第十三章　线性因子模型

许多深度学习的研究前沿均涉及构建输入的概率模型 $p_{\text{model}}(\boldsymbol{x})$。原则上说，给定任何其他变量的情况下，这样的模型可以使用概率推断来预测其环境中的任何变量。许多这样的模型还具有潜变量 $\boldsymbol{h}$，其中 $p_{\text{model}}(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{h}} p_{\text{model}}(\boldsymbol{x} \mid \boldsymbol{h})$。这些潜变量提供了表示数据的另一种方式。我们在深度前馈网络和循环网络中已经发现，基于潜变量的分布式表示继承了表示学习的所有优点。

在本章中，我们描述了一些基于潜变量的最简单的概率模型：**线性因子模型**（linear factor model）。这些模型有时被用来作为混合模型的组成模块 (Hinton *et al.*, 1995a; Ghahramani and Hinton, 1996; Roweis *et al.*, 2002) 或者更大的深度概率模型 (Tang *et al.*, 2012)。同时，也介绍了构建生成模型所需的许多基本方法，在此基础上更先进的深度模型也将得到进一步扩展。

线性因子模型通过随机线性解码器函数来定义，该函数通过对 $\boldsymbol{h}$ 的线性变换以及添加噪声来生成 $\boldsymbol{x}$。

有趣的是，通过这些模型我们能够发现一些符合简单联合分布的解释性因子。线性解码器的简单性使得它们成为了最早被广泛研究的潜变量模型。

线性因子模型描述如下的数据生成过程。首先，我们从一个分布中抽取解释性因子 $\boldsymbol{h}$

$$\mathbf{h} \sim p(\boldsymbol{h}), \tag{13.1}$$

其中 $p(\boldsymbol{h})$ 是一个因子分布，满足 $p(\boldsymbol{h}) = \prod_i p(h_i)$，所以易于从中采样。接下来，在给定因子的情况下，我们对实值的可观察变量进行采样

$$\boldsymbol{x} = \boldsymbol{W}\boldsymbol{h} + \boldsymbol{b} + \text{noise}, \tag{13.2}$$

其中噪声通常是对角化的（在维度上是独立的）且服从高斯分布。这在图 13.1 有具

体说明。

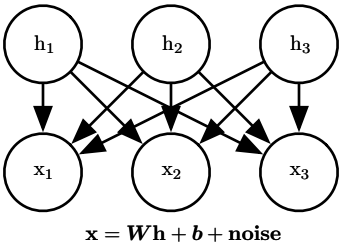

图 13.1: 描述线性因子模型族的有向图模型，其中我们假设观察到的数据向量 $\boldsymbol{x}$ 是通过独立的潜在因子 $\boldsymbol{h}$ 的线性组合再加上一定噪声获得的。不同的模型，比如概率 PCA，因子分析或者是 ICA，都是选择了不同形式的噪声以及先验 $p(\boldsymbol{h})$。

13.1 概率 PCA 和因子分析

概率 PCA（probabilistic PCA）、因子分析和其他线性因子模型是上述等式（式 (13.1) 和式 (13.2)）的特殊情况，并且仅在对观测到 $\boldsymbol{x}$ 之前的噪声分布和潜变量 $\boldsymbol{h}$ 先验的选择上有所不同。

在**因子分析**（factor analysis）(Bartholomew, 1987; Basilevsky, 1994) 中，潜变量的先验是一个方差为单位矩阵的高斯分布

$$\mathbf{h} \sim \mathcal{N}(\boldsymbol{h}; \mathbf{0}, \boldsymbol{I}), \tag{13.3}$$

同时，假定在给定 $\boldsymbol{h}$ 的条件下观察值 x_i 是**条件独立**（conditionally independent）的。具体来说，我们可以假设噪声是从对角协方差矩阵的高斯分布中抽出的，协方差矩阵为 $\boldsymbol{\psi} = \text{diag}(\boldsymbol{\sigma}^2)$，其中 $\boldsymbol{\sigma}^2 = [\sigma_1^2, \sigma_2^2, \ldots, \sigma_n^2]^\top$ 表示一个向量，每个元素表示一个变量的方差。

