华中科技大学

2023

计算机视觉实验　课程设计报告

专　　业：	计算机科学与技术
班　　级：	计卓 2101 班
学　　号：	U202112071
姓　　名：	王彬
邮　　件：	wangbin2002@hust.edu.cn
完成日期：	2024.1.17

目 录

结课报告：生成对抗网络和对抗样本攻击与防御

1.1 定义

对抗样本（Adversarial Examples），即对一个原始样本$x \in \Sigma$，以及我们得到的分类模型$f: \Sigma \to \Omega$，考虑向原先的样本添加一个微小扰动δ，使得其在人眼或其它度量距离内扰动的范围极小，也就是

$$D(x, x+\delta) < \epsilon \tag{1}$$

这时，模型会对扰动后的样本产生误判，即

$$f(x) \neq f(x+\delta) \tag{2}$$

也就是说，扰动后的样本为对抗样本，因为它在我们认定的度量空间内扰动较小，但它对于我们的分类模型产生了误判[1]。

本定义明确规定了被视作正确的第三方，而在计算机视觉领域中，这一第三方一般是人眼。我们认为小于一定范围的偏差的样本不应当显著改变模型的结果，亦即，这样的模型应当具备鲁棒性（Robustness）。

我们讨论这里的鲁棒性与准确性的问题，一般认为鲁棒性和准确率之间存在矛盾。但根据庞天宇等研究人员的论点，合理定义的鲁棒性和准确率之间应当没有矛盾[2]。

一般地，我们定义模型的准确率的标准为，损失函数视角下模型分布$p_\theta(y|x)$和真实数据分布$p_d(y|x)$之间的KL散度应当尽可能小，而在模型鲁棒性得到最优时，应当有，

$$p_{\theta^*(y|x)} = p_d(y|x) \tag{3}$$

这里我们引用常用的 Madry 等人的定义，引入模型损失值R_{Madry}来表述模型的鲁棒性，

$$R_{Madry} = \mathbb{E}_{p_d(x)}[\max KL(p_d(y|x)||p_\theta(y|x))] \tag{4}$$

然而此时，当这里的模型损失值达到最低时，却有模型分布与真实数据分布不一致。这从源头上讲，是由于 0-1 鲁棒错误率不可微，从而我们引入的模型损

失值只是鲁棒错误率的一个可微替代。这里我们根据庞等研究者的工作[2]，给出自洽鲁棒错误率（SCORE, Self-Consistent Robust Error）的定义，这样可以保持真实类别在局部保持不变，也就是说 0-1 鲁棒错误率允许扰动，亦即它在微小扰动下自洽。

$$\mathbf{R}_{\mathrm{SCORE}}^{\mathcal{D}}(\theta)=\mathbb{E}_{p_d(x)}\left[\max_{x'\in B(x)}\mathcal{D}\left(p_d(y|x')\|p_\theta(y|x')\right)\right]. \tag{5}$$

作者证明了R_{SCORE}的最优解的自洽性，即$p_{\theta^*(y|x)}=p_d(y|x)$。这里的度量空间 D 应当将之前的 KL 散度更换为任何一种新的度量空间，但是这里的距离度量应满足对称性（Symmetry）和三角不等式（Triangle Inequality）。而我们在先前的定义中使用的 KL 散度则不满足这两者。

1.2 生成式对抗网络 GANs 方法及其改进

1.2.1 生成对抗网络 GAN

作为该领域的开山鼻祖和早期杰出工作，我们首先介绍基于神经网络的生成式对抗网络 GAN（Generative Adversarial Networks）。该模型的工作流程如图 1 所示，GAN 模型通常具有两个模块，即生成器（Generator）和判别器（Discriminator）。其中，生成器负责对随机噪声进行处理，从而生成出“假数据”，而判别器则负责从数据中鉴别是否为由生成模型生成的假数据，从而使得生成模型生成的数据可以被判别器所鉴别。

