

信源(source)是产生信息的实体。在物联网无线节点中,传感器是一种信源。物联网无线节点使用无线通信技术,把信源产生的信息传送到一个或多个目的地。

传感器种类繁多,其输出既有数字信号,也有模拟信号。如图 5.1 所示,对于传感器输出的模拟信号,需要先经过数字化过程转换成数字信号,以便于后续的处理与传输。为了抑制信号中的某些频率分量,可以对信号进行进一步处理,例如滤波。在发送端传输数字信号之前,还可以对数字信号进行数据压缩编码,以减小传输的数据量。在接收端进行相反过程,即解码,也可对信号进行滤波等处理。

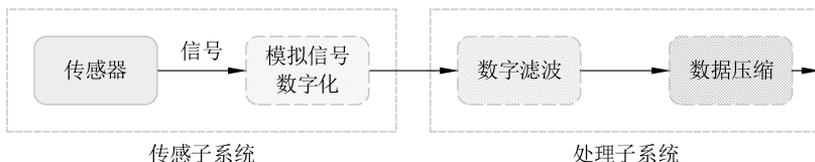


图 5.1 传感器信号与处理

本章首先讲述传感器的分类与特性,其次讲述信号的概念、模拟信号数字化过程以及信源编码方法,最后讲述频域与滤波的概念,并通过两个仿真实验加深读者对采样频率、频域、噪声、滤波的理解。

5.1 传感器

物联网中使用的无线节点,从供电角度看,大致可以分为无源节点和有源节点两类。无源节点多基于 RFID 技术,由读写器天线发射的磁场或电磁场通过无线节点(标签)的天线来为节点提供工作时所需的能量。有些无源节点上也配有传感器和处理器。有源节点多使用无线传感器网络等技术,通常具备传感器、处理器、无线收发器等部件,通过自带电池等方式供电,可以支持更远距离的无线数据传输。不论物联网无线节点使用 RFID 技术,还是使用无线传感器网络等技术,都是使用无线通信技术进行无线数据传输,只不过侧重点与技术复杂度略有不同。

不论是无源节点还是有源节点,在采集客观世界的的数据时,都离不开传感器。本书3.1节从硬件与应用的角度对传感器做了初步讲解,本节将进一步从分类与特性指标的角度讲解传感器。由于节点自身的特点,物联网无线节点中倾向于使用能耗较低、尺寸不大、数据传输率不高的传感器。

5.1.1 传感器的分类

可被测量的客观世界变量多种多样,相应地,传感器也种类繁多。从不同的角度,传感器可被划分为不同的类别。从被测量的角度,传感器可分为物理量传感器、化学量传感器以及生物量传感器。物理量传感器的分类趋于成熟,化学量传感器和生物量传感器的新产品不断涌现。国家标准 GB/T 36378.1—2018 中给出了物理量传感器的分类,如表 5.1 所示。化学量传感器和生物量传感器包括气体、氧气、湿度、结露、露点、水分、离子、氢离子浓度、pH 值、DNA、葡萄糖、尿素、胆固醇、血脂、谷丙转氨酶、血型、生化需氧量、谷氨酸、血气、血液 pH、血氧、血液二氧化碳、血液电解质、血钾、血钠、血氯、血钙、血压、食道压力、膀胱压力、胃肠内压、颅内压、心音、体温、皮温、血流、呼吸、呼吸流量、呼吸频率、细胞膜电压、细胞膜电容等传感器。

表 5.1 物理量传感器的分类

被测量类别	具体被测量	被测量详细划分
力学量传感器	压力传感器	绝压传感器
		表压传感器
		差压传感器
	重力传感器	
	张力传感器	
	应力传感器	
	应变传感器	
	力矩传感器	扭矩传感器
		转矩传感器
	位移传感器	线位移传感器
		角位移传感器
	速度传感器	线速度传感器
		角速度传感器
		转速传感器
		流速传感器
	加速度传感器	线加速度传感器
角加速度传感器		
振动传感器		
冲击传感器		

