机器学习与深度学习面试系列十六（自动编码器）

什么是自编码器（AE）？

自编码器(Auto-Encoder，AE)是通过无监督的方式来学习一组数据的有效编码(或表示)。深度学习大牛LeCun认为这里叫自监督学习（Self-supervised Learning）更为准确。

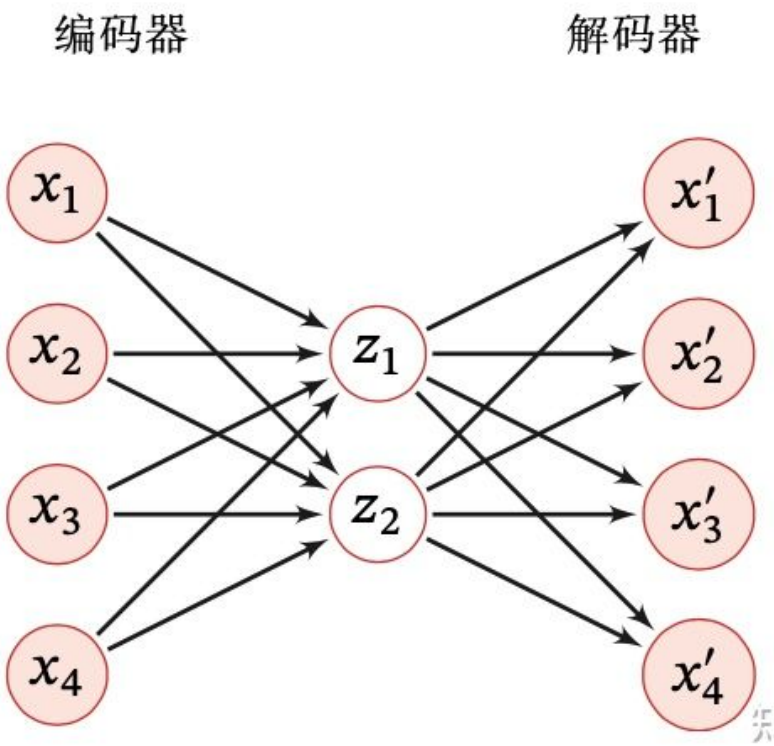
Yann LeCun

30 April 2019 ·

I now call it "self-supervised learning", because "unsupervised" is both a loaded and confusing term.

In self-supervised learning, the system learns to predict part of its input from other parts of it input. In other words a portion of the input is used as a supervisory signal to a predictor fed with the remaining portion of the input.

假设有一组D维的样本 $x^{(n)} \in R^D \quad 1 \leq n \leq N$ ，自编码器将这组数据映射到特征空间得到每个样本的编码 $z^{(n)} \in R^M \quad 1 \leq n \leq N$，并且希望这组编码可以重构出原来的样本。



自编码器的结构可分为两部分: 编码器(Encoder)和解码器(Decoder)。

编码器 $f: R^D \rightarrow R^M$

解码器 $g: R^M \rightarrow R^D$

自编码器的学习目标通常是最小化重构错误(Reconstruction Error)：

$$\mathcal{L} = \sum_{n=1}^{N} ||x^{(n)} - g(f(x^{(n)}))||^2$$

深度自编码器是什么?

对于很多数据来说，仅使用两层神经网络的自编码器还不足以获取一种好的数据表示。为了获取更好的数据表示，我们可以使用更深层的神经网络，即Encoder和Decoder分别具有多层。深层神经网络作为自编码器提取的数据表示一般会更加抽象，能够更好地捕捉到数据的语义信息。

堆叠自编码器是什么?

在实践中经常使用逐层堆叠的方式来训练一个深层的自编码器， 称为堆叠自编码器(Stacked Auto-Encoder，SAE)，又叫深度自编码器（细微区别本文不再区分）。

堆叠自编码器训练方式不是端到端，而是逐层贪婪训练：首先，以第一个编码器和最后一个解码器形成一个浅层自编码器；训练完这些层之后，使用编码层编码整个数据集；然后基于第二个编码器和倒数第二个解码器层形成另一个浅层自编码器，使用编码得到的数据集训练第二个浅层自编码器；重复这一过程，直到到达最内层。最终得到的深度自编码器由许多个浅层自编码器堆叠而成。

欠完备自编码器是什么?

如果 z 的维度M大于等于 x 的维度D，很显然自编码器只需要机械的复制输入到输出就可以使损失函数降为0，即 $g(f(x)) = x$ ，而这通常是没有什么意义的。

所以我们可以限制 编码 z 的维度比 x 小，这种编码维度小于输入维度的自编码器称为欠完备(undercomplete)自编码器。学习欠完备的表示将强制自编码器捕捉训练数据中最显著的特征。

1. 当解码器是线性的且损失函数是均方误差，欠完备的自编码器会学习出与 PCA 相同的生成子空间。这种情况下，自编码器在训练来执行复制任务的同时学到了训练数据的主元子空间。
2. 拥有非线性编码器函数 f 和非线性解码器函数 g 的自编码器能够学习出更强大的 PCA 非线性推广。

正则自编码器是什么?

