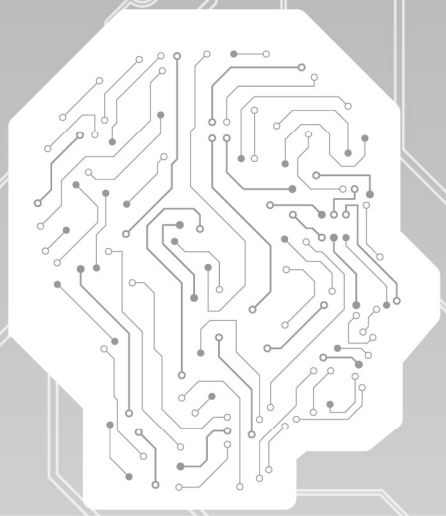

第 1 章

生成对抗网络介绍

生成对抗网络(GAN)是一种新兴的深度学习算法，是模仿给定数据集生成新数据样本的一种神经网络，它可以生成令人难以置信的逼真图像。学习构建和应用最先进的生成对抗网络是非常有价值的体验。例如，构建一个生成对抗网络来生成并未真实存在过的人的照片，或者让照片里的某个人变得更年轻或更衰老。使用 GAN 既可以将低分辨率的照片或视频转换为漂亮的高分辨率的照片或视频，也可以进行图像修复，智能去除遮挡或划痕，以获得完美的高清图片，还可以生成更多的数据以提供给学习算法。GAN 生成效果真实感强且清晰度高，因此广泛应用于电影和短视频的特效制作中。

本章介绍生成对抗网络的基本概念，讨论生成模型和判别模型之间的区别，概述 GAN 的基本架构，包括判别器和生成器，并介绍它们如何通过对抗过程进行训练，最后介绍训练 GAN 经常用到的公开数据集。

1.1 生成对抗网络与 PyTorch 简介

本节首先介绍生成对抗网络的发展简史，然后简单介绍生成对抗网络在人脸生成技术上的进步，以及“GAN”这个单词的来历，最后介绍 PyTorch 开发环境。

1.1.1 生成对抗网络介绍

首先介绍一下 GAN 能完成的工作，然后再粗浅介绍 GAN 是什么。

1. GAN 能完成的工作

2014 年，加拿大蒙特利尔大学的博士生伊恩·古德费罗(Ian Goodfellow)发明生成对抗网络 GAN，因而被誉为“GAN 之父”。该技术使得计算机能够通过使用两个独立的神经网络来生成逼真的数据，而不像通常那样只使用一个神经网络。其实 GAN 并不是第一个用于生成数据的计算机程序，但其结果的多样性和多功能性使它显得非常特别。GAN 取得了很多显著成果，比如，生成异常逼真的高质量伪造图像，能将涂鸦转变成像照片一样的高清图像，将马的视频转换为斑马的视频，等等，而且不需要大量的精心标记的训练数据(长期以来这都认定是人工智能系统无法做到的事情)。

神经网络专家杨立昆(Yann LeCun)称 GAN 为“机器学习领域近 20 年来最酷的想法”(原文是 The coolest idea in deep learning in the last 20 years.)。

使用 GAN 以后，以人脸合成为代表的数据生成技术取得了长足的进步，图 1.1 就是一个很好的例子。图中每张人脸图像下面的数字是论文发表的年份，2014—2017 年的图像来自论文 *The Malicious Use of Artificial Intelligence*: *Forecasting, Prevention, and Mitigation*[①]，2018 年的图像来自论文 *Thermal face generation using StyleGAN*[②]，2019 年和 2021 年的图像来自维基百科[③]，2024 年的图像来自当年访问的 This Person Does Not Exist(这个人并不存在)[④]网站。

