第3章 线性代数基础

本章学习目标

- 理解线性代数的相关概念；
- 掌握范数的有关知识；
- 掌握特征值和奇异值的运算；
- 理解主成分分析的方法和原理。

线性代数作为数学的一个分支，广泛应用于科学和工程中。深度学习会大量涉及线性代数相关的知识，掌握线性代数对理解和从事深度学习算法相关工作是非常必要的。如果大家已经掌握线性代数的有关知识，则可以直接跳过本章内容；如果之前没有接触过线性代数，可以通过阅读和学习本章来了解和掌握本书所需的线性代数知识。

3.1 标量、向量、矩阵和张量

学习线性代数首先需要了解以下几个重要的数学概念。

1. 标量(scalar)

一个标量便是一个单独的数，是计算的最小单元，一般采用斜体小写的英文字母表示。在介绍标量时通常会明确其所属的类型。例如，“$s\in\mathbf{R}$”表示 s 属于实数标量。

2. 向量(vector)

向量是由多个标量构成的一维数组，这些数是有序排列的。通过次序中的索引可以确定每个单独的数，通常赋予向量粗体的小写变量名称，比如 $\boldsymbol{x}$。通过下标索引来获取每一个元素，向量 $\boldsymbol{x}$ 的第一个元素用 x_1 表示，第二个元素用 x_2 表示，以此类推，第 n 个元素用 x_n 表示。当需要明确表示向量中的元素时一般会将元素排列成一个方括号包围的纵列，格式如下所示：

$$\boldsymbol{x}=\begin{bmatrix}x_1\\x_2\\\vdots\\x_n\end{bmatrix}$$

3. 矩阵(matrix)

矩阵是由标量构成的二维数组，其中的每一个元素被两个索引所确定。本书采用粗斜体的大写拉丁字母来表示矩阵，比如 $\boldsymbol{A}$，如果实数矩阵 $\boldsymbol{A}$ 的行数为 m，列数为 n，便可以用 $\boldsymbol{A}\in\mathbf{R}^{m\times n}$ 来表示 $\boldsymbol{A}$ 矩阵。在表示矩阵中的元素时，通常使用其名称以小写不加粗的斜体形

式，索引用逗号间隔。例如，$a_{1,1}$表示矩阵左上角的元素，$a_{m,n}$表示矩阵中右下角的元素。表示垂直坐标 i 中的所有元素时，用“:”表示水平坐标。比如通过 $a_{i,:}$表示 $\boldsymbol{A}$ 矩阵中垂直坐标 i 上的一横排元素，这也被称为 $\boldsymbol{A}$ 矩阵的第 i 行(row)。同样地，$a_{:,i}$表示 $\boldsymbol{A}$ 的第 i 列(column)。在需要明确表示矩阵中的元素时，可以将它们写在用方括号包围起来的数组中，具体形式如下所示：

$$\boldsymbol{A}=\begin{bmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & & \vdots \\ a_{m,1} & \cdots & a_{m,n} \end{bmatrix}$$

如果现在有 m 个用户的数据，每条数据含有 n 个特征，那么这些用户数据实际上对应了一个 $m\times n$ 的矩阵；一张图由 16×16 个像素点组成，那这个图就是一个 16×16 的矩阵。

4. 张量(tensor)

在某些情况下，需要用到高于二维坐标的数组。当一组数组中的元素分布在若干维坐标的规则网格中时被称为张量。本书使用黑斜体大写字母来表示张量，例如，用 $\boldsymbol{A}$ 来表示张量“A”。以二维空间下的三阶张量 $\boldsymbol{A}$ 为例，将张量 $\boldsymbol{A}$ 中的坐标为(i,j,k)的元素记作 $\boldsymbol{A}_{i,j,k}$。为了更好地理解张量，在此将如图 3.1 的一张 RGB 图片表示成一个三阶张量，张量的 3 个维度分别对应图片的高度、宽度和色彩数据。RGB 图片的色彩数据可以拆分为 3 张红色、绿色和蓝色的灰度图片，通过张量的形式来表示，如表 3.1 所示。

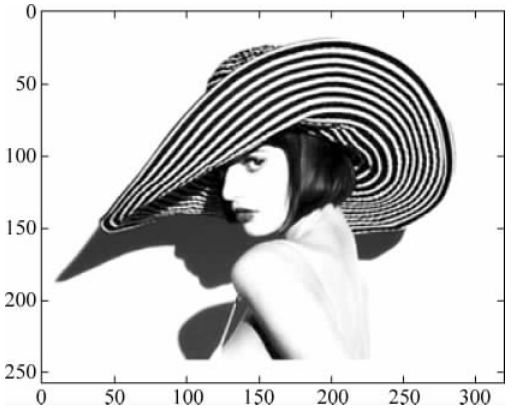

图 3.1　RGB 人像图

表 3.1　通过张量的形式表示 RGB 彩色图片

0	1	2	3	4	…	316	317	318	319
1	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	…	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]
2	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	…	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]

续表

0	1	2	3	4	…	316	317	318	319
3	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	…	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]
4	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	…	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]
5	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	…	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]	[1.0, 1.0, 1.0]

表 3.1 中的横轴表示图片的宽度值；纵轴表示图片的高度值(仅截取其中一部分数据)；表格中每个方格代表一个像素点，[1.0,0,0]表示红色(Red,后面简称 R)，[0,1.0,0]表示绿色(Green,后面简称 G)，[0,0,1.0]表示蓝色(Blue,后面简称 B)。以表 3.1 中第一行第一列为例，表中数据为[1.0,1.0,1.0]，表示 RGB 3 种颜色在图片的该位置的取值情况为 R=1.0,G=1.0,B=1.0。

张量在深度学习中非常重要，它是一个深度学习框架中的核心组件之一，后续的所有运算和优化算法几乎都是基于张量进行的。将各种各样的数据抽象成张量表示，然后再输入神经网络模型进行后续处理是一种非常必要且高效的策略。如果不采用将数据抽象成张量的方法，就需要根据各种不同类型的数据分别定义不同的组织形式，这样十分低效。通过将数据抽象成张量，还可以在数据处理完成后方便地将张量再转换回想要的格式。例如，Python 的 NumPy 包中 numpy.imread 和 numpy.imsave 两个方法，分别用来将图片转换成张量对象(即代码中的 Tensor 对象)，以及将张量再转换成图片保存起来。

矩阵的初等变换是矩阵的一种非常重要的运算，它在求解线性方程组、逆矩阵中都起到了非常重要的作用。对于任意一个矩阵 $\boldsymbol{A}\in\mathbf{R}^{m\times n}$，下面 3 种变换称为矩阵的初等行变换。

- 对调任意两行中的元素，记作 $r_i\rightarrow r_j$；
- 以不为 0 的数 k 乘以矩阵某一行的所有元素，记作 $r_i\times k$；
- 把某一行的所有元素的 k 倍加到另一行的对应的元素上去，记作 $r_i\times k+r_j$。

初等行变换和初等列变换同理，只需将上述 3 种变换中的行改成列即可。初等行变换和初等列变换统称为初等变换。

矩阵等价：如果矩阵 $\boldsymbol{A}$ 经过有限次初等变换变成矩阵 $\boldsymbol{B}$，那么称矩阵 $\boldsymbol{A}$ 与矩阵 $\boldsymbol{B}$ 等价，记作 $\boldsymbol{A}\rightarrow\boldsymbol{B}$。

矩阵之间的等价关系具有下面的性质。

- 自反性：$\boldsymbol{A}\sim\boldsymbol{A}$。
- 对称性：若 $\boldsymbol{A}\sim\boldsymbol{B}$，则 $\boldsymbol{B}\sim\boldsymbol{A}$。
- 传递性：若 $\boldsymbol{A}\sim\boldsymbol{B}$，$\boldsymbol{B}\sim\boldsymbol{C}$，则 $\boldsymbol{A}\sim\boldsymbol{C}$。

