第3章

状态特征的提取与迁移

3.1 状态特征提取概述

在装备故障诊断中，故障与征兆之间往往并不是简单的一一对应关系，当进行诸如异常检测、故障诊断等运维服务时，直接采用原始状态参数往往很难取得良好的效果。利用现代信号处理理论、方法和技术手段，对原始监测数据信号进行信号分离、特征提取、模式分类是装备故障诊断的前提。特征提取前首先需要进行特征选择。特征选择是指在原始数据空间中选择一些重要的特征，例如振动信号的振幅、相位等，这些特征能反映装备运行过程中的某些状态，基于这些特征可以实现装备的状态评价和故障诊断。状态特征提取方法可分为线性方法与非线性方法两大类。考虑到线性方法在处理复杂的非线性问题时往往不能取得理想的效果，本章主要介绍几种常用的非线性特征提取方法，包括核主元分析[1]、自动编码器、深度学习、迁移学习[2]等。

3.2 基于核主元分析的状态特征提取

常用的主元分析(principal component analysis，PCA)[3]通过线性变换输入变量的方法达到降维的目的。一些学者提出用线性逼近非线性的方法对 PCA 方法进行改进，如广义 PCA[4]、主曲线方法[5]、神经网络 PCA 方法[6]等，但这些方法对非线性问题的解决并不准确，且涉及复杂的算法变换问题。

在工程实际中，特征参数的变化往往呈现非线性，所以有必要采取非线性多元统计分析方法进行特征提取和分析。

基于核函数的主元分析方法(kernel PCA，KPCA)将输入数据映射到一个新的空间，这个过程是通过选定一个非线性函数实现的，然后在新空间中进行线性分析。该方法对于非线性数据特别有效，能提供更多的特征信息，并且提取的特征的识别效果更优[7]。核函数主元分析在机械设备状态和故障诊断应用中处于起步阶段。

3.2.1 主元分析的算法与分析

主元分析是一种对数据进行分析的技术，最重要的应用是对原有数据进行简化，可以有效找出数据中最“主要”的元素和结构，去除噪声和冗余，将原有的复杂数据降维，揭示隐藏在复杂数据背后的简单结构。主元分析在原始数据空间基础上通过构造一组新的变量代替原变量，新变量的维数低于原始数据，新变量中包含原变量的特征信息，从而大大降低了投影空间的维数[8-9]。由于投影空间中特征向量相互垂直，变量之间的相关性被消除，独立性增强。

设 $\boldsymbol{x}=[x_1,x_2,\cdots,x_p]^{\mathrm{T}}$ 为一个 p 维总体，假设 $\boldsymbol{x}$ 的期望和协方差矩阵均存在并已知，记 $E(\boldsymbol{x})=\mu$，$\mathrm{var}(\boldsymbol{x})=\boldsymbol{\Sigma}$，考虑如下线性变换[10]：

$$\begin{cases} y_1=a_{11}x_1+a_{12}x_2+\cdots+a_{1p}x_p=\boldsymbol{a}'_1\boldsymbol{x} \\ y_2=a_{21}x_1+a_{22}x_2+\cdots+a_{2p}x_p=\boldsymbol{a}'_2\boldsymbol{x} \\ \vdots \\ y_p=a_{p1}x_1+a_{p2}x_2+\cdots+a_{pp}x_p=\boldsymbol{a}'_p\boldsymbol{x} \end{cases} \tag{3-1}$$

式中，$\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_p$ 均为单位向量。下面求 $\boldsymbol{a}_1$，使得 y_1 的方差达到最大。

设 $\lambda_1,\lambda_2,\cdots,\lambda_p(\lambda_1\geqslant\lambda_2\geqslant\cdots\geqslant\lambda_p\geqslant0)$ 为 $\boldsymbol{\Sigma}$ 的 p 个特征值，$\boldsymbol{t}_1,\boldsymbol{t}_2,\cdots,\boldsymbol{t}_p$ 为相应的正交单位特征向量，即

$$\boldsymbol{\Sigma}\boldsymbol{t}_i=\lambda_i\boldsymbol{t}_i,\boldsymbol{t}'_i\boldsymbol{t}_i=1,\quad \boldsymbol{t}'_i\boldsymbol{t}_j=0,\quad i\neq j;\ i,j=1,2,\cdots,p$$

由矩阵知识可知

$$\boldsymbol{\Sigma}=\boldsymbol{T}\boldsymbol{\Lambda}\boldsymbol{T}'=\sum_{i=1}^{p}\lambda_i\boldsymbol{t}_i\boldsymbol{t}'_i$$