如果不强制要求 z 的维度M大于等于 x 的维度D，我们可以用某些形式的正则化来强迫网络学习到一些有用的表示。例如：稀疏自编码器、降噪自编码器。

什么是稀疏自编码器?

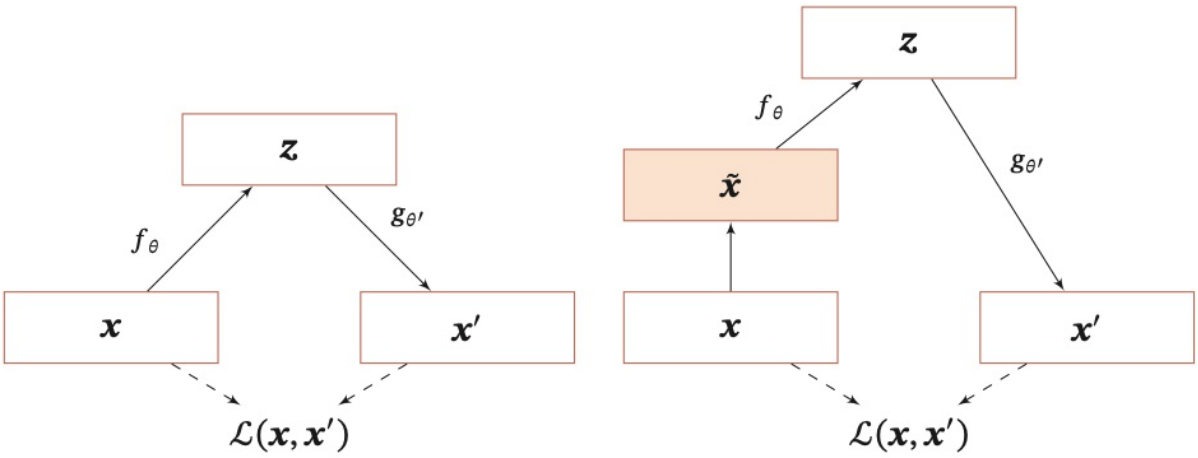
假设中间隐藏层 z 的维度M大于输入样本 x 的维度D，并让 z 尽量稀疏，这就是稀疏自编码器(Sparse Auto-Encoder)。

稀疏自编码器可以通过正则化损失函数轻松实现： $\mathcal{L} = \mathcal{L}_{MSE} + \Omega(z)$ ，稀疏自编码器一般用来学习特征，以便用于像分类这样的任务。稀疏正则化的自编码器必须反映训练数据集的独特统计特征，而不是简单地充当恒等函数。以这种方式训练，执行附带稀疏惩罚的复制任务可以得到能学习有用特征的模型。

什么是降噪自编码器?

我们使用自编码器是为了得到有效的数据表示，而有效的数据表示除了具有最小重构错误或稀疏性等性质之外，还可以要求其具备其他性质，比如对数据部分损坏(Partial Destruction)的鲁棒性。高维数据(比如图像)一般都具有一定的信息冗余，比如我们可以根据一张部分破损的图像联想出其完整内容。因此，我们希望自编码器也能够从部分损坏的数据中得到有效的数据表示，并能够恢复出完整的原始信息。

降噪自编码器(Denoising Auto-Encoder)就是一种通过引入噪声来增加编码鲁棒性的自编码器。例如：对于一个向量 x，我们首先根据一个 比例 u 随机将 x 的一些维度的值设置为0，得到一个被损坏的向量 $\hat{x}$ 。然后将被损坏的向量 $\hat{x}$ 输入给自编码器得到编码 z，并重构出原始的无损输入 x。再比如，我们可以对原始数据 x 随机加一个高斯噪声得到 $\hat{x}$ ，然后使用自编码器尝试重建 x 。注意，这里的损失函数是在重建结果（从噪声输入计算）与原始无噪声数据之间计算的。



(a) 自编码器　　(b) 降噪自编码器

自编码器的Encoder和Decoder是否需要对称?

不需要。在提出自编码器的早期阶段，研究者们确实倾向于Encoder和Decoder是对称结构，不仅层数是对称的，连权重都是对称的。举例来说：

对于编码器： $z = f(W^{(1)}x + b^{(1)})$

对于解码器： $x' = f(W^{(2)}z + b^{(2)})$

通常令 $W^{(2)}$ 等于 $W^{(1)}$ 的转置，即 $W^{(2)} = W^{(1)^T}$ ，这也被称为捆绑权重(Tied Weight)。捆绑权重自编码器的参数更少，因此更容易学习。此外，捆绑权重还在一定程度上起到正则化的作用。

现在自编码器在架构上是对称的情况依然很常见，但是没有这个模型本身没有限制必须对称。例如卷积自编码器，过去通常在编码器使用卷积层，在解码器使用反卷积层，但是现在更常见的是在解码器部分使用上采样层。

自编码器有哪些应用?

1. 数据去噪。
2. 降维、可视化。
3. 异常检测。如果一个输入序列不能被训练好的自编码器良好的重建，视为异常样本。