人工智能安全课程

第3讲：人工智能鲁棒性之对抗样本（二）

主讲人：王志波 杨子祺

浙江大学计算机科学与技术学院/网络空间安全学院

课程大纲

一 对抗样本检测

二 对抗样本防御

对抗样本检测分类

口 对抗样本检测概述

- 深度神经网络(DNN)在许多任务中取得了巨大的成功，但实验证明DNN会被对抗样本所欺骗，对抗样本是通过在正常样本中添加一些微小但有目的的扰动而产生的
- 随着DNN的发展与成功，对抗样本检测技术也获得了极大的关注
- 对抗样本检测技术早期的探索主要是对抗样本与原始样本的特征或数字特征之间的区别
- 近期学者将对抗样本检测技术与防御技术相结合，通过神经网络中间状态的输出作为检测器的输入，从而检测出对抗样本

对抗样本检测分类

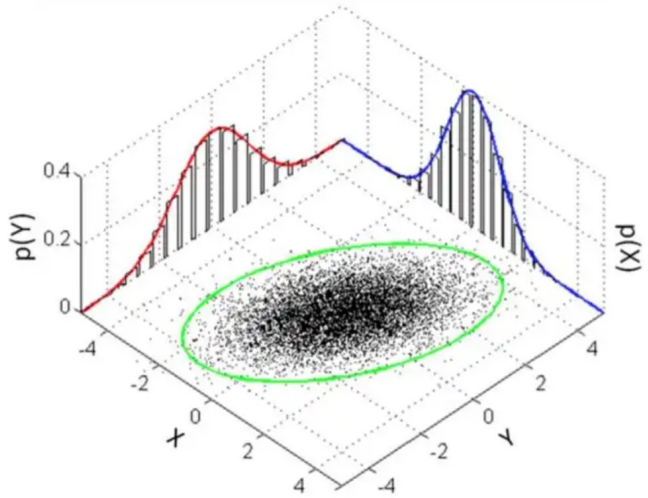
口 对抗样本检测方法分类

- 基于特征学习的对抗样本检测
  - 特征压缩
- 基于分布统计的对抗样本检测
  - softmax分布
- 基于中间输出的对抗样本检测
  - 对抗检测网络

基于特征学习的对抗样本检测

主要思想

- 特征学习：通过模型来学习，选择和抽取特征
- 基于特征学习的对抗样本检测主要利用对抗样本与原始样本的不同特征来进行对抗样本检测
- 在高维的数据下往往难以得到较好的特征学习结果来检测对抗样本，可以通过降维将高维的复杂数据转化为低维数据，降低特征学习的难度



数据降维

特征压缩

主要思想

- 输入样本的特征空间越大，产生对抗样本的可能性越大
- 特征压缩的目标是从输入中删除不必要的特征，在不损害分类器准确度的情况下减小对抗样本产生的可能性

主要方法

- 色深压缩
- 空间平滑

参考文献：[1] W. Xu, D. Evans, and Y. Qi, “Feature squeezing: Detecting adversarial examples in deep neural networks,” NDSS, 2017.

特征压缩

色深

- 数字图像中，最小的单位叫“像素”（Pixel），每一个像素都有自己独立的参数：
  - 在灰度图中（如MNIST数据集），每个像素有唯一确定的值来表示自己的颜色
  - 在RGB三通道图中（如CIFAR-10和ImageNet数据集），每一个像素都由R，G，B三个通道组成，其中每个通道又有确定的值来表示其颜色
- 色深为像素点可以取值颜色的大小，一般用位表示色深大小，色深为i位表示像素点取值的颜色有2^i种
- MNIST为8位的灰度图：即MNIST数据集的图像每个像素点的取值在$0\sim2^8-1$，0代表黑色，255代表白色，中间数值是渐变的灰色
- CIFAR-10和ImageNet为24位的彩色图，即每个像素含有R，G，B三个通道的图片，每个通道的值取值在$0\sim2^8-1$，故一个像素点所有的颜色取值范围在$0\sim2^{24}-1$

参考文献：[1] W. Xu, D. Evans, and Y. Qi, “Feature squeezing: Detecting adversarial examples in deep neural networks,” NDSS, 2017.

特征压缩

色深压缩的原因

- 较大的色深是会使得图片更接近于自然图像，但是也会引入很多不必要的特征
- 较大的色深对于图片的分类来说不是必须的，比如人眼可以很准确地分类出黑白图像，对于模型来说也是如此
- 减小色深就是在减小特征空间，以MNIST数据集为例，将MNIST数据集中的8位的图像色深减小到1位，那么特征空间将减小到原来的$1/128$

- 将MNIST中的“0”类色深压缩，色深由8位压缩到1位

参考文献：[1] W. Xu, D. Evans, and Y. Qi, “Feature squeezing: Detecting adversarial examples in deep neural networks,” NDSS, 2017.

特征压缩

色深压缩的实现

- 一般的模型在处理图片数据时为了减小方差会将图片的像素数值缩放到$[0,1]$
- 假设我们要将原来8位的色深压缩到i位（$1 \leq i \leq 7$）
- 将原像素处理后的$[0,1]$的值乘以$2^i - 1$，四舍五入（色深压缩操作）到整数后再除以$2^i - 1$

- 将CIFAR-10（第一行）和ImageNet（第二行）色深压缩，每个通道由8位压缩到1位

参考文献：[1] W. Xu, D. Evans, and Y. Qi, “Feature squeezing: Detecting adversarial examples in deep neural networks,” NDSS, 2017.

特征压缩

空间平滑

- 空间平滑（或称模糊）是一种广泛应用于图像处理的技术，目的是减少图像噪声

主要方法

- 局部平滑
- 非局部平滑

参考文献：[1] W. Xu, D. Evans, and Y. Qi, “Feature squeezing: Detecting adversarial examples in deep neural networks,” NDSS, 2017.

特征压缩

局部平滑

- 局部平滑方法利用邻近像素对每个像素进行平滑

主要方法

- 中值平滑
  - 取一个方形窗口对于整张图片上下移动，每次移动时，取出小窗内所有像素点的中位数，使用这个中位数代替中间像素点的值
  - 使用像素点领域的值代替该点的值，可以消除孤立的噪声点

参考文献：[1] W. Xu, D. Evans, and Y. Qi, “Feature squeezing: Detecting adversarial examples in deep neural networks,” NDSS, 2017.

特征压缩

□ 非局部平滑

- 非局部平滑不同于局部平滑，非局部平滑在一个更大的区域中对相似的像素块进行平滑，而不仅仅是邻近的像素
- 对于给定的图像，非局部平滑算法在图像的大面积区域内发现多个相似的像素块，并用这些相似像素块的平均值代替中心像素块
- 假设噪声的平均值为零，将相似的像素块求平均可以消除噪声

原图　　非局部平滑后的图

参考文献：[1] W. Xu, D. Evans, and Y. Qi, “Feature squeezing: Detecting adversarial examples in deep neural networks,” NDSS, 2017.

特征压缩

特征压缩方法在检测中的使用

- 使用不同的特征压缩方法（色深压缩）来压缩图片
- 使用不同的方法各自压缩图片，得到多张由不同方法与不同压缩程度的图片
- 将得到的特征压缩后的图片分别输入原模型中得到各自的输出
- 比较原图片输出与特征压缩后图片的差距，选择其中的最大值作为两者的最终差距并进行对抗样本的判定
- 不同的数据在不同特征压缩下的效果有区别，故需要比较多个特征压缩后的图片

参考文献：[1] W. Xu, D. Evans, and Y. Qi, “Feature squeezing: Detecting adversarial examples in deep neural networks,” NDSS, 2017.

特征压缩

检测方法的主要思想

- 将输入与特征压缩后的输入放入模型中，对于非对抗性的样本来说，模型的两次预测应该是相似的，反之则会产生较大的变化
- 度量函数：

$$score^{(\mathbf{x},\mathbf{x}_{squeezed})} = \|g(\mathbf{x}) - g(\mathbf{x}_{squeezed})\|_1$$

$score$：衡量图片压缩前后模型输出的差距　　$g(x)$：对于输入x 模型的输出

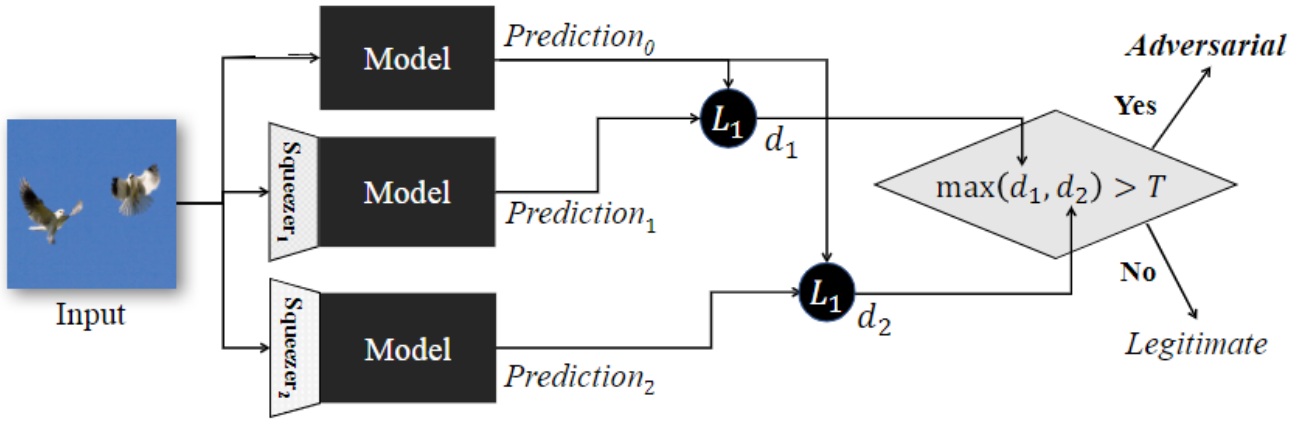
x：原始图片　　$x_{squeezed}$：特征压缩后的图片

- 检测的有效性取决于选择一个能够准确区分合法样本和对抗样本的阈值

参考文献：[1] W. Xu, D. Evans, and Y. Qi, “Feature squeezing: Detecting adversarial examples in deep neural networks,” NDSS, 2017.

特征压缩

检测过程



- 将模型原始输入经过不同方法的特征压缩后再通过模型得到不同的预测值
- 对比特征压缩后的预测值与原始图像输出之间的差距
- 选取最大的那个差距值与预先设定的阈值做对比，大于阈值则认为该样本为对抗样本，否则为正常样本

参考文献：[1] W. Xu, D. Evans, and Y. Qi, “Feature squeezing: Detecting adversarial examples in deep neural networks,” NDSS, 2017.

特征压缩

□ 特征压缩对抗样本检测结果（以MNIST为例）

	Configuration			L_∞ Attacks				L_2 Attacks			L_0 Attacks				Overall Detection Rate
	Squeezer	Parameters	Threshold	FGSM	BIM	CW_∞		Deep Fool	CW_2		CW_0		JSMA		
						Next	LL		Next	LL	Next	LL	Next	LL	
MNIST	Bit Depth	1-bit	0.0005	**1.000**	**0.979**	**1.000**	**1.000**	-	**1.000**	**1.000**	0.556	0.563	**1.000**	**1.000**	0.903
		2-bit	0.0002	0.615	0.064	0.615	0.755	-	0.963	0.958	0.378	0.396	0.969	**1.000**	0.656
	Median Smoothing	2x2	0.0029	0.731	0.277	**1.000**	**1.000**	-	0.944	**1.000**	0.822	0.938	0.938	**1.000**	0.868
		3x3	0.0390	0.385	0.106	0.808	0.830	-	0.815	0.958	0.889	**1.000**	0.969	**1.000**	0.781
	Best Attack-Specific Single Squeezer		-	**1.000**	**0.979**	**1.000**	**1.000**	-	**1.000**	**1.000**	0.889	**1.000**	**1.000**	**1.000**	-
	Best Joint Detection (1-bit, 2x2)		0.0029	**1.000**	**0.979**	**1.000**	**1.000**	-	**1.000**	**1.000**	**0.911**	0.938	**1.000**	**1.000**	**0.982**

- 图为MNIST数据集上不同的特征压缩方法对于不同攻击方法产生的对抗样本的检测成功率
- 对于不同的对抗样本，基于特征压缩的检测方法都有较好的检测成功率，将两种性能较好的特征压缩方法结合时，可以得到平均高达98.2%的检测成功率
- 在一定范围内，压缩的程度越大，对抗样本的检测成功率越高

参考文献：[1] W. Xu, D. Evans, and Y. Qi, “Feature squeezing: Detecting adversarial examples in deep neural networks,” NDSS, 2017.