续表

被测量类别	具体被测量	被测量详细划分
力学量传感器	流量传感器	质量流量传感器
		体积流量传感器
	位置传感器	物位传感器
		姿态传感器
	尺度传感器	测距传感器
		厚度传感器
		角度传感器
		倾角传感器
		表面粗糙传感器
	密度传感器	
黏度传感器		
硬度传感器		
热学量传感器	温度传感器	
	热流传感器	
	热导率传感器	
	热扩散率传感器	
光学量传感器	激光传感器	
	可见光传感器	
	红外光传感器	
	紫外光传感器	
	照度传感器	
	亮度传感器	
	色度传感器	
	光谱传感器	
	图像传感器	
	能见度传感器	
	浊度传感器	
磁学量传感器	磁场传感器	
	磁通量传感器	

续表

被测量类别	具体被测量	被测量详细划分
电学量传感器	电流传感器	
	电压传感器	
	电场传感器	
声学量传感器	超声波传感器	
	声压传感器	
	噪声传感器	
微波传感器		
射线传感器	X射线感光器	
	α 射线传感器	
	β 射线感光器	
	γ 射线传感器	
	射线剂量传感器	

从转换原理的角度,传感器可分为物理传感器(包括电阻式、电容式、电感式、压电式、磁电式、热电式、光电式、谐振式、声波式、辐射式等)、化学传感器和生物传感器三大类。从是否需要外部电源的角度,可分为无源传感器(例如热电偶、电磁麦克风、压电传感器、光电二极管等)和有源传感器(例如碳麦克风、热敏电阻、应变计、电容式和电感式传感器等)。从绝对与相对测量的角度,可分为绝对传感器(例如热敏电阻测量绝对温度)和相对传感器(例如热电偶测量温度差)。从输出信号类型的角度,可分为模拟传感器(输出为模拟信号)和数字传感器(输出为数字信号)。

5.1.2 传感器的基本指标

理想的传感器应具有以下特点。

- (1) 仅对被测量敏感。
- (2) 输出与输入之间具有唯一的、稳定的对应关系。
- (3) 输出可实时反映输入的变化。

实际中,传感器本身结构、电子电路器件、电路系统结构以及各种环境因素都可能会影响传感器的整体性能。

如果传感器的输入与输出之间存在某种对应关系,且对应关系不随时间发生变化,则可用传递函数来描述这个对应关系: $y=f(x)$ 。其中, x 为传感器的输入, y 为传感器的输出。大多数传感器具有线性传递函数:

$$y = a_0 + a_1 x \quad (5-1)$$

式中, a_0, a_1 为实数。线性传递函数的斜率 a_1 被称为传感器的灵敏度(sensitivity)。对于一些复杂的传感器,其非线性的传递函数不易求得,需做近似计算,例如用代数多项式

逼近,即

$$y = a_0 + a_1x + a_2x^2 + a_3x^3 + \cdots + a_nx^n \quad (5-2)$$

式中, a_0, a_1, \cdots, a_n 为实数。在许多情况下, n 可以取 2 或 3。也可使用线性分段逼近做近似计算,即把非线性传递函数分解为多个部分,每个部分视为线性。在 $n \geq 3$ 时,可使用样条插值做近似计算。对于非线性传递函数,灵敏度定义为传递函数的一阶导数。当传感器的输出取决于多个输入时,传递函数可能是多个变量的函数。例如湿度传感器,其输出取决于相对湿度和温度两个输入变量。

不会引起不可接受的较大误差的传感器输入 x 的范围称为满量程(Full Scale, FS)。在最大输入和最小输入下测得的传感器输出 y 之间的代数差,称为满量程输出(Full-Scale Output, FSO)。传感器在测量范围内可以检测到的输入的最小变化量,称为分辨力(resolution)。分辨力可以用最小变化量的典型值、平均值或最大值来表示。能使传感器的输出产生可测变化量的最小输入值,即零点附近的分辨力,被称为阈值(threshold)。

