课程大作业

（2025-2026 学年）

(一)课程大作业题目

从在线 excel 表中选择你感兴趣的题目。利用大模型、文献、书籍、博客等搜集相关资料，整理成一篇总结报告。

(二)格式要求

(1) 格式：全文小四宋体，1.5 倍行距，8-20 页。

(2) 在本 word 文档中第二页开始撰写。按照我给的目录提示来写。

(3) 文档中的公式要给出其中字母的含义。

(4) 参考他人图表要给出引用来源，不得盗用网络图。参考文献要交叉引用。

(5) 每段话的开头要有一句加黑的总领句或者关键词，如：

神经网络定义的通俗解释。想象一下，你的大脑就像一个巨大的网络，里面有很多小小的“电报员”——这就是神经网络中的“神经元”。这些“电报员”通过无数的“电线”连接在一起，接收来自其他“电报员”的信息，然后自己再发出信息。每根“电线”都有粗细，粗细代表着信息的重要性（权重）。当一个“电报员”收到了足够重要信息时，它就会被激活，发出自己的信号。神经网络就是模仿这种方式构建的，它会通过调整“电线”的粗细来学习，从而能够识别图像、理解语言，或者预测未来。

(三)大作业提交注意事项

(1) 截止日期：11 月 11 日晚 22:00

(2) 文档命名：姓名-学号

(3) 文档格式：只提交 word 版本

(4) 提交方式：发送到 XXX 邮箱 XXXXX@qq.com，邮件主题为：课程大作业（2025-2026 学年），主题写错的话无法搜到你发送过来的邮件

(四)下面撰写大作业正文

1.1 生成对抗网络

生成对抗网络的通俗解释。生成对抗网络（Generative Adversarial Nets, GAN）可以理解成一场“造假者”和“鉴宝师”的较量。“造假者”叫生成器，专门模仿真实事物造“赝品”，比如假图片、假文字；“鉴宝师”叫判别器，负责分辨送来的东西是真的还是生成器造的假。一开始，生成器水平很差，造的假一眼就能被判别器识破。但它会根据判别器的“差评”不断改进，比如模仿真画的笔触、色彩；而判别器也在这个过程中越练越精，对真假的判断越来越准。两者就像在“军备竞赛”中互相学习、共同进步，直到生成器造出的“赝品”逼真到判别器也无法分辨真假，这场较量就达到了平衡。此时的生成器就拥有了以假乱真的能力，能输出高质量的生成内容，这就是 GAN 的核心原理——通过对抗实现双方的迭代优化。

1.1.1 博弈论基础：生成器与判别器的对抗训练

二人零和博弈思想。二人零和博弈是指由两个参与者构成的博弈$G = \{S_1, S_2 : u_1, u_2\}$，且对任意策略组合，参与者 1 的收益与参与者 2 的收益之和恒为 0，即$u_1(s_1, s_2) + u_2(s_1, s_2) = 0$。求解二人零和博弈的方法是最大最小原则（Maxmin），当参与者 1 固定一个策略s_1时，参与者 2 会选择对它收益最高的那个策略s_2，与此相对的参与者 1 的收益最差，这个最小收益记为$\min\limits_{s_2 \in S_2} u_1(s_1, s_2)$，参与者 1 需要在所有固定策略的“最差收益”中选择那个能让这个收益最大的策略，即公式(1)。

