Information

Bit, Binary, Information Perspective

Definition

- Information refers to the result of processing, analyzing, and interpreting data, which provides meaning, relevance, and usefulness to the recipient.
- It is data in a context.
- It has many forms including text, image, audio, video…

Historical Concept: Hard to Define

- Ancient civilization: languages and symbols.
- Print press (1450s): books.
- Industrial revolution (18^{th}-19^{th} centuries): telegraph and telephone
- Digital age (20^{th} century): computer and digital technology.
- Modern era (21^{st} century): information explosion and IoT.

Claude Shannon’s Information Theory

- Published in a seminal 1948 paper “A Mathematical Theory of Communication.”
- Two main contributions:
  - Information as *a Measure of Uncertainty Reduction*.
    - When we receive a message, it reduces our uncertainty about the source or content of that message.
    - The more unpredictable the message, the more information it contains.
  - Bit as the basic unit of information.

Entropy

- Entropy measures the uncertainty or unpredictability of a random variable or the amount of information required to describe the state of a system.
  - The more chaos/unpredictable, the higher the entropy.
  - A sure thing has 0 entropy: sun rises tomorrow morning.
- Shannon provided a clear and precise mathematical definition of entropy/information, resolving thousands of years of confusion surrounding the concept.

A Fair Coin Toss

- Two outcomes: heads (H) or tails (T).
- Equal Probability: P(H) = P(T) = 0.5.
- Entropy H(X) measures the average amount of information produced by a stochastic source of data.
- The entropy/uncertainty is 1 bit – if you know the result, you have 1 bit of information.

$$
\begin{aligned}
H(X) &= -\sum_{i=1}^{n} p_i log_2(p_i) \\
&= -(0.5 log_2(0.5) + 0.5 log_2(0.5)) \\
&= -(log_2(0.5)) = 1
\end{aligned}
$$

Less Uncertainty: Unfair Coin Toss

- If the coin is not fair but comes up heads or tails with probabilities P(H) = 0.7 and P(T) = 0.3, then there is less uncertainty.
- Every time it is tossed, the head is more likely to come up than the tail.
- The information is less than 1 bit:

```
H(X) = -0.7 * log0.7 - 0.3 * log0.3 = 0.8816
```

A Bit

- A bit is a binary unit that can take one of two values: 0 or 1
- A bit can represent any two distinct values: yes/no, true/false, on/off, stand/sit, day/night, high/low, big/small, cat/dog,…

- Two coins, four multiple-choice questions
  - Two bits

Binary is Universal

- Binary system: The binary system uses only two digits: 0 and 1. This simplicity makes it ideal for electronic representation and manipulation.
- Mathematical convenience: Bits are easily manipulated using Boolean algebra, making mathematical operations and data processing straightforward.
- Universal applicability: Bits can represent various types of information, such as numbers, text, images, and audio, making them a universal unit of information.

Boolean Algebra

- AND (Conjunction): Represented by the symbol ∧. In binary, AND is performed by multiplying the two bits. Example: $1 \wedge 1 = 1$, $1 \wedge 0 = 0$, $0 \wedge 1 = 0$, $0 \wedge 0 = 0$.
- OR (Disjunction): Represented by the symbol ∨. In binary, OR is performed by adding the two bits. Example: $1 \vee 1 = 1$, $1 \vee 0 = 1$, $0 \vee 1 = 1$, $0 \vee 0 = 0$.
- NOT (Negation): Represented by the symbol ¬. In binary, NOT is performed by flipping the bit. Example: $\neg 1 = 0$, $\neg 0 = 1$.
- XOR (Exclusive OR): Represented by the symbol ⊕. In binary, XOR is performed by adding the two bits **modulo 2**. Example: $1 \oplus 1 = 0$, $1 \oplus 0 = 1$, $0 \oplus 1 = 1$, $0 \oplus 0 = 0$.

Boolean Algebra is Fundamental

- All these Boolean operations can be easily implemented using transistor-based logical gates.
- Boolean operations can implement math operations like addition, subtraction, and division using binary numbers and transistor-based logical gates.
- How transistors do math (Youtube): https://www.youtube.com/watch?v=VBDoT8o4q00

More Bits

- One bit can only represent two different things/states.
- More bits are needed in complex cases.
- Two bits can represent four states.
  - 00
  - 01
  - 10
  - 11
- n bits can represent 2 ^ n states or different things

Byte

- A byte is a group of 8 bits.
- It can represent $2 ^ 8 = 256$ different things.
- It is the basic unit of measurement for data storage and transmission.

- In old days, a cell phone text message has a size limit of 160 characters (160B): this limited the original Tweet length to 140 characters.