式中，$\boldsymbol{T}=[t_1,t_2,\cdots,t_p]$ 为正交矩阵，$\boldsymbol{\Lambda}$ 是对角线元素为 $\lambda_1,\lambda_2,\cdots,\lambda_p$ 的对角阵。考虑 y_1 的方差：

$$\begin{aligned} \mathrm{var}(y_1)&=\mathrm{var}(\boldsymbol{a}'_1\boldsymbol{x})=\boldsymbol{a}'_1\mathrm{var}(\boldsymbol{x})\boldsymbol{a}_1=\sum_{i=1}^{p}\lambda_i\boldsymbol{a}'_1\boldsymbol{t}_i\boldsymbol{t}'_i\boldsymbol{a}_1=\sum_{i=1}^{p}\lambda_i(\boldsymbol{a}'_1\boldsymbol{t}_i)^2 \\ &\leqslant\lambda_1\sum_{i=1}^{p}(\boldsymbol{a}'_1\boldsymbol{t}_i)^2=\lambda_1\boldsymbol{a}'_1\left(\sum_{i=1}^{p}\boldsymbol{t}_i\boldsymbol{t}'_i\right)\boldsymbol{a}_1=\lambda_1\boldsymbol{a}'_1\boldsymbol{T}\boldsymbol{T}'\boldsymbol{a}_1 \\ &=\lambda_1\boldsymbol{a}'_1\boldsymbol{a}_1=\lambda_1 \end{aligned} \tag{3-2}$$

由式(3-2)可知，当 $\boldsymbol{a}_1=\boldsymbol{t}_1$ 时，$y_1=\boldsymbol{t}'_1\boldsymbol{x}$ 的方差达到最大，最大值为 λ_1。称 $y_1=\boldsymbol{t}'_1\boldsymbol{x}$ 为第一主成分。类似的，如果第一主成分从原始数据中提取的信息还不够，还应考虑第二、……、第 i 主成分。

总方差中第 i 个主成分 y_i 的方差所占的比例 $\lambda_i/\sum_{j=1}^{p}\lambda_i(i=1,2,\cdots,p)$ 称为主成分 y_i 的贡献率。主成分的贡献率反映了主成分综合原始变量信息的能力或解释原始变量的能力。由贡献率的定义可知，p 个主成分的贡献率依次递减。前

$m(m\leqslant p)$个主成分的贡献率之和 $\sum_{i=1}^{m}\lambda_i/\sum_{j=1}^{p}\lambda_j$ 称为前 m 个主成分的累积贡献率，它反映了前 m 个主成分综合原始变量信息的能力。由于主成分分析的主要目的是降维，所以需要在信息损失不太多的情况下，用少数几个主成分来代替原始变量 $x_1,x_2,\cdots,x_p$，以进行后续的分析。通常的做法是取较小的 m，使得前 m 个主成分的累积贡献率不低于某一水平，这样就可以达到降维目的[11]。

3.2.2　主元中核函数的引入

基于核函数的主元分析方法(KPCA)的基本思想是[12]通过一个选定的映射$\boldsymbol{\Phi}$将输入样本 $\boldsymbol{x}$ 变换到其他空间，成为$\boldsymbol{\Phi}(\boldsymbol{x})$，然后对$\boldsymbol{\Phi}(\boldsymbol{x})$利用 PCA 线性特征提取方法进行计算，将非线性问题变换为线性问题，可用图 3-1 描述其过程。

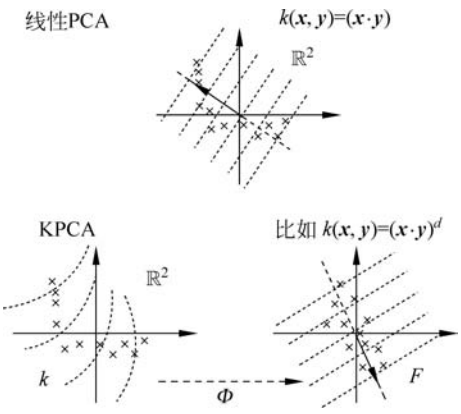

图 3-1　PCA/KPCA 空间转换示意图

假设 $x_1,x_2,\cdots,x_N$ 为训练样本，用$\{x_i\}$表示输入空间。选择的变换函数为$\boldsymbol{\Phi}$，变换到特征空间需要满足的条件为

$$\sum_{i=1}^{N}\boldsymbol{\Phi}(x_i)=\boldsymbol{0} \tag{3-3}$$

求其协方差矩阵为

$$\boldsymbol{\Sigma}=\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\Phi}(x_i)\boldsymbol{\Phi}(x_i)^{\mathrm{T}} \tag{3-4}$$