$$\max_{s_1 \in S_1} \min_{s_2 \in S_2} u_1(s_1, s_2) \tag{1}$$

对应的参与者 2 的收益为公式(2)。

$$\min_{s_2 \in S_2} \max_{s_1 \in S_1} u_2(s_1, s_2) \tag{2}$$

对于二人零和博弈，参与者 1 选择策略公式(3)，参与者 2 选择策略公式(4)。

$$a_1^* \in arg \max_{s_1 \in S_1} \min_{s_2 \in S_2} u_1(s_1, s_2) \tag{3}$$

$$a_2^* \in arg \min_{s_2 \in S_2} \max_{s_1 \in S_1} u_2(s_1, s_2) \tag{4}$$

仅当公式(1)等于公式(2)时，(a_1^*, a_2^*)构成一个纳什均衡，且这个均衡一定存在。

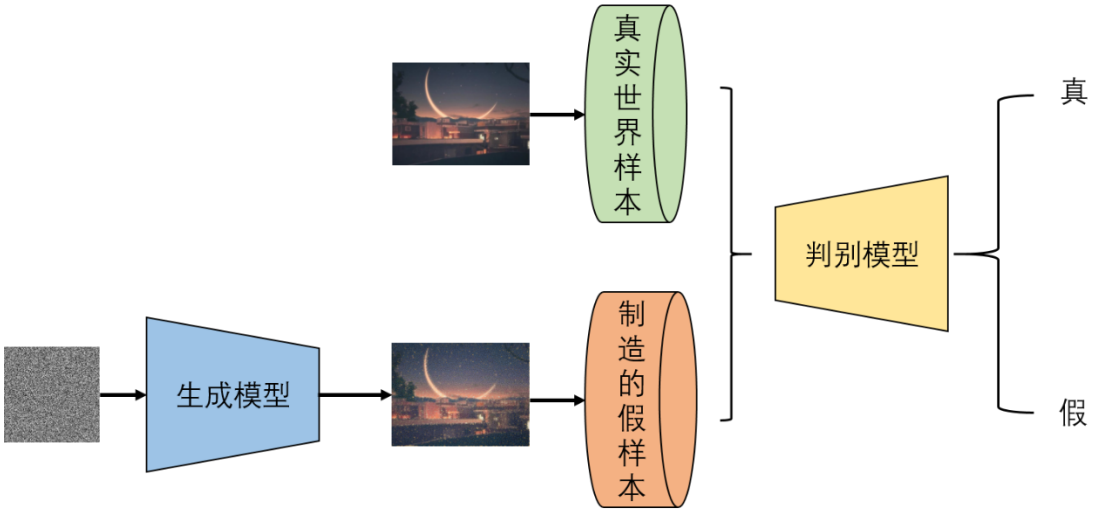


图 1 生成对抗网络结构示意图

生成对抗网络原理的文字性描述。生成对抗网络结构的示意图如图 1 所示。生成对抗网络的对抗训练是生成器与判别器交替优化、互相博弈的动态过程，核心围绕“真假分辨”与“以假乱真”的目标展开。训练初期，生成器从随机噪声中生成低质量的假样本，判别器则通过真实数据和这些假样本进行二分类学习——输出“1”代表真实样本，“0”代表生成样本。此时判别器能轻易识别假样本，生成器则根据判别器的反馈调整自身参数，逐步优化生成样本的逼真度。当判别器训练到一定精度后，切换到生成器训练：生成器尝试生成更接近真实的数据，让判别器难以分辨。而判别器也会在后续训练中，因生成器的“进步”不断提升分辨能力，通过学习新的真假样本特征更新参数。这个“生成器优化→判别器提升→生成器再优化”的交替循环持续进行，直到双方达到纳什均衡——判别器对真实样本和生成样本的判断概率均接近 0.5，无法有效区分真假，此时生成器便具备了生成高质量、逼真数据的能力，对抗训练完成。

生成对抗网络的数学原理。根据文献[1]，生成对抗网络的数学分析围绕“生成分布逼近真实分布”的核心目标展开，首先需明确关键变量与函数的定义：核心分布包括真实数据分布$p_{data}(x)$（训练数据样本的概率分布，为 GAN 最终学习目标）、噪声先验分布$p_z(z)$（生成器输入噪声服从的简单分布，是生成多样性样本的源头）、生成分布$p_g(x)$（由生成器隐式定义，即噪声$z{\sim}p_z(z)$时生成器输出$G(z)$构成的分布，训练过程需让$p_g(x) \to p_{data}(x)$）；模型函数包括生成器$G(z;\theta_g)$（可微函数，文章[1]中实现为多层感知机，参数为θ_g，将噪声空间的z映射到数据空间以输出模拟真实样本的生成数据，隐式定义生成分布$p_g(x)$）和判

别器$D(x;\theta_d)$（可微函数，文章[1]中实现为多层感知机，参数为θ_d，输出标量$D(x) \in [0,1]$以表示样本x来自$p_{data}(x)$而非$p_g(x)$的概率，本质是二分类模型的概率输出）。生成对抗网络的训练过程被形式化为“生成器最小化误差、判别器最大化误差”的 minmax 博弈，数学核心是价值函数$V(G,D)$的优化，如公式(5)所示。

$$\min_G \max_D V(G,D) = \mathbb{E}_{x\sim p_{data}(x)}[log\, D(x)] + \mathbb{E}_{z\sim p_z(z)}\left[log\left(1 - D\big(G(Z)\big)\right)\right] \quad (5)$$

对固定的生成器G，判别器的目标是精准区分真实样本与生成样本，即让真实样本$x\sim p_{data}(x)$的$D(x)$尽可能大，生成样本$z\sim p_z(z)$的$D\big(G(Z)\big)$尽可能小，等价于最大化二分类任务的对数似然函数；对固定的判别器D，生成器的目标是生成让判别器D无法区分的样本，即让生成样本$G(Z)$的$D(G(Z))$尽可能大，隐式的实现$p_g(x)$向$p_{data}(x)$逼近。

