第3章　数　　学

一种科学只有在成功地运用数学时，才算达到了真正完善的地步。

——马克思[①]

杨振宁曾在《20世纪数学与物理的分与合》演讲时说："现今只有两类数学著作。一类是你看完了第一页就不想看下去了，另一类是你看完了第一句话就不想看下去了"。

这句话杨振宁曾多次讲过，每次讲到这句话都引起哄堂大笑（看来大家感同身受）。此话事出有因，1969年，杨振宁察觉物理上的规范场理论和数学上的纤维丛理论可能有关系，就把著名拓扑学家 Norman E. Steenrod 的 *The Topology of Fibre Bundles* 一书拿来读，结果是一无所获。原因是该书从头至尾都是定义、定理、推论式的纯粹抽象演绎，生动活泼的实际背景淹没在形式逻辑的海洋之中，使人摸不着头脑。杨振宁说："看了以后完全不懂"。

当然不是杨振宁数学不好，没有任何人（包括专业的数学家）会嘲笑杨振宁的数学能力，英国数学家阿蒂亚爵士认为，杨-米尔斯理论实际上是数学科学大统一的核心，事实上，数学领域的最高奖（菲尔兹奖）有几个获奖者都和研究杨振宁提出的规范场方程有关。

杨振宁的这句话还在 *Mathematical Intelligencer* 一文中公开发表了。有一些数学家当然表示反对，认为数学书本来就应该如此，但是更多数学家对此是表示支持的。杨振宁也说："我相信会有许多数学家支持我，因为数学毕竟要让更多的人来欣赏，才会产生更大的效果"。

杨振宁尚且如此，就更不要说普通人了。大家学习的数学当然不会有杨振宁的那么深奥。不过，对于大多数人接触到的数学书籍，也有两种，一种是没看懂的，另一种是看懂了，但是不知道这些数学能做什么。

虽然按照惯例，本书会解释人工智能中一些名词的意思，对于"数学"一词，本书可不敢解释。好在"数学"是如此基础的一个词语，大家都知道它指代的是什么。

数学号称科学之母，不过，每个人对待数学的态度，其实是很矛盾的。一方面，一提到数学，大家都会觉得很难；另一方面，大部分人在考试的时候，都能得到不错的分数。而对数学的作用，大家有时高估，有时低估，高估时觉得数学无所不能，低估时觉得学完数学不知道有什么用，毕竟去菜市场买菜不需要微积分（连乘方、开方都不需要）。

本书会给大家复习一下高等数学、概率、线性代数里面的一些知识。同时，给一些有用有趣的例子，看看数学到底能做什么。毕竟，数学不只是用来考试的。

---

① 这句话摘自 Paul Lafargue 的《忆马克思》，这句话作为本章的题注很合适。事实上，没有数学，也就没有计算机科学，更别提人工智能了。

3.1 高等数学

高等数学其实是沿袭了苏联的叫法。在高等数学这门课中，大家学习到的内容主要是微积分知识，微积分是很多学科必用的基础工具，从这个角度来说，其实它是“初等”的(elementary)。

大家在学习微积分的时候，都是从微分开始的，然后再学习积分，事实上，从微积分的发展历史上来看，积分比微分要更早出现。积分法的起源是“测量图形的大小”，求面积、体积，都是积分的一种。微积分公式中积分符号∫也是取自拉丁语中“和”的单词 Summa 的首字母 S(莱布尼茨提出并使用)。积分原本就是“和”的意思，大家在学习积分的时候，也是从无数个无穷小量的和开始。计算机在求和的时候，天然就有优势，求和对计算机是很容易的一件事情。微分出现就晚了很多，虽然大家都是从微分开始学起。这里，先复习微分中最基本概念——导数。

3.1.1 导数

要讲导数，可以从函数(function，function 还有功能、方法的意思)说起。在计算机中，函数可以视为一种映射计算，有一种输入，就对应一种输出。例如，在机器翻译中，输入 artificial intelligence，经过某一个函数(方法)的计算，就可能会输出“人工智能”。图 3.1 也表示一个函数，输入一张图片，函数最可能的计算结果是鸟，当然，这个函数也可能有另外一个计算结果——树枝。计算机中的函数拓展了数学中函数的概念，这个函数可能没有解析表达式，只是一段程序。

图 3.2 列出了一些常见函数的导数。如果大家看到这些公式毫无压力，那么放心，本章中剩余的公式比这些公式还简单①。

图 3.1 函数

(图片来源：“八大山人”朱耷作品(局部))

$$c' = 0 \qquad (x^n)' = nx^{n-1}$$

$$(e^x)' = e^x \qquad (a^x)' = a^x \ln a$$

$$(\ln x)' = \frac{1}{x} \qquad (u+v)' = u' + v'$$

$$(uv)' = u'v + uv' \qquad \left(\frac{u}{v}\right)' = \frac{u'v - uv'}{v^2}$$

图 3.2 常见函数的导数公式

复合函数求导是高等数学比较基础的知识，需要用到链式法则。这个法则，也是一个非常重要的法则。图 3.3 是复合函数的导数公式，图中，$y = f(u)$，$u = g(x)$，y 是 u 的函数，u 是 x 的函数。则 y 对于 x 的求导法则如下。

① “一本书上每多一个公式，就会减少一半读者。”很多人说这句话是霍金说的，笔者比较怀疑这是他的原话，不过霍金是个有趣的人，他又是著名的科普书作家，因此，这句话很符合他的风格。减少一半是指数衰减，指数衰减很可怕，本书有上百个公式，全世界才 70 亿人。真的要是这样的话，恐怕本书一个读者也没有了。作为一本讲解人工智能的书，公式必不可少，大家不要害怕公式，仔细研读之后，会发现公式比文字更简洁，更有力量。不过本章虽然是介绍数学，公式并不多，大量公式要到机器学习部分才出现。

$$\frac{\mathrm{d}y}{\mathrm{d}x}=\frac{\mathrm{d}y}{\mathrm{d}u}\frac{\mathrm{d}u}{\mathrm{d}x}$$

对于多元复合函数的链式法则，需要用到偏导数的概念。图 3.4 是一个多元复合函数求偏导数的例子，其中，$z=f(u,v)$，$u=g(x,y)$，$v=h(x,y)$，图 3.4 中只画出 z 对 x 的偏导数。大家可以先记住这个图像，以后学习神经网络，会发现它们长得很像。

$$\frac{\partial z}{\partial x}=\frac{\partial z}{\partial u}\frac{\partial u}{\partial x}+\frac{\partial z}{\partial v}\frac{\partial v}{\partial x}$$

$$\frac{\partial z}{\partial y}=\frac{\partial z}{\partial u}\frac{\partial u}{\partial y}+\frac{\partial z}{\partial v}\frac{\partial v}{\partial y}$$

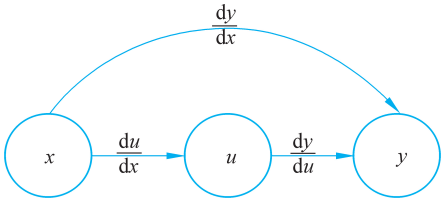


图 3.3　复合函数求导

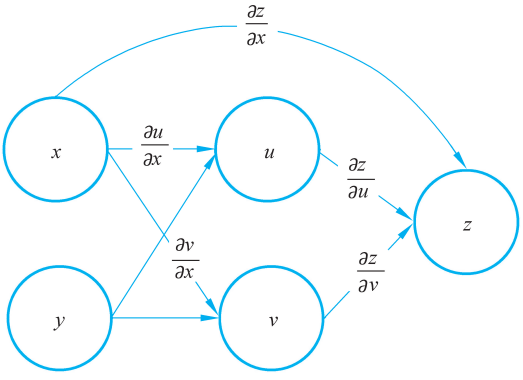


图 3.4　多元复合函数求导

3.1.2　Sigmoid()函数

Sigmoid()函数是本书中出现次数最多的一个函数，因此，这里单独介绍一下这个函数，Sigmoid()函数的解析式如式(3.1)所示。

$$\mathrm{Sigmoid}(x)=\frac{1}{1+\mathrm{e}^{-x}} \tag{3.1}$$

图 3.5 是函数 Sigmoid()的图像。四个图中 x 的取值范围分别是(−1,1)、(−5,5)、(−10,10)和(−100,100)。

画出一个函数的图像是一项非常重要的技能，在人工智能学习中，有很多问题是很抽象的，如果能够画出图来，对大家理解问题非常有帮助，希望大家掌握这种技能。3.5.1 节介绍了如何使用程序绘图。