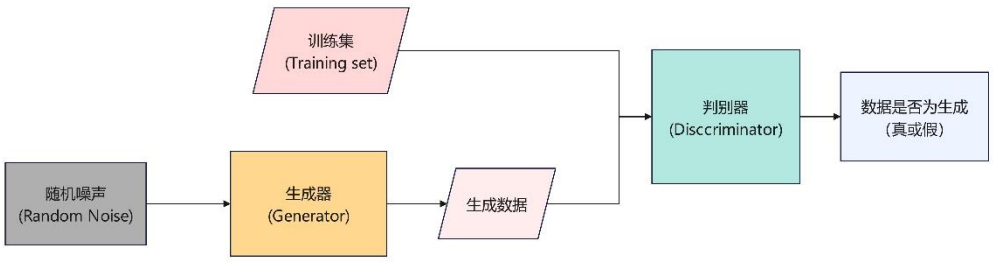

图 1 生成对抗网络工作流程

GAN 的目的是，使用对抗的方式训练一个生成模型，使其最大化判别模型犯错的可能性。这可以归结于一个经典的 Min-Max 问题，即下式，

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]. \quad (6)$$

通过生成对抗网络，将使生成器生成的数据分布逐渐与训练集趋于一致。它的具体训练过程是：

1）固定判别器，训练生成器，即用随机梯度下降的方式，训练生成器使得其生成的数据判别器无法区分；

2）固定生成器，训练判别器，即使用随机梯度下降的方式，增强判别器的鉴别力，使其能够判定出生成器所构造的假图片；

3）不断重复上述两个阶段的操作，以零和博弈的形式强化生成器和判别器的能力。

这样通过交替优化训练，我们最终使得判别器和生成器的能力都自动地得到提升。

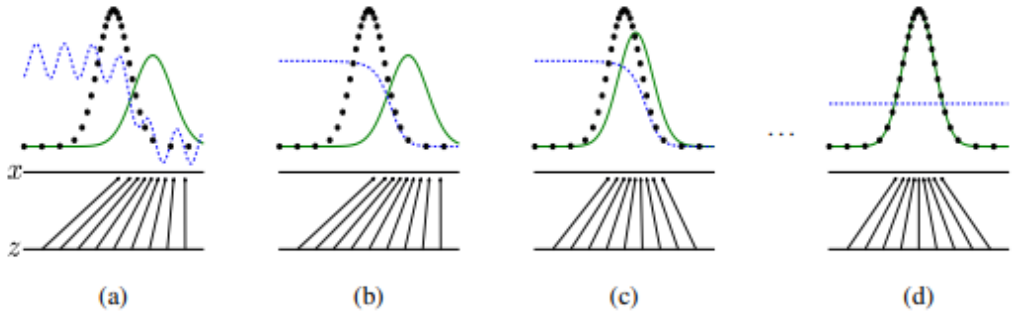

图 2 训练过程图示，即不断使得生成器得到的数据分布与原数据分布趋于一致[3]

该方法是经典的无监督学习框架，通过生成模型和判别模型的相互博弈的训练产生较好的输出。而且，模型的学习过程仅仅使用到了反向传播，无需近似计算概率；生成模型的参数更新不是直接来自数据，而是来自于判别器的反向传播。然而 GAN 的训练较难，稳定性较差。

实际训练过程中，生成器容易发散，判别器容易收敛。事实上，训练 GAN 生成对抗网络的生成器与判别器需要达到纳什均衡，但到目前为止还尚未找到很好的达到纳什均衡的方式，这使得其训练常常是不稳定的。而且，GAN 往往难以用在离散数据的生成上（例如文本），一般仅对图像生成有较好的应用，用于生成图像数据集实现数据增强，如对小样本数据集提供低成本的训练数据。GAN 还有一个问题在于，它所生成的数据分布没有显式的表示，换言之该模型的可解释性不良。

1.2.2 对 GAN 的改进工作：WGAN、WGAN-GP 方法

对于 GAN 网络的训练常常存在收敛困难的问题。我们首先讨论原始 GAN 存在的问题，根据 Arjovsky 等人的工作[4]，原本的生成对抗网络收敛发散可以归结于两点，一个是等价优化的距离量度不合理，原先使用了 KL 散度和 JS 散度，而在 1.1 节中我们已经论述了这些量度不具备对称性等优良性质；另一个问题在于判别器优化较好时，生成器会有梯度消失的问题发生。

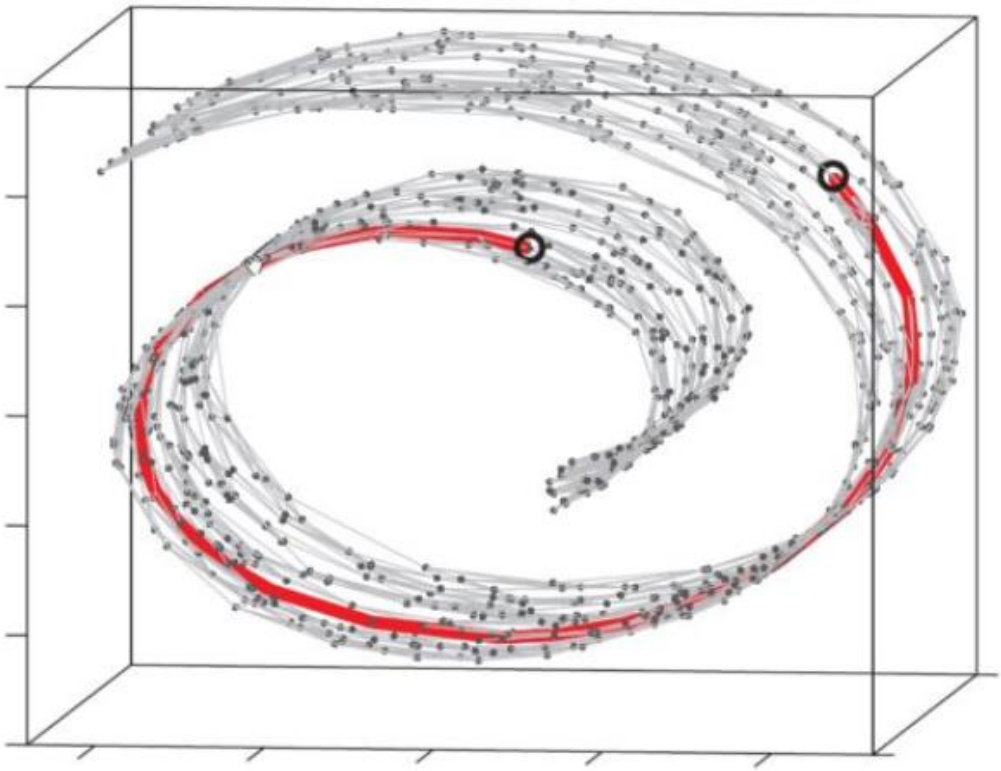

图 3 数据分布常为高维分布下的低维流形

理论上，真实的数据分布可以被看作高维度空间数据在低维度上的流形（Manifold），这里的流形指，数据虽然分布于高纬度空间，但是数据本身仅嵌于高维分布下的低维空间，因此使用低维空间足够表征数据。但是生成器所做的工作是将低维度空间映射到与真实数据类似的高维度空间内，而判别器仅在低维空间上对数据是否属于某一类别进行划分。