基于分布统计的对抗样本检测

主要思想

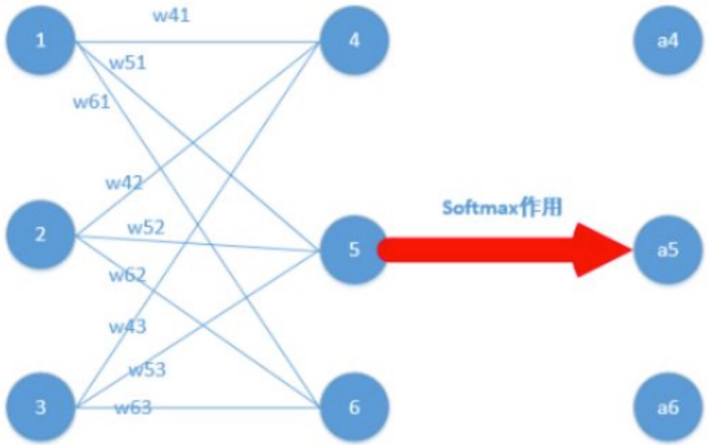
- 分布统计的核心思想是利用对抗样本与原始样本的不同数字特征（即样本通过网络后得到的概率分布的形状），通过检测输入是否符合正常样本的分布，从而判断输入是否具有对抗性

基于softmax分布的对抗样本检测的主要思想

- 分类网络的softmax输出向量可以很好地表现样本的数字特征分布
- 许多对抗样本的与正常样本具有差距较大的softmax输出向量，正常样本的softmax输出向量会比对抗样本更加分散（远离均匀分布），因此可以通过检测样本的softmax输出向量的分散程度确定样本是否具有对抗性

softmax分布

softmax函数



- softmax函数用于多分类过程中，它将多个神经元的输出，映射到（0,1）区间内
- softmax的输出可以看成概率来理解，第i个元素输出代表样本属于第i类的概率
- softmax函数计算方式：$S_i = \frac{e^i}{\sum_j e^j}$
- 其中等式左边的S_i表示第i个元素的输出值，等式右边的i表示经过softmax函数前的第i个神经元的值，分母为softmax函数前所有神经元经过指数函数后的累加和

参考文献：[1] “A baseline for detecting misclassified and out-of-distribution examples in neural networks,” ICLR, 2017.

softmax分布

基于softmax分布的对抗样本检测原理

- 正常样本往往比对抗样本的softmax向量中的最大概率值更大
- 与对抗样本相比，正常样本的softmax分布通常更远离均匀分布，即与均匀分布的差距更大，因为对抗样本一般只关注最大的概率值而忽视其他类的输出概率
- 对于一些希望在决策边界停止扰动（即添加的扰动越小越好）的对抗攻击来说，可以通过softmax分布的不同很有效地区分出对抗样本

参考文献：[1] “A baseline for detecting misclassified and out-of-distribution examples in neural networks,” ICLR, 2017.

softmax分布

□ 概率差异的度量：KL散度

$$D_{KL}(p\|q) = \sum_{i=1}^{N} [p(x_i)\log p(x_i) - p(x_i)\log q(x_i)]$$

p, q：两个概率分布
N：类别总数

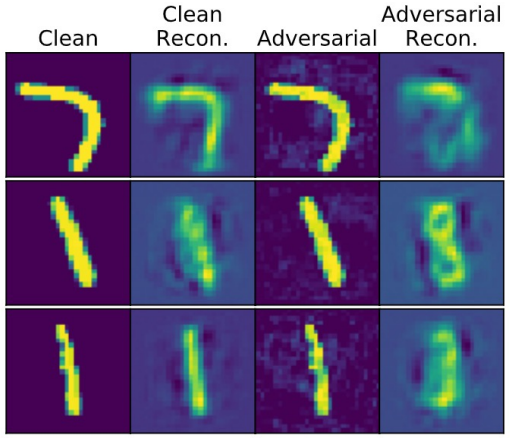
g(x)：对于输入x 模型的输出
x_i：第i个类别

- KL散度用于衡量两个概率分布的分散程度，分布差异越大，KL散度越大
- 对抗样本的检测可以通过测量均匀分布和样本softmax分布之间的KL散度来完成，然后根据预先设定的阈值，小于这个阈值则认为样本是对抗样本，反之认为是正常样本

参考文献：[1] “A baseline for detecting misclassified and out-of-distribution examples in neural networks,” ICLR, 2017.

softmax分布

口 基于softmax分布与输入重构的检测

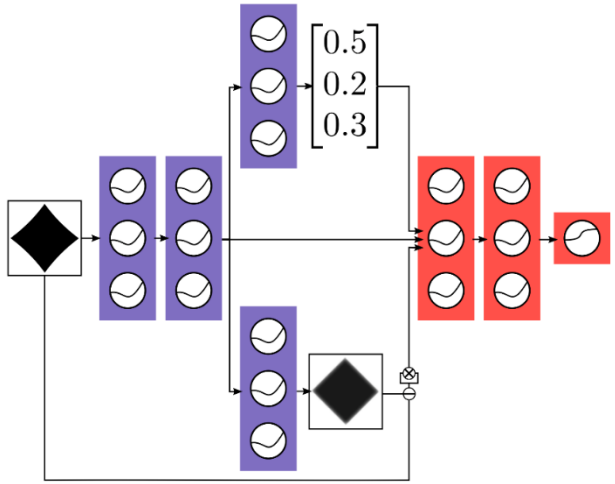


- 除了对于softmax的分布直接进行对比，研究者还提出了一种更有效的与输入重构相结合的方法
- 输入重构对于对抗样本检测的帮助：
- 如图，对抗样本与正常样本在原始分类网络中间层的输出进行重构后的图像有着明显的差距，对抗样本的重构图片相较于正常图片的重构图片更加不规则且更加模糊
- 通过对抗样本与正常样本输入重构的差异可以进行对抗样本的检测

参考文献：[1] “A baseline for detecting misclassified and out-of-distribution examples in neural networks,” ICLR, 2017.

softmax分布

基于softmax分布与输入重构的检测



- 检测流程如图，其中最右边的黑色菱形为原始输入图片
- 紫色部分属于神经元，在提取特征后进行分支处理
- 紫色的上半部分为普通的分类器，最终得到softmax的概率分布
- 紫色的下半部分进行输入重构，将重构后的图片与原图相减并将每个元素平方处理
- 红色部分训练一个二分类器，判别原始输入样本是否为对抗样本
- 红色部分的输入为神经网络提取的特征，输入重构的处理结果和 softmax 输出；并最终输出该样本为对抗样本的概率

参考文献：[1] “A baseline for detecting misclassified and out-of-distribution examples in neural networks,” ICLR, 2017.

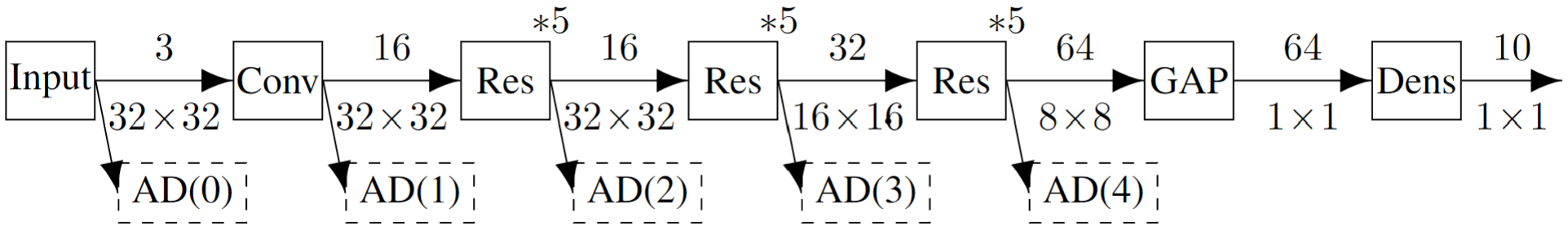
基于中间输出的对抗样本检测

基于中间输出的对抗样本检测概述

- 正常的样本与对抗样本的输入在深度神经网络中得到的中间输出状态有较大的差距，可以通过这一部分来检测对抗样本
- 基于中间输出的对抗样本检测的核心思想将深度神经网络的中间部分的输出作为检测器的输入，从而检测出对抗样本

对抗检测网络

对抗检测网络概述

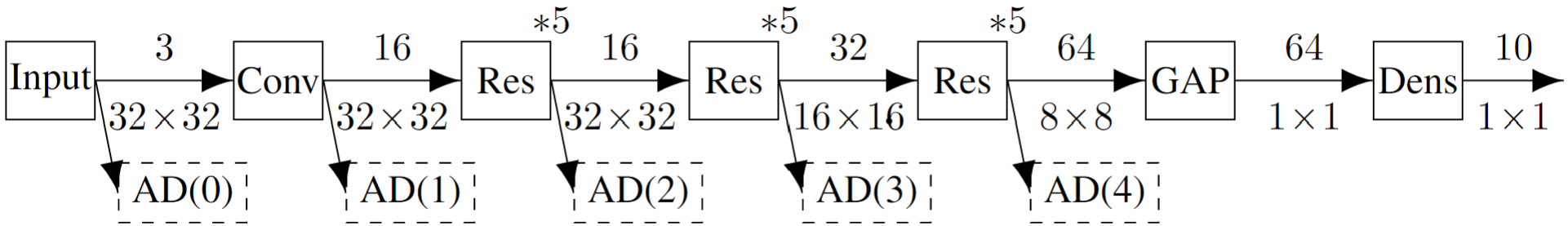


- 在原始分类网络的中间各层接入独立的训练好的分支网络（即图中的AD模块）作为检测器
- 检测器为一个训练好的二分类网络
- 检测器接收原网络的中间输出，并输出该样本为对抗样本的概率

参考文献：[1] J. H. Metzen, T. Genewein, V. Fischer, “On detecting adversarial perturbations,” ICLR, 2017.

对抗检测网络

□ 对抗检测网络使用（以ResNet上的使用为例）



AD：对抗检测网络

Res*5：序列长度为5的残差块

GAP：全局平均池化层

$Dens$：全连接层

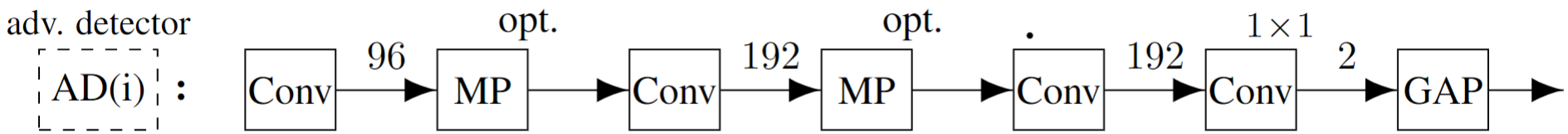
连接线上方的数字表示feature map的数目，下方的数字表示每个feature map的规格

- 在原始分类网络的不同层会存在有不同的对抗检测网络使用对应层的输出来判断样本是否为对抗样本

参考文献：[1] J. H. Metzen, T. Genewein, V. Fischer, “On detecting adversarial perturbations,” ICLR, 2017.

对抗检测网络

□ 对抗检测网络结构（以ResNet上的结构为例）



AD(i)：第i个对抗检测网络　　opt：表示这一块结构可选

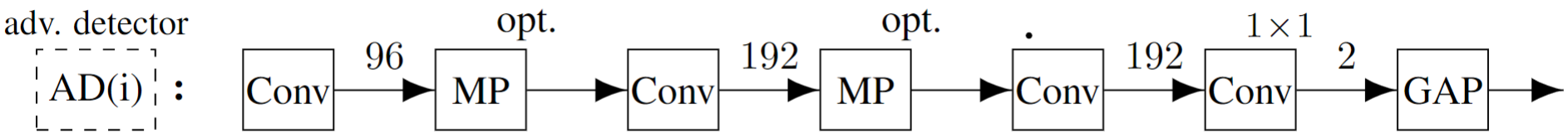
MP：最大化操作的的池化层

- 从原始分类网络不同层接出的对抗检测网络结构可以不同
- 对抗检测网络的训练根据不同的数据集与不同的原始分类网络而发生改变

参考文献：[1] J. H. Metzen, T. Genewein, V. Fischer, "On detecting adversarial perturbations," ICLR, 2017.