Sigmoid()函数在很多地方写作 σ 函数，有些地方写作 S 函数。记 Sigmoid()函数为 S 函数，那么它的导数如下：

$$S'=S\cdot(1-S)$$

Sigmoid()函数看起来平平无奇，事实上，它有非常多的优点。首先，这个函数能够把整个实数域的值平滑地映射到(0,1)，正好和概率的区间一致(虽然概率的值域是[0,1]，但是在实际解决问题的时候，概率值很难取到 0 或者 1)；其次，这个函数的导数和自身有相同的形式，也就是，如果需要用到该函数的导数，并不用真正去计算。不用计算在实际应用中是一个非常好的优点，无论在理论推导还是实际应用方面。

这个函数以后会多次出现。

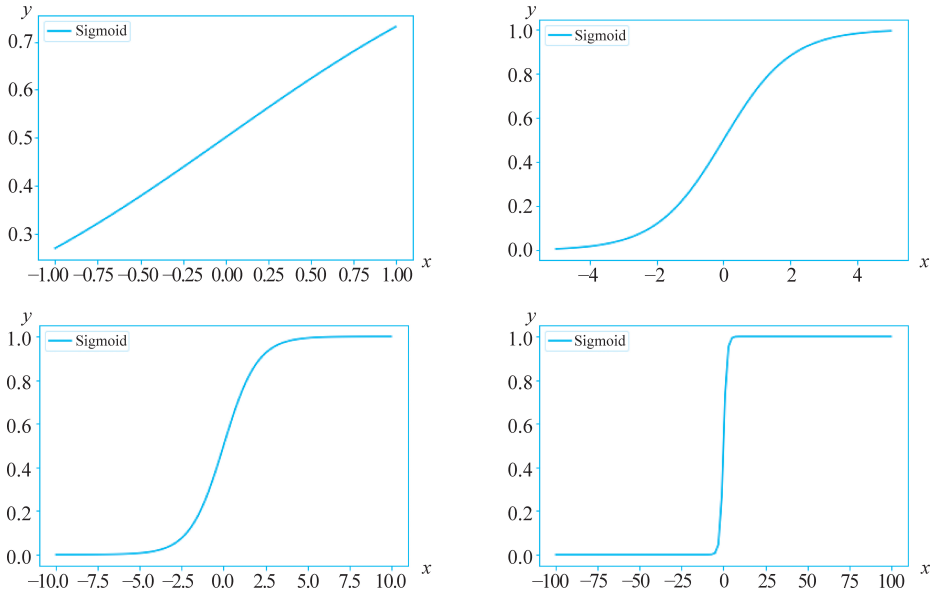


图 3.5 Sigmoid()函数图像

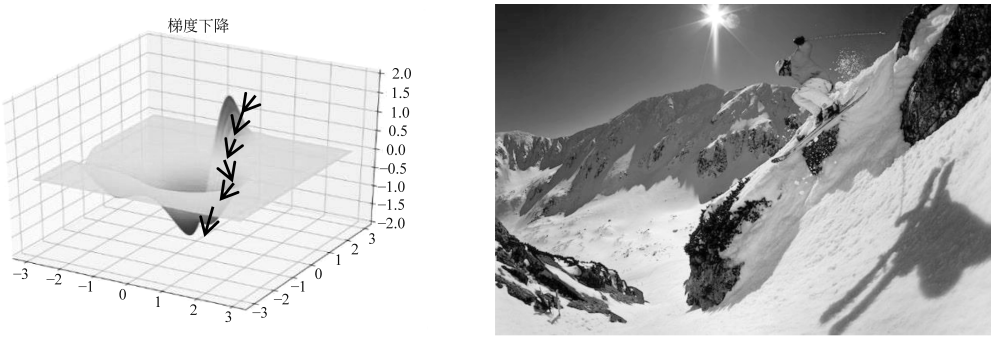
（用程序 3.1 绘制）

3.1.3 梯度下降算法

1. 基本概念

如果说 Sigmoid()函数是本书出现最多的函数，那么梯度下降(gradient descent)算法就是本书中出现最多的一个算法，这是一个非常基础且重要的算法。

所谓梯度，是一个向量，表示某一函数在该点处的方向导数沿着该方向最大，即函数在该点处沿着梯度的方向变化最快，变化率最大。简单地说，梯度即为某点所有方向导数中最大的那个，对于一元函数来说，梯度即为导数。之所以叫下降，因为在求函数最优化的算法里面，一般都习惯求函数的最小值，另外，虽然有梯度提升(gradient boosting)算法，思想上和梯度下降是一致的。图 3.6 是梯度下降算法示意图。



(a) 梯度下降路径
(程序3.2绘制，箭头为手动添加)

(b) 生活中的梯度下降
(图片来自网络，搜索"jasna ski"可得)

图 3.6 梯度下降算法示意图

梯度下降法的基本思想可以类比为找到山谷的过程(其实上山道理也是一样的,但是习惯上,所有的讲解都是讲如何下山,如果是爬山,目标就是如何找到最高峰)。假设这样一个场景:一个人在滑雪,需要最快地到达山谷(找到山的最低点)。但是在山上视野比较狭小,只能看到当前一小块区域,因此,下山的滑雪路径就无法确定,他必须利用自己周围的信息(局部信息)去找到下山的路径。这个时候,他就可以利用梯度下降算法来帮助自己下山。具体来说,以他当前所处的位置为基准,寻找这个位置最陡峭的地方(回忆梯度的概念,梯度就是所有方向导数中最大的那个),然后朝着这个方向滑下去。每走一段距离,都反复采用同一个方法,直到抵达山谷。

梯度下降的算法如下。

算法 1　梯度下降算法

输入:函数 $J(\theta)$,学习率 η

输出:$\operatorname{argmin}\theta$[①]

对于 θ 赋一个初值(可以随机给定);

循环:

按照如下公式改变 θ 的值,使得目标函数按照梯度下降的方向进行减少:

$$\theta \leftarrow \theta - \eta \frac{\partial J(\theta)}{\partial \theta}$$

当结果满足收敛条件的时候,停止循环。

该算法非常简单,也很容易理解。设学习率是 0.5,随机化初始值是 10,也就是初始滑雪者处于 10 这个位置,假设求得此处的梯度为 2,那么滑雪者就应该向梯度方向移动 0.5×2 个距离单位,此时滑雪者新的位置是 10−0.5×2(=9),完成一次循环迭代;此时滑雪者的位置是 9,如果此处的梯度是 3,假设学习率还是 0.5,那么滑雪者就应该向梯度方向移动 0.5×3个距离单位,此时滑雪者新的位置是 9−0.5×3(=7.5),完成一次新的循环迭代;以此类推……

在算法中,η 为学习率或步长,人为指定,过大会导致震荡即不收敛,若过小收敛速度会很慢。下面给出几个梯度下降的例子。

2. 求 $y=x^x$ 的最小值($x>0$)

读到此处,可以暂停一下,自己思考这个最小值是多少。

这个函数看起来很普通,当 x 趋近于正无穷的时候,y 趋近于正无穷;当 x 趋近于 0 的时候,y 趋近于 1;当 x 趋近于 1 的时候,y 趋近于 1。看起来它的最小值是 1。