传感器的一个重要技术指标是准确度(accuracy),这实际上意味着不准确度。不准确度是指传感器的输出值与真实值之间的最大偏差。不准确度可用多种方式衡量,例如绝对误差和相对误差。绝对误差为测量值与真实值的差。当误差独立于输入信号振幅时,常用此形式。相对误差为绝对误差与真实值的比。精度(precision)是指在相同的条件下反复测量相同的输入时,传感器给出相同输出的能力。精度描述这些测量读数之间的一致性,而这些读数不一定接近真实值。精度是准确度的必要条件,但不是充分条件。

实际中,真实值具有不确定性,因为无法完全确定真实值是多少。每个传感器都不是理想的,其性能也存在不确定性。无论单次测量的准确度有多高,即测量值与真实值有多接近,都无法保证测量值确实是准确的。由于实际传感器无法实现理想的传递函数,因此会产生一些类型的偏差,这些偏差会影响传感器的准确度,具体偏差如下。

(1) 滞后(hysteresis)误差。滞后是指传感器的输出因其输入是在增大过程中还是减小过程中达到同一输入值而有所不同。由此产生的传感器输出偏差称为滞后误差。产生滞后现象的主要原因是传感器机械部分存在不可避免的缺陷,包括材料结构变化等。

(2) 非线性误差。传感器的实际传递函数与理想的线性传递函数之间存在偏差,由此产生的传感器输出偏差称为非线性误差。

(3) 灵敏度误差。实际传感器的灵敏度可能与理想值有所不同,由此产生的传感器输出偏差称为灵敏度误差。

(4) 偏移量误差。如果传感器的输出值与真实值相差一个常数,则传感器存在偏移量误差。

(5) 动态误差。由被测量随时间的快速变化而引起的传感器输出偏差称为动态误差。

此外,传感器输出还会受到噪声等随机误差的影响。传感器也会受到漂移(drift)的影响。传感器的输出发生与被测量无关的、不需要的缓慢变化,称为漂移。它通常与传感器材料的老化有关,即材料某些方面的性能发生不可逆的变化。环境温度的变化也可能引起漂移。漂移表现为传感器的灵敏度、零点等特性随时间或温度发生变化。

当传感器的输入随时间快速变化时,传感器通常不能始终立即做出响应。传感器的这种与时间有关的特性,被称为动态特性。在控制系统理论中,通常通过常系数线性微分方程来研究传感器的动态特性。

5.2 信号

传感器输出的模拟电压信号、脉宽调制(Pulse-Width Modulation, PWM)信号、串行数字信号都是信号。

信号是信息的一种物理表示方式,是数据的传输载体。信号按物理属性分为电信号和非电信号。本书中提及的信号指的是电信号。电信号的基本形式是随时间变化的电压或电流。信号可以分为确定信号、随机信号、连续信号、离散信号、模拟信号、数字信号、周期信号、非周期信号等。在数学上,信号可表示为一个或多个自变量的函数。

在任意给定时刻 t , 都有确定的函数值 $f(t)$, 这样的信号称为确定信号。反之, 在任意给定时刻 t 都具有随机值的信号, 称为随机信号。

在连续的时间范围内有定义的信号, 称为连续信号。这里的连续是指函数的定义域(时间变量)是连续的, 其值域可以连续也可以不连续。若仅在一些离散的时间点才有定义的信号, 则称为离散信号或序列。离散点的时间间隔 $T_n = t_{n+1} - t_n$ 可以相等也可以不相等, 在通信中通常取等间隔, 即 $T_n = T, n \in \mathbb{Z}$ 。离散信号可表示为 $f(nT)$, 简写为 $f(n)$, 其中 n 是序号, $n \in \mathbb{Z}$ 。