因此，潜变量的作用是捕获不同观测变量 x_i 之间的依赖关系。实际上，可以容易地看出 $\boldsymbol{x}$ 服从多维正态分布，并满足

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{x}; \boldsymbol{b}, \boldsymbol{W}\boldsymbol{W}^\top + \boldsymbol{\psi}). \tag{13.4}$$

为了将 PCA 引入到概率框架中，我们可以对因子分析模型作轻微修改，使条件方差 σ_i^2 等于同一个值。在这种情况下，$\boldsymbol{x}$ 的协方差简化为 $\boldsymbol{W}\boldsymbol{W}^\top + \sigma^2\boldsymbol{I}$，这里的 σ^2

是一个标量。由此可以得到条件分布，如下：

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{x}; \boldsymbol{b}, \boldsymbol{W}\boldsymbol{W}^\top + \sigma^2 \boldsymbol{I}), \tag{13.5}$$

或者等价地

$$\mathbf{x} = \boldsymbol{W}\mathbf{h} + \boldsymbol{b} + \sigma \mathbf{z}, \tag{13.6}$$

其中 $\mathbf{z} \sim \mathcal{N}(\boldsymbol{z}; \mathbf{0}, \boldsymbol{I})$ 是高斯噪声。之后 Tipping and Bishop (1999) 提出了一种迭代的 EM 算法来估计参数 $\boldsymbol{W}$ 和 σ^2。

这个 **概率 PCA**（probabilistic PCA）模型利用了这样一种观察现象：除了一些微小残余的 **重构误差**（reconstruction error）（至多为 σ^2），数据中的大多数变化可以由潜变量 $\boldsymbol{h}$ 描述。通过Tipping and Bishop (1999) 的研究我们可以发现，当 $\sigma \to 0$ 时，概率 PCA 退化为 PCA。在这种情况下，给定 $\boldsymbol{x}$ 情况下 $\boldsymbol{h}$ 的条件期望等于将 $\boldsymbol{x} - \boldsymbol{b}$ 投影到 $\boldsymbol{W}$ 的 d 列所生成的空间上，与 PCA 一样。

当 $\sigma \to 0$ 时，概率 PCA 所定义的密度函数在 d 维的 $\boldsymbol{W}$ 的列生成空间周围非常尖锐。这导致模型会为没有在一个超平面附近聚集的数据分配非常低的概率。

13.2 独立成分分析

独立成分分析（independent component analysis, ICA）是最古老的表示学习算法之一 (Herault and Ans, 1984; Jutten and Herault, 1991; Comon, 1994; Hyvärinen, 1999; Hyvärinen *et al.*, 2001a; Hinton *et al.*, 2001; Teh *et al.*, 2003)。它是一种建模线性因子的方法，旨在将观察到的信号分离成许多潜在信号，这些潜在信号通过缩放和叠加可以恢复成观察数据。这些信号是完全独立的，而不是仅仅彼此不相关[1]。

许多不同的具体方法被称为 ICA。与我们本书中描述的其他生成模型最相似的 ICA 变种 (Pham *et al.*, 1992) 训练了完全参数化的生成模型。潜在因子 $\boldsymbol{h}$ 的先验 $p(\boldsymbol{h})$，必须由用户提前给出并固定。接着模型确定性地生成 $\boldsymbol{x} = \boldsymbol{W}\boldsymbol{h}$。我们可以通过非线性变化（使用式 (3.47)）来确定 $p(\boldsymbol{x})$。然后通过一般的方法比如最大化似然进行学习。

这种方法的动机是，通过选择一个独立的 $p(\boldsymbol{h})$，我们可以尽可能恢复接近独立的潜在因子。这是一种常用的方法，它并不是用来捕捉高级别的抽象因果因子，而是