早在 2014 年 GAN 刚出现时，计算机能够做到的极致的事就是生成一张模糊的人脸。

① https://arxiv.org/abs/1802.07228

② https://ieeexplore.ieee.org/document/9445031

③ https://en.wikipedia.org/wiki/Generative_adversarial_network

④ https://thispersondoesnotexist.com/

即使如此，这也被誉为突破性的成功。2017 年以后，GAN 的技术进步使合成的假脸质量可以与高分辨率人像照片相媲美。值得一提的是 This Person Does Not Exist 网站，该网站由软件工程师 Phillip Wang 于 2019 年 2 月发布，其利用 AI 人脸图像自动生成工具，基于 AI 技术生成现实生活中并不存在的人脸，在每次刷新时都会生成一张新的人脸图像，人脸的面部特征、表情和细节高度逼真。另一个可以定制图像的网站(https://this-person-does-not-exist.com/en)是由 Serhii Lopukha 发布的，该网站可以定制性别、年龄、种族，更容易满足用户个性化的需求。读者还可以自行探索另一个人脸生成的网站——https://thispersonnotexist.org/。

图 1.1　人脸生成技术上的进步

另一些有意思的类似网站如下。

- https://whichfaceisreal.com/index.php，要求用户在两张脸中判断哪一张是真实人脸。
- https://www.thiswaifudoesnotexist.net/，AI 自动生成一张二次元少女的头像。

除了能够生成图像外，GAN 还可以生成文本、语音、时间序列数据、关系数据、表格数据、连续事件日志数据、地理位置轨迹数据，等等。

2. GAN 是什么

GAN 是英文 Generative Adversarial Networks 的字首缩写，让我们先来看看这三个单词

的含义。

(1) Generative(生成)一词表明模型的总体目的是创建新的数据。训练后的 GAN 能生成什么数据取决于训练数据集。例如，如果想让 GAN 生成看起来像达·芬奇画作的图像，就要使用达·芬奇的画作作为训练数据集。

(2) Adversarial(对抗)一词是指构成GAN框架的生成器和判别器这两个模型之间的动态博弈和竞争。生成器的目标是伪造与训练集的真实数据相似度高甚至无法区分的样本数据。例如，生成看起来像达·芬奇画作的假画。判别器的目标是将生成器伪造的虚假样本与来自训练数据集的真实样本区分开来。可以这样认为：判别器就像一位艺术专家，评估达·芬奇画作的真伪。两个网络一直在试图战胜对方，生成器在伪造数据方面做得越好越逼真，判别器就要在区分真实样本和虚假样本方面做得越出色。

(3) Networks(网络)一词表示实现生成器和判别器最常见的机器学习模型是神经网络。根据 GAN 实现的不同复杂程度，其可以是简单的全连接神经网络，也可以是卷积神经网络，甚至是更复杂的神经网络，如 U-Net 等。

尽管 GAN 背后的数学计算很复杂，但是，为了让大众更容易理解 GAN，可以用现实世界中的一些例子来类比。前文讨论过伪造画作的例子，伪造假画的生成器试图欺骗艺术专家判别器，伪造者制作的假画越逼真，艺术专家就必须越善于鉴别真伪。反之亦然，艺术专家越善于鉴别一幅画的真伪，伪造者就越需要提高自己的伪造能力。

“GAN 之父”伊恩·古德费罗喜欢用假钞的比喻来描述 GAN：伪造假钞的罪犯(生成器)和一个试图抓住他的侦探(判别器)之间的对抗，假钞看起来越像真的钞票，就要求侦探越有鉴别假钞的能力，反之亦然。

可以使用更专业的术语来描述 GAN：生成器的目标是能够捕获训练数据集的模式并伪造几乎能以假乱真的样本。可以认为生成器是一个反向的目标识别模型。一般的目标识别算法通过学习图像中的模式来识别图像内容，但生成器不再识别模式，而是学习从头开始创建模式。实际上，生成器的输入通常只是一个随机向量，常用 z 来表示。

生成器通过从判别器的分类结果中得到反馈进行学习。判别器的目标是确定一个特定的样本是否真实：如果样本来自训练数据集，就应判定为真；如果样本由生成器伪造，则应判定为假。因此，每次当判别器被欺骗，将伪造图像判定为真时，生成器就成功了。相反，每次当判别器正确地将生成器伪造的图像判定为假时，通过判别器的反馈，生成器就会知道自己还需要改进。

判别器也在不断改进，像普通分类器那样从预测标签与真实标签的差值(或损失)中学习。因此，随着生成器伪造的数据越来越逼真，判别器在鉴别数据的真实性方面也做得越来越好，两个网络在对抗中得到提升。

