第3章

无监督学习

在监督学习中，我们使用带有标注的训练样本来训练模型。这些标注往往需要人工给出。如果没有这些标注，或者没有训练样本，我们是否还能够从数据中学习？此外，是否能够自动生成样本的标注？

在这些情况下，我们可以考虑使用无监督学习方法。所谓的**无监督学习**(unsupervised learning)，是一种从无标注数据中进行学习、发掘数据中隐藏结构的学习范式。

无监督学习中的一个常见任务是**聚类**(clustering)。聚类是指将数据集中相似的样本划分到同一个群组(cluster)中，如同“物以类聚，人以群分”。例如，在图书馆中我们将学科领域相同或相近的图书摆放在同一排书架上，这个过程就可以看作是聚类。在这个例子中，每一本图书是一个样本，每一排书架是一个群组。聚类就是将每一个样本都对应到一个群组中，使得同一群组中的样本都是在某种程度上相似的样本。

那么，如何进行聚类？一个常见的聚类方法为 **k 均值**(k-means)。

3.1 k 均值

k 均值在名称上与 k 近邻相像。二者之间的相同之处是，k 均值方法也使用两个样本的输入特征在欧几里得空间中的距离来度量两个样本之间的相似程度。但不同的是，k 近邻完成的是分类任务，借助于带有类别标注的训练样本，来预测无标注样本的类别；而 k 均值完成的是聚类任务，将一批无标注样本“自动”划分成多个群组。

3.1.1 k 均值聚类

在聚类中，我们把一定数量的样本(m 个样本)划分成若干个群组，使得每个群组中的样本都“比较相似”。那么，究竟划分为多少个群组“比较恰当”？不同的聚类方法给出的答案不同。在 k 均值中，将样本划分成 k 个群组，k 是使用该方法之前需要给出的超参数。

当已知群组数量 k 时，我们可将聚类问题描述为：输入 m 个样本的输入特征 $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(m)}$，输出 k 个样本集合 $\mathcal{G}_1, \mathcal{G}_2, \cdots, \mathcal{G}_k$。其中，$\boldsymbol{x}^{(i)} =$

$(x_1^{(i)},x_2^{(i)},\cdots,x_d^{(i)})^{\mathrm{T}}$，$d$ 为输入特征的维数，$x_1^{(i)}$ 表示第 i 个样本的第 1 个特征，$(\cdot)^{\mathrm{T}}$ 表示矩阵的转置，$i\in\{1,2,\cdots,m\}$，$2\leqslant k\leqslant m-1$。

接下来的问题是，如何用公式描述同一群组中的样本都“比较相似”？其中一类办法是，使用样本之间的欧几里得距离来度量样本之间的相似程度。

> **想一想**　有哪些可用来度量同一群组中样本相似程度的基于欧几里得距离的指标？

我们可以认为，如果两个样本在欧几里得空间中距离比较接近，那么这两个样本“比较相似”。如果我们希望同一群组中的样本都“比较相似”，那么这些样本之间的欧几里得距离都应该较小。如何度量这些样本之间在欧几里得距离上的接近程度？我们既可以用群组中任意两个样本之间距离的最大值来度量该群组样本之间的接近程度，也可以用群组中任意两个样本之间的平均距离来度量。样本 $\boldsymbol{x}^{(i)}$ 到群组 $\mathcal{G}_j$ 中所有样本的平均距离可表示为

$$\frac{1}{|\mathcal{G}_j|}\sum_{l=1}^{m}\|\boldsymbol{x}^{(i)}-\boldsymbol{x}^{(l)}\|_2[\boldsymbol{x}^{(l)}\in\mathcal{G}_j]$$

式中，$\|\cdot\|_2$ 表示 ℓ^2 范数，$\|\boldsymbol{x}^{(i)}-\boldsymbol{x}^{(l)}\|_2=\sqrt{(x_1^{(i)}-x_1^{(l)})^2+(x_2^{(i)}-x_2^{(l)})^2+\cdots+(x_d^{(i)}-x_d^{(l)})^2}$；$|\mathcal{G}_j|$ 表示群组 $\mathcal{G}_j$ 中的样本数量；$i\in\{1,2,\cdots,m\}$，$j\in\{1,2,\cdots,k\}$；这里的方括号仍为艾佛森括号，即

$$[\boldsymbol{x}^{(l)}\in\mathcal{G}_j]=\begin{cases}1, & \text{若 } \boldsymbol{x}^{(l)}\in\mathcal{G}_j\\ 0 & \text{若 } \boldsymbol{x}^{(l)}\notin\mathcal{G}_j\end{cases}$$

如果我们用平均距离来作为度量指标，那么很自然地我们希望所有样本到其群组中所有样本平均距离的平均值最小，以使每个群组中的样本都尽可能“相似”，即

$$\underset{\boldsymbol{x}^{(i)}\in\mathcal{G}_j}{\text{minimize}}\ \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{k}[\boldsymbol{x}^{(i)}\in\mathcal{G}_j]\frac{1}{|\mathcal{G}_j|}\sum_{l=1}^{m}\|\boldsymbol{x}^{(i)}-\boldsymbol{x}^{(l)}\|_2[\boldsymbol{x}^{(l)}\in\mathcal{G}_j]\tag{3-1}$$

我们希望通过调整将样本划分至不同的群组，来最小化式(3-1)中目标函数的值。为此，首先考查

$$\begin{aligned}\frac{1}{|\mathcal{G}_j|}\sum_{l=1}^{m}\|\boldsymbol{x}^{(i)}-\boldsymbol{x}^{(l)}\|_2[\boldsymbol{x}^{(l)}\in\mathcal{G}_j]&=\frac{1}{|\mathcal{G}_j|}\sum_{l=1}^{m}\|(\boldsymbol{x}^{(i)}-\boldsymbol{x}^{(l)})[\boldsymbol{x}^{(l)}\in\mathcal{G}_j]\|_2\\&\geqslant\left\|\frac{1}{|\mathcal{G}_j|}\sum_{l=1}^{m}(\boldsymbol{x}^{(i)}-\boldsymbol{x}^{(l)})[\boldsymbol{x}^{(l)}\in\mathcal{G}_j]\right\|_2\\&=\left\|\frac{1}{|\mathcal{G}_j|}|\mathcal{G}_j|\boldsymbol{x}^{(i)}-\frac{1}{|\mathcal{G}_j|}\sum_{l=1}^{m}\boldsymbol{x}^{(l)}[\boldsymbol{x}^{(l)}\in\mathcal{G}_j]\right\|_2\\&=\left\|\boldsymbol{x}^{(i)}-\frac{1}{|\mathcal{G}_j|}\sum_{l=1}^{m}\boldsymbol{x}^{(l)}[\boldsymbol{x}^{(l)}\in\mathcal{G}_j]\right\|_2=\|\boldsymbol{x}^{(i)}-\boldsymbol{c}_j\|_2\end{aligned}\tag{3-2}$$

式(3-2)中，$\boldsymbol{c}_j$ 为

$$\boldsymbol{c}_j=\frac{1}{|\mathcal{G}_j|}\sum_{l=1}^{m}\boldsymbol{x}^{(l)}[\boldsymbol{x}^{(l)}\in\mathcal{G}_j]=\frac{1}{|\mathcal{G}_j|}\sum_{\boldsymbol{x}\in\mathcal{G}_j}\boldsymbol{x}\tag{3-3}$$

即 $\boldsymbol{c}_j$ 为第 j 个群组 $\mathcal{G}_j$ 中所有样本输入特征的算术平均值，因此 $\boldsymbol{c}_j$ 是群组 $\mathcal{G}_j$ 的**形心**(centroid)，$j\in\{1,2,\cdots,k\}$。根据式(3-2)可将式(3-1)写为

$$\begin{aligned}\underset{\boldsymbol{x}^{(i)}\in\mathcal{G}_j}{\text{minimize}}\ \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{k}[\boldsymbol{x}^{(i)}\in\mathcal{G}_j]\frac{1}{|\mathcal{G}_j|}\sum_{l=1}^{m}\|\boldsymbol{x}^{(i)}-\boldsymbol{x}^{(l)}\|_2[\boldsymbol{x}^{(l)}\in\mathcal{G}_j]\\ =\underset{\boldsymbol{x}^{(i)}\in\mathcal{G}_j}{\text{minimize}}\ \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{k}[\boldsymbol{x}^{(i)}\in\mathcal{G}_j]\|\boldsymbol{x}^{(i)}-\boldsymbol{c}_j\|_2\end{aligned}\tag{3-4}$$

式(3-4)中的目标函数 $\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{k}[\boldsymbol{x}^{(i)}\in\mathcal{G}_j]\|\boldsymbol{x}^{(i)}-\boldsymbol{c}_j\|_2$ 表示 m 个欧几里得距离(即 ℓ^2 范数)的平均值。如果 $\boldsymbol{c}_1,\boldsymbol{c}_2,\cdots,\boldsymbol{c}_k$ 保持不变，那么最小化其中的每一个距离，都有助于最小化该目标函数。当 $\boldsymbol{c}_1,\boldsymbol{c}_2,\cdots,\boldsymbol{c}_k$ 保持不变时，由式(3-4)可得

$$\begin{aligned}\min_{\boldsymbol{x}^{(i)}\in\mathcal{G}_j}\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{k}[\boldsymbol{x}^{(i)}\in\mathcal{G}_j]\|\boldsymbol{x}^{(i)}-\boldsymbol{c}_j\|_2&=\frac{1}{m}\sum_{i=1}^{m}\min_{\boldsymbol{x}^{(i)}\in\mathcal{G}_j}\sum_{j=1}^{k}[\boldsymbol{x}^{(i)}\in\mathcal{G}_j]\|\boldsymbol{x}^{(i)}-\boldsymbol{c}_j\|_2\\ &=\frac{1}{m}\sum_{i=1}^{m}\min_{j\in\{1,2,\cdots,k\}}\|\boldsymbol{x}^{(i)}-\boldsymbol{c}_j\|_2\end{aligned}\tag{3-5}$$

也就是说，此时对所有的 $i\in\{1,2,\cdots,m\}$ 都最小化式(3-6)中目标函数的值，就可以使式(3-4)中目标函数的值最小。

$$\min_{j\in\{1,2,\cdots,k\}}\|\boldsymbol{x}^{(i)}-\boldsymbol{c}_j\|_2\tag{3-6}$$

亦即，将每一个样本 $\boldsymbol{x}^{(i)}$ 都划分至形心距其最近的群组，$i=1,2,\cdots,m$，就可以使式(3-4)中目标函数的值最小。

$$j^*=\underset{j\in\{1,2,\cdots,k\}}{\operatorname{argmin}}\|\boldsymbol{x}^{(i)}-\boldsymbol{c}_j\|_2\tag{3-7}$$

式(3-7)中，j^* 为形心距离样本 $\boldsymbol{x}^{(i)}$ 最近的群组的索引。注意到如果将式(3-7)中的 ℓ^2 范数 $\|\cdot\|_2$ 替换成 ℓ^2 范数的平方 $\|\cdot\|_2^2$，并不会影响计算结果，而且还可以省掉根号运算，有助于减少运算量。因此，我们用 ℓ^2 范数的平方来替代 ℓ^2 范数，即

$$j^*=\underset{j\in\{1,2,\cdots,k\}}{\operatorname{argmin}}\|\boldsymbol{x}^{(i)}-\boldsymbol{c}_j\|_2^2\tag{3-8}$$