函数值是连续值的连续信号称为模拟信号。函数值是连续值的离散信号称为采样信号。函数值为离散值的离散信号称为数字信号。

定义在 $(-\infty, \infty)$ 区间, 每隔一定时间 T (或整数 N), 按相同规律重复变化的信号, 称为周期信号。不具有周期性的信号称为非周期信号。对于连续的周期信号, 满足式(5-3)的最小 T 称为该连续周期信号的周期。

$$f(t) = f(t + mT) \quad m = 0, \pm 1, \pm 2 \dots \quad (5-3)$$

例如, 正弦信号 $f(t) = A \sin(2\pi ft + \varphi)$ 的周期为 $1/f$, f 为其频率, A 为振幅, φ 为初始相位。同样, 对于离散的周期信号, 满足式(5-4)的最小整数 N 称为该离散周期信号的周期, 即

$$f(n) = f(n + mN) \quad m = 0, \pm 1, \pm 2 \dots \quad (5-4)$$

例如, 正弦序列 $f(n) = \sin(\pi n)$ 的周期为 2。

由法国数学家、物理学家傅里叶(Fourier)发起的研究发现, 任何周期函数都可以用由正弦函数和余弦函数构成的无穷级数来表示。周期为 $1/f$ 的周期信号 $f(t)$ 的傅里叶级数分解为

$$f(t) = \frac{c}{2} + \sum_{n=1}^{\infty} a_n \cos(2\pi nft) + \sum_{n=1}^{\infty} b_n \sin(2\pi nft) \quad (5-5)$$

式中 a_n, b_n ——傅里叶系数;

$c/2$ ——周期信号的直流分量。

例如, 周期为 $1/f$ 的方波(即在每个周期的前半周期内函数值为 1, 后半周期内

函数值为-1)的傅里叶级数为

$$f(t) = \frac{4}{\pi} \sin(2\pi ft) + \frac{4}{3\pi} \sin(6\pi ft) + \frac{4}{5\pi} \sin(10\pi ft) + \dots \quad (5-6)$$

式(5-5)表明,构造任意周期信号需要无限多个正弦函数和余弦函数叠加,这些函数(即谐波)的频率随 n 的增加而增高,并且都是基频 f 的倍数。对于通信而言,若要在发送端和接收端之间传输任意的周期信号,则需要在信道中传输无限多个频率任意高的正弦波信号,这就要求信道的带宽无限大。而实际上,任何介质,包括空气、电缆以及发射机的带宽都是有限的,无法支持传输无限多个频率任意高的正弦波信号。这也是为何数字信号(方波)不能直接通过无线信道传输的原因。

因此在实际中,人们仅使用有限个正弦函数和余弦函数来构造信道中传输的周期信号(也是模拟信号)。可以将传输信号视为一个正弦波信号或多个不同频率正弦波信号的叠加。

5.3 模拟信号数字化

由于物联网无线节点的处理子系统一般使用处理器处理数据,因此传感器输出的模拟信号须先经过 ADC 转换为数字信号后再输入处理器。从数学上看,模拟信号转换为数字信号的过程有 3 个步骤,即采样、量化和编码,如图 5.2 所示。