若不满足式(3-3)的条件，其操作步骤详见文献[13]，可认为 $\boldsymbol{u}_i$ 位于$\boldsymbol{\Phi}(x_1)$，$\boldsymbol{\Phi}(x_2),\cdots,\boldsymbol{\Phi}(x_N)$张成的子空间中，即

$$\boldsymbol{u}_i=\sum_{j=1}^{N}\alpha_j^i\boldsymbol{\Phi}(x_j) \tag{3-5}$$

式中，$\boldsymbol{\alpha}^i=[\alpha_1^i,\alpha_2^i,\cdots,\alpha_N^i]^{\mathrm{T}}$。

事实上，式(3-5)对应特征空间中的目标函数可表达为如下拉格朗日函数：

$$g=\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\alpha}^{i\,\mathrm{T}}\boldsymbol{K}\boldsymbol{K}^{\mathrm{T}}\boldsymbol{\alpha}^i-\sum_{i=1}^{N}\lambda_i\boldsymbol{\alpha}^{i\,\mathrm{T}}\boldsymbol{K}\boldsymbol{\alpha}^i+\sum_{i=1}^{N}\lambda_i \tag{3-6}$$

式中，$(\boldsymbol{K})_{ij}=k(x_i,x_j)=\boldsymbol{\Phi}^{\mathrm{T}}(x_i)\boldsymbol{\Phi}(x_j)$，$g$ 取极值时，需满足

$$\frac{1}{N}\boldsymbol{K}^2\boldsymbol{\alpha}^i=\lambda_i\boldsymbol{K}\boldsymbol{\alpha}^i \tag{3-7}$$

令 $\lambda'=N\lambda_i$，则式(3-7)等价于如下特征方程：

$$\boldsymbol{K}\boldsymbol{\alpha}=\lambda'\boldsymbol{\alpha} \tag{3-8}$$

PCA 与 KPCA 方法的区别在于：在 PCA 分析中，新的特征是原始特征的线

性组合，代表点到直线的最小距离；在 KPCA 分析中，是通过选定的变换函数将原特征映射到新的空间形成新特征，代表点到曲面的最小距离。

3.2.3 核主元分析特征提取的形式化描述

由于映射的非线性，因此 KPCA 是一种非线性主元分析方法。如果原始数据存在复杂的非线性关系，相比主元分析而言，非线性主元分析更适合用作对其进行特征抽取。KPCA 即是一种非常成功的非线性主元分析方法。

根据式(3-8) 计算训练样本集$\{\boldsymbol{\Phi}(x_i)\}$的特征值为$\lambda_1,\lambda_2,\cdots,\lambda_m(m\leqslant N)$，相应的特征向量为$\boldsymbol{\alpha}^1,\boldsymbol{\alpha}^2,\cdots,\boldsymbol{\alpha}^m$，并假设特征空间中单位变换轴为$\boldsymbol{u}^1,\boldsymbol{u}^2,\cdots,\boldsymbol{u}^m$，则

$$\boldsymbol{u}^i=\frac{1}{\sqrt{\lambda_i}}\sum_{j=1}^{N}\boldsymbol{\alpha}_j^i\boldsymbol{\Phi}(x_j),\quad i=1,2,\cdots,m \tag{3-9}$$

变换轴$\boldsymbol{u}^i(i=1,2,\cdots,m)$的单位性($(\boldsymbol{u}^i)^{\mathrm{T}}\boldsymbol{u}^i=1$)，$\boldsymbol{u}^i$ 与 $\boldsymbol{u}^j$ 的正交性显而易见。基于式(3-9)，可以计算出特征空间中样本$\boldsymbol{\Phi}(x)$在 $\boldsymbol{u}^i$ 上投影的计算式，样本 $\boldsymbol{x}$ 在特征空间中的特征抽取结果为

$$\boldsymbol{y}=\left[\frac{1}{\sqrt{\lambda_1}}\sum_{j=1}^{N}\boldsymbol{\alpha}_j^1k(x_j,\boldsymbol{x}),\frac{1}{\sqrt{\lambda_2}}\sum_{j=1}^{N}\boldsymbol{\alpha}_j^2k(x_j,\boldsymbol{x}),\cdots,\frac{1}{\sqrt{\lambda_m}}\sum_{j=1}^{N}\boldsymbol{\alpha}_j^mk(x_j,\boldsymbol{x})\right]^{\mathrm{T}} \tag{3-10}$$