但是它的最小值却不是 1,因为如果 $x=1/2$,容易计算 $y=\sqrt{2}/2$,这个数约等于 0.7,因此判断 1 不是它的最小值,画出它的图像,如图 3.7 所示。

它的梯度(一元函数,梯度即导数)如下。

$$y' = (1+\ln x)\cdot x^x$$

① 在数学公式中,$\min\theta$ 表示求出 θ 的最小值,$\operatorname{argmin}\theta$ 表示求出当 θ 为何值时,目标函数取得最小值。

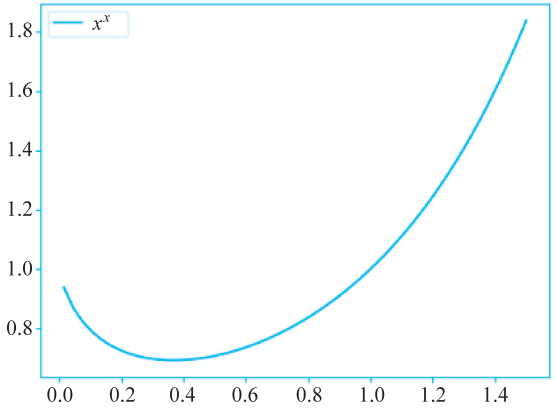


图 3.7　$y=x^x$ 的图像

（由程序 3.3 绘制）

当然，这个题目很简单，令 $y'=0$，最后可以得出结论，当 $x=1/e$ 的时候，y 取得最小值。可以对照式(3.1)，自己编程，利用梯度下降算法求出 x 的值，验证一下，是不是 $x=1/e$。在本章的 3.5 节有对应的程序(程序 3.3)。

这道题目可以通过求方程 $y'=0$ 的根得到，下面看一个不容易求根的例子。

3. 局部最优与全局最优

求下列函数的最小值。

$$y=x^4+0.86\times x^3-12.83\times x^2-9.41x+32$$

这个函数的梯度为

$$y'=4\times x^3+3\times 0.8\times x^2-2\times 12.83\times x-9.41$$

该导数是一个三次函数，这里 $y'=0$ 的根可不容易求。

该函数的图像如图 3.8 所示。图 3.8 中的图像有两个极小值，一个在(−3，−2)，一个在(2，3)，这两个极小值叫局部最优值，所有局部最优值中最优的那个，叫作全局最优值。如果初始值选在 2 附近，利用梯度下降算法可以得到一个全局最优的结果，如果初始值选在 −2

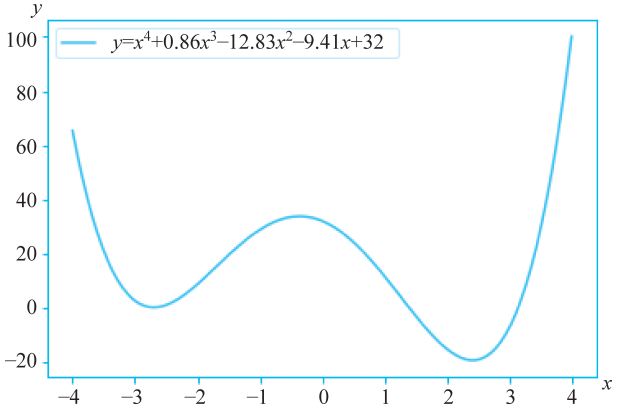


图 3.8　多个局部最优值

（由程序 3.4 绘制）

附近，那么可能只得到一个局部最优值。所以，在梯度下降算法中，如果初始值设置不好，很有可能只能得到局部最优值。

图 3.8 的函数是作者设计的，因为函数图形能够画出，所以很容易通过观察看出答案。事实上，对于生活中的大多数例子，都无法画出图形，无法知道解空间的概况，换句话说，面对一个问题，眼前是一片迷雾，根本不知道全局最优在哪里。很多问题都只能得到一个局部最优解（在人工智能领域，很多时候，能够得到一个局部最优解就已经很不错了）。

对于图 3.8 的函数，第 3.5 节中程序 3.4 显示如何得到不同局部最优值。

除了这种明显的山谷会陷入局部最优之外，还有很多情况，梯度下降也不会收敛，例如在图 3.9 中，函数有一个“平原”[①]。程序 3.5 给出了这个函数的梯度下降算法，事实上，如果初始值选在 0 附近，算法很可能在 0 附近就收敛。同样，如果试图通过调整学习率来跳出平原，结果也会出现震荡。

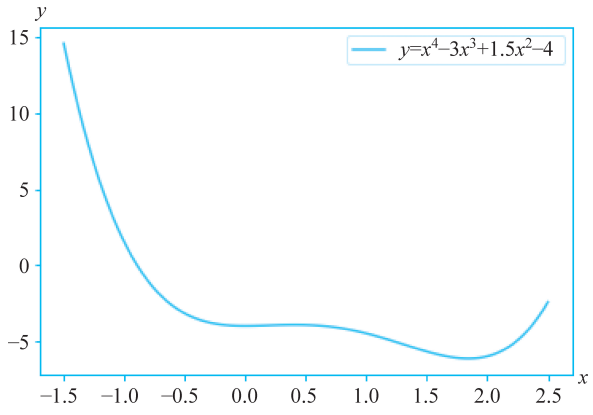


图 3.9　有一个“平原”的函数

（由程序 3.5 绘制）

大家可以把梯度下降理解为滑雪过程，那么，局部最优即是陷入一个山谷，那么，有没有可能跳出山谷？

一个合理的想法就是增大学习率，增大学习率就相当于步子变大了，那么就有可能跳出山谷。这个想法是很有道理，但是实际应用的时候要注意，学习率的控制其实也不是一个简单的事。仔细想一下，学习率的作用是为了控制步长，如果步长太大，就容易引起震荡，最终结果不收敛，事实上，程序 3.4 和程序 3.5 都表明通过控制步长来跳出局部最优并不容易。

学习率在迭代过程中并不是非要保持不变，可以想象一下，很可能开始的时候，离最优值比较远，学习率可以大一些，随着迭代的进行，越来越接近目标，学习率可以小一些。事实上，梯度下降有很多改进算法，包括 AdaGrad、RMSprop，都是让学习率动态变化，在本书第 12 章深度学习部分，会对梯度下降优化算法进一步深入介绍。

在梯度下降算法中，初始值与学习率都会影响最后结果。那么有没有办法能够得到一个好的初始值和学习率呢？原则上没有。当然，如果你对问题很熟的话，可以选择比较合适的初始值和学习率。绝大多数时候，都需要多试几次，例如，可以多试几个初始值，防止掉入

① 不同于图 3.8 中的“多谷”函数（图 3.8 以 π、$\sqrt{2}$、e 等特殊值为根，在纸上设计出来的），“平原”函数更难设计。实际上，图 3.9 是试出来的，图 3.9 看起来在 0 附近比较平，但不是真正的“平原”。

局部最优的坑里。

人工智能不是只为了解决简单问题的。很多问题没有可供使用的函数解析式，甚至很多时候，把问题形式化描述出来都算成功了。人工智能需要面对一个不确定的世界。

高等数学只是一个基本的工具，因为它只描述了一个确定性的世界。而真实生活，则面临一系列不确定。概率是解决不确定性最重要的一个工具。

3.2 概率论

在今天的人工智能技术中，有很多技术都是基于概率与统计知识。2018 年，萨金特在世界科技创新论坛上说："人工智能首先是一些很华丽的辞藻。人工智能其实就是统计学，只不过用了一个很华丽的辞藻"。这里不评价这句话，但是这句话从侧面说明了概率是学习人工智能的一块基石。

虽然本节标题叫概率论，但是这里不会讲解各种概率分布，并且把它们画出来，然后说一下各种分布的应用场景。这种训练，相信大家在学习概率过程中遇到过。本书介绍几个概率的有趣应用。

