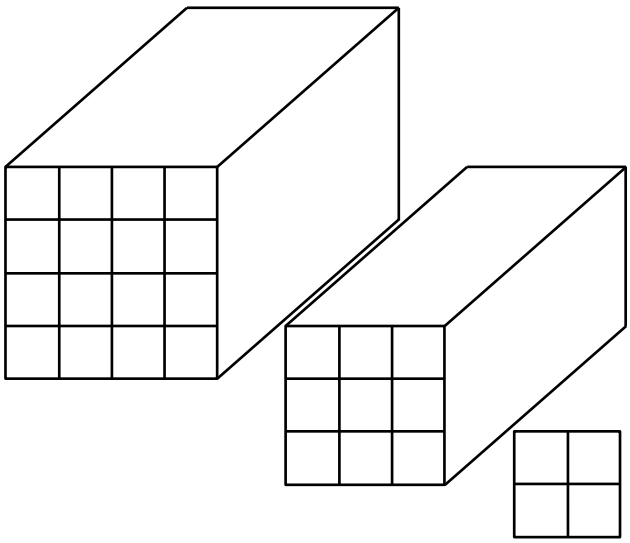

图 2.1 普通卷积计算方式

图 2.1 给出了输入特征大小h_{in}为 4，输入通道数为c_{in}，卷积核大小k为 3，步长s为 1，卷积核个数c_{out}为 1 的情况，因此会得到输出特征大小h_{out}为 2，输出通道数为 1 的结果。一个输出特征的组成单元是由一个卷积核与对应位置的输入特征进行加权求和得到的，因此获取一个特征组成单元所需要的计算量是$k^2 \cdot c_{in}$，由于输出特征大小是h_{out}，包含了${h_{out}}^2$个组成单元，因此获得单通道输出特征所需要的计算量为$k^2 \cdot c_{in} \cdot {h_{out}}^2$，当卷积核个数$c_{out}$不为 1（绝大多数情况不为 1）时，将获得$c_{out}$个通道的输出特征，因此其计算量为：

$$k^2 \cdot c_{in} \cdot {h_{out}}^2 \cdot c_{out}$$

由于大量的线性计算，普通卷积会带来巨大的信息冗余[71]，有人通过改变训练策略的方式来降低信息冗余量[73][72]，也有人改变卷积的计算方式，比如组卷机、空洞卷积等。

2.1.1.2 组卷积

组卷积是指在普通卷积的基础上对输入特征进行分组，因此卷积核的通道数不再等于输入特征的通道数c_{in}，假设分组数为g，那么卷积核的通道数为${c_{in}}'$：

$${c_{in}}' = \frac{c_{in}}{g}$$

图 2.2 给出了组卷积的计算方式，每一组输入通道（而不是像普通卷积一样整个通道）将配一个卷积核进行加权计算，获得g个通道的输出特征，图 2.2 中的分组数g为 2。与普通的卷积相比，我们考虑计算复杂度，在相同的计算量下，组卷积能获得g倍的输出通道数，那么当要求的输出通道数相同时，普通卷积的计算复杂度将是组卷积计算复杂度的g倍，因此再给定

输入通道数c_{in}、卷积核个数c_{out}、卷积核大小k以及卷积移动的步长大小s，定义出输出特征大小h_{out}后，组卷积的计算复杂度是：

$$\frac{k^2 \cdot c_{in} \cdot {h_{out}}^2 \cdot c_{out}}{g}$$

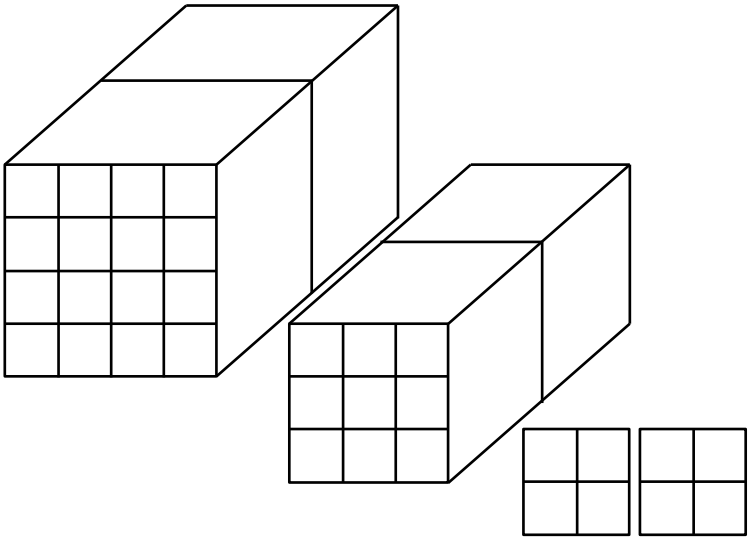

图 2.2 组卷积计算方式

组卷积大大降低了卷积计算的计算开销，同时也降低了参数的冗余程度[71]，但是如此分组计算的方式导致不同组的通道不能进行交互，zhang 等人提出了 Shufflenets[65]进行了改进。

2.1.1.3 空洞卷积

空洞卷积的目的是在做到相同感受野的同时减少计算复杂度，通过引入抑制比例r来从空间维度减少卷积核的参数，以此来减少计算量。抑制比例r的含义是每r个卷积核参数中只有一个参数有效，其他参数被抑制，如图 2.3 所示，给出了抑制比例r为 2 的情况，卷积核参数呈现出有效-无效的循环排列效果，因此普通卷积的计算复杂度是空洞卷积的r倍，则给定输入通道数c_{in}、卷积核个数c_{out}、卷积核大小k以及卷积移动的步长s后，定义出输出特征大小h_{out}，空洞卷积的计算复杂度是：

$$\frac{k^2 \cdot c_{in} \cdot {h_{out}}^2 \cdot c_{out}}{r}$$

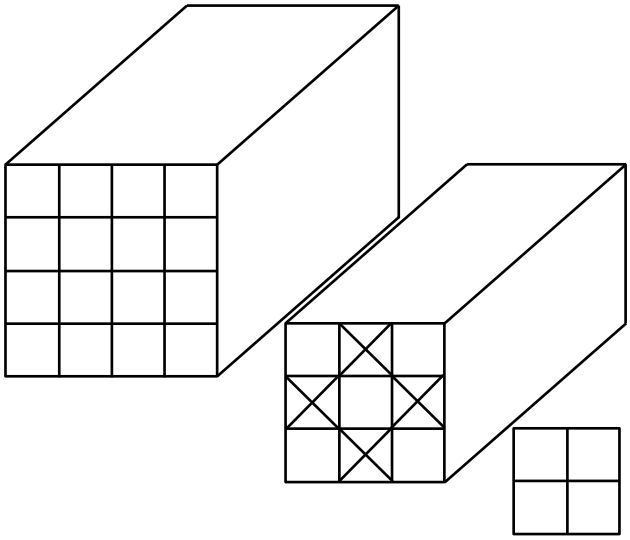

图 2.3 空洞卷积计算方式

空洞卷积被大量应用于图像分割任务，比如 DeepLab 系列[67][68]。由于图像分割是密集的像素级别预测任务，感受野的大小及其重要，因此空洞卷积表现出了良好的效果，因为当要求相同的计算复杂度时，空洞卷的感受野是普通卷积的r倍。

2.1.1.4 按通道分组的卷积

按通道分组的卷积是组卷积的极端，即分组数g等于输入通道数c_{in}，因此当给定卷积核个数c_{out}、卷积核大小k以及卷积移动的步长s后，定义出输出特征大小h_{out}，按通道分组的卷积的计算复杂度是：

$$\frac{k^2 \cdot c_{in} \cdot {h_{out}}^2 \cdot c_{out}}{c_{in}} = k^2 \cdot {h_{out}}^2 \cdot c_{out}$$

按通道分组的卷积被广泛用于小模型的设计[64][66]，因为小模型是为了部署在移动设备上而产生的，这些移动设备往往计算资源有限，大模型的巨大计算复杂度往往不能够做到对输入的实时反馈，因此需要计算量极小的小模型。

在近期，Pravendra Singh 等人研究了组卷积以及按通道分组的卷积，提出了计算效率更高的 HetConv[63]。与 Xception[66]以及 Mobile-Net[64]相比，减少了计算步骤，增加了计算效率。在后面的第四章中，我们提出了按通道分组的全连接层，用于进行通道信息融合。

2.1.2 卷积神经网络的基本构成

卷积神经网络由各种网络层组成，这些网络层协同作用，形成一系列性能不一的卷积网络。

在本章节，将讨论各种网络层以及他们的组合形式。

2.1.2.1 网络层

卷积神经网络中最基本的组成单元就是卷积层，由输入通道数c_{in}，输出通道数c_{out}，边界填充数p以及卷积类型组成，以普通卷积为例，给定卷积核大小k与步长s后，假设卷积层输入的特征大小是h_{int}，那么当双边填充时输出特征的大小h_{out}为：

$$h_{out} = \frac{h_{in} + 2p - k}{s} + 1$$

当单边填充时输出特征的大小h_{out}为：

$$h_{out} = \frac{h_{in} + p - k}{s} + 1$$

除了卷积层，与卷积层计算最相似的就是池化层，包括平均池化层与最大池化层，其中平均池化层可以理解为将通道分组的卷积层中的卷积核参数全部设为$1/k^2$，而最大池化层的计算方式也和通道分组的卷积层类似，只是卷积核参数只在输入特征的最大值处是 1，其余为 0。这两种池化层有特殊的情况，即全局平均池化与全局最大池化，也就是将滑动窗口的大小设置成与输入特征大小相同，如此就能将输入特征大小降为 1 维，这往往是为后面紧接着的全连接层服务。

为了增加非线性能力，往往还有激活层，包括 Sigmoid 层，ReLU 层等。其中 Sigmoid 函数是：

$$S(x) = \frac{1}{1 + e^{-x}}$$

ReLU 函数是：

$$R(x) = max(0, x) = \begin{cases} 0, x < 0 \\ x, x \geq 0 \end{cases}$$

2.1.2.2 神经网络组成

卷积神经网络的组成就是不同层之间的组合，上一层的输出作为本层的输入，本层的输出作为下一层的输入，可以形式化表示为函数的复合。假设输入特征为x，经过卷积层F以及激活层A后，将网络层对x的转化看成函数运算，那么输出为$A\big(F(x)\big)$，如此不断复合就形成了卷积神经网络。

2.1.3 分布式表示

卷积神经网络对输入图像的表示形式是分布式表示的，首次由 Bengio 等人提出[3]。图 2.4 展示了对帆船的分布式表示特点，卷积神经网络通过激活帆船的不同属性来表示“帆船”这一概念。

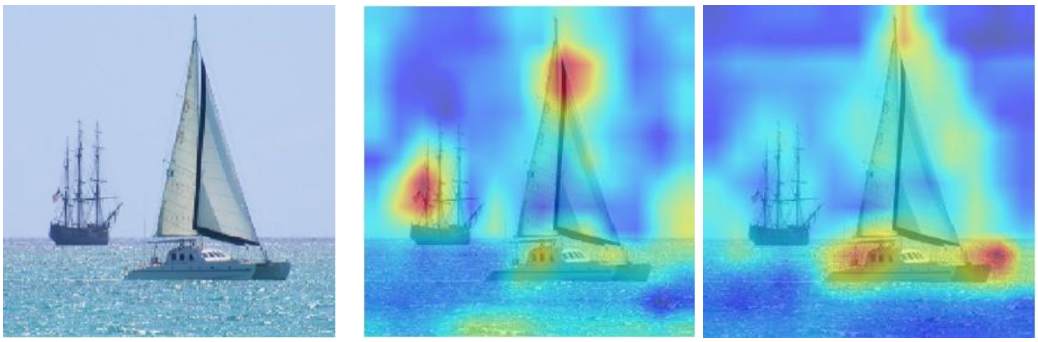

图 2.4 分布式表示

但是分布式表示特点也会带来大量的噪声激活，如图 2.5 所示，这些对神经网络的鲁棒性是巨大的损害，是亟待解决的问题。

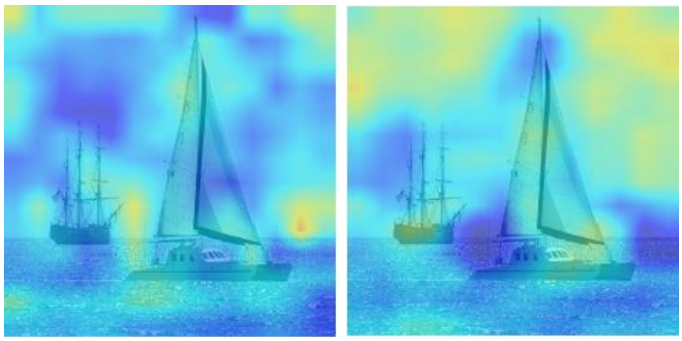

图 2.5 分布式表示的缺点

2.2 卷积神经网络结构发展

在众多研究人员提出的结构中，有一些神经网络的结构脱颖而出，成为当今流行的结构，本章节将讨论其中几种典型结构，也是本文所提出方法使用的几种神经网络，作为后续讨论的基础。

2.2.1 AlexNet

AlexNet 是由 Alex Krizhevsky 于 2012 年提出的[1]，其获得了同年 ImageNet 大规模图像分类大赛冠军。AlexNet 中首次使用了 ReLU 激活函数，同时也使用了 Dropout 的训练技巧，利用 GPU，将网络的宽度以及深度扩大，这是之前的研究者没有做到的。

AlexNet 的第一层是一个大卷积层，卷积核的大小是 11，同时后续迭代了大量的最大池化

层以及小卷积层。最后的分类层是三个全连接层，分别是 4096 维、4096 维以及 1000 维。

2.2.2 VGG

VGG 网络是由牛津大学 Visual Geometry Group 提出[2]，获得了 2014 年 ILSVRC 大赛的第二名。VGG 网络首次大规模使用了小卷积核和小池化核，这使得同参数情况下可以把网络做得更深，为后续的研究打下了良好的基础，提供了有效的经验实践。值得提出的是，VGG 网络的分类层是三个全连接层，包含了整个网络大部分的参数量，使得网络变得笨重，具体表现是巨大的参数量难以储存。

2.2.3 ResNet

ResNet 由 He Kaiming 等人提出[4]，其将跨层连接应用于深度神经网络中，利用巧妙的恒等映射大大降低了神经网络的训练难度。ResNet 由不断重复的残差学习块组成，这些残差学习块有两个版本，其中一种是由两个卷积核大小为 3 的卷积层以及跨层连接组成，另外一个版本是由卷积核大小为 1、卷积核大小为 3、卷积核大小为 1 的三层卷积层组成，并且配上跨层连接。

在原文中，作者阐述了将网络做到上千层的情况，在当今流行的版本中，152 层的 ResNet 也广为使用，其中也有浅层的 18 层网络。

2.2.4 ResNeXt

ResNeXt 网络是 ResNet 网络的扩展，是残差网络家族中的一员，由 Xie 等人提出[43]。在增加了残差学习块中卷积层的宽度后，对其进行分组，将组卷积技术应用于 ResNet，形成了升级版本的 ResNeXt。ResNeXt 的优点是能做到与 ResNet 的计算复杂度相同的情况下，性能远远超过 ResNet。

2.3 损失函数

为了提高网络的鲁棒性，除了神经网络结构的迭代更新，损失函数方面也得到了很多发展。除了最经典的 Softmax 损失函数以外，有人提出了基于角度正则的 Large-Margin Softmax 损失函数[74]，也有进行分类层参数模长约束的损失函数被提出[77]，最近有学者在这些基础上提出了对特征x模长进行约束的损失函数[78]。

2.3.1 Softmax 损失函数

Softmax 损失函数是最常用的损失函数，其中 Softmax 损失函数的形式是：

$$s = -log\left(\frac{e^{w_y \cdot x}}{\sum_{x,y} e^{w_y \cdot x}}\right)$$

其中w_y表示属于第y类的全连接层的分类向量，y是以x为特征的样本的标签。虽然 Softmax

广为使用，但是其存在一个问题，就是对训练数据类别不平衡敏感，导致训练的神经网络倾向于输出样本数目多的类别的类，从而使得神经网络鲁棒性下降，具体的我们将全连接层的内积打开：

$$s = -log\left(\frac{e^{\|w_y\|\cdot\|x\|\cdot cos\langle w_y,x\rangle}}{\sum_{x,y} e^{\|w_y\|\cdot\|x\|\cdot cos\langle w_y,x\rangle}}\right)$$

由于数据的驱动，训练得到的全连接层的参数向量w_y的模长以及与样本的角度$cos\langle w_y,x\rangle$会呈现出一个趋势，即样本多的类别的w_y的模长大，且角度小，从而导致$cos\langle w_y,x\rangle$值大，如此将会导致错分类，即将样本数少的类分成样本数目多的类。

2.3.2 L-Softmax 损失函数

L-Softmax 通过对余弦角增加约束[74]，改进了 Softmax 损失函数，具体形式为：

$$s = -log\left(\frac{e^{\|w_{y^*}\|\cdot\|x\|\cdot cos\left(m\cdot\langle w_{y^*},x\rangle\right)}}{e^{\|w_{y^*}\|\cdot\|x\|\cdot cos\left(m\cdot\langle w_{y^*},x\rangle\right)} + \sum_{x,y\neq y^*} e^{\|w_y\|\cdot\|x\|\cdot cos\langle w_y,x\rangle}}\right)$$

通过超参数m人为增大全连接层参数w_{y^*}与样本x的角度，从而增大不同类别样本之间的角度，使得不同类别的样本分得更开，达到提高网络鲁棒性的目的。

2.3.3 SphereFace

Weiyang Liu 等人在 L-Softmax 的基础上，通过对全连接层的模长进行约束，即将所有的$\|w_y\|$设为 1，使得分类面在球面上，在人脸识别领域提出了 SphereFace[77]：

$$s = -log\left(\frac{e^{\|x\|\cdot cos\left(m\cdot\langle w_{y^*},x\rangle\right)}}{e^{\|x\|\cdot cos\left(m\cdot\langle w_{y^*},x\rangle\right)} + \sum_{x,y\neq y^*} e^{\|x\|\cdot cos\langle w_y,x\rangle}}\right)$$