Size You Should Know

- A kilobyte (KB) is equal to 1,024 bytes, about a thousand bytes. It is about the size of a small text file.
- A megabyte (MB) is equal to 1,048,576 bytes, or 1,024KB, about a million bytes. A frame of HD video is about 6MB.
- A gigabyte (GB) is equal to 1,073,741,824 bytes, or 1,024MB, about a billion bytes. A typical laptop memory size is between 4GB and 64GB. An average 4K movie have a size of 100GB.
- A terabyte (TB) is equal to 1,099,511,627,776 bytes, or 1,024GB, about a trillion bytes. A home computer hard drive may have 1TB to 10TB.
- A petabyte (PB) is equal to 1,125,899,906,842,624 bytes, or 1,024TB. In 2023, YouTube hosted 4.3PB data each day and Facebook produced 4PB data each day. Source: Edge Delta.

Bigger: Better to Know

- An exabyte (EB) is equal to 1,024PB.
- A zettabyte (ZB) is equal to 1,024EB. In 2023, the world created around 120ZB of data. Source: Edge Delta.
- A yottabyte (YB) is equal to 1,024ZB.
- …

Telcom vs Computer

- In computer, when people talk about the size of memory and file, the unit of measurement is based on byte such as B, KB, MB, GB etc.
  - 1 KB = 1024B, 1MB = 1024 KB
  - The factor is 1024
- In telecom and communication, the unit of measurement is bit, using lowercase b.
  - 1 kb = 1000b, 1mb = 1000kb, 1gb = 1000mb, 1tb = 1000gb
  - it uses the factor of 1000, not 1024.

Binary Numbers

- A decimal number can be represented by one or more binary digits, where each bit represents a power of 2.
- Each binary digit has a corresponding place value.
  - The rightmost bit that has a place value of $1 = 2^0$
  - The 2nd bit from the right has a place value of $2 = 2^1$
  - the 3rd bit has a place value of $4 = 2^2$
  - and so on and so forth
  - It is a tradition in computer science to count from 0, therefore the nth bit has a place value of $2 = 2^n$

2^4	2^3	2^2	2^1	2^0	
16	8	4	2	1	
0	0	0	0	0	0 0

Binary Number Examples

Do it Manually: 1010

1. Write down the binary number: 1010
2. Start from the right, get the place value:

   2^0 = 1, 2^1 = 2, 2^2 = 4, 2^3 = 8
3. Multiply each binary digit by the corresponding power of 2:

   1×8 + 0×4 + 1×2 + 0×1 = 10
4. The result is the decimal equivalent: 10

Decimal to Binary

- We divide the decimal number by 2 repeatedly, keeping track of the remainders.
- The remainders, read from bottom to top, form the binary equivalent of the decimal number.
- For example, to convert the decimal number 10 to binary:
  - `10 ÷ 2 = 5 (remainder 0)`
  - `5 ÷ 2 = 2 (remainder 1)`
  - `2 ÷ 2 = 1 (remainder 0)`
  - `1 ÷ 2 = 0 (remainder 1)`
- Reading the remainders from bottom to top, we get the binary equivalent: 1010.

Hexadecimal

- Hexadecimal is used in computers because it provides a more efficient way to represent binary code, which is the fundamental language of computers.
- It uses 16 distinct symbols to represent numbers: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F.
  - A-F represent the numbers 10-15

Conversion Between Hexadecimal and Binary

- To convert a hexadecimal number to a binary number, write it as a four-digit bits: 0 as 0000, 1 as 0001, 2 as 0010, ..., F as 1111.
  - The hexadecimal number 2F3 is 0010 1111 0011.
- To convert a binary number to a hexadecimal number
  - from the rightmost bit, group the binary digits into sets of four
  - pad the leftmost group with zeros in its left
  - then convert each set to the corresponding hexadecimal number.
  - For example, 1011110011 is 0010 1111 0011 that is 2F3 in hexadecimal.

Hexadecimal to Decimal

- Write down the hexadecimal number and identify its digits and their positions.
- Multiply each digit by 16 raised to the power of its position, starting from 0 at the rightmost digit.
- Sum all the results to get the decimal equivalent.

For example, to convert the hexadecimal number 2F3 to decimal:

- Identify the digits and their positions: 2 is in position 2, F in position 1, 3 in position 0.
- Multiply each digit by 16 raised to the power of its position: 2 * (16 ^2) = 512, F * (16 ^ 1) = 15 * 16 = 240, 3 * (16 ^ 0) = 3.
- Sum the result: 512 + 240 + 3 = 755

Decimal to Hexadecimal

- Divide the decimal number by 16 and record the quotient and the remainder.
- Repeat the division using the quotient from the previous step until the quotient is 0.
- Write down the remainders in reverse order. These remainders represent the hexadecimal digits.