本书将深入研究使得上述一切成为可能的 GAN 算法。

1.1.2　PyTorch 介绍

PyTorch 框架是 Facebook 公司开发的广受欢迎的端到端深度学习平台之一，是一个用 Python、C++和 CUDA 语言编写的免费开源软件库，广泛用于语音识别、计算机视觉、自然语言处理等各种深度学习网络。PyTorch 主要提供两个高级功能：①具有强大 GPU 加速的类似 Numpy 的张量计算；②包含自动求导系统的深度神经网络。

PyTorch 的前身可追溯到 2002 年诞生于美国纽约大学的 Torch。Facebook 人工智能研究院(FAIR)团队于 2017 年 1 月在 GitHub 上开源了 PyTorch，并迅速占据 GitHub 热度榜榜首。PyTorch 是具有先进设计理念的框架，对 Tensor 之上的所有模块进行了重构，新增先进的自动求导系统，因此立刻引起广泛关注，并迅速在研究领域流行起来。

在开源框架领域，PyTorch 与 TensorFlow 之间的竞争一直存在，研究人员在写论文时也会有不同的偏向。但近年来，得益于 PyTorch 自身的一些优势，越来越多的学者偏向于选择 PyTorch，TensorFlow 的使用比例也因此逐渐下降。据 2024 年 4 月 6 日访问谷歌趋势(google trends)得到的 PyTorch 与 TensorFlow 趋势对比，我们可以看到 PyTorch 占据明显优势，如图 1.2 所示。

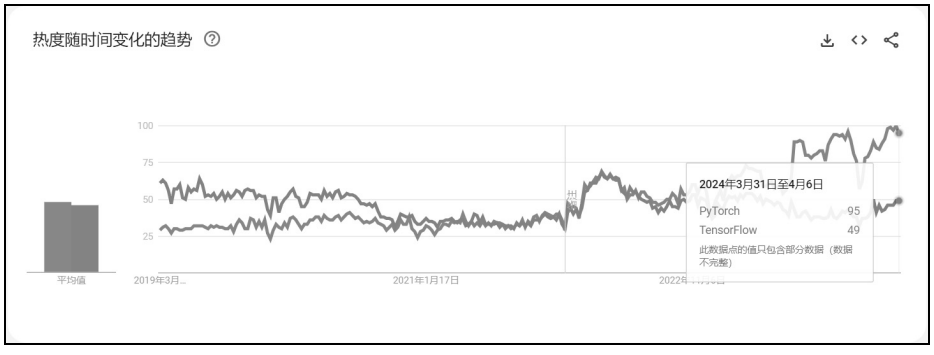

图 1.2　谷歌趋势的 PyTorch 与 TensorFlow 趋势对比

博文 *TensorFlow vs PyTorch*: *Deep Learning Frameworks*[2024][①]深度比较了 TensorFlow 和 PyTorch 的优缺点，并从性能、训练时间，以及内存使用、精度、调试、计算图定义多个

① 来源：https://www.knowledgehut.com/blog/data-science/pytorch-vs-tensorflow#difference-between%C2%A0tensorflow%C2%A0and%C2%A0pytorch%C2%A0

方面综合比较两者，值得一读。

就框架本身来说，越来越多的研究者在论文中选择使用 PyTorch，其原因可能有以下三个。

第一，操作简单。PyTorch 的工作方式与 Numpy 类似，很容易融入 Python 的生态系统。Numpy 用户感到最为亲切的就是 PyTorch 非常容易调试，但在 TensorFlow 中调试模型非常麻烦。

第二，合理的 API。多数研究者更喜欢 PyTorch 的 API，部分原因是 PyTorch 的 API 设计更加合理，另一部分原因是 TensorFlow 的 API 非常复杂，既有低级 API，又有 Keras 和 Estimators 等高级 API。

第三，不错的性能。PyTorch 使用动态图，不容易优化，但有一些非正式报告称 PyTorch 在速度上不亚于 TensorFlow。公平而言，至少 TensorFlow 在速度上还没有取得绝对优势。

