Numeri in virgola mobile

Note Title 2026-02-18

"numeri di macchina"

"floating point"

Rappresentazione scientifica in base β

Teo: Fissata una base $\beta > 1$, ogni reale $\neq 0$ si scrive come

$$x = \pm \beta^p \left(\sum_{k=0}^{\infty} c_k \beta^{-k} \right)$$

$[764.8888\ldots \to -10^2 \cdot 7.64888\ldots]$

c_k cifre in $\{0, 1, \ldots \beta-1\}$

Per avere una rappresentazione unica, supponiamo $c_0 \neq 0$, e che i c_k non siano tutti uguali a $\beta-1$ da un certo punto in poi

$c_0 \neq 0$ per avere unicità: $10^2 \cdot 7.6488\ldots = 10^3 \cdot 0.76488\ldots$

$= 10^4 \cdot 0.076488\ldots$

rappresentazione normalizzata

c_k non tutti uguali a $\beta-1$ da un certo punto:

$0.4999\ldots = 0.5$

$-10^{2} \cdot 7.64888\ldots$

esponente

segno

mantissa

Definizione: i numeri di macchina normalizzati sono i numeri della forma

$c_k \in \{0, 1, \ldots \beta-1\}$

$$x = \pm \beta^p \sum_{k=0}^{t-1} c_k \beta^{-k} \qquad p \in \{m, m+1, \dots M\} \qquad c_0 \neq 0$$

L'insieme dei numeri fatti in questo modo si indica con

$$F(\beta, t, m, M)$$

Esempio: scriviamo tutti i numeri di $F(10, 3, -2, 3)$

Partiamo a scrivere i numeri positivi con esponente $p = -2$

$$x = 10^{-2} \cdot (c_0 10^0 + c_1 10^{-1} + c_2 10^{-2}) = 0.0 c_0 c_1 c_2$$

w

0.0100, 0.0101, 0.0102, ... 0.0999

Poi con esponente $p = -1$: $x = 10^{-1}(c_0 10^0 + c_1 10^{-1} + c_2 10^{-2})$:

0.100, 0.101, 0.102, ... 0.999

$p=0$: 1.00, 1.01, 1.02, ... 9.99

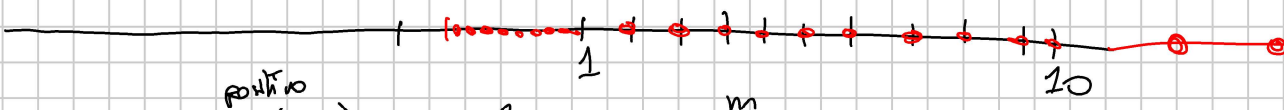
$p=1$: 10.0, 10.1, 10.2, ... 99.9

$p=2$: 100, 101, 102, ... 999

$p=3$: 1000, 1010, 1020, ... 9990 Ω

più tutti i loro negativi; questi sono tutti e soli gli elementi di $F(10, 3, -2, 3)$

- Numeri che non stanno in questo insieme saranno approssimati, ad es. $1001 \to 1000$
- I numeri non sono equispaziati:



- esiste un numero positivo Y più piccolo, $w = \beta^m \cdot 1.000..0$

e un numero più grande $\Omega = \beta^M \cdot (\beta-1)(\beta-1)\dots(\beta-1)$

t cifre uguali a $\beta - 1$

- per passare da un numero di macchina al suo successivo,

mi basta aggiungere 1 all'ultima cifra:

il successivo di

$$x = \beta^p \left(c_0 \beta^0 + c_1 \beta^{-1} + \dots + c_{t-1} \beta^{-(t-1)} \right)$$

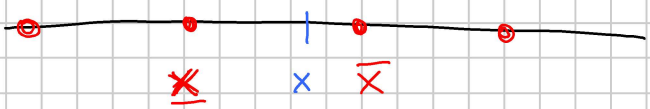
è $y = x + \beta^p \cdot \beta^{-(t-1)} = x + \beta^{p-t+1}$

Teo: dato un numero reale $x \in [-\Omega, -w] \cup [w, \Omega]$, esiste un numero di macchina $\tilde{x}$ tale che

$$\left| \frac{\tilde{x} - x}{x} \right| < \beta^{-(t-1)}.$$

errore relativo

Dim: possiamo assumere $x > 0$, l'altro caso è uguale, ma con dei meno davanti. Possiamo assumere che x non sia un numero di macchina.



Allora x sarà compreso tra due numeri di macchina successivi: $\underline{x} < x < \bar{x}$

$x = 10^{-6}$ $\tilde{x} = 10^{-2}$ errore: $\frac{10^{-2} - 10^{-6}}{10^{-6}} = \frac{10^{-6}(10^{4} - 1)}{10^{-6}} = 9999$ ($4-6$)

$x = 10^{-6}$ $\tilde{x} = 0$ errore: $\frac{0 - 10^{-6}}{10^{-6}} = -1$

Errore relativo = rapporto tra l'errore assoluto $\tilde{x} - x$ e la quantità che voglio misurare x.