对抗检测网络

对抗检测网络训练

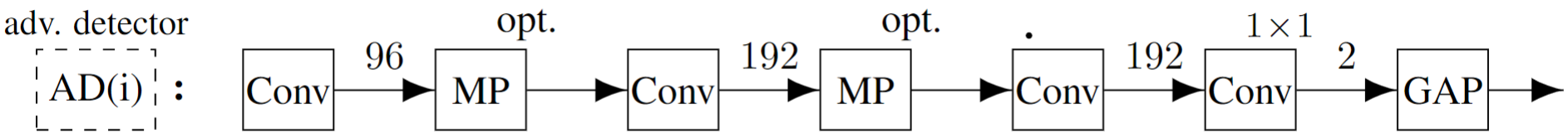


- 检测者已知的知识：原始分类网络结构，原始分类网络参数，原始训练数据，与训练数据集同分布的辅助数据集
- 训练集的获取：
- 从训练数据集与辅助数据集中选取图片，通过不同的对抗攻击方法得到不同的对抗样本
- 将正常样本与对抗样本分别输入原始分类网络中，得到原始分类网络与对抗检测网络接口处的原始分类网络的中间输出：正常样本数据（x_1，0)与对抗样本数据（x_2，1），收集这两种不同标签的数据作为这一接口处的对抗检测网络的训练数据集

参考文献：[1] J. H. Metzen, T. Genewein, V. Fischer, “On detecting adversarial perturbations,” ICLR, 2017.

对抗检测网络

对抗检测网络训练

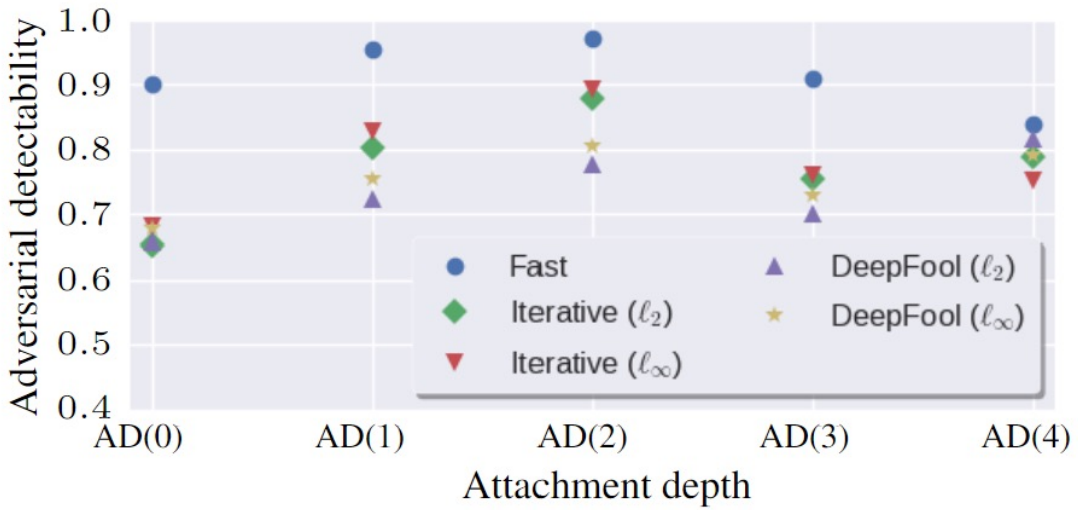


- 损失函数： $L = -[y log\ \hat{y} + (1-y) log\ (1-\hat{y})]$ ，y为原始标签，$\hat{y}$为预测结果；损失函数用来判定实际的输出与期望的输出的偏离程度，所以训练的目标就是最小化这个损失函数
- 训练方法：
- 通过监督学习的方法，每次将一个训练数据（x, y）输入对抗检测网络进行训练
- 在数据通过对抗检测网络后，得到预测结果$\hat{y}$，计算出这一轮损失函数L的值，使用随机梯度下降（SGD）的方法反向传播更新对抗检测网络的参数

参考文献：[1] J. H. Metzen, T. Genewein, V. Fischer, "On detecting adversarial perturbations," ICLR, 2017.

对抗检测网络

对抗检测网络实验结果



- 图为对于原始分类模型不同层的施加对抗检测网络后，不同对抗攻击方法下的对抗样本的检测成功率
- 不同深度的对抗检测网络对于检测不同攻击方法产生的对抗样本能力不同，故需要在原始模型的不同层的输出后都加上对抗检测网络

参考文献：[1] J. H. Metzen, T. Genewein, V. Fischer, "On detecting adversarial perturbations," ICLR, 2017.

课程大纲

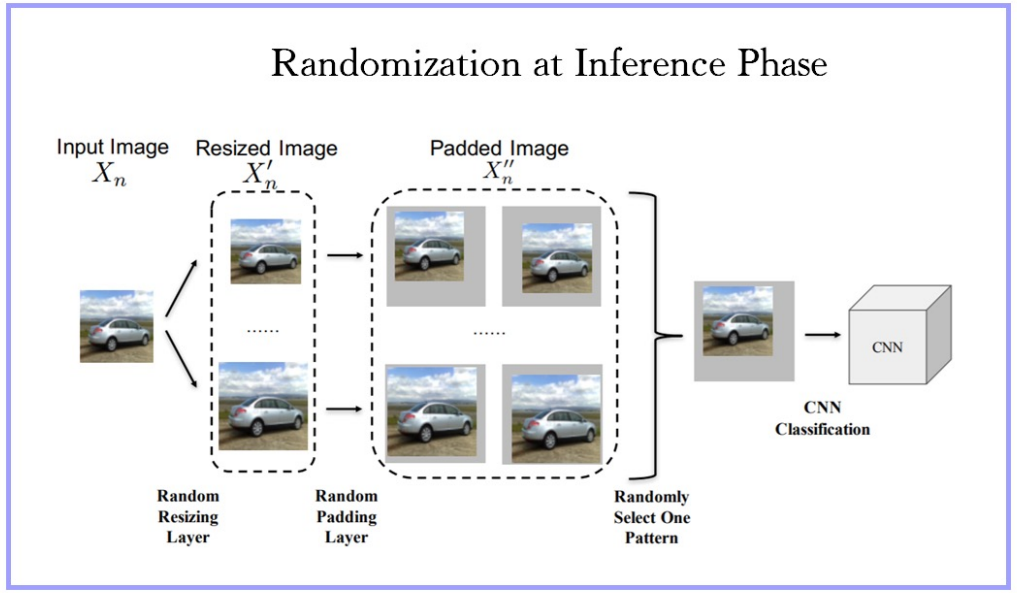
一 对抗样本检测

二 对抗样本防御

防御分类

基于经验的防御方法

- 主要破坏现有攻击方法的前置条件，增加其成本。例如攻击方法需要模型的完整输出，那他就将输出截断。如果攻击方法需要找梯度的最小值，它就在梯度上设置一些看起来像最小值的陷阱
- 非常依赖攻击方法条件，面对层出不穷的新式攻击易被攻破



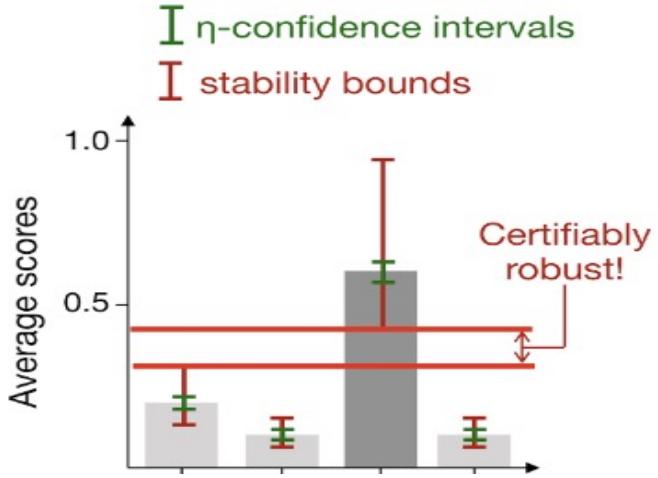
通过两种随机变换减轻推理时的对抗效果

参考文献：[1] 2018-Mitigating Adversarial Effects Through Randomization-Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, Alan Yuille

防御分类

基于理论的防御方法

- 不着眼于具体的攻击方法，而使将其抽象化。在实际操作时，攻击的方法被视为一个带范围约束的抽象操作，防御者只需使模型在抽象操作的范围内保持正确即防御对抗样本.
- 从理论的角度设计防御方式，为最坏情况提供下限，可防范现在尚未出现的攻击



对于任意的输入变换，约束分数可以改变的界限

参考文献：[1] 2019-Certified Robustness to Adversarial Examples with Differential Privacy- Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana

防御分类

基于经验的防御方法

- 对抗训练
- 特征去噪
- 防御蒸馏

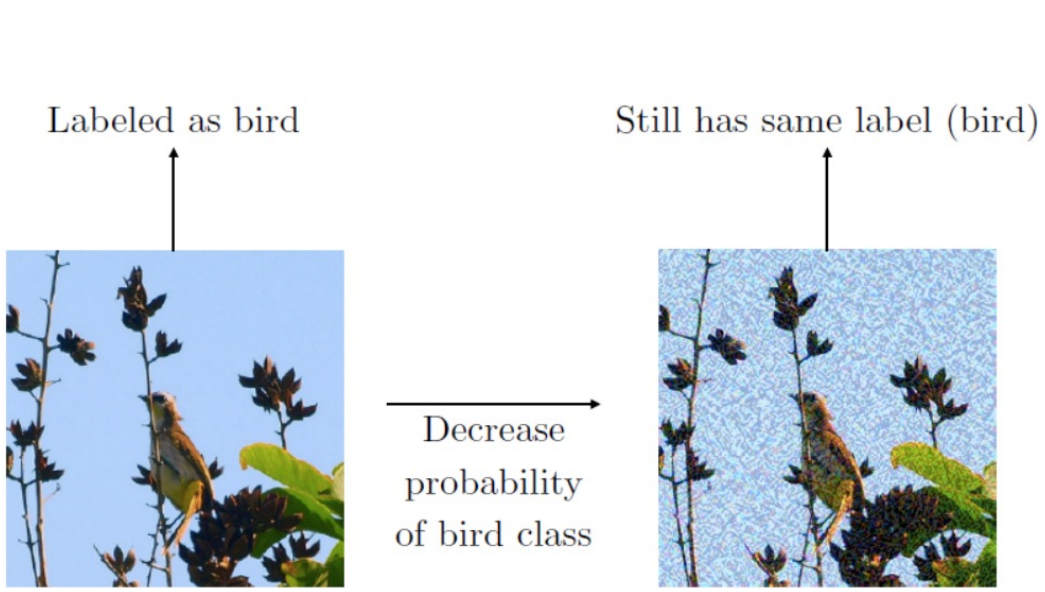
基于理论的防御方法

- 可证明式防御

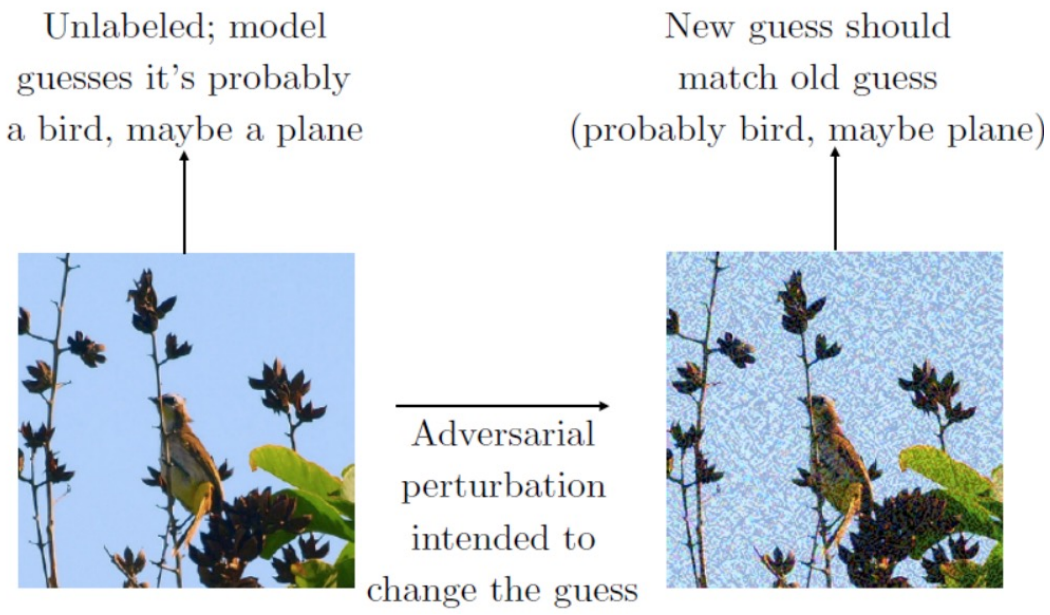
基于经验的防御方法

对抗训练

- 在模型的训练阶段，防御者可以主动生成对抗样本，纳入训练阶段对神经网络进行训练，构建鲁棒性更好的模型，达到防御对抗样本的目的。



有标签对抗训练



无标签对抗训练

基于经验的防御方法

不同的对抗训练

	生成对抗样本的攻击方法	训练样本	生成对抗样本的模型
FGSM对抗训练	FGSM	原始样本+对抗样本	目标模型
PGD对抗训练	PGD	对抗样本	目标模型
集成对抗训练（黑盒）	FGSM	对抗样本	预训练模型