转置(Transpose)是矩阵的重要操作之一。矩阵的转置是以对角线为轴的镜像，这条从左上角到右下角的对角线被称为主对角线(Main Diagonal)。将矩阵 $\boldsymbol{A}$ 的转置表示为 $\boldsymbol{A}^{\mathrm{T}}$ 的

表达式为$(\boldsymbol{A}^{\mathrm{T}})_{i,j}=\boldsymbol{A}_{j,i}$，矩阵转置如图 3.2 所示。

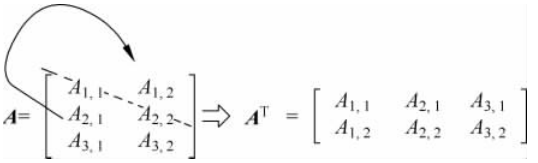

图 3.2　矩阵转置

向量可以看作是只有一列的矩阵，向量的转置可以看作是对只有一行元素的矩阵进行转置。可以将向量表示成行矩阵的转置，写在行中，然后使用转置将其变为标准的列向量，比如 $\boldsymbol{x}=[x_1,x_2,x_3]^{\mathrm{T}}$。

标量可以看作是只有一个元素的矩阵。因此，标量的转置等于它本身，$a=a^{\mathrm{T}}$。任何两个形状一样的矩阵之间都可以相加。两个矩阵相加是指对应位置的元素相加，比如 $\boldsymbol{C}=\boldsymbol{A}+\boldsymbol{B}$，其中 $c_{i,j}=a_{i,j}+b_{i,j}$。

标量和矩阵相乘，或是和矩阵相加时，将其与矩阵的每个元素相乘或相加，比如 $\boldsymbol{D}=a\times\boldsymbol{B}+c$，其中 $d_{i,j}=a\times b_{i,j}+c$。

在深度学习中，会用到一些非常规的符号。允许矩阵和向量相加，产生另一个矩阵：$\boldsymbol{C}=\boldsymbol{A}+\boldsymbol{b}$，其中 $\boldsymbol{C}_{i,j}=\boldsymbol{A}_{i,j}+b_j$。换言之，向量 $\boldsymbol{b}$ 和矩阵 $\boldsymbol{A}$ 的每一行相加。使用这种速记方法时无须在加法操作前定义复制向量 $\boldsymbol{b}$ 到矩阵 $\boldsymbol{A}$ 的每一行。这种隐式地复制向量 $\boldsymbol{b}$ 到很多位置的方式，被称为广播(broadcasting)，广播的相关内容可以回顾第 2 章中讲解的内容。

3.2　线性相关与生成子空间

本章在前面介绍了向量的定义，由多个同维度的列向量构成的集合称为向量组，矩阵可以看成是由行向量或者列向量构成的向量组。本节将讲解向量组中的线性组合、线性相关、最大线性无关组以及矩阵的秩等概念。

3.2.1　线性组合

假设存在向量组 $\boldsymbol{A}$：$\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_n(\boldsymbol{a}_i\in\mathbf{R}^m)$，对于任意一组实数 $k_1,k_2,\cdots,k_n$，有

$$k_1\boldsymbol{a}_1+k_2\boldsymbol{a}_2+\cdots+k_n\boldsymbol{a}_n$$

上述表达式为向量组 $\boldsymbol{A}$ 的线性组合，其中，$k_1,k_2,\cdots,k_n$ 被称为向量系数。如果存在一组不全为 0 实数 $k_1,k_2,\cdots,k_n$，使得任意一个向量 $\boldsymbol{b}$ 满足下列等式：

$$\boldsymbol{b}=k_1\boldsymbol{a}_1+k_2\boldsymbol{a}_2+\cdots+k_n\boldsymbol{a}_n$$

则称向量 $\boldsymbol{b}$ 可以被向量组 $\boldsymbol{A}$ 线性表示。

向量空间是由若干向量构成的非空集合，也称为线性空间。对于任意实数集$\{k_1,k_2,\cdots,k_n\}$，由 $k_1\boldsymbol{a}_1+k_2\boldsymbol{a}_2+\cdots+k_n\boldsymbol{a}_n$ 构成的所有向量集合称为向量空间$\{k_1\boldsymbol{a}_1+k_2\boldsymbol{a}_2+\cdots+k_n\boldsymbol{a}_n,k_i\in\mathbf{R}\}$。

通过人为定义向量空间来容纳“向量”“向量加法”和“向量标量乘法”及三者的性质，让“向量空间”作为这三者的“家”。向量空间是满足“向量加法”和“向量标量乘法”法则的集合(向量集)，假设给定一个域 F，一个非空集合 V 叫做域 F 上的一个向量空间存在向量空间

V 满足以下几个性质：

(1) 交换律：$\boldsymbol{\alpha}+\boldsymbol{\beta}=\boldsymbol{\beta}+\boldsymbol{\alpha}$ 对任意 $\boldsymbol{\alpha},\boldsymbol{\beta}\in V$ 成立；

(2) 结合律：$(\boldsymbol{\alpha}+\boldsymbol{\beta})+\boldsymbol{\gamma}=\boldsymbol{\alpha}+(\boldsymbol{\beta}+\boldsymbol{\gamma})$ 对任意的 $\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}\in V$ 成立；

(3) 存在一个 $\mathbf{0}\in V$，即零向量，满足 $\boldsymbol{\alpha}+\mathbf{0}=\boldsymbol{\alpha}$ 对任意 $\boldsymbol{\alpha}\in V$ 成立；

(4) 对任意的 $\boldsymbol{\alpha}\in V$，存在一个 $-\boldsymbol{\alpha}\in V$，满足 $\boldsymbol{\alpha}+(-\boldsymbol{\alpha})=\mathbf{0}$；

(5) 对任意的 $\boldsymbol{\alpha}\in V, a, b\in F$，有 $(a+b)\boldsymbol{\alpha}=a\boldsymbol{\alpha}+b\boldsymbol{\alpha}$；

(6) 对任意的 $\boldsymbol{\alpha},\boldsymbol{\beta}\in V, a\in \boldsymbol{F}$ 有 $a(\boldsymbol{\alpha}+\boldsymbol{\beta})=a\boldsymbol{\alpha}+a\boldsymbol{\beta}$；

(7) 对任意的 $a, b\in F, \boldsymbol{\alpha}\in V$ 有 $a(b\boldsymbol{\alpha})=(ab)(\boldsymbol{\alpha})$；

(8) $1\boldsymbol{\alpha}=\boldsymbol{\alpha}$。

3.2.2　线性相关

在线性代数中，向量空间的一组元素中，若没有向量可用有限个其他向量的线性组合来表示，则称为线性无关或线性独立(Linearly independent)，反之称为线性相关(Linearly dependent)。

假设存在向量组 $\boldsymbol{A}$：$\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_n$，若任意一组不全为 0 的实数 $k_1,k_2,\cdots,k_n$ 使

$$k_1\boldsymbol{a}_1+k_2\boldsymbol{a}_2+\cdots+k_n\boldsymbol{a}_n=0$$

则称向量组 $\boldsymbol{A}$ 是线性相关的，否则称它为线性无关。其中，$k_1,k_2,\cdots,k_n$ 被称为向量系数。

若向量组 $\boldsymbol{A}$ 是线性无关的，则只有当 $\lambda_1=\lambda_2=\cdots=\lambda_n=0$ 时，下式成立：

$$\lambda_1\boldsymbol{a}_1+\lambda_2\boldsymbol{a}_2+\cdots+\lambda_n\boldsymbol{a}_n=0$$

对于任意一个向量组，不是线性相关就是线性无关的。如果存在一组不全为 0 实数 $k_1,k_2,\cdots,k_n$，使得任意一个向量 $\boldsymbol{b}$ 满足下列等式。

$$\boldsymbol{b}=k_1\boldsymbol{a}_1+k_2\boldsymbol{a}_2+\cdots+k_n\boldsymbol{a}_n$$

则称向量 $\boldsymbol{b}$ 是向量组 $\boldsymbol{A}$ 的一个线性组合，这时也称向量 $\boldsymbol{b}$ 能由向量组 $\boldsymbol{A}$ 线性表示。