3.2.1 合取谬误

概率思维并不符合人类的直觉。

举一个例子（这个例子由卡尼曼提出[1]）。先不要看答案，自己快速给出一个选择。

琳达，31 岁，单身，一位直率又聪明的女士，主修哲学。在学生时代，她就对歧视问题和社会公正问题较为关心，还参加了反核示威游行。

快速回答，请问琳达更有可能是下面哪种情况？

1. 琳达是银行出纳。

2. 琳达是银行出纳，同时她还积极参与女权运动。

如果你选择 2，那么你和大多数人的选择一样，虽然这个选择是错误的。事实上，答案 2 是答案 1 的子集（还有很多银行出纳不参与女权运动），所以，1 是比 2 是更好的答案。很多人的选择是 2，这是人类智能的一个特点（但不一定是优点），喜欢过度诠释，因为 2 看起来更合理，有理有据。

下面再看一个经典概率问题，"星期二男孩"。

3.2.2 星期二男孩

招聘的时候，考官都喜欢问应聘者几个智力问题。假设你在应聘的时候，面试官问你这样一个问题："邻居家有两个孩子，已知一个是男孩，求另外一个孩子也是男孩的概率"。你的答案应该是多少？

当然，这里面有个双方都认可的假设，就是生男孩和生女孩的概率是相同的，都是二分之一（虽然统计结果表明这不是事实，但是不影响这里认可这个假设）。

"既然生男生女的概率一样，那么一个孩子是男孩，不会影响另一个孩子吧，所以，这道题的答案应该是 1/2"。

这是个错误答案，为什么？这涉及概率中最重要的一个概念，样本空间。邻居家有两个孩子，那么样本空间一共四个，应该是这样的：

老大：男孩 老二：男孩	老大：男孩 老二：女孩	老大：女孩 老二：男孩	老大：女孩 老二：女孩

问题中邻居家有两个孩子，当其中一个是男孩的时候，样本空间已经缩小了，变成了前三项（灰色底纹）。因此，如果问到另一个也是男孩的概率，那么其实是在这三个样本空间中选，答案是 1/3。

如果面试官接着问这样一个问题："邻居家有两个孩子，已知老大是男孩，求另外一个孩子也是男孩的概率"。这个时候的答案就是 1/2 了。

如果面试官接着问："邻居家有两个孩子，已知一个是男孩且出生在星期二，求另外一个孩子也是男孩的概率"。

男孩的概率和出生在星期几有什么关系？无论星期几生小孩，概率也是一样的。如果你这样想，先不要着急求得答案，希望你能停下来，自己想一想。

答案在表 3.1。

表 3.1　星期二男孩

老大 / 老二	周一	周二	周三	周四	周五	周六	周日	Mon.	Tues.	Wed.	Thur.	Fri.	Sat.	Sun.
周一														
周二														
周三														
周四														
周五														
周六														
周日														
Mon.														
Tues.														
Wed.														
Thur.														
Fri.														
Sat.														
Sun.														

这是一个流行的题目（笔者没有找到最初题目出自哪里），事实上，如果去网络搜索"星期二男孩"，会得到很多搜索结果，也有很多解释方法，这里依然使用样本空间的方法去解释。表 3.1 中，汉字周一、周二等表示男孩，英文缩写 Mon.、Tues.等表示女孩。星期二出生

的小孩的样本空间为表 3.1 中的浅灰色阴影部分，一共 27 个。这里面一共有 13 个样本是两个都是男孩(图中深灰色阴影部分)，因此，最后的答案是 13/27。

合取谬误和星期二男孩都反映了人类智能的局限，对于很多反直觉的概率不能很好地分辨。计算机在这方面就好得多，计算机擅长计算。因为计算机超强的计算能力，可以使用其进行概率模拟仿真，即蒙特卡罗模拟。

3.2.3 蒙特卡罗算法

计算机强大的计算能力表现在很多方面，其中的一方面就是能够进行随机模拟，随机模拟的意思是利用计算机随机地生成一些数据，模拟实际情况，这种模拟方法被称为蒙特卡罗算法。蒙特卡罗算法首先应用在“曼哈顿计划”中，由成员乌拉姆和冯·诺依曼首先提出。如今这个方法已经遍地开花，在很多地方得到了应用(货真价实的“很多”，包括金融、天体物理、气象学、地质统计学、保险业、生物学等几乎所有的日常生活以及科技一系列领域内都应用到了蒙特卡罗模拟算法，当然，这些应用也是计算机发展的必然结果)。

蒙特卡罗算法比较简单。主要包括三个步骤：

(1) 构造整个模拟问题的实际过程。

(2) 分析模拟过程的概率分布情况，生成符合条件的随机变量。

(3) 多次模拟，得到结果的估计量。

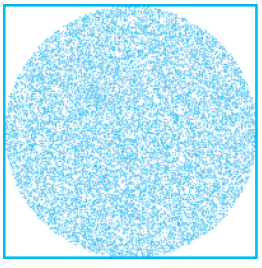


图 3.10 蒙特卡罗求 π
(由程序 3.6 绘制)

这里有一个简单的例子，利用蒙特卡罗算法求 π 的值。

蒙特卡罗求 π 的想法简单直接，如图 3.10 所示。在一个正方形框内按照均匀分布随机生成若干个点(一般要生成很多个，依照结果的精度而定，假设有 total 个)，假设正方形框内有一个内接圆，这样，就有一些点落在圆的内部(个数为 count)，其他点落在圆的外部。落在圆内部的点的个数 count 和总的点的个数 total 之比就应该等于它们的面积之比。简单分析可以得到，$\pi = 4 \times \text{count}/\text{total}$。

本章程序 3.6 给出了图 3.10 的绘制方法以及蒙特卡罗求 π 的过程。

虽然简单，但是蒙特卡罗算法却有着巨大的实用价值，例如，AlphaGo 即利用了蒙特卡罗树模拟可能的落子方法，在后面的博弈中会简单介绍。

3.2.4 三门问题

求 π 的问题比较简单，不用蒙特卡罗方法，大家也知道 π 的值。下面给出另外一个问题，这个问题困扰了很多人。

三门问题：假设你参加一个综艺节目，面前有三扇关闭的门，你只有一次打开门的机会。其中一扇门后面有一辆汽车，打开该扇门可赢得该汽车，另外两扇门后面则各藏有一只山羊，选中它们只能得到山羊。你的目标是获得一辆汽车。你选了一个门之后，先不要打开，主持人此时会在另外两扇门中打开一扇，露出门后面的羊。

这时候主持人问你，现在是否还要坚持自己原来的选择。图 3.11 是三门问题的示意图。

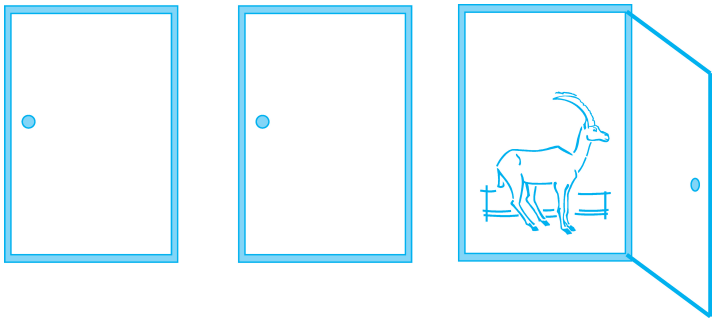


图 3.11　三门问题

这个问题困扰了很多人，据说著名数学家埃尔德什也出过错。乍一看，主持人打开了一扇门，还剩两扇门未开，这个时候哪个门的后面藏有小汽车的概率应该都一样，都是 1/2，所以换不换都一样。而且考虑到塞勒提出的禀赋效应[①]，如果你换了，结果还错了，那么这种后悔程度会更大。大多数人的选择是“不换”。

不过，这道题目的正确答案是：换。

可以这么想，你选择了一扇门，那么选中小汽车的概率就是 1/3，而剩下这两扇门加在一起，里面有小汽车的概率就是 2/3，现在主持人把一扇门打开了，相当于主持人替你把这 2/3 里面不是汽车的那一个选项去掉了。也就是你换了的话，概率是 2/3，不换的话，概率是 1/3。