最优性证明：当$p_g = p_{data}$时达到全局最优。本文首先在生成器G固定时求解判别器D的最优解，然后再证明生成器G的全局最优条件完成证明。首先，对于任意给定的生成器G，能最大化价值函数$V(G,D)$的最优判别器D_G^*如公式(6)所示，证明如下。

$$D_G^* = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \quad (6)$$

将价值函数中生成样本的积分变量从z转换为x，如公式(7)所示。

$$\begin{aligned} V(G,D) &= \int p_{data}(x)\, log\big(D(x)\big)\, dx + \int p_z(z)\, log\left(1 - D\big(G(z)\big)\right) dz \\ &= \int p_{data}(x) lo\, g\big(D(x)\big) + p_g(x)\, log\big(1 - D(x)\big)\, dx \end{aligned} \quad (7)$$

积分内的表达式是关于$D(x)$的单变量函数，如公式(8)。

$$f(y) = a\, log\, y + b\, log(1-y) \quad (8)$$

其中$a = p_{data}(x)$、$b = p_g(x)$、$y = D(X)$，对y求导得公式(9)。

$$\frac{df(y)}{dy} = \frac{a}{y} - \frac{b}{1-y} = 0 \quad (9)$$

令其导数等于 0 可得公式(10)，且该解在$y \in [0,1]$内有效，得证。

$$y = \frac{a}{a+b} \quad (10)$$

其次，生成器的“虚拟训练准则”定义为$C(G) = \max_D V(G,D)$，其全局最小值仅在$p_g = p_{data}$时取得且为$-\log 4$，证明如下。当$p_g = p_{data}$时，代入公式(6)可得公

式(11)。

$$D_G^* = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} = \frac{1}{2} \tag{11}$$

将公式(11)代入$C(G)$的表达式，计算得公式(12)。

$$C(G) = \mathbb{E}_{x\sim p_{data}}\left[log\frac{1}{2}\right] + \mathbb{E}_{x\sim p_g}\left[log\frac{1}{2}\right] = -log2 - log2 = -log4 \tag{12}$$

为证明该值是全局最小值，引入 Jensen-Shannon 散度（JS 散度，衡量两分布差异的非负指标，仅当两分布相等时为 0），将$C(G)$重写为公式(13)。

$$C(G) = -log4 + 2JSD(p_{data} \parallel p_g) \tag{13}$$

JS 散度$JSD(p_{data} \parallel p_g) \geq 0$，且仅当$p_g = p_{data}$时为 0，因此$C(G) \geq -log4$，得证。

生成对抗网络训练过程的描述。生成器与判别器是完全独立的两个模型，它们之间没有联系，训练采用的大原则是单独交替迭代训练。首先对生成器、判别器和超参数 k 进行初始化；然后循环更新 k 次判别器，判别器的目标是最大化“对真实样本判为真、对生成样本判为假”的对数似然，其随机梯度如公式(14)所示。

$$\nabla_{\theta_d}\frac{1}{m}\sum_{i=1}^{m}\left[logD(x^i) + log\left(1 - D\left(G(z^i)\right)\right)\right] \tag{14}$$

通过梯度上升更新判别器的参数θ_d，使D的区分能力在当前G的生成水平下尽可能提升，充分学习真实与生成样本的差异；其次更新 1 次生成器，生成器的目标是最小化$log\left(1 - D(G(Z))\right)$，其随机梯度如公式(15)所示。

$$\nabla_{\theta_g}\frac{1}{m}\sum_{i=1}^{m}log\left(1 - D\left(G(z^i)\right)\right) \tag{15}$$

通过梯度下降更新生成器的参数θ_g，使生成器生成的样本更难被判别器区分。重复执行迭代判别器和生成器的算法，直至达到训练迭代次数。整个算法流程如算法 1 所示。

算法 1 生成对抗网络的 minibatch 随机梯度下降训练。判别器的迭代次数 k 是一个超参数。

for 训练迭代次数 do

for k 次 do

从噪声先验分布$p_g(\mathrm{z})$中采样m个噪声样本$\{z^1, z^2, \ldots, z^m\}$

从真实数据分布$p_{data}(\mathrm{z})$中采样m个样本$\{x^1, x^2, \ldots, x^m\}$

通过提升判别器的随机梯度对其进行更新：

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[logD\left(x^i\right) + log\left(1 - D\left(G\left(z^i\right)\right)\right)\right]$$

end for

从噪声先验分布$p_g(\mathrm{z})$中采样m个噪声样本$\{z^1, z^2, \ldots, z^m\}$

通过降低生成器的随机梯度对其进行更新：

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} log\left(1 - D\left(G\left(z^i\right)\right)\right)$$

