第二章 数据驱动的数据再表达学习方法

数据驱动的数据再表达方法主要是指通过一定的样本学习得到满足要求的数据再表达的形式，我们称之为数据驱动的数据再表达学习。下面我们先给出定义，然后介绍各种模型和方法。

定义 2.1 首先建立训练架构，然后利用大量的训练样本，**通过学习训练得到给定**架构的参数，从而得到揭示输入数据潜在变化因素的数据再表达方法称为数据驱动的数据再表达方法。

除了对数据再表达一般要求能够分布式表达和不变性以外，我们希望数据再表达能够解开输入的样本数据隐藏的潜在的和不同的解释因素。输入样本数据的不同解释因素倾向于在输入分布中彼此独立地变化，并且当连续输入实际数据序列时，人们倾向于认为一次只有少数几个因素发生改变。

复杂数据来源的因素丰富，各个因素之间的作用相互交叉。这些因素一般在现实复杂的网络中相互作用，这可能使得与人工智能相关的任务（如目标分类或识别）变得复杂化。

例如，图像由一个或多个光源之间的相互作用，物体形状和图像中存在着各种表面的材料特性并相互作用而组成。场景中物体的阴影可以以复杂的模式相互交织在一起，产生出没有物体边界的错觉，并显著影响着感知的物体形状。如何应对这些复杂的相互交叉的作用？如何将物体和它们的阴影区分开？最终，克服这些挑战的方法还是必

须利用数据本身，使用大量未标记的样本数据，学习得到分离各种解释性来源的数据表示。这样做可以使对于人工智能相关任务的自然数据源中存在的复杂且结构变化丰富的数据具有更强大的表达。

重要的是要区分与学习相关的但对不同任务目标的不变特征或者变化不敏感特征，并学会解开解释因素。主要不同在于对信息的保留。根据定义，不变特征在不变性方向上变化不敏感。这是我们构建对数据变化不敏感特征的目标，这些特征对于我们将要完成的任务没有任何相关的信息。不幸的是，通常很难有先验知识知道哪组特征具有不变性并与我们将要完成的任务相关。

此外，如在深度学习方法的上下文中经常出现的情况那样，被训练的特征集合要用于可能具有相关特征的不同子集的多个任务中。诸如此类的考虑使我们得出结论，最有效的特征学习方法是尽可能多地解开因素，尽可能少地丢弃实际数据的有关信息。如果需要某种形式的降维，那么我们假设应该首先修剪在训练数据中有最少表示的局部变化方向（例如，在主成分分析中，它在全局而不是在每个训练样本周围进行修剪）。

数据驱动的数据再表达学习（简称数据再表达学习）的挑战之一是将其与后端的机器学习任务（如分类、识别）区分开来之后，难以建立明确的目标或训练目标。在分类的情况下，目标（至少在概念上）是明确的，我们希望最小化训练数据集上的错误分类的数量。在数据再表达学习的情况下，我们的目标与最终目标相去甚远，最终目标通常是学习分类器或其它预测器。我们的问题让人联想到强化学习中遇

到的同样问题：强化学习从环境状态到行为的映射，其目标是使得智能体选择的行为能够获得环境最大的奖赏，使得外部环境对学习系统在某种意义下的评价（或整个系统的运行性能）为最佳。