实践中，我们使用生成器生成的数据与真实数据的分布完全重合的可能性很低，这就导致判别器的收敛较为容易，在很多情况下我们都可以找到一个完美的判别器，这导致在生成器的训练过程中易发生反向传播的梯度消失问题。

我们的做法是，对生成样本和原始样本添加噪声，让低维的数据面散布至整个高维空间，让它们产生不可忽略的重叠。当两个分布足够靠近，它们之间的 JS 散度也将逐渐变小，但不会一直保持为一个常数，这就保证梯度消失问题可以得到解决。在训练过程中，可以对于噪声逐渐进行退火操作。直到最后两个低维流形的重叠足够多时，这里产生的噪声逐渐趋于零，而 JS 散度继续发挥作用，产生足量的有意义的梯度拉近两个低维流形，直至它们之间的距离趋于最小。

这样的技术仍然没有改变散度所固有的问题。我们知道 KL 散度和 JS 散度具有突变性，即[5]，

(7)

- $W(\mathbb{P}_0, \mathbb{P}_\theta) = |\theta|,$
- $JS(\mathbb{P}_0, \mathbb{P}_\theta) = \begin{cases} \log 2 & \text{if } \theta \neq 0\,, \\ 0 & \text{if } \theta = 0\,, \end{cases}$
- $KL(\mathbb{P}_\theta \| \mathbb{P}_0) = KL(\mathbb{P}_0 \| \mathbb{P}_\theta) = \begin{cases} +\infty & \text{if } \theta \neq 0\,, \\ 0 & \text{if } \theta = 0\,, \end{cases}$
- and $\delta(\mathbb{P}_0, \mathbb{P}_\theta) = \begin{cases} 1 & \text{if } \theta \neq 0\,, \\ 0 & \text{if } \theta = 0\,. \end{cases}$

也就是说，对于$\rho(P_0, P_\theta)$作为θ的函数时，EM 距离（即 Wasserstein 距离，$W(P_0, P_\theta)$）对于自变量连续，而 JS 散度和 KL 散度则具有不连续性。这样，理论上最小化 EM 距离来进行学习将是有意义的[5]。

事实上，原本的判别器的优化目标可以写作 JS 散度的形式，即，

$$C(G) = -\log(4) + 2 \cdot JSD\left(p_{\text{data}} \| p_g\right) \tag{8}$$

换言之，如果将判别器尽可能优化到收敛，GAN 的训练目标就退化为最小化真实数据分布和生成数据分布的 JS 散度，这也将带来梯度消失问题。

这样，作者将 EM 距离引入 GAN 的训练过程之中，以消解上面分析所得到的问题。这里原文献[5]中，作者通过一系列数学演算改进了 Wasserstein 距离的演算公式，得到，

$$K \cdot W(P_r, P_g) \approx \max_{\|f_\omega\|_L \leq K} \mathrm{E}_{x \sim P_r}[f_\omega(x)] - \mathrm{E}_{x \sim P_g}[f_\omega(x)] \tag{9}$$

这里的函数 f（表示为带参数的神经网络）应满足 K-Lipschitz 条件，也就是其

任意方向梯度的绝对值不会大于常数 K。这时我们需要增大（9）式的后者以近似拟合 Wasserstein 距离，然而原始的 GAN 分类器为判定是否属于原分布的真假二分类任务，最后一层函数使用 Sigmoid 进行二分类；而这里 WGAN 中的判别器需要拟合韦氏距离，属于回归任务，因此需要将最后一层 Sigmoid 函数去除。

WGAN 在技术上的改进具体为：

1）判别器去除最后一层 Sigmoid 函数；

2）生成器和判别器的损失函数不取 log；

3）每次更新判别器的参数后将其各个参数的绝对值进行截断，强制其小于一个固定常数；

4）不使用基于动量的优化算法（如 Momentum 或 Adam 等等），一般使用 SGD。

这是为了强制使得原神经网络函数满足 K-Lipschitz 条件，以满足求解条件。它的具体算法流程如下所示[5]，这里的 RMSProp 方法是一种梯度计算优化方法。

Algorithm 1 WGAN, our proposed algorithm. All experiments in the paper used the default values $\alpha = 0.00005$, $c = 0.01$, $m = 64$, $n_{\text{critic}} = 5$.

Require: : α, the learning rate. c, the clipping parameter. m, the batch size. n_{critic}, the number of iterations of the critic per generator iteration.