[1] 第 3.8 节讨论了不相关变量和独立变量之间的差异。

恢复已经混合在一起的低级别信号。在该设置中，每个训练样本对应一个时刻，每个 x_i 是一个传感器对混合信号的观察值，并且每个 h_i 是单个原始信号的一个估计。例如，我们可能有 n 个人同时说话。如果我们在不同位置放置 n 个不同的麦克风，则 ICA 可以检测每个麦克风的音量变化，并且分离信号，使得每个 h_i 仅包含一个人清楚地说话。这通常用于脑电图的神经科学，这种技术可用于记录源自大脑的电信号。放置在受试者头部上的许多电极传感器用于测量来自身体的多种电信号。实验者通常仅对来自大脑的信号感兴趣，但是来自受试者心脏和眼睛的信号强到足以混淆在受试者头皮处的测量结果。信号到达电极，并且混合在一起，因此为了分离源于心脏与源于大脑的信号，并且将不同脑区域中的信号彼此分离，ICA 是必要的。

如前所述，ICA 存在许多变种。一些版本在 $\boldsymbol{x}$ 的生成中添加一些噪声，而不是使用确定性的解码器。大多数方法不使用最大似然准则，而是旨在使 $\boldsymbol{h} = \boldsymbol{W}^{-1}\boldsymbol{x}$ 的元素彼此独立。许多准则能够达成这个目标。式 (3.47) 需要用到 $\boldsymbol{W}$ 的行列式，这可能是代价很高且数值不稳定的操作。ICA 的一些变种通过将 $\boldsymbol{W}$ 约束为正交来避免这个有问题的操作。

ICA 的所有变种均要求 $p(\boldsymbol{h})$ 是非高斯的。这是因为如果 $p(\boldsymbol{h})$ 是具有高斯分量的独立先验，则 $\boldsymbol{W}$ 是不可识别的。对于许多 $\boldsymbol{W}$ 值，我们可以在 $p(\boldsymbol{x})$ 上获得相同的分布。这与其他线性因子模型有很大的区别，例如概率 PCA 和因子分析通常要求 $p(\boldsymbol{h})$ 是高斯的，以便使模型上的许多操作具有闭式解。在用户明确指定分布的最大似然方法中，一个典型的选择是使用 $p(h_i) = \frac{d}{dh_i}\sigma(h_i)$。这些非高斯分布的典型选择在 0 附近具有比高斯分布更高的峰值，因此我们也可以看到独立成分分析经常用于学习稀疏特征。

按照我们对生成模型这个术语的定义，ICA 的许多变种不是生成模型。在本书中，生成模型可以直接表示 $p(\boldsymbol{x})$，也可以认为是从 $p(\boldsymbol{x})$ 中抽取样本。ICA 的许多变种仅知道如何在 $\boldsymbol{x}$ 和 $\boldsymbol{h}$ 之间变换，而没有任何表示 $p(\boldsymbol{h})$ 的方式，因此也无法在 $p(\boldsymbol{x})$ 上施加分布。例如，许多 ICA 变量旨在增加 $\boldsymbol{h} = \boldsymbol{W}^{-1}\boldsymbol{x}$ 的样本峰度，因为高峰度说明了 $p(\boldsymbol{h})$ 是非高斯的，但这是在没有显式表示 $p(\boldsymbol{h})$ 的情况下完成的。这就是为什么 ICA 多被用作分离信号的分析工具，而不是用于生成数据或估计其密度。

正如 PCA 可以推广到第十四章中描述的非线性自编码器，ICA 也可以推广到非线性生成模型，其中我们使用非线性函数 f 来生成观测数据。关于非线性 ICA 最初的工作可以参考 Hyvärinen and Pajunen (1999)，它和集成学习的成功结合可以参见 Roberts and Everson (2001); Lappalainen *et al.* (2000)。ICA 的另一个非线性扩展是**非线性独立成分估计**（nonlinear independent components estimation, NICE）

方法 (Dinh *et al.*, 2014)，这个方法堆叠了一系列可逆变换（在编码器阶段），其特性是能高效地计算每个变换的 Jacobian 行列式。这使得我们能够精确地计算似然，并且像 ICA 一样，NICE 尝试将数据变换到具有因子的边缘分布的空间。由于非线性编码器的使用，这种方法更可能成功。因为编码器和一个能进行完美逆变换的解码器相关联，所以可以直接从模型生成样本（首先从 $p(\boldsymbol{h})$ 采样，然后使用解码器）。