2.3.4 CosFace

在 SphereFace 的基础上，Hao Wang 等人对特征的模长进行了约束[78]，也令其为 1，最后使得分类只依靠余弦角的大小，其形式是：

$$s = -log\left(\frac{e^{a\cdot\left(cos\langle w_{y^*},x\rangle - m\right)}}{e^{a\cdot\left(cos\langle w_{y^*},x\rangle - m\right)} + \sum_{x,y\neq y^*} e^{a\cdot cos\langle w_y,x\rangle}}\right)$$

值得注意的是，与 L-Softmax 相比，对余弦角的约束从乘性变成了加性约束，同时又为了方便训练，通过超参数a对信号进行了放大。

2.4 本章小结

本章先介绍了卷积神经网络的基础，其中卷积计算方式以及分布式表示特点是影响卷积神

经网络鲁棒性的重点。紧接着是卷积神经网络的结构发展，挑选了几种典型结构进行介绍，最后是损失函数，从损失函数的层面介绍了影响卷积神经网络鲁棒性的原因。

第三章 卷积神经网络的对抗样本

深度神经网络（Deep neural networks）在机器学习领域的很多方面取得了重大进展，比如图像分类[1][2]、对象检测[5][6]、语音识别[7]、语言翻译[8]、语音合成[9]。深度学习与传统机器学习不同，只需要少量的甚至不需要手工设计特征。在大数据支持下，利用高效率的计算资源，深度神经网络可以从原始输入数据中直接提取复杂的特征[10]。

基于深度学习的技术已经支持众多应用产生。不管是 IT 行业的公司，还是汽车行业的公司，比如 Google、Telsa、Mercedes 和 Uber，都在进行自动驾驶汽车的研发，这需要大量的深度学习技术，比如目标检测、强化学习等。在众多应用中，人脸识别系统作为一种成熟的技术已经在我国的自动取款机上得到应用[11]，苹果公司也使用了人脸认证功能来解锁个人手机。异常检测技术也被应用于各种安全领域，其是建立在查找高级语义特征的基础上的[20][24][25][27]。

尽管深度学习技术已经在众多应用中获得良好的表现，但在这些应用中，其安全因素是至关重要的。最近的研究发现，深度学习技术很容易被稍作修改的输入信息所影响。这些样本可以很容易地欺骗一个性能良好的深度学习模型，但是人类却几乎不能发现这些异常的攻击样本。Szegedy 等人首次在图像上产生非常微小的扰动，就能使神经网络输出错误的结果[103]。这些可以使得神经网络错误分类但是人类无法察觉的样本被称为“对抗样本”。

虽然基于深度学习的技术应用已经可以被大量部署在现实世界中，但是最近的研究表明，对抗样本可以应用于现实世界，因此，提高深度神经网络的鲁棒性至关重要。例如，攻击者可以通过轻微改变交通识别标志的外观[37][75]，或将道路识别系统中的行人模糊[95]来构造对抗样本，对自动驾驶技术或者成熟的车辆识别技术进行攻击，但是人类无法察觉。同样的对于语音识别技术，攻击者可以针对自动语音识别（ASR）模型和语音控制系统（VCS）[28][94]生成对抗样本来进行攻击。

深度学习被认为是一个黑盒技术，即我们都知道它表现良好，但对原因的了解有限[29][69]。许多研究已经被提出用来解释深度神经网络[70][84][105][104]工作的原理，但是本质解释仍未出现。有趣的是，通过观察对抗样本，我们可以深入了解神经网络的语义表示[89]并找到决策边界，这反过来有助于提高神经网络的鲁棒性[98]并增强可解释性[33]。

本章对“对抗样本”的生成方法、应用以及相应的算法进行了总结，讨论了不同方法产生的对抗样本的特点。深度学习在计算机视觉领域有着非常优异的表现，因此大多数对抗样本都是针对计算机视觉模型产生的。本文主要讨论图像分类任务中的对抗样本。

3.1 神经网络中对抗样本的发现

本节介绍神经网络中一个非常有趣的现象：对于一张改变非常微小的图片，人类肉眼几乎无法察觉，但是神经网络会输出与原图不同的结果。其中图像分类任务的对抗样本一般这样产生：使用训练好的图像分类器，该分类器就是被攻击的模型，输入图像来获得该图像的预测标签，对抗样本是相对于原始输入图像来说的一个清晰图像，在原始图像的基础上加上微小的扰动，通常人类几乎无法识别，这种微小的扰动误导了分类器，产生了错误的图像分类结果。

可以形式化表示为给定一个经过训练的深度学习模型f，一个原始输入数据样本x，如何生成一个对抗样本x'，通常可以描述为一个约束优化问题：

$$\min_{x'} \|x - x'\|$$

其中要求：

$$f(x') = l'$$
$$f(x) = l$$
$$l \neq l'$$

其中l和l'表示x和x'的输出类别。

该优化问题在使得模型对预测进行错误分类的情况下，对输入数据进行限制，最小化了扰动，以此达到人类肉眼无法识别的目的。在不同假设下研究人员提出了这个优化问题的许多变体。如果扰动足够小，人类无法察觉，因此扰动可以被视为一个约束，目标函数可以是对输出结果进行限制，因此此变体问题可以通过以下方式描述：

$$\min_{x'} J_\theta(f(x'), l')$$

其中要求：

$$\|\delta\| < \epsilon$$
$$f(x) = l$$
$$l \neq l'$$

这里原始样本与对抗样本的差别被定义为$\delta \stackrel{\text{def}}{=} x - x'$，其中$\epsilon$是设定的超参数，意义为人类能够肉眼分辨的程度，值越小越难分辨，J_θ是参数为θ的神经网络的损失函数。

Szegedy 等人在 2014 年首次介绍了深度神经网络的对抗样本[103]。他们使用 L-BFGS 方法生成了对抗样本：

$$\min_{x'} c\|\delta\| + J(x', l')$$

选定合适的常数c，L- BFGS 通过搜索来确定x'的近似值。作者指出，生成的对抗样本也可以用于攻击其他的模型，即对抗样本的可转移性。他们认为，对抗样本存在的原因是模型无法学习到训练数据的完整空间信息。有学者也使用了 L-BFGS 攻击，但是实现了一个高效率的二

进制搜索方法，以确定最优值[102]。

图 3.1 给出了一个对抗样本的直观例子，其中左图是原图片，标签是阿尔卑斯山，中间的图是攻击算法产生的扰动，通过相加的方式来改变原图以产生右图的对抗样本，被分类器分成了“狗”这一概念[45]。

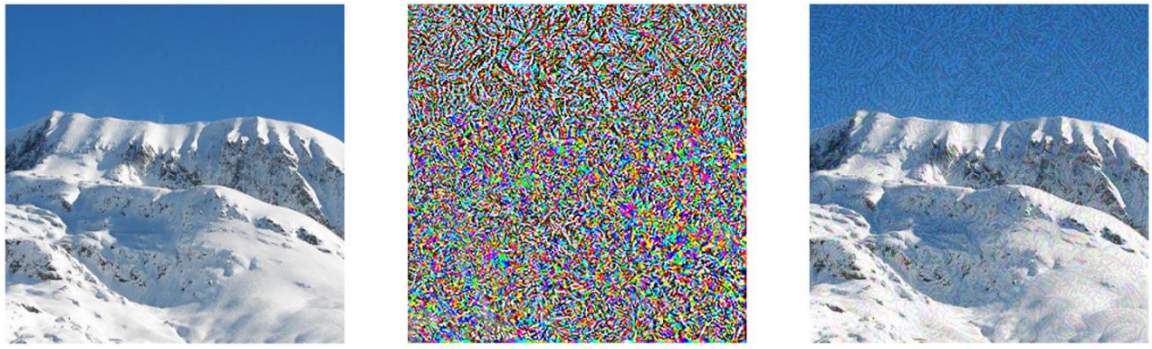

图 3.1 对抗样本展示

3.2 白盒攻击

在传统机器学习的攻击与防御领域，例如垃圾邮件过滤、入侵检测、人脸认证、欺诈检测等[30]，研究人员已经针对某些应用进行了大量研究。其中，经常对垃圾邮件进行微小改动以避免被发现[31][32][88]。Dalvi 等人首先提出了对抗样本，并将这个问题定义为博弈问题[33]，对抗样本的攻防成为一个迭代性问题。一种基于梯度的对抗样本产生方法首先被 Biggio 等人提出，用来攻击简单的线性分类器、支持向量机和神经网络[35]，他们还讨论了几种防御机制，并提出了提高模型鲁棒性的方法[108]。

在传统的机器学习中，攻击和防御方法都非常重视特征选择（甚至上一步：数据集），而很少关注对抗样本是否可以被人类察觉的问题，即扰动的大小不受限。但是在深度学习领域，对抗样本的扰动大小是受到限制的，过大的扰动会导致对抗样本与原始样本的差距较大，从而可以轻易被人类肉眼感知。与黑盒攻击相比，白盒攻击的方法允许攻击者使用被攻击模型的表示形式、参数，训练模型使用的方法以及数据集也能够被攻击者使用。本节回顾了近年来深度学习领域的白盒攻击方法。

3.2.1 快速梯度符号法 (FGSM)和其变种

因为 L-BFGS 攻击使用了计算复杂度颇高的搜索方法来确定最优值，Goodfellow 等人提出了一种快速生成对抗样本的方法，称为快速梯度符号法[48]。他们只在每个像素处沿着梯度符号的方向进行一次梯度更新，其产生的扰动可以表示为：

$$\delta = \epsilon \cdot sign\left(\nabla_x J_\theta(x, l)\right)$$

其中ϵ是扰动的缩放大小因子，生成的对抗样本x'可以由$x' = x + \delta$计算得到。这种扰动可

以简单地用反向传播来计算。

他们认为尽管线性计算加快了训练速度，但是对抗样本产生的原因在于深度神经网络中存在大量的线性单元，而且正则化方法在深度神经网络中的应用，如 Dropout、预训练等都不能提高网络的鲁棒性。

随后有人提出了一种新的方法，称为快速梯度值法，用原始梯度代替梯度符号：

$$\delta = \epsilon \cdot \nabla_x J_\theta(x, l)$$

快速梯度值法不受每个像素的约束，可以生成局部差异较大的图像。

后续又有人提出了 OTCM 方法[76]，通过最大化被攻击目标类的概率来进行指定目标的攻击：

$$x' = x - \epsilon \cdot sign\left(\nabla_x J_\theta(x, l')\right)$$

作者将这种攻击称为单步目标类方法。

有人发现基于 FGSM 的对抗训练得到的模型对比于梯度不可见的黑盒攻击来说更能抵抗白盒攻击[100]，他们提出了一种新的攻击方法，RAND-FGSM，用多步更新对抗样本的方式来产生攻击性能更好的对抗样本：

$$x_{tmp} = x + \alpha \cdot sign\left(\boldsymbol{\mathcal{N}}(\mathbf{0}^d, \boldsymbol{I}^d)\right),$$

$$x' = x_{tmp} + (\epsilon - \alpha) \cdot sign\left(\nabla_{x_{tmp}} J_\theta\left(x_{tmp}, l\right)\right)$$

这里的α和ϵ是超参数，其中$\epsilon > \alpha$。

3.2.2 基本迭代法（BIM）和最小近似类迭代法（ILLC）

先前的研究假设对抗样本可以直接输入到深度神经网络中，然而在许多应用中，人们只能通过设备（如摄像机、传感器）传递数据。Kurakin 等人将对抗样本应用于现实世界[75]，通过多次迭代来进行优化，每次迭代对样本产生较小的变化，扩展了快速梯度符号法。在每次迭代中，它们都会剪裁像素值，以避免在每个像素上进行较大的更改：

$$Clip_{x,\xi}\{x'\} = min\{255, x + \xi, max\{0, x - \epsilon, x'\}\}$$

其中$Clip_{x,\xi}\{x'\}$是每个迭代中由ξ限制的剪裁值。对抗样本是在多次迭代中产生的：

$$x_0 = x,$$
$$x_{n+1} = Clip_{x,\xi}\{x_n + \epsilon \cdot sign(\nabla_x J_\theta(x_n, y))\}$$

作者把这种方法称为基本迭代法。

为了进一步实现有目标的攻击，他们选择了原始样本置信度最低的类，并且通过最大化交叉熵损失来优化，此方法称为最小近似类迭代法：

$$x_0 = x,$$
$$y_{LL} = argmin_y\{p(y|x)\},$$
$$x_{n+1} = Clip_{x,\epsilon}\{x_n - \epsilon \cdot sign(\nabla_x J_\theta(x_n, y_{LL}))\}$$

他们成功地用一个手机摄像头拍摄的图像欺骗了神经网络，该图像就是通过基本迭代法和最小近似类迭代法产生的。他们还发现，快速梯度符号法对光照变换具有鲁棒性，但是迭代法不能抵抗光照变换。

3.2.3 基于雅可比的显著图攻击（JSMA）

Papernot 等人根据样本的显著性图，提出了一个有效的对抗样本产生方法，称为基于雅可比的显著图攻击方法[101]。他们首先计算给定样本x的雅可比矩阵：

$$J_f(x) = \frac{\partial f(x)}{\partial x} = \left[\frac{\partial f_j(x)}{\partial x_i}\right]_{i\times j}$$

通过这种方式，他们得到输入x中可以显著改变神经网络的输出的区域，即显著性图，根据显著性图来设计一个小扰动使得模型输出产生巨大的变化，做到小部分特征的变化可以欺骗整个神经网络的效果。

根据显著性图，攻击者可以发现原始样本中对网络输出影响最大的区域，这些区域往往是极小的，原文中通过修改每个样本 4.02%的输入特性，可以获得 97%的攻击成功率。然而，由于计算量巨大，该方法运行速度很慢。

3.2.4 DeepFool

Moosavi-Dezfooli 等人提出了 DeepFool[97]算法来找到原始输入的对抗样本，该对抗样本到决策边界的距离是最短的。为了克服高维非线性情况，作者采用线性逼近的方法进行迭代攻击。以分类为例子，他们发现分类的最小扰动是到分类超平面$\mathcal{F} = \{x: w^T x + b = 0\}$（如图 3.2）的距离，一个分类器$f$的扰动可以是：

$$\delta^*(x) = -\frac{f(x)}{\|w\|^2} w$$

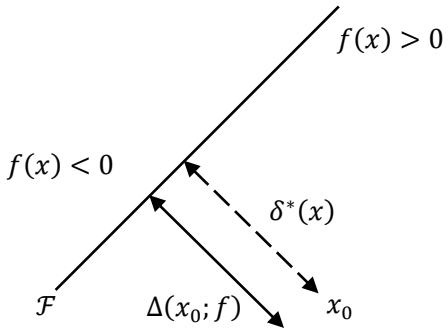

图 3.2 二分类器的对抗样本

如果f是一个二元可微分类器，则在迭代过程中，假设f在x_i周围线性化，则可以使用迭代法来逼近以寻找扰动的近似解，最小扰动计算如下：

$$\underset{\delta_i}{\operatorname{argmin}}\|\delta_i\|^2$$

其中要求：

$$f(x_i)+\nabla f(x_i)^T\delta_i=0$$

这个方法也可以通过查找最近的超平面扩展到多类问题中去。与 FGSM 和 JSMA 相比，DeepFool 提供的扰动更少，其中与 JSMA 相比，DeepFool 还降低了扰动的强度，也不需要选择特定的特征。

3.2.5 CPPN EA Fool

Nguyen 等人发现了一种新的攻击方法，即模式生成式网络编码器（CPPN），其中对抗样本可以被深度神经网络以高置信度的结果错分类，但是人类无法识别其类别[99]。我们将这种攻击归类为目标性攻击。

他们使用进化算法（EA）来产生对抗样本。作者首先用两种不同的方法对图像进行编码：直接编码（灰度或 HSV 值）和间接编码（模式生成式生成网络）。然后在每次迭代中，像一般的进化算法一样，选择一个随机模式特征进行变异，如果新的模式特征具有更高的攻击成功率，则将它作为当前模式特征。他们发现对于许多对抗样本，CPPN 可以像 JSMA 一样容易地定位重要特征来改变深度神经网络的输出。

3.2.6 C&W Attack

Carlini 和 Wagner 提出了一种新的攻击方法[44]，根据他们的进一步研究[36][38]，Carlini 和 Wagner 的攻击对大多数现有的对抗样本防御方法是有效的。作者依据优化的难易程度对目标函数进行了一些修改，定义了一个新的目标函数g，要求：

$$\min_{\delta}\ \|\delta\|_p+c\cdot g(x+\delta)$$

其中要求：

$$x+\delta\in[0,1]^n$$

这里$g(x+\delta)>0$当且仅当$f(x')\neq l'$。原文中作者列举了七个目标函数g的例子，通过实验评估，其中一个有效的目标函数是：

$$g(x')=max\left(\max_{l\neq l'}(Z(x')_l)-Z(x')_{l'},0\right)$$

这里Z就是 Softmax 函数，这样可以更好地优化距离和罚分条件。

其次，对于 L-BFGS 攻击方法，作者没有使用邻域约束的 L-BFGS 来寻找最小扰动，而是

引入了一个新的变量w来避免由像素值取值范围带来的约束，由w定义的扰动是：

$$\delta = \frac{1}{2}(tanh(w) + 1) - x$$

一般的深度学习优化方法（如 Adam 和 SGD）可以被用来产生对抗样本，通过二进制搜索找到最优常数c。他们发现如果$\|\delta\|_p$和$g(x+\delta)$的梯度不在同一尺度上，那么在梯度搜索的过程中很难找到合适的常数c来得到扰动的最优结果，由于这个原因，他们提出的方法并没有找到对抗样本的最优解，只能找到近似解。