如果这个解释还不太清楚的话，举另外一个例子。假设双色球中头奖的概率是 1700 万分之一，现在有 1700 万张彩票，里面有一个一等奖，你选对了就是一等奖，选不中就没奖，你只能开奖一次。你买了一张彩票，如果头脑正常的话，你不会认为你买的这张彩票是一等奖（事实上，大多数人都知道，这张是头奖的概率是 1/17000000）。好，这个时候，有个菩萨来帮助你——菩萨也要走个流程，她不会直接告诉你答案——她翻开一张彩票，告诉你这张不对，又翻开另外一张，告诉你这张也不是……一共翻开了 16999998 张，告诉你翻开的这些都不是一等奖。现在就剩 2 张了，一张是你手中的，一张还没有翻开，这两张肯定有一张是一等奖，你说，有奖的是哪张？谁都不会傻到相信先前手中这张正好中奖，合理的想法是马上就换菩萨剩下的那张。这个时候再分析就很清楚了，你手中的彩票中一等奖的概率是 1/17000000，而剩下的所有的彩票中一等奖的概率是 16999999/17000000，但是这堆彩票中已经有 16999998 张被翻开了，剩下的那张几乎就是一等奖了。

理解这道题目的关键就是主持人的做法，你选择了一扇门之后，还剩下两扇门，主持人从这两扇门中选择打开一扇门，注意，主持人的选择不是任意的，他只会打开后面是山羊的那扇门。

如果你还是不相信这个推理过程，那么，请参见 3.5 节部分程序 3.7。

现在能找到的三门问题的最早出处，来自美国的电视游戏节目 Let's Make a Deal。问题来自该节目的主持人蒙提·霍尔，所以三门问题也叫蒙提·霍尔问题。

① 禀赋效应是指个人一旦拥有某项物品，那么对该物品价值的评价要比未拥有之前大大提高。

文献[2]记载很多人在这个问题上出错。这是一个非常易错的题目，不管对于普通读者还是专业人士。美国有个节目叫 Ask Mailyn(去问玛丽莲)，这个节目的主持人 Mailyn 在美国 *Parade* 杂志上给出了这个题目的正确结果。但是一万多名读者给杂志写信，断言 Mailyn 搞错了，其中一千多名读者具有博士学位。在这个题目上出错的甚至包括埃尔德什。

不管埃尔德什正确或者错误，丝毫不影响他在数学上的地位。另外，如果他真的在这个问题上错误了，会使普通人对于数学有更多信心。

埃尔德什是数学大师(匈牙利天才之一)，被誉为 20 世纪的欧拉，1984 年沃尔夫奖获得者(与中国著名数学家陈省身同年得奖)。埃尔德什十分高产，发表论文数量在数学界排名第一(欧拉排第二，也许这就是为什么埃尔德什被称为 20 世纪的欧拉)，他十分乐于合作，乐于提携后代(著名华裔数学家陶哲轩即曾被埃尔德什提携)，因此很多人和埃尔德什合作过。这个特点也使得埃尔德什在学术网络领域有个专门的词，“埃尔德什数”，在 11.7 节会介绍这个数字。

3.2.5 信息论

为信息论做出贡献的人很多，其中，香农奠定了信息论的理论基础。

神级人物之香农

克劳德·艾尔伍德·香农(Claude Elwood Shannon，1916—2001 年)，美国信息学家、数学家、信息论之父。见图 3.12。

图 3.12 香农

香农对人类的贡献有很多，1948 年，香农发表论文 *A Mathematical Theory of Communication*(《通信的数学原理》)，最早提出了信息熵的概念，用于衡量信息的不确定性，提出了信息是可以度量的。可以说，没有信息论，就没有现代通信技术，互联网、手机等现代产品也不会这么快地出现。

香农硕士论文的题目是 *A Symbolic Analysis of Relay and Switching Circuits*(《继电器与开关电路的符号分析》)，这篇论文发表于 1938 年。尽管这个时候，晶体管还未出现，现代电子计算机还在萌芽阶段。他把布尔代数的“真”与“假”和电路系统的“开”与“关”对应起来，并用 1 和 0 表示。于是他用布尔代数分析并优化开关电路，这就奠定了数字电路的理论基础。哈佛大学的 Howard Gardner 教授说，“这可能是 20 世纪最重要、最著名的一篇硕士论文”。

和其他的天才人物一样，香农的研究方向也是多种多样，1940 年香农在 MIT 获得数学博士学位，而他的博士论文却是关于人类遗传学的，题目是 *An Algebra for Theoretical Genetics*(《理论遗传学的代数学》)。

如果说用“天才”一词来形容冯·诺依曼，那么形容香农可以用一个词“低调”。

香农被低估了，他并不为大众所熟知，就他的贡献而言，他应该比现在的名气更大。早期他的中文名字一度被翻译为“仙农”(虽然这个名字也很好听)。

《财富公式》一书中这样描写香农[5]："贝尔实验室和 MIT 有很多人将香农和爱因斯坦相提并论，而其他人也则认为这种对比是不公平的——对香农不公平。"

文献[5]说："香农的影响巨大怎么强调也不为过，就好比字母表的发明对于文学产生的巨大影响一样"。

香农不为大众所知，很可能与他自己低调的性格有关。在 1948 年发表了信息论之后，香农声名鹊起，但是他发表了一篇四段的文章，善意地敦促世界其他地方放弃他的"潮流"。正如他所说，"(信息理论)可能膨胀出的重要性已超出其实际成就。"(实际情况是重要性远远超过大家的想象，人类迎来了轰轰烈烈的信息时代)。

上面的最后一段话来自网络，未找到出处，但是依照香农的性格，这个很可能是真的。第 1 章介绍了达特茅斯会议，这个会议的由来其实是这样的。20 世纪 50 年代的时候，香农已经对机器与智能很感兴趣了，于是他把明斯基和麦卡锡招到自己的手下实习打工。研究过程中，研究组的人员(一种说法是麦卡锡，另一种说法是一个研究生 Jerry Rayna)建议香农汇编一个论文集，把机器模拟智能的论文汇编在一起。香农觉得机器模拟智能这个名称太高调了，低调地把它改成 *Automata Studies*(《自动机研究》)，但是因为这个名字实在不吸引人，大量的文章都和图灵机以及智能没什么关系(可见书名的重要性)，这次论文汇编没有起到预想的作用。所以在 1956 年夏，以香农为名义召集人，麦卡锡和明斯基为实际召集人，召开了达特茅斯会议。这次会议起名字时就吸取了教训，叫作 Summer Research Project on Artificial Intelligence。

另外一件事也可以证明香农的低调，香农信息论的奠基性论文最开始叫 *A Mathematical Theory of Communication*，后来论文集结集出版之后，才改成了 *The Mathematical Theory of Communication*。稍微懂英语都知道 A 和 The 的区别，就这个论文的意义来说，就应该用 The，用 A 确实很低调。

在三个神级人物里，香农是最长寿的，不过香农在 20 世纪 50 年代以后，就渐渐离开了学术界，隐居起来。以至于很多人以为他已经去世了，江湖上只留下他的传说。1985 年，离开学术界 30 年的香农参加了在英国举办的一次国际信息理论探讨会。香农的出现引爆了会场，大家排队要签名，研讨会主席 Robert McEliece 回忆说："那个画面，就好像牛顿他老人家忽然出现在现代物理学会议上。"

香农晚年对杂耍很感兴趣，真正字面意义的杂耍(图 3.12 中香农在骑独轮车)。香农家里最显著的地方，摆着杂耍学博士证书(Doctor of Juggling，但是美国并没有这样一个学位，所以很可能是香农的恶作剧)。在前文提到的 1985 年国际信息理论探讨会上，香农在演讲时，为了怕大家无聊，居然拿出三个手抛球开始玩杂耍。

香农晚年患上了阿尔茨海默病，2001 年辞世。

主要资料：文献[3-7]