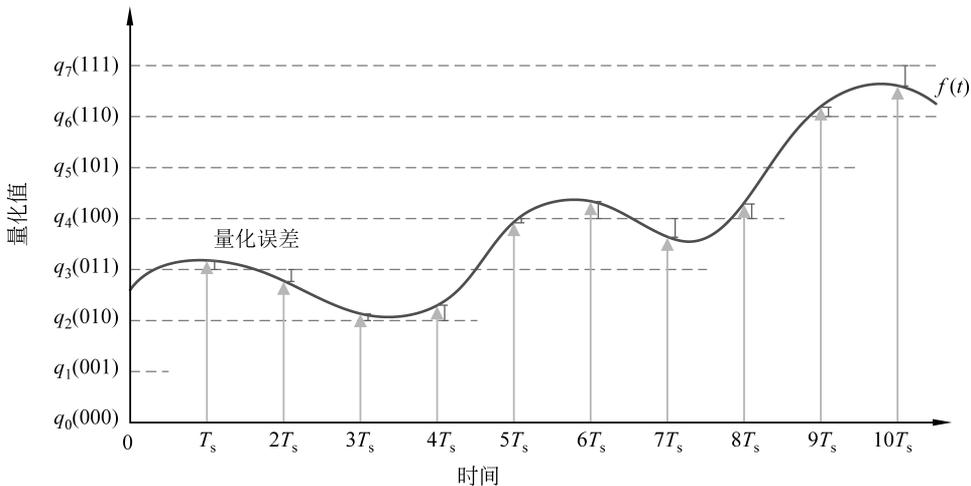


图 5.2 模拟信号数字化

(1) 采样。模拟信号被采样,成为采样信号。采样是指在一系列离散的时刻,对模拟信号采集其样值,得到一系列模拟信号在这些时刻的函数值。在通信中,通常采样的时间间隔相等,这个时间间隔记作 T_s 。近些年发展起来的压缩感知(compressed sensing)技术支持不等时间间隔采样。

通过采样所得到的模拟信号在一系列离散时刻上的采样值,显然与输入的被采样模拟信号不一样,只包含该模拟信号波形中的一部分值。那么,这是否意味着仅通过采样后

得到的一系列离散时刻上的采样值,无法准确恢复出被采样的模拟信号?

实际上,在采样时间间隔 T_s 满足一定条件时,可以由采样后得到的采样值准确恢复出被采样的模拟信号。奈奎斯特-香农采样定理(Nyquist-Shannon sampling theorem)指出:若模拟信号 $f(t)$ 的最高频率不超过 f_H 赫兹,则当以采样时间间隔 $T_s \leq 1/2f_H$ (即采样速率 $f_s \geq 2f_H$) 对其采样时,模拟信号 $f(t)$ 将被这些采样值完全确定。这个最低的采样频率 $2f_H$ 被称奈奎斯特速率,与此对应最大的采样时间间隔 $1/2f_H$ 被称为奈奎斯特间隔。在满足这个条件时,可以用一个截止频率为 f_H 的理想低通滤波器(也称重建滤波器)从采样信号中恢复出被采样的模拟信号。由于理想低通滤波器不可实现,实际的采样频率通常比奈奎斯特速率大一些,甚至数倍于奈奎斯特速率。

(2) 量化。采样值仍是一个取值连续的变量,即它可以有无数个可能的连续取值。如果用 N 个二进制位来表示采样值的大小,则最多可表示 $M=2^N$ 个不同的数值。为了能够用这 N 个二进制位来表示所有的采样值,把采样值的取值范围划分成 M 个不重叠的区间,将落在每个区间内的采样值用同一个数值 q_i 来表示, $i=0,1,\dots,M-1$ 。这个过程称为量化,这些区间称为量化区间。量化有多种方式,包括均匀量化(即均匀划分每个量化区间)和非均匀量化(即每个量化区间的划分并不均匀)、标量量化(即每次量化一个采样值)和矢量量化(即每次量化多个采样值)等。常见的标量均匀量化器包括中线(mid-tread)量化器和中升(mid-riser)量化器,前者的输入输出关系式为

$$Q(f(kT_s)) = \left\lfloor \frac{f(kT_s)}{\Delta} + \frac{1}{2} \right\rfloor \cdot \Delta = i \cdot \Delta \quad (5-7)$$

式中 $f(kT_s)$ ——模拟信号 $f(t)$ 在第 k 个采样时刻的采样值, $k \in \mathbb{Z}$, T_s 为采样时间间隔;

i —— M 个量化区间的索引, $i=0,1,\dots,M-1$;

Δ ——均匀量化的量化步长,也称为量化间隔;

$\lfloor \cdot \rfloor$ ——向下取整, $\lfloor x+1/2 \rfloor$ 表示对 x 四舍五入。

该式表明,采样值被与其距离最近的量化步长的整数倍(即量化值)所替代,如图 5.2 所示。从图 5.2 可以看出,采样值与其量化值之间通常存在一定的差异,这个差被称为量化误差。显然,对于同一输入信号,量化区间越多,即 M 越大,量化误差越小。