第三，讨论了扰动的三种距离测度：基于l_0、l_2和l_∞范数的距离。基于距离度量，提出了三种攻击：l_0攻击、l_2攻击和l_∞攻击。其中l_2攻击可以被描述为：

$$\min_w \left\|\frac{1}{2}(\tanh(w)+1)\right\|_2 + c \cdot g\left(\frac{1}{2}\tanh(w)+1\right)$$

3.2.7 Universal Perturbation

利用他们在 DeepFool 的工作，Moosavi Dezfooli 等人提出了一种新的对抗攻击[96]方法。他们提出的问题是找到一个满足下式的扰动：

$$\|\delta\|_p \le \epsilon$$
$$\mathcal{P}\big(f(x') \neq f(x)\big) \ge 1 - \tau$$

ϵ限制了扰动的大小，τ控制了所有对抗样本的失效率。

对于每次迭代，他们使用 DeepFool 方法针对每个输入数据获得最小扰动，并将单个样本的扰动更新为总扰动δ，实验发现在大多数样本被δ修改之前，这个循环不会停止。

3.2.8 特征攻击

Sabour 等人通过最大化神经网络中间层而不是输出层的表示距离来进行攻击[106]。把这样的攻击称为特征攻击。问题可以描述为以下形式：

$$\max_{x'} \|\phi_k(x) - \phi_k(x')\|$$

其中要求：

$$\|x - x'\|_\infty < \epsilon$$

这里的ϕ_k指的是神经网络的第k层输出。

3.2.9 Hot/Cold

Rozza 等人提出了一种 Hot/Cold 方法，用于对单个图像产生多个对抗样本[107]，他们认为只要不易察觉，就应该允许小的图像变换。他们定义了一种新的测量方法，即 Psychometric Perceptual Adversarial Similarity Score (PASS)，用来测量显著区域相似性。Hot/Cold 忽略了基于像素的不明显差异，并用 PASS 替换了广泛使用的l_p范数。PASS 包括两个阶段：1）将生成图

像与原始图像对齐；2）测量对齐图像与原始图像之间的相似性。

设$\phi(x', x)$为从对抗样本x'到原始样本x的变换。采用结构相似性（SSIM）指数[42]来衡量图像的显著性差异。Andras Rozsa 等人[107]利用 SSIM 并定义了一个新的衡量指标，即区域 SSIM 指数（RSSIM）：

$$RSSIM(x_{i,j}, x'_{i,j}) = L(x_{i,j}, x'_{i,j})^{\alpha} C(x_{i,j}, x'_{i,j})^{\beta} S(x_{i,j}, x'_{i,j})^{\gamma}$$

其中α是亮度$L(x_{i,j}, x'_{i,j})$的重要程度因子，β是对比度$C(x_{i,j}, x'_{i,j})$的重要程度因子，γ是结构$S(x_{i,j}, x'_{i,j})$的重要程度因子。那么 SSIM 可以被如下计算：

$$SSIM(x_{i,j}, x'_{i,j}) = \frac{1}{n \times m} \sum_{i,j} RSSIM(x_{i,j}, x'_{i,j})$$

新的距离 PASS 的定义如下：

$$PASS(x, x') = SSIM(\phi(x', x), x)$$

从而，攻击问题可以在新的距离下被重新定义为：

$$min\ D(x, x')$$

其中要求：

$$f(x') = l'$$

这里的$D(x, x')$是原始样本与对抗样本的距离，可以是$1 - PASS(x, x')$或者$\|x - x'\|_p$。

为了生成一组不同的对抗样本，作者将目标类l'定义为 Hot 类，将原始类l定义为 Cold 类。在每次迭代中，对抗样本都会移向目标（Hot）类，同时远离原始（Cold）类。他们的结果表明，他们生成的对抗样本与 FGSM 相当，并且具有更强的多样性。

3.3 黑盒攻击

在深度学习中，白盒攻击方法允许攻击者使用被攻击网络的训练方法以及训练数据集，结构和参数对攻击者来说也是可见的，从而可以得到网络的梯度，攻击者利用训练的数据集或者需要攻击的数据集，来生成对抗样本。这在现实世界中几乎是不匹配的，因为我们一般很难得到一个被攻击系统的模型，以人脸识别例子来说明，我们一般只能得到我们输入的图像以及返回的结果，那么只利用输入和输出信息，不接触网络信息的前提下，如何进行攻击呢，这就是黑盒攻击研究的问题，在本节，将介绍几种基于黑盒的攻击方法。

3.3.1 Zeroth Order Optimization (ZOO)

与基于梯度的对抗样本生成方法不同，Chen 等人提出了一种基于零阶优化（ZOO）的攻击方法[41]。此攻击方法可以直接应用在黑盒攻击中，因为基于此方法攻击者不需要知道网络的反传梯度。受 Carlini 和 Wagner 等人[44]的启发，作者将 Carlini 和 Wagner 等人方法[44]中的函数g稍

作修改，去掉了 Softmax 函数，成为了一个新的类似 hinge 的损失函数：

$$g(x') = max\left(\max_{l \neq l'}(log[f(x')]_l) - log[f(x')]_{l'}, 0\right)$$

并且利用定义来估计梯度和 Hessian：

$$\frac{\partial f(x)}{\partial x} \approx \frac{f(x + he_i) - f(x - he_i)}{2h}$$

$$\frac{\partial^2 f(x)}{\partial x^2} \approx \frac{f(x + he_i) - 2f(x) + f(x - he_i)}{h^2}$$

其中e_i表示第i分量为1的标准基向量，h是一个小常数。

通过采用梯度和 Hessian 的估计，ZOO 不需要访问所攻击的深度学习模型就能获得近似的梯度信息，然而查询和估计梯度需要巨大的计算量。作者在现有优化算法的基础上，提出了类似于 ADAM 的算法，即 ZOO-ADAM，用于对抗样本更新。实验表明，ZOO 的攻击性能与 C&W 攻击相当。

3.3.2 Natural GAN

Zhao 等人利用生成性对抗网络（GANs）作为生成对抗样本的方法 [40]，这使得对抗样本对人类来说更加自然，将这种方法命名为 Natural GAN。作者首先在数据集上训练了一个 WGAN 模型，其中生成器G将随机噪声映射到图像域，他们还训练了一个逆变器I将输入的图像映射到噪声域，因此，对抗样本是通过最小化两个不同噪声的距离而产生的。对抗样本是使用生成器生成的：$x' = G(z)$，优化目标简单又直观：

$$\min_z \|z - I(x)\|$$

其中要求：

$$f(G(z)) \neq f(x)$$

生成器G和逆变器I都是为了使对抗样本显得自然，优化的目的是找到可以产生对抗样本的z。Natural GAN 是目前许多深度学习领域的通用框架，已经应用于图像分类、文本识别和机器翻译。由于 Natural GAN 不需要原始被攻击神经网络的梯度，因此可以应用于黑盒攻击。

3.3.3 基于模型的集成攻击

Liu 等人在 ImageNet 上对深度神经网络的攻击可转移性进行了研究，并针对有目标的对抗样本攻击提出了一种基于模型的集成攻击方法[86]。作者认为，与无目标的对抗样本攻击相比，有目标的对抗样本攻击更难通过模型进行转换。使用基于模型的集成攻击可以生成可转移的对抗样本来攻击不同的黑盒模型。

作者在众多训练好的深度神经网络上通过白盒方法生成了对抗样本，并在黑盒模型上进行了测试。基于模型的集成攻击可以被描述为以下优化问题：

$$\underset{x'}{argmin}\ -log\left(\sum_{i=1}^{k}\alpha_i J_i(x',l)\right)+\lambda\|x'-x\|$$

其中k是深度神经网络的数目，f_i代表每个网络，那么J_i代表每个网络的损失函数，α_i是每个神经网络损失函数的权重，$\sum_i^k \alpha_i = 1$。结果表明，基于模型的集成攻击能够产生具有攻击可转移性的对抗样本。

3.4 本章小结

在本章节中，介绍了几种典型的对抗样本产生方法，包括基于梯度的白盒攻击方法和模型信息未知的黑盒攻击方法。尽管某些方法已经被新的方法所取代，但是其产生对抗样本的原理值得深入研究，此外对抗样本存在的原因也值得研究，这可以有助于提高深度神经网络鲁棒性的研究。

第四章 基于通道选择的鲁棒性提高方法

由于深度卷积神经网络的分布式表示特性，会在中间特征中产生噪声通道，如何减少噪声通道的影响对提高卷积神经网络的鲁棒性来说显得至关重要，一个直接的方法就是进行通道选择。所谓通道选择就是对每个通道重新定义权重，我们希望噪声通道的权重越小越好，非噪声通道的权重越大越好。

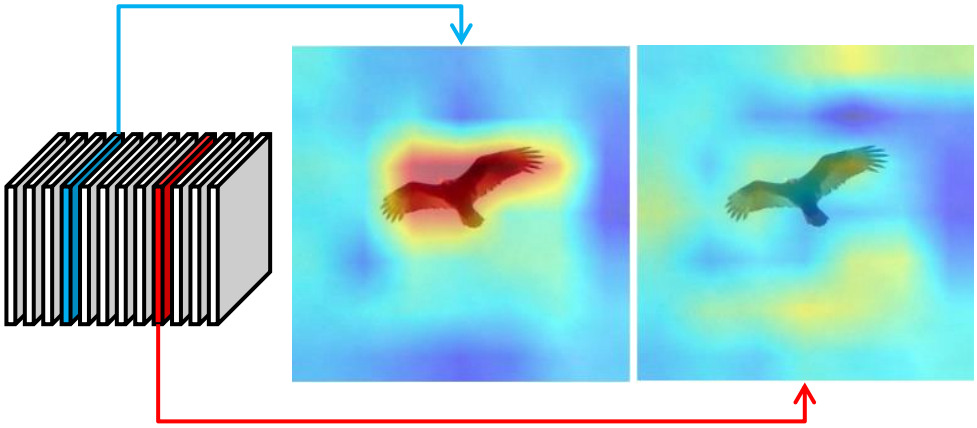

图 4.1 非噪声通道与噪声通道

如图 4.1 所示，蓝色线代表的非噪声通道与红色线代表的噪声通道都享有相同的权重值 1，在神经网络的图像分类中，会对后续的特征提取以及分类产生相同程度的正影响和负影响。为了改变这两种影响的程度，即降低噪声通道的影响，Hu Jie 等人提出了挤压与激励网络(SE-Net)[14]，通过引入 SE 模块显著提升网络性能。在每个 SE 模块中首先执行挤压操作，也就是全局平均池化（GAP），将输入信息聚合到一个通道描述子中，然后是激励操作，即两层的全连接网络，利用通道描述子来学习每个通道的权重。

在本章后续部分，首先会介绍 SE 的弊端，针对上述问题，我们提出了一种简单有效的两阶段信息融合过程。我们提出的两阶段方法包括“空间聚合”和“信息融合”。空间聚合旨在产生包含比全局池化更多局部信息的描述符，信息融合用来吸收空间聚合得到的描述符，并进一步返回一个表示能力强大的通道描述子，其维度和全局池化获得的一致，如此可被激励模块使用。最后我们会针对所提出的方法，进行实验分析以及总结。

4.1 基于信息融合的二阶段通道选择方法

4.1.1 SE 的弊端

尽管 SE 有令人信服的结果，但是基于全局平均池化的挤压操作是 SE 的弊端。因为全局平均池化模糊了局部特征，这对于识别不同通道是不利的。如图 4.2 所示，在没有局部信息提示下，在背景上具有噪声激活的通道也会被后面的激励模块误分配到高的权重，比如通道号为 1123 与 674 的两个瓶子响应图，经过 GAP 的挤压操作后有相同的激活值（0.023），将其作为后续的激励操作的输入后，导致激励操作无法准确区分两个通道而产生相近的通道权重（0.571 和 0.565），但是我们希望通道 1123 的权重远远小于通道 674。

图 4.2 挤压与激励模块中的弊端

全局池化的弊端在于可能会将噪声通道与信息通道聚合成相同的信号。使用全局平均池化将每个通道的空间信息直接压缩成 1 维是不科学的，因为不同的响应情况可能有相同的平均值。通过考虑以下事实可以很容易理解这一点：对局部前景区域激活的通道与对背景等不感兴趣区域激活的噪声通道会有一样的响应值，如图 4.2 所示，这将会导致激励模块难以区分他们而产生不科学的权重向量。

为了解决这个问题，一个有效的方法是在激励模块的输入（即挤压模块的输出）中引入额外的局部信息，但是这会引起一个问题：如何在引入附加的局部信息的同时保持原来 SE 模块的计算效率。在 4.1.2 小节，将介绍我们提出的二阶段通道选择方法，来解决 SE 的弊端。

4.1.2 二阶段通道选择方法

我们利用了两种不同的空间聚合策略：空间金字塔池化和基于分辨率的池化。空间金字塔池化可以利用更多输入信息[15][16][17]，基于分辨率的池化可以避免信息在浅层的时候被过早丢弃。

我们提出的基于分辨率的池化使用与神经网络最后层产生的特征图一样分辨率的池化窗口，并且步长等于边长，大分辨率特征被池化成分辨率较大的描述符，小分辨率特征被池化成分辨率较小的描述符。信息融合操作由按通道分组的全连接实现。图 4.3 和图 4.4 给出了我们提出的两个方法的流程图。

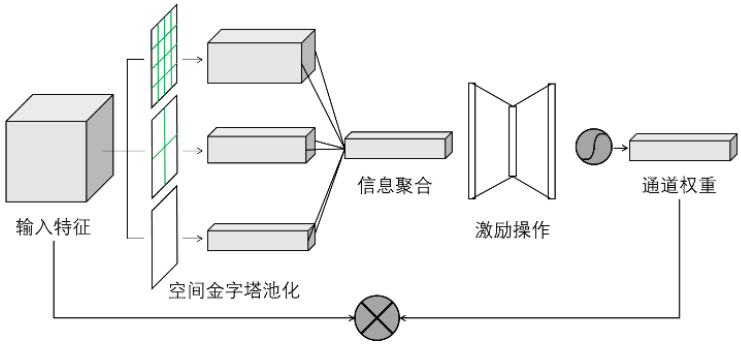

图 4.3 基于空间金字塔池化的挤压与激励模块

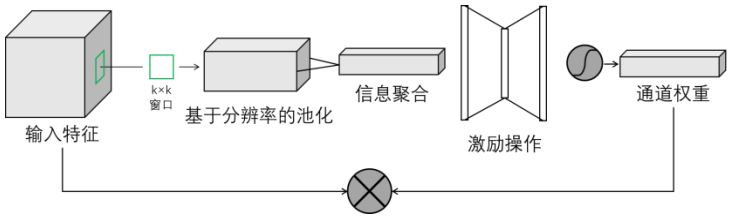

图 4.4 基于分辨率池化的挤压与激励模块

我们重新设计了挤压操作，使得其成为一个可以融合局部信息的高效模块。如图 4.3 和图 4.4 所示，我们的挤压操作是一个两阶段空间信息聚合过程，包括空间聚合(生成特征响应紧凑的中间表示)和信息融合(将中间表示的每一个通道聚合到 1 维)。

我们先定义出现的记号。定义x是一个卷积块的输入，经过一系列卷积操作后，输出$u \in R^{H\times W\times C}$，$H \times W$是空间维度，$C$是通道维度。然后挤压操作和激励操作依次被应用在$u$上来产生对应通道的权重。具体地，在挤压与激励网络（SE-Net）中，挤压操作利用了全局平均池化来聚合特征，即 $z = F_{sq}(u)$，$z \in R^C$。然后激励操作利用自编码器结构来生成权重向量，即$s = F_{ex}(z)$，$s \in R^C$，最后，每一个在特征u上的通道被重新赋予权重,即 $u' = F_{scale}(u, s) = u \times s, u' \in R^{H\times W\times C}$。我们改变了挤压操作$F_{sq}(\cdot)$，使用空间聚合和信息融合两个步骤来聚合特征，舍弃了原来的全局平均池化。

4.1.2.1 空间聚合

空间聚合的目的是将局部信息融合到中间描述符里。在产生的特征响应$u \in R^{H\times W\times C}$中，每个通道单位$u_i \in R^C$是其局部感受野的描述符。通过在空间维度中排列这些局部描述符，u形成整个输入图像的表示。在 SE 模块中，挤压u的空间维度以形成每个通道的紧凑描述符，这样做一方面增强了空间特征编码，另一方面降低了后续激励模块的计算成本。与直接将空间响应压缩为 1 维的原始 GAP 不同，我们的空间聚合阶段旨在生成每个通道的紧凑中间表示，同时又保留u中的局部空间信息。我们提出了两种空间聚合的策略：空间金字塔聚合和基于分辨率的聚合。空间金字塔聚合在多尺度表示和空间池化中具有令人瞩目的特性[17]，我们采用空间金字塔池化来聚合不同维度的特征以产生相同维度的表示, 即将特征聚合成 16、4、1 维，一共 21 维。但是其一个缺点是将不同阶段的特征聚合到同一个维度，忽略了在分辨率和语义方面的差异。由此我们又提出了基于分辨率的池化，我们的动机在于：在神经网络最后一个阶段产生的较粗分辨率的卷积特征上，全局池化表现出的效果较佳，但是在早期阶段，卷积特征的分辨率较大时，全局池化的效果极差。所以我们在每个阶段都使用与最后阶段一样的池化窗口来生成不同阶段有不同分辨率的中间特征。通过这样的方式，精细的早期特征被池化成较高分辨率的中间特征，粗糙的高级特征被池化成低分辨率的中间特征。

4.1.2.2 信息融合

信息融合的目的是将空间聚合产生的中间描述符进一步压缩到 1 维又不丢失局部信息。我们使用了按照通道分组的全连接层来进行信息融合，然后是 batch normalization[12]和 ReLU[21]，如此将每个通道的信息融合到 1 维。对于基于空间金字塔聚合产生的中间描述符，全连接层的输入维度在网络所有阶段都是相同的，即 21 维，但是对于基于分辨率的池化，输入维度随着阶段不同自适应改变。为了记号简单化，我们将在后面的内容使用 SPSE 来代表基于空间金字塔池化的挤压与激励模块，使用 RGSE 来代表基于分辨率池化的挤压与激励模块。