3.2.3　向量组的秩

假设存在向量组 $\boldsymbol{A}$：$\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_n$，如果能从中选出由 r 个子向量构成的子向量组 $\boldsymbol{A}_0$：$\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_r, r<n$，且满足以下两个条件的子向量组 $\boldsymbol{A}_0$：$\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_r$ 被称为向量组 $\boldsymbol{A}$ 的一个最大线性无关向量组，最大线性无关向量组包含的向量个数 r 称为向量组 $\boldsymbol{A}$ 的秩。

(1) 向量组 $\boldsymbol{A}_0$：$\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_r$ 线性无关；

(2) 向量组 $\boldsymbol{A}$ 的任意 $r+1$ 个向量构成的子向量组都是线性相关的。

向量组与矩阵存在等价关系，可以把矩阵看成是由所有行向量构成行向量组，矩阵的行秩相当于向量组的秩，矩阵的列秩可以看成是列向量组的秩。

设矩阵 $\boldsymbol{A}$ 的行秩为 $R(\boldsymbol{A})_r$，列秩为 $R(\boldsymbol{A})_c$，则有 $R(\boldsymbol{A})_r=R(\boldsymbol{A})_c$，因此，把矩阵 $\boldsymbol{A}$ 的行秩和列秩统称为矩阵 $\boldsymbol{A}$ 的秩。

3.2.4　实例：求解方程组

设，已知下列 3 个方程，对该方程组进行求解。

$$\begin{cases} x+y=3 \\ x-y=-2 \\ y=1 \end{cases}$$

将上述 3 个方程反映在二维直角坐标系中，如图 3.3 所示。

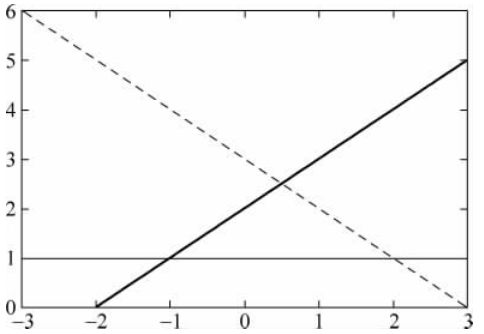

图 3.3　3 个函数的坐标图

由图 3.3 可以看出这 3 条直线没有共同交点，因此无法直接求出该方程组的解。此时，可以通过最小二乘逼近找出一个与 3 条直线距离最近的一个点。

首先，通过矩阵和向量的形式来表示上述方程组方程的表达式为 $Ax+b=y$。

$$\begin{bmatrix}1 & 1\\ 1 & -1\\ 0 & 1\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix}=\begin{bmatrix}3\\ -2\\ 1\end{bmatrix}$$

上述表达式的最小二乘逼近如下所示：

$$\begin{bmatrix}1 & 1 & 0\\ 1 & -1 & 1\end{bmatrix}\begin{bmatrix}1 & 1\\ 1 & -1\\ 0 & 1\end{bmatrix}\begin{bmatrix}x^*\\ y^*\end{bmatrix}=\begin{bmatrix}1 & 1 & 0\\ 1 & -1 & 1\end{bmatrix}\begin{bmatrix}3\\ -2\\ 1\end{bmatrix}$$

$$\begin{bmatrix}2 & 0\\ 0 & 3\end{bmatrix}\begin{bmatrix}x^*\\ y^*\end{bmatrix}=\begin{bmatrix}1\\ 6\end{bmatrix}$$

根据二阶方程的性质对矩阵 $\begin{bmatrix}2 & 0\\ 0 & 3\end{bmatrix}$ 求逆，结果为 $\begin{bmatrix}\frac{1}{2} & 0\\ 0 & \frac{1}{3}\end{bmatrix}$，代入上式可得：

$$\begin{bmatrix}x^*\\ y^*\end{bmatrix}=\begin{bmatrix}\frac{1}{2} & 0\\ 0 & \frac{1}{3}\end{bmatrix}\begin{bmatrix}1\\ 6\end{bmatrix}=\begin{bmatrix}\frac{1}{2}\\ 2\end{bmatrix}$$

最终，求得矩阵的近似解为 $\begin{bmatrix}\frac{1}{2}\\ 2\end{bmatrix}$，如图 3.4 所示。

接下来，通过 numpy.linalg.lstsq()函数在 Python 中实现最小二乘逼近的求解，实现代码如下所示：

```
import numpy
A = numpy.array([[1, 1], [1, -1], [0, 1]])
B = numpy.array([3, -2, 1])
x = numpy.linalg.lstsq(A,B)
print(x)
```

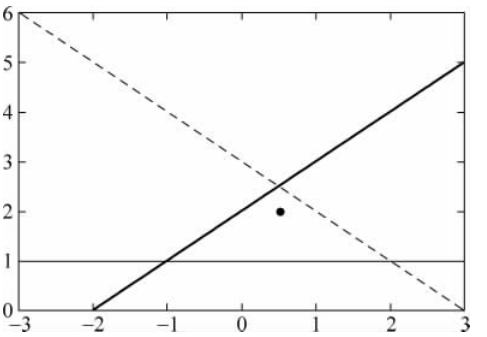

图 3.4　矩阵的近似解

运行结果如下：

```
(array([ 0.5, 2.0 ]), array([ 1.5]), 2, array([ 1.73205081, 1.41421356]))
```

在上述运行结果中，函数返回的 4 个值，从左往右分别对应以下内容：

(1) 最小二乘逼近解。如果 $\boldsymbol{B}$ 为二维，则逼近的结果存在多个列，每一列对应一个逼近解。上述示例中，逼近解为 $\begin{bmatrix}0.5\\2\end{bmatrix}$。

(2) 残差。即每一个 $\boldsymbol{B}-\boldsymbol{A}x$ 的长度的平方。在上述示例中，残差值为 1.5。

(3) 矩阵 $\boldsymbol{A}$ 的秩。上述示例中，矩阵 $\boldsymbol{A}$ 的秩为 2。

(4) 矩阵 $\boldsymbol{A}$ 的奇异值。上述示例中，矩阵 $\boldsymbol{A}$ 的奇异值为 $\begin{bmatrix}1.7320508\\1.41421356\end{bmatrix}$。

3.2.5　实例：线性回归

假设平面直角坐标系中存在 4 个坐标点，坐标分别为(−1,0)、(0,1)、(1,2)、(2,1)，求一条经过这 4 个点的直线方程，坐标点如图 3.5 所示。

设直线的方程为 $y=mx+b$。从图 3.5 可以很容易看出，这 4 个坐标点并不在一条直线上，因此不可能存在一条直线同时连接这 4 个点。此时，可以通过最小二乘的方法找到一条距离这 4 个点最近的直线，来求得近似解。具体方法如下所示。

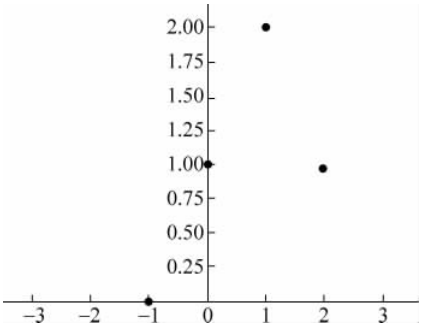

图 3.5　4 个点的坐标

首先，用方程组的形式分别表示图 3.5 中的 4 个点。

$$\begin{cases} f(-1) = -m+b = 0 \\ f(0) = 0+b = 1 \\ f(1) = m+b = 2 \\ f(2) = 2m+b = 1 \end{cases}$$

通过矩阵和向量的形式来表示上述方程组：

$$\underbrace{\begin{bmatrix}-1 & 1\\ 0 & 1\\ 1 & 1\\ 2 & 1\end{bmatrix}}_{A}\underbrace{\begin{bmatrix}m\\ b\end{bmatrix}}_{x}=\underbrace{\begin{bmatrix}0\\ 1\\ 2\\ 1\end{bmatrix}}_{b}$$