在应用中，根据实际情况选定 m 值，根据式(3-10)给出的特征提取结果进行故障分类。

3.2.4 核主元分析算法的改进

应用 KPCA 进行特征提取时，对系统的计算能力要求很高，这是因为需计算样本间的核函数，并加权求和，如果样本较多，系统的效率就会下降[14]。若能对 KPCA 方法的效率进行提升，将对实际应用非常重要[15]。

在用训练样本表示式(3-9)中的 $\boldsymbol{u}^i$ 时，不同样本的贡献率不一样，某些样本占较大权重，而另一些则相反。计算权重较小的样本耗费了计算时间，但对计算结果影响不大，若能从整体样本中找出对逼近最优变换轴影响大的那部分样本，则可减少 KPCA 特征提取的计算量。

因此，降低计算量的关键是确定找出重要样本的依据，根据特征方程特征值的大小判断训练样本在逼近最优变换轴方面的贡献率大小是可行的方案。特征值越大，相应最优变换轴对原数据的逼近程度就越强，采用该数据进行故障分类时包含原数据的信息越多。

假设

$$\boldsymbol{u}^i\approx\sum_{j=1}^{s}\beta_j\boldsymbol{\Phi}(x'_j),\quad s<N \tag{3-11}$$

式中，$\boldsymbol{\Phi}(x'_j)$来自训练样本集，称为特征空间中的节点。

令$\boldsymbol{\beta}^{(i)}=[\beta_1^{(i)},\beta_2^{(i)},\cdots,\beta_N^{(i)}]^{\mathrm{T}}$，则式(3-6)变形为

$$g=\frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{\beta}^{(i)})^{\mathrm{T}}\boldsymbol{K}_1\boldsymbol{K}_1^{\mathrm{T}}\boldsymbol{\beta}^{(i)}-\sum_{i=1}^{N}\lambda_i(\boldsymbol{\beta}^{(i)})^{\mathrm{T}}\boldsymbol{K}_2\boldsymbol{\beta}^{(i)}+\sum_{i=1}^{N}\lambda_i \tag{3-12}$$

式中

$$\boldsymbol{K}_1=\begin{bmatrix}k(x'_1,x_1) & \cdots & k(x'_1,x_l)\\ k(x'_2,x_1) & \cdots & k(x'_2,x_l)\\ \vdots & & \vdots\\ k(x'_s,x_1) & \cdots & k(x'_s,x_l)\end{bmatrix},\quad \boldsymbol{K}_2=\begin{bmatrix}k(x'_1,x'_1) & \cdots & k(x'_1,x'_s)\\ k(x'_2,x'_1) & \cdots & k(x'_2,x'_s)\\ \vdots & & \vdots\\ k(x'_s,x'_1) & \cdots & k(x'_s,x'_s)\end{bmatrix}$$

将式(3-12)对$\boldsymbol{\beta}^{(i)}$求导，可得如下广义特征方程：

$$\frac{1}{N}\boldsymbol{K}_1\boldsymbol{K}_1^{\mathrm{T}}\boldsymbol{\beta}^{(i)}=\lambda_i\boldsymbol{K}_2\boldsymbol{\beta}^{(i)} \tag{3-13}$$

在$\boldsymbol{K}_2$可逆条件下，令$\lambda'_i=N\lambda_i$，该特征方程可改写为

$$\boldsymbol{K}_2^{-1}\boldsymbol{K}_1\boldsymbol{K}_1^{\mathrm{T}}\boldsymbol{\beta}=\lambda'_i\boldsymbol{\beta}^{(i)}$$

将$\boldsymbol{\Phi}(x)$在这 m 个最优变换轴上的投影值组成向量，则特征空间中样本$\boldsymbol{\Phi}(x)$基于改进的 KPCA 方法的特征抽取结果为

$$\boldsymbol{y}=\left[\frac{1}{\sqrt{\lambda'_1}}\sum_{j=1}^{s}\beta_j^{(1)}k(x'_j,\boldsymbol{x}),\frac{1}{\sqrt{\lambda'_2}}\sum_{j=1}^{s}\beta_j^{(2)}k(x'_j,\boldsymbol{x}),\cdots,\frac{1}{\sqrt{\lambda'_m}}\sum_{j=1}^{s}\beta_j^{(m)}k(x'_j,\boldsymbol{x})\right]^{\mathrm{T}} \tag{3-14}$$

式中，$\lambda'_1,\lambda'_2,\cdots,\lambda'_m$为式(3-13)的前 m 个最大特征值；$\boldsymbol{\beta}^{(1)},\boldsymbol{\beta}^{(2)},\cdots,\boldsymbol{\beta}^{(m)}$为分别对应这 m 个特征值的特征向量，$\beta_j^{(i)}$ 为向量$\boldsymbol{\beta}^{(i)}$的第 j 维分量。

