对数的运算法则

从指数定义推导对数运算

Namishu

这份讲义从指数形式出发定义对数，并推导乘法法则、除法法则、指数法则和换底公式。重点是理解对数如何把乘法、除法和次方转化为更容易处理的加法、减法和乘法。

目录

给定一个大于 0 的数字 n，可以把它写成指数形式。例如，16 可以写成 2^4，表示 4 个 2 相乘，其中 4 是“指数”，2 是底数。16 还可以表示成 4^2，那么 4 是底数，2 是指数。

用符号 a 表示底数，那么 n 可以写成下面的形式。

$$n = a^x \tag{1}$$

注意，a 可以取哪些数值？因为 n 是一个正数，那么 a^x 的结果也应该是正数。所以 a 也必须大于 0。但是这样似乎还不够，因为如果 $a = 1$ 的话，无论多少个 a 相乘，结果都等于 1。因此，a 不能等于 1。

根据 (1) 式，我们把 x 叫做 n 的对数，记作

$$x = \log_a n \tag{2}$$

例如 $n = 16$。令 $a = 2$，我们有 $16 = 2^4$，那么 $\log_2 16 = 4$；令 $a = 4$，我们有 $16 = 4^2$，那么 $\log_4 16 = 2$；令 $a = 16$，我们有 $16 = 16^1$，那么 $\log_{16} 16 = 1$。

对数的定义

如果

$$n = a^x$$

其中 $n > 0$，$a > 0$ 且 $a \neq 1$，那么 x 叫做 n 以 a 为底的对数，记作

$$x = \log_a n.$$

现在把公式 (2) 带入公式 (1)，得到下面的“恒等式”。

$$n = a^{\log_a n} \tag{3}$$

它的意思是，n 可以表示成 a 的“几次方”的形式，那个“几”就是 n 的“对数”。比如，100 可以写成 3 的几次方？它是 $3^{\log_3 100}$，它也是 $5^{\log_5 100}, a^{\log_a 100}, b^{\log_b 100}$ 等等。

对数恒等式

$$n = a^{\log_a n}$$

这个式子把一个正数 n 放回到以 a 为底的指数形式中。后面对数运算法则的推导，都会反复用到它。

回顾这三个式子 (1)–(3)，其实是从不同的角度，来描述同一个“东西”。“对数”是通过指数来定义的，指数又可以用对数来表示。既然说的是同一个意思，那为什么还要引入对数呢？

打个比方，我们站在果树下，伸手摘不到果子，但是搭一个梯子就可以。这些定义就是梯子。对数的意义，不是引入了一个新符号，而是提供了一种新的看问题的方式：同一个数，既可以放在指数的世界里看，也可以放在对数的世界里看。

我们来看一看，如何使用这些“梯子”。

1 计算“乘法”的对数

考虑这样的问题，计算两个数相乘的对数，例如 $\log_2(32 \times 64)$。

根据定义计算。先算 32×64，得到 2048。然后计算 $\log_2 2048$。用 2048 除以 2 等于 1024，再用 1024 除以 2 得到 512，依次类推。我们一共除以 2 十次，最终得到 2，这说明 $2048=2^{11}$。

现在换一个方法，不直接计算 32×64。利用上面的公式 (3)，把 32×64 看成“一个数”，用公式 (3) 写成这个样子：

$$32\times 64=2^{\log_2(32\times 64)} \tag{4}$$

类似地，把 32 和 64 分别写成 $2^{\log_2 32}$ 和 $2^{\log_2 64}$，再把它们相乘得到

$$32\times 64=2^{\log_2 32}\times 2^{\log_2 64}=2^{\log_2 32+\log_2 64} \tag{5}$$