然而，当按照式(3-8)把所有样本都重新划分至新的群组之后，这些群组的形心可能会发生变化，因为形心是群组中所有样本输入特征的算术平均值，群组中的样本发生了变化，形心也可能发生变化。很自然地，我们会想到重新计算各个群组的形心。那么，如果我们用新计算出来的形心覆盖掉之前已有的形心，会起到什么作用？

此时，我们是在保持样本与群组之间对应关系不变的情况下，来调整式(3-4)中的 $\boldsymbol{c}_1,\boldsymbol{c}_2,\cdots,\boldsymbol{c}_k$，以使式(3-4)中目标函数的值最小，即

$$\underset{\boldsymbol{c}_1,\boldsymbol{c}_2,\cdots,\boldsymbol{c}_k}{\text{minimize}}\ \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{k}[\boldsymbol{x}^{(i)}\in\mathcal{G}_j]\|\boldsymbol{x}^{(i)}-\boldsymbol{c}_j\|_2\tag{3-9}$$

在这种情况下，式(3-9)中的目标函数是凸函数。因此，由式(3-9)给出的最优化问题在目标函数梯度为零处取得全局最优解。为了便于求解该最优化问题，人们将式(3-9)中的 ℓ^2 范数 $\|\cdot\|_2$ 同样替换成 ℓ^2 范数的平方 $\|\cdot\|_2^2$，尽管这样做求得的解与式(3-9)的解很多情况下并不相同，即

$$\underset{\boldsymbol{c}_1,\boldsymbol{c}_2,\cdots,\boldsymbol{c}_k}{\text{minimize}}\ \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{k}[\boldsymbol{x}^{(i)}\in\mathcal{G}_j]\|\boldsymbol{x}^{(i)}-\boldsymbol{c}_j\|_2^2\tag{3-10}$$

若令 $J(\boldsymbol{c}_1,\boldsymbol{c}_2,\cdots,\boldsymbol{c}_k)=\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{k}[\boldsymbol{x}^{(i)}\in\mathcal{G}_j]\parallel\boldsymbol{x}^{(i)}-\boldsymbol{c}_j\parallel_2^2$，则有

$$\frac{\partial J(\boldsymbol{c}_1,\boldsymbol{c}_2,\cdots,\boldsymbol{c}_k)}{\partial \boldsymbol{c}_j}=\frac{2}{m}\sum_{i=1}^{m}(\boldsymbol{c}_j-\boldsymbol{x}^{(i)})[\boldsymbol{x}^{(i)}\in\mathcal{G}_j]=\boldsymbol{0} \tag{3-11}$$

式(3-11)中，$j=1,2,\cdots,k$。由式(3-11)可解出

$$\boldsymbol{c}_j=\frac{1}{|\mathcal{G}_j|}\sum_{i=1}^{m}\boldsymbol{x}^{(i)}[\boldsymbol{x}^{(i)}\in\mathcal{G}_j] \tag{3-12}$$

式(3-12)中，$j=1,2,\cdots,k$。由此可见，当 $\boldsymbol{c}_1,\boldsymbol{c}_2,\cdots,\boldsymbol{c}_k$ 为各群组的形心时，式(3-10)中目标函数的值最小。所以，上述问题的答案是，如果我们用新计算出来的形心覆盖掉之前已有的形心，实际上是在最小化式(3-10)中目标函数的值。因此，在某种意义上这样做也有助于最小化式(3-4)中目标函数的值。

同样，在重新计算出各个群组的形心之后，又需要根据新的形心来将各个样本划分至新的群组。如此循环下去，直到式(3-4)中目标函数的值不再显著减小，即收敛。

那么，为什么式(3-4)中目标函数的值会收敛？如前所述，上述两步迭代中的每一步都试图在现有条件下最小化式(3-4)中的目标函数。因此从直觉上看，式(3-4)中目标函数的值大体上将会随着循环次数的增加而减小，直到不再显著减小。当样本与群组之间的对应关系不再发生改变时，群组的形心不再发生变化，式(3-4)中目标函数的值也不再发生变化。所以，当样本与群组之间的对应关系不再变化时，我们可认为式(3-4)中目标函数的值收敛。

接下来的问题是，在收敛之后，我们会得到全局最优解吗(使目标函数的值全局最小的样本与群组之间的对应关系)？由于式(3-4)给出的最优化问题，并不是凸优化问题，因此不能保证收敛后得到全局最优解。所以，k 均值的聚类结果仍会受到初始值的影响。

综上所述，k 均值聚类方法如下。

【k 均值聚类方法】

输入：m 个样本的输入特征 $\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)},\cdots,\boldsymbol{x}^{(m)}$；群组的数量 k。

输出：m 个样本与 k 个群组的对应关系(也可为 k 个群组的形心 $\boldsymbol{c}_1,\boldsymbol{c}_2,\cdots,\boldsymbol{c}_k$)。

(1) 初始化 k 个群组的形心 $\boldsymbol{c}_1,\boldsymbol{c}_2,\cdots,\boldsymbol{c}_k$。尽管我们可以用随机值来初始化，但是由于初始值对聚类结果有影响，常见的做法是：从 m 个样本中随机取 k 个样本，将这 k 个样本的输入特征分别作为 $\boldsymbol{c}_1,\boldsymbol{c}_2,\cdots,\boldsymbol{c}_k$ 的初始值。

(2) 重复如下两步迭代，直到满足停止条件。停止条件可以是：各个样本与群组之间的对应关系不再发生改变。

① 按照式(3-8)将 m 个样本分别划分至 k 个群组之一。

② 按照式(3-3)更新 k 个群组的形心 $\boldsymbol{c}_1,\boldsymbol{c}_2,\cdots,\boldsymbol{c}_k$。

当然，在上述迭代的第一步中，也可以只将一个样本划分至 k 个群组之一，这样在第二步中只需更新发生了变化的两个群组的形心。这种做法通常可以更早收敛，并且更便于实现。

3.1.2　k 值与轮廓系数

k 均值方法最初被用于向量量化(vector quantization)。向量量化是指用码本中的 k

个既定码字(向量)之一来替换一个输入向量。码本中的 k 个码字就是通过 k 均值聚类得到的 k 个群组的形心。因此,向量量化就是将输入向量替换为距离其最近的形心向量的过程。在向量量化中,k(码本中码字的数量)是训练码本时人工给出的值,通常为 2 的整数次幂,例如 64、256 等。

但是在一些其他应用里,我们事先并不清楚聚为多少个类“比较适合”,即不知道 k 的取值。如何找到“适合”的 k 值?

由于 k 是 k 均值的超参数,我们可以通过多次尝试来确定一个“比较适合”的 k 值。那么,什么是“比较适合”?如何度量?这涉及另外一个问题:如何评价聚类的结果?如果聚类的结果较好,那么就可以认为当前 k 的取值“比较适合”。

在无监督学习中,我们使用的数据集通常不带有标注数据,无法像在监督学习中那样通过比较预测值与标注之间的差距来评估模型的性能。因此,很多时候我们只好从对聚类的预期入手,通过度量聚类结果与我们预期结果之间的差距,来评估聚类结果。例如,我们希望聚类结果中同群组的样本之间距离较近,而不同群组的样本之间距离较远。一个相关的度量指标是**轮廓系数**(silhouette coefficient),其定义为

$$s_j^{(i)}=\frac{b_j^{(i)}-a_j^{(i)}}{\max(a_j^{(i)},b_j^{(i)})} \tag{3-13}$$

式(3-13)中,$s_j^{(i)}$ 为第 i 个样本(在第 j 个群组中)的轮廓系数;$\max(a_j^{(i)},b_j^{(i)})$ 表示取 $a_j^{(i)}$ 和 $b_j^{(i)}$ 两个数中的较大者;$a_j^{(i)}$ 为第 i 个样本到其所在群组 $\mathcal{G}_j$ 中所有其他样本的距离的平均值,$a_j^{(i)}\geqslant 0$;$b_j^{(i)}$ 为第 i 个样本到距其最近的另一个群组中所有样本的距离的平均值,$b_j^{(i)}\geqslant 0$,即

$$a_j^{(i)}=\frac{1}{|\mathcal{G}_j|-1}\sum_{l=1}^{m}\|\boldsymbol{x}^{(i)}-\boldsymbol{x}^{(l)}\|_2[\boldsymbol{x}^{(l)}\in\mathcal{G}_j] \tag{3-14}$$

$$b_j^{(i)}=\min_{h=1,2,\cdots,k;h\neq j}\left(\frac{1}{|\mathcal{G}_h|}\sum_{l=1}^{m}\|\boldsymbol{x}^{(i)}-\boldsymbol{x}^{(l)}\|_2[\boldsymbol{x}^{(l)}\in\mathcal{G}_h]\right) \tag{3-15}$$

可见,轮廓系数 $s_j^{(i)}$ 的取值范围为$[-1,1]$。当 $s_j^{(i)}>0$ 时,表明当前样本与其所在群组中的其他样本更为“相似”,聚类方法将其划分至了“正确”的群组;当 $s_j^{(i)}<0$ 时,表明当前样本与另一个群组中的样本更为“相似”,更应该被划分至另一个群组,聚类方法将其划分至了“错误”的群组。

为了用一个标量数值来作为聚类结果的评价指标,我们对 m 个样本的轮廓系数做算术平均,将该平均值记为 $\bar{s}$,则有

$$\bar{s}=\frac{1}{m}\sum_{l=1}^{m}\sum_{j=1}^{k}s_j^{(i)}[\boldsymbol{x}^{(i)}\in\mathcal{G}_j] \tag{3-16}$$

综上,我们可使用平均轮廓系数 $\bar{s}$ 来评价 k 均值聚类结果。当 k 值未知时,可以尝试使用不同的 k 值进行聚类并分别计算平均轮廓系数,然后选取最大的平均轮廓系数对应的 k 值,作为“适合”的 k 值。

3.1.3 k 均值实践

在本节中,我们编程实现 k 均值聚类方法,并使用第 2 章中多分类任务的轮椅数据集

(不包括其中的标注)以及平均轮廓系数,来评价 k 均值聚类。

实验 3-1　实现 k 均值聚类,并观察其在轮椅数据集上的聚类结果。更改随机种子后再次观察聚类结果。

提示:①可对输入特征做特征缩放;②由于轮椅数据集中包含 4 个类别的样本,所以 k 均值聚类中的 k 值可取 4;③既可使用多重循环的方式,又可使用矩阵形式实现 k 均值聚类;④在多个相邻整数中以相同的概率随机抽取一个或多个可使用 rng.integers()方法;⑤可能会用到 np.copy()、np.expand_dims()、np.array_equal()等函数;⑥如果对编写实验程序没有任何思路、无从下手,可参考 3.4 节。