3.3　基于自动编码器的状态特征提取

自动编码器(autoencoder)是一种神经网络，属于无监督学习模型，能够对高维数据进行有效的特征提取和特征表示。从 1988 年提出以来[16]，在基础形式的基础上，还出现了一些变种，如去噪自动编码器、稀疏自动编码器、收缩自动编码器等。

3.3.1　自动编码器

自动编码器包含编码器和解码器两部分，其模型结构如图 3-2 所示。

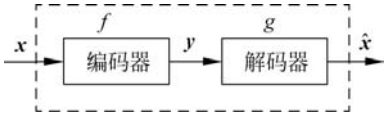

图 3-2　自动编码器的结构

编码器将输入样本 $\boldsymbol{x}$ 从原始特征空间映射到抽象特征空间中的样本 $\boldsymbol{y}$，而解码器将样本 $\boldsymbol{y}$ 从抽象特征空间映射回原始特征空间得到重构样本 $\hat{\boldsymbol{x}}$。模型的学习过程为通过最小化一个损失函数 $\boldsymbol{J}_{\mathrm{AE}}$ 来同时优化编码

器和解码器，从而学习得到针对输入样本 $\boldsymbol{x}$ 的抽象特征表示 $\boldsymbol{y}$，其中 $\boldsymbol{J}_{\mathrm{AE}}$ 为 $\hat{\boldsymbol{x}}$ 和 $\boldsymbol{x}$ 的重构误差。

可以发现，自动编码器在训练过程中无须使用样本标签，这种无监督的学习方式大大提升了模型的通用性。但如果自动编码器只是简单学会将输入样本 $\boldsymbol{x}$ 复制到 $\boldsymbol{y}$，那么自动编码器将没有什么用处[17]。为了能够从自动编码器中获得有用的特征，可以限制 $\boldsymbol{y}$ 的维度低于 $\boldsymbol{x}$ 的维度，这样将强制自动编码器提取训练样本中最显著的特征。此时，当解码器是线性的且重构误差取均方误差时，自动编码器会学习出与主成分分析相同的生成子空间。因此，拥有非线性编码器函数 f 和非线性解码器函数 g 的自动编码器可以看成主成分分析的非线性推广。存在的问题是如果编码器和解码器被赋予过大的容量，自动编码器会执行复制任务而难以提取到有效的抽象特征。

3.3.2 去噪自动编码器

为了使得自动编码器在编码器和解码器容量过大时仍能提取到有效特征，一个方法是在输入样本 $\boldsymbol{x}$ 中加入随机噪声，这就是去噪自动编码器（denoising autoencoder，DAE）[18]。目前添加噪声的方法主要有两种：添加服从特定分布的随机噪声；随机将输入样本 $\boldsymbol{x}$ 中的分量按特定比例置为 0。下面结合第二种方法进行介绍。去噪自动编码器由输入样本污染过程、编码器和解码器组成，其基本结构如图 3-3 所示。

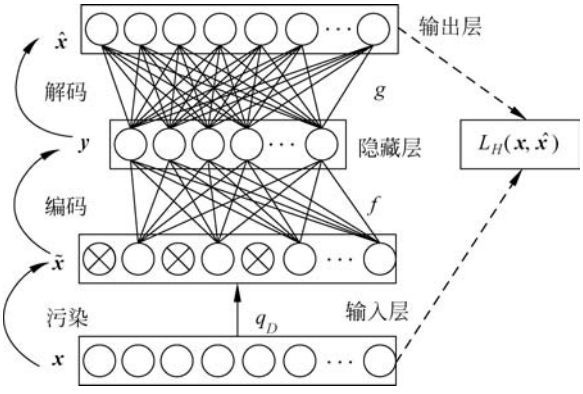

图 3-3 去噪自动编码器的结构

从图 3-3 中可以看出，去噪自动编码器包含输入层、隐藏层和输出层。若用 $\boldsymbol{x}\subseteq\mathbb{R}^n$ 表示原始输入数据，$\tilde{\boldsymbol{x}}$ 表示污染之后的输入数据，$\boldsymbol{y}\subseteq\mathbb{R}^m$ 表示隐藏层数据，$\hat{\boldsymbol{x}}\subseteq\mathbb{R}^n$ 表示输出层数据，则去噪自动编码器的工作过程如下。

1. 污染过程

这是指在输入层中将原始输入数据 $\boldsymbol{x}$ 通过函数 $q_D(\boldsymbol{x})$ 污染成 $\tilde{\boldsymbol{x}}$ 的过程。污染