ICA 的另一个推广是通过鼓励组内统计依赖关系、抑制组间依赖关系来学习特征组 (Hyvärinen and Hoyer, 1999; Hyvärinen *et al.*, 2001b)。当相关单元的组被选为不重叠时，这被称为 **独立子空间分析**（independent subspace analysis）。我们还可以向每个隐藏单元分配空间坐标，并且空间上相邻的单元组形成一定程度的重叠。这能够鼓励相邻的单元学习类似的特征。当应用于自然图像时，这种 **地质 ICA**（topographic ICA）方法可以学习 Gabor 滤波器，从而使得相邻特征具有相似的方向、位置或频率。在每个区域内出现类似 Gabor 函数的许多不同相位存在抵消作用，使得在小区域上的池化产生了平移不变性。

13.3 慢特征分析

慢特征分析（slow feature analysis, SFA）是使用来自时间信号的信息学习不变特征的线性因子模型 (Wiskott and Sejnowski, 2002)。

慢特征分析的想法源于所谓的 **慢性原则**（slowness principle）。其基本思想是，与场景中起描述作用的单个量度相比，场景的重要特性通常变化得非常缓慢。例如，在计算机视觉中，单个像素值可以非常快速地改变。如果斑马从左到右移动穿过图像并且它的条纹穿过对应的像素时，该像素将迅速从黑色变为白色，并再次恢复成黑色。通过比较，指示斑马是否在图像中的特征将不发生改变，并且描述斑马位置的特征将缓慢地改变。因此，我们可能希望将模型正则化，从而能够学习到那些随时间变化较为缓慢的特征。

慢性原则早于慢特征分析，并已被应用于各种模型 (Hinton, 1989; Földiák, 1989; Mobahi *et al.*, 2009; Bergstra and Bengio, 2009)。一般来说，我们可以将慢性原则应用于可以使用梯度下降训练的任何可微分模型。为了引入慢性原则，我们可以向代价函数添加以下项

$$\lambda \sum_t L(f(\boldsymbol{x}^{(t+1)}), f(\boldsymbol{x}^{(t)})), \tag{13.7}$$

其中 λ 是确定慢度正则化强度的超参数项，t 是样本时间序列的索引，f 是需要正则化的特征提取器，L 是测量 $f(\boldsymbol{x}^{(t)})$ 和 $f(\boldsymbol{x}^{(t+1)})$ 之间的距离的损失函数。L 的一个常见选择是均方误差。

慢特征分析是慢性原则中一个特别高效的应用。由于它被应用于线性特征提取器，并且可以通过闭式解训练，所以它是高效的。像 ICA 的一些变种一样，SFA 本身并不是生成模型，只是在输入空间和特征空间之间定义了一个线性映射，但是没有定义特征空间的先验，因此没有在输入空间上施加分布 $p(\boldsymbol{x})$。

SFA 算法 (Wiskott and Sejnowski, 2002) 先将 $f(\boldsymbol{x};\theta)$ 定义为线性变换，然后求解如下优化问题

$$\min_{\boldsymbol{\theta}} \mathbb{E}_t (f(\boldsymbol{x}^{(t+1)})_i - f(\boldsymbol{x}^{(t)})_i)^2 \tag{13.8}$$

并且满足下面的约束：

$$\mathbb{E}_t f(\boldsymbol{x}^{(t)})_i = 0 \tag{13.9}$$

以及

$$\mathbb{E}_t [f(\boldsymbol{x}^{(t)})_i^2] = 1. \tag{13.10}$$

学习特征具有零均值的约束对于使问题具有唯一解是必要的; 否则我们可以向所有特征值添加一个常数，并获得具有相等慢度目标值的不同解。特征具有单位方差的约束对于防止所有特征趋近于 0 的病态解是必要的。与 PCA 类似，SFA 特征是有序的，其中学习第一特征是最慢的。要学习多个特征，我们还必须添加约束

$$\forall i < j, \quad \mathbb{E}_t [f(\boldsymbol{x}^{(t)})_i f(\boldsymbol{x}^{(t)})_j] = 0. \tag{13.11}$$