4.2 模型复杂度分析

我们的两阶段空间聚合方法可以有效地提高原始 SE-Nets 的分类准确性。在本节，我们将展示这些性能的提升只会带来极少量的计算复杂度和模型参数。

4.2.1 计算量

为了详细说明计算复杂度对比，我们以 SE-ResNet-50，SPSE-ResNet-50 和 RGSE-ResNet-50 为例进行比较。我们的 SPSE-ResNet-50 和 RGSE-ResNet-50 只增加了 0.0114GFLOPs 和 0.0001GFLOPs 的计算量。具体而言，SE 模块的计算成本来自三个部分：挤压操作的全局池化，

激励操作的全连接层和最后的通道值放缩。因此我们提出的方法的额外计算成本仅来自于新的挤压操作。基于全局池化的挤压操作计算复杂度是$C \times H \times W$，实际上，可以很容易验证所有的非重复的空间池化与全局池化有相同的计算复杂度。单尺度空间金字塔池化和基于分辨率的池化都是非重复的池化，那么三尺度空间金字塔池化的计算量是全局池化的三倍。另外我们提出了非常有计算效率的按通道分组的全连接层(depth-wise FC)来进行特征融合，以 RGSE 为例，信息融合只增加了$C \times (HW/k^2)$的计算量，k是池化窗口大小，因此 SE-ResNet 与 RGSE-ResNet 有几乎相同的计算量，与巨大的基数相比，所增加的额外计算量几乎可忽略不计。

4.2.2 模型参数

模型参数大小分析将在这节介绍。我们的两阶段空间信息融合方法只有在按通道分组的全连接层中引入了额外参数，例如与 SE 对比，我们的 RGSE 方法只增加了$C \times (HW/k^2)$参数，SPSE 方法只增加了$C \times 21$参数，起决定作用的是C。因此我们的两阶段方法只会带来极小的参数负担，以 ResNet-50 为基础，SE-ResNet-50 增加了 2.52M 参数，我们的 SPSE-ResNet-50 和 RGSE-ResNet-50 分别只增加了 2.83M 和 2.62M。

4.3 实验与分析

这一部分介绍我们的性能对比实验和分析实验，其中性能对比实验涉及的计算机视觉任务主要是图像分类，同时目标检测和语义分割两个任务作为我们所提出方法的泛化性展示，也进行了实验。

4.3.1 实现细节

图像分类实验使用 ImageNet[34]数据集，实验设定参照了公认的 AlexNet[1]和 ResNet[4]的实验设定方案。具体是，使用随机梯度下降（SGD）进行优化，momentum 设置成 0.9，weight decay 是 0.0001，初始学习率设置成 0.1，每经过 30 轮训练后将学习率乘 0.1，一共进行 100 轮训练。输入图片的大小是 224×224，每张图片都减去均值后除以标准差，然后以 50%的概率进行水平翻转。我们的 batch size 是 256，并用 8 个 GPU 训练，每个 GPU 每次运行 32 张图片。

对于基于空间金字塔池化的两阶段空间信息聚合模块, 我们首先在特征$u \in R^{H \times W \times C}$上采用三个并行的池化层进行池化，并将空间维度的输出大小分别固定为 4×4，2×2 和 1×1。然后，我们重新整合并连接这些特征以形成 21×1×C 的表示。沿着通道方向，用 C 个 21×1 的全连接层将空间信息融合成 C×1 的表示。最后获得 C 维的通道描述符，其可以直接输入到激励模块。对于基于分辨率的池化，我们执行具有固定步长的单尺度的非重叠平均池化，其中池化窗口大小被设置为整个网络的最后卷积层的输出大小，步长大小与池化窗口边长相同以实现非重叠效果。

4.3.2 实验结果对比与分析

注意力机制被广泛利用来提升神经网络的表示能力。其中挤压与激励网络[14]通过对特征通道的重新加权来提升网络性能，是一种通道维度的注意力机制。CBAM[18]通过在 SE 后面继续引入空间维度的注意力机制提升了网络的表示性能，所以我们将 CBAM 也加入对比。

4.3.2.1 分类实验

我们在 ImageNet 数据及上做了大量的实验以证明我们提出的两阶段通道选择方法的有效性，将 SE-Nets[14]和 CBAM[18]作为我们的比较基准，这两个模块是当今性能最好的模块。我们使用的基本网络结构有 ResNet，加入组卷积后的 ResNeXt 和具有在深度学习中跨时代意义的 VGG。表 4.1 展示了我们的实验结果。

表 4.1　ImageNet 大规模图像分类实验

模型(Model)	计算复杂度 GFLOPs	Top-1 error	Top-5 error
ResNet-50	3.86	24.48	7.49
ResNet-50+SE	3.87	23.21	6.60
ResNet-50+CBAM	3.90	22.65	6.32
ResNet-50+SPSE	**3.88**	**22.40**	**6.18**
ResNet-50+RGSE	**3.87**	**22.27**	**6.15**
ResNet-101	7.58	23.28	6.70
ResNet-101+SE	7.60	22.35	6.13
ResNet-101+CBAM	7.64	21.49	5.68
ResNet-101+SPSE	**7.62**	**21.27**	**5.66**
ResNet-101+RGSE	**7.60**	**21.24**	**5.62**
ResNet-152	11.30	22.44	6.37
ResNet-152+SE	11.32	21.59	5.74
ResNet-152+CBAM	11.39	21.37	5.70
ResNet-152+SPSE	**11.36**	**21.16**	**5.60**
ResNet-152+RGSE	**11.32**	**21.09**	**5.58**
ResNeXt-50	3.89	22.72	6.44
ResNeXt-50+SE	3.90	21.89	6.02
ResNeXt-50+CBAM	3.93	21.91	5.89

ResNeXt-50+SPSE	**3.91**	**21.42**	**5.76**
ResNeXt-50+RGSE	**3.90**	**21.36**	**5.72**
ResNeXt-101	7.63	21.53	5.77
ResNeXt-101+SE	7.65	21.18	5.67
ResNeXt-101+CBAM	7.69	21.10	5.58
ResNeXt-101+SPSE	**7.67**	**20.71**	**5.45**
ResNeXt-101+RGSE	**7.65**	**20.67**	**5.43**
VGG-16	15.47	26.98	8.78
VGG-16+SE	15.48	25.20	7.69
VGG-16+CBAM	15.51	25.13	7.60
VGG-16+SPSE	**15.49**	**24.47**	**7.42**
VGG-16+RGSE	**15.48**	**24.42**	**7.44**
VGG-19	19.63	25.78	8.16
VGG-19+SE	19.64	24.15	7.26
VGG-19+CBAM	19.68	24.30	7.27
VGG-19+SPSE	**19.66**	**23.48**	**6.79**
VGG-19+RGSE	**19.64**	**23.37**	**6.68**

与 SE-Nets 的对比可以看出我们的网络一致性好于 SE-Nets。不仅如此，从表 4. 1 中还可以看出我们提出的 SPSE 和 RGSE 带来的额外计算量非常少，其中 RGSE 几乎与 SE 有一样的计算量。

与 CBAM 的对比结果也与 SE 的对比结果一样。CBAM 是最近提出的用来重新精修卷积特征的方法，它是 SE 方法的扩展。尽管 CBAM 的精度也优于 SE，但是其带来了额外更加多的计算复杂度。不管从精度方面还是从计算复杂度方面，我们的方法一致性优于 CBAM。值得注意的是，在 ResNeXt 系列和 VGG 系列，CBAM 只在 SE 的基础上提升了微乎其微的精度，有的甚至还低，但是我们的方法仍然有很好的表现。

根据表 4.1 的结果，我们会发现两个问题，一个是为什么 RGSE 始终优于 SPSE？另外一个是为什么 SPSE 和 RGSE 都只进行了通道上的注意力方法，却比 CBAM 性能优秀？值得注意的是 CBAM 不仅进行了通道的注意力方法还实施了空间的注意力方法。针对第一个问题，我们注意到空间金字塔池化可以生成特征响应的多尺度表示，应该会比单尺度的好，但是结果却差，那是因为 SPSE 将所有阶段的特征都池化到固定维度，这在早期阶段就丢失太多信息，在后期

引入太少信息。相比之下，RGSE 可以在早期阶段保留有用的局部信息，并且在后期提取更多信息表示。针对第二个问题，CBAM 学习所有通道共享的空间注意力特征，这与通道响应的多样性有冲突，而利用了局部信息产生的通道维度的权重则是符合了通道多样性特点。

4.3.2.2 检测与分割实验

为了进一步评估提出的两阶段空间信息聚合方法的泛化能力，我们在 MS-COCO[26]数据集上进行了目标检测和实例分割实验。实验设置与 SE[14]一样，我们使用 8 万张训练图像，4 万张测试图像。我们用 Faster R-CNN[6]进行目标检测，用 Mask R-CNN[22]进行实例分割。

目标检测的结果如表 4.2 所示，我们把骨干网络从 SE-Nets 改成 SPSE-Nets 和 RGSE-Nets 后进行了对比实验，最后我们获得了一致性的性能提高。实例分割的结果如表 4.3 所示，针对这样的像素级预测任务，我们提出的方法也优于 SE-Nets。

表 4.2 使用 Faster R-CNN 在 MS-COCO 上的目标检测实验

模型	mmAP	AP@0.50	AP@0.75	AR100
ResNet-50+SE	33.0	51.8	35.7	42.5
ResNet-50+SPSE	**34.1**	**54.3**	**36.4**	**45.8**
ResNet-50+RGSE	33.8	54.1	36.1	45.0
ResNet-101+SE	36.4	56.2	39.3	46.6
ResNet-101+SPSE	**36.9**	57.1	**39.8**	**48.0**
ResNet-101+RGSE	36.8	**57.5**	39.5	47.9

表 4.3 使用 Mask R-CNN 在 MS-COCO 上的实例分割实验

模型	mmAP	AP@0.50	AP@0.75	AR100
ResNet-50+SE	30.4	50.9	32.0	41.5
ResNet-50+SPSE	**30.6**	51.5	**32.2**	**42.3**
ResNet-50+RGSE	30.6	**51.6**	32.1	41.8
ResNet-101+SE	32.0	53.2	33.7	41.6
ResNet-101+SPSE	32.5	54.0	34.3	42.6
ResNet-101+RGSE	**32.6**	**54.5**	**34.3**	**43.4**

4.3.2.3 可视化分析

为了理解我们提出的两阶段方法的工作原理，我们进行了可视化分析实验。对于不同通道的特征图，我们给出了对应的全局平均响应、两阶段方法的响应和最后他们的激励权重值。如

图 4.5 和图 4.6 所示，我们发现当用原来的 SE 产生相似的中间响应（GAP 挤压）时，我们的方法可以产生不同的响应（SPP 信息融合与 RG 信息融合）来区分这些通道（有用通道和噪声通道），且在激励模块处理后产生准确的权重（激励响应）。

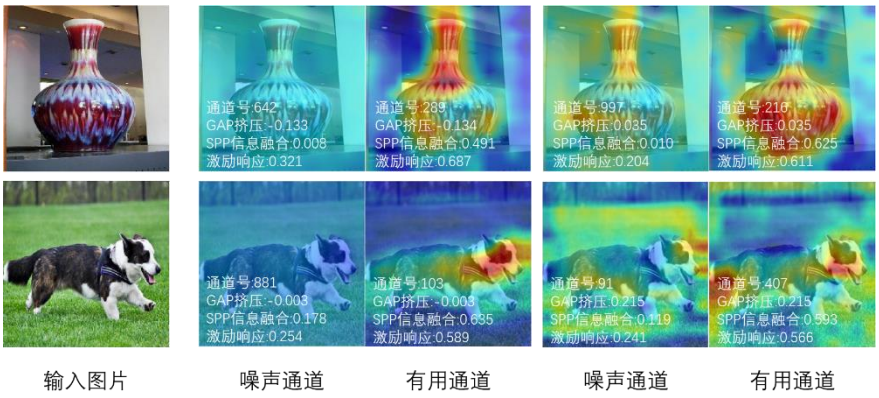

图 4.5 SPSE 的可视化分析

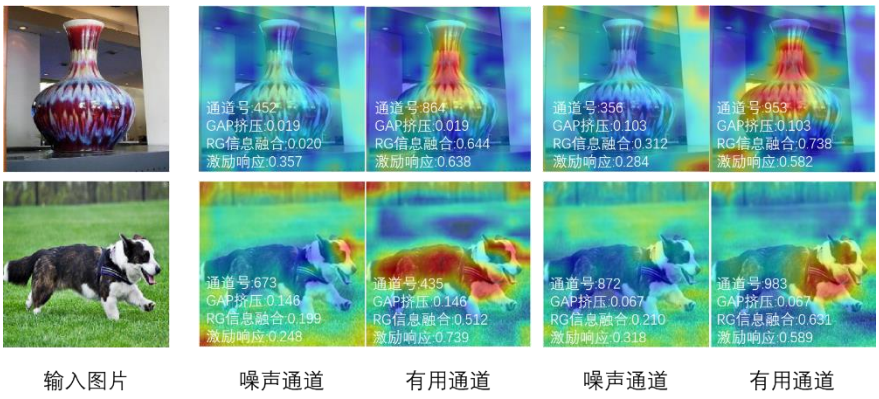

图 4.6 RGSE 的可视化分析

4.4 本章小结

在本章中，我们提出了一种简单有效的两阶段空间信息池化过程来提高卷积神经网络的鲁棒性，包括空间聚合和信息融合。在我们的方法中，空间聚合可以挤压并生成包含比全局平均池化更多局部信息的通道描述符。同时，通过信息融合吸收更多空间信息可以帮助激励操作以数据驱动的方式返回更准确的通道权重分数。大量的实验证明了我们的方法优于原始的 SE-Nets[14] 和 CBAM[18]，比如用于图像分类的 ImageNet 实验，以及用于目标检测和实例分割的 MS-COCO 实验。

第五章 基于特征过滤的鲁棒性提高方法

在本章，我们针对残差网络提出了一个非常轻量级的去噪模块，即一个卷积的编码-解码模块，我们称为 ED 模块。卷积核是卷积神经网络的核心，卷积神经网络由不断堆叠的卷积层构成，中间穿插着激活层、池化层等。通过端到端的训练，这些卷积核被损失函数所驱动，产生对输入的分布式表示[3]。但是如图 5.1 所示，是对“狗”这个概念的可视化效果，第一行“通道采样”展示了三张激活图，这些激活图是从残差网络 ResNet[4]最后一个残差模块的恒等映射中随机采样三个通道得到的，可以看到通道 35 正确地激活在原图的前景部分，通道 1140 和通道 558 错误地激活在原图的背景部分；第二行的“激活效果”图展示了原 ResNet 的所有通道的平均激活效果和我们提出的基于去噪的 ResNet 的激活效果，可以非常明显地看到，我们提出的方法成功的去掉了类似于通道 1140 和通道 558 的噪声信息，使得平均激活更加集中地分布在前景部分。

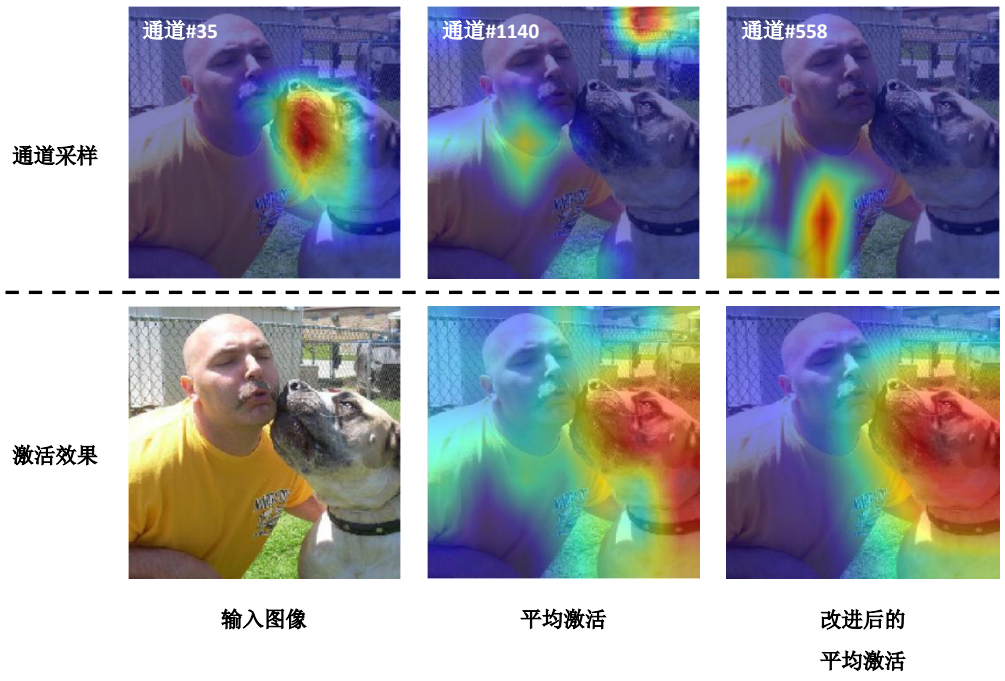

图 5.1 概念“狗”的通道采样与激活效果

既然这种噪声存在于神经网络中，又由于跨层连接的存在使得噪声可以被简单传输到整个网络，我们就有必要去处理噪声信息。因此，我们提出了一种轻量级特征过滤方法。

5.1 轻量级特征过滤方法

在这一节，我将介绍我们针对 ResNet 提出的去噪模块，然后展示可视化效果来证实我们的去噪模块在特征传递中的去噪作用，最后会给出一些定性分析。

5.1.1 去噪 ED 模块

5.1.1.1 去噪模块组成

我们的目标是针对 ResNet 等残差网络设计一个计算复杂度低的去噪模块，所以选择了3×3的卷积层作为编码器，使用了3×3的转置卷积作为解码器，两个层的步长都是 2。编码器的卷积核个数与残差模块的3×3卷积核个数相同，解码器的卷积核个数与残差模块的输出通道数相同。