end for

基于梯度的更新可以采用任何标准的基于梯度的学习规则。

1.1.2 训练动态与常见挑战：模式崩溃与梯度消失

梯度消失现象解释。梯度消失即是利用误差反向传播算法对深度神经网络进行训练时，梯度后向传播到浅层网络时基本不能引起数值的扰动，最终导致神经网络收敛很慢甚至不能收敛。生成对抗网络中的梯度消失现象，主要发生在训练初期生成器与判别器能力严重失衡的阶段，其核心表现为生成器因无法获得有效梯度信号而陷入参数更新停滞的状态。此时生成器水平较差，生成的样本与真实数据差距极大，判别器能轻松将其识别为“假样本”，对这些生成样本的预测概率几乎趋近于 0。这会使得损失函数对生成器参数的梯度也会趋近于 0，这种梯度信号的失效本质上是反馈传递的断裂，判别器“一边倒”的极端判断无法为生成器提供“如何改进”的有效方向，最终导致生成器难以通过参数更新提升生成样本的质量，陷入训练停滞的困境。

从数学的角度理解梯度消失现象。公式(13)中散度的相关内容如公式(16)所示。

$$JSD(P_1 \parallel P_2) = \frac{1}{2} KL\left(p_1 \parallel \frac{p_1 + p_2}{2}\right) + \frac{1}{2} KL\left(p_2 \parallel \frac{p_1 + p_2}{2}\right)$$
$$KL(p_1 \parallel p_2) = E_{x \sim p_1}\left(log \frac{p_1}{p_2}\right) \tag{16}$$

训练 GAN 网络需要$min\,JSD(p_{data} \parallel p_g)$，JS 散度的值越小表示两个分布之间越接近，这符合生成器的优化目标，即是要生成以假乱真的样本（两个样本之间的概率分布很接近）。当两个分布有重叠的时候，优化 JS 散度是可行的，而在两个分布完全没有重叠部分，或者重叠的部分可以忽略时，JS 散度是一个常数，此时梯度为 0，即生成器在训练的过程中得不到任何的梯度信息，出现梯度消失的现象。下面从两个分布是否有重叠部分分析梯度消失的原因。对于两个分布p_{data}和p_g有下面四种情况，如图 2 所示。

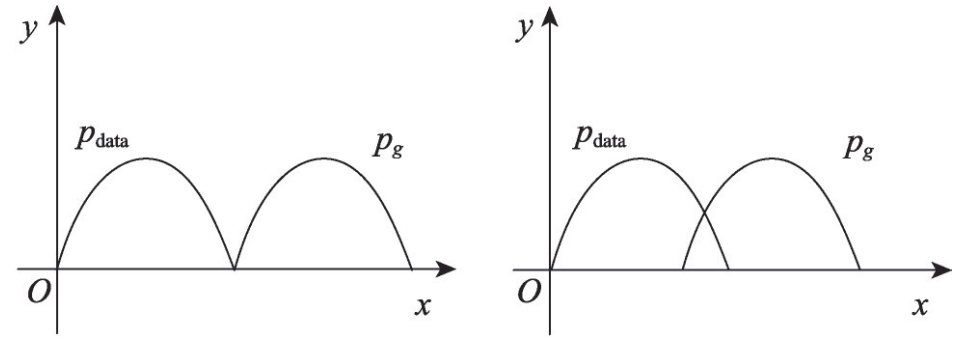
情况 1 $p_{data} = 0$且$p_g = 0$　　情况 3 $p_{data} = 0$且$p_g \neq 0$

情况 2 $p_{data} \neq 0$且$p_g \neq 0$　　情况 4 $p_{data} \neq 0$且$p_g \neq 0$

（1）两个分布没有重叠部分。对于情况 1，由于p_{data}和p_g的取值都在函数的支撑集之外，因此情况 1 对 JS 散度无贡献。情况 2 由于两个分布的支撑集没有交集，因此对最后的 JS 散度也无贡献。对于情况 3，将$p_{data} = 0$且$p_g \neq 0$代入公式(13)得到公式(17)。

$$JSD(p_{data} \parallel p_g) = 0 + \sum_{x \sim X} p_g \times log2 \tag{17}$$

由于$\sum_{x \sim X} p_g = 1$，有$JSD(p_{data} \parallel p_g) = lb2$。同理，对于情况 4，我们有$JSD(p_{data} \parallel p_g) = lb2$。说明此时生成器没有得到任何梯度信息，梯度消失。



(a) Two distributions do not overlap　(b) Two distributions overlap

(a) 两个分布不重叠　(b) 两个分布重叠

图 2 样本分布图[2]