Require: : w_0, initial critic parameters. θ_0, initial generator's parameters.

while θ has not converged **do**
  for $t = 0, ..., n_{\text{critic}}$ **do**
    Sample $\{x^{(i)}\}_{i=1}^{m} \sim \mathbb{P}_r$ a batch from the real data.
    Sample $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$ a batch of priors.
    $g_w \leftarrow \nabla_w [\frac{1}{m} \sum_{i=1}^{m} f_w(x^{(i)}) - \frac{1}{m} \sum_{i=1}^{m} f_w(g_\theta(z^{(i)}))]$
    $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$
    $w \leftarrow \text{clip}(w, -c, c)$
  end for
  Sample $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$ a batch of prior samples.
  $g_\theta \leftarrow -\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} f_w(g_\theta(z^{(i)}))$
  $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$
end while

WGAN 在训练稳定性方面有一定的进步，其效果如图 4 所示。然而由于 WGAN 使用权重修剪的方式使得判别器强制满足莱布尼兹约束，这将在训练过程中导致

一些预期之外的结果，使得生成的效果有时较差，也可能产生较难收敛的情形，而 Gulrajani 等人通过实验提出 WGAN 的权重修剪将会导致模型建模能力的弱化，甚至梯度消失的问题[6]。具体的问题在于，原作者通过限制神经网络f_w的所有参数不超过一定的范围，即，

$$w \in [-C, C]，\ 原文取 C = 0.01 \tag{10}$$

通过这样的方式强制限定 Lipschiz 约束，而这种方法的限制相对粗糙。

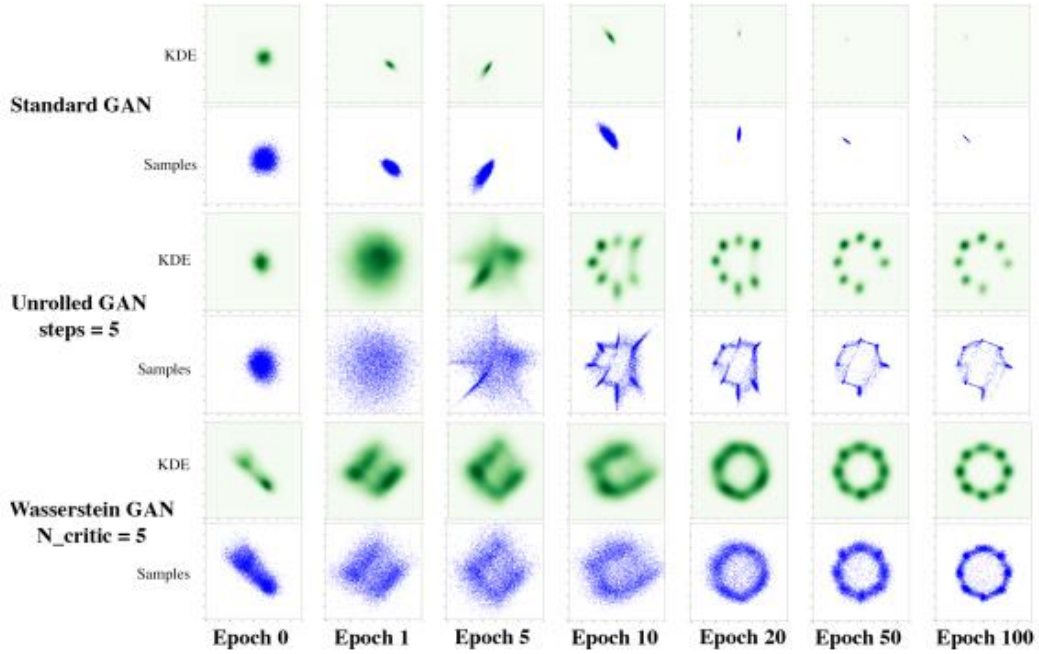

图 4 WGAN 优化效果[5]，其中数据集为 8 个高斯分布的点围成的圆环，可见其效果相对原始 GAN 有明显进步

Gulrajani 等人进一步地改进了这一点，我们下面介绍他们提出的 WGAN-GP 方法[6]。目前的目标是限定损失函数的梯度处于一定的区间之中，他们的替代方案是给原先判别器的损失函数引入梯度惩罚项（GP，Gradient Penalty）。

梯度惩罚项的设计方式是，一个可微函数满足 K-Lipschitz 条件当且仅当其任意处的梯度范数（Gradient Norm）小于常数 K，因此对于原先的 WGAN 目标函数添加了一个 L2 范数正则项，并通过双边约束将原始输入梯度限制在 1 附近。WGAN-GP 的分类器目标函数为，

$$L = \underbrace{\mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}[D(\tilde{\boldsymbol{x}})] - \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}[D(\boldsymbol{x})]}_{\text{Original critic loss}} + \underbrace{\lambda \mathop{\mathbb{E}}_{\hat{\boldsymbol{x}}\sim\mathbb{P}_{\hat{\boldsymbol{x}}}}\left[(\|\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})\|_2 - 1)^2\right]}_{\text{Our gradient penalty}}. \tag{11}$$

作者通过实验证明[6]，通过 WGAN 的梯度裁剪策略，将会导致学习到的绝大

多数权重居于两个边界值（也就是常数-C 和 C），而使用梯度惩罚项后的权重值则高斯分布于整个区间。更加关键的改进优势在于，WGAN-GP 对于较深的判别器网络，也不容易出现梯度消失问题，大大优化了其鲁棒性，这也将其生成的样本质量更高。

综合整个生成对抗网络的发展过程，GANs 最为突出的问题在于判别器容易收敛，而生成器易发散。研究人员通过多样的方式来试图解决梯度消失问题，特别是判别器的梯度易趋于零的问题。原本的 GAN 目标函数等价于判别两个分布（数据分布和生成分布）的 JS 散度，为了改进判别器的收敛问题，可以着手于定义新的距离量度，以得到不同的目标函数。