我们认为，良好的数据再表达能够解决变化的潜在因素，但如何将其转化为适当的学习模型与求解方法仍然是一个非常困难的问题。

数据驱动的数据再表达的学习目的是希望解开输入的样本数据隐藏的潜在的和不同的解释因素，其模型大致可以分成以下几种类型：

- 非概率模型
- 概率模型
- 概率图模型
- 流形上的学习模型

（1）非概率模型是通过学习得到决策函数 $y = f(x)$ 来进行数据再表达；

（2）概率模型是利用训练样本数据，通过学习条件概率分布 $p(x|y)$ 来进行数据再表达；

（3）概率图模型是由概率模型演化而来，它是用图结构来描述多元随机变量之间条件独立关系的概率模型；

（4）流形学习就是从高维采样数据中恢复低维流形结构，即找到高维空间中的低维流形，并求出相应的嵌入映射，以实现维数约简或者数据可视化，从而达到数据再表达的目标。

在后面三节内容就分别介绍这几种类型的数据再表达方法。

2.1 单层学习模块与深层学习模块

为了通过学习获得数据的良好表达，需要建立从输入到输出的学习模块。这些学习模块开始是单层的。单层学习模块既可以单独使用，也可以堆积变成多层学习。

所谓单层学习模块是指仅通过一层学习获得输入数据的再表达，也即相当于在数学上对输入数据只进行一次变换。训练阶段和数据再表达阶段分别见图 2.1 和 2.2。

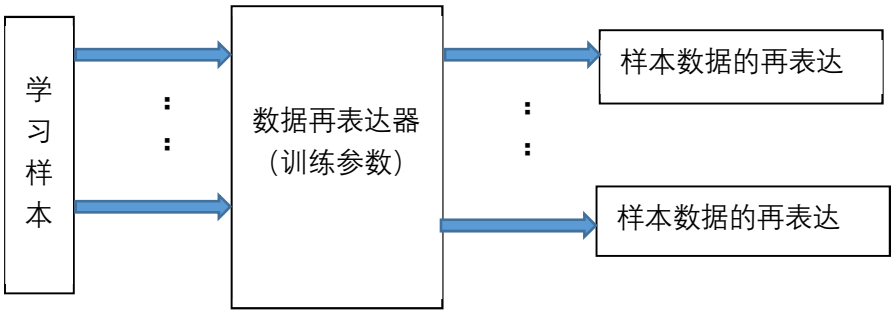

图 2.1 单层学习模块训练阶段

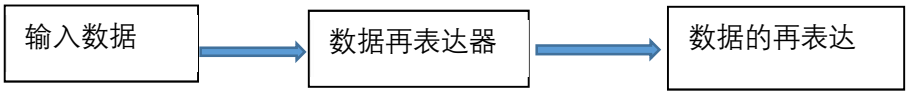

图 2.2 单层学习模块数据再表达阶段

单层数据驱动的数据再表达学习有两个常用的模型：非概率模型和概率模型。

在数据驱动的数据再表达学习中，概率模型是指利用训练样本数据，通过学习条件概率分布 $p(x|y)$ 来得到观察数据的分布。

非概率模型是指利用训练样本数据通过学习得到变换 $y=f(x)$，

从而实现数据的再表达。

神经网络模型是一种典型的非概率模型。

从根本上说，这两个模型的区别在于深度学习模型的分层体系结构是被解释为描述**概率图模型**还是描述**计算图模型**。简而言之，**被隐藏的单元被认为是潜在的随机变量还是计算节点**。虽然二者有很大的区别，但最新研究进展表明，它们的共同点都集中在单层模块的贪婪学习特性以及单层模块类型之间相似特性上。

概率模型代表性的例子是受限制的 Boltzmann 机（RBM），而非概率模型代表性的例子是自动编码器。事实上，在单层模型中，正如 Vincent[1]和 Swersky 等人[2]所示，在受限制的玻尔兹曼机的情况下，用得分匹配的归纳原理来训练模型与将正则化重建目标应用于自动编码器基本上是相同的。

下面我们以主成分分析为例，从概率、自动编码器和流形学习三个视角来解释单层无监督数据再表达学习算法。

主成分分析是一个最古老的特征提取算法。主成分分析算法学习一个从输入 $x \in \mathbb{R}^n$ 到输出 $h \in \mathbb{R}^m$ 的线性变换 $h = f(x) = W^T x + b$，其中矩阵 $W \in \mathbb{R}^{n \times m}$ 是正交矩阵，它的列构成了训练数据中最大方差的 m 个正交方向的正交基。

主成分分析的原理是设法将原来变量重新组合成一组新的相互无关的几个综合变量，同时根据实际需要从中可以取出几个较少的综合变量，尽可能多地反映原来变量的信息的统计方法。