- 目标模型：实施对抗攻击的模型
- 预训练模型：独立于目标模型而单独训练过的模型（结构可能和目标模型一样，也可能不一样）

基于经验的防御方法

FGSM对抗训练

- 主要思想：将生成的对抗样本加入到训练集中去，做一个数据增强，让模型在训练的时候就能够学习一遍对抗样本。
- 对抗训练目标函数为：

$$\tilde{J}(\boldsymbol{\theta}, \boldsymbol{x}, y) = \alpha J(\boldsymbol{\theta}, \boldsymbol{x}, y) + (1-\alpha) J(\boldsymbol{\theta}, \boldsymbol{x} + \epsilon \text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)), y)$$

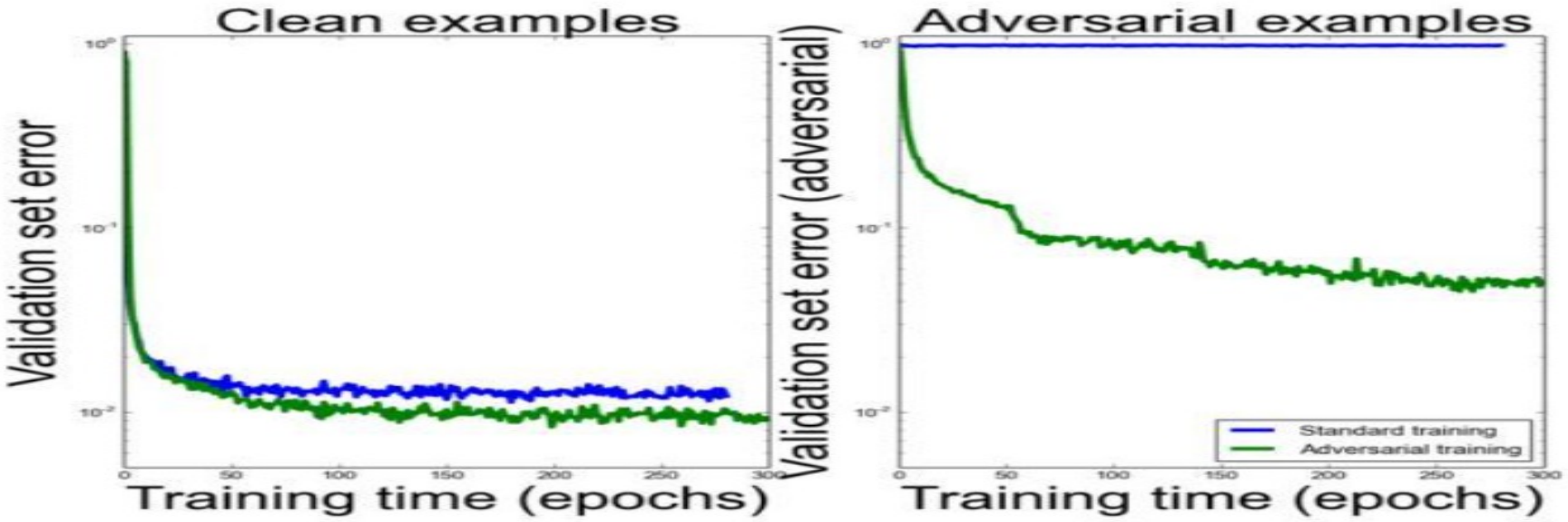
  - 式中 $x + \epsilon sign(\nabla_x J(\theta, x, y))$ 是良性样本 x 根据FGSM方法生成的对抗样本；
  - $J(\theta, x, y)$ 是识别网络（模型）的损失函数；$\tilde{J}(\boldsymbol{\theta}, \boldsymbol{x}, y)$是对抗网络的损失函数
  - α 是用于平衡良性和对抗样本的准确性
  - **优化目标**：对抗网络（模型）的损失函数最小，即对抗网络识别预测结果和真实结果尽可能接近
  - 这种对抗训练的方法意味着在训练过程中不断更新对抗样本，使他们抵抗当前版本的模型

参考文献：[1] 2015-Explaining and Harnessing Adversarial Examples- Ian J. Goodfellow, Jonathon Shlens, Christian Szegedy

基于经验的防御方法

FGSM对抗训练

- 训练效果并没有达到预期那么好，于是作者做了改进：
  - 增大网络的神经元个数240 →1600（如果没有对抗训练这可能会导致过拟合，此时测试集上错误率有1.14% ，因为模型越复杂容量越大，越容易过拟合）
- 结果：在MINST测试集上模型对 对抗样本的错误率从89.4%急剧下降至17.9%。
- 尽管该方法对FGSM的攻击有效，但是训练后的模型仍然容易受到基于迭代/优化方式的对抗攻击。



参考文献：[1] 2015-Explaining and Harnessing Adversarial Examples- Ian J. Goodfellow, Jonathon Shlens, Christian Szegedy

基于经验的防御方法

PGD对抗训练

- 传统对抗训练的期望损失函数：

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}[L(x,y,\theta)]$$

  - $\mathrm{E}_{(\mathrm{x,y})\sim\mathrm{D}}[\mathrm{L}]$：平均损失
  - 损失函数：度量模型一次预测的好坏
  - 风险函数/期望损失：度量平均意义下的模型预测好坏
- 传统的训练方法聚焦于提高某个特定方法的鲁棒性，通常整体对抗鲁棒性能很差
- 为了提高模型的对抗鲁棒性，需要适当的对范式进行扩展，提出一个具有通用性的范式：

$$\min_{\theta}\rho(\theta),\ \text{where}\quad \rho(\theta)=\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\max_{\delta\in\mathcal{S}}L(\theta,x+\delta,y)\right]$$

- 不同于传统训练 直接基于原始样本求目标函数，而是首先基于原始样本生成对抗样本，再基于对抗样本求解模型

参考文献：[1] 2018-Towards Deep Learning Models Resistant to Adversarial Attacks-Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, Adrian Vladu

基于经验的防御方法

PGD对抗训练

- 对抗训练可以概括为如下的最小最大化公式（ Min-Max最优化框架） [1]

$$\min_{\theta} \rho(\theta), \text{ where } \rho(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\max_{\delta\in\mathcal{S}} L(\theta, x+\delta, y)\right]$$

  - x ：样本的输入
  - y：样本的标签
  - δ：叠加在输入上的扰动
  - L：神经网络损失函数
- 内层$maxL$是攻击者的目标函数，旨在寻找损失函数最大的对抗样本。
- 外层$min\rho$是防御者的目标函数，即当扰动（对抗样本）固定的情况下，寻找损失函数最小的模型参数，使得该模型对这种扰动（对抗样本）有鲁棒性。
- 解决内层max和外层$min\rho$可以有不同的方法，目前大多数对抗训练工作都可以看作研究怎么去解决这两个子问题的

参考文献：[1] 2018-Towards Deep Learning Models Resistant to Adversarial Attacks-Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, Adrian Vladu

基于经验的防御方法

PGD对抗训练

- 该方法主要思想是“How can we train deep neural networks that are robust to adversarial inputs”
- 训练集：用对抗样本代替原始样本训练
- 针对内层max问题，提出PGD攻击

$$x^{t+1} = \Pi_{x+S}\left(x^t + \alpha \text{sign}(\nabla_x L(\theta, x, y))\right)$$

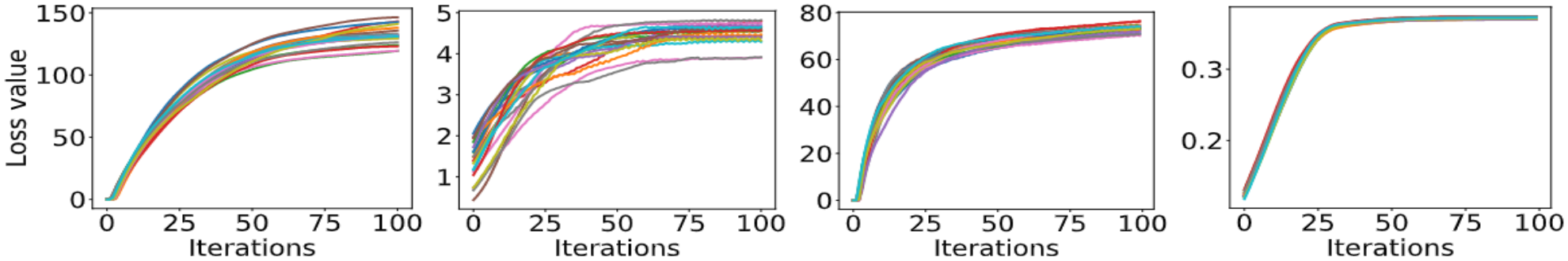
  - S：扰动的最大范围
  - α：每次移动的步长
  - $\prod$：投影操作
  - L：损失函数，其中L内的x指的是x^t
- PGD的目标是在扰动范围S的球体内寻找到一个点$x^* = x + \delta$，使得x^*处的损失函数的值最大
- 因为初始点（初始随机扰动）不同，会导致很多的局部最大值（生成一些很不同的对抗样本）。但是最后这些极大值点都会收敛到一个较集中的损失值。

参考文献：[1] 2018-Towards Deep Learning Models Resistant to Adversarial Attacks-Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, Adrian Vladu

基于经验的防御方法

PGD对抗训练

- 实验表明，不管是正常训练的模型还是对抗训练的模型，最终的loss都是很接近（集中）的，这个集中现象说明了一个问题： PGD是一阶方法中的“通用”攻击方法，即对于所有的一阶攻击方法，只要对PGD鲁棒，则对其它所有的方法也鲁棒。
  - 使用Cross-entropy loss 在 MNIST and CIFAR10数据集上创建对抗样本
  - 不同的颜色表示不同的初始条件，共有20组。



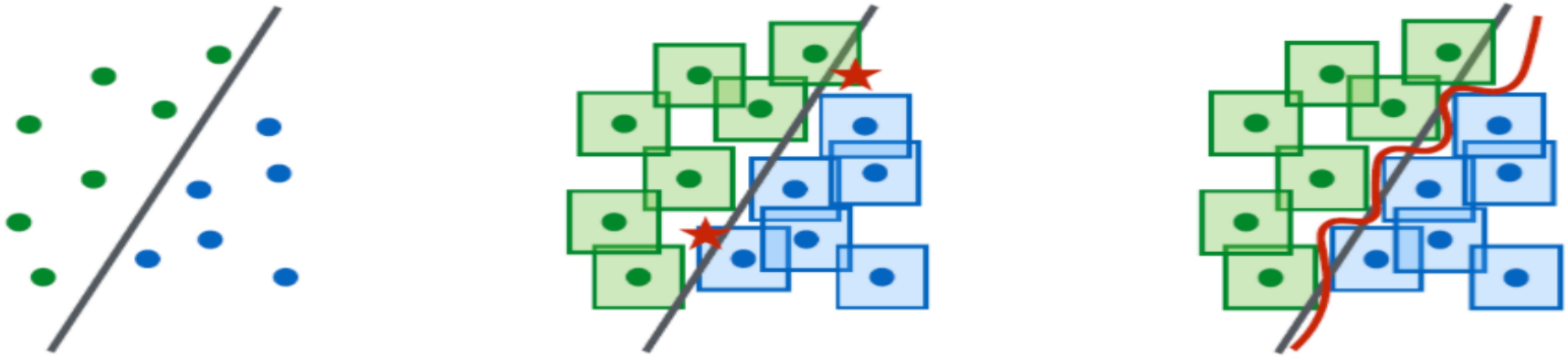
(a) MNIST Standard training　(b) MNIST Adversarial training　(c) CIFAR10 Natural training　(d) CIFAR10 Adversarial training

参考文献：[1] 2018-Towards Deep Learning Models Resistant to Adversarial Attacks-Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, Adrian Vladu

基于经验的防御方法

PGD对抗训练

- 对于训练神经网络而言，最终评价训练过程/网络性能的指标归根结底是loss value，即最终模型对对抗样本的损失（外层min 问题）。一般地，认为一个小的loss说明模型对对抗样本表现出鲁棒性。
- 训练神经网络的关键要素之一是模型复杂度。某种意义上来说，因为对抗样本的存在，将原始问题的边界变得更加复杂，也需要容量更大的模型。



