

Exercise 1

使用自动编码器执行的主要任务是什么？

答案：

以下是自动编码器用于的一些主要任务：

- 特征提取
- 无监督预训练
- 降维
- 生成模型
- 异常检测（自动编码器通常不擅长重建异常值）

Exercise 2

假设您想要训练一个分类器，并且您有大量未标记的训练数据，但只有几千个已标记的实例。自动编码器如何帮助？你将如何进行？

答案：

如果你想训练一个分类器并且你有大量未标记的训练数据但只有几千个标记实例，那么你可以首先在完整数据集（标记+未标记）上训练深度自动编码器，然后将其下半部分重新用于分类器（即，重用直到编码层的层，包括）并使用标记数据训练分类器。如果您的标记数据很少，您可能希望在训练分类器时冻结重复使用的层。

Exercise 3

如果一个自动编码器完美地重构了输入，那么它一定是一个好的自动编码器吗？你如何评估一个自动编码器的性能？

答案：

自动编码器完美地重构其输入的事实并不一定意味着它是一个好的自动编码器；也许它只是一个过度完整的自动编码器，学会了将其输入复制到编码层，然后再复制到输出。事实上，即使编码层包含单个神经元，非常深的自动编码器也有可能学习将每个训练实例映射到不同的编码（例如，第一个实例可以映射到 0.001，第二个映射到 0.002，第三个到 0.003，依此类推），并且它可以“记住”为每个编码重建正确的训练实例。它可以完美地重建其输入，而无需真正学习数据中的任何有用模式。在实践中，这种映射不太可能发生，但它说明了一个事实，即完美的重建并不能保证自动编码器学到任何有用的东西。然而，如果它产生非常糟糕的重建，那么它几乎可以肯定是一个糟糕的自动编码器。

要评估自动编码器的性能，一种选择是测量重建损失（例如，计算 **MSE**，或输出减去输入的均方）。同样，高重建损失是自动编码器不好的好兆头，但低重建损失并不能保证它是好的。您还应该根据自动编码器的用途来评估它。例如，如果您将其用于分类器的无监督预训练，那么您还应该评估分类器的性能。

Exercise 4

什么是不完整和过度完整的自动编码器？自动编码器的过度不完整的主要风险是什么？那么自动编码器过度完整的主要风险呢？

答案：

不完整的自动编码器是其编码层小于输入和输出层的编码器。如果它更大，那么它就是一个过度完整的自动编码器。过度不完整的自动编码器的主要风险是它可能无法重建输入。过完备自动编码器的主要风险是它可能只是将输入复制到输出，而没有学习任何有用的特征。

Exercise 5

你如何在一个堆叠自动编码器中绑定权重？这样做又有什么意义呢？

答案：

要绑定编码器层及其相应解码器层的权重，只需使解码器权重等于编码器权重的转置即可。

这将模型中的参数数量减少了一半，通常使训练收敛速度更快，训练数据更少，并降低了过度拟合训练集的风险。

Exercise 6

什么是生成式模型？你能说出一种生成式自动编码器的名字吗？

答案：

生成模型是一种能够随机生成类似于训练实例的输出的模型。例如，一旦在 **MNIST** 数据集上成功训练，生成模型可用于随机生成逼真的数字图像。输出分布通常类似于训练数据。例如，由于 **MNIST** 包含每个数字的许多图像，因此生成模型将输出每个数字大致相同数量的图像。

一些生成模型可以参数化——例如，只生成某些类型的输出。生成自动编码器的一个例子是变分自动编码器。

Exercise 7

什么是 GAN？你能说出几个 GAN 非常擅长的任务吗？

答案:

生成对抗网络是由两部分组成的神经网络架构，即生成器和鉴别器，它们具有相反的目标。生成器的目标是生成与训练集中的实例相似的实例，以愚弄鉴别器。鉴别器必须区分真实实例和生成实例。在每次训练迭代中，鉴别器像普通二元分类器一样被训练，然后生成器被训练以最大化鉴别器的误差。

GAN 用于高级图像处理任务，例如超分辨率、着色、图像编辑（用逼真的背景替换对象）、将简单的草图变成逼真的图像，或预测视频中的下一帧。它们还用于扩充数据集（训练其他模型），生成其他类型的数据（例如文本、音频和时间序列），以及识别其他模型的弱点并加以加强。