（2）两个分布有重叠部分。文献[4]指出，两个分布在高维空间是很难相交

的，即使相交，其相交部分其实是高维空间中的一个低维流形，其测度为 0，这说明两个分布相交部分可以忽略不计，此时 JS 散度的值和（1）讨论的一致，换言之，p_{data}和p_g两个分布只要它们没有重叠部分或者重叠部分可以忽略，那么 JS 散度就固定是常数$lb2$，这对于梯度下降法而言意味着梯度为 0。文献[6]证明了若存在一个判别器D无限接近于最优解时，即$\|D\sim D_*\| < \varepsilon$，有如公式(18)的关系。

$$\left\|\nabla_\theta E_{z\sim p(z)}\left[log\left(1-D\left(g_\theta(z)\right)\right)\right]\right\|_2 < M\frac{\varepsilon}{1-\varepsilon} \tag{18}$$

这说明了当判别器训练得越好，无限接近于最优判别器D_G^*时，生成器的梯度消失越严重，即是公式(19)。

$$\lim_{\|D\sim D_*\|\to 0}\left\|\nabla_\theta E_{z\sim p(z)}\left[log\left(1-D\left(g_\theta(z)\right)\right)\right]\right\|_2 = 0 \tag{19}$$

解决梯度消失现象的一些改进。[1]中，为了解决训练过程中出现的梯度消失问题，其将生成器的优化目标由$min\ log\ (1-D(G(Z)))$更换为$max\ log\ (D(G(z)))$，该目标函数与原目标在博弈平衡点处等价，且在训练初期能提供更强的梯度信号，避免梯度过早饱和，从而缓解梯度消失问题，帮助G在训练早期更好地学习以提升生成样本质量。LSGAN[3]中将判别器的交叉熵损失替换为最小二乘损失，通过惩罚远离决策边界的样本为G持续提供梯度。WGAN[4]系列用能反映分布“远近”的 Wasserstein 距离替代易陷入常数困境的 JS 散度，即使分布无重叠也能提供稳定梯度。WGAN-GP[5]使用一种梯度惩罚的方式来满足 Lipschitz 连续性条件，从而解决 WGAN 为满足 Lipschitz 连续性条件造成的梯度消失问题。

模式崩溃现象解释。生成对抗网络中的模式崩溃现象，指的是生成器在训练过程中陷入“局部最优解”而“偷懒”，最终只能生成少数几种固定类型的样本，无法覆盖真实数据全部多样性的问题。其最直观的表现是生成结果呈现明显的“单一化”。这些生成的样本或许质量较高，能够成功骗过判别器，但本质上是生成器放弃了对数据全分布的学习。导致这一现象的核心原因在于生成器的“短期收益”优先和判别器的反馈信号偏差：对生成器而言，找到几种能稳定骗过判别器的样本类型比学习所有数据模式更“省力”，可快速降低损失，于是主动选择这种“投机取巧”的路径；而当生成器集中生成某几类样本时，判别器会针对这些样本优化判断能力，对未被生成的样本类型缺乏判断经验，这种反馈偏差会进一步强化生成器的单一化行为，使其陷入“只生成少数类型样本”的循环中。

需要注意的是，模式崩溃与梯度消失不同，前者是生成器能更新但仅向单一方向更新，导致多样性差，后者则是生成器无法有效更新，导致生成质量差。

从数学的角度理解模式崩溃现象。为了解决 GAN 的梯度消失问题，Goodfellow 等人重新定义了损失函数，如公式(20)。

$$\min_G \max_D V(G,D) = \mathbb{E}_{x\sim p_{data}(x)}[log\, D(x)] + \mathbb{E}_{z\sim p_z(z)}\big[-log\, D\big(G(z)\big)\big] \quad (20)$$

在最优判别器D_G^*(x)下，结合公式(13)和公式(20)优化生成器的目标函数如公式(21)。

$$min\, C(G) = KL\big(p_g \parallel p_{data}\big) - 2JSD\big(p_{data} \parallel p_g\big) \quad (21)$$

要优化目标函数，需同时满足公式(22)和公式(23)。

$$\min(KL(p_g \parallel p_{data})) \quad (22)$$

$$\max(2JSD(p_{data} \parallel p_g)) \quad (23)$$

公式(22)要求p_{data}和p_g的概率分布一样，而公式(23)要求p_{data}和p_g的概率分布一样，这样就会产生矛盾，使得生成器无法稳定训练。放宽约束，只要求满足公式(22)同样不可行。如下：（1）当$p_g \to 0$且$p_{data} \to 1$时，$p_g log\frac{p_g}{p_{data}} \to 0$此时$KL(p_g \parallel p_{data})$趋近于 0。该情况说明了生成器生成了与真实样本相似的样本。（2）当$p_g \to 1$且$p_{data} \to 0$时，$p_g log\frac{p_g}{p_{data}} \to \infty$，此时$KL(p_g \parallel p_{data})$趋近于正无穷。该情况说明生成器生成了不真实的样本。情况（1）生成的样本缺乏多样性，惩罚较小，生成器宁可生成一些重复但很安全的样本，也不愿意生成多样性的样本，该现象就是模式崩溃。情况（2）是生成器生成了不真实的样本，样本缺乏准确性，惩罚较大。