上述表达式的最小二乘逼近如下所示。

$$\begin{bmatrix}-1 & 0 & 1 & 2\\ 1 & 1 & 1 & 1\end{bmatrix}\begin{bmatrix}-1 & 1\\ 0 & 1\\ 1 & 1\\ 2 & 1\end{bmatrix}\begin{bmatrix}m^*\\ b^*\end{bmatrix}=\begin{bmatrix}-1 & 0 & 1 & 2\\ 1 & 1 & 1 & 1\end{bmatrix}\begin{bmatrix}0\\ 1\\ 2\\ 1\end{bmatrix}$$

$$\begin{bmatrix}6 & 2\\ 2 & 4\end{bmatrix}\begin{bmatrix}m^*\\ b^*\end{bmatrix}=\begin{bmatrix}4\\ 4\end{bmatrix}$$

接下来，对矩阵$\begin{bmatrix}6 & 2\\ 2 & 4\end{bmatrix}$求逆，结果为$\frac{1}{20}\begin{bmatrix}4 & -2\\ -2 & 6\end{bmatrix}$，将结果代入上述表达式：

$$\begin{bmatrix}m^*\\ b^*\end{bmatrix}=\frac{1}{20}\begin{bmatrix}4 & -2\\ -2 & 6\end{bmatrix}\begin{bmatrix}4\\ 4\end{bmatrix}=\frac{1}{20}\begin{bmatrix}8\\ 16\end{bmatrix}=\begin{bmatrix}\frac{2}{5}\\ \frac{4}{5}\end{bmatrix}$$

因此，所求直线方程为 $y=\frac{2}{5}x+\frac{4}{5}$，具体如图 3.6 所示。

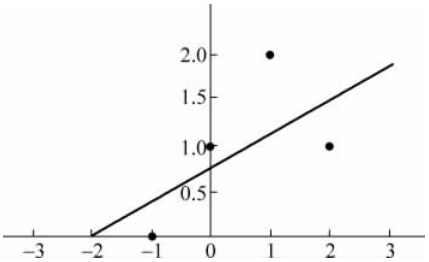

图 3.6　加入直线后的坐标图

图 3.6 中的直线便是经过 4 个点的直线的近似解。通过 Python 实现上述求解过程的代码如下所示：

```
import numpy
A = numpy.matrix('-1 1;0 1;1 1;2 1')
B = numpy.array([0, 1, 2, 1])
x = numpy.linalg.lstsq(A, B)
print (x)
```

运行结果如下：

```
(array([ 0.4, 0.8]), array([ 1.2]), 2, array([ 2.68999405, 1.66250775]))
```

上述运行结果中，numpy.linalg.lstsq 返回的 4 个值从左往右分别对应以下内容：

(1) 最小二乘逼近解。上述示例中，逼近解为$\begin{bmatrix}0.4\\0.8\end{bmatrix}$。

(2) 残差。上述示例中，残差值为 1.2。

(3) 矩阵 $\boldsymbol{A}$ 的秩。上述示例中，矩阵 $\boldsymbol{A}$ 的秩为 2。

(4) 矩阵 $\boldsymbol{A}$ 的奇异值。上述示例中，矩阵 $\boldsymbol{A}$ 的奇异值为$\begin{bmatrix}2.68999405\\1.66250775\end{bmatrix}$。

3.3 范　　数

范数(Norm)是一种定义在赋范线性空间中的函数，在机器学习中为了防止过拟合或使解可逆而引入范数，用范数衡量向量或矩阵的大小。范数包括向量范数和矩阵范数，向量范数表示向量空间中向量的大小，矩阵范数表示矩阵引起变化的大小。

监督类学习问题其实就是在规则化参数同时最小化误差。最小化误差是为了让模型拟合训练数据，而规则化参数是为了防止模型过拟合训练数据。在参数较多时，模型复杂度会极剧上升，容易出现过拟合，即模型的训练误差小，测试误差大。因此，需要尽可能降低模型的复杂程度，并在此基础上降低训练误差，这样才能保证优化后的模型测试误差也小。可以通过规则函数来降低模型的复杂程度。

规则化项可以是模型参数向量的范数，如 L^0、L^1、L^2 等。本节将分别对向量范数和矩阵范数进行讲解。

3.3.1 向量范数

在泛函分析中，向量范数是衡量向量大小的一种度量方式。向量范数(包括 L^p 范数)是将向量映射到非负值的函数。在形式上，向量范数是一个定义域为任意线性空间的向量 $\boldsymbol{x}$，其对应一个实值函数 $\|\boldsymbol{x}\|$，它把一个向量$\boldsymbol{v}$映射为一个非负实数值 R，即满足 $\|\boldsymbol{x}\|$：$\boldsymbol{v}\rightarrow R$。

向量 $\boldsymbol{x}$ 的范数是衡量从原点到点 x 的距离。向量范数需要具备下列 3 个性质。

(1) 正定性：$\|\boldsymbol{x}\|\geqslant 0$，且当 $\|\boldsymbol{x}\|=0$ 时，必有 $\boldsymbol{x}=\boldsymbol{0}$ 成立；

(2) 正齐次性：$\|\alpha\boldsymbol{x}\|=|\alpha|\times\|\boldsymbol{x}\|$，$\alpha\in F$；

(3) 三角不等次性：$\|\boldsymbol{x}\|+\|\boldsymbol{y}\|\geqslant\|\boldsymbol{x}+\boldsymbol{y}\|$。

机器学习领域经常会涉及对范数的应用。范数在机器学习中模型最优化的正则化中具有重要意义，它可以限制损失函数中模型参数的复杂程度。其中最常用到的是 L^p 范数，$p\in\mathbf{R}$，$p\geqslant 1$。L^p 范数的定义如下。

$$\|\boldsymbol{x}\|_p=\left(\sum_i|x_i|^p\right)^{\frac{1}{p}}$$

$\|\boldsymbol{x}\|_p$ 满足范数的 3 个性质。为了方便理解不同范数之间的区别，接下来将分别给出二维空间向量中不同的 p 值对应的单位范数，即 $\|\boldsymbol{x}\|_p=1$ 的图形表示。

1. 向量 L^1 范数

L^1 范数被称为绝对值范数，其大小等于向量的每个元素绝对值之和。L^1 范数表达式如下所示：

$$\| \boldsymbol{x} \|_1 = \sum_i | x_i |$$

其中，单位范数即满足 $\| \boldsymbol{x} \|_1 = 1$ 的点 $\{(x_1, x_2)\}$，如图 3.7 所示。

由于 L^1 范数的天然性质，对 L^1 范数优化的解属于稀疏解，因此 L^1 范数也被称为“稀疏规则算子”。通过 L^1 范数可以实现特征的稀疏，去掉一些没有信息的特征，例如在对用户的网络购物取向做分类的时候，用户有 100 个特征，可能只有十几个特征是对分类有用的，大部分特征如口音、智商等可能都是没有直接关系的特征，利用 L^1 范数就可以过滤掉这些“无用”特征。当机器学习问题中需要严格区分零和非零元素之间的差异时，通常会使用 L^1 范数。

