DIFFERENT METHODS OF INFORMATION ENCODING

Abdujabborov Madaminjon Voxidjon o'g'li
Teacher at the Department of Information Technologies, ASU

Otabayeva Sevarakhon
Teacher at the Department of Information Technologies, ASU

Abstract

This article discusses signal encoding, which can be achieved using various methods. It also provides information on both uniform and non-uniform encoding. Uniform encoding involves coding based on a single symbol such as "0" or "1". Non-uniform encoding, on the other hand, utilizes different representations within hardware or digital systems. This approach allows codes to be expressed in multiple forms, such as binary codes.

Keywords: Information, coding, analysis, signals, computers, bytes, data analysis, automated analysis, specialized tools, system duplication methods.

Introduction

A person receives information in an analog form through natural sensory organs. However, the reception of information in a discrete form, represented as a set of symbols, is considered universal. Specifically, the information inputted into a computer is processed and transmitted in the same discrete form through communication means.

Signal encoding involves representing symbols in convenient forms for future use. During encoding, a set of characters from one type is mapped to a set of characters from another type. In this context, the set of characters to be represented is known as the initial alphabet, while the set of characters used for representation is referred to as the encoding alphabet or alphabet code. For instance, consider a table that maps decimal numbers to binary and hexadecimal numbers. In this case, the decimal numbers serve as the initial alphabet, while 0 and 1 represent the binary encoding alphabet, and 0 to 9, along with characters A to F, form the hexadecimal coding alphabet.

A coding combination or code refers to a set of characters utilized to represent a single character or a combination of characters from the original alphabet using characters from the coding alphabet. For instance, if we assume that 8 is represented as 1000 in binary, then the combination of 8 and 1000 is referred to as a code combination. Additionally, the term "8" is used to represent the initial character. A collection of code combinations is called a code. The reverse process, which involves determining the initial symbol from a combination of codes, is known as decoding. A proper implementation of decoding must ensure unambiguous mapping, meaning that only one code corresponds to one initial character, and vice versa.

Information is transmitted through various communication devices using different methods. Examples of such devices include the telegraph, Morse code, radio, television, books, and more.

In Morse code, information is encoded and transmitted using short and long signals. In newspapers and magazines, information is encoded and transmitted using letters and symbols.

In computers, information is encoded and transmitted using the binary number system. One of the primary reasons for this is that the intricate circuits within a computer carry out specific tasks using small electrical energy pulses. Electric current has only two states, representing two states of pulse transfer. Consequently, the physical aspect of information storage and transmission in computers corresponds to 1 for pulse passing and 0 for non-passing state, reflecting its mathematical representation.

The performance of computer devices, specifically microcircuits, relies on the presence or absence of pulses passing through their components, categorized as "closed" or "open," "yes" or "no." By assigning the number 0 to one case and the number 1 to the other, it becomes evident that computer systems operate within the binary number system.

If there were a third case of impulse transmission in the network, it would operate within the ternary counting system.

To store, process, and transfer information on a computer, we utilize symbols from the keyboard along with special control keys. It is crucial to encode these symbols, which serve as information organizers within the computer system. Therefore, the question arises of how many digits in the binary number system are required to encode all the characters (more than 200 in total) used on a computer.

As you may recall from your mathematics course, in the two-digit binary number system, 2^2=4 characters can be encoded using number combinations. Similarly, in the three-digit binary system, 2^3=8 characters can be encoded, and so on. In order to code more than 200 symbols, a combination of eight-digit numbers in the binary number system (2^8=256) is required.

Therefore, the size of one character is typically assumed to be 8 bits.

Thus, a computer can exclusively process data comprising zeros and ones. Each 0 or 1 is referred to as a bit. Eight bits together form a byte, such as 00101011, 00000000, 11111111, 10101010. A byte serves as the fundamental unit for representing information on a computer. Consequently, information is represented in the computer system as a multitude of zeros and ones, which are composed of individual bytes. This form of information representation is known as digital or binary representation. The manipulation of data in this binary format, following specific rules, is also known as binary arithmetic.

In fact, the number of character combinations that can be encoded using one byte is 256, which corresponds to 2 raised to the power of 8, resulting in 256. Therefore, the number of combinations is as follows:

00000000=0	00000001=1	00000010=2	00000011=3	00000100=4	00000101=5
00000110=6	00000111= 7	00001000=8	00001001=9	00001010=10	00001011=11
00001100=12	00001101=13	. . .	. . .	11111110=254	11111111=255

If there is a requirement to encode numbers greater than 255, two bytes are combined, and 16 bits are utilized. This allows for a total of 2 to the power of 16, resulting in 65,536 combinations.

Numbers larger than this are represented using four bytes or 32 bits. One bit is allocated to determine the sign of the number.

Representing real numbers and performing operations can be more complex for a computer. As mentioned earlier, one or two bytes are typically sufficient to represent each character (including letters, numbers, punctuation marks, arithmetic operation symbols, etc.) when processing text-based information. Each character is assigned its own code.

For measuring information related to various phenomena, $I = -\sum_{i=1}^{N} p_i \log_2 p_i$ K. Shannon's formula is used to measure information and is expressed in the following form: I = $-\log_2(N)$ where I represents the measure of information, N is the number of possible events, and ri denotes the probability of each event.