主成分分析是设法将原来众多具有一定相关性的指标，重新组合

成一组新的互相无关的综合指标来代替原来的指标。

主成分分析的三种解释如下：

（1）它与概率模型有关，如概率主成分分析，因子分析和传统的多元高斯分布（协方差矩阵的主要特征向量是主要成分）；

（2）它学习的数据再表达与基本线性自动编码器学习得到的数据再表达基本相同；

（3）它可以看作线性流形学习的简单线性形式，即刻画一个数据密度达到峰值的输入空间中的低维区域。

因此，主成分分析可以作为一个共同的主线来理解不同类型的单层学习模型。但是，主成分分析只是一个线性模型，而线性特征的表现力非常有限：它们不能被堆栈成多层，以形成更深、更抽象的数据再表达，因为多个线变换的组合产生的只是另一个线性变换。所以，必须研究用于提取非线性特征的算法，这些算法可以堆栈成深层网络而形成更抽象的数据再表达。与主成分分析关系密切的该类算法是 Jutten 和 Herault[3]、 Bell 和 Sejnowski[4]、 Hyv¨arinen[5][6]提出的独立成分分析（ICA）。值得注意的是，虽然在最简单的完全无噪声情况下 ICA 产生线性特征，但在更一般的情况下，它可以等同于具有非高斯独立的潜在变量的线性生成模型，类似于稀疏编码，这导致了提取的是非线性特征。因此，ICA 及其变形，如独立和拓扑图 ICA（Hyv¨arinen 等[7]）可以并且已经被用于构建深层网络。获得独立成分的概念似乎与我们通过深层网络解开潜在解释因素的既定目标相似。然而，对于复杂的现实世界分布，真正独立的潜在因素与观察到