这要求学习的特征必须彼此线性去相关。没有这个约束，所有学习到的特征将简单地捕获一个最慢的信号。可以想象使用其他机制，如最小化重构误差，也可以迫使特征多样化。但是由于 SFA 特征的线性，这种去相关机制只能得到一种简单的解。SFA 问题可以通过线性代数软件获得闭式解。

在运行 SFA 之前，SFA 通常通过对 $\boldsymbol{x}$ 使用非线性的基扩充来学习非线性特征。例如，通常用 $\boldsymbol{x}$ 的二次基扩充来代替原来的 $\boldsymbol{x}$，得到一个包含所有 $x_i x_j$ 的向量。由此，我们可以通过反复地学习一个线性 SFA 特征提取器，对其输出应用非线性基扩展，然后在该扩展之上学习另一个线性 SFA 特征提取器的方式来组合线性 SFA 模块从而学习深度非线性慢特征提取器。

当在自然场景视频的小块空间部分上训练时，使用二次基扩展的 SFA 所学习到的特征与 V1 皮层中那些复杂细胞的特征有许多共同特性 (Berkes and Wiskott, 2005)。当在计算机渲染的 3D 环境内随机运动的视频上训练时，深度 SFA 模型能够学习的特征与大鼠脑中用于导航的神经元学到的特征有许多共同特性 (Franzius *et al.*, 2007)。因此从生物学角度上来说 SFA 是一个合理的有依据的模型。

SFA 的一个主要优点是，即使在深度非线性条件下，它依然能够在理论上预测 SFA 能够学习哪些特征。为了做出这样的理论预测，必须知道关于配置空间的环境动力（例如，在 3D 渲染环境中随机运动的例子中，理论分析是从相机位置、速度的概率分布中入手的）。已知潜在因子如何改变的情况下，我们能够通过理论分析解出表达这些因子的最佳函数。在实践中，基于模拟数据的实验上，使用深度 SFA 似乎能够恢复理论预测的函数。相比之下，在其他学习算法中，代价函数高度依赖于特定像素值，使得难以确定模型将学习到什么特征。

深度 SFA 也已经被用于学习用在对象识别和姿态估计的特征 (Franzius *et al.*, 2008)。到目前为止，慢性原则尚未成为任何最先进应用的基础。究竟是什么因素限制了其性能仍有待研究。我们推测，或许慢度先验太过强势，并且，最好添加这样一个先验使得当前时间步到下一个时间步的预测更加容易，而不是加一个先验使得特征近似为一个常数。对象的位置是一个有用的特征，无论对象的速度是高还是低。但慢性原则鼓励模型忽略具有高速度的对象的位置。

13.4 稀疏编码

稀疏编码（sparse coding）(Olshausen and Field, 1996) 是一个线性因子模型，已作为一种无监督特征学习和特征提取机制得到了广泛研究。严格来说，术语 “稀疏编码’’ 是指在该模型中推断 $\boldsymbol{h}$ 值的过程，而 ‘‘稀疏建模’’ 是指设计和学习模型的过程，但是通常这两个概念都可以用术语 “稀疏编码’’ 描述。

像大多数其他线性因子模型一样，它使用了线性的解码器加上噪声的方式获得一个 $\boldsymbol{x}$ 的重构，就像式 (13.2) 描述的一样。更具体地说，稀疏编码模型通常假设线性因子有一个各向同性精度为 β 的高斯噪声：

$$p(\boldsymbol{x} \mid \boldsymbol{h}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{W}\boldsymbol{h} + \boldsymbol{b}, \frac{1}{\beta}\boldsymbol{I}). \tag{13.12}$$

分布 $p(\boldsymbol{h})$ 通常选取为一个峰值很尖锐且接近 0 的分布 (Olshausen and Field,

1996)。常见的选择包括可分解的 Laplace、Cauchy 或者可分解的 Student-t 分布。例如，以稀疏惩罚系数 λ 为参数的 Laplace 先验可以表示为

$$p(h_i) = \text{Laplace}(h_i; 0, \frac{2}{\lambda}) = \frac{\lambda}{4} \mathrm{e}^{-\frac{1}{2}\lambda|h_i|}, \tag{13.13}$$