The fundamental formula for measuring uniform information was proposed by Ralph Hartley in 1928, providing a scientifically grounded approach to measuring information. It involves utilizing an alphabet comprising m letters to compile information. The total number of distinct types of information is given by N = mn. In this context, N represents the maximum number of information in various forms, m denotes the number of characters in the alphabet, and n represents the number of characters in the information.

For example, when using an alphabet consisting of two letters, "A" and "B," the number of data with a length of three characters is equal to 8. This is because with m = 2 and n = 3, we can generate eight different combinations of information: "AAA," "AAB," "ABA," "ABB," "BAA," "BAB," "BBA," and "BBB." There are no other possible options.

Hartley's formula is defined as I = $\log_2(N)$ = n * $\log_2(m)$, or N = 2^I. In this formula, I represents the amount of information in bits.

Coding can be classified as either even (uniform) or uneven (non-uniform). In flat coding, all characters are encoded using the same number or length of characters, while in uneven coding, different characters are encoded using varying numbers of characters. Decoding unevenly encoded information can be challenging.

To decode encoded information unambiguously from the beginning, it is necessary to ensure that no coded word precedes another coded word (referred to as the Fano condition). Similarly, to decode encoded information unambiguously from the end, it is crucial to avoid having one encoded word terminate with another encoded word (known as Fano's inverse condition).

If the words in the encoded information are not separated by a special character and the length of the encoding alphabet is consistent, flat encoding can be utilized. Decoding such information is straightforward, although the size of the encoded information is longer compared to non-uniform coding.

If the size of the coding alphabet is denoted as M and the code length is denoted as L, then it is possible to generate N = M^L different coded words. For flat codewords constructed using a binary alphabet (M = 2).

The number of different codewords of length L is equal to N = 2^L. Therefore, N characters from the alphabet can be encoded.

If textual information comprises letters, numbers, punctuation marks, and special characters, with each character occupying 8 bits of memory, it allows for the encoding of 2^8 = 256 characters. Alternatively, if the encoded word consists of 16 bits, it enables the encoding of 2^16 = 65,536 characters.

Let's explore some aspects related to flat coding (for information in the form of text, images, and sound). Although these topics may not receive extensive coverage in the standard high school computer science curriculum, we believe it is beneficial to introduce them as supplementary material for students.

Issues

Issue 1: The transmitter comprises 3 bulbs, each capable of operating in one of three states: "on," "off," or "blinking on and off." The goal is to determine the minimum number of bulbs required to transmit 18 different signals.

Solving: Given that the power of the coding alphabet is M = 3 (representing the 3 states of the bulbs) and the information power is N = 18 (the number of different signals), we can determine the minimum number of bulbs needed. We calculate I = log3(N) = $\log_3 18$, which falls between 2 and 3. Rounding up to the nearest larger number, we obtain a result of 3 bulbs required.

Answer 3.

Issue 2: The information comprises 4096 characters, and its size is specified as 1/512 Mbyte in flat encoding. The task is to determine the cardinality of the alphabet used in this information.

Solving: The strength of an alphabet refers to the number of characters it contains. To determine the strength of the alphabet used in the information, we need to convert the information size to bits.

Space is allocated for encoding a single badge. According to Hartley's formula, the cardinality of the alphabet is determined by N = 2^i = 2^4 = 16 bits.

Answer: 16

Issue 3: A black and white image has 8 levels of lighting. The image size is 10 x 15 cm, with a resolution of 300 dots per inch (1 inch = 2.5 cm). The question is how many kilobytes of memory the image occupies without compression.

Solution: Since N=8, each point requires i=log28=3 (8=23) bits of space.

Image size = 10*15 cm = 4*6 inches = 24 inches2

Since there are 300 points in 1 inch, it consists of inch2=3002 points=90000 points.

K=90000 dots*24 inch2=2160000 dots

I=K*i=2160000*3bit=6480000 bits = 810000 bytes = 810 Kbytes

Answer: 810

List of used Literature

1. Law of the Republic of Uzbekistan "On Information". - Bulletin of the Supreme Council of the Republic of Uzbekistan, 2004, No. 1-2, Article 10.

2. The Law of the Republic of Uzbekistan "On Keeping State Secrets". - Bulletin of the Supreme Council of the Republic of Uzbekistan, 1993, No. 5, Article 232.

3. Law of the Republic of Uzbekistan "On Electronic Digital Signature". - Bulletin of the Oliy Majlis of the Republic of Uzbekistan, 2004, No. 1-2, Article 12.

4. Law of the Republic of Uzbekistan "On Electronic Payments". - Collection of Laws of the Republic of Uzbekistan, 2005, No. 51, Article 373; 2009, No. 39, Article 423.

5. "Coding Theory Foundations" - Richard E. Blahut
6. "Information Coding" (Information Coding) - Jean-Gabriel Ganascia
7. "Information Coding: Practical Applications in Matlab" (Information Coding: Practical Applications in Matlab) - Gareth W. Peters and Yining Hu
8. "Fundamentals and Applications of Information Coding" - Andre J. G. Journault