的高维数据之间的关系可以通过线性变换充分表达还是值得怀疑的。

2.1.2 建立深度表示

为了获得更抽象的数据再表达形式，需要构建多层的数据再表达学习结构训练阶段和数据再表达阶段分别见图 2.3 和 2.4。

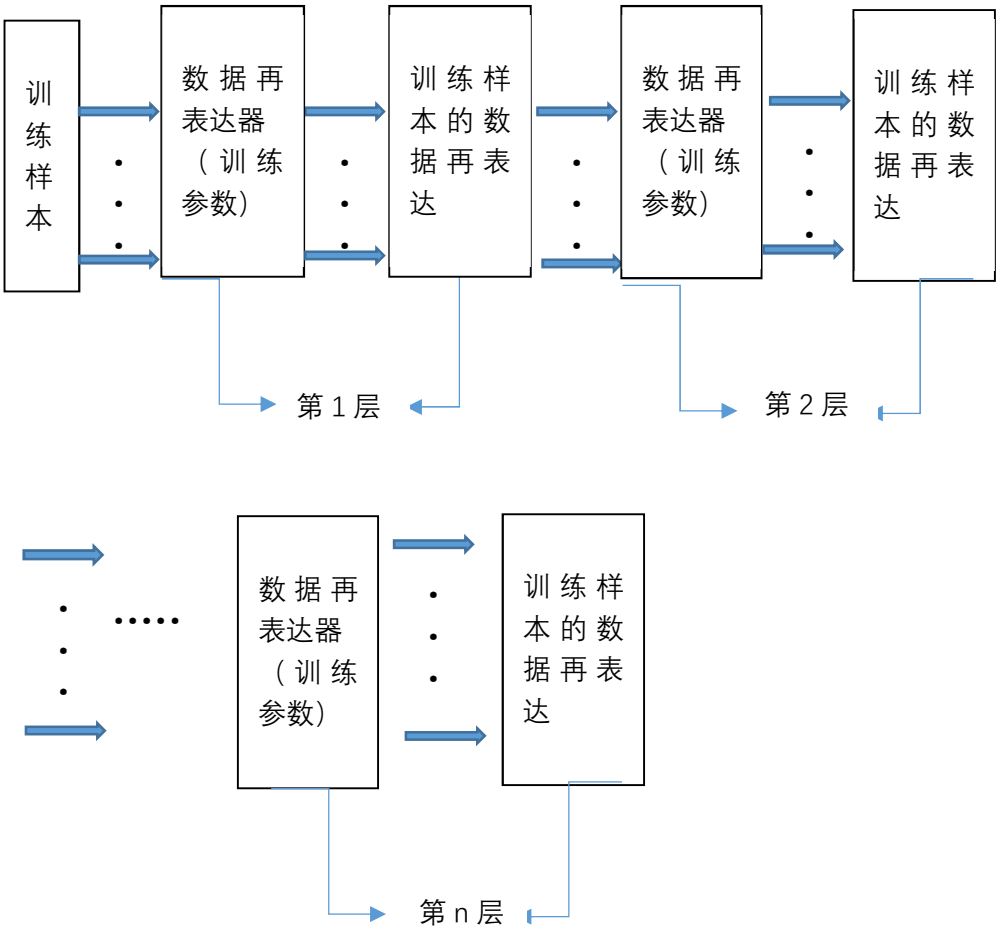

图 2.3 多层学习模块训练阶段

数据再表达阶段：

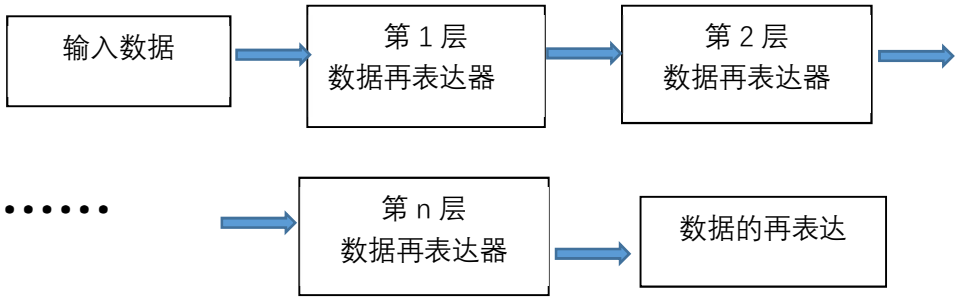

图 2.4 多层学习模块数据再表达阶段

2006 年，Hinton 等人[8]开始在特征学习和深度学习方面取得突破，并得到了快速发展。被称为**逐层贪婪无监督预训练，**其中心思想是一次学习一个特征层，使用无监督特征学习，在每个层次上学习一个新变换，类似于数学上若干变换组成复合变换；基本上，无监督特征学习的每次迭代都会向深度神经网络添加一层权重。最后，可以组合这些图层以初始化深度监督预测器，例如神经网络分类器，或深度生成模型，或深度波尔茨曼机（Salakhutdinov 和 Hinton[9]）。

我们主要关注可用于形成深层体系结构的数据再表达学习算法。特别地，根据经验观察到，逐层堆栈得到的数据再表达通常产生更好的表示，例如，在分类误差方面（Larochelle 等[10]； Erhan 等[11]）以及模型（Salakhutdinov 和 Hinton[12]）或学习特征的不变性方面 Goodfellow 等[13]。

下面我们简要介绍建立深层模型的概念，后面将介绍具体算法。在贪婪的分层无监督预训练之后，得到的深度特征既可以用作后端学习的标准监督机器学习预测器（例如 SVM）的输入，也可以用作深度监督神经网络的初始化（例如，通过附加逻辑回归层或多层神经网络的纯监督层）。分层程序也可以应用于纯监督环境，称为贪婪层监督预训练（Bengio et al[14]）。例如，在训练第一个隐藏层的多层感知器（MLP：Multi-Layer Perceptron）之后，丢弃其输出层，并且可以在其上堆栈成另一个单隐藏层 MLP 等

另一种改进（Seide 等[15]）是以监督的方式在迭代的每个步骤中预先训练所有先前添加的层，这种改进产生的预训练结果更好的。

尽管将单层组合成监督模型是直截了当的，但是不清楚应该组合多少层、如何组合无监督学习预训练层以形成更好的无监督模型。这仍然是一个困难的问题。最初的建议是将预先训练好的 RBM 堆叠成深置信网络（Hinton 等人[8]）或 DBN，其中顶层被解释为 RBM 而下层被解释为定向的 Sigmoid 置信网络。然而，目前尚不清楚如何近似最大似然训练以进一步优化这种生成模型。