因此，如果不是多年习惯使用 TensorFlow 的老用户，选择使用 PyTorch 无疑是明智之举。

1.2 判别模型与生成模型

判别模型是机器学习最常见的模型，通常用于机器学习中的分类。例如，学习如何区分猫和狗两种类别，通常将这样的判别模型称为分类器。判别模型使用一组特征 $\boldsymbol{x}$，从这些特征中确定图像中的动物类别是狗还是猫。换句话说，判别模型是在给定一组特征 $\boldsymbol{x}$(毛色、眼睛、体态、是否伸舌头等)的情况下，试图对类别 y(猫还是狗)的概率进行建模。判别模型可用条件概率表示为 $P(y|\boldsymbol{x})$，意思是在 $\boldsymbol{x}$ 的条件下 y 的概率。

生成模型试图学习如何对某些类别进行逼真的表示。例如，取一些通常称为噪声 z 的随机值输入到生成模型，让生成模型生成猫或狗的照片。如果输入只有噪声，而没有类别 y，那么生成模型既可能生成一只猫，也可能生成一只狗，这称为无条件生成。如果输入还包括类别 y，比如指定类别是狗，那么生成模型必须生成一只狗的照片，这称为条件生成。条件生成可用条件概率表示为 $P(\boldsymbol{x}|y)$，意思是在 y 的条件下 $\boldsymbol{x}$ 的概率。如果只要求生成模型生成一个类别，比如只使用全是狗的数据集，那就只会生成狗，也就不需要加上条件 y，只求特征 $\boldsymbol{x}$ 的概率即可，即 $P(\boldsymbol{x})$。

判别模型与生成模型对照如图 1.3 所示。其中，判别模型使用决策边界划分猫和狗，生成模型使用噪声 z 作为输入，输出猫或狗的照片。

为什么需要将噪声输入到生成模型？显然，只能生成固定的一只狗的模型意义不大。

增加随机噪声以后，生成模型就可以生成很多只不同种类、不同毛色、不同体型的狗，更具多样性。简单地说，生成模型试图捕捉 $\boldsymbol{x}$ 的概率分布，如眼神、体态、是否伸舌头、耳朵形状等不同的特征；噪声保证生成模型可以生成类别 y 的更真实且多样化的表示。

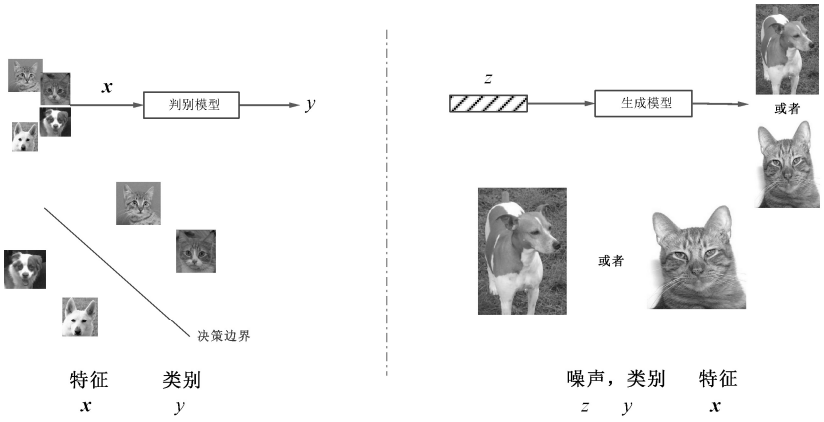

图 1.3　判别模型与生成模型对照

总之，生成模型通过学习来生成看起来很真实的样本，就像艺术家画出看起来很像真实照片的画作那样，可以认为生成模型是试图学习如何创造逼真艺术的艺术家；判别模型则是区分输入的不同类别，例如，区分猫和狗。当然，判别模型可以是生成模型的一个组件——判别器，其任务是判定输入样本的真伪。生成模型也有很多种，如变分自编码器(variational autoencoders)和扩散模型(diffusion models)。本书只专注于 GAN，感兴趣的读者可自行查阅相关文献。

1.3　GAN 架构介绍

生成对抗网络(GAN)是一种机器学习技术，由两个同时训练的网络模型组成：一个称为判别器(discriminator)，该模型用于从真假两种样本中识别出数据的真伪；另一个称为生成器(generator)，该模型用于生成虚假数据。