函数 $q_D(\boldsymbol{x})$为

$$\tilde{\boldsymbol{x}} = q_D(\boldsymbol{x}) \tag{3-15}$$

式中，$q_D(\cdot)$为随机匹配函数；其过程为在输入数据 $\boldsymbol{x}$ 中随机选择 $\nu\%$的样本并将其值设置为 0，$0<\nu<100$。

2. 编码过程

这是指将污染后的输入数据 $\tilde{\boldsymbol{x}}$ 经过编码函数 f_θ 映射到隐藏层 $\boldsymbol{y}$ 的过程。其编码过程中的非线性映射函数 f_θ 为

$$\boldsymbol{y} = f_\theta(\tilde{\boldsymbol{x}}) = S(\boldsymbol{W}\cdot\tilde{\boldsymbol{x}} + \boldsymbol{b}) \tag{3-16}$$

式中，参数$\theta = \{\boldsymbol{W}, \boldsymbol{b}\}$，其中 $\boldsymbol{W}$ 是一个 $m\times n$ 的权重矩阵，$\boldsymbol{b}\subseteq\mathbb{R}^m$ 是偏置向量；$\boldsymbol{y}\subseteq\mathbb{R}^m$ 表示隐藏层；$S(\cdot)$为节点激活函数，通常为 ReLU 函数：

$$S(x) = \begin{cases} x, & x > 0 \\ 0, & x \leqslant 0 \end{cases} \tag{3-17}$$

3. 解码过程

这是指将隐藏层 $\boldsymbol{y}$ 经过解码函数 $g_{\theta'}$ 重构得到向量 $\hat{\boldsymbol{x}}$ 的过程。其解码过程中的非线性映射函数 $g_{\theta'}$ 为

$$\hat{\boldsymbol{x}} = g_{\theta'}(y) = S(\boldsymbol{W}'\cdot\boldsymbol{y} + \boldsymbol{b}') \tag{3-18}$$

式中，参数$\theta' = \{\boldsymbol{W}', \boldsymbol{b}'\}$，其中 $\boldsymbol{W}' = \boldsymbol{W}^{\mathrm{T}}$ 是一个 $n\times m$ 的权重矩阵，$\boldsymbol{b}'\subseteq\mathbb{R}^n$ 是偏置向量；$\hat{\boldsymbol{x}}\subseteq\mathbb{R}^n$ 表示输出层。

4. 寻找最优参数

去噪自动编码器可以利用反向传播算法寻找最优参数$\{\theta, \theta'\} = \{\boldsymbol{W}, \boldsymbol{b}, \boldsymbol{W}', \boldsymbol{b}'\}$，使输出数据 $\hat{\boldsymbol{x}}$ 与输入数据 $\boldsymbol{x}$ 之间的重构误差最小。这里使用均方误差表示输入数据 $\boldsymbol{x}$ 与输出数据 $\hat{\boldsymbol{x}}$ 之间的重构误差：

$$\boldsymbol{J}_{\mathrm{DAE}}(\boldsymbol{\theta}, \boldsymbol{\theta}') = \sum_{x\in\mathbb{R}^n} \| \boldsymbol{x} - \hat{\boldsymbol{x}} \|^2 = \sum_{x\in\mathbb{R}^n} \| \boldsymbol{x} - g_{\theta'}(f_\theta(\tilde{\boldsymbol{x}})) \|^2 \tag{3-19}$$

式中，$\boldsymbol{J}_{\mathrm{DAE}}$ 是模型输入数据 $\boldsymbol{x}$ 与输出数据 $\hat{\boldsymbol{x}}$ 之间的重构误差。

最优参数$\{\theta, \theta'\}$确定后，去噪自动编码器中的隐藏层 $\boldsymbol{y}$ 即为提取到的抽象特征。

3.3.3　稀疏自动编码器

保证自动编码器在编码器和解码器容量过大时仍能提取到有效特征的另一个方法是对自动编码器的隐藏层增加稀疏性约束，即在自动编码器的损失函数中加入一个控制稀疏性的正则项，这样就得到了稀疏自动编码器(sparse autoencoder，SAE)[19]。稀疏自动编码器假设稀疏的表示往往比其他的表示更有效。稀疏性约束能够迫使编码器只有部分神经元被激活。如果激活函数为 Sigmoid 函数，那么

当神经元的输出接近于 1 的时候认为它被激活，而输出接近于 0 的时候认为它被抑制。

设 $\boldsymbol{x}$ 表示原始输入样本，$\boldsymbol{y}$ 表示编码器输出，y_j 表示编码器隐藏神经元 j 的输出，则 y_j 代表了该隐藏神经元的激活度。对于给定的 $\boldsymbol{x}$，$y_j(\boldsymbol{x})$ 表示输入为 $\boldsymbol{x}$ 时编码器隐藏神经元 j 的激活度。进一步，可以定义编码器隐藏神经元 j 的平均激活度 $\hat{\rho}_j$：

$$\hat{\rho}_j = \frac{1}{n}\sum_{i=1}^{n} y_j(\boldsymbol{x}^{(i)}) \tag{3-20}$$