除了我们介绍的基于 EM 距离的 WGAN 和 WGAN-GP 方法外，还有采用 f-divergence 的 LSGAN 方法等。但是 f-divergence 量度同样难以计算，实践中和 EM 距离类似，常用近似的方式进行估计。

目前 GAN 方法的研究仍然处于热门地位，仍有大量的改进方法产生。大量的改进方法均围绕着 GAN 常见的两大实践问题展开，即模式崩溃与训练崩溃。模式崩溃即训练时对于参数空间的优化难以收敛至原数据分布，而训练崩溃则在于反复迭代过程中，通过固定判别器以训练生成器的目标，和固定生成器以训练判别器的目标不一致。目前通过一些训练技巧可以部分地稳定训练，例如通过标签平滑、谱归一化、PatchGAN（对图像的每一个小 Patch 进行判别，以使生成器产生更加锐利的边缘）来稳定训练，未来仍然需要在量度与收敛问题上对生成对抗网络进行改进。

1.3 对抗样本攻击与防御

对抗样本攻防，则侧重于另一个领域，即深度学习安全领域，该方法需要使用精心设计的输入样本扰乱原本表现良好的深度学习模型。对抗样本更加侧重于攻击深度神经网络，而且该技术也可用于理解神经网络内部的语义层次，并提高神经网络的性能与鲁棒性，增强其可解释性[7]。

针对深度神经网络的对抗样本可以从三个维度进行分类[7]，即威胁模型、扰动和基准。威胁模型是基于特定的场景和知识对于特定攻击方式进行部署的方法，例如对抗性伪造（Adversarial Falsification），也就是说生成一个假阳性或假阴性样

本，让其在神经网络中得到误报的结果。另一类威胁模型则需要一些关于目标神经网络模型的知识，例如训练数据、模型架构、超参数等待，并通过计算模型梯度得到对抗样本。下面我们探讨对于对抗样本攻击技术的研究。

1.3.1 对抗样本攻击技术

(一)L-BFGS 方法

我们首先讨论神经网络模型的两个固有性质。第一，神经网络的高层单元中，图像的某种特征信息由多个神经元共同表示，而不是被某个特定的神经元表征；第二，输入-输出映射相对不连续，存在一种扰动使得神经网络进行错误分类[8]。L-BFGS 方法使用搜索的方式寻找上述的扰动，让神经网络被攻击，即输出错误结果。形式化地，我们定义扰动r，使得，

$$f(x) \neq l, f(x+r) = l \tag{12}$$

Szegedy 等人使用行搜索的方式最小化这样的扰动r，也就是说，我们的目标是找到合适的常数 C，使得[8]，

$$\begin{aligned} \min_{x'} \quad & c\|\eta\| + J_\theta(x', l') \\ s.t. \quad & x' \in [0,1]. \end{aligned} \tag{13}$$

这里作者的搜索方式是二分查找合适的常数 C，这一般要求函数具备凸性。然而神经网络一般非凸，因此可能找到一个近似值。该方法作为开山之作，有力地证明了神经网络可以被攻击，而且被攻击的效果良好，Szegedy 等人在 MNIST 数据集上进行了对抗测试，使得准确率下降到 51%[8]，取得了较好的攻击效果。

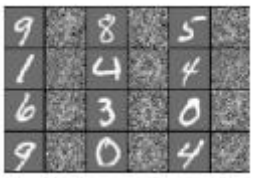

图 5 L-BFGS 方法对于高斯噪声为基准的攻击效果

然而，这里的搜索使用了复杂度较高的线搜索，而且要求神经网络所拟合的函数的凸性，后续的一系列工作将对此进行改进。

(二)快速梯度符号法：FGSM

Goodfellow 等人则证明了对抗样本的存在是因为深度神经网络的高维线性[9]。他们提出了快速梯度下降法（FGSM，Fast Gradient Sign Method），可以在白

盒环境下，亦即攻击者知道模型参数的情形，求得模型对于输入的梯度，然后反方向地乘以步长，得到的扰动与原先的输入相加，即得到对抗样本。

其攻击的思路是，通过基于梯度的攻击方法，使得损失函数向增大的方向变化。这是因为神经网络的反向传播是按照梯度方向更新网络权重，我们通过更改输入尽可能不使函数朝向较小的一方收敛。

其攻击表达式为，

$$x' = x + \varepsilon \cdot sign(\nabla_x J(x, y)) \tag{14}$$

其中，$sign$函数为去除目标维度的梯度方向。

文献[9]指出，这样攻击的效果是：1）如果激活函数为线性或近似线性（例如 Softplus、ReLU。Leaky ReLU 等等），它对于模型的攻击可以有梯度传导的效果，即扰动造成的影响将会随着模型深度的增加而越来越大；2）输入的维度越大，模型将会更加容易受到攻击。