2. 向量 L^2 范数

L^2 范数也被称为欧几里得范数(Euclidean norm)，其大小表示从原点出发到向量 $\boldsymbol{x}$ 的确定点的欧几里得距离(简称欧氏距离)。L^2 范数十分频繁地出现在机器学习中，经常简化表示为 $\| \boldsymbol{x} \|$，省略下标 2。L^2 范数为向量 $\boldsymbol{x}$ 各个元素平方和的开方，其表达式如下所示：

$$\| \boldsymbol{x} \|_2 = \sqrt{\sum_i x_i^2}$$

L^2 范数如图 3.8 所示。

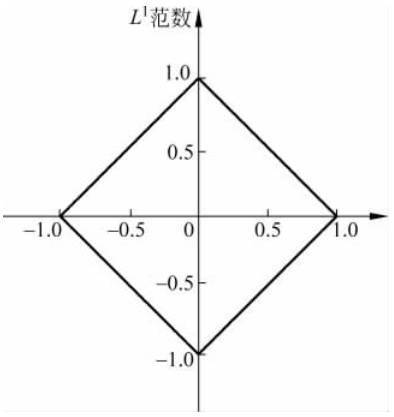

图 3.7　L^1 范数的图像

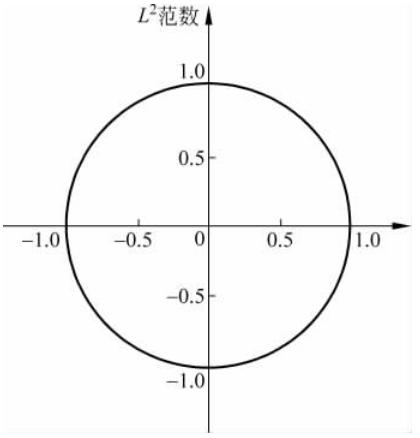

图 3.8　L^2 范数的图像

在衡量向量的值时可能会用到平方 L^2 范数，通过点积计算向量的值。通常，平方 L^2 范数比 L^2 范数更加方便。例如，平方 L^2 范数对向量 $\boldsymbol{x}$ 中每个元素的导数只取决于对应的元素，而 L^2 范数中每个元素的导数与整个向量相关。平方 L^2 范数虽然计算方便，但并不总是适用于任何场合，它在原点附近增长得十分缓慢，而某些机器学习应用中，需要严格区分元素值为零还是非零极小值。此时，更倾向于使用在各个位置斜率相同，且数学形式更简单的 L^1 函数。

3. 当 p 的值为 0 时

有时需要统计向量中非零元素的个数来衡量向量的大小。有些书籍将其称为“L^0 范数”，但是这个术语在数学意义上并不成立，因为向量的非零元素数目并不是范数。对标量放缩 n 倍并不会改变该向量非零元素的数目。因此，通常将 L^1 范数作为表示非零元素数目的替代函数。

4. 当 p 的值为∞时

另外一个经常在机器学习中出现的范数是 L^{∞} 范数，也被称为 max 范数，如图 3.9 所示。当 p 的值趋于∞时，范数表示向量中具有最大幅度的元素的绝对值：

$$\| \boldsymbol{x} \|_{\infty} = \max_i | x_i |$$

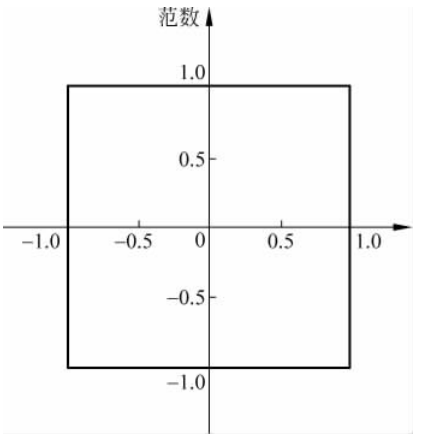

图 3.9　L^{∞} 范数的图像

3.3.2　矩阵范数

除了衡量向量的大小很多时候也需要衡量矩阵的大小，前面所讲到的有关向量范数的知识同样可以应用于矩阵。满足下列所有性质的任意函数 f 称为矩阵范数。

（1）正定性：$\| \boldsymbol{A} \| \geqslant 0$，且当 $\| \boldsymbol{A} \| = 0$ 时，必有 $\boldsymbol{A} = \boldsymbol{0}$ 成立；

（2）正齐次性：$\forall \boldsymbol{A} \in \mathbf{R}^{m \times n}$，$\| \alpha \boldsymbol{A} \| = |\alpha| \times \| \boldsymbol{A} \|$；

（3）三角不等次性：$\| \boldsymbol{A} \| + \| \boldsymbol{B} \| \geqslant \| \boldsymbol{A} + \boldsymbol{B} \|$；

（4）矩阵乘法的相容性：对于任意两个矩阵 $\boldsymbol{A} \in \mathbf{R}^{k \times m}$ 和矩阵 $\boldsymbol{B} \in \mathbf{R}^{n \times k}$，若 $\boldsymbol{A}$ 可以与 $\boldsymbol{B}$ 相乘，则满足 $\| \boldsymbol{A} \| \times \| \boldsymbol{B} \| \geqslant \| \boldsymbol{A} \times \boldsymbol{B} \|$。

本书采用与向量范数相似的表达式 $\| \boldsymbol{A} \|_p$ 来表示矩阵的 p 范数，矩阵通常采用的是诱导范数，诱导范数的定义如下：

（1）假设 $\| \boldsymbol{x} \|_m$ 是向量 $\boldsymbol{x}$ 的范数，$\| \boldsymbol{A} \|_n$ 是矩阵 $\boldsymbol{A}$ 的范数对于任意满足 $\| \boldsymbol{x} \|_m \times \| \boldsymbol{A} \|_n \geqslant \| \boldsymbol{x} \times \boldsymbol{A} \|_m$ 的向量 $\boldsymbol{x}$ 和矩阵 $\boldsymbol{A}$，有矩阵范数 $\| \boldsymbol{A} \|_n$ 与向量范数 $\| \boldsymbol{x} \|_m$ 相容。

（2）假设 $\| \boldsymbol{x} \|_p$ 是向量 $\boldsymbol{x}$ 的 p 范数，定义：$\| \boldsymbol{A} \|_p = \max_{\boldsymbol{x} \neq \boldsymbol{0}} \dfrac{\| \boldsymbol{x} \times \boldsymbol{A} \|_p}{\| \boldsymbol{x} \|_p} = \max_{\| \boldsymbol{x} \|_p = 1} \| \boldsymbol{x} \times \boldsymbol{A} \|_p$，则 $\| \boldsymbol{A} \|_p$ 是一个矩阵范数，并且称该矩阵范数是由向量范数 $\| \boldsymbol{x} \|_p$ 所诱导的诱导范数。

由向量的 p 范数诱导可得矩阵 p 范数。

1. 矩阵 L^2 范数

矩阵 L^2 范数也被称为谱范数，写作 $\| \boldsymbol{A} \|_2$，是由向量 L^2 范数诱导的矩阵范数，其表达式如下所示：

$$\| \boldsymbol{A} \|_2 = \max_j \sqrt{(\lambda_j (\boldsymbol{A}^{\mathrm{T}} \boldsymbol{A}))}$$

其中 $\lambda_j (\boldsymbol{A}^{\mathrm{T}} \boldsymbol{A})$ 表示矩阵 $\boldsymbol{A}^{\mathrm{T}} \boldsymbol{A}$ 的第 j 个特征值。

2. 矩阵 L^1 范数

矩阵 L^1 范数也被称为列和范数，写作 $\|\boldsymbol{A}\|_1$，是由向量 L^1 范数诱导的矩阵范数，其表达式如下所示：

$$\|\boldsymbol{A}\|_1 = \max_j\left(\sum_{i=1}^{m}|\alpha_{ij}|\right),\quad j=1,2,\cdots,m$$

当 p 的值趋于∞时，矩阵 L^∞ 范数也被称为行和范数，写作 $\|\boldsymbol{A}\|_\infty$，是由向量 L^1 范数诱导的矩阵范数，其表达式如下所示：

$$\|\boldsymbol{A}\|_\infty = \max_i\left(\sum_{j=1}^{n}|\alpha_{ij}|\right),\quad i=1,2,\cdots,n$$

在深度学习中，最常见的做法是使用 F 范数(Frobenius norm，简称 F 范数)，其表达式如下所示：

$$\|\boldsymbol{A}\|_F = \sqrt{\sum_{ij}A_{ij}^2}$$

其类似于向量 L^2 范数。在后续章节中处理模型最优化问题时将很多问题转化为 F 范数的最优化问题。

3.4 特殊的矩阵与向量

本节将对几种特殊的矩阵与向量进行讲解，通常任意矩阵都可以由以下几种特殊矩阵组合而成。

1. 对角矩阵(diagonal matrix)