相应的，Student-t 先验分布可以表示为

$$p(h_i) \propto \frac{1}{(1 + \frac{h_i^2}{\nu})^{\frac{\nu+1}{2}}}. \tag{13.14}$$

使用最大似然的方法来训练稀疏编码模型是不可行的。相反，为了在给定编码的情况下更好地重构数据，训练过程在编码数据和训练解码器之间交替进行。稍后在第 19.3 节中，这种方法将被进一步证明为是解决最大似然问题的一种通用的近似方法。

对于诸如 PCA 的模型，我们已经看到使用了预测 $\boldsymbol{h}$ 的参数化的编码器函数，并且该函数仅包括乘以权重矩阵。稀疏编码中的编码器不是参数化的编码器。相反，编码器是一个优化算法，在这个优化问题中，我们寻找单个最可能的编码值：

$$\boldsymbol{h}^* = f(\boldsymbol{x}) = \arg\max_{\boldsymbol{h}} p(\boldsymbol{h} \mid \boldsymbol{x}). \tag{13.15}$$

结合式 (13.13) 和式 (13.12)，我们得到如下的优化问题：

$$\arg\max_{\boldsymbol{h}} p(\boldsymbol{h} \mid \boldsymbol{x}) \tag{13.16}$$

$$= \arg\max_{\boldsymbol{h}} \log p(\boldsymbol{h} \mid \boldsymbol{x}) \tag{13.17}$$

$$= \arg\min_{\boldsymbol{h}} \lambda \|\boldsymbol{h}\|_1 + \beta \|\boldsymbol{x} - \boldsymbol{W}\boldsymbol{h}\|_2^2, \tag{13.18}$$

其中，我们扔掉了与 $\boldsymbol{h}$ 无关的项，并除以一个正的缩放因子来简化表达。

由于在 $\boldsymbol{h}$ 上施加 L^1 范数，这个过程将产生稀疏的 $\boldsymbol{h}^*$（详见第 7.1.2 节）。

为了训练模型而不仅仅是进行推断，我们交替迭代关于 $\boldsymbol{h}$ 和 $\boldsymbol{W}$ 的最小化过程。在本文中，我们将 β 视为超参数。我们通常将其设置为 1，因为它在此优化问题的作用与 λ 类似，没有必要使用两个超参数。原则上，我们还可以将 β 作为模型的参数，并学习它。我们在这里已经放弃了一些不依赖于 $\boldsymbol{h}$ 但依赖于 β 的项。要学习 β，必须包含这些项，否则 β 将退化为 0。

不是所有的稀疏编码方法都显式地构建了一个 $p(\boldsymbol{h})$ 和一个 $p(\boldsymbol{x} \mid \boldsymbol{h})$。通常我们只是对学习一个带有激活值的特征的字典感兴趣，当特征是由这个推断过程提取时，这个激活值通常为 0。

如果我们从 Laplace 先验中采样 $\boldsymbol{h}$，$\boldsymbol{h}$ 的元素实际上为 0 是一个零概率事件。生成模型本身并不稀疏，只有特征提取器是稀疏的。Goodfellow *et al.* (2013f) 描述了不同模型族中的近似推断，如尖峰和平板稀疏编码模型，其中先验的样本通常包含许多真正的 0。

与非参数编码器结合的稀疏编码方法原则上可以比任何特定的参数化编码器更好地最小化重构误差和对数先验的组合。另一个优点是编码器没有泛化误差。参数化的编码器必须泛化地学习如何将 $\boldsymbol{x}$ 映射到 $\boldsymbol{h}$。对于与训练数据差异很大的异常 $\boldsymbol{x}$，所学习的参数化编码器可能无法找到对应精确重构或稀疏的编码 $\boldsymbol{h}$。对于稀疏编码模型的绝大多数形式，推断问题是凸的，优化过程总能找到最优编码（除非出现退化的情况，例如重复的权重向量）。显然，稀疏和重构成本仍然可以在不熟悉的点上升，但这归因于解码器权重中的泛化误差，而不是编码器中的泛化误差。当稀疏编码用作分类器的特征提取器，而不是使用参数化的函数来预测编码值时，基于优化的稀疏编码模型的编码过程中较小的泛化误差可以得到更好的泛化能力。Coates and Ng (2011) 证明了在对象识别任务中稀疏编码特征比基于参数化的编码器（线性-sigmoid 自编码器）的特征拥有更好的泛化能力。受他们的工作启发，Goodfellow *et al.* (2013f) 表明一种稀疏编码的变体在标签极少（每类 20 个或更少标签）的情况中比相同情况下的其他特征提取器拥有更好的泛化能力。