本书的神级人物只写这三个人。当然，人工智能领域，涉及的传奇人物众多，但是这三个人，称之为神级人物是当之无愧的。

这三个人关系也很好，图灵是英国人，在美国和冯·诺依曼一起工作过，冯·诺依曼还试图说服图灵到美国工作。图灵在美国访问贝尔实验室期间和香农交流得很好，香农去英国期间和图灵也交流得很好，那时候图灵正好在研究会下棋的机器，香农也很感兴趣。香农和冯·诺依曼也交流密切，就是冯·诺依曼说服香农使用"熵"这个词来衡量信息。

1. 信息熵

严格说来，信息论不算是基础数学的一部分，介绍信息论，也不能说是对于数学的复习。但是，信息的度量是以概率的形式描述的，因此，把信息论的介绍，放到了概率一节。

事实上，信息(information)这个词很难定义，都知道今天是信息时代，但是对于信息，仍然没有一个严格的定义。香农在提出信息论的时候就说过，信息的定义本身不重要，更重要的是，信息能做什么。所谓信息，不过是对一些不确定性的度量。

伟大的思想都是相似的，图灵测试也是这样——重点不是它是什么，而是它能做什么。

信息是有价值的。假设今天是星期一，甲、乙两个人分别说了如下话语。

甲：明天是星期二。

乙：明天会下大雨。

大部分人的直觉都是第二句话的价值比较大，也就是信息量比较大。因为，是否下雨是个不确定的事件。而第一句话，对于绝大多数人来说，毫无信息量，因为今天周一，明天必然周二。换句话说，也就是甲这句话没有消除任何不确定性，因此，信息量为零。

这里说了信息量，大家不觉得违和，好像信息能够度量。这个就是香农的天才之处，对于信息这样一个抽象的概念，提出了度量的方法。

香农指出，信息和长度、质量这些物理属性一样，是种可以测量的东西，信息的单位是比特(bit)。香农用“熵”来度量信息，所以有的时候信息也称为信息熵。

物理学中的熵

如果大家有一些热力学的知识，会知道熵这个概念来自物理中的热力学，在热力学中，熵是分子热运动“杂乱程度”的度量，在平衡状态下熵值最大，这时分子处于最无序状态。

热力学定律提出，宇宙中的熵有逐渐增大的趋势。如果从概率角度来看，是因为无序的状态总是远大于有序的状态。例如，一个玻璃杯子，从高空坠落硬质地面，绝大多数时候都要摔碎，因为杯子保持完整只有一个状态，而摔碎成什么样子则几乎有无限种状态(虽然从原子个数角度来看，状态是有限的，但是可以认为是无限种状态)。换句话说，虽然会摔碎，但是即使初始值完全一样，它们摔碎的结果(包括碎片个数、大小等)也不可能一样。

1824 年，法国工程师卡诺提出了卡诺定理，说明热机的最大热效率只和其高温热源和低温热源的温度有关。克劳修斯和开尔文在热力学第一定律建立以后重新审视了卡诺定理，意识到卡诺定理必须依据一个新的定理，即热力学第二定律。他们分别于 1850 年和 1851 年提出了克劳修斯表述和开尔文表述。克劳修斯的表述是这样的：不可能把热量从低温物体传向高温物体而不引起其他变化。开尔文的表述是这样的：不可能制成一种循环动作的热机，从单一热源取热，使之完全变为功而不引起其他变化。这两种表述在理念上是等价的。

1854 年，克劳修斯首先引进了熵的概念，这是表示封闭体系杂乱程度的一个量。热力学第二定律也可以表述为：孤立系统的熵永不自动减少，熵在可逆过程中不变，在不可逆过程中增加。1877 年，玻尔兹曼用下面的关系式来表示系统无序性的大小：$S \propto \ln\Omega$。1900 年，普朗克引进了比例系数 k，将上式写为 $S=k\ln\Omega$，k 为玻尔兹曼常量，S 是宏观系统熵值，是分子运动或排列混乱程度的衡量尺度，Ω 是可能的微观态数，Ω 越大，系统就越混乱无序。

热力学第二定律是一个非常奇特的定律，在物理上，只有它可以定义时间。因为它是唯一区分过去和未来的基本物理定律，熵增的方向即是时间前进的方向，其他的物理定律在时间上都是可逆的，文献[9]写道：为什么第二定律能区分过去和未来，而其他定律不能，也许这是物理学中最大的谜团。

一条信息到底有多大？其实就是对其不确定性的度量。假设某一事情 X 包含 N 种情况，每种情况以 x_i 表示，$p(x_i)$代表每种情况发生的概率，那么这件事情的信息（不确定性）可以用式(3.2)度量。

$$H(X)=-\sum_{1}^{N}p(x_i)\log_2 p(x_i) \tag{3.2}$$

信息熵的表示符号一般都用 H，一般认为 H 是 Heat(热量)这个单词的首字母，熵的英文是 entropy，也有些资料使用 I(Information)表示信息的度量。

为什么式(3.2)前面有一个莫名其妙的负号？因为概率都是小于或等于 1 的，以 2 为底进行对数运算的时候，结果是负数，因此前面有个负号，保证结果是个正数。

如何理解香农熵的定义？这里给出一个例子，假设你要猜硬币，那么正常情况下，正面向上的概率 $p_1=1/2$，反面向上的概率 $p_2=1/2$，此时的信息熵是 $-(p_1\times\log_2 p_1+p_2\times\log_2 p_2)=1$。

这时那个好心的菩萨又来了，她告诉你其实这枚硬币做过手脚，如果此时你猜硬币，你最好猜正面向上，因为这枚硬币正面向上的概率是 $p_1=0.9$，反面向上的概率 $p_2=0.1$。那么此时信息熵是 $-(p_1\times\log_2 p_1+p_2\times\log_2 p_2)=0.469$。

正常的猜硬币游戏，信息熵是 1 比特，这是因为不确定性最高，而菩萨过来，帮你消除了不确定性，信息熵也就从 1 比特降到了 0.469 比特（菩萨帮你消除了 0.531 比特不确定性）。

当然，如果更进一步，令硬币投掷结果肯定是正面向上，即 $p_1=1$，$p_2=0$，此时由式(3.2)计算可知，最后的信息熵是 0，也就是没有任何不确定性。当然，在解决实际问题的时候，肯定发生和肯定不发生都极为罕见。

2. 香农三大定律

1）香农第一定律

香农第一定律，又称无失真信源编码定律。无失真，即不损失信息。信源编码，即把数据用某种形式编码。这个定律给信息编码指明了一个方向。

有这样一个例子能够说明信息和编码的关系。小仲马的法文版 *La Dame aux Camélias*、林纾先生翻译的古文版《巴黎茶花女遗事》和王振孙先生翻译的白话文版《茶花女》这三本书，哪本的信息量更大一点？

答案应该是一样大，因为它们讲述了同样一个故事，换句话说，它们消除的不确定性是一样的。在香农提出信息论之前，人们总把信息和信息编码弄混淆。实际上，一个信息可能有多种编码，例如，可以把法文、古文和白话文视为是描写同一个故事的不同的编码。不管用什么方式编码，这段内容的信息量是固定的，假设这段内容的信息熵是 H，用比特来表示。编码的总长度是 L 比特，香农第一定律告诉我们 $L \geqslant H$，也就是无论怎么编码，最好的结果也是编码的比特数等于信息的比特数，在大部分情况下，L 都会大于 H，也就是编码大多数情况下都有冗余。

林纾版的《巴黎茶花女遗事》，从出版信息查到的字数有 5.3 万余字，上海译文出版社（1993 年版）的《茶花女》（王振孙翻译）标记的字数是 29.6 万字。很明显，白话文的编码冗余要远远超过文言文。