For example, to convert the decimal number 755 to hexadecimal:

- Divide by 16: 755 / 16 = 47 remainder 3
- Divide the quotient by 16: 47 / 16 = 2 remainder 15
- Divide the quotient by 16: 2 / 16 = 0 remainder 2
- Write the remainders in reverse order: 2F3

Text in Binary (7/8 bits)

ASCII	Char	Hex	Bin	ASCII	Char	Hex	Bin
32	space	20	0010 0000	54	6	36	0011 0110
33	!	21	0010 0001	55	7	37	0011 0111
34	"	22	0010 0010	56	8	38	0011 1000
35	#	23	0010 0011	57	9	39	0011 1001
36	$	24	0010 0100	58	:	3A	0011 1010
37	%	25	0010 0101	59	;	3B	0011 1011
38	&	26	0010 0110	60	<	3C	0011 1100
39	'	27	0010 0110	61	=	3D	0011 1101
40	(	28	0010 1000	62	>	3E	0011 1110
41	)	29	0010 1001	63	?	3F	0011 1111
42	*	2A	0010 1010	64	@	40	0100 0000
43	+	2B	0010 1011	91	[	5B	0101 1011
44	,	2C	0010 1100	92	\	5C	0101 1100
45	-	2D	0010 1101	93	]	5D	0101 1101
46	.	2E	0010 1110	94	^	5E	0101 1110
47	/	2F	0010 1111	95	_	5F	0101 1111
48	0	30	0011 0000	96	`	60	0110 0000
49	1	31	0011 0001	123	{	7B	0111 1011
50	2	32	0011 0010	124	|	7C	0111 1100
51	3	33	0011 0011	125	}	7D	0111 1101
52	5	34	0011 0100	126	~	7E	0111 1110
53	5	35	0011 0101				

ASCII is not Enough

- With the advent of globalization and the internet, the need for a universal language of characters became increasingly important
- Unicode was first introduced in 1991 to create a standard that would allow computers to represent and exchange text in any language.
- The Unicode standard includes over 143,000 characters, supports over 150 languages and scripts.
- One of the key features of Unicode is its ability to represent a vast range of characters, including letters, digits, symbols, and even emojis.

Unicode

- Unicode is a system that enables computers to represent characters from any language using one to four bytes.
  - It starts with one byte for basic characters like English letters and common symbols.
  - As the complexity of the characters increases, more bytes are used: two bytes for extended characters found in languages like Latin, Greek, and Cyrillic;
  - Three bytes for characters from common languages such as Chinese, Arabic, and Hindi;
  - Four bytes for rare characters, emojis, and historic texts.

This flexible encoding allows Unicode to cover a vast array of characters from different languages and symbols, making it the standard for text representation in digital systems.

Unicode Examples

Latin: A (U+0041), ñ (U+00F1);

Greek: α (U+03B1), Ω (U+03A9);

Math: π (U+03C0), ∞ (U+221E);

Currency: $ (U+0024), € (U+20AC);

Emojis: 😊 (U+1F60A), 👍 (U+1F44D).

The "U+" notation represents the Unicode code point in hexadecimal form.

