Chapter 1

预训练网络

1.1 一个识别图像主体的预训练网络

在 Figure 1.1 中，输入图像从左侧进入并依次经过 5 个过滤器，每个过滤器生成一些输出图像。经过每个过滤器后，图像会被缩小。在过滤器堆栈中，最后一个过滤器产生的图像被排列成一个拥有 4096 个元素的一维向量，并被分类以产生 1000 个输出，每个输出对应一个类。

我们已经知道首字母大写的名称对应实现了许多流行模型的 Python 类，而首字母小写的名称是指用预定义的层和单元数实例化模型的函数，可以选择性地下载和加载预先训练好的权重。请注意，使用这些函数没有什么必要：通过它们只是为了方便地使用与预训练好的网络的构建方式相匹配的层和单元来实例化模型。

1.1.1 运行模型

在深度学习中，在新数据上运行训练过的模型的过程被称为推理（inference）。为了进行推理，我们需要将网络置于 eval 模式。如果我们忘记这样做，那么一些预先训练过的模型，如批归一化（Batch Normalization）和 Dropout 将不会产生有意义的答案，这仅仅是因为它们内部工作的方式。

1.2 一个足以以假乱真的预训练模型

1.2.1 GAN 游戏

GAN 是生成式对抗网络（generative adversarial network）的缩写，生成式（generative）意味着一些东西正在被创造出来，而对抗（adversarial）意味着这 2 个网络在竞争，其中一个要比另一个更聪明，而网络意义就显而易见了。

请记住，我们的首要目标是生成不能被识别为赝品的一类图像的合成示例。

请注意，无论是判别器获胜还是生成器获胜，都不应该被理解为字面意义上的获胜，因为二者

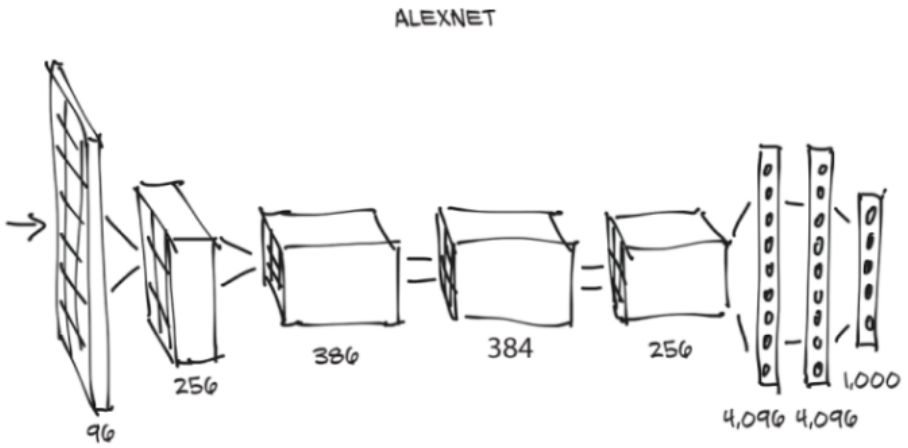

Figure 1.1: AlexNet 架构

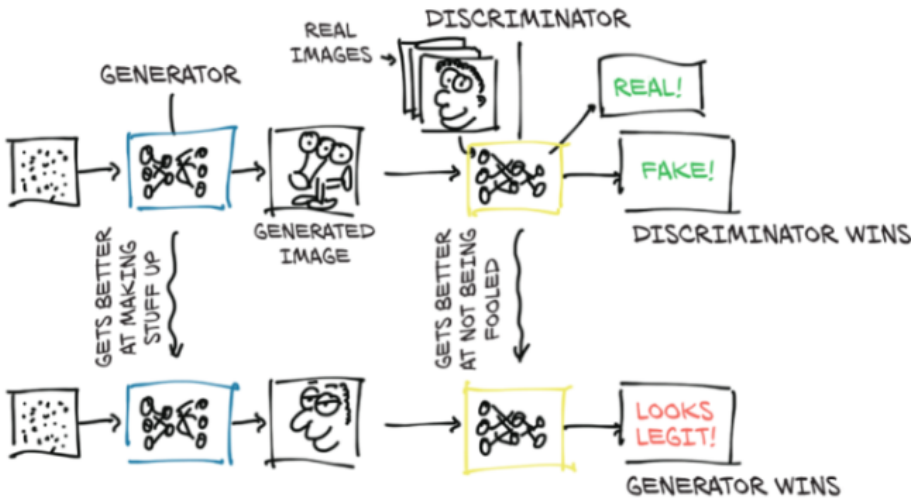

Figure 1.2: GAN 游戏的概念。生成器的最终目标是欺骗判别器，混淆真伪图像。判别器的最终目标是发现它何时被欺骗了，同时告知生成器在生成图像中可识别的错误。例如，一开始，生成器生成模糊的、3 只眼睛的怪物，看起来一点也不像伦勃朗的肖像画。此时判别器很容易把这幅画与真实的画区分开来。随着训练的推进，信息从判别器返回，而生成器使用这些信息进行改进。在训练结束时，生成器可以生成以假乱真的图像了，而判别器却不再能够识别出图像的真伪了。

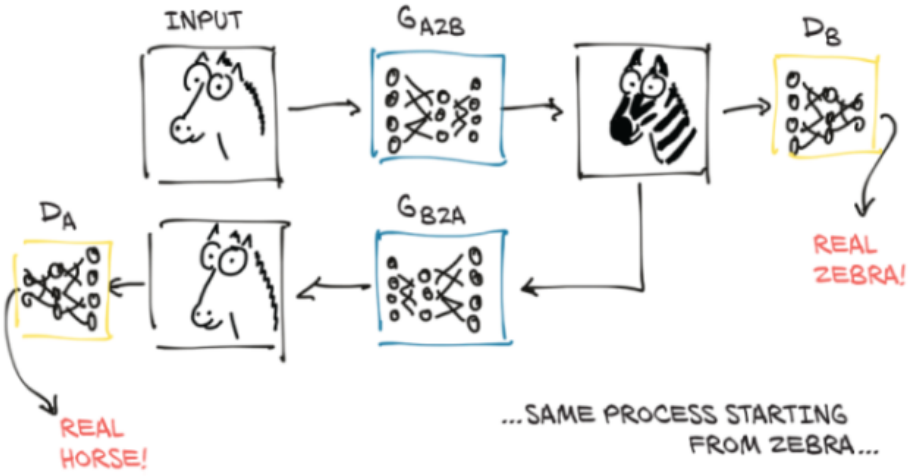

Figure 1.3: 一个经过训练可以欺骗 2 个判别器网络的 CycleGAN。第 1 个生成器学习从属于不同分布域的图像（本例是马），生成符合目标域的图像（本例是斑马）。因此判别器无法分辨出从马的照片中产生的图像是否真的是斑马的图像。同时，产生的假斑马图像通过另一个生成器发送到另一个判别器，由另一个判别器来判别，这就是 CycleGAN 中的 Cycle 前缀的意义所在。创建这样一个循环可以极大地使训练过程稳定，这就解决了 GAN 最初存在的一个问题。

之间没有明确的比赛关系。然而，2 个网络都是基于彼此网络的结果进行训练的，并推动彼此对网络参数进行优化。

1.2.2 CycleGAN

CycleGAN 是循环生成式对抗网络的缩写，它可以将一个领域的图像转换为另一个领域的图像，而不需要我们在训练集中显式地提供匹配对。