1.3.1　判别器

判别器是机器学习中的一种分类器，下面首先回顾分类器的基本原理，然后使用概率术语来表述分类器的学习建模过程，最后讲述如何将分类器转换为 GAN 判别器。

从机器学习的基础概念可知，分类器的目标是区分目标属性的不同类别，也就是分类。因此，给定若干手写数字的图片让分类器训练学习，分类器应该能够区分出某张照片是 5 还是其他数字，如图 1.4 所示。

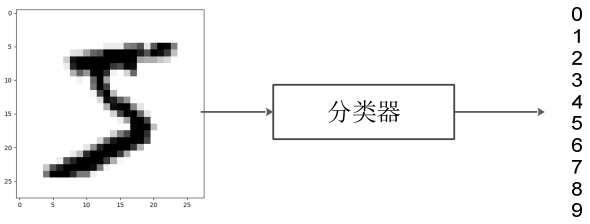

图 1.4　分类器的功能

分类器并不限于图像分类，它还可以将文本、语音等其他数据分类。

例如，分类器的输入为手写数字 5，像素为 x_1，x_2，…，x_d，一共有 d 个不同的特征，如 MNIST 数据集的 d=28×28=784。分类器通过一系列非线性运算，输出各个类别的概率。开始时，模型可能不知道如何正确分类，但是它会不断学习，根据数据中的真实标签来改进预测，提升预测性能。图 1.4 中分类器认定输入图像为 5 的概率为 0.85，为 4 的概率为 0.05，为 6 的概率为 0.10，如图 1.5 所示。

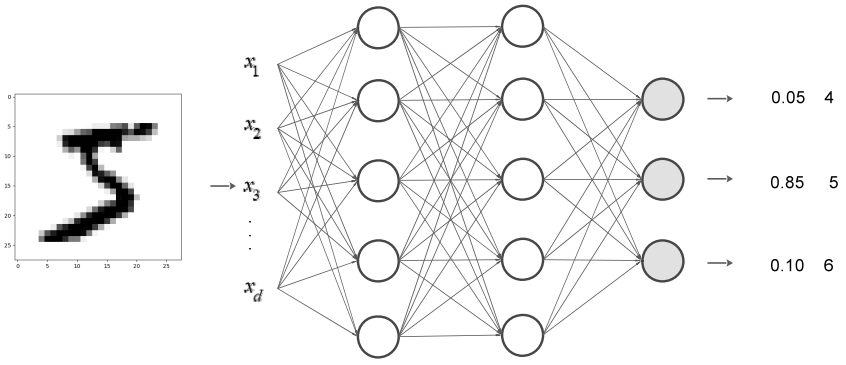

图 1.5　分类器进行图像分类

假如输入特征为 $\boldsymbol{x}$，类别标签为 y 模型。使用神经网络来接收这些特征时，预测输出为 $h(x;\theta)$，用代价函数 $J(\theta)$计算 $h(x;\theta)$与标签 y 之差，再根据代价函数进行反向传播，可使用梯度下降等优化算法优化网络参数θ，如图 1.6 所示。

一般可以将分类器建模为条件概率模型，给定特征输入 $\boldsymbol{x}$ 以后，求标签 y 的概率，公式如下(对于手写数字识别的例子，就是给定一张数字 5 的图像，要求分类器判断该数字是

几)：

$$P(y \mid \boldsymbol{x}) \tag{1.1}$$

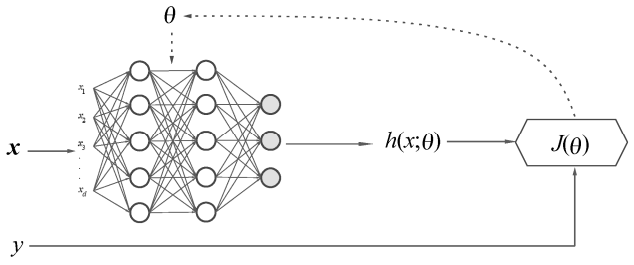

图 1.6　分类器的训练

这就是在给定输入特征 $\boldsymbol{x}$ 的情况下对类别 y 的概率进行建模，特征是从图像中提取的像素。因为是在特定特征集的条件下预测类别 y 的概率，所以公式中有一条竖线，说明是一个条件概率分布。