该模型的攻击效果如图 6 所示，也就是在添加扰动后，模型将会把熊猫以 99.3%的置信度误判为长臂猿[9]。

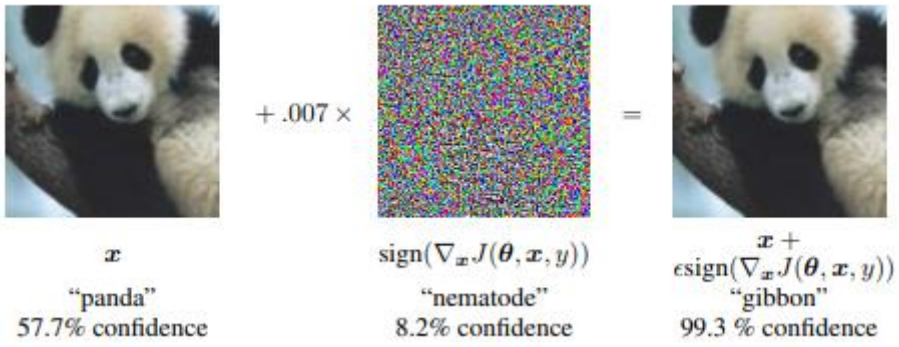

图 6 FGSM 攻击效果

这种攻击方式的速度相比前者大大提升。但它也具备一些不足：第一，它作为一种白盒攻击方式，需要得知神经网络模型的各方面的知识，例如模型的参数、数据集的分布等等；第二，Moosavi-Dezfooli 等人指出，FGSM 方法仅能提供最佳扰动向量的粗略近似，这是因为其特殊的梯度求解，而导致了次优解。

(三)寻找原始输入到对抗样本决策边界的最近距离：DeepFool

我们考虑利用决策边界缝隙的扰动，找到样本的决策超平面的边界，以实现对抗样本的生成。决策边界如图 7 所示，也就是通过样本数据的扰动，使得其原

本归属于类别 4，而通过扰动让其恰好归属于类别 3[10]。

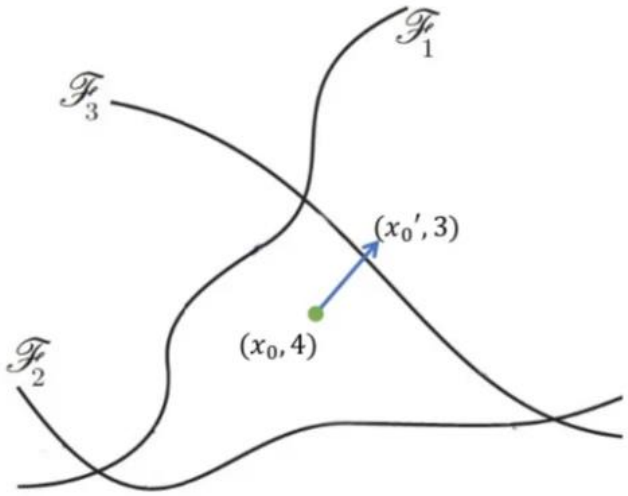

图 7 决策边界扰动，即将数据x_0通过扰动后为x_0'，以实现数据的错误分类

为了计算扰动后样本的偏差，形式化地，我们定义对抗最小扰动r，使得其可以改变估计标签$\hat{k}(x)$，

$$\Delta(x;\hat{k}) := min_r \, ||r||_2 \; subject\ to\ \hat{k}(x+r) \neq \hat{k}(x) \tag{15}$$

由于多分类问题可以归结为多个二分类问题的交，我们首先考虑二分类问题的扰动。我们对于改变分类器决策的扰动进行估计，为数据样本x_0到决策边界F上的正交投影（这一思路和感知机类似），其距离可以用解析公式给出[10]，

$$\begin{aligned} \boldsymbol{r}_*(\boldsymbol{x}_0) &:= \arg\min \|\boldsymbol{r}\|_2 \\ &\text{subject to } \operatorname{sign}(f(\boldsymbol{x}_0+\boldsymbol{r})) \neq \operatorname{sign}(f(\boldsymbol{x}_0)) \\ &= -\frac{f(\boldsymbol{x}_0)}{\|\boldsymbol{w}\|_2^2}\boldsymbol{w}. \end{aligned} \tag{16}$$

我们通过寻找最近的超平面，使用迭代方法来近似扰动，令分类器围绕数据寻找线性扰动。对于多分类器，我们将其定义为同样数量的二分类方案，并通过映射$f: R^n \to R^c$完成分类，

$$\hat{k}(\boldsymbol{x}) = \arg\max_k f_k(\boldsymbol{x}), \tag{17}$$

DeepFool 可以视为对 FGSM 的优化改进，因为其降低了扰动量，并且实现了对最优解更加近似的逼近。同时，该方法使用了单步迭代，通过计算最小距离来直接生成对抗样本，仍然可能陷入局部最优解，无法充分探索梯度空间，这些

都会导致其泛化能力具有一定的局限。

对于 DeepFool 方法的改进也可以从欧式空间着手。笔者曾经做过一段时间的知识图谱嵌入领域的实习，知识图谱由于其随着跳数的增加，关联实体的数量为指数增加的趋势，因此 Ivana 等人提出 MuRP[11]实现了双曲平面下的向量嵌入，类似的方式可能可以迁移至 DeepFool 方法内，亦即该方法是在欧式空间内实现的距离计算，或可在双曲空间内对于梯度进行计算。

(四)基于生成对抗网络的对抗样本生成：AdvGAN、GAP++

前面的各种方法为使用模型本身的性质，进行梯度计算，并通过可预见的、可解释的方式对于对抗样本进行生成。我们下面介绍两种基于我们在 1.2 节说明的生成对抗网络对于对抗攻击样本进行生成的方式。