但是冗余不一定是坏事，编码有冗余证明其容错性比较好。白话文删除一些文字，很多时候依然能看懂，但是文言文删除一些字，可能对意思影响就很大了。另外，冗余的信息在传送的时候，可靠性要更好。“重要的事情说三遍”，就是一种冗余，虽然有冗余，但是可靠性提高了。冗余还有一个好处，对于人类来说，冗余的信息也更容易让人明白，所以，如果本书有些地方啰唆（有冗余），那么实际上是为了让读者更容易理解。杨振宁所诟病的那些看不懂的书，可以认为那些书使用了更精简的编码。

既然编码的比特数总要大于或等于信息的比特数，有没有办法让编码的比特数接近信息的比特数呢？其实，这样的事情能够做到，请见下面的例子。

编码一章介绍过八卦，也就是八个卦象，它们的编码如下。

乾（☰）、坎（☵）、艮（☶）、震（☳）、巽（☴）、离（☲）、坤（☷）、兑（☱）

假设阳爻是 1，阴爻是 0，换成对应的二进制如下。

111、010、100、001、110、101、000、011

这个是正常的编码，每个卦象的编码长度是 3 比特。

这样编码的假设是各个卦象出现的概率相等，用香农熵公式计算一下，信息熵也是 3 比特（$L=H$）。现在假设这八个卦象出现的概率不等，分别是 1/2，1/4，1/8，1/16，1/32，1/64，1/128，1/128，那么，计算可得熵为 1.984 比特，刚才的编码还能使用，但是每个卦象需要 3 比特（$L>H$），如果换成如下的编码（一种霍夫曼编码）：

0、10、110、1110、11110、111110、1111110、1111111

平均编码长度是 1.984 比特。虽然最后一个卦象“兑”的编码变为 1111111，很长，但是因为它出现的概率也很小，所以，平均的编码长度很小。在概率角度上，和香农熵是一致的（$L=H$）。

但是，这样编码的代价就是编码会很复杂，在这个例子中，这种编码就比正常八卦的编码要复杂得多。这种编码的实质就是使新的编码的概率分布与数据的概率分布一致，从而使新的编码单个符号所含的信息量达到最大。香农第一定律是一个存在性定律，并没有给出具体的编码方法。

这个定律有很多意义，它为信息压缩指明了方向。现在上网看到的图片其实都是经过压缩的，香农第一定律指出了无损压缩的极限。当然，如果有的时候达到了香农极限，仍然不能让编码变小，可以丢失一些信息。例如，有些图片的细节不太重要，那么可以把部分信

息丢失，这种压缩叫有损压缩，本章后面会介绍一种简单的图片有损压缩方法。

2）香农第二定律

香农第二定律，也叫有噪声信道编码定律。噪声，就是表示在信息传输过程中和信息无关的信号；信道，即传输信息的通道。

在无线信息传输中，使用的信道就是一定频段的电磁波。空间充满了电磁波，包括宇宙射线、各种电器设备、雷电、太阳黑子等都可能产生电磁波，这些都是噪声。通信系统中，一般把信号(S)和噪声(N)之间的比值称为信噪比。香农第二定律指出了一个信道所能传递的信息的上限。式(3.3)给出了这个信道容量上限。

$$C = B \times \log_2\left(1 + \frac{S}{N}\right) \tag{3.3}$$

式(3.3)中，B 代表信道宽度，也就是常说的带宽。假如带宽是 1000Hz，信噪比是63∶1，那么信道的容量就是 $1000 \times \log_2(1+63/1) = 8000$(b/s)。

可以看到，要想在单位时间内传递更多的信息，有两个方法，第一个方法是提高带宽，也就是式(3.3)中的 B。第二个方法是提高信噪比，也就是 S/N 一项。

> 理论上，5G 要比 4G 快得多，一个重要的原因就是 5G 比 4G 的带宽大很多(还有其他技术保证 5G 比 4G 快)。5G 提高带宽的一个重要手段就是提高频率，当然，频率提高，问题也来了，频率越高，波长越短，能够传输的距离也越短。例如大家收听的广播电台，使用的电磁波是长波，它的频率很低，所以能够传得很远，所以大家打开收音机就能听到很远地方发送的广播，而高频的电磁波无论是传输距离还是绕过障碍物的能力都非常有限，所以 5G 要建更多的基站。

信噪比的提升是比较困难的，因为噪声无处不在。通常情况下，有线上网都要比无线上网更快，原因当然有很多，其中一个重要原因就是有线上网的噪声更少，信噪比更高。一般来说，距离无线信号发射器(例如无线路由器)越近，通信效果越好，因为距离发射器越远，信噪比越低(空间中充满电磁波噪声)；而一般来说，有线传输随着距离变远，信噪比也会降低，不过相比无线信号要好很多，这也是为什么一般距离变远之后，有线传输要比无线传输效果好很多。

香农第二定律对于通信系统的指导意义非常强。在香农提出信息论之前，人们就已经实现了无线通信(马可尼在 20 世纪初就实现了跨大西洋无线通信)，人们也知道无线通信的用处非常大，但是一直不能够大规模应用，当时的人们总觉得是工艺或功率等问题，总期待把工艺做得更精巧，或者通过提高功率等方式来实现无线通信。这种没有理论指导去试的方法，当然很难成功。有了香农第二定律之后，人们才知道了努力的方向。

3）香农第三定律

香农第三定律，也叫保真度准则下的信源编码定律，这个名字比第一定律和第二定律复杂。香农第一定律给出了信息进行编码的极限值(即无损压缩的极限值)，香农第二定律定义了信道传输的极限，香农第三定律讲的是总能够找到一种行之有效的编码方法，让信息的传输率无限接近信道的容量而不出错。设 $R(D)$ 为一离散无记忆信源的信息率失真函数，并且有有限的失真测度 D，那么只要码长足够长，一定存在一种编码，使得编码的平均失真度小于 D。

这个定律同样非常重要，在实际中，任何信道都不是完美的。人们将信号从一个地方传递到另一个地方时，信号会不可避免地受到噪声干扰，信号 1 也许会变为 0，0 也许会变为 1。香农证明了，只要加入足够的冗余，任何信息都可以通过不完美信道传输。香农对这个冗余进行了量化，信息在通过非完美信道传输时，所需冗余量大约等于信息受到干扰的熵。

在香农提出信息论之前，人们一直受困于如何克服通信中的噪声。一个正常的想法就是提高信号发送能量，让信号的强度远远超过噪声，如果通信系统只有一个信号传输，这没问题，但是通信系统会有多条信号同时传输，这就好比一个屋子里好多人在说话，为了让别人听清，就提高自己的声音，结果大家都得提高声音——信号内卷了起来。

香农给出了一个洞见，克服噪声的正确办法，是增加信息的冗余度。

举一个简单的例子，假设传输的所有信号都是由以下四个碱基符号组成：A、C、G、T。数字传输系统只能传送 0、1 两个数据，因此，对这四个符号编码如下。

A：00　　C：01　　G：10　　T：11

如果想传输的信号是 ACG，那么实际传送的数据是 000110。但是因为噪声的存在，传输的数据变成了 000111，那么接收到的消息就变成了 ACT，信息没有正确传输。

香农给出的洞见是给编码增加一些冗余度。例如，在对符号进行二进制编码的时候，可以编码如下。

A：00000　　C：00111　　G：11100　　T：11011

这样，如果接收到的数据是 10000，也容易知道其实发送的数据是信号 A。

虽然香农在实践中并没有发现这样做的编码方式，也没有发现将压缩代码和防错代码结合起来的方法，但他证明了这样的编码一定存在。

信息论告诉我们，信息是不确定性的度量，更进一步，通过学习香农三大定律，可以知道熵衡量的其实是数据的最优压缩，也是衡量将数据存储到硬盘上需要的最少比特数，或者说通过有限带宽传输数据的最短时间。香农定律说明无论怎样努力，都无法超越香农熵这一根本限制。

早期的通信系统通过模拟信号传播数据（早期的电话），如今的通信系统几乎都数字化，这些系统从香农理论中受益良多。今天看到这个香农的结论，觉得是很自然的，但是在那个时代，并非如此，在当时，人们仍然在模拟技术上纷纷押注。