GAN 判别器是一种分类器，但一般输出的不是多种类别，而是只有两种，所以可以将这样的分类器称为二元分类器。在图 1.7 中，输入一个数字后，不是像一般分类器那样要求判断这张图像中的数字到底是哪一个，而是要求判断这张图像是不是真的手写数字，这里判别器判定 80%是假的。用概率的术语来说，判别器是对给定一组输入样本 $\boldsymbol{x}$ 的真假概率进行建模。

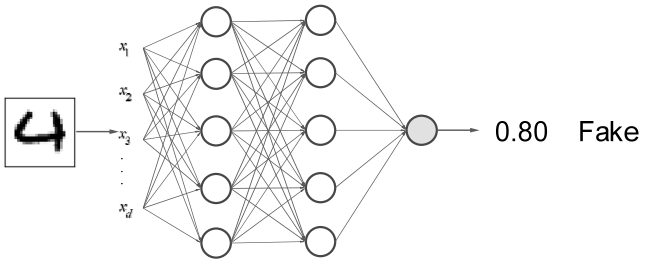

图 1.7　判别器判断示例

总之，判别器是一种分类器，在给定一组输入特征 $\boldsymbol{x}$ 的情况下，学习对样本的真假概率进行建模。判别器的输出概率能帮助生成器学习，以便生成更难以分辨真伪的样本。

1.3.2　生成器

生成器是 GAN 的核心，它是一个用于生成伪造样本的模型，是应该花时间和精力去改

进的网络，这样才能在训练过程结束后获取高性能的输出。下面将重新审视生成器的用途，并讲述如何提高其性能。

生成器的最终目标是能够生成某个特定类别的样本。因此，如果从手写数字的若干图片来训练生成器，那么生成器会进行计算并输出一张看起来很像手写数字的图片，如图 1.8 所示。

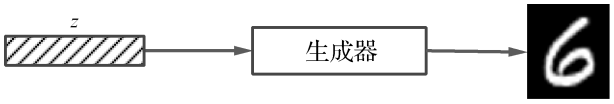

图 1.8　生成器生成示例

一般来说，不希望生成器在每次运行时都输出同一个数字。为了确保每次都能生成不同样本，需要输入不同的随机数，通常称为噪声向量。噪声向量是由一组随机数值组成的向量，一般作为生成器的输入，有时也将类别 y 输入到生成器网络。生成器网络对这些输入进行一系列的非线性计算，最终输出一张手写数字的图像。每个输出单元代表每个像素点的值，本例的输出单元数为 784，更高分辨率的例子甚至会输出高达几百万像素的图像。

不同的噪声向量会输出不同的数字，即便数字相同，其形状也可能不同。图 1.9 展示了噪声 z 可能让生成器生成的数字。

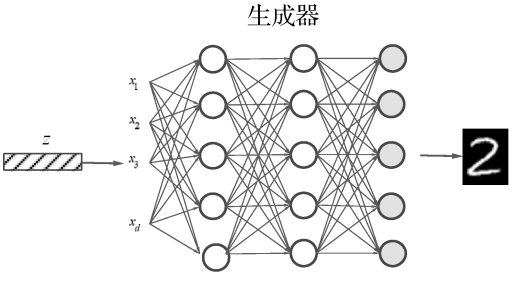

图 1.9　生成器输出示例

现在从概念上考虑如何通过训练来改进生成器，这就是如图 1.10 所示的生成器学习。其过程是：首先将噪声向量 z 输入到生成器网络，用于产生一组特征，这些特征可以构成手写数字图像。然后将这些特征输入到判别器网络中，判别器对它进行检查来确定其真假程度。基本上，生成器希望判别器的输出尽可能接近 1，也就是认定为真；而判别器试图让输出等于 0，也就是认定为假。可以计算一个代价函数来更新生成器的网络参数，使得生成器随着更多次的训练而得到改进，从而欺骗判别器。

如果得到性能不错的生成器，就可以保存生成器的网络参数。将来需要时，可以重新