式中，n 表示训练样本数量，$\boldsymbol{x}^{(i)}$ 表示第 i 个训练样本。

引进稀疏性参数 ρ，ρ 通常是一个接近于 0 的较小的正数。期望编码器隐藏神经元 j 的平均激活度 $\hat{\rho}_j = \rho$。为了实现这一约束，在自动编码器的损失函数中增加一个额外的惩罚因子。惩罚因子的具体形式有很多合理的选择，比如可选择

$$\sum_{j=1}^{m} \rho \log_2 \frac{\rho}{\hat{\rho}_j} + (1-\rho)\log_2 \frac{1-\rho}{1-\hat{\rho}_j} \tag{3-21}$$

式中，m 表示编码器隐藏神经元的数量。

式(3-21)也可表示为

$$\sum_{j=1}^{m} \mathrm{KL}(\rho \parallel \hat{\rho}_j) \tag{3-22}$$

式中，$\mathrm{KL}(\rho \parallel \hat{\rho}_j) = \sum_{j=1}^{m} \rho \log_2 \frac{\rho}{\hat{\rho}_j} + (1-\rho)\log_2 \frac{1-\rho}{1-\hat{\rho}_j}$，是一个以 ρ 为均值和一个以 $\hat{\rho}_j$ 为均值的两个伯努利随机变量之间的 KL 散度(也称为相对熵)。KL 散度是一种标准的用来测量两个分布之间差异的方法。当 $\hat{\rho}_j = \rho$ 时，$\mathrm{KL}(\rho \parallel \hat{\rho}_j) = 0$；$\hat{\rho}_j$ 和 ρ 差异越大，$\mathrm{KL}(\rho \parallel \hat{\rho}_j)$ 越大。

因此，稀疏自动编码器的损失函数为

$$\boldsymbol{J}_{\mathrm{SAE}}(\boldsymbol{\theta}, \boldsymbol{\theta}') = \boldsymbol{J}_{\mathrm{AE}}(\boldsymbol{\theta}, \boldsymbol{\theta}') + \beta \sum_{j=1}^{m} \mathrm{KL}(\rho \parallel \hat{\rho}_j) \tag{3-23}$$

式中，$\boldsymbol{J}_{\mathrm{AE}}$ 为自动编码器的损失函数；β 为控制稀疏性惩罚的权重。

除可采用式(3-21)或式(3-22)所示的稀疏性惩罚外，目前一些研究中还经常采用编码器输出的 L_1 范数或 L_2 范数作为稀疏性惩罚。当采用 L_1 范数时，稀疏自动编码器的损失函数为

$$\boldsymbol{J}_{\mathrm{SAE}}(\boldsymbol{\theta}, \boldsymbol{\theta}') = \boldsymbol{J}_{\mathrm{AE}}(\boldsymbol{\theta}, \boldsymbol{\theta}') + \beta \sum_{j=1}^{m} | y_j | \tag{3-24}$$

最优参数$\{\boldsymbol{\theta}, \boldsymbol{\theta}'\}$确定后，稀疏自动编码器中的编码器输出 $\boldsymbol{y}$ 即为提取到的特征。

3.3.4　收缩自动编码器

收缩自动编码器(contractive autoencoder,CAE)是 2011 年提出的一种新的自动编码器[20]。收缩自动编码器认为好的抽象特征应该对输入样本的微小变化具有鲁棒性,也就是当输入样本发生微小变化时,抽象特征不发生变化。为了实现这个想法,收缩自动编码器在自动编码器的损失函数上增加了一个额外的惩罚因子,即编码器激活函数对于输入的雅可比矩阵的 Frobenius 范数的平方,即

$$\boldsymbol{J}_{\mathrm{CAE}}(\theta,\theta')=\boldsymbol{J}_{\mathrm{AE}}(\theta,\theta')+\beta\sum_{j=1}^{m}\left\|\nabla_x y_j\right\|^2 \tag{3-25}$$