注意这两个式子 (4) 和 (5)，都表示同一个正数 32×64。于是，下面的式子成立。

$$2^{\log_2(32\times 64)}=2^{\log_2 32+\log_2 64}$$

等式两边的底数相同且不等于 1，那么它们的指数也相同。即，

$$\log_2(32\times 64)=\log_2 32+\log_2 64$$

容易看出来，32 是 5 个 2 相乘，64 是 6 个 2 相乘，即 $\log_2 32=5$，$\log_2 64=6$。代入上面的式子，得到下面的计算结果。

$$\log_2(32\times 64)=\log_2 32+\log_2 64=5+6=11$$

这个方法不用直接计算 32×64，它把“相乘的对数”变成“对数的相加”。它避免了乘法，这样算起来更简单。

我们也可以用更一般的语言来描述这个方法。假设我们要计算两个数相乘的对数 $\log_a(m\times n)$，其中 $a>0, a\neq 1$，并且 $m>0, n>0$，那么结果等于 $\log_a m$ 和 $\log_a n$ 的和。即

乘法法则

$$\log_a(m\times n)=\log_a m+\log_a n$$

这个公式称为对数的乘法法则。

2 计算“除法”的对数

如果是计算 $\log_a(m/n)$，那么按照上面的方法，可以得到怎样的规律？

继续利用公式 (3)，把 m/n 以及 m, n 分别用“对数符号”写成“指数表达式”，即

$$\frac{m}{n}=a^{\log_a(m/n)} \tag{6}$$

另外

$$m=a^{\log_a m},\quad n=a^{\log_a n}$$

m 除以 n 得到

$$\frac{m}{n}=\frac{a^{\log_a m}}{a^{\log_a n}}=a^{\log_a m-\log_a n} \tag{7}$$

结合式子 (6) 和 (7) 得到

$$a^{\log_a(m/n)}=a^{\log_a m-\log_a n}$$

等式两边底数相同（并且底数不等于 1），那么它们的指数也相同，因此

除法法则

$$\log_a(m/n) = \log_a m - \log_a n$$

这个公式称为对数的除法法则。

3 换一个底数

考虑这样的例子：2 的 5 次方等于 3 的几次方？用方程来描述就是，已知 $2^5 = 3^x$，求解 x。

可以根据定义来计算。第一步，计算 $2^5 = 32$，那么原来的方程就是 $3^x = 32$；第二步，根据对数的定义，得到 $x = \log_3 32$。我们知道 $3^3 = 27, 3^4 = 81$，因为 32 在 27 和 81 之间，这说明 $\log_3 32$ 是一个 3 到 4 之间的小数。它具体是多少，我们暂时不用关心，知道它是一个常数就可以了。

再看一个类似的问题。

已知 $2^{100} = 3^x$，求解 x。按照上面的步骤，我们试着算一下。

第一步，计算 2^{100}。等一等，它是 100 个 2 相乘！我们试着算到 $2^{10} = 1024$，还剩下 90 个 2 相乘。如果这么乘下去，结果是一个巨大的数字。这样算起来太复杂了。

我们换一个思路。先分析这个问题。已知数字 2^{100}，要把它的底数从 2 增加到 3，但是不改变这个数字的大小。那么，它的指数应该是增加还是减少？

容易发现，应该是减小。如果指数不减小，那么它至少是 3^{100}，这个数比 2^{100} 要大。因此，$x < 100$。这说明虽然 2^{100} 是一个巨大的数字，但是方程的解 x 不是很大，因此是有希望算出来的。

我们想到了恒等式 $n = a^{\log_a n}$。利用它先把 2 写成以 3 为底的指数形式，即

$$2 = 3^{\log_3 2}$$

接着把它代入 2^{100}，我们得到

$$2^{100} = (3^{\log_3 2})^{100} = 3^{100 \times \log_3 2}$$

即

$$2^{100} = 3^{100 \times \log_3 2}$$

再把这个等式代入要解的方程 $2^{100} = 3^x$。新的方程如下。

$$3^{100 \times \log_3 2} = 3^x$$

注意等式两边的底数相同（并且不等于 1），所以指数也应该相同。

$$x = 100 \times \log_3 2$$

容易发现，$\log_3 2$ 是一个介于 0 和 1 之间的数，用计算器算一下 $\log_3 2 \approx 0.631$。因此，

$$x = 100 \times 0.631 = 63.1$$

这个结果说明，即使原来的指数非常大，通过换底数，也可以把问题转化为“常数乘以一个对数”的形式。

4 指数法则

我们把“换底数”的问题用符号来描述：已知一个正数 n 可以写成 $n = a^x$。现在想给它换一个底数 b，写成 $n = b^y$ 的形式，那么 y 应该等于多少？换句话说，已知 $a^x = b^y$，求解 y（此时把 x 看成常数）。注意：底数 $a, b > 0$，并且 $a, b \neq 1$。