对角矩阵除了主对角线上含有非零元素，其他位置都是 0。假设存在矩阵 $\boldsymbol{D}\in\mathbf{R}^{i\times j}$，当且仅当对于所有的 $i\neq j$，$D_{i,j}=0$ 时，矩阵 $\boldsymbol{D}$ 为对角矩阵。其实本章前面提到的单位矩阵就是对角矩阵。值得注意的是，对角矩阵不一定是方阵，只要 $i\neq j$ 的位置元素值为 0 即可。

主对角元素从左上角到右下角的次序常常记为一个列向量：

$$\boldsymbol{\lambda}=[\lambda_1,\lambda_2,\cdots,\lambda_n]^{\mathrm{T}}=\begin{bmatrix}\lambda_1\\ \lambda_2\\ \vdots\\ \lambda_n\end{bmatrix}$$

以上述列向量为主对角元素的方阵就可以记为 $\mathrm{diag}(\boldsymbol{\lambda})$：

$$\mathrm{diag}(\boldsymbol{\lambda})=\begin{bmatrix}\lambda_1 & 0 & \cdots & 0\\ 0 & \lambda_2 & \cdots & 0\\ 0 & 0 & & 0\\ 0 & 0 & \cdots & \lambda_n\end{bmatrix}$$

用 $\mathrm{diag}(\boldsymbol{\lambda})$ 表示一个对角元素由向量 $\boldsymbol{\lambda}$ 中元素给定的对角方阵。对角矩阵受到关注，部分原因是对角矩阵的乘法计算很高效。计算乘法 $\mathrm{diag}(\boldsymbol{\lambda})\boldsymbol{x}$，只需要将 $\boldsymbol{x}$ 中的每个元素 x_i 放大 λ_i 倍。换言之，$\mathrm{diag}(\boldsymbol{\lambda})\boldsymbol{x}=\boldsymbol{\lambda}\circ\boldsymbol{x}$，其中$\circ$表示向量的点积运算。计算对角矩阵的逆矩阵也很高效。当且仅当对角元素都是非零值时，对角矩阵的逆矩阵存在，在这种情况下，$\mathrm{diag}(\boldsymbol{\lambda})^{-1}=\mathrm{diag}\left(\left[\frac{1}{\lambda_1},\frac{1}{\lambda_2},\cdots,\frac{1}{\lambda_n}\right]^{\mathrm{T}}\right)$。

在大部分情况下，可以根据任意矩阵导出一些通用的机器学习算法，但通过将一些矩阵限制为对角矩阵，可以得到计算代价较低的（并且描述语言较少的）算法。长方形的矩阵也有可能是对角矩阵。长方形对角矩阵没有逆矩阵，但仍然可以很快地计算它们的乘法。

2. 对称矩阵（symmetric matrix）

对称矩阵是其转置与自己相等的矩阵：

$$\boldsymbol{A} = \boldsymbol{A}^{\mathrm{T}}$$

当某些不依赖参数顺序的双参数函数生成元素时，对称矩阵经常会出现。例如，如果 $\boldsymbol{A}$ 是一个表示距离的矩阵，$A_{i,j}$ 表示点 i 到点 j 的距离，那么 $A_{i,j}=A_{j,i}$，因为距离函数是对称的。

3. 单位向量（unit vector）

单位向量是具有单位范数（unit norm）的向量，即满足向量的欧氏范数值为 1：

$$\| \boldsymbol{x} \|_2 = 1$$

上式中向量 $\boldsymbol{x}$ 即为单位向量。

4. 正交向量（orthogonal vector）

假设现有向量$\boldsymbol{v}=[v_1, v_2, \cdots, v_n]^{\mathrm{T}}$，$\boldsymbol{x}=[x_1, x_2, \cdots, x_n]^{\mathrm{T}}$，正交向量满足：

$$\boldsymbol{v}^{\mathrm{T}} \cdot \boldsymbol{x} = \boldsymbol{0}$$

由于$\boldsymbol{v}^{\mathrm{T}} \cdot \boldsymbol{x}=\boldsymbol{0}$，因此向量 $\boldsymbol{x}$ 和向量$\boldsymbol{v}$ 互相正交（orthogonal）。如果两个向量都有非零范数，那么这两个向量之间的夹角是 90°。如果这两个向量不仅互相正交，并且范数都为 1，便可以称它们是标准正交（orthonormal）。

5. 正交矩阵（orthogonal matrix）

正交矩阵是指行向量是标准正交的，列向量是标准正交的方阵：

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} = \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}} = \boldsymbol{I}$$

由此可得 $\boldsymbol{A}^{-1}=\boldsymbol{A}^{\mathrm{T}}$。

正交矩阵的优点之一是求逆计算代价小。需要注意正交矩阵的定义，正交矩阵的行向量不仅是正交的，还是标准正交的。对于行向量或列向量互相正交但不是标准正交的矩阵没有对应的专有术语。

3.5　特征值分解

“分而治之”的策略经常被用于机器学习的算法设计中，许多数学对象可以通过分解成多个子成分或者寻找相关特征，来更好地理解整体。例如，整数可以分解为质数。十进制整数 12 可以用十进制或二进制等不同方式表示，但素数分解是具有唯一性的，通过素数分解可以获得一些有用的信息。例如，$12=2\times2\times3$，通过这个表示可以知道 12 不能被 5 整除，或者 12 的倍数可以被 2 整除等信息。同理，可以通过将矩阵分解成多个子矩阵的方法来发现原矩阵中本来不明显的函数性质。本节将对特征值分解方法进行讲解。

特征值分解（Eigen-Decomposition）是使用最广的矩阵分解方法之一，是一种将矩阵分解成由其特征向量和特征值表示的矩阵之积的方法。

方阵 $\boldsymbol{A}$ 的特征向量是指与 $\boldsymbol{A}$ 相乘后相当于对该向量进行放缩的非 0 向量 $\boldsymbol{x}$：

$$\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}$$

标量 λ 被称为这个特征向量对应的特征值（Eigen value）。非 0 向量 $\boldsymbol{x}$ 称为方阵$\boldsymbol{A}$ 对应

于特征值 λ 的特征向量。

“特征”在模式识别和图像处理中是非常常见的一个词汇。要认识和描绘一件事物，首先要找出这个事物的特征。因此，要让计算机识别一个事物，首先就要让计算机学会理解或者抽象出事物的特征。不论某一类事物的个体如何变换，都存在于这类事物中的共有特点才能被作为“特征”。例如，计算机视觉中常用的 SIFT 特征点（Scale-Invariant Feature Transform）就是一种很经典的用于视觉跟踪的特征点，即使被跟踪的物体的尺度、角度发生了变化，这种特征点依然能够找到关联。

在机器学习中，特征向量选取是整个机器学习系统中非常重要的一步。线性代数中的“特征”是抽象的。矩阵乘法对应了一个变换，是把任意一个向量变成另一个方向或长度不同的新向量。在这个变换过程中，原向量会发生旋转、伸缩变化。如果矩阵对某一个向量或某些向量只发生伸缩（尺度）变化，而没有产生旋转变化（也就意味着张成的子空间没有发生改变），这样的向量就是特征向量。可以通过图 3.10 来理解特征向量的概念。

图 3.10　特征向量的概念

可以看出，图 3.10 中左图通过仿射变换，发生了形变，但是，图像的中心纵轴在变形后并未发生改变。对比上面左右两图中的浅色向量，可以看出其发生了方向的改变，但是深黑色向量的方向依然保持不变，因此深黑色向量可以看作该变换的一个特征向量。深黑色向量在从图 3.10 左图变换成右图时，既没有被拉伸也没有被压缩，其特征值为 1。所有沿着垂直线的向量也都是特征向量，它们的特征值相等。这些沿着垂直线方向的向量构成了特征值为 1 的特征空间。