如果对照轮椅数据集中的标注,可以看出 k 均值聚类为各个样本分配的群组的序号,比较接近于样本的类别标注。如果更改随机种子的值,聚类的结果将有所不同。

实验 3-2　计算实验 3-1 中 k 均值聚类的平均轮廓系数,并画出不同 k 值下的平均轮廓系数。

提示:计算向量之间的欧几里得距离(范数)可使用 np.linalg.norm()函数。

图 3-1 示出了当 $2 \leqslant k \leqslant 10$、随机种子为 11 时的平均轮廓系数。从图 3-1 中可以看出,平均轮廓系数为正数,表明大体上 k 均值将样本划分至了“正确”的群组。更改随机种子的值,观察平均轮廓系数的变化。可以看出,平均轮廓系数受随机种子的影响较大,这表明形心的初始值对聚类结果有较大影响。

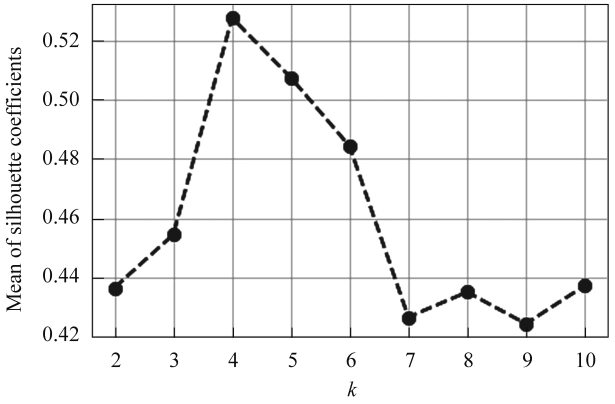


图 3-1　不同 k 值下的平均轮廓系数

3.2　主成分分析

除了聚类,无监督学习中的另一个常见任务是**降维**(dimensionality reduction)。顾名思义,降维是指把较高维的数据转换为较低维的数据,去除原数据中的冗余与噪声,在减

少数据量的同时尽量保留原数据的主要特性。例如,当我们用摆放在同一间教室内 4 个角落上的 4 个温度传感器的读数来预测教室内讲台处的室温时,预测模型的输入为由同一时刻 4 个温度传感器的读数组成的四维向量,显然这种情况下模型的输入数据中存在冗余,可使用降维方法降低输入数据的维数,例如从四维降至一维。

降维方法可分为线性方法及非线性方法。一种常见的线性降维方法是**主成分分析**(principal component analysis,PCA)。主成分分析是指计算数据中的"主要成分",也是一种特征提取(feature extraction)方法。

3.2.1 主成分分析降维

从直觉上看,如果把较高维的数据转换为较低维的数据,可能会损失数据中包含的一部分信息。因此,我们希望在转换过程中损失的信息越少越好,以便最大程度地保留数据中包含的信息。

我们知道,熵是用来描述随机变量取值不确定程度的度量指标。我们可以把数据集中样本的每维输入特征都看作一个随机变量。对于一个具有 m 个样本、d 维输入特征的数据集来说,共有 d 个随机变量。每个随机变量的熵越大,代表每维输入特征所包含的信息量越大。特别地,对于零均值正态分布的连续随机变量来说,其微分熵(differential entropy)是其方差的单调递增函数,即方差越小,微分熵越小。如果每维输入特征都服从均值为零的正态分布,并且各维输入特征之间都不相关,那么我们去掉方差最小的随机变量所对应的这维输入特征,就可以使降维过程中减少的信息量最小。去掉数据集中样本的部分输入特征的过程就是一种降维。

至此,我们已经得到了一种降维方法。但是,在上述降维过程中,我们不仅假设各维输入特征都服从零均值正态分布,而且假设各维输入特征之间都不相关。在机器学习中,我们经常假设各维输入特征都服从正态分布,这是因为根据中心极限定理(central limit theorem,CLT),许多个独立同分布随机变量之和趋近于正态分布,因此,自然界中的很多随机变量都服从正态分布,例如同质人群的身高。对于均值不为零的正态随机变量,我们只需要减去其样本平均值即可得到均值近似为零的正态随机变量,因为样本平均值是随机变量均值的无偏估计量。然而,在机器学习中,数据集中样本的各维输入特征之间都不相关是一种较为理想的假设,很多时候各维输入特征之间或多或少存在相关性。例如,在气温数据集中,在某大都市 A 点处传感器采集的气温数据,与同一时刻在该大都市 B 点处传感器采集的气温数据之间存在相关性。那么,在这种情况下该如何降维?

如果我们能从数据集中提取出一组新的各维不相关的样本输入特征,就可以继续使用上述降维方法来进行降维。我们知道,如果两个随机变量 X_1、X_2 的皮尔逊相关系数 $\rho_{X_1,X_2}=\dfrac{\mathbb{E}((X_1-\bar{X}_1)(X_2-\bar{X}_2))}{\sigma_{X_1}\sigma_{X_2}}=0$,即 $\mathbb{E}((X_1-\bar{X}_1)(X_2-\bar{X}_2))=\mathbb{E}(X_1X_2)-\mathbb{E}(X_1)\mathbb{E}(X_2)=0$,那么这两个随机变量不相关,即二者之间不存在线性关系。其中 $\mathbb{E}(\cdot)$ 代表取期望值,$\bar{X}_1$、$\bar{X}_2$ 分别是随机变量 X_1、X_2 的均值,$\mathbb{E}((X_1-\bar{X}_1)(X_2-\bar{X}_2))$ 是随机变量 X_1、X_2 的协方差(covariance)。更一般地,对于 d 个随机变量 $X_1,X_2,\cdots,X_d$

来说，如果在其 $d\times d$ 大小的协方差矩阵中，除主对角线上的元素之外，其余元素都为 0，即式(3-17)成立，那么这 d 个随机变量之间都不相关。

$$\boldsymbol{C}=\begin{pmatrix}\mathbb{E}((X_1-\bar{X}_1)(X_1-\bar{X}_1)) & \mathbb{E}((X_1-\bar{X}_1)(X_2-\bar{X}_2)) & \cdots & \mathbb{E}((X_1-\bar{X}_1)(X_d-\bar{X}_d))\\ \mathbb{E}((X_2-\bar{X}_2)(X_1-\bar{X}_1)) & \mathbb{E}((X_2-\bar{X}_2)(X_2-\bar{X}_2)) & \cdots & \mathbb{E}((X_2-\bar{X}_2)(X_d-\bar{X}_d))\\ \vdots & \vdots & \ddots & \vdots\\ \mathbb{E}((X_d-\bar{X}_d)(X_1-\bar{X}_1)) & \mathbb{E}((X_d-\bar{X}_d)(X_2-\bar{X}_2)) & \cdots & \mathbb{E}((X_d-\bar{X}_d)(X_d-\bar{X}_d))\end{pmatrix}$$

$$=\begin{pmatrix}\mathbb{E}((X_1-\bar{X}_1)^2) & 0 & \cdots & 0\\ 0 & \mathbb{E}((X_2-\bar{X}_2)^2) & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \mathbb{E}((X_d-\bar{X}_d)^2)\end{pmatrix} \tag{3-17}$$

式(3-17)中，$\bar{X}_1,\bar{X}_2,\cdots,\bar{X}_d$ 分别是随机变量 $X_1,X_2,\cdots,X_d$ 的均值。因此，如果我们能从数据集中提取出一组新的输入特征，满足式(3-17)，那么这些输入特征之间就都不相关。如何提取这组新的输入特征？

先考虑如何计算协方差矩阵。如果随机变量 $X_1,X_2,\cdots,X_d$ 的均值都为 0，则式(3-17)成为

$$\boldsymbol{C}=\begin{pmatrix}\mathbb{E}(X_1X_1) & \mathbb{E}(X_1X_2) & \cdots & \mathbb{E}(X_1X_d)\\ \mathbb{E}(X_2X_1) & \mathbb{E}(X_2X_2) & \cdots & \mathbb{E}(X_2X_d)\\ \vdots & \vdots & \ddots & \vdots\\ \mathbb{E}(X_dX_1) & \mathbb{E}(X_dX_2) & \cdots & \mathbb{E}(X_dX_d)\end{pmatrix}=\begin{pmatrix}\mathbb{E}(X_1X_1) & 0 & \cdots & 0\\ 0 & \mathbb{E}(X_2X_2) & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \mathbb{E}(X_dX_d)\end{pmatrix} \tag{3-18}$$

若数据集中有 m 个样本，样本输入特征的维数为 d，且每维输入特征的均值都是 0，那么可通过下式计算协方差矩阵：

$$\boldsymbol{C}=\frac{1}{m-1}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}} \tag{3-19}$$

式(3-19)中，$\boldsymbol{C}\in\mathbb{R}^{d\times d}$；$\boldsymbol{X}$ 为数据集中的样本组成的矩阵，$\boldsymbol{X}\in\mathbb{R}^{d\times m}$，$\boldsymbol{X}$ 由式(2-22)给出。式(3-19)中的除数是 $m-1$ 而不是 m，是因为方差的无偏估计量为 $\frac{1}{m-1}\sum_{i=1}^{m}(x^{(i)}-\bar{x})^2$，其中，$\bar{x}=\frac{1}{m}\sum_{i=1}^{m}x^{(i)}$。由式(3-18)可以看出 $\boldsymbol{C}=\boldsymbol{C}^{\mathrm{T}}$，即协方差矩阵 $\boldsymbol{C}$ 是对称矩阵。对称矩阵可以被正交矩阵(orthogonal matrix)对角化，即存在一个正交矩阵 $\boldsymbol{Q}$，使得

$$\boldsymbol{Q}^{-1}\boldsymbol{C}\boldsymbol{Q}=\boldsymbol{\Lambda}$$

其中，$\boldsymbol{Q}^{-1}$ 为矩阵 $\boldsymbol{Q}$ 的逆矩阵；$\boldsymbol{\Lambda}$ 为对角矩阵，即非主对角线上的元素都是 0。把式(3-19)代入上式可得

$$\frac{1}{m-1}\boldsymbol{Q}^{-1}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Q}=\boldsymbol{\Lambda}$$

又因为 $\boldsymbol{Q}$ 为正交矩阵,$\boldsymbol{Q}^{-1}=\boldsymbol{Q}^{\mathrm{T}}$,因此有

$$\frac{1}{m-1}(\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{X})(\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{X})^{\mathrm{T}}=\boldsymbol{\Lambda} \tag{3-20}$$

若令

$$\widetilde{\boldsymbol{X}}=\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{X} \tag{3-21}$$

式(3-21)中,$\widetilde{\boldsymbol{X}}\in\mathbb{R}^{d\times m}$,那么由式(3-19)和式(3-20)可得

$$\widetilde{\boldsymbol{C}}=\frac{1}{m-1}\widetilde{\boldsymbol{X}}\,\widetilde{\boldsymbol{X}}^{\mathrm{T}}=\frac{1}{m-1}(\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{X})(\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{X})^{\mathrm{T}}=\boldsymbol{\Lambda}$$

即 $\widetilde{\boldsymbol{X}}$ 的协方差矩阵 $\widetilde{\boldsymbol{C}}$ 是对角矩阵。也就是说,$\widetilde{\boldsymbol{X}}$ 中的各维输入特征之间都不相关。因此,$\widetilde{\boldsymbol{X}}$ 就是我们想要求得的具有不相关输入特征的 m 个样本的输入特征矩阵。在求得 $\widetilde{\boldsymbol{X}}$ 后,就可以按照如前所述的降维方法对 $\widetilde{\boldsymbol{X}}$ 进行降维。这种降维方法就是主成分分析降维。由于 $\boldsymbol{Q}$ 是正交矩阵,式(3-21)可以理解为 $\boldsymbol{X}$ 经过旋转(或反射、旋转反射)得到 $\widetilde{\boldsymbol{X}}$。