接下来考虑两种办法求解。

第一种方法，根据定义直接求解。

$$y = \log_b a^x \tag{8}$$

第二种方法，利用恒等式 $n = a^{\log_a n}$，先把 a 写成以 b 为底的指数形式。

$$a = b^{\log_b a}$$

然后把它代入 a^x 得到

$$a^x = (b^{\log_b a})^x = b^{x \log_b a}$$

即

$$a^x = b^{x \log_b a}$$

把这个式子代入要解的方程 $a^x = b^y$，得到新的方程如下。

$$b^{x \log_b a} = b^y$$

等式两边的底数相同（并且不等于 1），那么指数也相同。因此，

$$y = x \log_b a \tag{9}$$

这是第二种方法的结果。

再回过头观察式子：(8) 是第一种方法的结果；(9) 是第二种方法的结果。区别只是形式不同。结合这两个式子，我们得到

指数法则

$$\log_b a^x = x \log_b a$$

这个公式叫做对数的“指数法则”。

再回头来看前面提到的例子：已知 $2^{100} = 3^x$，求解 x。

第一步，等式的两边同时取对数（以 3 为底）。

$$\log_3 2^{100} = \log_3 3^x$$

第二步，根据指数法则，把“指数”变成“乘数”。

$$100 \log_3 2 = x \log_3 3$$

容易发现 $\log_3 3 = 1$。因此，

$$x = 100 \times \log_3 2$$

5 对数的换底公式

已知道一个正数 n。它以 a 为底的对数是 $\log_a n$；以 b 为底的对数是 $\log_b n$。我们的问题是，$\log_a n$ 和 $\log_b n$ 之间怎么换算？

继续用前面的思路。先用恒等式 (3) 把 n 写成 $b^{\log_b n}$ 的形式

$$\log_a n = \log_a b^{\log_b n} \tag{10}$$

然后根据指数法则（把指数变成乘数），得到下面的等式

$$\log_a b^{\log_b n} = \log_b n \times \log_a b \tag{11}$$

把 (11) 代入 (10)，于是

$$\log_a n = \log_b n \times \log_a b$$

把这个等式的两边同时除以 $\log_a b$，可以解出 $\log_b n$，即

换底公式

$$\log_b n = \frac{\log_a n}{\log_a b}$$

这个公式称为对数的“换底公式”。

6 对数的作用

回顾前面的讨论，我们一直在两个“世界”之间来回切换。这两个世界之间的梯子，就是前面反复出现的恒等式

$$n = a^{\log_a n}$$

正是借助这个梯子，我们才能把同一个数，从指数的世界搬到对数的世界，再从对数的世界搬回来。对数的核心力量，正体现在这种“在不同世界之间切换”的能力上。

在指数世界中，运算往往表现为乘法和次方；而一旦进入对数世界，这些复杂的运算就会变成更容易处理的加法和乘法。例如：

- 原来的世界：两个数相乘 $m \times n$；对数世界：$\log_a m + \log_a n$，乘法变成了加法。
- 原来的世界：n 个 x 相乘得到 x^n；对数世界：$n \times \log_a x$，次方变成了乘法。

在指数的世界中，数值的变化往往非常快；而取了对数之后，相当于把原来的世界“压扁”了，使得数值的变化变得更加缓慢，也更容易比较和计算。

接下来，我们通过几个具体的例子，来看对数是如何在实际问题中发挥作用的。

6.1 解指数方程

> 解方程：$2^x = 1000$。

第一步，等式两边取对数。底数大于 0 不等于 1，例如以 10 为底。

$$\log_{10} 2^x = \log_{10} 1000$$

第二步，根据对数的指数法则，我们有

$$\log_{10} 2^x = x \log_{10} 2$$

从而得到“一元一次方程”

$$x \log_{10} 2 = \log_{10} 1000 = 3$$

第三步，两边同时除以常数 $\log_{10} 2$ 得到

$$x = \frac{3}{\log_{10} 2}$$

6.2 乘积的比较

> 已知 $a = 2^{50} \times 3^{30}$，$b = 4^{40}$。请问 a 和 b 哪个更大？

对数登场。对 a 和 b 分别取对数（为了方便计算，选 2 为底数）。

$$\begin{aligned}\log_2 a &= \log_2(2^{50} \times 3^{30}) \\ &= \log_2 2^{50} + \log_2 3^{30} \\ &= 50 \log_2 2 + 30 \log_2 3 \\ &= 50 + 30 \log_2 3\end{aligned}$$