(3) 编码。即把量化后的每个量化值都映射为一个二进制数值。在图 5.2 中,采样信号经过中线量化器量化后,量化值 $i \cdot \Delta$ 被编码为量化区间索引 i 的二进制数值,这种码被称为自然二进制码。上述将模拟信号转换为二进制数字信号的方法称为脉冲编码调制(Pulse-Code Modulation, PCM)。PCM 通常在 ADC 上实现。

至此,模拟信号被转换为数字信号。

5.4 信源编码

在物联网无线节点中,减少无线传输的数据量有助于降低节点与网络的能耗(见 3.2 节)。如能减少传感器产生的数据量,则可减少无线节点通过无线传输的数据量。因此,可以对 5.3 节中模数转换后的数字信号做进一步编码(即数据压缩),以减少冗余。数据

压缩是减少数据量的过程。当然,编码与解码过程也会给处理器带来一些运算量,从而增加物联网无线节点的能耗,是否做数据压缩是一个时空复杂度与能耗的折中考虑。

信源编码,是将信源产生的消息编码为二进制码字的过程。信源是产生信息的实体,在物联网无线节点中,传感器是一种信源。消息是信息的具体体现形式,信息是消息中所包含的有效内容。信源编码有两个目的:①将任意信源消息唯一地映射为一个二进制码字,包括将信源产生的模拟信号转换成数字信号(见 5.3 节);②有效地将信源消息映射为最紧凑的二进制码字,即数据压缩。与信源编码对应的相反过程,称为信源解码。信源编码与信源解码之间的部分,即编码后数据的传送通道,称为编码信道,如图 5.3 所示。其中,信宿(sink)是接收信息的实体。

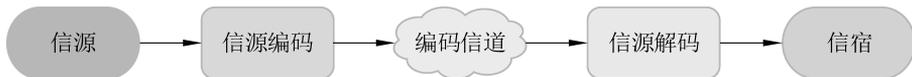


图 5.3 编码信道

先考虑一个数学问题。假如甲同学每次从 1、2、3、4 四个数字中随机抽出一个,且每个数字每次被抽到的概率都相等(即每个数字被抽到的概率都是 0.25)。乙同学不知道甲同学每次抽到的数字,只能通过向甲同学提问的方式获取有关数字大小的线索,而甲同学对乙同学的提问只能回答“是”或者“否”。那么,乙同学为了完全确定甲同学每次抽出的数字,需要针对每次抽出的数字平均至少提问几次?

显然,乙同学在提问时可以把这四个数字平均分成两组,从甲同学回答的“是”或“否”中,获知甲同学抽到的数字在哪一组中,然后再向甲同学提问这组两个数字中的一个是否为甲同学抽到的数字,从而总是可以通过两次提问完全确定甲同学抽到的数字。

如果这四个数字被甲同学抽到的概率不完全相等,例如 1 被抽到的概率为 0.5、2 被抽到的概率为 0.25、3 和 4 被抽到的概率为 0.125,那么在这种情况下,乙同学需要平均至少提问几次,才能完全确定甲同学每次抽出的数字?

由于 1 被抽到的概率最大,如果乙同学第一次提问的问题是“抽到的数字是否为 1”,则乙同学有 0.5 的概率提问一次就可确定这个数字为 1;如果抽到的数字不是 1,乙同学的第二个问题可以问是否为剩下的数字中被抽到概率最大的数字,即“抽到的数字是否为 2”,这时乙同学有 $0.25/(0.25+0.125+0.125)=0.5$ 的概率可确定这个数字为 2;如果经过两次提问都无法确定这个数字,那么通过第三次提问就可以确定这个数字具体为余下两个数字中的哪一个。采用这种方法提问,乙同学需要多至 3 次提问才能完全确定甲同学抽出的数字。但是,平均下来只需要 $1 \times 0.5 + 2 \times 0.25 + 3 \times 0.25 = 1.75$ 次提问,比之前等概率抽取情况下的平均提问次数还要少。