直接将 ED 模块组合到残差网络中会造成相当大的计算量负担，所以对 ED 模块的卷积进行了分组，也就是以组卷积的形式构成我们的去噪模块。具体来说，我们把3×3的卷积分成了 32 组，这大大降低了计算复杂度。

5.1.1.2 去噪模块预训练

为了使我们的去噪模块有一个好的初始化参数，我们对 ED 模块进行了预训练。首先训练 ED 模块的编码器部分，为了使得其具有一定的特征提取能力，我们在编码器的卷积层后面添加了全局平均池化层以及全连接层，在 ImageNet 数据集上训练分类任务，之后我们丢弃池化层和全连接层，只保留卷积层，为后续预训练解码器做准备。为了使得解码器拥有还原输入的能力，我们在预训练好的编码器后面加上解码器，并且在 ImageNet 上面训练重构任务，期间保持编码器参数不变，只更新解码器参数，并且我们使用 VGG 和 ResNet 抽取的特征增广了 ImageNet 数据集。最后为了增加 ED 模块的去噪能力，我们在增广的 ImageNet 数据集上添加噪声，产生带噪声的数据集，使用带噪声的数据和干净数据训练我们 ED 模块的去噪能力。

5.1.1.3 去噪模块与残差网络的组合

对于任何残差网络，比如 ResNet 和 ResNeXt[43]，我们的 ED 模块与其组合的方式都是非常简单的，图 5.2 的“去噪残差模块”展示了组合形式，即直接针对跨层连接进行去噪。记输入特征为x，残差函数为$F(x)$，我们的去噪函数为$ED(x)$，那么原始的残差模块的映射是$F(x)+x$，由于特征中的噪声通道会通过跨层连接被传递到整个网络中，故我们对跨层连接进行了去噪，所以去噪残差模块的跨层连接的函数映射方式是$A(ED(x)+x)$，其中A是激活函数，比如 Sigmoid、ReLU[21]等，最后，去噪残差模块的输出为$F(x)+A(ED(x)+x)$，值得注意的是，“线性投影”只会在输入与输出特征的分辨率不一样时使用。在后续的内容中，我们会证实去噪函

数会将输入的有效信息保留，噪声信息去除。

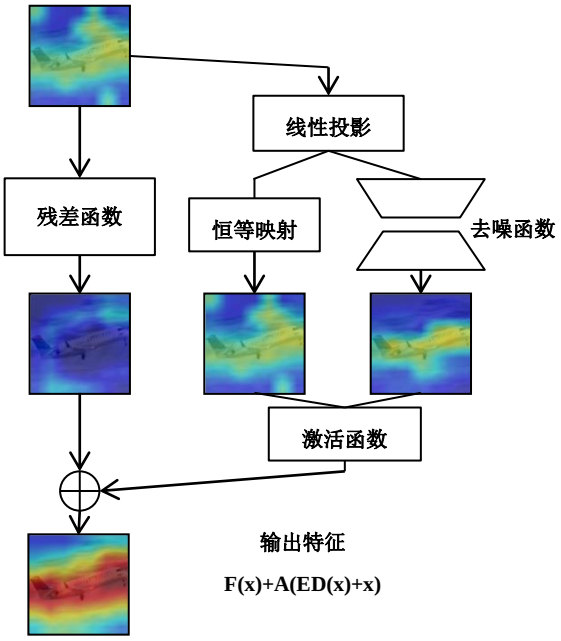

图 5.2 去噪残差模块

为了记号的方便性，我们将装备了 ED 模块的 ResNet 和 ResNeXt 记为 ED-ResNet 和 ED-ResNeXt，在后续的分析与实验中，我们将使用这个记号。

5.1.2 可视化与分析

为了可以直观地理解 ED 模块所扮演的角色，我们对残差模块进行了解耦分析。我们可视化了去噪残差模块的各个单元输出的特征，并且也分析了这些特征的组合。解耦得到的独立单元分别有原始的残差函数、恒等映射和我们提出的去噪函数。

图 5.3 给出了残差模块的特征可视化结果。每一行的组合特征是直接通过相加的形式得到的，这也还原了原本在网络中的特征组合形式，比如原始残差学习是$F(x)+x$，改进后的去噪残差学习是$F(x)+A(ED(x)+x)$，其中$A(ED(x)+x)$是我们提出的特征精修函数，也就是基于去噪函数的跨层连接。从图中我们可以看到输入特征x有着不集中的广泛的特征激活，这些激活包括不相关的背景区域，只通过残差函数来构造出原始的残差学习是不足以去掉这些噪声激活的，但是当把原始的特征作为输入的时候，我们的去噪函数可以产生集中的激活，这些激活都集中在原图最具有判别力的前景区域，同时抑制了背景区域。通过直接的像素级的相加并且激活后，去噪函数能够非常有效的改善原始跨层连接传递的特征质量，形成最后的精修特征。

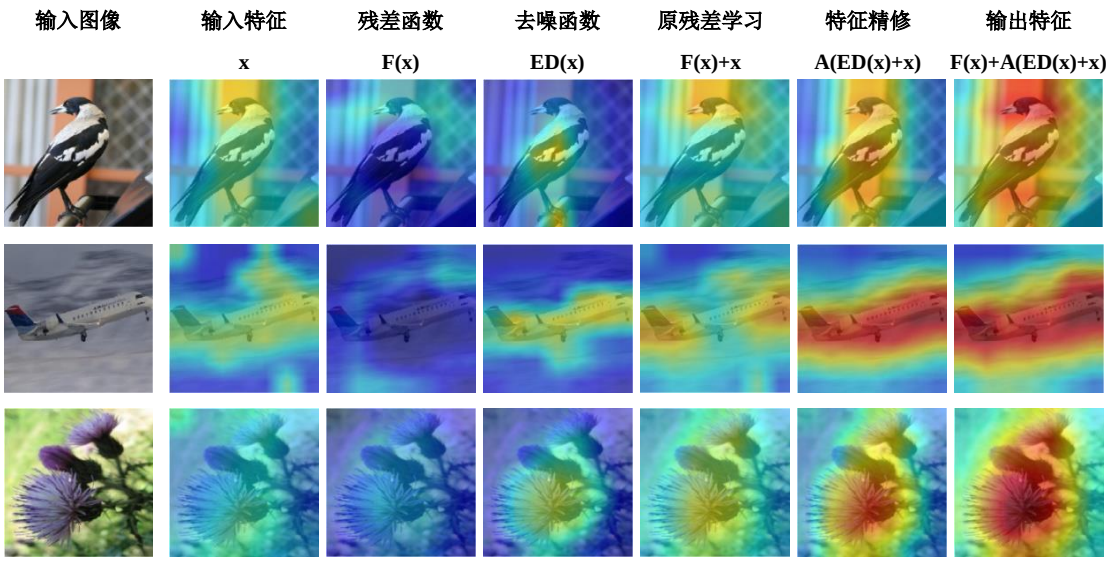

图 5.3 残差模块特征可视化

我们分析了残差函数$F(x)$和去噪函数$ED(x)$的具体扮演的角色。从广义上来说，无论是残差函数还是去噪函数，其作用都是特征转化。但是具体来说，残差函数主要做的是残差学习，即对输入特征的微小转化，而去噪函数做的是提取输入特征中有用的信息同时抑制无关的背景噪声信息。其中残差函数的输出与原文 ResNet[4]中一样，即 ResNet 中残差函数产生的影响对整个残差模块的影响非常小。另一方面，我们的 ED 去噪模块对原始的残差分支几乎不产生影响，我们只针对跨层连接传递的信息进行去噪精修。

5.1.3 与注意力机制对比

我们要强调的是，我们的方法与传统的基于注意力的方法（如 SENets[14]和 CBAM[18]）有着显著的不同。首先，我们不学习额外的注意力权重来重新定义特征响应。相反，我们利用编码器-解码器模块直接从原始输入中提取信息量最大的特征，然后通过简单的元素相加与激活来重新定义所学的表示，以此来抑制无关的背景噪声激活。其次，ED 模块的设计使得我们能够在不明显降低精度的情况下，减少 ResNet 中原始残差分支的计算负担，然而，基于注意力机制的方法在现有的文献中还没有发现这个优点。第三，我们的方法不依赖于任何注意力模块，而 CBAM 是一种高度依赖于 SE-Net 的贡献，因为它在 SE-Net 后插入了额外的空间注意力机制。

由于我们的方法从一个新的角度提高了 CNN 的表示能力，因此可以方便地与基于注意力的方法相结合，进一步提高识别精度。经实验验证，我们的 ED 模块与 SE-Net 集成后，可以提高分类准确度，在许多情况下都优于 CBAM。此外，虽然 CBAM 进一步在 SE-Net 的基础上利用了空间注意力，但在目标检测、实例分割和细粒度识别等具有挑战性的任务中，我们的方法

仍然获得比 CBAM 更好的结果。

5.2 方法实现和实验设置

在 ImageNet 数据集上，为了公平比较，我们的做法与 ResNet[4]保持一致。我们从 0.1 的学习率开始，每 30 个迭代除以 10，所有的模型都训练 100 个迭代。采用动量为 0.9，batch size 为 256 的随机梯度下降法进行优化，权值衰减为 0.0001，值得提出的是，在所有阶段中 ED 模块的学习率是主干网络的$1/10$。输入图像的大小是 224×224，减去每个像素的平均值，并进行标准数据增强和随机水平翻转，之后随机裁剪。我们采用了 MSRA 初始化方法[13]中提出的权值初始化方法。在测试时，我们对验证集图片进行统一的裁剪，即 224×224 个像素是从每个图像的中心裁剪得到。我们在卷积之后执行 batch normalization[12]，然后应用 ReLU[21]作为非线性激活函数。

5.3 实验与分析

我们进行了四个系列的实验，以验证所提出的编码器-解码器（ED）模块对于深度残差网络的有效性，包括 ImageNet-1K[34]分类、MS-COCO[26]目标检测和实例分割、CUB200-2011 数据集上的细粒度图像识别[39]和 CIFAR-10 的解耦研究[23]。为了公平比较，我们重新实现了所有模型。

5.3.1 ImageNet 分类实验

我们在 ImageNet 数据集上进行了三个对比实验，分别是与基本模型的对比、对注意力机制模型的提高以及基于计算复杂度的对比。

5.3.1.1 与基本模型的对比

我们认为 ResNet 和 ResNeXt 是我们方法的最重要的比较标准。我们还用了一个具有相同模型复杂度的两层3×3卷积替换编码器-解码器结构，实现了简单的对比实验，旨在证明 ED 模块的优势。

如表 5.1 所示，去噪模型 ED-ResNet 和 ED-ResNeXt 都比原始模型有更强的分类能力。具体来说，它将 ResNet-50 和 ResNet-101 的 top-1 误差分别降低了 1.22%和 0.91%。ED-ResNet-50 的 top-1 错误为 23.12%，与更深层的架构 ResNet-101 相同，但是 ResNet-101 几乎有两倍的计算复杂度。同样，ED-ResNet-101 的 top-1 和 top-5 错误分别为 22.21%和 6.23%，优于更深的 ResNet-152。

表 5.1 与基本模型的对比

模型	Top-1 error	Top-5 error
ResNet-50	24.34	7.32
ResNet-50 + 2conv	23.75	6.98
ResNet-50 + ED	**23.12**	**6.54**
ResNet-101	23.12	6.52
ResNet-101 + 2conv	22.69	6.34
ResNet-101 + ED	**22.21**	**6.23**
ResNet-152	22.44	6.37
ResNet-152 + 2conv	22.39	6.34
ResNet-152 + ED	**21.98**	**6.09**
ResNeXt-50	22.59	6.41
ResNeXt-50 + 2conv	22.56	6.37
ResNeXt-50 + ED	**22.01**	**6.11**
ResNeXt-101	21.34	5.66
ResNeXt-101 + 2conv	21.30	5.63
ResNeXt-101 + ED	**20.93**	**5.32**

为了确定 ED 的有效性，我们将 ED 进行了变体，即用两个卷积层代替了原来的编码器-解码器结构。我们的 ED-ResNet 和 ED-ResNeXt 始终优于“2conv”的表现，尽管后者也可以改进原始 ResNet 和 ResNeXt。我们注意到，与我们的 ED 相比，新增加的“2conv”分支在集成到 ResNeXt 时只能获得极小的性能改进，推测这是因为分组的“2conv”分支与 ResNeXt 的原始残差分支具有相似的行为，因此在训练过程中很难引入更多有用的信息。

5.3.1.2 对注意力机制模型的提高

本节我们分析了对 SE-Net 和 CBAM 的改进。由于我们的方法从不同的角度对 SE-Net 的残差网络[14]进行了改进，因此我们将 ED 模块集成到 SE-Net 中，以研究它是否能进一步提高识别精度。如表 5.2 所示，我们的 ED 版本的 SE-Net 比原来的 SE-Net 有了合理的改进。ED 模块能进一步降低 SE-ResNet-50 和 SE-ResNeXt-50 的 top-1 误差 0.81%和 0.41%。特别是，我们的 ED 模块的 SE-ResNeXt-50 甚至比更深层的 SE-ResNeXt-101 表现更好。对于 CBAM[18]，它通过结合全局平均池化（GAP）和全局最大池化（GMP）来改进 SE-Net，并将额外的空间注意力模块与 SE 结合起来，在表 5.2 中，我们还将我们的方法与 CBAM 进行了比较。我们的 SE-ResNet-50 和 SE-ResNet-101 的 ED 版本达到了与 CBAM 相当的性能，而 SE-ResNeXt-50 和 SE-ResNeXt-101 的 ED 版本则优于相应的 CBAM。

表 5.2 与注意力机制的结合

模型	Top-1 error	Top-5 error
SE-ResNet-50	23.27	6.59
SE-ResNet-50 + CBAM	22.66	6.31
SE-ResNet-50 + ED	**22.46**	**6.28**
SE-ResNet-101	22.37	6.13
SE-ResNet-101 + CBAM	**21.51**	**5.69**
SE-ResNet-101 + ED	21.71	6.04
SE-ResNeXt-50	21.61	5.72
SE-ResNeXt-50 + CBAM	21.92	5.91
SE-ResNeXt-50 + ED	**21.20**	**5.64**
SE-ResNeXt-101	21.32	5.54
SE-ResNeXt-101 + CBAM	21.07	5.59
SE-ResNeXt-101 + ED	**20.89**	**5.30**

5.3.1.3 基于计算复杂度的对比

我们所提出的 ED 去噪模块对基础网络的模型参数和计算量都有轻微的增加。为了进行公平的正面比较，我们从装备了 ED 模块的网络中移除一部分残差函数通道的分支，使它们的计算复杂度与基础网络相同。具体来说，对于 ED-ResNet 系列的 Conv2 到 Conv5，我们直接移除残差通道3×3卷积层的 4，8，16，32 个通道数；而对于 ED-ResNeXt 系列，我们在不同阶段从3×3卷积层中移除 20，40，80，160 个通道数，然后将所有阶段的通道分成 36 组。我们将这些简化的体系结构称为“ED-ResNet-A”和“ED-ResNeXt-A”。

如表 5.3 所示，在相同的设置下，ED-ResNet-A 和 ED-ResNeXt-A 的表现一致优于 ResNet 和 ResNeXt。例如，ED-ResNet-50-A 将 ResNet-50 的 Top-1 错误率和 Top-5 错误率减少了 1.26% 和 0.85%，甚至比不进行残差通道修剪的原始 ED-ResNet-50 稍好。ED-ResNet-101-A 使 ResNet-101 的 Top-1 错误率降低了 0.89%，与原 ED-ResNet-101 的性能相当。这些结果表明，我们的编码器-解码器模块可以在一定程度上减少残差分支的计算复杂度开销。

此外，由于我们的 ED-ResNeXt-A 将 ResNeXt 的原始分组从 32 更改为 36，我们还实现了 ResNeXt-50 的修改版本，以便进行公平比较。新架构为 ResNeXt-50-36×3d，共 36 组，每组卷积层宽度为 3（同 ED-ResNeXt50-A）。ResNeXt-50-36×3d 的 Top-1 错误率为 23.04%，Top-5 错误率为 6.46%，明显高于 ED-ResNeXt-50-A。

我们进一步又开发了非常有计算效率的 ED 网络实现，在 ED-ResNet 和 ED-ResNeXt 系列的基础上，移除了3×3残差卷积层的一半通道。这大大减少了原始网络的 FLOPs，即分别减少了原始 ResNet-50 和 ResNeXt-50 的 49%和 31%的计算量。我们将这些有效的架构称为

“ED-ResNet-B”和“ED-ResNeXt-B”。

表 5.3 基于计算复杂度的对比

模型	Top-1 error	Top-5 error	计算量（$\times 10^9$ FLOPs）
ResNet-50	24.34	7.32	4.1
ED-ResNet-50-A	**23.08**	**6.47**	4.0
ED-ResNet-50-B	23.94	6.95	**2.1**
ResNet-101	23.12	6.52	7.9
ED-ResNet-101-A	**22.23**	**6.24**	7.8
ED-ResNet-101-B	23.14	6.49	**3.9**
ResNet-152	22.44	6.37	11.7
ED-ResNet-152-A	**22.01**	**6.11**	11.5
ED-ResNet-152-B	22.52	6.41	**5.6**
ResNeXt-50	22.59	6.41	4.2
ED-ResNeXt-50-A	**22.03**	**6.12**	4.2
ED-ResNeXt-50-B	22.61	6.43	**2.9**
ResNeXt-101	21.34	5.66	8.0
ED-ResNeXt-101-A	**20.97**	**5.33**	7.9
ED-ResNeXt-101-B	21.57	5.71	**5.4**