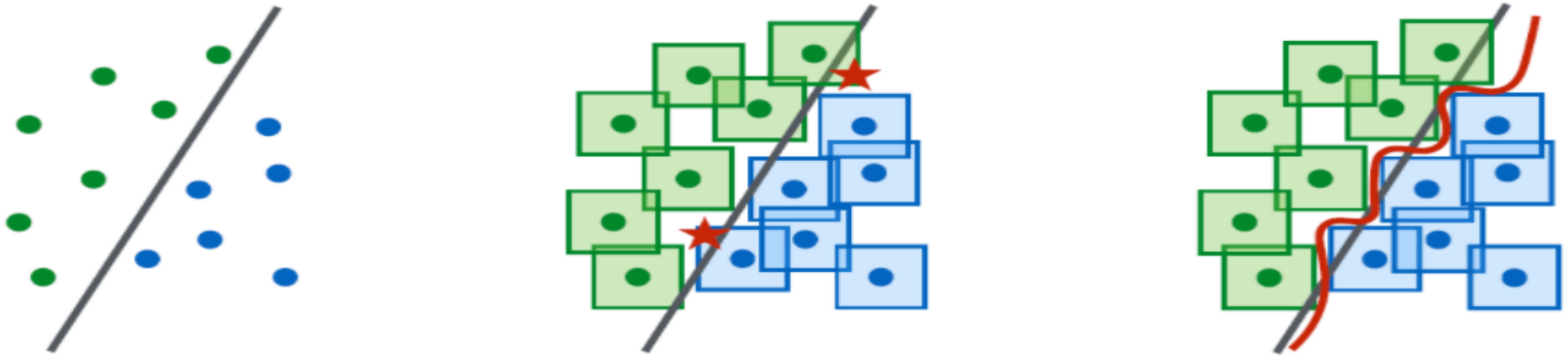
- 对于原始简单的样本，最左侧情况，此时决策边界也简单，则需要使用的模型架构/容量也简单

参考文献：[1] 2018-Towards Deep Learning Models Resistant to Adversarial Attacks-Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, Adrian Vladu

基于经验的防御方法

PGD对抗训练

- 对于训练神经网络而言，最终评价训练过程/网络性能的指标归根结底是loss value，即最终模型对对抗样本的损失（外层min 问题）。一般地，认为一个小的loss说明模型对对抗样本表现出鲁棒性。
- 训练神经网络的关键要素之一是模型复杂度。某种意义上来说，因为对抗样本的存在，将原始问题的边界变得更加复杂，也需要容量更大的模型。



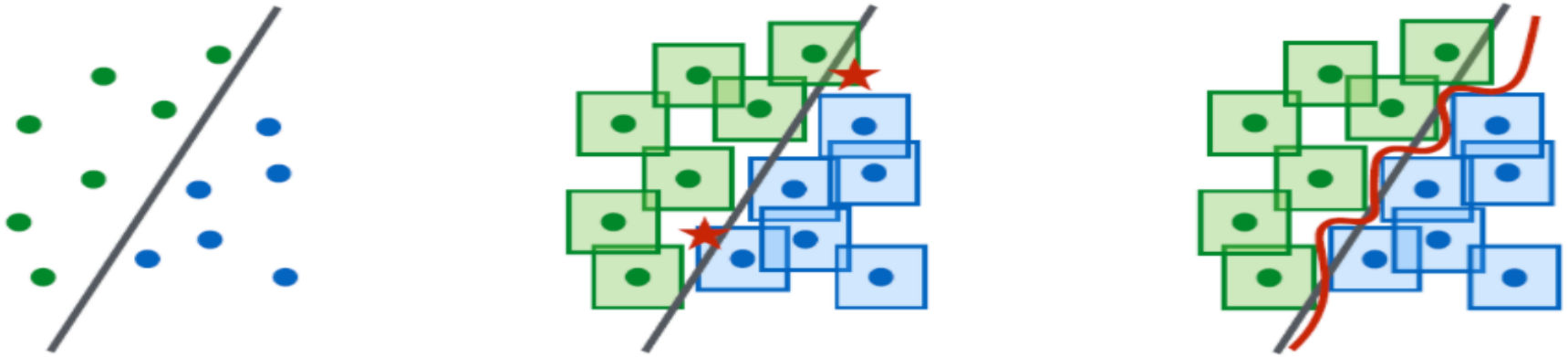
- 有对抗样本的情况，如中间图所示，此时如果仍然使用线性分类器，则不能很好的说明问题

参考文献：[1] 2018-Towards Deep Learning Models Resistant to Adversarial Attacks-Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, Adrian Vladu

基于经验的防御方法

PGD对抗训练

- 对于训练神经网络而言，最终评价训练过程/网络性能的指标归根结底是loss value，即最终模型对对抗样本的损失（外层min 问题）。一般地，认为一个小的loss说明模型对对抗样本表现出鲁棒性。
- 训练神经网络的关键要素之一是模型复杂度。某种意义上来说，因为对抗样本的存在，将原始问题的边界变得更加复杂，也需要容量更大的模型。



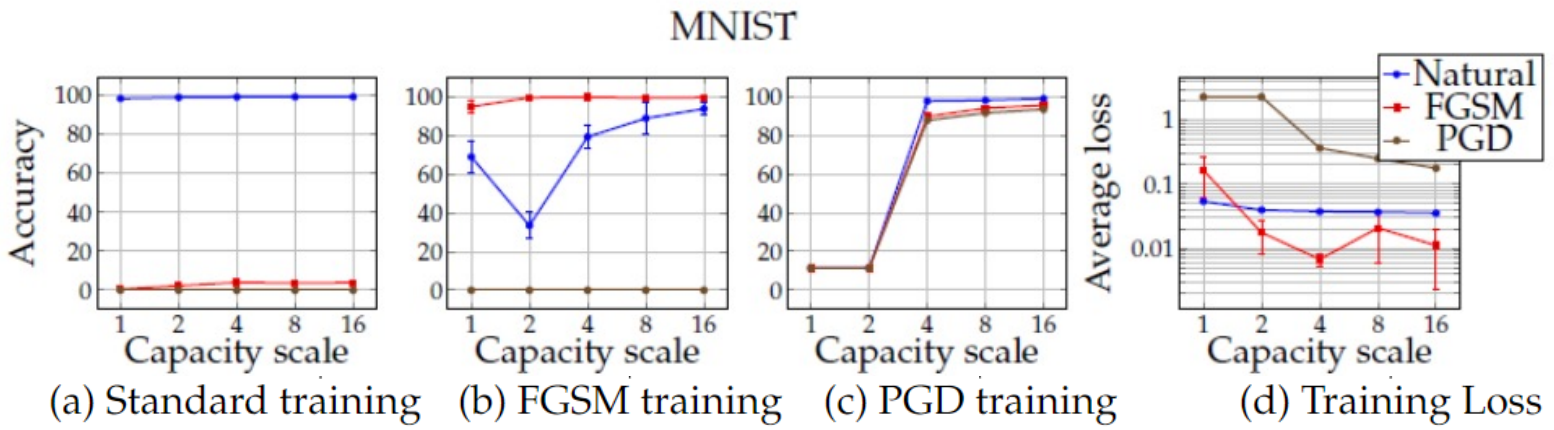
- 需要更复杂的边界/模型，如最右图所示，才能较好地表征问题

参考文献：[1] 2018-Towards Deep Learning Models Resistant to Adversarial Attacks-Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, Adrian Vladu

基于经验的防御方法

PGD对抗训练

- 随着网络规模的增大，模型的鲁棒性是提升的
  - 对于正常训练，可能很简单的模型就够了，因此训练精度变化很小；
  - 在FGSM训练中，对FGSM对抗样本分类准确率最高；对PGD样本无效；正常样本会先下降,随着模型容量的增加又会趋于高精度；
  - 对于PGD训练，三种样本的规律是一致的，即随着模型容量的增大，精度都变好；
  - 最后，给出三种训练模式下的loss，其中正常训练的loss变化不大，但是FGSM和PGD的loss随着模型容量的增大会逐渐减小。



参考文献：[1] 2018-Towards Deep Learning Models Resistant to Adversarial Attacks-Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, Adrian Vladu

基于经验的防御方法

PGD对抗训练

- 结果评估：
  - MNIST：若直接使用干净样本训练，则精度可达99.2%，但是使用FGSM攻击 准确率就下降到了6.4%。在PGD对抗训练下，干净样本的分类精度降了一点点（99.2%→98.8%）,但是对攻击方法鲁棒性大大提高（6.4%→95.6%）。
  - CIFAR10：对抗训练下，干净样本分类精度下降（95.2% → 87.3%）；
    PGD对抗训练提升了FGSM攻击下的分类精度（32.7% → 56.1%） 但没有FGSM训练效果好
    PGD攻击分类精度明显提升 （3.5% → 45.8%）

	标准训练	FGSM对抗训练	PGD对抗训练
Nature	95.2%	90.3%	87.3%
FGSM	32.7%	95.1%	56.1%
PGD	3.5%	0.0%	45.8%

参考文献：[1] 2018-Towards Deep Learning Models Resistant to Adversarial Attacks-Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, Adrian Vladu

基于经验的防御方法

集成对抗训练

- 原理：将对抗训练的过程和对抗样本的产生过程解耦，在对抗训练的过程中加入的对抗样本是通过攻击其他模型（预训练模型）产生的，增加了所加入对抗样本的多样性，从而提高模型抵御其他攻击方法（黑盒攻击）的能力。
- 在ImageNet集上对抗训练 Inception v3网络，集成对抗训练模型以Inception $v3_{adv-ens}$为例
- 集成对抗训练后，干净样本的识别准确率降低（78.0% → 75.8%）

白盒攻击下（有目标的FGSM攻击），集成对抗训练后识别准确率低于标准对抗训练（73.4% → 56.7%），高于原始网络（31.4% → 56.7%）

黑盒攻击下，集成对抗训练后识别准确率增加（48.8 % →66.6%）

	V3	$V3_{adv}$	$V3_{adv-ens}$
干净样本	78.0%	78.0%	75.8%
白盒攻击	31.4%	73.4%	56.7%
黑盒攻击	48.8%	59.2%	66.6%

参考文献：[1] 2018- Ensemble Adversarial Training: Attacks and Defenses - Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, Patrick McDaniel

基于经验的防御方法

对抗训练的一些结论

- 对抗训练模型需要比传统训练模型更大的容量，才能在干净数据上实现相同的准确率
- 对于没有对抗训练的模型，太大的模型或者太小的模型都会导致鲁棒性很差，而对于对抗训练的模型，由于条件限制，模型容量越大，鲁棒性显得越好[1]
- **标签泄露**：经过对抗训练的模型在对抗样本上可能比在干净样本上表现更好，因为对抗样本生成过程中使用了真实标签，模型可能学习到了对抗样本生成过程中的规律[1]

		Clean	$\epsilon = 2$	$\epsilon = 4$	$\epsilon = 8$	$\epsilon = 16$
No label leaking, training and eval using "step l.l."	top 1	77.3%	72.8%	73.1%	73.4%	72.0%
	top 5	93.7%	91.1%	91.1%	91.0%	90.3%
With label leaking, training and eval using FGSM	top 1	76.6%	86.2%	87.6%	88.7%	87.0%
	top 5	93.2%	95.9%	96.4%	96.9%	96.4%

无“标签泄露”：干净样本识别精度77.3%，对抗样本识别精度都要低些（72.8%，73.1%）
有“标签泄露”：干净样本识别精度76.6%，对抗样本识别精度反而高很多（86.2%，87.6%）

参考文献：[1] 2017-Adversarial Machine Learning at Scale-Alexey Kurakin, Ian J. Goodfellow, Samy Bengio
[2] 2018-Ensemble Adversarial Training: Attacks and Defenses-Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, Patrick McDaniel

基于经验的防御方法

去噪

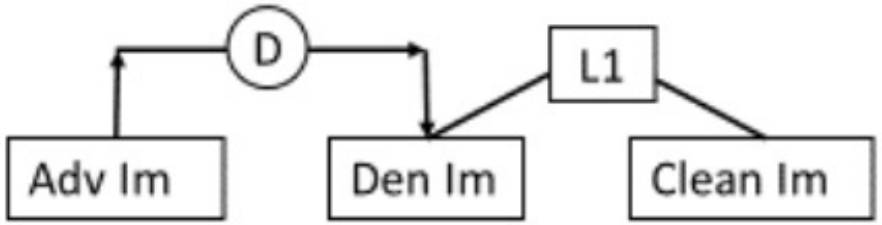
- 对抗样本通过向原始图像添加噪声来构造，使得输入模型后分类错误。如果在对抗样本输入模型之前，进行去噪处理，将攻击者千方百计添加到原始图像上的轻微干扰去除，则可以得到与原始图像近似的去噪后图像，从而分类依旧正确。
- 去噪主要有两个方向：输入去噪和特征去噪。
  - 输入去噪：模型测试阶段，防御者可以对输入数据进行去噪处理，试图消除输入数据中的部分或全部对抗性扰动。
  - 特征去噪：试图减轻对抗性干扰对DNN学习到的高级特征的影响。