1.1.3 代表性变体：DCGAN, StyleGAN

DCGAN 网络。深度卷积生成对抗网络[7]（Deep Convolutional GAN, DCGAN）是 Alec Radfor 等人于 2015 年提出的一种模型。该模型在原始的生成对抗网络基础上，开创性地将 CNN 和 GAN 相结合以实现对图像的处理，并提出了一系列对网络结构的限制以提高网络的稳定性。其首次提出稳定的 CNN-GAN 融合架构，为后续一系列高质量高质量生成模型奠定基础；验证了 GAN 在无监督特征学习中的价值，推动了“无监督预训练+监督微调”的迁移学习范式。其网络生

成器结构如图 3 所示，判别器结构如图 4 所示。DCGAN 对当时存在的 CNN 架构做了如下几个方面的修改：

1.使用步幅卷积替代池化层。判别器使用步幅卷积实现下采样；生成器使用分数步幅卷积实现上采样，让网络自主学习空间缩放。

2.取消全连接层。使用全局平均池化（Global Average Pooling）替代全连接层（Fully Connected Layer）。全局平均池化会降低收敛速度，但是可以提高模型的稳定性。对于生成器的输入采用均匀分布初始化，经矩阵乘法后重塑为 4 维张量，直接接入卷积栈；对于判别器，最后的卷积层先进行拉直操作，然后送入 Sigmoid 分类器。

3.批归一化（Batch Normalization, BN）。即将每一层的输入变换到 0 均值和单位方差，BN 被证明是深度学习中非常重要的加速收敛和减缓过拟合的手段。这样有助于解决初始化不足的问题并帮助梯度流向更深的网络。防止生成器把所有的随机输入都折叠到一个点。但是实践表明，将所有层都进行 Batch Normalization，会导致样本震荡和模型不稳定，因此只对生成器的输出层和鉴别器的输入层使用 BN。

4.对于生成器的输出层使用 tanh 激活函数，其余层使用 ReLU 激活函数；对于判别器，所有层都使用 LeakyReLU 激活函数。

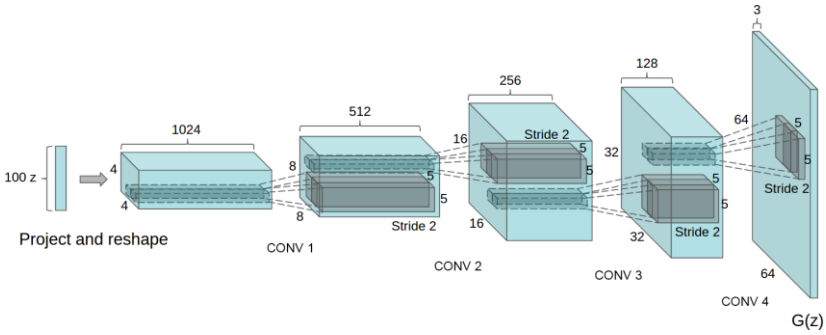


图 3 DCGAN 生成器网络结构[7]

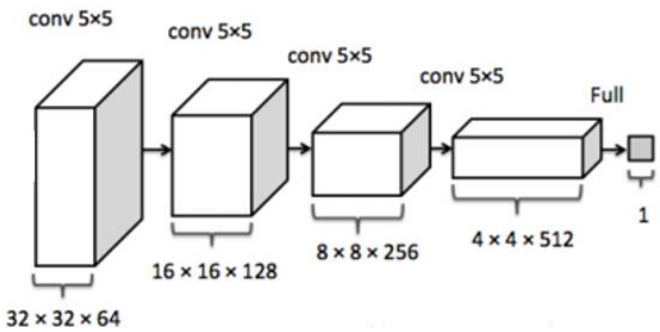


图 4 DCGAN 判别器网络结构

StyleGAN 网络。StyleGAN 从风格迁移的思路出发，重构 GAN 生成器架构。其主要有三个创新：一是引入了渐进式训练，二是提出了基于样式的生成器（Style-Based Generator），三是实现了对潜在空间较好的解耦。

渐进式训练。渐进式训练是一种“分阶段、逐步提升训练复杂度”的系统性方法，核心逻辑是将复杂任务拆解为低难度子目标，让训练对象从基础阶段起步，待适应后逐步增加训练负荷，最终实现稳定高效的目标达成。在 StyleGAN 中，首先从训练分辨率非常低的图像（如 4×4）的生成器和判别器开始，每次都增加一个更高的分辨率层，直到训练到高分辨率图像（1024×1024）结束，如图 5 所示。