如表 5.3 所示，虽然我们大大减少了计算量，但是 ED-ResNet-B 和 ED-ResNeXt-B 仍然获得了与基础网络相当甚至稍好的性能。例如，ED-ResNet-50-B 仍然在 ResNet-50 上获得更好的性能，并且 ED-ResNeXt-50-B 与 ResNeXt-50 相比，仅遭受非常轻微的性能降低。这些结果表明：1）编码器-解码器的 ED 模块在很大程度上减轻了原始残差分支的计算负担，性能没有明显下降；2）编码器-解码器的 ED 模块在模型压缩和有效模型设计方面具有潜力，是一个很有前途的未来工作方向。

此外，为了验证此特性是否是 ED 模块的独特性质，我们以 ED-ResNet-50-B 为例，将 ED 模块替换为 2conv 模块。这种 2conv 方案的 Top-1 错误率和 Top-5 错误率分别为 24.42%和 7.59%，均高于 ResNet-50 和 ED-ResNet-50-B，此实验证实了可以减少原始模型计算量的特性是 ED 模块独特的性质。

5.3.2 MS-COCO 检测与分割实验

我们利用 MS-COCO 数据集[26]进一步评估了去噪 ED 模块在目标检测和实例分割方面的泛化能力，该数据集包含 80k 训练图像和 40k 验证图像。

5.3.2.1 检测实验

我们训练 Faster R-CNN[6]作为我们的检测算法，然后在 40k 验证图像上对其进行评估。如表 5.4 所示，当改变主干网络时，我们的 ED-ResNet 可以显著提高用于目标检测的 Faster R-CNN 的性能。特别是，在 MS-COCO 的标准度量 mmAP 上，ED-ResNet-50 的表现比 ResNet-50 好 1.9%，在 AP@IoU=0.75 的情况下，提高了 2.3%，这比 AP@IoU=0.50 的情况下的改善更为显著。推测这是因为我们的 ED 模块有助于网络对感兴趣的对象产生更精确的激活，从而大大简化了 Faster R-CNN 回归分支的训练，得到更精确的定位结果。此外，ED-ResNet-101 提高了基于 ResNet-101 的 Faster R-CNN 的 2.5%的 mmAP，这在目标检测方面是一个显著的改进。

表 5.4 基于 Faster R-CNN 的目标检测实验

模型	mmAP	AP@0.50	AP@0.75	AR100
ResNet-50	31.0	50.9	33.1	44.0
ResNet-50 + CBAM	31.6	51.8	33.3	45.4
ResNet-50 + ED	**32.9**	**53.0**	**35.4**	**45.8**
ResNet-101	32.5	52.0	34.9	45.4
ResNet-101 + CBAM	34.2	54.4	36.1	46.2
ResNet-101 + ED	**35.0**	**54.9**	**37.5**	**47.9**

5.3.2.2 分割实验

实例分割是一项具有挑战性的工作，因为它需要进行正确的像素级预测。我们训练 Mask R-CNN[22]作为实例分割的基础算法。Mask R-CNN 是一个能同时预测目标对象框和像素级预测的通用框架。

表 5.5 和表 5.6 分别显示了 Mask R-CNN 的实例分割和目标检测结果。在这两个任务下，带有编码器-解码器的 ED 去噪模型一致优于原始 ResNet 模型。例如，ED-ResNet-50 比 ResNet-50 在分割方面的性能好 1.1%的 mmAP，在对象检测方面的性能好 1.4%的 mmAP。

表 5.5 基于 Mask R-CNN 的实例分割实验

模型	mmAP	AP@0.50	AP@0.75	AR100
ResNet-50	29.2	50.0	30.3	40.7
ResNet-50 + CBAM	28.4	48.6	29.4	40.7
ResNet-50 + ED	**30.3**	**51.4**	**31.6**	**41.5**
ResNet-101	30.6	52.0	31.9	42.0
ResNet-101 + CBAM	30.3	52.0	31.7	42.5
ResNet-101 + ED	**31.6**	**53.2**	**33.0**	**42.5**

表 5.6 基于 Mask R-CNN 的目标检测实验

模型	mmAP	AP@0.50	AP@0.75	AR100
ResNet-50	34.1	54.0	36.8	46.8
ResNet-50 + CBAM	32.9	52.4	35.1	46.5
ResNet-50 + ED	**35.5**	**55.1**	**38.1**	**47.9**
ResNet-101	36.3	56.1	39.0	49.0
ResNet-101 + CBAM	36.0	56.1	38.5	49.4
ResNet-101 + ED	**37.5**	**57.3**	**40.1**	**49.6**

5.3.3 细粒度图像分类实验

使用 CUB200-2011 数据集[39]，我们进一步评估了细粒度图像识别的性能，该数据集包含 200 种鸟类的 11788 幅图像。此外，CUB200-2011 还为鸟类提供了精确的实例分割标注，使我们能够对深度模型自动生成的感兴趣区域（ROI）进行定量研究。对于测试图像，突出区域应该是感兴趣的对象。对于这个数据集，我们在实验中将所有图像的大小调整为 448×448。

5.3.3.1 分类精准度

同样的，我们将我们的方法与原始模型 ResNet 和 ResNeXt 以及它们的 CBAM 版本进行了比较。如表 5.7 所示，我们的方法在不同深度下始终可以取得最佳结果，显示了其细粒度分类的潜力。

表 5.7 细粒度识别对比

模型	Error(Depth = 50)	Error(Depth=101)
ResNet	15.14	14.51
ResNet + CBAM	15.01	14.40
ResNet + ED	**14.84**	**14.35**
ResNeXt	14.52	14.23
ResNeXt + CBAM	14.41	14.11
ResNeXt + ED	**13.91**	**13.53**

5.3.3.2 感兴趣区域的分析

我们在 CUB200-2011 数据集上进行了定量评估，以验证我们基于 ED 的模型能够更准确地处理主要感兴趣对象的区域。首先得到 ResNet 和 ResNeXt 的最后一个卷积块产生的特征向量，然后通过深度方向相加得到二维矩阵，接下来，我们计算矩阵的平均值，并将其作为目标定位的阈值。我们使用预测的显著区域和实例分割标签之间的交并比（IoU）作为“感兴趣区域准

确性”的度量。预测的显著区域（感兴趣区域）如图 5.4 所示，对 CUB200-2011 中的一些图像的感兴趣区域进行了可视化，通过不同的模型计算最后的卷积输出：（a）中显示了五个鸟类图像和相应的实例分割标签；（b）中是分别由 ResNet-50、CBAM-ResNet-50（ResNet-50+CBAM）和 ED-ResNet-50（ResNet-50+ED）生成的激活图；（c）是通过使用平均值对原始激活图进行阈值化来划分感兴趣区域的。这清楚地表明 ED 模块有助于去除冗余激活并提取信息量最大的特征。

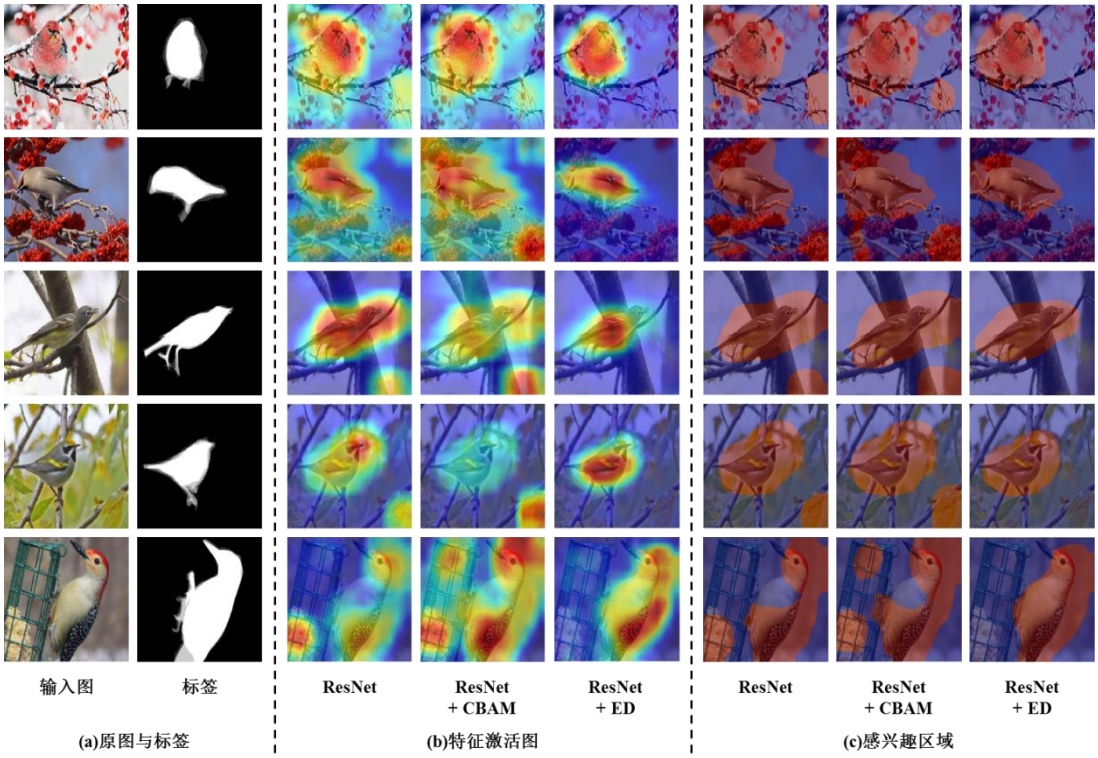

图 5.4 感兴趣区域可视化图

定量结果如表 5.8 所示，其中我们的方法始终优于其他方法。我们还研究了 ED 模块的输入特征和输出特征所产生的感兴趣区域，如表 5.9 所示，由 ED 模块的解码器特征产生的感兴趣区域比由输入特征产生的感兴趣区域更准确，这清楚地表明 ED 路径有利于提取信息量最大的特征和去除冗余激活。注意，由于我们的 ED 模块只是 ED 集成到 ResNet 和 ResNeXt 后的一条路径，因此解码器特征的“感兴趣区域准确性”低于完整输出，因为后者总结了原始残差分支、恒等映射和 ED 模块的输出。

表 5.8 感兴趣区域定量对比

模型	IoU(Depth=50)	IoU(Depth=101)
ResNet	54.99	55.57
ResNet + CBAM	58.93	59.04
ResNet + ED	**59.86**	**59.97**
ResNeXt	55.03	55.62
ResNeXt + CBAM	58.98	59.18
ResNeXt + ED	**59.92**	**60.12**

表 5.9 ED 模块的输入和输出对比

特征	IoU(Depth=50)	IoU(Depth=101)
ResNet 中的 ED 输入	51.76	52.43
ResNet 中的 ED 输出	**53.92**	**54.51**
ResNeXt 中的 ED 输入	52.08	52.97
ResNeXt 中的 ED 输出	**54.17**	**55.49**

5.3.4 解耦实验

5.3.4.1 与双卷积的对比

在 CIFAR-10 数据集[23]上，我们将我们的方法与 2conv 进行了比较，以进一步证明我们的 ED 模块的有效性和唯一性。对于 ResNet 系列的错误率，如表 5.10 所示，ED 模块和 2conv 都可以提高具有不同深度的 ResNet 的性能，但是 ED 模块始终表现得更好。

表 5.10 基于 ResNet 的 ED 与 2conv 对比

模型	原始版本	ED 版本	2conv 版本
ResNet-20	7.81	**7.33**	7.57
ResNet-32	7.32	**6.67**	6.83
ResNet-44	6.95	**6.21**	6.36
ResNet-56	6.47	**5.93**	6.27
ResNet-110	5.78	**5.63**	6.15

表 5.11 基于 ResNeXt-29 的 ED 与 2conv 对比

ResNeXt-29	原始版本	ED 版本	2conv 版本
保留所有残差通道	3.62	3.59	3.78
保留 25%残差通道	-	**3.41**	3.86
保留 50%残差通道	-	3.56	3.75
保留 75%残差通道	-	3.50	3.84

对于 ResNeXt 系列的错误率对比，我们采用了文献[43]中使用的 ResNeXt-29 体系结构进行公平比较，并改变原始残差分支卷积层的通道数的比例，进行了更深入的研究。我们观察到，最轻量级版本的 ED-ResNeXt-29 拥有最佳的性能，而 2conv 版本的表现比原始 ResNeXt-29 差（参见表 5.11）。这些结果也证实了我们的 ED 模块的有效性。

5.3.4.2 组卷积带来的影响

最近的研究[19]表明，组卷积可以大大降低模型的计算复杂度，而没有明显的性能下降。我们进一步研究了这个现象是否会发生在我们的编码器-解码器（ED）模块中。

如表 5.12 所示，具有和不具有组卷积的编码器-解码器（ED）均优于基本原始架构。我们观察到，与没有组卷积的 ED 版本相比，组卷积导致精度略有下降，这说明我们的精度提高主要来自编码器-解码器（ED）结构，而不是组卷积带来的收益。另一方面，由于组卷积显著减少了编码器-解码器（ED）模块的计算负担，所以我们将组卷积技术应用到了所有编码器-解码器（ED）模块中。

表 5.12 组卷积的影响

模型	**原始版本**		**不分组的 ED 版本**		**卷积分组的 ED 版本**	
(Model)	错误率	参数量	错误率	参数量	错误率	参数量
ResNet-20	7.81	0.27M	**6.77**	0.56M	7.33	**0.30M**
ResNet-32	7.32	0.46M	**6.46**	0.94M	6.67	**0.51M**
ResNet-44	6.95	0.66M	**5.97**	1.34M	6.21	**0.72M**
ResNet-56	6.47	0.85M	**5.79**	1.73M	5.93	**0.92M**
ResNet-110	5.78	1.70M	**5.53**	3.40M	5.63	**1.84M**

5.3.4.3 编码维度的确定

我们研究了 ED 模块中编码器对输入降低的空间维度对最终分类性能的影响。通过改变输入特征的填充数和卷积步长（将卷积核设为3×3），可以得到编码器输出的不同维度的特征。我们比较了不同的减少率（有 10%，20%，30%，40%，50%，60%，70%，80%，90%），当减少率等于 50%（填充数为 1，步长为 2）时，ED 模块达到了最佳性能。

5.4 本章小结

在本章节，我们提出了一种新的用于深度残差网络的轻量级特征过滤方法，并由卷积编码器-解码器模块实现，它可以被看作是对现有的恒等映射的增强模块。由于编码阶段的信息过滤能力，解码器部分既能突出语义高度相关的前景激活，又能抑制不相关或有噪声的背景响应。

通过去除原始残差分支中的一部分通道，我们可以获得 ED 网络的轻量级版本，而不会导致明显的精度下降。对多个大规模数据集的大量实验，验证了我们的方法在大规模图像分类、目标检测、实例分割和细粒度识别等方面能提高各种残差结构的性能。

第六章 基于对抗训练的鲁棒性提高方法

深度神经网络已经成功地应用于各种任务，包括图像分类[1]、语音识别[46]以及通过深度强化学习以人类的水平或者超过人类水平来玩游戏[47]。但是在最近几年，Szegedy 等人表明卷积神经网络可以很轻易得被微小的扰动所欺骗[103]。从那时起，许多对抗样本生成方法被提出，包括显著图攻击（JSMA）[101]、投影梯度下降（PGD）攻击[49]和 C&W[44]攻击。可以将攻击的形式分为两种：白盒攻击和黑盒攻击。在白盒攻击中，攻击者对目标网络有完整的了解，包括网络的结构和参数。而在黑盒攻击中，攻击者只有目标网络的输出信息[50]。

为了减轻对抗样本对神经网络的攻击，各种防御方法被提出以提高卷积神经网络的鲁棒性。在白盒攻击中，对抗性训练通过对抗样本增强了训练数据集，显示出良好的防御性能[51]。除了对抗性训练外，还有许多其他防御方法，包括防御蒸馏[52]、随机分组[53]等。

我们提出了基于生成网络（GAN）[54]的对抗样本产生方法，并将基于此方法的对抗训练称为生成对抗训练。在普通的 GAN 中，生成网络接受随机的低维向量并返回高维的生成图像（假图像）。判别网络接收真实图像和假图像，并返回二分类决策，0 表示假图像，1 表示真实图像。生成网络的目的是模仿真实图像生成假图像，以欺骗需要攻击的网络。在我们的方法中，我们不使用低维噪声作为生成网络的输入，而是将原始的训练样本作为生成网络的输入。因此，生成网络是一个自编码器式的结构，它将一个干净的图像映射成对该输入的扰动。然后，判别网络有两种类型的输入：原始干净图像和对抗样本（将干净图像与生成的扰动相加）。与普通 GAN 不同，我们的判别网络的目的是用正确的标签对干净样本和对抗样本都进行正确分类，而生成网络的目的是产生强大的扰动来愚弄判别网络以迫使判别器有更高的鲁棒性。

6.1 对抗样本生成方法介绍

在生成性对抗网络（GAN）[54]中，目标是训练神经网络，它可以模拟真实数据的分布。生成网络和判别网络是通过梯度下降一起训练的，在纳什均衡状态下，希望生成网络产生的样本与实际训练数据不可区分。我们采用类似 GAN 的方法产生对抗性噪声，作为提高判别模型鲁棒性的手段。

我们并不是针对一组不变的对抗样本进行训练，而是针对一个对抗噪声生成式网络进行训练。在实验中，我们证明了这一方法可以有效地提高神经网络的鲁棒性。

给定一个具有正确标签y的输入x，我们希望找出噪声Δx，使得$x+\Delta x$可以被神经网络错误地分类到其他一些标签$y' \neq y$上。我们将该噪声建模为$\epsilon G(x)$，其中G是生成特定样本噪声的生成神经网络，ϵ是控制扰动大小的比例因子。注意，与 FGSM 或 PGD 等白盒攻击方法不同，一