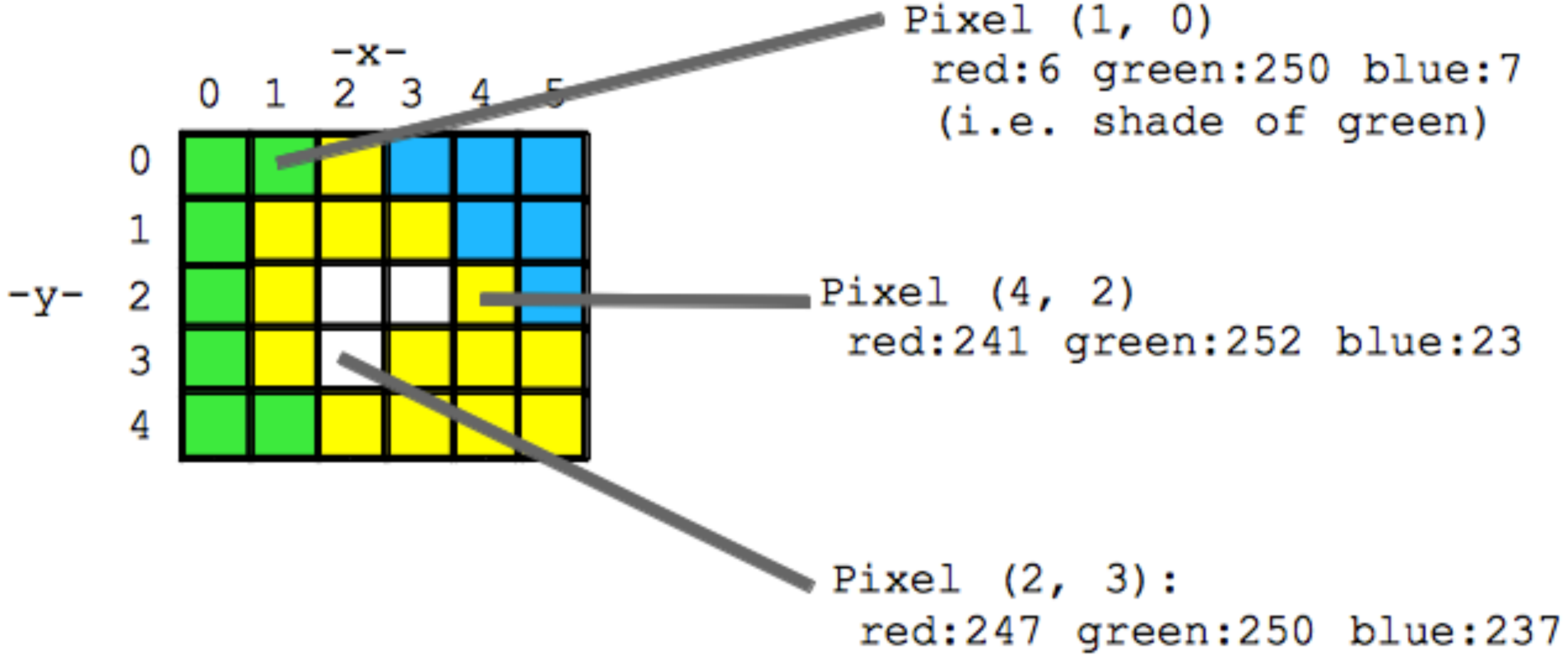
Screen Picture

- A picture in a computer screen is a collection of millions of these pixels, arranged in a grid to create a complete image.
- Resolutions:
  - 1080p (Full HD, FHD): 1920x1080 pixels, commonly used for gaming and streaming.
  - 1440p (2K, Quad HD, QHD): 2560x1440 pixels, versatile for various uses, including gaming and office work.
  - 4K (Ultra HD, UHD): 3840x2160 pixels, ideal for sharp image and text clarity, often used for professional applications.
  - 5K: 5120x2880 pixels, offers high pixel density and is commonly used with Mac computers for photo editing and other tasks requiring sharp images.

Pixel and RGB

- A screen pixel is the smallest unit of a digital image, representing a single point on a computer screen.
- It is made up of three-color components - red, green, and blue (RGB) - each with an intensity value ranging from 0 to 255.
- Each pixel's color and intensity contribute to the overall appearance of the picture, allowing for a wide range of colors, shades, and details to be displayed on the screen.
- As the pixels are arranged and lit up, they form a cohesive image, making up the digital pictures

Pixel and Color



An Example

- To display a red pixel of Red (R: 255, G: 0, B: 0)
- The binary Code is R: 11111111 (255), G: 00000000 (0), B: 00000000 (0).
- Combined Binary Code is 11111111 00000000 00000000, or FF0000 in hexadecimal.
- This combined binary code represents a single pixel with maximum red intensity and no green or blue intensity.
- Actual screen representation can be more complex, involving additional factors like alpha channels (transparency) and color depth.

CSS Color Codes

Color	CSS Color Name	Hex Code #RPGGBB	Decimal Code (R,G,B)
	Red	#FF0000	rgb (255,0,0)
	Orange	#FFA500	rgb (255,165,0)
	Yellow	#FFFF00	rgb (255,225,0)
	Green	#008000	rgb (0,128,0)
	Cyan	#00FFFF	rgb (0,255,225)
	Blue	#0000FF	rgb (0,0,225)
	Purple	#800080	rgb (128,0,128)
	Pink	#FFC0CB	rgb (255,192,203)
	Gray	#808080	rgb (128,128,128)
	Brown	#A52A2A	rgb (165,42,42)

Video

- Video is represented by breaking down each frame into tiny pixels.
- A 1080p video frame consists of 1920 x 1080 pixels
  - each represented by a 24-bit binary code (8 bits for red, 8 bits for green, and 8 bits for blue).
  - This results in a total of 2,073,600 binary digits per frame!
  - Digital video compression reduces the amount of data required to store or transmit video content.

Sound

- Sound is represented by sampling audio waves and assigning binary codes to each sample.
- A CD-quality audio signal
  - sampled 44,100 (44.1 kHZ) times per second
  - sample represented by a 16-bit binary code
  - two channel stereo, this results in a total of 44,100 * 16 * 2 = 1,411,200 (about 1.411 Mbps)
  - For a 74-minute CD, it is about 1411200 bit/second * 60 second/minute * 74 minute / 8 bit/byte = 740MB before compression
  - Algorithms like MP3, AAC, and FLAC compress the record to a size from 74MB to 370MB.

The Information Perspective

- Business: data processing
- Human: DNA sequences,
- Social life: communication and collaboration