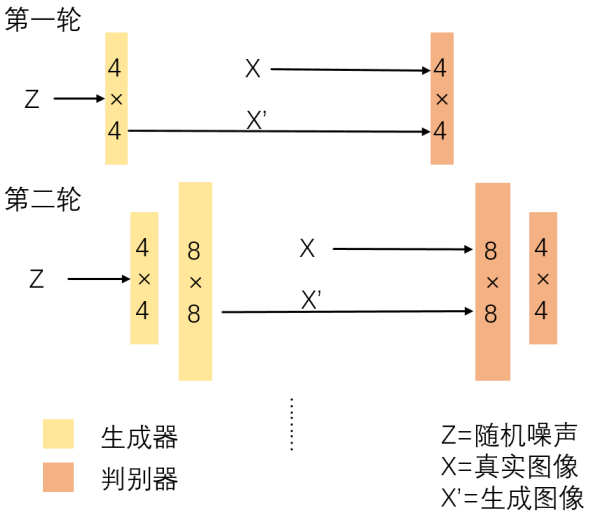


图 5 渐进式训练示意图

基于样式的生成器（Style-Based Generator）。生成器的结构如图 6 所示，传统的的生成器如图 6(a)所示，其仅使用随机隐码（latent code）作为生成器的输入。而 StyleGAN 首先将随机隐码输入到 8 个全连接层，并将其映射到一个中间的潜在空间w，w的不同元素控制不同的视觉特征，这种做法可以减少特征之间的相关性。然后w通过每个卷积层的 AdaIN 输入到生成器的每一个分辨率层中，图 6（b）中的 A 表示一个可学习的仿射变换，仿射变化 A 将w转化为 style 中 AdaIN 的平移和缩放因子$y = (y_{s,i}, y_{b,i})$。每个分辨率尺度下的计算方式是：首先对特征图x_i进行归一化处理，然后使用 style 中学习到的的平移和缩放因子进行尺度和平移变换，如公式(24)所示。

$$AdaIN(x_i, y) = y_{s,i}\frac{x_i - \mu(x_i)}{\sigma(x_i)} + y_{b,i} \tag{24}$$

在进行卷积操作和 AdaIN 操作之前在网络的每个分辨率层级上增加尺度化的噪

声（图 6（b）中的 B），噪声是由高斯噪声组成的单通道图像，以此使生成器生成更多的随机细节。在各分辨率层级增加的做法避免了特征纠缠现象，可以更好地控制噪声效果。为了进一步鼓励 styles 的局部化（减小不同层之间样式的相关性），其对生成器使用混合正则化。对给定比例的训练样本（随机选取）使用样式混合的方式生成图像。在训练过程中，使用两个随机隐码而不是一个，生成图像时，在合成网络中随机选择一个点（某层），从一个隐码切换到另一个隐码(称之为样式混合)。具体来说，通过映射网络运行两个隐码，并让对应的w_1、w_2控制样式，使w_1在交点前应用，w_2在交点后应用。这种正则化技术防止网络假设相邻样式是相关的，随机切换确保网络不会学习和依赖于级别之间的相关性。

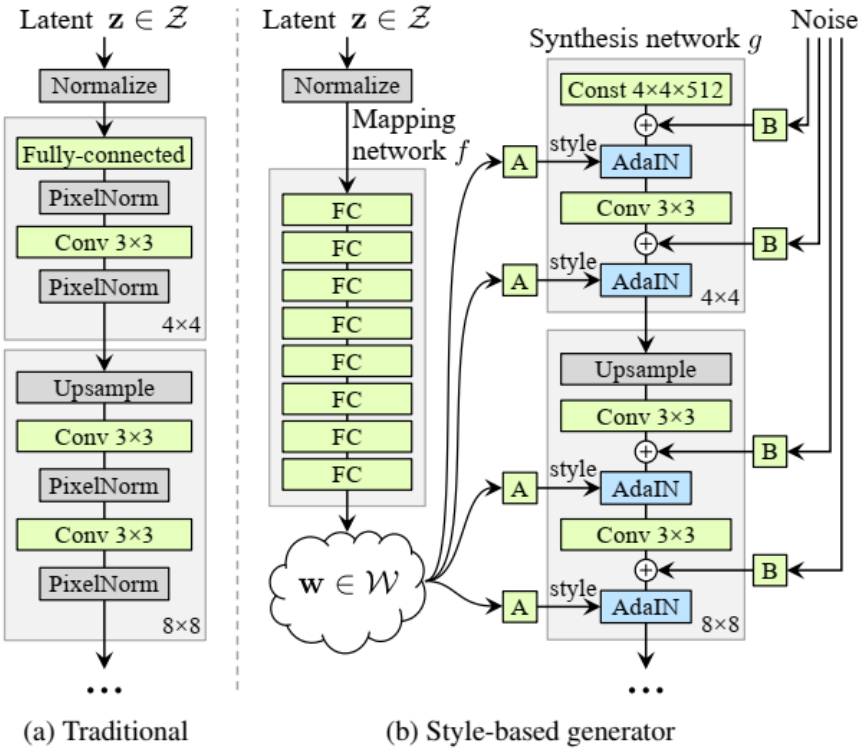


图 6 StyleGAN 生成器结构[8]