旦训练好G，就不需要知道它正在攻击的网络的参数，G还可以接受其他输入来产生对抗性噪声。为简单起见，我们规定G接受x作为输入。

假设我们有一个图像-标签对的训练集$\{(x_1, y_1), \ldots, (x_n, y_n)\}$。设$D_\theta$为$\theta$参数化的被攻击网络（用于分类），$G_\phi$为$\phi$参数化的扰动噪声生成网络。我们要解$D_\theta$和$G_\phi$之间的极小极大问题：

$$F(\theta, \phi) = \min_\theta \max_\phi \sum_{i=1}^{n} J(D_\theta(x_i), y_i) + \lambda \sum_{i=1}^{n} J\left(D_\theta\left(x_i + \epsilon G_\phi(x_i)\right), y_i\right)$$

这里的J是交叉熵损失函数，λ是权衡参数，权衡最小化正常样本的损失与最大化对抗样本的损失，ϵ是噪声的放缩大小。图 6.1 给出了流程图。

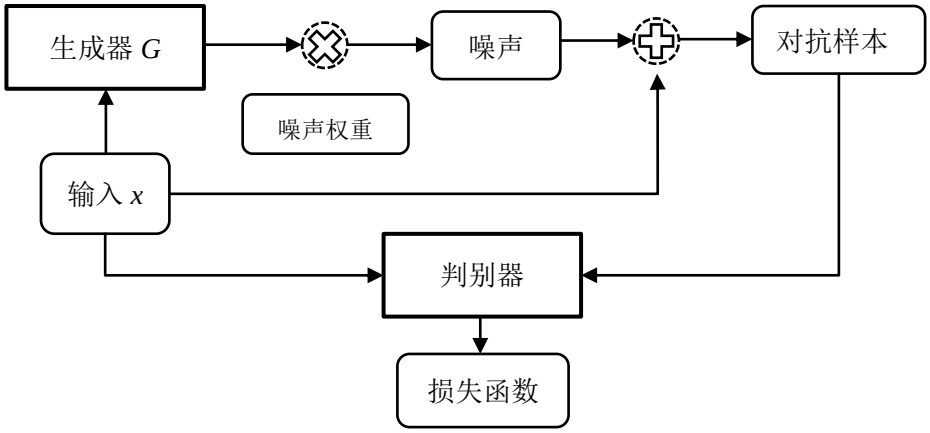

图 6.1 对抗样本生成方法

我们在这项工作中，主要研究基于l_∞范数的扰动。将 tanh 层作为生成网络G_ϕ的最后一层，可以很容易地实现将输出规范化到$[-1,1]$的范围。基于l_1或l_2范数的扰动可以通过在G_ϕ的最终层中增加适当的归一化层来实现。

对于判别网络，我们希望找到一个对于干净样本风险较小的θ解：

$$\min_\theta \ R(\theta) = \sum_{i=1}^{n} J(D_\theta(x_i), y_i)$$

同时也要求在最大扰动下找到对抗样本的最小风险以增加其鲁棒性：

$$\min_\theta \ R_{adv}(\theta) = \sum_{i=1}^{n} \max_{\Delta x, \|\Delta x\| \le \epsilon} J(D_\theta(x_i + \Delta x), y_i)$$

我们将对抗样本建模为一个具有有限容量的神经网络G_ϕ，而不是让扰动Δx完全自由，所以我们希望生成器能够尽量欺骗判别网络：

$$\max_{\phi} R_{adv}(\phi) = \sum_{i=1}^{n} J\left(D_\theta\left(x_i + \epsilon G_\phi(x_i)\right), y_i\right)$$

在这里，对抗噪声$G_\phi(x_i)$不依赖于被攻击网络的参数θ，生成网络的参数ϕ是在所有样本中共享的，而不是像Δx这样针对每个样本单独计算。在攻击领域，当不知道被攻击网络参数时，这更接近于黑盒攻击的情况，但是在训练具有鲁棒性的分类网络时，此生成对抗训练方法仍旧是白盒的，无论是何种攻击，我们的重点是提高神经网络的鲁棒性。然而，我们仍然希望G_ϕ具有足够的表现力来表示强大的攻击，以便D_θ有一个好的对手来训练。先前的研究[55][56]表明，有强大的G_ϕ可以有效地攻击训练有素的D_θ。

值得注意的是，我们的生成网络产生的噪声是多步骤的，每次得到当前的对抗样本后，将其作为“原图像”再次输入到生成网络来生成对抗性噪声以生成新的对抗样本：

$$x'_0 = x$$

$$x'_k = x'_{k-1} + \epsilon G_\phi(x'_{k-1})$$

在传统的生成性对抗网络中，我们最感兴趣的是由生成网络学习到的分布，判别网络是驱动训练的助手，可以在训练后丢弃。而在我们的工作中，我们对分类网络和生成网络都感兴趣。生成网络可以给我们一个强大的攻击模型，可以用来攻击其他判别器，而判别网络是通过生成对抗训练产生的，可以作为一个鲁棒性强大的分类器，可以抵御对抗噪声，因此我们的方法可以同时做到攻击与防御。

6.2 GAN 的交替训练

在实验中，我们使用随机初始化训练判别网络和生成网络，其中并没有使用预训练网络。我们不需要用干净的样本来预先训练判别网络，也不需要对某些固定的判别网络进行生成网络的预训练。

在实验中，我们发现如果我们对两个网络同时进行相同次数的参数更新，那么判别网络D_θ会使生成网络G_ϕ的表示能力过小，这可能会导致差的鞍点解，在差的鞍点解中，G_ϕ无法相对于D_θ进行局部改进，但实际上，只要在ϕ上进行更多的梯度下降，就可以使G_ϕ变得更强大。换言之，我们希望鞍点解周围的区域对于G_ϕ是相对平坦的。为了使生成网络更强大，使之成为噪声生成式网络，即可以产生有效对抗样本的生成网络，我们采用以下策略：对于D_θ的每次θ更新，我们使用相同的小批量数据对ϕ执行多个梯度下降。在小批量数据上进行多次梯度下降可以允许生成网络从小批量数据的输入中学习到对D_θ来说具有高损失的对抗性噪声。具体地在实验中，我们在每个小批量数据上运行 3 个梯度下降步骤。我们将干净样本的损失和对抗样本的损失的权重参数λ设为 1。

6.3 实验与分析

除了我们的生成对抗训练，我们还进行了模型的标准训练，同时也使用了基于 PGD 的对抗训练，我们将三者进行了比较。对于攻击，我们主要关注常用的快速梯度符号法（FGSM）和更强大的投影梯度下降（PGD）方法。

其中快速梯度符号法如下：

$$x_i' = x_i + \epsilon \cdot sign\left(\nabla_{x_i} J(D_\theta(x_i), y_i)\right)$$

投影梯度下降法如下：

$$x_i'^0 = Project_X(x_i + \epsilon u)$$

$$x_i'^{k+1} = Project_{B_\epsilon^\infty(x_i)\cap X}\left(x_i'^k + \epsilon \cdot sign\left(\nabla_{x_i} J(D_\theta(x_i'^k), y_i)\right)\right)$$

其中$Project_X$是投影函数，将输入投影到图像X的可行区域，其中u是$[-1,1]^d$中的一个均匀随机向量，ϵ是缩放因子，$B_\epsilon^\infty(x_i)$是一个半径为ϵ，中心为x_i的无穷范数球。

6.3.1 MNIST 实验

对于 MNIST，输入是大小为 28×28 的阿拉伯数字图像，图像是黑白的，像素值在 0 到 1 之间。我们将输入重新缩放到[-1,1]的范围。与先前的工作[57]一样，我们研究了$\epsilon = 0.3$时的扰动。我们使用一个简单的卷积神经网络作为我们的判别网络，对于我们的对抗性方法，我们使用编码器-解码器网络作为生成器。

我们使用学习率为 0.01、动量为 0.9、批量大小为 64 的 SGD 作为优化器，并对所有判别网络运行 10 万次迭代。经过 5 万次迭代后，学习率降低了 10 倍。对于生成网络，我们使用具有固定学习率为 0.01 且动量为 0.9 的 SGD 来优化。我们通过对G_ϕ进行更多的梯度下降来提高D_θ输入的不确定性，因此我们每更新一次D_θ就对G_ϕ进行 3 次更新。

表 6.1 显示了在扰动为 0.3 下，不同模型的白盒攻击精度。在 FGSM 和 PGD 攻击下，其准确率保持在 90%以上。我们的生成对抗训练模型比未设防的标准训练模型有更好的表现，但在 FGSM 攻击的准确性上与 PGD 训练模型仍有差距。然而，与标准训练模型和生成对抗训练模型相比，PGD 训练方法在干净样本上的精度有一个明显的下降。

表 6.1　MNIST 上白盒攻击下的分类准确度

训练方法/攻击方法	无噪声	FGS 攻击	PGD 攻击
标准训练 A	99.4	28.7	13.2
PGD 训练 B	97.7	95.9	92.3
生成对抗训练 C	99.6	95.7	93.3

表 6.2 也显示了不同模型的黑盒攻击精度。我们通过对代理模型A'、B'和C'运行 FGSM 和 PGD 攻击来生成黑盒攻击图像。这些代理模型的训练方法与它们的对应模型（标准训练对应 A、基于 PGD 的对抗训练对应 B、基于生成网络的对抗训练对应 C）相同，但使用不同的随机种子。我们注意到黑盒攻击倾向于对用相同方法训练的模型最有效。尽管基于 PGD 的对抗训练在基于 FGSM 的白盒攻击上优于我们的生成对抗训练方法，但它们在这些黑盒攻击上性能较差。

表 6.2　MNIST 上黑盒攻击下的分类准确度

训练方法/攻击方法	FGS-A’	PGD-A’	FGS-B’	PGD-B’	FGS-C’	PGD-C’
标准训练 A	49.4	23.4	91.5	89.1	92.2	90.3
PGD 训练 B	94.7	94.1	93.5	93.1	96.1	96.3
生成对抗训练 C	98.7	98.1	98.2	98.5	97.4	96.8

6.3.2 CIFAR 实验

对于 CIFAR-10，我们也将输入放缩到[-1,1]的范围。我们还通过随机填充和裁剪进行了图像的数据增强。我们将训练判别网络的批量大小设置为 64，一共进行了 10 万次训练迭代，每训练 2 万 5 千次迭代就将学习率降低 10 倍，使用 SGD 作为判别网络的优化器。我们使用学习率为 0.002 的 Adam 作为生成网络的优化器。

表 6.3 显示了在不同形式白盒攻击下的精度。基于 PGD 对抗训练的模型在干净数据和 FGSM 攻击下的精度比我们的生成对抗训练方法低。我们的生成对抗训练方法在干净数据的精确性方面能够与标准训练保持一致且稍好，并且在对抗 FGSM 和 PGD 攻击上有很好的鲁棒性。

表 6.3　CIFAR 上白盒攻击下的分类准确度

训练方法/攻击方法	无噪声	FGS 攻击	PGD 攻击
标准训练 A	91.7	55.3	19.1
PGD 训练 B	75.7	47.2	41.3
生成对抗训练 C	91.9	77.8	45.5

表 6.4 显示了模型的黑盒攻击精度。我们的生成对抗训练方法在总体上优于其他方法。基于 PGD 对抗训练的模型在 FGSM 攻击下也能很好的保持精度，但它的总体结果不是最好的，且它的缺点是在干净样本上的精度较低。

表 6.4　CIFAR 上黑盒攻击下的分类准确度

训练方法/攻击方法	FGS-A’	PGD-A’	FGS-B’	PGD-B’	FGS-C’	PGD-C’
标准训练 A	65.9	27.8	69.1	67.1	70.3	71.3
PGD 训练 B	73.6	73.1	57.2	55.6	73.9	73.7
生成对抗训练 C	82.7	80.5	79.7	79.4	78.5	76.9

6.3.3 实验可视化

在本节我们可视化了基于生成网络的对抗样本生成方法产生的图片，目的是为了展示我们的方法产生的图片并不是杂乱的，即符合对抗样本的一个严格要求：人类肉眼无法察觉。

图 6.2 展示了我们的生成式对抗样本产生方法在 MNIST 数据集上的可视化结果。其产生的方法是通过提取生成网络，即编码器-解码器网络的最后一层输出作为我们产生的对抗样本。可以看见我们的方法产生的样本与实际图片相差甚少，且人类肉眼无法识别其到底是不是对抗样本。

图 6.2 MNIST 上对抗样本可视化结果

图 6.3 展示了我们的生成式对抗样本产生方法在 CIFAR 数据集上的可视化结果。其产生的方法也是通过提取生成网络最后一层的输出作为我们产生的对抗图片，以此来攻击目标模型和进行对抗训练来提高卷积网络的鲁棒性。同样的可以看到，我们的方法产生的对抗样本与实际图片相比，人类肉眼是无法识别其到底是不是对抗样本的。

图 6.3 CIFAR 上对抗样本可视化结果

6.3.4 分析

在实验中，我们发现基于 PGD 的对抗样本通常在白盒攻击中效果较好，但是基于 PGD 对抗训练的模型在干净数据上的表现不尽人意，一个原因是 PGD 方法产生的对抗噪声往往对原始样本的改动较大，使得模型容量有限的分类器不能同时拟合原始样本和对抗样本。最近的一些研究表明，标准训练和对抗训练是两个截然不同的问题[58][59]，这使得基于 PGD 的对抗性训练在黑盒攻击下不能得到好的表现。

同时在我们的实验中也可以观察到，对于黑盒攻击来说，最有效的对抗样本产生方法是由使用相同训练方法得到的代理模型产生的，代理模型的训练方式只是更改了随机种子而已。这表明在防御黑盒攻击时，隐藏训练方法是一个重要的防御手段。当今也有研究表明防御黑盒攻击与对抗样本的可转移性密切相关[60]，可转移性指的是使用相同的对抗样本可以攻击不同的模型，比如使用白盒攻击产生的对抗样本可以攻击黑盒模型，但影响可转移性的因素仍不明了。

在我们的实验中，生成器结构的选择对提出方法的质量没有太大的影响。我们尝试了不同的网络结构来训练，但是其攻击成功率以及对分类器鲁棒性提高的帮助变化不大，反而训练的随机性（例如在梯度下降过程中每个网络的学习率和一共训练的迭代次数）比网络结构对最终学习得到的参数的质量有更大的影响。另外值得提出的是，当给定生成网络的结构并且保持不变，改变判别网络的结构进行训练后，会导致生成网络有不同的参数鞍点解。此外，最近的研究也表明，在神经网络损失值的等高线图中，其最小值周围存在连通区域[61][62]。

最后我们发现我们的方法可以扩展到多个判别网络与多个生成网络的竞争中。另外它还可以与不同的训练形式相结合，比如在进行分类器的对抗训练时，一些对抗样本来自预训练模型，而另一些则来自于跟判别网络一起竞争训练的生成网络。

6.4 本章小结

我们提出了一种噪声生成式对抗性训练方法，使用生成网络产生的对抗样本来提高卷积神经网络的鲁棒性，训练过程中产生的生成网络可以作为一种有效的对抗样本生成方法进行黑盒攻击。实验表明，我们提出的生成对抗训练方法在提高神经网络鲁棒性方面，都能取得最优结果。

第七章 总结与展望

7.1 工作总结

我们的工作集中在提高神经网络鲁棒性方面，首先我们梳理了到目前为止流行的提高网络鲁棒性的方法，包括神经网络架构方面、损失函数方面、神经网络的对抗样本方面。针对目前的一些在深度学习中的问题，我们从三个方面提出了三个方法。针对神经网络中的噪声通道问题，我们提出了基于通道选择的神经网络鲁棒性提高方法，针对残差网络族中的噪声信息传递问题，我们提出了基于特征过滤的鲁棒性提高方法，针对当今流行的对抗样本问题，我们提出了基于对抗训练的鲁棒性提高方法。

对于基于通道选择的神经网络鲁棒性提高方法，挤压和激励（SE）模块通过重新加权通道响应已经证明了对于最先进的深度网络结构具有显著的精度增益效果。SE 模块是一个集成了两个操作的结构单元：一个使用全局平均池化将空间卷积特征聚合到信道特征中的压缩操作，一个从压缩特征中学习样本特定通道权重以重新加权每个通道的激励操作。我们回顾了 SE 模块中的挤压操作，并说明了为什么以及如何以最小的额外成本将丰富的（全局和局部）信息嵌入到激励模块中。特别地，我们提出了一个简单而有效的两阶段空间池化过程：空间聚合和信息融合。空间聚合步骤旨在获得一组既包含全局特征又包含局部特征的中间描述符，这些描述符包含比全局平均池化更多的信息。同时，通过融合步骤吸收这些描述符所提供的更多信息，可以帮助激励操作以数据驱动的方式返回更准确的权重得分。通过在 ImageNet 上进行图像分类和在 MS-COCO 上进行目标检测和实例分割实验，验证了该方法的有效性。对于这些实验，我们的方法在所有任务上都比 SE 有了一致的提高。

对于基于信息过滤的鲁棒性提高方法，我们提出了一种新的用于深度残差神经网络的轻量级模块。该方法是一个简单的即插即用模块，即卷积编解码器（ED）。由于编码阶段的降维操作，解码器部分倾向于生成对前景有集中激活的特征，而不相关的响应被抑制。我们的 ED 模块增强了由恒等映射和原始变换分支导出的特征的表示能力。此外，我们还通过移除原始转换分支中的一部分通道来开发轻量级版本。幸运的是，我们的轻量级处理不会导致明显的性能下降，且会带来计算效率的提高。通过对 ImageNet、MS-COCO、CUB200 和 CIFAR 进行综合实验，我们证明了我们的 ED 模块对于各种残差结构有一致性的性能提高。具体来说，在 ImageNet 分类任务中，ResNet-50 和 ResNet-101 的 top-1 误差分别降低了 1.22%和 0.91%，在 MS-COCO 目标检测任务中，以 ResNet-101 为主干网络的 Faster R-CNN 的 mmAP 提高了 2.5%。

对于基于对抗训练的鲁棒性提高方法，其动机是近几年研究人员发现神经网络很容易被愚

弄，且微小扰动是人类无法察觉的。攻击算法产生的对抗样本可以使得神经网络输出错误结果。在训练中加入对抗样本是抵御攻击的常用鲁棒性提高方法。我们在生成性对抗网络（GAN）框架基础上提出了一种新的防御机制：我们使用一个生成网络对对抗性噪声进行建模，利用生成的对抗样本增加分类卷积神经网络的鲁棒性。

7.2 未来展望

在基于信息过滤的鲁棒性提高方法中，我们提出了用于特征去噪的 ED 模块，在实验中我们发现，ED 模块有利于降低模型的计算复杂度，这在模型压缩方向值得深入研究。我们会在未来的工作中争取做到保证模型性能的同时，降低模型的计算复杂度，实现高鲁棒性的轻量级模型设计。

在人工智能落地的热潮中，拥有高鲁棒性的轻量级模型是必须的，因为我们需要把模型部署到各种移动设备上，这些设备的计算资源往往是有限的，大模型的高计算复杂度往往不能做到实时反馈。但是另外一点重要的要求是需要保证模型的性能，当前的研究表明，降低模型计算复杂度的同时模型性能的损害是不能避免的，所以这一方面的工作亟待出现。

参考文献

[1] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In Conference and Workshop on Neural Information Processing Systems(NeurIPS), 2012, 1097–1105.