如果 $\boldsymbol{v}$ 是矩阵 $\boldsymbol{A}$ 的特征向量，那么任何放缩后的向量 $s\boldsymbol{v}$（$s\in\mathbf{R}, s\neq 0$）也是矩阵 $\boldsymbol{A}$ 的特征向量。此外，$s\boldsymbol{v}$ 和 $\boldsymbol{v}$ 有相同的特征值。基于特征向量的该特性，通常可以只考虑单位特征向量。假设矩阵 $\boldsymbol{A}$ 有 n 个线性无关的特征向量 $\{\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^n\}$，它们对应的特征值分别为 $\{\lambda^1, \lambda^2, \cdots, \lambda^n\}$。将 n 个线性无关的特征向量连接一个矩阵，使得每一列是一个特征向量 $\boldsymbol{V}=[\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^n]$，用特征值构成一个新的向量 $\boldsymbol{\lambda}=[\lambda_1, \lambda_2, \cdots, \lambda_n]^{\mathrm{T}}$，此时矩阵 $\boldsymbol{A}$ 的特征分解如下所示。

$$\boldsymbol{A}=\boldsymbol{V}\mathrm{diag}(\boldsymbol{\lambda})\boldsymbol{V}^{-1}$$

值得注意的是，并不是所有矩阵都可以特征值分解，一个大小为 $\mathbf{R}^{n\times n}$ 的矩阵 $\boldsymbol{A}$ 存在特征向量的充要条件是矩阵 $\boldsymbol{A}$ 含有 n 个线性无关的特征向量。

3.6　奇异值分解

3.5 节中探讨了如何将矩阵分解成特征向量和特征值，但特征值分解只适用于方阵，而现实中大部分矩阵都不是方阵，比如电影推荐网站有 m 个用户，每个用户有 n 种偏好，这样形成的一个 $m \times n$ 的矩阵，而 m 和 n 很可能不是方阵，此时可以通过奇异值分解（Singular Value Decomposition，SVD）来对矩阵进行分解。

奇异值分解适用于任意给定的 $m \times n$ 阶实数矩阵分解，它将矩阵分解为奇异向量（Singular Vector）和奇异值（Singular Value），通过奇异向量和奇异值来表述原矩阵的重要特征。除了适用于降维外，奇异值分解还能应用于很多机器学习的工程领域，如图像降噪、购物网站的推荐功能等。

在 3.5 节中，使用特征分解分析矩阵 $\boldsymbol{A}$ 时，得到特征向量构成的矩阵 $\boldsymbol{V}$ 和特征值构成的向量 $\boldsymbol{\lambda}$，现在通过如下形式重新表示矩阵 $\boldsymbol{A}$：

$$\boldsymbol{A} = \boldsymbol{V}\mathrm{diag}(\boldsymbol{\lambda})\boldsymbol{V}^{-1}$$

设 $\boldsymbol{A}$ 是一个 $m \times n$ 的矩阵，$\boldsymbol{U}$ 是一个 $m \times m$ 的矩阵，$\boldsymbol{D}$ 是一个 $m \times n$ 的矩阵，$\boldsymbol{V}$ 是一个 $n \times n$ 矩阵，则矩阵 $\boldsymbol{A}$ 可以分解为如下形式：

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^{\mathrm{T}}$$

这些矩阵每一个都拥有特殊的结构。矩阵 $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 都是正交矩阵，矩阵 $\boldsymbol{D}$ 是对角矩阵。

矩阵 $\boldsymbol{U}=\{\boldsymbol{u}^1,\boldsymbol{u}^2,\cdots,\boldsymbol{u}^m\}$ 是一个 m 阶方阵，其中 $\boldsymbol{u}^i$ 的值是矩阵 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ 的第 i 大的特征值对应的特征向量。$\boldsymbol{u}^i$ 也被称为矩阵 $\boldsymbol{A}$ 的左奇异向量（left singular vector）。

对角矩阵 $\boldsymbol{D}$ 对角线上的元素为 $(\lambda_1,\lambda_2,\cdots,\lambda_k)$，其中 λ_i 是矩阵 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ 的第 i 大的特征值的平方根，$\lambda_i=\sqrt{\lambda_i(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})}$ 被称为矩阵 $\boldsymbol{A}$ 的奇异值。

矩阵 $\boldsymbol{V}=\{\boldsymbol{v}^1,\boldsymbol{v}^2,\cdots,\boldsymbol{v}^n\}$ 是一个 n 阶方阵，其中 $\boldsymbol{v}^i$ 的值是矩阵的列向量，被称右奇异向量（Right Singular Vector）。

奇异值分解可以高效地表示数据。例如，假设想传输如图 3.11 所示的图像，每张图片实际上对应着一个矩阵，像素大小就是矩阵的大小，图中包含 15×25 个黑色或者白色像素。

可以看出，图 3.11 实际上是由如图 3.12 所示的 3 种类型的列所组成。

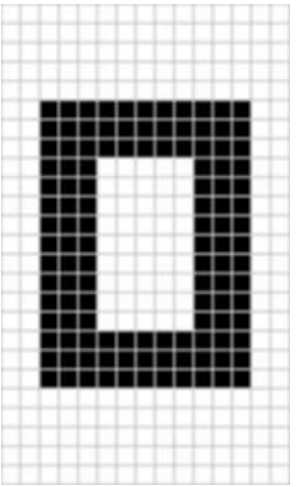

图 3.11　15×25 像素阵列

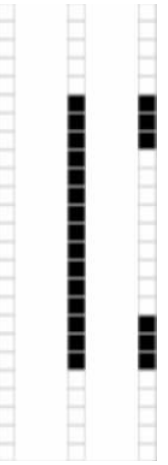

图 3.12　3 种类型的列

通过由各元素为 0 或 1 的 15×25 矩阵来表示图 3.11，其中，0 表示黑色像素，1 表示白色像素，矩阵如图 3.13 所示。

$$\boldsymbol{M}=\begin{bmatrix}
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&0&0&0&0&0&0&0&0&0&0&0&1&1\\
1&1&0&0&0&0&0&0&0&0&0&0&0&1&1\\
1&1&0&0&0&0&0&0&0&0&0&0&0&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&0&0&0&1&1&1&1&1&0&0&0&1&1\\
1&1&0&0&0&0&0&0&0&0&0&0&0&1&1\\
1&1&0&0&0&0&0&0&0&0&0&0&0&1&1\\
1&1&0&0&0&0&0&0&0&0&0&0&0&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1
\end{bmatrix}$$

图 3.13　图片的矩阵表示

奇异值往往对应着矩阵中隐含的重要信息，其包含信息的重要性与值的大小具有正相关性。每个矩阵都可以表示为一系列秩为 1 的“子矩阵”之和，通过奇异值来衡量这些“子矩阵”对应的权重。

如果对 $\boldsymbol{M}$ 进行奇异值分解，可以得到 3 个非零的奇异值(图 3.13 中只有 3 个线性独立的列，因此得到 3 个奇异值，矩阵的序为 3)。假设得到的这 3 个非零奇异值为 σ_1、σ_2、σ_3。矩阵可以通过如下表达式进行近似表达：

$$\boldsymbol{M}\approx \boldsymbol{u}_1\sigma_1\,\boldsymbol{v}_1^{\mathrm{T}}+\boldsymbol{u}_2\sigma_2\,\boldsymbol{v}_2^{\mathrm{T}}+\boldsymbol{u}_3\sigma_3\,\boldsymbol{v}_3^{\mathrm{T}}$$

从图 3.13 中可以看到，该图可以近似由 3 个包含 15 个元素的行向量 $\boldsymbol{v}_i$，3 个包含 25 个元素的列向量 $\boldsymbol{u}_i$，以及 3 个奇异值 σ_i 表达。因此，现在只需要 123(3×15+3×25+3=123)个元素就可以表示这个矩阵，远远少于原始矩中的 375 个元素。

一般情况下，奇异值越大，所对应的信息越重要，这一点可以被应用于数据的降噪处理中。假如，在通过扫描仪将图 3.11 输入到计算机中可能会因为扫描机的原因在原图上产生一些缺陷，这种缺陷通常被称为“噪声”。这时可以通过奇异值对图像去噪。图 3.14 是一张扫描后包含噪声的图片。现假设那些较小的奇异值是由噪声引起的，接下来可以通过令这些较小的奇异值为 0 来达到去除图像噪声的目的。