收缩自动编码器能够将输入样本点邻域映射到抽象特征空间中样本点处更小的邻域,这就是收缩自动编码器的局部空间收缩效应。

3.4　基于深度学习的状态特征提取

深度学习是一种深度神经网络,能够从原始数据中提取高层次、抽象的特征,适用于无监督学习、有监督学习、半监督学习等场景。在无监督学习方面,最典型的深度学习模型是将自动编码器及其变种堆叠起来构成的各种堆叠自动编码器。这些堆叠自动编码器最后一层编码器的输出即为最终提取的特征。在有监督学习方面,典型的深度学习模型有深度置信网络、卷积神经网络。对于分类问题,这些网络的最后一层为分类器,如采用 Softmax 激活函数实现激活,最后一层的输入即为最终提取的特征。本节首先对深度学习进行简单介绍,然后介绍深度置信网络、堆叠自动编码器和卷积神经网络。

3.4.1　深度学习简介

深度学习始于 2006 年多伦多大学教授 Geoffrey Hinton 和他的学生 Rudslan Salakhutdinov 在 *Science* 杂志上发表的论文 *Reducing the dimensionality of data with networks* 中提出的深度置信网络。深度学习不仅在理论研究上取得了突破性的进展,而且在实际应用方面也取得了重大成就。由于其强大的计算能力和特征提取能力,其被广泛地应用于分类问题、回归问题、数据降维问题、文本建模问题、图像分割问题和信息检索问题等众多领域。下面将从语音识别、图像处理以及自然语言处理 3 个领域对深度学习进行介绍。

1. 语音识别

微软研究院与 Hinton 教授在 2009 年合作开发了基于深度学习的语音识别框架[21]。其核心是一个具有多个隐藏层的深度神经网络,它利用多个隐藏层对语音信号进行特征提取的方式与人脑对语音处理的方式基本相同。2011 年,微软和谷

歌的研究人员率先将深度学习运用到语音识别中，取得了历史性突破——将错误识别率降低 20%～30%。2012 年，微软发布了其深度学习研制成果——全自动同声传译系统。从此，深度学习进入了更多人的视野。

近年来，随着深度学习被大量应用到语音识别领域[22-24]，研究人员开发出许多基于深度学习的新产品，例如 Google 的 Google Now、苹果的 Siri、百度的语音识别系统、微软的 Big Speech 等。

2. 图像处理

1989 年，Yann LeCun 等提出了卷积神经网络(convolutional neural network，CNN)的概念[25]。2006 年，研究者们首先将深度学习应用在 MINIST 手写字符图像分类问题中，取得了重大突破[26]。2012 年，Ciresan 等成功采用 CNN 技术把 MINIST 手写字符图像识别的错误率减小到 0.23%[27]。CNN 起始应用于小尺寸图像时获得了较好的效果，但用于大尺寸图像时效果并不理想。在 2012 年 ImageNet 竞赛上，Hinton 教授团队采用 CNN 对大规模的图库进行图像识别，将图像识别错误率从 26%降低到 15.3%[28]。同年，Google 发布了其深度学习成果——谷歌大脑。谷歌大脑是利用相互连接的上万台电脑来模仿人脑，通过大量样本对其训练之后，就能够自主地识别出图片。微软在 2014 年展示了其深度学习成果——Adam，假如谷歌能自主辨别出一只狗的图片，那么微软就能够辨别出这只狗的种类。微软在图像识别技术方面已经比谷歌更加成熟，更加进步[29]。


Adam

随着研究者们对深度学习在图像处理领域中的应用进行深入研究，深度学习技术在图像识别领域日趋完善。近几年深度学习技术已经成功地应用于行人目标检测、人脸识别等方面。

3. 自然语言处理

近年来，随着深度学习的快速发展，深度学习在自然语言处理领域也受到越来越多的关注，并且取得了一定的成就，但没有其他领域那么成熟。自然语言是人类特有的语言，对自然语言进行处理并不仅仅是单纯的语言学范畴。自然语言处理是指在软件系统以及计算机系统中能够实现直接使用自然语言进行计算机及计算机系统的通信。自然语言处理是一个多学科相交叉的学科，包括人工智能、关于计算机的语言学和人类语言学多个领域。目前基于统计机器学习的方法在自然语言处理领域中使用最为广泛，该方法所使用的特征主要来自基于 one-hot 向量表示的特征组合，这种组合特征的表示方式会使特征空间变得非常大，其优点是利用高维空间使多数任务变为近似线性可分。虽然该方法在一定程度上可以取得较满意的效果，但研究者们却更注重如何去提取有效的特征。早期研究者们采用的方法是将离散特征的分布式表示作为辅助特征引入传统的算法框架，虽然取得了一定的进展，但提升幅度不大。近两年来，随着研究者们对深度学习技术的理解日趋成熟，越来越多的研究者开始从输入到输出全部采用深度学习模型，并且已经在很多任务中取得了较为显著的突破。