综上所述,主成分分析降维方法如下。

【主成分分析降维方法】

输入:m 个样本的 d 维输入特征 $\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)},\cdots,\boldsymbol{x}^{(m)}$。

输出:m 个样本的 $\widetilde{d}$ 维新输入特征 $\widetilde{\boldsymbol{x}}^{(1)},\widetilde{\boldsymbol{x}}^{(2)},\cdots,\widetilde{\boldsymbol{x}}^{(m)},\widetilde{d}\leqslant d$。

(1) 零均值化。将输入的各个样本的输入特征 $\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)},\cdots,\boldsymbol{x}^{(m)}$ 减去其样本平均值,即

$$\boldsymbol{x}^{(i)}:=\boldsymbol{x}^{(i)}-\frac{1}{m}\sum_{j=1}^{m}\boldsymbol{x}^{(j)} \tag{3-22}$$

式(3-22)中,“:=”表示赋值;$i=1,2,\cdots,m$。

(2) 按照式(3-19)计算协方差矩阵 $\boldsymbol{C}$。

(3) 通过特征分解(eigendecomposition)求协方差矩阵 $\boldsymbol{C}$ 的 d 个特征值及对应的特征向量,并将它们按照特征值从大到小的顺序进行排列:λ_1 为最大的特征值,$\boldsymbol{q}_1$ 为 λ_1 对应的特征向量;λ_d 为最小的特征值,$\boldsymbol{q}_d$ 为 λ_d 对应的特征向量。$\boldsymbol{q}_1,\boldsymbol{q}_2,\cdots,\boldsymbol{q}_d$ 就是矩阵 $\boldsymbol{Q}$ 的 d 个列向量,$\lambda_1,\lambda_2,\cdots,\lambda_d$ 就是对角矩阵 $\boldsymbol{\Lambda}$ 主对角线上的 d 个元素。

(4) 取前 $\widetilde{d}$ 个特征值对应的特征向量 $\boldsymbol{q}_1,\boldsymbol{q}_2,\cdots,\boldsymbol{q}_{\widetilde{d}}$ 作为列向量,组成矩阵 $\widetilde{\boldsymbol{Q}},\widetilde{\boldsymbol{Q}}\in\mathbb{R}^{d\times\widetilde{d}}$。并参照式(3-21)计算矩阵 $\widetilde{\widetilde{\boldsymbol{X}}},\widetilde{\widetilde{\boldsymbol{X}}}=\widetilde{\boldsymbol{Q}}^{\mathrm{T}}\boldsymbol{X},\widetilde{\widetilde{\boldsymbol{X}}}\in\mathbb{R}^{\widetilde{d}\times m}$。$\widetilde{\widetilde{\boldsymbol{X}}}=(\widetilde{\widetilde{\boldsymbol{x}}}^{(1)},\widetilde{\widetilde{\boldsymbol{x}}}^{(2)},\cdots,\widetilde{\widetilde{\boldsymbol{x}}}^{(m)})$ 为待输出的 m 个 $\widetilde{d}$ 维样本输入特征 $\widetilde{\widetilde{\boldsymbol{x}}}^{(1)},\widetilde{\widetilde{\boldsymbol{x}}}^{(2)},\cdots,\widetilde{\widetilde{\boldsymbol{x}}}^{(m)}$ 组成的矩阵。

至此,我们得到降维后的 m 个样本的新输入特征 $\widetilde{\widetilde{\boldsymbol{x}}}^{(1)},\widetilde{\widetilde{\boldsymbol{x}}}^{(2)},\cdots,\widetilde{\widetilde{\boldsymbol{x}}}^{(m)}$,从而完成降维任务。需要说明的是,降维后的各维新输入特征的均值仍是 0。

那么,如何确定 $\widetilde{d}$ 的值?至少有以下 3 种办法:① 人工指定 $\widetilde{d}$ 的值,选择前 $\widetilde{d}$ 个较大的特征值,或根据系统需求确定 $\widetilde{d}$ 的值;② 选择使前 $\widetilde{d}$ 个特征值之和足够大的 $\widetilde{d}$ 值,即满足 $\dfrac{\sum_{j=1}^{\widetilde{d}}\lambda_j}{\sum_{j=1}^{d}\lambda_j}\geqslant\alpha$ 的最小 $\widetilde{d}$ 值,α 为预设门限,$0<\alpha<1$,例如取 $\alpha=0.9$;③ 结合后续的回归、

分类、聚类等模型，选择使后续模型性能较好的 $\tilde{d}$ 值。

上述降维过程用到了协方差矩阵的特征分解。是否还有其他办法可以从数据集中提取出新的各维不相关的输入特征，以满足式(3-18)？

考虑到特征分解仅适用于包括实对称矩阵在内的正规方阵，而奇异值分解(singular value decomposition，SVD)可用于任何复矩阵，比特征分解更具有普遍性。因此，我们尝试对样本的输入特征矩阵 $\boldsymbol{X}$ 做奇异值分解。通过奇异值分解，$\boldsymbol{X}$ 可被分解为 3 个矩阵之积，即

$$\boldsymbol{X}=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}$$

其中，$\boldsymbol{U}$、$\boldsymbol{V}$ 为正交矩阵，即 $\boldsymbol{U}^{\mathrm{T}}=\boldsymbol{U}^{-1}$，$\boldsymbol{V}^{\mathrm{T}}=\boldsymbol{V}^{-1}$，$\boldsymbol{U}\in\mathbb{R}^{d\times d}$，$\boldsymbol{V}\in\mathbb{R}^{m\times m}$；$\boldsymbol{\Sigma}$ 为(矩形)对角矩阵，$\boldsymbol{\Sigma}\in\mathbb{R}^{d\times m}$。若令

$$\tilde{\boldsymbol{X}}=\boldsymbol{U}^{\mathrm{T}}\boldsymbol{X} \tag{3-23}$$

可知 $\tilde{\boldsymbol{X}}\in\mathbb{R}^{d\times m}$，则由式(3-19)可得

$$\begin{aligned}\tilde{\boldsymbol{C}}&=\frac{1}{m-1}\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^{\mathrm{T}}=\frac{1}{m-1}(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{X})(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{X})^{\mathrm{T}}=\frac{1}{m-1}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{U}=\frac{1}{m-1}\boldsymbol{U}^{\mathrm{T}}(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}})(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}})^{\mathrm{T}}\boldsymbol{U}\\&=\frac{1}{m-1}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{\Sigma}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}=\frac{1}{m-1}\boldsymbol{U}^{-1}\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{-1}\boldsymbol{V}\boldsymbol{\Sigma}^{\mathrm{T}}\boldsymbol{U}^{-1}\boldsymbol{U}=\frac{1}{m-1}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{\mathrm{T}}\end{aligned} \tag{3-24}$$

由于 $\boldsymbol{\Sigma}$ 为 $d\times m$ 对角矩阵，式(3-24)中的 $\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{\mathrm{T}}$ 为 $d\times d$ 对角矩阵。也就是说，$\tilde{\boldsymbol{X}}$ 的协方差矩阵 $\tilde{\boldsymbol{C}}$ 为对角矩阵，因此 $\tilde{\boldsymbol{X}}$ 中的各维输入特征之间都不相关。$\tilde{\boldsymbol{X}}$ 就是我们想要求得的具有不相关输入特征的 m 个样本的输入特征矩阵。

可见，通过奇异值分解求 $\tilde{\boldsymbol{X}}$，并不需要计算协方差矩阵 $\boldsymbol{C}$。此外，在降维任务中，通常不必求得 $\boldsymbol{U}$ 的所有列向量，因此可使用简化版的奇异值分解，例如紧 SVD(compact SVD)、截尾 SVD(truncated SVD)等。所以人们常使用奇异值分解来进行主成分分析降维。

综上所述，使用奇异值分解的主成分分析降维方法如下。

【使用奇异值分解的主成分分析降维方法】

输入：m 个样本的 d 维输入特征 $\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)},\cdots,\boldsymbol{x}^{(m)}$。

输出：m 个样本的 $\tilde{d}$ 维新输入特征 $\tilde{\tilde{\boldsymbol{x}}}^{(1)},\tilde{\tilde{\boldsymbol{x}}}^{(2)},\cdots,\tilde{\tilde{\boldsymbol{x}}}^{(m)}$，$\tilde{d}\leqslant d$。

(1) 零均值化。将输入的各个样本的输入特征 $\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)},\cdots,\boldsymbol{x}^{(m)}$ 减去其样本平均值，见式(3-22)。

(2) 使用奇异值分解求出 $\tilde{d}$ 个最大的奇异值(即矩阵 $\boldsymbol{\Sigma}$ 主对角线上的元素)σ_1，$\sigma_2,\cdots,\sigma_{\tilde{d}}$ 所对应的 $\tilde{d}$ 个左奇异向量(即矩阵 $\boldsymbol{U}$ 的列向量)$\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_{\tilde{d}}$，组成矩阵 $\tilde{\boldsymbol{U}}$，$\tilde{\boldsymbol{U}}=(\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_{\tilde{d}})$，$\tilde{\boldsymbol{U}}\in\mathbb{R}^{d\times\tilde{d}}$。

(3) 参照式(3-23)计算矩阵 $\tilde{\tilde{\boldsymbol{X}}}$，$\tilde{\tilde{\boldsymbol{X}}}=\tilde{\boldsymbol{U}}^{\mathrm{T}}\boldsymbol{X}$，$\tilde{\tilde{\boldsymbol{X}}}\in\mathbb{R}^{\tilde{d}\times m}$。$\tilde{\tilde{\boldsymbol{X}}}=(\tilde{\tilde{\boldsymbol{x}}}^{(1)},\tilde{\tilde{\boldsymbol{x}}}^{(2)},\cdots,\tilde{\tilde{\boldsymbol{x}}}^{(m)})$ 即为待输出的 m 个样本的 $\tilde{d}$ 维输入特征 $\tilde{\tilde{\boldsymbol{x}}}^{(1)},\tilde{\tilde{\boldsymbol{x}}}^{(2)},\cdots,\tilde{\tilde{\boldsymbol{x}}}^{(m)}$ 组成的矩阵。

3.2.2 主成分分析实践

在本节中,我们分别使用特征分解和奇异值分解来实现主成分分析降维,并使用第 2 章中的气温数据集与线性回归模型来评价主成分分析降维。

实验 3-3 使用特征分解实现主成分分析降维,并对第 2 章回归任务中的气温数据进行降维。画出按降序排列的特征值。

提示:①对数据集中的五维气温数据进行降维;②计算对称矩阵的特征值与特征向量,可使用 np.linalg.eig()或者 np.linalg.eigh()函数,后者返回的结果按特征值升序排列,二者返回的特征向量略有差异。

图 3-2 示出了由气温数据集中五维气温数据得到的特征值。从图中可以看出,在同一城市的不同地点同时采集的气温数据,具有很强的相关性,数据之中存在较多的冗余,可以把这五维气温数据降至一维而不会减少太多的信息量。