容易发现 $\log_2 3 > 1$，因此 $\log_2 a > 50 + 30 = 80$。

$$\log_2 b = \log_2 4^{40} = 40 \log_2 4 = 40 \times 2 = 80$$

再从对数世界回到原来的世界。

$$\begin{aligned}a &= 2^{\log_2 a} > 2^{80} \\ b &= 2^{\log_2 b} = 2^{80}\end{aligned}$$

因此 $a > b$。

6.3 不等式证明

> 证明下面的不等式：
>
> $$\frac{a_1 + a_2 + \cdots + a_n}{n} \geq (a_1 a_2 \cdots a_n)^{1/n}$$
>
> 其中 $a_1, a_2, \ldots, a_n > 0$，$n > 0$ 是整数。

观察不等式右边，是多个数相乘，我们想到了对数。于是，考察它的对数形式

$$\log_b (a_1 a_2 \cdots a_n)^{1/n}$$

其中 $b > 1$。下面我们把 $\log_b$ 简写为 $\log$。

根据对数的指数法则和乘法法则，我们得到

$$\begin{aligned}\log(a_1 a_2 \cdots a_n)^{1/n} &= \frac{1}{n} \log(a_1 a_2 \cdots a_n) \\ &= \frac{1}{n}(\log a_1 + \log a_2 + \cdots + \log a_n)\end{aligned} \tag{12}$$

因为底数 $b>1$，因此 $\log x$ 是一个凹函数，它满足 Jensen 不等式。说明：如果你现在不了解凹函数或者 Jensen 不等式，先不用伤脑筋。那就直接忽略这一句话，暂且认为下面这个不等式是成立的。在更系统的数学学习中，这个不等式可以被严格证明。

于是，我们得到

$$\frac{1}{n}(\log a_1+\log a_2+\cdots+\log a_n)\le\log\left(\frac{a_1+a_2+\cdots+a_n}{n}\right) \tag{13}$$

结合 (12) 和 (13)，我们得到

$$\log(a_1a_2\cdots a_n)^{1/n}\le\log\left(\frac{a_1+a_2+\cdots+a_n}{n}\right)$$

最后再指数化，回到原来的世界

$$(a_1a_2\cdots a_n)^{1/n}\le\frac{a_1+a_2+\cdots+a_n}{n}$$

7 小结

对数并不是为了引入一个新符号，而是为了提供一种新的视角：把“乘法、除法、次方”这类指数世界里的运算，转化成对数世界里更容易处理的“加法、减法、乘法”。

对数的运算规律（$a>0,\ a\neq 1;\ m>0,\ n>0$）

$$\log_a(mn)=\log_a m+\log_a n$$

$$\log_a(m/n)=\log_a m-\log_a n$$

$$\log_a(m^x)=x\log_a m$$

$$\log_b n=\frac{\log_a n}{\log_a b}$$

这些规律背后反复出现的关键“梯子”，来自对数的定义所给出的恒等式

$$n=a^{\log_a n}.$$

它让我们可以把任意正数先“搬进”以 a 为底的指数形式（例如 $m=a^{\log_a m}$、$n=a^{\log_a n}$），再利用“同底指数相等 $\Rightarrow$ 指数相等”（底数 $a\neq 1$）去比较指数，从而把乘法/除法/次方一步步推导成加法/减法/乘法。

对数的典型应用场景包括：求解指数方程（两边取对数把未知指数变成乘数）、比较巨大乘积或幂的大小（先比较对数）、以及在数据处理中做“对数变换”（把数量级跨度很大的数据压缩尺度、把乘法模型线性化）。在科学与工程里，分贝（dB）、pH、地震震级等也都在用对数来表达“相对变化”与“数量级”。