潜在空间的解耦。解耦的目标是使潜在空间由线性子空间组成，即每个子空间（每个维度）控制一个变异因子（特征）。但是潜在空间Z中的各个因子组合的采样概率需要与训练数据中响应的密度匹配，这样会产生纠缠。而中间潜在空间w不需要根据任何固定分布进行采样；它的采样密度是由可学习的映射网络$f(z)$得到的，使变化的因素变得更加线性。生成器基于解耦的表示比基于纠缠的表示应该更容易产生真实的图像。因此，我们期望训练在无监督的情况下产生较少纠

缠的w。StyleGAN 提出了两种量化解耦方式：感知路径长度和线性可分性。

感知路径长度（Perceptual path length）。感知路径长度使用 10000 个样本计算，首先将两个隐空间之间的插值路径细分为小段，感知总长度定义为每段感知差异的总和。感知路径长度的定义是这个和在无限细的细分下的极限，实际上用一个小的细分$\epsilon = 10^{-4}$来近似它。隐空间 Z 中所有可能端点（在路径中的位置）的平均感知路径长度，计算如公式(25)。

$$l_Z = \mathbb{E}\left[\frac{1}{\epsilon^2} d\left(G\left(slerp(z_1, z_2; t)\right), G\left(slerp\left(slerp(z_1, z_2; t+\epsilon)\right)\right)\right)\right] \tag{25}$$

其中$z_1, z_2 \sim p(z)$，$t \sim U(0,1)$，$slerp$表示球面插值操作，G是生成器，d是计算得到生成图像之间的感知距离。d 的计算方式是使用基于感知的成对图像距离，测量连续图像之间的差异(两个 VGG16 embeddings 之间的差异，利用 VGG16 提取出图像的特征，在特征层面上计算距离)。计算隐空间w的感知路径长度与z的唯一不同是采用$lerp$线性插值，如公式(26)所示。

$$l_w = \mathbb{E}\left[\frac{1}{\epsilon^2} d\left(g\left(lerp(f(z_1), f(z_2); t)\right), g\left(lerp(f(z_1), f(z_2); t+\epsilon)\right)\right)\right] \tag{26}$$

线性可分性（Linear separability）。线性可分性的计算方式：首先训练 40 个辅助分类器，分别对 40 个二元属性进行区分。分类器与 StyleGAN 判别器结构相同，使用 CelebA-HQ 数据集训练得到，学习率10^{-3}，批次大小 8，Adam 优化器；然后使用生成器生成 200000 个图像，并使用辅助分类器进行分类，根据分类器的置信度对样本进行排序，去掉置信度最低的一半，得到 100000 个已知类别的隐空间向量；其次对于每个属性，拟合一个线性 SVM 来预测标签-基于传统的隐空间点或基于样式的隐空间点w并且根据这个超平面对这些隐空间点进行分类；最后用条件熵$H(Y|X)$度量超平面将点划分为正确类别的能力，X是 SVM 预测的类别，Y是预先训练好的辅助分类器确定的类；因此，根据 SVM 确定样本在超平面的哪一边，条件熵告诉我们需要多少额外的信息来确定样本的真实类别。

参考文献

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

[2] 梁俊杰,韦舰晶,蒋正锋.生成对抗网络 GAN 综述[J].计算机科学与探索,2020,14(01):1-17.

[3] Mao X, Li Q, Xie H, et al. Least squares generative adversarial networks [C]Proceedings of the IEEE international conference on computer vision. 2017: 2794-2802.

[4] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. arXiv preprint arXiv:1701.07875, 2017.

[5] Gulrajani I, Ahmed F, Arjovsky M, et al. Improved training of Wasserstein GANs[J]. arXiv:1704.00028, 2017.

[6] Browne M, Ghidary S. Convolutional neural networks for image processing: an application in robot vision[C]//LNCS2903: Australasian Joint Conference on Artificial Intelligence, Perth, Dec 3- 5, 2003. Berlin, Heidelberg: Springer, 2003: 641-652.

[7] Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks[J]. arXiv preprint arXiv: 1511.06434, 2015.

[8] Karras T, Laine S, Aila T. A style-based generator architecture for generative adversarial networks[C]Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 4401-4410.