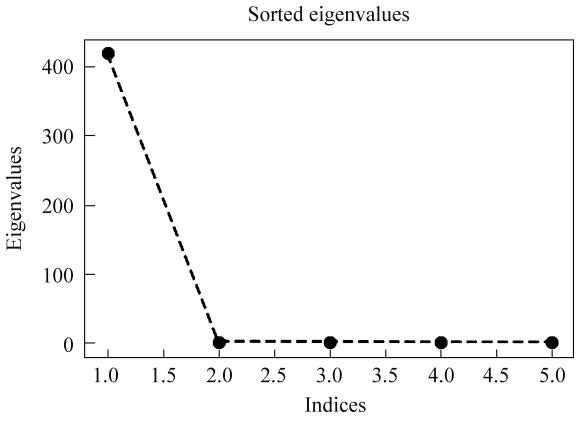


图 3-2 实验 3-3 中按降序排列的特征值

实验 3-4 使用奇异值分解实现主成分分析降维,并对第 2 章回归任务中的气温数据进行降维。画出按降序排列的奇异值。

提示:①对数据集中的五维气温数据进行降维;②对矩阵做奇异值分解可使用 np.linalg.svd()函数。

图 3-3 示出了由气温数据集中五维气温数据得到的奇异值。从图中可以看出,就主成分分析而言,奇异值之间按大小排列的顺序与特征值之间按大小排列的顺序相一致,我们可以根据奇异值的大小排列顺序来依次选取用于降维的奇异向量。在主成分分析中,按大小排列后的奇异值 σ_j 与按大小排列后的特征值 λ_j 之间的关系为 $\lambda_j=\frac{1}{m-1}\sigma_j^2$,$j=1,2,\cdots,d$。

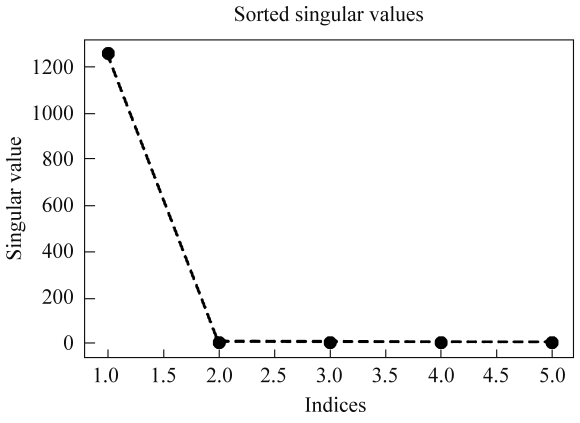


图 3-3　实验 3-4 中按降序排列的奇异值

实验 3-5　使用气温数据集及线性回归评价主成分分析降维。将主成分分析降维输出的降维后的输入特征，作为线性回归模型的输入特征，观察降维后的维数对线性回归模型性能的影响，并与使用未降维输入特征的线性回归模型进行性能比较。

表 3-1 给出了线性回归模型在不同输入特征下的均方根误差（RMSE）。本实验中，我们仍将气温数据集中的前 3000 个样本作为训练样本，后 960 个样本作为测试样本；使用批梯度下降法训练线性回归模型，学习率为 0.1，迭代次数为 100。从表 3-1 中可以看出，使用主成分分析对输入线性回归模型的气温数据进行降维，可以提高线性回归模型的性能，即便降维后输入特征的维数为 3；如果降维后输入特征的维数为 2 或者 1，线性回归模型性能的降幅不大，但是此时输入模型的数据量和相应的计算量都大为减小，体现出使用主成分分析降维的优势。

表 3-1　线性回归模型在不同输入特征下的均方根误差（RMSE）

数据集	使用未经过降维的输入特征（$d=4$）	使用降维后的四维输入特征（$\tilde{d}=4$）	使用降维后的三维输入特征（$\tilde{d}=3$）	使用降维后的二维输入特征（$\tilde{d}=2$）	使用降维后的一维输入特征（$\tilde{d}=1$）
训练数据集 RMSE	0.931	0.877	0.907	0.952	0.959
测试数据集 RMSE	0.742	0.644	0.702	0.76	0.78

3.3　自编码器

主成分分析降维是一种线性降维方法，适用于对各维输入特征之间存在线性关系的数据进行降维。如果使用线性降维方法得到的结果不及预期，或者知道各维输入特征之间不存在线性关系，可以考虑使用非线性方法对数据进行降维，例如使用**自编码器**（autoencoder）。

3.3.1 什么是自编码器

由于降维过程将较高维的输入特征映射为较低维的输入特征,故可以看作是一种定长数据压缩编码过程。因此,如果我们能够自动地对输入特征进行有效的“压缩编码”,也能实现降维。

所谓自编码器,指的是一类特殊的人工神经网络。最基本的自编码器是一种前馈神经网络,如图 3-4 所示,其主要特点是输出层节点的数量等于输入层节点的数量(偏差对应的常数 1 节点不计算在内),并且最中间隐含层节点的数量通常小于输入层和输出层节点的数量。自编码器可以有一个或多个隐含层,最中间的隐含层被称为**码字层**(code layer),因为该层节点的输出即为自编码器输出的编码后的码字。可见,在自编码器中,真正的编码输出层是“码字层”,而不是“输出层”,这与前馈神经网络有所不同。自编码器的输入层节点、隐含层节点、输出层节点与前馈神经网络相同。

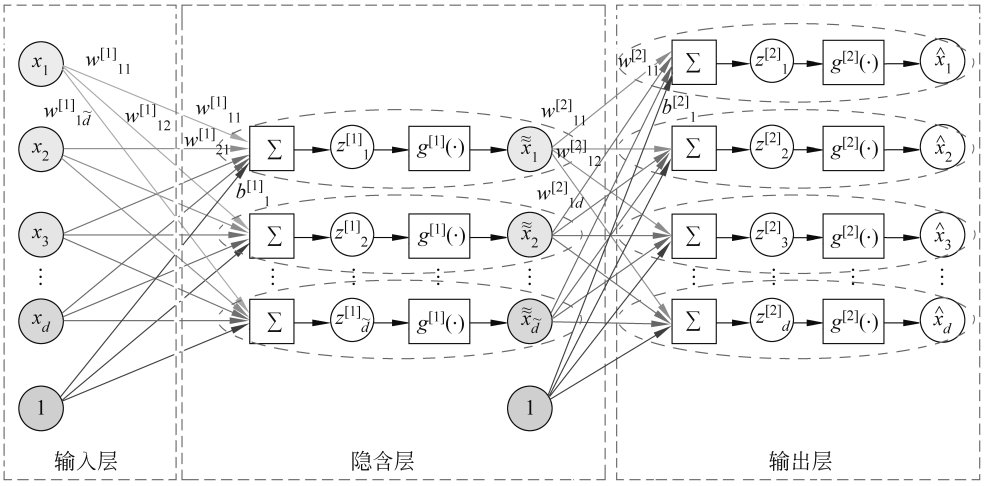


图 3-4 基本的自编码器

那么,自编码器为什么能够“自动编码”?这不仅归因于其具有相同数量的输入层节点和输出层节点,也归因于码字层节点的数量少于输入层和输出层节点的数量,起到了“压缩数据”的作用,还归因于我们训练构成自编码器的神经网络学习预测样本的输入特征。也就是说,在训练过程中,我们使用构成自编码器的包括输入层、隐含层、输出层在内的前馈神经网络,来完成监督学习中的回归任务或(分类任务),将待降维数据集中样本的输入特征作为该神经网络的输入,同时也将该输入特征作为样本的标注,让神经网络学习根据输入特征来预测输入特征。这个做法看上去莫名其妙、多此一举,既然已知输入特征,为何还要预测它们?这是因为自编码器真正的输出层是码字层(也是隐含层),输出层输出的预测值具体为多少并不重要,重要的是我们希望通过使预测值尽量接近于标注,来迫使码字层输出的码字(即降维后的输入特征)尽量对应降维前的输入特征。由于码字层节点的数量少于输入层和输出层节点的数量,所以根据输入特征来准确预测输入特征在很多时候并不是一件很容易做到的事情。

在自编码器中，我们可以将待降维的数据集划分为训练数据集和测试数据集，用训练数据集训练自编码器模型，然后用测试数据集来评价该自编码器模型。如果我们仅需对给定数据集中的样本进行降维，而无须泛化，那么也可以用数据集中的全部样本来训练自编码器模型。在训练完成后，与神经网络不同的是，我们仅使用该自编码器模型的输入层到码字层这部分模型，来对自编码器模型的输入特征进行降维，即在输入层输入待降维的输入特征，然后经过正向传播，在码字层得到模型输出的码字，也就是降维后的输入特征。

3.3.2　自编码器的训练与降维

既然基本的自编码器也是前馈神经网络，我们可以参照前馈神经网络的训练方法来训练基本的自编码器。

首先写出自编码器的正向传播过程。图 3-4 中，输入层和输出层节点的数量都为 d，隐含层(码字层)节点的数量为 $\widetilde{d}$。$\widetilde{d}$ 也是降维后的维数，是一个需要设置的超参数。根据图 3-4 可以写出

$$\widetilde{\widetilde{x}}_j = g^{[1]}(z_j^{[1]}) = g^{[1]}((\boldsymbol{w}_j^{[1]})^{\mathrm{T}}\boldsymbol{x} + b_j^{[1]}) = g^{[1]}\left(\sum_{l=1}^{d} w_{lj}^{[1]} x_l + b_j^{[1]}\right) \tag{3-25}$$

$$\hat{x}_k = g^{[2]}(z_k^{[2]}) = g^{[2]}((\boldsymbol{w}_k^{[2]})^{\mathrm{T}}\widetilde{\widetilde{\boldsymbol{x}}} + b_k^{[2]}) = g^{[2]}\left(\sum_{h=1}^{\widetilde{d}} w_{hk}^{[2]} \widetilde{\widetilde{x}}_h + b_k^{[2]}\right) \tag{3-26}$$

式(3-25)和式(3-26)中，$\widetilde{\widetilde{x}}_j$ 为输入特征 $\boldsymbol{x}$ 经过降维后得到的第 j 维特征；$g^{[1]}(\cdot)$ 为隐含层的激活函数；$z_j^{[1]}$ 为隐含层第 j 个节点线性回归部分的输出；$\boldsymbol{w}_j^{[1]}$ 为隐含层第 j 个节点线性回归部分的权重，$\boldsymbol{w}_j^{[1]} \in \mathbb{R}^{d\times 1}$，$\boldsymbol{w}_j^{[1]} = (w_{1j}^{[1]}, w_{2j}^{[1]}, \cdots, w_{dj}^{[1]})^{\mathrm{T}}$，$w_{dj}^{[1]}$ 为隐含层第 j 个节点的第 d 维输入特征的权重；$\boldsymbol{x} \in \mathbb{R}^{d\times 1}$，$\boldsymbol{x} = (x_1, x_2, \cdots, x_d)^{\mathrm{T}}$；$b_j^{[1]}$ 为隐含层第 j 个节点线性回归部分的偏差；$\hat{x}_k$ 为自编码器对输入特征 $\boldsymbol{x}$ 的第 k 维特征的预测值；$g^{[2]}(\cdot)$ 为输出层的激活函数；$z_k^{[2]}$ 为自编码器输出层第 k 个节点线性回归部分的输出；$\boldsymbol{w}_k^{[2]}$ 为输出层第 k 个节点线性回归部分的权重，$\boldsymbol{w}_k^{[2]} \in \mathbb{R}^{\widetilde{d}\times 1}$，$\boldsymbol{w}_k^{[2]} = (w_{1k}^{[2]}, w_{2k}^{[2]}, \cdots, w_{\widetilde{d}k}^{[2]})^{\mathrm{T}}$，$w_{\widetilde{d}k}^{[2]}$ 为输出层第 k 个节点的第 $\widetilde{d}$ 维输入的权重；$\widetilde{\widetilde{\boldsymbol{x}}} \in \mathbb{R}^{\widetilde{d}\times 1}$，$\widetilde{\widetilde{\boldsymbol{x}}} = (\widetilde{\widetilde{x}}_1, \widetilde{\widetilde{x}}_2, \cdots, \widetilde{\widetilde{x}}_{\widetilde{d}})^{\mathrm{T}}$；$b_k^{[2]}$ 为输出层第 k 个节点线性回归部分的偏差；$j = 1, 2, \cdots, \widetilde{d}$，$k = 1, 2, \cdots, d$。

