Statistical and Mathematical Methods for Artificial Intelligence

Academic Year 2023 – 2024

Alma Mater Studiorum · University of Bologna

Contents

1 Finite numbers

1.1 Sources of error

Measure error Precision of the measurement instrument.

Arithmetic error Propagation of rounding errors in each step of an algorithm.

Truncation error Approximating an infinite procedure into a finite number of iterations.

Inherent error Caused by the finite representation of the data (floating-point).

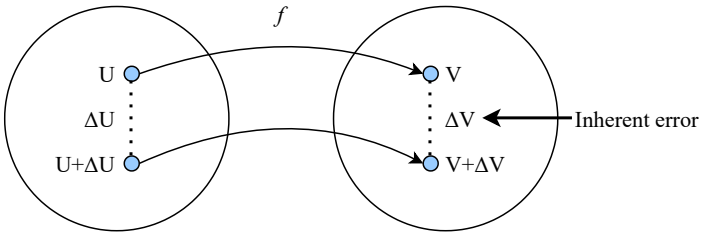

Figure 1: Inherent error visualization

1.2 Error measurement

Let x be a value and $\hat{x}$ its approximation. Then:

Absolute error

$$E_a = \hat{x} - x \tag{1}$$

Note that, out of context, the absolute error is meaningless.

Relative error

$$E_a = \frac{\hat{x} - x}{x} \tag{2}$$

1.3 Representation in base β

Let $\beta \in \mathbb{N}_{>1}$ be the base. Each $x \in \mathbb{R} \setminus \{0\}$ can be uniquely represented as:

$$x = \texttt{sign}(x) \cdot (d_1\beta^{-1} + d_2\beta^{-2} + \dots d_n\beta^{-n})\beta^p \tag{3}$$

where:

- $0 \leq d_i \leq \beta - 1$
- $d_1 \neq 0$
- starting from an index i, not all d_j $(j \geq i)$ are equal to $\beta - 1$

Equation (3) can be represented using the normalized scientific notation as:

$$x = \pm(0.d_1d_2\dots)\beta^p \tag{4}$$

where $0.d_1d_2\dots$ is the **mantissa** and β^p the **exponent**.

1.4 Floating-point

A floating-point system $\mathcal{F}(\beta, t, L, U)$ is defined by the parameters:

- β: base
- t: precision (number of digits in the mantissa)
- $[L, U]$: range of the exponent

Each $x \in \mathcal{F}(\beta, t, L, U)$ can be represented in its normalized form:

$$x = \pm(0.d_1 d_2 \dots d_t)\beta^p \quad L \leq p \leq U \tag{5}$$

Example 1.1. *In* $\mathcal{F}(10, 5, -3, 3)$, $x = 12.\bar{3}$ *is represented as:*

$$fl(x) = +0.12333 \cdot 10^2$$

1.4.1 Numbers distribution

Given a floating-point system $\mathcal{F}(\beta, t, L, U)$, the total amount of representable numbers is:

$$2(\beta - 1)\beta^{t-1}(U - L + 1) + 1$$

Representable numbers are more sparse towards the exponent upper bound and more dense towards the lower bound. It must be noted that there is an underflow area around 0.

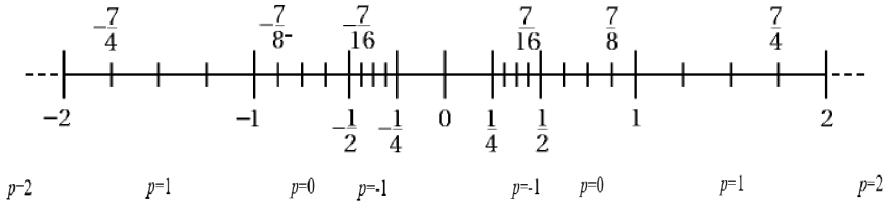

Figure 2: Floating-point numbers in $\mathcal{F}(2, 3, -1, 2)$

1.4.2 Numbers representation

Given a floating-point system $\mathcal{F}(\beta, t, L, U)$, the representation of $x \in \mathbb{R}$ can result in:

Exact representation if $p \in [L, U]$ and $d_i = 0$ for $i > t$.

Approximation if $p \in [L, U]$ but d_i may not be 0 for $i > t$. In this case, the representation is obtained by truncating or rounding the value.

Underflow if $p < L$. In this case, the values is approximated as 0.

Overflow if $p > U$. In this case, an exception is usually raised.

1.4.3 Machine precision

Machine precision $\varepsilon_{\text{mach}}$ determines the accuracy of a floating-point system. Depending on the approximation approach, machine precision can be computes as:

Truncation $\varepsilon_{\text{mach}} = \beta^{1-t}$

Rounding $\varepsilon_{\text{mach}} = \frac{1}{2}\beta^{1-t}$

Therefore, rounding results in more accurate representations.

$\varepsilon_{\text{mach}}$ is the smallest distance among the representable numbers (Figure 3).

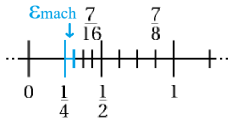

Figure 3: Visualization of $\varepsilon_{\text{mach}}$ in $\mathcal{F}(2, 3, -1, 2)$

In alternative, $\varepsilon_{\text{mach}}$ can be defined as the smallest representable number such that:

$$\texttt{fl}(1 + \varepsilon_{\text{mach}}) > 1.$$

1.4.4 IEEE standard

IEEE 754 defines two floating-point formats:

Single precision Stored in 32 bits. Represents the system $\mathcal{F}(2, 24, -128, 127)$.

1 (sign)	8 (exponent)	23 (mantissa)

Double precision Stored in 64 bits. Represents the system $\mathcal{F}(2, 53, -1024, 1023)$.

1 (sign)	11 (exponent)	52 (mantissa)

As the first digit of the mantissa is always 1, it does not need to be stored. Moreover, special configurations are reserved to represent `Inf` and `NaN`.

1.4.5 Floating-point arithmetic

Let:

- $+ : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be a real numbers operation.
- $\oplus : \mathcal{F} \times \mathcal{F} \to \mathcal{F}$ be the corresponding operation in a floating-point system.

To compute $x \oplus y$, a machine:

1. Calculates $x + y$ in a high precision register (still approximated, but more precise than the storing system)
2. Stores the result as $\texttt{fl}(x + y)$

A floating-point operation causes a small rounding error:

$$\left\| \frac{(x \oplus y) - (x + y)}{x + y} \right\| < \varepsilon_{\text{mach}} \tag{6}$$

Although, some operations may be subject to the **cancellation** problem which causes information loss.

Example 1.2. *Given* $x = 1$ *and* $y = 1 \cdot 10^{-16}$*, we want to compute* $x + y$ *in* $\mathcal{F}(10, 16, U, L)$*.*

$$\begin{aligned} z &= \mathit{fl}(x) + \mathit{fl}(y) \\ &= 0.1 \cdot 10^1 + 0.1 \cdot 10^{-15} \\ &= (0.1 + 0.\overbrace{0 \ldots 0}^{16 \text{ zeros}} 1) \cdot 10^1 \\ &= 0.1\overbrace{0 \ldots 0}^{15 \text{ zeros}} 1 \cdot 10^1 \end{aligned}$$

Then, we have that $\mathit{fl}(z) = 0.1\overbrace{0 \ldots 0}^{15 \text{ zeros}} \cdot 10^1 = 1 = x$*.*