Tra l'altro, riusciamo anche a scrivere $\underline{x}$ facilmente: è quello che si ottiene troncando le cifre di x

$$x = \beta^p \left(c_0 + c_1 \beta^{-1} + c_2 \beta^{-2} + c_3 \beta^{-3} + \dots \right)$$

$\underline{x}$ si ottiene troncando la somma dopo t addendi.

$\overline{x}$ è il num. di macchina successivo di $\underline{x}$,

cioè, $\underline{x} + \beta^{p+1-t}$

Sia $\tilde{x}$ preceduto $\underline{x}$ (troncamento), sia $\overline{x}$ (approssimatore per eccesso),

so che

$$|\tilde{x} - x| \leq \overline{x} - \underline{x} = \beta^{p+1-t}.$$

Devo stimare anche il denominatore della frazione:

$$x = \beta^p \left(c_0 \beta^0 + c_1 \beta^{-1} + c_2 \beta^{-2} + \dots \right) \geq \beta^p$$

c_0: almeno 1

$$\frac{|\tilde{x} - x|}{x} \leq \frac{\beta^{p+1-t}}{\beta^p} = \beta^{-(t-1)}. \quad \square$$

La costante $u = \beta^{-(t-1)}$ dipende solo dal sistema di numeri di macchina che sto usando, ed è detta precisione di macchina

Nel nostro esempio $\mathbb{F}(10, 3, -2, 3)$, $u = 10^{-2}$

$$\frac{\tilde{x} - x}{x} = \varepsilon \iff \tilde{x} = x(1 + \varepsilon)$$

---

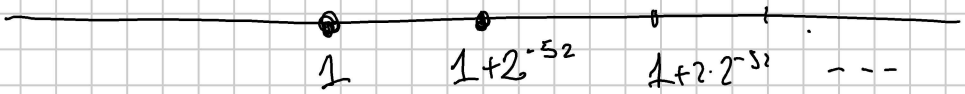
double / float64 / binary64 (IEEE)

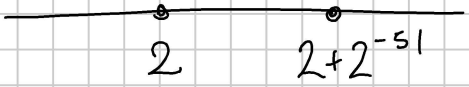
$\mathbb{F}(2, 53, -1022, 1023)$

β t m M

c_0 non si memorizza perché è sempre 1.

$$u = \beta^{-(t-1)} = 2^{-52} \approx 2.2 \cdot 10^{-16}$$





$$w \approx 10^{-308} \quad \Omega \approx 10^{308}$$

single / float / float32 : 32 bit, $u \approx 10^{-8}$

Dettaglio: oltre a questi numeri, vogliamo rappresentare:

- 0
- ∞ (inf), $-\infty$, -0 ad es. per $\frac{1}{\infty} = 0$.
- NaN, "not a number", $0/0$, $\infty - \infty$
- "numeri denormalizzati" $\in (-w, w)$.

⚠ Somme, prodotti, ecc. di numeri di macchina in generale non sono numeri di macchina → serve approssimare anche i risultati.

Si indica con un'operazione cerchiata, ad esempio

$a \oplus b$, $a \ominus b$, $a \odot b$ $a \oslash b$

quando effettuiamo un'operazione tra numeri di macchina e

poi approssimiamo il risultato in numeri di macchine.

Vediamo un po' di esempi in $\mathbb{F}(10, 3, -2, 3)$

$100 \oplus 0.01 = fl(100.01) = 100$

$100 \oplus 0.9 = fl(100.9) = 101$

$0.123 \oplus 3.45 = fl(3.573) = 3.57$

b)

$100 \oplus (0.4 \oplus 0.4) = 101$

$(100 \oplus 0.4) \oplus 0.4 = fl(100.4) \oplus 0.4 = 100 \oplus 0.4 = 100$

$5000 \oplus 5000 = \infty$ perché il risultato è $> \Omega$ (overflow)

$0.01 \otimes 0.01 = fl(0.0001) = 0$ (underflow)

perché $0.0001 < w$.

$(1 \oslash 3) \oplus (1 \oslash 3) \oplus (1 \oslash 3) = 0.333 \oplus 0.333 \oplus 0.333$

$= 0.999$

$a \oplus b = (a+b)(1+\varepsilon) \qquad |\varepsilon| \leq u = 2.2 \cdot 10^{-16}$

$0.1 \oplus 0.2 =$

$= \big((0.1)(1+\varepsilon_1) + (0.2)(1+\varepsilon_2)\big)(1+\varepsilon_3) \qquad |\varepsilon_1|, |\varepsilon_2|, |\varepsilon_3| \leq u$

$= 2.2 \cdot 10^{-16}$