进一步地，把由式(3-25)给出的 $\widetilde{d}$ 维降维后的输入特征，以及由式(3-26)给出的 d 维输入特征的预测值各自组成一个列向量，则有

$$\widetilde{\widetilde{\boldsymbol{x}}} = g^{[1]}(\boldsymbol{z}^{[1]}) = g^{[1]}((\boldsymbol{W}^{[1]})^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{b}^{[1]}) \tag{3-27}$$

$$\hat{\boldsymbol{x}} = g^{[2]}(\boldsymbol{z}^{[2]}) = g^{[2]}((\boldsymbol{W}^{[2]})^{\mathrm{T}}\widetilde{\widetilde{\boldsymbol{x}}} + \boldsymbol{b}^{[2]}) \tag{3-28}$$

式(3-27)和式(3-28)中，$\boldsymbol{z}^{[1]} \in \mathbb{R}^{\widetilde{d}\times 1}$，$\boldsymbol{z}^{[1]} = (z_1^{[1]}, z_2^{[1]}, \cdots, z_{\widetilde{d}}^{[1]})^{\mathrm{T}}$；$\hat{\boldsymbol{x}} \in \mathbb{R}^{d\times 1}$，$\hat{\boldsymbol{x}} = (\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_d)^{\mathrm{T}}$；$\boldsymbol{z}^{[2]} \in \mathbb{R}^{d\times 1}$，$\boldsymbol{z}^{[2]} = (z_1^{[2]}, z_2^{[2]}, \cdots, z_d^{[2]})^{\mathrm{T}}$；$\boldsymbol{b}^{[1]} \in \mathbb{R}^{\widetilde{d}\times 1}$，$\boldsymbol{b}^{[1]} = (b_1^{[1]}, b_2^{[1]}, \cdots, b_{\widetilde{d}}^{[1]})^{\mathrm{T}}$；$\boldsymbol{b}^{[2]} \in \mathbb{R}^{d\times 1}$，$\boldsymbol{b}^{[2]} = (b_1^{[2]}, b_2^{[2]}, \cdots, b_d^{[2]})^{\mathrm{T}}$；$\boldsymbol{W}^{[1]} \in \mathbb{R}^{d\times\widetilde{d}}$，由式(3-29)给出；$\boldsymbol{W}^{[2]} \in \mathbb{R}^{\widetilde{d}\times d}$，由式(3-30)给出。

$$\boldsymbol{W}^{[1]}=\begin{pmatrix} w_{11}^{[1]} & w_{12}^{[1]} & \cdots & w_{1\widetilde{d}}^{[1]} \\ w_{21}^{[1]} & w_{22}^{[1]} & \cdots & w_{2\widetilde{d}}^{[1]} \\ \vdots & \vdots & & \vdots \\ w_{d1}^{[1]} & w_{d2}^{[1]} & \cdots & w_{d\widetilde{d}}^{[1]} \end{pmatrix} \tag{3-29}$$

$$\boldsymbol{W}^{[2]}=\begin{pmatrix} w_{11}^{[2]} & w_{12}^{[2]} & \cdots & w_{1d}^{[2]} \\ w_{21}^{[2]} & w_{22}^{[2]} & \cdots & w_{2d}^{[2]} \\ \vdots & \vdots & & \vdots \\ w_{\widetilde{d}1}^{[2]} & w_{\widetilde{d}2}^{[2]} & \cdots & w_{\widetilde{d}d}^{[2]} \end{pmatrix} \tag{3-30}$$

更进一步地,我们把数据集中 m 个样本的 d 维输入特征 $\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)},\cdots,\boldsymbol{x}^{(m)}$ 组成矩阵 $\boldsymbol{X}$,$\boldsymbol{X}$ 由式(2-22)给出,$\boldsymbol{X}\in\mathbb{R}^{d\times m}$,$\boldsymbol{x}^{(i)}=(x_1^{(i)},x_2^{(i)},\cdots,x_d^{(i)})^{\mathrm{T}}$,$i\in\{1,2,\cdots,m\}$。为了便于编程实现、减少程序运行时间,我们将 m 个样本在隐含层输出的降维后的输入特征,以及在输出层输出的输入特征的预测值,分别组成一个矩阵,则有

$$\tilde{\tilde{\boldsymbol{X}}}=g^{[1]}(\boldsymbol{Z}^{[1]})=g^{[1]}((\boldsymbol{W}^{[1]})^{\mathrm{T}}\boldsymbol{X}+\boldsymbol{b}^{[1]}\boldsymbol{v}) \tag{3-31}$$

$$\hat{\boldsymbol{X}}=g^{[2]}(\boldsymbol{Z}^{[2]})=g^{[2]}((\boldsymbol{W}^{[2]})^{\mathrm{T}}\tilde{\tilde{\boldsymbol{X}}}+\boldsymbol{b}^{[2]}\boldsymbol{v}) \tag{3-32}$$

式(3-31)和式(3-32)中,$\tilde{\tilde{\boldsymbol{X}}}\in\mathbb{R}^{\widetilde{d}\times m}$,由式(3-33)给出;$\hat{\boldsymbol{X}}\in\mathbb{R}^{d\times m}$,由式(3-34)给出;$\boldsymbol{Z}^{[1]}\in\mathbb{R}^{\widetilde{d}\times m}$,由式(3-35)给出;$\boldsymbol{Z}^{[2]}\in\mathbb{R}^{d\times m}$,由式(3-36)给出;$\boldsymbol{v}$ 仍为 m 维元素都为 1 的行向量,$\boldsymbol{v}\in\mathbb{R}^{1\times m}$,即 $\boldsymbol{v}=(1,1,\cdots,1)$。

$$\tilde{\tilde{\boldsymbol{X}}}=\begin{pmatrix} \tilde{\tilde{x}}_1^{(1)} & \tilde{\tilde{x}}_1^{(2)} & \cdots & \tilde{\tilde{x}}_1^{(m)} \\ \tilde{\tilde{x}}_2^{(1)} & \tilde{\tilde{x}}_2^{(2)} & \cdots & \tilde{\tilde{x}}_2^{(m)} \\ \vdots & \vdots & & \vdots \\ \tilde{\tilde{x}}_{\widetilde{d}}^{(1)} & \tilde{\tilde{x}}_{\widetilde{d}}^{(2)} & \cdots & \tilde{\tilde{x}}_{\widetilde{d}}^{(m)} \end{pmatrix} \tag{3-33}$$

$$\hat{\boldsymbol{X}}=\begin{pmatrix} \hat{x}_1^{(1)} & \hat{x}_1^{(2)} & \cdots & \hat{x}_1^{(m)} \\ \hat{x}_2^{(1)} & \hat{x}_2^{(2)} & \cdots & \hat{x}_2^{(m)} \\ \vdots & \vdots & & \vdots \\ \hat{x}_d^{(1)} & \hat{x}_d^{(2)} & \cdots & \hat{x}_d^{(m)} \end{pmatrix} \tag{3-34}$$

$$\boldsymbol{Z}^{[1]}=\begin{pmatrix} z_1^{[1](1)} & z_1^{[1](2)} & \cdots & z_1^{[1](m)} \\ z_2^{[1](1)} & z_2^{[1](2)} & \cdots & z_2^{[1](m)} \\ \vdots & \vdots & & \vdots \\ z_{\widetilde{d}}^{[1](1)} & z_{\widetilde{d}}^{[1](2)} & \cdots & z_{\widetilde{d}}^{[1](m)} \end{pmatrix} \tag{3-35}$$

$$\boldsymbol{Z}^{[2]}=\begin{pmatrix} z_1^{[2](1)} & z_1^{[2](2)} & \cdots & z_1^{[2](m)} \\ z_2^{[2](1)} & z_2^{[2](2)} & \cdots & z_2^{[2](m)} \\ \vdots & \vdots & & \vdots \\ z_d^{[2](1)} & z_d^{[2](2)} & \cdots & z_d^{[2](m)} \end{pmatrix} \tag{3-36}$$

在自编码器中，我们希望样本输入特征的预测值 $\hat{\boldsymbol{X}}$ 尽量接近于样本输入特征 $\boldsymbol{X}$。由于输入特征的取值通常为连续值，所以可参照线性回归中的做法，通过最小化均方误差代价函数的值，来求得隐含层和输出层的各个权重与偏差，即

$$\boldsymbol{W}^{[1]*},\boldsymbol{b}^{[1]*},\boldsymbol{W}^{[2]*},\boldsymbol{b}^{[2]*}=\underset{\boldsymbol{W}^{[1]},\boldsymbol{b}^{[1]},\boldsymbol{W}^{[2]},\boldsymbol{b}^{[2]}}{\operatorname{argmin}} J(\boldsymbol{W}^{[1]},\boldsymbol{b}^{[1]},\boldsymbol{W}^{[2]},\boldsymbol{b}^{[2]}) \tag{3-37}$$

式(3-37)中，$\boldsymbol{W}^{[1]*}$、$\boldsymbol{b}^{[1]*}$、$\boldsymbol{W}^{[2]*}$、$\boldsymbol{b}^{[2]*}$ 分别为 $\boldsymbol{W}^{[1]}$、$\boldsymbol{b}^{[1]}$、$\boldsymbol{W}^{[2]}$、$\boldsymbol{b}^{[2]}$ 的最优解。参照式(2-14)，自编码器中的均方误差代价函数可以写为

$$\begin{aligned}J(\boldsymbol{W}^{[1]},\boldsymbol{b}^{[1]},\boldsymbol{W}^{[2]},\boldsymbol{b}^{[2]})&=\frac{1}{md}\sum_{i=1}^{m}\sum_{k=1}^{d}(\hat{x}_k^{(i)}-x_k^{(i)})^2=\frac{1}{md}\sum_{i=1}^{m}(\hat{\boldsymbol{x}}^{(i)}-\boldsymbol{x}^{(i)})^{\mathrm{T}}(\hat{\boldsymbol{x}}^{(i)}-\boldsymbol{x}^{(i)})\\&=\frac{1}{md}\sum_{i=1}^{m}(\hat{\boldsymbol{x}}-\boldsymbol{x})^{\mathrm{T}}(\hat{\boldsymbol{x}}-\boldsymbol{x})\big|_{\boldsymbol{x}=\boldsymbol{x}^{(i)}}\end{aligned} \tag{3-38}$$