假设通过奇异值分解得到了矩阵的以下奇异值，由大到小依次为：$\sigma_1=14.15$，$\sigma_2=4.67$，$\sigma_3=3.00$，$\sigma_4=0.21$，…，$\sigma_{15}=0.05$。可以看到，在 15 个奇异值中，从第 4 个奇异值开始数值变得较小，这些较小的奇异值便可能是所要剔除的“噪声”。此时，令这些较小奇异值为 0，仅保留前 3 个奇异值来构造新的矩阵，得到如图 3.15 所示的图像。

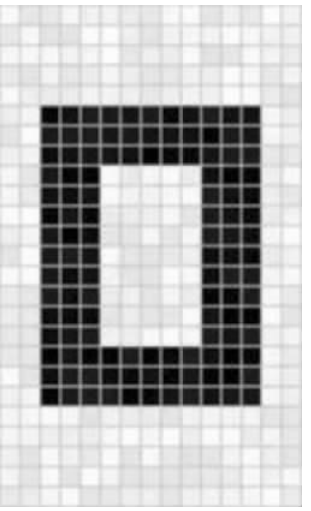

图 3.14　含有噪声的图像

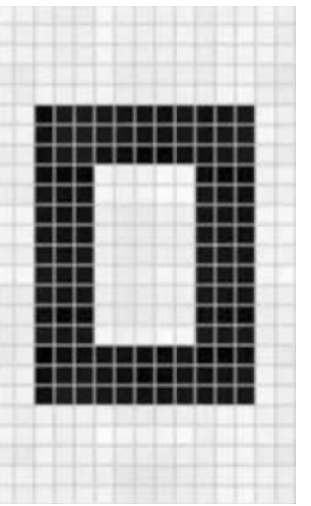

图 3.15　去噪后的图像

与图 3.14 相比，图 3.15 的白格子中灰白相间的图案减少了，通过这种对较小的奇异值置 0 的方法可降低图像噪声。

3.7　迹　运　算

在不使用求和符号的情况下，有的矩阵运算会难以描述，而通过矩阵乘法和迹运算符号，则可以清楚地进行表示。一个 n 阶方阵 $\boldsymbol{A}$ 的主对角线上各个元素的总和被称为矩阵 $\boldsymbol{A}$ 的迹，迹运算返回的是矩阵对角元素的和：

$$\mathrm{Tr}(\boldsymbol{A})=\sum_i \boldsymbol{A}_{i,i}$$

矩阵的迹有众多的性质，接下来将列出较为重要的几种。

(1) 迹运算提供了另一种描述矩阵 F 范数的方式：

$$\|\boldsymbol{A}\|_{\mathrm{F}}=\sqrt{\mathrm{Tr}(\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})}$$

(2) 矩阵的迹运算满足多个等价关系。例如，迹运算在转置运算下是不变的：

$$\mathrm{Tr}(\boldsymbol{A})=\mathrm{Tr}(\boldsymbol{A}^{\mathrm{T}})$$

多个矩阵相乘得到的方阵的迹，和将这些矩阵中最后一个挪到最前面之后相乘的迹是相同的(需要注意的是，在进行该操作时要保证挪动之后的矩阵乘积依然定义良好)。

$$\mathrm{Tr}(\boldsymbol{ABC})=\mathrm{Tr}(\boldsymbol{CAB})=\mathrm{Tr}(\boldsymbol{BCA})$$

n 个矩阵的有效迹的通用的表达式如下所示：

$$\mathrm{Tr}\left(\prod_{i=1}^{n}F^i\right)=\mathrm{Tr}\left(F^n\prod_{i=1}^{n-1}F^i\right)$$

即使循环置换后矩阵乘积得到的矩阵形状发生改变，迹运算的结果仍然保持不变。例如，假设矩阵 $\boldsymbol{A}\in\mathbf{R}^{m\times n}$，矩阵 $\boldsymbol{B}\in\mathbf{R}^{n\times m}$，可以得到如下表达式。

$$\mathrm{Tr}(\boldsymbol{AB}) = \mathrm{Tr}(\boldsymbol{BA})$$

尽管 $\boldsymbol{AB}\in\mathbf{R}^{m\times m}$ 和 $\boldsymbol{AB}\in\mathbf{R}^{n\times n}$。值得注意的是，标量在迹运算后仍然是它自身：$a=\mathrm{Tr}(a)$。

3.8 本章小结

线性代数是应用数学的一个重要分支，其中的矩阵运算是很多机器学习算法，尤其是深度学习算法的基础，通过本章的学习希望大家可以掌握与深度学习相关的线性代数知识点，如果需要进一步全面了解线性代数的相关知识，建议参考相关的专业书籍。

3.9 习　　题

1. 填空题

(1) 每个标量都是一个单独的数，是计算的最_____单元，一般采用斜体小写的英文字母表示。

(2) 当一组数组中的元素分布在若干维坐标的规则网格中时被称为_____。

(3) 对于任意一个向量组，不是线性_____就是线性_____的。

(4) 矩阵 L^{∞} 范数也被称为_____范数，写作 $\|\boldsymbol{A}\|_{\infty}$，是由向量 L^{1} 范数诱导的矩阵范数。

(5) 特征值分解是一种将矩阵分解成由其_____和_____表示的矩阵之积的方法。

2. 选择题

(1) 矩阵 $\boldsymbol{A}=\begin{bmatrix}1 & 0\\ 1 & 0\\ 1 & 1\end{bmatrix}$，则 $\boldsymbol{A}^{\mathrm{T}}$ 为(　　)。

A. $\begin{bmatrix}1 & 1 & 1\\ 0 & 0 & 1\end{bmatrix}$　　B. $\begin{bmatrix}1 & 1 & 1\\ 1 & 0 & 0\end{bmatrix}$　　C. $\begin{bmatrix}1 & 1\\ 0 & 0\\ 0 & 1\end{bmatrix}$　　D. $\begin{bmatrix}1 & 0 & 0\\ 1 & 1 & 1\end{bmatrix}$

(2) 若向量组 $\boldsymbol{A}$：$\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_n$ 是线性无关的，则只有当向量系数满足(　　)时，有 $\lambda_1\boldsymbol{a}_1+\lambda_2\boldsymbol{a}_2+\cdots+\lambda_n\boldsymbol{a}_n=0$ 成立。

A. $\lambda_1\times\lambda_2\times\cdots\times\lambda_n=0$　　B. $\lambda_1=\lambda_2=\cdots=\lambda_n=0$

C. $\lambda_1\times\lambda_2\times\cdots\times\lambda_n=1$　　D. $\lambda_1=\lambda_2=\cdots=\lambda_n=1$

(3) 已知向量 $\boldsymbol{a}=(1,k,1)^{\mathrm{T}}$ 是矩阵 $\boldsymbol{A}=\begin{bmatrix}2 & 1 & 1\\ 1 & 2 & 1\\ 1 & 1 & 2\end{bmatrix}$ 的逆矩阵 $\boldsymbol{A}^{-1}$ 的特征向量，则常数 k 的值为(　　)。

A. -2 或 1　　B. -2 或 -1　　C. -1 或 2　　D. 2 或 1

(4) 下列选项中,(　　)不是矩阵范数的所必须具备的性质。

A. 正齐次性　　B. 正定性

C. 三角等次性　　D. 矩阵乘法相容性

(5) 在迹运算的转置运算中 $\mathrm{Tr}(\boldsymbol{A})=$(　　)。

A. $\sqrt{\mathrm{Tr}(\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})}$　　B. $\mathrm{Tr}(\boldsymbol{A}^{\mathrm{T}})$　　C. $\mathrm{Tr}(\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})$　　D. $\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$

3. 思考题

(1) 简述 L^2 范数可以防止模型过拟合训练数据的原因。

(2) 简述奇异值分解的含义并列举两种可以应用奇异值分解的领域。