基于经验的防御方法

特征去噪

- Liao等[1]提出了一种基于高级表示法指导的去噪器（high-level representation guided denoiser, HGD），这种去噪器可以改善受对抗性扰动影响的特征。
- 传统去噪器：基于像素点的去噪器（Pixel guided denoiser，PGD ），将对抗样本转换为去噪样本，通过计算去噪样本和原始图像的 L_1范数来确定损失函数（基于像素点的距离）。

$$L = \|x - \hat{x}\|$$



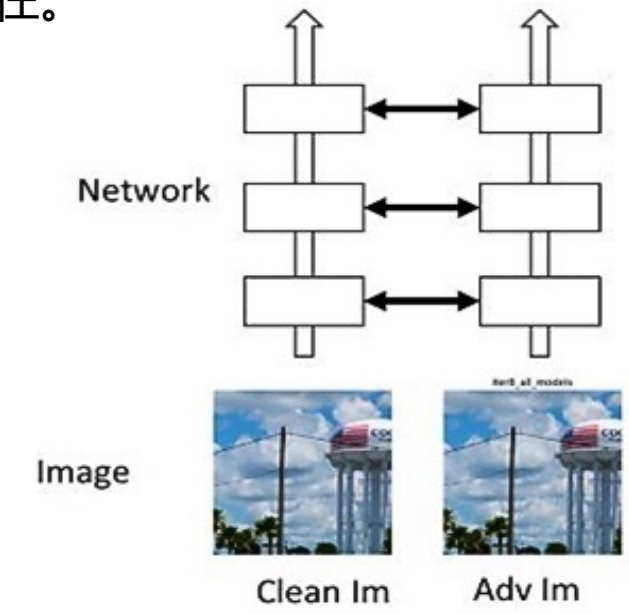
PGD去噪器

参考文献：[1] 2018-Defense against Adversarial Attacks Using High-Level Representation Guided Denoiser-Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, Jun Zhu

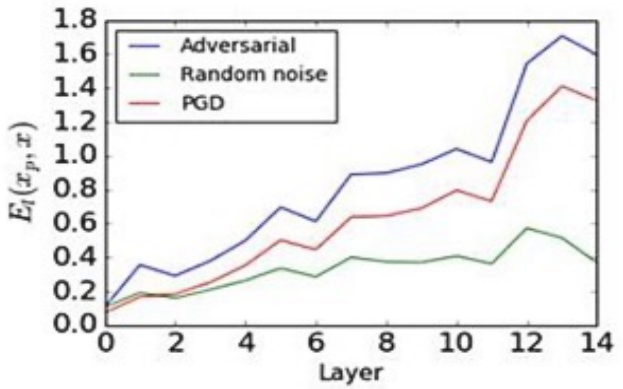
基于经验的防御方法

特征去噪

- 传统去噪器(像素级去噪)具有**误差放大效应**，在这种效应中，较小的残留对抗噪声影响会逐渐放大，最终导致错误的分类。
- 可以发现如果只是普通的噪声，比如高斯噪声，这些噪声的影响会随着网络的加深而逐渐变小,但是对于对抗样本的噪声，这些噪声的影响会随着网络的加深而逐渐变大。这一趋势在图像经过基本的去噪后仍然存在。



$$E_l(x_p, x) = \| f_l(x_p) - f_l(x) \| / \| f_l(x) \|$$



目标模型的逐层扰动

参考文献：[1] 2018-Defense against Adversarial Attacks Using High-Level Representation Guided Denoiser-Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, Jun Zhu

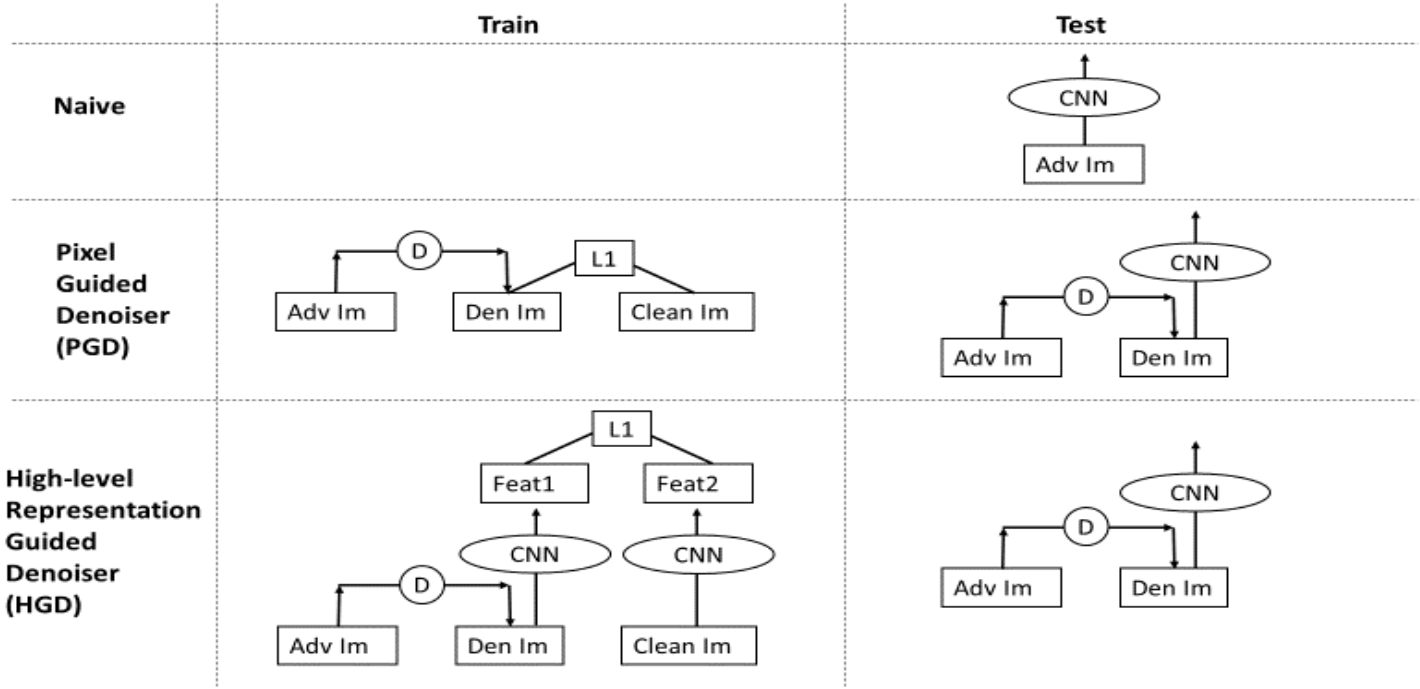
基于经验的防御方法

特征去噪

- 既然直接在输入图像上来消除噪声（PGD）并不能减少对抗样本中噪声带来的影响，那么我们直接将损失函数直接加在网络高层的特征层上（HGD）是不是可以抑制误差的放大？

$$L = \| f_l(\hat{x}) - f_l(x) \|$$

  - 去噪后的x在第l层输出与原始图像在第l层输出的 L_1 范数

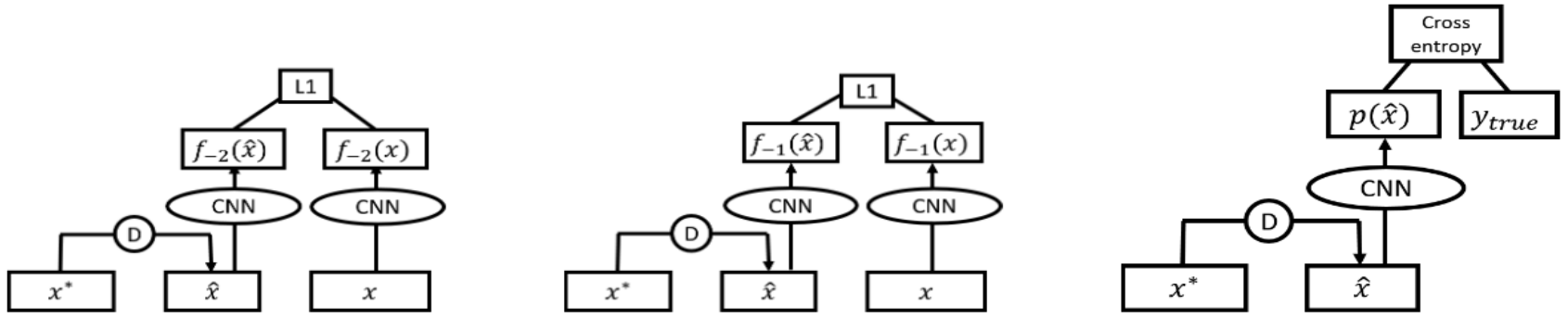


参考文献：[1] 2018-Defense against Adversarial Attacks Using High-Level Representation Guided Denoiser-Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, Jun Zhu

基于经验的防御方法

特征去噪

- 根据损失函数添加的位置和类型，提出三种HGD：
  - FGD： $l = -2$代表顶部卷积层
  - LGD： $l = -1$代表原始输出logits（softmax前一层）
  - CGD：先让CNN预测一个结果，然后与真实值作比较
- FGD和LGD不需要样本的标签信息，只需要将原图和对抗图像同时输入网络后拿到对应层的特征就好，属于非监督学习；CGD和普通的分类网络训练一样需要类别标签，让网络对于对抗样本也能分到原图的类别上，属于监督学习。

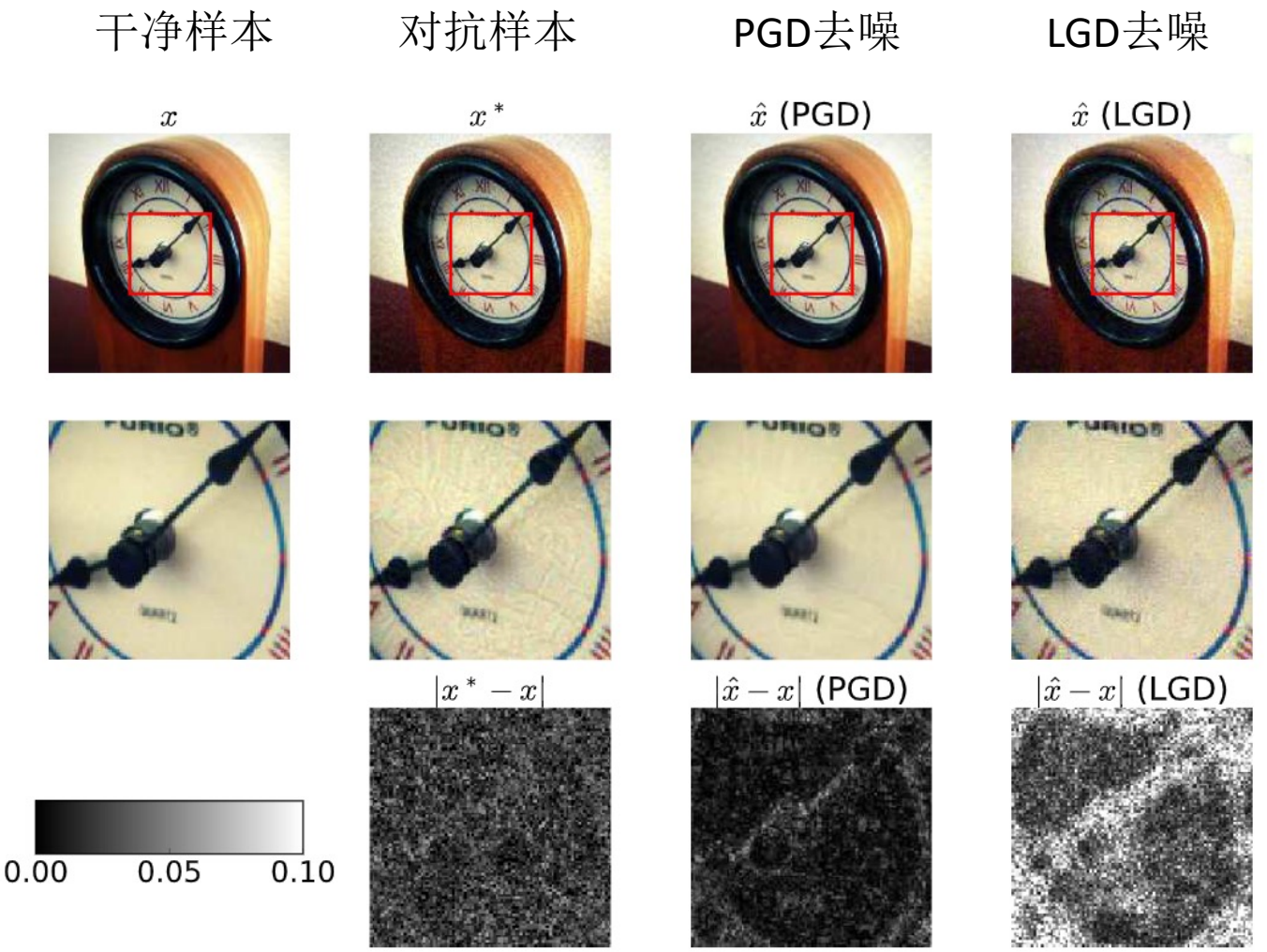


参考文献：[1] 2018-Defense against Adversarial Attacks Using High-Level Representation Guided Denoiser-Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, Jun Zhu