进一步写出式(3-38)的矩阵形式为

$$J(\boldsymbol{W}^{[1]},\boldsymbol{b}^{[1]},\boldsymbol{W}^{[2]},\boldsymbol{b}^{[2]})=\frac{1}{md}\mathrm{tr}((\hat{\boldsymbol{X}}-\boldsymbol{X})^{\mathrm{T}}(\hat{\boldsymbol{X}}-\boldsymbol{X})) \tag{3-39}$$

式(3-39)中，$\mathrm{tr}(\boldsymbol{A})$ 表示方阵 $\boldsymbol{A}$ 的迹，即方阵 $\boldsymbol{A}$ 的主对角线上的元素之和。

我们知道，在神经网络中，随着层数的增加，代价函数往往不是各层权重与偏差的凸函数。不过我们仍然可以使用梯度下降法求解由式(3-37)给出的最优化问题，尽管不能保证一定能得到全局最优解。在梯度下降法中，需要用到代价函数对权重与偏差的偏导数。为了简化计算式，我们写出中间变量，使用链式法则按照从输出层到输入层的顺序，逐层求代价函数对各层参数的偏导数。与神经网络相同，自编码器中各层的激活函数 $g^{[1]}(\cdot)$、$g^{[2]}(\cdot)$ 仍可使用 ReLU、sigmoid 等函数。在以下推导中，我们以使用 ReLU 激活函数为例，即 $g^{[1]}(z)=g^{[2]}(z)=g(z)=\max(0,z)=z[z>0]$、$\dfrac{\mathrm{d}g(z)}{\mathrm{d}z}=[z>0]$，这里的方括号为艾佛森括号。

练一练　求由式(3-38)给出的代价函数对各层参数 $\boldsymbol{b}^{[2]}$、$\boldsymbol{W}^{[2]}$、$\boldsymbol{b}^{[1]}$、$\boldsymbol{W}^{[1]}$ 的偏导数。

先求代价函数对 $b_k^{[2]}$、$\boldsymbol{w}_k^{[2]}$、$b_j^{[1]}$、$\boldsymbol{w}_j^{[1]}$ 的偏导数，$j=1,2,\cdots,\tilde{d}$，$k=1,2,\cdots,d$。写出式(3-38)中的损失函数为

$$L=\frac{1}{d}(\hat{\boldsymbol{x}}-\boldsymbol{x})^{\mathrm{T}}(\hat{\boldsymbol{x}}-\boldsymbol{x})=\frac{1}{d}\sum_{k=1}^{d}(\hat{x}_k-x_k)^2 \tag{3-40}$$

由此，式(3-38)成为

$$J(\boldsymbol{W}^{[1]},\boldsymbol{b}^{[1]},\boldsymbol{W}^{[2]},\boldsymbol{b}^{[2]})=\frac{1}{m}\sum_{i=1}^{m}L\big|_{\boldsymbol{x}=\boldsymbol{x}^{(i)}} \tag{3-41}$$

由式(3-40)、式(3-26)、式(3-25)可得

$$\frac{\partial L}{\partial b_k^{[2]}}=\frac{\partial L}{\partial \hat{x}_k}\cdot\frac{\partial \hat{x}_k}{\partial z_k^{[2]}}\cdot\frac{\partial z_k^{[2]}}{\partial b_k^{[2]}}=\frac{2}{d}(\hat{x}_k-x_k)[z_k^{[2]}>0] \tag{3-42}$$

$$\frac{\partial L}{\partial \boldsymbol{w}_k^{[2]}}=\frac{\partial L}{\partial \hat{x}_k}\cdot\frac{\partial \hat{x}_k}{\partial z_k^{[2]}}\cdot\frac{\partial z_k^{[2]}}{\partial \boldsymbol{w}_k^{[2]}}=\frac{2}{d}(\hat{x}_k-x_k)[z_k^{[2]}>0]\tilde{\boldsymbol{x}} \tag{3-43}$$

$$\frac{\partial L}{\partial b_j^{[1]}}=\sum_{k=1}^{d}\frac{\partial L}{\partial \hat{x}_k}\cdot\frac{\partial \hat{x}_k}{\partial z_k^{[2]}}\cdot\frac{\partial z_k^{[2]}}{\partial \tilde{\tilde{x}}_j}\cdot\frac{\partial \tilde{\tilde{x}}_j}{\partial z_j^{[1]}}\cdot\frac{\partial z_j^{[1]}}{\partial b_j^{[1]}}$$

$$=\sum_{k=1}^{d}\frac{2}{d}(\hat{x}_k-x_k)[z_k^{[2]}>0]w_{jk}^{[2]}[z_j^{[1]}>0]$$

$$=\frac{2}{d}[z_j^{[1]}>0]\sum_{k=1}^{d}(\hat{x}_k-x_k)[z_k^{[2]}>0]w_{jk}^{[2]} \tag{3-44}$$

$$\frac{\partial L}{\partial \boldsymbol{w}_j^{[1]}}=\sum_{k=1}^{d}\frac{\partial L}{\partial \hat{x}_k}\cdot\frac{\partial \hat{x}_k}{\partial z_k^{[2]}}\cdot\frac{\partial z_k^{[2]}}{\partial \tilde{\tilde{x}}_j}\cdot\frac{\partial \tilde{\tilde{x}}_j}{\partial z_j^{[1]}}\cdot\frac{\partial z_j^{[1]}}{\partial \boldsymbol{w}_j^{[1]}}$$

$$=\sum_{k=1}^{d}\frac{2}{d}(\hat{x}_k-x_k)[z_k^{[2]}>0]w_{jk}^{[2]}[z_j^{[1]}>0]\boldsymbol{x}$$

$$=\frac{2}{d}[z_j^{[1]}>0]\boldsymbol{x}\sum_{k=1}^{d}(\hat{x}_k-x_k)[z_k^{[2]}>0]w_{jk}^{[2]} \tag{3-45}$$

式(3-44)和式(3-45)中,加入求和公式是因为 $\hat{x}_1,\hat{x}_2,\cdots,\hat{x}_d$ 都是 $b_j^{[1]}$、$\boldsymbol{w}_j^{[1]}$ 的函数;这里的方括号为艾佛森括号。根据式(3-41)可得

$$\frac{\partial J(\boldsymbol{W}^{[1]},\boldsymbol{b}^{[1]},\boldsymbol{W}^{[2]},\boldsymbol{b}^{[2]})}{\partial b_k^{[2]}}=\frac{1}{m}\sum_{i=1}^{m}\frac{\partial L}{\partial b_k^{[2]}}\bigg|_{\boldsymbol{x}=\boldsymbol{x}^{(i)}}$$

$$=\frac{1}{m}\sum_{i=1}^{m}\frac{2}{d}(\hat{x}_k-x_k)[z_k^{[2]}>0]\bigg|_{\boldsymbol{x}=\boldsymbol{x}^{(i)}}$$

$$=\frac{2}{md}\sum_{i=1}^{m}(\hat{x}_k^{(i)}-x_k^{(i)})[z_k^{[2](i)}>0] \tag{3-46}$$

$$\frac{\partial J(\boldsymbol{W}^{[1]},\boldsymbol{b}^{[1]},\boldsymbol{W}^{[2]},\boldsymbol{b}^{[2]})}{\partial \boldsymbol{w}_k^{[2]}}=\frac{1}{m}\sum_{i=1}^{m}\frac{\partial L}{\partial \boldsymbol{w}_k^{[2]}}\bigg|_{\boldsymbol{x}=\boldsymbol{x}^{(i)}}$$

$$=\frac{1}{m}\sum_{i=1}^{m}\frac{2}{d}(\hat{x}_k-x_k)[z_k^{[2]}>0]\tilde{\tilde{\boldsymbol{x}}}\bigg|_{\boldsymbol{x}=\boldsymbol{x}^{(i)}}$$

$$=\frac{2}{md}\sum_{i=1}^{m}(\hat{x}_k^{(i)}-x_k^{(i)})[z_k^{[2](i)}>0]\tilde{\tilde{\boldsymbol{x}}}^{(i)} \tag{3-47}$$

$$\frac{\partial J(\boldsymbol{W}^{[1]},\boldsymbol{b}^{[1]},\boldsymbol{W}^{[2]},\boldsymbol{b}^{[2]})}{\partial b_j^{[1]}}=\frac{1}{m}\sum_{i=1}^{m}\frac{\partial L}{\partial b_j^{[1]}}\bigg|_{\boldsymbol{x}=\boldsymbol{x}^{(i)}}$$

$$=\frac{1}{m}\sum_{i=1}^{m}\frac{2}{d}[z_j^{[1]}>0]\sum_{k=1}^{d}(\hat{x}_k-x_k)[z_k^{[2]}>0]w_{jk}^{[2]}\bigg|_{\boldsymbol{x}=\boldsymbol{x}^{(i)}}$$

$$=\frac{2}{md}\sum_{i=1}^{m}[z_j^{[1](i)}>0]\sum_{k=1}^{d}(\hat{x}_k^{(i)}-x_k^{(i)})[z_k^{[2](i)}>0]w_{jk}^{[2]} \tag{3-48}$$

$$\frac{\partial J(\boldsymbol{W}^{[1]},\boldsymbol{b}^{[1]},\boldsymbol{W}^{[2]},\boldsymbol{b}^{[2]})}{\partial \boldsymbol{w}_j^{[1]}}=\frac{1}{m}\sum_{i=1}^{m}\frac{\partial L}{\partial \boldsymbol{w}_j^{[1]}}\bigg|_{\boldsymbol{x}=\boldsymbol{x}^{(i)}}$$

$$=\frac{1}{m}\sum_{i=1}^{m}\frac{2}{d}[z_j^{[1]}>0]\boldsymbol{x}\sum_{k=1}^{d}(\hat{x}_k-x_k)[z_k^{[2]}>0]w_{jk}^{[2]}\bigg|_{\boldsymbol{x}=\boldsymbol{x}^{(i)}}$$

$$=\frac{2}{md}\sum_{i=1}^{m}[z_j^{[1](i)}>0]\boldsymbol{x}^{(i)}\sum_{k=1}^{d}(\hat{x}_k^{(i)}-x_k^{(i)})[z_k^{[2](i)}>0]w_{jk}^{[2]} \tag{3-49}$$