Exercise 8

训练 GANs 的主要困难是什么？

答案:

众所周知，训练 GAN 非常困难，因为生成器和鉴别器之间存在复杂的动态关系。最大的困难是模式崩溃，生成器产生的输出几乎没有多样性。此外，训练可能非常不稳定：它可能开始时很好，然后突然开始振荡或发散，没有任何明显的原因。GAN 对超参数的选择也非常敏感。

Exercise 9

扩散模型擅长什么？它们的主要限制是什么？

答案:

扩散模型擅长生成多样化和高质量的图像。它们也比 GAN 更容易训练。然而，与 GAN 和 VAE 相比，它们在生成图像时要慢得多，因为它们必须经历反向扩散过程中的每一步。

Exercise 10

尝试使用去噪自动编码器对图像分类器进行预训练。如果您想要一个更大的挑战，您可以使用 MNIST（最简单的选项），也可以使用一个更复杂的图像数据集，如 CIFAR10。无论您使用的是哪个数据集，都请遵循以下步骤：

1. 将数据集分割成一个训练集和一个测试集。在全训练集上训练一个深度去噪的自动编码器。
2. 检查图像是否重建得相当好。可视化最能激活编码层中每个神经元的图像。
3. 构建分类 DNN，重用自动编码器的较低层。仅使用训练集中的 500 张图像对其进行训练。在有或没有预训练的情况下它表现得更好吗？

答案:

```
In [2]: import tensorflow as tf

(X_train, y_train), (X_test, y_test) = tf.keras.datasets.cifar10.load_data()
X_train = X_train / 255
X_test = X_test / 255
```

```
In [3]: tf.random.set_seed(42)

denoising_encoder = tf.keras.Sequential([
    tf.keras.layers.GaussianNoise(0.1),
    tf.keras.layers.Conv2D(32, 3, padding="same", activation="relu"),
    tf.keras.layers.MaxPool2D(),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(512, activation="relu"),
])
```

```
In [4]: denoising_decoder = tf.keras.Sequential([
    tf.keras.layers.Dense(16 * 16 * 32, activation="relu"),
    tf.keras.layers.Reshape([16, 16, 32]),
    tf.keras.layers.Conv2DTranspose(filters=3, kernel_size=3, strides=2,
                                    padding="same", activation="sigmoid")
])
```

```
In [ ]: denoising_ae = tf.keras.Sequential([denoising_encoder, denoising_decoder])

denoising_ae.compile(loss="binary_crossentropy", optimizer="nadam",
                    metrics=["mse"])

history = denoising_ae.fit(X_train, X_train, epochs=10,
                          validation_data=(X_test, X_test))
```

```
In [ ]: n_images = 5
new_images = X_test[:n_images]
new_images_noisy = new_images + np.random.randn(n_images, 32, 32, 3) * 0.1
new_images_denoised = denoising_ae.predict(new_images_noisy)

plt.figure(figsize=(6, n_images * 2))
for index in range(n_images):
    plt.subplot(n_images, 3, index * 3 + 1)
    plt.imshow(new_images[index])
    plt.axis('off')
    if index == 0:
        plt.title("Original")
    plt.subplot(n_images, 3, index * 3 + 2)
    plt.imshow(new_images_noisy[index].clip(0., 1.))
    plt.axis('off')
    if index == 0:
        plt.title("Noisy")
    plt.subplot(n_images, 3, index * 3 + 3)
    plt.imshow(new_images_denoised[index])
    plt.axis('off')
    if index == 0:
```

```
plt.title("Denoised")  
  
plt.show()
```

Exercise 11

在您选择的图像数据集上训练变分自动编码器，并使用它来生成图像。或者，您可以尝试找到您感兴趣的未标记数据集，看看是否可以生成新样本。

Exercise 12

训练一个 DCGAN 来处理您所选择的图像数据集，并使用它来生成图像。添加体验回放，看看这是否有帮助。将其转换为一个 条件 GAN，在那里您可以控制生成的类。

Exercise 13

浏览 KerasCV 的优秀稳定扩散教程，并生成一幅美丽的火蜥蜴读书的图画。如果你在推特上发布了你最好的作品，请标记我 @aureliengeron 。我很想看看你的作品！