基于经验的防御方法

特征去噪

- 去噪效果：
  - HGD方法源于去噪灵感，但是HGD不像PGD那样去除总体噪声，反而添加了更多的像素级扰动（噪声）。
  - 这也证明，像素层面上的去噪并不能真正的去掉对抗扰动。



参考文献：[1] 2018-Defense against Adversarial Attacks Using High-Level Representation Guided Denoiser-Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, Jun Zhu

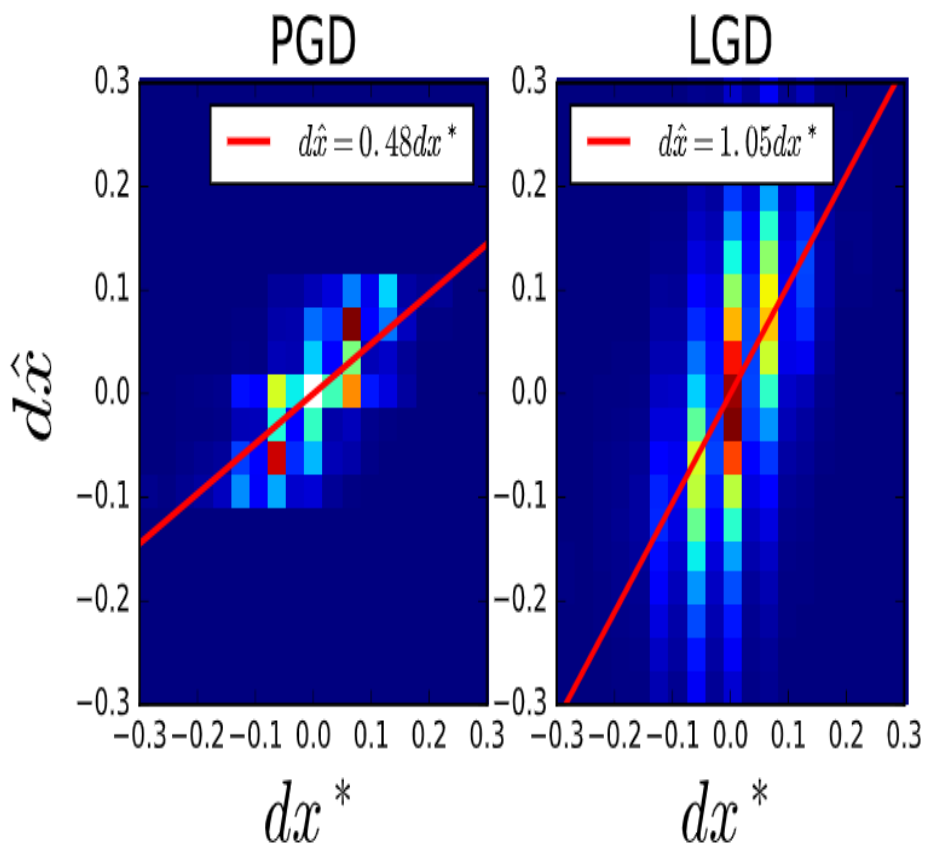
基于经验的防御方法

特征去噪

- 如图，绘制对抗扰动$dx^* = x^* - x$和通过去噪方法预测的扰动($\mathrm{d}\hat{x} = x^* - \hat{x}$) 的2D直方图 。
- 理想结果是$dx^* = \mathrm{d}\hat{x}$，意味着对抗扰动全部移除
- PGD去噪方法中：$\mathrm{d}\hat{x} = 0.48\mathrm{d}x^*$，意味着只能去掉部分对抗扰动
- LGD去噪方法中：$\mathrm{d}\hat{x} = 1.05\mathrm{d}x^*$，意味着对抗扰动几乎被全部移除。

- 由此可以知道，HGD（LGD）确实去除了绝大部分对抗扰动，同时也增加了一些像素级扰动（保护模型）

x干净样本， x^*对抗样本， $\hat{x}$去噪样本



参考文献：[1] 2018-Defense against Adversarial Attacks Using High-Level Representation Guided Denoiser-Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, Jun Zhu

基于经验的防御方法

特征去噪

- LGD是表现最均衡的去噪防御方法
- NIPS2017比赛中，HGD获得了防御赛道的第一名（107支队伍中效果最好、速度最快）

Table 3: The classification accuracy on test sets obtained by different defenses. NA means no defense.

Defense	Clean	WhiteTestSet		BlackTestSet	
		$\epsilon = 4$	$\epsilon = 16$	$\epsilon = 4$	$\epsilon = 16$
NA	76.7%	14.5%	14.4%	61.2%	41.0%
PGD	75.3%	20.0%	13.8%	67.5%	55.7%
ensV3 [31]	**76.9%**	69.8%	58.0%	72.4%	62.0%
FGD	76.1%	73.7%	67.4%	74.3%	71.8%
LGD	76.2%	75.2%	69.2%	**75.1%**	**72.2%**
CGD	74.9%	**75.8%**	**73.2%**	74.5%	71.1%

Table 1: Adversarial images generated by different models for training and testing.

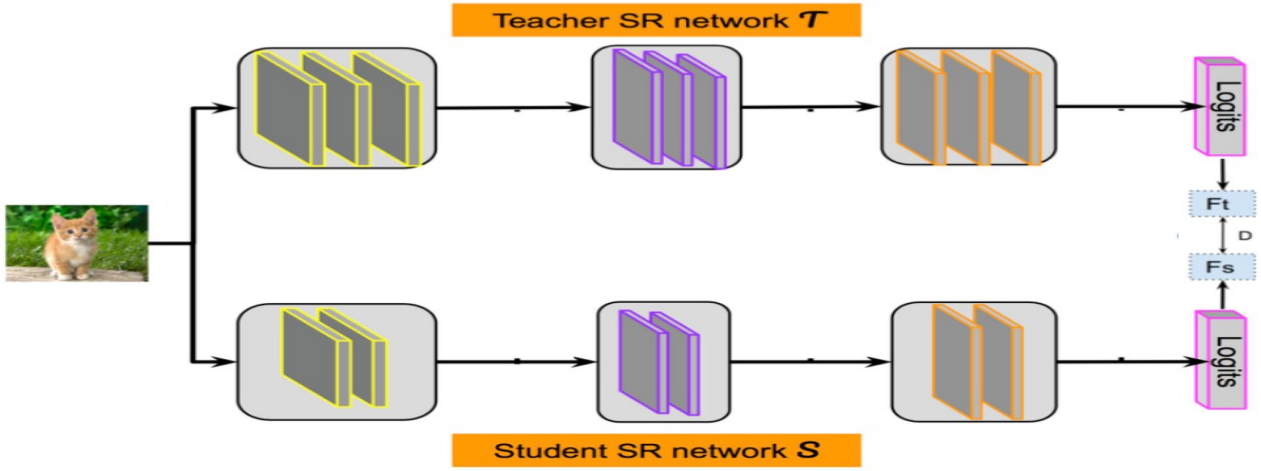
	Attacking method	Attacked model	ϵ
TrainSet and ValSet	FGSM	IncV3	[1,16]
	FGSM	IncResV2	
	FGSM	Res	
	FGSM	IncV3/IncResV2/Res	
	IFGSM2	IncV3/IncResV2/Res	
	IFGSM4	IncV3/IncResV2/Res	
	IFGSM8	IncV3/IncResV2/Res	
WhiteTestSet	FGSM	IncV3	{4,16}
	IFGSM4	IncV3/IncResV2/Res	
BlackTestSet	FGSM	IncV4	{4,16}
	IFGSM4	IncV4	

参考文献：[1] 2018-Defense against Adversarial Attacks Using High-Level Representation Guided Denoiser-Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, Jun Zhu

基于经验的防御方法

防御蒸馏

- **蒸馏**是一种将深层神经网络集合中的知识压缩为单一神经网络的方法，由原始网络和蒸馏网络2个网络组成
  - 原始网络：教师网络，一般为参数多且结构复杂的网络
  - 蒸馏网络：学生网络，一般为参数少且结构简单的网络
- 蒸馏方法可以将教师网络的知识有效地迁移到学生网络，可以压缩网络，实现降低DNN的计算复杂度。这样操作的一个好处是促进了深度学习在一些资源受限，不能使用GPUs计算设备中的部署



基于经验的防御方法

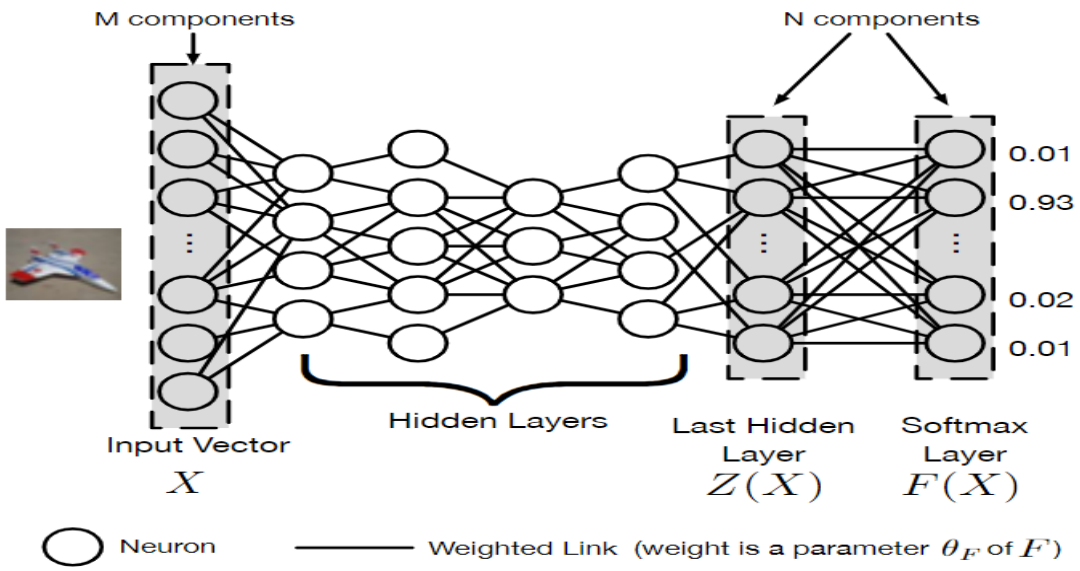
防御蒸馏

- **防御蒸馏**是蒸馏方法的扩展，并不是利用两个模型之间的知识迁移，而是提出了一种新的变体来对对抗样本进行防御。防御蒸馏不一样的地方在于：训练的两个模型架构是一样的，即目的不是了压缩，而是为了能防御对抗攻击。
- 原理：对学习到的模型进行平滑处理，以提高模型在训练数据集外的泛化性，降低对对抗样本的敏感度。
- 缺点：只对传统的简单的攻击方法生成的对抗样本具有鲁棒性。

基于经验的防御方法

防御蒸馏

- 假设：攻击者可以接触到目标模型F，即白盒攻击。
- 模型 F 架构如图，由输入层、隐含层、输出层、 softmax层构成。



深度神经网络架构

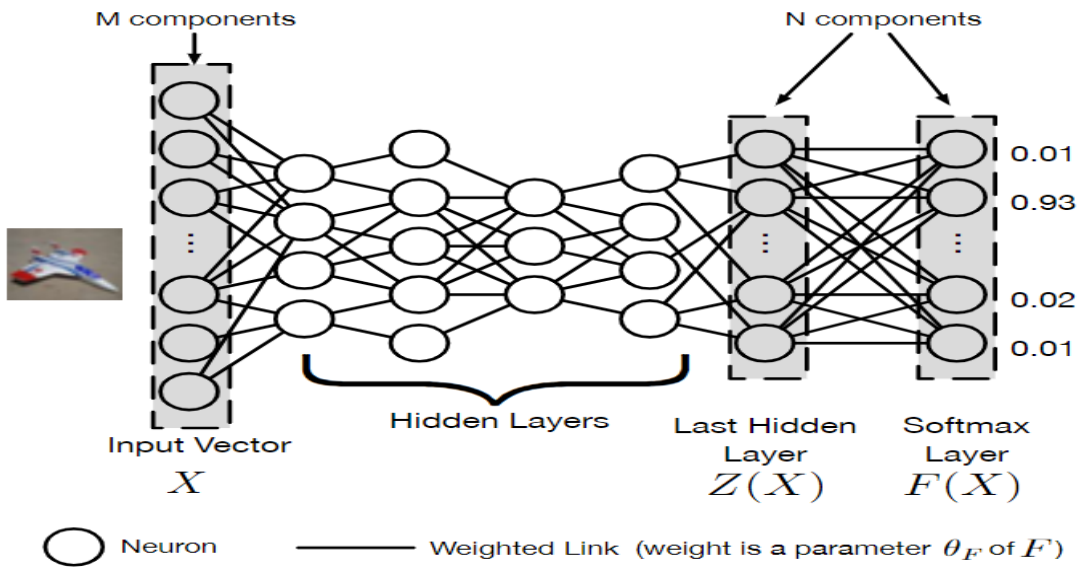
参考文献：[1] 2016-Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks-Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, Ananthram Swami