这个最少的平均提问次数,即为信息熵,常称作熵(entropy),用来衡量随机变量可能取值(抽出的数字)的不确定性。熵也可以解释为信源产生信息的平均速率。一般地,如果随机变量 S 的取值为 s_i 的概率是 $p(s_i)$,则 S 的熵 $H(S)$ 为

$$H(S) = - \sum_{i=1}^m p(s_i) \log(p(s_i)) \quad (5-8)$$

其中, S 的可能取值为 s_1, s_2, \dots, s_m 。如果式中对数的底取为 2,则熵 $H(S)$ 的单位为比

特(b)。

【例 5.1】 某传感器输出的模拟信号经分辨率为 3b 的 ADC 转换成数字信号。ADC 输出二进制数值“000”“001”“010”“011”“100”“101”“110”“111”的概率分别是 0.01、0.04、0.15、0.3、0.3、0.15、0.04、0.01。求该随机变量的熵。

【解】 根据式(5-8)计算其熵 $H(S)$ 为

$$\begin{aligned} H(S) &= - \sum_{i=1}^m p(s_i) \log_2(p(s_i)) \\ &= -(2 \times 0.01 \times \log_2 0.01 + 2 \times 0.04 \times \log_2 0.04 \\ &\quad + 2 \times 0.15 \times \log_2 0.15 + 2 \times 0.3 \times \log_2 0.3) \text{b} \\ &\approx 2.37 \text{b} \end{aligned}$$

例 5.1 中 ADC 的输出存在平均 $3\text{b} - 2.37\text{b} = 0.63\text{b}$ 的冗余。可以对传感器或 ADC 输出的数据进行数据压缩编码以减少其中的冗余。数据压缩分为有损压缩和无损压缩,前者在压缩过程中会丢弃掉一些细节信息,常用于压缩语音、图像、音视频等数据。对于传感器数据,通常人们不希望对其压缩时丢失任何信息,因此这里使用无损压缩方法。数据压缩方法的选择需要权衡压缩程度、压缩和解压缩所需的运算量等因素。无损数据压缩方法包括赫夫曼编码(Huffman coding)、行程编码(Run-Length Encoding, RLE)、算术编码(arithmetic coding)、非对称数字系统(Asymmetric Numeral Systems, ANS)等。

编码本质上是一种映射过程,即把输入的一组二进制比特(被称为符号),映射为输出的另一组二进制比特(即码字)。赫夫曼编码是一种变长码,即每个码字的长度并不完全相等。虽然赫夫曼编码并不总是最优的无损数据压缩方法,但其可以被有效实现。

赫夫曼编码的基本思想:把输入中出现概率较低的符号用较长的码字来表示,把输入中出现概率较高的符号用较短的码字来表示,从而减小输出码字的平均长度。编码的主要工作是找出输入符号与输出码字之间的最佳映射关系。

赫夫曼编码的基本做法:创建一棵二叉树,其各个叶节点为编码器输入中可能出现的符号,并且出现概率越小的符号所在的叶节点越可能位于二叉树的更深层。从二叉树根节点出发到某个符号所在的叶节点所经过的路径即为该符号编码后的输出。创建二叉树的算法如下,其中算法的输入为符号 s_i 的出现概率 $p(s_i)$, $i = 1, 2, \dots, m$, 输出为创建的二叉树。

(1) 创建 m 棵二叉树的集合 $A = \{T_1, T_2, \dots, T_m\}$ 。其中每棵二叉树 T_i 只有一个带有概率值 $p(s_i)$ 的根节点,其左、右子树均为空。

(2) 在 A 中选取根节点所带概率值最小的两棵树,分别作为左子树和右子树构造一棵新的二叉树。新二叉树根节点所带的概率值为其左、右两棵子树根节点所带的概率值之和。在 A 中删除选取的这两棵树,并将新的二叉树加入到 A 中。

(3) 重复上一步,直到 A 中只剩下一棵树为止。这棵树就是该算法创建的二叉树。

如果不知道符号 s_i 的出现概率 $p(s_i)$,可以使用自适应赫夫曼编码(adaptive Huffman coding)、自适应算术编码(adaptive arithmetic coding)等方法。