香农的理论对于通信系统非常重要，极大地避免了人们走弯路，这些定律告诉人们什么事情是不可能做到的。对于香农的信息论，可以这样类比：人们在了解了热力学定律之后，就不会试图去制作永动机。

3.3 线性代数

如果说微积分、概率还符合大家对数学的认知，还能找到一个应用场景或者和生活中的现象对应起来，那么线性代数经常让人一头雾水，莫名其妙。

这是因为，第一，线性代数“很简单”，这里的简单是指不需要太多其他的数学基础就可以学习线性代数。如果一个初中生来学习线性代数，不会有任何障碍，因为学习线性代数所需要的知识并未超过初中学习范围；第二，学完了不知道能做什么，线性代数的核心是向量和矩阵的各种运算，向量和矩阵就是一堆数的集合，对于这些数的集合的操作有什么用处？

而且，线性代数的名字也很古怪，线性代数的英文是 linear algebra。Algebra 一词即“代数”，最初来自阿拉伯语 al-jabr，al 为冠词，jabr 意为恢复或还原（解方程的时候，移项就是一种还原）。

大家很早就接触过代数，对这个词比较熟悉，那么什么是线性（linear）？线性这个词，生活中很常见，例如线性增长、线性组合。那么，到底什么是线性，什么又是非线性？

对于线性代数里面的线性，主要是要满足指线性映射，线性映射需要满足一些条件。假设在某维空间中有两个向量 $\boldsymbol{u}$ 和 $\boldsymbol{v}$，如果有映射 f 满足如下两个条件。

$$\text{可加性：}f(\boldsymbol{u}+\boldsymbol{v})=f(\boldsymbol{u})+f(\boldsymbol{v})$$

$$\text{齐次性：}f(\lambda\boldsymbol{u})=\lambda f(\boldsymbol{u})$$

就表示 f 是线性映射。

“线性”的概念在数学中其实也是混淆的，主要是因为线性代数还是一门较新的科学。按照以前大家学习的数学，线性大概就是类似 $y=kx+b$ 形式函数，在平面上的图形是一条直线。

事实上，在 $y=kx+b$ 中，如果 $b\neq0$ 的时候，它不满足齐次性，换句话说，按照线性映射的定义，这个方程不是线性的。

所以在描述“线性”一词的时候，需要知道它所在的场合。

在实际应用中，更关心可加性。说两种事物是线性关系，其实是说它们是不会互相影响的独立关系，而非线性则是会相互影响的关系，而正是这种相互影响，使得整体不再是简单地等于部分之和，而可能出现不同于“线性叠加”的增益或亏损。

举两个例子，第一个例子，如果人一只眼睛的视觉能力是 1 的话，那么两只眼睛加起来的视觉能力是多少？不是 2，而是 6～10 倍，也就是说它们之间不是 1+1 的线性关系。

第二个例子，工厂里有两个工人，甲每天能生产 100 个产品，乙每天能生产 120 个产品，那么甲、乙共同生产，一天能生产多少个产品？如果是线性组合的话，他们一天能生产 220 个，这里的线性组合就是你干你的，我干我的，互不影响；但是实际上，有人的地方就有江湖，两个人共同生产，就有很多问题，可能互相帮助，生产的数量就大于 220 个，也可能互相拆台，生产的数量就小于 220 个，这样，他们之间就是非线性的关系。

生活中线性的情景多还是非线性的情景多？肯定是非线性的情景多。事物在一起，不可避免地要互相产生作用。第 2 章最后的 More Is Different，就是因为事物组合在一起，它们之间是非线性的。

但是线性代数还是非常有用的，首先，如果事物之间产生的作用不多，可以在一定程度上，用线性关系替代非线性关系；其次，可以设计模型，利用线性关系之间的运算实现非线性关系。

人工智能科学中，线性代数是基本的工具。大家不用害怕，线性代数并不难（毕竟，初中基础即可学会）。

3.3.1　向量

先从向量的名字说起。

向量的英文是 vector，不同学科看待向量的方式不同。例如，物理学中把向量翻译成矢

量(也是译自 vector)。

数学中的向量更强调空间的概念，也就是向量是描述空间的工具。

在人工智能科学中，向量是这样一个含义：向量是一个有序的数组，是某组基(base)生成的空间中的点的坐标。这句话很重要，如果大家目前不能理解这句话，没关系，学完本书之后就会知道它的意义。基是线性空间一个基本的概念，例如，三维空间中的一组基，$x=(1,0,0)$、$y=(0,1,0)$和 $z=(0,0,1)$，三维空间中的点(3,4,5)可以表示该点在这组基中的坐标，其值是这组基的线性组合，$3x+4y+5z$。注意，三维空间中的基不止这一组，事实上，只要有三个三维向量是一组线性无关的向量，它们就组成了一组三维空间中的基，因此，有无穷组基(回忆一下，坐标变换就是从一组基变换到另外一组基)。

另外，在计算机领域中的一些资料默认以列向量描述数据。知道了这点能够解决阅读其他资料中的一些困惑，例如遇到这种写法，$(x_1,x_2,\cdots,x_n)^{\mathrm{T}}$ 或者 $(x_1,x_2,\cdots,x_n).\mathrm{T}$，这里的上标 T 或者.T 表示转置，表示有一个列向量，但是列向量写起来占空间，就像下面的 $\boldsymbol{a}$、$\boldsymbol{b}$、$\boldsymbol{c}$ 三个列向量：

$$\boldsymbol{a}=\begin{bmatrix}1\\2\\3\end{bmatrix},\quad \boldsymbol{b}=\begin{bmatrix}4\\5\\6\end{bmatrix},\quad \boldsymbol{c}=\begin{bmatrix}7\\8\\9\end{bmatrix}$$

如果想用一行来写，那么可以写成 $\boldsymbol{a}=(1,2,3)^{\mathrm{T}}$，$\boldsymbol{b}=(4,5,6)^{\mathrm{T}}$，$\boldsymbol{c}=(7,8,9)^{\mathrm{T}}$。

向量可以进行运算。计算这三个向量的线性组合 $\boldsymbol{d}=1\boldsymbol{a}+4\boldsymbol{b}-7\boldsymbol{c}$，得到结果 $\boldsymbol{d}=(-32,-34,-36)^{\mathrm{T}}$。

3.3.2 矩阵

矩阵是向量的组合，它几乎是线性代数的核心。在很多人眼中，矩阵比向量还莫名其妙。事实上，矩阵已经成为很多行业的重要工具，例如，海森堡即以矩阵描述量子力学，也称矩阵力学。工程中使用矩阵的例子更是数不胜数，机械、建筑、电器、管理等领域都用到了矩阵。

当然计算机用得更多。矩阵在计算机中的应用，至少有两个意义。

第一，矩阵用来描述数据。作为向量的组合，向量描述一个数据在某一个空间中的一个点(向量的值是该点的坐标)，那么矩阵就是这些点的集合。

第二，矩阵作为一种变换。矩阵的运算，可以视为是不同空间的转换。

这两个意义并不是独立的，可以互相包含。这两个意义都非常抽象。这里，试着简单说明一下，只有在大家学习了更多的内容之后，才能更深刻地理解这两个意义。

作为数据的描述：在第 2 章中，介绍过结构化数据和非结构化数据，结构化数据最典型的就是表格。表格数据天然就是矩阵，很方便使用矩阵进行描述，还有图像数据，也天然就是矩阵，这个在数据编码中已经见过了。

作为一种变换：矩阵有多种运算，矩阵的乘法即一种变换，在一个矩阵的帮助下，将一个空间中的数据转换为另一个空间中的数据，注意，这种转换是不可交换的，矩阵的乘法也是不可交换的。

关于矩阵的运算有很多，这里简单介绍其中两个比较重要的，第一是矩阵的特征值和特征向量，第二是矩阵的各种分解。