基于经验的防御方法

防御蒸馏

- softmax 层将原始输出转换成对应的概率分布。softmax 函数的形式如下：

$$F(X) = \left[\frac{e^{z_i(X)/T}}{\sum_{l=0}^{N-1} e^{z_l(X)/T}} \right]_{i \in 0..N-1}$$



深度神经网络架构

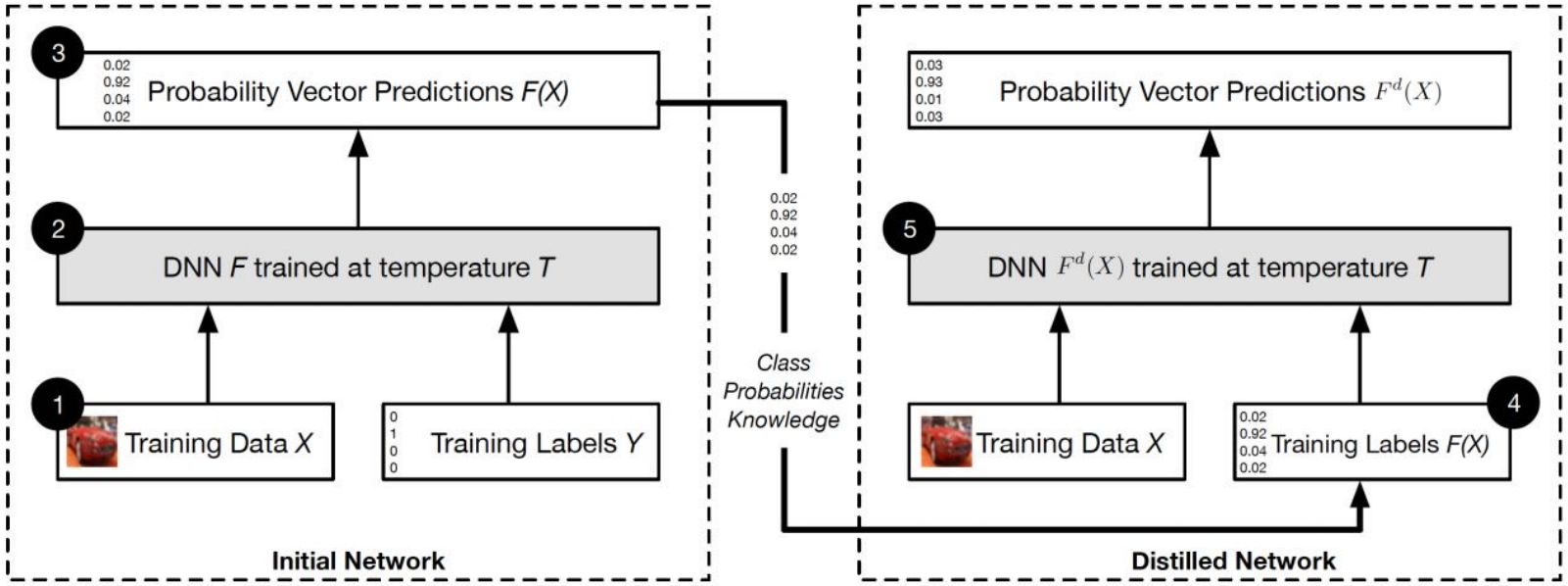
N：输出层的维度
$Z_i(X)$：隐藏层最后一层输出（原始输出logits）类别i 对应的输出分量。
T（Temperature）：训练过程控制的一个重要参数，称为蒸馏温度。引入了温度T，这是一个超参数。如果我们把T设置成1，就是标准的Softmax函数。

参考文献：[1] 2016-Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks-Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, Ananthram Swami

基于经验的防御方法

防御蒸馏

- 如图，利用蒸馏算法为原始模型训练一个蒸馏模型。
- 原始网络：
  - 训练数据集 X 和原始的标签数据 Y ；数据 Y 为硬标签
  - 训练初始的深度神经网络，得到概率分布$F(X)$



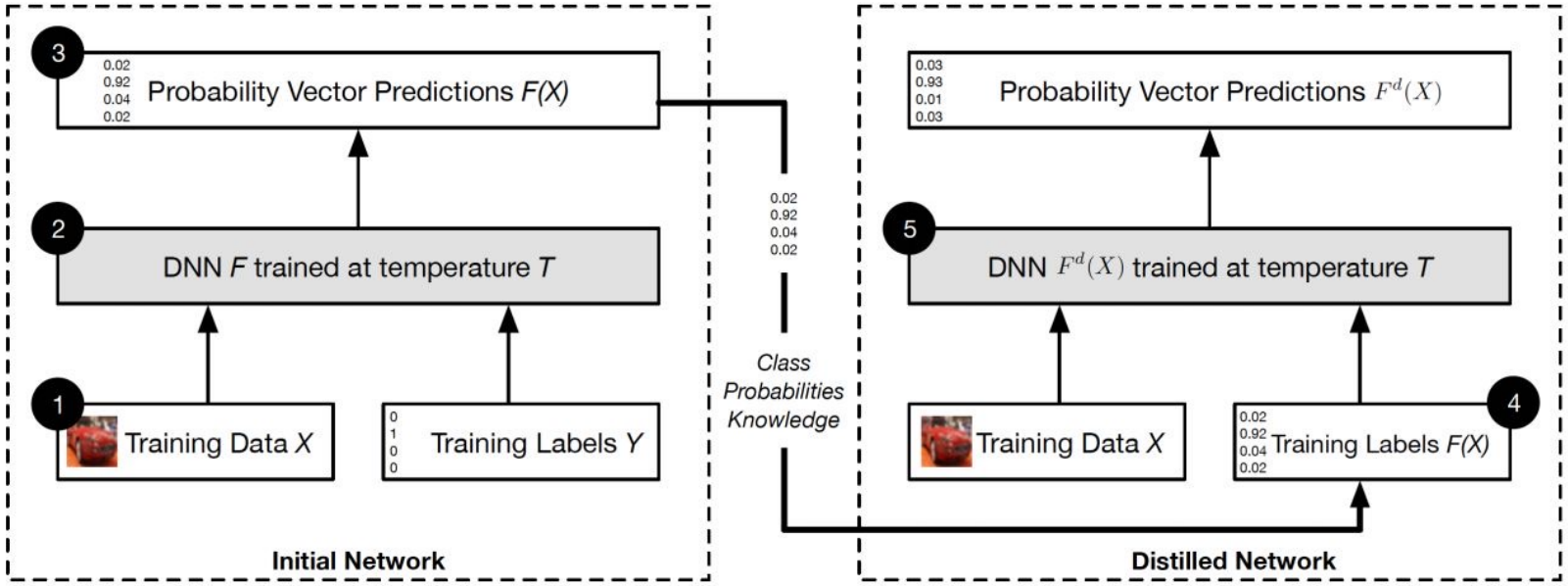
防御蒸馏架构图

参考文献：[1] 2016-Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks-Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, Ananthram Swami

基于经验的防御方法

防御蒸馏

- 如图，利用蒸馏算法为原始模型训练一个蒸馏模型，利用蒸馏模型来进行分类，能有效地防御对抗攻击。
- 蒸馏网络：
  - 基于原始网络的预测结果$F(X)$与原始输入数据集X 得到新的训练数据
  - 训练一个和原始网络架构相同、蒸馏温度T相同的蒸馏网络，得到新的概率分布$F^d(X)$(软标签)



防御蒸馏架构图

参考文献：[1] 2016-Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks-Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, Ananthram Swami

基于经验的防御方法

防御蒸馏

- 模型训练得到的预测概率$F(X)$不仅将输入X的正确分类信息编码到了网络中，同时将类别之间的相似信息也编码到了模型中
- 当 T 较小时，产生的类别概率就更离散；当 T 较大时，各个类别之间的概率分值差距会缩小，也即是强化那些非最大类别的存在感。当T 趋于无限大的时候，概率 F (X)将会收敛于1/N(每个类别没区别)
- 因此，使用较大的蒸馏温度 T 可以降低模型对输入变化的敏感性
- 预测阶段，我们需要将T设置为1，目的是为了照顾新样本。如果在预测阶段，我们将 T 设置的很大，我们的模型可能就对新样本的中的变化不那么敏感，这样就可能导致误判

参考文献：[1] 2016-Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks-Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, Ananthram Swami

基于经验的防御方法

防御蒸馏

- 在MNIST和CIFAR-10两个数据集上测试，应用防御蒸馏技术前后对抗样本的欺骗率
  - 随着 T 的增大，对抗样本攻击的成功率迅速的下降。说明这种防御方法能有效地抵御对抗样本的攻击。
  - 针对 MNIST数据集，对抗样本的成功欺骗率从95.86%降到0.45%，CIFAR-10数据集从 87.89%降到5.11%。

Distillation Temperature	MNIST Adversarial Samples Success Rate (%)	CIFAR10 Adversarial Samples Success Rate (%)
1	91	92.78
2	82.23	87.67
5	24.67	67
10	6.78	47.56
20	1.34	18.23
30	1.44	13.23
40	0.45	9.34
50	1.45	6.23
100	0.45	5.11
No distillation	95.89	87.89

参考文献：[1] 2016-Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks-Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, Ananthram Swami

基于理论的防御方法

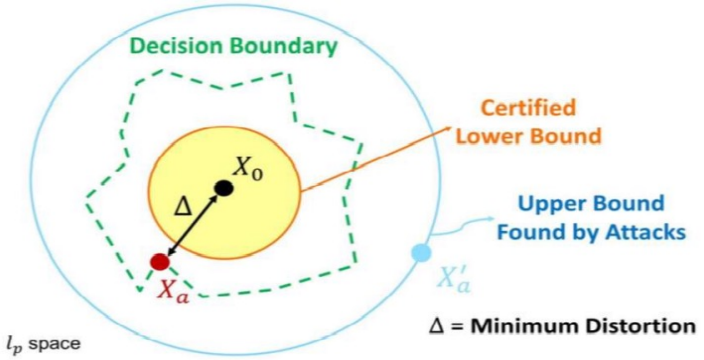
可证明式防御

- 之前介绍的所有防御都是启发式防御，这意味着这些防御的有效性只在实验上得到验证，而没有在理论上得到证明。
- 如果无法计算理论上的错误率，这些方法在未来可能会不断地被新的、更强大的攻击方法所打破，那么我们能否以某种方式结束这场攻击与防御的拉力赛?
- 因此许多研究者致力于探索可证明的防御方法，在一类定义明确的攻击下，这些方法始终能保持一定的准确性。

基于理论的防御方法

可证明式防御

- Robustness certificate $RC\ (x, F, \epsilon)$
  - 如果对于任意的 $x'\ \in\ B\ (x, \epsilon)$，都有 $F(x) = F(x')$，那么称模型在扰动ϵ下是完全鲁棒的。
  - $B\ (x, \epsilon)$：$\{x' | \parallel x' - x \parallel_p \leq \epsilon\}$
- 模型的鲁棒性由精准的决策边界来决定，这些决策边界可以由上界（upper bound）和下界（lower bound）来近似
- 对抗攻击寻找到渐进上界（asymptotic upper bound）
- 我们的挑战是如何计算（最小对抗损失的）下界，保证在这个范围空间内的样本预测结果始终不变



参考文献：[1] 2018-Certified defenses against adversarial examples-Aditi Raghunathan, Jacob Steinhardt, Percy Liang

未来挑战

有效防御白盒攻击

- 没有看到一种能够很好地平衡效果和效率的防御
- 有效性方面，对抗性训练表现出最好的性能，但计算成本很高。
- 在效率方面，许多基于随机和去噪的防御/检测系统的配置只需几秒钟。然而，最近的许多论文表明这些防御方法并没有他们声称的那样有效
- 可证明防御理论为实现对抗防御指明了一条道路，但其准确性和有效性都远远不能满足实际要求。

参考文献：[1] 2018-Certified defenses against adversarial examples-Aditi Raghunathan, Jacob Steinhardt, Percy Liang

论文阅读

- 题目：Adversarial Examples: Attacks and Defenses for Deep Learning
- 论文阅读报告