非参数编码器的主要缺点是在给定 $\boldsymbol{x}$ 的情况下需要大量的时间来计算 $\boldsymbol{h}$，因为非参数方法需要运行迭代算法。在第十四章中讲到的参数化自编码器方法仅使用固定数量的层，通常只有一层。另一个缺点是它不直接通过非参数编码器进行反向传播，这使得我们很难采用先使用无监督方式预训练稀疏编码模型然后使用监督方式对其进行精调的方法。允许近似导数的稀疏编码模型的修改版本确实存在但未被广泛使用 (Bagnell and Bradley, 2009)。

像其他线性因子模型一样，稀疏编码经常产生糟糕的样本，如图 13.2 所示。即使当模型能够很好地重构数据并为分类器提供有用的特征时，也会发生这种情况。这种现象发生的原因是每个单独的特征可以很好地被学习到，但是隐藏编码值的因子先验会导致模型包括每个生成样本中所有特征的随机子集。这促使人们开发更深的模型，可以在其中最深的编码层施加一个非因子分布，与此同时也在开发一些复杂的浅度模型。

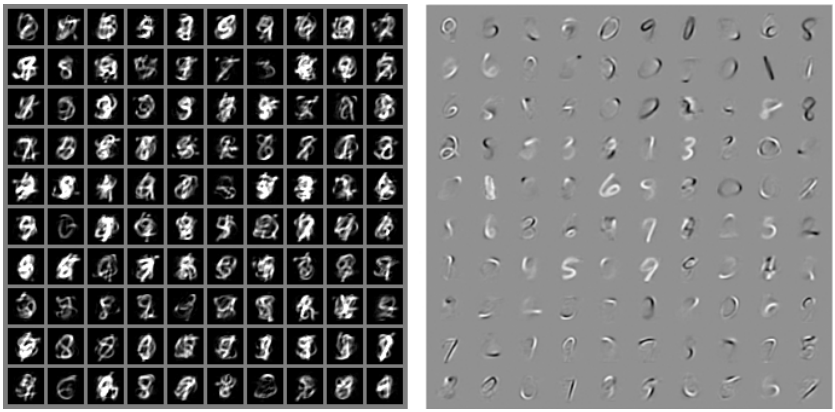

图 13.2: 尖峰和平板稀疏编码模型上在 MNIST 数据集训练的样例和权重。*(左)* 这个模型中的样本和训练样本相差很大。第一眼看来，我们可能认为模型拟合得很差。*(右)* 这个模型的权重向量已经学习到了如何表示笔迹，有时候还能写完整的数字。因此这个模型也学习到了有用的特征。问题在于特征的因子先验会导致特征子集合随机的组合。一些这样的子集能够合成可识别的 MNIST 集上的数字。这也促进了拥有更强大潜在编码分布的生成模型的发展。此图经 Goodfellow *et al.* (2013f) 允许转载。

13.5 PCA的流形解释

线性因子模型，包括 PCA 和因子分析，可以理解为学习一个流形 (Hinton *et al.*, 1997)。我们可以将概率 PCA 定义为高概率的薄饼状区域，即一个高斯分布，沿着某些轴非常窄，就像薄饼沿着其垂直轴非常平坦，但沿着其他轴是细长的，正如薄饼在其水平轴方向是很宽的一样。图 13.3 解释了这种现象。PCA 可以理解为将该薄饼与更高维空间中的线性流形对准。这种解释不仅适用于传统 PCA，而且适用于学习矩阵 $\boldsymbol{W}$ 和 $\boldsymbol{V}$ 的任何线性自编码器，其目的是使重构的 $\boldsymbol{x}$ 尽可能接近于原始的 $\boldsymbol{x}$。