进一步地，由式(3-46)～式(3-49)写出其向量或矩阵形式分别为

$$\frac{\partial J(\boldsymbol{W}^{[1]},\boldsymbol{b}^{[1]},\boldsymbol{W}^{[2]},\boldsymbol{b}^{[2]})}{\partial \boldsymbol{b}^{[2]}}=\frac{2}{md}((\hat{\boldsymbol{X}}-\boldsymbol{X})\circ[\boldsymbol{Z}^{[2]}>0])\boldsymbol{v}^{\mathrm{T}} \tag{3-50}$$

$$\frac{\partial J(\boldsymbol{W}^{[1]},\boldsymbol{b}^{[1]},\boldsymbol{W}^{[2]},\boldsymbol{b}^{[2]})}{\partial \boldsymbol{W}^{[2]}}=\frac{2}{md}\tilde{\tilde{\boldsymbol{X}}}((\hat{\boldsymbol{X}}-\boldsymbol{X})\circ[\boldsymbol{Z}^{[2]}>0])^{\mathrm{T}} \tag{3-51}$$

$$\frac{\partial J(\boldsymbol{W}^{[1]},\boldsymbol{b}^{[1]},\boldsymbol{W}^{[2]},\boldsymbol{b}^{[2]})}{\partial \boldsymbol{b}^{[1]}}=\frac{2}{md}((\boldsymbol{W}^{[2]}((\hat{\boldsymbol{X}}-\boldsymbol{X})\circ[\boldsymbol{Z}^{[2]}>0]))\circ[\boldsymbol{Z}^{[1]}>0])\boldsymbol{v}^{\mathrm{T}} \tag{3-52}$$

$$\frac{\partial J(\boldsymbol{W}^{[1]},\boldsymbol{b}^{[1]},\boldsymbol{W}^{[2]},\boldsymbol{b}^{[2]})}{\partial \boldsymbol{W}^{[1]}}=\frac{2}{md}\boldsymbol{X}((\boldsymbol{W}^{[2]}((\hat{\boldsymbol{X}}-\boldsymbol{X})\circ[\boldsymbol{Z}^{[2]}>0]))\circ[\boldsymbol{Z}^{[1]}>0])^{\mathrm{T}} \tag{3-53}$$

以上 4 式中，$\circ$ 表示阿达马积，即矩阵中的对应元素相乘；$\boldsymbol{b}^{[1]}\in\mathbb{R}^{\widetilde{d}\times 1}$，$\boldsymbol{b}^{[1]}=(b_1^{[1]},b_2^{[1]},\cdots,b_{\widetilde{d}}^{[1]})^{\mathrm{T}}$；$\boldsymbol{b}^{[2]}\in\mathbb{R}^{d\times 1}$，$\boldsymbol{b}^{[2]}=(b_1^{[2]},b_2^{[2]},\cdots,b_d^{[2]})^{\mathrm{T}}$；$\boldsymbol{W}^{[1]}\in\mathbb{R}^{d\times\widetilde{d}}$，由式(3-29)给出；$\boldsymbol{W}^{[2]}\in\mathbb{R}^{\widetilde{d}\times d}$，由式(3-30)给出；$\boldsymbol{X}\in\mathbb{R}^{d\times m}$，由式(2-22)给出；$\hat{\boldsymbol{X}}\in\mathbb{R}^{d\times m}$，由式(3-34)给出；$\tilde{\tilde{\boldsymbol{X}}}\in\mathbb{R}^{\widetilde{d}\times m}$，由式(3-33)给出；$\boldsymbol{Z}^{[1]}\in\mathbb{R}^{\widetilde{d}\times m}$，由式(3-35)给出；$\boldsymbol{Z}^{[2]}\in\mathbb{R}^{d\times m}$，由式(3-36)给出；$\boldsymbol{v}$ 仍为 m 维元素都为 1 的行向量，$\boldsymbol{v}\in\mathbb{R}^{1\times m}$，即 $\boldsymbol{v}=(1,1,\cdots,1)$。

至此，根据式(3-50)～式(3-53)以及式(2-199)，我们就可以使用批梯度下降法训练上述基本的自编码器。在训练过程中如果需要代价函数的值，可按照式(3-39)计算。

在经过训练得到 $\boldsymbol{W}^{[1]}$ 和 $\boldsymbol{b}^{[1]}$ 参数之后，我们就可以根据式(3-31)求出降维后的输入特征 $\tilde{\tilde{\boldsymbol{X}}}$，从而完成自编码器降维任务。式(3-31)中的激活函数同样为 ReLU，即 $g^{[1]}(z)=\max(0,z)=z[z>0]$。

3.3.3　自编码器实践

在本节中，我们动手编程实现基本的自编码器，用该自编码器对螺旋线数据集进行降维，并用第 2 章中的酒驾检测数据集与逻辑回归模型来评价自编码器降维。

helix_dataset

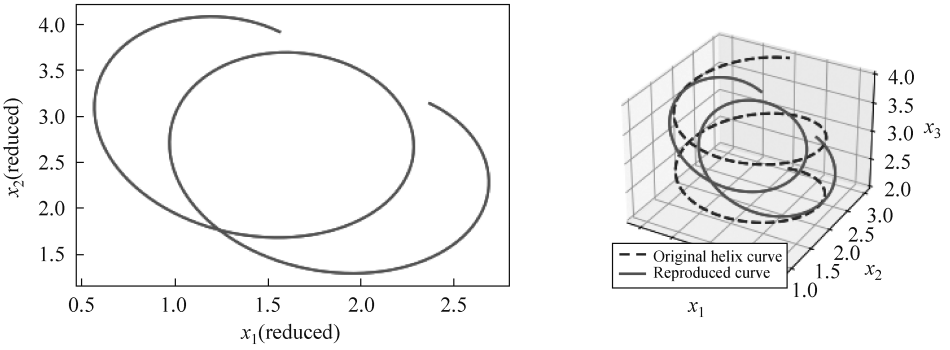
在螺旋线数据集中，样本输入特征的维数为 3，每个样本的三维输入特征对应于三维空间中的一个点，数据集中所有样本的输入特征构成了三维空间中的一条螺旋线。扫描二维码 3-1 可下载螺旋线数据集的 Jupyter Notebook Python 代码。

实验 3-6　实现基本的自编码器，并用该自编码器对螺旋线数据集进行降维，画出降维后的输入特征以及输入特征的预测值。

提示：可将该数据集中的所有样本都作为训练样本。

当数据集中输入特征的偏移为 2、降维后输入特征的维数为 2、随机种子为 1、学习率为 0.01、迭代次数为 100 000 时，得到如图 3-5(a)所示的降维后的二维输入特征，以及如图 3-5(b)中实线所示的输入特征预测值。图 3-5(b)中的虚线为螺旋线，即降维之前的输

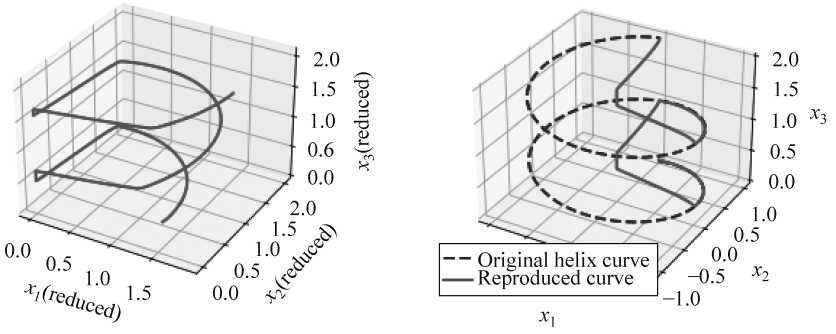
入特征。可见,如果各个样本的输入特征在同一仿射映射下的各维输出值都是正数,那么上述自编码器实际上是通过该仿射映射对输入特征降维,然后将降维后的输入特征再次通过另一个仿射映射映射到降维之前的较高维空间中,从而得到输入特征的预测值,并在这个较高维空间中,最小化输入特征与其预测值之间的欧几里得距离的平方之和(均方误差代价函数)。



(a) 降维后的输入特征(二维)　　(b) 输入特征的预测值

图 3-5　实验 3-6 中降维后的输入特征以及输入特征的预测值(二维、偏移为 2)

当数据集中输入特征的偏移为 0、降维后输入特征的维数为 3(即未实际降维)、随机种子为 1、学习率为 0.01、迭代次数为 100 000 时,得到如图 3-6(a)所示的降维后的三维输入特征,以及如图 3-6(b)中实线所示的输入特征预测值。图 3-6(b)中的虚线仍为螺旋线,即降维之前的输入特征。可见,如果只有一部分样本的输入特征在同一仿射映射下的输出值为正数,那么上述自编码器将仅使用这部分样本降维后的输入特征,来预测这部分样本的输入特征,并在降维之前的较高维空间中,最小化这部分样本的输入特征与其预测值之间的欧几里得距离的平方之和。



(a) 降维后的输入特征(三维)　　(b) 输入特征的预测值

图 3-6　实验 3-6 中降维后的输入特征以及输入特征的预测值(三维、偏移为 0)

实验 3-7　使用实验 3-6 中的自编码器,对酒驾检测数据集中的输入特征进行降维,并使用降维后的训练数据集来训练逻辑回归模型,观察逻辑回归模型在降维后的测试数据集上的性能。

提示：仅使用训练数据集来训练自编码器。

表 3-2 给出了逻辑回归模型在不同维数输入特征下的 F_1 值。这里，我们随机选取（随机种子为 1）酒驾检测数据集中的 250 个样本作为训练样本，余下的 134 个样本作为测试样本，并对样本的输入特征做标准化；使用批梯度下降法训练自编码器和逻辑回归模型，学习率分别为 0.01 和 0.1，迭代次数都为 5000。从表 3-2 中可以看出，当降维后的输入特征的维数为 2～4 时，使用自编码器对该数据集进行降维，可使逻辑回归模型获得与输入未降维输入特征时相仿的性能，即我们以更少的输入数据和更少的计算量获得了相仿的性能，体现出使用自编码器降维的一定优势。

表 3-2　逻辑回归模型在不同维数输入特征下的 F_1 值

数据集	使用未经过降维的输入特征（$d=5$）	使用降为四维后的输入特征（$\widetilde{d}=4$）	使用降为三维后的输入特征（$\widetilde{d}=3$）	使用降为二维后的输入特征（$\widetilde{d}=2$）	使用降为一维后的输入特征（$\widetilde{d}=1$）
训练数据集 F_1 值	0.977	0.976	0.953	0.976	0.898
测试数据集 F_1 值	0.992	0.968	0.984	0.984	0.824

3.4　本章实验分析

如果你已经独立完成了本章前 3 节中的各个实验，祝贺你，可以跳过本节学习。如果未能独立完成，也没有关系，因为在本节中，我们将对本章中出现的各个实验做进一步分析讨论。

实验 3-1　*实现 k 均值聚类，并观察其在轮椅数据集上的聚类结果。更改随机种子后再次观察聚类结果。*

分析：可参考实验 2-19 的程序，读取轮椅数据集，并对样本的输入特征做特征缩放。可借助 rng.integers()方法为每个群组的形心随机分配一个样本的输入特征作为其初始值。在 k 均值聚类中，由于我们事先不知道收敛之前需要进行多少次循环，所以主循环可使用 while 循环。在循环中的第一步，我们根据样本与各个群组形心之间的距离（或距离的平方），将每个样本都划分至一个距其最近的群组。为此，我们用一个数组来记录各个样本对应的群组的序号。我们既可以按照式(3-8)通过 for 循环直接计算出各个样本对应的群组序号，也可以用矩阵形式计算。如果用矩阵形式计算，可先将样本输入特征矩阵（二维数组）通过 np.expand_dims()函数扩展至三维数组，然后使用 Python 的广播操作，计算出所有样本的输入特征向量与所有形心向量之差，然后据此计算出距离的平方，再将与每个样本输入特征相距最近（距离的平方最小）的形心所在的群组的序号作为该样本所在的群组的序号，并保存在序号数组之中。由于我们根据样本与群组之间的对应关