事实上，这种对抗样本的生成方式具有天然的优势，即 GANs 网络本身就在试图“欺骗”判别器以获得更好的生成效果，我们需要合理地设计生成器和判别器，使其能够生成良好的对抗样本。

如图 8 所示[12]，我们首先将原始数据输入生成器中，以生成抗性扰动，再将生成的抗性扰动于原输入叠加作为对抗样本，并将其输入判别器和目标模型。通过对生成器和判别器的反复迭代训练，引导生成器生成最好的抗性扰动。

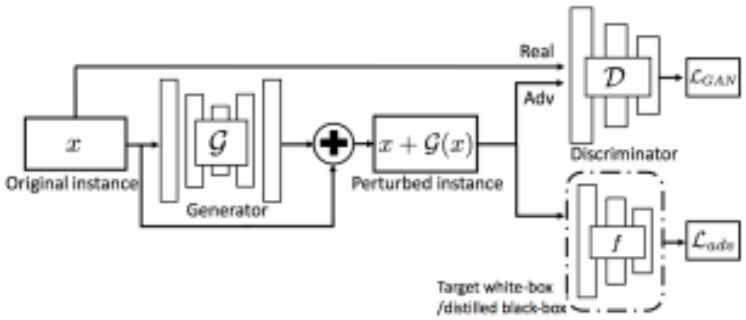

图 8 AdvGAN 工作流程

这里我们使用判别器 D 来代替视觉相似误差。这里需要定义判别损失L_{adv}，以引导生成器产生最好的抗性扰动，

$$\mathcal{L}_{\text{adv}}^{f} = \mathbb{E}_{x}\ell_{f}(x + \mathcal{G}(x), t), \tag{18}$$

综合原本 GAN 网络的损失，再加上 L2 正则项 Hinge-Loss，我们得到的目标损失函数即最优化，

$$\mathcal{L} = \mathcal{L}_{\text{adv}}^{f} + \alpha\mathcal{L}_{\text{GAN}} + \beta\mathcal{L}_{\text{hinge}}, \tag{19}$$

该方法属于黑盒攻击，因为其不需要实现得到神经网络模型的各个参数，我们也假定我们对于数据集和训练模型没有事先的知识。它在 MNIST 数据集上得到了 92.76%的成功率[12]，取得了较好的对抗效果。

GAP++则是阿里巴巴研究团队对于 GAP 模型的改进工作，该方法将扰动值限定在一个较小的范围内，即使用条件生成对抗网络（cGAN）限制图像输出。同时，该方法使用了基于模型攻击的新的 L0 范数，使得扰动数据像素个数可以由用户设置，确保图像在有效范围内进行裁剪，得到最终的生成对抗样本。

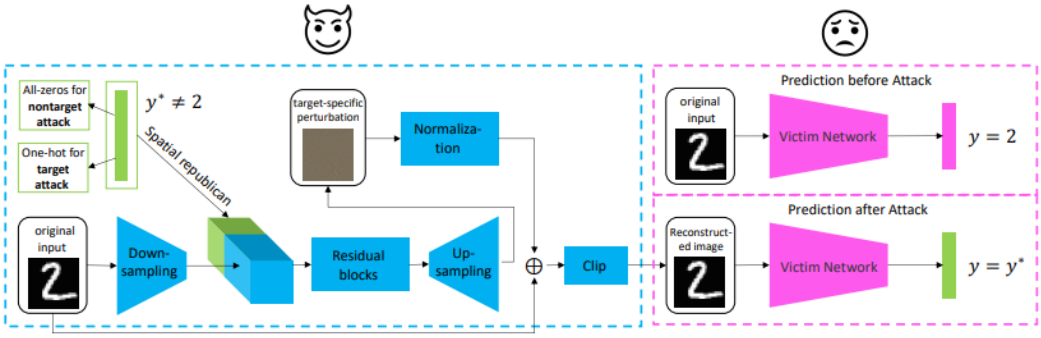

图 9 GAP++工作流程

然而，基于 GAN 实现的生成对抗网络在效果较好的同时，也具有着可移植性差、可解释性不足的缺陷。这需要进一步地研究加以优化。

1.3.2 对抗样本防御技术

我们简要介绍对抗样本的防御技术。对抗样本防御技术，即保护深度神经网络免受对抗攻击的影响[3]。一般有网络蒸馏法、梯度正则化等方式实现对抗样本防御。这里我们介绍网络蒸馏法。

网络蒸馏法（Difensive Distillation）即将知识从大型网络转移到小型网络，减少深度神经网络的规模。具体地，也就是将第一个 DNN 的最后一层全连接输出作为第二个 DNN 的输入，并在第二个深度神经网络中产生类别概率。

这样的蒸馏法的效果是，从深度神经网络中提取知识，从而提高鲁棒性。这是由于对于网络的对抗样本攻击主要利用深度网络的敏感性，而将最后一层的输出作为第二个深度神经网络的输入则可降低其小扰动的影响，在 CIFAR-10 数据集上降低了 JSMA 攻击的成功率 5%[14]。