编码器表示为

$$\boldsymbol{h} = f(\boldsymbol{x}) = \boldsymbol{W}^{\top}(\boldsymbol{x} - \boldsymbol{\mu}). \tag{13.19}$$

编码器计算 h 的低维表示。从自编码器的角度来看，解码器负责计算重构：

$$\hat{\boldsymbol{x}} = g(\boldsymbol{h}) = \boldsymbol{b} + \boldsymbol{V}\boldsymbol{h}. \tag{13.20}$$

能够最小化重构误差

$$\mathbb{E}[\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2] \tag{13.21}$$

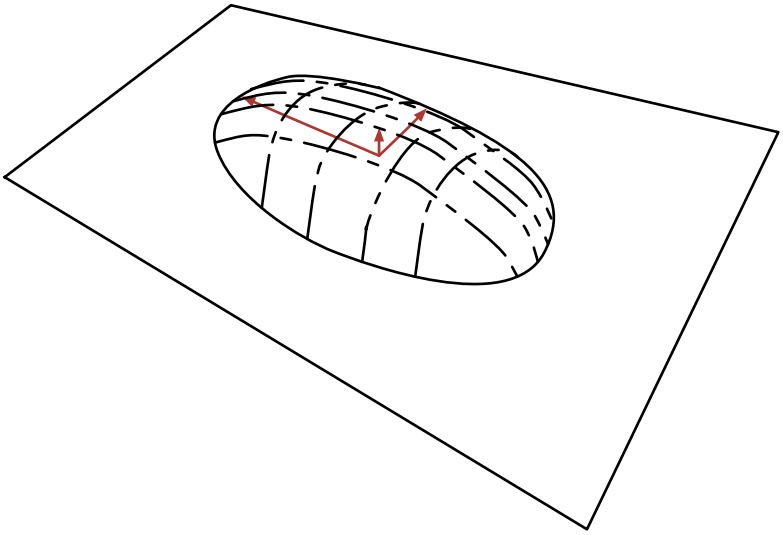

图 13.3: 平坦的高斯能够描述一个低维流形附近的概率密度。此图表示了 “流形平面” 上 “馅饼” 的上半部分，并且这个平面穿过了馅饼的中心。正交于流形方向（指向平面外的箭头方向）的方差非常小，可以被视作是 “噪声”，其他方向（平面内的箭头）的方差则很大，对应了 “信号” 以及降维数据的坐标系统。

的线性编码器和解码器的选择对应着 $\boldsymbol{V} = \boldsymbol{W}$，$\boldsymbol{\mu} = \boldsymbol{b} = \mathbb{E}[\boldsymbol{x}]$，$\boldsymbol{W}$ 的列形成一组标准正交基，这组基生成的子空间与协方差矩阵 $\boldsymbol{C}$

$$\boldsymbol{C} = \mathbb{E}[(\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^\top] \tag{13.22}$$

的主特征向量所生成的子空间相同。在 PCA 中，$\boldsymbol{W}$ 的列是按照对应特征值（其全部是实数和非负数）幅度大小排序所对应的特征向量。

我们还可以发现 $\boldsymbol{C}$ 的特征值 λ_i 对应了 $\boldsymbol{x}$ 在特征向量 $\boldsymbol{v}^{(i)}$ 方向上的方差。如果 $\boldsymbol{x} \in \mathbb{R}^D$，$\boldsymbol{h} \in \mathbb{R}^d$ 并且满足 $d < D$，则（给定上述的 $\boldsymbol{\mu}, \boldsymbol{b}, \boldsymbol{V}, \boldsymbol{W}$ 的情况下）最佳的重构误差是

$$\min \mathbb{E}[\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2] = \sum_{i=d+1}^{D} \lambda_i. \tag{13.23}$$

因此，如果协方差矩阵的秩为 d，则特征值 λ_{d+1} 到 λ_D 都为 0，并且重构误差为 0。

此外，我们还可以证明上述解可以通过在给定正交矩阵 $\boldsymbol{W}$ 的情况下最大化 $\boldsymbol{h}$ 元素的方差而不是最小化重构误差来获得。