[2] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[3] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence(IEEE TPAMI), 35(8):1798–1828, 2013.

[4] He KM, Zhang XY, Ren SQ, Sun J. Deep residual learning for image recognition. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, 770-778.

[5] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. arXiv preprint arXiv:1612.08242, 2016.

[6] Ren SQ, He KM, Girshick R, Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. In Conference and Workshop on Neural Information Processing Systems(NeurIPS), 2015, 91-99.

[7] George Saon, Hong-Kwang J Kuo, Steven Rennie, and Michael Picheny. The ibm 2015 english conversational telephone speech recognition system. arXiv preprint arXiv:1505.05899, 2015.

[8] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Advances in neural information processing systems, pages 3104–3112, 2014.

[9] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499, 2016.

[10] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. Nature, 521(7553):436–444, 2015.

[11] Charlotte Middlehurst. China unveils world's first facial recognition atm. http://www.telegraph.co.uk/news/worldnews/asia/china/11643314/China-unveils-worlds-first-facial-recognition-ATM.html, Jun 2015.

[12] Ioffe S, Szegedy C. Batch Normalization: Accelerating deep network training by reducing internal covariate shift. In International Conference on Machine Learning(ICML), 2015, 448–

456.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In IEEE International Conference on Computer Vision(ICCV), pages 1026–1034, 2015.

[14] Hu J, Shen L, Sun G. Squeeze-and-excitation networks. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, 7132–7141.

[15] Grauman K, Darrell T. The pyramid match kernel: Discriminative classification with sets of image features. In IEEE International Conference on Computer Vision(ICCV), 2005, 1458–1465.

[16] Lazebnik S, Schmid C, Ponce J. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2006, 2169–2178.

[17] He KM, Zhang XY, Ren SQ, Sun J. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence(IEEE TPAMI), 2015, 37(9):1904–1916.

[18] Woo S, Park J, Lee JY, Kweon IS. CBAM: Convolutional block attention module. In European Conference on Computer Vision(ECCV), 2018, 1–14

[19] Y. Ioannou, D. Robertson, R. Cipolla, A. Criminisi, et al. Deep roots: Improving CNN efficiency with hierarchical filter groups. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1231–1240, 2017.

[20] George E Dahl, Jack W Stokes, Li Deng, and Dong Yu. Large-scale malware classification using random projections and neural networks. In Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, pages 3422–3426. IEEE, 2013.

[21] Nair V, Hinton GE. Rectified linear units improve restricted boltzmann machines. In International Conference on Machine Learning(ICML), 2010, 807–814.

[22] He KM, Gkioxari G, Dollar P, Girshick G. Mask R-CNN. In IEEE International Conference on Computer Vision(ICCV), 2017, 2980–2988.

[23] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[24] Joshua Saxe and Konstantin Berlin. Deep neural network based malware detection using two dimensional binary program features. In Malicious and Unwanted Software (MALWARE), 2015 10th International Conference on, pages 11–20. IEEE, 2015.

[25] Ruimin Sun, Xiaoyong Yuan, Pan He, Qile Zhu, Aokun Chen, Andre Gregio, Daniela Oliveira,

and Li Xiaolin. Learning fast and slow: Propedeutica for real-time malware detection. arXiv preprint arXiv:1712.01145, 2017.

[26] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Doll ´ar, and C. L. Zitnick. Microsoft COCO: Common objects in context. In European Conference on Computer Vision(ECCV), 2014, 740–755.

[27] Zhenlong Yuan, Yongqiang Lu, Zhaoguo Wang, and Yibo Xue. Droidsec: deep learning in android malware detection. In ACM SIGCOMM Computer Communication Review, volume 44, pages 371–372. ACM, 2014.

[28] Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David Wagner, and Wenchao Zhou. Hidden voice commands. In USENIX Security Symposium, pages 513–530, 2016.

[29] Davide Castelvecchi. Can we open the black box of AI? Nature News, 538(7623):20, 2016.

[30] Marco Barreno, Blaine Nelson, Anthony D Joseph, and JD Tygar. The security of machine learning. Machine Learning, 81(2):121–148, 2010.

[31] Battista Biggio, Giorgio Fumera, and Fabio Roli. Multiple classifier systems for robust classifier design in adversarial environments. International Journal of Machine Learning and Cybernetics, 1(1-4):27–41, 2010.

[32] Nilesh Dalvi, Pedro Domingos, Sumit Sanghai, and Deepak Verma. Adversarial classification. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 99–108. ACM, 2004.

[33] Yinpeng Dong, Hang Su, Jun Zhu, and Fan Bao. Towards interpretable deep neural networks by leveraging adversarial examples. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. ImageNet large scale visual recognition challenge. Int. J. Comput. Vision, 115(3):211–252, 2015.

[35] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Srndic, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasionattacks against machine learning at test time. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pages 387–402. Springer, 2013.

[36] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. AISEC, 2017.

[37] Ivan Evtimov, Kevin Eykholt, Earlence Fernandes, Tadayoshi Kohno, Bo Li, Atul Prakash, Amir Rahmati, and Dawn Song. Robust physical-world attacks on deep learning models. arXiv preprint arXiv:1707.08945, 1, 2017.

[38] Nicholas Carlini and David Wagner. Magnet and efficient defenses against adversarial attacks are not robust to adversarial examples. arXiv preprint arXiv:1711.08478, 2017.

[39] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[40] Zhengli Zhao, Dheeru Dua, and Sameer Singh. Generating natural adversarial examples. arXiv preprint arXiv:1710.11342, 2017.

[41] 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. arXiv preprint arXiv:1708.03999, 2017.

[42] Jeremy R Flynn, Steve Ward, Julian Abich, and David Poole. Image quality assessment using the ssim and the just noticeable difference paradigm. In International Conference on Engineering Psychology and Cognitive Ergonomics, pages 23–30. Springer, 2013.

[43] S. Xie, R. Girshick, P. Doll ́ar, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 5987–5995, 2017.

[44] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In Security and Privacy (S&P), 2017 IEEE Symposium on, pages 39–57. IEEE, 2017.

[45] Dong Yinpeng and Liao Fangzhou. Boosting Adversarial Attacks with Momentum. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[46] Alex Graves, Abdel-Rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In Acoustics, speech and signal processing (ICASSP), 2013 IEEE International Conference on, pp. 6645–6649. IEEE, 2013.

[47] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare. Human-level control through deep reinforcement learning. Nature, 518(7540):529, 2015.

[48] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572, 2014.

[49] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In International Conference on

Learning Representations, 2018.

[50] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, pp. 506–519. ACM, 2017.

[51] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In International Conference on Learning Representations, 2017.

[52] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In 2016 IEEE Symposium on Security and Privacy (SP), pp. 582–597. IEEE, 2016.

[53] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. In International Conference on Learning Representations, 2018.

[54] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Advances in neural information processing systems, pp. 2672–2680, 2014.

[55] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, 2018.

[56] Shumeet Baluja and Ian Fischer. Adversarial transformation networks: Learning to generate adversarial examples. In AAAI Conference on Artificial Intelligence, 2018.

[57] Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. arXiv preprint arXiv:1803.06373, 2018.

[58] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. In Advances in Neural Information Processing Systems, 2018.

[59] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. arXiv preprint arXiv:1805.12152, 2018.

[60] Yanpei Liu, Xinyun Chen, Chang Liu, and DawnSong. Delving into transferable adversarial examples and black-box attacks. In International Conference on Learning Representations, 2017.

[61] Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry PVetrov, and Andrew Gordon Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. arXiv preprintar Xiv:1802. 10026, 2018.

[62] Felix Draxler, Kambis Veschgini, Manfred Salmhofer, and Fred A Hamprecht. Essentially no barriers in neural network energy landscape. In International Conference on Machine Learning, 2018.

[63] Pravendra Singh, Vinay Kumar Verma, Piyush Rai and Vinay P. Namboodiri. HetConv: HeterogeneousKernel-BasedConvolutionsforDeepCNNs. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[64] A.G.Howard,M.Zhu,B.Chen,D.Kalenichenko,W.Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, 2017.

[65] X. Zhang, X. Zhou, M. Lin, and J. Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 6848–6856, 2018.

[66] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1251–1258, 2017.

[67] Chen L C , Papandreou G , Kokkinos I , et al. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2016, 40(4):834-848.

[68] Chen L C , Papandreou G , Schroff F , et al. Rethinking Atrous Convolution for Semantic Image Segmentation. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[69] Will Knight. The dark secret at the heart of ai. MIT Technology Review, 2017.

[70] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. Proceedings of the International Conference on Machine Learning (ICML), 2017.

[71] Yani Ioannou, Duncan Robertson, Roberto Cipolla, Antonio Criminisi. Deep Roots: Improving CNN Efficiency with Hierarchical Filter Groups. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[72] Song Han, Jeff Pool,Sharan Narang,et al. DSD: Regularizing Deep Neural Networks with Dense-Sparse-Dense Training Flow. Proceedings of the International Conference on Learning Representations (ICLR), 2017

[73] Aaditya Prakash, James Storer, Dinei Florencio, Cha Zhang. RePr: Improved Training of Convolutional Filters. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[74] Weiyang Liu, Yandong Wen, Zhiding Yu, Meng Yang. Large-Margin Softmax Loss for

Convolutional Neural Networks. International Conference on Machine Learning (ICML), 2016.

[75] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533, 2016.

[76] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. Proceedings of the International Conference on Learning Representations (ICLR), 2017.

[77] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, Le Song. SphereFace: Deep Hypersphere Embedding for Face Recognition. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[78] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. CosFace: Large Margin Cosine Loss for Deep Face Recognition. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[79] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 580–587, 2014.

[80] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu. Object detection with deep learning: A review. IEEE Transactions on Neural Networks and Learning System, 2019.

[81] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3431–3440, 2015.

[82] J. Kim, A.-D. Nguyen, and S. Lee. Deep CNN-based blind image quality predictor. IEEE Transactions on Neural Networks and Learning System, 30(1):11–24, 2018.

[83] J. Cheng, J. Wu, C. Leng, Y. Wang, and Q. Hu. Quantized CNN: a unified approach to accelerate and compress convolutional networks. IEEE Transactions on Neural Networks and Learning System, 29(10):4730–4743, 2017.

[84] Zachary C Lipton. The mythos of model interpretability. International Conference on Machine Learning (ICML) Workshop, 2016.

[85] A. Aimar, H. Mostafa, E. Calabrese, A. Rios-Navarro, R. TapiadorMorales, I.-A. Lungu, M. B. Milde, F. Corradi, A. Linares-Barranco, S.-C. Liu, et al. Nullhop: A flexible convolutional neural network accelerator based on sparse representations of feature maps. IEEE Transactions on Neural Networks and Learning System, 30(3):644–656, 2018.

[86] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. Proceedings of the International Conference on Learning

Representations (ICLR), 2017.

[87] R. J. Cintra, S. Duffner, C. Garcia, and A. Leite. Low-complexity approximate convolutional neural networks. IEEE Transactions on Neural Networks and Learning System, 29(12):5981–5992, 2018.

[88] Daniel Lowd and Christopher Meek. Adversarial learning. In Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, pages 641–647. ACM, 2005.

[89] Jiajun Lu, Theerasit Issaranon, and David Forsyth. Safetynet: Detecting and rejecting adversarial examples robustly. In IEEE International Conference on Computer Vision(ICCV), 2017.

[90] N. Passalis and A. Tefas. Training lightweight deep convolutional neural networks using bag-of-features pooling. IEEE Transactions on Neural Networks and Learning System, 30(6):1705–1715, 2018.

[91] Ross Girshick. Fast R-CNN. In IEEE International Conference on Computer Vision(ICCV), 2015.

[92] Shang-Hua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, Philip Torr . Res2Net: A New Multi-scale Backbone Architecture. IEEE Transactions on Pattern Analysis and Machine Intelligence(IEEE TPAMI), 2019.

[93] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang. Residual attention network for image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3156–3164, 2017.

[94] Guoming Zhang, Chen Yan, Xiaoyu Ji, Taimin Zhang, Tianchen Zhang, and Wenyuan Xu. Dolphinatack: Inaudible voice commands. arXiv preprint arXiv:1708.09537, 2017.

[95] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In International Conference on Computer Vision. IEEE, 2017.

[96] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[97] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2574–2582, 2016.

[98] Yi Wu, David Bamman, and Stuart Russell. Adversarial training for relation extraction. In

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1779–1784, 2017.

[99] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 427–436, 2015.

[100] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. arXiv preprint arXiv:1705.07204, 2017.

[101] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In Security and Privacy (EuroS&P), 2016 IEEE European Symposium on, pages 372–387. IEEE, 2016.

[102] Pedro Tabacof and Eduardo Valle. Exploring the space of adversarial images. In Neural Networks (IJCNN), 2016 International Joint Conference on, pages 426–433. IEEE, 2016.

[103] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199, 2013.

[104] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. arXiv preprint arXiv:1703.00810, 2017.

[105] Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradientbased localization. arXiv preprint arXiv:1610.02391, 2016.

[106] Sara Sabour, Yanshuai Cao, Fartash Faghri, and David J Fleet. Adversarial manipulation of deep representations. Proceedings of the International Conference on Learning Representations (ICLR), 2016.

[107] Andras Rozsa, Ethan M Rudd, and Terrance E Boult. Adversarial diversity and hard positive generation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, pages 25–32, 2016.

[108] Fabio Roli, Battista Biggio, and Giorgio Fumera. Pattern recognition systems under attack. In Iberoamerican Congress on Pattern Recognition, pages 1–8. Springer, 2013.

致谢

度过了三年匆匆的研究生生活，仍旧有一种意犹未尽的感觉，对于数不胜数的未知知识，心里非常不甘。首先得感谢社会的发展与祖国的强盛，让越来越多的人可以享受学习的美妙。我便是这么多幸运儿中一个渺小的存在，能够从一无所知到窥得一斑。但是我深知，没有老师的指导，师兄的帮助，朋友的鼓励，父母的支持，我绝无可能完成毕业论文的谢幕。

我要感谢谭晓阳老师。谭老师是一个名副其实的学术研究者，他对学术严谨、敬畏的态度深深的感染了我，同时也熏陶着他周围的人。谭老师身材高大威猛，但是对我们却和蔼可亲，保有耐心，我总能够从每周的讨论中得到收获，听谭老师的课，就是一场享受，谭老师严谨的推演，耐心的教导帮助了我和师兄们。

我要感谢我的师兄们。王宇辉师兄是一个踏实的学术研究者，从他那里我收获了很多知识，每当遇到瓶颈，总可以从他那里得到解决的秘方。张魏宁师兄是一个沉着冷静的学术追求者，从他那里我学到了举一反三的意义。张文师兄是一个真真实实的技术控，遇到软件、开发环境问题总可以从他那里得到解决。宋歌师兄健康的生活方式改变了我的作息，让我明白运动的重要意义。与金鑫、王冬师兄交流，总可以开拓新的眼界。孙强与魏文戈师兄拥有踏实的工程实践能力，他们教会了我如何去编码。

我要感谢我的朋友。我要感谢王荣达和崔国华。希望我的朋友崔国华毕业后一帆风顺，王荣达在学术的道路上开满花。与你们相识甚是幸运，望来年常相聚。

我要感谢我的父母。在我失落的时候，父母的电话总会给予我振奋的力量，在我开心的时候，总能分享我的喜悦，在我做出决定的时候，总会帮我出谋划策。愿你们身体健康。

在学期间的研究成果及发表的学术论文

攻读硕士学位期间发表（录用）论文情况

1. 谢烟平, 谭晓阳. 基于信息修正的深度残差学习. 《数据采集与处理》. 2019（已录用）.

攻读硕士学位期间发表（在审）论文情况

1. Yanping Xie, Xin Jin, Xiu-Shen Wei, Borui Zhao, and Xiaoyang Tan. A lightweight encoder-decoder path for deep residual networks. IEEE Transactions on Neural Networks and Learning System (IEEE TNNLS).（在审）.
2. Yanping Xie, Xin Jin, Xiu-Shen Wei, Borui Zhao, and Xiaoyang Tan. Delving deep into spatial pooling for Squeeze-and-Excitation networks. IEEE Transactions on Neural Networks and Learning System (IEEE TNNLS).（在审）.

攻读硕士学位期间竞赛获奖情况

1. 第六届 FGVC 大赛 iNaturalist 赛道冠军. (CVPR Workshops).
2. 第六届 FGVC 大赛 Herbarium 赛道冠军. (CVPR Workshops).

攻读硕士学位期间专利情况

1. 谭晓阳, 谢烟平. 基于两阶段信息融合的图像识别技术.