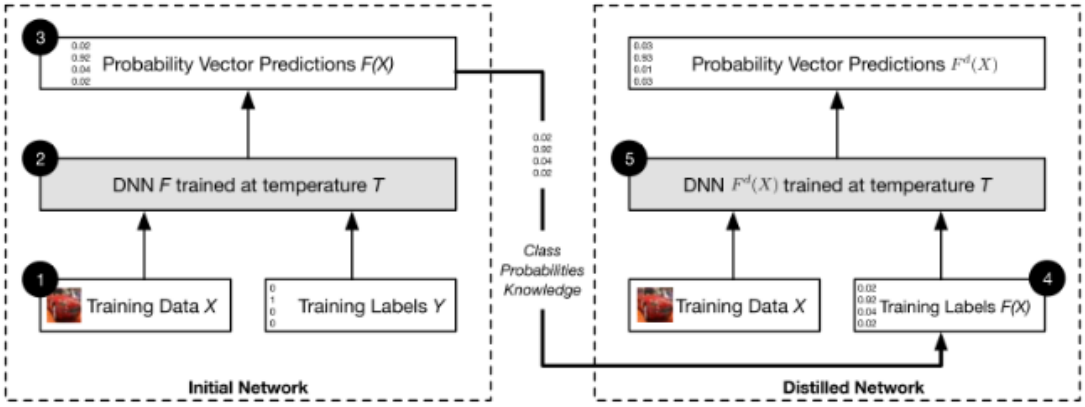

图 10 网络蒸馏图解

这样的防御之所以有效，因为先前的 DNN 在训练过程中得到的知识不仅被编码至网络的权重参数内，也编码在网络产生的概率向量中。这样的蒸馏方式则将这些网络产生的概率向量进一步提取类别知识，并转移至新的 DNN 体系和参数中[14]。这提高了模型的鲁棒性。

不过，这种蒸馏方法的对抗攻击能力是有限的。很显然，该方法是一种静态的防御方式，一旦采取新的对抗样本，这种蒸馏法将被再一次攻破。

1.4 对比：对抗样本和生成对抗网络

事实上，无论是对抗样本还是生成对抗网络，均是以博弈的方式对于神经网络模型的鲁棒性进行攻防实践。生成对抗网络使用生成器和分类器来生成新的图像，而对抗样本中的“尽可能小的扰动距离”则可以视作以人眼为分类器的生成网络博弈。

然而，这两种方法的侧重点全然不同。生成对抗网络侧重于生成样本，使得其尽可能欺骗过判别器，使得生成器的样本分布与实际数据分布重叠尽可能大；而对抗样本则主要是为了欺骗过深度神经网络，让其产生误分类。

如果转换思路，因为 GANs 的目的是使得生成数据尽可能“欺骗”过判别器，因此 AdvGAN、AdvGAN++、GAP 等工作均聚焦于两者之间的结合，将图片作为原输入，生成扰动项后，再将扰动项与原图片进行叠加，以对扰动样本进行生成。

对于生成样本的研究，是为了提高深度神经网络的鲁棒性。对抗学习提出模型的鲁棒性问题，提供了一个新的视角对于模型进行评估。而这样的数据增强的

方式，以及对抗学习中的防御方法，可以拓展到任意领域的应用。例如标签平滑的防御手段，亦可直接用于帮助缓解过拟合问题。

而对于生成对抗网络，该网络在离散程度较低的数据生成的效果较好（如图像），而对于离散程度高的文本数据则表现较差。我们可能在未来需要考虑新的基于博弈的离散数据生成方式，可能会让文本生成达到更好的性能。

参考文献：

[1] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, Alexey Kurakin. On Evaluating Adversarial Robustness https://github.com/evaluating-adversarial-robustness/adv-eval-paper/

[2] Pang T, Lin M, Yang X, et al. Robustness and accuracy could be reconcilable by (proper) definition[C]//International Conference on Machine Learning. PMLR, 2022: 17258-17277.

[3] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[J]. Advances in neural information processing systems, 2014, 27.

[4] Arjovsky M, Bottou L. Towards principled methods for training generative adversarial networks[J]. arXiv preprint arXiv:1701.04862, 2017.

[5] Arjovsky M, Chintala S, Bottou L. Wasserstein generative adversarial networks[C]//International conference on machine learning. PMLR, 2017: 214-223.

[6] Gulrajani I, Ahmed F, Arjovsky M, et al. Improved training of wasserstein gans[J]. Advances in neural information processing systems, 2017, 30.

[7] Yuan X, He P, Zhu Q, et al. Adversarial examples: Attacks and defenses for deep learning[J]. IEEE transactions on neural networks and learning systems, 2019, 30(9): 2805-2824.

[8] Szegedy C, Zaremba W, Sutskever I, et al. Intriguing properties of neural networks[J]. arXiv preprint arXiv:1312.6199, 2013.

[9] Goodfellow I J, Shlens J, Szegedy C. EXPLAINING AND HARNESSING ADVERSARIAL EXAMPLES[J]. stat, 2015, 1050: 20.

[10] Moosavi-Dezfooli S M, Fawzi A, Frossard P. Deepfool: a simple and accurate method to fool deep neural networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 2574-2582.

[11] Balazevic I, Allen C, Hospedales T. Multi-relational poincaré graph embeddings[J]. Advances in Neural Information Processing Systems, 2019, 32.

[12] Xiao C, Li B, Zhu J Y, et al. Generating adversarial examples with adversarial networks[C]//Proceedings of the 27th International Joint Conference on Artificial Intelligence. 2018: 3905-3911.

[13] Mao X, Chen Y, Li Y, et al. Gap++: Learning to generate target-conditioned adversarial examples[J]. arXiv preprint arXiv:2006.05097, 2020.

[14] Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network[J]. arXiv

preprint arXiv:1503.02531, 2015.